NetScaler 13.1

Machine translated content

Disclaimer

Contents

NetScaler リリースノート

August 15, 2023

リリースノートには、特定のビルドでソフトウェアがどのように変更されたか、およびビルドに存在する既知の問題が記載されています。

リリースノートドキュメントには、次のセクションのすべてまたは一部が含まれています。

- 新機能: ビルドでリリースされた機能強化とその他の変更。
- 修正された問題: ビルドで修正される問題。
- 既知の問題: ビルドに存在する問題。
- 注意点: ビルドを使用する際に留意すべき重要な点です。
- 制限事項: ビルドに存在する制限事項。

> 注
>
> - 問題の説明の下にある［# XXXXXX］ラベルは、NetScaler チームが内部的に使用する追跡 ID です。
> - これらのリリースノートには、セキュリティ関連の修正は記載されていません。セキュリティ関連の修正と勧告のリストについては、セキュリティ速報を参照してください。

NetScaler 13.1-52.19 ビルドのリリースノート

April 15, 2024

このリリースノートでは、NetScaler リリース Build 13.1-52.19 の拡張機能や変更、修正された問題と既知の問題について説明します。

メモ

- このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティに関する修正とアドバイスの一覧については、Citrix セキュリティ情報を参照してください。

新機能

ビルド 13.1-52.19 で利用できる機能強化と変更点。

NetScaler Secure Web Gateway

- **URL** フィルタリング機能の **URL** 分類の廃止

  URL フィルタリング機能の URL 分類は、このリリースでは廃止されました。

  注: 廃止された機能はすぐには削除されません。NetScaler は、将来のリリースで削除されるまで、廃止された機能を引き続き保持します。

  [NSSWG-1370]

プラットフォーム

- 新しい **AWS** リージョンのサポート

  NetScaler はハイデラバードとジャカルタの AWS リージョンをサポートするようになりました。詳細については、AWS-VPXサポートマトリックスを参照してください。

  [NSPLAT-28073]

- **NetScaler VPX** での **VMware ESX 8.0** アップデート **2** のサポート

  NetScaler VPX は、VMware ESX 8.0 アップデート 2 (ビルド 22380479) をサポートするようになりました。詳細については、「サポートマトリックスと使用ガイドライン」を参照してください。

  [NSPLAT-27755]

- **NetScaler VPX** でのプライマリディスク容量の増加のサポート

  NetScaler VPX プライマリディスク容量は、一度に 20 GB から 1 TB まで動的に増やすことができるようになりました。ディスク容量を増やすには、それぞれのクラウドまたはハイパーバイザー UI でプライマリディスクを 1 GB 以上拡張します。

  [NSPLAT-27622]

- **AWS R7iz** インスタンスタイプのサポート

  AWS クラウド上の NetScaler VPX は、R7iz インスタンスタイプをサポートするようになりました。詳細については、AWS-VPXサポートマトリックスを参照してください。

  [NSPLAT-26632]

SSL

- 証明書バンドルの更新操作をサポート

  証明書バンドルで更新操作がサポートされるようになりました。証明書バンドルを直接更新できるようになりました。以前は、最初にバンドルをバインド解除して削除し、次に証明書バンドルを追加してバインドする必要がありました。

  [NSSSL-12613]

ユーザーインターフェイス

- **NetScaler** 次世代 **API (**テクニカルプレビュー**)**

  NetScaler 次世代 API は、NetScaler アプライアンスをシンプルで使いやすい方法でプログラム的に構成できるようにする高度で堅牢な`RESTful`API です。この API は、宣言型で望ましい状態の、アプリケーション中心のインターフェースに基づいています。

  この API は、従来の NetScaler 構成の低レベルの複雑さの多くを抽象化して簡素化することを目的としています。これらの API 機能により、ネットワークや NetScaler の専門家ではないアプリケーション開発者に適しています。

  次世代 API のコアコンセプトはアプリケーションです。同じアプリケーションに関連するすべての構成要素がグループ化されているため、変更をアトミックに簡単かつ安全に適用できます。

  現在、次世代 API では以下の機能セットがサポートされています:

  1. コンテンツスイッチング (HTTP、HTTP プロトコル)
  2. ロード・バランシング (HTTP、HTTP プロトコル)
  3. SSL オフロード
  4. HTTP リダイレクト、HTTP レスポンダー
  5. サーバーヘルスチェック

  [NSCONFIG-8040]

解決された問題

ビルド 13.1-52.19 で対処されている問題。

分析インフラストラクチャ

- 高度な Syslog ポリシーが Syslog Global にバインドされている場合、SSLVPN に関連する一部のメッセージが ns.log ファイルに表示されません:

  - SSLVPN ログイン
  - SSL VPN ログアウト

  [NSHELP-37051]

- 管理パーティションで構成された NetScaler で分析を有効にすると、NetScaler でメモリリークが発生することがあります。

  [NSHELP-36584]

- 次の条件がすべて満たされると、NetScaler がクラッシュする可能性があります:

  - NetScaler の内部サービスが SNMP からのカウンター更新要求を処理しており、構成のクリア処理が同時に進行中です。
  - CPU 使用率が増加します。
  - SNMP が開始したパーティションに関連するクエリは削除されます。

  [NSHELP-36400]

- NetScaler を 13.1 49.x ビルドにアップグレードすると、CPU 使用率が高くなることがあります。

  [NSHELP-36389、NSCXLCM-3856]

認証、承認、監査

- 高可用性セットアップで、SAML IdP アクションと SAML IdP プロファイルがメタデータ URL で構成されている場合、次のいずれかの条件が満たされると NetScaler がクラッシュすることがあります:

  - HA フェイルオーバー中。
  - SAML アクションと SAML IdP プロファイルが削除されたとき。
  - 上記の両方。

  [NSHELP-36573]

- SAML SSO プロファイルまたは SAML IdP プロファイルがカスタムポリシー式を使用している場合、シングルサインオンが失敗することがあります。

  [NSHELP-36412、NSCXLCM-626]

- LDAP サーバーの応答が遅れると、セルフサービスパスワードリセットが設定された NetScaler がクラッシュすることがあります。

  [NSHELP-35586]

- 高可用性セットアップでは、ユーザーはメモリリークが原因でセカンダリ NetScaler インスタンスを頻繁に再起動します。

  [NSHELP-28659]

ボット管理

- 一部の良いボットはドロップされ、一部の悪いボットは許可されるため、誤検知が発生します。この問題は、自動更新プロセス中に AWS 署名ファイルが更新され、一部のボット署名に対して誤ったアクションが行われた場合に発生します。

  [NSBOT-1121]

負荷分散

- 負荷分散仮想サーバーは、MQTT プロトコルバージョン 5 タイプのサービスがバインドされたときに接続をリセットし、QoS レベル 1 のクエリを送信します。

  [NSHELP-37134]

- 管理パーティションのセットアップでは、グローバルサーバー負荷分散 (GSLB) 構成が存在するかどうかに関係なく、自動同期処理による CPU 使用率が高くなることがあります。この問題は通常、多数のパーティションが設定されている場合に発生します。

  [NSHELP-37015、NSCXLCM-3207]

- ポートタイプ ANY で構成された負荷分散仮想サーバーは、4 台以上の IPv6 サーバーを持つサービスグループがバインドされている場合、要求に応答しない場合があります。

  [NSHELP-36498]

- まれに、NetScaler が誤った IP アドレスで GSLB ドメインクエリに応答することがあります。この問題は、DNS ベースの自動スケール GSLB サービスグループが GSLB 仮想サーバーにバインドされている場合に発生します。

  [NSHELP-36393]

- IAM 権限がないために自己修復デーモンがクラッシュし、バックエンドで 10 分ごとに再起動されると、CPU 使用率が高くなることがあります。

  [NSHELP-36220]

- 高可用性セットアップでは、DNS ベースの GSLB サービスグループが GSLB 仮想サーバーにバインドされているときに、NetScaler が GSLB ドメインクエリに誤った IP アドレスで応答することがあります。この問題は、フェールオーバー後に発生します。

  [NSHELP-35633]

- ドメインにバインドされている間に IP アドレスが削除されると、NetScaler がクラッシュすることがあります。この問題は、TTL が低く、IP アドレスをバインドしてから削除するまでの間に競合状態が発生することが原因で発生します。

  [NSHELP-34546]

- レート制限レコードの取得とレコードのエージングプロセスの間のタイミングの問題により、NetScaler がクラッシュする可能性があります。

  [NSHELP-33349]

- いくつかの内部仮想サーバーでモニタープローブが失敗すると、NetScaler がクラッシュすることがあります。

  [NSHELP-30985]

その他

- HA モードでは、ICA パケットの処理中にセカンダリ NetScaler インスタンスがクラッシュします。

  [NSHELP-37256、NSCXLCM-3410、NSCXLCM-3573]

- アップグレード後、AppFlow が有効になっていると、HA モードの NetScaler がクラッシュします。

  [NSHELP-37142、NSCXLCM-3613]

- NetScaler Gateway でポリシーベースのルーティングが構成されている場合、ユーザーは UDP 経由で ICA セッションを起動できません。

  [NSHELP-36448]

- アップグレード後、ユーザーは NetScaler Gateway 構成ファイルを StoreFront からダウンロードできなくなります。

  [NSHELP-36322]

NetScaler Gateway

- NetScaler Gateway ポータルからアクセスするウェブページの URL には、HTTP が含まれています。今回の修正により、NetScaler Gateway からアクセスされるウェブページの URL の HTTP が HTTPS に置き換えられました。

  [NSHELP-36832]

- NetScaler では、冗長な文字列があるため、Syslog メッセージが複数行で表示されます。

  [NSHELP-36775]

- NetScaler GUI（[構成] > [システム] > [認証]）では、NetScaler Gateway ライセンスを持つユーザーの [詳細ポリシー] セクションが表示されません。

  [NSHELP-36762]

- アップグレード後、NetScaler Gateway のプロキシ設定がフル VPN モードで動作しなくなります。

  [NSHELP-35853]

- コンテンツスイッチアクションに認証仮想サーバーまたは VPN 仮想サーバーがターゲット仮想サーバーの 1 つとして含まれている場合、一部のイントラネットアプリケーションにアクセスできません。

  [NSHELP-35582]

- アップグレード後、[構成] > [ライセンスされていない機能を表示] オプションを選択すると、NetScaler GUI のナビゲーションペインに [セキュリティ] セクションの機能が表示されません。

  [CGOP-25521]

NetScaler SDX アプライアンス

- 管理サービスのアップグレード後、外部 LDAP サーバーへの認証が失敗することがあります。

  [NSHELP-36455, NSHELP-36842]

NetScaler Web App Firewall

- Web App Firewall プロファイルにバイパスリストと拒否リストを追加した後に show コマンドを実行すると、表示される設定の詳細が正しくありません。

  [NSHELP-37079]

- Web App Firewall は、自動更新プロセス中にカスタム署名ファイルから削除された署名 ID のトラフィックを引き続きブロックする可能性があります。この問題は、バージョンがデフォルトの署名ファイルバージョンと一致している場合に、自動更新プロセスが、更新されたカスタム署名オブジェクトファイルをパケットエンジンにプッシュできないために発生します。

  [NSHELP-37008]

- Web App Firewall がコマンドインジェクション保護チェックの実行に予想よりも時間がかかると、NetScaler がクラッシュすることがあります。

  [NSHELP-36343、NSCXLCM-2189]

ネットワーク

- NSIP（NSIP6）の IPv6 アドレスへの非 TCP トラフィックは、多数のポートリークが原因で失敗する可能性があります。

  [NSHELP-36764]

- 大規模な NAT（LSN）セットアップでは、LSN キューの処理における内部問題が原因で NetScaler がクラッシュすることがあります。

  [NSHELP-36605]

- RNAT 構成で`tcpproxy`パラメーターを有効にすると、NetScaler は IP セットとネットプロファイルに指定されていない SNIP をバックエンド通信の送信元 IP アドレスとして使用することがあります。

  [NSHELP-36562, NSCXLCM-2223]

- 大規模 NAT（LSN）セットアップでは、LSN フロントエンド接続とバックエンド接続で内部数の不一致があると、NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-36391]

- セッション参照カウントの処理に一貫性がないため、NetScaler がクラッシュすることがあります。

  [NSHELP-36379]

- 大規模 NAT（LSN）セットアップでは、ビットエンコーディング操作が正しくないために NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-36124]

- 高可用性セットアップでは、tag-all パラメータが有効になっている場合、HA パケットは VLAN タグ付きで送信されます。設定された HA デッドインターバル時間（デフォルト値は 3 秒）内に同期された構成がセカンダリノードに適用されない場合、プライマリノードとセカンダリノードの両方がプライマリステータスを主張しようとするスプリットブレインシナリオが発生する可能性があります。

  [NSHELP-35628]

- ネットワーク NIC インターフェイスイベントの遅延により、NetScaler の再起動後に BGP がデフォルトルートをインストールできませんでした。

  [NSHELP-30262]

プラットフォーム

- インターフェイス構成が無効になっていると、アプライアンスの再起動後に Tx Laser がオフにならないことがあります。

  [NSPLAT-28484、NSHELP-35272]

- NetScaler BLX の管理対象ホストを有効にしても、ホスト上で構成されているすべての IP アドレスが NetScaler BLX に追加されるわけではありません。

  [NSLINUX-1044]

- NetScaler BLX をアップグレードしても、nitro-python パッケージが更新されない場合があります。その結果、アップグレード中に次のエラーが表示されることがあります：

  「tar: /var/netscaler/nitro/ns_nitro-python_artesa_xx_xx.tar: アーカイブに見つかりません」

  [NSLINUX-1008]

- 専用モードでは、NetScaler BLX は HTTP と HTTPS のデフォルトの管理ポートとして、ポート 80 と 443 の代わりにポート 9080 と 9443 を使用します。

  [NSLINUX-977]

- VirtIO NIC を搭載した KVM プラットフォーム上の NetScaler VPX は、シングル PE モードで動作しているときに送信ストールを報告することがあります。

  [NSHELP-37106]

- Intel Fortville NIC を搭載した NetScaler SDX では、リンク状態が UP に変わってから数秒以内に物理 NIC の速度が変化しても、VPX インスタンスのインターフェイス速度は更新されません。

  [NSHELP-36361]

- Intel Fortville NIC を搭載した NetScaler SDX では、VPX インスタンスに Fortville インターフェイスが追加されるたびに、VPX インスタンスの管理 CPU 使用率が 1% 増加します。

  [NSHELP-35445, NSCXLCM-768]

- SDX プラットフォーム上の NetScaler VPX は、トラフィックが多いときにクラッシュする可能性があります。

  [NSHELP-34775]

ポリシー

- ns.conf ファイルにデフォルトの組み込みクラシック CMP バインディングが存在する場合、NetScaler のアップグレード中にエラーが発生することがあります。

  [NSPOLICY-5562]

SSL

- **NetScaler** から最後のパケットを再送信するように **DTLS** タイムアウトを構成する **DTLS** 展開では、**NetScaler** から最後のパケットを再送信するように初期タイムアウト値を構成できるようになりました。1 秒から 3 秒まで選択でき、デフォルト値は 3 秒に設定されています。

  以前は 3 秒がハードコーディングされていましたが、RFC では 1 秒が推奨されていました。

  [NSSSL-8868]

- SDX FIPS 14000 上の VPX インスタンスは、250K を超える SSL コンテキスト（セッション）で構成されている場合、メモリ破損の可能性があるためクラッシュする可能性があります。

  [NSHELP-36503、NSSVM-6019、NSCXLCM-1759、NSCXLCM-3465]

System

- サービスの SI_TOT_ResponseBytes カウンタ値は、仮想サーバー上の実際の応答サイズと比較してはるかに大きくなります。この状態は、NetScaler とサーバー間のネットワークでのパケットの順序変更またはパケット損失が原因で発生します。

  [NSHELP-36743]

- NetScaler は、すでに解放されたメモリセグメントを解放しようとするとクラッシュすることがあります。この問題は、アクションが RESET に設定されたレスポンダーポリシーにバインドされた TCP 負荷分散仮想サーバーでプロキシプロトコルデータを処理しているときに発生します。

[NSHELP-36729、NSCXLCM-2733、NSHELP-37102]

- クラスターデプロイメントで`start nstrace -nodes <nodeId>`コマンドを実行すると、すべてのノードでネットワークトレースが実行されます。ただし、予想される動作は、指定されたノードとクラスターコーディネーターでのみネットワークトレースをキャプチャすることです。

  [NSHELP-36489]

- HTTP/2 のウィンドウサイズが NetScaler からの gRPC 応答メッセージよりも小さい場合、クライアント（VIP）に送信される gRPC リクエストは失敗します。

  [NSHELP-36335]

- クライアント側では、NetScaler Console で分析（AppFlow のクライアント側測定）が有効になっていると、HTML ページが読み込まれないことがあります。

  [NSHELP-36318]

- NetScaler は、次の条件をすべて満たす SYN パケットをドロップします:

  - 古いシーケンス番号が付いている
  - タイムスタンプオプションが有効になっています
  - TIME_WAIT 状態にある既存の接続の 4 つのタプル (送信元 IP、送信元ポート、宛先 IP、および宛先ポート) と一致します。

  SYN パケットはドロップされるため、別の送信元ポートを使用して接続を再確立するように求められます。

  [NSHELP-35867]

- `unset nstcpprofile` コマンドを使用して TCP プロファイルパラメーターの設定を解除すると、NetScaler がコアをダンプすることがあります。

  [NSBASE-18724]

ユーザーインターフェイス

- NetScaler GUI を使用して IPv4 CIDR 形式のデータセットを作成することはできません。

  [NSHELP-36659]

- TACACS サーバでユーザアカウントパスワードの有効期限が切れると、同じユーザアカウントを使用してログインできないことがあります。

  すべてのユーザーアカウントのパスワードが、有効期限が切れる前またはそれ以前に更新されていることを確認してください。

  [NSHELP-36453、NSCXLCM-2312]

- NetScaler GUI は、GUI を使用して Splunk にデータをエクスポートする構成中に正しく応答しません。

  [NSHELP-36334]

- GUI を使用して NetScaler をアップグレードすると、GUI または SSH を使用してシステムアップグレードページにアクセスできなくなります。

  [NSHELP-35785]

- NetScaler GUI では、DNS レコードの表示に通常よりも時間がかかり、CPU 使用率が高くなることもあります。この問題は、DNS プロキシにキャッシュされたレコードの数が多い場合に発生します。

  [NSHELP-34788]

- 多数 (数千) の SSL 証明書を使用して構成された高可用性セットアップでは、構成の同期に通常より時間がかかる場合があります。その結果、同期状態が長時間続くことがあります。

  [NSHELP-32959、NSCXLCM-1752、NSCXLCM-1989、NSHELP-35003]

既知の問題

リリース 13.1-52.19 に存在する問題。

分析インフラストラクチャ

- NetScaler ADM を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに動作しません。

  回避策：管理ポッドを再起動します。

  [NSANINFRA-1504]

認証、承認、監査

- NetScaler は次のシナリオでクラッシュします:

  - メタデータ URL を使用して設定された SAML アクションが変更された場合。
  - ログアウト URL とメタデータ URL で構成された SAML アクションが保存され、NetScaler が再起動された場合。このクラッシュは繰り返し発生します。

  [NSAUTH-14541]

- SAML SP セットアップでは、負荷分散仮想サーバーで認証仮想サーバー名 (`authnVsName`) パラメーターが設定されている場合、SAML 認証が失敗します。

  [NSAUTH-14566]

- iOS 向け Citrix SSO のアップグレード後、認証のために受信するプッシュ通知に音が鳴らないことがあります。

  [NSHELP-27525]

- 管理者は、認証情報が無効であることが原因で発生した認証エラーのカスタムロギングを実行できません。この問題は、NetScaler Responder ポリシーがログインエラーのエラーを検出できないために発生します。

  [NSAUTH-11151]

- ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
  `show adfsproxyprofile <profile name>`

  回避策：クラスター内のアクティブなプライマリの NetScaler に接続し、`show adfsproxyprofile <profile name>`コマンドを実行します。プロキシプロファイルの状態が表示されます。

  [NSAUTH-5916]

- 次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

  - [LDAP 到達可能性をテスト] オプションが開きます。
  - 無効なログイン認証情報が入力され、送信されます。
  - 有効なログイン認証情報が入力され、送信されます。

  回避策：「LDAP 到達可能性テスト」オプションを閉じて開きます。

  [NSAUTH-2147]

負荷分散

- 高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

  [NSLB-7679]

- クラスタセットアップでは、コンテンツスイッチングポリシーがバインドされていると、コンテンツスイッチ仮想サーバーに誤った情報が一覧表示されることがあります。この問題は、ポリシールール式の長さが 255 文字を超える場合に発生します。

  [NSHELP-37118]

- 次の条件が満たされると、NetScaler がクラッシュする可能性があります：

  - 負荷分散または GSLB 仮想サーバーは、64 を超えるアクティブなサービスまたはサービスグループメンバーにバインドされています。
  - 静的近接負荷分散方式が設定されています。

  [NSHELP-37111]

- NetScaler は要求を受信すると、詳細を返信する代わりにクエリを返します。この問題は、次の条件が満たされた場合に発生することがあります：

  - ADNS サービスと DNS 負荷分散仮想サーバーは同じ NetScaler 上に構成されています。

- DNS クエリが仮想サーバーに送信され、仮想サーバーは否定的な応答を受け取ります。ただし、キャッシュ期間中は、同じクエリが ADNS サービスに送信されます。

[NSHELP-35878]

- NetScaler GUI を使用して次の手順を実行すると、実行構成に余分な CNAME レコードが表示されます:

  1. DNS レコードタイプ CNAME の GSLB 仮想サーバーを作成します。
  2. DNS レコードタイプ CNAME を設定します。
  3. 構成を保存します

  この問題は表面的なもので、機能には影響しません。

  [NSHELP-29217]

- サービスグループの`entityofs`トラップの ServiceGroupName 形式は次のとおりです。
  `<service(group)name>?<ip/DBS>?<port>`

  トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (“?”) は区切り文字として使用されます。NetScaler は、疑問符 (” ?”)。このフォーマットは、NetScaler ADM GUI でも同じように表示されます。これは予想される動作です。

  [NSHELP-28080]

NetScaler Gateway

- NetScaler UI（構成 > システム）では、アップグレード後に NetScaler Gateway ライセンスを追加すると、プロファイルセクションが表示されなくなります。

  [NSHELP-36496]

- 次の条件をすべて満たすと、NetScaler がクラッシュします:

  - VPN 仮想サーバーは IPv6 アドレスで構成されます。
  - IPv6 仮想サーバーでは、IPv4 IIP アドレスのみ (IPv6 IIP アドレスなし) が設定されます。

  [NSHELP-35559]

- スキーマをブラウズしているときに、「未定義のプロパティ「タイプ」を読み取れません」というエラーメッセージが表示されることがあります。

  [NSHELP-21897]

- LogStream トランスポートタイプが Insight 用に構成されている場合、クライアント IP とサーバー IP は HDX Insight SkipFlow レコードで反転されます。

  [CGOP-25905]

- Windows OS オプションは、NetScaler GUI の事前認証ポリシーと認証アクションの「エクスプレッションエディタ」ドロップダウンリストに表示されません。ただし、GUI または CLI を使用して以前の NetScaler ビルドで Windows OS スキャンをすでに構成している場合は、アップグレードしても機能に影響はありません。必要に応じて CLI を使用して変更できます。

  回避方法:

  設定には CLI コマンドを使用します。

  - nFactor 認証で高度な EPA アクションを設定するには、次のコマンドを使用します。
    add authentication epaAction adv_win_scan -csecexpr “sys.client_expr(“sys_0_WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]”)”
  - 従来の事前認証アクションを設定するには、次のコマンドを使用します。
    add aaa preauthenticationaction win_scan_action ALLOW
    add aaa preauthenticationpolicy win_scan_policy “CLIENT.SYSTEM(‘WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]’) EXISTS”win_scan_action

  [CGOP-22966]

- Windows ログオン機能の前に常時接続 VPN を使用するには、NetScaler Gateway を 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

  [CGOP-19355]

- NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

  [CGOP-11830]

- Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

  [CGOP-7269]

NetScaler Secure Web Gateway

- URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

  [NSHELP-22409]

ネットワーク

- FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続

に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

[NSNET-5233]

- 「show rnat」コマンドを実行すると、NetScaler 上で構成されている RNAT エンティティの不完全なリストが表示されることがあります。

  [NSHELP-36253]

- NetScaler アプライアンスは、コールドリスタート後に「ColdStart」SNMP トラップメッセージを生成しない場合があります。

  [NSHELP-27917]

- GUI または CLI に、同時セッションがある NetScaler 上で構成されているすべての SNMP マネージャーが表示されないことがある

  [NSHELP-25952]

- NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

  [NSHELP-21082]

プラットフォーム

- Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。「rm cloudprofile」コマンドを使用してプロファイルを削除します。

  [NSPLAT-4520]

- NetScaler CPX でテクニカルサポートバンドルを生成すると、一連のコアダンプファイル（仮想テレタイプシェルコアダンプ）が/_cpx/crash/_ ディレクトリに生成されます。これらのコアダンプファイルは機能的には重要ではないため、無視しても問題ありません。

  [NSLINUX-1016]

- DPDK を搭載した NetScaler BLX アプライアンスの Intel `X710 10G (i40e)`インターフェイスでは、次のインターフェイス操作はサポートされていません:

  - 無効化
  - 有効化
  - リセット

  [NSLINUX-64]

- DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

- NetScaler BLX アプライアンスには、少数の「巨大ページ」が割り当てられています。たとえば、1G です。
- NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

この問題は、エラーメッセージとして「/var/log/ns.log」に記録されます。

- 「BLX-DPDK: DPDK メモリプールを PE-x 用に初期化できませんでした」

注:x はワーカープロセス数以下の数です。

回避策:「巨大ページ」を多数割り当ててから、アプライアンスを再起動します。

[NSLINUX-42]

- DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

  [NSLINUX-17]

- DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

  [NSLINUX-14]

- Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

  「次のパッケージの依存関係は満たされていません:blx-core-libs: i386: preDepends: libc6: i386 (>= 2.19) しかしインストールできません」

  回避策: NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します:

  - dpkg –add-architecture i386
  - apt-get update
  - apt-get install libc6:i386

  [NSLINUX-5]

- AWS クラウドプロファイルの Autoscale グループの負荷分散サービスグループを更新すると、エラーが表示されることがあります。ただし、構成は期待どおりに正しく更新されます。

  [NSHELP-37030]

- 特定のシナリオでは、NetScaler BLX が ns.log ファイル内のシステムログをキャプチャしないことがあります。

  [NSHELP-36913]

ポリシー

- 処理データのサイズが設定されたデフォルトの TCP バッファーサイズを超えると、接続がハングアップすることがあります。

  回避策:TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

  [NSPOLICY-1267]

SSL

- NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

  回避方法:

  1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
  2. 構成を保存します。

  [NSSSL-9572]

- 認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

  [NSSSL-6478]

- 同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

  [NSSSL-6213]

- HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
  エラー:CRL 更新が無効です

  [NSSSL-6106]

- セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

  [NSSSL-4427]

- SSL プロファイル内の SSL プロトコルまたは暗号を変更しようとすると、「警告:SSL vserver/Service で使用可能な暗号が設定されていません」という誤った警告メッセージが表示されます。

  [NSSSL-4001]

- 期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

  [NSSSL-3184, NSSSL-1379, NSSSL-1394]

System

- 次の条件が満たされると、TCP 接続の RTT が高くなります:

  - 最大輻輳ウィンドウ（> 4 MB）が高く設定されている
  - TCP NILE アルゴリズムは有効になっています

  NetScaler アプライアンスが NILE アルゴリズムを使用して輻輳制御を行うには、条件がスロースタートのしきい値と最大輻輳ウィンドウを合わせた値を超える必要があります

  そのため、設定された最大輻輳時間帯に達するまで、NetScaler はデータを受け付け続け、最終的には RTT が高くなります。

  [NSHELP-31548、NSCXLCM-159]

ユーザーインターフェイス

- NetScaler GUI（[システム] **> [FIPS** システム情報] ページ）では、次のフィールドが空白になっています:

  - コントロールプレーンバージョン
  - データプレーンバージョン
  - グラフィックモジュールバージョン

  [NSUI-19559]

- NetScaler GUI では、「ダッシュボード」タブの下にある「ヘルプ」リンクが壊れています。

  [NSUI-14752]

- CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

  回避策：NetScaler GUI または CLI を使用して IPSec プロファイル、IP トンネル、および PBR ルールを追加して CloudBridge Connector を構成します。

  [NSUI-13024]

- GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

  [NSUI-6838]

- NetScaler GUI で、［システム］->［診断］->［アップグレード前/アップグレード後］->［差分レポートのエクスポート］オプションが期待どおりに機能しない。

  [NSHELP-37188]

- レート制限機能が有効になっているプロファイルの **Rate Limit Condition** フィールドを編集すると、次のエラーが表示される場合があります:
  名前付きのエンティティは既にバインドされています。バインドリソースはありません

  [NSHELP-36747、NSCXLCM-3290]

- NetScaler GUI を使用すると、データセットが宛先 IP パラメーターに設定されているデータセットベースの ACL を変更すると、次のエラーで失敗します:

  ```
  Argument pre-requisite missing [operator, destIP]
  ```

  [NSHELP-36666]

- GUI では、認証ポリシーページ（「認証仮想サーバー」>「認証ポリシー」>「ポリシーバインディング」）には、ページ分割の問題により 25 のポリシーしか表示されません。

  [NSHELP-36577]

- NetScaler GUI の［セキュリティ］>［SSL 転送プロキシ］>［SSL インターセプトポリシー］から SSL インターセプションポリシーを追加することはできません。

  [NSHELP-36166]

- 大規模な構成では、batch CLI コマンドを使用して設定を適用すると遅延が発生することがあります。

  [NSCONFIG-8607]

- 次の条件が満たされると、ユーザーはダウングレードされた NetScaler アプライアンスにログインできないことがあります:

  1. 次の手順のいずれかを実行します。
     - 現在のビルドにアップグレードしたら、システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更して、設定を保存します。
     - 現在のビルドで新しい NetScaler VPX、BLX、または CPX インスタンスをプロビジョニングします。
  2. アプライアンスを以下のいずれかのビルドにダウングレードします。
     - 13.1-4.x
     - 13.0-82.x またはそれ以前
     - 12.1-62.x またはそれ以前

  ダウングレード後に影響を受けるユーザーのリストを表示するには、コマンドプロンプトで次のように入力します。
  ```
  query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
  ```
  。回避策: 影響を受けるユーザーのパスワードをリセットします。詳細については、「[ルート管理者 (nsroot) のパスワードをリセットする方法](https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html」を参照してください。

注: 以前にアップグレードしたビルドをダウングレードする場合は、ダウングレード時に以前のビルドのバックアップされた設定ファイル (ns.conf) を使用してこの問題を回避してください。

[NSCONFIG-8068]

- 次の条件が満たされると、ユーザーはダウングレードされた NetScaler アプライアンスにログインできないことがあります:

  1. 次の手順のいずれかを実行します。
     - 現在のビルドにアップグレードしたら、システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更して、設定を保存します。
     - 現在のビルドで新しい VPX、BLX、または CPX インスタンスをプロビジョニングします。
  2. アプライアンスを以下のいずれかのビルドにダウングレードします。
     - 13.0-47.x またはそれ以前
     - 12.1-56.x またはそれ以前
     - 11.1-64.x またはそれ以前

  ダウングレード後に影響を受けるユーザーのリストを表示するには、コマンドプロンプトで次のように入力します。

  ```
  query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
  ```

  回避策: 影響を受けるユーザーのパスワードをリセットします。詳細については、「[ルート管理者 (nsroot) のパスワードをリセットする方法](https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html」を参照してください。

  注: 以前にアップグレードしたビルドをダウングレードする場合は、ダウングレード時に以前のビルドのバックアップされた設定ファイル (ns.conf) を使用してこの問題を回避してください。

  [NSCONFIG-3188]

NetScaler 13.1-51.15 ビルドのリリースノート

April 15, 2024

このリリースノートでは、NetScaler リリース Build 13.1-51.15 の拡張機能や変更、修正された問題、既知の問題について説明します。

メモ

- このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティに関する修正とアドバイスの一覧については、Citrix セキュリティ情報を参照してください。

- ビルド 13.1-51.15 以降のビルドは、https://support.citrix.com/article/CTX584986で説明されているセキュリティの脆弱性に対処します。
- ビルド 13.1-51.15 がビルド 13.1-51.14 に取って代わります。

新機能

ビルド 13.1-51.15 で利用できる機能強化と変更点。

NetScaler SDX アプライアンス

- **NetScaler SDX** でのインスタンスライセンスチェックアウトの制限の解除

  プールライセンスを使用する NetScaler SDX では、最小数のインスタンスライセンスをチェックアウトするという制限がなくなりました。つまり、少なくとも 1 つのインスタンスライセンスをチェックアウトできます。以前は、チェックアウトできるインスタンスライセンスの最小数はプラットフォームによって異なりました。

  詳しくは、「NetScaler プール容量」の表 1 を参照してください。

  [NSSVM-5881]

プラットフォーム

- **FreeBSD 11.4** と **FreeBSD 8.4** でタイマーの動作を同期させてください

  FreeBSD 11.4 のカーネルクロックとタイマーイベントは、NetScaler VPX プラットフォームでの安定性を実現するために、FreeBSD 8.4 と同様になっています。

  [NSPLAT-26973]

System

- 許可されていない管理 **UI URL** リクエストを追跡するための新しいカウンター

  カウンター (http_err_admin_ui_requests_drop) が追加され、ブロックされた許可されていない管理 UI URL リクエストの数を追跡できるようになりました。

  [NSBASE-18523]

解決された問題

ビルド 13.1-51.15 で対処されている問題。

分析インフラストラクチャ

- /netscaler/nsconmsg コマンドを実行すると、NetScaler は管理パーティション上の仮想サーバーの不完全なデバッグ情報を表示します。

  [NSHELP-36185]

- デフォルトのバインドタイプ SYSTEM_GLOBAL を使用して、ドメインベースのサービス（Syslog アクション用）の SYSLOG グローバルエンティティに SYSLOG ポリシーをバインドしようとすると、次のエラーメッセージが表示されます。「NO-SUCH RESOURCE」

  この問題は、以前に APPFW_GLOBAL のグローバルバインドタイプを使用して SYSLOG グローバルエンティティから SYSLOG ポリシーをバインド解除したことがあり、その結果、サーバーエンティティが内部的に削除された場合に発生します。

  [NSHELP-35668]

認証、承認、監査

- GUI で、nFactor フローにログインスキーマを追加し、クラスター IP アドレスが NetScaler で構成されている場合、「リソースはすでに存在します」というエラーが表示されます。

  [NSHELP-36180]

- 場合によっては、OAuth IdP が SP 証明書の古いコピーをキャッシュすると、OAuth SP として構成された NetScaler がクラッシュすることがあります。

  [NSHELP-36150]

- OAuth IdP として構成された NetScaler は、依存当事者への応答時に誤ったコンテンツタイプヘッダーを送信することがあります。

  [NSHELP-35918]

- SAML サービスプロバイダーとして構成された NetScaler は、SAML IdP サーバーに送信された要求への応答を期待するとクラッシュすることがあります。

  [NSHELP-35771]

負荷分散

- 自動同期オプションが有効になっている場合、GSLB サービスのパブリックポートパラメータはセカンダリ GSLB サイトで更新されません。この問題は、自動同期プロセスが set gslb service-PublicPort’ コマンドをサポートしていないセカンダリサイトで実行した場合に発生します。

  [NSHELP-36823]

- 存在しないネットプロファイルを評価ルールで構成された負荷分散モニターにバインドしようとした後に、ネットプロファイルを負荷分散モニターにバインドすると、NetScaler がクラッシュすることがあります。

  [NSHELP-36626]

- DNS SOA レコードが GSLB ドメインと同じ名前で構成されていて、NetScaler が DNS ネームサーバーリゾルバーとしても構成されている場合、NetScaler は A (IPv4) 以外のタイプの GSLB ドメインクエリに対して NXDOMAIN で正しく応答しないことがあります。

  [NSHELP-36451]

- ADNS 展開では、DNS サーバーが意図しないユーザーに DNS ビューの IP アドレスで応答することがあります。この問題は、DNS レスポンダーポリシーにドロップアクションを設定した場合に発生することがあります。

  [NSHELP-35928]

- まれに、TCP タイプの仮想サーバーに永続性を構成すると、NetScaler がクラッシュすることがあります。

  [NSHELP-35360]

- ユーザがドメインベースのサーバをサービスグループにバインドおよびバインド解除すると、モニタプローブが失敗します

  [NSHELP-29330]

その他

- NetScaler BLX を DPDK サポートで構成すると、NetScaler BLX は一部のファームウェアバージョンの NIC で DPDK 互換の NIC ポートを検出できないことがあります。その結果、NIC ポートは非 DPDK ポートとして NetScaler BLX に追加されます。

  [NSHELP-36290]

- UDP オーディオの問題が原因で、NetScaler がクラッシュすることがあります。

  [NSHELP-36188, NSCXLCM-2303]

- アップグレード後、ICA セッションデータのクリーンアップに失敗すると、NetScaler がクラッシュします。

  [NSHELP-36169]

- 仮想サーバーで Web App Firewall 機能が有効になっていて、gRPC スキーマが Web App Firewall プロファイルにバインドされていない場合、NetScaler は gRPC 要求をブロックすることがあります。

  [NSHELP-36131]

- プロトコル拡張を使用すると、NetScaler でメモリリークが発生します。

  [NSEXT-472]

NetScaler Gateway

- NetScaler Gateway GUI では、VPN 仮想サーバーにバインドされている ICA ポリシーが、［NetScaler ゲートウェイ］>［ポリシー］>［NetScaler Gateway ICA ポリシーおよびプロファイル］>［ICA ポリシー］セクションに表示されません。

  [NSHELP-36324]

- 次の条件が満たされている場合、「トラフィック管理」>「SSL」>「SSL ファイル」オプションが GUI に表示されません:

  - NetScaler Gateway ライセンスが使用されます。
  - ソフトウェアは NetScaler リリース 13.0 ビルド 9.1.x にアップグレードされます。

  [NSHELP-36186]

- VPN 仮想サーバーがデフォルトの TCP プロファイルを使用している場合、フル VPN モードのサービスに TCP プロファイルをバインドしても有効になりません。

  [NSHELP-36132]

- アップグレード後、VPN に接続すると、SNIP アドレスを介して HTTPS を使用して NetScaler UI にアクセスできなくなります。

  [NSHELP-35747]

- NetScaler Gateway 経由で VPN に接続している場合、アップグレード後に NetScaler GUI に HTTP 経由でアクセスできなくなることがあります。

  [NSHELP-35015]

- NetScaler Gateway GUI を使用して DTLS 仮想サーバーを構成する場合、DTLS 仮想サーバーでデフォルトの SSL プロファイルが有効になっていると、SSL 設定は削除されます。

  [NSHELP-34723]

- バッファオーバーフロー状態のため、NetScaler で EPA スキャンが断続的に失敗します。

  [NSHELP-34039]

NetScaler SDX アプライアンス

- 管理サービス UI で VPX インスタンスを編集すると、ゲートウェイ **IPv6** アドレスが管理サービス **UI** に表示されないため、管理サービスはインベントリから値を削除します。

  [NSSVM-5865]

- NetScaler SDX ダッシュボードでは、スループットデータとグラフは表示されません。

  [NSHELP-36586]

- 次の手順を実行すると、予想されるエラーメッセージが管理サービス UI に表示されません:
  - 管理チャネルで **Allow L2 Mode** オプションを有効にして VPX インスタンスを構成し、**VLAN** タグパラメータの値を入力します。
  - 同じ管理チャネルで **Allow L2 Mode** オプションを有効にしてさらに多くの VPX インスタンスを構成し、以前に構成した VPX インスタンスと同じ値を **VLAN** タグに入力します。

  [NSHELP-36321]
- NetScaler SDX バージョン 13.1 をビルド 42.47 からビルド 49.13 にアップグレードすると、管理サービスから XenServer への SCP/SSH 接続のタイムアウトが原因で失敗することがあります。管理サービス UI に次のエラーメッセージが表示されます:

  「Platform でアップグレードしようとしましたが、新しいバージョンが出ませんでした。」

  [NSHELP-36311, NSCXLCM-2086, NSCXLCM-2877]
- 管理サービス **UI** のダウンロードページから **SDX-MIB-SMIV2.mib** ファイルをダウンロードしても、**LAST-UPDATED** フィールドは更新されません。

  [NSHELP-36174]

NetScaler Web App Firewall

- クラスター展開では、Web App Firewall 機能を有効にすると NetScaler がクラッシュすることがあります。

  [NSHELP-36362]

ネットワーク

- 認証の失敗がユーザ名またはコミュニティ名と照合されると、SNMPv2 認証トラップロギングメッセージにユーザ名またはコミュニティ名の無効な値が表示されます。

  [NSHELP-36213, NSCXLCM-1848]
- 次の条件が満たされると、TCP 応答のルーティング中に NetScaler がクラッシュすることがあります:
  - トラフィックドメインと管理パーティションの両方が設定されています。
  - TCP パケットは、トラフィックドメインと管理パーティションの両方から同じポートと IP アドレスに同時に送信されます。

  [NSHELP-36202]
- IP over IP 構成の HA セットアップでは、HA フェイルオーバー中にセカンダリノードがクラッシュする可能性があります。

  [NSHELP-36060]

- アップグレード後、SSL VPN で構成された NetScaler は、HDX インサイトの処理中にエラーが発生したためにクラッシュします。

  [NSHELP-35895, NSCXLCM-1519, NSCXLCM-2321]

- MIB ファイルと実際の varbind が一致しない場合、SNMP トラップモニタリングがトラップの読み取りに失敗することがあります。

  [NSHELP-35629]

- デフォルト以外の HTTPS ポート上の内部 SSL サービスの場合、デフォルトの SSL 証明書のバインドを変更してアプライアンスを再起動しても、デフォルトの証明書は引き続き内部サービスにバインドされます。

  [NSHELP-24034]

プラットフォーム

- AWS にデプロイされた NetScaler VPX インスタンスに接続されているセカンダリディスクが、AWS Xen インスタンスタイプで検出されません。

  [NSHELP-36504]

ポリシー

- NITRO API を使用して、バージョン 13.0 以前のソフトウェアを実行している NetScaler をバージョン 13.1 以降にアップグレードすると、失敗することがあります。

  [NSHELP-36685]

- クラスターセットアップで、NOOP タイプのレスポンダーアクションを削除しようとすると、次のエラーが発生することがあります:

  「デフォルトアクションは削除できません」

  [NSHELP-36194]

SSL

- Intel SSL チップを搭載した NetScaler MPX および NetScaler SDX プラットフォームでは、初回再起動後またはウォームリブート後にカードが起動しないことがあります。

  [NSHELP-36506, NSCXLCM-2699, NSCXLCM-2796]

- TLS 1.3 クライアントが証明書の送信を拒否すると、認証仮想サーバーが次の（フォールバック）認証ポリシーを適用できない場合があります。

  [NSHELP-36479]

- TLS 1.3 構成でジャンボフレームを使用すると、NetScaler がクラッシュすることがあります。

  [NSHELP-36153]

- 次の条件を満たす場合、ログインを繰り返し試みた後、NetScaler がクラッシュすることがあります:

  - クライアント認証は有効になっています。
  - クライアントはルート証明書を使用して認証を試みます。

  [NSHELP-36094, NSCXLCM-2866]

- NetScaler は、クライアントが以前のハンドシェイクメッセージを再送信すると、エラーメッセージを返します。

  [NSHELP-34608, NSCXLCM-348]

System

- CONNECT HTTP リクエストメソッドを使用する HTTP/2 ストリームが終了すると、HTTP/2 を使用する Web ページが完全に読み込まれないことがあります。

  [NSHELP-36407、NSBASE-17449]

- ウィンドウスケールの値が 12 より大きい TCP プロファイルにアタッチされた負荷分散サービスは、HTTP タイプのモニターにバインドされたときに停止することがあります。

  [NSHELP-35947]

- start コマンドと stop コマンドは設定コマンドとして扱われるため、設定が変更されていなくても設定を保存するように求められます。

  [NSHELP-28413]

- AWS クラウド上の NetScaler BLX では、/var/tmp ディレクトリに保存されている nstrace ソケット（nstrace.sock）ファイルが、systemd クリーンアッププロセスによって削除されることがあります。

  [NSBASE-18548]

ユーザーインターフェイス

- 署名オブジェクトを編集または追加すると、「システムファイルの更新リクエストを完了できません」というエラーが表示されることがあります。この問題は、次の条件のいずれかが満たされた場合に発生します:

  - 署名オブジェクトはサイズが大きいです。
  - コールがタイムアウトになりました。

  [NSHELP-36751]

- 権限が制限されているユーザーは、より高い権限を持つユーザーが作成したレポートを削除できます。

  [NSHELP-36042]

- 外部認証ユーザーに対して大量のアカウンティング要求を受信すると、クラスタファイルの同期が正しく機能しない場合があります。この間、より多くのディスク容量が消費される可能性があります。

  [NSHELP-35985]

- **LB** アクションの設定ページの「値」フィールドにスペースが含まれている場合、GUI にはエラーメッセージは表示されません。スペースを含む値フィールドを編集すると、GUI はそのスペースをカンマに置き換え、構成が無効になります。

  [NSHELP-35532]

- NetScaler クラスターセットアップでは、`show nstrace` コマンドの CLI 出力がクラスターノードの NSIP アドレスに正しく表示されません。

  [NSHELP-33712]

- NetScaler GUI では、コマンドインターフェイスと比較して、表示されるキャッシュされたオブジェクトの数が少なくなります。

  [NSHELP-24337]

- NetScaler は、メモリ消費量が多いシナリオでクラッシュする可能性があります。

  [NSCONFIG-7972、NSCONFIG-7716、NSCXLCM-3380]

既知の問題

リリース 13.1-51.15 に存在する問題。

分析インフラストラクチャ

- NetScaler ADM を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに動作しません。

  回避策：管理ポッドを再起動します。

  [NSANINFRA-1504]

認証、承認、監査

- iOS 向け Citrix SSO のアップグレード後、認証のために受信するプッシュ通知に音が鳴らないことがあります。

  [NSHELP-27525]

- 管理者は、認証情報が無効であることが原因で発生した認証エラーのカスタムロギングを実行できません。この問題は、NetScaler Responder ポリシーがログインエラーのエラーを検出できないために発生します。

  [NSAUTH-11151]

- ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
  `show adfsproxyprofile <profile name>`

  回避策：クラスター内のアクティブなプライマリの NetScaler に接続し、`show adfsproxyprofile <profile name>`コマンドを実行します。プロキシプロファイルの状態が表示されます。

  [NSAUTH-5916]

- 次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

  - [LDAP 到達可能性をテスト] オプションが開きます。
  - 無効なログイン認証情報が入力され、送信されます。
  - 有効なログイン認証情報が入力され、送信されます。

  回避策：「LDAP 到達可能性テスト」オプションを閉じて開きます。

  [NSAUTH-2147]

負荷分散

- 高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

  [NSLB-7679]

- NetScaler は要求を受信すると、詳細を返信する代わりにクエリを返します。この問題は、次の条件が満たされた場合に発生することがあります：

  - ADNS サービスと DNS 負荷分散仮想サーバーは同じ NetScaler 上に構成されています。
  - DNS クエリが仮想サーバーに送信され、仮想サーバーは否定的な応答を受け取ります。ただし、キャッシュ期間中は、同じクエリが ADNS サービスに送信されます。

  [NSHELP-35878]

- NetScaler GUI を使用して次の手順を実行すると、実行構成に余分な CNAME レコードが表示されます：

  1. DNS レコードタイプ CNAME の GSLB 仮想サーバーを作成します。
  2. DNS レコードタイプ CNAME を設定します。
  3. 構成を保存します

  この問題は表面的なもので、機能には影響しません。

  [NSHELP-29217]

- サービスグループの`entityofs`トラップの ServiceGroupName 形式は次のとおりです。
  `<service(group)name>?<ip/DBS>?<port>`

  トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (“?”) は区切り文字として使用されます。NetScaler は、疑問符 (” ?”)。このフォーマットは、NetScaler ADM GUI でも同じように表示されます。これは予想される動作です。

  [NSHELP-28080]

その他

- DPDK を搭載した NetScaler BLX アプライアンスの Intel `X710 10G (i40e)`インターフェイスでは、次のインターフェイス操作はサポートされていません:
  - 無効化
  - 有効化
  - リセット

  [NSLINUX-64]

- DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:
  - NetScaler BLX アプライアンスには、少数の「巨大ページ」が割り当てられています。たとえば、1G です。
  - NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

  この問題は、エラーメッセージとして「/var/log/ns.log」に記録されます。

  - 「BLX-DPDK: DPDK メモリプールを PE-x 用に初期化できませんでした」

  注:x はワーカープロセス数以下の数です。

  回避策:「巨大ページ」を多数割り当ててから、アプライアンスを再起動します。

  [NSLINUX-42]

- DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

  [NSLINUX-17]

- DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

  [NSLINUX-14]

- Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

  「次のパッケージの依存関係は満たされていません:blx-core-libs: i386: preDepends: libc6: i386 (>= 2.19) しかしインストールできません」

  回避策：NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します:

  - dpkg –add-architecture i386
  - apt-get update
  - apt-get install libc6:i386

  [NSLINUX-5]

NetScaler Gateway

- NetScaler GUI（[構成] > [システム] > [認証]）では、NetScaler Gateway ライセンスを持つユーザーの [詳細ポリシー] セクションが表示されません。

  [NSHELP-36762]

- 次の条件をすべて満たすと、NetScaler がクラッシュします:

  - VPN 仮想サーバーは IPv6 アドレスで構成されます。
  - IPv6 仮想サーバーでは、IPv4 IIP アドレスのみ (IPv6 IIP アドレスなし) が設定されます。

  [NSHELP-35559]

- スキーマをブラウズしているときに、「未定義のプロパティ「タイプ」を読み取れません」というエラーメッセージが表示されることがあります。

  [NSHELP-21897]

- Windows OS オプションは、NetScaler GUI の事前認証ポリシーと認証アクションの「エクスプレッションエディタ」ドロップダウンリストに表示されません。ただし、GUI または CLI を使用して以前の NetScaler ビルドで Windows OS スキャンをすでに構成している場合は、アップグレードしても機能に影響はありません。必要に応じて CLI を使用して変更できます。

  回避方法:

  設定には CLI コマンドを使用します。

  - nFactor 認証で高度な EPA アクションを設定するには、次のコマンドを使用します。
    add authentication epaAction adv_win_scan -csecexpr “sys.client_expr(“sys_0_WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]”)”

- 従来の事前認証アクションを設定するには、次のコマンドを使用します。
  add aaa preauthenticationaction win_scan_action ALLOW
  add aaa preauthenticationpolicy win_scan_policy “CLIENT.SYSTEM(‘WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]’) EXISTS” win_scan_action

[CGOP-22966]

- Windows ログオン機能の前に常時接続 VPN を使用するには、NetScaler Gateway を 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

  [CGOP-19355]

- NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

  [CGOP-11830]

- Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

  [CGOP-7269]

NetScaler Secure Web Gateway

- URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

  [NSHELP-22409]

ネットワーク

- NetScaler CPX でテクニカルサポートバンドルを生成すると、一連のコアダンプファイル（仮想テレタイプシェルコアダンプ）が/_cpx/crash/_ ディレクトリに生成されます。これらのコアダンプファイルは機能的には重要ではないため、無視しても問題ありません。

  [NSNET-29141]

- FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

  [NSNET-5233]

- NSIP（NSIP6）の IPv6 アドレスへの非 TCP トラフィックは、多数のポートリークが原因で失敗する可能性があります。

  [NSHELP-36764]

- 大規模 NAT（LSN）セットアップでは、LSN フロントエンド接続とバックエンド接続で内部数の不一致があると、NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-36391]

- 大規模 NAT（LSN）セットアップでは、ビットエンコーディング操作が正しくないために NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-36124]

- NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

  [NSHELP-21082]

プラットフォーム

- Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。「rm cloudprofile」コマンドを使用してプロファイルを削除します。

  [NSPLAT-4520]

- Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
  回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノード上で設定する必要があります。

  [NSPLAT-4451]

ポリシー

- ns.conf ファイルにデフォルトの組み込みクラシック CMP バインディングが存在する場合、NetScaler のアップグレード中にエラーが発生することがあります。

  [NSPOLICY-5562]

- 処理データのサイズが設定されたデフォルトの TCP バッファーサイズを超えると、接続がハングアップすることがあります。

  回避策:TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

  [NSPOLICY-1267]

SSL

- NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

  回避方法:

  1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
  2. 構成を保存します。

  [NSSSL-9572]

- 認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

  [NSSSL-6478]

- 同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

  [NSSSL-6213]

- HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
  エラー:CRL 更新が無効です

  [NSSSL-6106]

- セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

  [NSSSL-4427]

- SSL プロファイル内の SSL プロトコルまたは暗号を変更しようとすると、「警告:SSL vserver/Service で使用可能な暗号が設定されていません」という誤った警告メッセージが表示されます。

  [NSSSL-4001]

- 期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

  [NSSSL-3184, NSSSL-1379, NSSSL-1394]

System

- 次の条件が満たされると、TCP 接続の RTT が高くなります:

  - 最大輻輳ウィンドウ（> 4 MB）が高く設定されている

- TCP NILE アルゴリズムは有効になっています

NetScaler アプライアンスが NILE アルゴリズムを使用して輻輳制御を行うには、条件がスロースタートのしきい値と最大輻輳ウィンドウを合わせた値を超える必要があります

そのため、設定された最大輻輳時間帯に達するまで、NetScaler はデータを受け付け続け、最終的には RTT が高くなります。

[NSHELP-31548、NSCXLCM-159]

- mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

  [NSBASE-18295]

- `unset nstcpprofile` コマンドを使用して TCP プロファイルパラメーターの設定を解除すると、NetScaler がコアをダンプすることがあります。

  回避策：代わりに、目的のパラメーター値を指定した`set nstcpprofile`コマンドを使用してください。

  [NSBASE-18724]

- LogStream トランスポートタイプが Insight 用に構成されている場合、クライアント IP とサーバー IP は HDX Insight SkipFlow レコードで反転されます。

  [NSBASE-8506]

ユーザーインターフェイス

- NetScaler GUI では、「ダッシュボード」タブの下にある「ヘルプ」リンクが壊れています。

  [NSUI-14752]

- CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

  回避策：NetScaler GUI または CLI を使用して IPSec プロファイル、IP トンネル、および PBR ルールを追加して CloudBridge Connector を構成します。

  [NSUI-13024]

- GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

  [NSUI-6838]

- レート制限機能が有効になっているプロファイルの **Rate Limit Condition** フィールドを編集すると、次のエラーが表示される場合があります：
  名前付きのエンティティは既にバインドされています。バインドリソースはありません

  [NSHELP-36747]

- NetScaler GUI の［セキュリティ］>［SSL 転送プロキシ］>［SSL インターセプトポリシー］から SSL インターセプションポリシーを追加することはできません。

  [NSHELP-36166]

- GUI を使用して NetScaler をアップグレードすると、GUI または SSH を使用してシステムアップグレードページにアクセスできなくなります。

  [NSHELP-35785]

- 次の条件が満たされると、ユーザーはダウングレードされた NetScaler アプライアンスにログインできないことがあります:

  1. 次の手順のいずれかを実行します。
     - 現在のビルドにアップグレードしたら、システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更して、設定を保存します。
     - 現在のビルドで新しい NetScaler VPX、BLX、または CPX インスタンスをプロビジョニングします。
  2. アプライアンスを以下のいずれかのビルドにダウングレードします。
     - 13.1-4.x
     - 13.0-82.x またはそれ以前
     - 12.1-62.x またはそれ以前

  ダウングレード後に影響を受けるユーザーのリストを表示するには、コマンドプロンプトで次のように入力します。
  `query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]`
  。回避策: 影響を受けるユーザーのパスワードをリセットします。詳細については、「[ルート管理者 (nsroot) のパスワードをリセットする方法](https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html」を参照してください。
  注: 以前にアップグレードしたビルドをダウングレードする場合は、ダウングレード時に以前のビルドのバックアップされた設定ファイル (ns.conf) を使用してこの問題を回避してください。

  [NSCONFIG-8068]

- 次の条件が満たされると、ユーザーはダウングレードされた NetScaler アプライアンスにログインできないことがあります:

  1. 次の手順のいずれかを実行します。
     - 現在のビルドにアップグレードしたら、システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更して、設定を保存します。
     - 現在のビルドで新しい VPX、BLX、または CPX インスタンスをプロビジョニングします。
  2. アプライアンスを以下のいずれかのビルドにダウングレードします。
     - 13.0-47.x またはそれ以前

- 12.1-56.x またはそれ以前
- 11.1-64.x またはそれ以前

ダウングレード後に影響を受けるユーザーのリストを表示するには、コマンドプロンプトで次のように入力します。

```
query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
```

回避策: 影響を受けるユーザーのパスワードをリセットします。詳細については、「[ルート管理者 (nsroot) のパスワードをリセットする方法](https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html」を参照してください。

注: 以前にアップグレードしたビルドをダウングレードする場合は、ダウングレード時に以前のビルドのバックアップされた設定ファイル (ns.conf) を使用してこの問題を回避してください。

[NSCONFIG-3188]

NetScaler 13.1-50.23 ビルドのリリースノート

February 15, 2024

このリリースノートドキュメントでは、NetScaler リリースビルド 13.1-50.23 に存在する機能強化と変更、修正された問題と既知の問題について説明します。

メモ

- このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティに関する修正とアドバイスの一覧については、Citrix セキュリティ情報を参照してください。

新機能

ビルド 13.1-50.23 で利用できる機能強化と変更。

その他

- テクニカルサポートバンドルをアップロードするための認証トークン

  テクニカルサポートバンドルを NetScaler から Citrix テクニカルサポートサーバーに直接アップロードする場合、認証トークンが必要になりました。以前は、テクニカルサポートバンドルをアップロードするには

Citrix のユーザー名とパスワードが必要でした。詳細については、「アプライアンスの問題を解決するためのテクニカルサポートバンドルを生成する方法」を参照してください。

[NSTOOLS-3019]

プラットフォーム

- レート制限メトリック用の新しい **SNMP MIB** のサポート

  NetScaler の 1 秒あたりのパケット数とビット/秒を取得するための新しい SNMP MIB を追加しました。この MIB は、NetScaler の現在のレート制限メトリックを理解するのに役立ちます。詳しくは、「NetScaler 13.1 SNMPOID リファレンス」を参照してください。

  [NSPLAT-26679]

- **10G/25G/40G NIC** を搭載した **NetScaler SDX** のアップデートされた **NIC** ドライバーとファームウェアのサポート

  シングルバンドルイメージ (SBI) バージョン 14.1-8.x 以降または 13.1-50.x 以降にアップグレードすると、次のプラットフォームで 10G/25G/40G NIC ドライバとファームウェアが自動的にバージョン 8.70 にアップグレードされます。NIC ファームウェアバージョン 8.70 は CVE-2020-8690、CVE-2020-8691、CVE-2020-8692、および CVE-2020-8693 を修正します。

  - SDX 8900
  - SDX 14000-40G
  - SDX 15000
  - SDX 15000-50G
  - SDX 25000-40G
  - SDX 16000
  - SDX 9100
  - SDX 26000
  - SDX 26000-50S

  [NSPLAT-23460]

解決された問題

ビルド 13.1-50.23 で対処されている問題。

分析インフラストラクチャ

- 複数の AppFlow ポリシーが NetScaler にグローバルにバインドされている場合、1 つのポリシーのバインドを解除すると、他のポリシーも無効になります。

[NSHELP-35960]

- 次の条件がすべて満たされると、NetScaler がクラッシュする可能性があります:
  - イベント、監査ログ、またはメトリックスは、分析プロファイルまたは AppFlow パラメータで有効になっています。
  - 応答側の書き換えポリシーが設定されます。

  [NSHELP-35550]
- NetScaler を再起動すると、SNMP アラームは以前に無効になっていても自動的に再び有効になります。

  [NSHELP-34745]
- 多要素認証が構成された NetScaler は、ポリシー評価中にクラッシュします。

  [NSHELP-33674]
- 再起動後、newnslog 構成ファイルが空の場合、VPX は再起動を続けます。

  [NSHELP-33373]
- クラスタ展開では、非 CCO ノードは、ノードで「強制クラスタ同期」または「再起動」操作を実行しても、TCP Syslog メッセージを外部 Syslog サーバに送信しません。

  [NSHELP-32925]
- `show syslogAction`コマンドは、次の条件の両方が満たされると、出力に未解決の IP アドレスを表示します。
  - トランスポートモード UDP でドメイン名を指定した SYSLOG アクションが使用されます。
  - ICMP はサーバ上で無効になっています。

  この問題は、サーバに ICMP 経由でアクセスできないため、ping のデフォルトモニタがサービスを DOWN とマークするために発生します。そのため、IP アドレスは解決されても出力には表示されません。

  [NSHELP-32886, NSHELP-33392]
- `ns.log` このファイルは、監査ログレベルが「なし」に設定されていて、設定されたファイルサイズ制限を超えている場合でも、デバッグログを生成します。この問題は、詳細ポリシーが不要であってもローカルロギングにバインドされていることが原因で発生します。

  [NSHELP-32404, NSCXLCM-1374, NSCXLCM-1551, NSCXLCM-1708]
- クラスター環境では、syslog ポリシーが lb 仮想サーバーにバインドされると、NetScaler で nsppe がクラッシュすることがあります。

  [NSHELP-30983, NSANINFRA-21]

認証、承認、監査

- アップグレード後、NetScaler ログファイルに「AAA DHT: サブタイプ 1 が無効なため、VPN エントリ再開通知が失敗しました」というメッセージが繰り返し表示されます。

  [NSHELP-35649]

- アップグレード後、ユーザーは期限切れの NetScaler Gateway パスワードをリセットできません。この問題は、二重認証ログインスキーマによる StoreFront 認証が nfactor フローの第 2 要素として構成されている場合に発生します。

  [NSHELP-35631]

- NetScaler GUI 認証サーバーページ（[構成] **>** [認証] **>** [ダッシュボード]）で、[ステータス] 列が正しく読み込まれず、「処理中の小さな画像」というメッセージが表示されます。この問題は、サーバー構成が進行中のエントリで発生します。

  [NSHELP-34534]

- Kerberos SSO は、同時に大量の要求を受信すると失敗することがあります。

  [NSHELP-34177]

- TACACS 認証が NAT IP アドレスで構成されている場合、NetScaler CLI を使用してコマンドを実行する際に遅延が発生する可能性があります。

  [NSHELP-32960]

- SAML サービスプロバイダーとして構成された NetScaler は、SAML ログアウト要求に対して 2 つの応答を送信する場合があります。この問題は、ログアウトリクエストに「SAML Request」パラメータが含まれている場合に発生します。

  [NSHELP-32555]

- 認証仮想サーバーをデフォルト以外のパーティションで使用すると、NetScaler がクラッシュすることがあります。

  [NSHELP-32054, NSCXLCM-640, NSCXLCM-1577]

- 次の認証方法のいずれかを第 2 要素として使用し、その後に nFactor フローでユーザー操作を必要とする要素が設定されると、NetScaler がクラッシュする可能性があります。

  - SAML
  - OAuth
  - クライアント証明書

  [NSHELP-29573, NSCXLCM-492, NSCXLCM-872, NSCXLCM-1216, NSHELP-32631, NSHELP-32765]

ボット管理

- まれに、デバイスの指紋検出技術を使用しているときに Web ページが読み込まれないことがあります。

  [NSHELP-34742]

負荷分散

- セカンダリノードの RPC パスワードが変更されると、セカンダリノードの GSLB 同期が失敗します。

  [NSLB-9788]

- GLSB レスポンスに 300 を超える IP アドレスが含まれていると、NetScaler がクラッシュすることがあります。

  [NSHELP-35792]

- トラフィックドメインの展開では、ネットプロファイルが DNS 仮想サーバーにバインドされると、DNS クエリの負荷分散が失敗することがあります。

  [NSHELP-35675]

- 次の一連の条件が満たされた後にドメイン名ベースのサービス（DBS）を参照すると、NetScaler がクラッシュすることがあります。

  1. ロケーションエントリは、DBS ドメイン名の解決先となる IP アドレスに設定されます。
  2. DBS ドメイン名が削除され、ネームサーバーから NXDOMAIN 応答が返されます。
  3. ロケーションエントリが削除されます。

  [NSHELP-35370]

- まれに、次の条件が満たされると、NetScaler アプライアンスがクラッシュしてコアダンプが生成されることがあります:

  - TCP ベースの DNS モニタプローブは、バックエンドサービスのモニタリングに使用されます。
  - アプライアンスのメモリが不足しています。

  [NSHELP-35289]

- HA セットアップでは、セカンダリノードを再起動すると静的近接データベースがロードされないことがあります。

  [NSHELP-35271]

- NTLM モニターは以下のオプションをサポートしていません。

  - NTLM バージョン 1 とバージョン 2 の両方の構成のモニターによる同時プロービング。
  - 「ScriptArg」パラメータの URL が別の IP アドレスに解決されたときに、プローブをサーバーの IP アドレスに転送します。

- NTLM バージョン 2。

[NSHELP-35185]

- 30 を超えるサービスがユーザーモニターにバインドされている場合、ユーザーモニターにバインドされているサービスが断続的に使用できなくなる可能性があります。

  [NSHELP-34669, NSCXLCM-1373]

- 8 ノード以上のクラスタ構成では、レート制限識別機能が意図したとおりに動作しない場合があります。

  [NSHELP-34555]

- タイムアウトの計算が正しくないため、StoreFront ユーザーモニターへのプローブが失敗することがあります。この問題は、StoreFront ユーザーモニターの構成時にタイムアウト値が 1 秒または 2 秒に設定されている場合に発生します。

  [NSHELP-34418]

- HA フェイルオーバー後、パーシスタンスタイムアウト期間が終了しても、パーシスタンスセッションのエントリはプライマリノードから削除されません。セッションエントリは、セカンダリノードが稼働するまで保持されます。

  [NSHELP-34378]

- 静的近接が GSLB 方式として設定されていて、データベースからのクライアントロケーション検索が失敗すると、CPU 使用率が高くなる可能性があります。

  [NSHELP-33823]

- 高可用性セットアップでは、フェールオーバー後にパーティションを削除すると、CPU 使用率が高くなる可能性があります。

  [NSHELP-33701]

- 次の条件が満たされると、NetScaler がクラッシュする可能性があります:

  - 負荷分散仮想サーバーは、複数のパーティションのリダイレクト URL で構成されます。
  - メモリ回復がトリガーされます。

  [NSHELP-33638, NSCXLCM-227, NSCXLCM-509]

- SOA の連絡先情報では、複数のドット文字を含む電子メールアドレス (例:john.doe.example.com) を入力すると、`john@doe.example.com`に変換されます。バックスラッシュ () をエスケープ文字として使用できるようになりました。その結果、john.doe.example.com は`john.doe@example.com`に翻訳されます。

  [NSHELP-33610]

- キャッシュリダイレクト機能は無効になっていますが、仮想サーバーは引き続きトラフィックを処理します。この問題は、キャッシュリダイレクト仮想サーバーの IP アドレスとポート番号が GSLB サービスに追加された場合に発生します。

[NSHELP-33495]

- インクリメンタル同期を実行する場合、モニターの「resptimeout」パラメーターが変更されると完全同期がトリガーされます。

  [NSHELP-31987]

その他

- ns.log ファイルの STA 応答ログは、デバッグレベルで出力されます。

  [NSHELP-35956]

- NetScaler が NetScaler が生成した Cookie を受信 HTTP 要求から削除してから上流の HTTP サーバーに送信すると、上流のサーバーはその要求を拒否することがあります。この問題は、Cookie の名前と値のペアを削除すると、Cookie ヘッダーフィールドが HTTP プロトコルの仕様を満たさなくなる可能性があるために発生します。

  [NSHELP-35855]

- ICA 接続がアクティブな VPN 仮想サーバーの名前を変更すると、NetScaler がクラッシュすることがあります。

  [NSHELP-35804]

- 次の条件が満たされると、NetScaler がクラッシュする可能性があります:

  - EDT 経由のアクティブな ICA 接続があります。
  - Citrix VDA と同じ IP アドレスとポート番号の UDP サービスが追加されます。
  - NetScaler Gateway と Citrix VDA の間には接続の問題があります。

  [NSHELP-35637]

- アップグレード後、HDX Insight が有効になっていると、NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-35058、NSCXLCM-1220、NSCXLCM-1467]

- クラスタ設定がアイドル状態の場合、ノード間メッセージング（NNM）は、指定された sndbuf サイズ（-S オプション付きの ping コマンド）の ping パケットに 20 ミリ秒の遅延を追加することがあります。

  [NSHELP-34774]

- HDX Insight で構成された NetScaler は、セカンダリノードが処理対象のパケットを受信すると再起動することがあります。

  [NSHELP-34152]

- NetScaler Gateway は、承認されたアクセス要求を SSO 障害として NetScaler ADM に報告します。その結果、NetScaler ADM UI の［Gateway］**>**［**Gateway Insight**］ページに、誤ったアラームの原因となる誤った SSO 障害レポートが表示されます。

  [NSHELP-27992]

- EDT ICA 接続が開始されると、NetScaler がクラッシュします。この問題は、HDX Insight の AppFlow 分析プロファイルが VPN 仮想サーバーにバインドされている場合に発生します。

  [GOPHDX-5014]

NetScaler Gateway

- バックエンドサーバーが HTTP/1.0 モードのプロキシリクエストを拒否するため、一部のアプリケーション (内部および外部) にアクセスできない場合があります。

  [NSHELP-35919]

- NetScaler GUI では、バインドに成功しても、VPN 仮想サーバーとの SAML IdP ポリシーバインディングは表示されません。

  [NSHELP-35663]

- NetScaler Gateway で高度なクライアントレス VPN アクセスを構成すると、ブックマークされた URL からページを読み込めないことがあります。

  [NSHELP-33771]

- NetScaler Gateway 経由で VPN 接続を確立すると、URL に誤ったテキストが含まれたホームページにリダイレクトされることがあります。この問題は、NetScaler が RFWebUI ポータルテーマで構成されている場合に発生します。

  [NSHELP-30097, NSCXLCM-481]

- EULA エンティティを作成すると、NetScaler Gateway の RFWebUI ポータルテーマにテキストが 1 行で表示されます。この問題は、HTML
 の改行タグが原因で発生します。使用許諾契約書では、
 すべての HTML タグが一時的に無効になっています。「n」を使用して改行を追加してみることができます。

  [CGOP-24534]

- NetScaler リリース 13.1 ビルド 50.23 以降、NetScaler Gateway 仮想サーバーまたは認証仮想サーバーを介してアクセスすると、すべてのディレクトリで PHP ファイルへのアクセスがブロックされます。

  [CHOP-22974]

NetScaler SDX アプライアンス

- NetScaler SDX FIPS では、NetScaler インスタンスをプロビジョニングまたは変更している間は、「FIPS を有効にする」オプションは使用できません。

[NSSVM-5848]

- VPX を内部管理ネットワークがサポートされていないバージョンからサポートされているバージョンにアップグレードした場合、NetScaler VPX の内部管理ネットワークを有効にすることはできません。

  [NSSVM-5634]

- NetScaler SDX に存在する VPX の IP アドレスで、それぞれのオクテットの桁数が異なる場合、`snmpwalk` get 呼び出しではすべての VPX に対して応答が返されないことがあります。

  [NSHELP-35877]

- 25G、40G、または 100G ポートを使用する LACP チャネル作成の最大スループットチェックの検証が失敗します。

  [NSHELP-35743]

NetScaler Secure Web Gateway

- まれに、NetScaler がパケットキャプチャ中にクラッシュすることがあります。この問題は、PCB 構造の TCP プロファイル変数に NULL 値がある場合に発生することがあります。

  [NSHELP-36081]

NetScaler Web App Firewall

- 変換ファイル要求のデバッグログは、URL 変換プロファイルが要求に適用されたときに生成されます。

  [NSWAF-10356]

- SQL 文法ベースの保護機能が有効になっている場合、特定の定義済みキーワードを使用すると、誤検知が発生する可能性があります。

  [NSHELP-36138]

- BOT レート制限トラフィックフィルターは、**Bursty** フィルターとして設定されていても、内部的にはスムーズフィルターとして機能します。

  [NSHELP-36095]

- Web App Firewall プロファイルに、SQL インジェクション保護をチェックする署名がバインドされていると、NetScaler がクラッシュすることがあります。

  [NSHELP-35989]

- URL 変換ポリシーを使用してホストヘッダーを更新すると、応答ヘッダーはポート番号で 2 回更新されます。

  [NSHELP-35840]

- ルールがキーと値のペアで構成されている場合、NetScaler GUI を使用して JSON クロスサイトスクリプティング緩和ルールを編集または削除することはできません。

  [NSHELP-35610]

- JSON コマンド・インジェクションの文法保護チェックは、日付形式を含むリクエストをブロックします。ただし、ログファイルのデータは更新されず、カウンタは増加しません。

  [NSHELP-35577]

- まれに、構成されたメモリが少なくて Web App Firewall プロファイルを使用すると、NetScaler がクラッシュすることがあります。

  [NSHELP-35463]

- 有効なセッションの Cookie が別のセッションで再利用されると、Cookie ハイジャック保護機能が意図したとおりに機能しません。NetScaler は要求をブロックせずに許可します。

  [NSHELP-33723]

ネットワーク

- NetScaler BLX アプライアンスを実行中にアンインストールしても、NetScaler BLX NIC の設定が Linux ホストに残っている場合があります。

  回避策。アプライアンスをアンインストールする前に、NetScaler BLX アプライアンスを停止してください。

  [NSNET-29109]

- デフォルト以外のパーティションからルートモニターをバインドすると、
「Operation not permitted」というエラーメッセージが表示されます。
ただし、対応するルートがデフォルト以外のパーティションに存在する場合、ルートモニターのステータスは UP と表示されます。

  [NSNET-28589]

- NetScaler BLX をアップグレードした後、新しい「blx.config」ファイルに「blx-managed-host」パラメーターと「ホスト IP アドレス」パラメーターがありません。その結果、管理対象ホストの IP アドレスへのアクセスが失われます。

  [NSHELP-36092]

- NetScaler アプライアンスが「SNMP GETBULK」リクエストに応答しないことがあります。

  [NSHELP-35902]

- VTYSH ターミナルは、VTYSH プロンプトに戻る前に show コマンド出力の最後に「more」と表示します。この問題は、NetScaler アプライアンスの基盤となる FreeBSD OS がバージョン 11.4 にアップグレードされたために発生します。

  [NSHELP-35829]

- 管理パーティションの 1 つを削除すると、NetScaler は他のパーティションのパケットバッファーも削除する場合があります。その結果、パケットバッファーが削除されたパーティションを削除すると、NetScaler がクラッシュすることがあります。

  [NSHELP-35595]

- 大規模 NAT（LSN）セットアップでは、LSN フロントエンド接続とバックエンド接続で内部数の不一致があると、NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-35318]

- 大規模な統合キャッシュまたは大規模 NAT 構成を含む NetScaler アプライアンスをリリース 12.1 または 13.0 からリリース 13.1 または 14.1 にアップグレードすると、パケットエンジンクラッシュからの回復時間は、アップグレード前のバージョンよりも比較的長くなります。

  [NSHELP-33797]

- 高可用性セットアップでは、次の条件がすべて満たされると、ノードの状態が UP になるまでに 60 秒以上かかります:

  - HA セットアップではフェイルセーフが有効になっています
  - HA モニタリングは複数のインターフェイスで有効になっています
  - HA モニタリング対応インターフェイスの 1 つがアクセス不能になる
  - HA モニタリングインターフェイスの少なくとも 1 つにアクセス可能

  今回の修正により、これらの条件がすべて満たされると、ノードの状態は直ちに UP になります。

  [NSHELP-32157]

プラットフォーム

- ESX ハイパーバイザー上の NetScaler VPX は、VMXNET3 インターフェイスでクラッシュする可能性があります。

  [NSPLAT-27262、NSHELP-36781、NSCXLCM-3163]

- まれに、10G NIC を使用してバージョン 13.1 を実行している NetScaler が、トラフィックがごくわずかなときにクラッシュすることがあります。

  [NSHELP-36274]

- Mellanox ConnectX3 NIC による Azure アクセラレーテッドネットワーキングをサポートする NetScaler VPX インスタンスでは、トラフィックが断続的にドロップすることがあります。

  [NSHELP-35666]

- Intel Fortville NIC を搭載した NetScaler SDX では、VPX インスタンスに Fortville インターフェイスが追加されるたびに、VPX インスタンスの管理 CPU 使用率が 1% 増加します。

  [NSHELP-35445, NSCXLCM-768]

- SSLv3、TLS 1.0、TLS 1.1 などの従来の SSL プロトコルを使用して NetScaler SDX でホストされている Citrix Hypervisor にアクセスすることはできなくなりました。

  [NSHELP-33196]

SSL

- ログファイルが異常に大きいと、NetScaler がクラッシュすることがあります。たとえば、SSL ハンドシェイクログを監視するためにログレベルを DEBUG に設定した場合、ファイルには他のモジュールの詳細なデバッグログも含まれるため、ログファイルのサイズが大幅に増加します。SSL デバッグログは、「nsapimgr」ノブを使用してログレベルを INFO に設定することで取得できます。その結果、ログファイルのサイズも小さくなります。

  [NSSSL-13206]

- NetScaler のメモリ使用量が高く、構成が頻繁にクリアされる場合、デフォルトの CA 証明書グループを削除すると NetScaler がクラッシュすることがあります。

  [NSHELP-35441]

- クライアントからの再送信されたハンドシェイクフライトの処理中にメモリが適切に解放されないため、NetScaler で DTLS トラフィックのメモリが大量に蓄積されることがあります。

  [NSHELP-35359、NSCXLCM-999、NSCXLCM-1968、NSCXLCM-2405、NSCXLCM-2482]

- ハイブリッドモードで動作する NetScaler MPX/SDX 14000 FIPS では、キー交換の一環として破損したデータを受信すると、キーメモリがリセットされることがあります。

  [NSHELP-35020]

- クライアントアプリケーションが TLS1.3 SSL 接続を介してパディング付きの大きなファイルをアップロードすると、アップロードが失敗することがあります。この障害は、パディングされたバイトが受信バッファから解放されないため、TCP 受信バッファがいっぱいになったために発生します。

  [NSHELP-34490]

- キー交換中に DH 512 暗号を使用すると、Intel Coleto または Intel Lewisburg チップを搭載した NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-34094]

- Coleto チップを搭載した NetScaler プラットフォームでは、ハンドシェイクメッセージのサイズが量子サイズよりも大きいと、SSL 再ネゴシエーションハンドシェイクが失敗します。

  [NSHELP-33924]

- SSL ハンドシェイクで交換されるクライアント、サーバー、および CA 証明書の合計サイズが 16K の制限を超えると、トラフィックが集中しているときに一時的にパフォーマンスに影響が出る可能性があります。

  [NSHELP-33905]

- SSL_TCP（フロントエンド）仮想サーバーと TCP（バックエンド）サービスを備えた NetScaler 展開環境では、クライアント要求が断続的に失敗することがあります。この障害は、NetScaler がフロントエンドで受信した SSL クライアントの Hello メッセージを転送するが、バックエンドでは処理できず、要求が失敗するために発生します。

  [NSHELP-32806]

- OpenSSL 3.x ベースの TLS クライアントは、RFC 5746 (再ネゴシエーション指示拡張またはセキュア・リネゴシエーション) のサポートに関するクライアントのアドバタイズメントをサーバーが承認しない限り、ハンドシェイクを途中で終了します。フロントエンドの仮想サーバーは、再ネゴシエーションが無効になっている場合、このアドバタイズメントを無視し、接続障害の原因となります。今回の修正により、再ネゴシエーションが無効になっていてもフロントエンドの仮想サーバーがアドバタイズメントを確認するようになり、互換性が向上しました。

  [NSHELP-35120]

システム

- TLS HTTP/2 接続では、事前に TCP FIN フラグがない状態で TLS クローズ通知メッセージを受信しても、NetScaler は HTTP/2 ゴーアウェイメッセージを送信しません。

  [NSHELP-36248]

- AppFlow と HTTP 圧縮の両方の機能が有効になっていると、NetScaler が誤った応答を送信することがあります。

  [NSHELP-35862]

- NetScaler は、クライアント TCP 接続で CONNECT HTTP 要求を受信しても、「407 プロキシ認証が必要です」という HTTP 応答がすでに送信されている以前のサーバー TCP 接続を再利用しません。代わりに、NetScaler は新しい TCP 接続で CONNECT HTTP リクエストをバックエンドサーバーに転送します。新しい TCP 接続で要求を転送すると、同じ TCP 接続で複数の HTTP 認証メッセージを交換する必要がある NTLM などのプロキシ認証プロトコルが破られます。

  [NSHELP-35717、NSCXLCM-1514、NSCXLCM-1835、NSCXLCM-1910]

- まれに、フロントエンド最適化（FEO）機能を有効にすると NetScaler がクラッシュすることがあります。

  [NSHELP-34861]

- 次の条件が満たされると、NetScaler はデータ転送を停止することがあります。

  - 複数の機能が有効になっています。
  - 複数の機能が TCP または HTTP ペイロードの同じ部分を削除しようとしています。

  [NSHELP-33793、NSCXLCM-1512、NSCXLCM-1954]

- バックエンドサーバーが HTTP リクエストに対して 464 エラーを送信すると、NetScaler はそのエラーをクライアントに転送しないため、クライアント側の接続は停止します。

  [NSHELP-33571, NSCXLCM-1098]

- NetScaler は、HTTP ベースの機能が大量のアプリケーションデータをバッファしようとすると、HTTP/2 接続でのデータ転送を停止することがあります。

  [NSHELP-32612]

- SSL サービスで構成された NetScaler アプライアンスは、アプライアンスが TCP FIN 制御パケットの後に TCP RESET 制御パケットを受信するとクラッシュします。

  [NSHELP-31656]

ユーザーインターフェイス

- HA セットアップでは、プライマリノードとセカンダリノードの NSIP アドレスに一意の SSL 証明書があっても、セカンダリノードの証明書はプライマリノードの証明書によって上書きされます。

  [NSHELP-35938]

- NetScaler GUI で GSLB 仮想サーバーを編集すると、負荷分散ポリシーを仮想サーバーにバインドした後、ポリシーセクションが **GSLB** 仮想サーバーページに表示されません。

  [NSHELP-35899]

- SSL デフォルトプロファイルが有効になっている場合、DTLS タイプの負荷分散仮想サーバーの暗号設定を GUI を使用して構成することはできません。

  [NSHELP-35704]

- CLI または GUI を使用して GSLB 構成を強制的に同期すると、同期が失敗し、次のメッセージが表示されます。
  「一部のコマンドが失敗しました」。

  この問題は、設定で bind DNS グローバルコマンドを使用している場合に発生します。

  [NSHELP-35699]

- NetScaler ADC インスタンスの作成時には、一部の組み込み構成を使用できません。

  [NSHELP-33451, NSCXLCM-502]

- NetScaler GUI では、nFactor 認証が失敗し、「認証中にアクティブなポリシーはありません」というエラーが表示されます。この問題は、割り当てアクションが設定されているが、認証ポリシーにバインドされていない場合に発生します。

  [NSHELP-33339]

既知の問題

リリース 13.1-50.23 に存在する問題。

分析インフラストラクチャ

- クラスタ導入環境では、CCO 以外のノードで「force cluster sync」コマンドを実行すると、ns.log ファイルに重複するログエントリが含まれます。

  [NSANINFRA-2850、NSGI-1293]

- NetScaler ADM を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに動作しません。

  回避策：管理ポッドを再起動します。

  [NSANINFRA-1504]

認証、承認、監査

- iOS 向け Citrix SSO のアップグレード後、認証のために受信するプッシュ通知に音が鳴らないことがあります。

  [NSHELP-27525]

- 管理者は、認証情報が無効であることが原因で発生した認証エラーに対してカスタムロギングを実行することはできません。この問題は、NetScaler Responder ポリシーがログインエラーのエラーを検出できないために発生します。

  [NSAUTH-11151]

- ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
  `show adfsproxyprofile <profile name>`

  回避策：クラスター内のアクティブなプライマリの NetScaler に接続し、`show adfsproxyprofile <profile name>`コマンドを実行します。プロキシプロファイルの状態が表示されます。

  [NSAUTH-5916]

- 次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

  - [LDAP 到達可能性をテスト] オプションが開きます。
  - 無効なログイン認証情報が入力され、送信されます。
  - 有効なログイン認証情報が入力され、送信されます。

  回避策：「LDAP 到達可能性テスト」オプションを閉じて開きます。

  [NSAUTH-2147]

負荷分散

- 高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

  [NSLB-7679]

- レート制限レコードの取得とレコードのエージングプロセスの間のタイミングの問題により、NetScaler がクラッシュする可能性があります。

  [NSHELP-33349]

- いくつかの内部仮想サーバーでモニタープローブが失敗すると、NetScaler がクラッシュすることがあります。

  [NSHELP-30985]

- ユーザーがドメインベースのサーバーをサービスグループにバインドまたはバインド解除すると、モニタープローブは失敗します。

  [NSHELP-29330]

- NetScaler GUI を使用して次の手順を実行すると、実行構成に余分な CNAME レコードが表示されます:

  1. DNS レコードタイプ CNAME の GSLB 仮想サーバーを作成します。
  2. DNS レコードタイプ CNAME を設定します。
  3. 構成を保存します

  この問題は表面的なもので、機能には影響しません。

  [NSHELP-29217]

- `entityofs` サービスグループのトラップの ServiceGroupName 形式は次のとおりです。
  `<service(group)name>?<ip/DBS>?<port>`

  トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (“?”) は区切り文字として使用されます。NetScaler は、疑問符 (” ?”)。このフォーマットは、NetScaler ADM GUI でも同じように表示されます。これは予想される動作です。

  [NSHELP-28080]

NetScaler Gateway

- 次の条件をすべて満たすと、NetScaler がクラッシュします:

  - VPN 仮想サーバーは IPv6 アドレスで構成されます。
  - IPv6 仮想サーバーでは、IPv4 IIP アドレスのみ (IPv6 IIP アドレスなし) が設定されます。

  [NSHELP-35559]

- NetScaler Gateway 経由で VPN に接続している場合、アップグレード後に NetScaler GUI に HTTP 経由でアクセスできなくなることがあります。

  [NSHELP-35015]

- スキーマをブラウズしているときに、「未定義のプロパティ「タイプ」を読み取れません」というエラーメッセージが表示されることがあります。

  [NSHELP-21897]

- Windows OS オプションは、NetScaler GUI の事前認証ポリシーと認証アクションの「エクスプレッションエディタ」ドロップダウンリストに表示されません。ただし、GUI または CLI を使用して以前の NetScaler ビルドで Windows OS スキャンをすでに構成している場合は、アップグレードしても機能に影響はありません。必要に応じて CLI を使用して変更を加えることができます。

  回避方法:

  設定には CLI コマンドを使用します。

  - nFactor 認証で高度な EPA アクションを設定するには、次のコマンドを使用します。
    認証 epaAction adv_win_scan-csecexpr「sys.client_expr (「sys_0_win-os_name_Anyof_win-10 [コメント:Windows OS]」)」を追加
  - 従来の事前認証アクションを設定するには、次のコマンドを使用します。
    add aaa preauthenticationaction win_scan_action ALLOW
    add aaa preauthenticationpolicy win_scan_policy “CLIENT.SYSTEM(‘WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]’) EXISTS” win_scan_action

  [CGOP-22966]

- Windows ログオン機能の前に常時接続 VPN を使用するには、NetScaler Gateway を 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を使用できます。

  [CGOP-19355]

- NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

  [CGOP-11830]

- Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、[重大なエラー] ダイアログボックスが表示されます。また、ページが応答しなくなります。

  [CGOP-7269]

NetScaler Secure Web Gateway

- URL フィルタリングのサードパーティベンダーで接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

[NSHELP-22409]

ネットワーク

- NetScaler BLX をアップグレードしても、nitro-python パッケージが更新されない場合があります。その結果、アップグレード中に次のエラーが表示されることがあります:

  「tar: /var/netscaler/nitro/ns_nitro-python_artesa_xx_xx.tar: アーカイブに見つかりません」

  回避方法:

  NetScaler BLX をアップグレードする前に、「rm-rf /var/netscaler/nitro/ns_nitro-python_ *.tar」コマンドを実行する必要があります。

  [NSNET-27927]

- DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

  [NSNET-25299]

- DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

  - NetScaler BLX アプライアンスには、少数の「巨大ページ」が割り当てられています。たとえば、1G です。
  - NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

  この問題は、エラーメッセージとして「/var/log/ns.log」に記録されます。

  - 「BLX-DPDK: DPDK メモリプールを PE-x 用に初期化できませんでした」

  注:x はワーカープロセス数以下の数です。

  回避策:「巨大ページ」を多数割り当ててから、アプライアンスを再起動します。

  [NSNET-25173]

- DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

  [NSNET-24449]

- DPDK を搭載した NetScaler BLX アプライアンスの Intel X710 10G (i40e)インターフェイスでは、次のインターフェイス操作はサポートされていません:

  - 無効化
  - 有効化

- リセット

[NSNET-16559]

- NetScaler BLX アプライアンスのインストールは、Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）で失敗し、次の依存関係エラーが発生することがあります：

  「次のパッケージは依存関係が満たされていません:blx-core-libs: i386: preDepends: libc6: i386 (>= 2.19) でもインストールできません」

  回避策：NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します：

  - dpkg –add-architecture i386
  - apt-get update
  - apt-get install libc6:i386

  [NSNET-14602]

- FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

  [NSNET-5233]

- NetScaler BLX を DPDK サポートで構成すると、NetScaler BLX は一部のファームウェアバージョンの NIC で DPDK 互換の NIC ポートを検出できないことがあります。その結果、NIC ポートは非 DPDK ポートとして NetScaler BLX に追加されます。

  [NSHELP-36290]

- NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

  [NSHELP-21082]

プラットフォーム

- ソフトウェアを 14.1-8.x 以降または 13.1-50.x 以降にアップグレードすると、銅線の 1G SFP トランシーバを備えた 25G インターフェイスがリモートスイッチまたはネットワークデバイスに接続できなくなります。

  この問題は、25G NIC を搭載した次のプラットフォームで発生します：

  - SDX 9100
  - SDX 15000
  - SDX 16000
  - SDX 26000

- SDX 26000-50S

[NSPLAT-27864]

- Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。「rm cloudprofile」コマンドを使用してプロファイルを削除します。

  [NSPLAT-4520]

- Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
  回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノードで設定する必要があります。

  [NSPLAT-4451]

- 高可用性セットアップのセカンダリ NetScaler SDX が共有 CPU コアで構成され、高可用性ハートビートが VLAN 経由で交換される場合、プライマリノードへの移行は失敗します。

  [NSHELP-32412、NSCXLCM-789、NSCXLCM-1847、NSCXLCM-2063、NSHELP-36440]

ポリシー

- 処理データのサイズが設定されたデフォルトの TCP バッファーサイズを超えると、接続がハングアップすることがあります。

  回避策:TCP バッファサイズを処理する必要があるデータの最大サイズに設定します。

  [NSPOLICY-1267]

SSL

- NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

  回避方法:

  1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
  2. 構成を保存します。

  [NSSSL-9572]

- 認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

[NSSSL-6478]

- 同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

  [NSSSL-6213]

- HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
  エラー:CRL 更新は無効です

  [NSSSL-6106]

- セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

  [NSSSL-4427]

- SSL プロファイル内の SSL プロトコルまたは暗号を変更しようとすると、「警告:SSL vserver/Service で使用可能な暗号が設定されていません」という誤った警告メッセージが表示されます。

  [NSSSL-4001]

- 期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

  [NSSSL-3184, NSSSL-1379, NSSSL-1394]

システム

- 次の条件が満たされると、TCP 接続で高い RTT が発生します:

  - 最大輻輳ウィンドウ（> 4 MB）が高く設定されている
  - TCP NILE アルゴリズムは有効になっています

  NetScaler アプライアンスが NILE アルゴリズムを使用して輻輳制御を行うには、条件がスロースタートのしきい値と最大輻輳ウィンドウを合わせた値を超える必要があります

  そのため、設定された最大輻輳時間帯に達するまで、NetScaler はデータを受け付け続け、最終的には RTT が高くなります。

  [NSHELP-31548]

- mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

  [NSBASE-18295]

- `unset nstcpprofile` コマンドを使用して TCP プロファイルパラメーターの設定を解除すると、NetScaler がコアをダンプすることがあります。

  回避策: 代わりに、目的のパラメーター値を指定した`set nstcpprofile`コマンドを使用してください。

  [NSBASE-18724]

- まれに、次の条件がすべて満たされると、NetScaler が HTTP/2 クライアント接続で内部エラーで HTTP/2 GoAway フレームを開始することがあります：

  – クライアントまたはバックエンドサーバーは、クライアントの HTTP/2 接続で最後の WebSocket または Connect ストリームを閉じようとします。
  – 多重化は有効になっています。

  このエラーは、クライアントの HTTP/2 接続で進行中のトランザクションには影響しません。

  回避策: 以下のコマンドを使用して、関連する HTTP/2 プロファイルの接続多重化を無効にします：

“set httpProfile <name> [-conMultiplex (DISABLED)]” ENABLED

[NSBASE-17449]

- Insight の LogStream 転送タイプが構成されている場合、HDX Insight SkipFlow レコードではクライアント IP とサーバー IP が逆になります。

  [NSBASE-8506]

ユーザーインターフェイス

- GUI を使用してテクニカルサポートバンドルを Citrix テクニカルサポートサーバーにアップロードすることはできません。

  回避策：CLI を使用してテクニカルサポートバンドルをアップロードします。

  [NSUI-19315]

- NetScaler GUI では、「ダッシュボード」タブの下にある「ヘルプ」リンクが壊れています。

  [NSUI-14752]

- CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

  回避策：NetScaler GUI または CLI を使用して IPsec プロファイル、IP トンネル、および PBR ルールを追加して CloudBridge Connector を構成します。

  [NSUI-13024]

- GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

  [NSUI-6838]

- GUI を使用して NetScaler をアップグレードすると、GUI または SSH を使用してシステムアップグレードページにアクセスできなくなります。

  [NSHELP-35785]

- 次の条件が満たされると、ユーザーはダウングレードされた NetScaler アプライアンスにログインできないことがあります:

  1. 次の手順のいずれかを実行します。
     - 現在のビルドにアップグレードしたら、システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更して、設定を保存します。
     - 現在のビルドで新しい NetScaler VPX、BLX、または CPX インスタンスをプロビジョニングします。
  2. アプライアンスを以下のいずれかのビルドにダウングレードします。
     - 13.1-4.x
     - 13.0-82.x またはそれ以前
     - 12.1-62.x またはそれ以前

  ダウングレード後に影響を受けたユーザーのリストを表示するには、コマンドプロンプトに「`query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]`
  回避策: 影響を受けたユーザーのパスワードをリセットする」と入力します。詳細については、「[ルート管理者 (nsroot) のパスワードをリセットする方法](https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html」を参照してください。
  注: 以前にアップグレードしたビルドをダウングレードする場合は、ダウングレード中に、以前のビルドのバックアップされた構成ファイル (ns.conf) を使用してこの問題を回避してください。

  [NSCONFIG-8068]

- 次の条件が満たされると、ユーザーはダウングレードされた NetScaler アプライアンスにログインできないことがあります:

  1. 次の手順のいずれかを実行します。
     - 現在のビルドにアップグレードしたら、システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更して、設定を保存します。
     - 現在のビルドで新しい VPX、BLX、または CPX インスタンスをプロビジョニングします。
  2. アプライアンスを以下のいずれかのビルドにダウングレードします。
     - 13.0-47.x またはそれ以前
     - 12.1-56.x またはそれ以前
     - 11.1-64.x またはそれ以前

  ダウングレード後に影響を受けるユーザーのリストを表示するには、コマンドプロンプトで次のように入力します。

  ```
  query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
  ```


回避策: 影響を受けるユーザーのパスワードをリセットします。詳細については、「[ルート管理者 (nsroot) のパスワードをリセットする方法](https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html」を参照してください。

注: 以前にアップグレードしたビルドをダウングレードする場合は、ダウングレード中に以前のビルドのバックアップされた構成ファイル (ns.conf) を使用してこの問題を回避してください。

[NSCONFIG-3188]

NetScaler 13.1—49.15 ビルドのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリース Build 13.1～49.15 の機能強化と変更、修正された問題と既知の問題について説明します。

メモ

- このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティに関する修正とアドバイスの一覧については、Citrix セキュリティ情報を参照してください。
- ビルド 13.1-49.15 以降のビルドは、CTX579459 で説明されているセキュリティの脆弱性を解決します。
- ビルド 13.1-49.15 はビルド 13.1-49.13 に取って代わります。
- ビルド 13.1-49.15 には、NSHELP-35981、NSHELP-35915、NSHELP-35734、NSHELP-35726、NSHELP-35042、および NSHELP-34825 の修正と、ビルド 13.1-49.13 で利用できるすべての機能強化とバグ修正が含まれています。
- SDX ビルド 49.14 は SDX ビルド 49.13 に取って代わります。この SDX ビルドには、以前の NetScaler ADC 13.1 リリースビルドに存在していた次の問題（NSSVM-5786）に対する修正が含まれています。この修正は SDX バンドルのみを対象としており、ADC nCore ファームウェアには影響しません。
- ビルド 13.1～49.13 以降のビルドは、CTX561482 で説明されているセキュリティの脆弱性に対処します。

新機能

ビルド 13.1—49.15 で利用できる機能強化と変更点。

ネットワーク

- **HA** 同期中はコマンド伝播が無効です。**\ n** 高可用性セットアップでは、**HA** 同期中にコマンド伝播が失敗するのを防ぐために
  、HA 同期中はコマンド伝播が無効になります。

[NSHELP-34253]

プラットフォーム

- **AWS** インスタンスタグを使用した **NetScaler** ソフトウェアバージョン情報の表示のサポート

  これで、NetScaler VPX バージョン情報が AWS インスタンスタグフィールドに追加されました。この変更により、NetScaler インスタンスにログインしなくてもソフトウェアのバージョンを確認できるようになりました。NetScaler インスタンスの起動時にこの情報を追加するには、デフォルトの IAM ロールに「ec2:CreateTags」権限が必要です。

  [NSPLAT-25066]

解決された問題

ビルド 13.1—49.15 で対処されている問題点

認証、承認、監査

- 楕円曲線証明書が VPN にグローバルにバインドされると、OAuth 認証ポリシーを使用して構成された NetScaler がクラッシュすることがあります。

  [NSHELP-34795]

- セッションの有効期限が切れた後、ユーザーが GSLB 構成の NetScaler で認証しようとすると、HTTP 404 エラーが表示されます。

  [NSHELP-34336]

ボット管理

- デバイスのフィンガープリントアクションが LOG、RESET、または REDIRECT に設定されている場合、ボットデバイスのフィンガープリントセッションリプレイ攻撃はドロップされます。

  [NSBOT-1117]

CallHome

- Call Home は、機能が無効になっていても、テレメトリデータを NetScaler テクニカルサポートサーバーに送信します。

  [NSHELP-33240]

負荷分散

- HA セットアップでは、以下の条件が満たされると、DNS サーバーは GSLB ドメインクエリに対して空の応答を断続的に送信することがあります。
  - パーシスタンスは GSLB 仮想サーバーで設定されます。
  - 多数の負荷分散デプロイメントが設定されています。
  - HA フェールオーバーが発生します。

  [NSHELP-35981]
- 次の条件が満たされると、セカンダリ NetScaler がクラッシュする可能性があります。
  - 高可用性セットアップでは、多数の負荷分散サーバーが負荷分散グループで構成されます。
  - 同期中は、負荷分散グループ内の負荷分散サーバーのいずれかで設定操作が実行されます。

  [NSHELP-34225]

その他

- ネットプロファイルがデフォルト以外のトラフィックドメインで設定され、AppFlow 設定で使用されると、システムポートが使い果たされ、トラフィックが影響を受けます。

  [NSHELP-34544]
- NetScaler SDX FIPS では、VPX で追加操作または編集操作を実行すると、次のエラーが表示されます。

  「is_fips_enabled が定義されていません」

  [NSSVM-5786]

NetScaler Gateway

- VPN と AppFlow が構成されている NetScaler がクラッシュし、HA フェイルオーバーが発生することがあります。

  [NSHELP-35734, NSCXLCM-1247]
- NetScaler Gateway のホームページにモバイルブラウザを使用してクライアントレス VPN モードでアクセスしようとすると、アプリが列挙されないことがあります。

  [NSHELP-35541、NSCXLCM-1132、NSCXLCM-1212、NSCXLCM-1248]

NetScaler Web App Firewall

- Web App Firewall プロファイルで「VerboseLogLevel」が「PatternPayloadHeader」に設定されていると、NetScaler がクラッシュすることがあります。

  [NSHELP-35915]

- NetScaler で永久ライセンスを使用している場合は、IP レピュテーションデータベースが Webroot によって更新されないことがあります。

  [NSHELP-33965]

ネットワーク

- 高可用性セットアップでは、変更中に HA 同期の一部としてルートがノードから削除されると、セカンダリノードがクラッシュします。

  [NSHELP-34927]

- 高可用性セットアップでは、次の両方の条件が満たされると、show ha ノードに誤った出力が表示されることがあります。

  - HA ハートビートは、1 つのインターフェイスまたは 1 つのチャネルでのみ交換されます。
  - インターフェイスまたはチャネルは無効になっています。

  [NSHELP-34193]

プラットフォーム

- AWS Autoscaling サービスを NetScaler VPX インスタンスに追加するためのクラウドプロファイルを作成する際、サービスコントロールポリシーがグローバルに設定されていると失敗することがあります。

  [NSHELP-35562]

- AWS プラットフォームの HA ペア構成では、次の構成のフェイルオーバー中に NetScaler VPX インターフェイスが正しく移行されませんでした。

  - HA デプロイメントは同じゾーンにあります。
  - 複数のインターフェイスが同じサブネットを使用しています。

  [NSHELP-35369]

- ファームウェアのアップグレード後、NetScaler MPX 5900/8900 アプライアンスの管理インターフェイスがダウンする可能性があります。その結果、アプライアンスにアクセスできなくなります。

  [NSHELP-31587]

ユーザーインターフェイス

- NetScaler GUI でレスポンダーポリシーまたは書き換えポリシーを構成するときに、［ログアクション］フィールドと［AppFlow Action］フィールドに値を追加しないで（必須ではない）、次のエラーが表示されます。

  ```
  Invalid name; names must begin with an alphanumeric character or underscore and must contain only alphanumerics, '_', '%23', '.', '', ':', '@', '='or '-'[logAction, ]
  ```

  [NSHELP-35726]

- 複数のパーティションを作成または削除すると、重複するパーティション ID が生成されることがあります。その結果、パーティションの作成時に次のエラーが表示されることがあります。

  ```
  Partition-id is already in use by another partition
  ```

  [NSHELP-35042]

- 同じユーザーが 2 つの異なるパーティションにバインドされている場合、ユーザーセッションが誤って計算されます。この 2 つのパーティションは、デフォルト、非デフォルト、またはその両方にすることができます。

  [NSHELP-34971]

- 次の両方の条件が満たされている場合、NetScaler アプライアンスの管理 CPU 使用率が高くなることがあります。

  - 管理パーティションはアプライアンス上で設定されます。
  - アプライアンスは NetScaler ADM によって管理されます。

  [NSHELP-34825、NSCXLCM-192、NSCXLCM-501、NSCXLCM-1279]

- NetScaler Gateway UI で承認ポリシー式を変更すると、「式エディター」ドロップダウンリストに **AAA** オプションが表示されません。

  [NSHELP-33509]

- ユーザーがコンテンツスイッチングポリシーのバインドを表示しても、コンテンツスイッチング仮想サーバーの詳細は [バインディングの表示] の同じ行に表示されません。

  [NSHELP-33149]

既知の問題

リリース 13.1 ～49.15 に存在する問題

認証、承認、監査

- 認証仮想サーバーをデフォルト以外のパーティションで使用すると、NetScaler アプライアンスがクラッシュする可能性があります。

[NSHELP-32054、NSCXLCM-640]

- iOS 向け Citrix SSO のアップグレード後、認証のために受信するプッシュ通知に音が鳴らないことがあります。

  [NSHELP-27525]

- 管理者は、認証情報が無効であることが原因で発生した認証エラーのカスタムロギングを実行できません。この問題は、NetScaler Responder ポリシーがログインエラーのエラーを検出できないために発生します。

  [NSAUTH-11151]

- ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
  `show adfsproxyprofile <profile name>`

  回避策：クラスター内のアクティブなプライマリの NetScaler に接続し、`show adfsproxyprofile <profile name>`コマンドを実行します。プロキシプロファイルの状態が表示されます。

  [NSAUTH-5916]

- 次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

  - [LDAP 到達可能性をテスト] オプションが開きます。
  - 無効なログイン認証情報が入力され、送信されます。
  - 有効なログイン認証情報が入力され、送信されます。

  回避策：「LDAP 到達可能性テスト」オプションを閉じて開きます。

  [NSAUTH-2147]

ボット管理

- BOT ポリシーが複雑なポリシールールを含むログアクションを使用すると、NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-34999]

負荷分散

- 高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

  [NSLB-7679]

- 次の一連の条件が満たされた後にドメイン名ベースのサービス（DBS）を参照すると、NetScaler がクラッシュすることがあります。

1. ロケーションエントリは、DBS ドメイン名の解決先となる IP アドレスに設定されます。
2. DBS ドメイン名が削除され、ネームサーバーから NXDOMAIN 応答が返されます。
3. ロケーションエントリが削除されます。

[NSHELP-35370]

- まれに、次の条件が満たされると、NetScaler アプライアンスがクラッシュしてコアダンプが生成されることがあります:

  - TCP ベースの DNS モニタープローブは、バックエンドサービスの監視に使用されます。
  - アプライアンスのメモリが不足しています。

  [NSHELP-35289]

- 静的近接が GSLB 方式として設定されていて、データベースからのクライアントロケーション検索が失敗すると、CPU 使用率が高くなる可能性があります。

  [NSHELP-33823]

- SOA 連絡先情報では、複数のドット文字 (たとえば `john.doe.example.com`) を含む電子メールアドレスを入力すると、[!`john@doe.example.com`[](`https://issues.citrite.net/images/icons/mail_small.png`)] (`mailto:john@doe.example.com`)。バックスラッシュ () をエスケープ文字として使用できるようになりました。結果として `john.doe.example.com`、[!`john.doe@example.com`[](`https://issues.citrite.net/images/icons/mail_small.png`)] (`mailto:john@doe.example.com`)。

  [NSHELP-33610]

- レート制限レコードの取得とレコードのエージングプロセスの間のタイミングの問題により、NetScaler がクラッシュする可能性があります。

  [NSHELP-33349]

- サービスグループの`entityofs`トラップの ServiceGroupName 形式は次のとおりです。
  `<service(group)name>?<ip/DBS>?<port>`

  トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (“?”) は区切り文字として使用されます。NetScaler は、疑問符 (” ?”)。このフォーマットは、NetScaler ADM GUI でも同じように表示されます。これは予想される動作です。

  [NSHELP-28080]

その他

- 高可用性セットアップで強制同期が行われると、アプライアンスはセカンダリノードで「set urlfilter parameter」コマンドを実行します。

その結果、セカンダリノードは、「TimeOfDayToUpdateDB」パラメーターで指定された次のスケジュール時刻まで、スケジュールされた更新をスキップします。

[NSSWG-849]

- NetScaler Gateway は、承認されたアクセス要求を SSO 障害として NetScaler ADM に報告します。その結果、NetScaler ADM UI の［Gateway ］>［Gateway Insight ］ページに、誤ったアラームの原因となる誤った SSO 障害レポートが表示されます。

  [NSHELP-27992]

- URL フィルタリングのサードパーティベンダーで接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

  [NSHELP-22409]

- クラスタ展開では、CCO 以外のノードで「force cluster sync」コマンドを実行すると、ns.log ファイルには重複するログエントリが含まれます。

  [NSANINFRA-2850、NSGI-1293]

- NetScaler ADM を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに動作しません。

  回避策：管理ポッドを再起動します。

  [NSANINFRA-1504]

- EDT Insight 機能を有効にすると、ネットワークの不一致時にオーディオチャンネルが失敗することがあります。

  [GOPHDX-1055]

- 高可用性セットアップでは、NetScaler のフェイルオーバー中に、NetScaler ADM のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

  [GOPHDX-1050]

NetScaler Gateway

- NetScaler Gateway 経由で VPN に接続している場合、アップグレード後に NetScaler GUI に HTTP 経由でアクセスできなくなることがあります。

  [NSHELP-35015]

- NetScaler Gateway で高度なクライアントレス VPN アクセスを構成すると、ブックマークされた URL からページを読み込めないことがあります。

  [NSHELP-33771]

- スキーマをブラウズしているときに、「未定義のプロパティ「タイプ」を読み取れません」というエラーメッセージが表示されることがあります。

  [NSHELP-21897]

- Windows OS オプションは、NetScaler GUI の事前認証ポリシーと認証アクションの「エクスプレッションエディタ」ドロップダウンリストに表示されません。ただし、GUI または CLI を使用して以前の NetScaler ビルドで Windows OS スキャンをすでに構成している場合は、アップグレードしても機能に影響はありません。必要に応じて CLI を使用して変更できます。

  回避方法:

  設定には CLI コマンドを使用します。

  - nFactor 認証で高度な EPA アクションを設定するには、次のコマンドを使用します。

    ```
    add authentication epaAction adv_win_scan -csecexpr "sys.
    client_expr("sys_0_WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows
    OS]")"
    ```

  - 従来の事前認証アクションを設定するには、次のコマンドを使用します。

    ```
    add aaa preauthenticationaction win_scan_action ALLOW
    add aaa preauthenticationpolicy win_scan_policy "CLIENT.
    SYSTEM('WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]')EXISTS
    "win_scan_action
    ```

  [CGOP-22966]

- Windows ログオン機能の前に常時接続 VPN を使用するには、NetScaler Gateway を 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

  [CGOP-19355]

- NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

  [CGOP-11830]

- Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

  [CGOP-7269]

NetScaler SDX アプライアンス

- NetScaler SDX FIPS では、NetScaler インスタンスをプロビジョニングまたは変更している間は、「FIPS を有効にする」オプションは使用できません。

  [NSSVM-5848]

NetScaler Web App Firewall

- ルールがキーと値のペアで構成されている場合、NetScaler GUI を使用して JSON クロスサイトスクリプティング緩和ルールを編集または削除することはできません。

  [NSHELP-35610]

ネットワーク

- DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

  [NSNET-25299]

- DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

  - NetScaler BLX アプライアンスには、少数の「巨大ページ」が割り当てられています。たとえば、1G です。
  - NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

  この問題は、エラーメッセージとして「/var/log/ns.log」に記録されます。

  - 「BLX-DPDK: DPDK メモリプールを PE-x 用に初期化できませんでした」

  注:x はワーカープロセス数以下の数です。

  回避策:「巨大ページ」を多数割り当ててから、アプライアンスを再起動します。

  [NSNET-25173]

- DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

  [NSNET-24449]

- DPDK を搭載した NetScaler BLX アプライアンスの Intel `X710 10G (i40e)`インターフェイスでは、次のインターフェイス操作はサポートされていません:

  - 無効化
  - 有効化
  - リセット

  [NSNET-16559]

- Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

「次のパッケージの依存関係は満たされていません:blx-core-libs: i386: preDepends: libc6: i386 (>= 2.19) しかしインストールできません」

回避策：NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します:

- dpkg –add-architecture i386
- apt-get update
- apt-get install libc6:i386

[NSNET-14602]

- FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

  [NSNET-5233]

- NetScaler アプライアンスは、SNMPv3 認証失敗トラップメッセージを NetScaler ログファイル (“/var/log/ns.log”) に記録しません。

  [NSHELP-33909]

- NetScaler アプライアンスで管理者パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は自動的に管理者パーティションの新しいメモリ制限に設定されます。

  [NSHELP-21082]

プラットフォーム

- NetScaler アプライアンスを 13.1～4.x 以降のバージョンから次のバージョンのいずれかにダウングレードすると、一部の Python パッケージがインストールされません:

  - 任意の 11.1 ビルド
  - 12.1—62.21 およびそれ以前
  - 13.0-81.x およびそれ以前

  [NSPLAT-21691]

- Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。「rm cloudprofile」コマンドを使用してプロファイルを削除します。

  [NSPLAT-4520]

- Azure の高可用性セットアップでは、GUI を使用してセカンダリノードにログオンすると、autoscale クラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。

回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノードで設定する必要があります。

[NSPLAT-4451]

- 高可用性セットアップのセカンダリ NetScaler SDX が共有 CPU コアで構成され、高可用性ハートビートが VLAN 経由で交換される場合、プライマリノードへの移行は失敗します。

  [NSHELP-32412、NSCXLCM-789]

ポリシー

- 処理データのサイズが設定されたデフォルトの TCP バッファーサイズを超えると、接続がハングアップすることがあります。

  回避策:TCP バッファサイズを処理する必要があるデータの最大サイズに設定します。

  [NSPOLICY-1267]

SSL

- NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

  回避方法:

  1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
  2. 構成を保存します。

  [NSSSL-9572]

- 認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

  [NSSSL-6478]

- 同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

  [NSSSL-6213]

- HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
  エラー:CRL 更新が無効です

  [NSSSL-6106]

- セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

  [NSSSL-4427]

- SSL プロファイル内の SSL プロトコルまたは暗号を変更しようとすると、「警告:SSL vserver/Service で使用可能な暗号が設定されていません」という誤った警告メッセージが表示されます。

  [NSSSL-4001]

- 期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

  [NSSSL-3184, NSSSL-1379, NSSSL-1394]

- クライアントからの再送信されたハンドシェイクフライトの処理中にメモリが適切に解放されないため、NetScaler で DTLS トラフィックのメモリが大量に蓄積されることがあります。

  [NSHELP-35359]

システム

- CONNECT HTTP リクエストメソッドを使用する HTTP/2 ストリームが終了すると、HTTP/2 を使用する Web ページが完全に読み込まれないことがあります。

  [NSHELP-36407、NSBASE-17449]

- NetScaler は、クライアント TCP 接続で CONNECT HTTP 要求を受信しても、「407 プロキシ認証が必要です」という HTTP 応答がすでに送信されている以前のサーバー TCP 接続を再利用しません。代わりに、NetScaler は新しい TCP 接続で CONNECT HTTP リクエストをバックエンドサーバーに転送します。新しい TCP 接続で要求を転送すると、同じ TCP 接続で複数の HTTP 認証メッセージを交換する必要がある NTLM などのプロキシ認証プロトコルが破られます。

  回避方法:

  1. CONNECT メソッドで受信した HTTP リクエストを INVALID としてマークし、さらに INVALID とマークされた HTTP リクエストがドロップされないようにするカスタム HTTP プロファイルを追加します。

```
add ns httpprofile fw-proxy-http-prof -markConnReqInval ENABLED -dropInvalReqs DISABLED
```

  1. このカスタム HTTP プロファイルを、フォワードプロキシサーバーのプールの負荷分散に使用される負荷分散仮想サーバーにバインドします。

```
set lb vs fw-proxy-vs -httpprofileName fw-proxy-http-prof
```

  注: NetScaler の機能ポリシーでは、INVALID とマークされた HTTP 要求または応答は評価されません。

  [NSHELP-35717、NSCXLCM-1514]

- SSL サービスで構成された NetScaler アプライアンスは、アプライアンスが TCP FIN 制御パケットの後に TCP RESET 制御パケットを受信するとクラッシュします。

  [NSHELP-31656]

- 次の条件が満たされると、TCP 接続の RTT が高くなります:

  - 最大輻輳ウィンドウ（> 4 MB）が高く設定されている
  - TCP NILE アルゴリズムは有効になっています

  NetScaler アプライアンスが NILE アルゴリズムを使用して輻輳制御を行うには、条件がスロースタートのしきい値と最大輻輳ウィンドウを合わせた値を超える必要があります

  そのため、設定された最大輻輳時間帯に達するまで、NetScaler はデータを受け付け続け、最終的には RTT が高くなります。

  [NSHELP-31548]

- mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

  [NSBASE-18295]

- Insight 用に LogStream トランスポートタイプが構成されている場合、クライアント IP とサーバー IP は HDX Insight SkipFlow レコードで反転されます。

  [NSBASE-8506]

ユーザーインターフェイス

- NetScaler GUI では、「ダッシュボード」タブの下にある「ヘルプ」リンクが壊れています。

  [NSUI-14752]

- CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

  回避策：NetScaler GUI または CLI を使用して IPSec プロファイル、IP トンネル、および PBR ルールを追加して CloudBridge Connector を構成します。

  [NSUI-13024]

- GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

  [NSUI-6838]

- **LB** アクションの設定ページの「値」フィールドにスペースが含まれている場合、GUI にはエラーメッセージは表示されません。スペースを含む値フィールドを編集すると、GUI はそのスペースをカンマに置き換え、構成が無効になります。

  [NSHELP-35532]

- あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

  1. NetScaler アプライアンスをいずれかのビルドにアップグレードします
     - 13.0 52.24 ビルド
     - 12.1 57.18 ビルド
     - 11.1 65.10 ビルド
  2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
  3. NetScaler アプライアンスを古いビルドにダウングレードします。

  CLI を使用してこれらのシステムユーザーのリストを表示するには、
  コマンドプロンプトで次のように入力します。

  ```
  query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
  ```

  回避策: この問題を解決するには、以下のいずれかの独立したオプションを使用してください。

  - NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3)、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
  - アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
  - 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

  詳細については、https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.htmlを参照してください。

  [NSCONFIG-3188]

NetScaler 13.1—48.47 リリースのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリース Build 13.1—48.47 の機能強化と変更、修正された問題と既知の問題について説明します。

メモ

- このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティに関する修正とアドバイスの一覧については、Citrix セキュリティ情報を参照してください。

新機能

ビルド 13.1—48.47 で利用できる機能強化と変更点。

負荷分散

- 静的近接負荷分散方式の強化

  現在、静的近接負荷分散方式を構成していて、複数のサーバーが異なる場所にある場合、サーバーは NetScaler ループバック IP アドレスではなくクライアント IP アドレスに基づいて選択されます。その結果、場合によっては応答時間が長くなることがあります。ProximityFromSelf というパラメーターが負荷分散パラメーターと負荷分散プロファイルに追加され、クライアントではなく NetScaler に近いサーバーを選択することで応答時間が短縮されます。
  詳しくは、「NetScaler ロケーションの静的近接性」を参照してください。

  [NSLB-9530]

- **HA** の状態が変化しても **GSLB** 完全同期はトリガーされません

  メイン GSLB サイトまたは下位サイトのいずれかで HA の状態が変化しても、GSLB の完全同期はトリガーされなくなりました。以前は、HA ノードが同期していて、HA 状態遷移中に GSLB 構成が変更されていない場合でも、下位サイトへの完全同期がトリガーされていました。GSLB の完全同期を開始しないことで、HA 状態の変更後に段階的に構成を変更しても、下位サイトとの同期が速くなりました。

  [NSLB-9477]

- **Oracle ECV** モニターは最新の **Oracle** 認証プロトコルをサポートします
  。NetScaler Oracle ECV モニターは、21c までのすべての Oracle バージョンとすべてのパスワードベースの認証プロトコルをサポートするようになりました。

  詳細については、「Oracle ECV モニター」を参照してください。

  [NSHELP-9819]

NetScaler Web App Firewall

- レート制限機能の強化

  レート制限タイプとレート制限条件パラメーターを使用して、トラフィックのタイプを制限したり、BOT レート制限機能に条件を追加したりできるようになりました。詳細については、「ボット検出」を参照してください。

  [NSWAF-9535]

ネットワーク

- **NetScaler** 構成、動的ルーティング構成、およびハードウェアセキュリティモジュール構成用の統合構成ファイル

  NetScaler アプライアンスは、NetScaler 構成（ns.conf）、動的ルーティング構成（zebos.conf）、およびハードウェアセキュリティモジュール（HSM）構成（chrystoki.conf）を含む統合構成ファイル（unified.conf）をサポートするようになりました。

  統合構成ファイルにより、さまざまなタイプの構成を 1 つのビューで確認できます。この統合構成ファイルは表示のみを目的としており、別の NetScaler アプライアンスの構成を適用するために使用することはできません。

  NetScaler アプライアンス内の統合構成ファイルの完全なパスは「/nsconfig/unified.conf」です。統合構成ファイルには、シェルのコマンドプロンプトを使用してアクセスできます。統合構成ファイルは、スタンドアロンの NetScaler アプライアンスと高可用性セットアップでのみサポートされます。

  [NSNET-27559]

- **NetScaler BLX** アプライアンスの **DPDK** ポートサポートとしての **VMware VMXNET3** ネットワークポート

  VMware 仮想化プラットフォーム上で動作する Linux ホスト仮想マシンの NetScaler BLX アプライアンスは、VMXNET3 ネットワークポートを DPDK ポートとしてサポートするようになりました。

  [NSNET-27244]

プラットフォーム

- **AWS EC2** インスタンス **IMDSv2** モードのサポート

  AWS EC2 インスタンスのインスタンスメタデータサービスバージョン 2（IMDSv2）モードが NetScaler アプライアンスでサポートされるようになりました。IMDSv1 と IMDSv2 は、実行中の AWS EC2 インスタンスからインスタンスメタデータにアクセスするために使用できる 2 つのモードで、IMDSv2 は IMDSv1 よりも安全です。以前は、IMDSv2 は NetScaler ではサポートされていませんでした。そのため、AWS EC2 インスタンスが IMDSv2 モードを使用していたとき、NetScaler アプライアンスはコールドリブート後に静的デフォルトルートを上書きしていました。

  [NSPLAT-21205]

システム

- **TCPm** タイプの仮想サーバーでのプロキシプロトコルのレスポンダーポリシーサポート **NetScaler** は、**TCP.rn** タイプの仮想サーバー上のプロキシプロトコルのレスポンダーポリシーをサポートするようになりま

した

。

以前は、プロキシプロトコルのレスポンダーポリシーは http.rn タイプの仮想サーバーでのみサポートされていました。

詳細については、「プロキシプロトコル」を参照してください。

[NSHELP-33193]

ユーザーインターフェイス

- **NetScaler GUI** ポリシー表現の **HTML** タグ

  ポリシー表現を作成する際に、NetScaler GUI で HTML タグがサポートされるようになりました。

  [NSUI-18918]

- 署名ビューのフィルター基準の表示ページの **CVE** フィルターカテゴリ

  CVE は、**** 署名ビューページの表示フィルター基準リストにカテゴリの ** 1 つとして追加されます。CVE をフィルターオプションとして使用すると、右側の「フィルター結果」ウィンドウにログ関連の詳細のみが表示されます。

  [NSUI-18512、NSCXLCM-616]

解決された問題

ビルド 13.1—48.47 で対処されている問題。

認証、承認、監査

- クラスター化された NetScaler 展開では、割り当てアクションを認証ポリシーにバインドすることはできません。

  [NSHELP-33974]

- **NetScaler** が **SAML** サービスプロバイダーとして構成されている場合、**SAML: StatusCode** タグの解析の問題が原因で、**SAML** アサーション検証が失敗する可能性があります。

  [NSHELP-33574]

- [セキュリティ] > [AAA-アプリケーショントラフィック] > [ポリシー] > [セッションポリシーとプロファイル] > [セッションプロファイル] ページでセッションプロファイルを編集すると、セッションプロファイルの作成時にオフに設定されていた場合でも、[Web アプリケーションへのシングルサインオン] オプションがオンに設定されます。

  [NSHELP-33067]

- OTP シークレットの暗号化または復号化は、複数値の属性では失敗する可能性があります。

  [NSHELP-31057]

負荷分散

- ADNS サーバーとして構成されている NetScaler は、UDP または TCP プロトコルでクエリを受信すると、構成に基づいて応答を送信します。ただし、同じ TCP または UDP セッションで複数のクエリが送信された場合、最初のクエリへの応答のみが正しく送信されます。DNS ポリシーは、同じ接続での以降のクエリに対して UNDEF を発生させます。

  [NSLB-10103]

- 次の一連の条件が満たされると、NetScaler がクラッシュする可能性があります。

  1. GSLB サービスは、優先度で GSLB 仮想サーバーにバインドされます。
  2. GSLB 仮想サーバーの負荷分散方式は、バックアップ負荷分散方式と同じです。
  3. すべての GSLB サービスは、GSLB 仮想サーバーからバインド解除されます。
  4. GSLB 仮想サーバーは削除されます。

  [NSHELP-34694]

- NetScaler では、統計収集の遅延が原因でパケットドロップが発生します。遅延は、複数のサービスグループが異なるポート上の同じサービス IP アドレスにバインドされているために発生します。

  [NSHELP-34171、NSCXLCM-319]

- 「show server name」コマンドは、サービスがサーバーにバインドされている場合でも、サービスのステータスを「不明」として表示します。

  [NSHELP-33668]

- 多数の AutoscaleGSLB サービスグループが構成されている場合、NetScaler がクラッシュしてコアをダンプする可能性があります。

  [NSHELP-33545]

- NetScaler アプライアンスは、サーバー数の計算が間違っているため、サーバー接続が高い場合に誤った SNMP アラートをトリガーします。

  [NSHELP-31582]

NetScaler Gateway

- NetScaler Gateway で ICA プロキシが有効になっている NetScaler は、ダブルホップ DMZ 展開でクラッシュする可能性があります。

  [NSHELP-33369]

- クラスター化された NetScaler 展開では、ICA のみパラメーターが ON に設定されている場合、強制タイムアウト設定が有効になっていても、NetScaler Gateway が断続的にユーザーセッションの切断に失敗します。

  [NSHELP-33014]

- 特定のユーザーに追加された RDP ブックマークは、これらの URL をブックマークしていない他のユーザーにも表示されます。

  [NSHELP-29904]

- GUI または CLI を使用して構成をクリアすると、セキュアトークン認証局（STA）関連のエンティティがクリアされると NetScaler アプライアンスがクラッシュする可能性があります。

  [CGOP-23152]

NetScaler SDX アプライアンス

- まれに、IP アドレスなどの一部のフィールドに不要な値が入力されているために、NetScaler SDX がクラッシュしてアクセス不能になることがあります。

  [NSHELP-34925]

NetScaler Web App Firewall

- NetScaler アプライアンスは、無効な HTTP ヘッダー情報が原因でクラッシュする可能性があります。この問題は、次の条件が満たされた場合に発生します。
  - HTTP リクエストボディで SQL/XSS 違反が発生しています。
  - 詳細ロギングは「patternPayloadHeader」に設定されています。

  [NSHELP-35297]

- NetScaler は、リクエストカウンターの合計数よりも多い数の Web Application Firewall リクエストカウンターを報告します。これは、XML リクエストではリクエストカウンターが 2 回インクリメントされるためです。

  [NSHELP-34591]

- まれに、ポスト本体の制限をより高い値に設定すると、NetScaler がより多くのメモリを消費することがあります。

  [NSHELP-34507]

ネットワーク

- 構成を保存した後で NetScaler CPX を再起動すると、NetScaler CPX は起動に失敗します。

  [NSNET-28691]

- NetScaler アプライアンスは、アプライアンス上の古い IPv6 一時マッピングをクリーンアップするための内部タイマーの問題により、受信したパケットをドロップすることがあります。

  [NSHELP-34607]

- BGP を設定する場合、redistribute コマンドを入力した後に Tab キーを押しても、VTYSH コマンドラインは自動補完もコマンド候補も表示しません。

  [NSHELP-34332]

- PMTU が有効になっているレイヤー 3 モードでは、NetScaler アプライアンスは、ESP トラフィックの「フラグメンテーションが必要だが DF ビットが設定されている」とマークされた ICMP パケットを転送する代わりにドロップします。

  [NSHELP-34318]

- 大規模な NAT（LSN）セットアップでは、LSN キューの処理における内部の問題により、NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-33499]

プラットフォーム

- メンバーインターフェイスが設定されていない LA チャネルに VRID をバインドすると、NetScaler アプライアンスがクラッシュします。

  [NSPLAT-26707]

- Azure 上の NetScaler VPX が Azure 高速ネットワークを使用している場合、Azure 高速ネットワークのシングルルート I/O 仮想化（SR-IOV）インターフェースは、NetScaler の実行中に Azure によって動的にデタッチおよび再アタッチできます。動的な NIC のデタッチと再接続により、NetScaler は特定のシナリオで応答しない場合があります。

  [NSHELP-34515、NSCXLCM-171、NSCXLCM-908]

- NetScaler SDX アプライアンスをシャットダウンしようとすると、アプライアンスは 1 回目のシャットダウンではなく再起動します。この現象は、アプライアンスがシャットダウンしようとしているときにコアダンプを生成した場合に発生する可能性があります。

  [NSHELP-33276、NSHELP-33192]

ポリシー

- HA セットアップでは、ALL オプションと空の置換文字列を設定すると、REGEX_REPLACE 式がループに陥り、フェイルオーバーが発生する可能性があります。

  [NSHELP-34640]

SSL

- クラスター設定では、デフォルトプロファイルまたはカスタムプロファイルを SSL 内部サービスにアタッチすることはできません。

  [NSSSL-12763]

- チェーンが長く、チェーン内の中間証明書の 1 つがクロス署名ルート証明書である場合、クロス署名証明書の検証は失敗します。

  [NSHELP-34615]

- Intel Coleto または Intel Lewisburg チップを搭載した NetScaler アプライアンスは、ピアサーバーが最初にネゴシエートした暗号とは異なる暗号をネゴシエートすると、バックエンドの再ネゴシエーションフェーズでクラッシュする可能性があります。

  [NSHELP-34324]

- キー交換中に DH 512 暗号を使用すると、Intel Coleto または Intel Lewisburg チップを搭載した NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-34094]

- 内部サービスにバインドされたカスタム暗号のみを含むクラスタ設定では、クラスタ設定をリリース 13.0 からリリース 13.1 にアップグレードすると、DEFAULT 暗号グループも内部サービスにバインドされます。

  [NSHELP-33883]

システム

- NetScaler アプライアンスの SYSLOG 監査モジュールは、アプライアンスを 13.0～88.16 ビルドよりも後のビルドにアップグレードすると、クラッシュして複数のコアファイルをダンプすることがあります。

  [NSHELP-33505]

- AppQoE 構成に RetryOnTimeout パラメーターが構成されている場合、NetScaler アプライアンスがバックエンドサーバーから 1xx HTTP 応答（たとえば「続行」100）を受信するとクラッシュすることがあります。

  [NSHELP-33438]

- Syslog メッセージのタイムスタンプは、サマータイム期間中は正しくありません。

  [NSHELP-30137]

ユーザーインターフェイス

- GUI を使用して HTTP_QUIC 仮想サーバーを作成するときに HTTP プロファイルを選択することはできません。この問題は、HTTP_QUIC 仮想サーバーを作成するための HTTP プロファイルが無効になっているために発生します。

[NSUI-18816]

- GUI では、E メールアクションに関連する高度な認証ポリシーを作成することはできません。これは、認証ポリシーを作成するときに、「Action Type」フィールドのドロップダウンリストに「Email」オプションが表示されないためです。

  [NSHELP-35065]

- クラスタ設定で、CLI または GUI を使用してパターンセットファイルを追加すると失敗します。

  [NSHELP-34996]

- GUI または NITRO API を使用すると、デフォルト以外のパーティションへのユーザーログインが失敗することがあります。

  [NSHELP-34849]

- HTTPD デーモンは、NITRO API バルクバインディングの HTTP GET リクエストの処理中に例外が発生するとクラッシュする可能性があります。

  [NSHELP-34399]

- NetScaler アプライアンスに 6 つ以上の管理パーティションが含まれている場合、NetScaler アプライアンスのバックアップおよび復元機能がアプライアンスを適切にバックアップしない場合があります。

  [NSHELP-34370]

- NetScaler GUI では、負荷分散ポリシーを UDP および SSL タイプの仮想サーバーにバインドすることはできません。これらのオプションは、**LB** Policy **Manager** ページのプロトコルにリストされていないためです。

  [NSHELP-33724]

- 保存された構成と実行中の構成に大きな違いがある場合、NetScaler UI に次のエラーが表示されます。

  「設定の取得中にエラーが発生しました」

  [NSHELP-32752]

- NetScaler で管理パーティション機能を構成し、セカンダリノードパーティション内で構成コマンドを継続的に実行すると、`save ns config` コマンドを使用してセカンダリノードパーティションに構成を保存できない場合があります。

  [NSHELP-31663]

- NetScaler GUI では、保存済み構成と実行構成画面（[システム] > [診断]）に、プレーンテキストではなく HTML タグが誤って表示されます。

  [NSHELP-27169]

- NetScaler GUI を使用すると、DTLS 負荷分散サービスのネットプロファイルを追加できないことがあります。

  [NSHELP-23676]

既知の問題

リリース 13.1—48.47 に存在する問題。

認証、承認、監査

- 認証仮想サーバーをデフォルト以外のパーティションで使用すると、NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-32054、NSCXLCM-640]

- 次の条件が満たされると、NetScaler がクラッシュします。

  - 401 ベースの証明書認証は、負荷分散仮想サーバーを介して行われます。
  - 認証仮想サーバーにバインドされた認証ポリシーはありません。
  - デバッグログは有効になっています。

  [NSAUTH-13259]

- 管理者は、認証情報が無効であることが原因で発生した認証エラーのカスタムロギングを実行できません。この問題は、NetScaler Responder ポリシーがログインエラーのエラーを検出できないために発生します。

  [NSAUTH-11151]

- ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
  `show adfsproxyprofile <profile name>`

  回避策：クラスター内のアクティブなプライマリの NetScaler に接続し、`show adfsproxyprofile <profile name>`コマンドを実行します。プロキシプロファイルの状態が表示されます。

  [NSAUTH-5916]

- 次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

  - [LDAP 到達可能性をテスト] オプションが開きます。
  - 無効なログイン認証情報が入力され、送信されます。
  - 有効なログイン認証情報が入力され、送信されます。

  回避策：「LDAP 到達可能性テスト」オプションを閉じて開きます。

  [NSAUTH-2147]

ボット管理

- BOT ポリシーが複雑なポリシールールを含むログアクションを使用すると、NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-34999]

負荷分散

- 高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

  [NSLB-7679]

- まれに、次の条件が満たされると、NetScaler アプライアンスがクラッシュしてコアダンプが生成されることがあります:

  - TCP ベースの DNS モニタープローブは、バックエンドサービスの監視に使用されます。
  - アプライアンスのメモリが不足しています。

  [NSHELP-35289]

- サービスグループの`entityofs`トラップの ServiceGroupName 形式は次のとおりです。
  `<service(group)name>?<ip/DBS>?<port>`

  トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (“?”) は区切り文字として使用されます。NetScaler は、疑問符 (” ?”)。このフォーマットは、NetScaler ADM GUI でも同じように表示されます。これは予想される動作です。

  [NSHELP-28080]

その他

- 高可用性セットアップで強制同期が行われると、アプライアンスはセカンダリノードで「set urlfilter parameter」コマンドを実行します。
  その結果、セカンダリノードは、「TimeOfDayToUpdateDB」パラメーターで指定された次のスケジュール時刻まで、スケジュールされた更新をスキップします。

  [NSSWG-849]

- クラスタ設定がアイドル状態の場合、ノード間メッセージング（NNM）は、指定された sndbuf サイズ（-S オプション付きの ping コマンド）の ping パケットに 20 ミリ秒の遅延を追加することがあります。

  [NSHELP-34774]

- レジストリ値が 2000 バイトを超えると、AlwaysOnAllow リストレジストリが期待どおりに動作しません。

  [NSHELP-31836]

- URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

  [NSHELP-22409]

NetScaler Gateway

- Chrome を使用する Mac デバイスで、2 つの FQDN にアクセス中に VPN 拡張機能がクラッシュします。

  [NSHELP-32144]

- カスタマイズされた EPA 障害ログメッセージは、NetScaler Gateway ポータルには表示されません。代わりに、「内部エラー」というメッセージが表示されます。

  [NSHELP-31434]

- ユーザーが Always-On サービスモードで Windows マシンにログインすると、Windows 自動ログオンが機能しないことがあります。マシントンネルはユーザートンネルに遷移しません。「Connecting…」というメッセージが VPN プラグイン UI に表示されます。

  [NSHELP-31357、CGOP-21192、NSCXLCM-612]

- `networkAccessOnVPNFailure`Always on プロファイルパラメーターを`fullAccess`から`onlyToGateway`に変更すると、ユーザーは NetScaler Gateway アプライアンスに接続できなくなります。

  [NSHELP-30236]

- 次の条件を満たす場合、Windows ログオン後に VPN プラグインはトンネルを確立しません。

  - NetScaler Gateway アプライアンスが常時オン機能用に構成されている
  - アプライアンスは、2 要素認証が「オフ」の証明書ベースの認証に設定されています

  [NSHELP-23584]

- スキーマをブラウズしているときに、「未定義のプロパティ「タイプ」を読み取れません」というエラーメッセージが表示されることがあります。

  [NSHELP-21897]

- NetScaler クラスター設定では、HDX Insight と Gateway Insight を同時に有効にすることはできません。

  [CGOP-23570]

- Windows OS オプションは、NetScaler GUI の事前認証ポリシーと認証アクションの「エクスプレッションエディタ」ドロップダウンリストに表示されません。ただし、GUI または CLI を使用して以前の NetScaler ビルドで Windows OS スキャンをすでに構成している場合は、アップグレードしても機能に影響はありません。必要に応じて CLI を使用して変更できます。

  回避方法:

  設定には CLI コマンドを使用します。

  - nFactor 認証で高度な EPA アクションを設定するには、次のコマンドを使用します。

    ```
    add authentication epaAction adv_win_scan -csecexpr "sys.
    client_expr("sys_0_WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows
    OS]")
    ```


  - 従来の事前認証アクションを設定するには、次のコマンドを使用します。

```
add aaa preauthenticationaction win_scan_action ALLOW
add aaa preauthenticationpolicy win_scan_policy "CLIENT.
SYSTEM('WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]')EXISTS
"win_scan_action
```

[CGOP-22966]

- Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

  [CGOP-19355]

- Gateway Insight レポートでは、SAML エラーエラーの認証タイプフィールドに「SAML」ではなく「Local」という値が誤って表示されます。

  [CGOP-13584]

- 高可用性セットアップでは、NetScaler フェイルオーバー中に、NetScaler ADM のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

  [CGOP-13511]

- EDT Insight 機能が有効になっていると、ネットワークの不一致時にオーディオチャネルに障害が発生することがあります。

  [CGOP-13493]

- NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

  [CGOP-11830]

- Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

  [CGOP-7269]

ネットワーク

- DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

  [NSNET-25299]

- DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

  - NetScaler BLX アプライアンスには、少数の「巨大ページ」が割り当てられています。たとえば、1G です。

- NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

この問題は、エラーメッセージとして「/var/log/ns.log」に記録されます。

- 「BLX-DPDK: DPDK メモリプールを PE-x 用に初期化できませんでした」

注:x はワーカープロセス数以下の数です。

回避策:「巨大ページ」を多数割り当ててから、アプライアンスを再起動します。

[NSNET-25173]

- DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

[NSNET-24449]

- DPDK を搭載した NetScaler BLX アプライアンスの Intel `X710 10G (i40e)` インターフェイスでは、次のインターフェイス操作はサポートされていません:

  - 無効化
  - 有効化
  - リセット

[NSNET-16559]

- Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

「次のパッケージの依存関係は満たされていません:blx-core-libs: i386: preDepends: libc6: i386 (>= 2.19) しかしインストールできません」

回避策: NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します:

  - dpkg –add-architecture i386
  - apt-get update
  - apt-get install libc6:i386

[NSNET-14602]

- FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

[NSNET-5233]

- NetScaler アプライアンスは、コールドリスタート後に「ColdStart」SNMP トラップメッセージを生成しない場合があります。

  [NSHELP-27917]

- NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

  [NSHELP-21082]

プラットフォーム

- NetScaler アプライアンスを 13.1〜4.x 以降のバージョンから次のバージョンのいずれかにダウングレードすると、一部の Python パッケージがインストールされません:

  - 任意の 11.1 ビルド
  - 12.1—62.21 およびそれ以前
  - 13.0-81.x およびそれ以前

  [NSPLAT-21691]

- Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。「rm cloudprofile」コマンドを使用してプロファイルを削除します。

  [NSPLAT-4520]

- Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
  回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノード上で設定する必要があります。

  [NSPLAT-4451]

- 高可用性セットアップのセカンダリ NetScaler SDX が共有 CPU コアで構成され、高可用性ハートビートが VLAN 経由で交換される場合、プライマリノードへの移行は失敗します。

  [NSHELP-32412]

ポリシー

- 処理データのサイズが設定されたデフォルトの TCP バッファーサイズを超えると、接続がハングアップすることがあります。

  回避策:TCP バッファサイズを処理する必要があるデータの最大サイズに設定します。

  [NSPOLICY-1267]

SSL

- NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

  回避方法:

  1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
  2. 構成を保存します。

  [NSSSL-9572]

- 認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

  [NSSSL-6478]

- 同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

  [NSSSL-6213]

- HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
  エラー:CRL 更新が無効です

  [NSSSL-6106]

- セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

  [NSSSL-4427]

- SSL プロファイル内の SSL プロトコルまたは暗号を変更しようとすると、「警告:SSL vserver/Service で使用可能な暗号が設定されていません」という誤った警告メッセージが表示されます。

  [NSSSL-4001]

- 期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

  [NSSSL-3184, NSSSL-1379, NSSSL-1394]

- OpenSSL 3.x ベースの TLS クライアントは、RFC 5746 (再ネゴシエーション指示拡張またはセキュア・リネゴシエーション) のサポートに関するクライアントのアドバタイズメントをサーバーが承認しない限り、ハンドシェイクを途中で終了します。フロントエンドの仮想サーバーは、再ネゴシエーションが無効になっている場合、このアドバタイズメントを無視し、接続障害の原因となります。今回の修正により、再ネゴシエーションが無効になっていてもフロントエンドの仮想サーバーがアドバタイズメントを確認するようになり、互換性が向上しました。

[NSHELP-35120]

システム

- CONNECT HTTP リクエストメソッドを使用する HTTP/2 ストリームが終了すると、HTTP/2 を使用する Web ページが完全に読み込まれないことがあります。

  [NSHELP-36407、NSBASE-17449]

- 次の条件がすべて満たされると、NetScaler がクラッシュする可能性があります:

  - イベント、監査ログ、またはメトリックは、分析プロファイルまたは AppFlow パラメータで有効になっています。
  - 応答側の書き換えポリシーが設定されます。

  [NSHELP-35550]

- 多要素認証が構成された NetScaler は、ポリシー評価中にクラッシュします。

  [NSHELP-33674]

- SSL サービスで構成された NetScaler アプライアンスは、アプライアンスが TCP FIN 制御パケットの後に TCP RESET 制御パケットを受信するとクラッシュします。

  [NSHELP-31656]

- 次の条件が満たされると、TCP 接続の RTT が高くなります:

  - 最大輻輳ウィンドウ（> 4 MB）が高く設定されている
  - TCP NILE アルゴリズムは有効になっています

  NetScaler アプライアンスが NILE アルゴリズムを使用して輻輳制御を行うには、条件がスロースタートのしきい値と最大輻輳ウィンドウを合わせた値を超える必要があります

  そのため、設定された最大輻輳時間帯に達するまで、NetScaler はデータを受け付け続け、最終的には RTT が高くなります。

  [NSHELP-31548]

- mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

  [NSBASE-18295]

- LogStream トランスポートタイプが Insight に設定されている場合、HDX Insight SkipFlow レコードでクライアント IP とサーバー IP が反転します。

  [NSBASE-8506]

ユーザーインターフェイス

- NetScaler GUI では、「ダッシュボード」タブの下にある「ヘルプ」リンクが壊れています。

  [NSUI-14752]

- CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

  回避策: NetScaler GUI または CLI を使用して IPsec プロファイル、IP トンネル、および PBR ルールを追加して CloudBridge Connector を構成します。

  [NSUI-13024]

- GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

  [NSUI-6838]

- NetScaler BLX アプライアンスの高可用性セットアップでは、プライマリノードが応答しなくなり、CLI または API 要求がブロックされることがあります。

  回避策: プライマリノードを再起動します。

  [NSCONFIG-6601]

- あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

  1. NetScaler アプライアンスをいずれかのビルドにアップグレードします
     - 13.0 52.24 ビルド
     - 12.1 57.18 ビルド
     - 11.1 65.10 ビルド
  2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
  3. NetScaler アプライアンスを古いビルドにダウングレードします。

  CLI を使用してこれらのシステムユーザーのリストを表示するには、
  コマンドプロンプトで次のように入力します。

  ```
  query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
  ```

  回避策: この問題を解決するには、以下のいずれかの独立したオプションを使用してください。

  - NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3）、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
  - アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。

- 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳細については、「ルート管理者 (nsroot) パスワードをリセットする方法」を参照してください。

[NSCONFIG-3188]

NetScaler 13.1-45.64 リリースのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリース Build 13.1-45.64 の機能強化と変更、修正された問題と既知の問題について説明します。

メモ

- このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティに関する修正とアドバイスの一覧については、Citrix セキュリティ情報を参照してください。
- ビルド 13.1-45.61 以降のビルドは、CTX477714 で説明されているセキュリティの脆弱性に対処します。
- ビルド 13.1-45.64 がビルド 13.1-45.61 とビルド 13.1-45.63 に置き換わります。ただし、Build 13.1-45.61 にアップグレードした場合、構成が失われる可能性があります。修復手順については、CTX547038 を参照してください 。
- ビルド 13.1-45.63 には、NSSSL-12761 と NSHELP-35058 の修正に加えて、ビルド 13.1-45.61 で利用できるすべての拡張機能とバグ修正が含まれています。
- ビルド 13.1-45.64 には、NSBASE-18162（NSHELP-35288）の修正と、ビルド 13.1-45.63 で利用可能なすべての機能強化とバグ修正が含まれています。

新機能

ビルド 13.1-45.64 で利用できる機能強化と変更点。

NetScaler SDX アプライアンス

- **SDX** アプライアンスのアップグレード中の追加チェック

  これで、管理サービスから XenServer/Citrix Hypervisor への Secure Shell（SSH）接続に障害が発生した場合、NetScaler SDX アプライアンスのアップグレードは許可されません。

  [NSSVM-5114]

- 管理者プロファイルの作成時にパスワードの複雑さを有効または無効にする

  NetScaler SDX アプライアンスは、GUI または CLI を使用して VPX インスタンスのパスワードの複雑性を有効または無効にできるようになりました。

  - パスワードの複雑性が有効になっている場合、パスワードの最小長は 4 文字ですが、以前は 6 文字でした。
  - パスワードの複雑度が無効になっている場合、必要なパスワードの最小長は 1 文字です。

  [NSSVM-4889]

NetScaler Web App Firewall

- **NetScaler Web App Firewall**、ボット、**IP** レピュテーションのプロキシ認証の設定

  NetScaler Web App Firewall の署名更新、ボット署名の更新、およびレピュテーション更新のプロキシ認証を構成できるようになりました。プロキシ認証は、アプライアンスのセキュリティをさらに強化します。プロキシ認証が有効になっている NetScaler アプライアンスは、インターネットからアップデートをダウンロードする前に、プロキシサーバーで認証を行います。これにより、アプライアンスを悪意のあるダウンロードから保護できます。
  プロキシ認証を設定するには、次のセキュリティ機能の設定でプロキシのユーザー名とパスワードを指定します。

  - NetScaler Web App Firewall。詳細については、「エンジン設定」を参照してください。
  - ボット。詳細については、「ボット検出」を参照してください。
  - IP レピュテーション。詳細については、「IP レピュテーション」を参照してください。

  [NSWAF-9532]

- **apache_mode** 属性は廃止されました

  `add appfw profile`コマンドの`invalidPercentHandling`パラメータの`apache_mode`属性は廃止されました。

  [NSWAF-4110]

負荷分散

- カスタムエントリの最大数の増加

  IP アドレス範囲のロケーション修飾子を指定するために、最大 3000 のカスタムロケーションエントリを追加できるようになりました。これらのエンティティは、GSLB の静的近接方式やロケーションマッチポリシーで使用されます。

  詳細については、「静的近接データベースへのカスタムエントリの追加」を参照してください。

  [NSLB-9755]

ネットワーク

- **NetScaler BLX** アプライアンスの自動構成サポート

  NetScaler BLX アプライアンスには次の自動構成機能が追加されました。

  - NetScaler BLX アプライアンスは、すべての Linux ホスト NIC ポートをアプライアンスの専用ポートとして自動的に追加するように構成できます。この自動構成では、`blx-managed-host` を 1 に設定し、NetScaler BLX 構成ファイル () **`interface`** のパラメーターを含む両方の行にコメントする必要があります。`blx.conf` アプライアンスは、すべての Linux ホスト NIC ポートを専用ポートとして自動的に追加します。また、アプライアンスは DPDK 互換の NIC ポートを自動的に検出し、Linux ホスト上の DPDK VFIO モジュールにバインドします。
  - NetScaler BLX アプライアンスを専用モードで構成して、アプライアンスの NSIP アドレスとデフォルトゲートウェイを自動的に設定できます。この自動構成では、`blx-managed-host`を 1 に設定し、NetScaler BLX 構成ファイル（`blx.conf`）内の`ipaddress`および**`default`**パラメーターを含む行にコメントする必要があります。アプライアンスは、Linux ホストで最も優先順位の高いゲートウェイルートを持つデフォルトポートとして、専用の NIC ポートの 1 つを選択します。デフォルトポートの IP アドレスとデフォルトゲートウェイは、NetScaler BLX アプライアンスの NSIP アドレスとデフォルトゲートウェイとして設定されます。

  [NSNET-27468]

- **NetScaler BLX** アプライアンスに対する **RHEL** バージョン **9.x** のサポート

  NetScaler BLX アプライアンスは、Red Hat Enterprise Linux (RHEL) バージョン 9.x プラットフォームでサポートされるようになりました。

  [NSNET-27421]

ポリシー

- **NetScaler BLX** および **CPX** アプライアンスで **NSPEPI** ツールを使用する機能

  NSPEPI と無効チェック構成ツールが NetScaler CPX および BLX アプライアンスでサポートされるようになりました。

  [NSPOLICY-4872]

SSL

- 不明なサーバー名で **SSL** ハンドシェイクを続行

  NetScaler アプライアンスでは、不明なサーバー名でも SSL ハンドシェイクを続行できるようになり、ハンドシェイクの削除または完了の決定はクライアントに任されています。

以前、アプライアンスは、不明なサーバー名のクライアント hello を受信したときに SSL ハンドシェイクを終了していました。

[NSSSL-10918]

システム

- **HTTP PUT** リクエストメソッドの圧縮サポート
  NetScaler アプライアンスは、PUT リクエストメソッドを使用する HTTP リクエストについて、サーバーから受信した HTTP レスポンスを圧縮するようになりました。

  [NSHELP-32695]

- メトリックコレクターのエクスポート頻度の設定

  デフォルトでは、メトリクスコレクターは 30 秒ごとの時系列分析データのエクスポートをサポートします。時系列分析プロファイルデータを NetScaler からエクスポートする間隔を決定できるように、30 秒から 300 秒までの値として構成できるようになりました。

  [NSBASE-17561]

- 監査ログの **Splunk** への直接エクスポートのサポート

  監査ログを使用すると、NetScaler のさまざまなモジュールによって収集された NetScaler の状態とステータス情報を記録できます。監査ログを NetScaler から Splunk にエクスポートして、トラブルシューティングに役立つ有意義な洞察を得ることができます。この機能により、Splunk が提供する HTTP イベントコレクターを使用して、監査ログを HTTP（または HTTPS）経由で NetScaler から Splunk に直接送信できます。

  [NSBASE-17559]

- ウェブソケット **HTTP/2** 接続マルチプレクシングのサポート

  NetScaler アプライアンスは、WebSocket 接続の多重化をサポートするようになりました。WebSocket 接続は HTTP/2 経由でサポートされています。CLI または GUI を使用して WebSocket 接続を有効にできます。

  [NSBASE-17307]

解決された問題

ビルド 13.1-45.64 で対処されている問題。

AppFlow

- NetScaler インスタンスのメトリックコレクターが断続的に応答を停止します。その結果、メトリックスコレクターが応答を停止しても、1 間隔 (30 秒) の分析データがエクスポートされないことがあります。

  [NSHELP-34048]

認証、承認、監査

- GSLB が有効になっている一部の NetScaler アプライアンスでは、URL 計算が無効なため、認証仮想サーバーから負荷分散仮想サーバーへのリダイレクトが失敗します。

  [NSHELP-33459]

- NetScaler を OpenID プロバイダー（OAuth IdP）として使用し、GSLB がそれを使用して構成されている場合、依存パーティ（RP）での OAuth 認証はトークンの検証中に失敗し、OAuth リレーパーティ（RP）での認証が失敗する可能性があります。

  [NSHELP-33455]

- NetScaler アプライアンスが SAML サービスプロバイダーとして構成され、SSL 証明書が更新されると、クラッシュする可能性があります。

  [NSHELP-33243、NSHELP-32966、NSHELP-33242、NSHELP-34366]

- NetScaler アプライアンスでの OAuth 認証は、トークンの解析に関する問題により失敗します。

  [NSHELP-31573]

ボット管理

- NetScaler アプライアンスは、IP レピュテーション機能が無効になっている場合、IP データベースデータのダウンロードを試みます。

  [NSHELP-34488]

キャッシュ

- キャッシュされたオブジェクトの Cache-Control ヘッダーの Max-Age 値がバックエンドサーバーで変更されると、NetScaler アプライアンスが再起動することがあります。

  [NSHELP-34078]

- クラスタ設定で CLIP アドレスを使用してクラスタ設定にアクセスすると、GUI または CLI に表示されるキャッシュグローバルポリシー情報が不完全になります。

  [NSCACHE-521]

NetScaler SDX アプライアンス

- 管理サービスダッシュボードから「コア割り当て」にアクセスしようとすると、NetScaler SDX アプライアンスがクラッシュすることがあります。

  [NSHELP-34537]

- VPX インスタンスに割り当てられた非対称暗号ユニット（ACU）と対称暗号ユニット（SCU）がパケットエンジン（PE）コアの倍数でない場合、NetScaler SDX アプライアンスが期待どおりに動作しないことがあります。つまり、1000* 個の PE コア数です。

  [NSHELP-34389]

- 管理サービス UI から VPX インスタンスのプロパティのいずれかを編集しているときに、管理サービス（SVM）がクラッシュする可能性があります。

  [NSHELP-34297]

- [構成**] > [**システム**] > [**セットアップウィザード**] > [**管理ネットワーク**] > [**サポート **IP** の編集**]** に移動して **NetScaler SDX** アプライアンスのサポート **IP** アドレスを変更しようとすると、変更を保存できません。プロンプトで「はい」をクリックすると、変更が停止します。未定義の参照エラーがブラウザに表示されます。

  修正: 参照する前に未定義のオブジェクトを確認してください。

  [NSHELP-34141]

NetScaler Gateway

- アップグレード後、HDX Insight が有効になっていると、NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-35058]

- アップグレード後、RDP プロキシ接続を起動すると NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-33420]

- VPN セッションアクションを再設定すると、VPN セッションアクションの Always On プロファイルは設定解除されます。

  [NSHELP-33396]

- アップグレード後、NetScaler アプライアンスは最初の HA 同期中にクラッシュする可能性があります。

  [NSHELP-32957]

NetScaler Web App Firewall

- Web App Firewall の署名ルールに次のオブジェクトのいずれかが含まれている場合、HA 展開中に NetScaler アプライアンスがクラッシュする可能性があります。

  - Patsets
  - データセット
  - 文字列マップ

- 名前付き表現

[NSHELP-34338]

- 緩和ルールをエクスポートすると、ダウンロードに時間がかかり、ファイルが完全にダウンロードされません。この問題は、ファイルサイズが 5 MB を超える場合に発生します。

  [NSHELP-34044]

- Web App Firewall ポリシーが仮想サーバー上で更新されると、次の問題が発生します。

  - NetScaler GUI と CLI が応答しなかったか、通常よりも時間がかかりました。
  - パケットの CPU 使用率が 100% に増加しました
  - パーシスタンスセッションの数が増えました。

  [NSHELP-33975]

- JSON コマンドインジェクション緩和ルールは、緩和ルールにセミコロン (;) またはピリオド (.) が含まれていると機能しない場合があります。

  [NSHELP-33606]

負荷分散

- 次の条件が満たされ、すべてのサービスをバインド解除して再度バインドすると、NetScaler アプライアンスがクラッシュします。

  - 負荷分散仮想サーバーは、ハッシュベースの方法で構成されます。
  - サービスは優先的にこの仮想サーバーにバインドされます。

  [NSHELP-34314]

- HA セットアップでは、複数の仮想サーバーにバインドされているサービスグループを削除すると、NetScaler アプライアンスがクラッシュします。

  [NSHELP-34029]

- NetScaler アプライアンスの負荷分散構成を追加または変更すると、次のエラーが表示されることがあります。

  構成に一貫性がない可能性があります。「show configstatus」コマンドで確認するか、再起動してください。

  この問題は、set lb vserver コマンドを httpsRedirectURL パラメーターと RedirectFromPort パラメーターとともに使用した場合に発生します。

  [NSHELP-33912]

- まれに、nsmap がクラッシュすることがあります。その結果、位置情報データベースを使用する一部の NetScaler アプライアンスが意図したとおりに機能しない場合があります。

  [NSHELP-33840]

- 高可用性設定でサービスを無効にしてから有効にすると、フェイルオーバーが発生したときに一部のモニターが SKIP_OFS 状態になることがあります。

  [NSHELP-33717]

- show cs vserver コマンドでは、パラメータがコンテンツスイッチングポリシーで設定され、コンテンツスイッチング仮想サーバにバインドされていても、ルールパラメータは表示されません。

  [NSHELP-33506]

- 接続ミラーリング中に、書き換えポリシーが 30 バイトを超えると、NetScaler アプライアンスがクラッシュします。

  [NSHELP-32902]

- 帯域幅の使用量が設定された制限内であっても、SNMP アラートが生成されます。この問題は、2 つの異なるデータ型を比較し、インクリメントするときに 1 つのパラメーターがラップアラウンドする場合に発生します。

  [NSHELP-32509]

- 接続ミラーリングが設定された NetScaler アプライアンスは、ジャンボパケットが送信されるとクラッシュします。

  [NSHELP-31072]

- NetScaler VPX アプライアンスは、次の条件が満たされるとクラッシュします。

  1. autosync オプションは設定を他の GSLB サイトと同期させるために使われます。
  2. GSLB キャッシュの取得に使用されるインカネーション番号は 1024 の倍数です。

  [NSHELP-30075]

- GSLB 設定では、SSL 証明書が下位サイトに見当たりません。この問題は、自動同期オプションが有効になっていて、下位サイトにマスターサイトでは使用できない SSL 証明書がある場合に発生します。

  [NSHELP-29309]

その他

- NetScaler アプライアンスで「ns_hw_err.bash」スクリプトを実行すると、次のエラーメッセージが表示されます。
  「エラー：ファイル‘ns_hw_plugins.py’を開けません:[Errno 2] そのようなファイルまたはディレクトリはありません」

  [NSHELP-32991]

- クラスタ設定では、クラスタ IP アドレスが NSIP アドレスのサブネットとは異なるサブネットに設定されている場合、ファイルの自動同期は失敗します。

  [NSHELP-29988]

プラットフォーム

- ADC アプライアンスをリリース 13.1 ビルド 42.47 にアップグレードした後、一部のパブリッククラウド VPX 展開では、HTTP と TCP サービスのアップ状態とダウン状態の間でフラップが発生する場合があります。

  [NSPLAT-26310]

- BMC ファームウェアバージョン 4.08 を実行している SDX アプライアンスで、13.0 ビルド 84.X から 1 つのバンドルアップグレードを実行すると、システム起動中に Lightout Management（LOM）ファームウェアの 4.14 へのアップグレードが断続的に停止し、30 分後にタイムアウトすることがあります。

  [NSPLAT-26148]

- AWS クラウド上の NetScaler VPX インスタンスの HA セットアップでは、/var/log/の場所に保存されている「cloud-ha-daemon.log」ファイルのコンテンツが 1 回ではなく 2 回印刷されます。

  [NSPLAT-25687]

- NetScaler SDX アプライアンスでは、トラフィックのバーストを処理するのに十分なスループットが SDX アプライアンスに用意されている場合でも、VPX インスタンスはバーストモードの一部として構成された最小スループット値で動作する場合があります。

  [NSHELP-33875、NSHELP-34667]

- NetScaler MPX 9100、MPX 9100T、MPX 16000、および MPX 16000T プラットフォームでは、ライセンスホスト ID が変更されると、アプライアンスがライセンスなしで起動することがあります。

  [NSHELP-33745、NSHELP-33756、NSHELP-33801]

SSL

- 証明書とキーのペアと ECC カーブを SSL サービス、サービスグループ、または内部サービスにバインドするコマンドは、設定 (ns.conf) に保存されません。

  [NSSSL-12761]

- リリース 13.1 ビルド 37.x にアップグレードすると、構成が変更されていなくても、TLSv1.0 プロトコルを使用してネゴシエートできなくなる場合があります。

  [NSHELP-34345]

システム

- NetScaler アプライアンスによって圧縮された HTTP 応答は、Content-Length HTTP 応答ヘッダーフィールドの値に先頭にスペース文字が追加されるため、一部の HTTP (S) クライアントで障害が発生する可能性があります。

  [NSHELP-34660]

- すべての HTTP ヘッダーをログに記録するように構成された NetScaler アプライアンスは、20 を超える長いヘッダーを含む HTTP 要求または応答を受信するとクラッシュします。

  [NSHELP-34145]

- 管理パーティションに rest タイプの AppFlow コレクターが設定されている場合、NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-33600]

- NetScaler バージョン 13.1 ビルド 33.47 以降では、GUI または CLI を使用してイベント、指標、および監査ログパラメーターを有効または無効にすることはできません。

  [NSHELP-33247]

- 以下の条件が満たされると、gRPC クライアントは gRPC ステータスヘッダーの解析に失敗します。

  - gRPC ステータスヘッダーは、末尾ヘッダーだけに追加されるのではなく、先頭ヘッダーと末尾ヘッダーの両方に追加されます。

  [NSHELP-31640]

- 次の両方の条件が満たされると、NetScaler アプライアンスでメモリリークが発生する可能性があります。

  - HTTP 圧縮機能が有効になっています。
  - 接続はトランザクションの途中でリセットされます。

  [NSHELP-30631]

- HTTP/2 対応の仮想サーバーがバックエンドサービスに要求を転送するのではなく、HTTP/2 要求に対する応答を生成すると、Citrix ADC アプライアンスがクラッシュする可能性があります。

  [NSBASE-18162、NSHELP-35288]

- NetScaler アプライアンスからクライアントへのヘッダーのみの gRPC 応答には、gRPC ステータスおよび gRPC メッセージは含まれません。

  [NSBASE-17802]

ユーザーインターフェイス

- 管理パーティションを使用している場合、GUI を使用して SSL 証明書を削除することはできません。

  [NSHELP-34429]

- NetScaler GUI の［コンテンツスイッチングポリシーバインドの設定］ページの［バインド先］列には、ポリシーがバインドされているコンテンツスイッチ仮想サーバーの実際の名前の代わりに、「CS Virtual Server」という文字列が表示されます。

  [NSHELP-34374]

- NetScaler GUI を使用すると、HTTP プロファイルの代替サービスを構成できないことがあります。

  [NSHELP-34304]

- AppFW プロファイルをログ表現にバインドすると、state パラメータはデフォルトで有効に設定されます。ただし、システムをアップグレードすると、パラメータは無効に戻されます。

  [NSHELP-34187]

- NetScaler GUI の [診断] ページ（[システム] > [診断]）にあるコアファイルのダウンロードは、エラーで失敗することがあります。

  [NSHELP-33644]

- NetScaler GUI では、特定のタイプの SNMP トラップの編集ボタンをクリックすると、特定タイプの SNMP トラップの代わりに汎用タイプの SNMP トラップの詳細が表示されます。

  [NSHELP-33520]

- NITRO Python SDK GET を名前で呼び出すと、次のリソースに対して「割り当て前にローカル変数 'response' が参照されました」というエラーメッセージが表示されて失敗します。

  - `appfwhtmlerrorpage`
  - `appfwjsonerrorpage`
  - `appfwprotofile`
  - `appfwsignatures`
  - `appfwwsdl`
  - `appfwxmlerrorpage`
  - `appfwxmlschema`
  - `botsignature`
  - `responderhtmlpage`

  [NSHELP-32525]

- クラスタ設定では、CLIP アドレスに対して実行される`show HTTP monitor`操作では、複数値の HTTP 応答コードは表示されません。

  [NSCONFIG-7107]

既知の問題

リリース 13.1-45.64 に存在する問題。

認証、承認、監査

- 認証仮想サーバーをデフォルト以外のパーティションで使用すると、NetScaler アプライアンスがクラッシュする可能性があります。

[NSHELP-32054]

- 管理者は、認証情報が無効であることが原因で発生した認証エラーのカスタムロギングを実行できません。この問題は、NetScaler Responder ポリシーがログインエラーのエラーを検出できないために発生します。

  [NSAUTH-11151]

- ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
  `show adfsproxyprofile <profile name>`

  回避策：クラスター内のアクティブなプライマリの NetScaler に接続し、`show adfsproxyprofile <profile name>`コマンドを実行します。プロキシプロファイルの状態が表示されます。

  [NSAUTH-5916]

- 次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

  - [LDAP 到達可能性をテスト] オプションが開きます。
  - 無効なログイン認証情報が入力され、送信されます。
  - 有効なログイン認証情報が入力され、送信されます。

  回避策：「LDAP 到達可能性テスト」オプションを閉じて開きます。

  [NSAUTH-2147]

NetScaler Gateway

- Citrix Secure Access クライアントでスプリットトンネルがオフに設定されていると、なりすまし IP アドレス範囲と重複するイントラネットリソースにアクセスできません。

  [NSHELP-34334]

- ゲートウェイサーバーにアクセスできるため、常時接続の VPN 接続が起動時に断続的に失敗します。

  [NSHELP-33500]

- Citrix Secure Access 関連のレジストリ値が 1500 文字を超える場合、ログコレクターはエラーログを収集できません。

  [NSHELP-33457]

- Windows フィルタリングプラットフォーム (WFP) ドライバを使用している場合、VPN の再接続後にイントラネットアクセスが機能しないことがあります。

  [NSHELP-32978]

- Citrix Secure Access クライアントのバージョン 21.7.1.2 以降では、管理者権限のないユーザーが新しいバージョンにアップグレードできません。この問題は、Citrix Secure Access クライアントのアップグレードが NetScaler アプライアンスから行われる場合にのみ発生します。

[NSHELP-32793]

- ユーザーが Windows 向け Citrix Secure Access 画面の［ホームページ］タブをクリックすると、ページに接続拒否エラーが表示されます。

  [NSHELP-32510]

- Chrome を使用する Mac デバイスで、2 つの FQDN にアクセス中に VPN 拡張機能がクラッシュします。

  [NSHELP-32144]

- NetScaler Gateway 13.0 または 13.1 のプロキシ設定が空の場合、Citrix SSO が不適切なプロキシ設定を作成することがあります。

  [NSHELP-31970]

- Citrix Secure Access によって確立されたトンネルの外部にあるリソースへの直接接続は、大幅な遅延または輻輳が発生すると失敗することがあります。

  [NSHELP-31598]

- カスタマイズされた EPA 障害ログメッセージは、NetScaler Gateway ポータルには表示されません。代わりに、「内部エラー」というメッセージが表示されます。

  [NSHELP-31434]

- ユーザーが Always-On サービスモードで Windows マシンにログインすると、Windows 自動ログオンが機能しないことがあります。マシントンネルはユーザートンネルに遷移しません。「Connecting…」というメッセージが VPN プラグイン UI に表示されます。

  [NSHELP-31357、CGOP-21192、NSHELP-34211]

- Always on が設定されている場合、aoservice.exe ファイルのバージョン番号（1.1.1.1）が正しくないため、ユーザトンネルが失敗します。

  [NSHELP-30662]

- ［NetworkAccessonVPNFailure］プロファイルパラメーターを［フルアクセス］から［OnlyToGateway］に変更すると、ユーザーは NetScaler Gateway アプライアンスに接続できなくなります。

  [NSHELP-30236]

- ゲートウェイのホームページは、ゲートウェイプラグインが VPN トンネルを正常に確立した直後には表示されません。この問題を解決するために、次のレジストリ値が導入されます。

  HKLMSoftwareCitrixSecure Access ClientSecureChannelResetTimeoutSeconds
  タイプ: DWORD

  デフォルトでは、このレジストリ値は設定も追加もされません。「SecureChannelResetTimeoutSeconds」の値が 0 または追加されていない場合、遅延を処理するための修正が機能しません。これはデフォルトの動作です。管理者は、このレジストリをクライアントに設定して修正を有効にする必要があります (つまり、ゲートウェイプラグインが VPN トンネルを正常に確立した直後にホームページを表示する)。

[NSHELP-30189]

- Windows VPN クライアントは、サーバからの「SSL close notify」アラートを尊重せず、同じ接続で転送ログイン要求を送信します。

  [NSHELP-29675]

- macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

  [NSHELP-28551]

- クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

  [NSHELP-28404]

- 次の条件を満たす場合、Windows ログオン後に VPN プラグインはトンネルを確立しません。

  - NetScaler Gateway アプライアンスが常時オン機能用に構成されている
  - アプライアンスは、2 要素認証を「オフ」にした証明書ベースの認証用に設定されています。

  [NSHELP-23584]

- スキーマをブラウズしているときに、「未定義のプロパティ「タイプ」を読み取れません」というエラーメッセージが表示されることがあります。

  [NSHELP-21897]

- NetScaler クラスター設定では、HDX Insight と Gateway Insight を同時に有効にすることはできません。

  [CGOP-23570]

- Windows OS オプションは、NetScaler GUI の事前認証ポリシーと認証アクションの「エクスプレッションエディタ」ドロップダウンリストに表示されません。ただし、GUI または CLI を使用して以前の NetScaler ビルドで Windows OS スキャンをすでに構成している場合は、アップグレードしても機能に影響はありません。必要に応じて CLI を使用して変更できます。

  回避方法:

  設定には CLI コマンドを使用します。

  - nFactor 認証で高度な EPA アクションを設定するには、次のコマンドを使用します。
    add authentication epaAction adv_win_scan -csecexpr “sys.client_expr(“sys_0_WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]”)”
  - 従来の事前認証アクションを設定するには、次のコマンドを使用します。
    add aaa preauthenticationaction win_scan_action ALLOW
    add aaa preauthenticationpolicy win_scan_policy “CLIENT.SYSTEM(‘WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]’) EXISTS”win_scan_action

  [CGOP-22966]

- Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

  [CGOP-19355]

- Gateway Insight レポートでは、SAML エラーエラーの認証タイプフィールドに「SAML」ではなく「Local」という値が誤って表示されます。

  [CGOP-13584]

- 高可用性セットアップでは、NetScaler フェイルオーバー時に、NetScaler Console のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

  [CGOP-13511]

- ICA 接続が MAC Receiver バージョン 19.6.0.32 または Citrix Virtual Apps and Desktops バージョン 7.18 から起動されると、HDX Insight 機能は無効になります。

  [CGOP-13494]

- EDT Insight 機能が有効になっていると、ネットワークの不一致時にオーディオチャネルに障害が発生することがあります。

  [CGOP-13493]

- ブラウザからローカルホスト接続を受け付けると、macOS の Accept Connection ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

  [CGOP-13050]

- 一部の言語では、Citrix SSO アプリ > ホームページの「ホームページ」というテキストが切り捨てられます。

  [CGOP-13049]

- NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

  [CGOP-11830]

- Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

  [CGOP-7269]

負荷分散

- 高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

  [NSLB-7679]

- サービスグループのentityofsトラップの ServiceGroupName 形式は次のとおりです。
  <service(group)name>?<ip/DBS>?<port>

  トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (“?”) は区切り文字として使用されます。NetScaler は、疑問符（” ?”)。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

  [NSHELP-28080]

その他

- 高可用性セットアップで強制同期が行われると、アプライアンスはセカンダリノードで「set urlfilter parameter」コマンドを実行します。
  その結果、セカンダリノードは、「TimeOfDayToUpdateDB」パラメーターで指定された次のスケジュール時刻まで、スケジュールされた更新をスキップします。

  [NSSWG-849]

- レジストリ値が 2000 バイトを超えると、AlwaysOnAllow リストレジストリが期待どおりに動作しません。

  [NSHELP-31836]

- URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

  [NSHELP-22409]

ネットワーク

- DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

  [NSNET-25299]

- DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

  - NetScaler BLX アプライアンスには、少数の「巨大ページ」が割り当てられています。たとえば、1G です。
  - NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

  この問題は、エラーメッセージとして「/var/log/ns.log」に記録されます。

  -「BLX-DPDK: DPDK メモリプールを PE-x 用に初期化できませんでした」

注:x はワーカープロセス数以下の数です。

回避策:「巨大ページ」を多数割り当ててから、アプライアンスを再起動します。

[NSNET-25173]

- DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

  [NSNET-24449]

- DPDK を搭載した NetScaler BLX アプライアンスの Intel `X710 10G (i40e)` インターフェイスでは、次のインターフェイス操作はサポートされていません:

  - 無効化
  - 有効化
  - リセット

  [NSNET-16559]

- Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

  「次のパッケージの依存関係は満たされていません:blx-core-libs: i386: preDepends: libc6: i386 (>= 2.19) しかしインストールできません」

  回避策: NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します:

  - dpkg –add-architecture i386
  - apt-get update
  - apt-get install libc6:i386

  [NSNET-14602]

- FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

  [NSNET-5233]

- NetScaler アプライアンスは、コールドリスタート後に「ColdStart」SNMP トラップメッセージを生成しない場合があります。

  [NSHELP-27917]

- NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

  [NSHELP-21082]

プラットフォーム

- NetScaler アプライアンスを 13.1～4.x 以降のバージョンから次のバージョンのいずれかにダウングレードすると、一部の Python パッケージがインストールされません：

  - 任意の 11.1 ビルド
  - 12.1-62.21 およびそれ以前
  - 13.0-81.x およびそれ以前

  [NSPLAT-21691]

- Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。「rm cloudprofile」コマンドを使用してプロファイルを削除します。

  [NSPLAT-4520]

- Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
  回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノード上で設定する必要があります。

  [NSPLAT-4451]

- メンバーインターフェイスが設定されていない LA チャネルに VRID をバインドすると、NetScaler アプライアンスがクラッシュします。

  回避策：VRID を LA チャネルにバインドする前に、LA チャネルのメンバーインターフェイスを設定します。

  [NSPLAT-26707]

ポリシー

- 処理データのサイズが設定されたデフォルトの TCP バッファーサイズを超えると、接続がハングアップすることがあります。

  回避策:TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

  [NSPOLICY-1267]

SSL

- NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

  回避方法：

1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
2. 構成を保存します。

[NSSSL-9572]

- 認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

  [NSSSL-6478]

- 同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

  [NSSSL-6213]

- HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
  エラー:CRL 更新が無効です

  [NSSSL-6106]

- セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

  [NSSSL-4427]

- SSL プロファイル内の SSL プロトコルまたは暗号を変更しようとすると、「警告:SSL vserver/Service で使用可能な暗号が設定されていません」という誤った警告メッセージが表示されます。

  [NSSSL-4001]

- 期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

  [NSSSL-3184, NSSSL-1379, NSSSL-1394]

システム

- CONNECT HTTP リクエストメソッドを使用する HTTP/2 ストリームが終了すると、HTTP/2 を使用する Web ページが完全に読み込まれないことがあります。

  [NSHELP-36407、NSBASE-17449]

- 次の条件が満たされると、TCP 接続の RTT が高くなります:
  - 最大輻輳ウィンドウ（> 4 MB）が高く設定されている
  - TCP NILE アルゴリズムは有効になっています

NetScaler アプライアンスが NILE アルゴリズムを使用して輻輳制御を行うには、条件がスロースタートのしきい値と最大輻輳ウィンドウを合わせた値を超える必要があります

そのため、設定された最大輻輳時間帯に達するまで、NetScaler はデータを受け付け続け、最終的には RTT が高くなります。

[NSHELP-31548]

- アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

  [NSHELP-21240]

- mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

  [NSHELP-10972]

- クラスタ展開では、CCO 以外のノードで「force cluster sync」コマンドを実行すると、ns.log ファイルには重複するログエントリが含まれます。

  [NSBASE-16304、NSGI-1293]

- NetScaler Console を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに機能しません。

  回避策：管理ポッドを再起動します。

  [NSBASE-15556]

- LogStream トランスポートタイプが Insight 用に構成されている場合、クライアント IP とサーバー IP は HDX Insight SkipFlow レコードで反転されます。

  [NSBASE-8506]

ユーザーインターフェイス

- NetScaler GUI では、「ダッシュボード」タブの下にある「ヘルプ」リンクが壊れています。

  [NSUI-14752]

- CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

  回避策：NetScaler GUI または CLI を使用して IPSec プロファイル、IP トンネル、および PBR ルールを追加して CloudBridge Connector を構成します。

  [NSUI-13024]

- GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

  [NSUI-6838]

- NetScaler BLX アプライアンスの高可用性セットアップでは、プライマリノードが応答しなくなり、CLI または API 要求がブロックされることがあります。

  回避策: プライマリノードを再起動します。

  [NSCONFIG-6601]

- あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

  1. NetScaler アプライアンスをいずれかのビルドにアップグレードします
     - 13.0 52.24 ビルド
     - 12.1 57.18 ビルド
     - 11.1 65.10 ビルド
  2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
  3. NetScaler アプライアンスを古いビルドにダウングレードします。

  CLI を使用してこれらのシステムユーザーのリストを表示するには、
  コマンドプロンプトで次のように入力します。

  ```
  query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
  ```

  回避策: この問題を解決するには、以下のいずれかの独立したオプションを使用してください。

  - NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3）、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
  - アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
  - 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

  詳細については、https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.htmlを参照してください。

  [NSCONFIG-3188]

NetScaler 13.1-42.47 リリースのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリースビルド 13.1-42.47 の機能強化と変更、修正された問題と既知の問題について説明します。

メモ

- このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正と勧告のリストについては、セキュリティ速報を参照してください。

新機能

ビルド 13.1-42.47 で利用できる機能強化と変更。

ボット管理

- ボット設定での **IP** レピュテーションのダウンロード停止のサポート

  IP レピュテーション機能を無効にしたら、****NetScaler** のボット管理設定でデフォルトの非侵入プロファイルを **BOT_BYPASS** に設定します ****** 。この設定では、IP レピュテーションのダウンロードが停止します。

  ボット管理設定を変更するには、[セキュリティ**] > [NetScaler bot** 管理**] > [NetScaler bot** 管理設定の変更] に移動します。

  [NSBOT-1050、NSHELP-34310、NSHELP-33835、NSHELP-34410]

- **NetScaler** コンソールの **GUI** に新しいボット違反が表示される

  NetScaler コンソールの GUI では、以下のボット違反が新たに導入されました:

  - ユーザーエージェントヘッダーなし
  - 複数のユーザーエージェントヘッダー

  アプリケーションサーバーは、ユーザーエージェントヘッダー情報を使用して、受信したリクエストの詳細を知ります。一部のボットリクエストには、複数のユーザーエージェントヘッダーがある場合や、ユーザーエージェントヘッダーがない場合があります。このようなボット違反は、NetScaler のボット管理プロファイルを使用して検出できます。次に、NetScaler コンソール GUI を使用してボット違反を監視します。詳細については、「違反カテゴリ」を参照してください。

  [NSBOT-1023]

NetScaler SDX アプライアンス

- **SD-WAN** サポートは管理サービスから廃止されました

  リリース 13.1 ビルド 42.x 以降では、NetScaler SDX アプライアンスから SD-WAN サポートが廃止されました。

  [NSSVM-5465]

- **VPX** のプロビジョニングまたは編集中は、「ゲートウェイ」フィールドと「ネクストホップ」フィールドはオプションです

  NetScaler SDX アプライアンス管理サービスでは、次の条件が満たされると、VPX のプロビジョニング、編集、バックアップの作成、または復元に`Gateway`および`Nexthop`フィールドが必須ではなくなります。

  - 次のいずれかのオプションに該当します。
    - VPX では「内部ネットワーク経由の管理」が有効になっています。
    - VPX IP アドレスは、管理サービスの IP アドレスと同じサブネットにあります。
  - VPX は、バージョン 13.0-88.9 または 13.1-37.8、およびそれらの上位バージョンでプロビジョニングされます。

  詳しくは、「NetScaler インスタンスのプロビジョニング」を参照してください。

  [NSSVM-5307]

NetScaler Gateway

- **EDT** の **DF** ビット伝播をデフォルトで有効にするサポート

  NetScaler Gateway アプライアンスでは、EDT パス最大伝送ユニット検出（PMTUD）オプションの DF ビットエンフォースメントがデフォルトで有効になりました。このオプションは、パフォーマンスの低下やセッション確立の失敗につながる可能性のある EDT フラグメンテーションを防ぎます。以前は、このオプションはデフォルトで無効になっていました。管理者は ICA パラメータ設定を使用してオプションを有効にする必要がありました。

  [CGOP-22615]

NetScaler Web App Firewall

- **CLI** または **API** を使用して **Citrix Web App Firewall** ールの署名を有効にする

  CLI コマンドまたは API 呼び出しを使用して、NetScaler Web App Firewall で個々の署名を有効にできるようになりました。そのためには、署名を ID またはカテゴリで選択し、アクションを設定します。以前は、署名ファイルをアップロードすることによってのみ署名を有効にできました。

  例 1:

  ```
  import appfw signature DEFAULT object_name -sigRuleId 1001 9882
  2000 1250 810 -Enabled ON -Action LOG BLOCK
  ```

  例 2:

  ```
  import appfw signature DEFAULT object_name -sigCategory web-misc
  -Enabled ON -Action LOG BLOCK
  ```

  CLI を使用して個々の署名を追加するには、を参照してください。

[NSWAF-9333]

- **NetScaler Web App Firewall** シグネチャの新しいマッチパターン

  NetScaler Web App Firewall シグネチャでは、次の新しいマッチパターンを選択できるようになりました。

  - コマンドインジェクション
  - SQL インジェクション文法
  - コマンドインジェクション文法

  NetScaler Web App Firewall は、選択したパターンを探して攻撃を分類します。

  注: シグニチャルールパターンを変更できるのは、カスタムシグニチャだけです。

  詳細については、「署名ルールパターンの追加」を参照してください。

  [NSWAF-9280]

- **WAF** をバイパスするか、リクエストを拒否するようにグローバルリストを設定する

  NetScaler Web App Firewall プロファイルのグローバルリストを構成して、Web App Firewall をバイパスしたり、リクエストを拒否したりできるようになりました。受信リクエストがグローバルバイパスリストと一致する場合、NetScaler の Web App Firewall はスキップされます。受信要求がグローバル拒否リストと一致する場合、NetScaler Web App Firewall はそれらの要求をブロックし、定義されたアクションを適用します。

  バイパスリストと拒否リストは、URL、IPv4、および IPv6 アドレスをサポートします。リテラル、PCRE、および式を使用して指定できます。詳しくは、「グローバルリストを管理して WAF をバイパスするか、リクエストを拒否する」を参照してください。

  [NSWAF-8981]

- **NetScaler Web App Firewall** プロファイルの作成を簡略化して **CVE** から保護しました

  NetScaler Web App Firewall に適切な署名を適用して、NetScaler アプライアンスを保護します。他のセキュリティチェックを行わずに、アプライアンスを CVE から保護したい場合があります。この場合、NetScaler Web App Firewall からの残りのチェックを無効にするプロファイルを作成できるようになりました。

  NetScaler Web App Firewall プロファイルで、デフォルトとして **CVE** オプションを選択します。このオプションでは、署名を追加してバインドするだけで済みます。残りのチェックは自動的に無効になります。以前は、プロファイルからセキュリティチェックを 1 つずつ手動で無効にする必要がありました。

  詳細については、「Web App Firewall プロファイルの作成」を参照してください。

  [NSWAF-8970]

プラットフォーム

- **VMware vSphere 8.0.0b** のサポート

NetScaler VPX インスタンスが VMware vSphere 8.0.0b（ビルド 20513097）をサポートするようになりました。

[NSPLAT-25844]

- パブリッククラウドの同じ **Autoscaling** グループによる複数のサービスのサポート

  パブリッククラウドのバックエンド自動スケーリング機能では、NetScaler VPX インスタンスが同じ自動スケーリンググループの複数のサービスをサポートするようになりました。この機能は Azure、AWS、および GCP クラウドでサポートされています。NetScaler GUI では、クラウド内の同じ自動スケーリンググループを使用して、さまざまなサービスに対して（異なるポートを使用して）さまざまなクラウドプロファイルを作成できます。

  以前は、NetScaler VPX インスタンスのサポートは、自動スケーリンググループごとに 1 つのサービスに制限されていました。サービスごとに異なる自動スケーリンググループを追加する必要がありました。

  [NSPLAT-21596]

- **VMware ESXi** ハイパーバイザでの **SR-IOV** 搭載 **Mellanox ConnectX-4 NIC** のサポート

  NetScaler VPX インスタンスは、VMware ESXi ハイパーバイザー上の SR-IOV を搭載した Mellanox ConnectX-4 NIC をサポートするようになりました。

  [NSPLAT-20295]

ポリシー

- パターンセットにバインドできるパターンの制限の増加

  NetScaler アプライアンスでは、50000 個のパターンをパターンセットにバインドできるようになりました。パターンセットファイルでは、パターンセットにバインドできるパターンは 10000 個だけです。また、パターンセットをストリーミングで使用する場合、そのパターンセットにバインドできるパターンは 5000 個だけです。ストリーミング用のパターンセットは、リライトアクションの検索パラメータ、HTTP 本文、または TCP ペイロードベースの式で使用されます。以前は、パターンセットにバインドできるパターンは 5000 個まででした。

  [NSPOLICY-2733]

- クライアント側とサーバー側の **UDP** ヘッダーとペイロードに関連するすべての表現のサポート

  クライアント側とサーバー側の UDP ヘッダーとペイロードに対して以下の拡張が行われました。

  - UDP プロトコルに関連する式は、クライアント側とサーバー側の式に分けられます。
  - 以前のサポートはクライアント側の式のみで、サーバー側でも同じ式が使用されていました。
  - UDP プロトコルがサーバー側の表現をサポートするようになりました。この式を使用して、UDP 送信元ポート、宛先ポート、長さ、チェックサム、およびペイロードを抽出できます。
  - クライアント側の表現も強化され、特定の UDP パケットから長さ、チェックサム、ペイロードを抽出できるようになりました。

- 下位互換性のため、クライアント側の式がサーバー側で使用されている場合は引き続きサポートされます。Citrix では、サーバー側にはサーバー側の式を使用することをお勧めします。

詳細については、「TCP、UDP、および VLAN データの式」を参照してください。

[NSPOLICY-1829]

SSL

- クロス署名証明書検証のサポート

  NetScaler アプライアンスはクロス署名証明書検証をサポートするようになりました。証明書が複数の発行者によって署名されている場合、ルート証明書への有効なパスが少なくとも 1 つあれば検証は成功します。

  以前は、証明書チェーン内の証明書の 1 つがクロス署名されていて、ルート証明書へのパスが複数ある場合、ADC アプライアンスはパスを 1 つだけチェックしていました。そのパスが有効でない場合、検証は失敗しました。

  [NSSSL-11259]

システム

- **NetScaler** アプライアンスから **Prometheus** へのメトリクスの直接エクスポートのサポート

  NetScaler は、Prometheus へのメトリクスの直接エクスポートをサポートするようになりました。この機能により、Prometheus は外部のエクスポーターを必要とせずに、NetScaler インスタンスから直接メトリクスを取得できます。以前は、NetScaler から Prometheus サーバーにメトリックをエクスポートするには、アプライアンスの外部にエクスポーターリソースが必要でした。

  詳しくは、「Prometheus を使用した NetScaler とアプリケーションの監視」を参照してください。

  [NSBASE-17100]

ユーザーインターフェイス

- `systemfile` **NITRO API** の **8 MB** のアップロード制限サポート

  `systemfile` NITRO API の最大アップロード制限が 2 MB から 8 MB に引き上げられました。

  [NSCONFIG-7089]

- **NITRO API** レスポンスでの **64** ビットの数値のサポート

  以前は、NetScaler アプライアンスは、これらのタイプでは整数応答がサポートされていなかったため、NITRO API 応答で符号なし整数または長いプロパティタイプの値を文字列として返していました。また、アプライアンスはダブルデータタイプの stats-counter-rate 値を整数として返しました。

NITRO API は 64 ビットの整数をサポートするようになりました。このサポートにより、アプライアンスは NITRO API レスポンスで以下を返すことができます。

- 符号なし整数または長整数データ型の文字列の代わりに正確な整数値を指定します。
- 整数の代わりにシリアル化された正確なカウンタレート値。

NITRO API で 64 ビット整数サポートを有効にするための新しいクエリパラメータ`largeintsupport`が導入されました。

NITRO API リクエストで`largeintsupport`を`yes`に設定すると、NetScaler アプライアンスは NITRO API レスポンスで正確な整数値を返します。`largeintsupport`を`no`に設定しても、以前の機能は保持されます。これもデフォルト設定です。

[NSCONFIG-5399]

解決された問題

ビルド 13.1-42.47 で対処されている問題

認証、承認、監査

- NetScaler アプライアンスをアップグレードすると、ユーザーは RADIUS 認証を使用して NetScaler アプライアンスにアクセスできなくなります。

  [NSHELP-33200]

- NetScaler GUI の［認証仮想サーバー］ページの［応答ポリシー］セクションには、応答側タイプのキャッシュポリシーが表示されません。

  [NSHELP-33111]

- CWA クライアントまたはネイティブ VPN クライアントによるゲートウェイ認証は、`ns_aaa_relaystate_param_w` patset に文字列がないために失敗する可能性があります。

  [NSHELP-33054]

- SSO 認証情報に間違ったユーザープリンシパル名を使用すると、高度な暗号化タイプの Kerberos SSO 偽装が失敗することがあります。

  [NSHELP-32890, NSHELP-34087]

ボット管理

- 署名ファイルの形式が無効な場合、ボット署名の処理中に NetScaler アプライアンスがクラッシュします。

  [NSHELP-33690]

- NetScaler GUI では、ユーザー定義のボット署名に誤った基本バージョンが表示されます。

  [NSHELP-33546]

NetScaler SDX アプライアンス

- NetScaler SDX アプライアンスをアップグレードすると、まれに管理サービス GUI に次の誤ったイベントが表示されます。

  「SVM バージョンとハイパーバイザーバージョンには互換性がありません」

  [NSHELP-32949]

NetScaler Gateway

- VPN URL のポリシーを評価すると、NetScaler Gateway アプライアンスがクラッシュします。

  [NSHELP-33683, CGOP-20369, NSHELP-34002, NSHELP-34030, NSHELP-34052, NSHELP-34076, NSHELP-34077, NSHELP-34100, NSHELP-34151, NSHELP-34180, NSHELP-34243, NSHELP-34276, NSHELP-34327, NSHELP-34402]

- NetScaler アプライアンスをアップグレードすると、RDP プロキシ URL が X1 ポータルテーマで動作せず、「Http/1.1 オブジェクトが見つかりません」というメッセージが表示されます。

  [NSHELP-33676, NSHELP-33845, NSHELP-33921, NSHELP-34032]

- NetScaler アプライアンスをアップグレードすると、UDP トラフィックの処理中にアプライアンスがクラッシュする可能性があります。

  [NSHELP-33417, NSHELP-34031]

- NetScaler アプライアンスをアップグレードすると、RDP プロキシ URL にアクセスできなくなり、「Http/1.1 Object Not Found」というエラーメッセージが表示されます。この問題は、RDP URL のカスタムパラメータにスペースが含まれている場合に発生します。

  [NSHELP-33333]

- NetScaler Gateway の高可用性設定では、フェイルオーバー中にプライマリアプライアンスとセカンダリアプライアンスがクラッシュする可能性があります。

  [NSHELP-33198, NSHELP-33483]

- 一部の VPN セッションは、フェイルオーバー後にセカンダリ ADC アプライアンスからクリアまたは削除されることがあります。

  [NSHELP-33125]

- HDX Insight が有効になっていて、ユーザーがログアウト後すぐに StoreFront にログインすると、NetScaler Gateway アプライアンスがクラッシュすることがあります。

  [NSHELP-32907, NSHELP-33079, NSHELP-33289]

- まれに、VPN 展開で STA モニターを取得中に NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-32893]

- NetScaler Gateway アプライアンスをアップグレードすると、NetScaler GUI に [構成] > [NetScaler 製品との統合] セクションが表示されません。

  [NSHELP-32335]

- CA 証明書のドメインが異なる場合、クライアントデバイスの CA 証明書を確認するための EPA スキャンが NetScaler アプライアンスで失敗します。

  [NSHELP-32118]

- NetScaler アプライアンスで GSLB が有効になっていると、macOS 用の Citrix EPA プラグインがクラッシュします。

  [CGOP-22722]

NetScaler Web App Firewall

- NetScaler Web App Firewall では、ストリーミングとフィールドの一貫性チェックを有効にすると、ペイロードのオリジンサーバーへの転送が遅れます。その結果、ペイロードの POST メソッドは失敗します。

  [NSHELP-33700]

- Cookie ハイジャックリダイレクトは、リクエスト URL からクエリパラメータを削除します。その結果、リダイレクトされたリクエストが失敗する可能性があります。

  [NSHELP-33633, NSHELP-33812]

負荷分散

- 同じ GSLB 仮想サーバーを複数の GSLB 仮想サーバーのバックアップとして使用すると、セカンダリノードがクラッシュする可能性があります。

  [NSHELP-33400, NSHELP-34247]

- GSLB 仮想サーバーで次の設定が構成されている場合、NetScaler アプライアンスは GSLB ドメインクエリの正しいサービス IP アドレスで応答しません。

  1. ECS オプションは有効です。
  2. 静的近接は負荷分散方法として設定されます。

  [NSHELP-32879]

ネットワーク

- INC モードの高可用性セットアップでは、HA バージョンの不一致があると、セカンダリノードがプライマリノードから無効なルートを学習する可能性があります。

  [NSHELP-33948]

- OSPF ルーティングが設定された NetScaler アプライアンスでは、OSPF デフォルトルート LSA が存在する場合でも、デフォルトルートはインストールされません。

  [NSHELP-33070]

- SSH セッションのいくつかの受信パケットの`nstrace`で、次の条件がすべて満たされると、異なる受信インターフェイス番号と VLAN ID が誤って表示されることがあります。

  - SSH セッションのクライアント用の ECMP ルートは、NetScaler アプライアンスにあります。
  - SSH セッションは数秒間アイドル状態です。

  [NSHELP-32734]

- ファイル内の SNMP トラップ名`dataStreamRateLimitHit`がキャメルケースではないため、SNMP MIB ファイルをネットワークモーニングツールにロードできないことがあります。

  [NSHELP-32634]

- 大規模な NAT 64 セットアップでは、内部パケットエンジンの不一致の問題により NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-31985]

- GSLB サイトの IP アドレスが管理パーティションに設定されている GSLB セットアップでは、アップストリームルーターからのこの GSLB サイト IP アドレスに対する ARP リクエストが管理パーティションに到達しません。この問題は、次の条件がすべて満たされた場合に発生します。

  - 共有 VLAN は admin パーティションにバインドされます。
  - 共有 VLAN には、GSLB サイトの IP アドレスと同じサブネットの SNIP IP アドレス、たとえば SNIP-1 が存在します。
  - GSLB サイトの IP アドレスと同じサブネットにある別の SNIP IP アドレス、たとえば SNIP-2 が追加され、SNIP-1 が削除されます。

  [NSHELP-30552]

プラットフォーム

- VMXNET3 インターフェイスを備えた VMware ESX ハイパーバイザー上の NetScaler VPX リリース 13.1 ビルド 37.38 では、高可用性セットアップで次の動作が見られます。

高可用性ノード間の通信が確立されていないため、NetScaler VPX HA ペアは構成されていません。その結果、ピアノードのステータスは UNKNOWN と表示されます。

[NSPLAT-25677]

- ESX vSphere クライアントの OVF テンプレートにプレブートユーザーデータを提供すると、ESXi ホストはプレブート構成を適用しません。

  [NSPLAT-24233、NSPLAT-25551]

- AWS VPC の DHCP オプションセットに 3 つ以上の DNS サーバー名を設定すると、DNS 解決が失敗します。この問題は、13.1 ビルド 42.x より前のリリースの NetScaler VPX インスタンスで見られます。

  [NSHELP-33171]

- NetScaler SDX 8015/8400/8600 プラットフォームでは、Xen サーバーのメモリ消費量が増加する可能性があります。

  [NSHELP-32260]

- 10G インターフェイスを備えた NetScaler SDX アプライアンスでは、このインターフェイスで大量のトラフィックが送信されると、送信が停止することがあります。

  [NSHELP-31232]

SSL

- NetScaler アプライアンスのメモリが不足し、TLS 1.3 ハンドシェイクの開始中にメモリ割り当て要求が失敗するため、TLS1.3 接続の失敗により仮想サーバーがクラッシュします。

  この修正により、TLS 1.3 接続は失敗しますが、アプライアンスはクラッシュしません。

  [NSSSL-12200]

- 仮想サーバーは、次の条件が満たされると、TLS `decrypt_error` 1.3 ハンドシェイクをアラートで誤って終了する可能性があります。

  - クライアントは証明書で認証中です。
  - 仮想サーバーは、OCSP または CRL を使用して証明書のステータスチェックを実行するように構成されています。
  - クライアントは、証明書メッセージと証明書検証メッセージの両方を同じ TLS レコードで送信します。

  [NSHELP-33355]

- DEFAULT 暗号をバインド解除した後、仮想サーバー上のプロトコルバージョンを無効にし、後で説明に記載されているこのプロトコルで暗号をバインドしようとすると、次のエラーメッセージが表示されます。

  ```
  No usable ciphers configured on the SSL vserver/service
  ```


暗号は仮想サーバーで有効になっている他のプロトコルでサポートされているため、このメッセージは正しくありません。例:

暗号名:TLS1-ECDHE-RSA-AES256-SHA
説明:SSLv3 KX=ECC-DHE AU=RSA ENC=AES (256) MAC=SHA1 HEXCODE=0xC014

この暗号は、SSLv3 以降のすべてのプロトコル (SSLv3、TLS1、TLS11、TLS12) でサポートされています。仮想サーバーで SSLv3 を無効にしてからこの暗号をその仮想サーバーにバインドしようとすると、仮想サーバーで TLS1、TLS11、TLS12 プロトコルがまだ有効になっていても警告が表示されます。

今回の修正により、構成で暗号がサポートされていない場合にのみ警告が表示されます。

[NSHELP-32739]

- NetScaler アプライアンスでは、1970 年より古い`notBefore date`の証明書を構成することはできません。

  [NSHELP-32677]

- 次の条件が満たされると、NetScaler アプライアンスがクラッシュする可能性があります。

  - クライアントは、クライアントハローメッセージの TLS1.3 初期データを SSL Insight 仮想サーバーに送信します。
  - この仮想サーバーでは ECDHE 暗号が有効になっています。

  [NSHELP-31560]

システム

- RFC（RFC 7230）に準拠していないお客様のアプリケーションは、NetScaler 13.1 へのアップグレード後に機能しなくなる可能性があります。この障害は、RFC 7230 に準拠するように NetScaler アプライアンスに強制的なコンプライアンスチェックが実施されているために発生します。

  修正の一環として、この特定のコンプライアンスチェックは HTTP プロファイルパラメーター「-markrfc7230NonCompliantInVal」に移動されました。“お客様は、以前に実施されたこのコンプライアンスチェックを無効にできます。

  [NSHELP-34046]

- NetScaler アプライアンスは、次の両方の条件が満たされるとクラッシュする可能性があります。

  - コンテンツ検査デバイスが ADC アプライアンスにリセット (RST) 応答を送信しますが、侵入防止システム (IPS) リソースの 1 つが正しくクリアされていません。
  - 以降のトランザクションでは、同じ IPS リソースにアクセスします。

  [NSHELP-33691]

- 場合によっては、TIME_WAIT 状態のサーバー接続から送信された修正確認を処理しているときに、NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-33469]

- NetScaler アプライアンスは、解放された ICAP 上のリソースにアクセスしようとするとクラッシュする可能性があります。この状態は、ICAP が応答修正 (RESPMOD) モードの場合に発生します。

  [NSHELP-33403]

- NetScaler アプライアンスは、パーティションから Logstream データを一貫して送信することができません。

  [NSHELP-33237]

- NetScaler アプライアンスは、チャンクされた値の解析に失敗すると接続を中止します。この問題は、Transfer-Encoding ヘッダーに複数の値があり、Chunked が最初の値ではない場合に発生します。

  [NSHELP-32420]

- NetScaler アプライアンスは、サーバー側の TCP 接続に関連する修正 ACK パケットを処理するとクラッシュする可能性があります。

  [NSHELP-32290]

- SSL サービスで構成された NetScaler アプライアンスは、アプライアンスが TCP FIN 制御パケットの後に TCP RESET 制御パケットを受信するとクラッシュします。

  [NSHELP-31656]

ユーザーインターフェイス

- JSON タイプの NetScaler Web App **Firewall** プロファイルを作成してプロファイル設定を更新しようとすると、**JSON** エラーオブジェクトに空のリストが表示されます。

  [NSUI-18453]

- システムグローバル設定の一部として Allow Default Partition オプションが有効になっていても、管理者パーティションのセットにバインドされたシステムユーザーアカウントは NITRO API を介してデフォルトパーティションにアクセスできない場合があります。

  [NSHELP-33990]

- NetScaler ボット管理プロファイルのリンクが、[トラフィック管理**] > [**コンテンツスイッチング] ページに正しく表示されません。そのリンクをクリックすると、空白のページが表示されます。この問題は、ボットポリシーをコンテンツスイッチング仮想サーバーにバインドした場合に発生します。

  [NSHELP-33697]

- ユーザー名またはドメイン名に特殊文字が含まれていると、NetScaler GUI へのログオンが失敗します。

  [NSHELP-33684]

- 実行中の NetScaler 構成をクリアすると、`RBAconfig` パラメータが NO に設定されていても、従来の TACACS 構成で作成された NetScaler 管理セッションは切断されます。

  [NSHELP-33655]

- ユーザーがコンテンツスイッチングポリシーのバインドを表示しても、コンテンツスイッチング仮想サーバーの詳細は [バインディングの表示] の同じ行に表示されません。

  [NSHELP-33149]

- シャットダウン時の **NITRO API** でのパワーオフオプションのサポート

  `shutdown` NITRO API は、NetScaler アプライアンスをシャットダウンして電源を切る「-p now」オプションをサポートするようになりました。

  例:

  次の curl リクエストの例では、`shutdown` NITRO API を「-p now」オプションとともに使用して、IP アドレス 192.0.0.33 の NetScaler アプライアンスをシャットダウンして電源を切ります。

  ‘curl -v -X POST -H Content-Type: application/json -u nsroot:examplepassword http://192.0.0.33/nitro/v1/config/install?warning=yes -d ‘{ “shutdown” : { “args” :” -p now” }}’ ‘

  [NSHELP-32915]

- NetScaler Web App Firewall のプロファイルを作成し、[システム**] > [**レポート**]** でアプリケーションファイアウォールの構成レポートを生成しようとすると、次のエラーが表示されます。

  「PDF ドキュメントを読み込めませんでした。」

  [NSHELP-32469]

- クラスタ設定では、NetScaler GUI を使用して仮想サーバーを作成すると、プロトコルリストに TFTP オプションが表示されません。

  [NSHELP-32036]

- NetScaler GUI では、[システムログファイル] ページ（[構成] > [システム] > [監査] > [Syslog メッセージ]）と [ログ] ページ（[構成] > [認証] > [ログ]）がログファイルを読み込めません。

  [NSHELP-30868]

- NetScaler GUI では、保存済み構成と実行構成画面（[システム] > [診断]）に、プレーンテキストではなく HTML タグが誤って表示されます。

  [NSHELP-27169]

- コンテンツスイッチポリシーラベルにバインドされたポリシーを NetScaler GUI で表示すると、そのポリシーラベルにバインドされたポリシーが他にもあっても、25 個のポリシーしか表示されません。

  [NSHELP-23428]

既知の問題

リリース 13.1-42.47 に存在する問題。

AppFlow

- HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

  [NSINSIGHT-943]

認証、承認、監査

- 管理者は、認証情報が無効であることが原因で発生した認証エラーのカスタムロギングを実行できません。この問題は、NetScaler Responder ポリシーがログインエラーのエラーを検出できないために発生します。

  [NSAUTH-11151]

- ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
  `show adfsproxyprofile <profile name>`

  回避策：クラスター内のアクティブなプライマリの NetScaler に接続し、`show adfsproxyprofile <profile name>`コマンドを実行します。プロキシプロファイルの状態が表示されます。

  [NSAUTH-5916]

- 次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

  - [LDAP 到達可能性をテスト] オプションが開きます。
  - 無効なログイン認証情報が入力され、送信されます。
  - 有効なログイン認証情報が入力され、送信されます。

  回避策：「LDAP 到達可能性テスト」オプションを閉じて開きます。

  [NSAUTH-2147]

NetScaler SDX アプライアンス

- 次の条件が満たされている場合、NetScaler SDX アプライアンスでホストされている VPX インスタンスでパケットドロップが表示されます。

  - スループット割り当てモードはバーストです。
  - スループットと最大バーストキャパシティには大きな違いがあります。

  [NSHELP-21992]

NetScaler Gateway

- Citrix Secure Access 関連のレジストリ値が 1500 文字を超える場合、ログコレクターはエラーログを収集できません。

  [NSHELP-33457]

- Windows フィルタリングプラットフォーム (WFP) ドライバを使用している場合、VPN の再接続後にイントラネットアクセスが機能しないことがあります。

  [NSHELP-32978]

- Citrix Secure Access クライアントのバージョン 21.7.1.2 以降では、管理者権限のないユーザーが新しいバージョンにアップグレードできません。この問題は、Citrix Secure Access クライアントのアップグレードが NetScaler アプライアンスから行われる場合にのみ発生します。

  [NSHELP-32793]

- ユーザーが Windows 向け Citrix Secure Access 画面の［ホームページ］タブをクリックすると、ページに接続拒否エラーが表示されます。

  [NSHELP-32510]

- Chrome を使用する Mac デバイスで、2 つの FQDN にアクセス中に VPN 拡張機能がクラッシュします。

  [NSHELP-32144]

- NetScaler Gateway リリース 13.0 または 13.1 のプロキシ設定が空の場合、Citrix SSO によって不適切なプロキシ設定が作成されることがあります。

  [NSHELP-31970]

- Citrix Secure Access クライアントのデバッグログ制御は、NetScaler Gateway に依存せず、マシントンネルとユーザートンネルの両方のプラグイン UI から有効または無効にできるようになりました。

  [NSHELP-31968]

- Citrix Secure Access によって確立されたトンネルの外部にあるリソースへの直接接続は、大幅な遅延または輻輳が発生すると失敗することがあります。

  [NSHELP-31598]

- カスタマイズされた EPA 障害ログメッセージは、NetScaler Gateway ポータルには表示されません。代わりに、「内部エラー」というメッセージが表示されます。

  [NSHELP-31434]

- ユーザーが Always-On サービスモードで Windows マシンにログインすると、Windows 自動ログオンが機能しないことがあります。マシントンネルはユーザートンネルに遷移しません。「Connecting…」というメッセージが VPN プラグイン UI に表示されます。

  [NSHELP-31357, CGOP-21192, NSHELP-34211]

- Always on が設定されている場合、aoservice.exe ファイルのバージョン番号（1.1.1.1）が正しくないため、ユーザトンネルが失敗します。

  [NSHELP-30662]

- ［NetworkAccessonVPNFailure］プロファイルパラメーターを［フルアクセス］から［OnlyToGateway］に変更すると、ユーザーは NetScaler Gateway アプライアンスに接続できなくなります。

  [NSHELP-30236]

- ゲートウェイのホームページは、ゲートウェイプラグインが VPN トンネルを正常に確立した直後には表示されません。この問題を解決するために、次のレジストリ値が導入されます。

  HKLMSoftwareCitrixSecure Access ClientSecureChannelResetTimeoutSeconds
  タイプ: DWORD

  デフォルトでは、このレジストリ値は設定も追加もされません。「SecureChannelResetTimeoutSeconds」の値が 0 または追加されていない場合、遅延を処理するための修正が機能しません。これはデフォルトの動作です。管理者は、このレジストリをクライアントに設定して修正を有効にする必要があります (つまり、ゲートウェイプラグインが VPN トンネルを正常に確立した直後にホームページを表示する)。

  [NSHELP-30189]

- Windows VPN クライアントは、サーバからの「SSL close notify」アラートを尊重せず、同じ接続で転送ログイン要求を送信します。

  [NSHELP-29675]

- macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

  [NSHELP-28551]

- クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

  [NSHELP-28404]

- 次の条件を満たす場合、Windows ログオン後に VPN プラグインはトンネルを確立しません。

  - NetScaler Gateway アプライアンスが常時オン機能用に構成されている
  - アプライアンスは、2 要素認証が「オフ」の証明書ベースの認証に設定されています

  [NSHELP-23584]

- スキーマをブラウズしているときに、「未定義のプロパティ「タイプ」を読み取れません」というエラーメッセージが表示されることがあります。

  [NSHELP-21897]

- NetScaler クラスター設定では、HDX Insight と Gateway Insight を同時に有効にすることはできません。

  [CGOP-23570]

- Windows OS オプションは、NetScaler GUI の事前認証ポリシーと認証アクションの「エクスプレッションエディタ」ドロップダウンリストに表示されません。ただし、GUI または CLI を使用して以前の NetScaler ビルドで Windows OS スキャンをすでに構成している場合は、アップグレードしても機能に影響はありません。必要に応じて CLI を使用して変更できます。

  回避方法:

  設定には CLI コマンドを使用します。

  - nFactor 認証で高度な EPA アクションを設定するには、次のコマンドを使用します。
    add authentication epaAction adv_win_scan -csecexpr “sys.client_expr(“sys_0_WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]”)”
  - 従来の事前認証アクションを設定するには、次のコマンドを使用します。

    ```
    add aaa preauthenticationaction win_scan_action ALLOW
    add aaa preauthenticationpolicy win_scan_policy "CLIENT.
    SYSTEM('WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]')EXISTS
    "win_scan_action
    ```

  [CGOP-22966]

- Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースにはない、リリース 13.0 で導入された追加の拡張機能を活用できます。

  [CGOP-19355]

- Gateway Insight レポートでは、SAML エラーエラーの認証タイプフィールドに「SAML」ではなく「Local」という値が誤って表示されます。

  [CGOP-13584]

- 高可用性セットアップでは、NetScaler フェイルオーバー時に、NetScaler Console のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

  [CGOP-13511]

- ICA 接続が MAC Receiver バージョン 19.6.0.32 または Citrix Virtual Apps and Desktops バージョン 7.18 から起動されると、HDX Insight 機能は無効になります。

  [CGOP-13494]

- EDT Insight 機能が有効になっていると、ネットワークの不一致時にオーディオチャネルに障害が発生することがあります。

  [CGOP-13493]

- ブラウザからローカルホスト接続を受け付けると、macOS の Accept Connection ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

  [CGOP-13050]

- 一部の言語では、Citrix SSO アプリ > ホームページの「ホームページ」というテキストが切り捨てられます。

  [CGOP-13049]

- NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

  [CGOP-11830]

- Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

  [CGOP-7269]

負荷分散

- 高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

  [NSLB-7679]

- サービスグループの`entityofs`トラップの ServiceGroupName 形式は次のとおりです。
  `<service(group)name>?<ip/DBS>?<port>`

  トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (“?”) は区切り文字として使用されます。NetScaler は、疑問符 (” ?”)。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

  [NSHELP-28080]

その他

- 高可用性セットアップで強制同期が実行されると、アプライアンスはセカンダリノードで`set urlfiltering parameter`コマンドを実行します。
  その結果、セカンダリノードは、「TimeOfDayToUpdateDB」パラメーターで指定された次のスケジュール時刻まで、スケジュールされた更新をスキップします。

  [NSSWG-849]

- レジストリ値が 2000 バイトを超えると、AlwaysOnAllow リストレジストリが期待どおりに動作しません。

  [NSHELP-31836]

- URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

  [NSHELP-22409]

ネットワーク

- DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

  [NSNET-25299]

- DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

  - NetScaler BLX アプライアンスは、hugepagesの小さい数が割り当てられます。たとえば、1G です。
  - NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

  この問題は、エラーメッセージとして「/var/log/ns.log」に記録されます。

  - `BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x`

  注:x はワーカープロセス数以下の数です。

  回避策: hugepages 多数の数を割り当ててから、アプライアンスを再起動します。

  [NSNET-25173]

- DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

  [NSNET-24449]

- DPDK を搭載した NetScaler BLX アプライアンスの Intel X710 10G (i40e)インターフェイスでは、次のインターフェイス操作はサポートされていません:

  - 無効化
  - 有効化
  - リセット

  [NSNET-16559]

- Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

  ```
  The following packages have unmet dependencies: blx-core-libs:
  i386 : PreDepends: libc6:i386 (>= 2.19)but it is not installable
  ```

  回避策：NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します:

  - `dpkg --add-architecture i386`


```
- apt-get update
- apt-get install libc6:i386
```

[NSNET-14602]

- FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

  [NSNET-5233]

- NetScaler アプライアンスは、コールドリスタート後に「ColdStart」SNMP トラップメッセージを生成しない場合があります。

  [NSHELP-27917]

- NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

  [NSHELP-21082]

プラットフォーム

- NetScaler アプライアンスを 13.1～4.x 以降のバージョンから次のバージョンのいずれかにダウングレードすると、一部の Python パッケージがインストールされません:

  - 任意の 11.1 ビルド
  - 12.1-62.21 およびそれ以前
  - 13.0-81.x およびそれ以前

  [NSPLAT-21691]

- Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。`rm cloudprofile` コマンドを使用して、プロファイルを削除します。

  [NSPLAT-4520]

- Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
  回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノード上で設定する必要があります。

  [NSPLAT-4451]

ポリシー

- 処理データのサイズが設定されたデフォルトの TCP バッファーサイズを超えると、接続がハングアップすることがあります。

  回避策:TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

  [NSPOLICY-1267]

SSL

- NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

  回避方法:

  1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
  2. 構成を保存します。

  [NSSSL-9572]

- 認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

  [NSSSL-6478]

- 同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

  [NSSSL-6213]

- HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
  `ERROR: crl refresh disabled`

  [NSSSL-6106]

- セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

  [NSSSL-4427]

- SSL プロファイル内の SSL プロトコルまたは暗号を変更しようとすると、「警告:SSL vserver/Service で使用可能な暗号が設定されていません」という誤った警告メッセージが表示されます。

  [NSSSL-4001]

- 期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

  [NSSSL-3184, NSSSL-1379, NSSSL-1394]

システム

- 次の条件が満たされると、TCP 接続の RTT が高くなります。

  - 最大輻輳ウィンドウ（> 4 MB）が高く設定されている
  - TCP NILE アルゴリズムは有効になっています

  NetScaler アプライアンスが NILE アルゴリズムを使用して輻輳制御を行うには、条件がスロースタートのしきい値と最大輻輳ウィンドウを合わせた値を超える必要があります

  そのため、設定された最大輻輳時間帯に達するまで、NetScaler はデータを受け付け続け、最終的には RTT が高くなります。

  [NSHELP-31548]

- アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

  [NSHELP-21240]

- mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

  [NSHELP-10972]

- まれに、NetScaler が次の条件をすべて満たすと、HTTP/2 クライアント接続で内部エラーで HTTP/2 GoAway フレームを開始することがあります。

  - クライアントまたはバックエンドサーバーは、クライアントの HTTP/2 接続上の最後の WebSocket または Connect ストリームを閉じようとします。
  - 多重化は有効になっています。

  このエラーは、クライアントの HTTP/2 接続で進行中のトランザクションには影響しません。

  回避策: 以下のコマンドを使用して、関連する HTTP/2 プロファイルの接続多重化を無効にします:

  ```
  set httpProfile <name> [-conMultiplex ( ENABLED | DISABLED )]
  ```

  [NSBASE-17449]

- クラスタ展開では、CCO 以外のノードで「force cluster sync」コマンドを実行すると、ns.log ファイルには重複するログエントリが含まれます。

  [NSBASE-16304、NSGI-1293]

- NetScaler Console を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに機能しません。

回避策：管理ポッドを再起動します。

[NSBASE-15556]

- LogStream トランスポートタイプが Insight に設定されている場合、HDX Insight SkipFlow レコードでクライアント IP とサーバー IP が反転します。

  [NSBASE-8506]

ユーザーインターフェイス

- NetScaler GUI では、「ダッシュボード」タブの下にある「ヘルプ」リンクが壊れています。

  [NSUI-14752]

- CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

  回避策：NetScaler GUI または CLI を使用して IPSec プロファイル、IP トンネル、および PBR ルールを追加して CloudBridge Connector を構成します。

  [NSUI-13024]

- GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

  [NSUI-6838]

- 高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

  - HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

  回避策：HA 同期が完了した後にのみ、手動で HA フェイルオーバーを連続して実行してください（両方のノードが同期成功状態になっています）。

  [NSHELP-25598]

- あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

  1. NetScaler アプライアンスをいずれかのビルドにアップグレードします
     - 13.0 52.24 ビルド
     - 12.1 57.18 ビルド
     - 11.1 65.10 ビルド
  2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
  3. NetScaler アプライアンスを古いビルドにダウングレードします。

  CLI を使用してこれらのシステムユーザーのリストを表示するには、
  コマンドプロンプトで次のように入力します。


```
query ns config -changedpassword [-config <full path of the
configuration file (ns.conf)>]
```

回避策: この問題を解決するには、以下のいずれかの独立したオプションを使用してください。

- NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3）、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
- アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
- 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳細については、/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.htmlを参照してください。

[NSCONFIG-3188]

NetScaler 13.1-37.38 リリースのリリースノート

April 15, 2024

このリリースノートドキュメントでは、NetScaler リリースビルド 13.1-37.38 の機能強化と変更、修正された問題と既知の問題について説明します。

メモ

- このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正と勧告のリストについては、セキュリティ速報を参照してください。
- NetScaler SDX バンドルビルド 13.1-37.39 は、ビルド 13.1-37.38 に代わるものです。

新機能

ビルド 13.1-37.38 で利用できる機能強化と変更点。

NetScaler SDX アプライアンス

- アップグレードプロセスの強化

NetScaler SDX アプライアンスでは、アップグレードプロセスに 2 回の再起動ではなく 1 回の再起動が必要になりました。

[NSSVM-5299]

- **SDX UI** からのサードパーティインスタンスサポートの削除

  NetScaler SDX アプライアンスは、UI インターフェイスからのサードパーティインスタンスをサポートしなくなりました。「サードパーティインスタンス」ビューは、SDX UI インターフェイスの「設定」タブから削除されました。

  回避策:Management Service で引き続きサードパーティのインスタンスを使用したい場合は、以下の手順を使用してください。

  1. 管理サービスシェルにログオンします。
  2.「/mpsconfig」ディレクトリに「.ThirdPartyVM」というファイルを作成します。
  3. 管理サービスシェルで`svmd restart`コマンドを実行して、管理サービスを再起動します。

  [NSSVM-5229]

NetScaler Gateway

- 認証クッキーの **HTTPOnly** フラグのサポート

  VPN シナリオの認証クッキー、つまり NSC_ 認証、承認、および監査 C および NSC_TMAS クッキーで HttpOnly フラグがサポートされるようになりました。NSC_TMAS 認証 Cookie は nFactor 認証時に使用され、NSC_Authentication、承認、および AuditingC Cookie は認証セッションに使用されます。Cookie の HTTPOnlyFlag は、JavaScript ドキュメントの Cookie オプションを使用して Cookie へのアクセスを制限します。これにより、クロスサイトスクリプティングによる Cookie の盗難を防ぐことができます。

  [CGOP-14004]

負荷分散

- 自動遅延 **TROFS** ステートの設定

  DNS 応答から IP アドレスが削除されたときに、サービスグループのメンバーが TROFS 状態に正常に移行するように設定できます。自動遅延 TROFS が有効になっている場合、NetScaler は、メンバーを TROFS 状態に移行する前に、サービスグループに接続されているすべてのモニターで最も長い応答タイムアウトを待ちます。

  詳細については、「ドメインベースのサービスグループの自動スケーリングの設定」を参照してください。

  [NSLB-9371]

ネットワーク

- **AMD** プロセッサーベースの **Linux** ホスト上の **NetScaler BLX** アプライアンスに対する **DPDK** サポート

  AMD プロセッサーベースの Linux ホスト上の NetScaler BLX アプライアンスが DPDK をサポートするようになりました。アプライアンスは、Linux ホスト上の指定された DPDK 互換 NIC ポートを自動的に検出します。その後、アプライアンスはそれらを DPDK モードで初期化します。NetScaler BLX アプライアンスを起動すると、DPDK ポートが専用ポートとしてアプライアンスに追加されます。

  「blx.conf」ファイルで DPDK 互換 NIC ポートを 1 つ以上指定する代わりに、同じ IOMMU グループに属する DPDK 互換 NIC ポートをすべて指定する必要があります。それ以外の場合、DPDK 互換の NIC ポートは非 DPDK 専用ポートとして NetScaler BLX アプライアンスに追加されます。

  [NSNET-19219]

プラットフォーム

- **GCP** の共有コアインスタンスのパフォーマンスの向上

  NetScaler VPX インスタンスでは、GCP の共有コアインスタンスに対して CPU 利回りパラメータがデフォルトで有効になっています。これにより、共有コアインスタンスの GCP のパフォーマンスが向上します。GCP の共有コアマシンの種類について詳しくは、Google Cloud のドキュメントをご覧ください。

  GCP の共有コアインスタンスを使用する ADC HA セットアップでは、ログイン時に次の警告メッセージが表示されます。

  高いパフォーマンスと可用性を実現するには、共有コアマシンから Google Cloud Platform 上の汎用またはコンピューティング/メモリ最適化インスタンスタイプに移行することをお勧めします。

  [NSPLAT-23748]

- **Azure DV5** シリーズでの **Citrix ADC VPX** インスタンスのサポート

  Azure クラウド上の NetScaler VPX インスタンスを Azure DV5 シリーズの仮想マシンで実行できるようになりました。

  [NSPLAT-22730]

- **NetScaler MPX 16000** プラットフォームのサポート

  このリリースは、NetScaler MPX 16000 プラットフォームをサポートしています。このプラットフォームには、2 つの 16 コアプロセッサと 128 GB (16 x 8 GB DIMM) のメモリが搭載されています。アプライアンスには、合計 8 つの 25G SFP+ ポートと 4 つの 100G QSFP28 イーサネットポートが搭載されています。
  詳細については、https://docs.citrix.com/en-us/citrix-hardware-platforms/mpx/netscaler-hardware-platforms/mpx-16000.htmlを参照してください。

  [NSPLAT-25436]

- **NetScaler SDX 16000** プラットフォームのサポート

  このリリースは、NetScaler SDX 16000 プラットフォームをサポートしています。このプラットフォームには、2 つの 16 コアプロセッサと 256 GB (16 x 16 GB DIMM) のメモリが搭載されています。アプライアンスには、合計 8 つの 25G SFP+ ポートと 4 つの 100G QSFP28 イーサネットポートが搭載されています。
  詳細については、https://docs.citrix.com/en-us/citrix-hardware-platforms/sdx/hardware-platforms/sdx-16000.htmlを参照してください。

  [NSPLAT-21608]

SSL

- 証明書の有効期限が切れるまでの定期的な通知のサポート

  NetScaler アプライアンスは、証明書の有効期限が切れるまで、1 日に 1 つの通知を送信するようになりました。以前は、証明書の有効期限が切れる前に設定された日数だけ通知が送信されていました。

  [NSSSL-11874]

システム

- **Syslog** 接続障害を警告するための **SNMP** アラーム

  外部の syslog サーバーへのネットワーク接続障害について警告するために、新しい SNMP アラーム「SyslogConnectionDropped」が NetScaler アプライアンスに導入されました。

  [NSBASE-16823]

ユーザーインターフェイス

- Subscription Advantage の日付が異なる 1 つまたは複数のライセンスファイルをアップロードすると、NetScaler Console はそれらを 1 つのプールにマージできません。その結果、NetScaler インスタンスは、ライセンスファイルの制限を超えると容量をチェックアウトできません。

  [NSCONFIG-6590, NSHELP-30854]

解決された問題

ビルド 13.1-37.38 で対処されている問題。

AppFlow

- AppFlow を構成すると、アプライアンスがバックエンドサーバーから空の HTTP チャンク応答を受信すると、NetScaler アプライアンスは TCP 接続をリセットします。

  この問題は、関連する AppFlow アクションの「ClientSideMeasurements」パラメーターが有効になっている場合に発生します。

  [NSHELP-32250]

認証、承認、監査

- アプライアンスに Standard Edition ライセンスがある場合、NetScaler アプライアンスを再起動すると、NO_AUTHN 認証アクションは持続しません。

  [NSHELP-32522]

- NetScaler Gateway GSLB セットアップでは、次の条件が満たされると、GSLB サイト間のプロキシ接続ループが検出される場合があります。

  - すべての GSLB サイトが同じバージョンではありません。
  - NetScaler Gateway は高度な認証を使用して構成されています。

  [NSHELP-32487]

- NetScaler アプライアンスは、Content-Type ヘッダーの文字セットサフィックスを削除し、次の両方を構成した場合に`Content-Type: application/x-www-form-urlencoded`を送信します。

  - SSO フォームベース認証
  - `nsapimgr knob - nsapimgr_wr.sh -ys call=ns_formsso_use_ctype_simple_en knob`

  [NSHELP-31977]

- SAML 認証が設定されている場合、ログアウト時に問題が発生する可能性があります。

  [NSHELP-31962]

- SSO の処理に必要なベアラートークンがないトラフィックで SSO が有効になっていると、シングルサインオン (SSO) が失敗します。

  [NSHELP-31362]

キャッシュ

- キャッシュされたコンテンツがクライアントに提供されると、NetScaler アプライアンスがクラッシュします。

[NSHELP-31760]

- キャッシュ制御ブロックで「max_age」および「s_maxage」パラメータ値が動的に設定されていないと、NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-27758]

NetScaler SDX アプライアンス

- NetScaler SDX アプライアンスの GUI では、ユーザーがイベントルールに障害オブジェクトを追加すると、入力フィールドがクロスサイトスクリプティング攻撃の影響を受けやすく、保存されているクロスサイトスクリプティングに対してページセキュリティが脆弱になりました。この問題を防ぐため、ユーザー入力が有効であることを確認するために、入力フィールドがサニタイズされるようになりました。

  [NSHELP-32600]

NetScaler Gateway

- Gateway Insight と Web Insight の機能のいずれかまたは両方が有効になっていると、NetScaler アプライアンスがクラッシュします。

  [NSHELP-33345, NSHELP-33347]

- 接続ブローカーの存在下では、RDP プロキシが機能しないことがあります。

  [NSHELP-33063]

- NetScaler Gateway アプライアンスのポートが不足しているため、アプリケーションを NetScaler Gateway 経由で起動できない場合があります。

  [NSHELP-32418]

- クライアントレス VPN アクセス用に構成された NetScaler Gateway アプライアンスは、ダミーセッションの処理中にクラッシュする可能性があります。

  [NSHELP-32399]

- HDX Insight が有効になっていると、NetScaler Gateway アプライアンスがクラッシュする可能性があります。

  [NSHELP-32120]

- UDP セッションを起動すると、セッションを閉じた後でも古い接続が存在しているように見えます。ただし、これらは実際には古い接続ではなく、カウンタの問題です。

  [NSHELP-32009]

- ユーザーが NetScaler アプライアンスにログオンし、Citrix Workspace がインストールされていない場合、Citrix Workspace をダウンロードするためのリンクが誤って Citrix Receiver を指します。

  [NSHELP-31877]

- Gateway Insight の認証失敗レコードには、NOAUTH が第 1 要素として設定されていて、認証情報が無効であるために 2 番目の要素認証が失敗した場合、ユーザー名は「匿名」と表示されます。この問題は、nFactor ビジュアライザーでは設計上、1 番目のファクターが NOAUTH として設定されているため、nFactor ビジュアライザーを使用して構成を実行した場合にのみ発生します。

  [NSHELP-31795]

- 「show vpn icaconnection」コマンドでは、ICA 接続のシリアル番号が正しく表示されません。この問題は、「show vpn icaconnection」コマンドの実行時にシリアル番号が任意にリセットされるために発生します。

  [CGOP-22205]

NetScaler Web App Firewall

- 次のソフトウェアバージョンで Citrix Web App Firewall のシグネチャオブジェクトを構成すると、スタンドアロンの NetScaler アプライアンスまたは HA セットアップのセカンダリモードがクラッシュする可能性があります。

  - 13.0 ビルド 88.5 およびそれ以降
  - 13.1 ビルド 33.41 およびそれ以降

  [NSHELP-33250]

- ブロック、ログ、`cookieHijackingAction` または統計を設定すると、NetScaler アプライアンスでメモリリークが発生します。

  [NSHELP-33187]

- NetScaler Web App Firewall では、コンテンツタイプのヘッダーにプロトコル（アプリケーション/pkcs7-signature）を指定すると、ヘッダーが正しく解析されません。その結果、ファイアウォールは有効なリクエストをブロックします。

  [NSHELP-32844]

- WAF プロファイルの復元時には、一部の緩和ルールがインポートされません。

  [NSHELP-32729]

- NetScaler Web App Firewall は、コマンドインジェクションの検出に時間がかかることがあります。その結果、Pitboss は NetScaler アプライアンスを再起動します。

  [NSHELP-32654]

- 正当な Cookie はログに保存され、重複した Cookie 違反ログが表示されます。

  [NSHELP-32369]

負荷分散

- 特定のシナリオでは、サービスグループにバインドされたサーバーが、無効な Cookie 値を表示します。トレースログで正しい Cookie 値を確認できます。

  [NSHELP-21196]

その他

- NetScaler アプライアンスは、Web サーバーのログ機能のバッファサイズを 16MB ではなく 3MB という誤ったデフォルト値に設定しています。

  [NSHELP-32429]

ネットワーク

- NetScaler BLX クラスター設定では、次の操作はエラーメッセージなしで失敗します。

  - 強制基本レベルでのコンフィグレーションのクリア (「クリアコンフィグ-フォースベーシック」)
  - 強制拡張レベルでの設定のクリア (「設定をクリア-強制拡張」)
  - 強制拡張 + レベルでの設定のクリア (「設定をクリア-強制拡張 +」)

  [NSNET-27132]

- 高可用性セットアップでは、多数の LSN セッションをクリアしているときに、メモリ破損によりプライマリノードがクラッシュする可能性があります。

  [NSHELP-32467]

- 次の条件をすべて満たすと、NetScaler アプライアンスがクラッシュする可能性があります。

  - TTL ベースの ACL がタイムアウト
  - NetScaler アプライアンスには多数の ACL が設定されています。

  [NSHELP-31307]

プラットフォーム

- NetScaler MPX アプライアンスで Mellanox インターフェイスを無効にすると、インターフェイスにリンクされているピアスイッチは、リンクダウン状態ではなくリンクアップ状態で表示されます。

  [NSPLAT-24422]

- NetScaler VPX インスタンスは、次の条件の両方が満たされた場合、クライアントからのパケットをドロップします。

  - VPX インスタンスは、VMXNET3 アダプタを使用して AWS 上の VMware クラウドでホストされます。
  - VMXNET3 アダプタはパケットの RSS ハッシュを生成できません。

  [NSHELP-33150]

ポリシー

- NetScaler アプライアンスでは、NSPEPI ツールを使用して従来のポリシーから高度なポリシーに移行されたコンテンツスイッチポリシーは、次の条件が満たされると機能しない場合があります。

  - ポリシーはコンテンツスイッチ仮想サーバーにバインドされます。
  - 「CaseSensitive」パラメータはオフに設定されています。

  [NSHELP-31951]

SSL

- 仮想サーバーが Azure Key Vault に保存されている秘密鍵を使用するように構成されている場合、NetScaler アプライアンスは TLS 1.3 ハンドシェイク中にクラッシュする可能性があります。

  [NSHELP-32451]

- NetScaler アプライアンスは、次の条件が満たされるとクラッシュします。

  - ハンドシェイクが完了する前に、クライアントから別のクライアントに hello が送信されます。
  - リクエストには、最初のクライアント hello に特別な暗号セットが含まれています。

  [NSHELP-32422]

- クラスタ IP（CLIP）アドレスを介してアクセスされる NetScaler GUI には、SSL 仮想サーバーへのサーバー証明書バインディングは表示されません。

  [NSHELP-31602]

- 有効な CA 証明書がデフォルトの証明書バンドルに存在しない場合、SSL インターセプト中に OCSP 応答検証が失敗することがあります。この失敗は、設定された証明書バンドルの代わりにデフォルトの証明書バンドルを使用して OCSP 応答検証が誤って行われたことが原因です。

  [NSHELP-30594]

システム

- NetScaler Console サーバーは、固有の URL を含む大量の HTTP トラフィックを受信すると、大量のメモリを消費します。その結果、NetScaler コンソールサーバーにアクセスできなくなります。

  [NSHELP-32922]

- NetScaler アプライアンスでは、ヘッダー変更フレームワークによりメモリが破損します。この状態は、NetScaler アプライアンスが消費するクッキーが転送される前に特定の順序で削除された場合に発生します。

  [NSHELP-32799]

- HTTP リクエストで PATCH メソッドを使用すると、VPN 認証が失敗します。この問題は、HTTP PATCH メソッドが未知の認証方法として認識されるために発生します。

  [NSHELP-32214]

- コンテンツ検査機能を使用する場合、ペイロードによる Rewrite ヘッダーの挿入が正しく機能しない場合があります。

  [NSHELP-30088]

ユーザーインターフェイス

- Management Service ライセンスページでは、ライセンスノードにアクセスしたり更新したりしても、プールされたライセンス情報は更新されません。代わりに、プールされたライセンス情報は、ログアウトして再度ログインしたときにのみ更新されます。

  [NSHELP-33203]

- ユーザーがコンテンツスイッチングポリシーのバインドを表示しても、コンテンツスイッチング仮想サーバーの詳細は [バインディングの表示] の同じ行に表示されません。

  [NSHELP-33149]

- ユーザーがトラフィックポリシーをコンテンツスイッチングまたは負荷分散仮想サーバーにバインドしても、バインディングの詳細は GUI に表示されません。

  [NSHELP-32751]

- NetScaler GUI を使用して NetScaler アプライアンスを次のいずれかのビルドにアップグレードまたはダウングレードすると、失敗することがあります。

  - リリース 13.1 ビルド 30.52
  - リリース 13.1 ビルド 27.59

  [NSHELP-32673]

- NetScaler GUI を使用して、カスタムパーティションでホストされている DNS および DNS_TCP プロトコルを使用して仮想サーバーを作成または編集しているときに、次のエラーが表示されます。

  ```
  Error: Invalid object name [lbvserver_scpolicy_binding]
  ```

  [NSHELP-32534]

- NetScaler GUI には次の問題が表示されます。

  - NetScaler GUI を使用してサーバー証明書が SSL 仮想サーバーにバインドされている場合、証明書バインディングは GUI に表示されません。CA 証明書のバインドは GUI に通常どおり表示されます。
  - 組み込みレスポンダーポリシーの非表示ボタンをクリックすると、手動で作成されたレスポンダーポリシーも非表示になります。

  クラスター設定では、NetScaler GUI にさらに次の問題が表示されます。

  - 暗号グループを内部サービスにバインドすると、エラーが発生して失敗します。
  - 組み込みの書き換えアクションは GUI には表示されません。

  [NSHELP-32499]

- 管理パーティションのある NetScaler アプライアンスでは、パーティション内の「ns」パラメータ設定は再起動後に失われます。この状態は、組み込み構成が間違っているために発生します。

  [NSHELP-32486]

- ユーザーがログインした後、NetScaler アプライアンスのログインページに有効なユーザー名が表示されない場合があります。

  [NSHELP-31759]

- 高可用性セットアップでは、HA 構成の同期後に暗号化された構成がセカンダリノードで失われます。

  [NSHELP-30897]

既知の問題

リリース 13.1-37.38 に存在する問題。

AppFlow

- HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

  [NSINSIGHT-943]

認証、承認、監査

- NetScaler アプライアンスは、重複したパスワードログインの試行を認証せず、アカウントのロックアウトを防ぎます。

  [NSHELP-563]

- NetScaler アプライアンスでコンテンツセキュリティポリシー（CSP）機能が有効になっている場合、DUO 認証は失敗します。

  [NSAUTH-12687]

- 管理者は、認証情報が無効であることが原因で発生した認証エラーのカスタムロギングを実行できません。この問題は、NetScaler Responder ポリシーがログインエラーのエラーを検出できないために発生します。

  [NSAUTH-11151]

- ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
  `show adfsproxyprofile <profile name>`

  回避策：クラスター内のアクティブなプライマリの NetScaler に接続し、`show adfsproxyprofile <profile name>`コマンドを実行します。プロキシプロファイルの状態が表示されます。

  [NSAUTH-5916]

- 次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

  - [LDAP 到達可能性をテスト] オプションが開きます。
  - 無効なログイン認証情報が入力され、送信されます。
  - 有効なログイン認証情報が入力され、送信されます。

  回避策：「LDAP 到達可能性テスト」オプションを閉じて開きます。

  [NSAUTH-2147]

NetScaler SDX アプライアンス

- 次の条件が満たされている場合、NetScaler SDX アプライアンスでホストされている VPX インスタンスでパケットドロップが表示されます。

  - スループット割り当てモードはバーストです。
  - スループットと最大バーストキャパシティには大きな違いがあります。

  [NSHELP-21992]

NetScaler Gateway

- Citrix Secure Access クライアントのバージョン 21.7.1.2 以降では、管理者権限のないユーザーが新しいバージョンにアップグレードできません。この問題は、Citrix Secure Access クライアントのアップグレードが NetScaler アプライアンスから行われる場合にのみ発生します。

  [NSHELP-32793]

- ユーザーが Windows 向け Citrix Secure Access 画面の［ホームページ］タブをクリックすると、ページに接続拒否エラーが表示されます。

  [NSHELP-32510]

- Chrome を使用する Mac デバイスで、2 つの FQDN にアクセス中に VPN 拡張機能がクラッシュします。

  [NSHELP-32144]

- NetScaler Gateway リリース 13.0 または 13.1 のプロキシ設定が空の場合、Citrix SSO によって不適切なプロキシ設定が作成されることがあります。

  [NSHELP-31970]

- Citrix Secure Access クライアントのデバッグログ制御は、NetScaler Gateway に依存せず、マシントンネルとユーザートンネルの両方のプラグイン UI から有効または無効にできるようになりました。

  [NSHELP-31968]

- Citrix Secure Access によって確立されたトンネルの外部にあるリソースへの直接接続は、大幅な遅延または輻輳が発生すると失敗することがあります。

  [NSHELP-31598]

- カスタマイズされた EPA 障害ログメッセージは、NetScaler Gateway ポータルには表示されません。代わりに、「内部エラー」というメッセージが表示されます。

  [NSHELP-31434]

- ユーザーが Always-On サービスモードで Windows マシンにログインすると、Windows 自動ログオンが機能しないことがあります。マシントンネルはユーザートンネルに遷移しません。「Connecting…」というメッセージが VPN プラグイン UI に表示されます。

  [NSHELP-31357, CGOP-21192]

- Always on が設定されている場合、aoservice.exe ファイルのバージョン番号（1.1.1.1）が正しくないため、ユーザトンネルが失敗します。

  [NSHELP-30662]

- 「NetworkAccessonVPNFailure」プロファイルパラメータを`fullAccess`から`onlyToGateway`に変更すると、ユーザーは NetScaler Gateway アプライアンスに接続できなくなります。

  [NSHELP-30236]

- ゲートウェイのホームページは、ゲートウェイプラグインが VPN トンネルを正常に確立した直後には表示されません。この問題を解決するために、次のレジストリ値が導入されます。

  HKLMSoftwareCitrixSecure Access ClientSecureChannelResetTimeoutSeconds
  タイプ: DWORD

  デフォルトでは、このレジストリ値は設定も追加もされません。「SecureChannelResetTimeoutSeconds」の値が 0 または追加されていない場合、遅延を処理するための修正が機能しません。これはデフォルトの動作です。管理者は、このレジストリをクライアントに設定して修正を有効にする必要があります (つまり、ゲートウェイプラグインが VPN トンネルを正常に確立した直後にホームページを表示する)。

  [NSHELP-30189]

- Windows VPN クライアントは、サーバからの「SSL close notify」アラートを尊重せず、同じ接続で転送ログイン要求を送信します。

  [NSHELP-29675]

- rdx.js ファイルにいくつかの Citrix 内部 IP アドレスがあることに気付くかもしれません。

  [NSHELP-28682]

- macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

  [NSHELP-28551]

- クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

  [NSHELP-28404]

- 次の条件を満たす場合、Windows ログオン後に VPN プラグインはトンネルを確立しません。

  - NetScaler Gateway アプライアンスが常時オン機能用に構成されている
  - アプライアンスは、2 要素認証を「オフ」にした証明書ベースの認証用に設定されています。

  [NSHELP-23584]

- スキーマをブラウズしているときに、「未定義のプロパティ「タイプ」を読み取れません」というエラーメッセージが表示されることがあります。

  [NSHELP-21897]

- NetScaler クラスター設定では、HDX Insight と Gateway Insight を同時に有効にすることはできません。

  [CGOP-22849]

- Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

  [CGOP-19355]

- 無効な STA チケットによるアプリケーションの起動失敗は、Gateway Insight では報告されません。

  [CGOP-13621]

- Gateway Insight レポートでは、SAML エラーエラーの認証タイプフィールドに「SAML」ではなく「Local」という値が誤って表示されます。

  [CGOP-13584]

- 高可用性セットアップでは、NetScaler フェイルオーバー時に、NetScaler Console のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

  [CGOP-13511]

- ICA 接続が MAC Receiver バージョン 19.6.0.32 または Citrix Virtual Apps and Desktops バージョン 7.18 から起動されると、HDX Insight 機能は無効になります。

  [CGOP-13494]

- EDT Insight 機能が有効になっていると、ネットワークの不一致時にオーディオチャネルに障害が発生することがあります。

  [CGOP-13493]

- ブラウザからローカルホスト接続を受け付けると、macOS の Accept Connection ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

  [CGOP-13050]

- 一部の言語では、Citrix SSO アプリ > ホームページの「ホームページ」というテキストが切り捨てられます。

  [CGOP-13049]

- NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

  [CGOP-11830]

- Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

  [CGOP-7269]

負荷分散

- 高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

  [NSLB-7679]

- サービスグループの`entityofs`トラップの ServiceGroupName 形式は次のとおりです。
  `<service(group)name>?<ip/DBS>?<port>`

トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (“?”) は区切り文字として使用されます。NetScaler は、疑問符 (” ?”)。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

[NSHELP-28080]

その他

- 高可用性セットアップで強制同期が行われると、アプライアンスはセカンダリノードで「set urlfilter parameter」コマンドを実行します。
  その結果、セカンダリノードは、「TimeOfDayToUpdateDB」パラメーターで指定された次のスケジュール時刻まで、スケジュールされた更新をスキップします。

  [NSSWG-849]

- レジストリ値が 2000 バイトを超えると、AlwaysOnAllow リストレジストリが期待どおりに動作しません。

  [NSHELP-31836]

- URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

  [NSHELP-22409]

ネットワーク

- DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

  [NSNET-25299]

- DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

  - NetScaler BLX アプライアンスには、少数の「巨大ページ」が割り当てられています。たとえば、1G です。
  - NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

  この問題は、エラーメッセージとして「/var/log/ns.log」に記録されます。

  -「BLX-DPDK: DPDK メモリプールを PE-x 用に初期化できませんでした」

  注:x はワーカープロセス数以下の数です。

  回避策:「巨大ページ」を多数割り当ててから、アプライアンスを再起動します。

  [NSNET-25173]

- DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

  [NSNET-24449]

- DPDK を搭載した NetScaler BLX アプライアンスの Intel X710 10G (i40e)インターフェイスでは、次のインターフェイス操作はサポートされていません:

  - 無効化
  - 有効化
  - リセット

  [NSNET-16559]

- Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

  「次のパッケージの依存関係は満たされていません:blx-core-libs: i386: preDepends: libc6: i386 (>= 2.19) しかしインストールできません」

  回避策：NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します:

  - dpkg –add-architecture i386
  - apt-get update
  - apt-get dist-upgrade
  - apt-get install libc6:i386

  [NSNET-14602]

- FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

  [NSNET-5233]

- NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

  [NSHELP-21082]

プラットフォーム

- Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。「rm cloudprofile」コマンドを使用してプロファイルを削除します。

[NSPLAT-4520]

- Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
  回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノード上で設定する必要があります。

  [NSPLAT-4451]

- NetScaler SDX 8015/8400/8600 プラットフォームでは、Xen サーバーのメモリ消費量が増加する可能性があります。
  回避策：Xen Server で次のコマンドを実行し、アプライアンスを再起動します。
  /opt/xensource/libexec/xen-cmdline –set-xen “dom0_mem=1024M,max:1024M”

  [NSHELP-32260]

ポリシー

- 処理データのサイズが設定されたデフォルトの TCP バッファーサイズを超えると、接続がハングアップすることがあります。

  回避策:TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

  [NSPOLICY-1267]

SSL

- NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

  回避方法：

  1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例：`set ssl vserver <name> -SSL3 DISABLED`。
  2. 構成を保存します。

  [NSSSL-9572]

- 認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

  [NSSSL-6478]

- 同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

  [NSSSL-6213]

- HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
  エラー:CRL 更新は無効です

  [NSSSL-6106]

- セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

  [NSSSL-4427]

- SSL プロファイル内の SSL プロトコルまたは暗号を変更しようとすると、「警告:SSL vserver/Service で使用可能な暗号が設定されていません」という誤った警告メッセージが表示されます。

  [NSSSL-4001]

- 期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

  [NSSSL-3184, NSSSL-1379, NSSSL-1394]

システム

- 次の条件が満たされると、TCP 接続の RTT が高くなります。

  - 最大輻輳ウィンドウ（> 4 MB）が高く設定されている
  - TCP NILE アルゴリズムは有効になっています

  NetScaler アプライアンスが NILE アルゴリズムを使用して輻輳制御を行うには、条件がスロースタートのしきい値と最大輻輳ウィンドウを合わせた値を超える必要があります

  そのため、設定された最大輻輳時間帯に達するまで、NetScaler はデータを受け付け続け、最終的には RTT が高くなります。

  [NSHELP-31548]

- アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

  [NSHELP-21240]

- mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

  [NSHELP-10972]

- まれに、NetScaler が次の条件をすべて満たすと、HTTP/2 クライアント接続で内部エラーで HTTP/2 GoAway フレームを開始することがあります。

  - クライアントまたはバックエンドサーバーは、クライアントの HTTP/2 接続上の最後の WebSocket または Connect ストリームを閉じようとします。
  - 多重化は有効になっています。

このエラーは、クライアントの HTTP/2 接続で進行中のトランザクションには影響しません。

回避策: 以下のコマンドを使用して、関連する HTTP/2 プロファイルの接続多重化を無効にします:

```
set httpProfile <name> [-conMultiplex ( ENABLED | DISABLED )]
```

[NSBASE-17449]

- クラスタ展開では、CCO 以外のノードで「force cluster sync」コマンドを実行すると、ns.log ファイルには重複するログエントリが含まれます。

  [NSBASE-16304、NSGI-1293]

- NetScaler Console を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに機能しません。

  回避策：管理ポッドを再起動します。

  [NSBASE-15556]

- LogStream トランスポートタイプが Insight 用に構成されている場合、クライアント IP とサーバー IP は HDX Insight SkipFlow レコードで反転されます。

  [NSBASE-8506]

ユーザーインターフェイス

- MQTT リライト機能では、GUI のエクスプレッションエディタを使用してエクスプレッションを削除することはできません。

  回避策:CLI を使用して MQTT タイプのアクションの追加または編集コマンドを使用します。

  [NSUI-18049]

- NetScaler GUI では、「ダッシュボード」タブの下にある「ヘルプ」リンクが壊れています。

  [NSUI-14752]

- CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

  回避策：NetScaler GUI または CLI を使用して IPSec プロファイル、IP トンネル、および PBR ルールを追加して CloudBridge Connector を構成します。

  [NSUI-13024]

- GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

  [NSUI-6838]

- 高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

  - HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

回避策：HA 同期が完了した後にのみ、手動で HA フェイルオーバーを連続して実行してください（両方のノードが同期成功状態になっています）。

[NSHELP-25598]

- NetScaler BLX アプライアンスの高可用性セットアップでは、プライマリノードが応答しなくなり、CLI または API 要求がブロックされることがあります。

  回避策: プライマリノードを再起動します。

  [NSCONFIG-6601]

- あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

  1. NetScaler アプライアンスをいずれかのビルドにアップグレードします
     - 13.0 52.24 ビルド
     - 12.1 57.18 ビルド
     - 11.1 65.10 ビルド
  2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
  3. NetScaler アプライアンスを古いビルドにダウングレードします。

  CLI を使用してこれらのシステムユーザーのリストを表示するには、
  コマンドプロンプトで次のように入力します。

  ```
  query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
  ```

  回避策: この問題を解決するには、以下のいずれかの独立したオプションを使用してください。

  - NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3）、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
  - アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
  - 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

  詳細については、「ルート管理者 (nsroot) パスワードをリセットする方法」を参照してください。

  [NSCONFIG-3188]

NetScaler 13.1-33.54 リリースのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリースビルド 13.1-33.54 の機能強化と変更、修正された問題と既知の問題について説明します。

メモ

- このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正と勧告のリストについては、セキュリティ速報を参照してください。
- ビルド 13.1-33.47 以降のビルドは、https://support.citrix.com/article/CTX463706で説明されているセキュリティの脆弱性を解決します。
- ビルド 33.54 は、ビルド 33.52、ビルド 33.49、ビルド 33.47 を置き換えます。
- ビルド 33.54 には、NSHELP-33250、NSHELP-33345、および NSHELP-33063 の問題に対する修正が含まれています。
- ビルド 33.52 には、NSHELP-32907 の問題の修正が含まれています。
- ビルド 33.49 には、NSHELP-32709、NSHELP-32697、NSHELP-32410、NSHELP-31790、NSHELP-31478、および NSCONFIG-7098 の問題に対する修正が含まれていました。

新機能

ビルド 13.1-33.54 で利用できる機能強化と変更点。

ボット管理

- **BOT** 関連の新表現

  次の式が追加され、BOT プロファイルがロギングモードに設定されている場合に使用できます。

  - `HTTP.REQ.BOT.IS_SUSPECTED` -クライアントが BOT であると疑われる場合は true を返します。
  - `HTTP.REQ.BOT.TYPE.EQ(<bot type>)` -クライアントの BOT タイプが引数と同じ場合は true を返します。ボットタイプに設定可能な値: 良好、不良、不明
  - `HTTP.REQ.BOT.TYPE.NE(<bot type>)` -クライアントの BOT タイプが引数と同じでない場合は true を返します。ボットタイプに設定可能な値: 良好、不良、不明
  - `HTTP.REQ.BOT.TYPE.ENUM_NAME` -BOT タイプを文字列で返します。たとえば、良い、悪い、不明。
  - `HTTP.REQ.BOT.DETECTION_METHODS` -クライアントを BOT として検出するために使用される検出手法のリスト。

  [NSBOT-842]

NetScaler Gateway

- SmartControl を設定すると、対応する認証、承認、および監査セッションが存在しない場合でも、セッションの信頼性がサポートされます。ネットワーク障害からの回復後に NetScaler アプライアンスがクライアントデバイスから受信した再接続要求は、対応する認証、承認、および監査セッションが存在しない場合でも処理されます。

  [CGOP-21040]

NetScaler Web App Firewall

- 新しいデフォルト **Web App Firewall** プロファイル

  コア WAF 保護機能を備えたコアと呼ばれる新しいデフォルトプロファイルが利用可能になりました。コアプロファイルでは次のチェックが有効になっています。

  - 文法ベースの SQL インジェクション
  - 文法ベースの CMD インジェクション
  - XSS か
  - BOF
  - ブロック表現

  [NSWAF-9133]

- **JSON** ペイロードのカスタムキーワードサポート

  任意のキーワードを追加して、設定したキーワードが JSON ペイロードに存在するかどうかを確認できます。構成されたキーワードが着信要求で検出された場合、NetScaler アプライアンスを構成して、要求をブロックしたり、ログを更新したり、ログカウンタを増やしたりすることができます。

  利点は、SQL インジェクションとコマンドインジェクションのチェックでカバーされていないキーワードを追加して、誤検出を減らすことができることです。

  [NSWAF-9076]

プラットフォーム

- **NetScaler** ライセンスの不正使用を防ぐ

  NetScaler アプライアンスをバージョン 13.1 にアップグレードする場合、NetScaler ライセンスシステムは、Customer Success Services 有効期限に従ってライセンス検証を実施するようになりました。この日付が Customer Success Services 対象日より前の場合、既存のライセンスはアップグレードされたバージョンの ADC アプライアンスでは機能しません。この動作により、ライセンスの不正使用を防ぐことができます。

  NetScaler 製品とその対象期間の一覧については、https://support.citrix.com/article/CTX111618/citrix-product-customer-success-services-eligibility-datesを参照してください。

[NSPLAT-24522]

- **Azure** 高速ネットワークでの動的 **NIC** 削除の処理

  NetScaler VPX インスタンスは、Azure アクセラレーテッドネットワークでの動的な NIC の取り外しと、取り外された NIC の再接続をシームレスに処理できるようになりました。

  Azure では、ホストのメンテナンス作業のために、高速ネットワークのシングルルート I/O 仮想化 (SR-IOV) 仮想機能 (VF) NIC を削除できます。NIC が Azure VM から削除されるたびに、NetScaler VPX インスタンスはインターフェイスのステータスを「リンクダウン」と表示し、トラフィックは仮想インターフェイスのみを経由します。取り外した NIC が再接続されると、VPX インスタンスは再接続された SR-IOV VF NIC を使用します。このプロセスはシームレスに行われ、設定は不要です。

  [NSPLAT-23300]

- **Python 3.7** のサポート

  Python 2.7 が廃止されたため、NetScaler アプライアンスは Python 3.7 をサポートするようになりました。

  現在の Python スクリプトを Python 3.7 と互換性を持たせるにはアップグレードする必要があります。

  [NSPLAT-20832]

SSL

- 証明書の有効期限が切れるまでの定期的な通知のサポート

  NetScaler アプライアンスは、証明書の有効期限が切れるまで、1 日に 1 つの通知を送信するようになりました。以前は、証明書の有効期限が切れる前に設定された日数だけ通知が送信されていました。

  [NSSSL-11874]

- 証明書作成リクエストのメールアドレスの長さの増加

  NetScaler アプライアンスでは、証明書作成要求の電子メールアドレスの制限が 255 文字に引き上げられました。以前の制限は 39 文字でした。

  [NSSSL-10917]

- **Intel Coleto** および **Intel Lewisburg** ベースのプラットフォームでの **Thales Luna HSM** のサポート

  Thales Luna HSM は現在、NetScaler、Intel Coleto、および Intel Lewisburg SSL チップベースのプラットフォームでサポートされています。

  以下のアプライアンスには、Intel Coleto チップが搭載されています。

  - MPX 5900
  - MPX/SDX 8900
  - MPX/SDX 15000

- MPX/SDX 15000-50G
- MPX/SDX 26000
- MPX/SDX 26000-50S
- MPX/SDX 26000-100G

以下のプラットフォームには、Intel Lewisburg チップが付属しています。

- MPX 9100
- SDX 9100

[NSSSL-9707]

システム

- **HTTP** プロファイルに新しいパラメータが追加されました

  バックエンドサーバーへの攻撃を防ぐために、新しいパラメーター PassProtocolUpgrade が HTTP プロファイルに追加されました。このパラメータの状態に応じて、アップグレードヘッダーはバックエンドサーバーに送信されるリクエストで渡されるか、リクエストを送信する前に削除されます。

  - PassProtocolUpgrade パラメーターが有効になっている場合、アップグレードヘッダーはバックエンドに渡されます。サーバーはアップグレード要求を受け入れ、応答で通知します。
  - このパラメータを無効にすると、アップグレードヘッダーが削除され、残りのリクエストがバックエンドに送信されます。

  PassProtocolUpgrade パラメータが次のプロファイルに追加されます。

  - nshttp_default_profile はデフォルトで有効になっています
  - nshttp_default_strict_validation デフォルトでは無効になっています
  - nshttp_default_internal_apps はデフォルトで無効になっています
  - nshttp_default_http_quic_profile はデフォルトで有効になっています

  Citrix では、このパラメータをデフォルトでは無効にすることを推奨しています。詳細については、『NetScaler セキュア導入ガイド』を参照してください。

  [NSBASE-17423]

- 複数の時系列プロファイルのサポート

  NetScaler アプライアンスは、最大 3 つの時系列プロファイル構成をサポートするようになりました。
  各時系列プロファイルには次の内容を設定できます。

  - そのコレクター
  - メトリクスコレクターがエクスポートするのに必要なカウンターのセットを含むスキーマファイル
  - メトリックをエクスポートできるデータ形式。
  - 指標、監査ログ、イベントを有効または無効にするオプション。

複数の時系列プロファイルがサポートされているため、メトリックコレクターは（構成されたスキーマファイルに基づいて）メトリックの異なるセットを異なるフォーマット（AVRO、Prometheus、Influx）のさまざまなコレクターに同時にエクスポートできます。

詳細については、「AppFlow 機能の構成」を参照してください。

[NSBASE-16809]

- syslog は、特定の時間間隔で TCP 経由でエクスポートされません。この状態のため、syslog は監査バッファーに無期限に留まり、ログがないことが認識されます。この syslog は、バッファーがいっぱいになったときにのみ送信されます。

  今回の修正により、syslog は、監査バッファーがいっぱいになったとき、または 20 秒おきに、いずれか早いほうの間隔で TCP 経由でエクスポートされます。

  [NSBASE-16698]

- **QUIC** の暗号オフロードサポート

  NetScaler アプライアンスは、ソフトウェアからハードウェアへの暗号処理のオフロードをサポートするようになりました。これにより、QUIC トランザクションが高速化されます。NetScaler アプライアンスには、暗号アクセラレーションを透過的に実行する SSL ハードウェアチップが搭載されています。

  詳細については、QUICを参照してください。

  [NSBASE-12046]

ユーザーインターフェイス

- 内部サービスの **TLS 1.2** 設定に基づく安全な **RPC** 通信

  NetScaler アプライアンスを以下のいずれかのビルドからリリース 13.1 ビルド 33.x 以降にアップグレードすると、内部 RPCS および KRPCS サービスの TLS 1.2 設定（有効または無効）に基づいて、RPC ノードの「セキュア」オプションが有効または無効になります。

  - リリース 13.0 ビルド 64.35 以前
  - リリース 12.1 ビルド 61.18 またはそれ以前

  「セキュア」オプションが有効になっている場合、RPC 通信は次の設定の NetScaler ノード間で暗号化されます。

  - 高可用性
  - クラスター
  - GSLB か

  「セキュア」オプションでは、NetScaler ノード間の RPC 接続にセキュアプロトコル TLS1.2 とポート番号 3008 および 3009 を使用します。

安全な RPC 通信を確保するために、Citrix ではこれらのセットアップをアップグレードする前に次の操作を実行することをお勧めします。

- 内部 RPCS および KRPCS サービスでは TLS 1.2 を有効にする必要があります。
  - nsrpcs-127.0.0.1-3008
  - nskrpcs-127.0.0.1-3009
  - nsrpcs-: 1l-3008
- 3008 と 3009 は、NetScaler ノード間のファイアウォールでブロック解除する必要があります。

NetScaler CLI または GUI を使用して、セキュアオプションを有効または無効にできます。

[NSCONFIG-6485]

- **NetScaler CPX** ライセンスアグリゲーターのサポート

  これで、NetScaler が提供する新しい Kubernetes マイクロサービスである NetScaler CPX ライセンスアグリゲーターを使用して、NetScaler CPX のライセンスを取得できるようになりました。NetScaler CPX を起動するときは、NetScaler CPX ライセンスアグリゲーターの IP アドレスまたはドメイン名を使用して環境変数 CLA を構成する必要があります。環境変数が構成されている場合、NetScaler CPX ライセンスアグリゲーターは、接続されているすべての NetScaler CPX の集約ライセンスをチェックアウトします。

  [NSCONFIG-6394]

- **NITRO API** インストールの非同期オプションのサポート「**NITRO API** のインストール」に新しいオプション「async」が導入されました。「async」オプションはインストール操作のジョブ ID を返します。これを「nsjob NITRO」API 呼び出しで使用すると、インストール操作のステータスの詳細を取得できます。

  例:

  次の curl リクエストの例では、インストール NITRO API を async オプションとともに使用しています。レスポンスペイロードにはジョブ ID が 2 として含まれています。

  Curl 要求:
  “curl -v -X POST -H “Content-Type: application/json” -u nsroot:examplepassword http://192.0.0.33/nitro/v1/config/install?warning=yes -d ‘{“install”: {“url”: “https://example-repo.citrite.net/build-13.1-36.11_nc_64.tgz” , “async” :” 1” }}’ ”

  レスポンスペイロード:

  ” { “install” :{ “url” : "<file path>", “y” : false, “l” : false, “a” : false, “enhancedupgrade” : false, “resizeswapvar” : false, “async” : true, “id” : “2” }”

  次の curl リクエストの例では、「nsjob NITRO API」を使用して、インストール操作の ID であるジョブ ID 2 のステータス詳細を取得します。

Curl 要求:
“curl -v -X GET -H “Content-Type: application/json” -u nsroot:examplepassword http://192.0.0.33/nitro/v1/config/nsjob/2”

レスポンスペイロード:

” { “errorcode” : 0, “message” : “Done” , “severity” : “NONE” , “nsjob” : [

{ “name” : “install” , “id” : “2” , “status” : “Success” , “progress” : “nInstallation has completed.nnReboot is required for configuration changes to take effect.Installation succeeded. Reboot required.n” , “timeelapsed” : 148, “errorcode” : “5221” , “message” : “The configuration changes will not take effect until the system is rebootedn” }

]}”

[NSCONFIG-5870]

解決された問題

ビルド 13.1-33.54 で対処されている問題。

認証、承認、監査

- NetScaler アプライアンスは、MEM_SSLVPN モジュールのメモリリークにより要求の処理を停止します。

  [NSHELP-32646]

- NetScaler Gateway Duo 認証ログオンページは、RFWebUI 以外のテーマでは読み込まれません。

  [NSHELP-32463]

- デバイスを NetScaler Gateway アプライアンスに登録すると、Citrix Secure Access（Citrix SSO）に「プッシュ登録に失敗しました」というメッセージが表示されます。

  [NSHELP-32461]

- LDAP 認証と SAML 認証の両方がカスケードで構成されている場合、ログオン中にエラーページが表示されます。

  [NSHELP-32378]

- Citrix Workspace アプリを使用したゲートウェイへの認証が成功しないことがあります。

  [NSHELP-32333]

- NetScaler アプライアンスでコンテンツセキュリティポリシー（CSP）機能が有効になっている場合、SAML 認証は失敗します。

  [NSHELP-32203]

キャッシュ

- 統合キャッシュ機能が有効で、アプライアンスのメモリが不足している場合、NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-22942]

NetScaler SDX アプライアンス

- NetScaler SDX アプライアンスでは、リリース 13.1 ビルド 30.52 からそれ以前のリリースまたはビルドにダウングレードしても、クリーンインストールオプションが機能しません。

  [NSSVM-5419]

- NetScaler VPX インスタンスを SDX と ADM を使用してバックアップすると、いくつかの冗長ハードウェアセキュリティモジュール（HSM）構成ファイルもバックアップされます。

  [NSHELP-32539]

- NetScaler SDX アプライアンスの管理サービス Syslog には、日付が誤って 2 回表示されます。

  [NSHELP-32311]

NetScaler Gateway

- Gateway Insight と Web Insight の機能のいずれかまたは両方が有効になっていると、NetScaler アプライアンスがクラッシュします。

  [NSHELP-33345]

- 接続ブローカーの存在下では、RDP プロキシが機能しないことがあります。

  [NSHELP-33063]

- HDX Insight が有効になっていて、ユーザーがログアウト後すぐに StoreFront にログインすると、NetScaler Gateway アプライアンスがクラッシュすることがあります。

  [NSHELP-32907, NSHELP-33079, NSHELP-33289]

- Patset ベースの MAC アドレス EPA スキャンは、デバイス証明書スキャンと同じ要素では機能しません。

  [NSHELP-32760]

- NetScaler アプライアンスは、認証トラフィックに使用される認証方法が不明な HTTP パケットをすべてドロップします。認証トラフィックに認証および承認仮想サーバーが使用されている場合、認証方法が不明な場合、負荷分散操作で問題が発生し、展開が中断されます。不明な認証方法は、デフォルトでは無効になっています。

  [NSHELP-32709]

- [ログイン情報の転送] ダイアログボックスには [転送] ボタンは表示されません。

  [NSHELP-32614]

- StoreFront でコールバック URL が構成されている場合、StoreFront サーバーからのログアウト要求 POST /CitrixAuthservice.asmx を処理中に NetScaler アプライアンスがクラッシュします。

  [NSHELP-32207]

- NetScaler Gateway アプライアンスでは、VPN パラメーターがセッションアクションレベルで設定されていない場合、グローバル VPN パラメーターは有効になりません。

  高可用性セットアップをアップグレードする前に、セカンダリアプライアンスの HA 同期を手動で無効にしてください。詳細については、https://docs.citrix.com/en-us/citrix-adc/13-1/upgrade-downgrade-citrix-adc-appliance/upgrade-downgrade-ha-pair.htmlを参照してください

  [NSHELP-31478, CGOP-21737]

- NetScaler Gateway ログオンページのタイトルとポータルテーマが正しく表示されません。

  [NSHELP-29202]

- IIP プール（IP アドレスとマスク）の構成中に、IP アドレスが範囲内の最初の IP アドレスと一致しない場合、NetScaler CLI および GUI には 1 つのブロックのみが表示され、すべては表示されません。

  例:
  バインド VPN vserver vpn_ssl-IntraneTip 172.168.1.1 255.255.255.0
  バインド VPN vserver vpn_ssl-IntranetIP 172.168.2.1 255.255.255.0

  この場合、VPN vserver vpn_ssl を表示しているときの CLI または GUI には 172.168.2.1 プールのみが表示され、172.168.2.2 は表示されません。

  [NSHELP-29084]

NetScaler Web App Firewall

- 次のソフトウェアバージョンで Citrix Web App Firewall のシグネチャオブジェクトを構成すると、スタンドアロンの NetScaler アプライアンスまたは HA セットアップのセカンダリモードがクラッシュする可能性があります。

  - 13.0 ビルド 88.5 およびそれ以降
  - 13.1 ビルド 33.41 およびそれ以降

  [NSHELP-33250]

- プロキシサーバーとプロキシポートが設定されている場合、WAF シグネチャの更新は失敗します。シグネチャ自動更新プロセスの毎時実行中、ADC アプライアンスは、設定されたプロキシサーバーとプロキシポートを経由する代わりに、自動更新ホストに接続して更新されたファイルをダウンロードします。その結果、自動更新ホストにアクセスできないと、更新が失敗します。

[NSHELP-32613]

- 次の条件が満たされると、NetScaler アプライアンスがクラッシュする可能性があります。
  - アプライアンスに高い負荷がかかっています。
  - 構成の変更が行われています。
  - 署名の削除には長い時間がかかります。

  [NSHELP-32454]

- ボットデバイスの指紋認証セッションリプレイ攻撃は、ドロップされるのではなくログに記録されます。

  [NSHELP-31949]

負荷分散

- `useencryptedPersistenceCookieset lb param` コマンドでオプションを有効にすると、サービスグループを変更すると Cookie ハッシュが変更されます。

  [NSHELP-32697]

- まれに、コンテンツスイッチング仮想サーバーで SSL セッション ID ベースの永続性と SSL セッションチケットベースの処理が有効になっていると、NetScaler アプライアンスがクラッシュしてコアダンプを生成することがあります。

  [NSHELP-32228]

- 設定した属性がサーバに存在しない場合でも、LDAP モニタのステータスはアップのままです。

  [NSHELP-32025]

その他

- `ns_hw_err.bash`が NetScaler アプライアンスで実行してハードウェアの問題を検出すると、正常なディスクがあっても「起動時に HDD が見つかりません」というエラーが表示されることがあります。

  [NSHELP-31571]

- 次の条件が満たされると、クラスターノードはパケットループに入ります。
  - 宛先 IP アドレスが CLIP の UDP パケットがクラスタノードに送信されます。
  - CCO は、クラスターインスタンスの存続期間中に 1 つのノードから別のノードに変更されました。

  [NSHELP-30804]

ネットワーク

- ConfigMaps でファイルベースのスタートアップ構成を使用すると、クラッシュ後に NetScaler CPX がデフォルトのルート構成を復元できません。この動作により、接続が失われます。

  [NSNET-27124]

- NetScaler アプライアンスが、UDP パケットの IP ヘッダーに誤った IP チェックサムを追加する可能性があります。

  [NSHELP-32587]

- NetScaler BLX クラスター設定では、次の条件が満たされると、VTYSH が起動しないことがあります。

  - Linux ホストが再起動すると、NetScaler BLX ルートヘルスインジェクション（RHI）プロセスのオーダーループが発生します。

  [NSHELP-32473]

- 仮想サーバーを削除すると、次の条件が満たされている場合、NetScaler アプライアンスは関連する VIP RHI 状態を誤って DOWN に設定します。

  - 仮想サーバーにはバックアップ仮想サーバーがあります。
  - 仮想サーバは DOWN 状態で、少なくとも 1 つのバックアップ仮想サーバが UP 状態です。

  [NSHELP-29972]

プラットフォーム

- AMD プロセッサ上で動作する NetScaler アプライアンスは、ソフトウェアバージョンをリリース 13.1 ビルド 30.x にアップグレードすると、起動中にクラッシュすることがあります。

  [NSPLAT-24968、NSHELP-32808]

- 高可用性フェイルオーバーは AWS および GCP クラウドでは機能しません。AWS および GCP クラウド、および NetScaler VPX オンプレミスでは、管理 CPU が 100% の容量に達する可能性があります。これらの問題はいずれも、次の条件が満たされた場合に発生します。

  1. NetScaler アプライアンスの初回起動時には、プロンプトが表示されたパスワードは保存されません。
  2. その後、NetScaler アプライアンスを再起動します。

  [NSPLAT-22013]

- Mellanox NIC を含む NetScaler SDX アプライアンスを、VLAN フィルタリングが無効になっているビルドからアップグレードし、管理サービスがアップグレードの一環として VLAN フィルタリングを無効にしようとすると、操作が失敗します。その結果、すべてのインターフェイスとチャネルで VLAN フィルタリングが有効になります。

  [NSHELP-32759]

- ファームウェアのアップグレード後、NetScaler MPX 5900/8900 アプライアンスの管理インターフェイスがダウンする可能性があります。その結果、アプライアンスにアクセスできなくなります。

  [NSHELP-31587]

ポリシー

- 次の条件が満たされると、NetScaler アプライアンスが patset によるポリシー追加中にクラッシュすることがあります。

  - NSB に関連するフラグが TCP の書き換えシナリオで間違った順序で設定されています。

  [NSHELP-31064]

SSL

- 仮想サーバーは、無効なパディングを含む TLS 1.3 レコードを受信すると、「予期しないメッセージ」アラートの代わりに、致命的な「decode_error」アラートを送信します。

  [NSSSL-11890]

- Intel QAT 対応の暗号アクセラレーションハードウェアを搭載した NetScaler MPX および SDX プラットフォームでは、TLS 1.3 接続を介して仮想サーバーに送信される要求に、SOURCEIP パーシステンスタイプが一貫して適用されません。つまり、1 つの送信元 IP アドレスから送信されたリクエストは、複数の異なるバックエンドサーバーに分散される可能性があります。

  [NSHELP-32410, NSHELP-32895, NSHELP-32572, NSHELP-32688]

- Cavium SSL カードを搭載した NetScaler アプライアンスが、クライアントに DTLS ALERT メッセージを送信しているときにクラッシュすることがあります。

  [NSHELP-32031]

- 証明書認証ルールが評価され、同じ要求で 2 回トリガーされると、NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-31785]

システム

- 管理者パーティションで AppFlow 機能を有効にできるのは、デフォルトパーティションで ULFD モードを有効にした後だけです。

  [NSHELP-32670]

- NetScaler アプライアンスは、受信 TCP セグメントに部分的な HTTP 要求メソッドが存在する場合、HTTP 要求を無効な要求として扱うことがあります。

  [NSHELP-32462]

- 次の条件が満たされると、NetScaler アプライアンスがクラッシュする可能性があります。

  - HTTP2 と SSL の組み合わせでメモリ使用量が多い場合、NetScaler アプライアンスはメモリを割り当てることができません。

  [NSHELP-32255]

- IP または PORT フィルターを使用して nstrace パケットキャプチャを開始すると、NetScaler アプライアンスが VPN セットアップでクラッシュします。

  [NSHELP-31790]

- 以下の条件が満たされると、gRPC クライアントは gRPC ステータスヘッダーの解析に失敗します。

  - gRPC ステータスヘッダーは、末尾ヘッダーだけに追加されるのではなく、先頭ヘッダーと末尾ヘッダーの両方に追加されます。

  [NSHELP-31640]

- SACK を有効にすると、NetScaler アプライアンスは再送信リストの最後の 1 バイト TCP セグメントを再送信しません。理由は次のとおりです。アプライアンスは、最後の 1 バイトの TCP セグメントを再送信リストの最後を示すダミーセグメントとして使用します。

  [NSHELP-28778]

ユーザーインターフェイス

- NetScaler GUI を使用して GSLB サービスを GSLB 仮想サーバーにバインドすることはできません。**GSLB** サービスグループバインディング **> GSLB** サービスバインディング **> GSLB** サービスのリストが空と表示されているためです。

  [NSHELP-32236]

- NetScaler GUI（[システム] > [ネットワーク] > [ルート]）を使用して静的ルートを変更すると、次のエラーメッセージが表示されて誤って失敗することがあります。

  - 「必要な引数が見つかりません [ゲートウェイ]」

  [NSHELP-32024]

- HA/Cluster セットアップでは、RSA 以外の SSH キーを設定すると、設定の同期が失敗します。たとえば、ECDSA キーや DSA キーなどです。

  [NSHELP-31675]

- NetScaler GUI で、**[**システム**] > [SNMP] > [**トラップ**]** の下に既存の **SNMP** トラップ宛先がある場合、その宛先の編集に失敗し、次のエラーメッセージが表示されます。

  – 「SNMP トラップを取得中にエラーが発生しました」

  [NSHELP-3161]

- NetScaler アプライアンスの GUI には、構成済みの SAML および OAuth IDP ポリシーの正しい数が表示されません。

  [NSHELP-31480]

- NetScaler アプライアンスでは、GUI インターフェイスを使用しているときに、レスポンダーポリシーページに次の問題が表示されます。

  – カスタム作成されたレスポンダーポリシーは、組み込みレスポンダーポリシーの下に表示される場合があります。

  [NSHELP-31428]

- NetScaler HA セットアップでは、構成を保存して更新ボタンをクリックすると、NetScaler GUI に次の問題が発生します。

  – アプライアンスに未保存の設定変更がない場合でも、GUI の [保存] ボタンにオレンジ色のドットが誤って表示されます。

  [NSHELP-30031]

- GSLB 仮想サーバーの統計情報は、管理パーティションモードでは使用できません。

  [NSHELP-28524]

- NetScaler Console からライセンスをチェックアウトした NetScaler アプライアンスは、アプライアンスが ADM から切断されると猶予期間に入ります。アプライアンスは ADM ではライセンスされていないように見え、ADM に再接続した後も猶予期間中も継続します。

  [NSCONFIG-7098]

既知の問題

リリース 13.1-33.54 に存在する問題。

AppFlow

- HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

  [NSINSIGHT-943]

認証、承認、監査

- CWA クライアントまたはネイティブ VPN クライアントによるゲートウェイ認証は、ns_aaa_relaystate_param_w patset に文字列がないために失敗する可能性があります。

  回避方法:

  ```
  bind policy patset ns_aaa_relaystate_param_whitelist "citrixauthwebviewdon
  ://"-index 1 -charset ASCII

  bind policy patset ns_aaa_relaystate_param_whitelist "citrixsso
  ://"-index 2 -charset ASCII

  bind policy patset ns_aaa_relaystate_param_whitelist "citrixng://
  "-index 3 -charset ASCII
  ```

  [NSHELP-33054]

- NetScaler アプライアンスは、Content-Type ヘッダーの文字セットサフィックスを削除し、次の両方を構成した場合にContent-Type: application/x-www-form-urlencodedを送信します。

  - SSO フォームベース認証
  - nsapimgr knob - nsapimgr_wr.sh -ys call=ns_formsso_use_ctype_simple_en knob

  [NSHELP-31977]

- SAML 認証が設定されている場合、ログアウト時に問題が発生する可能性があります。

  [NSHELP-31962]

- NetScaler アプライアンスは、重複したパスワードログインの試行を認証せず、アカウントのロックアウトを防ぎます。

  [NSHELP-563]

- ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
  show adfsproxyprofile <profile name>

  回避策: クラスター内のアクティブなプライマリの NetScaler に接続し、show adfsproxyprofile <profile name>コマンドを実行します。プロキシプロファイルの状態が表示されます。

  [NSAUTH-5916]

- 次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

  - [LDAP 到達可能性をテスト] オプションが開きます。
  - 無効なログイン認証情報が入力され、送信されます。
  - 有効なログイン認証情報が入力され、送信されます。

回避策:「LDAP 到達可能性テスト」オプションを閉じて開きます。

[NSAUTH-2147]

キャッシュ

- キャッシュされたコンテンツがクライアントに提供されると、NetScaler アプライアンスがクラッシュします。

  [NSHELP-31760]

- 統合キャッシュ機能が有効で、アプライアンスのメモリが不足している場合、NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-22942]

NetScaler SDX アプライアンス

- 次の条件が満たされている場合、NetScaler SDX アプライアンスでホストされている VPX インスタンスでパケットドロップが表示されます。
  - スループット割り当てモードはバーストです。
  - スループットと最大バーストキャパシティには大きな違いがあります。

  [NSHELP-21992]

NetScaler Gateway

- Citrix Secure Access クライアントのバージョン 21.7.1.2 以降では、管理者権限のないユーザーが新しいバージョンにアップグレードできません。この問題は、Citrix Secure Access クライアントのアップグレードが NetScaler アプライアンスから行われる場合にのみ発生します。

  [NSHELP-32793]

- ユーザーが Windows 向け Citrix Secure Access 画面の［ホームページ］タブをクリックすると、ページに接続拒否エラーが表示されます。

  [NSHELP-32510]

- Chrome を使用する Mac デバイスで、2 つの FQDN にアクセス中に VPN 拡張機能がクラッシュします。

  [NSHELP-32144]

- EPA の障害が断続的に発生するため、ユーザーは VPN にログオンできません。

  [NSHELP-32138]

- デバイスに適切なクライアント証明書がない場合、オプションのクライアント証明書による nFactor 認証は失敗します。

  [NSHELP-32127]

- HDX Insight が有効になっていると、NetScaler Gateway アプライアンスがクラッシュする可能性があります。

  [NSHELP-32120]

- クラスターセットアップでは、CGP_FINISH_REQUEST 要求をクライアントに送信中に NetScaler アプライアンスがクラッシュします。

  [NSHELP-32029]

- UDP セッションを起動すると、セッションを閉じた後でも古い接続が存在しているように見えます。ただし、これらは実際には古い接続ではなく、カウンタの問題です。

  [NSHELP-32009]

- NetScaler Gateway リリース 13.0 または 13.1 のプロキシ設定が空の場合、Citrix SSO によって不適切なプロキシ設定が作成されることがあります。

  [NSHELP-31970]

- Citrix Secure Access クライアントのデバッグログ制御は、NetScaler Gateway に依存せず、マシントンネルとユーザートンネルの両方のプラグイン UI から有効または無効にできるようになりました。

  [NSHELP-31968]

- Microsoft Edge がデフォルトのブラウザの場合、Citrix Secure Access UI のホームページリンクは機能しません。

  [NSHELP-31894]

- ユーザーが NetScaler アプライアンスにログオンし、Citrix Workspace がインストールされていない場合、Citrix Workspace をダウンロードするためのリンクが誤って Citrix Receiver を指します。

  [NSHELP-31877]

- Gateway Insight の認証失敗レコードには、NOAUTH が第 1 要素として設定されていて、認証情報が無効であるために 2 番目の要素認証が失敗した場合、ユーザー名は「匿名」と表示されます。この問題は、nFactor ビジュアライザーでは設計上、1 番目のファクターが NOAUTH として設定されているため、nFactor ビジュアライザーを使用して構成を実行した場合にのみ発生します。

  [NSHELP-31795]

- Citrix Secure Access によって確立されたトンネルの外部にあるリソースへの直接接続は、大幅な遅延または輻輳が発生すると失敗することがあります。

  [NSHELP-31598]

- カスタマイズされた EPA 障害ログメッセージは、NetScaler Gateway ポータルには表示されません。代わりに、「内部エラー」というメッセージが表示されます。

  [NSHELP-31434]

- ユーザーが Always-On サービスモードで Windows マシンにログインすると、Windows 自動ログオンが機能しないことがあります。マシントンネルはユーザートンネルに遷移しません。「Connecting…」というメッセージが VPN プラグイン UI に表示されます。

  [NSHELP-31357, CGOP-21192]

- ポリシーベースルーティング（PBR）ポリシーは、VPN 経由の DNS トラフィックには有効になりません。

  [NSHELP-31123]

- Always on が設定されている場合、aoservice.exe ファイルのバージョン番号（1.1.1.1）が正しくないため、ユーザトンネルが失敗します。

  [NSHELP-30662]

- ［NetworkAccessonVPNFailure］プロファイルパラメーターを［フルアクセス］から［OnlyToGateway］に変更すると、ユーザーは NetScaler Gateway アプライアンスに接続できなくなります。

  [NSHELP-30236]

- ゲートウェイのホームページは、ゲートウェイプラグインが VPN トンネルを正常に確立した直後には表示されません。この問題を解決するために、次のレジストリ値が導入されます。

  HKLMSoftwareCitrixSecure Access ClientSecureChannelResetTimeoutSeconds
  タイプ: DWORD

  デフォルトでは、このレジストリ値は設定も追加もされません。「SecureChannelResetTimeoutSeconds」の値が 0 または追加されていない場合、遅延を処理するための修正が機能しません。これはデフォルトの動作です。管理者は、このレジストリをクライアントに設定して修正を有効にする必要があります (つまり、ゲートウェイプラグインが VPN トンネルを正常に確立した直後にホームページを表示する)。

  [NSHELP-30189]

- Windows VPN クライアントは、サーバからの「SSL close notify」アラートを尊重せず、同じ接続で転送ログイン要求を送信します。

  [NSHELP-29675]

- IIP プール（IP アドレスとマスク）の構成中に、IP アドレスが範囲内の最初の IP アドレスと一致しない場合、NetScaler CLI および GUI には 1 つのブロックのみが表示され、すべては表示されません。

  例:
  バインド VPN vserver vpn_ssl-IntraneTip 172.168.1.1 255.255.255.0
  バインド VPN vserver vpn_ssl-IntranetIP 172.168.2.1 255.255.255.0

  この場合、VPN vserver vpn_ssl を表示しているときの CLI または GUI には 172.168.2.1 プールのみが表示され、172.168.2.2 は表示されません。

回避策：範囲内の最初の IP アドレスを使用して IIP ブロックを設定します。

例：

```
bind vpn vserver vpn_ssl -intranetIP 172.168.1.0 255.255.255.0
bind vpn vserver vpn_ssl -intranetIP 172.168.2.0 255.255.255.0
```

[NSHELP-29084]

- サーバー証明書が信頼されていると、サーバー検証コードが失敗することがあります。その結果、エンドユーザーはゲートウェイにアクセスできなくなります。

  [NSHELP-28942]

- rdx.js ファイルにいくつかの Citrix 内部 IP アドレスがあることに気付くかもしれません。

  [NSHELP-28682]

- macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

  [NSHELP-28551]

- クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

  [NSHELP-28404]

- 次の条件を満たす場合、Windows ログオン後に VPN プラグインはトンネルを確立しません。

  - NetScaler Gateway アプライアンスが常時オン機能用に構成されている
  - アプライアンスは、2 要素認証を「オフ」にした証明書ベースの認証用に設定されています。

  [NSHELP-23584]

- スキーマをブラウズしているときに、「未定義のプロパティ「タイプ」を読み取れません」というエラーメッセージが表示されることがあります。

  [NSHELP-21897]

- 「show vpn icaconnection」コマンドでは、ICA 接続のシリアル番号が正しく表示されません。この問題は、「show vpn icaconnection」コマンドの実行時にシリアル番号が任意にリセットされるために発生します。

  [CGOP-22205]

- Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

  [CGOP-19355]

- 無効な STA チケットによるアプリケーションの起動失敗は、Gateway Insight では報告されません。

  [CGOP-13621]

- Gateway Insight レポートでは、SAML エラーエラーの認証タイプフィールドに「SAML」ではなく「Local」という値が誤って表示されます。

  [CGOP-13584]

- 高可用性セットアップでは、NetScaler フェイルオーバー時に、NetScaler Console のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

  [CGOP-13511]

- ICA 接続が MAC Receiver バージョン 19.6.0.32 または Citrix Virtual Apps and Desktops バージョン 7.18 から起動されると、HDX Insight 機能は無効になります。

  [CGOP-13494]

- EDT Insight 機能が有効になっていると、ネットワークの不一致時にオーディオチャネルに障害が発生することがあります。

  [CGOP-13493]

- ブラウザからローカルホスト接続を受け付けると、macOS の Accept Connection ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

  [CGOP-13050]

- 一部の言語では、Citrix SSO アプリ > ホームページの「ホームページ」というテキストが切り捨てられます。

  [CGOP-13049]

- NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

  [CGOP-11830]

- Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

  [CGOP-7269]

NetScaler Web App Firewall

- NetScaler Web App Firewall は、コマンドインジェクションの検出に時間がかかることがあります。その結果、Pitboss は NetScaler アプライアンスを再起動します。

  [NSHELP-32654]

- ボットデバイスの指紋認証セッションリプレイ攻撃は、ドロップされるのではなくログに記録されます。

  [NSHELP-31949]

負荷分散

- 高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

  [NSLB-7679]

- GSLB 仮想サーバーで次の設定が構成されている場合、NetScaler アプライアンスは GSLB ドメインクエリの正しいサービス IP アドレスで応答しません。

  1. ECS オプションは有効です。
  2. 静的近接は負荷分散方法として設定されます。

  [NSHELP-32879]

- ユーザー監視スクリプトが 1024 バイトを超える応答を返すと、NetScaler アプライアンスがクラッシュしてコアがダンプされる可能性があります。

  [NSHELP-32097]

- 設定した属性がサーバに存在しない場合でも、LDAP モニタのステータスはアップのままです。

  [NSHELP-32025]

- まれに競合状態が発生するため、ローカルサイトとリモートサイトの間に不整合がある可能性があります。この不整合は、リモートサイトがローカルサイトから動的メンバーを学習していないことが原因である可能性があります。

  リモートサイトの動的メンバーの削除は、パケットエンジン間の通信中に問題が発生したため、正常に削除されない場合があります。

  [NSHELP-31982]

- vserverAdvancesSLConfigTable OID に対応する SNMP WALK 要求は、仮想サーバーの優先順位が構成されているときにコアダンプになります。

  [NSHELP-31704]

- サービスグループの`entityofs`トラップの ServiceGroupName 形式は次のとおりです。
  `<service(group)name>?<ip/DBS>?<port>`

  トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (“?”) は区切り文字として使用されます。NetScaler は、疑問符（” ?”)。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

  [NSHELP-28080]

- 特定のシナリオでは、サービスグループにバインドされたサーバーが、無効な Cookie 値を表示します。トレースログで正しい Cookie 値を確認できます。

  [NSHELP-21196]

その他

- 高可用性セットアップで強制同期が行われると、アプライアンスはセカンダリノードで「set urlfilter parameter」コマンドを実行します。
  その結果、セカンダリノードは、「TimeOfDayToUpdateDB」パラメーターで指定された次のスケジュール時刻まで、スケジュールされた更新をスキップします。

  [NSSWG-849]

- レジストリ値が 2000 バイトを超えると、AlwaysOnAllow リストレジストリが期待どおりに動作しません。

  [NSHELP-31836]

- 次の条件が満たされると、クラスターノードはパケットループに入ります。

  - 宛先 IP アドレスが CLIP の UDP パケットがクラスタノードに送信されます。
  - CCO は、クラスターインスタンスの存続期間中に 1 つのノードから別のノードに変更されました。

  回避策：宛先 IP アドレスを CLIP アドレスとする特定の UDP パケットにドロップ ACL を適用することで、このパケットループを回避または終了できます。

  [NSHELP-30804]

- URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

  [NSHELP-22409]

ネットワーク

- DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

  [NSNET-25299]

- DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

  - NetScaler BLX アプライアンスには、少数の「巨大ページ」が割り当てられています。たとえば、1G です。
  - NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

  この問題は、エラーメッセージとして「/var/log/ns.log」に記録されます。

  - 「BLX-DPDK: DPDK メモリプールを PE-x 用に初期化できませんでした」

注:x はワーカープロセス数以下の数です。

回避策:「巨大ページ」を多数割り当ててから、アプライアンスを再起動します。

[NSNET-25173]

- DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

  [NSNET-24449]

- DPDK を搭載した NetScaler BLX アプライアンスの Intel `X710 10G (i40e)` インターフェイスでは、次のインターフェイス操作はサポートされていません:

  - 無効化
  - 有効化
  - リセット

  [NSNET-16559]

- Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

  「次のパッケージの依存関係は満たされていません:blx-core-libs: i386: preDepends: libc6: i386 (>= 2.19) しかしインストールできません」

  回避策: NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します:

  - dpkg –add-architecture i386
  - apt-get update
  - apt-get dist-upgrade
  - apt-get install libc6:i386

  [NSNET-14602]

- FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

  [NSNET-5233]

- NetScaler アプライアンスで ECMP が構成されている場合、SSH 負荷分散接続で次の問題が発生する可能性があります。

  - NetScaler アプライアンスは、最初のパケットを同じフローの他のパケットとは異なるルートで送信します。

  [NSHELP-32089]

- 次の条件が満たされると、NetScaler アプライアンスが一部のシナリオでクラッシュする可能性があります。
  - NetScaler アプライアンスは、オフセットが異なる複数の最初のフラグメントを受信します。
  - NetScaler アプライアンスはフラグメントを再構成しません。

  [NSHELP-32084]
- 仮想サーバーで「セッションレス」オプションを有効にし、サーバー側で ECMP を有効にした負荷分散構成では、次の問題が発生する可能性があります。
  - NetScaler アプライアンスは、常に同じルートでパケットをサーバーに送信します。

  [NSHELP-32061]
- 次の条件をすべて満たすと、NetScaler アプライアンスがクラッシュする可能性があります。
  - TTL ベースの ACL がタイムアウト
  - NetScaler アプライアンスには多数の ACL が設定されています。

  [NSHELP-31307]
- 仮想サーバーを削除すると、次の条件が満たされている場合、NetScaler アプライアンスは関連する VIP RHI 状態を誤って DOWN に設定します。
  - 仮想サーバーにはバックアップ仮想サーバーがあります。
  - 仮想サーバは DOWN 状態で、少なくとも 1 つのバックアップ仮想サーバが UP 状態です。

  [NSHELP-29972]
- NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

  [NSHELP-21082]

プラットフォーム

- 高可用性フェイルオーバーは AWS および GCP クラウドでは機能しません。AWS および GCP クラウド、および NetScaler VPX オンプレミスでは、管理 CPU が 100% の容量に達する可能性があります。これらの問題はいずれも、次の条件が満たされた場合に発生します。
  1. NetScaler アプライアンスの初回起動時には、プロンプトが表示されたパスワードは保存されません。
  2. その後、NetScaler アプライアンスを再起動します。

  [NSPLAT-22013]
- 13.0/12.1/11.1 ビルドから 13.1 ビルドにアップグレードするか、13.1 ビルドから 13.0/12.1/11.1 ビルドにダウングレードすると、一部の Python パッケージが NetScaler アプライアンスにインストールされません。この問題は、次の NetScaler バージョンで修正されています。

- 13.1-4.x
- 13.0-82.31 以降
- 12.1-62.21 以降

NetScaler バージョンを 13.1-4.x から次のバージョンのいずれかにダウングレードすると、Python パッケージはインストールされません。

- 任意の 11.1 ビルド
- 12.1-62.21 およびそれ以前
- 13.0-81.x およびそれ以前

[NSPLAT-21691]

- Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。「rm cloudprofile」コマンドを使用してプロファイルを削除します。

  [NSPLAT-4520]

- Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
  回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノード上で設定する必要があります。

  [NSPLAT-4451]

- NetScaler SDX 8015/8400/8600 プラットフォームでは、Xen サーバーのメモリ消費量が増加する可能性があります。
  回避策：Xen Server で次のコマンドを実行し、アプライアンスを再起動します。
  /opt/xensource/libexec/xen-cmdline –set-xen “dom0_mem=1024M,max:1024M”

  [NSHELP-32260]

- NetScaler リリース 13.1 以降、NetScaler アプライアンスは、8 つを超える VMXNET3 ネットワークインターフェイスを備えた ESXi ハイパーバイザーで起動に失敗します。

  [NSHELP-31266]

ポリシー

- 処理データのサイズが設定されたデフォルトの TCP バッファーサイズを超えると、接続がハングアップすることがあります。

  回避策:TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

  [NSPOLICY-1267]

- NetScaler アプライアンスでは、NSPEPI ツールを使用して従来のポリシーから高度なポリシーに移行されたコンテンツスイッチポリシーは、次の条件が満たされると機能しない場合があります。
  - ポリシーはコンテンツスイッチ仮想サーバーにバインドされます。
  - 「CaseSensitive」パラメータはオフに設定されています。

  [NSHELP-31951]
- 次の条件が満たされると、NetScaler アプライアンスが patset によるポリシー追加中にクラッシュすることがあります。
  - NSB に関連するフラグが TCP の書き換えシナリオで間違った順序で設定されています。

  [NSHELP-31064]

SSL

- NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

  回避方法:

  1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
  2. 構成を保存します。

  [NSSSL-9572]
- 認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

  [NSSSL-6478]
- 同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

  [NSSSL-6213]
- HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
  エラー:CRL 更新が無効です

  [NSSSL-6106]
- セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

  [NSSSL-4427]

- SSL プロファイル内の SSL プロトコルまたは暗号を変更しようとすると、「警告:SSL vserver/Service で使用可能な暗号が設定されていません」という誤った警告メッセージが表示されます。

  [NSSSL-4001]

- 期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

  [NSSSL-3184, NSSSL-1379, NSSSL-1394]

- Cavium SSL カードを搭載した NetScaler アプライアンスが、クライアントに DTLS ALERT メッセージを送信しているときにクラッシュすることがあります。

  [NSHELP-32031]

- 次の条件が満たされると、SSL ハンドシェイクが失敗することがあります。

  1. Hello 検証要求 (HVR) は DTLS で有効になっています。
  2. NetScaler アプライアンスは HVR をクライアントに送信します。
  3. クライアントは HVR を受信しません。
  4. クライアントは、HVR にセッション Cookie で応答する代わりに、最初のクライアント hello を再送信しようとします。

  注: 再送信されたクライアントの hello メッセージに応答して、ADC アプライアンスは HVR をクライアントに最大 3 回送信します。適切な応答が受信されない場合、アプライアンスはハンドシェイクに失敗します。

  [NSHELP-31808]

- 証明書認証ルールが評価され、同じ要求で 2 回トリガーされると、NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-31785]

- クラスタ IP（CLIP）アドレスを介してアクセスされる NetScaler GUI には、SSL 仮想サーバーへのサーバー証明書バインディングは表示されません。

  [NSHELP-31602]

- 有効な CA 証明書がデフォルトの証明書バンドルに存在しない場合、SSL インターセプト中に OCSP 応答検証が失敗することがあります。この失敗は、設定された証明書バンドルの代わりにデフォルトの証明書バンドルを使用して OCSP 応答検証が誤って行われたことが原因です。

  [NSHELP-30594]

- ソフトウェアモードで SSL トラフィックを処理すると、NetScaler アプライアンスがクラッシュする可能性があります。

  [NSHELP-29996]

システム

- NetScaler アプライアンスでは、ヘッダー変更フレームワークによりメモリが破損します。この状態は、NetScaler アプライアンスが消費するクッキーが転送される前に特定の順序で削除された場合に発生します。

  [NSHELP-32799]

- NetScaler アプライアンスでは、HTTP プロファイルの「MaxHeaderFieldLen」パラメーターのデフォルト値により、次の問題が発生します。

  - 13.0 ビルドにアップグレードした後のトラフィック障害。

  [NSHELP-32079]

- AppFlow がクライアント側でのみ有効になっていると、NetScaler アプライアンスがクラッシュすることがあります。

  [NSHELP-31892]

- SSL サービスで構成された NetScaler アプライアンスは、アプライアンスが TCP FIN 制御パケットの後に TCP RESET 制御パケットを受信するとクラッシュします。

  [NSHELP-31656]

- 以下の条件が満たされると、gRPC クライアントは gRPC ステータスヘッダーの解析に失敗します。

  - gRPC ステータスヘッダーは、末尾ヘッダーだけに追加されるのではなく、先頭ヘッダーと末尾ヘッダーの両方に追加されます。

  [NSHELP-31640]

- 次の条件が満たされると、TCP 接続の RTT が高くなります。

  - 最大輻輳ウィンドウ（> 4 MB）が高く設定されている
  - TCP NILE アルゴリズムは有効になっています

  NetScaler アプライアンスが NILE アルゴリズムを使用して輻輳制御を行うには、条件がスロースタートのしきい値と最大輻輳ウィンドウを合わせた値を超える必要があります

  そのため、設定された最大輻輳時間帯に達するまで、NetScaler はデータを受け付け続け、最終的には RTT が高くなります。

  [NSHELP-31548]

- NetScaler アプライアンスでは、コンテンツスイッチングまたは負荷分散仮想 IP（VIP）の HTTP/2 構成を有効にすると、次の問題が発生します。

- コンテンツ検査機能を使用する場合、ペイロードによる Rewrite ヘッダーの挿入が正しく機能しない場合があります。

  [NSHELP-30088]

- アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

  [NSHELP-21240]

- mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

  [NSHELP-10972]

- クラスタ展開では、CCO 以外のノードで「force cluster sync」コマンドを実行すると、ns.log ファイルには重複するログエントリが含まれます。

  [NSBASE-16304、NSGI-1293]

- NetScaler Console を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに機能しません。

  回避策：管理ポッドを再起動します。

  [NSBASE-15556]

- LogStream トランスポートタイプが Insight 用に構成されている場合、クライアント IP とサーバー IP は HDX Insight SkipFlow レコードで反転されます。

  [NSBASE-8506]

ユーザーインターフェイス

- MQTT リライト機能では、GUI のエクスプレッションエディタを使用してエクスプレッションを削除することはできません。

  回避策:CLI を使用して MQTT タイプのアクションの追加または編集コマンドを使用します。

  [NSUI-18049]

- NetScaler GUI では、「ダッシュボード」タブの下にある「ヘルプ」リンクが壊れています。

  [NSUI-14752]

- CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

  回避策：NetScaler GUI または CLI を使用して IPSec プロファイル、IP トンネル、および PBR ルールを追加して CloudBridge Connector を構成します。

  [NSUI-13024]

- GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

  [NSUI-6838]

- NetScaler Web App Firewall のプロファイルを作成し、[システム**] > [**レポート**]** でアプリケーションファイアウォールの構成レポートを生成しようとすると、次のエラーが表示されます。

  「PDF ドキュメントを読み込めませんでした。」

  [NSHELP-32469]

- 高可用性 (HA) セットアップで、nsconf ツールのローカル IP アドレスをフェッチしているときに、次の問題が発生します。

  - ローカルホスト接続ログインに失敗しました。この障害は、RPC ノードのパスワードが HA セットアップのプライマリノードとセカンダリノードで異なる場合に発生します。

  回避策:HA セットアップでは、プライマリノードとセカンダリノードの両方の RPC ノードパスワードが同じであることを確認します。

  [NSHELP-32083]

- NetScaler リリース 13.0 では、［優先負荷分散仮想サーバーサービスの構成］ページの ［**OK** ］ボタンがグレー表示になっています。

  [NSHELP-32007]

- ユーザーがログインした後、NetScaler アプライアンスのログインページに有効なユーザー名が表示されない場合があります。

  [NSHELP-31759]

- HA/Cluster セットアップでは、RSA 以外の SSH キーを設定すると、設定の同期が失敗します。たとえば、ECDSA キーや DSA キーなどです。

  [NSHELP-31675]

- NetScaler GUI で、**[**システム**] > [SNMP] > [**トラップ**]** の下に既存の **SNMP** トラップ宛先がある場合、その宛先の編集に失敗し、次のエラーメッセージが表示されます。

  -「SNMP トラップを取得中にエラーが発生しました」

  [NSHELP-3161]

- NetScaler アプライアンスの GUI には、構成済みの SAML および OAuth IDP ポリシーの正しい数が表示されません。

  [NSHELP-31480]

- NetScaler アプライアンスでは、GUI インターフェイスを使用しているときに、レスポンダーポリシーページに次の問題が表示されます。

  - カスタム作成されたレスポンダーポリシーは、組み込みレスポンダーポリシーの下に表示される場合があります。

  [NSHELP-31428]

- NetScaler HA セットアップでは、構成を保存して更新ボタンをクリックすると、NetScaler GUI に次の問題が発生します。

  - アプライアンスに未保存の設定変更がない場合でも、GUI の [保存] ボタンにオレンジ色のドットが誤って表示されます。

  [NSHELP-30031]

- GSLB 仮想サーバーの統計情報は、管理パーティションモードでは使用できません。

  [NSHELP-28524]

- 高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

  - HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

  回避策：HA 同期が完了した後にのみ、手動で HA フェイルオーバーを連続して実行してください（両方のノードが同期成功状態になっています）。

  [NSHELP-25598]

- NetScaler BLX アプライアンスの高可用性セットアップでは、プライマリノードが応答しなくなり、CLI または API 要求がブロックされることがあります。

  回避策: プライマリノードを再起動します。

  [NSCONFIG-6601]

- あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

  1. NetScaler アプライアンスをいずれかのビルドにアップグレードします
     - 13.0 52.24 ビルド
     - 12.1 57.18 ビルド
     - 11.1 65.10 ビルド
  2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
  3. NetScaler アプライアンスを古いビルドにダウングレードします。

  CLI を使用してこれらのシステムユーザーのリストを表示するには、
  コマンドプロンプトで次のように入力します。

  ```
  query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
  ```

  回避策: この問題を解決するには、以下のいずれかの独立したオプションを使用してください。

  - NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3）、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。

- アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
- 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳細については、https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.htmlを参照してください。

[NSCONFIG-3188]

NetScaler 13.1—30.52 リリースのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリース Build 13.1—30.52 に存在する拡張機能と変更、修正された既知の問題について説明します。

メモ

このリリースノートドキュメントには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正とアドバイザリのリストについては、セキュリティ速報を参照してください。

新機能

ビルド 13.1～30.52 で利用できる機能強化と変更。

ネットワーク

ASDOT 形式の 4 バイト BGP ASN のサポート　NetScaler アプライアンスは、RFC 5396 で定義されている asdot 形式の 4 バイトの BGP 自律システム番号（ASN）の構成と表示をサポートするようになりました。NetScaler アプライアンスは、BGP ASN に対して次の 2 つの形式を全体的にサポートしています。

- asplain-2 バイトと 4 バイトの両方の ASN が 10 進数値で表される 10 進値表記。たとえば、65527 は 2 バイト ASN で、234567 は 4 バイト ASN です。
- asdot-自律システムのドット表記。2 バイトの ASN は 10 進数値で表され (asplain と同じ)、4 バイトの ASN はドット表記で表されます。たとえば、65527 は 2 バイト ASN で、3.37959 は 4 バイト ASN です（3.37959 は 234567 の 10 進数のドット形式です）。

[NSNET-26101]

NetScaler BLX アプライアンスに対する **AWS** クラウド上の **Amazon Linux 2** のサポート　NetScaler BLX アプライアンスは、AWS クラウド上の Amazon Linux 2 でサポートされるようになりました。NetScaler BLX は、Amazon Linux 2 の DPDK ポートとして AWS エラスティックネットワークアダプター (ENA) での実行をサポートします。

[NSNET-25802]

利用可能なルートにモニタープローブを均等に分散　13.1～30.x から、NetScaler アプライアンスは、次の 5 つのタプルに基づくハッシュアルゴリズムを使用して、負荷分散モニタープローブのルートを選択します。

- 送信元 IP アドレス
- 送信元ポート
- 宛先 IP アドレス
- 送信先ポート
- プロトコル番号

5 つのタプル情報に基づいてルートを選択することで、利用可能なルートにモニタープローブが均等に分散されます。この均等な分散により、ルート内のトラフィックの過負荷を防ぐことができます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/networking/ip-routing/route-selection-based-on-five-tuples.htmlを参照してください。

[NSNET-24646]

SSL

OCSP マルチホチキス止めソリューションのサポート　TLS 1.3 プロトコルを使用すると、すべての中間証明書に、クライアントからのステータス要求への応答に OCSP 応答拡張が含まれるようになりました。以前は、サーバー証明書のみが、クライアントからのステータス要求への応答にこの拡張を含んでいました。

[NSSSL-9281]

ユーザーインターフェイス

より短い時間でライセンスを取得できるように **`show ns licenseserverpool`** コマンドを最適化しました　`show ns licenseserverpool`コマンドを実行すると、ライセンスの取得にかかる時間が短くなります。ライセンスモードを指定する新しいパラメーター`licensemode`が`add ns licenseserver`コマンドに追加されます。そのため、`show ns licenseserverpool`コマンドは指定されたライセンスモードに基づくライセンスのみを表示します。すべてのライセンスのインベントリが必要な場合は、`show ns licenseserverpool -get alllicenses`コマンドを使用してください。

以前は、`show ns licenseserverpool`コマンドは、設定されているライセンスモードに関係なくすべてのライセンスを表示するために使用されていました。その結果、コマンドがすべてのライセンスを取得するのに時間がかかっていました。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/licensing.html#citrix-adc-self-managed-pool-licenseを参照してください:

[NSCONFIG-6961]

セルフマネージドプールライセンスのサポート　NetScaler アプライアンスがセルフマネージドプールライセンスをサポートするようになりました。これにより、購入後のライセンスサーバーへのライセンスファイルのアップロードが簡単かつ自動化されます。NetScaler Console を使用して、共通の帯域幅または vCPU とインスタンスプールで構成されるライセンスフレームワークを作成できます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/licensing.html#citrix-adc-self-managed-pool-licenseを参照してください:

[NSCONFIG-6592]

NetScaler CPX ライセンスアグリゲーターのサポート　これで、NetScaler が提供する新しい Kubernetes マイクロサービスである NetScaler CPX ライセンスアグリゲーターを使用して、NetScaler CPX のライセンスを取得できるようになりました。NetScaler CPX を起動するときは、NetScaler CPX ライセンスアグリゲーターの IP アドレスまたはドメイン名を使用して環境変数 CLA を構成する必要があります。環境変数が構成されている場合、NetScaler CPX ライセンスアグリゲーターは、接続されているすべての NetScaler CPX の集約ライセンスをチェックアウトします。

[NSCONFIG-6394]

解決された問題

ビルド 13.1—30.52 で対処された問題。

認証、承認、監査

構成内の SAML メタデータ URL がバックスラッシュ（/）で終わらない、またはバックスラッシュ（/）を含んでいない場合、NetScaler アプライアンスがクラッシュする可能性があります。

[NSHELP-31937]

Syslog サーバを設定した場合、SAML 関連のログが 2 行に 1 つ表示されます。

[NSHELP-31750]

認証仮想サーバーでコンテンツセキュリティポリシー (CSP) の書き換えポリシーを適用する際に、アプリケーションの書き換えに問題が発生する可能性があります。

[NSHELP-31583]

LDAP アクションが IP アドレスではなく FQDN に設定されている場合、非 ASCII 文字が nsvpn.log に記録されます。

[NSHELP-27281]

NetScaler GUI には、VPN 仮想サーバーにバインドされたデフォルトのキャッシュポリシーが表示されません。

[NSHELP-26874]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、システムグループの作成または編集が失敗します。

[NSHELP-32359]

NetScaler SDX アプライアンスは、ハイパーバイザーディスクの使用状況に関する SNMP トラップを NetScaler コンソールに送信しません。

[NSHELP-32323]

NetScaler SDX アプライアンスでは、VLAN ホワイトリストが NetScaler VPX インスタンスに割り当てられた Mellanox インターフェイスの正しい値で更新されません。

[NSHELP-31849]

NetScaler SDX アプライアンスをアップグレードすると、ハイパーバイザーのバージョンが現在の SDX バージョンとアップグレード後の SDX バージョンの両方で同じであっても、管理サービス GUI に次の誤ったイベントが通知されます。

SVM と Hypervisor のバージョンが一致しません

[NSHELP-31769]

証明書名またはキー名にスペースが含まれていると、NetScaler SDX アプライアンスへの SSL 証明書のインストールが失敗します。

[NSHELP-31711]

NetScaler SDX アプライアンスを 13.0 から 13.1 ファームウェアにアップグレードすると、プラットフォームのアップグレード中にインストール後のスクリプトファイル（postinst.sh）の Citrix Hypervisor へのアップロードが失敗することがあります。

[NSHELP-31125]

NetScaler Gateway

クラスターセットアップでは、CGP_FINISH_REQUEST 要求をクライアントに送信中に NetScaler アプライアンスがクラッシュします。

[NSHELP-32029]

イントラネット IP アドレスをクライアントに割り当てるときに、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-31712]

ポリシーベースルーティング（PBR）ポリシーは、VPN 経由の DNS トラフィックには有効になりません。

[NSHELP-31123]

従来の EPA ポリシーと nFactor 認証が構成されている場合、認証に成功した場合の Gateway Insight イベントは NetScalerApplication Delivery Management に送信されません。

[NSHELP-30901]

ns_aaa_json.c ファイルに NS_AUDITLOG_STR* ログ用の余分な行が表示される場合があります。

[NSHELP-28160]

GUI を使用してクラシック認可ポリシーをバインド解除することはできません。ただし、CLI を使用して認証、承認、および監査認可ポリシーのバインドを解除することはできます。

今回の修正により、GUI を使用して承認ポリシーのバインドを解除できるようになりました。

[NSHELP-27064]

Gateway Insight は、VPN ユーザに関する正確な情報を表示しません。

[NSHELP-23937]

ログフラグの脆弱性は、クライアントの送信元 IP アドレスをキャプチャしません。これらのログは次のとおりです。

- 無効なヘッダー/バージョンの HTTP リクエストをドロップしています
- パストラバーサルが検出されました
- 不要な場所に‘/vpns/’が見つかりました
- 無効な HTTP リクエストをドロップする

[CGOP-18190]

NetScaler Web App Firewall

NetScaler アプライアンスでは、コンソールにログメッセージが殺到し、アプライアンスが DNS クエリを Webroot パブリッククラウドサービスプロバイダーに送信する可能性があります。これは、IP レピュテーション機能を無効にすると、24 時間に 1 回ではなく 5 分おきに実行されるためです。

[NSWAF-9299]

負荷分散

ユーザー監視スクリプトが 1024 バイトを超える応答を返すと、NetScaler アプライアンスがクラッシュしてコアがダンプされる可能性があります。

[NSHELP-32097]

まれに、DNSSEC 処理が有効になっていて、DNS ゾーン構成が存在する場合、NetScaler アプライアンスがクラッシュしてコアをダンプすることがあります。

[NSHELP-31993]

まれに競合状態が発生するため、ローカルサイトとリモートサイトの間に不整合がある可能性があります。この不整合は、リモートサイトがローカルサイトから動的メンバーを学習していないことが原因である可能性があります。

リモートサイトの動的メンバーの削除は、パケットエンジン間の通信中に問題が発生したため、正常に削除されない場合があります。

[NSHELP-31982]

vserverAdvancesSLConfigTable OID に対応する SNMP WALK 要求は、仮想サーバーの優先順位が構成されているときにコアダンプになります。

[NSHELP-31704]

ネットワーク

DPDK を搭載した NetScaler BLX アプライアンスは、次の条件が満たされると再起動に失敗することがあります。

- NetScaler BLX アプライアンスには`hugepages`の大きい数が割り当てられます。たとえば、16 GB と入力します。

この問題は、`/var/log/ns.log`にエラーメッセージとして記録されます。

- `EAL: rte_mem_virt2phy(): cannot open /proc/self/pagemap: Too many open files`

[NSNET-24727]

NetScaler アプライアンスで ECMP が構成されている場合、SSH 負荷分散接続で次の問題が発生する可能性があります。

- NetScaler アプライアンスは、最初のパケットを同じフローの他のパケットとは異なるルートで送信します。

[NSHELP-32089]

次の条件が満たされると、NetScaler アプライアンスが一部のシナリオでクラッシュする可能性があります。

- NetScaler アプライアンスは、オフセットが異なる複数の最初のフラグメントを受信します。
- NetScaler アプライアンスはフラグメントを再構成しません。

[NSHELP-32084]

仮想サーバーで`sessionless`オプションが有効で、サーバー側で ECMP が有効になっている負荷分散構成では、次の問題が発生する可能性があります。

- NetScaler アプライアンスは、常に同じルートでパケットをサーバーに送信します。

[NSHELP-32061]

大規模な NAT44 セットアップでは、次の理由により、SIP トラフィックの受信中に NetScaler アプライアンスがクラッシュすることがあります。

- 古いフィルタリングエントリが原因です。

[NSHELP-28895]

プラットフォーム

NetScaler SDX アプライアンスでは、Mellanox インターフェースのリングサイズが 1024 エントリから 2048 エントリに増加します。

[NSPLAT-24539]

/var/log/waagent フォルダーに保存されているファイルのログローテーションが失敗し、より多くのディスク領域を占有します。この障害は、NetScaler VPX インスタンスから取得したバックアップ構成を、復元機能を使用して Azure クラウドでホストされている別の NetScaler VPX インスタンスに適用したときに発生します。

[NSHELP-31599]

NetScaler リリース 13.1 以降、NetScaler アプライアンスは、8 つを超える VMXNET3 ネットワークインターフェイスを備えた ESXi ハイパーバイザーで起動に失敗します。

[NSHELP-31266]

ポリシー

NetScaler アプライアンスでは、次のことが観察されます。

- まれなケースでのメモリアカウントに関連する問題。
- 特定のエンティティのメモリ割り当て/割り当て解除に関連する問題。

また、特定のエンティティの割り当て/割り当て解除の追跡が追加/改善されました。

[NSHELP-29215]

SSL

RSA と ECDSA の両方の証明書とキーのペアが仮想サーバにバインドされ、ピアが互換性のある署名アルゴリズムをサポートしている場合、TLS 1.3 サーバは ECDSA 証明書とキーのペアを選択します。以前は、TLS 1.3 サーバーは RSA 証明書とキーのペアを選択していました。この変更により、TLS 1.3 サーバーは TLS 1.2 サーバーと同じように動作するようになりました。

[NSSSL-11650]

TLS 1.3 サーバーは、複数の TLS レコードに分割 (フラグメント) された TLS 1.3 ハンドシェイクメッセージを検出すると、`decode_error` アラートを返します。これは、クライアントが証明書を使用して認証を行っていて、クライアントの証明書が最大 TLS レコードサイズ (約 16 KB) よりも大きい場合に、ハンドシェイクの正常な完了に影響を与える可能性があります。

[NSSSL-2940]

次の条件が満たされると、SSL ハンドシェイクが失敗することがあります。

1. Hello 検証要求 (HVR) は DTLS で有効になっています。
2. NetScaler アプライアンスは HVR をクライアントに送信します。
3. クライアントは HVR を受信しません。
4. クライアントは、セッション Cookie を使用して HVR に応答する代わりに、最初のクライアント Hello を再送信しようとします。注：再送信されたクライアントの Hello メッセージに応答して、ADC アプライアンスは HVR をクライアントに最大 3 回送信します。適切な応答が受信されない場合、アプライアンスはハンドシェイクに失敗します。

[NSHELP-31808]

メモリ使用率が 80% を超えると、SSL トラフィックを処理するように構成された NetScaler アプライアンスがクラッシュする可能性があります。

[NSHELP-29996]

システム

NetScaler アプライアンスが syslog アクション構成フローでクラッシュする。このクラッシュは、セカンダリノードでの高可用性同期中に発生します。

[NSHELP-32254, NSHELP-32397]

NetScaler アプライアンスでは、HTTP プロファイルの`maxHeaderFieldLen`パラメータのデフォルト値によって次の問題が発生します。

- 13.0 ビルドにアップグレードした後のトラフィック障害。

[NSHELP-32079]

AppFlow がクライアント側でのみ有効になっていると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-31892]

次の条件が満たされると、NetScaler アプライアンスがクラッシュすることがあります。

- 分析プロファイルと AppFlow ポリシーの両方がバインドされ、プロファイルで`httpAllHdrs`オプションが有効になっています。

[NSHELP-30628]

NetScaler アプライアンスでは、コンテンツスイッチングまたは負荷分散仮想 IP（VIP）の HTTP/2 構成を有効にすると、次の問題が発生します。

- NetScaler アプライアンスを介して HTTP/2 ヘッダーとデータフレームを Web サイトに転送する際の遅延が最大 100 ミリ秒に増加します。

[NSHELP-30094、NSHELP-34672]

ユーザーインターフェイス

高可用性 (HA) セットアップで、nsconf ツールのローカル IP アドレスをフェッチしているときに、次の問題が発生します。

- ローカルホスト接続ログインに失敗しました。この障害は、RPC ノードのパスワードが HA セットアップのプライマリノードとセカンダリノードで異なる場合に発生します。

[NSHELP-32083]

SSL 仮想サーバーと証明書とキーのペアのバインディングを削除しようとすると、次の例外が Python API SDK で見られます。
TypeError: ‘str’ オブジェクトと ‘bool’ オブジェクトを連結できません

[NSHELP-31746]

負荷分散サーバーの統計情報が、NetScaler GUI ダッシュボードでずれている。

[NSHELP-20752]

既知の問題

リリース 13.1～30.52 に存在する問題。

AppFlow

HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

[NSINSIGHT-943]

認証、承認、監査

NetScaler アプライアンスは、重複したパスワードログインの試行を認証せず、アカウントのロックアウトを防ぎます。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema が NetScaler GUI のログインスキーマエディタ画面に正しく表示されません。

[NSAUTH-6106]

ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
`show adfsproxyprofile <profile name>`

回避方法:

クラスタ内のプライマリのアクティブな NetScaler に接続し、`show adfsproxyprofile <profile name>` コマンドを実行します。プロキシプロファイルの状態が表示されます。

[NSAUTH-5916]

次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

- [LDAP 到達可能性をテスト] オプションが開きます。
- 無効なログイン認証情報が入力され、送信されます。
- 有効なログイン認証情報が入力され、送信されます。

回避方法:

[LDAP 到達可能性のテスト] オプションを閉じて開きます。

[NSAUTH-2147]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、CLAG が Mellanox NIC 上に作成されている場合、VPX インスタンスの再起動時に CLAG MAC が変更されます。VPX インスタンスへのトラフィックは再起動後に停止します。これは、MAC テーブルに古い CLAG MAC エントリがあるためです。

[NSSVM-4333]

NetScaler Gateway

Chrome を使用している MAC デバイスで、2 つの FQDN にアクセスしているときに VPN 拡張機能がクラッシュする。

[NSHELP-32144]

Citrix Secure Access によって確立されたトンネルの外部にあるリソースへの直接接続は、大幅な遅延または輻輳が発生すると失敗することがあります。

[NSHELP-31598]

Always on が設定されている場合、aoservice.exe ファイルのバージョン番号（1.1.1.1）が正しくないため、ユーザトンネルが失敗します。

[NSHELP-30662]

[NetworkAccessonVPNFailure］プロファイルパラメーターを［フルアクセス］から［OnlyToGateway］に変更すると、ユーザーは NetScaler Gateway アプライアンスに接続できなくなります。

[NSHELP-30236]

ゲートウェイのホームページは、ゲートウェイプラグインが VPN トンネルを正常に確立した直後には表示されません。この問題を解決するために、次のレジストリ値が導入されます。

\ HKLM\ ソフトウェア\ Citrix\ Secure Access Client
セキュアチャネルリセットタイムアウト (秒) タイプ:DWORD

デフォルトでは、このレジストリ値は設定も追加もされません。SecureChannelResetTimeoutSeconds の値が 0 の場合、または追加されていない場合、遅延を処理する修正は機能しません。これはデフォルトの動作です。管理者は、このレジストリをクライアントに設定して修正を有効にする必要があります (つまり、ゲートウェイプラグインが VPN トンネルを正常に確立した直後にホームページを表示する)。

[NSHELP-30189]

Windows VPN クライアントは、サーバからの「SSL close notify」アラートを尊重せず、同じ接続で転送ログイン要求を送信します。

[NSHELP-29675]

サーバー証明書が信頼されていると、サーバー検証コードが失敗することがあります。その結果、エンドユーザーはゲートウェイにアクセスできなくなります。

[NSHELP-28942]

rdx.js ファイルにいくつかの Citrix 内部 IP アドレスがあることに気付くかもしれません。

[NSHELP-28682]

macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

[NSHELP-28551]

クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

[NSHELP-28404]

次の条件が満たされている場合、VPN プラグインは Windows ログオン後にトンネルを確立しません。

- NetScaler Gateway アプライアンスが常時オン機能用に構成されている
- アプライアンスは、二要素認証`off`の証明書ベースの認証用に構成されています

[NSHELP-23584]

スキーマを参照しているときに、エラーメッセージ`Cannot read property 'type'of undefined`が表示されることがあります。

[NSHELP-21897]

Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を使用できます。

[CGOP-19355]

無効な STA チケットによるアプリケーションの起動失敗は、Gateway Insight では報告されません。

[CGOP-13621]

Gateway Insight レポートでは、SAML エラーエラーの場合、「認証タイプ」フィールドの値`SAML`が誤って`Local`と表示されます。

[CGOP-13584]

高可用性セットアップでは、NetScaler フェイルオーバー時に、NetScaler Console のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

[CGOP-13511]

ICA 接続が MAC Receiver バージョン 19.6.0.32 または Citrix Virtual Apps and Desktops バージョン 7.18 から起動されると、HDX Insight 機能は無効になります。

[CGOP-13494]

EDT Insight 機能が有効になっていると、ネットワークの不一致時にオーディオチャネルに障害が発生することがあります。

[CGOP-13493]

ブラウザからローカルホスト接続を受け付けると、macOS の **Accept Connection** ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

[CGOP-13050]

一部の言語では、Citrix SSO アプリ > ホームページのテキストHome Pageが切り捨てられます。

[CGOP-13049]

NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

[CGOP-11830]

Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

[CGOP-7269]

負荷分散

高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

[NSLB-7679]

高可用性（HA）セットアップでは、ルートは新しいプライマリノードでドロップされ、次の条件が満たされた場合に再度学習されません。

- 重大なインターフェイス障害により、動的ルートの削除と HA フェールオーバーが同時に発生します。

[NSHELP-32264]

entityofs サービスグループのトラップの ServiceGroupName 形式は次のとおりです。
<service(group)name>?<ip/DBS>?<port>

トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (?) はセパレータとして使用されます。NetScaler は、疑問符（?）付きのトラップを送信します。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

[NSHELP-28080]

特定のシナリオでは、サービスグループにバインドされたサーバーが、無効な Cookie 値を表示します。トレースログで正しい Cookie 値を確認できます。

[NSHELP-21196]

その他

高可用性セットアップで強制同期が行われると、アプライアンスはセカンダリノードでset urlfiltering parameterコマンドを実行します。
その結果、セカンダリノードは、TimeOfDayToUpdateDB パラメータで指定された次のスケジュールされた時刻まで、スケジュールされた更新をスキップします。

[NSSWG-849]

レジストリ値が 2000 バイトを超えると、AlwaysOnAllow リストレジストリが期待どおりに動作しません。

[NSHELP-31836]

URL フィルタリングのサードパーティベンダーで接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

[NSHELP-22409]

ネットワーク

DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

[NSNET-25299]

DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

- NetScaler BLX アプライアンスは、hugepagesの小さい数が割り当てられます。たとえば、1G です。
- NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

この問題は、/var/log/ns.logにエラーメッセージとして記録されます。

- `BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x`

注:x はワーカープロセス数以下の数です。

回避方法:

多数のhugepagesを割り当て、アプライアンスを再起動します。

[NSNET-25173]

DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

[NSNET-24449]

DPDK を搭載した NetScaler BLX アプライアンスの Intel X710 10G (i40e)インターフェイスでは、次のインターフェイス操作はサポートされていません:

- 無効化
- 有効化
- リセット

[NSNET-16559]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

回避方法:

NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します。

- dpkg –add-architecture i386
- apt-get update
- apt-get dist-upgrade
- apt-get install libc6:i386

[NSNET-14602]

FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

[NSNET-5233]

NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

[NSHELP-21082]

プラットフォーム

13.0/12.1/11.1 ビルドから 13.1 ビルドにアップグレードするか、13.1 ビルドから 13.0/12.1/11.1 ビルドにダウングレードすると、一部の Python パッケージが NetScaler アプライアンスにインストールされません。この問題は、次の NetScaler バージョンで修正されています。

- 13.1-4.x
- 13.0—82.31 およびそれ以降
- 12.1—62.21 およびそれ以降

NetScaler バージョンを 13.1-4.x から次のバージョンのいずれかにダウングレードすると、Python パッケージはインストールされません。

- 任意の 11.1 ビルド
- 12.1—62.21 およびそれ以前
- 13.0-81.x およびそれ以前

[NSPLAT-21691]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、2 番目のノードと CLIP で CLAG MAC の不一致があります。

- CLAG は Mellanox NIC 上に作成されます。
- クラスターと CLAG セットアップに別の VPX インスタンスを追加します。

その結果、VPX インスタンスへのトラフィックが停止します。

[NSPLAT-21049]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、CLIP テーブルと MAC テーブルで MAC アドレスの不一致が原因で最初のノードがダウンします。

- CLAG は Mellanox NIC 上に作成されます。
- クラスタから 2 つ目のノードを削除します。

[NSPLAT-21042]

Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。`rm cloudprofile` コマンドを使用して、プロファイルを削除します。

[NSPLAT-4520]

Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノードで設定する必要があります。

[NSPLAT-4451]

ポリシー

処理データのサイズが、設定されているデフォルトの TCP バッファサイズを超えると、接続がハングすることがあります。回避策：TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

[NSPOLICY-1267]

SSL

NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

回避方法：

1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
2. 構成を保存します。

[NSSSL-9572]

認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

[NSSSL-6478]

同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

[NSSSL-6213]

HSM タイプとして Key Vault を指定せずに HSM キーを削除すると、次のような誤ったエラーメッセージが表示されます。
エラー:CRL 更新は無効です

[NSSSL-6106]

セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

[NSSSL-4427]

SSL プロファイルの SSL プロトコルまたは暗号を変更しようとすると、不正な警告メッセージ`Warning: No usable ciphers configured on the SSL vserver/service`,が表示されます。

[NSSSL-4001]

期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。[NSSSL-3184, NSSSL-1379, NSSSL-1394]

NetScaler SDX アプライアンスをリリース 13.1 ビルド 21.50 以降にアップグレードすると、SSL 復号化と MAC 比較が失敗することがあります。その結果、SSL ハンドシェイクの失敗、VPX ステータスのフラッピング、VPX インスタンス GUI の利用不能、仮想サーバーとアプリケーションのダウンが発生する可能性があります。

注: この問題は、SDX 8900、SDX 15000、SDX 15000-50G、SDX 26000、および SDX 26000-50S のプラットフォームで発生します。

[NSHELP-31672]

システム

アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

[NSHELP-21240]

mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

[NSHELP-10972]

クラスタ展開では、非 CCO ノードで`force cluster sync`コマンドを実行すると、ns.log ファイルに重複したログエントリが含まれます。[NSBASE-16304、NSGI-1293]

NetScaler Console を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに機能しません。

回避策: 管理ポッドを再起動します。

[NSBASE-15556]

LogStream トランスポートタイプが Insight に設定されている場合、HDX Insight SkipFlow レコードでクライアント IP とサーバー IP が反転します。

[NSBASE-8506]

SSL サービスで構成された NetScaler アプライアンスは、アプライアンスが TCP FIN 制御パケットの後に TCP RESET 制御パケットを受信するとクラッシュします。

[NSHELP-31656]

ユーザーインターフェイス

MQTT リライト機能では、GUI のエクスプレッションエディタを使用してエクスプレッションを削除することはできません。

回避方法:

CLI から MQTT タイプの add または edit action コマンドを使用します。

[NSUI-18049]

NetScaler GUI では、`Dashboard`タブの下にある`Help`リンクが壊れています。

[NSUI-14752]

CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

回避方法:

NetScaler GUI または CLI を使用して、IPSec プロファイル、IP トンネル、および PBR ルールを追加して、CloudBridge Connector を構成します。

[NSUI-13024]

GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

[NSUI-6838]

高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

- HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

回避方法:

HA 同期が完了した後（両方のノードが同期成功状態にある）のみ、手動の HA フェールオーバーを連続して実行します。

[NSHELP-25598]

NetScaler BLX アプライアンスの高可用性セットアップでは、プライマリノードが応答しなくなり、CLI または API 要求がブロックされることがあります。

回避方法:

プライマリノードを再起動します。

[NSCONFIG-6601]

あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

1. NetScaler アプライアンスをいずれかのビルドにアップグレードします。
   - 13.0 52.24 ビルド
   - 12.1 57.18 ビルド
   - 11.1 65.10 ビルド
2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更して、構成を保存します。
3. NetScaler アプライアンスを古いビルドにダウングレードします。

CLI を使用してこれらのシステムユーザのリストを表示するには、次の手順を実行します。

コマンドプロンプトで入力します:

```
query ns config -changedpassword [-config <full path of the configuration
 file (ns.conf)>]
```

回避方法:

この問題を修正するには、次の独立したオプションのいずれかを使用します。

- NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3）、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
- アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
- 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.htmlを参照してください。

[NSCONFIG-3188]

NetScaler 13.1—27.59 リリースのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリース Build 13.1—27.59 に存在する拡張機能と変更、修正された既知の問題について説明します。

メモ

このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正と勧告のリストについては、セキュリティ速報を参照してください。

新機能

ビルド 13.1～27.59 で利用できる機能強化と変更。

認証、承認、監査

新しい **Microsoft Graph API** で **Intune NAC v2** 構成をユーザーが使用できるようにする　非推奨の AAD グラフ API の代わりに、新しい Microsoft Graph API で Intune NAC v2 構成を使用できるようになりました。

詳細については、「Microsoft Intune インテグレーションとAzure AD Graph の拡張サポート」を参照してください。

[NSAUTH-11897]

ボット管理

NetScaler Gateway デバイスでの **WAF/**ボット管理のためのスタイルブック　NetScaler Gateway デバイスの WAF および BOT ポリシーを構成して、Gateway ログインページを保護できるようになりました。NetScaler Gateway デバイスの WAF/Bot 管理用に、2 つの新しいデフォルトスタイルブックが利用可能になりました。

- WAF と BOT を使用した NetScaler Gateway ログオンサイト保護のためのスタイルブック
- Waf と Bot のセキュリティ違反を伴う WAF と BOT を使用した NetScaler Gateway ログオンサイト保護のためのスタイルブック

ゲートウェイで WAF/Bot 管理にデフォルトの Stylebook を使用するには、[アプリケーション] > [構成] > [StyleBook] に移動します。検索フィールドに StyleBook の名前を入力し、Enter キーを押します。詳しくは、https://docs.citrix.com/en-us/citrix-application-delivery-management-software/current-release/stylebooks/how-to-use-default-stylebooks.html%23to-create-a-configuration-from-a-default-stylebookを参照してください:

[NSBOT-755]

すべての **NetScaler** プレミアム資格でボット検出機能を有効にする　ボット検出機能と署名および IP レピュテーションチェックは、すべての NetScaler Premium 資格に対してデフォルトで有効になりました。

環境に入ってくるボットトラフィックと、NetScaler アプライアンスが実行したアクションを表示できます。また、ADC アプライアンスは、SNMP ログメッセージに次のボットトラフィック情報をキャプチャします。

- 検出されたボットの数
- 検出されたボットの上位 2 つのカテゴリ
- 検出されたボットの詳細を確認できる場所

詳細については、「ボットの検出」を参照してください。

[NSBOT-752]

NetScaler Web App Firewall

新しい署名を自動有効にする　「新しい署名を自動有効にする」を選択して、更新後に新しい WAF 署名のデフォルトルールを自動的に有効にできるようになりました。

[NSWAF-8825]

WAF プロファイルの機密フィールド　WAF プロファイルに機密項目を追加できるようになりました。これらのフィールドはマスクされ、違反が発生しても ADC ログに記録されません。以前は、これらのフィールドは設定を使用してのみ追加できました。

[NSWAF-8525]

HTML ペイロードのカスタムキーワードサポート　任意のキーワードを追加して、これらの設定済みキーワードが HTML ペイロードに存在するかどうかを確認できます。構成されたキーワードが着信要求で検出された場合、NetScaler アプライアンスを構成して、要求をブロックしたり、ログを更新したり、ログカウンタを増やしたりすることができます。

この機能を使用すると、SQL インジェクションおよびコマンドインジェクションチェックでカバーされていないキーワードを追加して、誤検出を減らすことができます。

[NSWAF-8520]

HTML ペイロードでのコマンドインジェクション検出のための文法ベースのアプローチ　NextGen NetScaler Web App Firewall ソリューションは、コマンドインジェクション検出のための文法ベースのアプローチをサポートするように強化されました。このアプローチにより、HTML ペイロードの誤検出が減少します。

以前は、パターンベースのアプローチのみがサポートされていました。

[NSWAF-8270]

ネットワーク

これで、NetScaler CPX の NSIP: 8080 ポートを仮想サーバー構成に使用できます。以前は、このポートは予約されており、ユーザー構成には使用できませんでした。

[NSNET-25399]

Geneve トンネルはクラスタセットアップでサポートします　Geneve トンネルは、NetScaler アプライアンスのクラスタ設定でサポートされるようになりました。

[NSNET-24773]

SNMP Trap メッセージを送信する際の重大度レベルを含めるための機能強化　NetScaler VPX アプライアンスでは、SNMP トラップメッセージに重要度レベルが変数バインドとして含まれるようになりました。以下のコマンドを **severityInfoIntrap** オプションとともに使用します。

- **snmp** オプションを設定する**-severityInfoIntrap** 有効

このオプションを有効にすると、トラップの重大度レベルが SNMP トラップメッセージに含まれます。

[NSNET-21603]

プラットフォーム

AWS での **NetScaler** の高可用性のための **IPv6** アドレスのサポート　NetScaler VPX の高可用性ペアは、同じ AWS アベイラビリティーゾーン内の IPv6 アドレスをサポートするようになりました。以前は、IPv4 アドレスのみがサポートされていました。

[NSPLAT-16672]

ユーザーインターフェイス

Microsoft は、2022 年 6 月から Internet Explorer ブラウザのサポートを終了しました。詳しくは、https://support.microsoft.com/en-us/windows/internet-explorer-help-23360e49-9cd3-4dda-ba52-705336cc0de2を参照してください。

NetScaler リリース 13.1 27.x 以降、NetScaler アプライアンスは、GUI にアクセスするための Internet Explorer をサポートしなくなりました。

Internet Explorer を使用して NetScaler GUI にアクセスすると、NetScaler アプライアンスは、Internet Explorer がサポートされていないというメッセージを表示します。また、GUI にアクセスするためにサポートされているブラウザのリストも推奨されます。

[NSUI-18224]

NetScaler GUI で機能を有効または無効にするための確認プロンプト　NetScaler GUI で、GUI で NetScaler 機能を有効または無効にするときに、操作の確認を求めるプロンプトが表示されるようになりました。確認プロンプトは、NetScaler 機能の偶発的な有効化または無効化を防ぎます。

[NSUI-18098]

解決された問題

ビルド 13.1—27.59 で対処された問題。

認証、承認、監査

`/logon/LogonPoint/Resources/List and /cgi/Resources/List`などのエンドポイントの書き換えポリシーはサポートされていません。

[NSHELP-29488]

NetScaler SDX アプライアンス

Citrix Service 仮想マシンのタイムゾーン設定が期待どおりに動作しません。

[NSHELP-32114]

NetScaler SDX アプライアンスでは、大量の SNMP データ処理により、より高いメモリ使用量が検出されます。

[NSHELP-30222]

SDX-ROOT-MIB:: XenTable の NetScaler SDX アプライアンスで実行されている SNMP ウォークアプリケーションは、予想よりも時間がかかります。

[NSHELP-30085]

NetScaler Gateway

ユーザは、高度なクライアントレス VPN モードでブックマークにアクセスできない場合があります。

[NSHELP-30939]

UDP オーディオ接続用に ICA プロキシモードで構成された NetScaler Gateway アプライアンスは、メモリ破損によりクラッシュすることがあります。

[NSHELP-30919]

ICA アプリの起動は、次の条件で失敗します。

- コンテンツセキュリティポリシー (CSP) 機能が有効になっています。
- ユーザーはブラウザーからログインしますが、Citrix Workspace アプリを使用してアプリを起動します。

[NSHELP-30534]

HDX Insight が有効で、NSAP が無効になっている場合、チャネル解析中に NetScaler Gateway アプライアンスがクラッシュすることがあります。

[NSHELP-30029]

Gateway Insight は、認証ルールがログインフロー内の要求の 1 つと一致するように設定されている場合、ユーザがログイン用の資格情報を送信する前であっても、誤った認証エラーを報告します。

[NSHELP-29313]

セッションプロファイルに StoreFront の FQDN が含まれていると、資格情報を入力するとアプリの起動に失敗します。次のエラーが表示されます。

‘Http/1.1 内部サーバーエラー 43531’

今回の修正により、お客様はセッションプロファイルの WI アドレスの代わりに FQDN を IP に入力できるようになりました。

[NSHELP-26671]

NetScaler Web App Firewall

`No user-agent header action`および`multi user-agent header action`のログは、IP レピュテーションチェックのログメッセージを正しく使用していない可能性があります。

[NSHELP-31935]

NetScaler アプライアンスは、低速の DNS サーバーで BOT 署名ルックアップを処理中にクラッシュすることがあります。

[NSHELP-31642]

署名ルールでクロスサイトスクリプティングが有効になっていると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-31617]

負荷分散

場合によっては、サービスの状態がモニターの状態と同期していません。

[NSHELP-31747]

次の条件が満たされると、ネームサーバーの削除中に NetScaler アプライアンスがクラッシュします。

- DNS サーバーとネームサーバーは、同じ IP アドレスとポートで構成されます。
- リッスンポリシーは DNS サーバーに設定されます。

[NSHELP-31142]

永続性エントリが存在し、多数のダミー負荷分散仮想サーバーとグループ仮想サーバーが構成されている場合、NetScaler アプライアンスがクリア構成中にクラッシュする可能性があります。

[NSHELP-30051]

NetScaler アプライアンスに未解決の WIHOME 構成が存在する場合、ワイルドカード仮想サービスの作成は失敗します。

[NSHELP-25627]

その他

NetScaler アプライアンスでは、追加の HDD がアプライアンスに追加されると、クラッシュフォルダー/var/crash/nslogに/var/nslogファイルへのリンクが作成されます。クラッシュフォルダにあるnewnslogファイルは、テクニカルサポートが生成するコレクタフォルダには収集されません。

[NSHELP-31354]

リソースに割り当てられたメモリが解放されないと、NetScaler SWG アプライアンスがクラッシュし、トラフィックがない場合でもメモリ使用量が高くなる可能性があります。

[NSHELP-31290]

公開鍵システム認証が構成された NetScaler クラスターセットアップでは、次の問題が発生します。

- VTYSH は、クラスタ構成コーディネータ (CCO) のすべてのクラスタノードの情報を表示しません。

[NSHELP-28762]

プラットフォーム

SDX 26000 プラットフォーム（SDX 26100-100G、26160-100G、26200-100G、26250-100G）では、単一の VPX インスタンスに割り当てることができる CPU コアの最大数が 26 から 25CPU コアに変更されました。

[NSPLAT-21233]

BYOL ライセンスは、ALI クラウドプラットフォームで実行されている NetScaler VPX インスタンスには適用できません。

[NSHELP-31546]

SSL

暗号ユニットが VPX インスタンスに割り当てられ、ジャンボ構成が有効になっていると、NetScaler SDX アプライアンスがクラッシュします。

[NSHELP-30950]

NetScaler アプライアンスは、次のシナリオでクラッシュする可能性があります。

- SSL タイプの負荷分散モニターと SSL サービスの名前は同じです
- SSL サービスの名前が変更された
- 負荷分散モニターが削除される

[NSHELP-30445]

SSL インターセプトが有効で、DNS サーバーが有効な DNS 応答を返さない場合、Web サイトへのアクセスはブロックされます。

[NSHELP-30201]

次のすべての状態が発生すると、NetScaler アプライアンスがクラッシュします。

- デフォルトの RSA 証明書とキーのペアは、内部サービスにバインドされています。
- 非 RSA 証明書とキーのペアは、同じサービスにバインドされます。
- 高可用性同期が発生します。

[NSHELP-30084]

rc.netscaler ファイルの一部であるカスタマイズは、システムの初期化中に実行されないため、適用されません。

[NSHELP-31914]

システム

管理している NetScaler コンソールアプライアンスのネットワーク MTU が 1500 を超えると、NetScaler アプライアンスがクラッシュします。

[NSHELP-30835]

クライアント側の測定構成を持つ NetScaler アプライアンスは、次の条件下で変数を破損し、ページの読み込みに失敗する可能性があります。

- HTTP レスポンスには、2000 バイトを超える JavaScript 変数が含まれています。

[NSHELP-30026]

NetScaler アプライアンスでは、デフォルトの高度なグローバルポリシーのバインドを解除して構成を保存すると、次回の再起動時に変更が反映されません。

[NSHELP-19867]

NetScaler アプライアンスは、ヘッダー名フィールドにドット文字を含むカスタム HTTP ヘッダーを含むパケットをドロップします。このアクションは、AllowOnlyWordCharactersAndHyphen パラメーターがデフォルトの HTTP プロファイルでデフォルトで有効になっているために発生します。

13.1-27.x 以降では、デフォルトの HTTP プロファイルセットの allowOnlyWordCharactersAndHyphen パラメーターがデフォルトで無効になっています。ただし、セキュリティを強化するために、このパラメーターを有効にしておくことをお勧めします。

[NSBASE-16722]

ユーザーインターフェイス

NetScaler バージョン 13.0 バージョン 85.15 ビルドでは、GUI を使用して負荷分散サービスグループのメンバーをバインド解除することはできません。

[NSHELP-31474]

NetScaler GUI の **[**システム**] > [**診断] ページには、Advanced ライセンスをお持ちのお客様のページの詳細は表示されません。

[NSHELP-31330]

管理パーティションでは、パケットトレースの記録が期待どおりに動作しない場合があります。

[NSHELP-31321]

CLI インターフェイスで`show run`コマンドを実行中に`CTRL`+`C`を入力すると、NetScaler アプライアンスへの再接続が失敗し、次のエラーが表示されます。

- `Invalid username or password`

この問題は、キーとパスワードの文字が同じ場合に発生します。

[NSHELP-30817]

アップグレードのインストール順序が正しくないため、NetScaler アプライアンスで次の問題が発生します。

- カーネルイメージが最初に更新され、数ステップ後に暗号化キーがコピーされます。これらの手順の間に何らかの障害が発生し、ADC アプライアンスが新しいイメージを作成します。新しいイメージに暗号化キーがないと、復号化に失敗し、設定が失われます。

[NSHELP-30755]

既知の問題

リリース 13.1～27.59 に存在する問題。

AppFlow

HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

[NSINSIGHT-943]

認証、承認、監査

NetScaler アプライアンスは、重複したパスワードログインの試行を認証せず、アカウントのロックアウトを防ぎます。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema が、NetScaler GUI のログインスキーマエディタ画面に正しく表示されません。

[NSAUTH-6106]

ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
`show adfsproxyprofile <profile name>`

回避方法:

クラスタ内のプライマリのアクティブな NetScaler に接続し、`show adfsproxyprofile <profile name>` コマンドを実行します。プロキシプロファイルの状態が表示されます。

[NSAUTH-5916]

次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

- [LDAP 到達可能性をテスト] オプションが開きます。
- 無効なログイン認証情報が入力され、送信されます。
- 有効なログイン認証情報が入力され、送信されます。

回避方法:

[LDAP 到達可能性のテスト] オプションを閉じて開きます。

[NSAUTH-2147]

キャッシュ

統合キャッシュ機能が有効で、アプライアンスのメモリが不足している場合、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-22942]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、CLAG が Mellanox NIC 上に作成されている場合、VPX インスタンスの再起動時に CLAG MAC が変更されます。VPX インスタンスへのトラフィックは再起動後に停止します。これは、MAC テーブルに古い CLAG MAC エントリがあるためです。

[NSSVM-4333]

NetScaler SDX アプライアンスでは、VLAN ホワイトリストが NetScaler VPX インスタンスに割り当てられた Mellanox インターフェイスの正しい値で更新されません。

[NSHELP-31849]

NetScaler SDX アプライアンスをアップグレードすると、ハイパーバイザーのバージョンが現在の SDX バージョンとアップグレード後の SDX バージョンの両方で同じであっても、管理サービス GUI に次の誤ったイベントが通知されます。

SVM と Hypervisor のバージョンが一致しません

[NSHELP-31769]

証明書名またはキー名にスペースが含まれていると、NetScaler SDX アプライアンスへの SSL 証明書のインストールが失敗します。

[NSHELP-31711]

NetScaler Gateway

Citrix Secure Access によって確立されたトンネルの外部にあるリソースへの直接接続は、大幅な遅延または輻輳が発生すると失敗することがあります。

[NSHELP-31598]

Always on が設定されている場合、aoservice.exe ファイルのバージョン番号（1.1.1.1）が正しくないため、ユーザトンネルが失敗します。

[NSHELP-30662]

[NetworkAccessonVPNFailure］プロファイルパラメーターを［フルアクセス］から［OnlyToGateway］に変更すると、ユーザーは NetScaler Gateway アプライアンスに接続できなくなります。

[NSHELP-30236]

ゲートウェイのホームページは、ゲートウェイプラグインが VPN トンネルを正常に確立した直後には表示されません。この問題を解決するために、次のレジストリ値が導入されます。

\ HKLM\ ソフトウェア\ Citrix\ Secure Access Client
セキュアチャネルリセットタイムアウト (秒) タイプ:DWORD

デフォルトでは、このレジストリ値は設定も追加もされません。SecureChannelResetTimeoutSeconds の値が 0 の場合、または追加されていない場合、遅延を処理する修正は機能しません。これはデフォルトの動作です。管理者は、このレジストリをクライアントに設定して修正を有効にする必要があります (つまり、ゲートウェイプラグインが VPN トンネルを正常に確立した直後にホームページを表示する)。

[NSHELP-30189]

Windows VPN クライアントは、サーバからの「SSL close notify」アラートを尊重せず、同じ接続で転送ログイン要求を送信します。

[NSHELP-29675]

サーバー証明書が信頼されていると、サーバー検証コードが失敗することがあります。その結果、エンドユーザーはゲートウェイにアクセスできなくなります。

[NSHELP-28942]

rdx.js ファイルにいくつかの Citrix 内部 IP アドレスがあることに気付くかもしれません。

[NSHELP-28682]

macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

[NSHELP-28551]

クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

[NSHELP-28404]

GUI を使用してクラシック認可ポリシーをバインド解除することはできません。ただし、CLI を使用して認証、承認、および監査認可ポリシーのバインドを解除することはできます。

今回の修正により、GUI を使用して承認ポリシーのバインドを解除できるようになりました。

[NSHELP-27064]

Gateway Insight は、VPN ユーザに関する正確な情報を表示しません。

[NSHELP-23937]

次の条件が満たされている場合、VPN プラグインは Windows のログオン後にトンネルを確立しません。

- NetScaler Gateway アプライアンスが常時オン機能用に構成されている
- アプライアンスは 2 要素認証`off`による証明書ベースの認証用に設定されています。

[NSHELP-23584]

スキーマを参照しているときに、エラーメッセージ`Cannot read property 'type'of undefined`が表示されることがあります。

[NSHELP-21897]

Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

[CGOP-19355]

無効な STA チケットによるアプリケーションの起動失敗は、Gateway Insight で報告されません。

[CGOP-13621]

Gateway Insight レポートでは、SAML エラーエラーの [認証タイプ] フィールドの`SAML`ではなく、値`Local`が誤って表示されます。

[CGOP-13584]

高可用性セットアップでは、NetScaler フェイルオーバー時に、NetScaler Console のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

[CGOP-13511]

ICA 接続が MAC Receiver バージョン 19.6.0.32 または Citrix Virtual Apps and Desktops バージョン 7.18 から起動されると、HDX Insight 機能は無効になります。

[CGOP-13494]

EDT Insight 機能が有効になっていると、ネットワークの不一致時にオーディオチャネルに障害が発生することがあります。

[CGOP-13493]

ブラウザからローカルホスト接続を受け付けると、macOS の Accept Connection ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

[CGOP-13050]

一部の言語では、Citrix SSO アプリ > ホームページのテキスト`Home Page`が切り捨てられます。

[CGOP-13049]

NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

[CGOP-11830]

Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

[CGOP-7269]

NetScaler Web App Firewall

NetScaler アプライアンスは、低速の DNS サーバーで BOT 署名ルックアップを処理中にクラッシュすることがあります。

[NSHELP-31642]

署名ルールでクロスサイトスクリプティングが有効になっていると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-31617]

負荷分散

高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

[NSLB-7679]

場合によっては、サービスの状態がモニターの状態と同期していません。

[NSHELP-31747]

次の条件が満たされると、NetScaler アプライアンスがクラッシュしてコアがダンプされる可能性があります。

- 静的近接または RTT は、プライマリまたはバックアップの負荷分散方法として使用されます。
- 送信元 IP アドレスの永続性が有効になっている

[NSHELP-31735]

`entityofs` サービスグループのトラップの ServiceGroupName 形式は次のとおりです。
`<service(group)name>?<ip/DBS>?<port>`

トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (?) はセパレータとして使用されます。NetScaler は、疑問符（?）付きのトラップを送信します。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

[NSHELP-28080]

特定のシナリオでは、サービスグループにバインドされたサーバーが、無効な Cookie 値を表示します。トレースログで正しい Cookie 値を確認できます。

[NSHELP-21196]

その他

高可用性セットアップで強制同期が実行されると、アプライアンスはセカンダリノードで`set urlfiltering parameter`コマンドを実行します。
その結果、セカンダリノードは、`TimeOfDayToUpdateDB` パラメータで指定された次のスケジュールされた時刻まで、スケジュールされた更新をスキップします。

[NSSWG-849]

レジストリ値が 2000 バイトを超えると、AlwaysOnAllow リストレジストリが期待どおりに動作しません。

[NSHELP-31836]

URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

[NSHELP-22409]

ネットワーク

DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

[NSNET-25299]

DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

- NetScaler BLX アプライアンスは、`hugepages`の小さい数が割り当てられます。たとえば、1G です。
- NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

この問題は、`/var/log/ns.log`にエラーメッセージとして記録されます。

- `BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x`

注:x はワーカープロセス数以下の数です。

回避方法:

多数の`hugepages`を割り当て、アプライアンスを再起動します。

[NSNET-25173]

DPDK を搭載した NetScaler BLX アプライアンスは、次の条件が満たされると再起動に失敗することがあります。

- NetScaler BLX アプライアンスにはhugepagesの大きい数が割り当てられます。たとえば、16 GB と入力します。

この問題は、/var/log/ns.logにエラーメッセージとして記録されます。

- EAL: rte_mem_virt2phy(): cannot open /proc/self/pagemap: Too many open files

回避方法:

この問題を解決するには、次のいずれかの回避策を使用します。

- ulimitコマンドを使用するか、limits.confファイルを編集して、Linux ホストで開くことができるファイルの上限を増やします。
- 割り当てられたhugepagesの数を減らします。

[NSNET-24727]

DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

[NSNET-24449]

DPDK を搭載した NetScaler BLX アプライアンスの Intel X710 10G (i40e)インターフェイスでは、次のインターフェイス操作はサポートされていません:

- 無効化
- 有効化
- リセット

[NSNET-16559]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

The following packages have unmet dependencies: blx-core-libs:i386 : PreDepends: libc6:i386 (>= 2.19)but it is not installable

回避方法:

NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します。

- dpkg –add-architecture i386
- apt-get update
- apt-get dist-upgrade
- apt-get install libc6:i386

[NSNET-14602]

FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

[NSNET-5233]

NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

[NSHELP-21082]

プラットフォーム

高可用性フェイルオーバーは AWS および GCP クラウドでは機能しません。AWS および GCP クラウド、および NetScaler VPX オンプレミスでは、管理 CPU が 100% の容量に達する可能性があります。これらの問題はいずれも、次の条件が満たされた場合に発生します。

1. NetScaler アプライアンスの初回起動時には、プロンプトが表示されたパスワードは保存されません。
2. その後、NetScaler アプライアンスを再起動します。

[NSPLAT-22013]

13.0/12.1/11.1 ビルドから 13.1 ビルドにアップグレードするか、13.1 ビルドから 13.0/12.1/11.1 ビルドにダウングレードすると、一部の Python パッケージが NetScaler アプライアンスにインストールされません。この問題は、次の NetScaler バージョンで修正されています。

- 13.1-4.x
- 13.0-82.31 以降
- 12.1-62.21 以降

NetScaler バージョンを 13.1-4.x から次のバージョンのいずれかにダウングレードすると、Python パッケージはインストールされません。

- 任意の 11.1 ビルド
- 12.1-62.21 およびそれ以前
- 13.0-81.x およびそれ以前

[NSPLAT-21691]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、2 番目のノードと CLIP で CLAG MAC の不一致があります。

- CLAG は Mellanox NIC 上に作成されます。
- クラスターと CLAG セットアップに別の VPX インスタンスを追加します。

その結果、VPX インスタンスへのトラフィックが停止します。

[NSPLAT-21049]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、CLIP テーブルと MAC テーブルで MAC アドレスの不一致が原因で最初のノードがダウンします。

- CLAG は Mellanox NIC 上に作成されます。
- クラスタから 2 つ目のノードを削除します。

[NSPLAT-21042]

Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。`rm cloudprofile` コマンドを使用して、プロファイルを削除します。

[NSPLAT-4520]

Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノード上で設定する必要があります。

[NSPLAT-4451]

NetScaler リリース 13.1 以降、NetScaler アプライアンスは、8 つを超える VMXNET3 ネットワークインターフェイスを備えた ESXi ハイパーバイザーで起動に失敗します。

[NSHELP-31266]

ポリシー

処理データのサイズが、設定されているデフォルトの TCP バッファサイズを超えると、接続がハングすることがあります。回避策：TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

[NSPOLICY-1267]

SSL

NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

回避方法:

1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
2. 構成を保存します。

[NSSSL-9572]

認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

[NSSSL-6478]

同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

[NSSSL-6213]

HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。エラー:CRL 更新は無効です

[NSSSL-6106]

セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

[NSSSL-4427]

SSL プロファイルの SSL プロトコルまたは暗号を変更しようとすると、不正な警告メッセージ`Warning: No usable ciphers configured on the SSL vserver/service,`が表示されます。

[NSSSL-4001]

期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。[NSSSL-3184, NSSSL-1379, NSSSL-1394]

NetScaler アプライアンスは、次のシナリオでクラッシュする可能性があります。

- SSL タイプの負荷分散モニターと SSL サービスの名前は同じです
- SSL サービスの名前が変更された
- 負荷分散モニターが削除される

[NSHELP-30445]

システム

アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

[NSHELP-21240]

mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

[NSHELP-10972]

クラスタ展開では、非 CCO ノードで`force cluster sync`コマンドを実行すると、ns.log ファイルに重複したログエントリが含まれます。[NSBASE-16304、NSGI-1293]

NetScaler Console を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに機能しません。

回避策: 管理ポッドを再起動します。

[NSBASE-15556]

LogStream トランスポートタイプが Insight に設定されている場合、HDX Insight SkipFlow レコードでクライアント IP とサーバー IP が反転します。

[NSBASE-8506]

ユーザーインターフェイス

MQTT リライト機能では、GUI のエクスプレッションエディタを使用してエクスプレッションを削除することはできません。

回避方法:

CLI から MQTT タイプの add または edit action コマンドを使用します。

[NSUI-18049]

NetScaler GUI では、`Dashboard`タブの下にある`Help`リンクが壊れています。

[NSUI-14752]

CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

回避方法:

NetScaler GUI または CLI を使用して、IPSec プロファイル、IP トンネル、および PBR ルールを追加して、CloudBridge Connector を構成します。

[NSUI-13024]

GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

[NSUI-6838]

高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

- HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

回避方法:

HA 同期が完了した後（両方のノードが同期成功状態にある）のみ、手動の HA フェールオーバーを連続して実行します。

[NSHELP-25598]

NetScaler BLX アプライアンスの高可用性セットアップでは、プライマリノードが応答しなくなり、CLI または API 要求がブロックされることがあります。

回避方法:

プライマリノードを再起動します。

[NSCONFIG-6601]

あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

1. NetScaler アプライアンスをいずれかのビルドにアップグレードします。
    - 13.0 52.24 ビルド
    - 12.1 57.18 ビルド
    - 11.1 65.10 ビルド
2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
3. NetScaler アプライアンスを古いビルドにダウングレードします。

CLI を使用してこれらのシステムユーザーの一覧を表示するには、
コマンドプロンプトで次のように入力します。

```
query ns config -changedpassword [-config <full path of the configuration
 file (ns.conf)>]
```

回避方法:

この問題を修正するには、次の独立したオプションのいずれかを使用します。

- NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3）、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
- アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
- 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.htmlを参照してください。

[NSCONFIG-3188]

NetScaler 13.1-24.38 リリースのリリースノート

March 20, 2024

このリリースノートのドキュメントでは、NetScaler リリースビルド 13.1-24.38 に存在する機能強化と変更、修正された既知の問題について説明します。

メモ

このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正と勧告のリストについては、セキュリティ速報を参照してください。

ビルド 13.1-24.38 以降のビルドは、https://support.citrix.com/article/CTX457836で説明されているセキュリティの脆弱性に対処します。

新機能

ビルド 13.1-24.38 で利用できる機能強化と変更。

負荷分散

高可用性 **INC** モードの接続フェイルオーバーサポート　NetScaler は、次の条件がすべて満たされた場合に、高可用性 INC モードの接続フェイルオーバーをサポートするようになりました。

- 仮想サーバーのサービスタイプは ANY です。
- モードは DSR (MAC、IPTUNNEL、または TOS) です。
- USIP は、仮想サーバーにバインドされたサービスで有効になっています。

[NSLB-9121]

CAA レコードのサポート　NetScaler アプライアンスは、認証局認証（CAA）レコードの追加をサポートするようになりました。CAA レコードは、ドメインの所有者が、ドメインの SSL 証明書を発行できる認証局 (CA) を指定できるドメインネームシステム (DNS) レコードの一種です。

この機能強化により、Web プレゼンスをさらに保護するレイヤーが提供されます。CAA レコードがないと、誰でもドメインの証明書署名要求 (CSR) を生成し、任意の CA によって署名された証明書を取得できるため、セキュリティ上のリスクが生じる可能性があります。

[NSLB-9007]

プラットフォーム

NetScaler SDX 8015 プラットフォームでは、ライトアウト管理（LOM）バージョンが 3.21 から 3.56 にアップグレードされます。

NetScaler SDX 14000、SDX 14000-40G、SDX 14000-40S、および SDX 14000-FIPS プラットフォームでは、LOM バージョンが 4.08 から 4.14 にアップグレードされます。

[NSPLAT-23416]

リソースグループ全体で **NetScaler** バックエンド **Autoscale** を **VMSS** で **Azure** でサポート　NetScaler VPX インスタンスは、次のシナリオでリソースグループ全体で Azure バックエンド自動スケーリングをサポートするようになりました。

Azure VMSS と NetScaler VPX インスタンスは、同じ Azure 仮想ネットワークに展開されます。
Azure VMSS と NetScaler VPX インスタンスは、同じ Azure サブスクリプション内の異なる Azure 仮想ネットワークに展開されます。これら 2 つの仮想ネットワークは、Azure の仮想ネットワークピアリング機能を使用して接続する必要があります。

この機能により、アプリケーションとネットワークリソースを異なるリソースグループに分離できます。

以前は、Azure での NetScaler バックエンド AAutoscale は、VMSS と NetScaler VPX インスタンスが同じリソースグループに展開されている場合にのみ機能します。

[NSPLAT-16664]

システム

メトリクスコレクターのカウンターを購読する　NetScaler アプライアンスは、メトリックコレクターでカウンターをサブスクライブするオプションをサポートするようになりました。
メトリクスコレクターは、AVRO、Prometheus 形式、Influx DB 形式などのさまざまな形式で、30 秒ごとの時系列分析データのエクスポートをサポートします。メトリクスコレクターは、必要なカウンターをスキーマファイルに追加できるカウンターの動的更新をサポートしています。スキーマファイル名は CLI インターフェイスを使用して設定できます。メトリクスコレクターは、スキーマファイルからカウンター名を読み取り、エクスポートします。

以前は、メトリクスコレクターは、コンパイル時に定義済みのカウンターセットのエクスポートのみをサポートしていました。カウンターのリストを変更すると、ビルドのアップグレードが必要でした。
詳細については、https://docs.citrix.com/en-us/citrix-adc/13-1/ns-ag-appflow-intro-wrapper-con/ns-ag-appflow-config-tsk.htmlを参照してください。

[NSBASE-11595]

ユーザーインターフェイス

NetScaler ライセンス有効期限アラートを構成する　NetScaler ライセンスの有効期限が切れる前の指定された日数から次のアラート操作を実行するように NetScaler アプライアンスを構成できるようになりました。

- NetScaler GUI にライセンス有効期限の警告バナーを表示します。
- SNMP アラームが有効な場合、ライセンスの有効期限情報を含む NS_LICENSE_EXPIRY SNMP トラップを、設定されたトラップリスナーに定期的に送信します。

[NSCONFIG-6360]

解決された問題

ビルド 13.1-24.38 で対処されている問題。

認証、承認、監査

ユニファイドゲートウェイのセットアップでは、認証が成功した後でも、ユニファイドゲートウェイの背後にあるサービスにアクセスすると、まれに再ログインページが表示されることがあります。

[NSHELP-31148、NSHELP-2799]

フォームベースの SSO は、URL クエリで key-value パラメーターを送信するバックエンドサーバーで失敗します。

[NSHELP-30975]

OAuth 構成にターゲット URL がないため、大量のメモリ割り当てが原因で NetScaler アプライアンスがクラッシュする可能性があります。

[NSHELP-30963]

Chrome をシークレットモードで使用しているときに、RADIUS 認証で断続的な問題が発生することがあります。

[NSHELP-30944]

NetScaler アプライアンスの認証、承認、および auditingD モジュールは、パケットエンジンから認証、承認、および auditingD への着信パスワードの長さがないか正しくないためにクラッシュする可能性があります。

[NSHELP-30911]

NetScaler アプライアンスは、nFactor プッシュ操作中にクラッシュします。

[NSHELP-30577]

NetScaler アプライアンスによる誤ったヘッダー追加により、Outlook アプリを介して Outlook 交換サーバーに接続する際に断続的な障害が発生する可能性があります。

[NSHELP-30555]

NetScaler アプライアンスは、コア間の通信障害時にメモリ破損によりクラッシュする可能性があります。

[NSHELP-30275]

パスワード変更イベントがトリガーされると、認証セッション中にシングルサインオンが失敗します。この問題は、PersistentLogin attempts パラメーターが有効になっている場合にのみ発生します。

[NSHELP-28085]

RADIUS `invalid credentials` 認証プロセス中にエラーメッセージが表示されることがあります。このエラーは、Google Chrome ブラウザを使用してクライアントデバイスから NetScaler アプライアンスにアクセスすると表示されます。

[NSHELP-27113]

NetScaler アプライアンスがネストされた LDAP グループ検索を実行すると、NetScaler アプライアンスの動作が無効なため、Active Directory のグループ情報の一部が失われます。`groupSearchSubAttribute` パラメータが適切に設定されている場合でも、ADC アプライアンスは誤った値をとります。

[NSHELP-26316]

NetScaler アプライアンスは、NOAUTH が 401 ベースの認証フローの最初の要素として構成され、後続の要素としてネゴシエートが構成されている場合にコアをダンプします。

[NSHELP-25203]

NetScaler SDX アプライアンス

NetScaler SDX GUI で、NTP 構成ファイル（ntp.conf）のいずれかの行にスペースしか含まれていない場合、NTP サーバーを表示するとユーザーインターフェイスがフリーズする可能性があります。

[NSHELP-31530]

Mellanox NIC を搭載した NetScaler SDX アプライアンスで、Mellanox NIC を持つ VPX インスタンスのスループットを変更すると、VPX インスタンスが再起動します。

[NSHELP-31305]

NetScaler SDX アプライアンスをリリース 13.1 ビルド 21.50 以降にアップグレードすると、SSL 復号化と MAC 比較が失敗することがあります。その結果、SSL ハンドシェイクの失敗、VPX ステータスのフラッピング、VPX インスタンス GUI の利用不能、仮想サーバーとアプリケーションのダウンが発生する可能性があります。

注：この問題は、SDX 8900、SDX 15000、SDX 15000-50G、SDX 26000、および SDX 26000-50S のプラットフォームで発生します。

[NSHELP-31672]

NetScaler Gateway

まれに、VPN 仮想サーバーで構成された NetScaler アプライアンスが、NetScaler Gateway へのログインに成功した後にクラッシュすることがあります。

[NSHELP-31481]

ICA DTLS セットアップで、STA チケットを処理すると NetScaler Gateway アプライアンスがクラッシュします。

[NSHELP-31211]

NetScaler アプライアンスは、`UDPFLOWSTAT` 承認ポリシーによって拒否された UDP トラフィックに関するトラフィックを示すメッセージを誤って記録します。`Allowed`

[NSHELP-29542]

送信プロキシが構成されている場合、NetScaler アプライアンスでメモリリークが発生する。

[NSHELP-29234]

[アクティブユーザセッション（Active Users Session）] ページには、エントリ数が 1 ページあたり 2000 に変更されない限り、すべてのアクティブユーザセッションは表示されません。

今回の修正により、すべてのユーザーセッションと接続を一覧表示する新しいリンク `All user session`（NetScaler Gateway-> 接続を監視 > すべてのユーザーセッション）が管理 UI に追加されました。

[NSHELP-29151]

`show vpn icaConnection` コマンド出力に、ICA 接続のシリアル番号が正しく表示されません。この問題は、`show vpn icaconnection` の実行時にシリアル番号が任意にリセットされるために発生します。

[NSHELP-25646]

NetScaler Web App Firewall

Web App Firewall ポリシーは、構成 (`ns.conf`) ファイルに 2 回保存できます。

[NSHELP-30899]

引用符（一重引用符、二重引用符、またはバックティック）を含む WAF SQL インジェクションでは、パターンを攻撃としてマークするために開始引用符と終了引用符が存在する必要があります。ただし、パターンにコメントがある場合、閉じ引用符は不要です。

[NSHELP-30379]

負荷分散

ADC アプライアンスで ECS が有効になっていて、場所が見つからない場合、スコーププレフィックスが正しく設定されません。この問題の結果、誤った永続性エントリが作成されます。不正な永続性エントリは、非静的近接ベースの GSLB 方式の要求で受信した ECS IP アドレスではなく、LDNS IP アドレスに基づいて作成されます。

[NSHELP-30846]

まれな競合状態のシナリオでは、NetScaler アプライアンスに次の構成が存在する場合、パケットエンジンがコアダンプでクラッシュする可能性があります。

- GSLB 仮想サーバーは、ソース IP アドレスベースの永続性で構成され、DNS ロギングは ADNS サービスにバインドされた DNS プロファイルで有効になります。
- DNS 負荷分散サーバーは、DNS プロファイルで DNS ログが有効になっていない状態で構成されます。

[NSHELP-29791]

その他

ポータル jQuery UI が 1.12.1 から 1.13.1 に更新され、セキュリティ情報:CVE-2021-41182、CVE-2021-41183、および CVE-2021-41184 で説明されている脆弱性に対処します。

[NSHELP-30209]

ネットワーク

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスは、BLX 構成ファイル（`/etc/blx/blx.conf`）の設定に関係なく、常に共有モードで展開されます。この問題は、Debian ベースの Linux システムにデフォルトで存在する`mawk`が、`blx.conf`ファイルにある awk コマンドの一部を実行しないために発生します。

[NSNET-14603]

大規模な NAT44 セットアップでは、次の理由により、SIP トラフィックの受信中に NetScaler アプライアンスがクラッシュすることがあります。

- LSN フィルタリングとマッピングのエントリは、アプライアンスには存在しません。

[NSHELP-30225]

一部のパケットが ACL ルールに一致したときに、ACL ルールからデータセットをバインド解除すると、NetScaler アプライアンスがクラッシュする可能性があります。

[NSHELP-30221]

大規模な NAT44 セットアップでは、次の理由により、SIP トラフィックの受信中に NetScaler アプライアンスがクラッシュすることがあります。

- フィルタリングエントリの削除中、セッション参照カウントはゼロではありません。

[NSHELP-29348]

プラットフォーム

シングルバンドルイメージ（SBI）および VPX バージョン 13.1～24.x 以降を備えた NetScaler SDX アプライアンスでは、Fortville NIC で VRRP を使用するアクティブ-アクティブ展開がサポートされます。この展開は L2 モードではサポートされていません。

展開には次の点が適用されます。

- 関連する VPX インスタンスをアップグレードまたはダウングレードする前に、管理サービスから VRID 構成を削除することをお勧めします。アップグレードまたはダウングレード操作が完了したら、管理サービスから VRID 構成を追加します。
- 前述の推奨事項に従わない場合は、Management Service から VPX インスタンスを手動で再検出して、VRRP コンバージェンスを有効にする必要があります。

[NSHELP-30670]

RPC ノードのパスワードに特殊文字が含まれていると、GCP および AWS クラウド上の NetScaler VPX インスタンスの HA フェイルオーバーが失敗します。

[NSHELP-28600]

ポリシー

一部のシナリオでは、AppExpert 変数のクリア操作で割り当てアクションを使用すると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-29766]

SSL

NetScaler MPX/SDX 14000 FIPS アプライアンスは、SAML などの内部アプリケーションによる暗号化操作のための API の継続的な使用により、一定期間にわたってクラッシュする可能性があります。

[NSHELP-27952]

システム

AppFlow パラメーター`TimeSeriesOverNSIP`が有効になっていても、REST コレクターは停止します。

[NSHELP-30759]

NetScaler アプライアンスでは、次の条件が満たされると、HTTP/2 トランザクションで遅延の問題が発生します。

- バックエンドサービスで HTTP/2 SSL 設定が有効になっている
- サービスは HTTP/2 プロトコルをサポートしていません。

[NSHELP-30020]

NetScaler アプライアンスは、サービス SYN フラッドカウンターで誤った SNMP アラームを報告します。

[NSHELP-28710, NSHELP-28713]

ユーザーインターフェイス

プールライセンスで構成された NetScaler アプライアンスをアップグレードすると、アプライアンスが部分的な構成で再起動することがあります。

[NSHELP-30926]

NetScaler アプライアンスで、GUI インターフェイスを使用してグローバルまたはデフォルトグローバルをオーバーライドするようにキャッシュポリシーをバインドすると、次のエラーで失敗します。

- 必須の引数がありません。

このエラーは、CLI インターフェイスを使用してキャッシュポリシーをバインドしている間は発生しません。

[NSHELP-30826]

検索フィルターは、NetScaler GUI の［証明書の管理］>［CSR］ページの［名前］キーには使用できません。

[NSHELP-30274]

既知の問題

リリース 13.1-24.38 に存在する問題。

AppFlow

HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

[NSINSIGHT-943]

認証、承認、監査

NetScaler アプライアンスは、重複したパスワードログインの試行を認証せず、アカウントのロックアウトを防ぎます。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema が NetScaler GUI のログインスキーマエディタ画面に正しく表示されません。

[NSAUTH-6106]

ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
`show adfsproxyprofile <profile name>`

回避方法:

クラスタ内のプライマリのアクティブな NetScaler に接続し、`show adfsproxyprofile <profile name>` コマンドを実行します。プロキシプロファイルの状態が表示されます。

[NSAUTH-5916]

次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

- [LDAP 到達可能性をテスト] オプションが開きます。
- 無効なログイン認証情報が入力され、送信されます。
- 有効なログイン認証情報が入力され、送信されます。

回避方法:

[LDAP 到達可能性のテスト] オプションを閉じて開きます。

[NSAUTH-2147]

キャッシュ

統合キャッシュ機能が有効で、アプライアンスのメモリが不足している場合、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-22942]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、CLAG が Mellanox NIC 上に作成されている場合、VPX インスタンスの再起動時に CLAG MAC が変更されます。VPX インスタンスへのトラフィックは再起動後に停止します。これは、MAC テーブルに古い CLAG MAC エントリがあるためです。

[NSSVM-4333]

証明書名またはキー名にスペースが含まれていると、NetScaler SDX アプライアンスへの SSL 証明書のインストールが失敗します。

[NSHELP-31711]

NetScaler Gateway

Always on が設定されている場合、aoservice.exe ファイルのバージョン番号（1.1.1.1）が正しくないため、ユーザトンネルが失敗します。

[NSHELP-30662]

[NetworkAccessonVPNFailure］プロファイルパラメーターを［フルアクセス］から［OnlyToGateway］に変更すると、ユーザーは NetScaler Gateway アプライアンスに接続できなくなります。

[NSHELP-30236]

ゲートウェイのホームページは、ゲートウェイプラグインが VPN トンネルを正常に確立した直後には表示されません。この問題を解決するために、次のレジストリ値が導入されます。
\HKLM\Software\Citrix\Secure Access Client\SecureChannelResetTimeoutSeconds
種類：DWORD

[NSHELP-30189]

Windows VPN クライアントは、サーバからの「SSL close notify」アラートを尊重せず、同じ接続で転送ログイン要求を送信します。

[NSHELP-29675]

サーバー証明書が信頼されていると、サーバー検証コードが失敗することがあります。その結果、エンドユーザーはゲートウェイにアクセスできなくなります。

[NSHELP-28942]

rdx.js ファイルにいくつかの Citrix 内部 IP アドレスがあることに気付くかもしれません。

[NSHELP-28682]

macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

[NSHELP-28551]

クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

[NSHELP-28404]

GUI を使用してクラシック認可ポリシーをバインド解除することはできません。ただし、CLI を使用して認証、承認、および監査認可ポリシーのバインドを解除することはできます。

今回の修正により、GUI を使用して承認ポリシーのバインドを解除できるようになりました。

[NSHELP-27064]

Gateway Insight は、VPN ユーザに関する正確な情報を表示しません。

[NSHELP-23937]

次の条件を満たす場合、Windows ログオン後に VPN プラグインはトンネルを確立しません。

- NetScaler Gateway アプライアンスが常時オン機能用に構成されている
- アプライアンスは 2 要素認証`off`による証明書ベースの認証用に設定されています。

[NSHELP-23584]

スキーマを参照しているときに、エラーメッセージ`Cannot read property 'type'of undefined`が表示されることがあります。

[NSHELP-21897]

Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

[CGOP-19355]

無効な STA チケットによるアプリケーションの起動失敗は、Gateway Insight では報告されません。

[CGOP-13621]

Gateway Insight レポートでは、SAML エラーエラーの [認証タイプ] フィールドの`SAML`ではなく、値`Local`が誤って表示されます。

[CGOP-13584]

高可用性セットアップでは、NetScaler フェイルオーバー時に、NetScaler Console のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

[CGOP-13511]

ICA 接続が MAC Receiver バージョン 19.6.0.32 または Citrix Virtual Apps and Desktops バージョン 7.18 から起動されると、HDX Insight 機能は無効になります。

[CGOP-13494]

EDT Insight 機能が有効になっていると、ネットワークの不一致時にオーディオチャネルに障害が発生することがあります。

[CGOP-13493]

ブラウザからローカルホスト接続を受け付けると、macOS の Accept Connection ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

[CGOP-13050]

一部の言語では、Citrix SSO アプリ > ホームページのテキストHome Pageが切り捨てられます。

[CGOP-13049]

NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

[CGOP-11830]

Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

[CGOP-7269]

負荷分散

高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

[NSLB-7679]

entityofs サービスグループのトラップの ServiceGroupName 形式は次のとおりです。
<service(group)name>?<ip/DBS>?<port>

トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (?) はセパレータとして使用されます。NetScaler は、疑問符（?）付きのトラップを送信します。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

[NSHELP-28080]

特定のシナリオでは、サービスグループにバインドされたサーバーが、無効な Cookie 値を表示します。トレースログで正しい Cookie 値を確認できます。

[NSHELP-21196]

その他

高可用性セットアップで強制同期が実行されると、アプライアンスはセカンダリノードでset urlfiltering parameterコマンドを実行します。
その結果、セカンダリノードは、TimeOfDayToUpdateDB パラメータで指定された次のスケジュールされた時刻まで、スケジュールされた更新をスキップします。

[NSSWG-849]

URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

[NSHELP-22409]

ネットワーク

DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

[NSNET-25299]

DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

- NetScaler BLX アプライアンスは、hugepagesの小さい数が割り当てられます。たとえば、1G です。
- NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

この問題は、/var/log/ns.logにエラーメッセージとして記録されます。

- `BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x`

注:x はワーカープロセス数以下の数です。

回避方法:

多数のhugepagesを割り当て、アプライアンスを再起動します。

[NSNET-25173]

DPDK を搭載した NetScaler BLX アプライアンスは、次の条件が満たされると再起動に失敗することがあります。

- NetScaler BLX アプライアンスにはhugepagesの大きい数が割り当てられます。たとえば、16 GB と入力します。

この問題は、/var/log/ns.logにエラーメッセージとして記録されます。

- `EAL: rte_mem_virt2phy(): cannot open /proc/self/pagemap: Too many open files`

回避方法:

この問題を解決するには、次のいずれかの回避策を使用します。

- ulimitコマンドを使用するか、limits.confファイルを編集して、Linux ホストで開くことができるファイルの上限を増やします。
- 割り当てられたhugepagesの数を減らします。

[NSNET-24727]

DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

[NSNET-24449]

DPDK を搭載した NetScaler BLX アプライアンスの Intel X710 10G (i40e)インターフェイスでは、次のインターフェイス操作はサポートされていません:

- 無効化
- 有効化
- リセット

[NSNET-16559]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

回避方法:

NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します。

- dpkg –add-architecture i386
- apt-get update
- apt-get dist-upgrade
- apt-get install libc6:i386

[NSNET-14602]

FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

[NSNET-5233]

NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

[NSHELP-21082]

プラットフォーム

高可用性フェイルオーバーは AWS および GCP クラウドでは機能しません。AWS および GCP クラウド、および NetScaler VPX オンプレミスでは、管理 CPU が 100% の容量に達する可能性があります。これらの問題はいずれ

も、次の条件が満たされた場合に発生します。

1. NetScaler アプライアンスの初回起動時には、プロンプトが表示されたパスワードは保存されません。
2. その後、NetScaler アプライアンスを再起動します。

[NSPLAT-22013]

13.0/12.1/11.1 ビルドから 13.1 ビルドにアップグレードするか、13.1 ビルドから 13.0/12.1/11.1 ビルドにダウングレードすると、一部の Python パッケージが NetScaler アプライアンスにインストールされません。この問題は、次の NetScaler バージョンで修正されています。

- 13.1-4.x
- 13.0-82.31 以降
- 12.1-62.21 以降

NetScaler バージョンを 13.1-4.x から次のバージョンのいずれかにダウングレードすると、Python パッケージはインストールされません。

- 任意の 11.1 ビルド
- 12.1-62.21 およびそれ以前
- 13.0-81.x およびそれ以前

[NSPLAT-21691]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、2 番目のノードと CLIP で CLAG MAC の不一致があります。

- CLAG は Mellanox NIC 上に作成されます。
- クラスターと CLAG セットアップに別の VPX インスタンスを追加します。

その結果、VPX インスタンスへのトラフィックが停止します。

[NSPLAT-21049]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、CLIP テーブルと MAC テーブルで MAC アドレスの不一致が原因で最初のノードがダウンします。

- CLAG は Mellanox NIC 上に作成されます。
- クラスタから 2 つ目のノードを削除します。

[NSPLAT-21042]

Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。`rm cloudprofile` コマンドを使用して、プロファイルを削除します。

[NSPLAT-4520]

Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノード上で設定する必要があります。

[NSPLAT-4451]

NetScaler リリース 13.1 以降、NetScaler アプライアンスは、8 つを超える VMXNET3 ネットワークインターフェイスを備えた ESXi ハイパーバイザーで起動に失敗します。

[NSHELP-31266]

ポリシー

処理データのサイズが、設定されているデフォルトの TCP バッファサイズを超えると、接続がハングすることがあります。回避策：TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

[NSPOLICY-1267]

SSL

NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

回避方法:

1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
2. 構成を保存します。

[NSSSL-9572]

認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

[NSSSL-6478]

同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

[NSSSL-6213]

HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
エラー:CRL 更新は無効です

[NSSSL-6106]

セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

[NSSSL-4427]

SSL プロファイルの SSL プロトコルまたは暗号を変更しようとすると、不正な警告メッセージ`Warning: No usable ciphers configured on the SSL vserver/service,`が表示されます。

[NSSSL-4001]

期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。[NSSSL-3184, NSSSL-1379, NSSSL-1394]

MPX 8900 および MPX 15000 FIPS 認定アプライアンスでは、ECDHE トラフィックを実行するとメモリリークが発生する可能性があります。

[NSHELP-30744]

rc.netscaler ファイルの一部であるカスタマイズは、システムの初期化中に実行されないため、適用されません。

[NSHELP-31914]

システム

アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

[NSHELP-21240]

mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

[NSHELP-10972]

クラスタ展開では、非 CCO ノードで`force cluster sync`コマンドを実行すると、ns.log ファイルに重複したログエントリが含まれます。[NSBASE-16304、NSGI-1293]

NetScaler Console を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに機能しません。

回避策：管理ポッドを再起動します。

[NSBASE-15556]

LogStream トランスポートタイプが Insight 用に構成されている場合、クライアント IP とサーバー IP は HDX Insight SkipFlow レコードで反転されます。

[NSBASE-8506]

NetScaler アプライアンスは、ヘッダー名フィールドにドット（”.”）の付いたカスタム HTTP ヘッダーを含むパケットをドロップします。このアクションは、`allowOnlyWordCharactersAndHyphen` パラメータがデフォルトの HTTP プロファイルでデフォルトで有効になっているために発生します。

回避策: デフォルトの HTTP プロファイルで`allowOnlyWordCharactersAndHyphen`を無効にします。ただし、Citrix では有効にしておくことをお勧めします。

[NSBASE-16722]

ユーザーインターフェイス

MQTT リライト機能では、GUI のエクスプレッションエディタを使用してエクスプレッションを削除することはできません。

回避方法:

CLI から MQTT タイプの add または edit action コマンドを使用します。

[NSUI-18049]

NetScaler GUI では、`Dashboard`タブの下にある`Help`リンクが壊れています。

[NSUI-14752]

CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

回避方法:

NetScaler GUI または CLI を使用して、IPsec プロファイル、IP トンネル、および PBR ルールを追加して、CloudBridge Connector を構成します。

[NSUI-13024]

GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

[NSUI-6838]

高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

- HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

回避方法:

HA 同期が完了した後（両方のノードが同期成功状態にある）のみ、手動の HA フェールオーバーを連続して実行します。

[NSHELP-25598]

NetScaler BLX アプライアンスの高可用性セットアップでは、プライマリノードが応答しなくなり、CLI または API 要求がブロックされることがあります。

回避方法:

プライマリノードを再起動します。

[NSCONFIG-6601]

あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

1. NetScaler アプライアンスをいずれかのビルドにアップグレードします。

- 13.0 52.24 ビルド
- 12.1 57.18 ビルド
- 11.1 65.10 ビルド

1. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
2. NetScaler アプライアンスを古いビルドにダウングレードします。

CLI を使用してこれらのシステムユーザーの一覧を表示するには、
コマンドプロンプトで次のように入力します。

```
query ns config -changedpassword [-config <full path of the configuration
 file (ns.conf)>]
```

回避方法:

この問題を修正するには、次の独立したオプションのいずれかを使用します。

- NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3）、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
- アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
- 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.htmlを参照してください。

[NSCONFIG-3188]

メモ

April 15, 2024

このリリースノートドキュメントでは、NetScaler リリース Build 13.1～21.50 に存在する機能強化と変更、修正された問題、既知の問題について説明します。

このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正と勧告のリストについては、セキュリティ速報を参照してください。

ビルド 13.1-21.50 以降のビルドは、https://support.citrix.com/article/CTX457048で説明されているセキュリティ脆弱性に対処します。

新機能

ビルド 13.1—21.50 で使用できる機能強化と変更点。

ボット管理

ユーザーの地理的位置に基づくボットレート制限手法　ボットレート制限検出技術により、ユーザーの地理的位置に基づいてトラフィックボットを制限できるようになりました。この設定では、国名を URL または Cookie 名に似た値として設定できます。これにより、国ごとに異なるレート制限を適用できます。以前の検出技術では、クライアントの IP アドレス、セッション、または URL のみに基づいてトラフィックをレート制限できました。

[NSBOT-753]

ヘッドレスブラウザ検出のためのデバイスフィンガープリント **(DFP)** 技術の拡張　ハッカーは、複数ユーザーアカウントの作成、チケットの予約、価格の廃棄、クレデンシャルスタッフィング、チケットスピン攻撃などのプロセスを自動化することで、ヘッドレスブラウザを介してサーバーリソースにアクセスできます。

ボットプロファイルのデバイスフィンガープリント (DFP) 検出技術が、ヘッドレスボットと Web ドライバーボットを検出するインテリジェンスによって強化されました。ヘッドレスブラウザボットトラフィックを軽減するには、[ヘッドレスブラウザ検出] オプションと [デバイスフィンガープリント検出] 機能を有効にする必要があります。

[NSBOT-747]

NetScaler Web App Firewall

JSON コマンドインジェクション攻撃に対するきめ細かい緩和　NetScaler アプライアンスでは、JSON コマンドインジェクション攻撃に対してきめ細かい緩和を構成できるようになりました。

[NSWAF-8511]

JSON クロスサイトスクリプティング攻撃に対するきめ細かい緩和　NetScaler アプライアンスでは、JSON クロスサイトスクリプティング攻撃に対してきめ細かい緩和を構成できるようになりました。

[NSWAF-8510]

JSON SQL インジェクション攻撃に対するきめ細かい緩和　NetScaler アプライアンスでは、JSON SQL インジェクション攻撃に対してきめ細かい緩和を構成できるようになりました。

[NSWAF-8509]

負荷分散

強化された望ましい状態 **API** エラーメッセージ　サービスグループメンバーの IP アドレスが、CS 仮想サーバーなどの他の NetScaler エンティティにすでに関連付けられている場合に表示されるエラーメッセージが拡張されます。失敗の理由がエラーメッセージで明らかになりました。以前は、エラーメッセージが失敗した理由は不明でした。

[NSLB-9005]

Desired State API は、既存のサーバ **IP** アドレスと名前の再利用をサポート　Desired State API では、サービスグループメンバーの IP アドレスが既存のサーバと一致する場合でも、サービスグループメンバーをサービスグループにバインドできるようになりました。既存のサーバの IP アドレスと名前は、サービスグループメンバーのバインド中に再利用されます。

以前は、IP アドレスが一致すると、サービスグループメンバーをサービスグループにバインドできませんでした。

[NSLB-9004]

ネットワーク

拡張 **ACL** の **IPv4** データセットにおける **CIDR** ベースのバインディングのサポート　拡張 ACL は、CIDR 表記で指定された IPv4 アドレス範囲を含む IPv4 データセットをサポートするようになりました。

[NSNET-24452]

DPDK モードでの **NetScaler BLX** アプライアンスのソフトウェア受信側スケーリングのサポート　DPDK モードの NetScaler BLX アプライアンスは、より多くのパケットエンジンで構成されており、送信（Tx）および受信（Rx）キューの数が少ない NIC ポートをサポートしません。

DPDK モードの NetScaler BLX アプライアンスは、次の両方の条件が満たされる場合、NIC ポートを使用しません。

- アプライアンスには、限られた数の送信キュー（Tx）と受信キュー（Rx）をサポートする NIC ポートがあります。たとえば、7 と入力します。
- アプライアンスは、より多くのパケットエンジンを使用して構成されています。たとえば、28 と入力します。

この問題を解決するために、ビルド 13.1 21.x 以降、NetScaler BLX アプライアンスはソフトウェア受信側スケーリング（RSS）を使用して、受信したパケットを複数のパケットエンジンにわたって NIC ポートで効率的に分散します。

ソフトウェア RSS モジュールは、各 NIC ポートに Rx と Tx の論理キューペアを割り当てます。その後、キューペアはパケットエンジン PE-0 にマッピングされます。

NIC ポートの Rx キュー内のパケットごとに、PE-0 は RSS ハッシュアルゴリズムを使用してパケットエンジンを選択します。次に PE-0 は、選択したパケットエンジンにパケットを送信して処理します。パケットの処理が完了すると、PE-0 は NIC ポートの Tx キューにパケットを送信します。

[NSNET-23133]

NetScaler GUI、**NetScaler CLI**、または **NetScaler NITRO API** を使用して、内部 **HTTP GUI** サービスを構成します　NetScaler アプライアンスでは `/etc/httpd.conf` 、NetScaler GUI への接続を管理する内部 HTTP GUI サービスの構成ファイルです。

`httpd.conf`ファイルを使用して内部 HTTP GUI サービスを構成する代わりに、NetScaler GUI、NetScaler CLI、または NetScaler NITRO API を使用できるようになりました。たとえば、NetScaler CLI を使用して、内部 HTTP サービスに一度に接続できるクライアントの最大数を変更できます。

内部 HTTP GUI サービスの名前形式は次のとおりです。`nshttpd-gui-<loop back IP address >-80`

NetScaler サービスコマンド操作を使用して、内部 HTTP GUI サービスを構成します。

[NSNET-20350]

プラットフォーム

NetScaler MPX 9100 プラットフォームのサポート　このリリースでは、NetScaler MPX 9100 プラットフォームがサポートされています。MPX 9110、MPX 9120、および MPX9130 モデルが含まれています。詳細については、「NetScaler MPX 9100」を参照してください。

[NSPLAT-23308]

NetScaler SDX 9100 プラットフォームのサポート　このリリースでは、NetScaler SDX 9100 プラットフォームがサポートされています。SDX 9120 モデルと SDX 9130 モデルが含まれています。詳細については、「NetScaler SDX 9100」を参照してください。

[NSPLAT-23299]

AWS および **GCP** クラウドで **SSL-TPS** のパフォーマンスを向上させる　パケットエンジン (PE) の重みを均等に分散することで、AWS と GCP クラウドで SSL-TPS のパフォーマンスを向上させることができます。これを行うには、NetScaler CLI で次のコマンドを実行して PE モードを設定します。

```
set cpuparam pemode [CPUBOUND | Default]
```

Azure クラウドでは、PE の重みは既定で均等に分散されます。この機能によって Azure インスタンスのパフォーマンスは向上しません。

[NSPLAT-22570]

NetScaler VPX インスタンスでの **VMware ESXi 7.0** アップデート **3c** のサポート　NetScaler VPX インスタンスは、VMware ESXi バージョン 7.0 アップデート 3c（ビルド 19193900）をサポートするようになりました。

[NSPLAT-22468]

SSL

NetScaler プラットフォームでの **SSL** チップ使用率の詳細を表示する　リリース 13.1 ビルド 21.x から、Intel Coleto チップおよび MPX 9100 (Lewisburg) プラットフォームに付属する MPX および SDX プラットフォームでの SSL チップ使用率に関する詳細を表示するカウンタが追加されました。サポートされていないプラットフォームでは、これらのカウンタの値は 0.0 と表示されます。

詳細については、Intel Coleto および Lewisberg SSL チップベースのプラットフォームのサポートを参照してください。

[NSSSL-10996]

DTLS を使用した **ECDSA** 証明書と暗号のサポート　ECDSA 証明書と暗号は、仮想サーバーやサービスなどの DTLS エンティティで使用できるようになりました。

[NSSSL-9535]

システム

TCP オプションヘッダーでのクライアント詳細の送信に関する拡張機能

- NetScaler アプライアンスは、最初のデータパケットに加えて、3 ウェイハンドシェイクの最後の ACK パケットにクライアント IP アドレスを挿入するようになりました。以前は、アプライアンスは最初のデータパケットでのみクライアント IP アドレスを送信していました。
- NetScaler アプライアンスは、挿入モード構成の TCP オプションでクライアントポートの送信をサポートするようになりました。この機能を有効または無効にするためのパラメータ`Send Client Port in Tcp Option`（`sendClientPortInTcpOption`）が TCP プロファイルに導入されました。

[NSBASE-15635]

解決された問題

ビルド 13.1—21.50 で解決されている問題。

認証、承認、監査

SAML 構成で使用されている SSL 証明書とキーのペアの更新中にエラーが発生すると、NetScaler アプライアンスがクラッシュすることがあります。この問題を解決するには、証明書のバインドを解除し、証明書を更新してから再度バインドします。

[NSHELP-30270]

SAML を使用したログイン要求に「」（一重引用符）以外の空白文字が含まれている場合、ユーザーは NetScaler アプライアンスにログインできません。今回の修正により、すべての空白文字が許可されるようになりました。

[NSHELP-29773]

KCD SSO の一部である委任されたユーザーに AS_REQ 要求を送信する際、NetScaler アプライアンスは、ドメインコントローラー（DC）がすべての暗号化タイプを公開するときに、次の優先度を持つ暗号化タイプを選択します。

1. ETYPE_ARCFOUR_HMAC_MD5
2. ETYPE_AES128_CTS_HMAC_SHA1_96
3. ETYPE_AES256_CTS_HMAC_SHA1_96Instead of
4. ETYPE_AES256_CTS_HMAC_SHA1_96
5. ETYPE_AES128_CTS_HMAC_SHA1_96
6. ETYPE_ARCFOUR_HMAC_MD5

[NSHELP-28681]

認証、承認、および auditing.login.Password を使用すると、認証が失敗することがあります。

[NSHELP-28101]

次の両方の条件が満たされると、NetScaler アプライアンスがバックエンドサーバーと SSO ループに入り、メモリが蓄積される可能性があります。

- ADC アプライアンスは、バックエンドサーバーとのネゴシエーションおよび NTLM SSO 認証を実行します。
- バックエンドサーバーは両方の認証を実行できません。

[NSHELP-27757]

プライマリコントローラカードとセカンダリコントローラカード間でセッションとキー構成の同期が行われると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-26891]

NetScaler SDX アプライアンス

ファクトリパーティションに十分な領域がないためにクリーンインストールが失敗すると、誤ったメッセージが表示されます。

[NSHELP-30136]

[クラスタノードの追加（Add Cluster Node）] ページの [バックプレーン（backplane）] フィールドは、次のいずれかの条件が満たされない

- このノードグループは、レイヤ 3 クラスタ用にすでに存在しています。
- これはレイヤー 2 クラスタです。

[NSHELP-29701]

NetScaler Gateway

SAML と EPA が nFactor 認証の連続要素として設定されている場合、VPN クライアントユーザは正常にログアウトできません。今回の修正により、ユーザーは問題なくログアウトできるようになりました。

[NSHELP-30193]

NetScaler GSLB および SSL VPN のセットアップで、DTLS ICA 接続の処理中にメモリリークが観察されます。その結果、接続が切断され、メモリが増加します。

[NSHELP-30182]

PCoIP アプリとデスクトップをブラウザーから起動すると起動に失敗し、エラーメッセージ`VMware client missing`が表示されます。この問題は、`vmware-view` プロトコルが許可されたプロトコルのリストに追加されていないために発生します。

[NSHELP-30062]

macOS でアンチウイルスの最後のフルシステムスキャンを確認するための EPA スキャンが失敗します。

[NSHELP-29571]

バイナリレスポンスが有効になっていると、NetScaler Gateway VPN フルトンネルが期待どおりに機能しません。その結果、NSAAC Cookie が破損します。今回の修正により、バイナリレスポンスは以前の VPN プラグインで動作するようになりました。ただし、JSON 応答と互換性のある最新の VPN プラグインを使用することをお勧めします。

[NSHELP-28729]

負荷分散

パーティション化された NetScaler アプライアンスは、追加のヘッダー（EDNS）を含む DNS 要求パケットの処理中にコアをダンプすることがあります。

[NSHELP-30796]

Autoscale DNS デプロイメントでは、TROFS 状態のメンバーはヘルスチェックの失敗を検出して応答しません。

[NSHELP-29628]

次の条件が満たされると、書き換えポリシーを負荷分散仮想サーバーにバインドしているときに、NetScaler アプライアンスがクラッシュすることがあります。

1. 2 つ目の式を評価すると、進行中の 1 つ目の式のポリシー状態変数が上書きされます。
2. DETERMINE_SERVICES ポリシーの状態変数は、負荷分散仮想サーバーによって定義されたルールによって上書きされます。

[NSHELP-29449]

show service コマンドを実行したときに表示されるモニタの応答時間が正しくない場合があります。

[NSHELP-28994]

SMPP 再試行メッセージは、要求が成功した場合でもクラスタ内のすべてのノードに送信されます。このシナリオでは、NetScaler アプライアンスのメモリ消費量が高くなります。

[NSHELP-28332]

ネットワーク

NetScaler BLX アプライアンスをリリース 13.1 ビルド 17.x にアップグレードすると、アプライアンスが起動しないことがあります。

[NSNET-25002]

`jsonschema` ホストに python モジュールがない場合、RHEL ベースの Linux ホストへの NetScaler BLX アプライアンスのインストールは失敗します。

[NSNET-24638]

NetScaler BLX アプライアンスを DPDK でアップグレードすると、次の条件がすべて満たされると失敗します。

- NetScaler BLX アプライアンスは Debian ベースの Linux ホストで実行されています
- NetScaler リリース 13.0 ビルド 82.x 以前からリリース 13.1 ビルド 17.x へのアップグレードが行われます。

[NSNET-24622]

ポート設定で TCP ACL ルールを設定した後に ICMP ACL ルールを設定すると、次の問題が発生する場合があります。

- NetScaler アプライアンスは、TCP ACL の同じポート設定を ICMP ACL にも誤って追加します。

[NSHELP-31114]

次の条件が満たされると、GUI を使用して INAT ルール内のプライベート IP アドレスを変更すると失敗します。

- INAT ルールでは接続フェールオーバーが有効になっています。

[NSHELP-30792]

NetScaler アプライアンスのシリアルコンソールで、VTYSH プロンプトまたはシェルプロンプトに出力が表示されないことがあります。

[NSHELP-30446]

IP セットが既にバインドされているネットプロファイルを変更すると、次のエラーで失敗することがあります。

- `IP set is already bound to the network profile`

[NSHELP-29363]

大規模な NAT44 セットアップでは、次の理由により、SIP トラフィックの受信中に NetScaler アプライアンスがクラッシュすることがあります。

- アプライアンスの LSN モジュールの参照カウントのフィルタリングとマッピングは 0 以外です。

[NSHELP-28842]

プラットフォーム

仮想マシンが起動の初期段階にある場合、Azure クラウドでホストされている NetScaler VPX インスタンスのシリアルコンソールにはアクセスできません。

[NSPLAT-23010]

NetScaler VPX HA フェイルオーバー中に、IP セットを IP アドレスにバインドせずに IPset を構成すると、AWS クラウドでの Elastic IP アドレスの移動が失敗します。

[NSHELP-29425]

SSL

RC4 暗号スイートは、`Illegal parameter error` メッセージ付きの SSL ハンドシェイク中に失敗します。

[NSSSL-11463]

SSL インターセプトが有効になっていて、期限切れの証明書を使用してバックエンドサーバーにアクセスする複数の並列要求がある場合、NetScaler アプライアンスがクラッシュします。

[NSHELP-29520]

クラスタのセットアップでは、次の問題が発生する可能性があります。

- CLIP の SSL 内部サービスにバインドするデフォルトの証明書とキーのペアに対するコマンドがありません。ただし、古いビルドからアップグレードする場合は、デフォルトの証明書とキーのペアを、CLIP 上の影響を受ける SSL 内部サービスにバインドする必要があります。

- 内部サービスに対するデフォルトの set コマンドの CLIP とノード間の設定の不一致。
- ノードで実行される show running config コマンドの出力で、SSL エンティティに対するデフォルトの暗号バインドコマンドがありません。省略は表示上の問題にすぎず、機能への影響はありません。このバインディングは show ssl <entity> <name> コマンドを使用して表示できます。

[NSHELP-25764]

システム

次のいずれかの状況が発生すると、NetScaler アプライアンスがクラッシュします。

- syslog アクションはドメイン名で設定され、GUI または CLI を使用して設定をクリアします。
- 高可用性同期はセカンダリノードで行われます。[NSHELP-30987、NSHELP-28121、NSHELP-29843]

NetScaler アプライアンスから転送されるすべてのデータパケットには、構成された TTL 値ではなく、クライアントまたはサーバーから送信された値が含まれます。

[NSHELP-30683]

NetScaler アプライアンスは、HTTP 以外のデータパケットの一部をバックエンドサーバーに転送できません。

[NSHELP-30192]

特定のシナリオでは、次の条件が満たされた場合、NetScaler アプライアンスは一部の HTTP パケットをバックエンドサーバーに転送しません。

- NetScaler 機能が内部で HTTP パケットのクローンを作成する場合。

[NSHELP-29958]

NetScaler アプライアンスが、IPv6 トランザクションに関連する AppFlow レコードに IPv4 アドレスを誤って追加することがあります。

[NSHELP-29261]

ICAP モジュールからクライアントへのチャンク応答を再生すると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-28788]

Pitboss 障害は、再送信キューに大量のパケットをループさせると発生します。

[NSHELP-26071]

TCP プロトコルを使用して外部 SYSLOG サーバにログインすると、一部の SYSLOG メッセージがドロップされます。

[NSHELP-24522]

特定のシナリオでは、IP アドレスベースのフィルタを適用すると、nstrace パケットキャプチャですべてのパケットが失われます。

[NSHELP-23483]

ユーザーインターフェイス

NetScaler GUI でキャッシュフィルタリングが期待どおりに機能しないことがあります。

[NSHELP-30392]

NetScaler アプライアンスが外部認証サーバーを使用するように構成されている場合、グローバルに無効に設定されている rbaonResponse パラメーターに関係なく、stat コマンドの実行が遅れることがあります。パラメータは GUI または CLI から無効にできます。

[NSHELP-30289]

NetScaler GUI は RAPI 呼び出しを処理しないため、GUI の一部のコンポーネントが応答しなくなります。

[NSHELP-30231]

場合によっては、NetScaler GUI の［SSL キー］タブから SSL キーを読み込めないことがあります。

[NSHELP-28870]

フィルター付きの NITRO GET リクエストに対する API レスポンスには、フィルターに記載されていない場合でも追加情報が含まれている場合があります。

[NSHELP-28598]

管理パーティションの設定で、証明書失効リスト (CRL) ファイルのアップロードと追加が失敗する。

[NSHELP-20988]

既知の問題

リリース 13.1 ～21.50 に存在する問題。

AppFlow

HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

[NSINSIGHT-943]

認証、承認、監査

NetScaler アプライアンスは、重複したパスワードログインの試行を認証せず、アカウントのロックアウトを防ぎます。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema が、NetScaler GUI のログインスキーマエディタ画面に正しく表示されません。

[NSAUTH-6106]

ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
`show adfsproxyprofile <profile name>`

回避方法:

クラスタ内のプライマリのアクティブな NetScaler に接続し、`show adfsproxyprofile <profile name>` コマンドを実行します。プロキシプロファイルの状態が表示されます。

[NSAUTH-5916]

次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

- [LDAP 到達可能性をテスト] オプションが開きます。
- 無効なログイン認証情報が入力され、送信されます。
- 有効なログイン認証情報が入力され、送信されます。

回避方法:

[LDAP 到達可能性のテスト] オプションを閉じて開きます。

[NSAUTH-2147]

キャッシュ

統合キャッシュ機能が有効で、アプライアンスのメモリが不足している場合、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-22942]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、CLAG が Mellanox NIC 上に作成されている場合、VPX インスタンスの再起動時に CLAG MAC が変更されます。VPX インスタンスへのトラフィックは再起動後に停止します。これは、MAC テーブルに古い CLAG MAC エントリがあるためです。

[NSSVM-4333]

Mellanox NIC を搭載した NetScaler SDX アプライアンスで、Mellanox NIC を持つ VPX インスタンスのスループットを変更すると、VPX インスタンスが再起動します。

[NSHELP-31305]

NetScaler SDX アプライアンスをリリース 13.1 ビルド 21.50 以降にアップグレードすると、SSL 復号化と MAC 比較が失敗することがあります。その結果、SSL ハンドシェイクの失敗、VPX ステータスのフラッピング、VPX インスタンス GUI の利用不能、仮想サーバーとアプリケーションのダウンが発生する可能性があります。

注：この問題は、SDX 8900、SDX 15000、SDX 15000-50G、SDX 26000、および SDX 26000-50S のプラットフォームで発生します。

[NSHELP-31672]

NetScaler Gateway

場合によっては、ポート 53 を使用する一部の非 DNS プロトコル（STUN など）の問題が原因で、macOS 向けの Citrix Secure Access が接続を切断することがあります。

[NSHELP-31004]

Always on が設定されている場合、aoservice.exe ファイルのバージョン番号（1.1.1.1）が正しくないため、ユーザトンネルが失敗します。

[NSHELP-30662]

[NetworkAccessonVPNFailure］プロファイルパラメーターを［フルアクセス］から［OnlyToGateway］に変更すると、ユーザーは NetScaler Gateway アプライアンスに接続できなくなります。

[NSHELP-30236]

Windows VPN クライアントは、サーバからの「SSL close notify」アラートを尊重せず、同じ接続で転送ログイン要求を送信します。

[NSHELP-29675]

サーバー証明書が信頼されていると、サーバー検証コードが失敗することがあります。その結果、エンドユーザーはゲートウェイにアクセスできなくなります。

[NSHELP-28942]

macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

[NSHELP-28551]

クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

[NSHELP-28404]

GUI を使用してクラシック認可ポリシーをバインド解除することはできません。ただし、CLI を使用して認証、承認、および監査認可ポリシーのバインドを解除することはできます。

今回の修正により、GUI を使用して承認ポリシーのバインドを解除できるようになりました。

[NSHELP-27064]

高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

- HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

回避方法:

HA 同期が完了した後（両方のノードが同期成功状態にある）のみ、手動の HA フェールオーバーを連続して実行します。

[NSHELP-25598]

Gateway Insight は、VPN ユーザに関する正確な情報を表示しません。

[NSHELP-23937]

次の条件を満たす場合、Windows ログオン後に VPN プラグインはトンネルを確立しません。

- NetScaler Gateway アプライアンスが常時オン機能用に構成されている
- アプライアンスは 2 要素認証`off`による証明書ベースの認証用に設定されています。

[NSHELP-23584]

スキーマを参照しているときに、エラーメッセージ`Cannot read property 'type'of undefined`が表示されることがあります。

[NSHELP-21897]

Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

[CGOP-19355]

無効な STA チケットによるアプリケーションの起動失敗は、Gateway Insight で報告されません。

[CGOP-13621]

Gateway Insight レポートでは、SAML エラーエラーの [認証タイプ] フィールドの`SAML`ではなく、値`Local`が誤って表示されます。

[CGOP-13584]

高可用性セットアップでは、NetScaler フェイルオーバー時に、NetScaler Console のフェイルオーバー数の代わりにストレージリポジトリ数が増加します。

[CGOP-13511]

ICA 接続を MAC レシーバーバージョン 19.6.0.32 または Citrix Virtual Apps and Desktops バージョン 7.18 から起動すると、HDX Insight 機能は無効になります。

[CGOP-13494]

EDT Insight 機能を有効にすると、ネットワークの不一致時にオーディオチャンネルが失敗することがあります。

[CGOP-13493]

ブラウザからローカルホスト接続を受け付けると、macOS の Accept Connection ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

[CGOP-13050]

一部の言語では、Citrix SSO アプリ > ホームページのテキストHome Pageが切り捨てられます。

[CGOP-13049]

NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

[CGOP-11830]

Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

[CGOP-7269]

負荷分散

高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

[NSLB-7679]

entityofs サービスグループのトラップの ServiceGroupName 形式は次のとおりです。
<service(group)name>?<ip/DBS>?<port>

トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (?) はセパレータとして使用されます。NetScaler は、疑問符（?）付きのトラップを送信します。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

[NSHELP-28080]

その他

高可用性セットアップで強制同期が行われると、アプライアンスはセカンダリノードでset urlfiltering parameterコマンドを実行します。

その結果、セカンダリノードは、TimeOfDayToUpdateDB パラメータで指定された次のスケジュールされた時刻まで、スケジュールされた更新をスキップします。

[NSSWG-849]

URL フィルタリングのサードパーティベンダーで接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

[NSHELP-22409]

ネットワーク

DPDK をサポートする NetScaler BLX アプライアンスでは、DPDKIntel i350 NIC ポートではタグ付き VLAN はサポートされません。これは、DPDK ドライバに存在する既知の問題であるため、確認されています。

[NSNET-25299]

DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

- NetScaler BLX アプライアンスは、hugepagesの小さい数が割り当てられます。たとえば、1G です。
- NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

この問題は、/var/log/ns.logにエラーメッセージとして記録されます。

- `BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x`

注:x はワーカープロセス数以下の数です。

回避方法:

多数のhugepagesを割り当て、アプライアンスを再起動します。

[NSNET-25173]

DPDK を搭載した NetScaler BLX アプライアンスは、次の条件が満たされると再起動に失敗することがあります。

- NetScaler BLX アプライアンスにはhugepagesの大きい数が割り当てられます。たとえば、16 GB と入力します。

この問題は、/var/log/ns.logにエラーメッセージとして記録されます。

- `EAL: rte_mem_virt2phy(): cannot open /proc/self/pagemap: Too many open files`

回避方法:

この問題を解決するには、次のいずれかの回避策を使用します。

- ulimitコマンドを使用するか、limits.confファイルを編集して、Linux ホストで開くことができるファイルの上限を増やします。
- 割り当てられたhugepagesの数を減らします。

[NSNET-24727]

DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

[NSNET-24449]

DPDK を搭載した NetScaler BLX アプライアンスの Intel X710 10G (i40e)インターフェイスでは、次のインターフェイス操作はサポートされていません:

- 無効化
- 有効化
- リセット

[NSNET-16559]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

回避方法:

NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します。

- dpkg –add-architecture i386
- apt-get update
- apt-get dist-upgrade
- apt-get install libc6:i386

[NSNET-14602]

FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

[NSNET-5233]

NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリの制限は、管理パーティションの新しいメモリ制限に自動的に設定されます。

[NSHELP-21082]

プラットフォーム

高可用性フェイルオーバーは AWS および GCP クラウドでは機能しません。AWS および GCP クラウド、および NetScaler VPX オンプレミスでは、管理 CPU が 100% の容量に達する可能性があります。これらの問題はいずれも、次の条件が満たされた場合に発生します。

1. NetScaler アプライアンスの初回起動時には、プロンプトが表示されたパスワードは保存されません。
2. その後、NetScaler アプライアンスを再起動します。

[NSPLAT-22013]

13.0/12.1/11.1 ビルドから 13.1 ビルドにアップグレードするか、13.1 ビルドから 13.0/12.1/11.1 ビルドにダウングレードすると、一部の Python パッケージが NetScaler アプライアンスにインストールされません。この問題は、次の NetScaler バージョンで修正されています。

- 13.1-4.x
- 13.0—82.31 およびそれ以降
- 12.1—62.21 およびそれ以降

NetScaler バージョンを 13.1-4.x から次のバージョンのいずれかにダウングレードすると、Python パッケージはインストールされません。

- 任意の 11.1 ビルド
- 12.1—62.21 およびそれ以前
- 13.0-81.x およびそれ以前

[NSPLAT-21691]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、2 番目のノードと CLIP で CLAG MAC の不一致があります。

- CLAG は Mellanox NIC 上に作成されます。
- クラスターと CLAG セットアップに別の VPX インスタンスを追加します。

その結果、VPX インスタンスへのトラフィックが停止します。

[NSPLAT-21049]

NetScaler SDX アプライアンスでのクラスター設定では、次の条件が満たされると、CLIP テーブルと MAC テーブルで MAC アドレスが一致しないため、最初のノードがダウンします。

- CLAG は Mellanox NIC 上に作成されます。
- クラスタから 2 つ目のノードを削除します。

[NSPLAT-21042]

Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。`rm cloudprofile` コマンドを使用して、プロファイルを削除します。

[NSPLAT-4520]

Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノードで設定する必要があります。

[NSPLAT-4451]

NetScaler リリース 13.1 以降、NetScaler アプライアンスは、8 つを超える VMXNET3 ネットワークインターフェイスを備えた ESXi ハイパーバイザーで起動に失敗します。

[NSHELP-31266]

ポリシー

処理データのサイズが、設定されたデフォルトの TCP バッファサイズを超えると、接続がハングすることがあります。回避策：TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

[NSPOLICY-1267]

一部のシナリオでは、AppExpert 変数のクリア操作で割り当てアクションを使用すると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-29766]

SSL

NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

回避方法：

1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例：`set ssl vserver <name> -SSL3 DISABLED`。
2. 構成を保存します。

[NSSSL-9572]

認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

[NSSSL-6478]

同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

[NSSSL-6213]

HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。エラー:CRL 更新は無効です

[NSSSL-6106]

セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

[NSSSL-4427]

SSL プロファイルの SSL プロトコルまたは暗号を変更しようとすると、不正な警告メッセージ`Warning: No usable ciphers configured on the SSL vserver/service,`が表示されます。

[NSSSL-4001]

期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

[NSSSL-3184, NSSSL-1379, NSSSL-1394]

MPX 8900 および MPX 15000 FIPS 認定アプライアンスでは、ECDHE トラフィックを実行するとメモリリークが発生する可能性があります。

[NSHELP-30744]

rc.netscaler ファイルの一部であるカスタマイズは、システムの初期化中に実行されないため、適用されません。

[NSHELP-31914]

システム

アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

[NSHELP-21240]

mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

[NSHELP-10972]

クラスタ展開では、CCO 以外のノードで`force cluster sync`コマンドを実行すると、ns.log ファイルに重複するログエントリが含まれます。[NSBASE-16304、NSGI-1293]

NetScaler Console を Kubernetes クラスターにインストールすると、必要なプロセスが実行されない可能性があるため、期待どおりに機能しません。

回避策：管理ポッドを再起動します。

[NSBASE-15556]

LogStream トランスポートタイプが Insight 用に構成されている場合、HDX Insight SkipFlow レコードでは、クライアント IP とサーバー IP が反転します。

[NSBASE-8506]

NetScaler アプライアンスは、ヘッダー名フィールドにドット（”.”）の付いたカスタム HTTP ヘッダーを含むパケットをドロップします。このアクションは、allowOnlyWordCharactersAndHyphen パラメータがデフォルトの HTTP プロファイルでデフォルトで有効になっているために発生します。

回避策: デフォルトの HTTP プロファイルでallowOnlyWordCharactersAndHyphenを無効にします。ただし、Citrix では有効にしておくことをお勧めします。

[NSBASE-16722]

ユーザーインターフェイス

MQTT リライト機能では、GUI のエクスプレッションエディタを使用してエクスプレッションを削除することはできません。

回避方法:

CLI から MQTT タイプの add または edit action コマンドを使用します。

[NSUI-18049]

NetScaler GUI では、Dashboardタブの下にあるHelpリンクが壊れています。

[NSUI-14752]

CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

回避方法:

NetScaler GUI または CLI を使用して、IPsec プロファイル、IP トンネル、および PBR ルールを追加して、CloudBridge Connector を構成します。

[NSUI-13024]

GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

[NSUI-6838]

NetScaler BLX アプライアンスの高可用性セットアップでは、プライマリノードが応答しなくなり、CLI または API 要求がブロックされることがあります。

回避方法:

プライマリノードを再起動します。

[NSCONFIG-6601]

あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

1. NetScaler アプライアンスをいずれかのビルドにアップグレードします。

- 13.0 52.24 ビルド
- 12.1 57.18 ビルド
- 11.1 65.10 ビルド

1. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
2. NetScaler アプライアンスを古いビルドにダウングレードします。

CLI を使用してこれらのシステムユーザーの一覧を表示するには、
コマンドプロンプトで次のように入力します。

```
query ns config -changedpassword [-config <full path of the configuration
 file (ns.conf)>]
```

回避方法:

この問題を修正するには、次の独立したオプションのいずれかを使用します。

- NetScaler アプライアンスがまだダウングレードされていない場合（前述の手順の手順 3)、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
- アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
- 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳細については、「ルート管理者 (nsroot) パスワードをリセットする方法」を参照してください。

[NSCONFIG-3188]

NetScaler 13.1—17.42 リリースのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリース Build 13.1—17.42 に存在する機能強化と変更、修正された問題と既知の問題について説明します。

メモ

このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正と勧告のリストについては、セキュリティ速報を参照してください。

新機能

ビルド 13.1 〜17.42 で利用できる機能強化と変更。

ボット管理

IPv6 アドレッシングのサポート　NetScaler ボット管理では、ボット検出技術でインターネットプロトコルバージョン 6（IPv6）アドレス指定がサポートされるようになりました。

[NSBOT-690]

NetScaler Gateway

NetScaler Gateway を介した **EDT** の **DF** ビット伝播　NetScaler Gateway アプライアンスは、EDT パスの最大伝送ユニット検出（PMTUD）機能に対して DF ビットの強制をサポートするようになりました。パス MTU ディスカバリ機能は、EDT セッションの確立時に最大伝送ユニット（MTU）を動的に決定するのに役立ちます。DF ビット強制により、パフォーマンスの低下やセッションの確立の失敗につながる可能性がある EDT フラグメンテーションが防止されます。

以前のリリースでは、NetScaler Gateway は EDT パス MTUD をサポートしていましたが、DF ビットの強制はサポートしていませんでした。

[CGOP-18438]

NetScaler Web App Firewall

複数のクロスサイトスクリプティング **(XSS)** 違反を学習するためのサポートの強化　NetScaler Web App Firewall の学習プロセスが強化され、クロスサイトスクリプティング攻撃の誤検出が減少しました。

学習を有効にすると、リクエスト内のすべての違反を学習し、すべてのタグ、属性、パターンに一度に緩和を適用できます。以前は、一度に 1 つの違反しか報告できず、複数の違反に対してこのプロセスを繰り返す必要がありました。

たとえば、ペイロードに 15 個のカスタムタグがあり、それぞれ違反が発生した場合、最初の違反に対して緩和を適用し、リクエストを実行して別のカスタムタグに違反のフラグを立てることができます。すべてのカスタムタグに 1 つずつ緩和を適用するには、このプロセスを繰り返す必要があります。

[NSWAF-7545]

負荷分散

LB および **GSLB Autoscale** サービスグループのメンバーを有効または無効にするオプション　LB または GSLB (DNS ベース) の Autoscale サービスグループの特定のメンバーを直接有効または無効にできるようになりました。そのため、LB または GSLB (DNS ベース) の Autoscale サービスグループの管理が容易になりました。

以前は、個々のメンバーを有効または無効にするには、LB または GSLB Autoscale サービスグループ全体を有効または無効にする必要がありました。個々のメンバーを有効または無効にするオプションがあるのは、Autoscale 以外のサービスグループだけでした。

[NSLB-8109]

ネットワーク

ISSU 統計情報の機能拡張　ISSU 統計情報に次の 2 つの拡張機能が追加されました。

- 古いプライマリノードが現在処理している既存の接続の一覧を表示するオプション dumpsession (Dump Session) がshow migrationオペレーションに追加されました。dumpsession オプションを指定した show migration オペレーションは、新しいプライマリノードでのみ実行する必要があります。
- show migration 操作（オプションなし）では、ISSU 移行操作に関連する次の追加情報が表示されます。
- ISSU 移行操作の一部として処理される接続の総数
- ISSU 移行操作の一部として処理されている残りの接続数

詳細については、https://docs.citrix.com/en-us/citrix-adc/13-1/upgrade-downgrade-citrix-adc-appliance/issu-high-availability.htmlを参照してください。

[NSNET-23577]

SNMP を使用して、**NetScaler** アプライアンスのバックエンド接続のポート使用量を監視する　PORT-ALLOC-EXCEED SNMP アラームを使用して、NetScaler アプライアンスのバックエンド接続のポート使用率を監視できます。

PORT-ALLOC-EXCEED SNMP アラームには、NetScaler が所有する IP アドレスに割り当てられたポートの合計数をパーセンテージで指定するhigh-thresholdおよびnormal-thresholdパラメータが含まれます。たとえば、high-threshold パラメータが 90 に設定されている場合、NetScaler アプライアンスは次のイベントが発生するとトラップメッセージを生成して送信します。

- NetScaler が所有するバックエンド接続の IP アドレスのいずれかで、ポート割り当ての割合が 90%を超える場合

SNMP アラートは、使用可能な空きポートがほぼ使い果たされそうになった場合に、NetScaler が所有する IP アドレスを増やす必要があるかどうかを判断するのに役立ちます。

[NSNET-21719]

GENEVE プロトコルのサポート　NetScaler アプライアンスは、RFC 8926 で定義されている汎用ネットワーク仮想化カプセル化（GENEVE）プロトコルをサポートするようになりました。

サーバ仮想化とクラウドコンピューティングアーキテクチャにより、データセンターにおける分離されたレイヤ 2 ネットワークの需要が高まっています。VLAN 制限の 4094 は不十分であることが判明しており、この制限を克服するために VXLAN や NVGRE などのカプセル化プロトコルが導入されました。

これらのプロトコルは、主にコントロールプレーンの実装が異なります。GENEVE プロトコルでは、コントロールプレーンの仕様は定義されていません。このプロトコルは、コントロールプレーンの仕様を定義するために実装に任されています。

GENEVE プロトコルは、レイヤ 2 フレームを UDP パケットにカプセル化することにより、レイヤ 3 インフラストラクチャ上にレイヤ 2 オーバーレイネットワークを構築することを目的としたカプセル化テクノロジです。各 VLAN は、VNID と呼ばれる一意の 24 ビット識別子によって識別されます。同じセグメント ID (VNID) 内でのみ相互に通信できます。

NetScaler アプライアンスは、UDP ポート 6081 での GENEVE カプセル化をサポートしています。

[NSNET-21717]

NetScaler BLX アプライアンスを専用モードで実行している **Linux** ホストへの **SSH** アクセスを構成します　デフォルトでは、NetScaler BLX アプライアンスを専用モードで実行している Linux ホストへの SSH アクセスは、アプライアンスの専用インターフェイスを介して行うことはできません。

NetScaler BLX アプライアンスの専用インターフェイスを介して、Linux ホストへの SSH アクセスを構成できます。この機能は、NetScaler BLX アプライアンスを専用モードで実行している単一インターフェイスの Linux ホストで役立ちます。

Linux ホストへの直接 SSH アクセスは、次のいずれかのタイプで設定できます。

- NetScaler BLX アプライアンスの NetScaler IP（NSIP）のポート 9022 で SSH アクセスを提供します。- <NetScaler IP address (NSIP)>:9022
- NetScaler IP（NSIP）のサブネットに新しい IP アドレスを定義し、ポート 22 で SSH アクセスを提供します。- <**new** IP address on the NetScaler IP address (NSIP)subnet>:22
  また、Linux ホストの他のすべてのポートには、新しい IP アドレスを使用して到達できます。たとえば、Linux ホストでポート 514/UDP で実行されている`rsyslog`サーバは、新しい IP アドレスのポート 514 で到達可能になりました。

[NSNET-21586]

DPDK ポートを備えた **NetScaler BLX** アプライアンスの導入を簡素化　DPDK ポートを備えた NetScaler BLX アプライアンスを展開する手順が簡素化され、次の機能が強化されました。

- NetScaler BLX アプライアンスは、DPDK バージョン 20.11.1 でコンパイルされたライブラリを使用するようになりました。アプライアンスは DPDK VFIO カーネルモジュールを Linux ホストに自動的にロードします。

- dpdk-configパラメーターは、NetScaler BLX 構成 (blx.conf) ファイルから削除されました。既存のworker-processesパラメータは、DPDK ポートを持つ NetScaler BLX アプライアンスにも適用されるようになりました。worker-processesは、NetScaler BLX アプライアンスのパケットエンジンの数を指定します。つまり、worker-processes は、モード（共有、専用、または DPDK）に関係なく、NetScaler BLX アプライアンスの共通パラメータになりました。worker-process が設定されていない場合、NetScaler BLX アプライアンスはデフォルトで 1 つのパケットエンジンで構成されます。
- interfacesパラメータは、非 DPDK NIC ポートに加えて、DPDK 互換 NIC ポートを指定するようになりました。NetScaler BLX アプライアンスは、interfaces パラメータに指定されたポートのリストから、DPDK 互換の NIC ポート（存在する場合）を自動的に検出します。アプライアンスは、検出された DPDK 互換 NIC ポートを Linux ホスト上の DPDK VFIO モジュールにバインドします。NetScaler BLX アプライアンスを起動すると、DPDK および非 DPDK NIC ポートがアプライアンスの一部として自動的に追加されます。
- DPDK にバインドされた Mellanox NIC ポートを指定するdpdk-non-uio-intfパラメーターは、NetScaler BLX 構成（blx.conf）ファイルから削除されました。interfacesパラメータは、NetScaler BLX アプライアンスで DPDK ポートとして使用される Mellanox NIC ポートを指定するようになりました。NetScaler BLX アプライアンスの Mellanox NIC ポートを指定する前に、Linux ホストに Mellanox OFED DPDK ライブラリとカーネルモジュールをインストールする必要があります。NetScaler BLX アプライアンスは、指定された Mellanox NIC ポートを自動的に検出し、DPDK モードで初期化します。NetScaler BLX アプライアンスを起動すると、DPDK にバインドされた Mellanox NIC ポートがアプライアンスの一部として追加されます。
- Linux ホストで DPDK 用にhugepagesを設定するための新しいパラメータtotal-hugepage-mem が NetScaler BLX 構成（blx.conf）ファイルに導入されました。total-hugepage-memパラメータでは、hugepages サイズを MB または GB 単位で指定します (たとえば、1024 MB と 2 GB)。
- DPDK ポートを使用して NetScaler BLX アプライアンスをアップグレードすると、アップグレードモジュールは既存の構成を NetScaler BLX 構成（blx.conf）ファイルの新しい形式に自動的に変換します。

[NSNET-20524]

NetScaler アプライアンスで使用可能な空きポートを監視して、新しいバックエンド接続がないか確認する
NetScaler アプライアンスは、物理サーバーまたは他のピアデバイスとの通信に、Citrix が所有する IP アドレスをソース IP アドレスとして使用します。NetScaler アプライアンスは IP アドレスのプールを維持し、サーバーとの接続中に IP アドレスを動的に選択します。物理サーバが配置されているサブネットに応じて、アプライアンスは使用する IP アドレスを決定します。このアドレスプールは、トラフィックおよびモニタプローブの送信に使用されます。

新しいバックエンド接続で NetScaler が所有する IP アドレスで使用可能な空きポートの総数を表示できます。この情報は、使用可能な空きポートがほぼ使い果たされそうになった場合に、NetScaler が所有する IP アドレスを増やす必要があるかどうかを判断するのに役立ちます。

NetScaler アプライアンスに次の情報を入力して、新しいバックエンド接続に使用できる空きポートの総数を計算できます。

- Citrix が所有する IP アドレス（オプション）

- 宛先 IP アドレス
- Destination port
- TCP または非 TCP プロトコル

[NSNET-20410]

プラットフォーム

KVM ハイパーバイザーでの **NetScaler** アプライアンスの初回起動時の **NetScaler VPX** 構成のサポート　KVM ハイパーバイザーで NetScaler アプライアンスを初めて起動するときに、NetScaler VPX 構成を適用できるようになりました。したがって、VPX インスタンスでのお客様のセットアップは、はるかに短時間で構成できます。

[NSPLAT-21571]

バックアップ操作中に **NetScaler** 管理パーティションから **nstrace** フォルダーを除外する　管理パーティションを持つ NetScaler アプライアンスでは、nstrace フォルダーのバックアップ操作は除外されます。これにより、重要なデータを失うことなく、NetScaler の全体的なバックアップサイズが縮小されます。

[NSPLAT-21433]

ポリシー

ポリシーデータセットの **IPv4** および **IPv6** アドレスでの **CIDR** サブネット表記のサポート　IPv4 および IPv6 アドレスのポリシーデータセットで、バインドされた値を CIDR 表記 (a.b.c.d/n など) を使用したサブネットにできるようになりました。CIDR 表記法では、サブネットのアドレスと範囲を指定します。以前は、ポリシーデータセットにサブネットを追加するオプションはありませんでした。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/appexpert/pattern-sets-data-seta/configuring-data-sets.htmlを参照してください。

[NSPOLICY-3828]

SSL

NetScaler アプライアンスのフロントエンド **SSL** サービスで非セキュアプロトコルを無効にする　標準のセキュリティスキャンでは、NetScaler アプライアンスの起動時にデフォルトで作成されるフロントエンド SSL サービスで、セキュリティで保護されていないプロトコルに対してアラートがトリガーされる場合があります。このようなアラートを回避するために、アプライアンスの起動時にフロントエンド SSL サービスではこれらのプロトコルがデフォルトで無効になっています。非セキュアプロトコルの例としては、SSLv3、TLv1、および TLSv1.1 があります。

デフォルトの SSL プロファイルを有効にすると、これらのプロトコルが無効になった新しい SSL プロファイルが作成されます。この新しいプロファイルは、フロントエンド SSL サービス

(ns_default_ssl_profile_internal_frontend_service) にバインドされます。このプロファイルは編集可能です。

[NSSSL-9985]

RSASSA-PSS アルゴリズムを使用して署名された証明書のサポート　すべての NetScaler プラットフォームで、RSASSA-PSS アルゴリズムで署名された証明書がサポートされるようになりました。これらのアルゴリズムは X.509 証明書パスの検証でサポートされています。

[NSSSL-9289]

解決された問題

ビルド 13.1-17.42 で対処される問題。

認証、承認、監査

ADFSPIP URL が type`http://`に設定されている場合、NetScaler アプライアンスがクラッシュします。ADFSPIP は `https://` URL タイプのみをサポートします。

[NSHELP-29838]

要求処理に大幅な遅延がある場合、SAML IdP フロー中にアプライアンスがクラッシュすることがあります。

[NSHELP-29789]

`/logon/LogonPoint/Resources/List and /cgi/Resources/List`などのエンドポイントの書き換えポリシーはサポートされていません。

[NSHELP-29488]

まれに、ログの位置が正しくないために NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-29267]

OAuth サービスプロバイダーを使用して認証するように構成された NetScaler アプライアンスは、IDP TokenEndpoint で認証するように「client-secrete_post」で構成することはできません。

今回の修正により、IDP のトークンエンドポイントと通信するときに ADC の OAuth サービスプロバイダー機能に認証方法`client_secret_basic`が追加されました。

[NSHELP-28945]

SAML 認証が進行中で、サイズが 1800 バイト以上の X.509 証明書が SAML 認証に使用されると、アプライアンスが応答しないことがあります。

[NSHELP-28608]

有効期限が切れたときにユーザーパスワードが変更されると、マルチコア NetScaler アプライアンスで認証、承認、および auditing.user.Attribute 式で空の値が返されることがあります。

[NSHELP-28419]

NetScaler アプライアンスは、OAuth 証明書利用者として構成されている場合、ID トークンから抽出された「電子メール」および「ユーザー名」フィールド情報を、認証、承認、および監査セッションのハッシュ属性に追加しません。

[NSHELP-28262]

SAML メタデータが設定されると、SSL 証明書でメモリリークが観察されます。

[NSHELP-27846]

ユーザーが SAML ログアウトを実行しても、ログアウトはすぐには行われず、次のエラーメッセージが表示されます。

```
Unsupported mechanisms found in Assertion; Please contact your
administrator.
```

このエラーは、顧客が設定した IDP が異なる URL エンコード技術を使用して、応答内の署名アルゴリズムパラメータをエンコードしているために発生します。この修正により、複数の URL エンコード技術を使用した SAML レスポンスの署名アルゴリズムパラメーターのエンコードがサポートされるようになりました。

[NSHELP-27621]

nFactor が設定されていると、ログアウトメッセージに不正な IP アドレスが記録されることがあります。

[NSHELP-26692]

NetScaler アプライアンスは、次の両方の条件が満たされるとクラッシュします。

- 電子メール OTP が設定されている
- メールサーバーが応答しない、またはメールサーバーにネットワークの問題がある

[NSHELP-26137]

高可用性セットアップでは、強制同期が開始されると NetScaler アプライアンスがクラッシュします。

[NSAUTH-11876]

Intune NAC v2 は Android 11 以降ではサポートされていません。

[NSAUTH-11872]

パスワードに特定の特殊文字が含まれている場合や、引数にスペースが含まれている場合、管理者は LDAP または RADIUS 接続ツールを使用できません。

[NSAUTH-11322]

ボット管理

CAPTCHA チャレンジが進行中の場合、NetScaler ボット管理は、CAPTCHA の再試行に対してユーザーが設定した構成値を尊重しません。

[NSBOT-801]

CallHome

プールされたライセンスを使用している NetScaler MPX アプライアンスでは、CallHome の登録が失敗することがあります。CallHome が NetScaler サポートサーバーへのアプライアンスの登録に誤ったシリアル番号を使用しているため、登録は失敗します。

[NSHELP-28667]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスをバックアップから復元しても、CLI プロンプト文字列は復元されません。

[NSHELP-30238]

NetScaler SDX 115xx アプライアンスでは、アプライアンスのバックアップに 3 つ以上のインスタンスが含まれていると、CPU コア数が多い（3〜5 コア）割り当てられた VPX の復元が失敗することがあります。

[NSHELP-30135]

NetScaler SDX アプライアンスでは、`Hypervisor Disk Usage High` アラートでアラームを発生させるデフォルト値が 98% に増加します。

[NSHELP-29688]

インターフェイス速度値が 4 Gbps を超えると、整数のオーバーフローが原因で誤った値が返されます。

[NSHELP-29658]

まれに、NetScaler SDX アプライアンスで ADC インベントリが発生しないことがあります。

[NSHELP-29607]

NetScaler SDX アプライアンスでは、電源、電圧、またはディスクの障害が複数回発生した場合、管理サービスは syslog または電子メール通知を送信しません。

[NSHELP-29443]

NetScaler Gateway

Chrome 98 または Edge 98 ブラウザバージョンにアップグレードすると、ユーザーは EPA プラグインまたは VPN プラグインを起動できません。この問題を解決するには、次の手順に従います。

1. VPN プラグインをアップグレードする場合、エンドユーザーは初めて VPN Client を使用して接続し、マシンで修正プログラムを入手する必要があります。それ以降のログイン試行では、接続するブラウザまたはプラグインを選択できます。
2. EPA のみのユースケースでは、エンドユーザーはゲートウェイに接続するための VPN クライアントを持ちません。この場合は、次の操作を行います。

    a) ブラウザを使用してゲートウェイに接続します。
    b) ダウンロードページが表示されるのを待ち、nsepa_setup.exe をダウンロードします。
    c) ダウンロード後、ブラウザを閉じて nsepa_setup.exe ファイルをインストールします。
    d) クライアントを再起動します。

[NSHELP-30641]

TCP SYSLOG 設定を使用した高可用性セットアップでは、HA フェールオーバーまたは設定のクリア操作中にノードがクラッシュすることがあります。

[NSHELP-29251]

NetScaler Gateway ポータルページでは、**RDP** プロキシリンクアイコンは **RFWebUI** ポータルテーマでも変わりません。

[NSHELP-28974]

NetScaler Gateway アプライアンスをバージョン 13.0 にアップグレードすると、セッションプロファイルのプロキシ構成が意図したとおりに機能しなくなります。非 HTTP NS プロキシが設定されている場合、プロキシ接続はバイパスされます。

例:
VPN セッションアクションプロキシ NS-HttpProxy 192.0.2. 0:24-SSLProxy 192.0.2. 0:24

この例では、-HttpProxy は意図したとおりに動作しますが、-sslProxy は機能しません。

[NSHELP-28640]

割り当てられたメモリがリセットされないため、DTLS Audio で STA を処理しているときに NetScaler Gateway アプライアンスがクラッシュします。

[NSHELP-28432]

NetScaler アプライアンスは、非推奨の VPND プロセスに関連する古いメッセージをログに記録します。

[NSHELP-28163]

StoreFront にバックアップ負荷分散仮想サーバー経由でアクセスすると、VPN 仮想サーバーを介した StoreFront へのアクセスが失敗します。

[NSHELP-27852]

既存の ICA セッションに再接続すると、NetScaler Gateway アプライアンスがクラッシュすることがあります。

[NSHELP-27441]

GUI を使用してクラシック認可ポリシーをバインド解除することはできません。ただし、CLI を使用して認証、承認、および監査認可ポリシーのバインドを解除することはできます。

今回の修正により、GUI を使用して承認ポリシーのバインドを解除できるようになりました。

[NSHELP-27064]

NetScaler Web App Firewall

XML ライブラリバージョン 2.9.12 にアップグレードすると、WAF シグネチャ関連の XML ファイルが解析中に破損します。

[NSWAF-8662]

HTTP リクエストが Web App Firewall モジュールによってブロックされた場合でも、JSON コマンドインジェクション保護が ns.log メッセージで`Not blocked`に表示されます。

[NSHELP-29709]

Web App Firewall ログメッセージがクロスサイトスクリプティング (XSS) URL 属性違反で`BAD URL`が表示され、そのカテゴリが属するカテゴリ (タグ、パターン、属性など) に関する用語`Bad URL`が不明確になります。

[NSHELP-29358]

SSL タイプの負荷分散仮想サーバーでボット管理ポリシーが有効になっていると、ボットデバイスのフィンガープリント投稿 URL が失敗することがあります。

[NSHELP-29198]

Web App Firewall 署名 ID 1048 は、NetScaler Gateway ページの読み込みをブロックします。

[NSHELP-29113]

次のモジュールが有効になっていると、NetScaler アプライアンスがクラッシュすることがあります。

- 高度なセキュリティチェック機能を備えた Web App Firewall
- Appqoe。

[NSHELP-28251]

負荷分散

Autoscale タイプの DNS サービスグループのメンバーが TROFS 状態で、同じメンバーが再びグループに追加された場合、このメンバーの状態は伝達されません。

[NSHELP-29493]

ポリシー式にワイルドカードが含まれている`add dns action`コマンドおよび`add location`コマンドでは、増分同期が失敗します。

[NSHELP-29301]

静的にバインドされたメンバーと動的に解決された DNS レコードの間に競合がある場合、一部のサービスグループメンバーは Autoscale サービスグループリストから削除されません。この問題が原因でメモリが破損する。

[NSHELP-28949]

show コマンドと stat コマンドに表示されるサービスグループの状態が矛盾しています。

[NSHELP-28931]

まれに、ロケーションデータベースの構成が構成 (ns.conf) ファイルに含まれていないことがあります。

[NSHELP-28570]

ピアが既存の接続のリセット要求を送信すると、SQL または Oracle Type Monitor がクラッシュします。

[NSHELP-28478]

永続性が有効な展開では、コンテキスト保存中に不正な仮想サーバが保存される。

[NSHELP-28342]

高可用性フェイルオーバー後、または NetScaler アプライアンスが再起動されると、LB グループの永続性構成が失われます。

[NSHELP-28071]

デフォルトモニタがサービスにバインドされている場合でも、デフォルトモニタの設定済み状態は無効と表示されます。

[NSHELP-27669]

その他

アプライアンスを NetScaler バージョン 12.1 ビルド 63.22 にアップグレードすると、次の問題が発生します。

- 拡張機能の検索 API は、アップグレード後に動作しないことがあります。

[NSHELP-29860]

ネットワーク

NetScaler アプライアンスは、次の条件をすべて満たすとクラッシュすることがあります。

- 負荷分散ルートは、アプライアンスのトラフィックドメインに設定されます。
- アプライアンス上で設定のクリア操作が実行されます。

[NSNET-23847]

大規模な NAT44 セットアップでは、次の理由により、SIP トラフィックの受信中に NetScaler アプライアンスがクラッシュすることがあります。

- LSN モジュールは、参照カウントのデクリメント中、またはサービスを削除している間は、サービスを検索しません。

[NSHELP-29134]

大規模な NAT44 展開では、次の理由により、SIP トラフィックの受信中にアプライアンスがクラッシュすることがあります。

- LSN モジュールが、既に削除されたサービスのメモリ位置にアクセスしました。

[NSHELP-28815]

偶数のパケットエンジン（PE）を搭載した NetScaler アプライアンスでは、アクティブなインターフェイスのステータスが冗長インターフェイスセット（LR チャネル）の非アクティブとして誤って表示されます。この問題は、NetScaler アプライアンスの機能には影響しません。

[NSHELP-28099]

NetScaler アプライアンスは、コールドリスタート後に`coldStart`SNMP トラップメッセージを生成しないことがあります。

[NSHELP-27917]

プラットフォーム

`ntpdate` コマンドがクラッシュし、コアダンプが発生する。

[NSHELP-29649]

SSL

エクスポート暗号スイートを使用すると、NetScaler MPX 7500 アプライアンスがクラッシュします。

[NSSSL-11294]

まれに、次のプラットフォームで DTLS の処理中にクラッシュが発生することがあります。

- MPX 5900
- MPX/SDX 8900
- MPX/SDX 15000
- MPX/SDX 15000-50G
- MPX/SDX 26000

- MPX/SDX 26000-50S
- MPX/SDX 26000-100G

[NSHELP-29538]

高可用性セットアップでは、プライマリノードとセカンダリノード間で証明書タイプが正しく同期されません。

[NSHELP-27589]

VPN 展開では、NetScaler アプライアンスはセッションを再利用するために SSL セッションをキャッシュから取得し、プロキシまたはバックエンドサーバーと通信します。これは、クライアントから受信した SNI と、キャッシュされたセッションに存在する SNI を一致させることなく行われます。

その結果、キャッシュされたデータに応じて SNI が送信されないか、異なる SNI が送信されます。

[NSHELP-27439]

システム

侵入防御システム（IPS）リソースに割り当てられたメモリをクリアすると、NetScaler アプライアンスでメモリリークが観察されます。

[NSHELP-29992]

SSL プロファイルと SSL 証明書キーを HTTP QUIC 仮想サーバーに関連付ける構成操作は、NetScaler クラスター展開で失敗することがあります。

[NSHELP-29655]

次の条件が満たされると、同じクライアント接続に対する 2 番目の要求は失敗します。

- ClientSideMeasurements は有効になっています。
- HEAD 要求が受信されました。

[NSHELP-29353]

一部のシナリオでは、NetScaler アプライアンスが次の状況でクラッシュすることがあります。

- TCP ジャンボフレームが使用されます。
- 永続性は TCP 負荷分散仮想サーバで構成されます。

[NSHELP-29162]

NetScaler アプライアンスは、次の条件が満たされるとクラッシュします。

- クライアント側の測定オプションは AppFlow アクションで有効になります。
- チャンクヘッダーはパケット境界上にあります。

[NSHELP-29049]

NetScaler アプライアンスは、HTTP パイプライン（1 つまたは複数の要求）のサイズが 128 KB を超えると、接続をリセットします。この問題は、パイプラインのサイズが 128 KB に制限されているために発生します。

[NSHELP-28846]

NetScaler 侵入防御システム（IPS）では、次の条件が満たされた場合、データの挿入または変更時に書き換えポリシーの問題が確認されます。

- NetScaler アプライアンスは、バックエンドサーバー接続が開く前にデータパケットを IPS サーバーに送信します。

[NSHELP-28496]

高可用性セットアップでは、次の理由により、セカンダリノードで Admin パーティション設定の HA 同期が失敗します。

- セカンダリノードでの設定の負荷が大きいため、メモリ不足の問題が発生する

[NSHELP-28409]

クライアントが複数の TCP ストリームがある接続をリセットすると、サーバー側のトランザクションレコードは送信されないため、それらのデータストリームの L4 レコードが失われます。

[NSHELP-28281]

TCP 接続では、NetScaler アプライアンスは、次の条件がすべて満たされた場合、サーバーから受信した FIN パケットをクライアントに転送するのではなく、ドロップすることがあります。

- TCP バッファリングは有効です。
- サーバは FIN パケットとデータパケットを別々に送信します。

[NSHELP-274]

クラスタセットアップでは、`set ratecontrol`コマンドは NetScaler アプライアンスを再起動した後にのみ機能します。

[NSHELP-21811]

NetScaler アプライアンスが FIN フラグが設定された順序の悪い TCP パケットを受信すると、次の問題が発生することがあります。

- NetScaler アプライアンスが不正な SACK を送信します。これは、アプライアンスが 1 バイトの順不同の TCP パケットではなく 2 バイトを受信したことを示します。
- NetScaler アプライアンスは、順番どおりの TCP パケットを受信して TCP FIN パケットを認識しません。

[NSBASE-15735]

ユーザーインターフェイス

確認のプロンプトが表示されないため、誤って SSL 証明書をリンク解除してしまう可能性があります。今回の修正により、ユーザーがリンクされた証明書をクリックすると、証明書のリンクを解除する前に確認を求められるようになりました。

[NSUI-17897]

NetScaler GUI を使用して、接続フェイルオーバーがすでに有効になっている ACL ベースの RNAT ルールを変更すると、次のエラーで失敗することがあります。

- `Invalid argument value [connfailover]`

[NSHELP-29243]

NetScaler GUI を使用して SSL 証明書を構成または確認するときに、エラー`Directory doesn't exist`が表示されることがあります。この問題は、**SSL** フォルダ/`nsconfig`/`ssl`に 2 つの連続したドット (`..`) を含むファイル名が存在する場合に発生します。

[NSHELP-28589]

高可用性セットアップで、組み込みポリシーパターンセットがプライマリノードで変更された場合、組み込みポリシーパターンセットバインディングの HA 同期が失敗することがあります。

[NSHELP-28460]

ADC GUI で RPC ノードの secure オプションを選択解除すると、次のエラーメッセージが表示されます。

引数の前提条件がありません [validateCert、セキュア == はい]

[NSHELP-28239]

サイドパネルビューでリストのページサイズを変更しようとすると、ページがゆがんでしまいます。

[NSHELP-28220]

`create ssl rsakey`や`create ssl cert`など、一部の SSL コマンドで引数に特殊文字が使用されている場合、余分なバックスラッシュ文字が誤って導入される。

[NSHELP-27378]

ping または ping6 コマンドにインターフェイス (-I) オプションを指定すると、次のエラーで失敗することがあります。

- **`interface`** `option not supported`

[NSHELP-26962]

既知の問題

リリース 13.1 〜17.42 に存在する問題。

AppFlow

HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

[NSINSIGHT-943]

認証、承認、監査

NetScaler アプライアンスは、重複したパスワードログインの試行を認証せず、アカウントのロックアウトを防ぎます。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema が NetScaler GUI のログインスキーマエディタ画面に正しく表示されません。

[NSAUTH-6106]

ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
`show adfsproxyprofile <profile name>`

回避方法:

クラスタ内のプライマリのアクティブな NetScaler に接続し、`show adfsproxyprofile <profile name>` コマンドを実行します。プロキシプロファイルの状態が表示されます。

[NSAUTH-5916]

次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

- [LDAP 到達可能性をテスト] オプションが開きます。
- 無効なログイン認証情報が入力され、送信されます。
- 有効なログイン認証情報が入力され、送信されます。

回避方法:

[LDAP 到達可能性のテスト] オプションを閉じて開きます。

[NSAUTH-2147]

キャッシュ

統合キャッシュ機能が有効で、アプライアンスのメモリが不足している場合、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-22942]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、CLAG が Mellanox NIC 上に作成されている場合、VPX インスタンスの再起動時に CLAG MAC が変更されます。VPX インスタンスへのトラフィックは再起動後に停止します。これは、MAC テーブルに古い CLAG MAC エントリがあるためです。

[NSSVM-4333]

NetScaler Gateway

サーバー証明書が信頼されていると、サーバー検証コードが失敗することがあります。その結果、エンドユーザーはゲートウェイにアクセスできなくなります。

[NSHELP-28942]

NetScaler Gateway の高可用性設定では、Gateway Insight が有効になっていると、セカンダリノードがクラッシュする可能性があります。

[NSHELP-28856]

macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

[NSHELP-28551]

クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

[NSHELP-28404]

高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

- HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

回避方法:

HA 同期が完了した後（両方のノードが同期成功状態にある）のみ、手動の HA フェールオーバーを連続して実行します。

[NSHELP-25598]

Gateway Insight は、VPN ユーザに関する正確な情報を表示しません。

[NSHELP-23937]

次の条件を満たす場合、Windows ログオン後に VPN プラグインはトンネルを確立しません。

- NetScaler Gateway アプライアンスが常時オン機能用に構成されている
- アプライアンスは 2 要素認証`off`による証明書ベースの認証用に設定されています。

[NSHELP-23584]

スキーマを参照しているときに、エラーメッセージ`Cannot read property 'type'of undefined`が表示されることがあります。

[NSHELP-21897]

Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

[CGOP-19355]

無効な STA チケットによるアプリケーションの起動失敗は、Gateway Insight では報告されません。

[CGOP-13621]

Gateway Insight レポートでは、SAML エラーエラーの場合、「認証タイプ」フィールドの値`SAML`が誤って`Local`と表示されます。

[CGOP-13584]

高可用性セットアップでは、NetScaler フェイルオーバー時に、NetScaler Console のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

[CGOP-13511]

ブラウザからローカルホスト接続を受け付けると、macOS の **Accept Connection** ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

[CGOP-13050]

一部の言語では、**Citrix SSO** アプリ **>** ホームページの`Home Page`テキストが切り捨てられます。

[CGOP-13049]

NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

[CGOP-11830]

Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

[CGOP-7269]

クラスタ展開では、非 CCO ノードで`force cluster sync`コマンドを実行すると、ns.log ファイルに重複したログエントリが含まれます。

[CGOP-6794]

負荷分散

高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

[NSLB-7679]

entityofs サービスグループのトラップの ServiceGroupName 形式は次のとおりです。
<service(group)name>?<ip/DBS>?<port>

トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (?) はセパレータとして使用されます。NetScaler は、疑問符（?）付きのトラップを送信します。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

[NSHELP-28080]

その他

高可用性セットアップで強制同期が行われると、アプライアンスはセカンダリノードでset urlfiltering parameterコマンドを実行します。
その結果、セカンダリノードは、TimeOfDayToUpdateDB パラメータで指定された次のスケジュールされた時刻まで、スケジュールされた更新をスキップします。

[NSSWG-849]

URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

[NSHELP-22409]

ネットワーク

DPDK を搭載した NetScaler BLX アプライアンスは、次の条件をすべて満たすと再起動に失敗することがあります:

- NetScaler BLX アプライアンスは、hugepagesの小さい数が割り当てられます。たとえば、1G です。
- NetScaler BLX アプライアンスには多数のワーカープロセスが割り当てられています。たとえば、28 と入力します。

この問題は、/var/log/ns.logにエラーメッセージとして記録されます。

- BLX-DPDK:DPDK Mempool could Not be Initialized **for** PE-x

注:x はワーカープロセス数以下の数です。

回避方法:

多数のhugepagesを割り当て、アプライアンスを再起動します。

[NSNET-25173]

DPDK を搭載した NetScaler BLX アプライアンスは、次の条件が満たされると再起動に失敗することがあります。

- NetScaler BLX アプライアンスにはhugepagesの大きい数が割り当てられます。たとえば、16 GB と入力します。

この問題は、/var/log/ns.logにエラーメッセージとして記録されます。

- EAL: rte_mem_virt2phy(): cannot open /proc/self/pagemap: Too many open files

回避方法:

この問題を解決するには、次のいずれかの回避策を使用します。

- ulimitコマンドを使用するか、limits.confファイルを編集して、Linux ホストで開くことができるファイルの上限を増やします。
- 割り当てられたhugepagesの数を減らします。

[NSNET-24727]

DPDK モードの NetScaler BLX アプライアンスは、DPDK 簡易機能のため、再起動に少し時間がかかる場合があります。

[NSNET-24449]

NetScaler BLX アプライアンス 13.0 61.x ビルドから 13.0 64.x ビルドにアップグレードすると、BLX 構成ファイルの設定が失われます。その後、BLX 構成ファイルがデフォルトにリセットされます。

[NSNET-17625]

DPDK を搭載した NetScaler BLX アプライアンスの Intel X710 10G (i40e)インターフェイスでは、次のインターフェイス操作はサポートされていません:

- 無効化
- 有効化
- リセット

[NSNET-16559]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスは、BLX 構成ファイル（/etc/blx/blx.conf）の設定に関係なく、常に共有モードで展開されます。この問題は、Debian ベースの Linux システムではデフォルトで存在するmawkが、blx.confファイル内に存在するawkコマンドの一部を実行しないために発生します。

回避方法:

NetScaler BLX アプライアンスをインストールする前にgawkをインストールします。Linux ホスト CLI で次のコマンドを実行してgawkをインストールできます。

- apt-get install gawk

[NSNET-14603]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

回避方法:

NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します。

- dpkg –add-architecture i386
- apt-get update
- apt-get dist-upgrade
- apt-get install libc6:i386

[NSNET-14602]

FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

[NSNET-5233]

NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

[NSHELP-21082]

プラットフォーム

高可用性フェイルオーバーは AWS および GCP クラウドでは機能しません。AWS および GCP クラウド、および NetScaler VPX オンプレミスでは、管理 CPU が 100% の容量に達する可能性があります。これらの問題はいずれも、次の条件が満たされた場合に発生します。

1. NetScaler アプライアンスの初回起動時には、プロンプトが表示されたパスワードは保存されません。
2. その後、NetScaler アプライアンスを再起動します。

[NSPLAT-22013]

13.0/12.1/11.1 ビルドから 13.1 ビルドにアップグレードするか、13.1 ビルドから 13.0/12.1/11.1 ビルドにダウングレードすると、一部の Python パッケージが NetScaler アプライアンスにインストールされません。この問題は、次の NetScaler バージョンで修正されています。

- 13.1-4.x
- 13.0—82.31 およびそれ以降
- 12.1—62.21 およびそれ以降

NetScaler バージョンを 13.1-4.x から次のバージョンのいずれかにダウングレードすると、Python パッケージはインストールされません。

- 任意の 11.1 ビルド
- 12.1—62.21 およびそれ以前
- 13.0-81.x およびそれ以前

[NSPLAT-21691]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、2 番目のノードと CLIP で CLAG MAC の不一致があります。

- CLAG は Mellanox NIC 上に作成されます。
- クラスターと CLAG セットアップに別の VPX インスタンスを追加します。

その結果、VPX インスタンスへのトラフィックが停止します。

[NSPLAT-21049]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、CLIP テーブルと MAC テーブルで MAC アドレスの不一致が原因で最初のノードがダウンします。

- CLAG は Mellanox NIC 上に作成されます。
- クラスタから 2 つ目のノードを削除します。

[NSPLAT-21042]

Azure リソースグループから Autoscale 設定または仮想マシンスケールセットを削除する場合は、NetScaler インスタンスから対応するクラウドプロファイル構成を削除します。`rm cloudprofile` コマンドを使用して、プロファイルを削除します。

[NSPLAT-4520]

Azure の高可用性セットアップでは、GUI を使用してセカンダリノードにログオンすると、Autoscale クラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは必ずプライマリノードで設定してください。

[NSPLAT-4451]

RPC ノードのパスワードに特殊文字が含まれていると、GCP および AWS クラウド上の NetScaler VPX インスタンスの HA フェイルオーバーが失敗します。

[NSHELP-28600]

ポリシー

処理データのサイズが、設定されているデフォルトの TCP バッファサイズを超えると、接続がハングすることがあります。回避策：TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

[NSPOLICY-1267]

SSL

NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

回避方法：

1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例：`set ssl vserver <name> -SSL3 DISABLED`。
2. 構成を保存します。

[NSSSL-9572]

認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

[NSSSL-6478]

同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

[NSSSL-6213]

HSM タイプとして Key Vault を指定せずに HSM キーを削除すると、次のような誤ったエラーメッセージが表示されます。
エラー:CRL 更新は無効です

[NSSSL-6106]

セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

[NSSSL-4427]

SSL プロファイルの SSL プロトコルまたは暗号を変更しようとすると、不正な警告メッセージ`Warning: No usable ciphers configured on the SSL vserver/service,`が表示されます。

[NSSSL-4001]

期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

[NSSSL-3184]

システム

アプライアンスがクライアントから **max_concurrent_stream** 設定フレームを受信しない場合、**MAX_CONCURRENT_STREAMS** 値はデフォルトで **100** に設定されます。

[NSHELP-21240]

mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

[NSHELP-10972]

NetScaler GUI（ナビゲーション：[システム**] > [**レポート**] > [PCI DSS** レポートの生成**]**）で **PCI DSS** レポートを生成する際に問題が発生します。

[NSBASE-16225]

LogStream トランスポートタイプが Insight に設定されている場合、HDX Insight SkipFlow レコードでクライアント IP とサーバー IP が反転します。

[NSBASE-8506]

NetScaler アプライアンスは、ヘッダー名フィールドにドット（”.”）の付いたカスタム HTTP ヘッダーを含むパケットをドロップします。このアクションは、`allowOnlyWordCharactersAndHyphen` パラメータがデフォルトの HTTP プロファイルでデフォルトで有効になっているために発生します。

回避策: デフォルトの HTTP プロファイルで`allowOnlyWordCharactersAndHyphen`を無効にします。ただし、Citrix では有効にしておくことをお勧めします。

[NSBASE-16722]

ユーザーインターフェイス

MQTT リライト機能では、GUI のエクスプレッションエディタを使用してエクスプレッションを削除することはできません。

回避方法:

CLI から MQTT タイプの add または edit action コマンドを使用します。

[NSUI-18049]

NetScaler GUI では、`Dashboard`タブの下にある`Help`リンクが壊れています。

[NSUI-14752]

CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

回避方法:

NetScaler GUI または CLI を使用して、IPSec プロファイル、IP トンネル、および PBR ルールを追加して、CloudBridge Connector を構成します。

[NSUI-13024]

GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

[NSUI-6838]

管理パーティションの設定で、証明書失効リスト (CRL) ファイルのアップロードと追加が失敗する。

[NSHELP-20988]

NetScaler アプライアンスのバージョン 13.0-71.x を以前のビルドにダウングレードすると、ファイル権限の変更のために一部の NITRO API が機能しないことがあります。

回避方法:

`/nsconfig/ns.conf` の権限を 644 に変更します。

[NSCONFIG-4628]

あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

1. NetScaler アプライアンスをいずれかのビルドにアップグレードします。
    - 13.0 52.24 ビルド
    - 12.1 57.18 ビルド
    - 11.1 65.10 ビルド
2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
3. NetScaler アプライアンスを古いビルドにダウングレードします。

CLI を使用してこれらのシステムユーザーの一覧を表示するには、
コマンドプロンプトで次のように入力します。

```
query ns config -changedpassword [-config <full path of the configuration
 file (ns.conf)>]
```

回避方法:

この問題を修正するには、次の独立したオプションのいずれかを使用します。

1. NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3)、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。

2. アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
3. 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.htmlを参照してください。

[NSCONFIG-3188]

NetScaler 13.1—12.51 リリースのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリース Build 13.1—12.51 に存在する機能強化と変更、修正された問題と既知の問題について説明します。

ビルド 13.1 ～12.51 は、ビルド 13.1 ～12.50 を置き換えます。

このビルドには、NSWAF-8668 の問題の修正も含まれています。

メモ

このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正と勧告のリストについては、セキュリティ速報を参照してください。

新機能

ビルド 13.1 ～12.51 で利用できる機能強化と変更。

認証、承認、監査

最新バージョンの **Intune NAC API** のサポート　Intune NAC API の最新バージョンでは、NetScaler Gateway の Intune ネットワークアクセス制御（NAC）のサポートが強化されました。

[NSAUTH-9722]

接続プロキシを使用した **nFactor** 認証のための **GSLB** アクティブ/アクティブ展開のサポート

接続プロキシを使用した nFactor 認証の GSLB アクティブ/アクティブ展開のサポートが追加されました。このサポートは、NetScaler Gateway と認証、承認、監査の両方のシナリオに適用されます。
現在、nFactor 認証にさまざまな要素が設定されていて、ゲートウェイが GSLB 用に設定されている場合、クライアントのリクエストが異なる GSLB サイトに到達すると、認証が中断する可能性があります。

たとえば、LDAP が第 1 要素として設定され、RADIUS が 2 番目の要素として設定されている場合、次のシナリオでは認証が中断される可能性があります。

- LDAP に対するクライアント要求が GSLB サイト 1 に届く。
- RADIUS 要求は GSLB サイト 2 に到着します。
  認証を完了してトラフィックを処理するために、要求を正しい GSLB サイトにルーティングするために、接続プロキシが使用されるようになりました。

[NSAUTH-7141]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、管理サービスがバックグラウンドで NetScaler インスタンスをポーリングして、SSL 証明書、ネットワーク機能、構成監査などの操作を確認します。これで、要件に応じてこのポーリングを有効または無効にできます。このポーリングを無効にすると、管理サービスと ADC インスタンスのパフォーマンスが向上します。

[NSSVM-4991]

NetScaler Web App Firewall

JSON セキュリティーチェック **(SQL**、**CMD**、および **XSS)** の冗長ロギング　NetScaler アプライアンスでは、JSON セキュリティチェック用のパターン、パターンペイロード、HTTP ヘッダーの詳細などの違反の詳細をログに記録するための詳細なログレベルパラメーターを構成できるようになりました。ログの詳細は、監視とトラブルシューティングのために NetScaler Console サーバーに送信されます。詳細ログメッセージは ns.log ファイルには保存されません。

[NSWAF-8269]

Web App Firewall Classic 監査ログポリシーを廃止する

Web App Firewall ポリシーをグローバルにバインドするために、`bind audit syslogGlobal`および`bind audit nslogGlobal`コマンドで新しいグローバルバインディングタイプ`APPFW_GLOBAL`を構成できるようになりました。グローバルバインドされた監査ログポリシーは、Web App Firewall のロギングコンテキストで評価されます。

[NSWAF-406]

負荷分散

MQTT プロトコルの書き換えポリシーサポート　リライト機能で MQTT プロトコルがサポートされるようになりました。MQTT クライアント要求とサーバ応答のパラメータに基づいてアクションを実行するように、書き換えポリシーを設定できます。

[NSLB-8661]

サービスの優先順位

サービスの優先順位機能を使用すると、負荷分散の選択プリファレンスに基づいて、サービスまたはサービスグループの順序に優先順位を付けることができます。サービスまたはサービスグループを LB または GSLB 仮想サーバにバインドするときに、サービスの選択順序を設定できるようになりました。新しいパラメータ-order <number> が bind コマンドに追加され、サービスセレクションプリファレンスが設定されます。

既定では、最小の順序番号が優先されます。ただし、この既定の選択動作を延期することはできます。新しい LB action コマンドと policy コマンドを使用して、着信クライアントトラフィックに基づいてサービスセレクション順序を設定できるようになりました。

サービスの優先順位付け機能は、より少ない構成コマンドでプライマリおよびバックアップ仮想サーバチェーン機能の動作を模倣します。

[NSLB-8039]

ネットワーク

セッションレス負荷分散を設定するには、**IP Tunnel** 外部ヘッダーにクライアント **IP** アドレスを挿入します　次の設定のセッションレス負荷分散構成では、カプセル化器の NetScaler アプライアンスは、IP トンネルの外部ヘッダーのソース IP として、クライアント IP アドレスではなく SNIP アドレスを使用します。

- 負荷分散仮想サーバ:
- リダイレクションモード (m): IP トンネル
- セッションレス: 有効
- IP トンネルグローバルパラメータ:
- クライアントソース IP アドレスを使用 (useClientSourceIP): 有効

ただし、一部のシナリオでは、トンネルカプセル化解除（バックエンド NetScaler またはバックエンドサーバー）がクライアントの IP アドレスを認識する必要があります。

この要件を満たすために、NetScaler アプライアンスは、IP トンネルの外部ヘッダーのソース IP としてクライアント IP アドレスを使用するようになりました。

詳細については、「IP Over IP を使用して DSR モードで負荷分散を構成する」を参照してください。

[NSNET-21804]

プラットフォーム

VMware ESXi イメージが仮想ハードウェアバージョン **13** まで起動する　VMware ESXi イメージ（12.1 以降）から NetScaler VPX インスタンスを展開すると、仮想マシンはデフォルトでハードウェアバージョン 13 で起動します。

[NSPLAT-21416]

Citrix Hypervisor での **Intel** イーサネットコントローラー **X710** および **XL710** シリーズのサポート

Citrix Hypervisor で実行されている NetScaler VPX インスタンスを、次の NIC でシングルルート I/O 仮想化（SR-IOV）を使用して構成できるようになりました。

- Intel X710 10G
- Intel XL710 40G

[NSPLAT-21410]

AWS 共有 **VPC** でプライベート **IP** アドレスを使用して **VPX** 高可用性ペアをデプロイする

AWS 共有仮想プライベートクラウド (VPC) を使用して、異なる AWS ゾーンのプライベート IP アドレスを使用して VPX 高可用性ペアをデプロイできるようになりました。VPC 共有により、複数の AWS アカウントがアプリケーションリソースを、一元管理された共有の VPC に作成できます。NetScaler VPX インスタンスは、AWS 共有 VPC 内に作成できます。共有 VPC を使用すると、作成および管理する VPC の数が減り、請求とアクセスコントロールには個別のアカウントが使用されます。

[NSPLAT-21401]

SSL

JA3 SSL フィンガープリントに基づいてマルウェアを検出する新しい表現　新しい SSL 式 CLIENT.SSL.JA3_FINGERPRINT が追加されました。この式は、設定された JA3 フィンガープリントとリクエストを比較することで、悪意のあるリクエストを識別するのに役立ちます。

例:

```
add ssl policy ja3_pol -rule "CLIENT.SSL.JA3_FINGERPRINT.EQ(bb4c15a90e93a25ddc
)"-action reset
```

[NSSSL-10156]

クラスタでの証明書バンドルのサポート

証明書バンドルがクラスタセットアップでサポートされるようになりました。

[NSSSL-9854]

SSL 証明書バンドルのサポート

証明書バンドル機能が拡張され、バンドルがエンティティとして扱われるようになりました。したがって、中間証明書ごとにファイルを作成する必要はありません。2 つの証明書バンドルで、中間証明書チェーンの一部を共有できるようになりました。証明書バンドルの一部でもある同じサーバ証明書とキーを使用して、証明書とキーのペアを追加することもできます。証明書バンドルの削除も簡略化されます。

以前は、証明書バンドルを追加すると、設定に複数のコマンドが追加されていました。2 つのバンドルが共通の中間証明書を共有している場合、別の証明書バンドルを追加することはできません。削除も手動のプロセスでした。

[NSSSL-9425]

システム

html インジェクション関連のコマンドは 13.1 リリースで削除されました。この変更により、すべてのバックエンドコードが削除されます。

[NSBASE-14742]

解決された問題

ビルド 13.1—12.51 で対処される問題。

認証、承認、監査

電子メール OTP が構成されている場合、NetScaler アプライアンスがクラッシュします。

[NSHELP-29312]

ネイティブ OTP 暗号化ツールでは、デバイス名に特殊文字を使用できません。

[NSHELP-28795]

NetScaler アプライアンスにログインすると、次の両方の条件が満たされると、空白のパスワードフィールドが表示されます。

- Duo の 2 要素認証が設定されている
- RFWebUI ポータルテーマが使用されています

[NSHELP-27868]

次の条件が満たされると、サービスへのアクセスは拒否されます。

- このサービスは認証仮想サーバにバインドされます。
- 401 認証は、サービスとサービスがバインドされている仮想サーバーで構成されます。

[NSHELP-26903]

まれに、次の条件を満たすと、高可用性セットアップのセカンダリノードがクラッシュすることがあります。

- `aaa groups`および/または`aaa users`は NetScaler アプライアンスで構成されます。

[NSHELP-26732]

LDAP、RADIUS、または TACACS サービスの管理者パスワードに二重引用符（”）が含まれている場合、NetScaler アプライアンスは`Test Connectivity`チェック中にそれを削除し、接続に失敗します。

[NSHELP-23630]

NetScaler SDX アプライアンス

NetScaler SDX 14000-40G、15000、および 15000-50G のプラットフォームでは、CLI を使用してインターフェイス速度を設定すると失敗します。

[NSHELP-29388]

NetScaler SDX プラットフォームでホストされている ADC インスタンスのプロファイルを変更すると、ログファイルに`save config`コマンドのエントリが追加されることがあります。

[NSHELP-29343]

NetScaler SDX アプライアンスでは、管理サービスで実行されている SNMP エージェントが、存在しない OID に対して誤ったエラーコードを返します。

[NSHELP-29209]

データレコードの総数が 5000 未満の場合、ADC イベントテーブルのデータをページ間でソートできるようになりました。

[NSHELP-29170]

NetScaler Gateway

EPA が構成され、十分なメモリが利用できない場合、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-28329]

ディレクトリ /var/netscaler/logon/logonPoint/Custom/ は、ディレクトリが最初に存在しなかった場合、アップグレード後に作成されません。

[NSHELP-28223]

ns_aaa_json.c ファイルに NS_AUDITLOG_STR* ログ用の余分な行が表示される場合があります。

[NSHELP-28160]

VPN 接続が確立されると、DNS 登録は機能しません。

この問題を修正するには、nsapimgr ノブ nsapimgr_wr.sh -ys call=toggle_vpn_configured_dns_disable_override を有効にする必要があります。

[NSHELP-27760]

転送ログイン中に、イントラネット IP サブネットがクライアント側で正しく表示されないことがあります。

[NSHELP-26904]

L7 レイテンシーが有効になっている場合、セッションの ICA レイテンシーは Citrix Director で 64,000 ミリ秒と誤って記録されます。L7 レイテンシーは、`nsapimgr`ノブ`enable_ica_l7_latency`が 1 に設定されているときに有効になります。

[NSHELP-23459]

ユーザーが NetScaler Gateway アプライアンスにログインして ICA アプリにアクセスすると、Gateway Insight ログファイルに次のメッセージが殺到します。

```
GwInsight: Func=ns_aaa_copy_email_id_to_vpn_record input hash_attrs_len
 is zero Oct 25 23:01:31 <local0.err> 10.217.24.1Oct 25 23:01:31 <
local0.err> 10.217.24.101 10/26/2021:06:01:31 GMT NSGWTHDR 0-PPE-0 :
default SSLVPN Message 10491736 0 : GwInsight: Func=ns_aaa_copy_email_id_to_vp
 input hash_attrs_len is zero
```

[CGOP-19685]

NetScaler Gateway ポータルのエンタープライズブックマーク機能は、次のプロトコルのみをサポートします。その他のブックマークはすべてブロックされます。`http://`、`https://`、`rdp://`、および`ftp://`。

[CGOP-19543]

NetScaler Web App Firewall

WAF 署名を使用している場合は、ビルドをアップグレードした後、デフォルトの署名を含むすべての WAF 署名を最新バージョンに更新する必要があります。次に、必要なシグニチャルールを再度有効にします。

[NSWAF-8668]

ボット管理システムでトラップ URL が自動生成されると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-29339]

負荷分散

失敗したコマンドに ENUM 値が欠落しているため、GSLB サービスグループはモニターの更新を処理できません。

[NSHELP-29050]

NetScaler アプライアンスは、解放されているパーティションとは異なるパーティションに割り当てられているメモリを解放しようとするとクラッシュします。

[NSHELP-29038]

ZONE タイプの DNS レコードが親ドメインで使用できる場合、既存の NS レコードを持つ子ドメインのクエリを実行すると、子ドメイン NS レコードではなく親ドメインの SOA レコードになります。

[NSHELP-28793]

GSLB 仮想サーバーが次のように構成されている場合、NetScaler アプライアンスは、予想される GSLB サービス IP アドレスを持つ GSLB ドメインクエリに応答しないことがあります。
永続タイプ:
送信元 IP アドレス負荷分散アルゴリズム:
静的近接バックアップ負荷分散方式: ラウンドトリップタイム (RTT)

[NSHELP-28668]

ワイルドカードポートを使用すると、負荷分散または GSLB ドメインベースの Autoscale サービスグループの状態は DOWN のままになります。

[NSHELP-28548]

GSLB サービスグループにバインドされたモニターに対して、最後の応答メッセージが正しく表示されない。

[NSHELP-28393]

GET 操作中に cookieTimeout 値が正しく設定されないため、CS 仮想サーバーの更新操作が失敗します。

[NSHELP-27979]

MySQL タイプのモニターのモニタープローブを処理すると、アプライアンスが失敗し、最終的にシステムが再起動することがあります。

[NSHELP-27953]

その他

64 ビットアーキテクチャと 1 TB のファイルストレージを備えた Linux システムで実行されている NetScaler CPX インスタンスは、証明書とキーファイルをロードできるようになりました。

[NSHELP-28986]

IDNA2008 標準ドメインでは、URL セットのパターンマッチングが失敗します。

[NSHELP-28902]

VXLAN で MAC ベース転送（MBF）が有効になっていると、ステートフル TCP セッションが確立されていませんでした。

[NSHELP-27125]

ネットワーク

管理パーティションを持つ NetScaler アプライアンスをアップグレードすると、次の条件が満たされると、構成が一部失われることがあります。

- 使用可能なシステムメモリ全体が管理パーティションに割り当てられている場合。

[NSNET-23031]

制限事項-

VLAN ID 2 は内部使用のために予約されています

VLAN ID 2 は、ブリッジおよび none モードでの展開のための内部使用のために予約されています。NetScaler CPX は、0/1 以外のすべてのインターフェイスを VLAN ID 2 にバインドし、VLAN ID 2 の MTU（最大伝送ユニット）は eth0 インターフェイスの MTU と等しく設定されます。VLAN を設定し、それにインターフェイスをバインドする場合は、インターフェイス MTU が 1500 バイト未満の場合は、VLAN の MTU を Linux で設定されているとおりにインターフェイスの MTU に設定します。

[NSNET-22807]

Web アプリケーションファイアウォールプロファイルが高度なセキュリティ保護チェックで構成されている場合、DPDK モードの NetScaler BLX アプライアンスがクラッシュすることがあります。

[NSNET-22654]

次の条件が満たされると、関連サービスのモニタープローブの作成中に NetScaler アプライアンスがクラッシュすることがあります。

- IPv4 アドレスが 1 つ以上あり、IPv6 アドレスがない IP セットを持つネットプロファイル。ネットプロファイルはモニターにバインドされ、モニターは IPv6 サービスに設定されます。

- IPv6 アドレスが 1 つ以上あり、IPv4 アドレスがない IP セットを持つネットプロファイル。ネットプロファイルはモニターにバインドされ、モニターは IPv4 サービスに設定されます。

[NSHELP-29382]

NetScaler アプライアンスでは、メモリ割り当ての失敗後にパッシブ FTP データ接続が失われることがあります。

[NSHELP-26522]

プラットフォーム

VMXNET3 ドライバーを使用する NetScaler VPX インスタンスは、インスタンスが次のいずれかの NetScaler ビルドで実行されている場合、ランダムにクラッシュすることがあります。

- NetScaler 13.1 ビルド 4.x
- NetScaler 13.1 ビルド 9.x

[NSHELP-29120]

ポリシー

NetScaler アプライアンスは、次の条件でクラッシュすることがあります。

- 監査メッセージアクションは、リクエストの本文に 1 つ以上の REGEX 関数を適用した文字列ビルダ式で構成されます。
- [ストリーミング] オプションを有効にして構成されたアプリケーションファイアウォールプロファイル。

たとえば、HTTP.REQ.BODY (10000000) .REGEX_SELECT (再/名前 = [^\ r\ n] * [\ r\ n] +/) などです。

[NSHELP-27895]

SSL

リクエストバインドポイントですでにバインドされているポリシーに対してポリシーアクションが`Forward`に設定されている場合、HTTP リクエストの処理中に NetScaler アプライアンスがクラッシュします。

[NSHELP-29115]

NetScaler アプライアンスは、次の手順を実行するとクラッシュします。

1. SSL タイプのモニタが追加されます。
2. 証明書とキーのペアがモニタにバインドされます。
3. モニタが取り外される。
4. 同じ名前のモニタがもう 1 つ追加されます。
5. 証明書とキーのペアが更新されます。

[NSHELP-28666]

SAN 証明書内のすべての IP アドレスが表示されます。以前は、SAN 証明書に含まれるすべての IP アドレスのうち、最後の SAN IP アドレスのみが表示されていました。

[NSHELP-27336]

外部 HSM で DH 暗号を使用すると、SSL ハンドシェイクが失敗します。

[NSHELP-25307]

システム

NetScaler アプライアンスがクライアントから HTTP/2 GOWAY フレームを受信すると、ストリーム ID が約束 ID（最後にピアが開始したストリーム識別子）よりも大きいストリームがすべて誤ってリセットされます。

[NSHELP-29328]

NetScaler コンソールでは、ADM-エージェントの問題が原因で、ADM-エージェントから高いメモリ使用量が報告されることがあります。

[NSHELP-29285]

NetScaler アプライアンスは、次の条件をすべて満たすとクラッシュします。

- サーバの IP アドレスを持つコンテンツ検査アクションでは、サービスの内部データがすでに設定されている場合はそれを使用します。
- その結果、CI アクションが削除されると、サービスの内部データも削除されます。
- 実際のサービスが削除されると、NetScaler アプライアンスはすでに削除された内部データへのアクセスと削除を試みます。

[NSHELP-28293]

管理者パーティションがある NetScaler アプライアンスでは、デフォルト以外のパーティションで`nstrace`ユーティリティが正しく実行されないことがある

[NSBASE-15738]

クラスタ構成では、ネットワークの問題により、CCO 優先度のノードが Open vSwitch (OVS) から切断されます。ノードがクラスタ構成に再参加した後は、最新の SYN Cookie を受信しません。

[NSBASE-14419]

ユーザーインターフェイス

プールされた容量で構成されたクラスターモードの ADC インスタンスがダウンする。この問題は、クラスタノードにホスト名が設定されている場合や、ノードが起動時に ADM ライセンスサーバに接続するのに時間がかかる場合に発生します。

[NSHELP-28613]

NetScaler GUI では、すべてのクラスターノードではなく、1 つのノードのみのクラスターテクニカルサポートバンドルを誤って生成することがあります。

[NSHELP-28606]

NetScaler GUI を使用してクラスターテクニカルサポートバンドルを生成すると、エラーが発生して失敗することがあります。

[NSHELP-28586]

NetScaler CLI インターフェイスでは、コマンドプロンプトにコマンドを入力しているときに<Tab>キーを押しても、コマンドをバインドするオプションは自動入力されません。

たとえば、次のコマンドを入力すると、<Tab> キーを使用するとオブジェクトは自動入力されません。

`bind authentication vserver <authvservername> -policy <Tab>`。

ここで、認証仮想サーバは、RADIUS ポリシー、IdapPolicy、証明書ポリシー、TACAS ポリシー、高度な認証ポリシーなど、複数のオブジェクトタイプにバインドできます。

[NSCONFIG-6340]

既知の問題

リリース 13.1 〜12.51 に存在する問題。

AppFlow

HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

[NSINSIGHT-943]

認証、承認、監査

SSO 機能をプロキシサーバーで使用すると、NetScaler アプライアンスでメモリリークが発生する場合があります。

[NSHELP-27744]

NetScaler アプライアンスは、重複したパスワードログインの試行を認証せず、アカウントのロックアウトを防ぎます。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema が NetScaler GUI のログインスキーマエディタ画面に正しく表示されません。

[NSAUTH-6106]

ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。

```
show adfsproxyprofile <profile name>
```

回避方法:

クラスタ内のプライマリのアクティブな NetScaler に接続し、`show adfsproxyprofile <profile name>` コマンドを実行します。プロキシプロファイルの状態が表示されます。

[NSAUTH-5916]

次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

- [LDAP 到達可能性をテスト] オプションが開きます。
- 無効なログイン認証情報が入力され、送信されます。
- 有効なログイン認証情報が入力され、送信されます。

回避方法:

[LDAP 到達可能性のテスト] オプションを閉じて開きます。

[NSAUTH-2147]

キャッシュ

統合キャッシュ機能が有効で、アプライアンスのメモリが不足している場合、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-22942]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、CLAG が Mellanox NIC 上に作成されている場合、VPX インスタンスの再起動時に CLAG MAC が変更されます。VPX インスタンスへのトラフィックは再起動後に停止します。これは、MAC テーブルに古い CLAG MAC エントリがあるためです。

[NSSVM-4333]

NetScaler Gateway

サーバー証明書が信頼されていると、サーバー検証コードが失敗することがあります。その結果、エンドユーザーはゲートウェイにアクセスできなくなります。

[NSHELP-28942]

VPN の切断後、DNS リゾルバがホスト名の解決に失敗することがあります。これは、VPN の切断中に DNS サフィックスが削除されるためです。

[NSHELP-28848]

macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

[NSHELP-28551]

クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

[NSHELP-28404]

Windows プラグインは、認証中にクラッシュすることがあります。

[NSHELP-28394]

Gateway Insight は、VPN ユーザに関する正確な情報を表示しません。

[NSHELP-23937]

次の条件を満たす場合、Windows ログオン後に VPN プラグインはトンネルを確立しません。

- NetScaler Gateway アプライアンスが常時オン機能用に構成されている
- アプライアンスは 2 要素認証`off`による証明書ベースの認証用に設定されています。

[NSHELP-23584]

スキーマを参照しているときに、エラーメッセージ`Cannot read property 'type'of undefined`が表示されることがあります。

[NSHELP-21897]

Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは利用できない、リリース 13.0 で導入された追加の拡張機能を活用できます。

[CGOP-19355]

無効な STA チケットによるアプリケーションの起動失敗は、Gateway Insight では報告されません。

[CGOP-13621]

Gateway Insight レポートでは、SAML エラーエラーの [認証タイプ] フィールドの`SAML`ではなく、値`Local`が誤って表示されます。

[CGOP-13584]

高可用性セットアップでは、NetScaler フェイルオーバー時に、NetScaler Console のフェイルオーバー数ではなくストレージリポジトリ数が増加します。

[CGOP-13511]

ブラウザからローカルホスト接続を受け付けると、macOS の Accept Connection ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

[CGOP-13050]

一部の言語では、Citrix SSO アプリ > ホームページのテキストHome Pageが切り捨てられます。

[CGOP-13049]

NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

[CGOP-11830]

Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

[CGOP-7269]

クラスタ展開では、非 CCO ノードでforce cluster syncコマンドを実行すると、ns.log ファイルに重複したログエントリが含まれます。

[CGOP-6794]

負荷分散

高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

[NSLB-7679]

entityofs サービスグループのトラップの ServiceGroupName 形式は次のとおりです。
<service(group)name>?<ip/DBS>?<port>

トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (?) はセパレータとして使用されます。NetScaler は、疑問符（?）付きのトラップを送信します。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

[NSHELP-28080]

その他

高可用性セットアップで強制同期が実行されると、アプライアンスはセカンダリノードでset urlfiltering parameterコマンドを実行します。
その結果、セカンダリノードは、TimeOfDayToUpdateDB パラメータで指定された次のスケジュールされた時刻まで、スケジュールされた更新をスキップします。

[NSSWG-849]

URL フィルタリングのサードパーティベンダーとの接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

[NSHELP-22409]

ネットワーク

NetScaler BLX アプライアンス 13.0 61.x ビルドから 13.0 64.x ビルドにアップグレードすると、BLX 構成ファイルの設定が失われます。その後、BLX 構成ファイルがデフォルトにリセットされます。

[NSNET-17625]

DPDK を搭載した NetScaler BLX アプライアンスの Intel `X710 10G (i40e)`インターフェイスでは、次のインターフェイス操作はサポートされていません:

- 無効化
- 有効化
- リセット

[NSNET-16559]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスは、BLX 構成ファイル（`/etc/blx/blx.conf`）の設定に関係なく、常に共有モードで展開されます。この問題は、Debian ベースの Linux システムにデフォルトで存在する`mawk`が、`blx.conf`ファイルにある awk コマンドの一部を実行しないために発生します。

回避方法:

NetScaler BLX アプライアンスをインストールする前に`gawk`をインストールします。Linux ホスト CLI で次のコマンドを実行して`gawk`をインストールできます。

- apt-get install gawk

[NSNET-14603]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

回避方法:

NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します。

- dpkg –add-architecture i386
- apt-get update
- apt-get dist-upgrade

- apt-get install libc6:i386

[NSNET-14602]

FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

[NSNET-5233]

高可用性設定では、両方のノード間で HA バージョンが一致しない場合、動的ルートはセカンダリノードに同期されません。セカンダリノードのアクセシビリティが動的ルートに依存している場合、セカンダリノードには到達できません。

修正として、HA バージョンが一致しない場合でも、動的ルートはセカンダリノードに同期されます。

[NSHELP-28326]

NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

[NSHELP-21082]

プラットフォーム

高可用性フェイルオーバーは AWS および GCP クラウドでは機能しません。AWS および GCP クラウド、および NetScaler VPX オンプレミスでは、管理 CPU が 100% の容量に達する可能性があります。これらの問題はいずれも、次の条件が満たされた場合に発生します。

1. NetScaler アプライアンスの初回起動時には、プロンプトが表示されたパスワードは保存されません。
2. その後、NetScaler アプライアンスを再起動します。

[NSPLAT-22013]

13.0/12.1/11.1 ビルドから 13.1 ビルドにアップグレードするか、13.1 ビルドから 13.0/12.1/11.1 ビルドにダウングレードすると、一部の Python パッケージが NetScaler アプライアンスにインストールされません。この問題は、次の NetScaler バージョンで修正されています。

- 13.1-4.x
- 13.0-82.31 以降
- 12.1-62.21 以降

NetScaler バージョンを 13.1-4.x から次のバージョンのいずれかにダウングレードすると、Python パッケージはインストールされません。

- 任意の 11.1 ビルド

- 12.1-62.21 およびそれ以前
- 13.0-81.x およびそれ以前

[NSPLAT-21691]

バージョン 13.1 を実行している NetScaler SDX アプライアンスで、バージョン 12.0 XVA で VPX インスタンスをプロビジョニングすると失敗します。

VPX バージョン 12.1 以降のみがサポートされています。SBI をバージョン 13.1 にアップグレードする前に、VPX バージョンをアップグレードします。

[NSPLAT-21442]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、2 番目のノードと CLIP で CLAG MAC の不一致があります。

- CLAG は Mellanox NIC 上に作成されます。
- クラスターと CLAG セットアップに別の VPX インスタンスを追加します。

その結果、VPX インスタンスへのトラフィックが停止します。

[NSPLAT-21049]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、CLIP テーブルと MAC テーブルで MAC アドレスの不一致が原因で最初のノードがダウンします。

- CLAG は Mellanox NIC 上に作成されます。
- クラスタから 2 つ目のノードを削除します。

[NSPLAT-21042]

Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。`rm cloudprofile` コマンドを使用して、プロファイルを削除します。

[NSPLAT-4520]

Azure の高可用性セットアップで、GUI からセカンダリノードにログオンすると、自動スケーリングクラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノード上で設定する必要があります。

[NSPLAT-4451]

ポリシー

処理データのサイズが、設定されているデフォルトの TCP バッファサイズを超えると、接続がハングすることがあります。回避策：TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

[NSPOLICY-1267]

SSL

NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

回避方法:

1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
2. 構成を保存します。

[NSSSL-9572]

Update コマンドは、次の add コマンドでは使用できません。

- Azure アプリケーションの追加
- Azure Key Vault を追加する
- hsmkey オプションで ssl 証明書キーを追加する

[NSSSL-6484]

認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

[NSSSL-6478]

同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

[NSSSL-6213]

HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。エラー:CRL 更新は無効です

[NSSSL-6106]

セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

[NSSSL-4427]

SSL プロファイルの SSL プロトコルまたは暗号を変更しようとすると、不正な警告メッセージ`Warning: No usable ciphers configured on the SSL vserver/service`,が表示されます。

[NSSSL-4001]

期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

[NSSSL-3184]

システム

アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

[NSHELP-21240]

mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

[NSHELP-10972]

gRPC トラフィックの大きなストリームを処理すると、TCP アドバタイズされたウィンドウが指数関数的に増加し、メモリ使用量が増加します。

[NSBASE-15447]

LogStream トランスポートタイプが Insight 用に構成されている場合、クライアント IP とサーバー IP は HDX Insight SkipFlow レコードで反転されます。

[NSBASE-8506]

ユーザーインターフェイス

MQTT リライト機能では、GUI のエクスプレッションエディタを使用してエクスプレッションを削除することはできません。

回避方法:

CLI から MQTT タイプの add または edit action コマンドを使用します。

[NSUI-18049]

NetScaler GUI では、`Dashboard`タブの下にある`Help`リンクが壊れています。

[NSUI-14752]

CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

回避方法:

NetScaler GUI または CLI を使用して、IPsec プロファイル、IP トンネル、および PBR ルールを追加して、CloudBridge Connector を構成します。

[NSUI-13024]

GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

[NSUI-6838]

高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

- HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

回避方法:

HA 同期が完了した後（両方のノードが同期成功状態にある）のみ、手動の HA フェールオーバーを連続して実行します。

[NSHELP-25598]

NetScaler アプライアンスのバージョン 13.0-71.x を以前のビルドにダウングレードすると、ファイル権限が変更されたために一部の Nitro API が機能しないことがあります。

回避方法:

`/nsconfig/ns.conf` の権限を 644 に変更します。

[NSCONFIG-4628]

あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

1. NetScaler アプライアンスをいずれかのビルドにアップグレードします。

- 13.0 52.24 ビルド
- 12.1 57.18 ビルド
- 11.1 65.10 ビルド

1. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
2. NetScaler アプライアンスを古いビルドにダウングレードします。

CLI を使用してこれらのシステムユーザーの一覧を表示するには、
コマンドプロンプトで次のように入力します。

```
query ns config -changedpassword [-config <full path of the configuration
 file (ns.conf)>]
```

回避方法:

この問題を修正するには、次の独立したオプションのいずれかを使用します。

- NetScaler アプライアンスがまだダウングレードされていない場合（上記の手順のステップ 3)、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
- アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
- 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.htmlを参照してください。

[NSCONFIG-3188]

NetScaler 13.1—9.60 リリースのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリース Build 13.1—9.60 に存在する機能強化と変更、修正された問題と既知の問題について説明します。

メモ

このリリースノートには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正と勧告のリストについては、セキュリティ速報を参照してください。

新機能

ビルド 13.1 ～9.60 で利用できる機能強化と変更。

ボット管理

IP レピュテーションに対する **IPv6** プロトコルのサポート　NetScaler Web App Firewall の IP レピュテーション機能は、ポリシー構成用の IPv6 プロトコルをサポートし、不要な要求を送信する不正な IP アドレスからのセキュリティ保護を強化しました。

IPv6 プロトコルでは、次の脅威カテゴリがサポートされています。

- スパム送信元
- Windows エクスプロイト
- ウェブ攻撃
- ボットネット
- スキャナー
- サービス拒否
- レピュテーション
- フィッシング
- プロキシ
- ネットワーク

- クラウドプロバイダー
- モバイルの脅威
- Tor プロキシー

[NSBOT-585]

Webroot パブリッククラウドサービスプロバイダーのボット署名のカテゴリ

IP レピュテーション技術に基づく NetScaler ボット検出機能が強化され、受信クライアントがパブリッククラウド IP アドレスかどうかを検出できるようになりました。ボット管理機能の設定では、IP レピュテーション機能を有効にする必要があります。NetScaler アプライアンスは、Webroot パブリッククラウドサービスプロバイダーのカテゴリを使用して、ポリシー評価のためにクライアント IP アドレスをクラウドサービスプロバイダーの IP アドレスデータベースに対して検証できます。

ボットプロファイルにバインドできるパブリッククラウドの種類は次のとおりです。

- AWS
- GCP
- Azure
- Oracle
- IBM
- Salesforce

[NSBOT-50]

NetScaler SDX アプライアンス

プールされたライセンスでの **SDX** アプライアンスの復元のサポート　プールされたライセンスを使用している NetScaler SDX アプライアンスの復元に対するサポートが追加されました。ライセンスページも拡張されました。これで、そのページからライセンスを追加および変更できます。

詳しくは、https://docs.citrix.com/en-us/sdx/current-release/configuring-management-service/backup-restore.html%23restore-the-applianceを参照してください:

[NSSVM-4750]

ユーザーは、NetScaler SDX アプライアンスの管理者プロファイルを編集して、ADC インスタンスに新しい資格情報を適用できるようになりました。

詳しくは、https://docs.citrix.com/en-us/sdx/current-release/provision-netscaler-instances.html%23update-an-admin-profileを参照してください:

[NSSVM-4409]

工場出荷時のパーティションからのログが「techsupport」バンドルに含まれるようになり、工場出荷時のリセット履歴をキャプチャできるようになりました。

[NSSVM-2190]

NetScaler Gateway

ホワイトリストに登録された **MAC** アドレスの **EPA** スキャン　式にすべての IP アドレスを一覧表示しなくても、ホワイトリストに登録された MAC アドレスの EPA スキャンを設定できます。代わりに、この構成にパターンセットを使用できます。NetScaler リリース 13.1 より前では、ホワイトリストに登録されたすべての MAC アドレスを EPA 式の一部として指定する必要がありました。

[CGOP-17928]

NetScaler Web App Firewall

セキュリティ保護の強化をサポート　2 つの新しいリラクゼーションカウンタが追加され、次のセキュリティチェックが追加されました。このデータは、構成内の古い緩和を追跡するために使用されます。

- コンテンツタイプの保護
- JSON コマンドインジェクション保護

[NSWAF-6950]

ネットワーク

NetScaler BLX アプライアンスの新しい帯域幅およびサブスクリプションベースのローカルライセンス
NetScaler BLX アプライアンスでは、次の帯域幅ベースのサブスクリプションベースのローカルライセンスが利用可能になりました。

- NetScaler VPX/BLX サブスクリプション 10 Mbps スタンダード、アドバンスト、プレミアムエディション
- NetScaler VPX/BLX サブスクリプション 100 Gbps スタンダード、アドバンスト、プレミアムエディション

詳しくは、https://docs.citrix.com/en-us/citrix-adc-blx/current-release/licensing-blx.htmlを参照してください:

[NSNET-21527]

NetScaler BLX アプライアンスでのメトリックコレクターのサポート

NetScaler BLX アプライアンスが NetScaler メトリックコレクター機能をサポートするようになりました。

[NSNET-15095]

プラットフォーム

VMware ESX ハイパーバイザーでの **NetScaler** アプライアンスの初回起動時の **NetScaler VPX** 構成のサポート
VMware ESX ハイパーバイザーでの NetScaler アプライアンスの初回起動時に、NetScaler VPX 構成を適用できるようになりました。これにより、特定のセットアップまたは VPX インスタンスが非常に短い時間で起動されることがあります。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/deploying-vpx/install-vpx-on-esx/apply-preboot-userdata-on-esx-vpx.htmlを参照してください。

[NSPLAT-21021]

NetScaler VPX インスタンスでの **VMware ESX 7.0** アップデート **1d** サポート

NetScaler VPX インスタンスは、VMware ESX バージョン 7.0 アップデート 1d（ビルド 17551050）をサポートするようになりました。

[NSPLAT-19667]

ポリシー

サフィックスを削除した **URL** パスを返すポリシー式　NetScaler では、`HTTP.REQ.URL.STRIP_SUFFIX` サフィックスが削除された URL パスを返す新しいポリシー式がサポートされるようになりました。

例:

URL: /testsite/file5.html

HTTP.REQ.URL.STRIP_SUFFIX は、テキストを次のように返します。`/testsite/file5`

[NSPOLICY-825]

システム

マルチパス **TCP** バージョン **1** のサポート　NetScaler アプライアンスは、MPTCP バージョン 0 に対する既存のサポートに加えて、マルチパス TCP（MPTCP）バージョン 1 をサポートするようになりました。MPTCP バージョン 1 のサポートは RFC 8684 に準拠しています。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/system/tcp-configurations.htmlを参照してください。

[NSBASE-9237]

gRPC ヘルスモニタのサポート

NetScaler アプライアンスは、サーバーの gRPC ヘルスステータスをプローブする gRPC ヘルスモニターをサポートするようになりました。gRPC ヘルスモニタは、gRPC サービスの全体的なヘルスまたは特定のサービスのヘルスをチェックします。

ヘルスチェックプロトコルは、HTTP2 モニター設定で grpcHealthCheck、grpcStatusCode、および grpcServiceName を設定することによって実装されます。プロトコルを実装しているクライアントは、サーバーのステータス (正常、正常ではない、不明、またはサービスが実装されていない) を照会し、サーバーはステータスメッセージで応答します。

[NSBASE-6455]

ユーザーインターフェイス

NetScaler BLX チェックインおよびチェックアウトライセンス　NetScaler BLX アプライアンスには、NetScaler Application Delivery Management（ADM）からオンデマンドでライセンスを割り当てることができます。ADM ソフトウェアはライセンスを保存および管理します。ライセンスは、スケーラブルで自動化されたライセンスプロビジョニングを提供するライセンスフレームワークを備えています。

NetScaler BLX アプライアンスは、NetScaler BLX アプライアンスが展開されているときに、NetScaler コンソールからライセンスをチェックアウトできます。NetScaler BLX アプライアンスが削除または破棄されると、アプライアンスはライセンスを NetScaler Console ソフトウェアにチェックバックします。

詳しくは、https://docs.citrix.com/en-us/citrix-adc-blx/current-release/licensing-blx.htmlを参照してください:

[NSCONFIG-5777]

NITRO オートメーションツールの使い方

NetScaler Console サービスコネクトは、Ansible、Terraform、NITRO SDK などの自動化ツールの使用状況をキャプチャするようになりました。

[NSCONFIG-4515]

解決された問題

ビルド 13.1 〜9.60 で対処される問題。

認証、承認、監査

次の条件が満たされると、NetScaler アプライアンスがクラッシュすることがあります。

1. アプライアンスにメモリ不足が加えられています。
2. 監査ログは有効で、INFO レベルに設定されています。
3. ユーザー認証が進行中です。

[NSHELP-29053]

NetScaler アプライアンスが認証用に`SameSite`Cookie 属性とドメイン属性に対して構成されている場合、認証は失敗します。これは、`SameSite` cookie 属性値と Domain 属性がセミコロンで区切られていないために発生します。

[NSHELP-28971]

次の条件を満たすと、NetScaler アプライアンスがクラッシュすることがあります。

1. アプライアンスにメモリ不足が加えられています。
2. SAML は認証方法の 1 つとして設定されています。

[NSHELP-28855]

VPN 仮想サーバが SAML SP として構成されている場合、不正な logout (`/cgi/tmlogout`) URL が返されます。この問題は、SAML メタデータに誤ったログアウト URL が生成されたために発生します。

[NSHELP-28726]

マルチコア環境では、認証、承認、監査 TM 仮想サーバの背後にあるリソースにクライアントブラウザがアクセスできないことがあります。

[NSHELP-28474]

NetScaler 高可用性セットアップでは、同期の問題により、CLI 構成中に一部の認証コマンドが表示されます。

[NSHELP-28448]

フォーム SSO が有効になっている場合、NetScaler アプライアンスは、コンテンツタイプヘッダーとともにフォームを追加することで、バックエンドサーバーからの資格情報要求に応答します。この追加により、ヘッダーが既に存在する場合、ヘッダーが重複します。

[NSHELP-28405]

`DualAuthOrPush.xml` ログインスキーマが使用されている場合、NetScaler アプライアンスはサーバー検証エラーをスローします。

[NSHELP-28063]

NetScaler アプライアンスが 401 ベースの認証用に構成されている場合、`SameSite`Cookie 属性は認証 Cookie に追加されません。

[NSHELP-27764]

RADIUS `invalid credentials` 認証プロセス中にエラーメッセージが表示されることがあります。このエラーは、Google Chrome ブラウザを使用してクライアントデバイスから NetScaler アプライアンスにアクセスすると表示されます。

[NSHELP-27113]

抽出されたグループの識別名が NULL の場合、Active Directory グループの抽出中に NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-26899]

式で Authentication、承認、および auditing.user.Domain が使用されている場合、ログインユーザーに対して正しくない SSO ドメイン名が入力されます。

[NSHELP-26443]

プロキシサーバーで SSO 機能を使用すると、NetScaler アプライアンスで NSB リークが発生する場合があります。

[NSHELP-25492]

キャッシュ

`set cache contentGroup`コマンドで`insertAge`パラメータが有効になっている場合、キャッシュ応答で追加のヘッダー情報が送信されます。

[NSHELP-27772]

キャッシュ制御ブロックで`Max_age`パラメータ値と`s_maxage`パラメータ値が動的に設定されていない場合、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-27758]

次の条件が満たされた場合、NetScaler アプライアンスがクラッシュすることがあります。

- アプライアンスは統合キャッシュからコンテンツを提供しています。
- キャッシュされたコンテンツは再検証されます。
- 異なるクライアントから、同じキャッシュされたオブジェクトに対する新しい要求が ADC に届きます。

[NSHELP-22596]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、SDX ライセンスが猶予期間内にない場合、システムは猶予期間内にないアラームが 1 回だけではなく継続的に生成されます。

[NSHELP-28740]

NetScaler SDX アプライアンスの管理サービスでは、SNMP マネージャーのインターフェイス速度が、ビット/秒ではなく Kbps/Mbps で表示されます。

[NSHELP-28724]

SNMP v2 トラップ宛先のコミュニティストリングは、NetScaler SDX アプライアンスでマスクされます。

[NSHELP-28625]

NetScaler SDX アプライアンスでは、プールされたライセンスの猶予期間（30 日）が経過した後でも、VPX インスタンスのスループットを変更できます。

[NSHELP-28553]

Python バージョンのアップグレードにより、管理サービスの Python SDK の読み込みが構文エラーにより失敗することがあります。

[NSHELP-27897]

NetScaler SDX アプライアンスでは、`Hypervisor Disk Usage High`のアラームをオンにするデフォルト値が 98% に増加します。

[NSHELP-27854]

NetScaler SDX アプライアンスでは、次の一連の条件が満たされると、管理チャネルの一部であるインターフェイスが管理チャネルとともに表示されます。

1. VPX インスタンスはクラスターの一部です。
2. 管理チャネルが作成されます。

[NSHELP-27487]

NetScaler Gateway

SSL VPN ライセンスビットは、GCP マーケットプレイスで VPX に設定されていません。その結果、マーケットプレイスの購読者は GCP で SSL VPN を使用できません。

[NSHELP-29107]

NetScaler アプライアンスが UDP トラフィックの処理中にクラッシュすることがあります。

[NSHELP-28802]

HTTP ルールを含む AppFlow ポリシーが NetScaler Gateway にバインドされている場合、VPN ログオン中に NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-28705]

3G /テザー接続ユーザーの場合、NetScaler Gateway のログオンページを読み込めないことがあります。

[NSHELP-28367]

まれに、解放されたセッションにアクセスすると、転送ログイン中に NetScaler Gateway アプライアンスがクラッシュすることがあります。

[NSHELP-28022]

着信カプセル化セキュリティペイロード（ESP）トラフィックの処理中に NetScaler アプライアンスがクラッシュし、セキュリティアソシエーション（SA）が見つからない。

[NSHELP-27991]

nFactor 認証の最後の要素として SAML が設定されていて、クラシック EPA も設定されている場合、転送ログインで問題が発生することがあります。

[NSHELP-27983]

次の両方の条件を満たすと、NetScaler アプライアンスがクラッシュすることがあります。

- アプライアンスは ICA プロキシモード用に展開されます。
- ICA フローの Gateway Insight 機能が有効になっています。

[NSHELP-27982]

まれに、NetScaler Gateway ポータルページに Internet Explorer ブラウザーで EPA プラグインの［ダウンロード］ボタンが表示されないことがあります。

[NSHELP-27849]

非同期がブロックされ、コンテンツスイッチングポリシー構成を変更すると、NetScaler Gateway アプライアンスがクラッシュする可能性があります。

[NSHELP-27570]

NetScaler アプライアンスが UDP トラフィックの処理中にクラッシュすることがあります。

[NSHELP-27536]

ユーザーの個人用ブックマークファイルを、ある NetScaler Gateway アプライアンスから別のアプライアンスにコピーすることはできません。

[NSHELP-27389]

セッションポリシーで不明な VPN クライアントオプションが設定されている場合、NetScaler Gateway アプライアンスがクラッシュすることがあります。

[NSHELP-27380]

無効なメモリ位置にアクセスすると、NetScaler Gateway アプライアンスがクラッシュすることがあります。

[NSHELP-27343]

Gateway Insight が有効になっていると、ローカルの ns.log ファイルに SSL VPN ログメッセージが殺到するため、NetScaler Gateway アプライアンスが予期せず再起動します。

[NSHELP-27040]

NetScaler Gateway ポータルのローカライズは、Internet Explorer ブラウザーと互換性がありません

[NSHELP-26822]

VPN セッションプロファイルを編集すると、NetScaler Gateway GUI にメッセージ`Invalid IP or Port`が表示されます。

[NSHELP-26722]

グローバル syslog パラメータで syslog サーバを変更しても、`show audit messages` 出力には最新のログは表示されません。

[NSHELP-19430]

NetScaler Web App Firewall

NetScaler Web App Firewall 学習エンジンは、違反が確認された場合にのみフィールド形式のルールを学習します。

[NSWAF-7677]

次の条件が満たされた場合、NetScaler アプライアンスがクラッシュすることがあります。

- Web App Firewall の Cookie プロキシが有効になっています。
- セッション Cookie とパーシステント Cookie は同じ名前です。

[NSHELP-28181]

負荷分散

ユーザーモニターおよび組み込みのモニター関連コマンドのパラメーター値のテキストの間にスペースがある場合、パラメーター値は切り捨てられ、スペースに続くテキストは無視されます。

例:

```
add lb monitor ftp_user USER -scriptName nsftp.pl -scriptArgs `file=
    test.txt;username=NS user;password=test123` -dispatcherIP 127.0.0.1
    -dispatcherPort 3013`
<!--NeedCopy-->
```

この例では、ユーザー名は`NS user`に設定されていますが、`NS`が送信されるだけで、その後のテキストはスペースのために切り捨てられま。

[NSLB-8915]

Autoscale を有効にして GSLB サービスグループを構成すると、VPX プライマリサイトとセカンダリサイトがクラッシュしました。

[NSHELP-28530]

HTTP プローブの監視中に HTTP 応答を送信した後、NSB メモリが解放されないため、HA セットアップの NetScaler アプライアンスは接続を失います。

[NSHELP-28466]

マルチ PE システムでは、システムで何回か障害が発生しても、ドメインベースのグループが UP 状態に回復しないことがあります。この問題は、CLI と内部モニターの競合状態が原因で発生します。

[NSHELP-27965]

場合によっては、show running configuration コマンドを発行すると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-27815]

クラスタセットアップで、1 つ以上のノードがDOWN状態になると、バックアップノードがクラスタノードグループに参加できないことがあります。この障害により、一部の NetScaler 機能が失敗します。

[NSHELP-27664]

パイプライン化された RADIUS 要求を受信すると、NetScaler アプライアンスが応答に適切なパケット識別子を挿入しないことがあります。この問題により、クライアントは無効な応答を受信します。

[NSHELP-27391]

次の条件が満たされると、GSLB 設定が部分的に失われる可能性があります。

- NetScaler アプライアンスが再起動されます。
- ADNS サービスは、リモート GSLB サイトと同じ IP アドレスで設定されます。

[NSHELP-26816]

ネットワークレイテンシが高い複数の GSLB サイトで多数の GSLB サービスを構成すると、リモート GSLB サイトで GSLB サービスのステータスが更新されないことがあります。

[NSHELP-23799]

その他

`add URLF categorization`コマンドはデータベースの更新に失敗し、内部エラーが発生します。

[NSSWG-1315]

次の条件が満たされた場合、処理を再開した後に NetScaler アプライアンスがクラッシュすることがあります。

- SSL 転送プロキシ機能が使用されます。
- SSL 転送プロキシ要求のプロトコル情報は、複数の非同期パケットで受信されます。アプライアンスはパケット処理を一時停止し、リクエストのプロトコル詳細をすべて受信すると再開します。

[NSHELP-28447]

インラインデバイスがカスタムメッセージを送信してからリセットすると、NetScaler アプライアンスはインラインデバイスの応答をクライアントに転送する前に接続をリセットします。

[NSHELP-27676]

ネットワーク

次の条件が満たされると、NetScaler VPX インスタンスがクラッシュすることがあります。

- 多数の FTP データ接続が存在します。
- NetScaler アプライアンスでフェイルオーバーが発生します。
- クライアント側またはサーバー側の NATPCB 接続がクリアされます。

[NSHELP-27816]

高可用性設定では、次の条件が満たされた場合、ダイナミックルーティングが有効な SNIP アドレスはリブート時に VTYSH に公開されません。

- ダイナミックルーティングが有効な SNIP アドレスは、デフォルト以外のパーティション内の共有 VLAN にバインドされます。

修正の一環として、NetScaler アプライアンスは、動的ルーティングが有効な SNIP アドレスをデフォルト以外のパーティションの共有 VLAN にバインドすることを許可しなくなりました

[NSHELP-24000]

プラットフォーム

NetScaler アプライアンスのウォームリブート中に、AWS クラウド内の NetScaler VPX インスタンスがクラッシュします。

[NSPLAT-21979]

ソフトウェアバージョン 13.1 ビルド 4.43 の NetScaler VPX インスタンスは、AWS クラウド内の C5n ファミリーのインスタンスをサポートしていません。

[NSPLAT-21451]

Azure クラウドおよび Microsoft Hyper-V サーバー上の NetScaler VPX インスタンスでは、特定の状況では、Hyper-V 仮想インターフェイスの送信側で輻輳パケットのドロップが発生することがあります。これらのパケットドロップにより、NetScaler アプライアンスからの送信が停止する可能性があります。

[NSHELP-28375]

NetScaler MPX 5900 および MPX 8900 プラットフォームでは、LCD 画面に誤ったプラットフォーム番号が表示される。

[NSHELP-28207]

SDX プラットフォームのステータスは LOM コンソールに UNKNOWN と表示されます。これは表示上の問題であり、機能への影響はありません。

[NSHELP-20009]

ポリシー

FIX サービスタイプをレイヤ 2 およびレイヤ 3 モードで使用すると、NetScaler がクラッシュすることがあります。

[NSHELP-28468]

非 TCP ベースのプロトコルで MATCHES () 式が使用されている場合、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-26062]

SSL

メモリ割り当ての失敗により、証明書とキーのペアの追加が失敗することがあります。その結果、CA 証明書とキーのペアの検索が失敗し、アプライアンスがクラッシュします。

[NSHELP-28197]

SSL 仮想サーバーで非同期ポリシーが構成されている場合、NetScaler MPX プラットフォームで SSL ハンドシェイクの再ネゴシエーションが失敗することがあります。

[NSHELP-27870]

NetScaler アプライアンスは、コンテンツ長の HTTP ヘッダーがない場合、OCSP 応答を受け入れません。

[NSHELP-27039]

CRL を発行した CA 証明書名は 32 文字に切り捨てられます。ただし、証明書キーの名前は最大 64 文字です。この問題は、CRL フィールドの文字数が 32 文字に制限されているために発生します。

[NSHELP-26986]

NetScaler MPX/SDX 14000 FIPS アプライアンスで、EDT データグラムサイズが 1 K を超える EDT 構成を使用すると、メモリリークが発生することがあります。

[NSHELP-25375]

システム

NetScaler インスタンスを NetScaler コンソールに登録すると、ADC カウンターにポート割り当てエラーが表示されます。

[NSHELP-28779]

NetScaler バージョン 13.0 ビルド 64-x 以降にアップグレードすると、`Unexpected data received from the server on probe connection for SSL_BRIDGE service type - Server.` メッセージ付きの警告ログが多すぎます。

[NSHELP-28656]

リリース 13.0 ビルド 82.x 以降を実行している NetScaler アプライアンスは、`ns mode pmtud`が有効でパーティションが使用されている場合、クラッシュすることがあります。

[NSHELP-28068]

受信したヘッダーサイズが最大ヘッダーテーブルサイズより大きい場合、アプライアンスはテーブルサイズをゼロにリセットします。その結果、HTTP2 リクエストは数回リクエストすると失敗します。

[NSHELP-27977]

分析プロファイルが参照する AppFlow コレクタポインタが破損しています。

[NSHELP-27924]

キューに保留中のトランザクションがある場合、ADM はメモリ使用率が高いという重大なアラートをランダムに報告します。

[NSHELP-27913]

TCP Zombie timeout は、接続の高速側ではハーフクローズタイムアウトが発生するため、アクティブなサーバーまたはクライアントの接続をフラッシュします。

[NSHELP-27502]

接続チェーン TCP オプションが NetScaler RPC 接続に追加されます。この問題は、GSLB サイト通信との相互運用性の問題を引き起こします。

[NSHELP-27417]

パブリッククラウド MPTCP クラスタの展開では、リンクセットが無効になっていると、パケットの再送信が増加します。

[NSHELP-27410]

NetScaler アプライアンスは、MPTCP 接続で、SACK ブロック、タイムスタンプ、MPTCP データ ACK などの TCP オプションとともに、無効な TCP パケットを送信することがあります。

[NSHELP-27179]

NSWL クライアントは、パケットエンジン（PE-0）からデータを複数回ログに記録することがありますが、他のパケットエンジンからのログはスキップされます。

[NSHELP-27138]

次の条件が満たされた場合、NetScaler アプライアンスがクラッシュすることがあります。

- Logstream メタデータレコードを処理する場合。
- AppFlow 機能が有効になっている。

[NSHELP-26942]

NetScaler アプライアンスとデータローダーで Logstream レコードの不一致が見られます。

[NSHELP-25796]

ユーザーインターフェイス

仮想サーバーの場合、NetScaler GUI（バージョン 13.1 ビルド 4.43）の［トラフィック設定］でパラメーターを編集すると、次のエラーメッセージが表示されます。

```
Invalid argument [pq]
```

[NSHELP-29492]

`ns.conf` ファイルを読み取る操作を実行すると、次の問題が発生します。例: `show ns saved config`。

- HTTPD プロセスがフリーズし、GUI と NITRO API にアクセスできなくなる可能性があります。

[NSHELP-28249]

ADC GUI で RPC ノードの secure オプションを選択解除すると、次のエラーメッセージが表示されます。

引数の前提条件がありません [validateCert、セキュア == はい]

[NSHELP-28239]

クラスタ設定では、次の理由により、設定の同期プロセス中に 2 つ以上のパスワードを持つシングルトンエンティティまたはグローバルエンティティがノードで失敗することがあります。

- シーケンスの最初のパスワードをスキップすると、同期中のノードで後続のパスワード復号化が失敗します。同期中のノードには存在しない CCO ローカルキーが検索されるため、復号化は失敗します。

[NSHELP-28035]

高可用性セットアップまたはクラスタセットアップをリリース 13.0 ビルド 74.14 以降にアップグレードすると、次の理由で設定の同期が失敗することがあります。

- `ssh_host_rsa_key`秘密鍵と公開鍵はどちらも正しくないペアです。

[NSHELP-27834]

高可用性設定では、次の条件が満たされると、システムユーザー認証プロセス中に NetScaler アプライアンスがクラッシュすることがあります。

- パスワードハッシュの計算には、ハートビートが 5 回欠落するまでに時間がかかります。

[NSHELP-27066]

負荷分散サーバーの統計情報が、NetScaler GUI ダッシュボードでずれている。

[NSHELP-20752]

ボットプロファイルからレート制限 URL をバインド解除すると、内部データベースエラーが発生します。

[NSCONFIG-6231]

NetScaler アプライアンスは、NITRO API 呼び出しの一部の GSLB および統計パラメーターについて誤ってZeroが返されます。

[NSCONFIG-6104]

CLI カラーモードで有効になっている NetScaler アプライアンスは、CLI 成功テキストメッセージを緑色ではなく白色で表示します。

[NSCONFIG-5689]

NetScaler BLX アプライアンスが NetScaler Console を使用してライセンスされている場合、アプライアンスをリリース 13.0 ビルド 83.x にアップグレードした後にライセンスが失敗することがあります。

[NSCONFIG-4834]

ビデオの最適化

ビデオ最適化機能を有効にすると、メモリ割り当てに失敗したため、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-28752]

既知の問題

リリース 13.1 〜9.60 に存在する問題。

AppFlow

HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

[NSINSIGHT-943]

認証、承認、監査

まれに、ログの位置が正しくないために NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-29267]

マルチコア NetScaler アプライアンスでは、有効期限が切れたときにユーザーパスワードが変更されると、認証、承認、および監査.user.Attribute 式で空の値が返されることがあります。

[NSHELP-28419]

SSO 機能をプロキシサーバーで使用すると、NetScaler アプライアンスでメモリリークが発生する場合があります。

[NSHELP-27744]

NetScaler アプライアンスは、次の両方の条件が満たされるとクラッシュします。

- 電子メール OTP が設定されている
- メールサーバーが応答しない、またはメールサーバーにネットワークの問題がある

[NSHELP-26137]

NetScaler アプライアンスは、重複したパスワードログインの試行を認証せず、アカウントのロックアウトを防ぎます。

[NSHELP-563]

NetScaler GUI のログインスキーマエディター画面に DualAuthPushOrOTP.xml LoginSchema が正しく表示されません。

[NSAUTH-6106]

ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。

```
show adfsproxyprofile <profile name>
```

回避方法:

クラスタ内のプライマリのアクティブな NetScaler に接続し、`show adfsproxyprofile <profile name>` コマンドを実行します。プロキシプロファイルの状態が表示されます。

[NSAUTH-5916]

次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

- [LDAP 到達可能性をテスト] オプションが開きます。
- 無効なログイン認証情報が入力され、送信されます。
- 有効なログイン認証情報が入力され、送信されます。

回避方法:

[LDAP 到達可能性のテスト] オプションを閉じて開きます。

[NSAUTH-2147]

キャッシュ

統合キャッシュ機能が有効で、アプライアンスのメモリが不足している場合、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-22942]

Call Home

プールされたライセンスを使用している NetScaler MPX アプライアンスでは、Call Home の登録が失敗することがある。Call Home が NetScaler サポートサーバーへのアプライアンスの登録に誤ったシリアル番号を使用しているため、登録は失敗します。

[NSHELP-28667]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、CLAG が Mellanox NIC 上に作成されている場合、VPX インスタンスの再起動時に CLAG MAC が変更されます。VPX インスタンスへのトラフィックは再起動後に停止します。これは、MAC テーブルに古い CLAG MAC エントリがあるためです。

[NSSVM-4333]

NetScaler SDX アプライアンスでは、電源、電圧、またはディスクの障害が複数回発生した場合、管理サービスは syslog または電子メール通知を送信しません。

[NSHELP-29443]

NetScaler Gateway

スプリットトンネルが `Reverse`, DNS に設定されている場合、イントラネットドメインの解決は失敗します。

[NSHELP-29371]

TCP SYSLOG 設定を使用した高可用性セットアップでは、HA フェールオーバーまたは設定のクリア操作中にノードがクラッシュすることがあります。

[NSHELP-29251]

NetScaler Gateway ポータルページでは、RFWebUI ポータルテーマによって **RDP** プロキシリンクアイコンは変わりません。

[NSHELP-28974]

サーバー証明書が信頼されていると、サーバー検証コードが失敗することがあります。その結果、エンドユーザーはゲートウェイにアクセスできなくなります。

[NSHELP-28942]

VPN の切断後、DNS リゾルバがホスト名の解決に失敗することがあります。これは、VPN の切断中に DNS サフィックスが削除されるためです。

[NSHELP-28848]

NetScaler Gateway アプライアンスをバージョン 13.0 にアップグレードすると、セッションプロファイルのプロキシ構成が意図したとおりに機能しなくなります。非 HTTP NS プロキシが設定されている場合、プロキシ接続はバイパスされます。

例:

```
add vpn sessionAction-proxy NS -httpProxy 192.0.2.0:24 -sslProxy
192.0.2.0:24
```

この例では、-HttpProxy は意図したとおりに動作しますが、-sslProxy は機能しません。

[NSHELP-28640]

macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

[NSHELP-28551]

クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

[NSHELP-28404]

Windows プラグインは、認証中にクラッシュすることがあります。

[NSHELP-28394]

StoreFront にバックアップ負荷分散仮想サーバー経由でアクセスすると、VPN 仮想サーバーを介した StoreFront へのアクセスが失敗します。

[NSHELP-27852]

既存の ICA セッションに再接続すると、NetScaler Gateway アプライアンスがクラッシュすることがあります。

[NSHELP-27441]

GUI を使用してクラシック認可ポリシーをバインド解除することはできません。ただし、CLI を使用して認証、承認、および監査認可ポリシーのバインドを解除することはできます。

今回の修正により、GUI を使用して承認ポリシーのバインドを解除できるようになりました。

[NSHELP-27064]

次のいずれかの状況が発生すると、NetScaler アプライアンスがクラッシュします。

- syslog アクションはドメイン名で設定され、GUI または CLI を使用して設定をクリアします。
- 高可用性同期はセカンダリノードで行われます。

回避方法:

syslog サーバーのドメイン名の代わりに syslog サーバーの IP アドレスを使用して syslog アクションを作成します。

[NSHELP-25944]

高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

- HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

回避方法:

HA 同期が完了した後（両方のノードが同期成功状態にある）のみ、手動の HA フェールオーバーを連続して実行します。

[NSHELP-25598]

Gateway Insight は、VPN ユーザに関する正確な情報を表示しません。

[NSHELP-23937]

次の条件を満たす場合、Windows ログオン後に VPN プラグインはトンネルを確立しません。

- NetScaler Gateway アプライアンスが常時オン機能用に構成されている
- アプライアンスは 2 要素認証`off`による証明書ベースの認証用に設定されています。

[NSHELP-23584]

スキーマを参照しているときに、エラーメッセージ`Cannot read property 'type'of undefined`が表示されることがあります。

[NSHELP-21897]

Windows ログオン機能の前に常時接続 VPN を使用したい場合は、NetScaler Gateway 13.0 以降にアップグレードすることをお勧めします。これにより、12.1 リリースでは使用できない、リリース 13.0 で導入された追加の拡張機能を適用できます。

[CGOP-19355]

無効な STA チケットによるアプリケーションの起動失敗は、Gateway Insight で報告されません。

[CGOP-13621]

Gateway Insight レポートでは、SAML エラーエラーの [認証タイプ] フィールドの`SAML`ではなく、値`Local`が誤って表示されます。

[CGOP-13584]

高可用性セットアップでは、NetScaler フェイルオーバー中に、NetScaler Console のフェイルオーバー数の代わりにストレージリポジトリ数が増加します。

[CGOP-13511]

ブラウザからローカルホスト接続を受け付けると、macOS の **Accept Connection** ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

[CGOP-13050]

一部の言語では、**Citrix SSO** アプリ **>** ホームページの`Home Page`テキストが切り捨てられます。

[CGOP-13049]

NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

[CGOP-11830]

Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

[CGOP-7269]

クラスタ展開では、非 CCO ノードで`force cluster sync`コマンドを実行すると、ns.log ファイルに重複したログエントリが含まれます。

[CGOP-6794]

NetScaler Web App Firewall

SSL タイプの負荷分散仮想サーバーでボット管理ポリシーが有効になっていると、ボットデバイスのフィンガープリント投稿 URL が失敗することがあります。

[NSHELP-29198]

次のモジュールが有効になっていると、NetScaler アプライアンスがクラッシュすることがあります。

- 高度なセキュリティチェック機能を備えた Web App Firewall
- Appqoe。

[NSHELP-28251]

負荷分散

高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

[NSLB-7679]

ポリシー式にワイルドカードが含まれている`add dns action`コマンドおよび`add location`コマンドでは、増分同期が失敗します。

[NSHELP-29301]

show コマンドと stat コマンドに表示されるサービスグループの状態が矛盾しています。

[NSHELP-28931]

ZONE タイプの DNS レコードが親ドメインで使用できる場合、既存の NS レコードを持つ子ドメインのクエリを実行すると、子ドメイン NS レコードではなく親ドメインの SOA レコードになります。

[NSHELP-28793]

entityofs サービスグループのトラップの ServiceGroupName 形式は次のとおりです。
<service(group)name>?<ip/DBS>?<port>

トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (?) はセパレータとして使用されます。NetScaler は、疑問符（?）付きのトラップを送信します。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

[NSHELP-28080]

その他

高可用性セットアップで強制同期が行われると、アプライアンスはセカンダリノードでset urlfiltering parameterコマンドを実行します。
その結果、セカンダリノードは、TimeOfDayToUpdateDB パラメータで指定された次のスケジュールされた時刻まで、スケジュールされた更新をスキップします。

[NSSWG-849]

64 ビットアーキテクチャと 1 TB のファイルストレージを備えた Linux システムで実行されている NetScaler CPX インスタンスは、証明書とキーファイルをロードできるようになりました。

[NSHELP-28986]

URL フィルタリングのサードパーティベンダーで接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

[NSHELP-22409]

ネットワーク

NetScaler アプライアンスは、次の条件をすべて満たすとクラッシュすることがあります。

- 負荷分散ルートは、アプライアンスのトラフィックドメインに設定されます。
- アプライアンス上で設定のクリア操作が実行されます。

[NSNET-23847]

NetScaler BLX アプライアンス 13.0 61.x ビルドから 13.0 64.x ビルドにアップグレードすると、BLX 構成ファイルの設定が失われます。その後、BLX 構成ファイルがデフォルトにリセットされます。

[NSNET-17625]

DPDK を搭載した NetScaler BLX アプライアンスの Intel X710 10G (i40e)インターフェイスでは、次のインターフェイス操作はサポートされていません:

- 無効化

- 有効化
- リセット

[NSNET-16559]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスは、BLX 構成ファイル（/etc/blx/blx.conf）の設定に関係なく、常に共有モードで展開されます。この問題は、Debian ベースの Linux システムではデフォルトで存在するmawkが、blx.confファイル内に存在するawkコマンドの一部を実行しないために発生します。

回避方法:

NetScaler BLX アプライアンスをインストールする前にgawkをインストールします。Linux ホスト CLI で次のコマンドを実行してgawkをインストールできます。

- apt-get install gawk

[NSNET-14603]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

回避方法:

NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します。

- dpkg --add-architecture i386
- apt-get update
- apt-get dist-upgrade
- apt-get install libc6:i386

[NSNET-14602]

FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

[NSNET-5233]

大規模な NAT44 セットアップでは、次の理由により、SIP トラフィックの受信中に NetScaler アプライアンスがクラッシュすることがあります。

- LSN モジュールは、参照カウントのデクリメント中、またはサービスを削除している間は、サービスを検索しません。

[NSHELP-29134]

NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

[NSHELP-21082]

プラットフォーム

高可用性フェイルオーバーは AWS および GCP クラウドでは機能しません。AWS および GCP クラウド、および NetScaler VPX オンプレミスでは、管理 CPU が 100% の容量に達する可能性があります。これらの問題はいずれも、次の条件が満たされた場合に発生します。

1. NetScaler アプライアンスの初回起動時には、プロンプトが表示されたパスワードは保存されません。
2. 次に、NetScaler アプライアンスを再起動します。

[NSPLAT-22013]

13.0/12.1/11.1 ビルドから 13.1 ビルドにアップグレードするか、13.1 ビルドから 13.0/12.1/11.1 ビルドにダウングレードすると、一部の Python パッケージが NetScaler アプライアンスにインストールされません。この問題は、次の NetScaler バージョンで修正されています。

- 13.1-4.x
- 13.0—82.31 およびそれ以降
- 12.1—62.21 およびそれ以降

NetScaler バージョンを 13.1-4.x から次のバージョンのいずれかにダウングレードすると、Python パッケージはインストールされません。

- 任意の 11.1 ビルド
- 12.1—62.21 およびそれ以前
- 13.0-81.x およびそれ以前

[NSPLAT-21691]

バージョン 13.1 を実行している NetScaler SDX アプライアンスで、バージョン 12.0 XVA で VPX インスタンスをプロビジョニングすると失敗します。

VPX バージョン 12.1 以降のみがサポートされています。SBI をバージョン 13.1 にアップグレードする前に、VPX バージョンをアップグレードします。

[NSPLAT-21442]

NetScaler SDX アプライアンスのクラスタセットアップで、次の条件が満たされると、2 番目のノードと CLIP で CLAG MAC の不一致があります。

- CLAG は Mellanox NIC 上に作成されます。

- クラスターと CLAG セットアップに別の VPX インスタンスを追加します。

その結果、VPX インスタンスへのトラフィックが停止します。

[NSPLAT-21049]

NetScaler SDX アプライアンスでのクラスター設定では、次の条件が満たされると、CLIP テーブルと MAC テーブルで MAC アドレスが一致しないため、最初のノードがダウンします。

- CLAG は Mellanox NIC 上に作成されます。
- クラスタから 2 つ目のノードを削除します。

[NSPLAT-21042]

Azure リソースグループから Autoscale 設定または仮想マシンスケールセットを削除する場合は、NetScaler インスタンスから対応するクラウドプロファイル構成を削除します。`rm cloudprofile` コマンドを使用して、プロファイルを削除します。

[NSPLAT-4520]

Azure の高可用性セットアップでは、GUI を使用してセカンダリノードにログオンすると、Autoscale クラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノードで設定する必要があります。

[NSPLAT-4451]

VMXNET3 ドライバーを使用する NetScaler VPX インスタンスは、インスタンスが次のいずれかの NetScaler ビルドで実行されている場合、ランダムにクラッシュすることがあります。

- NetScaler 13.1 ビルド 4.x
- NetScaler 13.1 ビルド 9.x

[NSHELP-29120]

ポリシー

処理データのサイズが、設定されたデフォルトの TCP バッファサイズを超えると、接続がハングすることがあります。回避策：TCP バッファサイズを、処理する必要があるデータの最大サイズに設定します。

[NSPOLICY-1267]

SSL

NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

回避方法：

1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
2. 構成を保存します。

[NSSSL-9572]

認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

[NSSSL-6478]

同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

[NSSSL-6213]

HSM の種類として KEYVAULT を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。エラー:CRL 更新は無効です

[NSSSL-6106]

セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

[NSSSL-4427]

SSL プロファイルの SSL プロトコルまたは暗号を変更しようとすると、不正な警告メッセージ`Warning: No usable ciphers configured on the SSL vserver/service,`が表示されます。

[NSSSL-4001]

期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

[NSSSL-3184]

高可用性セットアップでは、プライマリノードとセカンダリノード間で証明書タイプが正しく同期されません。

[NSHELP-27589]

システム

NetScaler アプライアンスがクライアントから HTTP/2 GOWAY フレームを受信すると、約束された ID（最後にピアが開始したストリーム識別子）より大きいストリーム ID を持つすべてのストリームが誤ってリセットされます。

[NSHELP-29328]

X-Forwarder ヘッダーは、NetScaler アプライアンスからバックエンドサーバーに送信される一部の要求には追加されません。

[NSHELP-29142]

NetScaler アプライアンスは、次の条件が満たされるとクラッシュします。

- クライアント側の測定オプションは AppFlow アクションで有効になります。
- チャンクヘッダーはパケット境界上にあります。

[NSHELP-29049]

高可用性セットアップでは、次の理由により、セカンダリノードで Admin パーティション設定の HA 同期が失敗します。

- セカンダリノードでの設定の負荷が大きいため、メモリ不足の問題が発生する

[NSHELP-28409]

TCP 接続では、NetScaler アプライアンスは、次の条件がすべて満たされた場合、サーバーから受信した FIN パケットをクライアントに転送するのではなく、ドロップすることがあります。

- TCP バッファリングは有効です。
- サーバは FIN パケットとデータパケットを別々に送信します。

[NSHELP-274]

Pitboss 障害は、再送信キューに大量のパケットをループさせると発生します。

[NSHELP-26071]

アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

[NSHELP-21240]

mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

[NSHELP-10972]

管理者パーティションがある NetScaler アプライアンスでは、デフォルト以外のパーティションで`nstrace`ユーティリティが正しく実行されないことがある

[NSBASE-15738]

gRPC トラフィックの大きなストリームを処理すると、TCP アドバタイズされたウィンドウが指数関数的に増加し、メモリ使用量が増加します。

[NSBASE-15447]

LogStream トランスポートタイプが Insight 用に構成されている場合、HDX Insight SkipFlow レコードでは、クライアント IP とサーバー IP が反転します。

[NSBASE-8506]

ユーザーインターフェイス

NetScaler GUI では、`Dashboard`タブの下にある`Help`リンクが壊れています。

[NSUI-14752]

CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

回避方法:

NetScaler GUI または CLI を使用して、IPSec プロファイル、IP トンネル、および PBR ルールを追加して、CloudBridge Connector を構成します。

[NSUI-13024]

GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

[NSUI-6838]

NetScaler GUI を使用して SSL 証明書を構成または確認するときに、エラー`Directory doesn't exist`が表示されることがあります。この問題は、2 つの連続するドット (`..`) が付いたファイル名が SSL フォルダー/`nsconfig`/`ssl`に存在する場合に発生します。

回避方法:

これらのファイルをフォルダから削除するか、/`nsconfig`/`ssl` フォルダから移動します。

[NSHELP-28589]

高可用性セットアップで、組み込みポリシーパターンセットがプライマリノードで変更された場合、組み込みポリシーパターンセットバインディングの HA 同期が失敗することがあります。

[NSHELP-28460]

サイドパネルビューでリストのページサイズを変更しようとすると、ページがゆがんでしまいます。

[NSHELP-28220]

インターフェイス (-I) オプションを指定した ping または ping6 コマンドは、次のエラーで失敗することがあります。

- **`interface`** `option not supported`

[NSHELP-26962]

管理パーティションの設定で、証明書失効リスト (CRL) ファイルのアップロードと追加が失敗する。

[NSHELP-20988]

NetScaler アプライアンスのバージョン 13.0-71.x を以前のビルドにダウングレードすると、ファイル権限の変更のために一部の NITRO API が機能しないことがあります。

回避方法:

/`nsconfig`/`ns.conf` の権限を 644 に変更します。

[NSCONFIG-4628]

あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

1. NetScaler アプライアンスをいずれかのビルドにアップグレードします。
    - 13.0 52.24 ビルド
    - 12.1 57.18 ビルド
    - 11.1 65.10 ビルド
2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
3. NetScaler アプライアンスを古いビルドにダウングレードします。

CLI を使用してこれらのシステムユーザーの一覧を表示するには、
コマンドプロンプトで次のように入力します。

```
query ns config -changedpassword [-config <full path of the configuration
 file (ns.conf)>]
```

回避方法:

この問題を修正するには、次の独立したオプションのいずれかを使用します。

- NetScaler アプライアンスがまだダウングレードされていない場合（前述の手順の手順 3)、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
- アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
- 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.htmlを参照してください。

[NSCONFIG-3188]

NetScaler 13.1—4.44 リリースのリリースノート

March 20, 2024

このリリースノートドキュメントでは、NetScaler リリース Build 13.1～4.44 に存在する機能強化と変更、修正済みおよび既知の問題について説明します。

メモ

- このリリースノートドキュメントには、セキュリティ関連の修正は含まれていません。セキュリティ関連の修正とアドバイザリのリストについては、セキュリティ速報を参照してください。
- Citrix Secure Access クライアント（以前は Windows 用 NetScaler Gateway プラグインとして知られていました）ビルド 21.9.1.2 以降には、https://support.citrix.com/article/CTX341455の修正が含まれています。Windows ビルド 21.9.1.2 用の NetScaler Gateway プラグインは、NetScaler ビルド 13.1〜4.44 に含まれています。
- ビルド 13.1 〜4.44 以降のビルドでは、https://support.citrix.com/article/CTX330728で説明されているセキュリティの脆弱性に対処します。
- ビルド 4.44 はビルド 4.43 を置き換えます。
- このビルドには、NSHELP-29519 という問題の修正も含まれています。

新機能

ビルド 13.1 〜4.44 で使用できる機能強化と変更点。

認証、承認、監査

Kerberos SSO 認証のルートドメインからツリードメインへのトラバーサルはサポートされています　NetScaler アプライアンスからのバックエンドサーバーの Kerberos SSO 認証中に、ルートドメインからツリードメインへのトラバーサルがサポートされるようになりました。詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/aaa-tm/single-sign-on-types/kerberos-single-sign-on/setup-citrix-adc-single-sign-on.htmlを参照してください。

[NSAUTH-9836]

ボット管理

NetScaler ボット管理に関する詳細なロギング　着信トラフィックがボットとして識別される場合、NetScaler アプライアンスでは、ドメインアドレス、URL、ユーザーエージェントヘッダー、Cookie ヘッダーなどの追加の HTTP ヘッダーの詳細を記録するためのボット詳細ログ機能を構成できるようになりました。ログの詳細は、監視とトラブルシューティングの目的で ADM サーバーに送信されます。詳細ログメッセージは ns.log ファイルには保存されません。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/bot-management/bot-detection.htmlを参照してください：

[NSBOT-273]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスのクラスター形成ページの機能強化　`Add Node to Cluster` ページの GUI で次の変更が加えられました。クラスタに新しいノードを追加するときに、SNIP アドレスを追加するように求めるプロンプトがシステムによって表示されるようになりました。これらの機能強化は、厳密な送信元 IP アドレスチェックのセキュリティ問題に対処します。

- SNIP のオプションフィールドが提供されるようになりました。
- ノードをクラスタ IP アドレス (CLIP) に追加する際に SNIP を動的に作成するための`Add`ボタンも用意されています。

[NSSVM-4170]

NetScaler SDX 管理者は、ロックアウト間隔が期限切れになる前にユーザーのロックを解除できるようになりました。ユーザーがコンソール経由で管理サービスにログインする場合、ロックアウトは適用されません。ロックアウト間隔も秒から分に変更されます。最小値 = 1 分。最大値 = 30 分。

GUI を使用してユーザのロックを解除するには、次の手順を実行します。

1. [設定] **>** [システム] **>** [ユーザ管理] **>** [ユーザ] に移動します。
2. ロックを解除するユーザーを選択します。
3. ロック解除] をクリックします。**CLI** を使用してユーザのロックを解除するには、次の手順を実行します。

コマンドプロンプトで入力します:

```
set systemuser id=`<ID>` unlock=true
<!--NeedCopy-->
```

[NSSVM-4144]

NetScaler Gateway

追加言語のサポート　NetScaler Gateway ユーザーポータルは、ロシア語、韓国語、中国語（繁体字）で利用できるようになりました。

[CGOP-17095]

Gateway Insight の OAuth-OpenID 接続認証のサポート

NetScaler Gateway インサイトは、OAuth-OpenID Connect 認証関連のイベント（ユーザーのログオンの成功と失敗）を報告するようになりました。

詳しくは、https://docs.citrix.com/en-us/citrix-application-delivery-management-software/current-release/analytics/gateway-insight.htmlを参照してください:

[CGOP-16907]

NetScaler Web App Firewall

高度なポリシー式を使用したクライアント **IP** アドレスの抽出　NetScaler アプライアンスは、高度なポリシー式を使用して、HTTP 要求ヘッダー、要求本文、要求 URL からクライアント IP アドレスを抽出します。抽出された値は、監査ログ、セキュリティインサイト、およびクライアントの地理位置情報の計算のために ADM サーバーに送信されます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/bot-management/bot-detection.htmlを参照してください:

[NSWAF-7260]

BOT TPS 検出メカニズムのオプションを有効にする

有効化オプションは、ボットプロファイル設定の各 TPS ボット検出ルールで利用できるようになりました。デフォルトでは、この値は ON です。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/bot-management/bot-detection.htmlを参照してください:

[NSHELP-25777]

負荷分散

コンテンツスイッチング仮想サーバーでの **HTTP** から **HTTPS** へのリダイレクトのサポート　サービスタイプが SSL のコンテンツスイッチング仮想サーバーで、HTTP トラフィックのリダイレクトがサポートされるようになりました。2 つの新しいパラメータ、`HttpsRedirectUrl`および`RedirectFromPort`が`add cs vserver`コマンドに追加されました。`RedirectFromPort`パラメータで指定されたポートに到着するすべての HTTP トラフィックは、`HttpsRedirectUrl`パラメータで指定された URL にリダイレクトされます。`HttpsRedirectUrl`が設定されていない場合、HTTP トラフィックは着信 HTTP 要求のホストヘッダーの値にリダイレクトされます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/ssl/how-to-articles/ssl-config-https-vserver-to-accept-http-traffic.htmlを参照してください:

[NSLB-8224]

保存コンフィグコマンドをリモート **GSLB** サイトに同期するサポート

これで、`save ns config`コマンドをリモートの GSLB サイトに同期できるようになりました。この機能を有効にするために、新しいパラメータ`GSLBSyncSaveConfigCommand`が`set gslb parameter`コマンドに追加されます。`GSLBSyncSaveConfigCommand`を有効にすると、`save ns config`コマンドは別

の GSLB コマンドとして扱われ、リモート GSLB サイトに同期されます。`save ns config`コマンドを同期するには、この`AutomaticConfigSync`オプションを有効にする必要があります。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/global-server-load-balancing/synchronizing-configuration-in-gslb-setup/real-time-synchronization.htmlを参照してください:

[NSLB-7831]

ユーザーモニター用のセキュアなスクリプト引数のサポート

新しいパラメーター、`-secureargs`が`add lb monitor`コマンドに追加されます。このパラメータは、スクリプト引数をプレーンテキスト形式ではなく暗号化された形式で格納します。このパラメータ（ユーザ名やパスワードなど）を使用して、ユーザモニタのスクリプトに関連する機密データを保護できます。スクリプトに関連する機密データについては、パラメーター`-scriptargs`の代わりにパラメーター`-secureargs`を使用することをお勧めします。両方のパラメータを一緒に使用する場合、`-scriptname`で指定されたスクリプトは、`<scriptargs>` `<secureargs>`の順序で引数を受け入れる必要があります。つまり、引数の定義順序を維持して、`<scriptargs>`の最初のいくつかのパラメータと、`<secureargs>`の残りのパラメータを指定する必要があります。Secure 引数は内部ディスパッチャにのみ適用されます。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/load-balancing/load-balancing-custom-monitors/configure-user-monitor.htmlを参照してください:

[NSLB-6314]

ネットワーク

拡張 **ACL** に対する数値型データセットのサポート　NetScaler アプライアンスは、拡張 ACL の数値タイプのデータセットをサポートするようになりました。番号タイプのデータセットを使用して、拡張 ACL ルールの送信元ポートまたは宛先ポート、またはその両方を指定できます。

[NSNET-20235]

IPSet にバインドされた **VIP** アドレスの **RHI** サポート

NetScaler アプライアンスは、次の条件がすべて満たされている場合、IPSet にバインドされた VIP アドレスをカーネルルートとしてアドバタイズします。

- VIP アドレスでは、`host route`オプションが有効になっています。
- IPSet は、マルチ IP 負荷分散仮想サーバーなどの構成にバインドされます。

[NSNET-20209]

ボリュームマウントを使用した **ADM** との **NetScaler CPX** 登録のサポート

NetScaler CPX は、Kubernetes ConfigMaps と Secret によるボリュームマウントを使用した NetScaler Console への登録をサポートするようになりました。NetScaler CPX は、NetScaler CPX のファイルシステムにあるボリュームマウントから派生した構成の詳細を使用して、ADM エージェントへの登録を開始します。

[NSNET-19058]

プラットフォーム

NetScaler VPX インスタンスでの **VMware ESX 7.0** アップデート **2a** サポート　NetScaler VPX インスタンスは、VMware ESX バージョン 7.0 アップデート 2a（ビルド 17867351）をサポートするようになりました。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/deploying-vpx/supported-hypervisors-features-limitations.htmlを参照してください：

[NSPLAT-20104]

ESXi 上の **NetScaler VPX** インスタンスに対する **AMD** プロセッサのサポート

VMware ESXi ハイパーバイザー上の NetScaler VPX インスタンスは、AMD プロセッサをサポートするようになりました。詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/deploying-vpx/install-vpx-on-esx.htmlを参照してください：

[NSPLAT-17853]

Azure マーケットプレイスでの **NetScaler VPX 5000** サブスクリプションのサポート

NetScaler VPX 5000 サブスクリプションプランは、Azure Marketplace でサポートされるようになりました。このサブスクリプションベースのプランでは、次のライセンスが提供されます。

- Standard
- 詳細
- Premium

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/deploying-vpx/deploy-vpx-on-azure.html#citrix-adc-vpx-licensingを参照してください：

[NSPLAT-13663]

ポリシー

高度なポリシー式での **IP** ヘッダーフィールドのサポート　高度なポリシー式により、IP パケットから次のヘッダーフィールドを取得できるようになりました。

- DSCP
- ECN
- TTL
- バージョン
- 識別
- ヘッダーの長さ
- ヘッダーチェックサム
- オプション
- Payload

[NSPOLICY-2441]

NetScaler バージョン **13.1** 以降からの廃止予定の機能の削除

廃止された多数の機能が削除され、NetScaler アプライアンスでは構成できなくなりました。

次のようなものがあります。

- フィルタ機能 (コンテンツフィルタまたは CF とも呼ばれる)-アクション、ポリシー、およびバインディング。
- SPDY、確実接続（SC)、プライオリティキューイング（PQ)、HTTP サービス拒否（DoS)、および HTML インジェクション機能。
- SSL、コンテンツスイッチング、キャッシュリダイレクト、圧縮、およびアプリケーションファイアウォールに関する従来のポリシー。
- コンテンツスイッチングポリシーの`url`パラメータと`domain`パラメータ。
- 負荷分散永続性ルールのクラシック式。
- 書き換えアクションの`pattern`パラメータ。
- 書き換えアクションの`bypassSafetyCheck`パラメータ。
- 高度な定義式の`SYS.EVAL_CLASSIC_EXPR`。
- `patclass`設定エンティティ。
- 高度な定義式で引数のない`HTTP.REQ.BODY`です。
- 高度な定義式の Q および S プレフィックス。
- cmp パラメータ設定の`policyType`パラメータ。(CLI コマンド`set cmp parameter`)。

すでに文書化されているように、変換には`nspepi`ツールを使用できます。このツールは NetScaler アプライアンスバージョン 13.0 または 12.1 で実行する必要があります。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/appexpert/policies-and-expressions/introduction-to-policies-and-exp/classic-policy-deprecation-faq.htmlを参照してください:

また、最新バージョンのツールを使用して、クラシック設定から高度な設定、およびトラフィックドメインから管理パーティションに移行する方法については、https://github.com/citrix/ADC-scriptsを参照してください。

[NSPOLICY-186]

システム

QUIC ブリッジの統計情報の表示　QUIC bridge `stat` コマンドは、QUIC ブリッジ統計情報の詳細なサマリーを提供するようになりました。

[NSBASE-13883]

NetScaler 13.1 以降の廃止予定の機能の削除

次の非推奨の機能とその構成はサポートされなくなり、NetScaler アプライアンスから削除されます。

- SureConnect (SC)
- プライオリティキューイング（PQ）
- HTTP DoS 保護 (HDOSP)
- `HTMLInjection`

別の方法として、SureConnect、プライオリティキューイング、および HTTP DoS 保護に AppQoE を使用し、`HTMLInjection`のクライアント側の測定値を使用することをお勧めします。

詳しくは、https://docs.citrix.com/en-us/citrix-adc/13-1/appexpert/policies-and-expressions/introduction-to-policies-and-exp/classic-policy-deprecation-faq.htmlを参照してください:

[NSBASE-13780]

ユーザーインターフェイス

NITRO コールのバッチ **API** サポート　Citrix ADC アプライアンスが`batchapi`API をサポートするようになりました。`batchapi`API は、1 つのリクエストで複数の NITRO 呼び出しを処理できるため、ネットワークトラフィックを最小限に抑えることができます。`batchapi`を使用して、次の操作を実行できます。

- バッチ API を使用して、複数の異種リソースを同時に作成、更新、および削除できます。
- バッチ API を使用して、複数の異種リソースを取得できます。

[NSCONFIG-4061]

解決された問題

ビルド 13.1 〜4.44 で対処された問題。

認証、承認、監査

LDAP モニターをサービスにバインドすると、NetScaler アプライアンスが Active Directory に不正なパスワードを送信するため、モニターがダウンします。

[NSHELP-27961]

マルチカスケード AD では、最後のカスケードでユーザが見つからない場合、ユーザのアカウントはロックされません。

[NSHELP-27948]

NetScaler アプライアンスが SAML 認証用に構成されている場合、アプライアンスは RSA 以外の証明書を使用するときにコアをダンプします。

[NSHELP-27813]

場合によっては、ロールベースのアクセスが構成されているときに、特定のユーザーの認証要求を処理中に NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-27655]

Azure AD が NetScaler 認証仮想サーバーで OAuth IdP として構成されている場合、ユーザーは Citrix Workspace アプリからログインできません。

[NSHELP-27462]

場合によっては、StoreFront を使用してアプリケーションにアクセスすると、Workspace アプリで SAML 認証が失敗することがあります。

[NSHELP-27338]

場合によっては、要求に認証 Cookie がない場合、認証、承認、および監査 TM 仮想サーバーへの HTTP POST 要求が正しく処理されないことがあります。POST 本文は処理中に失われます。

[NSHELP-27227]

NetScaler アプライアンスは、認証、承認、および監査 TM および 401 Lb ベースのトラフィックを処理中に頻繁にクラッシュします。

[NSHELP-27094]

場合によっては、NetScaler Gateway のユーザー認証および認証、承認、監査-トラフィック管理展開の実行中に、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-26555]

間違った OTP を入力すると、エラーメッセージ`Email Auth failed. No further action to continue`が表示されます。

[NSHELP-26400]

特定のシナリオでは、ポリシー名がイントラネットアプリケーション名よりも長い場合、[認証、承認、および監査グループのバインド] コマンドが失敗することがあります。

[NSHELP-25971]

SAML ID プロバイダー（IdP）として構成された NetScaler アプライアンスは、サービスプロバイダー（SP）に引用符が含まれている場合、リレー状態をサービスプロバイダー（SP）から切り捨てます。

20131]

パスワードの復号化の問題により、ネットワーク接続テストのチェックが失敗しました。ただし、認証機能は正常に動作します。

[NSAUTH-10216]

ボット管理

1 秒あたりのトランザクション (TPS) ボット検出メカニズムでは、CAPTCHA チャレンジ後のレスポンスの取得中に、バックエンドアプリケーションサーバーが 304 レスポンスを返します。

[NSBOT-626]

キャッシュ

高可用性セットアップでは、HA フェールオーバー中に`memLimit`キャッシュパラメータ設定の HA 同期が失敗します。

[NSHELP-28428]

高可用性セットアップでは、プライマリノードがキャッシュされたオブジェクトではなく NULL ポインタにアクセスした後にクラッシュします。

[NSHELP-26967]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスでは、インスタンスがソフトウェアバージョン 13.0-76.x 以前で作成されている場合、インスタンスの復元が失敗することがあります。

[NSHELP-28429]

NetScaler SDX アプライアンスでは、管理サービスが ADC インスタンスの不正なデータ使用を報告します。

[NSHELP-28208]

NetScaler SDX アプライアンスでは、管理サービスコンソールで CLI プロンプトを変更することはできません。

[NSHELP-28030]

NetScaler SDX アプライアンスでは、インベントリで実行されているジョブとスケジューラの増加により、管理サービスが約 80% の高いメモリ使用率を報告することがあります。

[NSHELP-27805]

NetScaler SDX アプライアンスでは、システムファイル（snmpd.conf および ntp.conf）に改行文字が含まれていると、アップグレードが失敗することがあります。

[NSHELP-27713]

NetScaler SDX アプライアンスでは、インベントリで実行されているジョブとスケジューラの増加により、管理サービスが約 80% の高いメモリ使用率を報告することがあります。

[NSHELP-27396]

NetScaler Gateway

最新バージョンにアップグレードすると、RDP セッションの起動に失敗することがあります。

[NSHELP-29519]

カスタムテーマの CSS 属性を編集しようとすると、エラーメッセージが表示されます。

[NSHELP-28648]

評価中にブロック状態になる可能性があるレスポンダーポリシーが仮想サーバーにバインドされている場合、Citrix Workspace へのログオンが失敗します。

[NSHELP-27819]

クライアントレス VPN を使用して NetScaler Gateway アプライアンスにアクセスすると、コアダンプが生成されることがあります。

[NSHELP-27653]

NetScaler Gateway アプライアンスは、サーバーから開始された UDP トラフィックの処理中にクラッシュする可能性があります。

[NSHELP-27611]

ユーザーは、Microsoft Outlook にログインすると、他のユーザーのメールボックスを表示できます。回避策として、多重化を無効にします。

[NSHELP-27538]

`clearconfig`、`kill ica connection`、`stop dtls listener`などの EDT 関連コマンドがアプライアンスによって処理されると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-27398]

NetScaler Gateway アプライアンスは、UDP トラフィックの処理中にクラッシュすることがあります。

[NSHELP-27317]

Syslog ポリシーが仮想サーバーにバインドされ、対応する syslog アクションが変更されると、NetScaler Gateway アプライアンスがクラッシュします。

[NSHELP-27171]

Gateway Insight が有効になっていると、NetScaler ログにログメッセージ`GwInsight: Func=ns_sslvpn_send_app_launch_fail_record Appflow policy evaluation has failed`があふれることがあります。

[NSHELP-26750]

次の両方の条件が満たされている場合、構成をクリアしようとすると、NetScaler Gateway アプライアンスがクラッシュします。

- SSL プロファイルと証明書とキーのペアは、デフォルトの TCP モニタにバインドされます。
- 同じデフォルト TCP モニタが syslog アクションにバインドされます。

[NSHELP-26685]

[NetScaler Gateway トラフィックプロファイルの作成] ページでプロキシとして FQDN を入力すると、メッセージ`Invalid Proxy Value`が表示されます。

[NSHELP-26613]

NetScaler GUI を使用して RDP クライアントプロファイルを作成しているときに、次の条件が満たされるとエラーメッセージが表示されます。

- デフォルトの事前共有キー (PSK) が設定されています。
- [RDP クッキー有効期間 (秒)] フィールドの RDP クッキー有効性タイマーを変更しようとしています。

[NSHELP-25694]

SNMP OID は、不正な現在の接続セットを VPN 仮想サーバに送信します。

[NSHELP-25596]

複数の VPN プラグインクライアントが 1800 バイト以上の X.509 証明書を使用してトンネルを設定すると、Citric ADC アプライアンスがクラッシュします。

[NSHELP-25195]

STA サーバにバインドされている VPN 仮想サーバの名前を変更すると、show コマンドを実行すると STA サーバのステータスは DOWN と表示されます。

[NSHELP-24714]

まれに、イントラネット IP（IIP）アドレスが有効で、IIP アドレスへのサーバー起動接続がある場合、NetScaler Gateway アプライアンスがクラッシュすることがあります。

[NSHELP-23819]

`show tunnel global`コマンド出力には、高度なポリシー名が含まれます。以前は、出力には高度なポリシー名が表示されませんでした。

例:

新しい出力:

```
> show tunnel global
Policy Name: ns_tunnel_nocmp Priority: 0

Policy Name: ns_adv_tunnel_nocmp Type: Advanced policy
Priority: 1
Global bindpoint: REQ_DEFAULT

Policy Name: ns_adv_tunnel_msdocs Type: Advanced policy
Priority: 100
Global bindpoint: RES_DEFAULT
Done
>
<!--NeedCopy-->
```

前の出力:

```
> show tunnel global
Policy Name: ns_tunnel_nocmp Priority: 0 Disabled

Advanced Policies:

Global bindpoint: REQ_DEFAULT
Number of bound policies: 1

Done
<!--NeedCopy-->
```

[NSHELP-23496]

ICA 開始/停止イベントに対して RADIUS アカウンティングを構成した場合、ICA 開始の RADIUS アカウンティング要求のセッション ID はすべてゼロで表示されます。

[NSHELP-22576]

NetScaler Web App Firewall

NetScaler クラスタセットアップでは、1 つ以上のノードが NetScaler バージョン 12.0、12.1、または 13.0 ビルド 52.x 以前のビルドからアップグレードされると、ノードの 1 つがクラッシュします。クラッシュは、Web App Firewall クッキーの形式とサイズに互換性がないために発生します。

[NSWAF-7689]

Web App Firewall では、区切り文字としてカンマがある場合、`Cookie-transformation`パラメータは応答側の Cookie 値を分割します。

[NSHELP-28411]

コマンドインジェクション違反が特定の順序で観察され、次の条件が満たされると、NetScaler アプライアンスがクラッシュすることがあります。

- リクエストに複数の Cookie が存在します
- `URLDecodeRequestCookies`機能がオフになっている

[NSHELP-28365]

NetScaler アプライアンスは、Samesite 属性と Web アプリケーションファイアウォール機能が有効になっている HTTP 応答を解析するときに、メモリ使用率が高いことを示すことがあります。

[NSHELP-27722]

Internet Explorer ブラウザは SSL 接続を再利用しないため、クッキーハイジャック機能では Internet Explorer ブラウザのサポートが制限されています。この制限のため、1 つの要求に対して複数のリダイレクトが送信され、最終的に Internet Explorer ブラウザで`MAX REDIRECTS EXCEEDED`エラーが発生します。

[NSHELP-27193]

NetScaler バージョン 13.0 ビルド 76.29 にアップグレードし、アプライアンスでファイルアップロード機能を有効にすると、次の問題が発生します。

- SQL およびクロスサイトスクリプティング保護チェックは、すべての Web アプリケーションのファイルアップロードプロセスをブロックします。

[NSHELP-27140]

負荷分散

GSLB セットアップでは、GSLB サイトで統計がクリアされた後、リモートサービスのステータスは更新されません。回避策として、同じ GSLB サイトでもう一度統計をクリアします。リモートサービスのステータスが更新されます。

[NSHELP-28169]

高可用性セットアップでは、次の条件が満たされると、セカンダリノードがクラッシュする可能性があります。

- 両方のノードの物理メモリの容量は、互いに異なります。
- データセッションが正しく同期されていません。

[NSHELP-26503]

クラスタ設定では、GSLB 仮想サーバーバインディングを介してアクセスすると、GSLB サービスの IP アドレスが GUI に表示されません。これは表示上の問題であり、機能に影響はありません。

[NSHELP-20406]

その他

NetScaler アプライアンスは、トンネルまたはサービスの種類（TOS）仮想サーバーが作成されると、追加の L2 情報を追加します。

[NSHELP-27825]

ネットワーク

Debian ベースの Linux ホストで実行されている NetScaler BLX アプライアンス（バージョン 13.0 ビルド 82.x）をアップグレードすると、SSH は共有モードで意図したとおりに動作しません。

[NSNET-23020]

NetScaler BLX アプライアンスをリリース 13.1 ビルド 4.x にアップグレードした後、Web アプリケーションファイアウォールがコンテンツタイプヘッダーのない要求を誤ってブロックすることがあります。

[NSNET-21415]

NetScaler BLX アプライアンスでは、タグ付きnon-dpdkインターフェイスでバインドされた NSVLAN が期待どおりに動作しないことがあります。タグなしnon-dpdkインターフェイスでバインドされた NSVLAN は正常に動作します。

[NSNET-18586]

NetScaler アプライアンスでは、内部ドライバーレイヤーが誤ったデータバッファを使用し、データが破損し、アプライアンスがクラッシュする可能性があります。

[NSHELP-27858]

修正された問題:

サイドカーとして展開され、複数のネットワークに接続された NetScaler CPX は、宛先サブネットの正しい送信元 IP アドレスを選択できませんでした。

[NSHELP-27810]

高可用性セットアップでは、WAF プロファイルとロケーションファイルの設定で HA 同期が失敗することがあります。

[NSHELP-27546]

次の条件がすべて満たされると、負荷分散構成でパケットループが観察されます。

- 仮想サーバーはポート 80 でリッスンするように構成されており、接続フェールオーバー (connfailover) パラメーターはステートレスに設定されています。
- 仮想サーバは、以下の 2 つの要求パケットを受信します。
  - 送信元ポート = 80

- 宛先ポート = 80 以外の番号
- 宛先 IP アドレス = 仮想サーバーの IP アドレス (VIP)

[NSHELP-22431]

プラットフォーム

ターゲットインスタンスを作成しない場合でも、GCP Console にFailed to create target instanceエラーメッセージが表示されます。この問題は、GCP サービスアカウントに compute.targetInstances.get IAM 権限がない場合に発生します。このリリースから、NetScaler VPX は、VIP スケーリング機能を使用する仮想マシンのターゲットインスタンスを作成します。

[NSPLAT-20952]

NetScaler アプライアンスは、NetScaler アプライアンスがライセンスの PPS 制限に達する前であっても、偽のパケット/秒（PPS）レート制限アラートを生成します。

[NSHELP-26935]

ポリシー

グローバルスコープを持つ NS 変数は、HTTP/2 トラフィックでは機能しません。

[NSHELP-27095]

SSL

クラスタ設定では、インストールされた 2 つの証明書が OCSP AIA 拡張を持つ 1 つのサーバ証明書の発行者である場合、サーバ証明書を削除するとアプライアンスは到達不能になります。

[NSHELP-28058]

高可用性セットアップでは、次の両方の条件が満たされると、CRL 自動更新が断続的に失敗します。

- プライマリノードからセカンダリノードにファイルが同期されています。
- CRL ファイルが CRL サーバから同時にダウンロードされています。

[NSHELP-27435]

NetScaler アプライアンスでは、-expiryMonitor を有効にして証明書とキーのペアを追加すると、翌日に誤った証明書の有効期限通知が記録されます。

[NSHELP-27348]

クラスタデータベースでは、クライアント Hello バインドポイントの仮想サーバに SSL ポリシーを異なる優先順位で複数回バインドすると、バインディングは正しく更新されません。その結果、仮想サーバからポリシーをバインド解除した後でも、ポリシーを削除するとエラーが表示されます。

[NSHELP-27301]

構成ファイルの組み込み証明書（`ns-server-certificate`）の名前を変更すると、再起動中に NetScaler アプライアンスがクラッシュします。

[NSHELP-26858]

クラスタのセットアップでは、次の問題が発生する可能性があります。

- CLIP の SSL 内部サービスにバインドするデフォルトの証明書とキーのペアに対するコマンドがありません。ただし、古いビルドからアップグレードする場合は、デフォルトの証明書とキーのペアを、CLIP 上の影響を受ける SSL 内部サービスにバインドする必要があります。
- 内部サービスに対するデフォルトの set コマンドの CLIP とノード間の設定の不一致。
- ノードで実行される show running config コマンドの出力で、SSL エンティティに対するデフォルトの暗号バインドコマンドがありません。省略は表示上の問題にすぎず、機能への影響はありません。バインドは、`show ssl <entity> <name>` コマンドを使用して表示できます。

[NSHELP-25764]

システム

NetScaler アプライアンスは、ICAP オプション応答でクラッシュすることがあります。この問題は、許可されたヘッダー値に 204 以外の値が含まれている場合に発生します。

[NSHELP-27879]

AppFlow では、フローレコードのレイヤー 4 バイトカウントが HTTP 仮想サーバトランザクションと一致しません。カウント値は、レイヤー 7 仮想サーバのバイトカウント値より小さくなっています。

[NSHELP-27495]

NetScaler アプライアンスが NetScaler コンソールに登録されている場合、tcpCurClientConn カウンターに大きな値が表示されます。

[NSHELP-27463]

AppFlow 機能が無効になってから有効に戻ると、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-27236]

まれに、NetScaler アプライアンスがバックエンドサーバーからクライアントを転送するときに、誤った TCP SACK シーケンス番号をクライアントに送信することがあります。この問題は、TCP プロファイルで TCP 選択的 ACK（SACK）オプションが有効になっている場合に発生します。

[NSHELP-24875]

`HTTP.REQ.*`式を含むポリシーが`HTTP_QUIC`仮想サーバーの RESPONSE バインドポイントにバインドされると、NetScaler アプライアンスがクラッシュすることがあります。同じポリシーを仮想サーバーとともに HTTP または SSL タイプの`HTTP_QUIC`仮想サーバーにバインドすると、この問題は発生しません。

[NSBASE-14612]

ユーザーインターフェイス

圧縮ポリシーマネージャ GUI で、関連するバインドポイントと接続タイプを指定して、圧縮ポリシーを HTTP プロトコルにバインドできない。

[NSUI-17682]

`show systemfile`コマンドを使用して ADC インスタンスからファイルの内容をフェッチすると、ADC コンソールにダウンロード失敗のエラーメッセージが表示されます。この問題は、ファイルの内容が NULL バイトで始まる場合に発生します。

[NSHELP-28227]

`admautoregd`SYSLOG フラッドは、内部システムの問題
(Python バイナリファイルが見つからない）が原因で、カスタマーリソース定義（CRD）の誤分類と誤診断につながります。

修正:Python バイナリがまだ見つからない場合、30 分後に`admautoregd`プロセスの監視を停止します。

[NSHELP-28185]

KEK を使用して構成された AWS 上の VPX インスタンスを、NetScaler リリース 13.0 ビルド 76.x 以降にアップグレードすると、構成が失われる可能性があります。リブート後に設定がロードされると、KEK を使用して暗号化された機密データはすべて失敗します。

[NSHELP-28010]

`create ssl rsakey`や`create ssl cert`など、一部の SSL コマンドの引数内で特殊文字が使用されている場合、追加のバックスラッシュ文字が誤って導入されます。

[NSHELP-27378]

高可用性セットアップでは、次の条件のいずれかが満たされると、HA 同期または HA 伝播が失敗することがあります。

- RPC ノードのパスワードには特殊文字が含まれています。
- RPC ノードのパスワードは 127 文字です (最大文字数)。

[NSHELP-27375]

入力構成ファイルのサイズが非常に大きい場合、`nsconfigaudit`ツールがクラッシュする可能性があります。

[NSHELP-27263]

NetScaler GUI を使用して、サービスまたはサービスグループを優先負荷分散仮想サーバーにバインドすることはできません。

[NSHELP-27252]

NetScaler アプライアンスでシステムクロックが更新されると、レポート機能が機能しなくなることがあります。

[NSHELP-25435]

NetScaler VPX アプライアンスでは、ライセンスサーバーを追加した後、容量設定操作が失敗することがあります。この問題は、サポートされているタイプのチェックインおよびチェックアウト（CICO）ライセンスの数が多いため、フレクセラ関連コンポーネントの初期化に時間がかかるために発生します。

[NSHELP-23310]

ログ式がボットプロファイルにバインドされている場合、`botprofile_logexpression_binding` NITRO API GET 呼び出しは応答を返しません。

[NSCONFIG-5490]

クラスター構成では、Web App Firewall プロファイルをきめ細かなルールでバインドし、`non-fine-graned` ルールを同じ URL にバインドすると、きめ細かなルールがデータベースから削除されます。その結果、クラスタ IP アドレスには細かな粒度のないルールのみが表示されます。

[NSCONFIG-5389]

既知の問題

リリース 13.1 〜4.44 に存在する問題。

AppFlow

HDX Insight は、ユーザーがアクセスできないアプリケーションまたはデスクトップを起動しようとしたことによるアプリケーションの起動失敗を報告しません。

[NSINSIGHT-943]

認証、承認、監査

VPN 仮想サーバが SAML SP として構成されている場合、不正な logout (`/cgi/tmlogout`) URL が返されます。この問題は、SAML メタデータに誤ったログアウト URL が生成されたために発生します。

[NSHELP-28726]

SSO 機能をプロキシサーバーで使用すると、NetScaler アプライアンスでメモリリークが発生する場合があります。

[NSHELP-27744]

まれに、次の条件を満たすと、高可用性セットアップのセカンダリノードがクラッシュすることがあります。

- `aaa groups`または`aaa users`の両方が NetScaler アプライアンスで構成されている。

[NSHELP-26732]

NetScaler アプライアンスは、重複したパスワードログインの試行を認証せず、アカウントのロックアウトを防ぎます。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema が、NetScaler GUI のログインスキーマエディタ画面に正しく表示されません。

[NSAUTH-6106]

ADFS プロキシプロファイルは、クラスタ展開で構成できます。次のコマンドを発行すると、プロキシプロファイルのステータスが誤って空白として表示される。
`show adfsproxyprofile <profile name>`

回避方法:

クラスタ内のプライマリのアクティブな NetScaler に接続し、`show adfsproxyprofile <profile name>` コマンドを実行します。プロキシプロファイルの状態が表示されます。

[NSAUTH-5916]

次の手順を実行すると、NetScaler GUI の［認証 LDAP サーバーの構成］ページが応答しなくなります。

- [LDAP 到達可能性をテスト] オプションが開きます。
- 無効なログイン認証情報が入力され、送信されます。
- 有効なログイン認証情報が入力され、送信されます。

回避方法:

[LDAP 到達可能性のテスト] オプションを閉じて開きます。

[NSAUTH-2147]

キャッシュ

統合キャッシュ機能が有効で、アプライアンスのメモリが不足している場合、NetScaler アプライアンスがクラッシュすることがあります。

[NSHELP-22942]

NetScaler SDX アプライアンス

NetScaler SDX アプライアンスで、ソフトウェアバージョン 12.0 XVA イメージを使用して ADC インスタンスを作成すると失敗します。その結果、インスタンスは到達不能になります。

[NSHELP-28408]

NetScaler Gateway

VPN の切断後、DNS リゾルバがホスト名の解決に失敗することがあります。これは、VPN の切断中に DNS サフィックスが削除されるためです。

[NSHELP-28848]

NetScaler Gateway アプライアンスをバージョン 13.0 にアップグレードすると、セッションプロファイルのプロキシ構成が意図したとおりに機能しなくなります。非 HTTP NS プロキシが設定されている場合、プロキシ接続はバイパスされます。

例:VPN
セッションアクションの追加-プロキシ NS-HttpProxy 192.0.2. 0:24-SSLProxy 192.0.2. 0:24

この例では、-HttpProxy は意図したとおりに動作しますが、-sslProxy は機能しません。

[NSHELP-28640]

macOS キーチェーンにクライアント証明書がない場合、Citrix SSO for macOS のクライアント証明書認証が失敗します。

[NSHELP-28551]

クライアントのアイドルタイムアウトが設定されていると、ユーザーが数秒以内に NetScaler Gateway からログアウトすることがあります。

[NSHELP-28404]

Windows プラグインは、認証中にクラッシュすることがあります。

[NSHELP-28394]

次のいずれかの状況が発生すると、NetScaler アプライアンスがクラッシュします。

- syslog アクションはドメイン名で設定され、GUI または CLI を使用して設定をクリアします。
- 高可用性同期はセカンダリノードで行われます。

回避方法:

syslog サーバーのドメイン名の代わりに syslog サーバーの IP アドレスを使用して syslog アクションを作成します。

[NSHELP-25944]

Gateway Insight は、VPN ユーザに関する正確な情報を表示しません。

[NSHELP-23937]

次の条件が満たされている場合、VPN プラグインは Windows ログオン後にトンネルを確立しません。

- NetScaler Gateway アプライアンスが常時オン機能用に構成されている
- アプライアンスは 2 要素認証`off`による証明書ベースの認証用に設定されています。

[NSHELP-23584]

スキーマを参照しているときに、エラーメッセージ`Cannot read property 'type'of undefined`が表示されることがあります。

[NSHELP-21897]

無効な STA チケットによるアプリケーションの起動失敗は、Gateway Insight で報告されません。

[CGOP-13621]

Gateway Insight レポートでは、SAML エラーエラーの場合、「認証タイプ」フィールドの値`SAML`が誤って`Local`と表示されます。

[CGOP-13584]

高可用性セットアップでは、NetScaler フェイルオーバー中に、NetScaler Console のフェイルオーバー数の代わりにストレージリポジトリ数が増加します。

[CGOP-13511]

ブラウザからローカルホスト接続を受け付けると、macOS の **Accept Connection** ダイアログボックスには、選択した言語に関係なく英語でコンテンツが表示されます。

[CGOP-13050]

一部の言語では、**Citrix SSO** アプリ **>** ホームページの`Home Page`テキストが切り捨てられます。

[CGOP-13049]

NetScaler GUI からセッションポリシーを追加または編集すると、エラーメッセージが表示されます。

[CGOP-11830]

Outlook Web App (OWA) 2013 では、[設定] メニューの [オプション] をクリックすると、重大なエラーダイアログボックスが表示されます。また、ページが応答しなくなります。

[CGOP-7269]

クラスタ展開では、非 CCO ノードで`force cluster sync`コマンドを実行すると、ns.log ファイルに重複したログエントリが含まれます。

[CGOP-6794]

NetScaler Web App Firewall

Web App Firewall 署名 ID 1048 は、NetScaler Gateway ページの読み込みをブロックします。

[NSHELP-29113]

負荷分散

高可用性設定では、プライマリノードのサブスクライバセッションがセカンダリノードに同期されないことがあります。これはまれなケースです。

[NSLB-7679]

失敗したコマンドに ENUM 値が欠落しているため、GSLB サービスグループはモニターの更新を処理できません。

[NSHELP-29050]

GSLB 仮想サーバーが次のように構成されている場合、NetScaler アプライアンスは、予想される GSLB サービス IP アドレスを持つ GSLB ドメインクエリに応答しないことがあります。
永続タイプ:
送信元 IP アドレス負荷分散アルゴリズム:
静的近接バックアップ負荷分散方式：ラウンドトリップタイム (RTT)

[NSHELP-28668]

Autoscale を有効にして GSLB サービスグループを構成すると、VPX プライマリサイトとセカンダリサイトがクラッシュしました。

回避策：GSLB サービスを追加したり、IP ポートを GSLB サービスグループにバインドしたりするときは、
コンテンツスイッチング仮想サーバーなどのダミー仮想サーバーを追加しないでください。

[NSHELP-28530]

HTTP プローブの監視中に HTTP 応答を送信した後、NSB メモリが解放されないため、HA セットアップの NetScaler アプライアンスは接続を失います。

[NSHELP-28466]

`entityofs` サービスグループのトラップの ServiceGroupName 形式は次のとおりです。
`<service(group)name>?<ip/DBS>?<port>`

トラップ形式では、サービスグループは IP アドレスまたは DBS 名とポートで識別されます。疑問符 (?) はセパレータとして使用されます。NetScaler は、疑問符（?）付きのトラップを送信します。フォーマットは NetScaler コンソールの GUI でも同じように表示されます。これは予想される動作です。

[NSHELP-28080]

その他

高可用性セットアップで強制同期が行われると、アプライアンスはセカンダリノードで`set urlfiltering parameter`コマンドを実行します。
その結果、セカンダリノードは、`TimeOfDayToUpdateDB` パラメータで指定された次のスケジュールされた時刻まで、スケジュールされた更新をスキップします。

[NSSWG-849]

IDNA2008 標準ドメインでは、URL セットのパターンマッチングが失敗します。

[NSHELP-28902]

VXLAN で MAC ベース転送（MBF）が有効になっていると、ステートフル TCP セッションが確立されていませんでした。

[NSHELP-27125]

URL フィルタリングのサードパーティベンダーで接続の問題が発生した場合、管理 CPU の停滞により NetScaler アプライアンスが再起動することがあります。

[NSHELP-22409]

ネットワーク

Web アプリケーションファイアウォールプロファイルが高度なセキュリティ保護チェックで構成されている場合、DPDK モードの NetScaler BLX アプライアンスがクラッシュすることがあります。

回避方法：

WAF の高度なセキュリティ保護設定を削除します。

[NSNET-22654]

NetScaler BLX アプライアンス 13.0 61.x ビルドから 13.0 64.x ビルドにアップグレードすると、BLX 構成ファイルの設定が失われます。その後、BLX 構成ファイルがデフォルトにリセットされます。

[NSNET-17625]

DPDK を搭載した NetScaler BLX アプライアンスの Intel `X710 10G (i40e)`インターフェイスでは、次のインターフェイス操作はサポートされていません：

- 無効化
- 有効化
- リセット

[NSNET-16559]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスは、BLX 構成ファイル（`/etc/blx/blx.conf`）の設定に関係なく、常に共有モードで展開されます。この問題は、Debian ベースの Linux システムではデフォルトで存在する`mawk`が、`blx.conf`ファイル内に存在する`awk`コマンドの一部を実行しないために発生します。

回避方法：

NetScaler BLX アプライアンスをインストールする前に`gawk`をインストールします。Linux ホスト CLI で次のコマンドを実行して`gawk`をインストールできます。

- apt-get install gawk

[NSNET-14603]

Debian ベースの Linux ホスト（Ubuntu バージョン 18 以降）では、NetScaler BLX アプライアンスのインストールが次の依存関係エラーで失敗することがあります:

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

回避方法:

NetScaler BLX アプライアンスをインストールする前に、Linux ホスト CLI で次のコマンドを実行します。

```
1 -  dpkg --add-architecture i386
2 -  apt-get update
3 -  apt-get dist-upgrade
4 -  apt-get install libc6:i386
5 <!--NeedCopy-->
```

[NSNET-14602]

FTP データ接続の場合、NetScaler アプライアンスは NAT 操作のみを実行し、TCP MSS ネゴシエーションのパケットに対して TCP 処理は実行しない場合があります。その結果、最適なインターフェイス MTU は接続に対して設定されません。この誤った MTU 設定により、パケットのフラグメンテーションが発生し、CPU のパフォーマンスに影響します。

[NSNET-5233]

高可用性セットアップで 2 つの NetScaler アプライアンスの大規模な NAT 展開では、高可用性構成に`stayprimary`または`staysecondary`オプションが設定されていると、IPsec ALG が正しく機能しないことがあります。

[NSNET-1646]

NetScaler アプライアンスで管理パーティションのメモリ制限が変更されると、TCP バッファリングメモリ制限は管理パーティションの新しいメモリ制限に自動的に設定されます。

[NSHELP-21082]

高可用性（HA）セットアップでは、Gratuitous ARP（GARP）が無効になっていると、HA フェールオーバー後にアップストリームルータがトラフィックを新しいプライマリに転送しないことがあります。

[NSHELP-20796]

プラットフォーム

13.0/12.1/11.1 ビルドから 13.1 ビルドにアップグレードするか、13.1 ビルドから 13.0/12.1/11.1 ビルドにダウングレードすると、一部の Python パッケージが NetScaler アプライアンスにインストールされません。この問題は、次の NetScaler バージョンで修正されています。

- 13.1-4.x

- 13.0—82.31 およびそれ以降
- 12.1—62.21 およびそれ以降

NetScaler バージョンを 13.1-4.x から次のバージョンのいずれかにダウングレードすると、Python パッケージはインストールされません。

- 任意の 11.1 ビルド
- 12.1-62.21 およびそれ以前
- 13.0-81.x およびそれ以前

[NSPLAT-21691]

バージョン 13.1 を実行している NetScaler SDX アプライアンスで、バージョン 12.0 XVA で VPX インスタンスをプロビジョニングすると失敗します。

VPX バージョン 12.1 以降のみがサポートされています。SBI をバージョン 13.1 にアップグレードする前に、VPX バージョンをアップグレードします。

[NSPLAT-21442]

Azure リソースグループからオートスケール設定または仮想マシンスケールセットを削除する場合は、対応するクラウドプロファイル構成を NetScaler インスタンスから削除します。`rm cloudprofile` コマンドを使用して、プロファイルを削除します。

[NSPLAT-4520]

Azure の高可用性セットアップでは、GUI を使用してセカンダリノードにログオンすると、autoscale クラウドプロファイル構成の初回ユーザー (FTU) 画面が表示されます。
回避策: 画面をスキップし、プライマリノードにログオンしてクラウドプロファイルを作成します。クラウドプロファイルは、常にプライマリノード上で設定する必要があります。

[NSPLAT-4451]

VMXNET3 ドライバーを使用する NetScaler VPX インスタンスは、インスタンスが次のいずれかの NetScaler ビルドで実行されている場合、ランダムにクラッシュすることがあります。

- NetScaler 13.1 ビルド 4.x
- NetScaler 13.1 ビルド 9.x

[NSHELP-29120]

ポリシー

処理データのサイズが、設定されたデフォルトの TCP バッファサイズを超えると、接続がハングすることがあります。回避策：TCP バッファサイズを、処理が必要なデータの最大サイズに設定します。

[NSPOLICY-1267]

SSL

NetScaler SDX 22000 アプライアンスと NetScaler SDX 26000 アプライアンスの異機種クラスタでは、SDX 26000 アプライアンスを再起動すると SSL エンティティの構成が失われます。

回避方法:

1. CLIP で、仮想サーバ、サービス、サービスグループ、内部サービスなど、既存および新規のすべての SSL エンティティで SSLv3 を無効にします。例: `set ssl vserver <name> -SSL3 DISABLED`。
2. 構成を保存します。

[NSSSL-9572]

Update コマンドは、次の add コマンドでは使用できません。

```
1 -  add azure application
2 -  add azure keyvault
3 -  add ssl certkey with hsmkey option
4 <!--NeedCopy-->
```

[NSSSL-6484]

認証 Azure Key Vault オブジェクトが既に追加されている場合は、Azure Key Vault オブジェクトを追加できません。

[NSSSL-6478]

同じクライアント ID とクライアントシークレットを持つ複数の Azure Application エンティティを作成できます。NetScaler アプライアンスはエラーを返しません。

[NSSSL-6213]

HSM タイプとして`KEYVAULT`を指定せずに HSM キーを削除すると、次の誤ったエラーメッセージが表示されます。
エラー: `crl refresh disabled`

[NSSSL-6106]

セッションキーの自動更新がクラスタ IP アドレスで無効と誤って表示される。(このオプションは無効にできません。)

[NSSSL-4427]

SSL プロファイルの SSL プロトコルまたは暗号を変更しようとすると、不正な警告メッセージ`Warning: No usable ciphers configured on the SSL vserver/service,`が表示されます。

[NSSSL-4001]

期限切れのセッションチケットは、HA フェールオーバー後、非 CCO ノードと HA ノードで保持されます。

[NSSSL-3184]

リクエストバインドポイントですでにバインドされているポリシーに対してポリシーアクションが`Forward`に設定されている場合、HTTP リクエストの処理中に NetScaler アプライアンスがクラッシュします。

[NSHELP-29115]

システム

NetScaler アプライアンスが HTTP/2 ヘッダーフレームを処理すると、TCP ウィンドウリークが観察されます。

[NSHELP-28475]

クライアントが複数の TCP ストリームがある接続をリセットすると、サーバー側のトランザクションレコードは送信されないため、それらのデータストリームの L4 レコードが失われます。

[NSHELP-28281]

クラスタセットアップでは、`set ratecontrol`コマンドは NetScaler アプライアンスを再起動した後にのみ機能します。

回避方法:

`nsapimgr_wr.sh -ys icmp_rate_threshold=<new value>` コマンドを使用します。

[NSHELP-21811]

アプライアンスがクライアントから max_concurrent_stream 設定フレームを受信しない場合、MAX_CONCURRENT_STREAMS 値はデフォルトで 100 に設定されます。

[NSHELP-21240]

mptcp_cur_session_without_subflow カウンタが誤ってゼロではなく負の値にデクリメントします。

[NSHELP-10972]

LogStream トランスポートタイプが Insight 用に構成されている場合、HDX Insight SkipFlow レコードでは、クライアント IP とサーバー IP が反転します。

[NSBASE-8506]

ユーザーインターフェイス

NetScaler GUI では、`Dashboard`タブの下にある`Help`リンクが壊れています。

[NSUI-14752]

CloudBridge Connector の作成/監視ウィザードが応答しなくなるか、CloudBridge Connector の設定に失敗することがあります。

回避方法:

NetScaler GUI または CLI を使用して、IPSec プロファイル、IP トンネル、および PBR ルールを追加して、CloudBridge Connector を構成します。

[NSUI-13024]

GUI を使用して ECDSA キーを作成する場合、カーブのタイプは表示されません。

[NSUI-6838]

`ns.conf` ファイルを読み取る操作を実行すると、次の問題が発生します。例: `show ns saved config`。

- HTTPD プロセスがフリーズし、GUI と NITRO API にアクセスできなくなる可能性があります。

[NSHELP-28249]

高可用性セットアップでは、次の条件が満たされると、VPN ユーザセッションが切断されます。

- HA 同期の進行中に、2 つ以上の連続した手動の HA フェールオーバー操作が実行される場合。

回避方法:

HA 同期が完了した後（両方のノードが同期成功状態にある）のみ、手動の HA フェールオーバーを連続して実行します。

[NSHELP-25598]

管理パーティションの設定で、証明書失効リスト (CRL) ファイルのアップロードと追加が失敗する。

[NSHELP-20988]

NetScaler アプライアンスのバージョン 13.0-71.x を以前のビルドにダウングレードすると、ファイル権限の変更のために一部の NITRO API が機能しないことがあります。

回避方法:

`/nsconfig/ns.conf` の権限を 644 に変更します。

[NSCONFIG-4628]

あなた（システム管理者）が NetScaler アプライアンスで次の手順をすべて実行すると、システムユーザーがダウングレードされた NetScaler アプライアンスにログインできないことがあります。

1. NetScaler アプライアンスをいずれかのビルドにアップグレードします。
    - 13.0 52.24 ビルド
    - 12.1 57.18 ビルド
    - 11.1 65.10 ビルド
2. システムユーザーを追加するか、既存のシステムユーザーのパスワードを変更し、設定を保存します。
3. NetScaler アプライアンスを古いビルドにダウングレードします。

CLI を使用してこれらのシステムユーザーの一覧を表示するには、
コマンドプロンプトで次のように入力します。

```
query ns config -changedpassword [-config <full path of the configuration
 file (ns.conf)>]
```

回避方法:

この問題を修正するには、次の独立したオプションのいずれかを使用します。

- NetScaler アプライアンスがまだダウングレードされていない場合（前述の手順の手順 3）、同じリリースビルドの以前にバックアップされた構成ファイル（ns.conf）を使用して NetScaler アプライアンスをダウングレードします。
- アップグレードされたビルドでパスワードが変更されていないシステム管理者は、ダウングレードされたビルドにログインし、他のシステムユーザーのパスワードを更新できます。
- 上記のいずれのオプションも機能しない場合、システム管理者はシステムユーザパスワードをリセットできます。

詳細については、「root 管理者パスワードをリセットする方法」を参照してください。

[NSCONFIG-3188]

次の NetScaler アップグレード操作を行うと、ローカルシステムユーザーアカウントのログインに失敗することがあります。

- NetScaler 13.0-83.x ビルドから NetScaler 13.1-4.x ビルドへ
- NetScaler 12.1-63.x ビルドから NetScaler 13.1-4.x ビルドへ
- NetScaler 12.1-63.x ビルドから NetScaler 13.0-82.x ビルドへ

この問題は、次のいずれかの条件を満たすローカルシステムユーザーアカウントに対してのみ発生します。

- アップグレード操作を実行する前に、NetScaler ビルド（13.0-83.x または 12.1-63.x）のローカルシステムアカウントのユーザーパスワードが変更されました。
- アップグレード操作を実行する前に、ローカルシステムユーザーアカウントが NetScaler ビルド（13.0-83.x または 12.1-63.x）に追加されている。

回避方法:

システム管理者は、ログイン失敗の問題が発生しているローカルシステムユーザアカウントのパスワードをリセットできます。

詳細については、「root 管理者パスワードをリセットする方法」を参照してください。

[NSCONFIG-5650]

NetScaler の製品概要

August 15, 2023

このトピックでは、NetScaler アプライアンスの基本機能と構成の詳細について説明します。ネットワーク機器を設置および構成するシステムおよびネットワーク管理者は、この内容を参照してください。

NetScaler を理解する

NetScaler アプライアンスは、アプリケーション固有のトラフィック分析を実行して、Web アプリケーションのレイヤー 4 レイヤー 7（L4–L7）ネットワークトラフィックをインテリジェントに分散、最適化、保護するアプリケーションスイッチです。たとえば、NetScaler アプライアンスは、長寿命の TCP 接続ではなく、個々の HTTP 要求に対する決定を負荷分散します。負荷分散機能は、サーバーの障害を遅らせ、クライアントとの切断を少なくします。ADC の機能は次のように大まかに分類できます。

1. データの切り替え
2. ファイアウォールのセキュリティ
3. 最適化
4. ポリシーインフラストラクチャ
5. パケットフロー

データの切り替え

NetScaler は、アプリケーションサーバーの前に展開すると、クライアント要求の送信方法によってトラフィックの最適な分散が保証されます。管理者は、HTTP または TCP 要求の本文に含まれる情報と、URL、アプリケーションデータタイプ、または Cookie などの L4～L7 ヘッダー情報に基づいて、アプリケーショントラフィックをセグメント化できます。多数の負荷分散アルゴリズムと広範なサーバーヘルスチェックによって、クライアント要求が適切なサーバーに確実に送信されるので、アプリケーションの可用性が向上します。

ファイアウォールのセキュリティ

NetScaler セキュリティと保護は、アプリケーション層攻撃から Web アプリケーションを保護します。ADC アプライアンスでは適正なクライアント要求を許可して、不正な要求をブロックできます。サービス拒否（Denial Of Service：DoS）攻撃に対する防御機能を組み込んでおり、サーバーに大きな負担をかけるアプリケーショントラフィックの適正なサージから保護する機能をサポートしています。組み込まれたファイアウォールは、バッファーオーバーフローの悪用、SQL インジェクション、クロスサイトスクリプト攻撃など、アプリケーション層の攻撃から Web アプリケーションを保護します。また、ファイアウォールは、企業の機密情報と重要な顧客データを保護する、個人情報盗難保護機能を備えています。

最適化

最適化により、Secure Sockets Layer (SSL) 処理、データ圧縮、Client Keep-Alive、TCP バッファリング、静的および動的コンテンツのサーバーからのキャッシュなど、リソースを大量に消費する操作がオフロードされます。これにより、サーバーファーム内のサーバーのパフォーマンスが向上し、アプリケーションの処理速度が上昇します。ADC アプライアンスは、レイテンシが長く、ネットワークリンクの輻輳が原因で発生する問題を軽減する、透過的な TCP 最適化をいくつかサポートしています。これにより、クライアントやサーバーの構成を変更することなく、アプリケーションの配信を迅速化できます。

ポリシーインフラストラクチャ

「ポリシー」は、NetScaler のトラフィックフィルタリングと管理の詳細を定義し、「式」と「アクション」の 2 つの部分で構成されます。式は、ポリシーと一致する要求の種類を定義します。このアクションは、リクエストが式に一致した場合の対処方法を ADC アプライアンスに指示します。たとえば、セキュリティ攻撃の特定の URL パターンを、接続をドロップまたはリセットするように設定されたと照合する式などです。各ポリシーには優先度があり、優先度によってポリシーを評価する順序が決定されます。

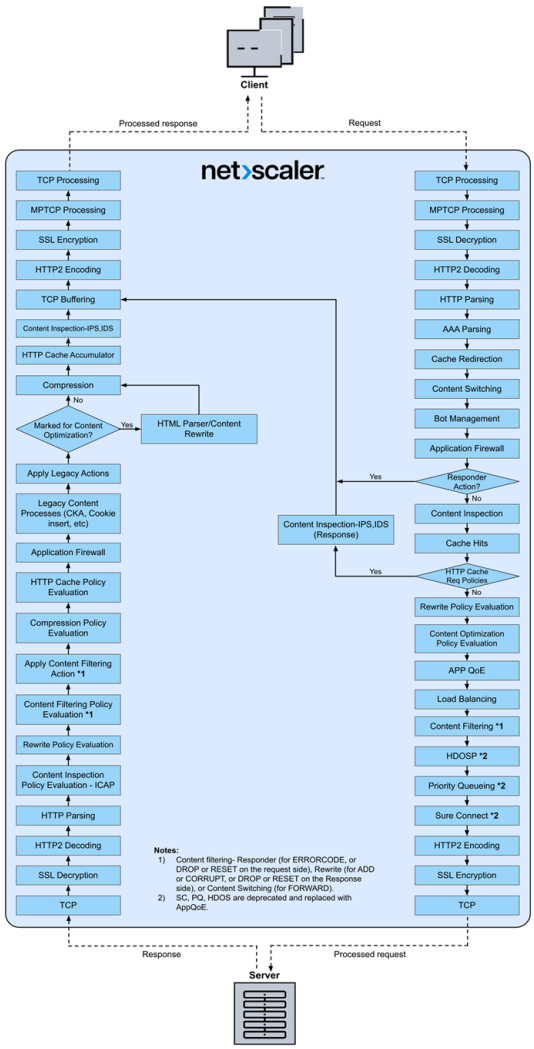
ADC アプライアンスがトラフィックを受信すると、適切なポリシーリストによってトラフィックの処理方法が決定されます。一覧の各ポリシーには 1 つまたは複数の式が含まれており、それらが一緒になって、ポリシーと一致するために接続が満たす必要のある条件を定義します。

rewrite を除くすべてのポリシータイプについて、アプライアンスは要求が一致した最初のポリシーのみを実装します。Rewrite ポリシーの場合、ADC アプライアンスはポリシーを順番に評価し、関連するアクションを同じ順序で実行します。必要な結果を得るには、ポリシーの優先度が重要です。

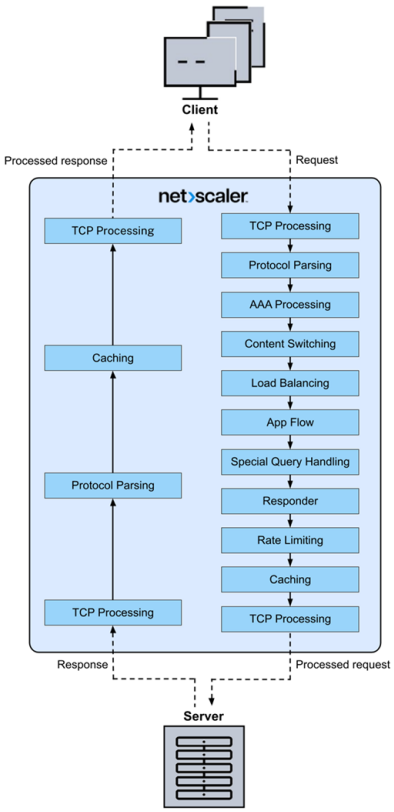
パケットフロー

要件に応じて、複数の機能を設定するように選択できます。たとえば、圧縮と SSL オフロードの両方を構成できます。この場合、発信パケットは圧縮されてから暗号化されて、クライアントに送信されます。

次の図は、NetScaler アプライアンスの HTTP2 パケットフローを示しています。



次の図は、NetScaler アプライアンスのデータストリームクエリ処理フローを示しています。DataStream は、MySQL と MS SQL のデータベースでサポートされています。DataStream 機能について詳しくは、DataStream を参照してください。



> 注**:** トラフィックがコンテンツスイッチング仮想サーバーに対するものである場合、アプライアンスはポリシーを次の順序で評価します。
>
> 1. グローバルオーバーライドにバインドされます。
> 2. 負荷分散仮想サーバにバインドされています。
> 3. コンテンツスイッチング仮想サーバにバインドされています。
> 4. グローバルデフォルトにバインドされます。
>
> このようにして、ポリシールールが true で gotoprioritыexpression が END の場合、それ以降のポリシー評価は停止されます。
> コンテンツスイッチングでは、負荷分散仮想サーバーが選択されていないか、コンテンツスイッチング仮想サーバーにバインドされていない場合、コンテンツスイッチング仮想サーバーにのみバインドされたレスポンダーポリシーが評価されます。

システムの制限

NetScaler ソフトウェア 9.2 以降をインストールする場合、各 NetScaler 機能にはシステム制限があります。詳しくは、Citrix の記事CTX118716を参照してください。

NetScaler アプライアンスはネットワークのどこに適合しますか?

September 12, 2023

NetScaler は、ネットワーク内のクライアントとサーバーの間にあります。クライアントとサーバー間を流れるトラフィックを処理する仲介者の役割を果たします。クライアントからのトラフィックの場合、NetScaler はサーバーとして機能し、要求を受信します。クライアント要求を受信すると、NetScaler はクライアントに代わって新しい要求をサーバーに送信します。要求をサーバーに送信する際、NetScaler はクライアントとして機能します。

NetScaler が適している一般的なネットワーク展開は次のとおりです。

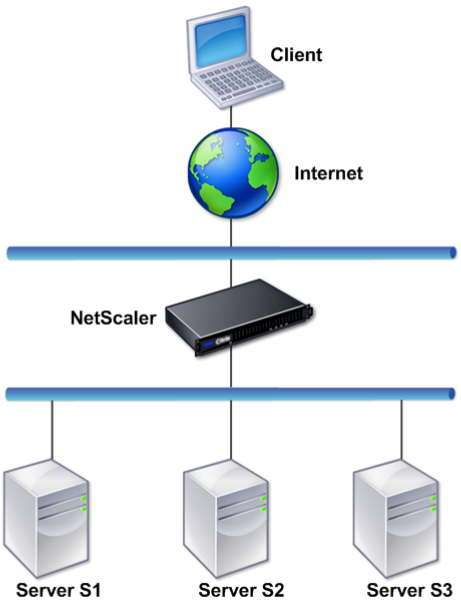
- ゲートウェイ -NetScaler を組織の内部ネットワーク（またはイントラネット）の境界にあるゲートウェイとして使用して、内部ネットワークに存在するサーバー、アプリケーション、およびその他のネットワークリソースへの安全な単一アクセスポイントを提供できます。
- アプリケーションファイアウォール - NetScaler をアプリケーションファイアウォールとして使用して、セキュリティ侵害、データ損失、および機密のビジネス情報や顧客情報にアクセスする Web サイトへの不正な変更を防ぐことができます。そのために、リクエストとレスポンスの両方をフィルタリングし、悪意のあるアクティビティの証拠がないか調べ、そのようなアクティビティを示すリクエストをブロックします。
- ロードバランサー -NetScaler をロードバランサーとして動作させ、クライアント要求を複数のサーバーに分散してリソースの使用率を最適化できます。限られた数のサーバーが多くのクライアントにサービスを提供している現実のシナリオでは、サーバーが過負荷になり、サーバーファームのパフォーマンスが低下する可能性があります。NetScaler アプライアンスは、負荷分散基準を使用して、各クライアント要求を、要求が到着したときに処理するのに最適なサーバーに転送することにより、ボトルネックを防ぎます。
- グローバルサーバーロードバランサー -NetScaler をグローバルサーバーロードバランサー（GSLB）として構成して、災害復旧を実現し、WAN の障害発生点に対してアプリケーションの継続的な可用性を確保できます。GSLB は、クライアントのリクエストを最も近いまたは最もパフォーマンスの高いデータセンター、または障害が発生した場合は存続しているデータセンターに転送することで、データセンター全体の負荷を分散します。
- パケットフォワーダー -NetScaler をパケットフォワーダーとして使用して、NetScaler が所有していない IP にパケットを転送できます。NetScaler はルーターのように動作し、学習したルート、またはパケットを転送するように構成されたルートを確認します。

物理的な展開モード

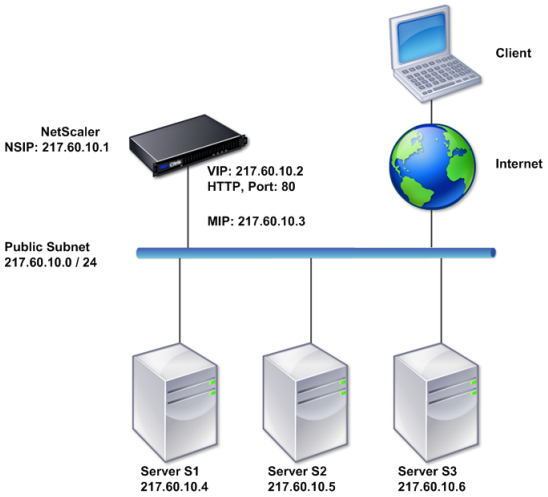
クライアントとサーバーの間に論理的に配置されている NetScaler アプライアンスは、次の 2 つの物理モードのいずれかで展開できます。

- インラインモードまたはツーアームモード
- ワンアームモード

インラインモードでは、アプライアンスは複数のネットワークインターフェースを使用してさまざまなイーサネットセグメントに接続し、クライアントとサーバーの間に位置します。1 つ以上の冗長インターフェースを使用してサーバーネットワークに接続でき、アプライアンスとサーバーの両方を別々のサブネットに配置できます。アプライアンスの L4〜L7 機能を透過的に利用して、サーバーをパブリックネットワーク内に配置し、クライアントがアプライアンスを介してサーバーに直接アクセスするよう構成できます。通常は、実際のサーバーを抽象化した仮想サーバー（後述）を構成します。次の図は、一般的なインライン展開の例を示しています。



ワンアームモードでは、アプライアンスの 1 つのネットワークインターフェイスのみが、Ethernet セグメントに接続されます。この場合のアプライアンスは、ネットワークのクライアント側とサーバー側を分離せずに、構成済みの仮想サーバーを介してアプリケーションへのアクセスを提供します。一部の環境では、ワンアームモードを使用すると、NetScaler ADC の設定に必要なネットワーク変更を簡略化することができます。



インライン（2 アーム）およびワンアーム配置の例については、一般的なネットワークトポロジについてを参照してください。

NetScaler とクライアント/サーバーとの通信方法

August 15, 2023

NetScaler アプライアンスは通常、サーバーファームの前に展開され、クライアント側で構成を変更しなくても、クライアントとサーバーの透過的な TCP プロキシとして機能します。この基本的な動作モードは「Request Switching 技術」と呼ばれ、NetScaler 機能の中核を成しています。Request Switching により、アプライアンスは TCP 接続を多重化してオフロードし、固定接続を維持し、要求（アプリケーションレイヤー）レベルでトラフィックを管理することができます。これらの機能が実現されるのは、アプライアンスが HTTP 要求をその TCP 接続から分離できるからです。

構成によっては、アプライアンスが要求をサーバーに転送する前に、トラフィックを処理する場合があります。たとえば、クライアントがサーバー上の安全なアプリケーションにアクセスしようとする場合に、アプライアンスは必要な SSL 処理を実行してから、トラフィックをサーバーに送信することがあります。

サーバーリソースへの効率的で安全なアクセスを実現するため、アプライアンスは、NetScaler 所有 IP アドレスと呼ばれる IP アドレスのセットを使用します。ネットワークトラフィックを管理するには、NetScaler 所有 IP アドレスを、構成の構築ブロックになる仮想エンティティに割り当てます。たとえば、負荷分散を構成するには、仮想サーバーを作成し、クライアント要求を受信してサービスに配布します。これらのサービスは、サーバー上のアプリケーションとして振る舞うエンティティです。

NetScaler 所有 IP アドレスについて

NetScaler アプライアンスでは、プロキシとして機能するためにさまざまな IP アドレス（「NetScaler 所有 IP アドレス」）が使用されます。主な NetScaler 所有 IP アドレスは、次のとおりです。

- NSIP IP（NSIP）アドレス

  NSIP アドレスは、アプライアンス自体に対する管理アクセスや一般的なシステムアクセス、および高可用性構成のアプライアンス間の通信用の IP アドレスです。

- 仮想サーバー IP（VIP）アドレス

  VIP アドレスは仮想サーバーに関連付けられた IP アドレスです。クライアントが接続するパブリック IP アドレスです。広範なトラフィックを管理するアプライアンスでは、多くの VIP が構成されます。

- サブネット IP（SNIP）アドレス

  SNIP アドレスは、接続の管理とサーバーの監視で使用します。各サブネットに複数の SNIP アドレスを指定できます。SNIP アドレスは VLAN にバインドできます。

- IP セット

  IP セットは、アプライアンス上で SNIP として構成される IP アドレスのセットです。IP セットには、そのセットに含まれる IP アドレスの用途を識別するためのわかりやすい名前を付けます。

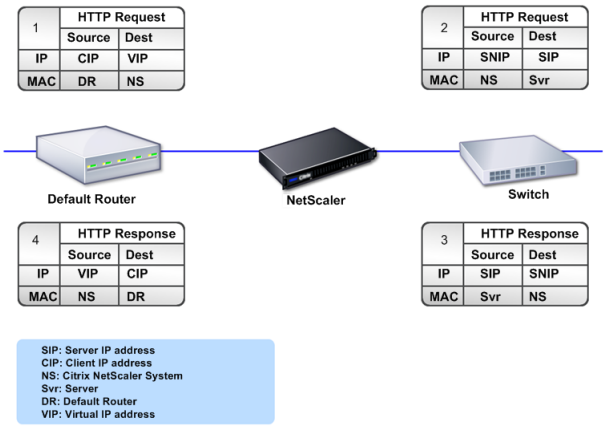
- ネットプロファイル

  ネットプロファイル（ネットワークプロファイル）には、1 つの IP アドレスまたは IP セットが含まれます。ネットプロファイルは負荷分散またはコンテンツスイッチ仮想サーバー、サービス、サービスグループ、またはモニターにバインドされます。アプライアンスが物理サーバーまたはピアと通信するときは、このプロファイルでソース IP アドレスとして指定されているアドレスが使用されます。

トラフィックフローの管理方法

NetScaler アプライアンスは TCP プロキシとして機能するので、IP アドレスを変換してから、パケットをサーバーに送信します。仮想サーバーを構成した場合、クライアントはサーバーに直接接続する代わりに NetScaler アプライアンス上の VIP に接続します。仮想サーバーの設定に基づく判断として、アプライアンスは適切なサーバーを選択し、クライアントの要求をそのサーバーに送信します。デフォルトでは、次の図に示すように、アプライアンスは SNIP アドレスを使用して、サーバーとの接続を確立します。

図 1：仮想サーバーベースの接続



仮想サーバーがない場合、アプライアンスは受信した要求をサーバーへ透過的に転送します。この動作は、透過モードと呼ばれます。透過モードで動作している場合、アプライアンスは、着信したクライアント要求のソース IP アドレスを SNIP アドレスに変換しますが、宛先 IP アドレスは変更しません。このモードが動作するには、L2 または L3 モードが適切に構成されている必要があります。

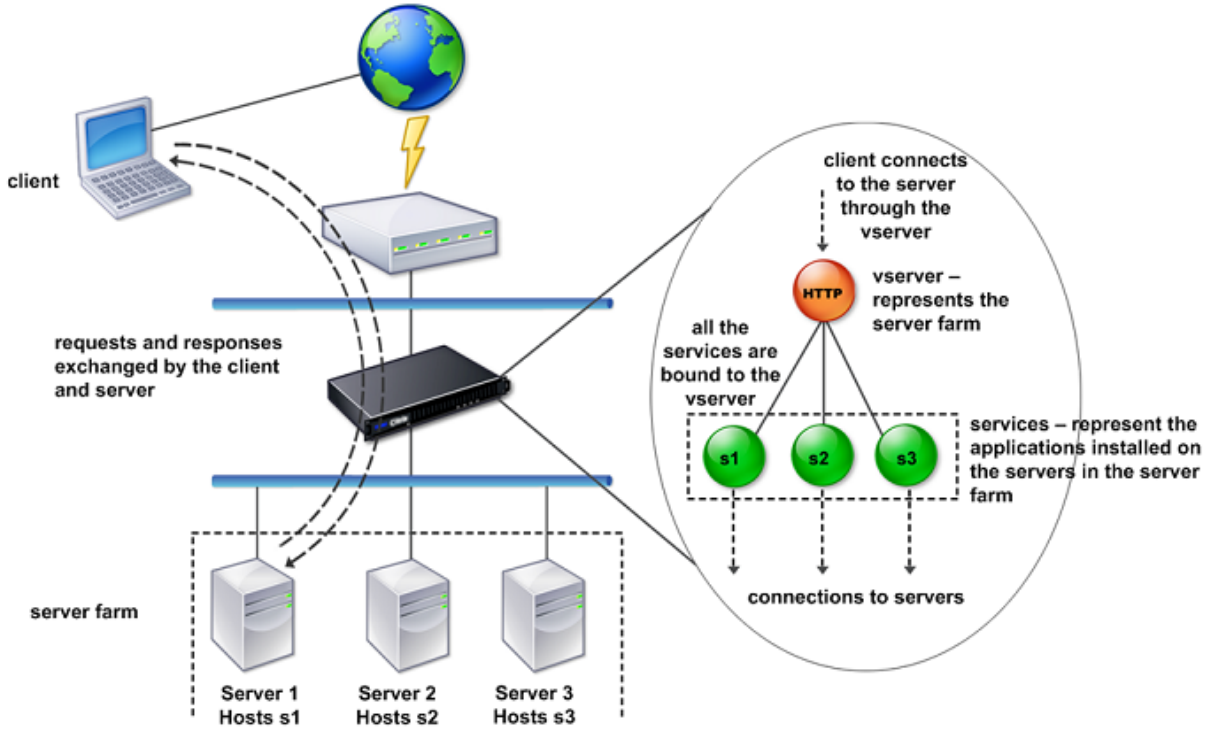
サーバーが実際のクライアント IP アドレスを必要とする場合には、アプライアンスを構成して、クライアント IP アドレスを追加フィールドとして挿入して HTTP ヘッダーを変更するか、またはサーバーとの接続に SNIP ではなくクライアント IP アドレスを使用することができます。

トラフィック管理構築ブロック

通常、NetScaler アプライアンスの構成は、トラフィック管理用の構築ブロックとして動作する一連の仮想エンティティで構築されます。この構築ブロックの手法により、トラフィックフローを分離できます。仮想エンティティは抽象型であり、通常、トラフィックを処理するための IP アドレス、ポート、およびプロトコルハンドラーを表しています。クライアントは、これらの仮想エンティティを介して、アプリケーションとリソースにアクセスします。最もよく使用されるエンティティは、「仮想サーバー」と「サービス」です。仮想サーバーはサーバーファームまたはリモートネットワーク内のサーバーグループとして振る舞い、サービスは各サーバー上の個々のアプリケーションとして機能します。

ほとんどの機能とトラフィックの設定値は、仮想エンティティを介して有効化されます。たとえば、特定の仮想サーバー経由でサーバーファームに接続するクライアントへのすべてのサーバー応答が、アプライアンスにより圧縮されるように構成できます。特定の環境に合わせてアプライアンスを構成するには、適切な機能を確認して仮想エンティティの正しい組み合わせを選択し、それらの機能を提供する必要があります。ほとんどの機能は、相互にバインドされた仮想エンティティをカスケードすることで提供されます。この場合の仮想エンティティは、提供されるアプリケーションの最終的な構造に組み込まれるブロックのようなものです。仮想エンティティを追加、削除、変更、バインド、有効化、および無効化して、機能を構成できます。次の図は、ここで説明されている概念を示しています。

図 2: トラフィック管理構築ブロックのしくみ

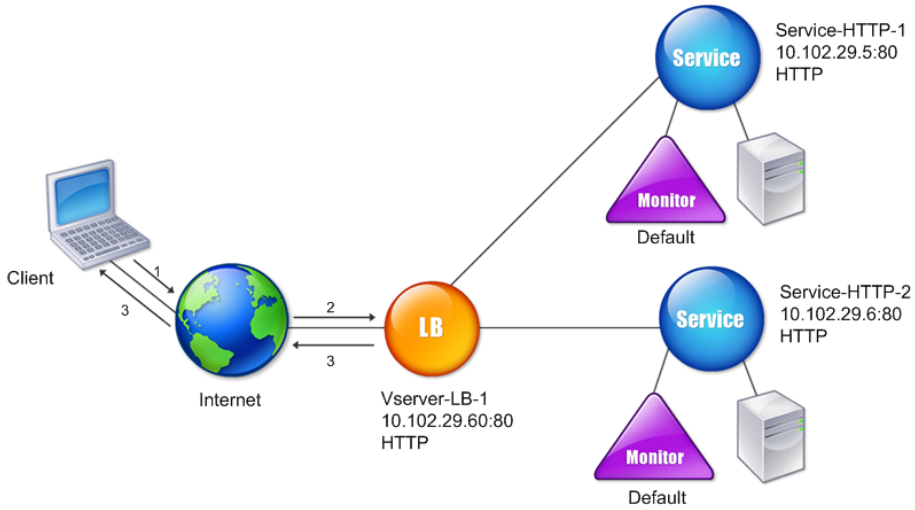


シンプルな負荷分散構成

次の図の例では、NetScaler アプライアンスがロードバランサーとして機能するように構成されています。この構成では、負荷分散に固有の仮想エンティティを構成し、それらを特定の順序でバインドする必要があります。ロードバランサーとして機能する場合、アプライアンスはクライアント要求を複数のサーバー間に分散して、リソース使用率を最適化します。

一般的な負荷分散構成の基本的な構築ブロックは、サービスと負荷分散仮想サーバーです。サービスはサーバー上のアプリケーションとして振る舞い、仮想サーバーはクライアントが接続する単一の IP アドレスを提供してサーバーを抽象化します。クライアント要求がサーバーに送信されるようにするため、各サービスを仮想サーバーにバインドする必要があります。つまり、各サーバーに対してサービスを作成し、サービスを仮想サーバーにバインドする必要があります。クライアントは VIP アドレスを使用して NetScaler アプライアンスに接続します。アプライアンスは VIP アドレスにクライアント要求を受信すると、負荷分散アルゴリズムによって決定されたサーバーに要求を送信します。負荷分散機能は、モニターと呼ばれる仮想エンティティを使用して、特定の構成済みサービス（サーバーおよびアプリケーション）が要求を受信できるかどうかを追跡します。

図 3: 負荷分散仮想サーバー、サービス、およびモニター



負荷分散アルゴリズムを構成するほか、負荷分散構成の動作やパフォーマンスに関する複数のパラメーターを構成できます。たとえば、送信元の IP アドレスに基づいてパーシステンスが維持されるように仮想サーバーを構成できます。この場合、特定の IP アドレスからのすべての要求が同じサーバーに送信されます。

仮想サーバーについて

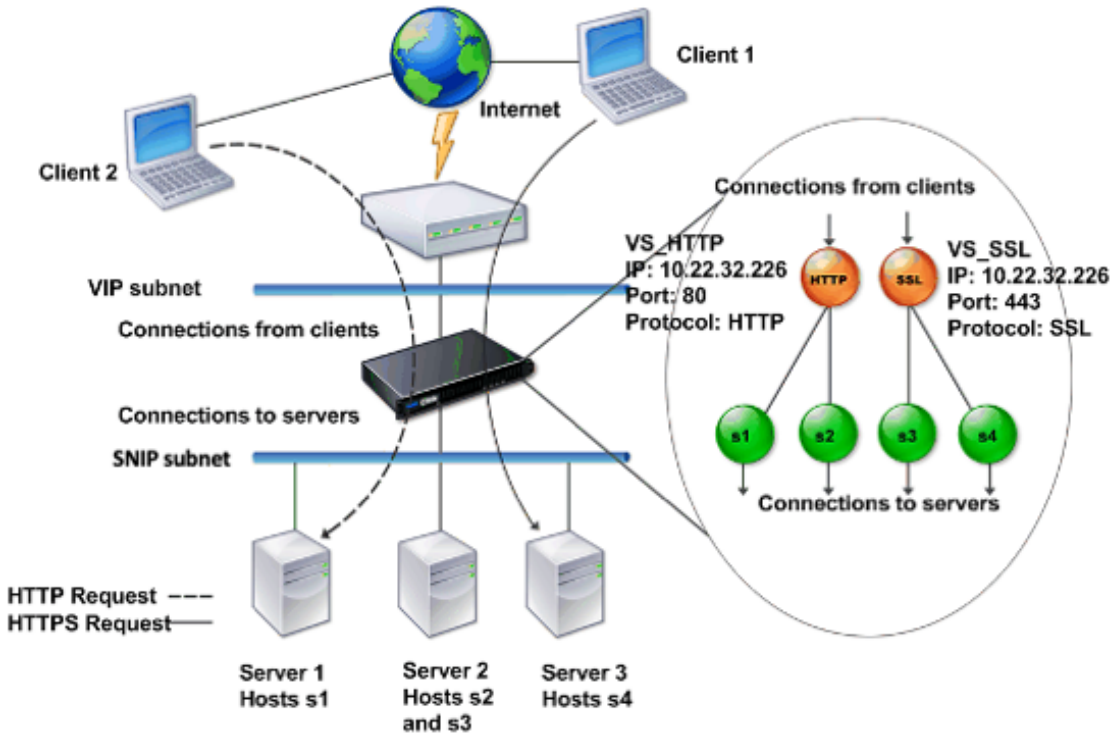
仮想サーバーは名前付きの NetScaler エンティティであり、外部クライアントはそのサーバー上でホストされたアプリケーションにアクセスします。仮想サーバーは英数字名、仮想 IP（VIP）アドレス、ポート、およびプロトコルによって表されます。仮想サーバーの名前はローカル上でのみ意味を持ち、仮想サーバーを識別しやすくするために指定されます。クライアントがサーバー上のアプリケーションにアクセスを試みる場合、クライアントは物理サーバーの IP アドレスではなく、VIP に要求を送信します。アプライアンスが VIP アドレスで要求を受信すると、仮想サーバーでの接続を終了して、クライアントに代わってサーバーとの独自の接続を使用します。仮想サーバーのポートおよびプロトコル設定値によって、その仮想サーバーが振る舞うアプリケーションが決定されます。たとえば、Web サーバーは、ポートとプロトコルがそれぞれ 80 と HTTP に設定された仮想サーバーとサービスによって構成されます。複数の仮想サーバーで同じ VIP アドレスを使用して、異なるプロトコルとポートを使用することもできます。

仮想サーバーは、さまざまな機能の配信ポイントとして動作します。圧縮、キャッシュ、SSL オフロードなどのほとんどの機能は、通常、仮想サーバーで有効になっています。アプライアンスは VIP アドレスで要求を受信すると、要求を受信したポートとそのプロトコルによって、適切な仮想サーバーを選択します。次にアプライアンスは、仮想サ

ーバーに構成されている機能に従って要求を処理します。

ほとんどの場合、仮想サーバーはサービスと協調して動作します。複数のサービスを 1 つの仮想サーバーにバインドすることができます。これらのサービスは、サーバーファーム内の物理サーバーで動作するアプリケーションとして振る舞います。アプライアンスは、VIP アドレスで受信した要求を処理した後、仮想サーバーで設定された負荷分散アルゴリズムの決定に従って、要求をサーバーに転送します。次の図は、これらの概念を示しています。

図 4：単一の VIP アドレスを持つ複数の仮想サーバー



上の図は、VIP アドレスが同じでポートとプロトコルが異なる、2 つの仮想サーバーで構成された環境を示しています。これらの各仮想サーバーには、2 つのサービスがバインドされています。サービス s1 と s2 は VS_HTTP にバインドされており、サーバー 1 とサーバー 2 の HTTP アプリケーションとして動作しています。サービス s3 と s4 は VS_SSL にバインドされており、サーバー 2 とサーバー 3 の SSL アプリケーションとして動作しています（サーバー 2 は、HTTP アプリケーションと SSL アプリケーションの両方を提供します）。アプライアンスが VIP アドレスで HTTP 要求を受信すると、VS_HTTP の設定値により指定されたとして要求を処理し、サーバー 1 またはサーバー 2 に要求を送信します。同様に、アプライアンスが VIP アドレスで HTTPS 要求を受信すると、VS_SSL の設定値により指定されたとして要求を処理し、サーバー 2 またはサーバー 3 に要求を送信します。

仮想サーバーの IP アドレス、ポート番号、またはプロトコルに特定の値を指定せずに、ワイルドカード文字を使用して指定することもできます。このような仮想サーバーは、ワイルドカード仮想サーバーと呼ばれます。たとえば、特定の VIP の代わりにワイルドカード文字を使用し、特定のポート番号で仮想サーバーを構成した場合、アプライアンスは、そのプロトコルおよびポート宛のすべてのトラフィックをインターセプトして処理します。特定の VIP およびポート番号の代わりにワイルドカード文字を使用して仮想サーバーを構成した場合は、そのプロトコルのすべてのト

ラフィックをインターセプトして処理します。

仮想サーバーは、以下のカテゴリに分類できます。

- 負荷分散仮想サーバー

  要求を受信して、適切なサーバーにリダイレクトします。適切なサーバーの選択は、ユーザーが設定したさまざまな負荷分散方式に基づいて行われます。

- キャッシュリダイレクト仮想サーバー

  動的コンテンツに対するクライアント要求を配信元のサーバーにリダイレクトし、静的コンテンツに対するクライアント要求をキャッシュサーバーにリダイレクトします。キャッシュリダイレクト仮想サーバーは、通常、負荷分散仮想サーバーと一緒に動作します。

- コンテンツスイッチ仮想サーバー

  クライアントが要求したコンテンツに基づいて、トラフィックをサーバーに送信します。たとえば、画像に対するすべてのクライアント要求を、画像のみを処理するサーバーに送信するコンテンツスイッチ仮想サーバーを作成できます。コンテンツスイッチ仮想サーバーは、通常、負荷分散仮想サーバーと一緒に動作します。

- VPN（Virtual Private Network：仮想プライベートネットワーク）仮想サーバー

  トンネリングされたトラフィックを復号化して、イントラネットアプリケーションに送信します。

- SSL 仮想サーバー

  SSL トラフィックを受信して復号化し、適切なサーバーにリダイレクトします。適切なサーバーの選択は、負荷分散仮想サーバーの選択と類似しています。

サービスについて

サービスは、サーバー上のアプリケーションとして機能します。通常、サービスは仮想サーバーと組み合わされていますが、仮想サーバーがなくてもアプリケーション固有のトラフィックを管理できます。たとえば、NetScaler アプライアンスで Web サーバーアプリケーションとして振る舞う HTTP サービスを作成できます。この Web サーバーでホストされた Web サイトへのアクセスをクライアントが試みると、アプライアンスが HTTP 要求をインターセプトして Web サーバーとの透過的な接続を作成します。

サービス専用モードでは、アプライアンスがプロキシとして機能します。NetScaler はクライアント接続を終了し、SNIP アドレスを使用してサーバーとの接続を確立し、着信したクライアント要求のソース IP アドレスを SNIP アドレスに変換します。クライアントは要求をサーバーの IP アドレスに直接送信しますが、サーバーは要求が SNIP アドレスから送られてきたものと見なします。アプライアンスは IP アドレス、ポート番号、およびシーケンス番号を変換します。

サービスは、機能を適用するポイントでもあります。SSL Acceleration の例を考えてみましょう。この機能を使用するには、SSL サービスを作成して、そのサービスに SSL 証明書をバインドする必要があります。アプライアンスは HTTPS 要求を受信すると、トラフィックを復号化し、クリアテキストとしてサーバーに送信します。サービス専用モードでは、限られたわずかな機能しか設定できません。

サービスは「モニター」と呼ばれるエンティティを使用して、アプリケーションのヘルスを追跡します。すべてのサービスには、サービスタイプに基づく「デフォルトモニター」がバインドされています。モニターで設定された値に従って、アプライアンスは定期的にアプリケーションにプローブを送信し、アプリケーションの状態を判定します。プローブが失敗した場合、アプライアンスはサービスがダウンしたものとしてマークします。このような場合、アプライアンスは、適切なエラーメッセージでクライアント要求に応答するか、設定された負荷分散ポリシーに従って要求を転送します。

NetScaler 製品ラインの概要

August 15, 2023

NetScaler の製品ラインは、アプリケーションレベルのセキュリティ、最適化、およびトラフィック管理を単一の統合アプライアンス上に集約して、インターネットおよびプライベートネットワーク経由のアプリケーション配信を最適化します。ユーザーはサーバールームに NetScaler アプライアンスを設置し、NetScaler を介してすべての接続を管理対象サーバーにルーティングできます。次に、有効化された NetScaler 機能と設定されたポリシーが、着信および発信トラフィックに適用されます。

NetScaler アプライアンスは、既存の負荷分散装置、サーバー、キャッシュ、およびファイアウォールを補完する目的で、ネットワークに統合できます。クライアント側またはサーバー側にソフトウェアを追加する必要はなく、NetScaler の Web ベースの GUI および CLI 構成ユーティリティを使用して設定することができます。

このセクションでは、以下のトピックについて説明します:

- NetScaler のハードウェアプラットフォーム
- NetScaler のエディション
- NetScaler ADC ハードウェアでサポートされるリリース
- サポートされているブラウザー

NetScaler のハードウェアプラットフォーム

NetScaler ハードウェアは、さまざまなハードウェア仕様を持つさまざまなプラットフォームで利用できます。

NetScaler MPX のハードウェアプラットフォーム

NetScaler SDX のハードウェアプラットフォーム

NetScaler のエディション

NetScaler オペレーティングシステムには、次の 3 つのエディションがあります。

- Standard

- 詳細設定
- Premium

Standard エディションと Advanced エディションでは、使用できる機能が制限されています。すべてのエディションで機能のライセンスが必要です。

NetScaler ソフトウェアエディションの詳細については、NetScaler エディションのデータシートを参照してください。

ライセンスを取得してインストールする方法については、「ライセンス」を参照してください。

NetScaler ADC ハードウェアでサポートされているリリース

すべての NetScaler ハードウェアプラットフォームおよびこれらのプラットフォームでサポートされているソフトウェアリリースについては、次の互換性マトリックス表を参照してください。

NetScaler MPX ハードウェア-ソフトウェア互換性マトリックス

NetScaler SDX ハードウェア-ソフトウェア互換性マトリックス

互換性のあるブラウザー

NetScaler GUI にアクセスするには、ワークステーションに互換性のある Web ブラウザーが必要です。

次の表に、NetScaler GUI バージョン 12.0、12.1、および 13.0 と互換性のあるブラウザーを示します。

オペレーティングシステム	**Web** ブラウザー	バージョン
Windows 7 およびそれ以降	Internet Explorer	11、Edge、およびそれ以降
Windows 7 およびそれ以降	Mozilla Firefox	45 およびそれ以降
Windows 7 およびそれ以降	Chrome	60 およびそれ以降
MAC	Mozilla Firefox	45 およびそれ以降
MAC	Safari	10.1.1 およびそれ以降

NetScaler 11.1 と互換性のあるブラウザーのバージョンは次のとおりです。

オペレーティングシステム	**Web** ブラウザー	バージョン
Windows 7 およびそれ以降	Internet Explorer	8、9、10、11、エッジ
Windows 7 およびそれ以降	Mozilla Firefox	45 およびそれ以降
Windows 7 およびそれ以降	Chrome	60 およびそれ以降

オペレーティングシステム	**Web** ブラウザー	バージョン
MAC	Mozilla Firefox	45 およびそれ以降
MAC	Safari	10.1.1 およびそれ以降

ハードウェアをインストールします

August 15, 2023

NetScaler アプライアンスをインストールする前に、インストール前のチェックリストを確認してください。

SDX アプライアンスを使用するには、表に記載されているリソースに記載されている手順に従って、以下のタスクを完了する必要があります。指定された順序でタスクを完了します。

タスク

説明

1. 安全、注意、警告、およびその他の情報を読む

   製品を設置する前に、知っておく必要のある注意と危険に関する情報をお読みください。

2. インストールの準備

   新しいアプライアンスを設置する前に、アプライアンスを開梱し、すべての部品が納品されたことを確認し、サイトとラックを準備し、基本的な電気安全上の注意事項に従ってください。

3. ハードウェアをインストールします

   アプライアンスをラックマウントし、トランシーバー（使用可能な場合）を取り付け、アプライアンスをネットワークと電源に接続します。

4. アプライアンスを構成します。

   GUI またはシリアルコンソールを使用して、NetScaler アプライアンスの初期設定を構成します。

   これらのタスクを完了するには、次のドキュメントに記載されている手順に従ってください。

   - NetScaler MPX ハードウェアのドキュメント
   - NetScaler SDX ハードウェアのドキュメント

NetScaler アプライアンスへのアクセス

April 15, 2024

NetScaler アプライアンスには、コマンドラインインターフェイス（CLI）と GUI の両方があります。GUI には、アプライアンスを構成するための構成ユーティリティと、ダッシュボードと呼ばれる統計ユーティリティがあります。初回のアクセス用に、すべてのアプライアンスには出荷時にデフォルトの NSIP（NetScaler IP）アドレス 192.168.100.1 とデフォルトのサブネットマスク 255.255.0.0 が割り当てられています。初回構成時に、新しい NSIP アドレスとそのサブネットマスクを割り当てることができます。

複数の NetScaler 装置の展開時に IP アドレスの競合が発生した場合は、以下の点について確認してください。

- ネットワーク上の別のデバイスに既に割り当てられている IP アドレスを、NSIP として選択していないかどうか。
- 複数の NetScaler アプライアンスに同じ NSIP を割り当てていないかどうか。
- NSIP は、すべての物理ポートでアクセス可能です。NetScaler のポートはホストポートであり、スイッチポートではありません。

次の表は、使用可能なアクセス方法の一覧です。

アクセス方法	ポート	デフォルト IP アドレス（必要/不要）
CLI	コンソール	N
CLI と GUI	イーサネット	Y

コマンドラインインターフェイス

CLI には、ワークステーションをコンソールポートに接続してローカルでアクセスすることも、同じネットワーク上の任意のワークステーションから SSH（Secure Shell）を介して接続してリモートアクセスすることもできます。

コンソールポートを使用したコマンドラインインターフェイスへのログオン

アプライアンスには、ワークステーションに接続するためのコンソールポートがあります。アプライアンスにログオンするには、シリアルクロスオーバーケーブルと、端末エミュレーションプログラムを備えたワークステーションが必要です。

コンソールポートを使用して CLI にログオンするには、次の手順を実行します：

1. コンソールポートをワークステーションのシリアルポートに接続します。詳細については、「コンソールケーブルの接続」を参照してください。

2. ワークステーションで、ハイパーターミナルまたはその他のターミナルエミュレーションプログラムを起動します。ログオンプロンプトが表示されない場合は、Enter キーを 1 回または数回押すことが必要な場合があります。
3. [User name] にnsrootを入力します。Password にnsrootを入力し、パスワードが機能しない場合はアプライアンスのシリアル番号を入力します。シリアル番号のバーコードは、アプライアンスの背面にあります。

SSH を使用したコマンドラインインターフェイスへのログオン

SSH プロトコルを使用すると、同じネットワーク上の任意のワークステーションからアプライアンスにリモートアクセスできます。SSH Version 1（SSH1）または SSH Version 2（SSH2）を使用できます。

動作している SSH クライアントがない場合は、以下の SSH クライアントプログラムをダウンロードしてインストールできます。

- PuTTY

  複数のプラットフォームでサポートされている、オープンソースソフトウェアです。次のサイトから入手できます。

  http://www.chiark.greenend.org.uk/~sgtatham/putty/

- Vandyke Software SecureCRT

  Windows プラットフォームでサポートされている、市販のソフトウェアです。次のサイトから入手できます。

  http://www.vandyke.com/products/securecrt/

これらのプログラムは NetScaler チームによってテストされ、NetScaler アプライアンスで正しく動作することが確認されています。そのほかのプログラムも正常に動作する可能性がありますが、テストは実施されていません。

SSH クライアントが正しくインストールされていることを確認するには、対象のクライアントを使用して、SSH 接続を受け付ける、ネットワーク上の任意のデバイスに接続します。

SSH クライアントを使用して NetScaler アプライアンスにログオンするには、次の手順を実行します:

1. ワークステーションで、SSH クライアントを起動します。
2. 初期構成には、デフォルトの IP アドレス（NSIP）である 192.168.100.1 を使用します。その後のアクセスでは、初回構成時に割り当てる NSIP を使用します。プロトコルとして SSH1 または SSH2 のいずれかを選択します。
3. [User name] にnsrootを入力します。Password にnsrootを入力し、パスワードが機能しない場合はアプライアンスのシリアル番号を入力します。シリアル番号のバーコードは、アプライアンスの背面にあります。たとえば、

```
1 login as: nsroot
2
3
```



```
Using keyboard-interactive authentication.

Password:

Last login: Tue Jun 16 10:37:28 2009 from 10.102.29.9




Done

>

<!--NeedCopy-->
```

NetScaler GUI

> 重要:
>
> Citric ADC GUI への HTTPS アクセスには、証明書とキーのペアが必要です。ADC では、証明書とキーのペアは内部サービスに自動的にバインドされます。デフォルトのキーサイズは、MPX または SDX アプライアンスで 1024 バイト、VPX インスタンスで 512 バイトです。ただし、最新の Web ブラウザーの多くは 1024 バイト未満のキーを受け入れません。このため、VPX 構成ユーティリティへの HTTPS アクセスがブロックされてしまいます。

また、ライセンスがない状態で MPX アプライアンスを起動して、その後でライセンスを追加してからアプライアンスを再起動すると、証明書のバインドが失われることがあります。

GUI への HTTPS アクセスのために、アプライアンスに少なくとも 1024 バイトの証明書とキーのペアをインストールすることをお勧めします。また、アプライアンスを起動する前に適切なライセンスをインストールしてください。

GUI には、構成ユーティリティと「ダッシュボード」と呼ばれる統計ユーティリティが含まれています。アプライアンスのイーサネットポートに接続されたワークステーションを介して、これらのツールにアクセスします。

GUI を実行するワークステーションのシステム要件は、次のとおりです。

- Windows ベースのワークステーションの場合は、Pentium 166 MHz 以上のプロセッサが必要です。
- Linux ベースのワークステーションの場合は、Linux カーネル v2.2.12 以降と`glibc`バージョン 2.12-11 以降を実行する、Pentium プラットフォームが必要です。32MB 以上の RAM が必要であり、48MB 以上を推奨します。ワークステーションは、ディスプレイをローカルホストに設定し、16 ビットカラーモードで KDE および KWM のウィンドウマネージャーをサポートする必要があります。
- Solaris ベースのワークステーションの場合は、Solaris 2.6、Solaris 7、または Solaris 8 を実行する Sun が必要です。

構成ユーティリティとダッシュボードにアクセスするには、サポートされている Web ブラウザーがワークステーションにインストールされている必要があります。

次の表は、NetScaler GUI バージョン 12.1、13.0、および 13.1 と互換性のあるブラウザーの一覧です。

オペレーティングシステム	**Web** ブラウザー	バージョン
Windows 10 以降	Edge	110.1587.63 およびそれ以降
Windows 10 以降	Mozilla Firefox	102 およびそれ以降
Windows 10 以降	Chrome	108 およびそれ以降
MAC	Mozilla Firefox	110.0.1 およびそれ以降
MAC	Safari	15.5 およびそれ以降

NetScaler GUI を使用する

構成ユーティリティにログオンしたら、状況依存ヘルプが付属するグラフィックインターフェイスを介して、アプライアンスを構成できます。

GUI にログオンするには、次の手順を実行します:

1. Web ブラウザーを開いて、NetScaler IP（NSIP）を HTTP アドレスとして入力します。初回構成がまだ完了していない場合は、デフォルトの NSIP（http://192.168.100.1）を入力します。NetScaler のログオンページが表示されます。

   注: 2 台の NetScaler アプライアンスが高可用性ペアとしてセットアップされている場合は、GUI にアクセスするときにセカンダリ NetScaler アプライアンスの IP アドレスを入力しないでください。このような方法でアクセスすると、GUI を使用してセカンダリアプライアンスを構成しても、その変更内容がプライマリ NetScaler アプライアンスに適用されません。

2. [User Name] ボックスに「`nsroot`」と入力します。
3. [Password] ボックスに、初回構成時に`nsroot`アカウントに割り当てた管理パスワードを入力し、[**Login**] をクリックします。パスワードが機能しない場合はアプライアンスのシリアル番号を入力します。シリアル番号のバーコードは、アプライアンスの背面にあります。

   オンラインヘルプにアクセスするには、右上隅の［Help］メニューの［Help］を選択します。

統計ユーティリティの使用

ダッシュボード（統計ユーティリティ）は、NetScaler アプライアンスのパフォーマンスを監視できる図と表を表示する、ブラウザーベースのアプリケーションです。

ダッシュボードにログオンするには、次の手順を実行します:

1. Web ブラウザーを開いて、NSIP を HTTP アドレスとして入力します。NetScaler のログオンページが表示されます。
2. [User Name] ボックスに「nsroot」と入力します。
3. [Password] ボックスに、初回構成時にnsrootアカウントに割り当てた管理パスワードを入力します。パスワードが機能しない場合はアプライアンスのシリアル番号を入力します。シリアル番号のバーコードは、アプライアンスの背面にあります。

ADC の初回構成

April 15, 2024

NetScaler MPX アプライアンスの初期構成については、「NetScaler MPX アプライアンスの初期構成」を参照してください。

NetScaler SDX アプライアンスの初期構成については、「NetScaler SDX アプライアンスの初期構成」を参照してください。

NITRO API

NITRO API を使用して NetScaler アプライアンスを構成できます。NITRO では、Representational State Transfer（REST）インターフェイスを介して機能が提供されます。そのため、NITRO アプリケーションはあらゆるプログラミング言語で開発することができます。さらに、Java、.NET、または Python で開発する必要があるアプリケーションの場合、NITRO API は、個別のソフトウェア開発キット（SDK）としてパッケージ化された関連ライブラリを介して提供されます。詳しくは、「NITRO API」を参照してください。

NetScaler ADC の導入を保護する

February 15, 2024

NetScaler アプライアンスの展開ライフサイクルを通じてセキュリティを維持するために、次のセキュリティの側面を考慮することをお勧めします。

- 物理的セキュリティ
- アプライアンスのセキュリティ
- Network Security
- 管理と管理

展開が異なれば、セキュリティに関する考慮事項も異なる場合があります。NetScaler の安全な導入ガイドラインは、特定のセキュリティ要件に基づいて適切な安全な導入を決定するのに役立つ一般的なセキュリティガイダンスを提供します。

NetScaler ADC アプライアンスを安全に展開するためのガイドラインの詳細については、NetScaler ADC セキュア展開のガイドラインを参照してください。

高可用性を構成する

August 15, 2023

2 台の NetScaler アプライアンスを高可用性構成で展開できます。この構成では、1 台の装置がアクティブに接続を受け付けてサーバーを管理し、2 台目の装置は 1 台目の装置を監視します。高可用性構成では、アクティブに接続を受け付けてサーバーを管理する NetScaler アプライアンスはプライマリ装置と呼ばれ、もう 1 台はセカンダリ装置と呼ばれます。プライマリ装置が故障した場合は、セカンダリ装置がプライマリになって、アクティブに接続の受け付けを開始します。

高可用性ペアの各 NetScaler アプライアンスは、「ハートビートメッセージ」または「ヘルスチェック」と呼ばれる定期的なメッセージを送信してもう一方の装置を監視し、ピアノードのヘルスまたは状態を判定します。プライマリ装置のヘルスチェックが失敗した場合、セカンダリ装置は指定された時間、接続を再試行します高可用性の詳細については、「高可用性」を参照してください。指定された時間内に再試行が成功しない場合、セカンダリ装置は「フェールオーバー」と呼ばれるプロセスによって、プライマリ装置の役割を引き継ぎます。次の図は、2 つの高可用性構成を示しています。1 つはワンアームモード構成で、もう 1 つはツーアームモード構成です。

図 1: ワンアームモードでの高可用性

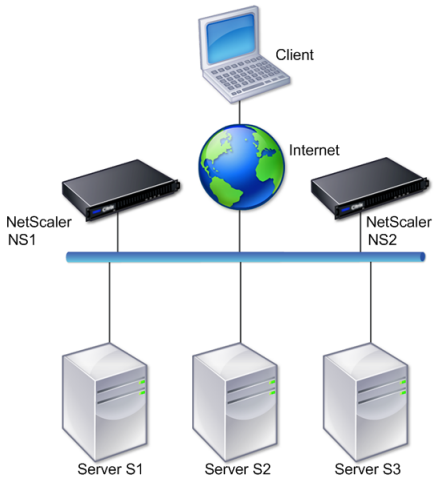
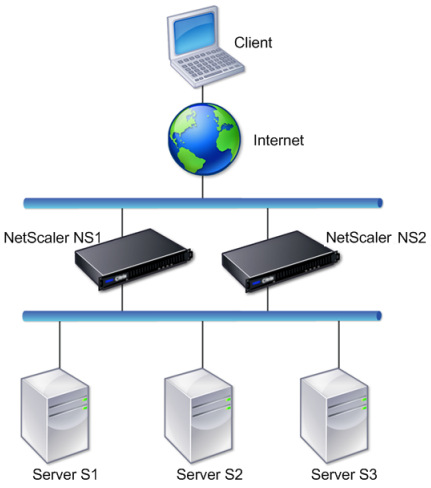


図 2: ツーアームモードでの高可用性



ワンアーム構成では、NS1 と NS2 の両方、およびサーバー S1、S2、S3 がスイッチに接続されています。

ツーアーム構成では、NS1 と NS2 の両方が 2 つのスイッチに接続されています。サーバー S1、S2、S3 は、2 番目のスイッチに接続されています。クライアントとサーバーの間のトラフィックは、NS1 または NS2 のいずれかを経由します。

高可用性環境をセットアップするには、1 台の ADC アプライアンスをプライマリとして、もう 1 台のアプライアンスをセカンダリとして設定します。各 ADC アプライアンスで次のタスクを実行します。

- ノードを追加します。
- 未使用のインターフェイスの高可用性モニターを無効にします。

ノードを追加

ノードは、ピア NetScaler アプライアンスを論理的に表したものです。ID と NSIP でピア装置を識別します。アプライアンスはこれらのパラメーターを使用して、ピアと通信してその状態を追跡します。ノードを追加すると、プライマリ装置とセカンダリ装置は、非同期的にハートビートメッセージを交換します。ノード ID は 64 以下の整数です。

CLI 経由

コマンドラインインターフェイスを使用してノードを追加するには、次の手順に従います。

コマンドプロンプトで次のコマンドを入力し、ノードを追加して構成を確認します。

- add HA node <id> <IP アドレス >

- show HA node <id>

  例

```
add HA node 0 10.102.29.170
 Done
> show HA node 0
1)      Node ID:      0
        IP:   10.102.29.200 (NS200)
        Node State: UP
        Master State: Primary
        SSL Card Status: UP
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 1:0:41:50 (days:hrs:min:
           sec)
<!--NeedCopy-->
```

GUI 経由

GUI を使用してノードを追加するには、次の手順に従います。

1. ［**System**］ > ［**High Availability**］に移動します。
2. ［**Nodes**］タブで ［**Add**］をクリックします。
3. ［**Create HA Node**］ページの ［**Remote Node IP Address**］テキストボックスに、リモートノードの NSIP アドレス（たとえば、10.102.29.170）を入力します。
4. ［**Configure remote system to participate in High Availability setup**］チェックボックスがオンになっていることを確認します。［**Remote System Login Credentials**］の下のボックスに、リモートノードのログイン情報を入力します。
5. ［**Turn off HA monitor on interfaces/channels that are down**］チェックボックスをオンにして、ダウンしているインターフェイスでの HA モニターを無効にします。

追加したノードが［Nodes］タブの一覧に表示されていることを確認します。

未使用のインターフェイスの高可用性モニターを無効にします

高可用性モニターは、インターフェイスを監視する仮想エンティティです。接続されていない、またはトラフィックに使用されていないインターフェイスのモニターを、無効にする必要があります。ステータスが DOWN になっているインターフェイスでモニターが有効になっている場合、ノードの状態は NOT UP になります。高可用性構成では、プライマリノードが NOT UP 状態になると、高可用性フェールオーバーが行われる可能性があります。以下のような場合、インターフェイスには DOWN のマークが付けられます。

- インターフェイスが接続されていない。
- インターフェイスが正常に動作していない。
- インターフェイスを接続するケーブルが正常に機能していない。

CLI 経由

コマンドラインインターフェイスを使用して未使用のインターフェイスの高可用性モニターを無効化するには、次の手順を実行します

コマンドプロンプトで次のコマンドを入力し、未使用のインターフェイスの高可用性モニターを無効にして構成を確認します。

- set interface <id> -haMonitor OFF
- show interface <id>

  例

```
> set interface 1/8 -haMonitor OFF
 Done
> show interface 1/8
        Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
        flags=0x4000 <ENABLED, DOWN, down, autoneg, 802.1q>
        MTU=1514, native vlan=1, MAC=00:d0:68:15:fd:3d, downtime
           238h55m44s
        Requested: media AUTO, speed AUTO, duplex AUTO, fctl OFF,
                  throughput 0

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0)
           Muted(0)
        Bandwidth thresholds are not set.
<!--NeedCopy-->
```

  未使用のインターフェイスで高可用性モニターが無効になっている場合、そのインターフェイスの showinterface コマンドの出力には「HAMON」が含まれません。

GUI 経由

GUI を使用して未使用のインターフェイスの高可用性モニターを無効にするには、次の手順に従います。

1. ［System］> ［Network］> ［Interfaces］に移動します。
2. モニターを無効にする必要があるインターフェイスを選択します。
3. ［開く］をクリックします。［Modify Interface］ダイアログボックスが開きます。
4. ［HA Monitoring］で［OFF］をクリックします。
5. 「OK」をクリックします。
6. インターフェイスを選択すると、ページ下部の詳細に「HA Monitoring: OFF」が表示されることを確認してください。

RPC ノードのパスワードを変更

December 8, 2023

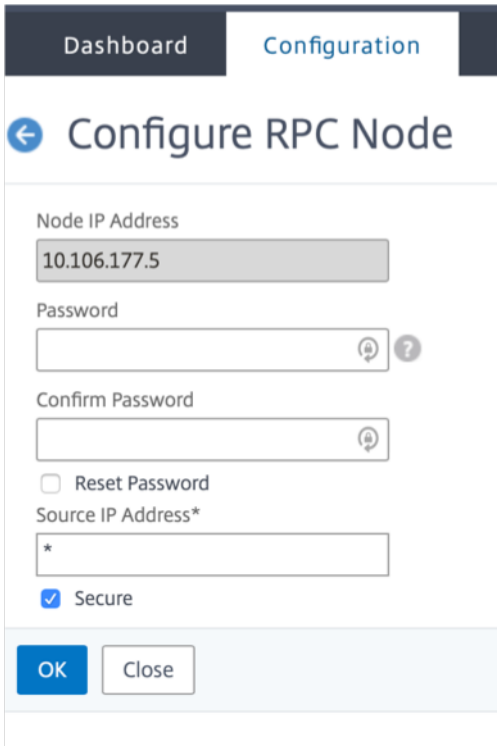
各アプライアンスがほかの NetScaler アプライアンスと通信するには、それらの NetScaler アプライアンスについての知識（認証方法など）が必要です。RPC ノードは、構成およびセッション情報のシステム間通信に使用される内部システムエンティティです。1 つの RPC ノードが各 NetScaler アプライアンスに存在し、他の NetScaler アプライアンスの IP アドレスや認証に使用されるパスワードなどの情報を格納します。他の NetScaler アプライアンスに接続する NetScaler アプライアンスは、RPC ノード内のパスワードをチェックします。

> 注:
>
> NetScaler アプライアンスを次のビルドのいずれかからリリース 13.1 ビルド 33.x 以降にアップグレードすると、`secure` RPC ノードのオプションは、内部 RPC および KRPCS サービスに存在する TLS 1.2 設定（有効または無効）に基づいて有効または無効になります。
>
> - リリース 13.0 ビルド 64.35 以前
> - リリース 12.1 ビルド 61.18 以前
>
> `Secure`オプションが有効になっている場合、RPC 通信は次の設定の NetScaler ノード間で暗号化されます。
>
> - 高可用性
> - クラスター
> - GSLB
>
> `secure`オプションは、NetScaler ノード間の RPC 接続に、安全なプロトコル TLS1.2 とポート番号 3008 と 3009 を使用します。
>
> 安全な RPC 通信を確保するために、Citrix ではこれらのセットアップをアップグレードする前に次の操作を実行することをお勧めします。
>
> - 内部 RPCS および KRPCS サービスでは TLS 1.2 を有効にする必要があります。
>
>   ```
>   - nsrpcs-127.0.0.1-3008
>   - nskrpcs-127.0.0.1-3009
>   - nsrpcs-::1l-3008
>   ```
>
> - 3008 と 3009 は、NetScaler ノード間のファイアウォールでブロック解除する必要があります。
>
> `secure`オプションは、NetScaler CLI または GUI を使用して有効または無効にできます。

GUI を使用して RPC ノードのパスワードを変更するには

1. [システム] > [ネットワーク] > [**RPC**] に移動します。
2. **RPC** ペインで、ノードを選択して ［**Edit**］をクリックします。

3. [**Configure RPC Node**] に、新しいパスワードを入力します。
4. [**Source IP Address**] に、ピアシステムノードとの通信に使用する既存のノードの IP アドレスを入力します。



5. [**Secure**] を選択し、[**OK**] をクリックします。

> 注
>
> セキュリティを強化するために、RPC ノードで [**Secure**] オプションを有効にすることをおすすめします。[**Secure**] オプションを有効にすると、アプライアンスは 1 つの ADC ノードから他の ADC ノードに送信されるすべての RPC 通信を暗号化して、RPC 通信を保護します。このセキュアな通信では、ポート番号 3008 を使用します。ADC ノード間のファイアウォールがポート番号 3008 をブロックしている場合は、ブロックを解除して続行します。そうしないと、構成の同期と構成の伝播が失敗する可能性があります。

CLI を使用して RPC ノードのパスワードを変更するには

コマンドラインで、次のコマンドを入力します:

```
set ns rpcNode <IPAddress> {
 -password }
  [-secure ( YES | NO )]
show ns rpcNode
<!--NeedCopy-->
```

例:


```
> set ns rpcNode 192.0.2.4 -password mypassword -secure YES
 Done
> show rpcNode
.
.
.
 IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
     SrcIP:  *           Secure:  ON
Done
>

<!--NeedCopy-->
```

FIPS アプライアンスの初回構成

August 15, 2023

> 注
>
> - FIPS FAQ はここにあります: FIPS FAQ

構成ユーティリティへの HTTPS アクセスおよびセキュアなリモートプロシージャコールには、証明書とキーのペアが必要です。RPC ノードは、構成およびセッション情報のシステム間通信に使用される内部システムエンティティです。アプライアンスごとに 1 つの RPC ノードが存在します。このノードに格納されるパスワードは、接続するアプライアンスによって提供されるパスワードと比較して調べられます。各アプライアンスがほかの NetScaler アプライアンスと通信するには、それらのアプライアンスについての知識（認証方法など）が必要です。RPC ノードはこの情報を保持しており、それにはほかの NetScaler アプライアンスの IP アドレスや、認証に使用されるパスワードなどが含まれます。

NetScaler MPX アプライアンス仮想アプライアンスでは、証明書とキーのペアは内部サービスに自動的にバインドされます。FIPS アプライアンスでは、FIPS カードのハードウェアセキュリティモジュール（HSM）に証明書とキーのペアをインポートする必要があります。そのためには、FIPS カードを構成し、証明書とキーのペアを作成して、それを内部サービスにバインドする必要があります。

CLI を使用してセキュアな HTTPS の構成

CLI を使用してセキュアな HTTPS を構成するには、次の手順を実行します

1. アプライアンスの FIPS カードでハードウェアセキュリティモジュール（HSM）を初期化します。HSM の初期化については、次のリンクのいずれかを参照してください。
   - MPX の場合: HSM を設定します。

- SDX の場合: SDX 14030/14060/14080 FIPS アプライアンス上のインスタンスに HSM を設定します。

2. アプライアンスが高可用性セットアップの一部である場合は、SIM を有効にします。プライマリアプライアンスおよびセカンダリアプライアンスで SIM を有効にする方法については、「高可用性セットアップでの FIPS アプライアンスの構成」を参照してください。
3. FIPS キーをアプライアンスの FIPS カードの HSM にインポートします。コマンドプロンプトで入力します。

```
import ssl fipskey serverkey -key ns-server.key -inform PEM
```

4. 証明書とキーのペアを追加します。コマンドプロンプトで入力します。

```
add certkey server -cert ns-server.cert -fipskey serverkey
```

5. 前の手順で作成した証明書キーを次の内部サービスにバインドします。コマンドプロンプトで入力します。

```
bind ssl service nshttps-127.0.0.1-443 -certkeyname server

bind ssl service nshttps-::11-443 -certkeyname server
```

GUI を使用して安全な HTTPS を構成する

GUI を使用して安全な HTTPS を構成するには、次の手順に従います。

1. アプライアンスの FIPS カードでハードウェアセキュリティモジュール（HSM）を初期化します。HSM の初期化については、次のリンクのいずれかを参照してください。
   - MPX の場合: HSM を設定します。
   - SDX の場合: SDX 14030/14060/14080 FIPS アプライアンス上のインスタンスに HSM を設定します。
2. アプライアンスが高可用性セットアップの一部である場合は、セキュア情報システム（SIM）を有効にします。プライマリアプライアンスおよびセカンダリアプライアンスで SIM を有効にする方法については、「高可用性セットアップでの FIPS アプライアンスの構成」を参照してください。
3. FIPS キーをアプライアンスの FIPS カードの HSM にインポートします。FIPS キーのインポートの詳細については、「既存の FIPS キーのインポート 」セクションを参照してください。
4. [**Traffic Management**] **>** [**SSL**] **>** [**Certificates**] に移動します。
5. 詳細ペインで、[Install] をクリックします。
6. [Install Certificate] ダイアログボックスで、証明書の詳細を入力します。
7. [Create] をクリックしてから、[Close] をクリックします。
8. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
9. 詳細ペインの [Action] タブで、[Internal Services] をクリックします。
10. 一覧から`nshttps-127.0.0.1-443`を選択し、[Open] をクリックします。
11. [Available] ペインの [SSL Settings] タブで、手順 7 で作成した証明書を選択して [Add] をクリックし、[OK] をクリックします。

12. 一覧からnshttps-::11-443を選択し、[Open]をクリックします。
13. [Available]ペインの[SSL Settings]タブで、手順 7 で作成した証明書を選択して[Add]をクリックし、[OK]をクリックします。
14. 「OK」をクリックします。

CLI を使用してセキュア RPC を構成する

CLI を使用してセキュア RPC を設定するには、次の手順に従います。

1. アプライアンスの FIPS カードでハードウェアセキュリティモジュール（HSM）を初期化します。HSM の初期化については、次のリンクのいずれかを参照してください。

   - MPX の場合: HSM を設定します。
   - SDX の場合: SDX 14030/14060/14080 FIPS アプライアンス上のインスタンスに HSM を設定します。

2. 安全な情報システム（SIM）を有効にします。プライマリアプライアンスおよびセカンダリアプライアンスで SIM を有効にする方法については、「高可用性セットアップでの FIPS アプライアンスの構成」を参照してください。
3. FIPS キーをアプライアンスの FIPS カードの HSM にインポートします。コマンドプロンプトで入力します。

   ```
   import ssl fipskey serverkey -key ns-server.key -inform PEM
   ```

4. 証明書とキーのペアを追加します。コマンドプロンプトで入力します。

   ```
   add certkey server -cert ns-server.cert -fipskey serverkey
   ```

5. 証明書とキーのペアを次の内部サービスにバインドします。コマンドプロンプトで入力します。

   ```
   bind ssl service nsrpcs-127.0.0.1-3008 -certkeyname server
   bind ssl service nskrpcs-127.0.0.1-3009 -certkeyname server
   bind ssl service nsrpcs-::1l-3008 -certkeyname server
   ```

6. セキュア RPC モードを有効にします。コマンドプロンプトで入力します。

   ```
   set ns rpcnode \<IP address\> -secure YES
   ```

   RPC ノードのパスワードの変更の詳細については、「RPC ノードのパスワードを変更する」を参照してください。

GUI を使用してセキュア RPC を構成する

GUI を使用してセキュア RPC を設定するには、次の手順に従います。

1. アプライアンスの FIPS カードでハードウェアセキュリティモジュール（HSM）を初期化します。HSM の初期化については、次のリンクのいずれかを参照してください。
   - MPX の場合: HSM を設定します。
   - SDX の場合: SDX 14030/14060/14080 FIPS アプライアンス上のインスタンスに HSM を設定します。
2. 安全な情報システム（SIM）を有効にします。プライマリおよびセカンダリアプライアンスで SIM を有効にする方法については、高可用性セットアップで FIPS アプライアンスを構成します。
3. FIPS キーをアプライアンスの FIPS カードの HSM にインポートします。FIPS キーのインポートの詳細については、「既存の FIPS キーのインポート 」セクションを参照してください。
4. [**Traffic Management**] **>** [**SSL**] **>** [**Certificates**] に移動します。
5. 詳細ペインで、[Install] をクリックします。
6. [Install Certificate] ダイアログボックスで、証明書の詳細を入力します。
7. [Create] をクリックしてから、[Close] をクリックします。
8. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
9. 詳細ペインの [Action] タブで、[Internal Services] をクリックします。
10. 一覧から nsrpcs-127.0.0.1-3008 を選択し、[Open] をクリックします。
11. [Available] ペインの [SSL Settings] タブで、手順 7 で作成した証明書を選択して [Add] をクリックし、[OK] をクリックします。
12. 一覧から nskrpcs-127.0.0.1-3009 を選択し、[Open] をクリックします。
13. [Available] ペインの [SSL Settings] タブで、手順 7 で作成した証明書を選択して [Add] をクリックし、[OK] をクリックします。
14. 一覧から`nsrpcs-::11-3008`を選択し、[Open] をクリックします。
15. [Available] ペインの [SSL Settings] タブで、手順 7 で作成した証明書を選択して [Add] をクリックし、[OK] をクリックします。
16. 「OK」をクリックします。
17. [システム**] > [**ネットワーク**] > [RPC**] に移動します。
18. 詳細ペインで IP アドレスを選択して、[Open] をクリックします。
19. [Configure RPC Node] ダイアログボックスで、[Secure] を選択します。
20. 「OK」をクリックします。

一般的なネットワークトポロジ

August 15, 2023

「NetScaler ADC アプライアンスはネットワーク内のどこに適合しますか?」の「物理展開モード」セクションで説明されているようにでは、NetScaler ADC アプライアンスをクライアントとサーバー間でインライン展開するか、ワンアームモードで展開できます。インラインモードでは、一般的な導入タイプであるツーアームトポロジを使用します。

一般的なツーアームトポロジを設定します

ツーアームトポロジでは、1 つのネットワークインターフェイスはクライアントネットワークに接続され、もう 1 つのネットワークインターフェイスはサーバーネットワークに接続されます。これにより、すべてのトラフィックが仮想アプライアンスを通過するようになります。このトポロジでは、ハードウェアの再接続が必要になり、一時的にダウン時間が発生する場合もあります。ツーアームトポロジの基本的なバリエーションは複数サブネットと透過モードです。複数サブネットでは、通常、仮想アプライアンスがパブリックサブネット上に、サーバーがプライベートサブネット上に配置されます。透過モードでは、仮想アプライアンスとサーバーの両方がパブリックネットワーク上に配置されます。

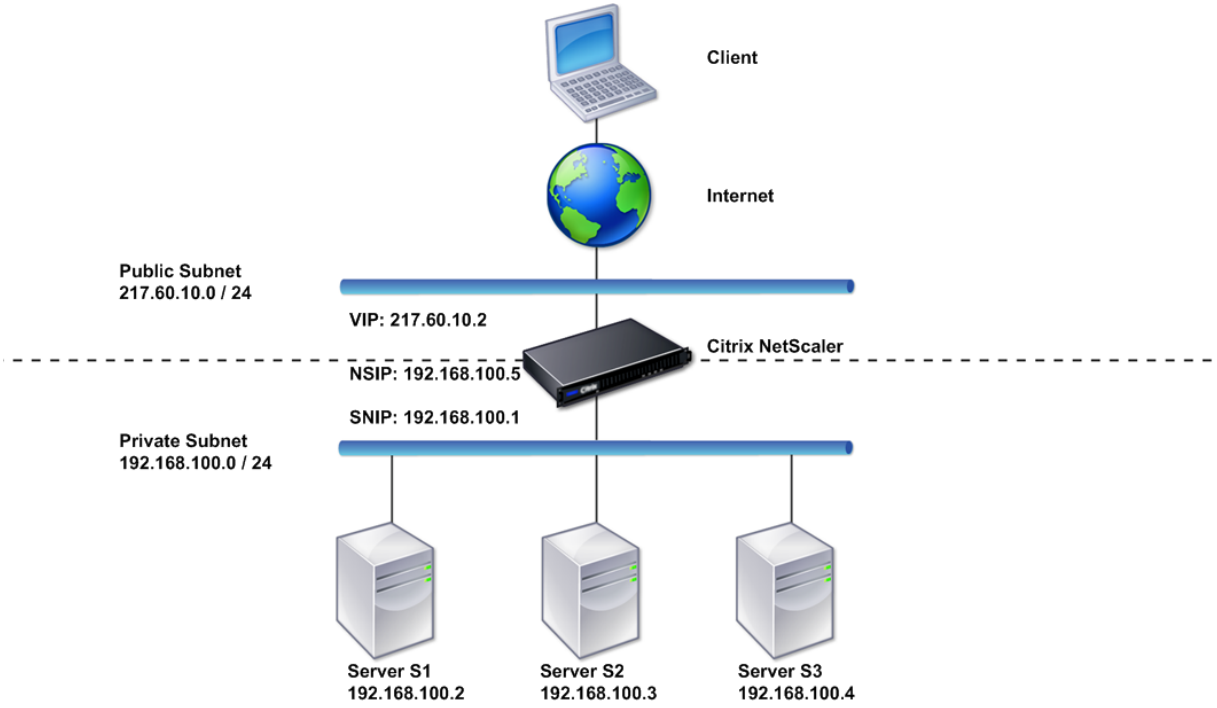
シンプルなツーアーム複数サブネットトポロジのセットアップ

最もよく使用されるトポロジの 1 つで、NetScaler アプライアンスはクライアントのサーバーの間にインラインに配置され、仮想サーバーはクライアント要求を処理するように設定されます。この構成は、クライアントとサーバーが異なるサブネット上にある場合に使用されます。たいていの場合、クライアントとサーバーは、それぞれパブリックサブネット上とプライベートサブネット上にあります。

たとえば、サーバー S1、S2、および S3 を管理するためにツーアームモードで展開されたアプライアンスについて考えてみましょう。アプライアンス上で HTTP の仮想サーバーが構成されており、各サーバー上で HTTP サービスが実行されています。これらのサーバーはプライベートサブネット上にあり、アプライアンスでこれらのサーバーと通信するための SNIP が構成されています。MIP の代わりに SNIP を使用するため、アプライアンスで USNIP（Use SNIP）オプションを有効にする必要があります。

次の図に示すように、VIP はパブリックサブネット 217.60.10.0 上にあり、NSIP、サーバー、および SNIP はプライベートサブネット 192.168.100.0/24 上にあります。

図 1：ツーアームモード、複数のサブネットのトポロジ図



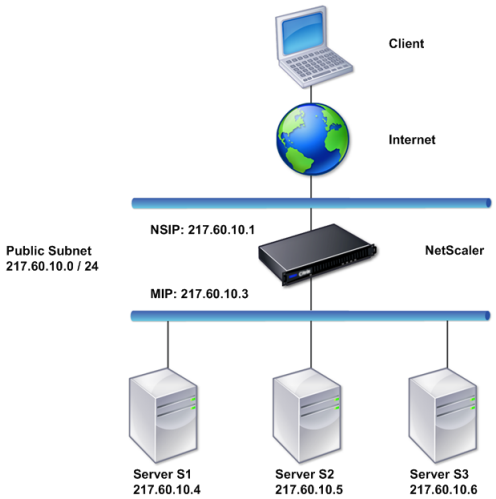
複数のサブネットを持つツーアームモードで NetScaler ADC アプライアンスを展開するには、次の手順に従います。

1. NetScaler IP アドレス（NSIP）の構成の説明に従って、NSIP とデフォルトゲートウェイを構成します。
2. 「サブネット IP アドレスの設定」の説明に従って、SNIPを設定します。
3. 「USNIP モードを有効または無効にするには」セクションの説明に従って、USNIP オプションを有効にします 。
4. 仮想サーバーの作成セクションおよび「サービスの設定」セクションの説明に従って、[仮想サーバーとサービスを構成します](/ja-jp/citrix-adc/13-1/load-balancing/load-balancing-setup.html) 。
5. 一方のネットワークインターフェイスをプライベートサブネットに、もう一方のインターフェイスをパブリックサブネットに接続します。

シンプルなツーアーム透過トポロジのセットアップ

クライアントが仮想サーバーの仲介なしでサーバーに直接アクセスする必要がある場合は、透過モードを使用します。クライアントはサーバーにアクセスできる必要があるので、サーバー IP アドレスはパブリックにする必要があります。次の図に示す例では、NetScaler アプライアンスがクライアントとサーバーの間に配置されています。そのため、トラフィックはアプライアンスを経由する必要があります。パケットをブリッジするため、L2 モードを有効にする必要があります。NSIP と MIP は、同じパブリックサブネット上（217.60.10.0/24）にあります。

図 2: ツーアーム、透過モードのトポロジ図



NetScaler ADC アプライアンスをツーアームの透過モードで展開するには、次の手順に従います。

1. NetScaler IP アドレス（NSIP）の構成の説明に従って、NSIP とデフォルトゲートウェイを構成します。
2. 「レイヤ 2 モードの有効化および無効化」の説明に従って、L2 モードを有効にします。
3. 管理対象サーバーのデフォルトゲートウェイを、MIP として構成します。
4. ネットワークインターフェイスを、スイッチの適切なポートに接続します。

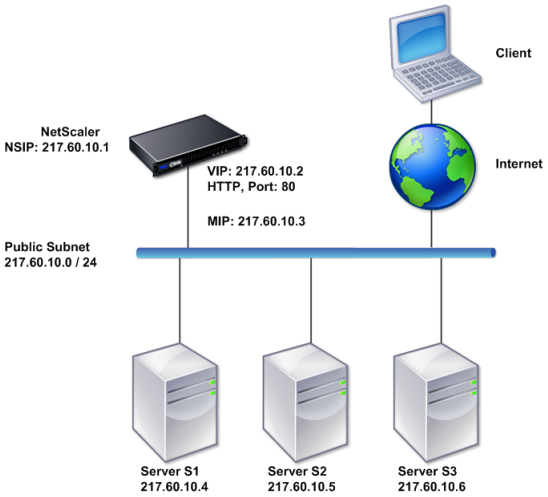
一般的なワンアームトポロジのセットアップ

ワンアームトポロジの 2 つの基本的なバリエーションは、1 つのサブネットを持つトポロジと複数のサブネットを持つトポロジです。

シンプルなワンアームシングルサブネットトポロジのセットアップ

クライアントとサーバーが同じサブネット上にある場合は、単一のサブネットでワンアームトポロジを使用できます。たとえば、サーバー S1、S2、および S3 を管理するためにワンアームモードで展開された NetScaler アプライアンスについて考えてみましょう。ADC アプライアンスで HTTP の仮想サーバーが構成されており、そのサーバー上で HTTP サービスが実行されています。次の図に示すように、NetScaler IP アドレス（NSIP）、マップされた IP アドレス（MIP）、およびサーバーの IP アドレスは同じパブリックサブネット上（217.60.10.0/24）にあります。

図 3: ワンアームモード、シングルサブネットのトポロジ図



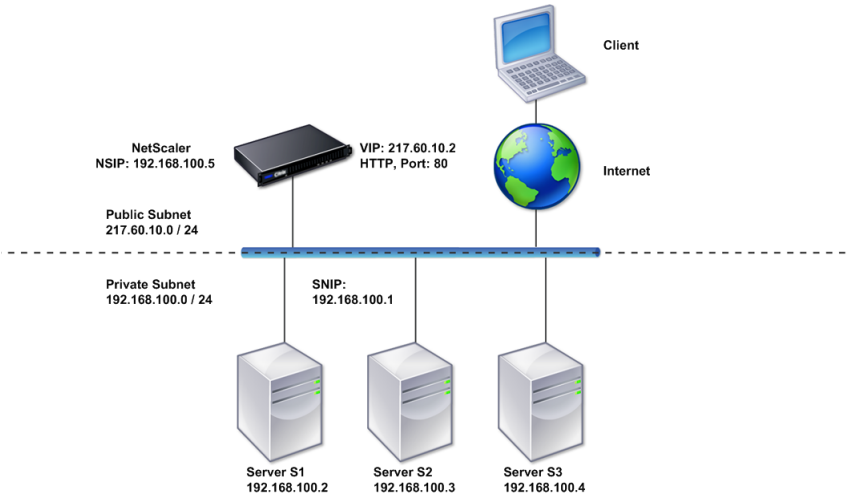
NetScaler ADC アプライアンスを単一サブネットのワンアームモードで展開するには、次の手順に従います。

1. NetScaler IP アドレス（NSIP）の構成の説明に従って、NSIP とデフォルトゲートウェイを構成します。
2. 仮想サーバーの作成セクションおよび「サービスの設定」セクションの説明に従って、[仮想サーバーとサービスを構成します](/ja-jp/citrix-adc/13-1/load-balancing/load-balancing-setup.html) 。
3. 一方のネットワークインターフェイスをスイッチに接続します。

シンプルなワンアーム複数サブネットトポロジのセットアップ

クライアントとサーバーが異なるサブネット上にある場合は、複数のサブネットを持つワンアームトポロジを使用できます。たとえば、サーバー S1、S2、および S3 を管理するためにワンアームモードで導入された NetScaler アプライアンスについて考えてみましょう。これらのサーバーはネットワーク上のスイッチ SW1 に接続されています。アプライアンスで HTTP の仮想サーバーが構成されており、そのサーバー上で HTTP サービスが実行されています。これら 3 つのサーバーはプライベートサブネット上にあるので、SNIP（Subnet IP：サブネット IP）アドレスはこれらのサーバーと通信するように設定されています。アプライアンスが MIP の代わりに SNIP を使用するように、[Use Subnet IP address (USNIP)] を有効にする必要があります。次の図に示すように、仮想 IP アドレス (VIP）はパブリックサブネット上（217.60.10.0/24）にあります。NSIP、SNIP、およびサーバーの IP アドレスはプライベートサブネット上（192.168.100.0/24）にあります。

図 4：ワンアームモード、複数のサブネットのトポロジ図



複数のサブネットを持つワンアームモードで NetScaler ADC アプライアンスを展開するには、次の手順に従います。

1. NetScaler IP アドレス（NSIP）の構成の説明に従って、NSIP とデフォルトゲートウェイを構成します。
2. 「サブネット IP アドレスの設定」の説明に従って、SNIP を設定し、USNIPオプションを有効にします。
3. 仮想サーバーの作成セクションおよび「サービスの設定」セクションの説明に従って、[仮想サーバーとサービスを構成します](/ja-jp/citrix-adc/13-1/load-balancing/load-balancing-setup.html) 。
4. 一方のネットワークインターフェイスをスイッチに接続します。

システム管理設定

August 15, 2023

初期構成が整ったら、NetScaler アプライアンスの動作を定義し、接続管理を容易にする設定を構成できます。HTTP 要求と応答を処理するためのいくつかのオプションがあります。ルーティング、ブリッジング、および MAC ベースの転送モードは、NetScaler アプライアンスにアドレス指定されていないパケットを処理するために使用できます。ネットワークインターフェイスの特性を定義し、インターフェイスを集約できます。タイミングの問題を防ぐために、Citrix クロックをネットワークタイムプロトコル（NTP）サーバーと同期させることができます。NetScaler アプライアンスは、権限のあるドメインネームサーバー（ADNS）としてなど、さまざまな DNS モードで動作できます。システム管理用に SNMP を設定し、システムイベントの syslog ロギングをカスタマイズできます。展開する前に、構成が完全で正しいことを確認してください。

システム設定

August 15, 2023

システム設定の構成には、接続のキープアライブとサーバーオフロードを有効にするための HTTP ポートの設定、各サーバーの最大接続数の設定、接続あたりの最大要求件数の設定などの基本的なタスクが含まれます。プロキシ IP アドレスが適さない環境では、クライアント IP アドレス挿入を有効にして、HTTP Cookie バージョンを変更できます。

データ接続用のエフェメラルポートの代わりに、特定のポート範囲で FTP 接続が開かれるように NetScaler アプライアンスを構成することもできます。ファイアウォールですべてのポートを開くのは危険なので、この方法によってセキュリティが向上します。1,024～64,000 までの任意の範囲を設定できます。

展開前に、確認チェックリストを使用して構成内容を確認します。HTTP パラメーターと FTP ポート範囲を構成するには、NetScaler GUI を使用します。

次の表に記載された HTTP パラメーターの種類を変更できます。

パラメータータイプ: HTTP ポート情報

指定：管理対象サーバーが使用する Web サーバーの HTTP ポート。ポートを指定すると、アプライアンスは、指定されたポートと一致する宛先ポートを持つクライアント要求に対して要求のスイッチ操作を実行します。

> 注：着信したクライアント要求が、アプライアンスで指定されたサービスまたは仮想サーバー宛でない場合、要求の宛先ポートは、グローバルに構成されたいずれかの HTTP ポートと一致する必要があります。これにより、アプライアンスは接続のキープアライブとサーバーオフロードを実行できます。

パラメータータイプ: 制限

指定：各管理対象サーバーへの最大接続数、および、各接続を介して送信される要求の最大件数。たとえば、［Max Connections］が「500」に設定され、アプライアンスが 3 台のサーバーを管理している場合、3 台のサーバーそれぞれに対し、最大 500 個の接続を開くことができます。デフォルトでは、アプライアンスは管理する任意のサーバーに対して任意の数の接続を作成できます。接続あたりの要求数を無制限に指定するには、［Max Requests］を「0」に設定します。

> 注：Apache HTTP サーバーを使用している場合は、［Max Connections］を、Apache httpd.conf ファイルの MaxClients パラメーターと同じ値に設定する必要があります。その他の Web サーバーの場合、このパラメーターの設定はオプションとなります。

パラメータータイプ: クライアント IP の挿入

指定：HTTP 要求ヘッダーへのクライアント IP アドレスの挿入を有効または無効にします。隣接するテキストボックスで、ヘッダーフィールドの名前を指定できます。アプライアンスが管理する Web サーバーが SNIP アドレスを受信すると、サーバーはそのアドレスをクライアント IP アドレスとして識別します。一部のアプリケーションでは、ログを記録するため、または Web サーバーが提供するコンテンツを動的に決定するために、クライアント IP アドレスが必要です。

クライアントから、アプライアンスが管理している 1 台、数台、またはすべてのサーバーに送信された HTTP ヘッダー要求に、実際のクライアント IP アドレスを挿入する機能を有効にできます。これによって、（Apache モジュール、ISAPI インターフェイス、または NSAPI インターフェイスを使用して）サーバーを少し変更するだけで、挿入されたアドレスにアクセスできるようになります。

パラメータータイプ: クッキーバージョン

指定：COOKIEINSERT パーシステンスが仮想サーバーに設定されている場合に使用される HTTP Cookie バージョン。デフォルトでは、インターネットで最も一般的な種類であるバージョン 0 を使用します。代わりにバージョン 1 を指定することも可能です。

パラメータータイプ: 要求/応答

指定：特定の種類の要求を処理し、HTTP エラー応答のログを有効または無効にするオプション。

パラメータータイプ: サーバーヘッダーの挿入

指定：NetScaler が生成した HTTP 応答にサーバーヘッダーを挿入します。

GUI を使用して HTTP パラメーターを設定するには、次の手順に従います。

1. ナビゲーションペインで、[システム] を展開し、[設定] をクリックします。
2. 詳細ペインで、[**Change HTTP Parameters**] をクリックします。
3. [**Configure HTTP parameters**] ダイアログボックスで、上記の表に一覧表示された見出しの下に表示されている一部またはすべてのパラメーターの値を指定します。
4. [**OK**] をクリックします。

GUI を使用して FTP ポート範囲を設定するには、次の手順に従います。

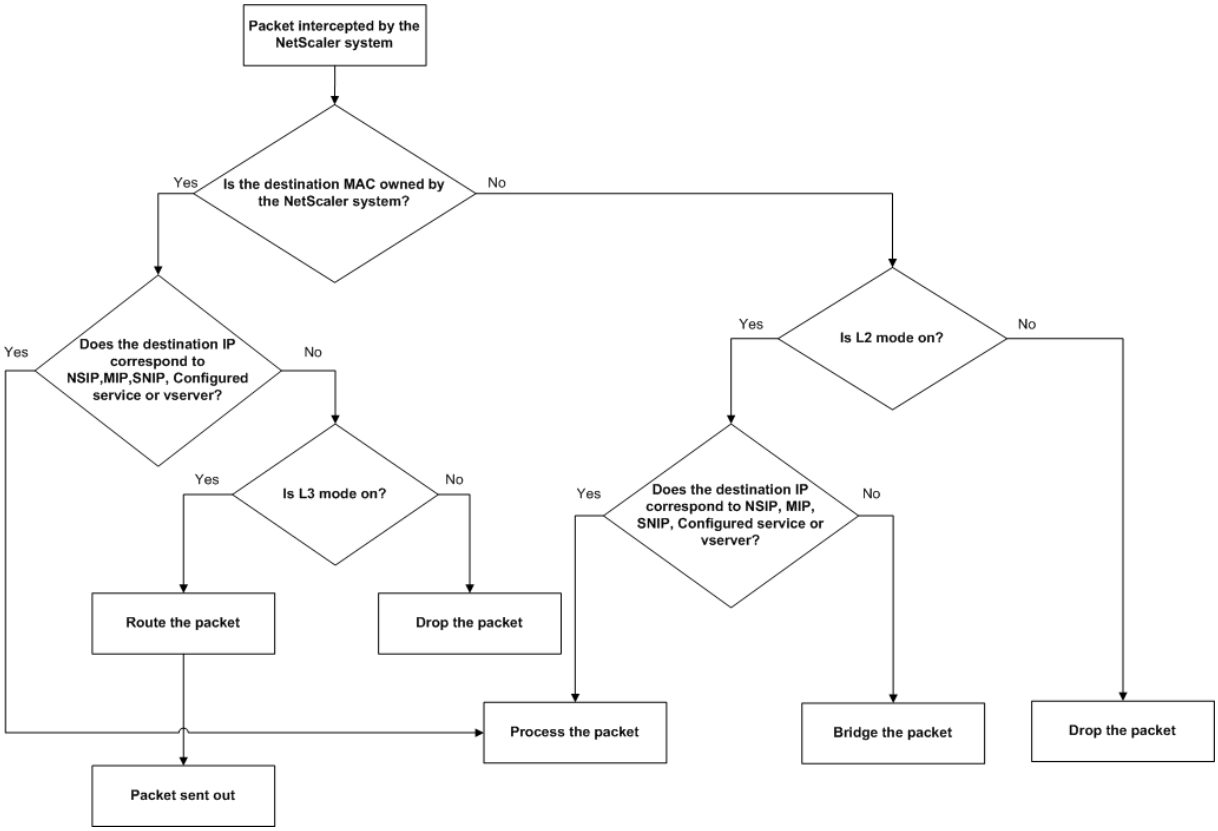
1. ナビゲーションペインで、[**System**] を展開し、[**Settings**] をクリックします
2. 詳細ペインの [**Settings**] で、[**Change global system settings**] をクリックします。
3. [**FTP Port Range**] の [**Start Port**] および [**End Port**] ボックスに、指定する範囲の最小ポート番号と最大ポート番号（たとえば、「5000」と「6000」）をそれぞれ入力します。
4. [**OK**] をクリックします。

パケット転送モード

August 15, 2023

NetScaler アプライアンスは、アプライアンスが所有する IP アドレス宛ではないパケット（つまり NSIP、MIP、SNIP、構成済みサービス、または構成済み仮想サーバーの IP アドレス宛でないパケット）をルーティングまたはブリッジできます。デフォルトでは、L3 モード（ルーティング）が有効になり、L2 モード（ブリッジ）が無効になりますが、この構成は変更できます。アプライアンスがパケットを評価し、パケットの処理、ルーティング、ブリッジ、廃棄のいずれかを行う方法を次のフローチャートに示します。

図 1: レイヤー 2 モードとレイヤー 3 モード間の相互作用



アプライアンスは次のモードを使用して、受信したパケットを転送できます。

- レイヤー 2（L2）モード
- レイヤー 3（L3）モード
- MAC ベース転送モード

レイヤー **2** モードの有効化と無効化

レイヤー 2 モードは、レイヤー 2 フォワード（ブリッジ）機能を制御します。このモードを使用して、NetScaler アプライアンスをレイヤー 2 デバイスとして動作させ、自分宛ではないパケットをブリッジするように設定することができます。このモードを有効にした場合、パケットはどの MAC アドレスにも転送されません。これは、パケットがアプライアンスの任意のインターフェイスに着信することができ、各インターフェイスが独自の MAC アドレスを持っているからです。

レイヤー 2 モードを無効にした場合（デフォルト）、アプライアンスは、自分の MAC アドレス宛ではないパケットをドロップします。別のレイヤー 2 デバイスがアプライアンスと並列に設置されている場合は、レイヤー 2 モードを無効にしてブリッジ（レイヤー 2）ループを防ぐ必要があります。構成ユーティリティまたはコマンドラインを使用して、レイヤー 2 モードを有効にできます。

注: アプライアンスはスパニングツリープロトコルをサポートしていません。L2 モードが有効な場合に、ループを避

けるために、アプライアンス上の 2 つのインターフェイスを同じブロードキャストドメインに接続しないでください。

CLI を使用してレイヤー **2** モードを有効または無効にするには

コマンドプロンプトで次のコマンドを入力し、レイヤー 2 モードを有効または無効にして、有効または無効になっていることを確認します。

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

例

```
    > enable ns mode l2
    Done
    > show ns mode

    Mode Acronym Status
    ------- ------- ------
    1) Fast Ramp FR ON
    2) Layer 2 mode L2 ON
    .
    .
    .
    Done
    >

    > disable ns mode l2
    Done
    > show ns mode

    Mode Acronym Status
    ------- ------- ------
    1) Fast Ramp FR ON
    2) Layer 2 mode L2 OFF
    .
    .
    .
    Done
    >
<!--NeedCopy-->
```

GUI を使用してレイヤー 2 モードを有効または無効にするには

1. ナビゲーションペインで、[システム] を展開し、[設定] をクリックします。
2. 詳細ペインの [モード と 機能] で、[モードの構成] をクリックします。
3. [**Configure Modes**] ダイアログボックスで、レイヤ 2 モードを有効にするには、[レイヤ **2** モード] チェックボックスをオンにします。レイヤー 2 モードを無効にするには、チェックボックスをオフにします。

4. [**OK**] をクリックします。有効化/無効化モードですか? というメッセージが詳細ペインに表示されます。
5. [はい] をクリックします。

レイヤー 3 モードの有効化と無効化

レイヤー 3 モードは、レイヤー 3 フォワード機能を制御します。このモードを使用して、NetScaler アプライアンスがルーティングテーブルを参照して自分宛ではないパケットを転送するように設定できます。レイヤー 3 モードを有効にした場合（デフォルト）、アプライアンスはルートテーブルのルックアップを実行して、アプライアンス所有の IP アドレス宛ではないすべてのパケットを転送します。レイヤー 3 モードを無効にした場合、アプライアンスはこれらのパケットをドロップします。

CLI を使用してレイヤ 3 モードを有効または無効にする

コマンドプロンプトで次のコマンドを入力し、レイヤー 3 モードを有効または無効にして、有効または無効になっていることを確認します。

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

例

```
> enable ns mode l3
Done
> show ns mode

Mode Acronym Status
------- ------- ------
1) Fast Ramp FR ON
2) Layer 2 mode L2 OFF
.
.
.
9) Layer 3 mode (ip forwarding) L3 ON
.
.
.
Done
>

> disable ns mode l3
Done
> show ns mode

Mode Acronym Status
------- ------- ------
```



```
      1) Fast Ramp FR ON
      2) Layer 2 mode L2 OFF
      .
      .
      .
      9) Layer 3 mode (ip forwarding) L3 OFF
      .
      .
      .
      Done
      >
<!--NeedCopy-->
```

GUI を使用してレイヤ **3** モードを有効または無効にする

1. ナビゲーションペインで、[システム] を展開し、[設定] をクリックします。
2. 詳細ウィンドウの [モードと機能] で、[モードの構成] をクリックします。
3. [モードの構成] ダイアログボックスで、レイヤ 3 モードを有効にするには、[レイヤ **3** モード **(IP** 転送**)**] チェックボックスをオンにします。To disable Layer 3 mode, clear the check box.
4. [**OK**] をクリックします。有効化**/**無効化モードですか? というメッセージが詳細ペインに表示されます。
5. [はい] をクリックします。

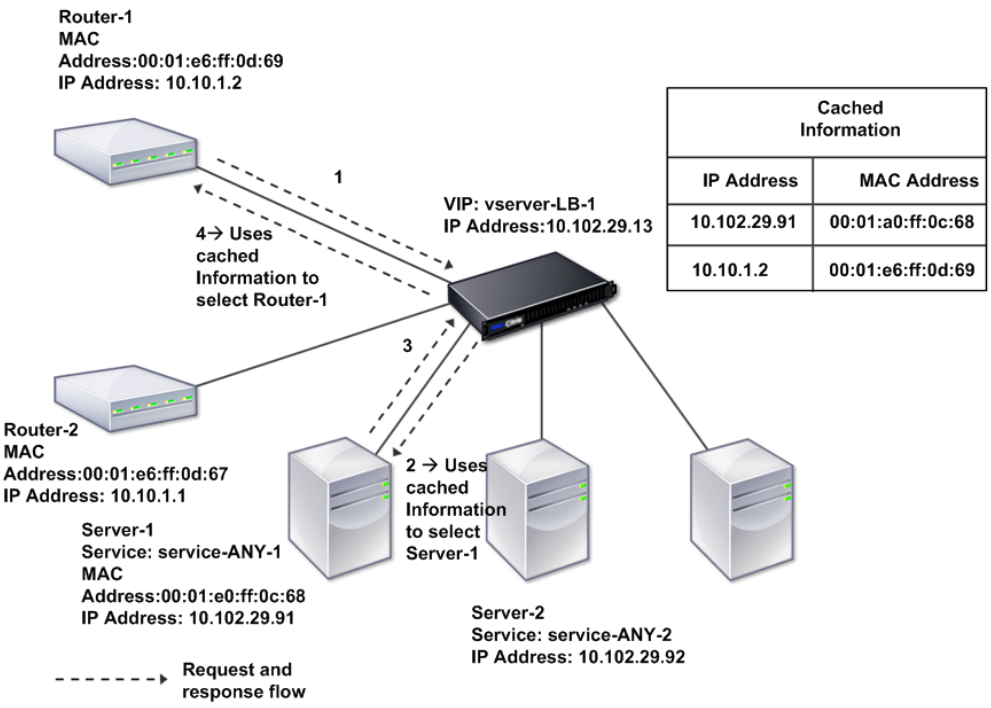
MAC ベースの転送モードを有効または無効にする

NetScaler アプライアンスがソースの MAC アドレスを記憶しているので、MAC ベース転送を使用して、パケットの転送時にトラフィックをより効率的に処理し、複数のルートや ARP ルックアップを防ぐことができます。複数のルックアップを防ぐため、アプライアンスは、ARP ルックアップを実行するすべての接続のソース MAC アドレスをキャッシュして、データを同じ MAC アドレスに返します。

MAC ベース転送は、VPN デバイスを使用している場合に便利です。これは、アプライアンスによって、特定の VPN を経由するすべてのトラフィックが同じ VPN デバイスを通過するようになるからです。

次の図は、MAC ベース転送のプロセスを示しています。

図 2: MAC ベースの転送プロセス



MAC ベース転送を有効にした場合、アプライアンスは次の MAC アドレスをキャッシュします。

- 受信接続のソース（ルーター、ファイアウォール、VPN デバイスなどの通信デバイス）
- 要求に応答するサーバー

サーバーがアプライアンスを介して応答する場合、アプライアンスは、応答パケットの宛先 MAC アドレスをキャッシュしたアドレスに設定し、トラフィックが対称的に流れるようにして、応答をクライアントに転送します。このプロセスでは、ルートテーブルのルックアップ機能と ARP ルックアップ機能が回避されます。ただし、アプライアンスが接続を開始した場合は、ルックアップ機能でルートと ARP テーブルが使用されます。MAC ベース転送を有効にするには、構成ユーティリティを使用するか、またはコマンドラインを使用します。

一部の展開環境では、着信および発信パスが、異なるルーターを経由する必要があります。このような状況では、MAC ベース転送がトポロジデザインに違反します。グローバルサーバー負荷分散 (GSLB) サイトで、着信パスと送信パスが異なるルーターを経由する必要がある場合、MAC ベースの転送を無効にし、アプライアンスのデフォルトルーターを発信ルーターとして使用する必要があります。

MAC ベース転送を無効にして、レイヤー 2 またはレイヤー 3 接続を有効にした場合、ルートテーブルは、発信接続と着信接続に別のルーターを指定できます。MAC ベース転送を無効にするには、構成ユーティリティを使用するか、またはコマンドラインを使用します。

CLI を使用して **MAC** ベースの転送を有効または無効にする

コマンドプロンプトで次のコマンドを入力し、MAC ベース転送を有効または無効にして、有効または無効になっていることを確認します。

- <enable ns mode \< モード\>
- <disable ns mode \< モード\>
- <show ns mode
  Example

  “‘ pre codeblock

  > enable ns mode mbf
  > Done
  > show ns mode

```
1   Mode                          Acronym           Status
2   -------                       -------           ------   1)  Fast
       Ramp                    FR                  ON   2)  Layer 2
       mode                 L2                  OFF   .   .   .   6)
       MAC-based forwarding        MBF                 ON   .   .   .
         Done  >
```

  > disable ns mode mbf
  > Done
  > show ns mode

```
1   Mode                          Acronym           Status
2   -------                       -------           ------   1)  Fast
       Ramp                    FR                  ON   2)  Layer 2
       mode                 L2                  OFF   .   .   .   6)
       MAC-based forwarding        MBF                 OFF   .   .   .
          Done  >  <!--NeedCopy--> ```
```

GUI を使用して MAC ベースの転送を有効または無効にするには

1. ナビゲーションペインで、[システム] を展開し、[設定] をクリックします。
2. 詳細ペインの［**Modes and Features**］グループで［**Configure modes**］をクリックします。
3. [モードの構成] ダイアログボックスで、MAC ベースの転送モードを有効にするには、[**MAC** ベースの転送] チェックボックスをオンにします。MAC ベース転送を無効にするには、このチェックボックスをオフにします。
4. ［**OK**］をクリックします。有効化/無効化モードですか? というメッセージが詳細ペインに表示されます。
5. ［**Yes**］をクリックします。

ネットワークインターフェイス

August 15, 2023

NetScaler のインターフェイスには、スロット/ポート表示に番号が付けられています。個々のインターフェイスの特性を変更することに加えて、特定のホストグループだけにトラフィックが許可されるように VLAN を構成できます。リンクを高速チャネルに集約することもできます。

仮想 **LAN**

NetScaler アプライアンスは、（レイヤー 2）ポートと IEEE802.1Q タグ付きの VLAN（Virtual LAN：仮想 LAN）をサポートしています。VLAN 構成は、トラフィックを特定のワークステーショングループだけに制限しなければならない場合に便利です。IEEE 802.1q タグ付け機能を使用して複数の VLAN に属するように、ネットワークインターフェイスを構成できます。

構成した VLAN は、IP サブネットにバインドできます。これにより、（サブネット上のホストのデフォルトルーターとして構成されている場合）ADC アプライアンスは、これらの VLAN 間で IP 転送を実行します。

NetScaler アプライアンスは、次のタイプの VLAN をサポートします。

- デフォルト VLAN

  デフォルトでは、NetScaler アプライアンスのネットワークインターフェイスは、タグなしのネットワークインターフェイスとして、単一のポートベース VLAN に含まれています。このデフォルト VLAN は、VID が 1 であり、永続的に存在します。デフォルト VLAN を削除したり、その VID を変更したりすることはできません。

- ポートベース VLAN

  排他的なレイヤー 2 ブロードキャストドメインを共有するネットワークインターフェイスのセットは、ポートベース VLAN のメンバーシップを定義します。複数のポートベース VLAN を構成できます。インターフェイスをタグなしメンバーとして新しい VLAN に追加すると、デフォルト VLAN から自動的に削除されます。

- タグ付き VLAN

  ネットワークインターフェイスは、VLAN のタグ付きまたはタグなしメンバーになることができます。各ネットワークインターフェイスは、唯一の VLAN（ネイティブ VLAN）のタグなしメンバーです。タグなしネットワークインターフェイスは、タグなしフレームとしてネイティブ VLAN のフレームを転送します。タグ付きネットワークインターフェイスは、複数の VLAN の一部になることができます。タグ付きを設定する場合は、リンクの両端で VLAN 設定が一致していることを確認してください。構成ユーティリティを使用し、VLAN のタグ付きメンバーとしてポートをバインドできる、タグ付き VLAN（nsvlan）を定義できます。この VLAN を構成するには ADC アプライアンスを再起動する必要があるので、ネットワークの初回構成中に実行する必要があります。

リンクアグリゲートチャネル

リンクアグリゲーションは、複数ポートからの着信データを、1 つの高速リンクに結合します。リンクアグリゲートチャネルを構成すると、NetScaler アプライアンスとほかの接続デバイス間の通信チャネルの容量と可用性が増加します。アグリゲートされたリンクは「チャネル」とも呼ばれます。

ネットワークインターフェイスをチャネルにバインドした場合、チャネルのパラメーターは、ネットワークインターフェイスのパラメーターよりも優先されます。ネットワークインターフェイスは、1 つのチャネルにのみバインドできます。ネットワークインターフェイスをリンクアグリゲートチャネルにバインドすると、VLAN 構成が変更されます。つまり、ネットワークインターフェイスをチャネルにバインドすると、ネットワークインターフェイスは以前属していた VLAN から削除され、デフォルト VLAN に追加されます。ただし、チャネルを元の VLAN や新しい VLAN にバインドすることができます。たとえば、ネットワークインターフェイス 1/2 と 1/3 を、ID が 2 の VLAN にバインドしていて、それらをリンクアグリゲートチャネル LA/1 にバインドした場合、ネットワークインターフェイスはデフォルト VLAN に移動されますが、VLAN 2 にバインドすることができます。

注：また、リンクアグリゲーション制御プロトコル（Link Aggregation Control Protocol：LACP）を使用して、リンクアグリゲーションを構成することもできます。詳細については、「リンク集約制御プロトコルを使用したリンク集約の設定」を参照してください。

クロック同期

August 15, 2023

NetScaler ADC アプライアンスを設定して、ローカルの時刻を、NTP（Network Time Protocol：ネットワークタイムプロトコル）サーバーの時刻と同期することができます。これにより、NetScaler のクロックの設定は、ネットワーク上のほかのサーバーと同じ日付と時刻になります。NTP は、UDP（User Datagram Protocol：ユーザーデータグラムプロトコル）ポート 123 を、トランスポートレイヤーとして使用します。NTP 設定ファイルに NTP サーバを追加して、アプライアンスがこれらのサーバから定期的に更新を取得できるようにします。

ローカルの NTP サーバーがない場合は、公式 NTP サイト（http://www.ntp.org）で、パブリックなオープンアクセス NTP サーバーの一覧を検索できます。

アプライアンスでクロック同期を構成するには、次の手順に従います：

1. コマンドラインにログオンし、shell コマンドを入力します。
2. シェルプロンプトで、ntp.conf ファイルを/etc ディレクトリから/nsconfig ディレクトリにコピーします。ファイルが/nsconfig ディレクトリに既に存在する場合は、ntp.conf ファイルから次のエントリが削除されていることを確認します。

```
restrict localhost
restrict 127.0.0.2
```


これらのエントリは、デバイスをタイムサーバーとして実行する場合のみ必要となります。ただし、この機能は NetScaler アプライアンスではサポートされていません。

3. /nsconfig/ntp.conf を編集するには、ファイルのサーバの下の NTP サーバの IP アドレスを入力し、エントリを制限します。
4. /nsconfig ディレクトリに rc.netscaler という名前のファイルがない場合は作成します。
5. 次のエントリを追加して、/nsconfig/rc.netscaler /bin/sh /etc/ntpd_ctl full_startを編集します。

   このエントリは ntpd サービスを開始し、ntp.conf ファイルをチェックします。

   時間差が大きく、強制的に時刻を同期したくない場合は、日付を手動で設定してから ntpd を再び開始できます。シェルで次のコマンドを実行すると、アプライアンスとタイムサーバの時間差を確認できます。

```
ntpdate -q <IP address or domain name of the NTP server>
<!--NeedCopy-->
```

6. アプライアンスを再起動して、クロック同期を有効にします。

   注：アプライアンスを再起動せずに時刻同期を開始する場合は、シェルプロンプトで次のいずれかのコマンドを入力します。

```
/usr/sbin/ntpd -c /nsconfig/ntp.conf -g -p /var/run/ntpd.pid -l /
   var/log/ntpd.log &

or

/bin/sh /etc/ntpd_ctl full_start

<!--NeedCopy-->
```

DNS の構成

August 15, 2023

ADNS（Authoritative Domain Name Server）、DNS プロキシサーバー、エンドリゾルバー、またはフォワーダーとして機能するように、NetScaler アプライアンスを構成できます。SRV レコード、AAAA レコード、A レコード、MX レコード、NS レコード、CNAME レコード、PTR レコード、SOA レコードなど、DNS リソースレコードを追加できます。また、アプライアンスは外部 DNS サーバーの負荷を分散できます。

アプライアンスをフォワーダーとして構成する方法が一般的です。この構成では、外部ネームサーバーを追加する必要があります。外部ネームサーバーを追加したら、構成が正しいことを確認する必要があります。

外部ネームサーバーを追加、削除、有効化、および無効化することができます。IP アドレスを指定してネームサーバーを作成するか、既存の仮想サーバーをネームサーバーとして設定できます。

ネームサーバーを追加する場合は、IP アドレスまたは VIP（Virtual IP：仮想 IP）アドレスを指定できます。IP アドレスを使用する場合、アプライアンスはラウンドロビン方式で、構成したネームサーバーに要求を負荷分散します。VIP を使用する場合は、任意の負荷分散方式を指定できます。

CLI を使用してネームサーバーを追加します

コマンドプロンプトで次のコマンドを入力し、ネームサーバーを追加して構成を確認します。

- <add dns nameServer \<IP\>
- <show dns nameServer \<IP\>

  例

```
1  > add dns nameServer 10.102.29.10
2   Done
3  > show dns nameServer 10.102.29.10
4  1)        10.102.29.10  -  State: DOWN
5   Done
6
7  <!--NeedCopy-->
```

GUI を使用してネームサーバーを追加する

1. [**Traffic Management**] > [**DNS**] > [**Name Servers**] の順に選択します。
2. 詳細ペインで、[追加] をクリックします。
3. [**Create Name Server**] ダイアログボックスで、[**IP Address**] を選択します。
4. [**IP Address**] ボックスにネームサーバーの IP アドレス（たとえば、「10.102.29.10」）を入力します。外部ネームサーバーを追加する場合は、[**Local**] チェックボックスをオフにします。
5. [作成] をクリックし、[閉じる] をクリックします。
6. 追加したネームサーバーが [**Name Servers**] ペインに表示されることを確認します。

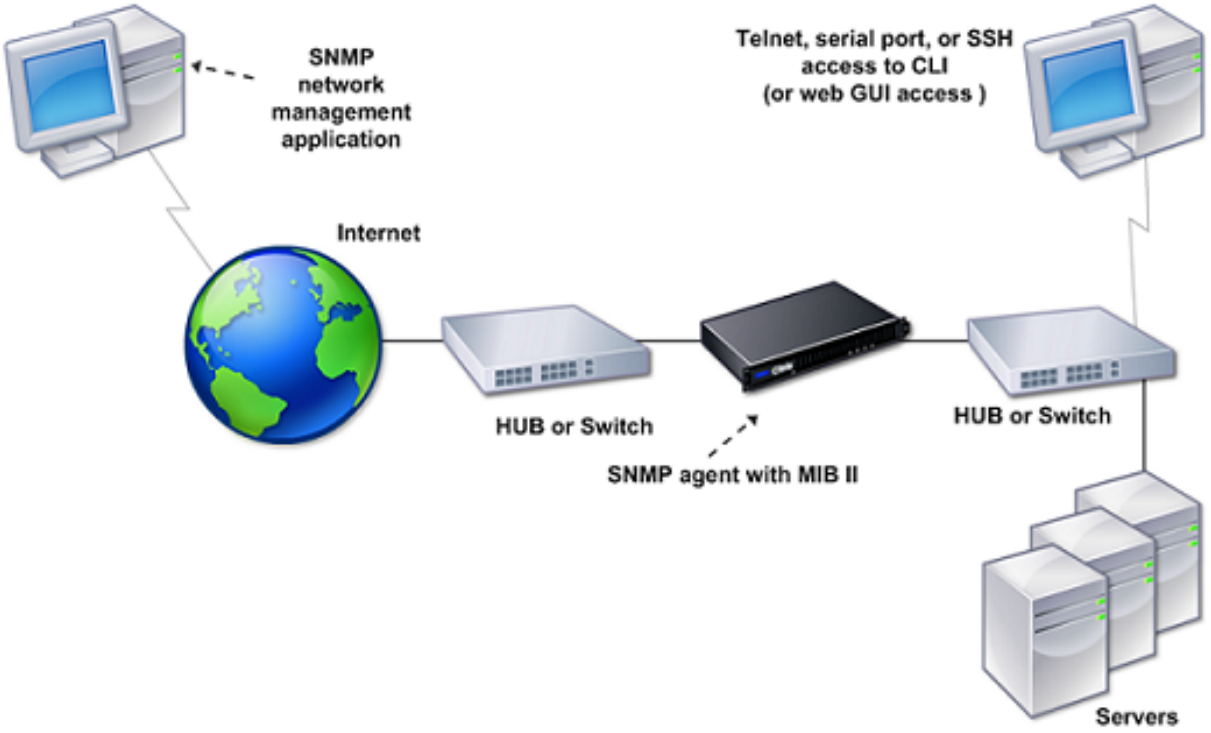
SNMP 構成

August 15, 2023

外部のコンピューターで実行されている SNMP（Simple Network Management Protocol：簡易ネットワーク管理プロトコル）ネットワーク管理アプリケーションは、NetScaler アプライアンスの SNMP エージェントにクエリを発行します。エージェントは、ネットワーク管理アプリケーションで要求されたデータを MIB（Management Information Base：管理情報ベース）で検索して、データをアプリケーションに送信します。

SNMP の監視では、トラップメッセージとアラームを使用します。SNMP トラップメッセージは、異常な状態を通知するためにエージェントが生成する非同期イベントです。このメッセージは、アラームによって通知されます。たとえば、CPU 使用率が 90% を超えたときに通知する場合は、その条件に対するアラームをセットアップできます。次の図は、SNMP が有効で設定済みの NetScaler アプライアンスが含まれたネットワークを示しています。

図 1: NetScaler アプライアンスの SNMP



NetScaler アプライアンスの SNMP エージェントは、SNMP version 1 (SNMPv1)、SNMP version 2 (SNMPv2)、および SNMP version 3（SNMPv3）をサポートしています。エージェントはバイリンガルモードで動作しているので、SNMPv2 クエリ（Get-Bulk など）と SNMPv1 クエリを処理できます。また、SNMP エージェントは SNMPv2 に準拠したトラップを送信し、counter64 などの SNMPv2 データタイプをサポートしています。SNMPv1 マネージャー（ADC アプライアンスからの SNMP 情報を要求する、他のサーバー上のプログラム）は、SNMP クエリを処理するときに NS-MIB-smiv1.mib ファイルを使用します。SNMPv2 マネージャーは、NS-MIB-smiv2.mib ファイルを使用します。

NetScaler アプライアンスは、次のエンタープライズ固有の MIB をサポートしています。

- 標準 MIB-2 グループのサブセット。MIB-2 グループの SYSTEM、IF、ICMP、UDP、および SNMP を提供します。
- システムエンタープライズ MIB。システム固有の設定と統計情報を提供します。

SNMP の設定には、SNMP エージェントにクエリを発行できるマネージャーを指定し、SNMP トラップメッセージを受信する SNMP トラップリスナーを追加し、SNMP アラームを設定する作業が含まれます。

SNMP マネージャーを追加する

SNMP Version 1、2、または 3 に準拠する管理アプリケーションを実行するワークステーションを構成して、アプライアンスにアクセスできます。このようなワークステーションは、「SNMP マネージャー」と呼ばれます。アプライアンスに SNMP マネージャーを指定しない場合、アプライアンスは、ネットワーク上のすべての IP アドレスから SNMP クエリを受け付けて応答します。1 つまたは複数の SNMP マネージャーを設定する場合、アプライアンスは、それらの特定の IP アドレスからのみ SNMP クエリを受け付けて応答します。SNMP マネージャーの IP アドレスを指定する場合は、ネットマスクパラメーターを使用して、サブネット全体からのアクセス権を付与できます。最大 100 個の SNMP マネージャーまたはネットワークを追加できます。CLI を使用して SNMP マネージャーを追加するには

コマンドプロンプトで次のコマンドを入力し、SNMP マネージャーを追加して構成を確認します。

```
add snmp manager <IPAddress> ... [-netmask <netmask>]
show snmp manager <IPAddress>
```

例**:**

```
1 add snmp manager 10.102.29.5 -netmask 255.255.255.255
2 Done
3 show snmp manager 10.102.29.5
4 10.102.29.5 255.255.255.255
5 Done
6 <!--NeedCopy-->
```

GUI を使用して **SNMP** マネージャーを追加するには:

1. ナビゲーションペインで ［**System**］、［**SNMP**］の順に展開して、［**Managers**］をクリックします。
2. 詳細ペインで、[追加] をクリックします。
3. ［**Add SNMP Manager**］ダイアログボックスの ［**IP Address**］ボックスに、管理アプリケーションを実行しているワークステーションの IP アドレス（たとえば、「10.102.29.5」）を入力します。
4. [作成] をクリックし、[閉じる] をクリックします。
5. 追加した SNMP トラップが、ペインの下部にある ［**Details**］セクションに表示されていることを確認します。

SNMP トラップリスナーの追加

アラームを構成したら、アプライアンスによるトラップメッセージの送信先となるトラップリスナーを指定します。トラップリスナーの IP アドレスや宛先ポートなどのパラメーターを指定する以外に、トラップの種類（汎用または専用）と SNMP のバージョンを指定できます。

汎用または専用のトラップを受信するために、最大 20 のトラップリスナーを構成できます。

CLI を使用して **SNMP** トラップリスナーを追加するには

コマンドプロンプトで次のコマンドを入力し、SNMP トラップを追加して構成を確認します。

- `add snmp trap specific <IP>`
- `show snmp trap`

例:

```
Trap type: SPECIFIC
Destination IP: 10.102.29.3
TD: 0
Destination Port: 162
Source IP: NetScaler IP
Version: V2
Min-Severity: -
AllPartition: DISABLED
Community: public
<!--NeedCopy-->
```

GUI を使用して **SNMP** トラップリスナーを追加するには

1. ナビゲーションペインで、[システム]、[**SNMP**] の順に展開し、[トラップ] をクリックします。
2. 詳細ペインで、[追加] をクリックします。
3.「**SNMP** トラップ送信先の作成」ダイアログ・ボックスの「送信先 **IP** アドレス」テキスト・ボックスに、IP アドレス (10.102.29.3 など) を入力します。
4. [作成] をクリックし、[閉じる] をクリックします。
5. 追加した **SNMP** トラップがペインの下部にある詳細セクションに表示されていることを確認します。

SNMP アラームを構成する

いずれかのアラームに該当するイベントが発生した場合にアプライアンスがトラップメッセージを生成するように構成できます。アラームを構成するには、アラームを有効にして、トラップを生成する重要度レベルを設定します。Critical、Major、Minor、Warning、および Informational という、5 つの重要度レベルがあります。アラームの重要度が、トラップに指定した重要度と一致する場合のみ、トラップが送信されます。

一部のアラームは、デフォルトで有効になっています。SNMP アラームを無効にすると、該当するイベントが発生してもアプライアンスはトラップメッセージを生成しません。たとえば、Login-Failure SNMP アラームを無効にすると、ログインが失敗してもアプライアンスはトラップメッセージを生成しません。

CLI を使用してアラームを有効または無効にするには

コマンドプロンプトで次のコマンドを入力し、アラームを有効または無効にして、有効または無効になっていることを確認します。

set snmp alarm [-state ENABLED DISABLED]

-
- show snmp alarm \< トラップ名\>

  例

```
set snmp alarm LOGIN-FAILURE -state ENABLED
Done
show snmp alarm LOGIN-FAILURE
Alarm Alarm Threshold Normal Threshold Time State Severity Logging
----- --------------- ---------------- ---- -------- --------- --------
LOGIN-FAILURE N/A N/A N/A ENABLED - ENABLED
Done
<!--NeedCopy-->
```

CLI を使用してアラームの重大度を設定するには

コマンドプロンプトで次のコマンドを入力し、アラームの重要度を設定して重要度が正しく設定されていることを確認します。

- `set snmp alarm <trapName> [-severity <severity>]`
- `show snmp alarm <trapName>`

例**:**

```
set snmp alarm LOGIN-FAILURE -severity Major
Done
show snmp alarm LOGIN-FAILURE
Alarm Alarm Threshold Normal Threshold Time State Severity Logging
----- --------------- ---------------- ---- -------- --------- --------
LOGIN-FAILURE N/A N/A N/A ENABLED Major ENABLED
Done
<!--NeedCopy-->
```

GUI を使用してアラームを構成するには

1. ナビゲーションペインで、[システム]、[SNMP] の順に展開し、[アラーム] をクリックします。
2. 詳細ペインでアラーム **(LOGIN-FAILURE** など**)** を選択し、「開く」をクリックします。
3. 「**SNMP** アラームの設定」ダイアログ・ボックスで、アラームを有効にするには、「状態」ドロップダウン・リストで「有効」を選択します。アラームを無効にするには、[Disabled] を選択します。
4. 「重要度」ドロップダウンリストで、重要度オプション (「メジャー」など) を選択します。
5. 「**OK**」をクリックし、「閉じる」をクリックします。

6. ペインの下部にある「詳細」セクションを表示して、設定した SNMP アラームのパラメータが正しく設定されていることを確認します。

構成を確認する

August 15, 2023

システムの設定が終了したら、次のチェックリストに記入して設定を確認します。

設定チェックリスト

- 実行中のビルド：
- 非互換性の問題はない（非互換性の問題は対象ビルドのリリースノートに記載されています）。
- ポート設定（速度、二重、フロー制御、監視）がスイッチのポートと同じである。
- ピーク時にすべてのサーバー側接続をサポートするように、SNIP アドレスが十分に設定されている。
  - 設定済みの SNIP IP アドレス数：__
  - 予想される同時サーバー接続数：
    [ ] 62,000 [ ] 124,000 [ ] Other_____

トポロジ設定チェックリスト

ルートを使用して、他のサブネット上のサーバーを解決した。

入力したルート

- NetScaler アプライアンスがパブリック-プライベートトポロジの場合、RNAT（Reverse NAT：リバース NAT）が設定されている。
- ADC アプライアンスで設定されたフェールオーバー（高可用性）設定が、ワンアームまたはツーアーム構成で解決される。使用されないネットワークインターフェイスをすべて無効化：

- ADC アプライアンスが外部負荷分散装置の後ろに配置されている場合、外部負荷分散装置の負荷分散ポリシーが「least connection」ではない。

  外部負荷分散装置に設定されている負荷分散ポリシー：

- ADC アプライアンスがファイアウォールの前に配置されている場合、ファイアウォールのセッションタイムアウトが 300 秒以上に設定されている。

  注：NetScaler アプライアンスでの TCP アイドル接続のタイムアウトは 360 秒です。ファイアウォール上でも 300 秒以上のタイムアウトが設定されている場合、接続が先に閉じられないためにアプライアンスで TCP 接続の多重化が行われることがあります。

  セッションタイムアウトに設定されている値：_____

サーバー設定チェックリスト

- 「キープアライブ」がすべてのサーバーで有効になっている。

  キープアライブタイムアウトに設定されている値：_____

- デフォルトゲートウェイが正しい値に設定されている（デフォルトゲートウェイは、NetScaler アプライアンスまたはアップストリームルーターにする必要があります）。デフォルトゲートウェイ：

- サーバーのポート設定（速度、二重、フロー制御、監視）がスイッチのポート設定と同じである。

- Microsoft® Internet Information Server を使用している場合、バッファリングがサーバーで有効になっている。
- Apache Server を使用している場合、MaxConn（最大接続数）パラメーターがサーバーと NetScaler アプライアンスで設定されている。

  設定されている MaxConn（最大接続数）の値：

- Netscape® Enterprise Server™ を使用している場合、接続ごとの最大要求数パラメーターが NetScaler アプライアンスで設定されている。設定されている接続ごとの最大要求数の値：

ソフトウェア機能の設定チェックリスト

- レイヤー 2 モード機能を無効にする必要があるかどうか（別のレイヤー 2 デバイスが NetScaler アプライアンスと並行して動作している場合は、無効にしてください）。

  有効または無効にする理由

- MAC ベース転送機能を無効にする必要があるかどうか（リターントラフィックが使用する MAC アドレスが異なる場合は、無効にする必要があります）。

  有効または無効にする理由

- ホストベースの再使用を無効にする必要があるかどうか（サーバーに仮想ホストがあるかどうか）。

  有効または無効にする理由

- サージ保護機能のデフォルト設定を変更する必要があるかどうか。

  設定を変更または保持する理由

アクセスチェックリスト

- クライアント側ネットワークから、システム IP の ping を実行できる。
- サーバー側ネットワークから、システム IP の ping を実行できる。
- NetScaler 経由で、管理対象サーバーの ping を実行できる。
- 管理対象サーバーから、インターネットホストの ping を実行できる。
- ブラウザーを介して、管理対象サーバーにアクセスできる。
- ブラウザーを使用して、管理対象サーバーからインターネットにアクセスできる。
- SSH を使用してシステムにアクセスできる。
- すべての管理対象サーバーへの管理者アクセスが機能している。

> 注: ping ユーティリティを使用している場合は、ping されるサーバーで ICMP ECHO を有効にしてください。そうしないと、ping が失敗します。

ファイアウォールチェックリスト

次のファイアウォール要件が満たされている。

- UDP 161（SNMP）
- UDP 162（SNMP トラップ）
- TCP/UDP 3010（GUI）
- HTTP 80（GUI）
- TCP 22（SSH）

NetScaler アプライアンスにおけるトラフィックの負荷分散

August 15, 2023

負荷分散機能は、クライアント要求を複数のサーバーに分散して、リソース使用率を最適化します。限られた数のサーバーが多数のクライアントにサービスを提供する実際のシナリオでは、サーバーが過負荷になり、サーバーファームのパフォーマンスが低下する可能性があります。NetScaler アプライアンスは、負荷分散基準を使用して、各クライアント要求を、要求が到着したときに処理するのに最適なサーバーに転送することにより、ボトルネックを防ぎます。

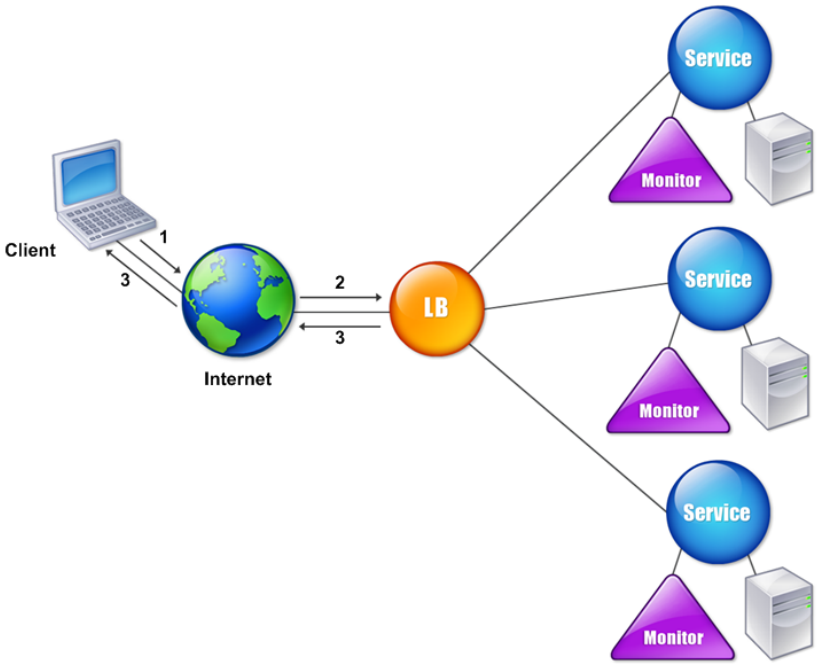
負荷分散を構成するには、サーバーファーム内の複数のサーバーをプロキシし、それらの間で負荷を分散する仮想サーバーを定義します。

クライアントがサーバーへの接続を開始すると、仮想サーバーはクライアント接続を終了し、選択したサーバーとの新しい接続を開始するか、サーバーとの既存の接続を再利用して負荷分散を実行します。負荷分散機能は、レイヤー 4（TCP および UDP）からレイヤー 7（FTP、HTTP、および HTTPS）までのトラフィック管理を提供します。

NetScaler アプライアンスは、負荷分散方法と呼ばれるいくつかのアルゴリズムを使用して、サーバー間で負荷を分散する方法を決定します。デフォルトの負荷分散方法は、最小接続方法です。

一般的な負荷分散の展開は、次の図で説明するエンティティで構成されます。

図 1：負荷分散アーキテクチャ



エンティティは次のように機能します。

- 仮想サーバー。IP アドレス、ポート、およびプロトコルで表されるエンティティ。仮想サーバーの IP アドレス（VIP）は通常、パブリック IP アドレスです。クライアントはこの IP アドレスに接続要求を送信します。仮想サーバーは、サーバーのバンクを表します。
- **Service**. サーバーまたはサーバー上で実行されているアプリケーションの論理表現。サーバーの IP アドレス、ポート、およびプロトコルを識別します。サービスは仮想サーバーにバインドされています。
- サーバーオブジェクト。IP アドレスで表されるエンティティ。サーバーオブジェクトは、サービスを作成するときに作成されます。サービスの IP アドレスは、サーバーオブジェクトの名前として使用されます。サーバーオブジェクトを作成してから、サーバーオブジェクトを使用してサービスを作成することもできます。
- モニター。サービスの状態を追跡するエンティティ。アプライアンスは、各サービスにバインドされたモニターを使用してサーバーを定期的にプローブします。サーバーが指定された応答タイムアウト内に応答せず、指定された数のプローブが失敗した場合、サービスは DOWN とマークされます。次に、アプライアンスは残りのサービス間で負荷分散を実行します。

負荷分散

August 15, 2023

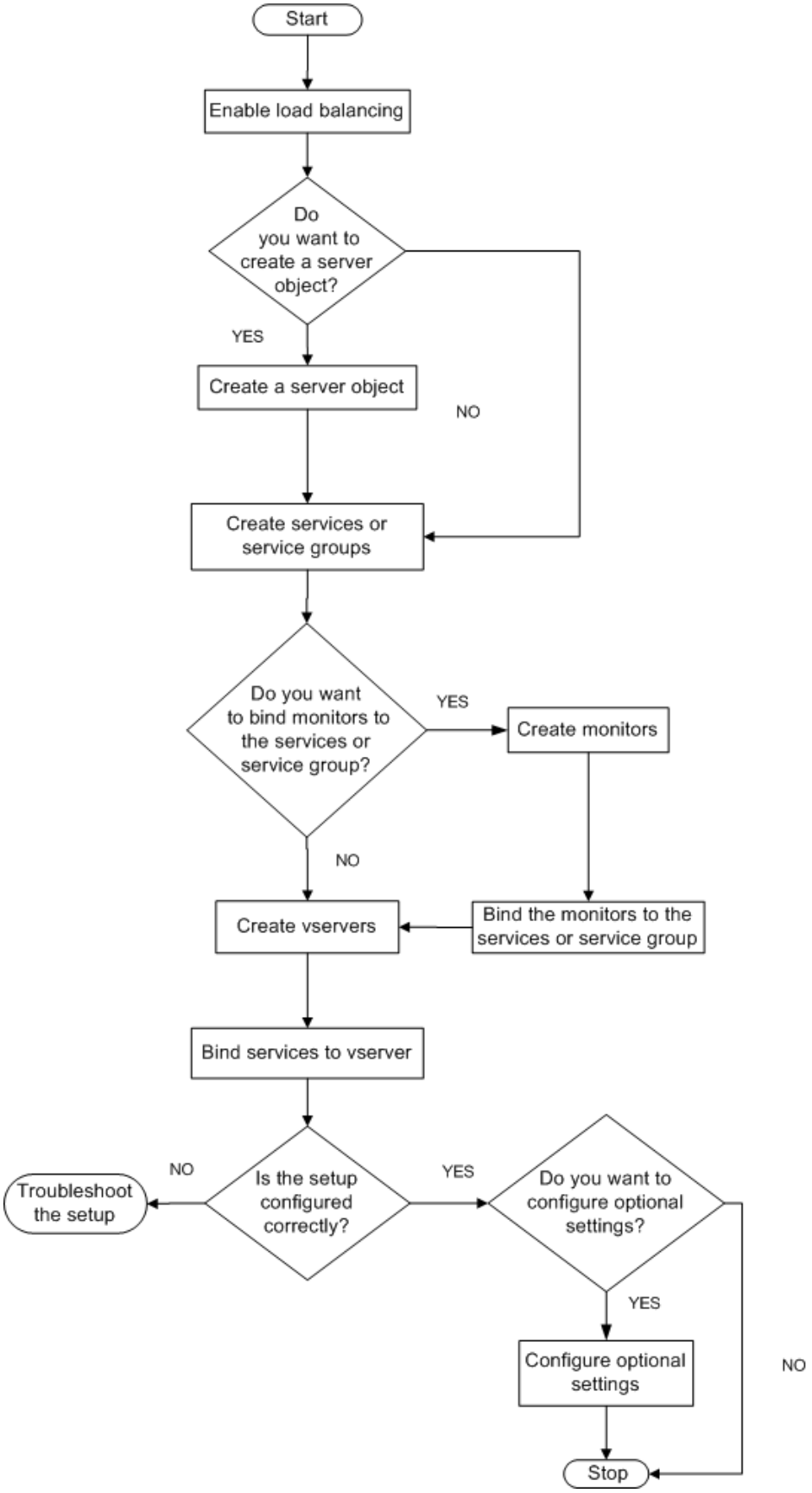
負荷分散を構成するには、まずサービスを作成する必要があります。次に、仮想サーバーを作成して、サービスをその仮想サーバーにバインドします。デフォルトでは、NetScaler アプライアンスは各サービスに 1 つのモニターをバインドします。サービスをバインドしたら、すべての構成内容が正しいことを確認します。

> 注：構成を適用した後で、各エンティティがどのように実行されているかを示す統計情報を表示できます。統計ユーティリティ、または
> stat lb vserver < 仮想サーバー名 > コマンドを使用します。仮想サーバー名 >

オプションで、サービスに重要度（Weight）を割り当てることができます。割り当てられた重要度に基づいてサービスが負荷分散されます。ただし、負荷分散機能の導入時には詳細な重要度を構成せずに、基本的なパーシステンス設定（特定サーバーへの接続の保持）と構成保護設定のみを行えます。

次のフローチャートは、一連の構成タスクを示しています。

図 1：負荷分散を構成するための一連のタスク



負荷分散を有効にする

負荷分散を構成する前に、負荷分散機能が有効になっているか確認します。

CLI を使用して負荷分散を有効にするには

コマンドプロンプトで次のコマンドを入力し、負荷分散を有効にして構成を確認します。

- enable feature lb
- show feature

例

“‘ pre codeblock

> enable feature lb
> Done
> show feature

```
1   Feature                  Acronym          Status
2   -------                  -------          ------   1)  Web
       Logging               WL               OFF   2)  Surge
       Protection         SP                OFF   3)  Load Balancing
                  LB                ON   .   .   .   9)  SSL
       Offloading            SSL              ON   .   .   .  Done
       <!--NeedCopy--> ```
```

GUI を使用して負荷分散を有効にするには

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. 詳細ペインの [Modes and Features] で、[Change basic features] をクリックします。
3. [Configure Basic Features] ダイアログボックスで、[Load Balancing] チェックボックスをオンにして [OK] をクリックします。
4. 「Enable/Disable Feature(s)?」メッセージが表示されたら、[はい] をクリックします。

サービスと仮想サーバーの構成

負荷分散するサービスを特定したら、負荷分散の初回構成を実装できます。これを行うには、サービスオブジェクトと負荷分散仮想サーバーを作成して、それらをバインドします。

CLI を使用して初期負荷分散構成を実装するには

コマンドプロンプトで次のコマンドを入力し、初回構成を実装して確認します。

- <add service \< 名前\> \<IP アドレス\> \< サービスタイプ\> \< ポート\>
- <add lb vserver \< 仮想サーバー名\> \< サービスタイプ\> \[\<IP アドレス\> \< ポート\>\]
- <bind lb vserver \< 名前\> \< サービス名\>
- <show service bindings \< サービス名\>

  例

```
> add service service-HTTP-1 10.102.29.5 HTTP 80
 Done
> add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
 Done
> bind lb vserver vserver-LB-1 service-HTTP-1
 Done
> show service bindings service-HTTP-1
   service-HTTP-1 (10.102.29.5:80) - State : DOWN

     1)      vserver-LB-1 (10.102.29.60:80) - State : DOWN
 Done
<!--NeedCopy-->
```

GUI を使用して初期負荷分散構成を実装するには

1. ［Traffic Management］［Load Balancing］の順に選択します。
2. 詳細ペインの［Getting Started］で、［Load Balancing wizard］をクリックし、ウィザードの説明に従って基本的な負荷分散セットアップを作成します。
3. ナビゲーションペインに戻り、［Load Balancing］を展開して、［Virtual Servers］をクリックします。
4. 構成した仮想サーバーを選択し、ページの下部に表示されるパラメーターが正しく構成されていることを確認します。
5. ［開く］をクリックします。
6. ［Services］タブの各サービスで［Active］チェックボックスがオンになっていることを確認し、各サービスが仮想サーバーにバインドされていることを確認します。

パーシステンス設定

August 15, 2023

仮想サーバーにより実行されるサービスへの接続を維持したい場合（電子商取引で使用される接続など）は、その仮想サーバーに対してパーシステンスを構成する必要があります。アプライアンスは、まず構成されている負荷分散方

式に基づいてサーバーを選択しますが、それ以降は同じクライアントからのすべての要求を同じサーバーに転送します。

パーシステンスを構成すると、サーバーの初回選択時以降の要求で、負荷分散方式が無視されます。構成したパーシステンスの適用先サービスがダウンしている場合は、負荷分散方式に基づいて新しいサービスが選択され、同じクライアントからのそれ以降の要求はそのサービスに永続的に割り当てられます。選択したサービスが Out Of Service 状態の場合、未処理の要求の処理は続行されますが、新しい要求や接続は受け付けられません。シャットダウン期間が経過すると、既存の接続が閉じます。次の表は、設定できるパーシステンスの種類を示しています。

永続性タイプ	固定接続数
Source IP、SSL Session ID、Rule、DESTIP、SRCIPDESTIP	250K*
CookieInsert、URL passive、Custom Server ID	メモリの上限。CookieInsert の場合、タイムアウトが 0 でなければ、メモリの上限に達するまで任意の数の接続が許可されます。

前の表の「*」は次のことを指しています。

コアあたり 250K セッションは、パケットエンジンあたりのデフォルトです。パケットエンジンあたり 100 万のセッションエントリを設定するには、次のコマンドを実行します。

```
set lb parameter -sessionsthreshold <1000000*number of PE>
```

3 PE システムの場合は、次のコマンドを実行します。

```
set lb parameter -sessionsthreshold 3000000
```

表 1. 同時持続的接続の数の制限

アプライアンスのリソース不足により構成済みのパーシステンスが維持できない場合は、負荷分散方式に基づいてサーバーが選択されます。パーシステンスは、その種類で構成された時間だけ保持されます。一部のパーシステンスの種類は、特定の仮想サーバーに固有です。次の表は、それらの関係を示しています。

Persistence TypeHeader 1	**HTTP**	**HTTPS**	**TCP**	**UDP/IP**	**SSL_Bridge**
接続元 IP	はい	はい	はい	はい	はい
CookieInsert	はい	はい	いいえ	いいえ	いいえ
SSL Session ID	いいえ	はい	いいえ	いいえ	はい
URL Passive	はい	はい	いいえ	いいえ	いいえ
Custom Server ID	はい	はい	いいえ	いいえ	いいえ

Persistence TypeHeader 1	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge
規則	はい	はい	いいえ	いいえ	いいえ
SRCIPDESTIP	-	-	はい	はい	-
DESTIP	-	-	はい	はい	-

表 2. 仮想サーバーのタイプごとに使用可能な永続性タイプ

仮想サーバーのグループに対して、パーシステンスを指定することもできます。グループに対してパーシステンスを有効にすると、クライアント要求を受信した仮想サーバーに関係なく、クライアント要求は同じサーバーに送信されます。パーシステンスの構成時間が経過すると、着信したクライアント要求に対して、グループの任意の仮想サーバーが選択されます。

一般的に使用される 2 つのパーシステンスの種類は、Cookie に基づくパーシステンスと URL のサーバー ID に基づくパーシステンスです。

Cookie に基づくパーシステンスの設定

Cookie に基づく永続性を有効にすると、Citrix **ADC** アプライアンスは **HTTP** 応答の **Set-Cookie** ヘッダーフィールドに **HTTP** Cookie を追加します。Cookie には、HTTP 要求の送信先のサービスに関する情報が含まれています。クライアントは Cookie を保存して、それ以降のすべての要求に含めます。ADC は Cookie を使用して、これらの要求に対するサービスを選択します。HTTP または HTTPS タイプの仮想サーバーに対して、この種類のパーシステンスを使用できます。

NetScaler アプライアンスは、<NSC_XXXX>= <ServiceIP> <ServicePort> の Cookie を挿入します。

各項目の意味は次のとおりです:

- «NSC_XXXX> は、仮想サーバー名から導出される仮想サーバー ID です。
- «ServiceIP> は、サービスの IP アドレスの 16 進数値です。
- «ServicePort> は、サービスのポートの 16 進数値です。

`useEncryptedPersistenceCookie`オプションを有効にすると、ADC はクッキーを挿入するときに SHA2 ハッシュアルゴリズムを使用して ServiceIP と ServicePort を暗号化し、Cookie を受信すると復号化します。

注: クライアントが HTTP Cookie を保存できない場合は、以降の要求に HTTP Cookie が含まれなくなり、パーシステンスは適用されません。

デフォルトでは、ADC アプライアンスは Netscape 仕様に準拠して、HTTP Cookie バージョン 0 を送信します。また、RFC 2109 に準拠して、バージョン 1 を送信することもできます。

HTTP Cookie に基づくパーシステンスに対して、タイムアウト値を設定できます。以下の点に注意してください:

- HTTPCookie バージョン 0 を使用する場合、NetScaler アプライアンスは、ADC アプライアンス上の現在の GMT 時間とタイムアウト値の合計として計算された Cookie の有効期限（HTTP クッキーの期限切れ属性）の絶対協定世界時（GMT）を挿入します。
- HTTP Cookie バージョン 1 が使用されている場合、ADC アプライアンスは相対有効期限（HTTP Cookie の Max-Age 属性）を挿入します。この場合、クライアントソフトウェアが実際の有効期限を計算します。

注: 現在インストールされているほとんどのクライアントソフトウェア（Microsoft Internet Explorer と Netscape ブラウザー）は、HTTP Cookie バージョン 0 を理解しますが、一部の HTTP プロキシは HTTP Cookie バージョン 1 を理解します。

タイムアウト値を 0 に設定すると、使用されている HTTP Cookie バージョンに関係なく、ADC アプライアンスは有効期限を指定しなくなります。この場合、有効期限はクライアントソフトウェアに依存し、そのような Cookie は、そのソフトウェアがシャットダウンすると、無効になります。この種類のパーシステンスはシステムリソースを消費しません。したがって、好きな数だけ永続的なクライアントを含めることができます。

管理者は HTTPCookie のバージョンを変更できます。

CLI を使用して **HTTPCookie** のバージョンを変更するには

コマンドプロンプトで次を入力します。

```
set ns param [-cookieversion ( 0 | 1 )]
<!--NeedCopy-->
```

例**:**

```
set ns param -cookieversion 1
<!--NeedCopy-->
```

GUI を使用して **HTTPCookie** のバージョンを変更するには

1. ［**System**］**>** ［**Settings**］に移動します。
2. 詳細ペインで、［Change HTTP Parameters］をクリックします。
3. ［Configure HTTP Parameters］ダイアログボックスの［Cookie］で、［Version 0］または［Version 1］を選択します。

注: パラメーターの詳細については、「Cookie に基づくパーシステンスの構成」を参照してください。

CLI を使用して **Cookie** に基づいて永続性を構成するには

コマンドプロンプトで次のコマンドを入力し、Cookie に基づくパーシステンスを構成して確認します。


```
set lb vserver <name> -persistenceType COOKIEINSERT

show lb vserver <name>
<!--NeedCopy-->
```

例:

```
set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
Done
show lb vserver vserver-LB-1
    vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
    .
    .
    .
    Persistence: COOKIEINSERT (version 0)
    Persistence Timeout: 2 min
    .
    .
    .
 Done
<!--NeedCopy-->
```

GUI を使用して **Cookie** に基づいて永続性を構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ペインで、パーシステンスを設定する仮想サーバー（たとえば、vserver-LB-1）を選択し、[Open] をクリックします。
3. [Configure Virtual Server (Load Balancing)] ダイアログボックスの [Method and Persistence] タブにある [Persistence] リストで、[COOKIEINSERT] を選択します。
4. [Time-out (min)] テキストボックスに、タイムアウト値（たとえば、「2」）を入力します。
5. 「OK」をクリックします。
6. パーシステンスを設定した仮想サーバーを選択して、ペインの下部にある [Details] セクションを表示し、仮想サーバーが正しく構成されていることを確認します。

URL のサーバー ID に基づくパーシステンスの構成

NetScaler アプライアンスは、URL に含まれるサーバー ID に基づいてパーシステンスを保持できます。「URL パッシブパーシステンス」と呼ばれる方法では、ADC はサーバー応答からサーバー ID を抽出して、クライアント要求の URL クエリに埋め込みます。サーバー ID は IP アドレスで、ポートは 16 進数で指定します。ADC は、以降のクライアント要求からサーバー ID を抽出し、それを使用してサーバーを選択します。

URL パッシブパーシステンスでは、ペイロード式またはポリシーインフラストラクチャ式を設定し、クライアント要求に含まれるサーバー ID の場所を指定する必要があります。式の詳細については、ポリシーの設定とリファレンスを参照してください。

注：サーバー ID をクライアント要求から抽出できない場合、サーバーの選択は負荷分散方式に基づいて行われます。

例：ペイロード式

URLQUERY contains sid= という式は、トークン sid= に一致した後、クライアント要求の URL クエリからサーバー ID を抽出するようにシステムを設定します。したがって、URL`http://www.citrix.com/index.asp?\\&sid;=c0a864100050`を含むリクエストは、IP アドレス 10.102.29.10 とポート 80 のサーバーに送信されます。

タイムアウト値は、この種類のパーシステンスには影響しません。このパーシステンスは、サーバー ID がクライアント要求から抽出できる限り維持されます。この種類のパーシステンスはシステムリソースを消費しないため、保持されるクライアント数に制限はありません。

> 注: パラメータの詳細については、「負荷分散」を参照してください。

CLI を使用して **URL** のサーバー **ID** に基づいて永続性を構成するには

コマンドプロンプトで次のコマンドを入力し、URL のサーバー ID に基づくパーシステンスを構成して確認します。

```
set lb vserver <name> -persistenceType URLPASSIVE

<show lb vserver <name>
<!--NeedCopy-->
```

例**:**

```
set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
Done
show lb vserver vserver-LB-1
    vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
    .
    .
    .
    Persistence: URLPASSIVE
    Persistence Timeout: 2 min
    .
    .
    .
Done
<!--NeedCopy-->
```

GUI を使用して **URL** のサーバー **ID** に基づいて永続性を構成するには

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ペインで、パーシステンスを設定する仮想サーバー（たとえば、vserver-LB-1）を選択し、［Open］をクリックします。

3. [Configure Virtual Server (Load Balancing)] ダイアログボックスの [Method and Persistence] タブにある [Persistence] リストで、[URLPASSIVE] を選択します。
4. [Time-out (min)] テキストボックスに、タイムアウト値（たとえば、「2」）を入力します。
5. [Rule] ボックスに、有効な式を入力します。また、[Rule] ボックスの横にある [Configure] をクリックし、[Create Expression] ダイアログボックスを使用して式を作成します。
6. 「OK」をクリックします。
7. パーシステンスを設定した仮想サーバーを選択して、ペインの下部にある [Details] セクションを表示し、仮想サーバーが正しく構成されていることを確認します。

負荷分散設定を保護する機能の構成

August 15, 2023

正しく動作していない仮想サーバーに関する通知を提供するように URL リダイレクトを構成できます。また、プライマリ仮想サーバーが使用できなくなった場合にその役割を引き継ぐバックアップ仮想サーバーを構成することもできます。

URL リダイレクトの構成

HTTP または HTTPS タイプの仮想サーバーがダウンしたり無効になったりしたときに、アプライアンスのステータスを通信するためのリダイレクト URL を構成できます。この URL は、ローカルリンクでもリモートリンクでも構いません。アプライアンスでは、HTTP 302 リダイレクトが使用されます。

リダイレクトは、絶対 URL でも相対 URL でも構いません。構成したリダイレクト URL に絶対 URL が含まれている場合、着信した HTTP 要求で指定された URL に関係なく、その絶対 URL にリダイレクトされます。構成したリダイレクト URL にドメイン名のみが含まれている場合（相対 URL）、そのドメインに着信 URL を追記した場所にリダイレクトされます。

注：負荷分散仮想サーバーで、バックアップ仮想サーバーとリダイレクト URL の両方を構成した場合、バックアップ仮想サーバーがリダイレクト URL よりも優先されます。この場合は、プライマリおよびバックアップ仮想サーバーの両方がダウンしているときに、リダイレクトが使用されます。

CLI を使用してクライアント要求を URL へリダイレクトするように仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力し、クライアント要求が URL にリダイレクトされるように仮想サーバーを構成して確認します。

```
set lb vserver <name> -redirectURL <URL>

show lb vserver <name>
```



```
<!--NeedCopy-->
```

例:

```
    > set lb vserver vserver-LB-1 -redirectURL <http://www.newdomain.
       com/mysite/maintenance>
     Done
    > show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jun 17 08:56:34 2009 (+666 ms)
        .
        .
        .
        Redirect URL: <http://www.newdomain.com/mysite/maintenance>
        .
        .
        .
     Done
    >
<!--NeedCopy-->
```

GUI を使用してクライアント要求を **URL** へリダイレクトするように仮想サーバーを構成するには

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、URL リダイレクトを構成する仮想サーバー（たとえば、vserver-LB-1）を選択し、[Open] をクリックします。
3. [Configure Virtual Server (Load Balancing)]ダイアログボックスで、[Advanced]タブの[Redirect URL]テキストボックスに、URL(たとえば、「`http://www.newdomain.com/mysite/maintenance`」）を入力して［OK］をクリックします。
4. サーバーに設定したリダイレクト URL が、ペインの下部にある［Details］セクションに表示されていることを確認します。

バックアップ仮想サーバーの設定

プライマリ仮想サーバーがダウンしているか無効である場合、アプライアンスは接続またはクライアント要求をバックアップ仮想サーバーに送信し、クライアントトラフィックをバックアップ仮想サーバーからサービスに転送できます。アプライアンスは、サイトの停止またはメンテナンスに関する通知メッセージをクライアントに送信することもできます。バックアップ仮想サーバーはプロキシであり、クライアントに対して透過的です。

仮想サーバーを作成したり、既存の仮想サーバーのオプションパラメーターを変更したりする場合は、バックアップ仮想サーバーを構成できます。また、既存のバックアップ仮想サーバーに対してバックアップ仮想サーバーを構成し、カスケードされたバックアップ仮想サーバーを作成することもできます。バックアップ仮想サーバーをカスケードする最大の深さは、10 です。アプライアンスは、起動しているバックアップ仮想サーバーを検索し、その仮想サーバーにアクセスしてコンテンツを提供します。

プライマリおよびバックアップ仮想サーバーがダウンしたり、それらのサーバーの処理要求数がしきい値に達したりしたときに、プライマリで URL がリダイレクトされるように構成できます。

注：バックアップ仮想サーバーが存在しない場合は、リダイレクト URL を構成しないとエラーメッセージが表示されます。バックアップ仮想サーバーとリダイレクト URL の両方が構成されている場合は、バックアップ仮想サーバーが優先されます。

CLI を使用してバックアップ仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力し、バックアップサーバーを構成して確認します。

```
set lb vserver <name> [-backupVserver <string>]

show lb vserver <name>
<!--NeedCopy-->
```

例**:**

```
    > set lb vserver vserver-LB-1 -backupVserver vserver-LB-2
     Done
    > show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jun 17 08:56:34 2009 (+661 ms)
        .
        .
        .
        Backup: vserver-LB-2
        .
        .
        .
     Done
    >
<!--NeedCopy-->
```

GUI を使用してバックアップ仮想サーバーをセットアップするには

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ペインで、バックアップ仮想サーバーを設定する仮想サーバー（たとえば、vserver-LB-1）を選択し、［Open］をクリックします。
3. ［Configure Virtual Server (Load Balancing)］ダイアログボックスの［Advanced］タブにある［Backup Virtual Server］リストで、バックアップ仮想サーバー（たとえば、vserver-LB-2）を選択し、［OK］をクリックします。
4. 設定したバックアップ仮想サーバーが、ペインの下部にある［Details］セクションに表示されていることを確認します。

注：ダウンしたプライマリサーバーが復帰した場合でも、その仮想サーバーをプライマリとして明示的に再設定するまでバックアップ仮想サーバーがプライマリサーバーとして動作するようにするには、［Disable Primary When Down］チェックボックスをオンにします。

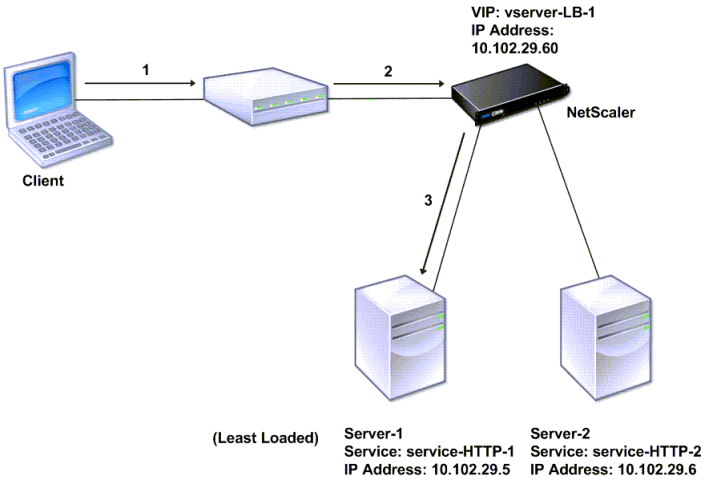
一般的な負荷分散シナリオ

August 15, 2023

負荷分散セットアップでは、NetScaler アプライアンスはクライアントとサーバーファームの間に論理的に配置され、サーバーへのトラフィックフローを管理します。

次の図は、基本的な負荷分散構成のトポロジを示しています。

図 1：基本的な負荷分散トポロジ

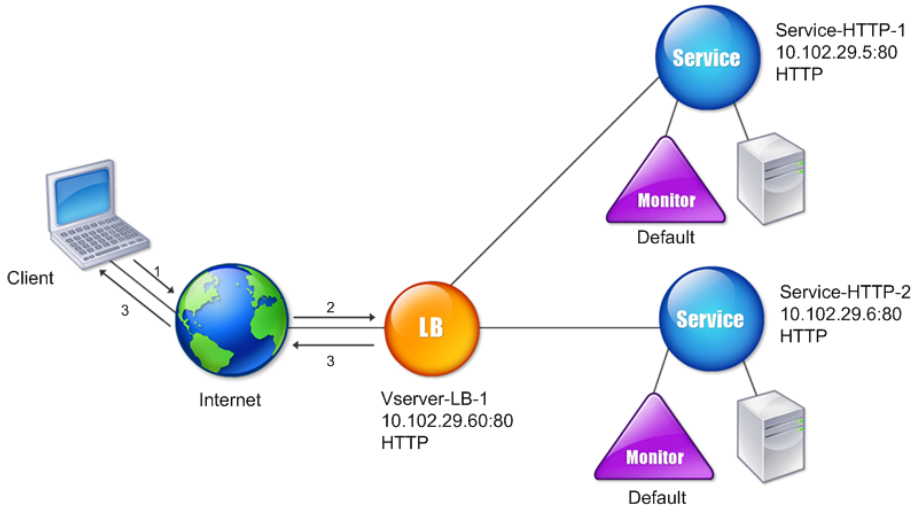


仮想サーバーは、クライアントからの要求に対してサービスを選択して割り当てます。サービス「service-HTTP-1」と「service-HTTP-2」が作成されて、「virtual server-LB-1」という仮想サーバーにバインドされている、前の図のシナリオについて考えてみましょう。virtual server-LB-1 は、クライアント要求を service-HTTP-1 または service-HTTP-2 に転送します。システムは Least Connections 負荷分散方式を使用して、各要求のサービスを選択します。次の表は、システムで設定する必要がある基本的なエンティティの名前と値を示しています。

表 1. LB 構成パラメーター値

次の図は、前の表で説明した負荷分散のサンプル値と、必須パラメーターを示しています。

図 2：負荷分散エンティティモデル



次の表に、コマンドラインインターフェイスを使用してこの負荷分散セットアップを構成するためのコマンドを示します。

タスク	コマンド
負荷分散を有効にする	enable feature lb
「service-HTTP-1」というサービスを作成する	add service service-HTTP-1 10.102.29.5 HTTP 80
「service-HTTP-2」というサービスを作成する	add service service-HTTP-2 10.102.29.6 HTTP 80
「vserver-LB-1」という仮想サーバーを作成する	add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
「service-HTTP-1」というサービスを「vserver-LB-1」という仮想サーバーにバインドする	bind lb vserver vserver-LB-1 service-HTTP-1
「service-HTTP-2」というサービスを「vserver-LB-1」という仮想サーバーにバインドする	bind lb vserver vserver-LB-1 service-HTTP-2

表 2. 初期構成タスク

初期構成タスクの詳細については、「基本的な負荷分散の設定」を参照してください。

タスク	コマンド
「vserver-LB-1」という仮想サーバーのプロパティを表示する	show lb vserver vserver-LB-1
「vserver-LB-1」という仮想サーバーの統計情報を表示する	stat lb vserver vserver-LB-1
「service-HTTP-1」というサービスのプロパティを表示する	show service service-HTTP-1
「service-HTTP-1」というサービスの統計情報を表示する	stat service service-HTTP-1
「service-HTTP-1」というサービスのバインド情報を表示する	show service bindings service-HTTP-1

表 3. 検証タスク

タスク	コマンド
「vserver-LB-1」という仮想サーバーにパーシステンスを構成する	set lb vserver vserver-LB-1 -persistenceType SOURCEIP -persistenceMask 255.255.255.255 -timeout 2
「vserver-LB-1」という仮想サーバーに COOKIEINSERT パーシステンスを構成する	set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
「vserver-LB-1」という仮想サーバーに URLPassive パーシステンスを構成する	set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
クライアント要求を「vserver-LB-1」という仮想サーバー上の URL へリダイレクトするように仮想サーバーを構成する	set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
「vserver-LB-1」という仮想サーバーにバックアップ仮想サーバーを設定する	set lb vserver vserver-LB-1 -backupVserver vserver-LB-2

表 4. カスタマイズタスク

パーシステンスの構成の詳細については、永続性設定の選択と構成を参照してください。クライアント要求を URL にリダイレクトするように仮想サーバーを構成し、バックアップ仮想サーバーを設定する方法については、「負荷分散構成を保護するための機能の構成」を参照してください。

ユースケース：**NetScaler** アプライアンスを使用して **Web** サイトに対してセキュアおよび **HTTPOnly Cookie** オプションを強制する方法

December 8, 2023

Web 管理者は、Secure または HttpOnly、またはセッション ID のフラグと Web アプリケーションによって生成される認証 Cookie の両方を強制する場合があります。HTTP 負荷分散仮想サーバーを使用し、NetScaler アプライアンスでポリシーを書き換えることで、これらの 2 つのオプションを含めるように Set-Cookie ヘッダーを変更できます。

- **HttpOnly** - Cookie のこのオプションにより、Web ブラウザーは HTTP または HTTPS プロトコルのみを使用して Cookie を返します。JavaScript ドキュメントクッキー参照などの HTTP 以外のメソッドは、クッキーにアクセスできません。このオプションは、クロスサイトスクリプティングによる Cookie の盗難を防ぐのに役立ちます。

  > 注
  >
  > JavaScript やクライアント側の Java アプレットなどのクライアント側のスクリプトを使用して Web アプリケーションが Cookie のコンテンツにアクセスする必要がある場合は、HttpOnly オプションを使用できません。このドキュメントに記載されている方法を使用して、NetScaler アプライアンスによって生成された Cookie ではなく、サーバーで生成された Cookie のみを書き換えることができます。たとえば、AppFirewall、永続性、VPN セッションの Cookie などです。

- **Secure** - Cookie のこのオプションにより、送信が SSL で暗号化されている場合、Web ブラウザーは Cookie の値のみを返します。このオプションは、接続の盗聴によるクッキーの盗難を防ぐために使用できます。

> 注
>
> 次の手順は、VPN 仮想サーバーには適用されません。

CLI を使用して既存の **HTTP** 仮想サーバーに対してセキュアフラグと **HTTPOnly** フラグを強制するように **NetScaler** アプライアンスを構成するには

1. 書き換えアクションを作成します。

   この例では、Secure フラグと HttpOnly フラグの両方を設定するように設定されています。どちらかが欠落している場合は、必要に応じて他の組み合わせに合わせて変更します。

```
1 add rewrite action act_cookie_Secure replace_all http.RES.
     full_Header ""Secure; HttpOnly; path=/"" -search "regex(re!(
     path=/\\; Secure; HttpOnly)|(path=/\\; Secure)|(path=/\\;
     HttpOnly)|(path=/)!)"
2 <!--NeedCopy-->
```


このポリシーは、「path=/」、「path=/; Secure」、「path=/; Secure; httpOnly」および「path=/; HttpOnly」のすべてのインスタンスを「セキュア; httpOnly; path=/」に置き換えます。大文字と小文字が一致しない場合、この正規表現 (regex) は失敗します。

2. アクションをトリガーする書き換えポリシーを作成します。

```
add rewrite policy rw_force_secure_cookie "http.RES.HEADER("Set-Cookie").EXISTS" act_cookie_Secure
<!--NeedCopy-->
```

3. 書き換えポリシーをセキュリティで保護する仮想サーバーにバインドします。Secureオプションを使用する場合は、SSL 仮想サーバーを使用する必要があります。

```
bind lb vserver mySSLVServer -policyName rw_force_secure_cookie -priority 100 -gotoPriorityExpression NEXT -type RESPONSE
<!--NeedCopy-->
```

例:

次の例は、HttpOnly フラグを設定する前にクッキーを示しています。

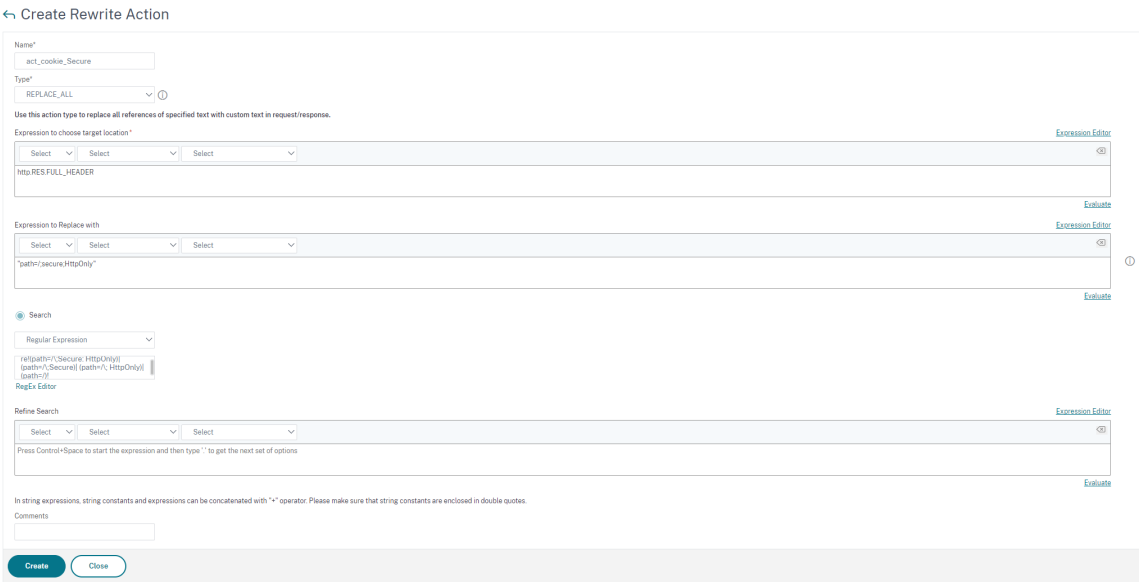
```
Set-Cookie: CtxsAuthId=C5614491; path=/Citrix/ProdWeb
<!--NeedCopy-->
```

以下は、httpOnly フラグを設定した後の Cookie の例です

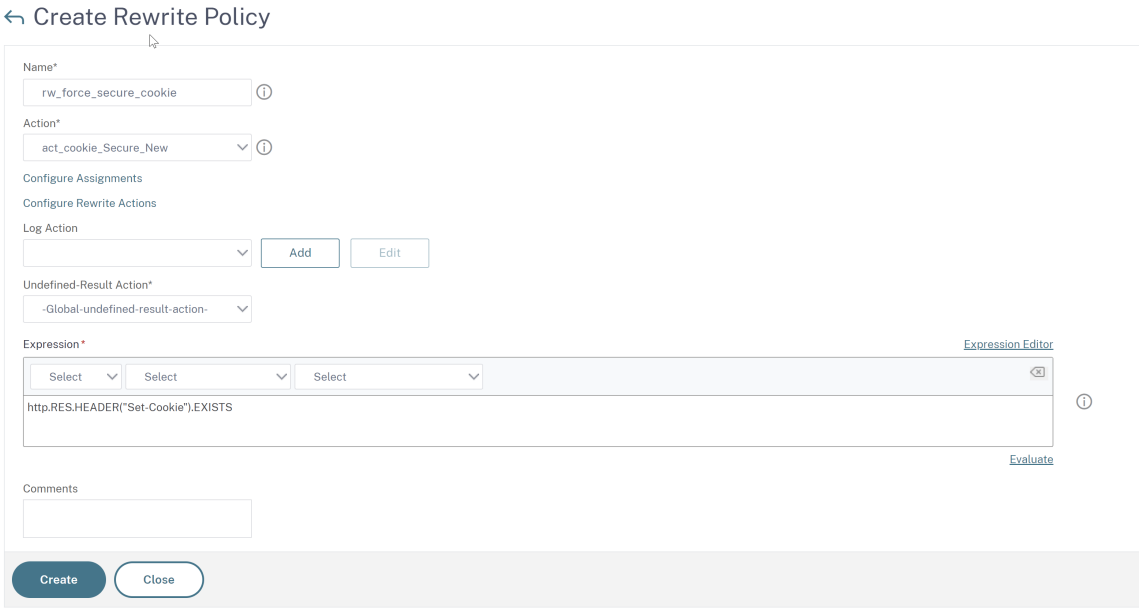
```
Set-Cookie: CtxsAuthId=C5614491; Secure; HttpOnly; path=/Citrix/ProdWeb/
<!--NeedCopy-->
```

GUI を使用して既存の **HTTP** 仮想サーバーに対してセキュアフラグと **HTTPOnly** フラグを強制するように **NetScaler** アプライアンスを構成するには

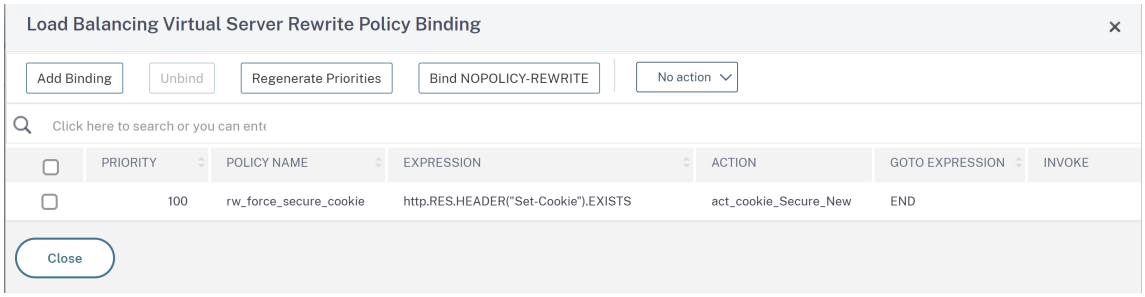
1. [**AppExpert**] > [**Rewrite**] > [**Actions**] の順に移動し、[**Add**] をクリックして新しい書き換えアクションを追加します。



2. **AppExpert >** 書き換え **>** ポリシーに移動し、[追加] をクリックして新しい書き換えポリシーを追加します。



3. [トラフィック管理] **> [**負荷分散**] > [**仮想サーバー] に移動し、書き換え (応答) ポリシーを対応する SSL 仮想サーバーにバインドします。



圧縮による負荷分散トラフィックの速度向上

August 15, 2023

圧縮は、帯域幅の使用を最適化する一般的な手段であり、ほとんどの Web ブラウザーで圧縮データがサポートされています。圧縮機能を有効にすると、NetScaler アプライアンスがクライアントからの要求をインターセプトして、そのクライアントが圧縮コンテンツに対応しているかどうかを判断します。また、アプライアンスがサーバーからの HTTP 応答を受信すると、そのコンテンツを調べて圧縮可能かどうかを決定します。コンテンツが圧縮可能な場合、アプライアンスはコンテンツを圧縮し、応答のヘッダーを変更して実行した圧縮の種類を示し、圧縮コンテンツをクライアントに転送します。

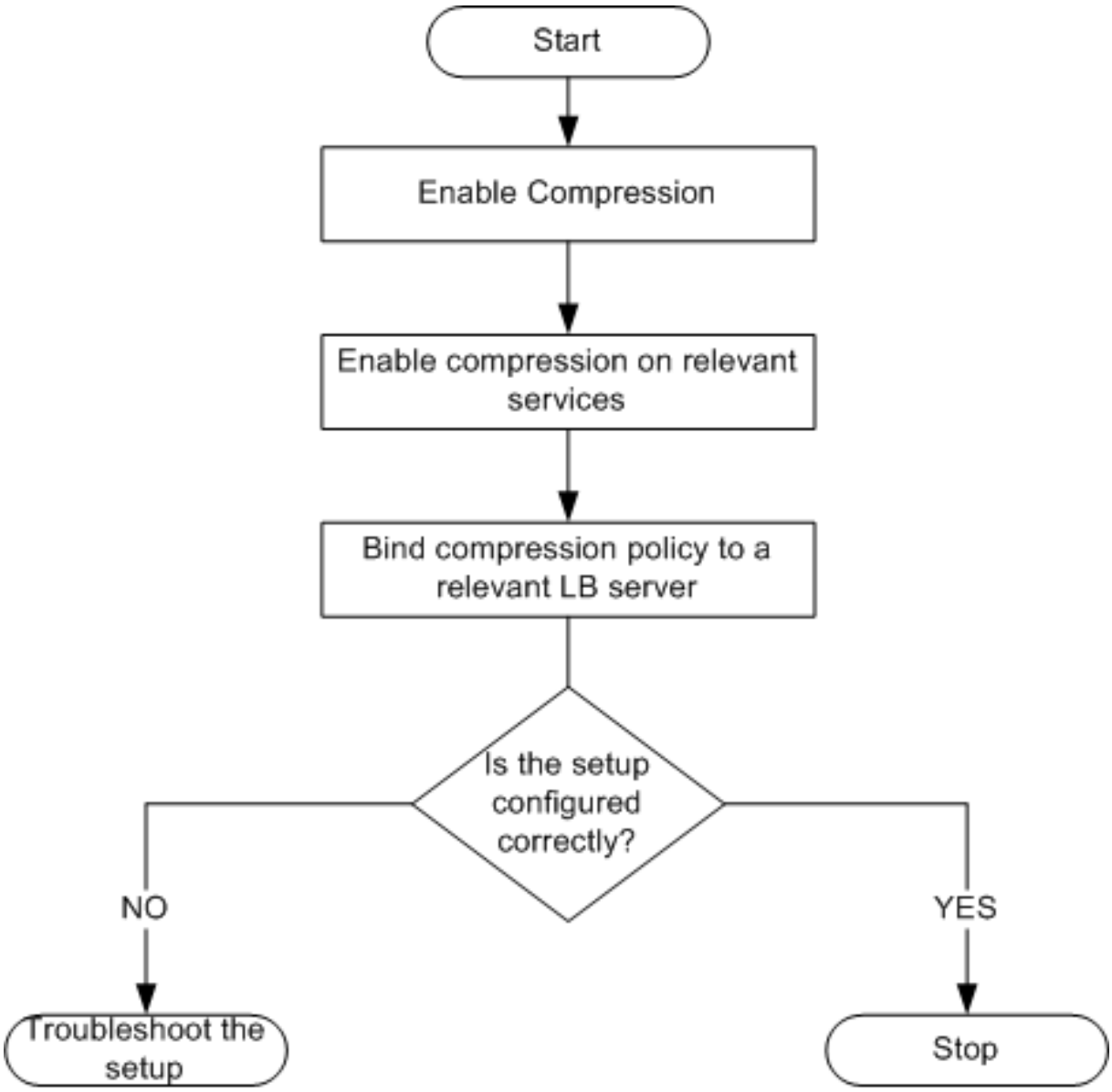
NetScaler の圧縮は、ポリシーベースの機能です。ポリシーは要求と応答をフィルタリングして圧縮される応答を特定し、各応答に適用する圧縮の種類を指定します。アプライアンスは、text/html、text/plain、text/xml、text/css、text/rtf、application/msword、application/vnd.ms-excel、application/vnd.ms-powerpoint などの一般的な MIME タイプを圧縮する複数の組み込みポリシーを提供します。また、カスタムポリシーを作成することもできます。アプライアンスは、application/octet-stream、binary、bytes などの圧縮済みの MIME タイプや、GIF、JPEG などの画像形式を圧縮しません。

圧縮を構成するには、グローバルな圧縮機能を有効にしてから、圧縮対象の応答を配信するサービスごとに圧縮を有効にする必要があります。負荷分散またはコンテンツスイッチ向けに仮想サーバーを構成済みの場合は、それらの仮想サーバーにポリシーをバインドする必要があります。それ以外の場合、アプライアンスを経由するすべてのトラフィックにポリシーが適用されます。

圧縮を構成するタスクの順序

次のフローチャートは、負荷分散セットアップで基本的な圧縮を構成するタスクの順序を示しています。

図 1：圧縮を構成するためのタスクの順序



> 注：上図の手順では、負荷分散が構成済みであることが想定されています。

圧縮を有効にする

デフォルトでは、圧縮が無効になっています。クライアントに送信される HTTP 応答の圧縮を許可するには、圧縮機能を有効にする必要があります。

CLI を使用して圧縮を有効にするには

コマンドプロンプトで次のコマンドを入力し、圧縮を有効化して構成を確認します。

- enable ns feature CMP

- show ns feature

```
    > enable ns feature CMP



    Done

    > show ns feature



          Feature                        Acronym              Status

          -------                        -------              ------

    1)    Web Logging                    WL                   ON

    2)    Surge Protection               SP                   OFF

    .

    7) Compression Control CMP ON

       .

    Done
<!--NeedCopy-->
```

GUI を使用して圧縮を有効にするには

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. 詳細ペインの [Modes and Features] で、[Change basic features] をクリックします。
3. [Configure Basic Features] ダイアログボックスで、[Compression] チェックボックスをオンにしてから [OK] をクリックします。
4. [Enable/Disable Feature(s)?] ダイアログボックスで、[Yes] をクリックします。

データを圧縮するサービスの設定

グローバルな圧縮設定を有効にしたら、圧縮対象のファイルを配信するサービスごとに圧縮を有効にする必要があります。

CLI を使用して特定のサービスの圧縮を有効にするには

コマンドプロンプトで次のコマンドを入力し、特定のサービスの圧縮を有効化して構成を確認します。

- set service \< サービス名\> -CMP YES
- show service \< 名前\>

```
> show service SVC_HTTP1

SVC_HTTP1 (10.102.29.18:80) - HTTP

State: UP

Last state change was at Tue Jun 16 06:19:14 2009 (+737 ms)

Time since last state change: 0 days, 03:03:37.200

Server Name: 10.102.29.18

Server ID : 0   Monitor Threshold : 0

Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits

Use Source IP: NO

Client Keepalive(CKA): NO

Access Down Service: NO

TCP Buffering(TCPB): NO

HTTP Compression(CMP): YES

```



```
39
40 Idle timeout: Client: 180 sec   Server: 360 sec
41
42
43 Client IP: DISABLED
44
45
46 Cacheable: NO
47
48
49 SC: OFF
50
51
52 SP: OFF
53
54
55 Down state flush: ENABLED
56
57 1)      Monitor Name: tcp-default
58
59
60 State: DOWN     Weight: 1
61
62
63 Probes: 1095    Failed [Total: 1095 Current: 1095]
64
65
66 Last response: Failure - TCP syn sent, reset received.
67
68
69 Response Time: N/A
70
71
72 Done
73
74 <!--NeedCopy-->
```

GUI を使用して特定のサービスの圧縮を有効にするには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 詳細ペインで圧縮を設定するサービス（たとえば、[service-HTTP-1]）を選択し、[Open] をクリックします。
3. [Advanced] タブの [Settings] で、[Compression] チェックボックスをオンにして [OK] をクリックします。
4. サービスが選択されている場合はペインの下部にある [**Details**] に [HTTP Compression(CMP): ON] が表示されていることを確認します。

仮想サーバーへの圧縮ポリシーのバインド

仮想サーバーに圧縮ポリシーをバインドすると、ポリシーは、その仮想サーバーに関連付けられたサービスによってのみ評価されます。仮想サーバーへの圧縮ポリシーのバインドは、[Configure Virtual Server (Load Balancing)]ダイアログボックスまたは［Compression Policy Manager］ダイアログボックスを使用して行います。このトピックには、[Configure Virtual Server（Load Balancing)］ダイアログボックスを使用して、圧縮ポリシーを負荷分散仮想サーバーにバインドする手順が含まれています。

コマンドラインを使用して仮想サーバーへの圧縮ポリシーをバインドまたはバインド解除するには

コマンドプロンプトで次のコマンドを入力し、負荷分散仮想サーバーへの圧縮ポリシーをバインドまたはバインド解除して構成を確認します。

(バインド	<name\ > バインド解除) lb vserver-ポリシー名\ <string\ >

-
- show lb vserver \< 名前\>

例:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp
Done
> showlbvserverlbvip

lbvip(8.7.6.6:80)-HTTPType:ADDRESS
State:UP
LaststatechangewasatThuMay2805:37:212009(+685ms)
Timesincelaststatechange:19days,04:26:50.470
EffectiveState:UP
ClientIdleTimeout:180sec
Downstateflush:ENABLED
DisablePrimaryVserverOnDown:DISABLED
PortRewrite:DISABLED
No.ofBoundServices:1(Total)1(Active)
ConfiguredMethod:LEASTCONNECTION
CurrentMethod:RoundRobin,Reason:Boundservice'sstatechangedtoUP
Mode:IP
Persistence:NONE
VserverIPandPortinsertion:OFF
Push:DISABLEDPushVServer:
PushMultiClients:NO
PushLabelRule:

BoundServiceGroups:
1)GroupName:Service-Group-1
```



```
26
27 1)Service-Group-1(10.102.29.252:80)-HTTPState:UPWeight:1
28
29 1)Policy:ns_cmp_msappPriority:0
30
31 Done
32
33 <!--NeedCopy-->
```

GUI を使用して負荷分散仮想サーバーへの圧縮ポリシーをバインドまたはバインド解除するには

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ペインで、圧縮ポリシーのバインドまたはバインド解除を行う仮想サーバー（たとえば、[Vserver-LB-1]）を選択し、［Open］をクリックします。
3. ［Configure Virtual Server (Load Balancing)］ダイアログボックスで、[Policies] タブの［Compression］をクリックします。
4. 次のいずれかを行います：
   - 圧縮ポリシーをバインドする場合は、［Insert Policy］をクリックしてから仮想サーバーにバインドするポリシーを選択します。
   - 圧縮ポリシーをバインド解除する場合は、仮想サーバーからバインド解除するポリシーの名前を選択し、［Unbind Policy］をクリックします。
5. 「OK」をクリックします。

SSL による負荷分散トラフィックのセキュリティ保護

August 15, 2023

NetScaler SSL オフロード機能は、SSL トランザクションを実行する Web サイトのパフォーマンスを透過的に向上させます。SSL オフロードでは、CPU 負荷の高い SSL 暗号化および復号化タスクをローカル Web サーバーからアプライアンスにオフロードすることにより、SSL データの処理によるサーバーパフォーマンスの低下を引き起こすことなく、Web アプリケーションを安全に配信できます。SSL トラフィックを復号化すると、あらゆる標準サービスで処理できるようになります。SSL プロトコルは、さまざまな種類の HTTP および TCP データとシームレスに機能して、このようなデータを使用するトランザクションに、セキュリティ保護されたチャネルを提供します。

SSL を設定するには、まず SSL を有効にする必要があります。次に、アプライアンスで HTTP または TCP サービスおよび SSL 仮想サーバーを構成し、そのサービスを仮想サーバーにバインドします。証明書とキーのペアを追加して SSL 仮想サーバーにバインドする必要もあります。Outlook Web Access サーバーを使用する場合は、SSL サポートを有効にするアクションとそのアクションに適用するポリシーを作成する必要があります。SSL 仮想サーバーは、暗号化された着信トラフィックをインターセプトして、ネゴシエートしたアルゴリズムを使用してそのトラフィ

ックを復号化します。復号化されたデータは、アプライアンス上のほかのエンティティに転送され、適切に処理されます。

SSL オフロードの詳細については、SSL オフロードとアクセラレーションを参照してください。

SSL を設定するタスクの順序

SSL を設定するには、まず SSL を有効にする必要があります。その後、NetScaler アプライアンスで SSL 仮想サーバーと HTTP または TCP サーバーを作成する必要があります。最後に、有効な SSL 証明書と設定済みのサービスを、SSL 仮想サーバーにバインドする必要があります。

SSL 仮想サーバーは、暗号化された着信トラフィックを傍受して、ネゴシエートしたアルゴリズムを使用してそのトラフィックを解読します。復号化されたデータは、NetScaler アプライアンス上のほかのエンティティに転送され、適切に処理されます。

次のフローチャートには、基本的な SSL オフロードセットアップを設定するタスクの順序が示されています。

図 1：SSL オフロードを構成するための一連のタスク

SSL オフロードを有効にする

まず、SSL 機能を有効にします。SSL 機能を有効にしなくてもアプライアンス上で SSL ベースのエンティティを設定できますが、それらのエンティティは SSL を有効にするまで動作しません。

CLI を使用して SSL を有効にする

コマンドプロンプトで次のコマンドを入力し、SSL オフロードを有効にして構成を確認します。

```
-  enable ns feature SSL
-  show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature ssl

Done


> show ns feature


Feature Acronym Status


------- ------- ------


1) Web Logging WL ON


2) SurgeProtection SP OFF


3) Load Balancing LB ON . . .


 9) SSL Offloading SSL ON


10) Global Server Load Balancing GSLB ON . .


Done >
<!--NeedCopy-->
```


GUI を使用して **SSL** を有効にする

次の手順を実行します:

1. ナビゲーションペインで、[システム] を展開し、[設定] をクリックします。
2. 詳細ペインの [モードと機能] で、[基本機能の変更] をクリックします。
3.「**SSL** オフロード」チェックボックスを選択し、「**OK**」をクリックします。
4.「機能の有効化/無効化」では? メッセージボックスで、「はい」をクリックします。

HTTP サービスを作成する

アプライアンス上の各サービスは、サーバー上の個々のアプリケーションとして機能します。構成したサービスは、アプライアンスがネットワーク上のサーバーにアクセスしてその状態を監視できるようになるまで無効状態になります。このトピックでは、HTTP サービスを作成する手順について説明します。

注: TCP トラフィックの場合、次の手順を実行しますが、HTTP サービスの代わりに TCP サービスを作成します。

CLI を使用して **HTTP** サービスを追加します

コマンドプロンプトで次のコマンドを入力し、HTTP サービスを追加して構成を確認します。

```
-  add service <name> (<IP> | <serverName>) <serviceType> <port>
-  show service <name>
<!--NeedCopy-->
```

例:

```
> add service SVC_HTTP1 10.102.29.18 HTTP 80


 Done


> show service SVC_HTTP1


        SVC_HTTP1 (10.102.29.18:80) - HTTP


        State: UP


        Last state change was at Wed Jul 15 06:13:05 2009


        Time since last state change: 0 days, 00:00:15.350


```



```
        Server Name: 10.102.29.18

        Server ID : 0   Monitor Threshold : 0

        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits

        Use Source IP: NO

        Client Keepalive(CKA): NO

        Access Down Service: NO

        TCP Buffering(TCPB): NO

        HTTP Compression(CMP): YES

        Idle timeout: Client: 180 sec   Server: 360 sec

        Client IP: DISABLED

        Cacheable: NO

        SC: OFF

        SP: OFF

        Down state flush: ENABLED



1)      Monitor Name: tcp-default

                State: UP       Weight: 1

                Probes: 4       Failed [Total: 0 Current: 0]

```



```
75
76                  Last response: Success - TCP syn+ack received.
77
78
79                  Response Time: N/A
80
81
82   Done
83  <!--NeedCopy-->
```

GUI を使用して **HTTP** サービスを追加します

次の手順を実行します:

1. [**Traffic Management**] > [**SSL Offload**] > [**Servers**] の順に選択します。
2. 詳細ペインで、[追加] をクリックします。
3. [**Create Service**] ダイアログボックスで、サービスの名前、IP アドレス、およびポートを入力します（たとえば、SVC_HTTP1, 10.102.29.18、および 80)。
4. プロトコルリストで、サービスのタイプ (HTTP など) を選択します。
5. [作成] をクリックし、[閉じる] をクリックします。構成した HTTP サービスが、[Services] ページに表示されます。
6. 作成したサービスを選択して、ペインの下部にある [Details] セクションを表示し、パラメーターが正しく構成されていることを確認します。

SSL ベースの仮想サーバーを追加する

基本的な SSL オフロードセットアップでは、SSL 仮想サーバーは暗号化されたトラフィックをインターセプトおよび復号化して、仮想サーバーにバインドされているサービスにクリアテキストメッセージを送信します。CPU 負荷の高い SSL 処理をアプライアンス側にオフロードすると、バックエンドサーバーでより多くの要求を処理できるようになります。

CLI を使用して **SSL** ベースの仮想サーバーを追加します

コマンドプロンプトで次のコマンドを入力し、SSL ベースの仮想サーバーを追加して構成を確認します。

```
1  -  add lb vserver <name> <serviceType> [<IPAddress> <port>]
2  -  show lb vserver <name>
3  <!--NeedCopy-->
```

> 注意**:** 安全な接続を確保するには、有効な SSL 証明書を SSL ベースの仮想サーバーを有効にする前に、有効な SSL 証明書をバインドする必要があります。

例**:**


```
> add lb vserver vserver-SSL-1 SSL 10.102.29.50 443
  Done


  > show lb vserver vserver-SSL-1


  vserver-SSL-1 (10.102.29.50:443) - SSL Type: ADDRESS


  State: DOWN[Certkey not bound] Last state change was at Tue Jun 16
     06:33:08 2009 (+176 ms)


  Time since last state change: 0 days, 00:03:44.120


  Effective State: DOWN Client Idle Timeout: 180 sec


  Down state flush: ENABLED


  Disable Primary Vserver On Down : DISABLED


  No. of Bound Services : 0 (Total) 0 (Active)


  Configured Method: LEASTCONNECTION Mode: IP


  Persistence: NONE


  Vserver IP and Port insertion: OFF


  Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule:
     Done
<!--NeedCopy-->
```

GUI を使用して **SSL** ベースの仮想サーバーを追加します

次の手順を実行します:

1. [**Traffic Management**] > [**SSL Offload**] > [**Virtual Servers**] の順に選択します。
2. 詳細ペインで、[追加] をクリックします。
3. [**Create Virtual Server (SSL Offload)**] ダイアログボックスで、仮想サーバーの名前、IP アドレス、およびポートを入力します。

4. プロトコルリストで、仮想サーバーのタイプ (SSL など) を選択します。
5. [作成] をクリックし、[閉じる] をクリックします。
6. 作成した仮想サーバーを選択して、ペインの下部にある［Details］セクションを表示し、パラメーターが正しく構成されていることを確認します。証明書とキーのペアとサービスが仮想サーバーにバインドされていないため、仮想サーバーは DOWN としてマークされます。

> 注意**:** 安全な接続を確保するには、有効な SSL 証明書を SSL ベースの仮想サーバーを有効にする前に、有効な SSL 証明書をバインドする必要があります。

サービスの **SSL** 仮想サーバーへのバインド

SSL 仮想サーバーが復号化した受信データは、その仮想サーバーにバインドされたサービスに転送されます。

アプライアンスとサーバーの間のデータ転送は、暗号化したりクリアテキストで送信したりできます。アプライアンスとサーバーの間のデータ転送を暗号化する場合、トランザクション全体がエンドツーエンドで保護されることになります。エンドツーエンドのセキュリティのためのシステムの構成の詳細については、「SSL オフロードとアクセラレーション」を参照してください。

CLI を使用してサービスを仮想サーバーにバインドします

コマンドプロンプトで次のコマンドを入力し、サービスを SSL 仮想サーバーにバインドして構成を確認します。

```
-  bind lb vserver <name> <serviceName>
-  show lb vserver <name>
<!--NeedCopy-->
```

例:

```
> bind lb vserver vserver-SSL-1 SVC_HTTP1



  Done


  > show lb vserver vserver-SSL-1 vserver-SSL-1 (10.102.29.50:443) -
     SSL Type:


  ADDRESS State: DOWN[Certkey not bound]


  Last state change was at Tue Jun 16 06:33:08 2009 (+174 ms)


  Time since last state change: 0 days, 00:31:53.70
```



```
19
20
21    Effective State: DOWN Client Idle
22
23
24    Timeout: 180 sec
25
26
27    Down state flush: ENABLED Disable Primary Vserver On Down :
28
29
30    DISABLED No. of Bound Services : 1 (Total) 0 (Active)
31
32
33    Configured Method: LEASTCONNECTION Mode: IP Persistence: NONE Vserver
          IP and
34
35
36    Port insertion: OFF Push: DISABLED Push VServer: Push Multi Clients:
         NO Push Label Rule:
37
38
39
40
41
42    1) SVC_HTTP1 (10.102.29.18: 80) - HTTP
43
44
45    State: DOWN Weight: 1
46
47
48    Done
49  <!--NeedCopy-->
```

GUI を使用してサービスを仮想サーバーにバインドする

1. [**Traffic Management**] > [**SSL Offload**] > [**Virtual Servers**] の順に選択します。
2. 詳細ペインで仮想サーバーを選択して、[**Open**] をクリックします。
3. [サービス] タブの [アクティブ] 列で、選択した仮想サーバーにバインドするサービスの横にあるチェックボックスを選択します。
4. [**OK**] をクリックします。
5. ペインの下部にある [Details] セクションの [Number of Bound Services] カウンターが、仮想サーバーにバインドしたサービスの数だけ増加することを確認します。

証明書とキーのペアを追加します

SSL 証明書は、SSL キー交換および暗号化/復号化プロセスに含まれるエレメントです。証明書は、SSL サーバーのアイデンティティを確立するために SSL ハンドシェイク中に使用されます。NetScaler アプライアンスに含まれている、有効な既存の SSL 証明書を使用するか、独自の SSL 証明書を作成できます。アプライアンスは、最大 4096 ビットの RSA 証明書をサポートします。

次の曲線のみを持つ ECDSA 証明書がサポートされています。

- prime256v1 (ADC では P_256)
- secp384r1 (ADC では P_384)
- secp521r1 (ADC では P_521、VPX でのみサポート)
- secp224r1 (ADC では P_224、VPX でのみサポート)

> 注: 信頼される証明機関から発行された有効な SSL 証明書を使用することをお勧めします。無効な証明書や自分で作成した証明書は、一部の SSL クライアントと互換性がありません。

SSL 処理に使用する前に、証明書を対応するキーとペアにする必要があります。次に、証明書とキーのペアを仮想サーバーにバインドすると、SSL 処理に使用できるようになります。

CLI を使用して証明書キーペアを追加します

> 注**:** ECDSA 証明書とキーのペアの作成については、「ECDSA 証明書とキーペアの作成」を参照してください。

コマンドプロンプトで次のコマンドを入力し、証明書とキーのペアを作成して構成を確認します。

```
-  add ssl certKey <certkeyName> -cert <string> [-key <string>]
-  show sslcertkey <name>
<!--NeedCopy-->
```

例**:**

```
> add ssl certKey CertKey-SSL-1 -cert ns-root.cert -key ns-root.key

 Done


> show sslcertkey CertKey-SSL-1


    Name: CertKey-SSL-1 Status: Valid,


    Days to expiration:4811 Version: 3


    Serial Number: 00 Signature Algorithm: md5WithRSAEncryption Issuer:
       C=US,ST=California,L=San
```



```
16
17
18     Jose,O=Citrix ANG,OU=NS Internal,CN=de fault
19
20
21     Validity Not Before: Oct 6 06:52:07 2006 GMT Not After : Aug 17
           21:26:47 2022 GMT
22
23
24     Subject: C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,
           CN=d efault Public Key
25
26
27     Algorithm: rsaEncryption Public Key
28
29
30     size: 1024
31
32
33   Done
34  <!--NeedCopy-->
```

GUI を使用して証明書キーペアを追加します

次の手順を実行します:

1. [**Traffic Management**] **>** [**SSL**] **>** [**Certificates**] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [証明書のインストール**]** ダイアログボックスの **[**証明書とキーのペア名] テキストボックスに、追加する証明書キーペアの名前 (CertKey-SSL-1 など) を入力します。
4. 「詳細」の「証明書ファイル名」で、「ブラウズ（アプライアンス）」をクリックして証明書を探します。証明書とキーはともに、アプライアンスの/nsconfig/ssl/ディレクトリに保存されます。ローカルシステムにある証明書を使用するには、[Local] を選択します。
5. 使用する証明書を選択し、[選択] をクリックします。
6. 「プライベート・キー・ファイル名」で、「ブラウズ（アプライアンス）」をクリックしてプライベート・キー・ファイルを探します。ローカルシステムにある秘密キーを使用するには、[Local] を選択します。
7. 使用するキーを選択し、[選択] をクリックします。証明書とキーのペアで使用するキーを暗号化するには、暗号化に使用するパスワードを [Password] ボックスに入力します。
8. [**Install**] をクリックします。
9. 証明書キーのペアをダブルクリックし、[証明書の詳細] ウィンドウで、パラメーターが正しく構成されて保存されていることを確認します。

SSL 証明書キーペアの仮想サーバーへのバインド

SSL 証明書とそれに対応するキーをペアにしたら、証明書とキーのペアを SSL 仮想サーバーにバインドして、SSL 処理に使用できるようにする必要があります。セキュリティで保護されたセッションでは、クライアントコンピューターとアプライアンス上の SSL ベースの仮想サーバーの間に接続を確立する必要があります。その後、仮想サーバーで着信トラフィックに対して SSL 処理が実行されます。したがって、アプライアンスで SSL 仮想サーバーを有効にする前に、有効な SSL 証明書を SSL 仮想サーバーにバインドする必要があります。

CLI を使用して、SSL 証明書キーペアを仮想サーバーにバインドします

コマンドプロンプトで次のコマンドを入力し、SSL 証明書とキーのペアを仮想サーバーにバインドして構成を確認します。

```
-  bind ssl vserver <vServerName> -certkeyName <string>
-  show ssl vserver <name>
<!--NeedCopy-->
```

例:

```
> bind ssl vserver Vserver-SSL-1 -certkeyName CertKey-SSL-1

Done

> show ssl vserver Vserver-SSL-1



     Advanced SSL configuration for VServer Vserver-SSL-1:

     DH: DISABLED

     Ephemeral RSA: ENABLED Refresh Count: 0

     Session Reuse: ENABLED Timeout: 120 seconds

     Cipher Redirect: ENABLED

     SSLv2 Redirect: ENABLED

     ClearText Port: 0

```



```
32
33      Client Auth: DISABLED
34
35
36      SSL Redirect: DISABLED
37
38
39      Non FIPS Ciphers: DISABLED
40
41
42      SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
43
44
45
46
47
48  1) CertKey Name: CertKey-SSL-1 Server Certificate
49
50
51  1) Cipher Name: DEFAULT
52
53
54     Description: Predefined Cipher Alias
55
56
57  Done
58  <!--NeedCopy-->
```

GUI を使用して、**SSL** 証明書キーペアを仮想サーバーにバインドします

次の手順を実行します:

1. [**Traffic Management**] > [**SSL Offload**] > [**Virtual Servers**] の順に選択します。
2. 証明書とキーのペアをバインドする仮想サーバー（たとえば、[Vserver-SSL-1]）を選択して、[Open] をクリックします。
3. [**Configure Virtual Server (SSL Offload)**] ダイアログボックスの [**SSL Settings**] タブにある [**Available**] で、仮想サーバーにバインドする証明書とキーのペアを選択します。次に、[追加] をクリックします。
4. [**OK**] をクリックします。
5. 選択した証明書キーのペアが [Configured] 領域に表示されていることを確認します。

Outlook Web Access に対するサポートの構成

NetScaler アプライアンスで Outlook Web Access（OWA）サーバーを使用している場合は、OWA サーバー宛の HTTP 要求に特別なヘッダーフィールド「FRONT-END-HTTPS: ON」を挿入するようにアプライアンスを構成して、URL リンクが「`http://`」ではなく「`https://`」として生成されるようにします。

注**:** OWA サポートは HTTP ベースの SSL 仮想サーバーとサービスに対してのみ有効にできます。TCP ベースの SSL 仮想サーバーと SSL サービスでは OWA をサポートできません。

OWA サポートを構成するには、次の操作を実行します。

- OWA サポートを有効にする SSL アクションを作成します。
- SSL ポリシーを作成します。
- ポリシーを SSL 仮想サーバーにバインドします。

OWA サポートを有効にする **SSL** アクションを作成します

Outlook Web Access（OWA）サポートを有効にする前に、SSL アクションを作成する必要があります。SSL アクションは SSL ポリシーにバインドされ、着信データがポリシーで指定された規則と一致すると実行されます。

CLI を使用して **OWA** サポートを有効にする **SSL** アクションを作成します

コマンドプロンプトで次のコマンドを入力し、OWA サポートを有効にする SSL アクションを作成して構成を確認します。

```
-  add ssl action <name> -OWASupport ENABLED
-  show SSL action <name>
<!--NeedCopy-->
```

例**:**

```
     > add ssl action Action-SSL-OWA -OWASupport enabled



     Done

     > show SSL action Action-SSL-OWA

     Name: Action-SSL-OWA

     Data Insertion Action: OWA

     Support: ENABLED

     Done
<!--NeedCopy-->
```


GUI を使用して **OWA** サポートを有効にする **SSL** アクションを作成します

次の手順を実行します:

1. [**Traffic Management**] > [**SSL**] > [**Policies**] に移動します。
2. 詳細ペインの [アクション] タブで、[追加] をクリックします。
3. 「**SSL** アクションの作成」ダイアログの「名前」テキストボックスに「action-SSL-OWA」と入力します。
4. Outlook Web Access で [有効] を選択します。
5. [作成] をクリックし、[閉じる] をクリックします。
6. [Action-SSL-OWA] が [**SSL Actions**] ページに表示されていることを確認します。

SSL ポリシーを作成する

SSL ポリシーは、ポリシーインフラストラクチャを使用して作成します。各 SSL ポリシーには SSL アクションがバインドされ、アクションは、着信トラフィックがポリシーで設定された規則と一致すると実行されます。

CLI を使用して **SSL** ポリシーを作成します

コマンドプロンプトで次のコマンドを入力し、SSL ポリシーを作成して構成を確認します。

```
-  add ssl policy <name> -rule <expression> -reqAction <string>
-  show ssl policy <name>
<!--NeedCopy-->
```

例**:**

```
> add ssl policy-SSL-1 -rule ns_true -reqaction Action-SSL-OWA

Done

> show ssl policy-SSL-1

Name: Policy-SSL-1 Rule: ns_true

Action: Action-SSL-OWA Hits: 0

Policy is bound to following entities

1) PRIORITY : 0

Done
<!--NeedCopy-->
```

GUI を使用して **SSL** ポリシーを作成します

次の手順を実行します:

1. [**Traffic Management**] > [**SSL**] > [**Policies**] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [**SSL** ポリシーの作成] ダイアログボックスの [名前] テキストボックスに、SSL ポリシーの名前 (Policy-SSL-1 など) を入力します。
4. 「リクエストアクション」で、このポリシーに関連付ける設定済みの SSL アクション（Action-SSL-OWA など）を選択します。ns_true 汎用式により、成功した SSL ハンドシェイクトラフィックのすべてにこのポリシーが適用されます。特定の応答に対してのみポリシーを適用する場合は、より高い詳細レベルのポリシーを作成できます。詳細なポリシー式の構成の詳細については、「SSL アクションとポリシー」を参照してください。
5. [**Named Expressions**] で、組み込みの汎用式 ns_true を選択し、[**Add Expression**] をクリックします。式 ns_true が [Expression] ボックスに表示されます。
6. [作成] をクリックし、[閉じる] をクリックします。
7. ポリシーを選択して、ペイン下部にある [Details] セクションを表示し、ポリシーが正しく構成されていることを確認します。

SSL ポリシーを **SSL** 仮想サーバーにバインドします

Outlook Web Access に SSL ポリシーを設定したら、Outlook 着信トラフィックをインターセプトする仮想サーバーにポリシーをバインドします。着信データが SSL ポリシーで構成された規則と一致すると、そのポリシーに関連付けられたアクションが実行されます。

CLI を使用して **SSL** ポリシーを **SSL** 仮想サーバーにバインドします

コマンドプロンプトで次のコマンドを入力し、SSL ポリシーを SSL 仮想サーバーにバインドして構成を確認します。

```
-  bind ssl vserver <vServerName> -policyName <string>
-  show ssl vserver <name>
<!--NeedCopy-->
```

例**:**

```
> bind ssl vserver Vserver-SSL-1 -policyName Policy-SSL-1

 Done

> show ssl vserver Vserver-SSL-1

Advanced SSL configuration for VServer Vserver-SSL-1:

DH: DISABLED

Ephemeral RSA: ENABLED

Refresh Count: 0

```



```
Session Reuse: ENABLED

Timeout: 120 seconds

Cipher Redirect: ENABLED

SSLv2 Redirect: ENABLED

ClearText Port: 0

Client Auth: DISABLED

SSL Redirect: DISABLED

Non FIPS Ciphers: DISABLED

SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1) CertKey Name: CertKey-SSL-1 Server Certificate

1) Policy Name: Policy-SSL-1 Priority: 0

1) Cipher Name: DEFAULT Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

GUI を使用して **SSL** ポリシーを **SSL** 仮想サーバーにバインドします

次の手順を実行します:

1. [**Traffic Management**] > [**SSL Offload**] > [**Virtual Servers**] の順に選択します。
2. 詳細ペインで仮想サーバー **(**たとえば、**VServer-SSL-1)** を選択し、「開く」をクリックします。
3. [仮想サーバーの設定 **(SSL** オフロード**)**] ダイアログボックスで、[ポリシーの挿入] をクリックし、SSL 仮想サーバーにバインドするポリシーを選択します。必要に応じて、[Priority] ボックスをダブルクリックして、新しい優先度を入力することもできます。
4. [**OK**] をクリックします。

一目でわかる機能

August 15, 2023

NetScaler の機能は、特定のニーズに対応するために、個別に構成することも、組み合わせて構成することもできます。一部の機能は複数のカテゴリに当てはまりますが、多数の NetScaler ADC 機能は、一般に、アプリケーション

スイッチングおよびトラフィック管理機能、アプリケーションアクセラレーション機能、アプリケーションセキュリティおよびファイアウォール機能、およびアプリケーション可視性機能として分類できます。

フィーチャが処理を実行する順序を理解するには、「フィーチャの処理の順序」セクションを参照してください。

アプリケーションスイッチングとトラフィック管理機能

August 15, 2023

以下は、アプリケーションスイッチングとトラフィック管理機能です。

SSL オフロード

Web サーバーから SSL 暗号化および解読を透過的にオフロードして、コンテンツ要求の処理用にサーバーのリソースを解放します。SSL はアプリケーションのパフォーマンスにとって大きな負担となり、多くの最適化方法が無効になることがあります。SSL オフロードおよび SSL アクセラレーションでは、Citrix Request Switching 技術のすべての利点を SSL トラフィックに適用できるため、エンドユーザーのパフォーマンスを低下させることなく、Web アプリケーションのセキュリティ保護されたデリバリを実現できます。

詳細については、SSL オフロードとアクセラレーションを参照してください。

アクセス制御リスト

着信パケットとアクセス制御リスト（ACL：Access Control List）を比較します。パケットが ACL 規則と一致した場合は、規則で指定されたアクションがパケットに適用されます。一致しない場合は、デフォルトアクション（ALLOW）が適用され、パケットは通常どおりに処理されます。アプライアンスが着信パケットと ACL を比較されるようにするには、ACL を適用する必要があります。すべての ACL はデフォルトで有効になっていますが、NetScaler アプライアンスが着信パケットを ACL と比較するためには、管理者が ACL を適用する必要があります。ルックアップテーブルに含める必要がなくても保持すべき ACL がある場合は、それを無効にしてから ACL を適用する必要があります。ADC アプライアンスは、着信パケットを無効な ACL と比較しません。

詳細については、アクセス制御リストを参照してください。

負荷分散

負荷分散の決定は、ラウンドロビン、最小接続数、加重最小帯域幅、加重最小パケット数、最小応答時間、および URL、ドメインソース IP、宛先 IP に基づくハッシュなど、さまざまなアルゴリズムに基づいて行われます。TCP と UDP プロトコルの両方がサポートされているので、これらのプロトコルに基づくすべてのトラフィック（たとえば、HTTP、HTTPS、UDP、DNS、FTP、NNTP、および一般的なファイアウォールトラフィック）を負荷分散す

ることができます。また、ADC アプライアンスは、ソース IP、Cookie、サーバー、グループ、または SSL セッションに基づくセッションパーシステンスを維持できます。サーバー、キャッシュ、ファイアウォール、およびそのほかのインフラストラクチャデバイスが正常に動作して適切なコンテンツがユーザーに提供されるように、カスタムの Extended Content Verification（ECV）を適用できます。また、ping、TCP、または HTTP URL を使用してヘルスチェックを実行したり、Perl スクリプトによるモニターを作成したりできます。
高度な WAN 最適化を提供するには、データセンターで展開されている CloudBridge アプライアンスを NetScaler アプライアンスで負荷分散することができます。これにより、帯域幅と同時セッションの数を大幅に改善できます。

詳細については、「負荷分散」を参照してください。

トラフィックドメイン

トラフィックドメインを使用すると、単一の NetScaler アプライアンス内にいくつかの論理 ADC パーティションを作成できます。トラフィックドメインを使用すると、異なるアプリケーション用にネットワークトラフィックを分離できます。トラフィックドメインを使用すると、リソース間のやり取りが行われない分離環境を複数作成できます。特定のトラフィックドメインに属しているアプリケーションは、そのドメイン内のエンティティおよびプロセストラフィックとのみ通信します。あるトラフィックドメインに属しているトラフィックは、別のトラフィックドメインの境界を越えることはできません。そのため、アドレスが同じドメイン内で重複していない限り、アプライアンスで重複する IP アドレスを使用できます。

詳細については、「トラフィックドメイン」を参照してください。

ネットワークアドレス変換

ネットワークアドレス変換（NAT）では、NetScaler アプライアンスを通過する IP パケットの送信元/宛先 IP アドレスや TCP/UDP ポート番号が変更されます。アプライアンスで NAT を有効にすると、プライベートネットワークのセキュリティが強化されます。また、データが NetScaler アプライアンスを通過するときにプライベートネットワークの送信元 IP アドレスが変更されるため、インターネットなどのパブリックネットワークからプライベートネットワークが保護されます。

NetScaler アプライアンスでは、次の種類のネットワークアドレス変換がサポートされます。

INAT：受信 NAT（Inbound NAT：INAT）では、NetScaler アプライアンスで構成された IP アドレス（通常はパブリックアドレス）がサーバーの代わりに接続要求を待機します。アプライアンスがそのパブリック IP アドレスで要求パケットを受信した場合、Citrix ADC は、宛先 IP アドレスをサーバーのプライベート IP アドレスに置き換えます。つまり、アプライアンスはクライアントとサーバー間のプロキシとして機能します。INAT 構成には、NetScaler アプライアンスの IP アドレスとサーバーの IP アドレスの間の 1 対 1 の関係を定義する INAT 規則が含まれます。

RNAT：逆ネットワークアドレス変換（Reverse Network Address Translation：RNAT）では、サーバーによって開始されたセッションについて、NetScaler アプライアンスは、サーバーが生成したパケットの送信元 IP アドレスをアプライアンスで設定された IP アドレス（種類：SNIP）に置き換えます。これにより、サーバーが生成したパ

ケットでサーバーの IP アドレスがさらされるのを防止します。RNAT 構成には、条件を指定する RNAT 規則が含まれます。アプライアンスは、条件に一致するパケットに対して RNAT 処理を実行します。

ステートレス **NAT46** 変換：ステートレス NAT46 は、セッション情報を NetScaler アプライアンスに保持せずに、IPv4 パケットと IPv6 パケットを相互に変換することにより、IPv4 ネットワークと IPv6 ネットワーク間の通信を実現します。ステートレス NAT46 構成には、IPv4-IPv6 INAT 規則と NAT46 IPv6 プレフィックスが含まれます。

ステートフル **NAT64** 変換：ステートフル NAT64 機能は、セッション情報を NetScaler アプライアンスに保持しながら、IPv6 パケットと IPv4 パケットを相互に変換することにより、IPv4 クライアントと IPv6 サーバー間の通信を実現します。ステートレス NAT64 構成には、NAT64 規則と NAT64 IPv6 プレフィックスが含まれます。

詳しくは、「ネットワークアドレス変換の構成」を参照してください。

マルチパス **TCP** のサポート

NetScaler アプライアンスでは、マルチパス TCP（Multipath TCP：MPTCP）がサポートされます。MPTCP は、ホスト間で使用可能な複数のパスを識別および使用して TCP セッションを保持する TCP/IP プロトコル拡張機能です。TCP プロファイルで MPTCP を有効にして仮想サーバーにバインドする必要があります。MPTCP が有効な場合、仮想サーバーは MPTCP ゲートウェイとして機能し、クライアントとの MPTCP 接続を、サーバーとの間で保持している TCP 接続に変換します。

詳細については、「MPTCP (マルチパス TCP)」を参照してください。

コンテンツスイッチ

ポリシーを切り替えるコンテンツの構成に基づいて要求を送信するサーバーを決定します。ポリシールールは、IP アドレス、URL、HTTP ヘッダーに基づいて設定できます。これにより、そのときのユーザー、使用されているエージェントの種類、ユーザーが要求したコンテンツなど、ユーザーとデバイスの特性に基づいて、スイッチを決定することができます。

詳しくは、「コンテンツスイッチ」を参照してください。

広域サーバー負荷分散（**Global Server Load Balancing**：**GSLB**）

NetScaler のトラフィック管理機能を拡張して、分散インターネットサイトとグローバル企業に対応します。設置場所が、複数のネットワークの場所や 1 箇所の複数のクラスターに分散していても、NetScaler は可用性を維持し、それらの間でトラフィックを分散します。インテリジェントな DNS 決定を行って、ダウンまたは過負荷状態のサイトにユーザーが割り当てられるのを防ぎます。近接ベースの GSLB 方式が有効な場合、NetScaler は、さまざまなサイトからクライアントのローカル DNS サーバー（LDNS）までの距離に基づいて、負荷分散の決定を行うことができます。距離ベースの GSLB 方式の最大の長所は、最も近い使用可能なサイトが選択されて、応答時間が短くなることです。

詳しくは、「グローバルサーバー負荷分散」を参照してください。

動的ルーティング

ルーターが、隣接するルーターからトポロジ情報、ルート、および IP アドレスを自動的に取得できるようにします。動的ルーティングが有効な場合、対応するルーティングプロセスはルート更新をリスンして、ルート情報を提供します。ルーティングプロセスはパッシブモードにすることもできます。ルーティングプロトコルを利用して、アップストリームルーターは Equal Cost Multipath 手法を使用し、2 台のスタンドアロン NetScaler 装置にホスティングされた同一の仮想サーバーに、トラフィックを負荷分散することができます。

詳細については、ダイナミックルートの設定を参照してください。

リンク負荷分散

複数の WAN リンクを負荷分散して、リンクフェールオーバーを提供し、ネットワークのパフォーマンスをさらに最適化して、ビジネスの継続性を保証します。インテリジェントなトラフィック制御とヘルスチェックを行って、アップストリームルーター間で効率的にトラフィックを分散することにより、ネットワーク接続の高い可用性を維持します。ポリシーとネットワーク状態に基づいて、着信トラフィックと送信トラフィックの両方をルーティングする最適な WAN リンクを特定し、高速な障害検出とフェールオーバーによって、WAN やインターネットリンク障害からアプリケーションを保護します。

詳細については、「リンク負荷分散」を参照してください。

TCP 最適化

TCP プロファイルを使用すると、TCP トラフィックを最適化できます。TCP プロファイルでは、NetScaler 仮想サーバーによる TCP トラフィックの処理方法を定義します。管理者は、組み込みの TCP プロファイルを使用するか、カスタムプロファイルを作成することができます。TCP プロファイルを定義した後、そのプロファイルを 1 つまたは複数の仮想サーバーにバインドできます。

TCP プロファイルで有効にできる主要な最適化機能のいくつかは次のとおりです。

- TCP Keep-Alive - リンクが切断されるのを防ぐために、指定された間隔で通信先の動作状態をチェックします。
- SACK（Selective Acknowledgment：選択的確認応答）- 特に LFN（Long Fat Network：広帯域高遅延ネットワーク）において伝送のパフォーマンスを向上させます。
- TCP ウィンドウスケーリング—LFN 経由の効率的なデータ転送を可能にします。

TCP プロファイルの詳細については、TCP プロファイルの設定を参照してください。

CloudBridge Connector

Citrix OpenCloud フレームワークの基本的な部分である NetScaler CloudBridge Connector 機能は、クラウド拡張データセンターの構築に使用されるツールです。OpenCloud Bridge により、ネットワークを再構成すること

なく、クラウド上の 1 つ以上の NetScaler アプライアンスまたは NetScaler 仮想アプライアンスをネットワークに接続することができます。クラウドがホストするアプリケーションは、組織内の単一ネットワーク上で実行されているかのように動作します。OpenCloud Bridge の主な目的は、企業がアプリケーションをクラウドに移行しながら、コストやアプライアンス障害のリスクを削減できるようにすることにあります。また、OpenCloud Bridge は、クラウド環境のネットワークセキュリティを向上します。OpenCloud Bridge は、クラウドインスタンス上の NetScaler アプライアンスまたは NetScaler 仮想アプライアンスを LAN 上の NetScaler アプライアンスまたは NetScaler 仮想アプライアンスに接続するレイヤー 2 ネットワークブリッジです。接続は、GRE（Generic Routing Encapsulation）プロトコルを使用するトンネルを介して確立されます。GRE プロトコルは、さまざまなネットワークプロトコルからのパケットをカプセル化し、別のプロトコル経由で転送するメカニズムを提供しています。IPSec（Internet Protocol Security：インターネットプロトコルセキュリティ）プロトコルは、OpenCloud Bridge のピア間の通信を確保します。

詳細については、CloudBridgeを参照してください。

DataStream

NetScaler DataStream 機能は、送信中の SQL クエリに基づいて要求を分散することで、データベース層で要求の割り振りを実行するインテリジェントなメカニズムを提供します。

データベースサーバーの前に NetScaler を導入すれば、アプリケーションサーバーまたは Web サーバーからのトラフィックを最適に分散することができます。管理者は、SQL クエリの情報と、データベース名、ユーザー名、文字セット、およびパケットサイズに基づいて、トラフィックをセグメント化できます。

負荷分散を構成して、負荷分散アルゴリズムに基づいて要求を割り振ることができます。または、ユーザー名、データベース名、コマンドパラメーターなどの SQL クエリパラメーターに基づくコンテンツスイッチを構成して、スイッチ条件を詳細に設定できます。さらに、モニターを構成してデータベースサーバーの状態を監視することもできます。

NetScaler アプライアンスの高度なポリシーインフラストラクチャには、要求の評価と処理に使用できる式が含まれています。高度な式により、MySQL データベースサーバーに関連付けられたトラフィックが評価されます。高度なポリシーの要求ベースの式（MYSQL.CLIENT および MYSQL.REQ で始まる式）を使用すると、コンテンツスイッチ仮想サーバーのバインドポイントで要求スイッチの意思決定を行うことが可能になり、応答ベースの式（MYSQL.RES で始まる式）を使用すると、ユーザー設定のヘルスモニターへのサーバー応答を評価することができます。

> 注：DataStream は、MySQL と MS SQL のデータベースでサポートされています。

詳細については、DataStreamを参照してください。

アプリケーションの速度向上機能

August 15, 2023

- AppCompress

  gzip 圧縮プロトコルを使った HTML とテキストファイルに対する透過的圧縮一般的な 4：1 の圧縮率で、データセンターの帯域幅要件が、最大 50% 削減されます。また、ユーザーのブラウザーに渡す必要があるデータ量が削減されるため、エンドユーザーの応答時間が大幅に短縮されます。

- キャッシュリダイレクト

  リバースプロキシ、透過プロキシ、またはフォワードプロキシのキャッシュファームに対するトラフィックのフローを管理します。すべての要求を検査して、キャッシュ不能な要求を特定し、固定接続を介してそれらを発信元のサーバーに直接送信します。キャッシュ不能な要求を発信元の Web サーバーへインテリジェントにリダイレクトすることによって、NetScaler ADC アプライアンスはキャッシュリソースを解放し、キャッシュヒット率を上げながら、これらの要求に対する全体的な帯域幅消費と応答遅延を削減します。

  詳細については、「キャッシュリダイレクト」を参照してください。

- AppCache

  静的および動的コンテンツの両方に対して、高速インメモリ HTTP/1.1 および HTTP/1.0 準拠の Web キャッシュを提供し、Web コンテンツとアプリケーションデータデリバリーを最適化します。このオンボードキャッシュは、着信要求がセキュリティで保護されたり、データが圧縮されたりしている場合でも、着信アプリケーション要求の結果を保存し、データを再利用して、同じ情報に対する今後の要求に対応します。オンボードキャッシュから直接データを提供することによって、静的および動的コンテンツ要求をサーバーに送信する必要がなくなるので、アプライアンスはページの再生成時間を削減できます。

  詳細については、「統合キャッシュ」を参照してください。

- TCP バッファリング

  サーバーの応答をバッファリングして、そのクライアントの速度でクライアントに応答を提供し、より高速にサーバーをオフロードするので、Web サイトのパフォーマンスが向上します。

アプリケーションセキュリティとファイアウォール機能

August 15, 2023

以下は、セキュリティとファイアウォールの機能です。

サービス不能（**DoS**）攻撃に対する防御

悪意ある分散型サービス不能（DDoS：Distributed Denial of Service）攻撃や、そのほかの悪意ある攻撃をサーバーに達する前に検出して阻止し、ネットワークおよびアプリケーションのパフォーマンスに悪影響を及ぼさないようにします。NetScaler アプライアンスは適正なクライアントを特定して、その優先度を引き上げ、疑わしいクライア

ントが不相応に大量のリソースを消費してサイトをダウンすることがないようにします。アプライアンスは、次のような種類の悪意ある攻撃に対するアプリケーションレベルの保護を提供します。

- SYN フラッド攻撃
- パイプライン攻撃
- ティアドロップ攻撃
- ランド攻撃
- 帯域枯渇攻撃
- ゾンビ接続攻撃

アプライアンスは、これらの接続へのサーバーリソースの割り当てを禁止して、これらの種類の攻撃を積極的に防御します。これによって、これらのイベントに関連するパケットが原因でサーバーに大きな負担がかかることがないようにします。

アプライアンスは、ICMP 率の制限と積極的な ICMP パケットチェックを使用して、ICMP ベースの攻撃からもネットワークリソースを保護します。強力な IP 再構築を実行して、さまざまな疑わしい、間違った形式のパケットをドロップし、サイトトラフィックにアクセス制御リスト（ACL：Access Control List）を適用して保護機能を強化します。

詳細については、「AppQoE」を参照してください。

コンテンツフィルタリング

レイヤー 7 レベルで、Web サイトを悪意のある攻撃から保護します。アプライアンスは、HTTP ヘッダーに基づくユーザー構成の規則に従って、各着信要求を検査し、ユーザーが構成したアクションを実行します。アクションとして、接続の再設定、要求のドロップ、ユーザーのブラウザーへのエラーメッセージの送信などを設定できます。これにより、不要な要求を除去して、サーバーが攻撃にさらされる危険性を減らすことができます。

この機能は、HTTP GET と POST 要求を分析して、既知の不正なシグニチャを排除できるので、HTTP ベースの攻撃からサーバーを防御できます。

詳細については、「コンテンツフィルタ」を参照してください。

レスポンダー

高度なフィルタリングのように機能し、アプライアンスからクライアントへの応答を生成するために使用できます。この機能の一般的な用途は、リダイレクト応答、ユーザー定義応答、およびリセットの生成です。

詳しくは、「レスポンダー」を参照してください。

書き換え

HTTP ヘッダーと本文のテキストを変更します。再書き込み機能を使用して、HTTP 要求または応答に HTTP ヘッダーを追加したり、個別の HTTP ヘッダーを変更したり、HTTP ヘッダーを削除したりできます。また、要求と応答

の HTTP ボディを変更することもできます。

アプライアンスは、要求を受信したり応答を送信したりするときに書き換え規則をチェックして、適切な規則を要求や応答に適用してから Web サーバーまたはクライアントコンピューターに渡します。

詳細については、書き換えを参照してください。

サージ保護

サーバーへのユーザー要求のフローを調整し、サーバー上のリソースへ同時にアクセスできるユーザー数を制御して、サーバーの容量に達した場合には、追加の要求をキューに入れます。接続を確立できるレートを制御することによって、アプライアンスは、サーバーに渡される要求のサージをブロックし、サイトがオーバーロード状態になるのを防ぎます。

詳しくは、「サージ保護」を参照してください。

NetScaler Gateway

NetScaler Gateway はアプリケーションアクセスのセキュリティを保護するソリューションで、詳細なアプリケーションレベルのポリシーと操作の制御機能を管理者に提供し、ユーザーがどこにいても作業できるようにすると同時に、アプリケーションとデータへのアクセスのセキュリティを保護します。IT 管理者は一拠点からツールを使用して、企業内外の規制順守および高度な情報セキュリティの確保を支援できます。それと同時に、ユーザーは役割、デバイス、およびネットワークに応じて最適化された単一のアクセスポイントを経由して、必要なエンタープライズアプリケーションとデータを使用できます。この 2 つの機能のユニークな組み合わせによって、今日のモバイルワーカーの生産性を最大限に向上させることができます。

詳しくは、Netscaler Gateway のドキュメントを参照してください。

アプリケーションファイアウォール

保護された各 Web サーバーと、その Web サーバー上の Web サイトに接続するユーザー間のトラフィックをフィルター処理して、クロスサイトスクリプティング攻撃、バッファーオーバーフロー攻撃、SQL インジェクション攻撃、強制的ブラウズなど、ハッカーやマルウェアによる悪用からアプリケーションを保護します。アプリケーションファイアウォールは、Web サーバーセキュリティに対する攻撃や、Web サーバーリソースの悪用の形跡がないかどうか、すべてのトラフィックを調べて適切なアクションを実行し、これらの攻撃を未然に防ぎます。

詳しくは、「アプリケーションファイアウォール」を参照してください。

アプリケーションの可視性機能

March 20, 2024

- NetScaler Application Delivery Management

  NetScalerApplication Delivery Management（ADM）は、Web および HDX（ICA）トラフィック全体のユーザーエクスペリエンスをエンドツーエンドで可視化する高性能コレクターです。NetScaler アプライアンスによって生成された HTTP および ICA AppFlow レコードを収集し、レイヤー 3 からレイヤー 7 の統計を含む分析レポートを作成します。NetScaler Console では、過去 5 分間のリアルタイムデータと、過去 1 時間、1 日、1 週間、1 か月で収集された履歴データを詳細に分析できます。

  HDX（ICA）分析ダッシュボードでは、HDX ユーザー、アプリケーション、デスクトップ、およびゲートウェイレベルの情報をドリルダウンできます。同様に、HTTP 分析では、Web アプリケーション、アクセスされた URL、クライアント IP アドレス、サーバー IP アドレス、およびそのほかのダッシュボードの概観を表示します。管理者は、ユースケースに合わせて、これらのダッシュボードからドリルダウンして問題点を明らかにできます。

- AppFlow を使用した拡張されたアプリケーションの可視性

  NetScaler アプライアンスは、データセンター内のすべてのアプリケーショントラフィックを一元的に制御します。これは、アプリケーションパフォーマンスの監視、分析、およびビジネスインテリジェンスアプリケーションにとって有効なフローとユーザーセッションレベルの情報を収集します。AppFlow は、RFC 5101 で定義されたオープンな IETF（Internet Engineering Task Force：インターネット技術標準化委員会）標準である IPFIX（Internet Protocol Flow Information eXport）形式を使用して、この情報を送信します。IPFIX（Cisco 社製 NetFlow の標準化バージョン）は、ネットワークフロー情報を監視するために幅広く使用されています。AppFlow は、新しい情報要素を定義してアプリケーションレベルの情報を表現します。

  トランスポートプロトコルとして UDP を使用して、AppFlow は フローレコードと呼ばれる収集されたデータを 1 つまたは複数の IPv4 コレクターに送信します。コレクターはフローレコードを集約し、リアルタイムレポートまたは履歴レポートを生成します。

  AppFlow は、HTTP、SSL、TCP、および SSL_TCP フローのトランザクションレベルでの可視性を実現します。監視対象のフロータイプのサンプリングとフィルタリングを行うことが可能です。

  アプリケーショントラフィックのサンプリングとフィルタリングを行うことで監視するフロータイプを制限する場合は、AppFlow を仮想サーバー向けに有効化できます。AppFlow では、仮想サーバーの統計情報も提供しています。

  また、AppFlow を特定のサービス向けに有効化してアプリケーションサーバーを表現し、そのアプリケーションサーバーへのトラフィックを監視することもできます。

  詳細については、AppFlowを参照してください。

- ストリーム分析

  Web サイトやアプリケーションのパフォーマンスは、最も頻繁に要求されるコンテンツの配信をどのように最適化するかにより決まります。キャッシュや圧縮などの方法は、クライアントへのサービス配信の高速化に役立ちますが、最も頻繁に要求されるリソースを特定し、それらのリソースをキャッシュまたは圧縮できるようにする必要があります。Web サイトやアプリケーショントラフィックに関するリアルタイム統計を集計すれば、最も頻繁に使用されるリソースを特定できます。リソースごとのアクセス頻度や消費帯域幅などの統計

によって、サーバーパフォーマンスとネットワーク使用率を改善するために、それらのリソースをキャッシュまたは圧縮する必要があるかどうかを判断できます。応答時間やアプリケーションへの同時接続数などの統計は、サーバー側のリソースを強化する必要があるかどうかを判断するのに役立ちます。

Web サイトやアプリケーションが頻繁に更新されない場合、統計データを収集する製品を使用して、その統計を手動で分析し、コンテンツの配信を最適化できます。ただし、最適化を手動で行わない場合や、Web サイトまたはアプリケーションのコンテンツが動的に生成される場合、統計データを収集するだけでなく、その統計に基づいてリソースの配信を自動的に最適化できるインフラストラクチャが必要です。NetScaler アプライアンスでは、この機能はストリーム分析機能によって提供されます。この機能は単一の NetScaler アプライアンス上で動作し、定義した基準に基づいて実行時の統計を収集します。NetScaler ポリシーと共に使用すると、この機能によって自動的なリアルタイムトラフィックの最適化に必要なインフラストラクチャも提供されます。

詳細については、アクション分析を参照してください。

NetScaler ADC ソリューション

August 15, 2023

NetScaler ソリューションを使用すると、頻繁に導入される構成をセットアップする作業が簡単になります。解決策が追加されているか参照するために、このページを時々確認してください。

このセクションには、次のソリューションが含まれています。

- Citrix Virtual Apps and Desktops 用の NetScaler ADC の設定
- グローバルサーバー負荷分散（Global Server Load Balancing：GSLB）による優先ゾーン
- NetScaler ADC でのエニーキャストのサポート
- NetScaler を使用して AWS にデジタル広告プラットフォームをデプロイする
- NetScaler による AWS でのクリックストリーム分析の強化
- Microsoft Windows Azure パックと Cisco ACI によって管理されるプライベートクラウド内の NetScaler

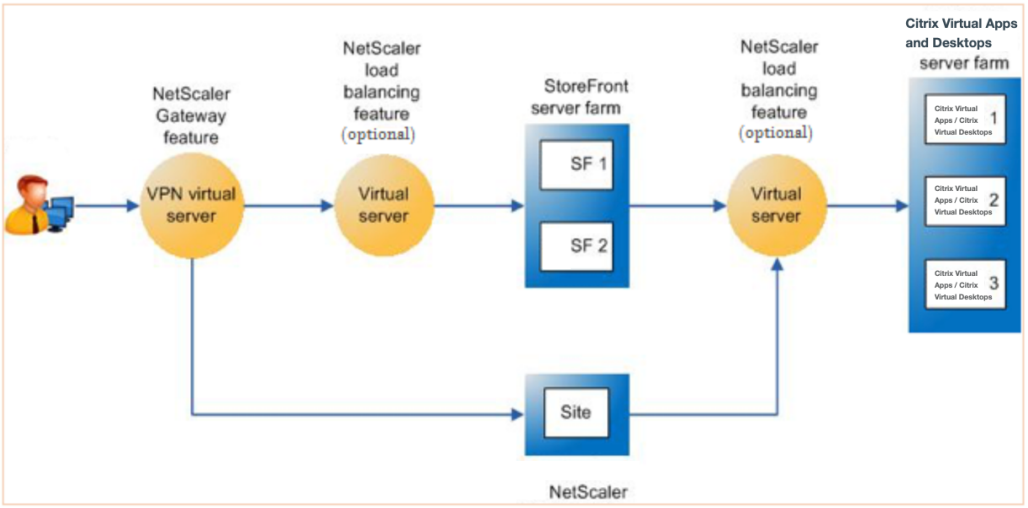
Citrix Virtual Apps and Desktops 用の NetScaler ADC の設定

August 15, 2023

NetScaler ADC アプライアンスは、Citrix Virtual Apps and Desktops アプリケーションへの負荷分散された安全なリモートアクセスを提供できます。NetScaler 負荷分散機能を使用して、Citrix Virtual Apps and Desktops サーバー全体にトラフィックを分散できます。NetScaler Gateway 機能を使用して、サーバーへの安全なリモートアクセスを提供できます。

NetScaler ADC は、トラフィックフローを高速化および最適化し、Citrix Virtual Apps and Desktops の導入に役立つ可視化機能も提供します。

図 1: Citrix Virtual Apps and Desktops セットアップの NetScaler アプライアンス



上の図は、この導入に関連するコンポーネントを示しています。

- で接続する必要があります。ユーザーアクセス用の URL を提供し、ユーザーを認証することでセキュリティを提供します。
- **NetScaler** 負荷分散仮想サーバー。StoreFront サーバーのトラフィックを負荷分散します。また、Citrix Virtual Apps および Desktop サーバーの前に負荷分散仮想サーバーを展開して、XML ブローカーや Desktop Delivery Controller（DDC）サーバーなどの主要コンポーネントの負荷を分散することもできます。
- **Citrix Virtual Apps and Desktops.** ユーザーがアクセスしたいアプリケーションを提供します。

NetScaler GUI を使用して NetScaler ADC for Citrix Virtual Apps and Desktops をセットアップするには

前提条件

- Citrix Virtual Apps とデスクトップサーバーが構成され、使用可能になっています。
- NetScaler Gateway、NetScaler、Citrix Virtual Apps and Desktops、StoreFront に関する実務知識がある
- 仮想サーバーとサービスを構成し、サービスを仮想サーバーにバインドしていることを確認します。詳しくは、次のトピックを参照してください:
  - Citrix 仮想アプリとデスクトップの負荷分散

- Citrix 仮想アプリとデスクトップの負荷分散

手順:

1. NetScaler アプライアンスにログオンし、[構成] タブで [**XenApp**] と [**XenDesktop**] をクリックします。
2. 詳細ペインで、「はじめに」をクリックします。NetScaler にセットアップが存在する場合は、変更する各セクションに対応する「編集」リンクをクリックします。
3. 導入環境内で Citrix Virtual Apps and Desktops アプリケーションにアクセスするためのインターフェイスを提供する製品（StoreFront）を選択します。
4. 安全なリモートアクセスを設定します。
   a) **NetScaler Gateway** 設定セクションで、VPN 仮想サーバーの詳細を指定し、「続行」をクリックします。
   b)「サーバー証明書」セクションで、既存の証明書を選択するか、新しい証明書をインストールして、「続行」をクリックします。
   c)「認証」セクションで、使用するプライマリ認証メカニズムを設定し、サーバーの詳細を指定するか、既存のサーバーを使用して、「続行」をクリックします。
   d) **StoreFront** セクションで、アプリケーションにアクセスするためのインターフェイスを提供するサーバーの詳細を指定し、「続行」をクリックします。
   e) 複数の SF サーバーを指す LB 仮想サーバーを StoreFStoreFront サーバーとして使用できます。
5. [完了] をクリックして設定を完了します。

グローバルサーバー負荷分散（**Global Server Load Balancing**：**GSLB**）による優先ゾーン

August 15, 2023

GSLB 搭載ゾーンの優先設定は、Citrix Virtual Apps and Desktops、StoreFront、および NetScaler ADC を統合して、クライアントがクライアントの場所に基づいて最も最適化されたデータセンターにアクセスできるようにする機能です。

Citrix Virtual Apps and Desktops の分散展開では、複数のデータセンターから同等のリソースが複数ある場合、StoreFront が最適なデータセンターを選択しない場合があります。このような場合、StoreFront はデータセンターをランダムに選択します。リクエストを送信するクライアントとの距離に関係なく、任意のデータセンターにある任意の Citrix Virtual Apps and Desktops サーバーにリクエストを送信できます。

クライアント IP アドレスは、HTTP リクエストが NetScaler Gateway アプライアンスに到着したときに調べられます。実際のクライアント IP アドレスを使用して、StoreFront に転送されるデータセンターの優先リストを作成します。NetScaler ADC アプライアンスがゾーン優先ヘッダーを挿入するように構成されている場合、StoreFront

3.5 以降では、アプライアンスから提供された情報を使用してデリバリーコントローラーの一覧を並べ替え、クライアントと同じゾーンにある最適なデリバリーコントローラーに接続できます。StoreFront は、選択したデータセンターゾーンに最適なゲートウェイ VPN 仮想サーバーを選択し、この情報を適切な IP アドレスを使用して ICA ファイルに追加し、クライアントに送信します。次に、Storefront は、優先データセンターの配信コントローラーでホストされているアプリケーションを起動してから、他のデータセンターにある同等のコントローラーに接続しようとします。

このソリューションの構成の詳細については、ここをクリックしてください。

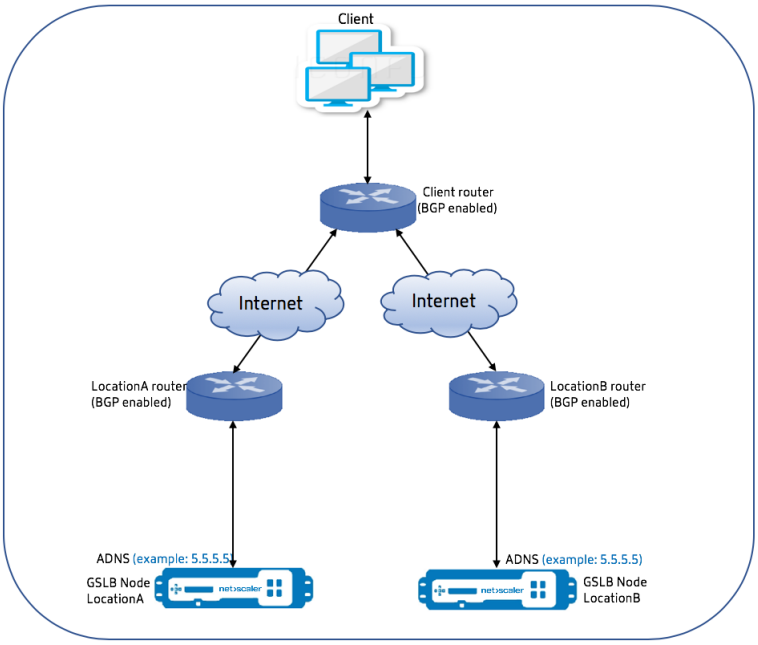
NetScaler ADC でのエニーキャストのサポート

August 15, 2023

エニーキャストは、複数のサーバーが IP アドレスを共有するネットワークの一種です。クライアントリクエストは、ルーティングテーブルに基づいて地形的に最も近いサーバーに送信されます。このルーティングにより、レイテンシーの問題が軽減され、高可用性が確保され、ダウンタイムが最小限に抑えられます。

NetScaler は、グローバルサーバー負荷分散（GSLB）と DNS 機能を備えたエニーキャストネットワークをサポートしています。

次の図は、NetScaler のエニーキャストのトポロジー図を示しています。



エニーキャスト **GSLB**

NetScaler GSLB 機能により、災害復旧とともにグローバルに分散したサイト間の負荷分散が可能になり、アプリケーションの継続的な可用性が保証されます。

停電時には、GSLB は最も近いデータセンターまたは最もパフォーマンスの高いデータセンターにトラフィックをルーティングすることで、即時の障害復旧を実現します。ただし、GSLB は以下を制御できません。

- DNS トラフィックが地理的に異なる場所にある GSLB ノードにルーティングされる方法。
- DNS クエリが GSLB ノードにルーティングされる間にどれだけの待ち時間が追加されるか。

一般的な GSLB 設定では、各データセンターには DNS クエリを受信するようにサイト固有の権限のあるドメインネームサーバー（ADNS）で構成された GSLB ノードがあります。各サイトの ADNS は、DNS リゾルバーのネームサーバーとして設定されます。GSLB ノードの数が増えると、ネームサーバーレコードの数も増えます。このような場合、データセンターに障害が発生した場合、LDNS は別のネームサーバーで解決を再試行する必要があります。この再試行により、DNS 解決の待ち時間が長くなります。
また、GSLB ノードが追加されるたびに、ネームサーバーのレコードを更新する必要があります。

これらの欠点を克服するには、エニーキャスト ADNS を使用できます。エニーキャスト ADNS では、すべての GSLB ノードに単一の ADNS IP アドレスが使用され、DNS トラフィックは動的ルーティングを使用して GSLB ノードにルーティングされます。

たとえば、GSLB サイトがダウンしている場合、ルーティングテーブルが更新され、このサイトへのルートが削除されます。その結果、DNS クエリはダウンしているサイトには送信されません。その結果、再試行は行われません。

新しい GSLB ノードが追加された場合、新しいノードには同じ ADNS IP アドレスが割り当てられます。動的ルーティングは、ルーティングアルゴリズムに基づいて新しいサイトへのルートを含むルーティングテーブルを自動的に更新します。そのため、DNS ネームサーバーレコードを更新する必要はありません。Anycast を使用すると、新しい GSLB サイトの展開がより簡単かつ迅速になります。

エニーキャストモードで **ADNS IP** アドレスを設定する方法

NetScaler アプライアンスの ADNS IP でホストルーティングを有効にし、適切なルートヘルスインジェクション（RHI）レベルを設定します。ほとんどの場合、ADNS IP には仮想サーバーが存在しないため、RHI レベルを NONE として選択する必要があります。ADNS IP でホストルートを有効にすると、カーネルルートになります。その後、任意の動的ルーティングを有効にし、ルーティングプロトコルを設定してカーネルルートを再配布できます。

ADNS IP 構成—例

コマンドプロンプトで次を入力します。

```
add service adns_public 5.5.5.5 ADNS 53

set ip 5.5.5.5 -hostRoute ENABLED -vserverRHILevel ALL_VSERVERS
```



```
<!--NeedCopy-->
```

GSLB サイトの **BGP** 設定—例

```
Site1#sh run
!
hostname Site1
!
log syslog
log record-priority
!
ns route-install bgp
!
interface lo0
 ip address 127.0.0.1/8
 ipv6 address fe80::1/64
 ipv6 address ::1/128
!
interface vlan0
 ip address 10.102.148.94/25
 ipv6 address fe80::e84c:f4ff:fe74:4588/64
!
interface vlan2
 ip address 172.18.30.15/24
!
router bgp 5
 redistribute kernel -----> redistributing the kernel routes
 neighbor 172.18.30.30 remote-as 4
 neighbor 172.18.30.30 advertisement-interval 1
 neighbor 172.18.30.30 timers 4 16
!
End

Site1#
<!--NeedCopy-->
```

GSLB サイトルーティングテーブル-例

```
Site1#sh ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, I - Intranet
       * - candidate default

K       5.5.5.5/32 via 0.0.0.0 ---------------------------------------->
    Kernel Route for ADNS
C       10.102.148.0/25 is directly connected, vlan0
C       127.0.0.0/8 is directly connected, lo0
B       172.18.10.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
```



```
B       172.18.20.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
C       172.18.30.0/24 is directly connected, vlan2
B       192.168.3.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
B       192.168.5.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
B       192.168.10.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h

Gateway of last resort is not set
Site1#
<!--NeedCopy-->
```

エニーキャスト **DNS**

NetScaler 上の DNS プロキシ仮想サーバーにはエニーキャスト DNS を使用できます。複数の DNS ネームサーバーが設定されている場合、DNS リゾルバーはラウンドロビン方式に基づいて応答します。たとえば、リゾルバーが最初のサーバーから応答を受け取らない場合、設定されたタイムアウト値の期限が切れると、リゾルバーは 2 番目のサーバーに切り替わります。1 番目のサーバーから 2 番目のサーバーに切り替えると、DNS 解決の待ち時間が長くなります。DNS リゾルバーがエニーキャストで構成されている場合、このレイテンシーは解消できます。

DNS 設定—例

コマンドプロンプトで次を入力します。

```
add lb vserver dns DNS 5.5.5.50 53

set ip 5.5.5.50 -hostRoute ENABLED -vserverRHILevel ALL_VSERVERS
<!--NeedCopy-->
```

NetScaler を使用して **AWS** にデジタル広告プラットフォームをデプロイする

March 20, 2024

デジタルプラットフォームの性質が進化するにつれて、さまざまな広告アプリケーションが利用できるようになっています。たとえば、ソーシャルメディア、ダイレクトメール、ビデオ、バナー、ポップス、インタースティシャル、リッチメディアなどです。広告主は急速に動画広告ネットワークを採用しており、広告トラフィックの約 40% を占めています。しかし、現代のユーザーによるモバイルの使用が増えるにつれ、モバイルプラットフォームでの動画広告の掲載が急増しています。

デジタル広告プラットフォームはいくつかの課題に直面しています。課題のいくつかは以下のとおりです。

- セキュリティ脅威
- 高い運用コスト

- インターネット経由でトラフィックを送信するには、さまざまなデバイスを使用できます。リアルタイム通信のさまざまなプロトコルには、次のような課題があります。
  - WebRTC
  - アダプティブストリーミング
  - ビデオ用の UDP (WebRTC は HTTP 経由で UDP を使用する)

広告プラットフォームの複雑な挙動に対処するため、NetScaler ソリューションは機能一式を AWS とうまく統合し、いつでもどこでもデジタル広告インベントリに瞬時に安全かつ確実にアクセスできるようにします。NetScaler は、デジタルプラットフォーム向けの SaaS および Web アプリを提供する上で重要な役割を果たします。

NetScaler とのデジタル広告プラットフォームの統合

デジタル広告プラットフォームの概要

デジタル広告プラットフォームは、以下の主要コンポーネントで構成されています。

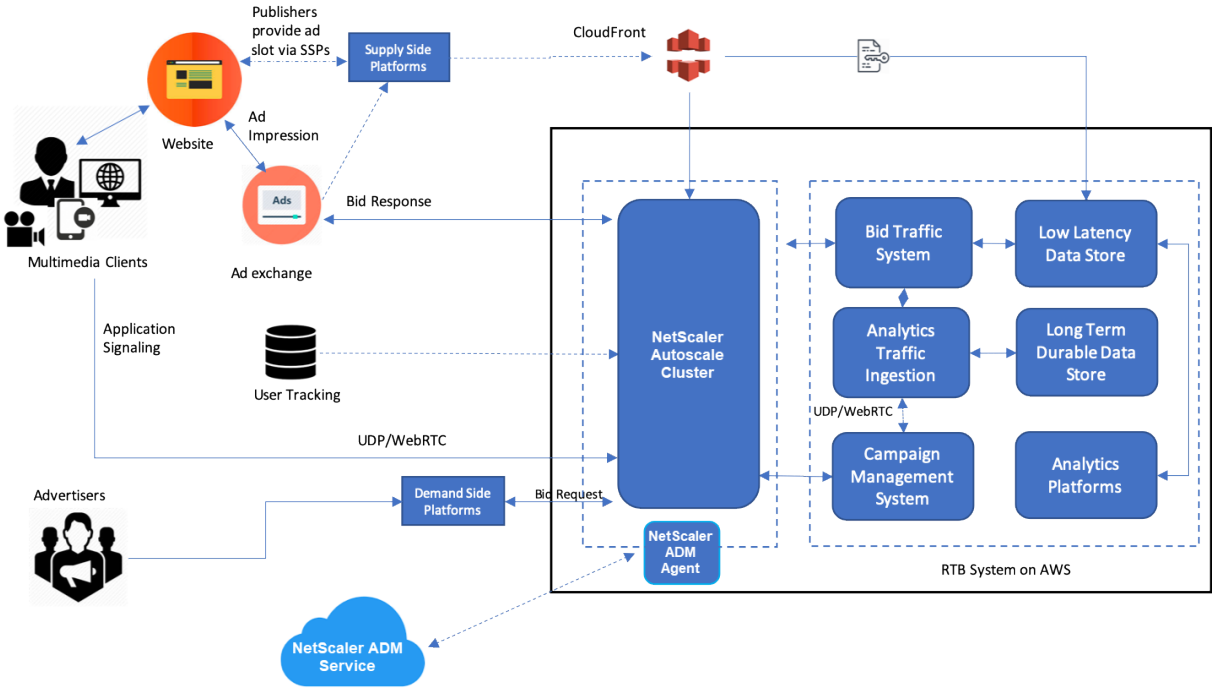
- 広告交換
- 広告ネットワーク
- デマンドサイドプラットフォーム (DSP)
- サプライサイドプラットフォーム (SSP)
- リアルタイム入札 (RTB) システム

広告システムで行われるプロセスの概要は次のとおりです。

- 最初のトランザクションは、ユーザーがウェブサイトにアクセスしたときに発生します。
- これにより、入札/広告リクエスト（ユーザーの人口統計情報を含む）がトリガーされ、広告サーバーまたは発行元が広告エクスチェンジに連絡して送信されます。
- 広告発行元は、SSP を通じて広告リクエストを広告エクスチェンジに送信します。
- アドエクスチェンジは、このリクエストとそれに付随するデータを DSP に送信し、インプレッションまたは広告リクエストが可能であることを伝えます。そのため、複数の広告主がリアルタイムで自動的に入札を行い、広告を掲載することができます。
- 一方、広告主は DSP でキャンペーンを設定する必要があります。データ管理プラットフォーム（DMP）からのユーザーに関する情報を使用して、ユーザーに広告を配信するためにユーザーが支払う意思のある金額を評価します。
- DSP は、広告エクスチェンジに配信されるため、広告インプレッションのたびにリアルタイムに入札を行います。
- 広告エクスチェンジまたは SSP が設定した期間内に最も多く入札した入札者が、発行元から広告を配信するための広告枠を獲得します。そうしないと、主要なユーザー層に適した広告を入手する機会が失われます。

デジタル広告プラットフォームが **NetScaler** とどのように統合されているか

次の図は、広告プラットフォームのさまざまなコンポーネントが NetScaler および CitrixApplication Delivery Management（ADM）と通信してオンライン広告を提供する方法を示しています。



NetScaler がどのように貢献するか　広告公開プロセスでは、NetScaler ソリューションが絶え間なく流入する入札トラフィックの処理と処理に役立ちます。すべてのトラフィックのエントリポイントとして機能し、アベイラビリティーゾーン全体のスケーラビリティと可用性を確保します。広告トラフィックの弾力性に応えるため、ウェブアプリケーションやデータベースサーバーの前にある自動スケーリンググループにデプロイされます。

AWS の NetScaler ソリューションを使用した広告プラットフォームを使用すると、世界中でリアルタイムのパフォーマンス、高いスケーラビリティ、および高可用性を実現できます。リッチメディア、動画、モバイル、ネイティブ広告をリアルタイムで売買できます。これにより、広告プラットフォームの運営に伴う全体的な運用コストとレイテンシが削減されます。Autoscale e 中にバックエンドサーバーを正常に削除したり、接続を多重化したり、エンドユーザーのトラフィックに影響が及ばないようにしたりする豊富な機能を備えた、最高のパフォーマンスを備えたプロキシです。NetScaler は、広告プラットフォームで使用される HTTP、UDP、WebRTC、および RTSP プロトコルの負荷分散をサポートしています。

NetScaler は、以下の主な特徴により AWS 環境に一貫して適合します。

- コンテンツスイッチング—ホスト名に基づいて適切なプラットフォームに切り替えます。
- セキュリティ保護—Web アプリケーションファイアウォール (WAF) 機能、レート制限 (クライアント IP 経由)、および DDoS 攻撃からの保護を使用します。
- フロントエンドとバックエンドの両方のトラフィックの自動スケーリング。

- ADM を活用することで、ADC アプライアンス全体をエンドツーエンドで可視化し、異常を検知します。
- 低レイテンシー。

NetScaler コンソールの貢献方法　NetScaler は NetScaler Console を活用して、デジタル広告プラットフォームが直面する以下の課題を克服しています:

- 予想されるパフォーマンスからの傾向の偏差を特定
- リアルタイムのアプリケーションパフォーマンス分析
- キャパシティモニタリング

NetScaler および **ADM** との広告プラットフォーム統合の利点

NetScaler ソリューションには、デジタル広告プラットフォームベンダーに次のような機能とメリットがあります。

低コスト

- AWSAutoscaling サービスと統合された NetScaler VPX インスタンスは、フロントエンドとバックエンドのリソースを自動的にスケールアップまたはスケールダウンできます。これにより、広告プラットフォームの弾力性に合わせたゼロタッチ構成が可能になります。
- 単一のポイントからすべてのタイプのトラフィックを配信する統合。

AWS 自動スケーリングの詳細については、「バックエンド AWS 自動スケーリングサービスの追加」を参照してください。

高可用性

- あるアベイラビリティーゾーンが使用できなくなった場合、NetScaler はフォールトトレランス機能を適用して、トラフィックを中断することなく別のアベイラビリティーゾーンのサーバーを自動検出します。
- また、クライアント接続の損失を回避してサーバーを正常に終了します。

詳細については、「AWS での高可用性の仕組み」を参照してください。

アプリケーション・パフォーマンス分析　NetScaler Console のインテリジェントな分析とアプリケーションパフォーマンス分析により、次のことが保証されます:

- エンドユーザーエクスペリエンスを悩ませている問題 (サーバー応答の異常、5XX エラーなど) を可視化します。
- 管理者に警告して、すぐに修正措置を講じてください。

詳細については、「アプリケーション分析のパフォーマンス指標」を参照してください。

ファイアウォールのセキュリティ　最も一般的なセキュリティ脆弱性は、ネットワークではなくウェブアプリケーションで発生します。ボット、データ盗難、アプリケーション層攻撃などの不正アクセスからウェブアプリケーションを保護することは不可欠です。

NetScaler は、以下を含む包括的な統合レイヤー 4～レイヤー 7 セキュリティを提供します。

- Web App Firewall（WAF）は、定期的に更新されるボットシグネチャと行動ベースの検出機能により、ウェブアプリケーションを保護し、悪意のあるボットを識別して軽減します。
- 広告プラットフォームが圧倒されるのを防ぐためのレート制限。

詳細については、NetScaler Web App Firewallを参照してください。

広告プラットフォームに適した **AWS** インスタンスタイプを選択します

次の 2 つの要素に応じて、ADC に適した AWS インスタンスタイプを選択してください。

- 広告プラットフォームに同時にアクセスするユーザーの数。
- プラットフォーム上の平均ユーザー数。

NetScaler は、c5、c5n、m5 などを含むさまざまな EC2 インスタンスにデプロイできます。広告プラットフォームには、次の AWS インスタンスタイプを使用してください。

- c5 または c5n は SSL の多いトラフィックの処理に適しています。
- c5.large は、最大 1000 の SSL TPS を処理できます。

詳細については、VPX-AWS サポートマトリックスを参照してください。

NetScaler による **AWS** でのクリックストリーム分析の強化

March 20, 2024

お客様は、モバイルアプリ、SaaS アプリなどのさまざまなアプリケーションを通じて企業製品にアクセスするようになってきています。したがって、アプリケーションはカスタマーエクスペリエンスデータの地雷になる可能性があります。顧客行動をオンラインで追跡するために、顧客中心の企業は、この顧客行動データを使用して、各顧客に対してデータドリブンなプロファイルを形成します。

クリックストリームは、ウェブサイトまたはモバイルアプリケーションでのユーザーアクション（クリック）を表すイベントのシーケンスまたはストリームです。ただし、クリックストリームの範囲はクリックを超えています。これには、商品検索、インプレッション、購入、およびビジネスに関連する可能性のあるイベントが含まれます。カスタマーエクスペリエンスのデータを収集して保存するだけではあまり価値がありません。非常に複雑なデータを、適切なタイミングで適切なベンダーにシームレスに配信する必要があります。企業は、データから価値を引き出し、意識

的な意思決定を迅速に実行して、戦略を改善することができます。したがって、企業はクリックストリーム分析を使用して、アプリのカスタマーエクスペリエンスのジャーニーに関するインサイトを収集するようになっています。

このドキュメントでは、クリックストリームデータが最も重要である理由、収集、保存、分散、有意義で実用的な分析に変換される方法について、十分に理解しています。

NetScaler は NetScaler Console と統合され、Amazon Kinesis Data Firehose などの AWS サービスに付加価値をもたらし、ユーザーのクリックストリームを中心に展開するクラス最高の分析ソリューションを企業に提供します。

この NetScaler ソリューションは、複雑なビジネス上の問題を効率的かつ非常に簡単に解決するのに役立ちます。NetScaler と AWSKinesis は、不十分に設計されたワークフローの問題を把握するのに役立ちます。NetScaler Console は、関連するフィルターを適用することで、Web アプリやネットワークのパフォーマンスに関連する問題を把握するのに役立ちます。NetScaler と NetScaler Console および AWS Kinesis を組み合わせることで、各フェーズで大量に流入するクリックストリームデータを管理および分析できます。このソリューションは、可用性、拡張性、堅牢性を備え、継続的かつセキュアなデリバリーを保証します。したがって、実用的なインサイトを導き出すことができます。

企業がクリックストリーム分析を選択する理由

企業は主に、ユーザーがアプリケーションとのやりとりを理解し、アプリケーションの目標を改善するためのインサイトを得るために、クリックストリームを選択します。クリックストリーム分析は、ユーザーの行動、ナビゲーションの習慣などを追跡する情報検索ユースケースです。クリックストリーム分析では、次の情報が表示されます。

- 顧客がクリック頻度が高くどのリンクを、どの時点でクリックしているのですか。
- ウェブサイトにアクセスする前の訪問者はどこにいましたか?
- 訪問者は各ページにどれくらいの時間を費やしましたか?
- 訪問者がウェブブラウザの「戻る」ボタンをクリックしたのはいつですか?
- 訪問者がショッピングカートに追加（または削除）した商品はどれですか?
- 訪問者が私のウェブサイトを出たのはどのページからですか?

Amazon Kinesis を使用してクリックストリームデータを管理する分析サービス

Amazon Kinesis を使用して、クリックストリーム分析を実行できます。Amazon Kinesis によって、次のサービスでクリックストリーム分析ができます:

- Amazon Kinesis データファイヤーホース
- Amazon Kinesis データ分析
- Amazon Kinesis データストリーム

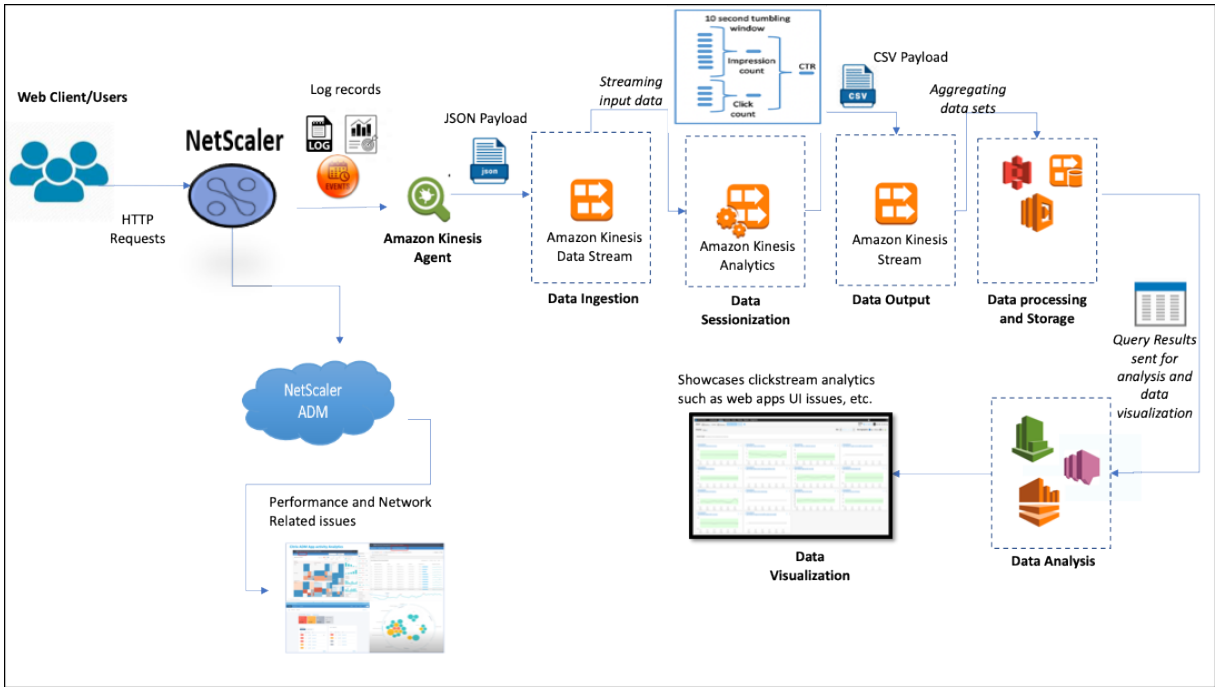
Amazon Kinesis を使用すると、あらゆる規模で膨大なデータセットを収集して分析できます。AWS Kinesis は、次のようなさまざまなソースからのデータを処理できます。

- モバイルおよびウェブアプリケーション（ゲーム、e コマースなど）
- IoT デバイス
- ソーシャルネットワーキングアプリケーション
- 金融取引サービス
- 地理空間サービス

NetScaler がクリックストリーム分析を可能にする方法

NetScaler ソリューションは、訪問したウェブサイト、消費された帯域幅、ナビゲーションフローなど、ユーザーのアクティビティに関する情報を安全に照合して提供します。企業は、この高いスループットと継続的なクリックストリームデータを分析して、次の効果を実証します。

- サイトレイアウト
- マーケティングキャンペーン
- 新しいアプリケーション機能



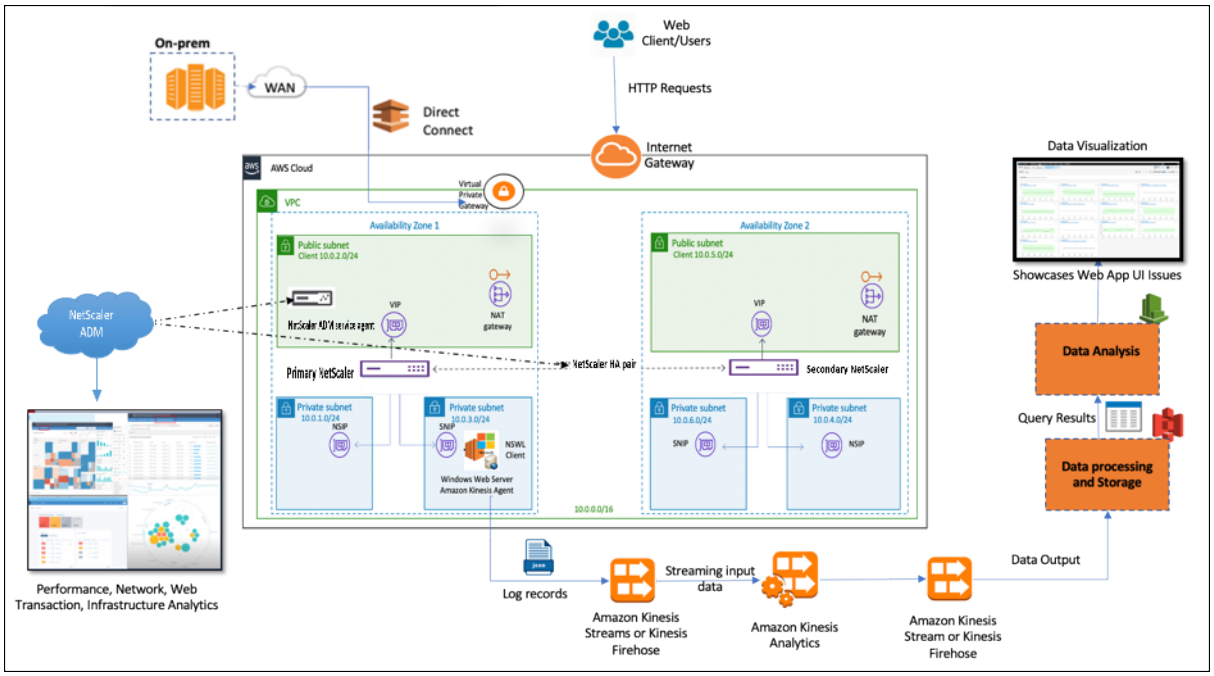
エンタープライズ環境に復元力のあるネットワーク保護を提供する NetScaler の機能により、計算量の多いタスクをオフロードし、このデータでセッションを実行することにより、サーバーのコストが大幅に削減されます。これにより、企業は常に高可用性、セキュリティ、低遅延でイベントをリアルタイムで識別できます。

構成情報については、「クリックストリーム分析用の NetScaler ソリューションを構成する」を参照してください。

NetScaler と NetScaler コンソールが AWS 環境を補完する方法

次の図は、AWS インフラストラクチャでクリックストリーム分析を実行するためのエンドツーエンドのユーザーワークフローを示しています。この図は、次のプロセスを理解するのに役立ちます。

- ユーザーが NetScaler と対話する方法
- NetScaler がユーザーのアクションをキャプチャしてクリックストリームデータを生成する方法
- クリックストリームデータが AWS サービスに配信される方法 (Amazon Kinesis)
- Amazon Kinesis がデータログを処理して保存し、有意義なクリックストリーム分析を生成する方法



NetScaler は AWS 環境と NetScaler Console にシームレスに統合されるため、企業は変動する量や多様なクリックストリームデータに対応できます。ストリーミングの知識を簡単に読み込み、分析するサービスを提供します。また、特殊な要望に合わせてカスタムストリーミングナレッジアプリケーションを作成することもできます。

Amazon Kinesis

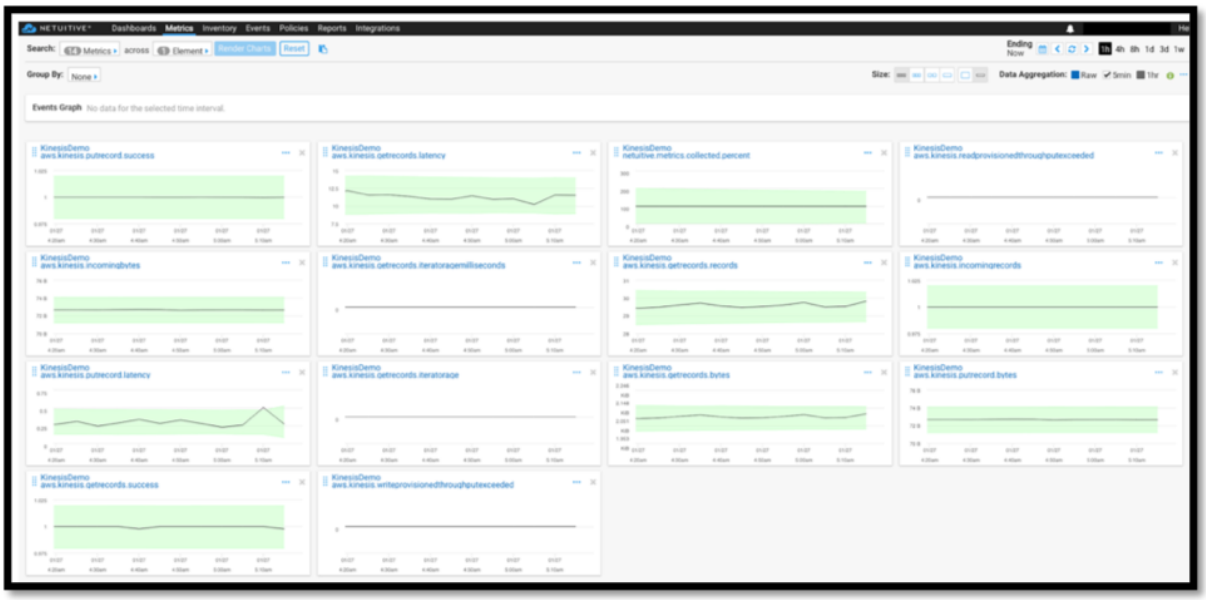
AWS 環境には、NetScaler によってキャプチャされたユーザーイベント、ログ、メトリックスの分析を実行するさまざまなサービスがあります。データには、ウェブサイトのクリックストリーム、財務トランザクション、ソーシャルメディアフィード、IT ログ、および位置追跡イベントが含まれます。

- Amazon Kinesis Data Streams は、複数のソースから 1 秒あたりのデータを継続的にキャプチャできる、スケーラブルで耐久性のあるリアルタイムデータストリーミングを含むシナリオで分析を実行します。
- Amazon Kinesis Data Analytics は、さまざまなデータセットの集約にかかる時間が短いため、セッション生成間のレイテンシーが短いシナリオで使用できます。

- Microsoft Windows 版 Amazon Kinesis エージェントは、入力データを収集、解析、フィルタリング、および Kinesis データストリームにストリーミングします。
- データがクラウドで稼働したら、正確なデータパイプラインを実装して、必要な結果を得ることができます。たとえば、この情報を Amazon Quick Sight で使用できます。Amazon Quick Sight は、ダッシュボードの作成に使用される視覚化ツールです。

AWS Kinesis ダッシュボードには、次のサービスがあります。

- ウェブアプリの UI の問題を紹介する
- 時間あたりのイベント、訪問者数、リファラーなど、Web 使用状況の指標をほぼリアルタイムで視覚化します。
- セッション単位の分析



NetScaler コンソール分析

NetScaler Console を NetScaler と併用することで、すべてのビジネス環境を一元的に把握できます。NetScaler でキャプチャされたログは NetScaler Console に送られ、個々のアプリケーションを 1 つのエンティティとして扱います。次の ADM 機能を使用すると、貴重なインサイトを獲得し、問題を効果的にトラブルシューティングできます:

- インテリジェント・アナリティクス
- Web トランザクション分析
- 異常検出
- パフォーマンスおよびネットワーク関連の問題

次の ADM サービスダッシュボードは、問題を効果的にトラブルシューティングするための貴重なインサイトを得るのに役立ちます。

NetScaler コンソールとクリックストリーム分析の相関関係

クリックストリーム分析データを ADM 分析と関連付けて、アプリケーションのパフォーマンスを記述、予測、および改善できます。

NetScaler コンソールの詳細については、次を参照してください。「[NetScaler コンソール](https://docs.citrix.com/en-us/tech-zone/design/reference-architectures/citrix-adm.html#:~:text=Citrix%20Application%20Delivery%20Management%20(ADM)%20is%20a%20centralized%20management%20solution.&text=You%20can%20use%20ADM%20to,from%20a%20single%2C%20unified%20console)

たとえば、組織がログを分析しているときに、ほとんどのユーザーがサイトを放棄していることに気づきました。しかし、このユーザー行動の根本的な原因を見つけるには、アプリケーションのどの部分が悪いのかを調べる必要があります。クリックストリーム分析データと ADM 分析を使用すると、次のインサイトを導き出して、ユーザーがサイトを放棄した理由を分析できます。

- レイテンシー、5xx エラーが原因でユーザーが放棄されていますか?
- SSL ハンドシェイクエラーはありますか。
- パフォーマンスやネットワークに関連する問題があるアプリケーションの一部はありますか。
- 404 エラーがあるか、ページの読み込み時間が応答するのに永遠にかかる、など。
- 顧客はサーバー応答の異常に直面していますか。

NetScaler Console サービスが提供する Web インサイトにより、IT 管理者は以下の機能を使用して問題を迅速に解決できます:

- NetScaler によって提供されるすべての Web アプリケーションの統合されたリアルタイムの監視を提供します。
- 観測性ツール（グローバルサービスグラフなど）を使用して、時間、レイテンシー、および通常のユーザーの動作に関するアプリケーションのパフォーマンスに関する全体的なビューを取得します。
- インテリジェントな分析を実行して、サーバー応答の異常を理解します。
- SSL Insight は、5xx および 4xx エラーの解決に貢献します。
- 以下を含むすべての Web セッションの記録を維持するには
  - すべてのウェブトランザクションの詳細ログ
  - 関連ログを検索する検索機能
  - ADC・ツー・エンド・ユーザーを隔離する機能 ADC からサーバーへの問題

ADC がクリックストリーム分析用にエクスポートしたデータの種類

NetScaler は、次のようなさまざまな形式のデータを生成するさまざまなソースをキャプチャします。

- Web サーバーログ

  Web サーバロギング機能は、HTTP および HTTPS 要求のログをクライアントシステムに送信して、ストレージと取得を行います。これらのログには膨大な量のデータが含まれており、理解するのが難しく、そこから意味があります。分析ツールは、それを理解し、価値を引き出すのに役立ちます。設定の詳細については、このドキュメントの「**Web** ロギングの設定」セクションを参照してください。

- Syslogs

  syslog の主な用途は、システム管理です。プロアクティブな Syslog モニタリングは、インフラストラクチャ内のサーバーやその他のデバイスのダウンタイムを大幅に削減するため、効果があります。Syslog は、重要なネットワークの問題を特定し、プロアクティブに報告します。

- アクセスログ

  アクセスログには、Web サーバーで発生したイベントに関する情報が格納されます。たとえば、誰かがあなたのウェブサイトを訪問すると、ログが記録され、ウェブサーバー管理者に訪問者の IP アドレス、閲覧していたページ、ステータスコード、使用されたブラウザなどの情報が提供されます。ログを理解するための適切な知識が不足している場合、ログにアクセスするのは圧倒的かもしれません。
  システムを次のものに統合するようにプログラムできます。

  - シームレスな配信のための NetScaler
  - ビジネスに役立つ実用的なインサイトのための Kinesis

- 監査ログ

監査ログ機能を使用すると、カーネルおよびユーザーレベルのデーモン内のさまざまなモジュールによって収集された NetScaler の状態とステータス情報をログに記録できます。

- エラーログ

  エラーログファイルは、管理者が Web サーバーで発生した特定のエラーに関する詳細情報を提供するための補助となります。

クリックストリーム分析用に **NetScaler** ソリューションを構成する

Web サーバーのログ機能を使用すると、HTTP および HTTPS 要求のログをクライアントシステムに送信して、ストレージと取得を行うことができます。

Web サーバーロギング用に NetScaler を構成するには、次のことを行う必要があります。

- Web ログ機能を有効にする
- Web ログサーバーが NetScaler で実行されるため、ログエントリを一時的に保存するようにバッファのサイズを構成します。

CLI を使用して Web サーバロギングを設定するには、次の手順を実行します。

1. Web サーバロギング機能を有効にします。

```
enable ns feature WL
<!--NeedCopy-->
```

2. [オプション] ログに記録された情報を保存するバッファサイズを変更/設定します。

```
set ns weblogparam -bufferSizeMB 60
<!--NeedCopy-->
```

3. NetScaler Web ロギング（NSWL）クライアントをインストールします。詳細については、「NetScaler Web ロギング（NSWL）クライアントのインストール」を参照してください。
4. Windows で NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

   a) nswl_win-.zip< release number > ファイルを展開して、< build number > NSWL クライアントをインストールする Windows システムにパッケージからコピーします。

   b) Windows システムでは、ファイルをディレクトリ (を参照 < NSWL-HOME>) に解凍します。ビン、サンプル、およびその他のディレクトリが抽出されます。

   c) コマンドプロンプトで、< NSWL-HOME >\ bin ディレクトリから次のコマンドを実行します。

```
nswl -install -f < path of the log.conf file >\log.conf
<!--NeedCopy-->
```


> 注:
>
> NSWL クライアントをアンインストールするには、コマンドプロンプトで < NSWL-HOME >\ bin ディレクトリから次のコマンドを実行します。

```
nswl -remove
<!--NeedCopy-->
```

5. NSWL クライアントをインストールしたら、NSWL 実行可能ファイルを使用して NSWL クライアントを構成します。これらの構成は、NSWL クライアント構成ファイル (log.conf) に保存されます。

   NSWL 実行可能ファイルが配置されているディレクトリから次のコマンドを実行します。

```
\ns\bin
<!--NeedCopy-->
```

6. NSWL クライアント構成ファイル（log.conf）で、クライアントシステムのコマンドプロンプトで次のコマンドを実行して、NSWL クライアントがログを収集する NetScaler IP アドレス（NSIP）を追加します。

```
nswl -addns -f < Path to the configuration(log.conf) file >\log.
    conf
<!--NeedCopy-->
```

7. NetScaler アプライアンスの NSIP（IP アドレス）、ユーザー名、nsroot およびパスワードを「インスタンス ID/設定したパスワード」として入力します。

   - NSWL クライアントは、NetScaler IP アドレス（NSIP）を NSWL 構成ファイルに追加した後、ADC に接続します。
   - ADC は、HTTP および HTTPS 要求ログエントリをクライアントに送信する前にバッファリングします。
   - クライアントは、エントリを保存する前に（log.conf ファイルを変更して）エントリをフィルタリングできます。

> 注
>
> NetScaler デフォルトパスワードを変更し、構成を続行します。次のコマンドを入力して、パスワードを変更します。

```
set system user nsroot -password <your password>
<!--NeedCopy-->
```

Amazon Kinesis エージェントの設定

AWS ウェブコンソールで次の手順を実行します。Amazon Kinesis エージェントを設定します。

1. 設定ファイル (appsettings.json) を作成してデプロイします。構成ファイルは、ソースをシンクに接続するソース、シンク、およびパイプのセットと、オプションの変換を定義します。

次の例は、Windows アプリケーションログイベントを Kinesis Data Firehose にストリーミングするように Kinesis `appsettings.json` エージェントを設定する完全な設定ファイルです。

```
{

"Sources": [
    {

        "Id": "NSWLog",
        "SourceType": "DirectorySource",
        "Directory": "C:\Users\Administrator\Downloads\nswl_win
           -13.0-52.24\bin",
        "FileNameFilter": "*.log"
        "RecordParser": "TimeStamp",
        "TimestampFormat": "yyyy-MM-dddd HH:mm:ss.ffff", //
           Optional parameter required only by the timestamp
           record parser
        "TimeZoneKind": "UTC", //Local or UTC
        "SkipLines": 0 //Skip a number of lines at the beginning
           of each file
     }

    ],
"Sinks": [
    {

        "Id": "ApplicationLogKinesisFirehoseSink",
        "SinkType": "KinesisFirehose",
        "StreamName": "Delivery-ik-logs",
        "AccessKey": "Your Access Key",
        "SecretKey": "YourSecretKey",
        "Region": "ap-south-1"
     }

    ],
"Pipes": [
    {

    "Id": "ApplicationLogSourceToApplicationLogKinesisFirehoseSink
       ",
    "SourceRef": "ApplicationLogSource",
    "SinkRef": "ApplicationLogKinesisFirehoseSink"
     }

    ],
"Telemetrics":
    {

    "off": "true"
     }

 }

```



```
<!--NeedCopy-->
```

2. データソースに KinesisAgent をセットアップしてデータを収集し、AmazonKinesis に継続的に送信します Firehose/Kinesis データ分析。詳細については、「Microsoft Windows 用 Amazon Kinesis エージェントの使用開始」を参照してください。
3. Amazon Kinesis Firehoseを使用して、エンドツーエンドのデータ配信ストリームを作成します。配信ストリームは、エージェントから宛先にデータを送信します。送信先には、Amazon Kinesis Analytics、Amazon Redshift、Amazon Elasticsearch サービス、Amazon S3 が含まれます。[ソース] で、[直接 **PUt** またはその他のソース] を選択して Kinesis Data Firehose 配信ストリームを作成します。
4. Amazon Kinesis アナリティクスで SQL クエリを使用して受信ログデータを処理します。
5. 処理されたデータを Kinesis アナリティクスから Amazon Elasticsearch サービスにロードして、データのインデックスを作成します。
6. Kibana や AWS QuickInsight Services などの可視化ツールを使用して、処理されたデータを分析し、視覚化します。

参照ドキュメント

- Syslog メッセージの表示とエクスポート
- ハイブリッドマルチクラウド向け Citrix Networking
- Kinesis エージェントを使用した AWK Kinesis データストリームへの書き込み

Microsoft Windows Azure パックと **Cisco ACI** によって管理されるプライベートクラウド内の **NetScaler**

April 15, 2024

NetScaler アプライアンスは、Microsoft Windows Azure Pack を介して管理されるプライベートクラウドでの負荷分散に使用できます。プライベートクラウドのネットワークは、Cisco ACI と NetScaler を使用して自動化されています。

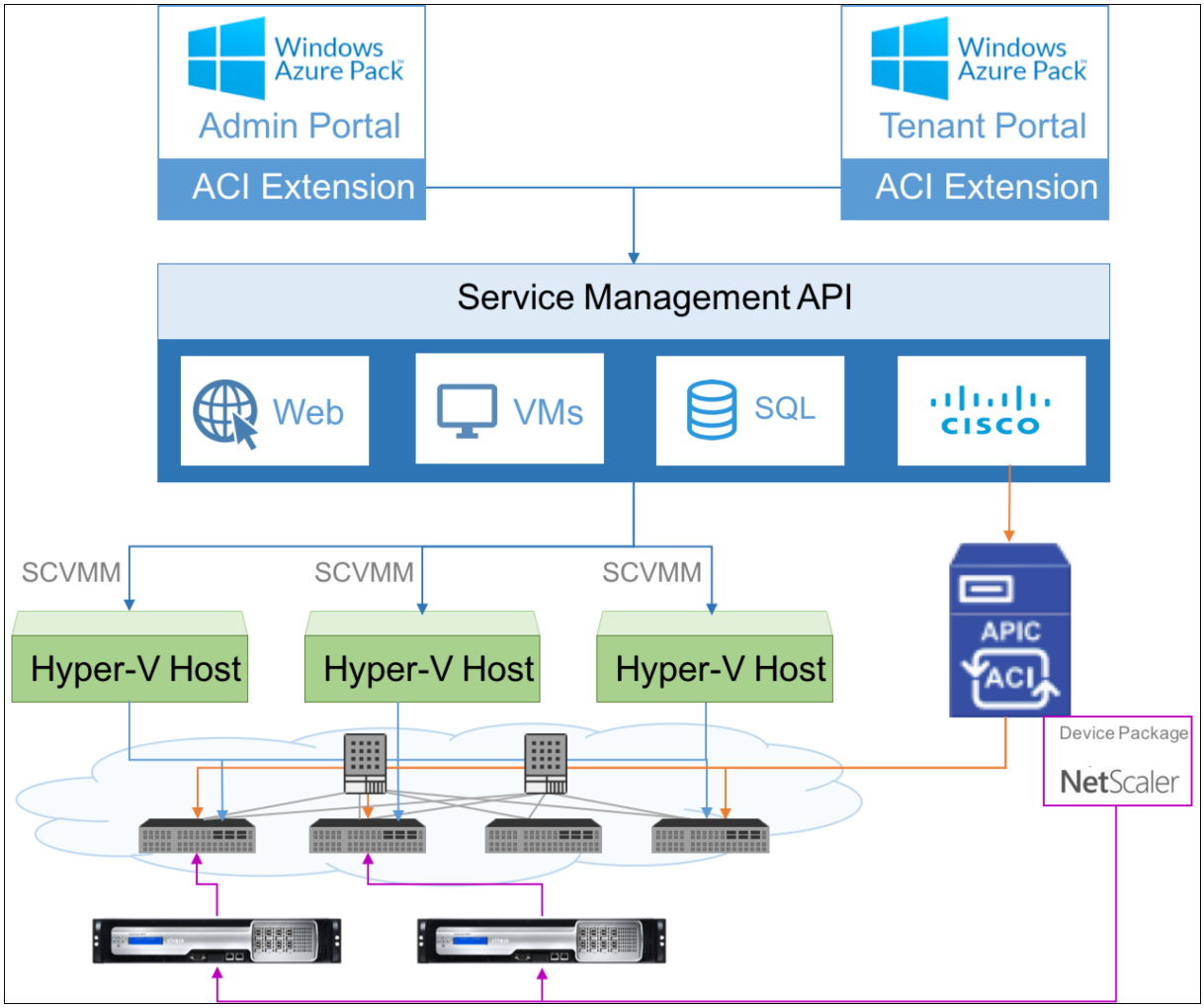
このソリューションには、Windows Azure Pack（WAP）から Cisco APIC、Cisco APIC からシステムセンター仮想マシンマネージャー（SCVMM）、Cisco APIC から NetScaler への統合など、多くの統合ポイントが含まれます。プライベートクラウドのテナントとして、NAT を有効にしたり、ネットワークサービスをプロビジョニングしたり、ロードバランサーを追加したりできます。

WAP はテナントポータルと管理者ポータルをサポートしており、管理者は ACI 登録、VIP 範囲、NetScaler デバイスと仮想マシンクラウドの関連付け、テナントユーザーアカウントの作成などの管理タスクを実行できます。テナン

トは WAP テナントポータルにログオンし、ネットワーク、ブリッジドメイン、仮想ルーティングと転送（VRF）を設定し、NetScaler の負荷分散と RNAT 機能を利用できます。

> 重要
>
> - このソリューションでは、NetScaler アプライアンスは基本的な負荷分散のみを提供します。
> - テナントは同じネットワークに異なるポートを持つ複数の VIP アドレスを配置できますが、IP とポートの組み合わせが一意であることを確認する必要があります。
> - NetScaler デバイスパッケージは、単一コンテキスト展開のみをサポートします。各テナントは専用の NetScaler インスタンスを取得します。
> - WAP は、NetScaler SDX プラットフォームにデプロイされた NetScaler VPX インスタンスを含む、NetScaler MPX アプライアンスと NetScaler VPX 仮想アプライアンスをサポートします。

次の図は、ソリューションの概要を示しています。



前提条件

以下の点について確認してください:

- Cisco ACI コンポーネントと NetScaler に関する概念的な知識がある。
  - Cisco ACI とそのコンポーネントの詳細については、次の URL にある製品マニュアルを参照してください。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html。
  - NetScaler の詳細については、次の URL にある NetScaler 製品ドキュメントを参照してください。https://docs.netscaler.com/。
- データセンターの Cisco APIC を含む Cisco ACI のすべての必須コンポーネントがセットアップされ、設定されます。Cisco ACI とそのコンポーネントの詳細については、次の URL にある製品マニュアルを参照してください。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html。
- Cisco ACI を Microsoft Windows Azure Pack と統合する方法を知っています。次の製品ドキュメントを参照してください http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/2-x/virtualization/b_ACI_Virtualization_Guide_2_2_1.html。
- Microsoft Windows Azure パックの概念的な知識がある。次の製品ドキュメントを参照してください https://www.microsoft.com/en-in/cloud-platform/windows-azure-pack。
- NetScaler ソフトウェアリリース 11.1 以降がインストールされている。
- NetScaler は Cisco ACI で設定して、Cisco APIC を使用して管理できるようにします。
- Cisco APIC から、次のことを確認してください。
  - Cisco APIC と NetScaler の管理接続が確立されました。
  - *NetScaler* デバイスパッケージバージョン *11.1*～*52.3* をアップロードし、Cisco APIC を使用して NetScaler デバイスを Cisco ACI に登録します。
  - NetScaler アプライアンスを Cisco APIC の共通テナントで構成し、Cisco APIC に障害がないことを確認します。
  - VLAN プール、L3OutServicesDOM、L3extOut、リソースプールなど、APIC 固有の構成をすべて設定しました。詳細については、*Cisco* のマニュアルを参照してください。

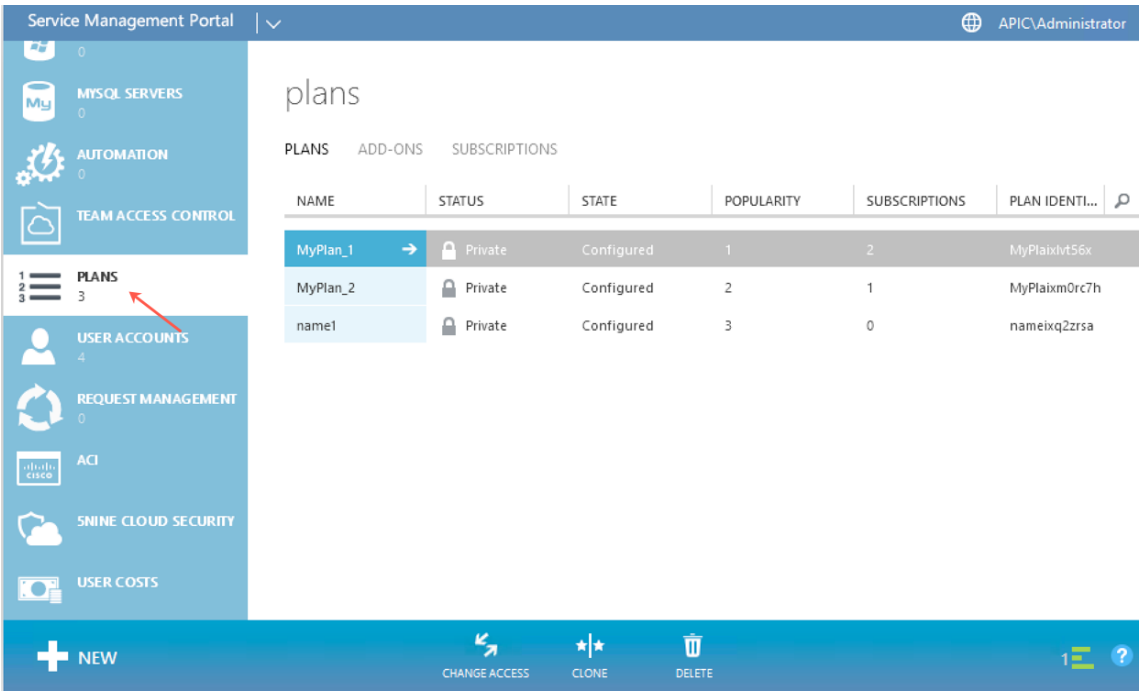
サービス管理ポータル（管理者ポータル）のプランでの **NetScaler ADC** ロードバランサーの作成

August 15, 2023

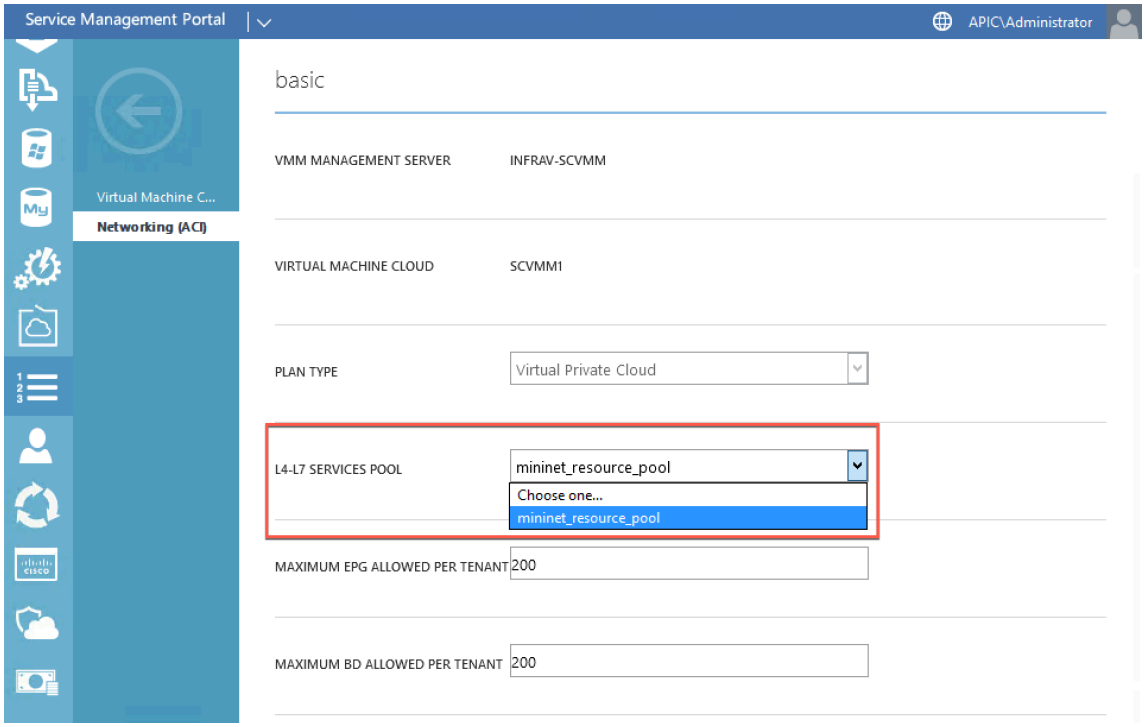
WAP のサービス管理ポータルでは、管理者は Cisco APIC を WAP に登録したり、ホスティングプランを作成したりできます。プランの一部として、VIP 範囲を指定したり、NetScaler ロードバランサーをプランに関連付けたり、テナントユーザーアカウントを作成したりできます。

管理ポータルのプランに **NetScaler** ロードバランサーを作成するには:

1. サービス管理ポータル (管理者ポータル) にログインします。
2. ナビゲーションペインで「PLANS」を選択します。



3. プランペインで、ロードバランサーを追加するプランを選択します。
4. 選択したプランのペインで、[ネットワーク **(ACI)**] を選択します。
5. [ネットワーク（**ACI**）] ペインの [**L4-L7** サービスプール] ドロップダウンリストで、Cisco APIC で作成した L4-L7 リソースプールを選択します。



6. テナントユーザーアカウントを作成し、作成したプランにユーザーを関連付けます。

サービス管理ポータル（テナントポータル）を使用して **NetScaler** ロードバランサーを構成する

August 15, 2023

WAP では、テナントがブリッジドメイン（BD）、VRF、およびネットワークを作成すると、テナントはサービス管理ポータル（テナントポータル）を通じて NetScaler ロードバランサーを構成できます。

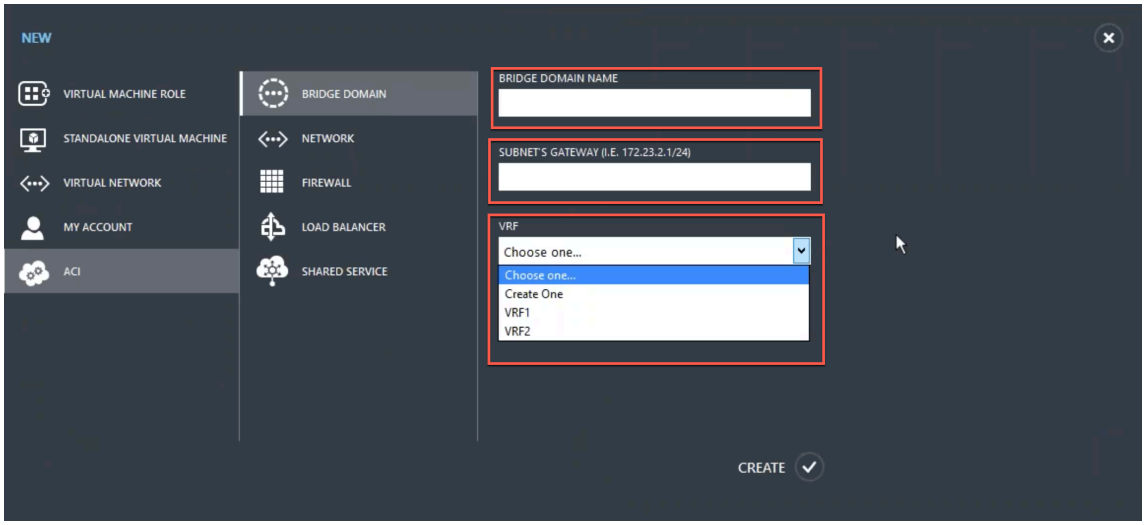
サービス管理ポータル（テナントポータル）で NetScaler ロードバランサーを構成するには

1. サービス管理ポータル (テナントポータル) にログインします。
2. 次のように、ブリッジドメインと VRF を作成します。

   a. ナビゲーションペインで **ACI** を選択します。

   b.「新規」をクリックします。

   c. 新しいペインで、ブリッジドメインを選択します。



d.「ブリッジドメイン」フィールドに、ブリッジドメイン名 (たとえば、BD01) を入力します。

e. (オプション)「サブネットのゲートウェイ」フィールドに、サブネットのゲートウェイを入力します (たとえば、192.168.1.1/24)。

f. **VRF** フィールドで、すでにサブスクリプションに含まれている VRF を選択するか、**Create One** を選択して VRF を作成します。

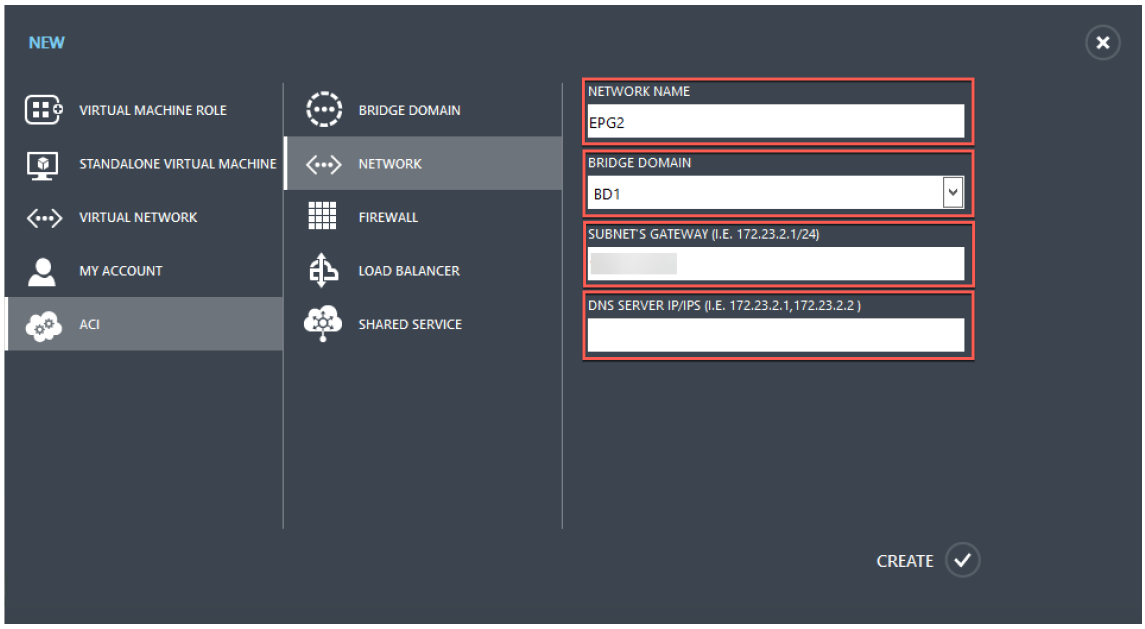
g.［**CREATE**］をクリックします。

3. ネットワークを作成し、作成したブリッジドメインに関連付けます。以下を実行します:

a. ナビゲーションペインで **ACI** を選択します。

b.「新規」をクリックします。

c.「新規」ペインで、「ネットワーク」を選択します。



d.「ネットワーク名」フィールドに、ネットワーク名 (たとえば、S01) を入力します。

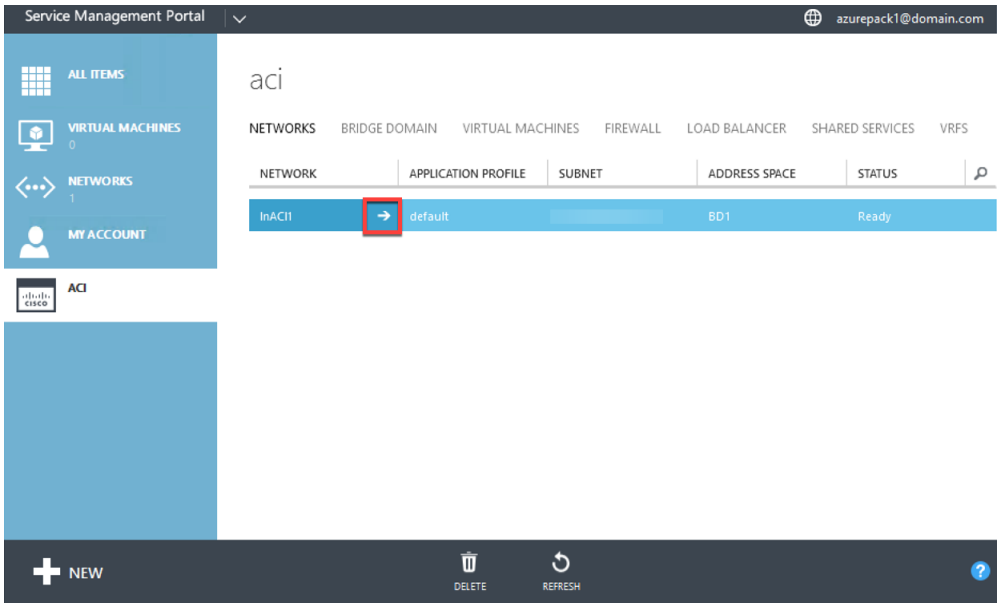
e. **BRIDGE** ドメインドロップダウンリストで、作成したブリッジドメインを選択します。(たとえば、BD01)。

f. サブネットの **GATEWAY** フィールドに、サブネットのゲートウェイアドレス (172.23.2.1/24 など) を入力します。

g. （オプション）「**DNS** サーバー **IP/IPS** 」フィールドに、DNS サーバーの詳細を入力します。

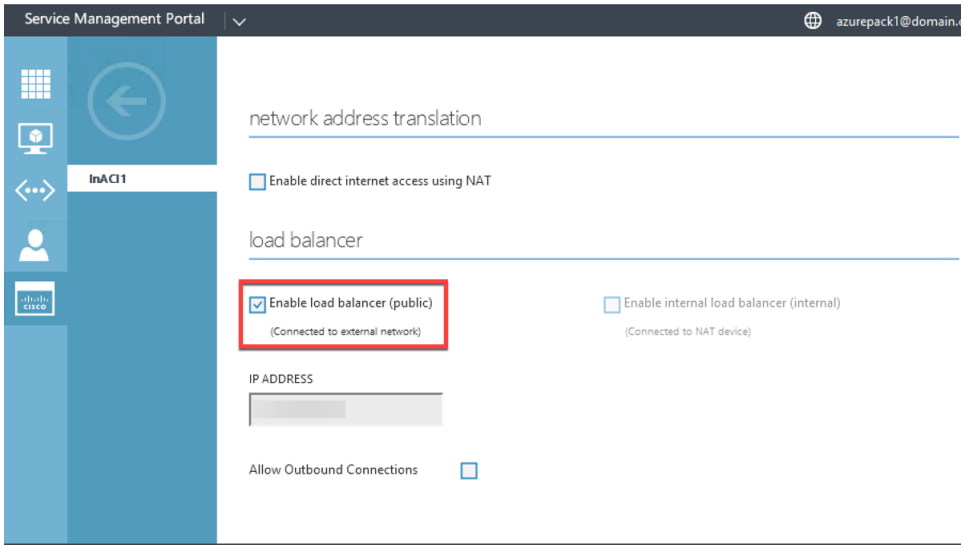
h.［**CREATE**］をクリックします。

4. **ACI** ペインで、「ネットワーク」を選択します。

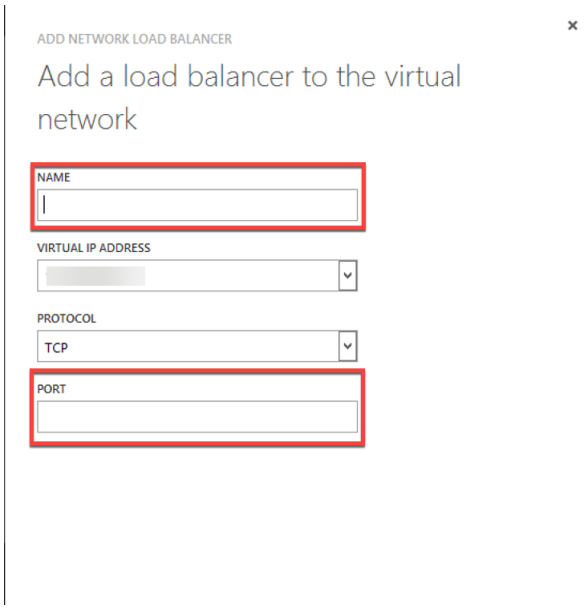


5. 作成したネットワークをダブルクリックします。次に、ネットワークペインで [ロードバランサーを有効にする **(**パブリック**)**] を選択します。[**IP** アドレス] フィールドでは、管理者が管理者ポータルで設定した VIP 範囲から VIP が自動的に割り当てられます。詳細については、サービス管理ポータル（管理者ポータル）の「プランでの NetScaler ADC ロードバランサーの作成」を参照してください。

6. 作成したネットワークをダブルクリックします。次に、ネットワークペインで [ロードバランサーを有効にする **(**パブリック**)**] を選択します。[**IP** アドレス] フィールドでは、管理者が管理者ポータルで設定した VIP 範囲から VIP が自動的に割り当てられます。詳細については、サービス管理ポータル（管理者ポータル）の「プランでの NetScaler ADC ロードバランサーの作成」を参照してください。



7. ネットワークペインで「ロードバランサー」タブを選択し、「追加」をクリックします。



8. [ネットワークロードバランサーの追加] ペインで、次の操作を行います。

   a. **NAME** フィールドに、ロードバランサーの名前を入力します。

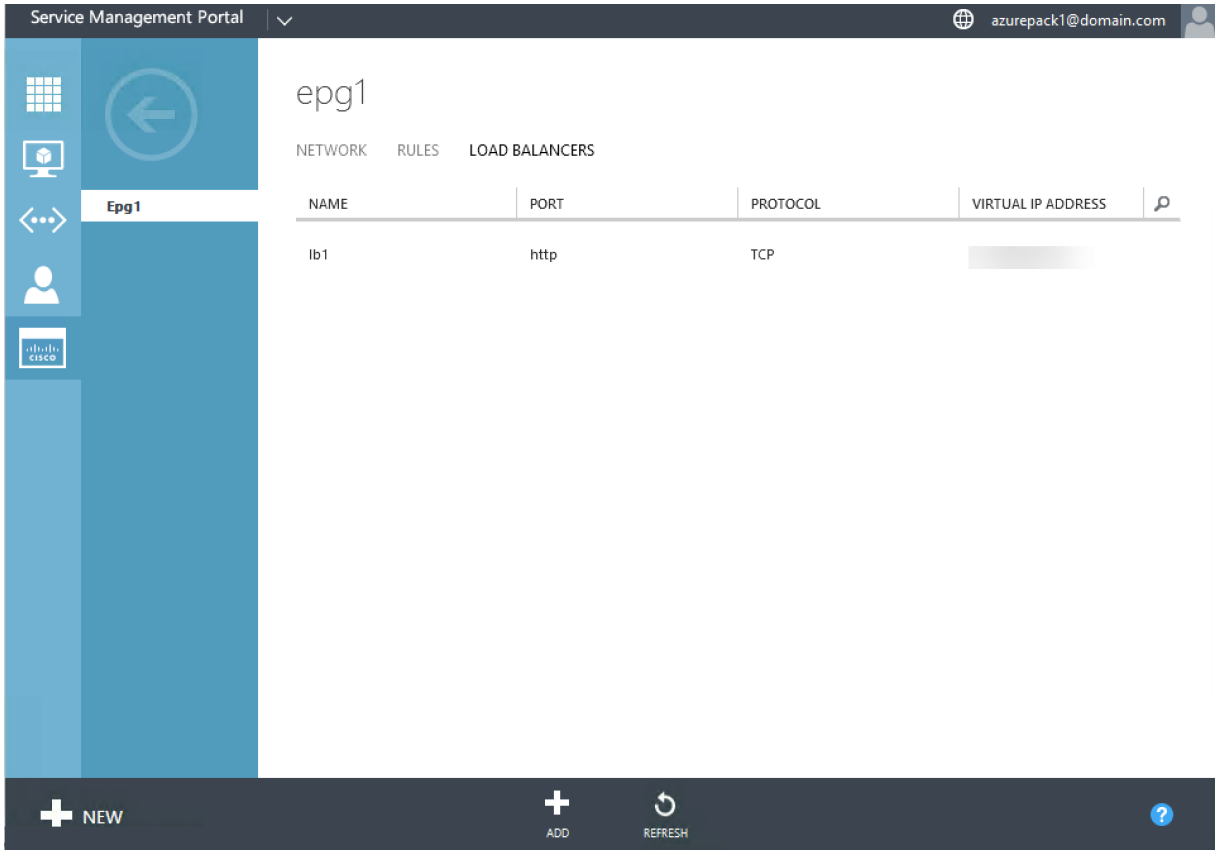
   b. オプションで、**VIRTUAL IP ADDRESS** フィールドで、以前に定義した VIP 範囲の VIP アドレスをロードバランサーに割り当てます。

   c. オプションで、「プロトコル」フィールドで「**TCP**」を選択します。

   d.「**PORT** 」フィールドに、ポート番号を入力します。

9. [**CREATE**] をクリックします。

   NetScaler ロードバランサーは [ロードバランサー] タブに表示され、**NetScaler** ロードバランサーはデータパス対応です。



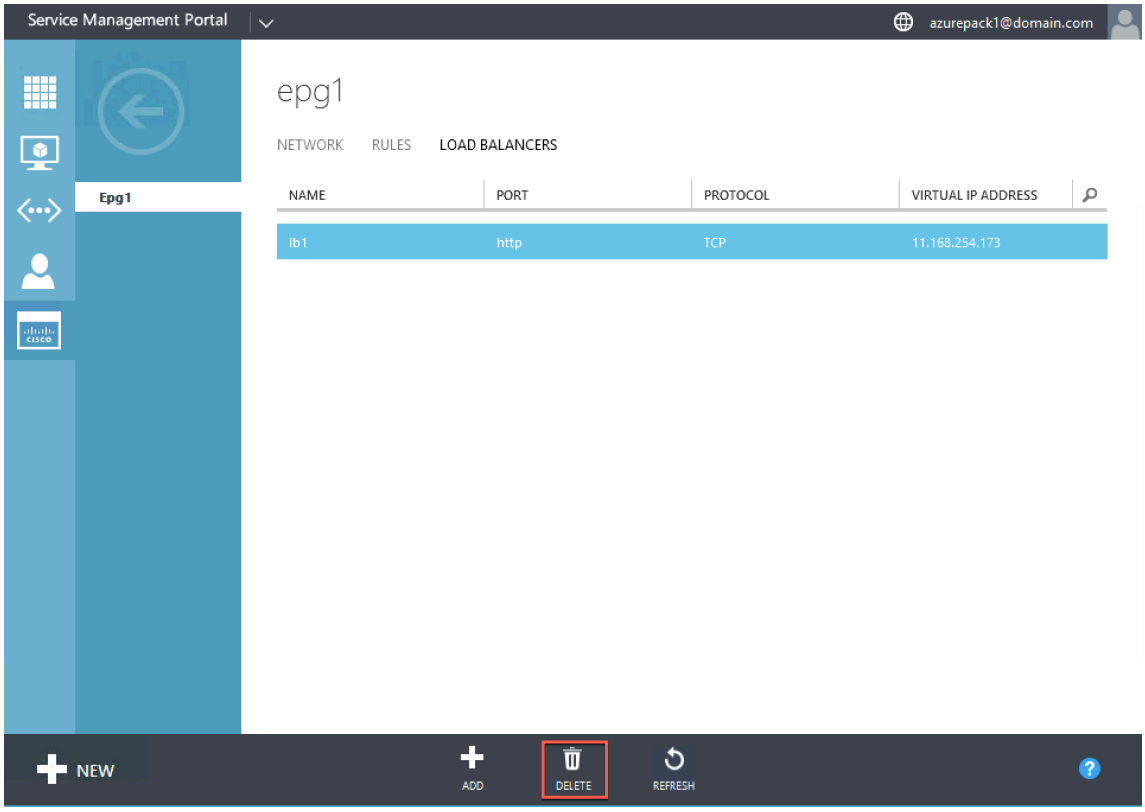
ネットワークからの **NetScaler ADC** ロードバランサーの削除

August 15, 2023

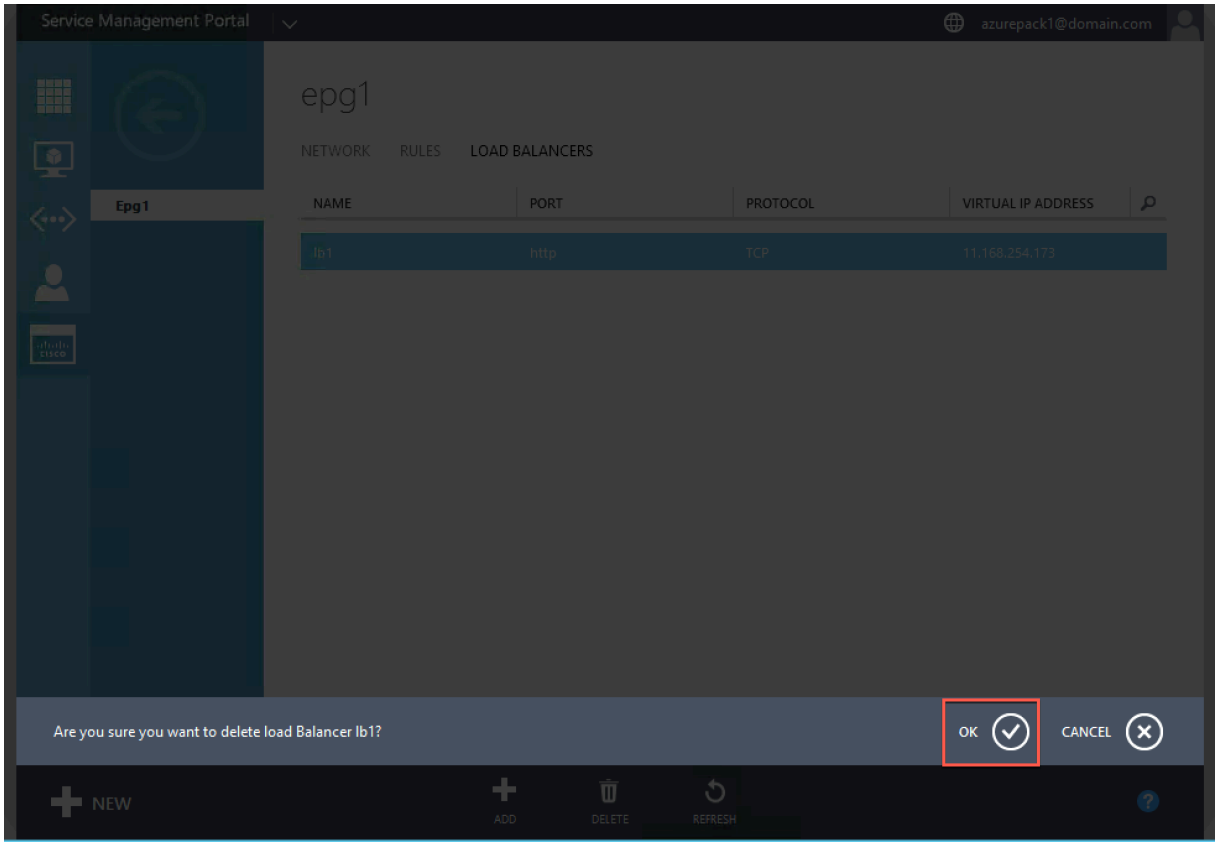
サービス管理ポータル（テナントポータル）を使用して、作成した NetScaler ロードバランサーをネットワークから削除できます。

NetScaler ロードバランサーをネットワークから削除するには:

1. サービス管理ポータル (テナントポータル) にログインします。
2. ナビゲーションペインで **ACI** を選択します。
3. **ACI** ペインの [ネットワーク] タブで、作成したネットワークをクリックします。
4. 選択したネットワークのペインで、**NetScaler** ロードバランサーを選択し、**[削除]** をクリックします。



5. **OK** をクリックして NetScaler ロードバランサーを削除します。



Kubernetes ベースのマイクロサービス向け **NetScaler** クラウドネイティブソリューション

March 20, 2024

企業は、イノベーションのスピードを上げ、お客様により近づくことができるように変化するために、社内プロセスを再設計し、組織内の境界を取り払っています。こうした企業は、同じチーム内の適切なスキルセットを統合するために、縦割り構造を取り除いています。目標の 1 つは、スピード、俊敏性、効率性を備えたソフトウェアアプリケーションを作成して提供することです。こうした理由で、マイクロサービスに基づく最新のアプリケーションアーキテクチャが、さらに多くの企業に採用されるようになっています。
マイクロサービスアーキテクチャを使用すると、個別に展開、更新、スケーリングできる疎結合のサービスのセットとして、アプリケーションを作成できます。

クラウドネイティブは、次の主要な属性を持つアプリケーションを構築および展開するために、マイクロサービスアーキテクチャに依存するアプローチです:

- 疎結合されたマイクロサービスまたはコンテナとしてアプリケーションを展開する
- 非常に高度な自動化を伴う
- アジャイル DevOps プロセスと継続的デリバリーワークフローを実装する

- 相互作用とコラボレーションのための API を中心に置く

Kubernetes はクラウドネイティブのプロセスでどのように役立ちますか?

必要なレベルの俊敏性と安定性を提供するために、クラウドネイティブアプリケーションには、高レベルのインフラストラクチャの自動化、セキュリティ、ネットワーキング、および監視が必要です。コンテナを大規模に効率的に管理できるコンテナオーケストレーションシステムが必要です。Kubernetes は、コンテナのデプロイとオーケストレーションの最も一般的なプラットフォームとして登場しました。Kubernetes は、開発者やオペレーターからコンテナーを実行、デプロイ、管理するという複雑なタスクを抽象化し、ノードのクラスター間でコンテナーを自動的にスケジュールします。Kubernetes とクラウドネイティブコンピューティングファンデーション（CNCF）エコシステムは、クラウドネイティブソリューションのプラットフォームを構築するのに役立ちます。

Kubernetes を使用するいくつかの主なメリット:

- オンプレミス、ハイブリッド、またはパブリッククラウドインフラストラクチャのアプリケーション展開を簡素化します
- アプリケーションの開発と展開を加速します
- アプリケーションの俊敏性、柔軟性、およびスケーラビリティを向上させます

NetScaler クラウドネイティブソリューションとは何ですか?

実稼働環境で Kubernetes を使用するメリットを最大化するには、Kubernetes をいくつかのツール、ベンダー提供およびオープンソースのコンポーネントと統合する必要があります。クラウドネイティブアプリケーションで実稼働環境レベルの信頼性とセキュリティを確保することは、多くの組織が直面している課題です。
業界をリードする NetScaler のプロバイダーである NetScaler は、Kubernetes の本番環境における課題に対処するための NetScaler クラウドネイティブソリューションを提供しています。

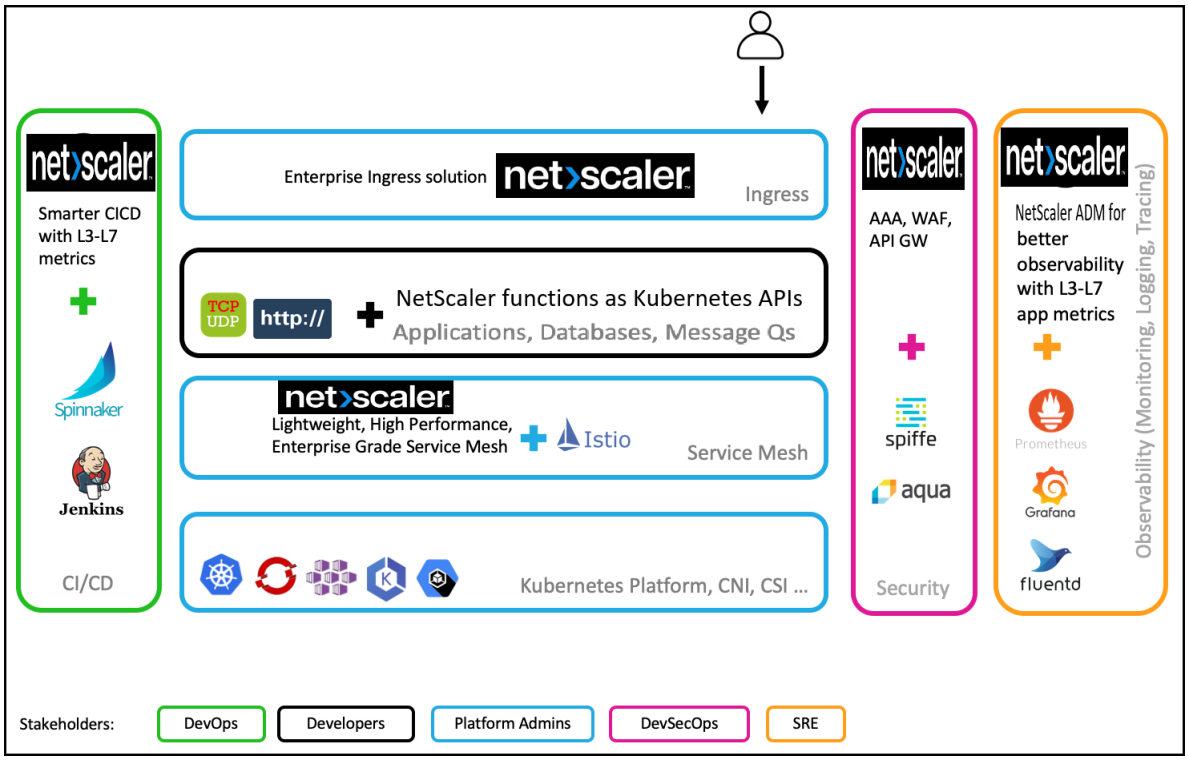
NetScaler のクラウドネイティブソリューションは、NetScalers の高度なトラフィック管理、オブザーバビリティ、および包括的なセキュリティ機能を活用して、エンタープライズグレードの信頼性とセキュリティを確保します。Kubernetes 環境のアプリケーショントラフィックを完全に可視化し、即座にフィードバックを提供し、アプリケーションのパフォーマンスに関して有意義な詳細情報を得るのに役立ちます。

次の表に、Ingress ソリューションを実装する際の、さまざまな利害関係者にとっての主な要件を示します。

利害関係者	職務権限	ニーズ
プラットフォーム管理者	Kubernetes クラスタの可用性を確保	複数のクラスタ、運用、およびプラットフォームのライフサイクル管理のために展開されたアプリケーションを管理するための、より簡単な方法

利害関係者	職務権限	ニーズ
DevOps	アプリケーションの実稼働環境への展開を加速	CI/CD パイプラインとの統合、より高速な展開のためのカナリアやブルーグリーンなどの展開手法のサポート
開発者	マイクロサービスの開発とテスト	トラフィックを Kubernetes クラスタに誘導するため、トレースとデバッグ、アプリケーションのレート制限、アプリケーションの認証
SRE	サービスレベルアグリーメントを満たすためのアプリケーションの可用性を確保	アプリケーションとインフラストラクチャ向けの高度な利用統計情報
SecOP	セキュリティコンプライアンスの確保	セキュアな Ingress トラフィック、API 保護、Kubernetes クラスタ内のマイクロサービス間におけるセキュアな通信のためのサービスメッシュ

次の図は、NetScaler クラウドネイティブソリューションと、クラウドネイティブへの移行において利害関係者が直面するさまざまな課題にどのように対処するかを説明しています。



NetScaler クラウドネイティブソリューションには、次のような主なメリットがあります。

- 開発者、SRE、devOps、ネットワークまたはクラスタ管理者のニーズに応える、高度な Kubernetes Ingress ソリューションを提供します。
- TCP または UDP トラフィックに基づいてレガシーアプリケーションを Kubernetes 環境に移動するときに、それらを書き直す必要がなくなります。
- Kubernetes API として公開されている NetScaler ポリシーでアプリケーションを保護します。
- North-Souch トラフィックと East-West トラフィックに、高性能のマイクロサービスを展開するのに役立ちます。
- NetScaler Console サービスグラフを使用して、すべてのマイクロサービスのオールインワンビューを提供します。
- TCP、UDP、HTTP、HTTPS、SSL などのさまざまな種類のトラフィックにわたるマイクロサービスを、すばやくトラブルシューティングできます。
- API を保護します。
- カナリア展開で CI/CD パイプラインを自動化します。
- CNCF オープンソースツールとの、すぐに使える統合を提供します。

Citrix が提供するさまざまなクラウドネイティブソリューションの詳細については、次のリンクを参照してください。

- Kubernetes Ingress ソリューション
- サービスメッシュ
- 監視性のためのソリューション
- Kubernetes での API ゲートウェイ

NetScaler クラウドネイティブソリューションのコンポーネント

次の表は、NetScaler クラウドネイティブソリューションの主なコンポーネントを説明しています。

コンポーネント	説明
NetScaler Ingress Controller	このコンテナは Kubernetes Ingress Controller の実装であり、NetScaler（NetScaler CPX、VPX、または MPX）を使用してトラフィックを管理し、Kubernetes クラスターにルーティングします。NetScaler Ingress Controller を使用すると、イングレスルールに従って NetScaler CPX、VPX、または MPX を構成し、お使いの NetScaler を Kubernetes 環境に統合できます。

コンポーネント	説明
NetScaler オブザーバビリティエクスポーター	NetScaler Observability Exporter は、NetScaler からメトリックとトランザクションを収集し、サポートされているエンドポイントに適した形式（JSON、AVRO など）に変換するコンテナです。NetScaler オブザーバビリティエクスポーターによって収集されたデータを目的のエンドポイントにエクスポートできます。エンドポイントにエクスポートされたデータを分析することで、NetScalers がプロキシするアプリケーションのマイクロサービスレベルで貴重な洞察を得ることができます。
NetScaler xDS アダプター	NetScaler xDS アダプターは、NetScaler を xDS API（Istio、Consul など）に基づくサービスメッシュコントロールプレーン実装と統合するためのコンテナです。サービスメッシュコントロールプレーンと通信し、コントロールプレーン API サーバに対する gRPC クライアントとして動作することによって更新をリッスンします。NetScaler xDS-adapter は、コントロールプレーンからの更新に基づいて、同等の NetScaler 構成を生成します。
NetScaler CPX	NetScaler CPX は、Docker ホストでプロビジョニングできるコンテナベースのアプリケーション配信コントローラーです。NetScaler CPX を使用することにより、Docker エンジン機能を利用し、NetScaler の負荷分散機能とトラフィック管理機能を、コンテナベースのアプリケーション向けに活用できます。1 つまたは複数の NetScaler CPX インスタンスを、スタンドアロンインスタンスとして Docker ホストで展開できます。

Kubernetes Ingress ソリューション

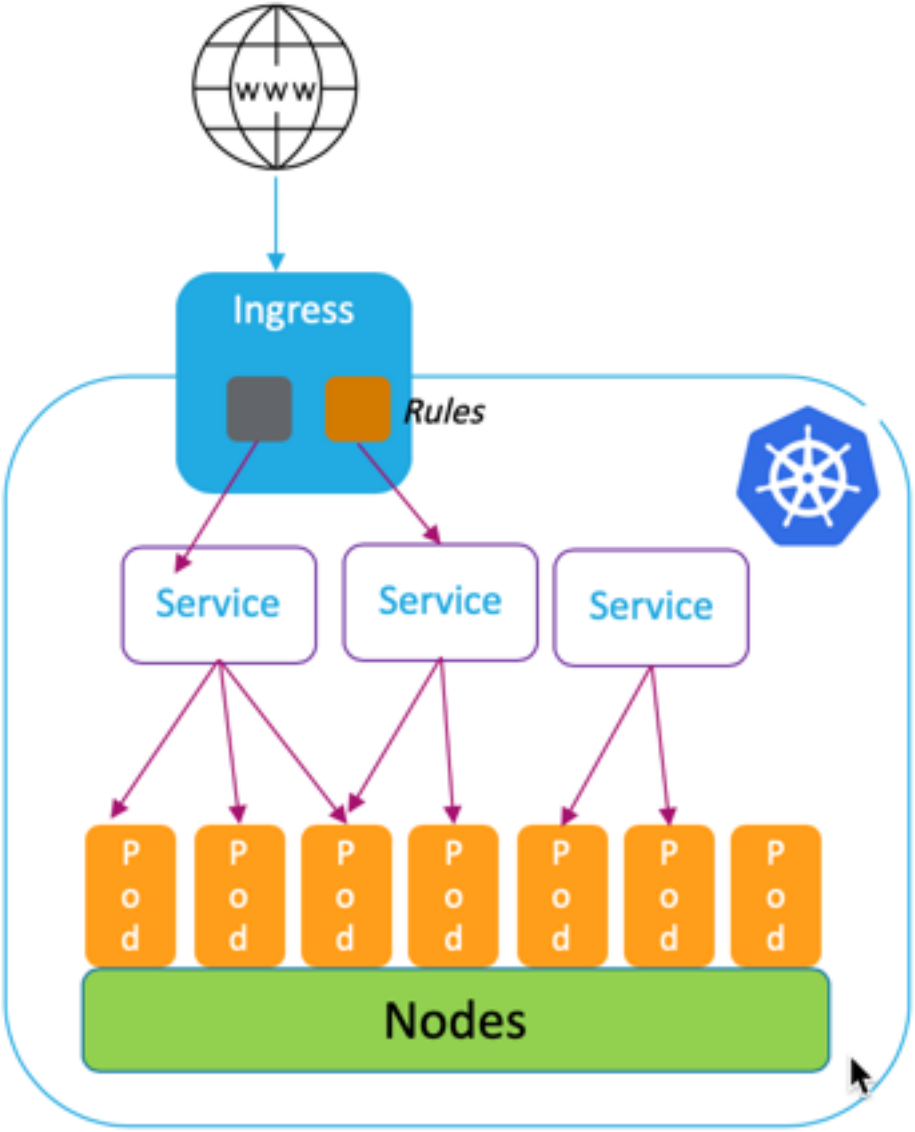
August 15, 2023

このトピックでは、NetScaler が提供する Kubernetes Ingress ソリューションの概要と、その利点について説明します。

Kubernetes イングレスとは何ですか**?**

Kubernetes クラスタ内でアプリケーションを実行する場合、外部ユーザーが Kubernetes クラスタの外部からアプリケーションにアクセスできるようにする必要があります。Kubernetes には、安定した IP アドレスを使用して複数のサービスを公開する最も効果的な方法を提供する Ingress というオブジェクトが用意されています。Kubernetes Ingress オブジェクトは常に 1 つ以上のサービスと関連付けられ、外部ユーザーがクラスター内で実行されているサービスにアクセスするための単一エントリポイントとして機能します。

次の図は、Kubernetes イングレスの仕組みを説明しています。



Kubernetes Ingress の実装は、以下のコンポーネントで構成されています。

- 入力リソース。Ingress リソースでは、クラスター外からアプリケーションにアクセスするためのルールを定

義できます。

- 入力コントローラー。Ingress Controller は、Ingress で定義されているルールを解釈するクラスター内にデプロイされたアプリケーションです。Ingress Controller は、Ingress ルールをクラスタと統合した負荷分散アプリケーションの構成指示に変換します。ロードバランサーは、Kubernetes クラスター内で実行されるソフトウェアアプリケーション、またはクラスター外で実行されるハードウェアアプライアンスです。
- **Ingress** デバイス。入力デバイスは、NetScaler CPX、VPX、MPX などの負荷分散アプリケーションで、入力 Controller が提供する構成指示に従って負荷分散を実行します。

Citrix の Kubernetes Ingress ソリューションは何ですか

このソリューションでは、NetScaler（NetScaler CPX、VPX、または MPX）を使用してトラフィックを管理および Kubernetes クラスターにルーティングするための Kubernetes Ingress コントローラーの実装が NetScaler によって提供されます。NetScalerIngress Controller は、NetScaler を Kubernetes 環境に統合し、イングレスルールに従って NetScaler CPX、VPX、または MPX を構成します。

標準的な Kubernetes Ingress ソリューションは、レイヤ 7（HTTP または HTTPS トラフィック）でのみ負荷分散を提供します。場合によっては、TCP、UDP、またはアプリケーションに依存する多くのレガシーアプリケーションを公開し、それらのアプリケーションの負荷分散の方法が必要になる場合があります。NetScalerIngress Controller ソリューションは、標準の HTTP または HTTPS イングレスとは別に、TCP、TCP-SSL、および UDP トラフィックをサポートします。また、複数のクラウドやオンプレミスのデータセンターにまたがってシームレスに動作します。

NetScaler ADC は、書き換えポリシーやレスポンダーポリシーなどのエンタープライズグレードのトラフィック管理ポリシーを提供し、レイヤー 7 でトラフィックの負荷を効率的に分散します。ただし、Kubernetes Ingress にはこのようなエンタープライズグレードのトラフィック管理ポリシーがありません。Citrix の Kubernetes Ingress ソリューションを使用すると、NetScaler が提供する CRD を使用して、Kubernetes 環境のアプリケーショントラフィックにリライトポリシーとレスポンダーポリシーを適用できます。

Citrix の Kubernetes Ingress ソリューションは、CI/CD アプリケーションパイプラインの自動カナリアデプロイもサポートしています。このソリューションでは、は Spinnaker プラットフォームと統合され、Kayenta を使用して Canary 展開を分析するための正確なメトリックを提供するソースとして機能します。メトリクスを分析した後、Kayenta はカナリアの集計スコアを生成し、カナリアバージョンを昇格または失敗することを決定します。NetScaler ADC ポリシーインフラストラクチャを使用して、Canary バージョンへのトラフィック配信を規制することもできます。

次の表は、Citrix の Ingress ソリューションが Kubernetes Ingress よりも優れている点をまとめたものです。

Features	Kubernetes Ingress	Citrix の Ingress ソリューション
HTTP と HTTPS のサポート	はい	はい
URL ルーティング	はい	はい
TLS	はい	はい

Features	Kubernetes Ingress	Citrix の Ingress ソリューション
負荷分散	はい	はい
TCP、TCP-SSL	いいえ	はい
UDP	いいえ	はい
HTTP/2	はい	はい
CI/CD ツールによる自動カナリア導入サポート	いいえ	はい
NetScaler の書き換えポリシーとレスポンダーポリシーの適用のサポート	いいえ	はい
認証 (オープンオーソライゼーション (OAuth)、相互認証 TLS (mTLS))	いいえ	はい
Citrix レート制限ポリシーの適用のサポート	いいえ	はい

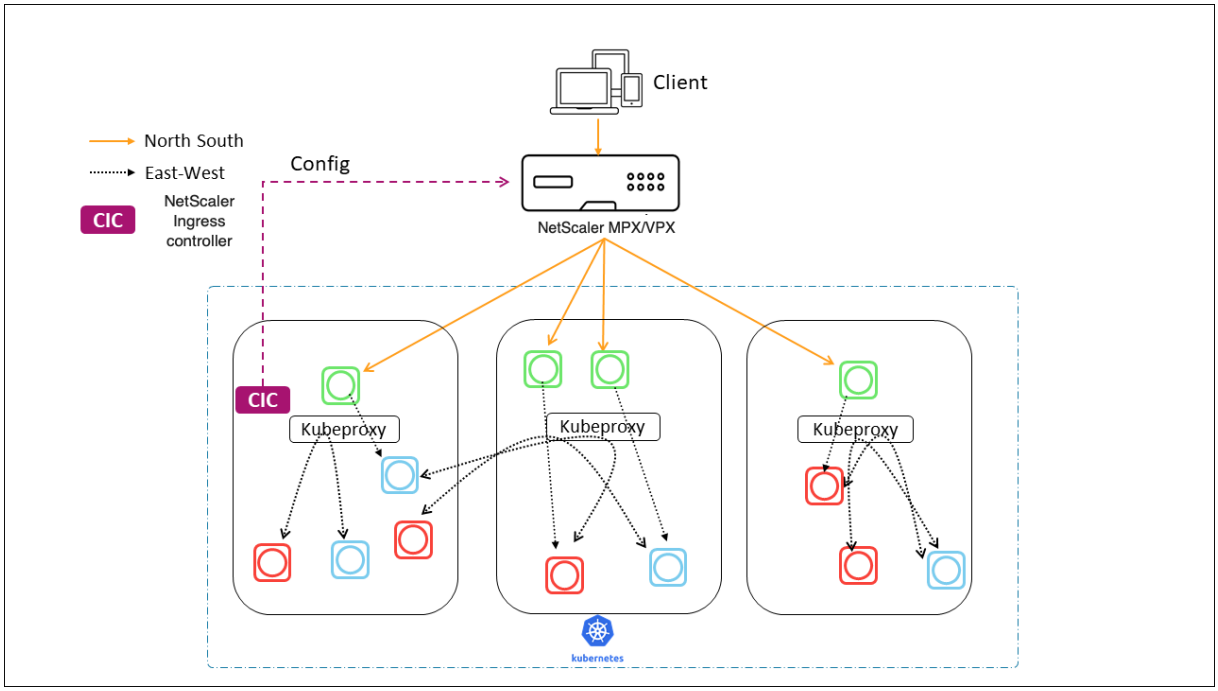
Kubernetes Ingress ソリューションのデプロイオプション

NetScaler の Kubernetes Ingress ソリューションでは、NetScaler と Kubernetes 環境をどのように管理したいかに応じて、柔軟なアーキテクチャを実現できます。

統合入力（単一層）

統合されたイングレス（単層）アーキテクチャでは、Kubernetes クラスターの外部にデプロイされた NetScaler MPX または VPX デバイスは、NetScaler Ingress Controller を使用して Kubernetes 環境と統合されます。NetScaler Ingress Controller は Kubernetes クラスターにポッドとしてデプロイされ、マイクロサービスまたは Ingress リソースの変更に基づいて NetScaler の構成を自動化します。NetScaler デバイスは、インバウンドトラフィックに対して負荷分散、TLS ターミネーション、HTTP または TCP プロトコルの最適化などの機能を実行し、トラフィックを Kubernetes クラスター内の適切なマイクロサービスにルーティングします。このアーキテクチャは、同じチームが Kubernetes プラットフォームとアプリケーションデリバリーコントローラ (ADC) を含むその他のネットワークインフラストラクチャを管理するシナリオに最適です。

次の図は、統合された Ingress アーキテクチャを使用したデプロイメントを示しています。



統合された Ingress ソリューションには以下の主なメリットがあります。

- 既存の NetScaler インフラストラクチャの機能を Kubernetes 環境に拡張する方法を提供します
- インバウンドトラフィックにトラフィック管理ポリシーを適用できます
- ネットワークに精通した DevOps チームに適したシンプルなアーキテクチャを提供します
- マルチテナンシーをサポート

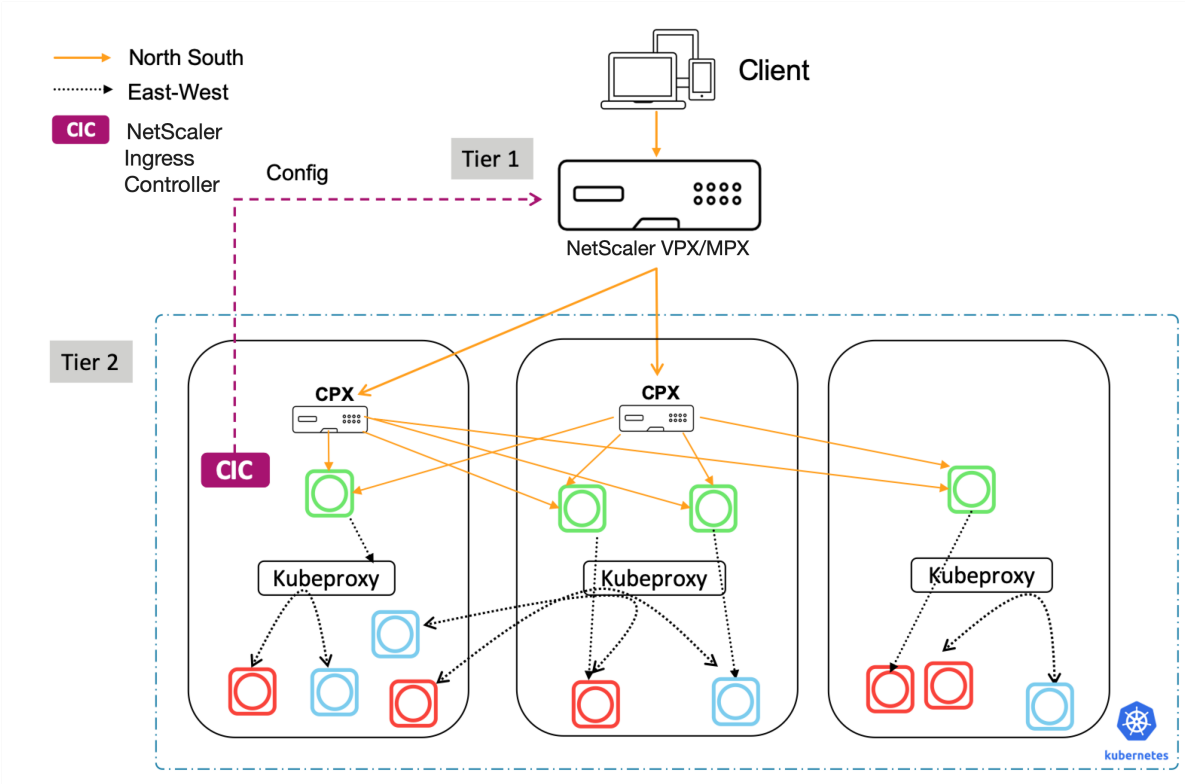
デュアル・ティア・イングレス

デュアルティアアーキテクチャでは、Kubernetes クラスターの外部にデプロイされた NetScaler ADC（MPX または VPX）がティア 1 で動作し、クラスター内で実行されている NetScaler ADC CPX への North-South トラフィックの負荷分散を行います。NetScaler CPX は階層 2 で動作し、Kubernetes クラスタ内のマイクロサービスの負荷分散を実行します。

個別のチームが Kubernetes プラットフォームとネットワークインフラストラクチャを管理するシナリオでは、デュアルティアアーキテクチャが最適です。

ネットワークチームは、GSLB、ハードウェアプラットフォームでの TLS ターミネーション、TCP 負荷分散などのユースケースにティア 1 の NetScaler を使用します。Kubernetes プラットフォームチームは、階層 2 の NetScaler（CPX）を使用して、レイヤー 7（HTTP/HTTPS）の負荷分散、相互 TLS、マイクロサービスのオブザーバビリティまたは監視を行うことができます。階層 2 の NetScaler（CPX）は、新たに利用可能になった機能に対応するために、階層 1 の NetScaler とは異なるソフトウェアリリースバージョンを用意できます。

次の図は、2 層アーキテクチャの導入を示しています。



デュアルティア Ingress には以下の主なメリットがあります。

- 開発者やプラットフォームチームのアプリケーション開発を高速化します
- Kubernetes クラスタ内のマイクロサービスに開発者主導のトラフィック管理ポリシーを適用できます
- クラウドの拡張とマルチテナンシーを実現

詳細については、NetScaler Ingress Controller ドキュメントを参照してください。

はじめに

Citrix の Kubernetes Ingress ソリューションを使い始めるには、以下の例を試してみてください。

- ミニクベの NetScaler CPX による入力トラフィックの負荷分散
- NetScaler CPX プロキシを使用して North-South 方向の入力トラフィックの負荷分散
- NetScaler CPX プロキシを使用して East-West のマイクロサービストラフィックの負荷分散

サービスメッシュ

August 15, 2023

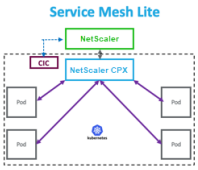
サービスメッシュは、API を使用してクラウドネイティブアプリケーションのサービス間通信を処理するためのインフラストラクチャレイヤーです。マイクロサービスを接続、保護、監視する方法を提供します。NetScaler には、お客様のサービスメッシュ要件を満たす 2 つのソリューションが用意されています。

- サービスメッシュライト
- サービスメッシュ（NetScaler ADC と Iistio との統合）

サービスメッシュライト

本格的なサービスメッシュの実装は複雑で、急な学習曲線が必要です。同様の利点を持つサービスメッシュのシンプルな実装をお探しの場合は、NetScaler の Service Mesh Lite というより複雑度の低いソリューションが提供されています。このソリューションでは、NetScaler CPX は Kubernetes クラスタ内の集中型ロードバランサーとして実行され、マイクロサービス間で East-West トラフィックを負荷分散します。NetScaler CPX は、インバウンドトラフィックとコンテナ間トラフィックにポリシーを適用します。

次の図は、サービスメッシュライトアーキテクチャを示しています。



詳細については、サービスメッシュ Lite のドキュメントを参照してください。

サービスメッシュ（**NetScaler ADC** と **Iistio** との統合）

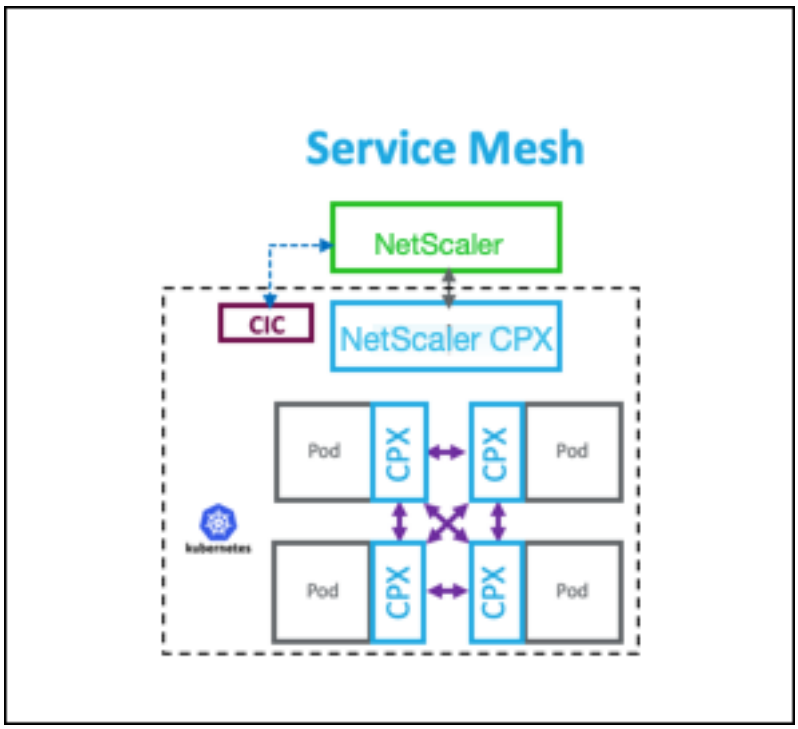
NetScaler は、NetScaler を Istio と統合することでサービスメッシュソリューションを提供します。オープンソースでプラットフォームに依存しないサービスメッシュである Istio は、最も一般的なサービスメッシュ実装の 1 つです。NetScaler を Istio と統合することで、NetScaler の機能を活用して、サービスメッシュ内のアプリケーションのトラフィックを保護および最適化できます。

NetScaler は、次の方法で Istio と統合できます。

- NetScaler MPX、VPX、または CPX をサービスメッシュへの Istio 入力ゲートウェイとして使用して、トラフィックを Kubernetes クラスターに公開します。
- NetScaler CPX は、サービスメッシュ内のアプリケーションコンテナを備えたサイドカープロキシとして機能し、アプリケーション間の通信を制御します。

どちらかの統合を単独で使用することも、両方の方法を組み合わせて統一されたデータプレーンソリューションを構築することもできます。

次の図は、サービスメッシュアーキテクチャを示しています。



サービスメッシュは安全性の高いアプリケーションに最適で、次のような利点もあります。

- コンテナごとのきめ細かい（モジュール化）トラフィック管理が可能
- サイドカーの実装により、より充実したオブザーバビリティ、分析、セキュリティ（相互 TLS）を実現
- NetScaler CPX が組み込まれた各コンテナのカナリア導入を自動化できます
- クラウドの移植性をサポート
- アプリケーションによって実行される機能のいくつかをサイドカーにオフロードできます。
- サイドカーレイテンシを低減
- オープンソースツールとの統合を実現
- 拡張性を提供

詳細については、NetScaler ADC と Istio の統合に関するドキュメントを参照してください。

監視性のためのソリューション

March 20, 2024

マイクロサービスベースのアーキテクチャでは、効率的で回復力のあるアーキテクチャを構築するには、サービス間の通信を可視化することが不可欠です。従来のロギングとモニタリングの方法では、マイクロサービスアーキテクチャの課題に対処することはできません。Citrix のオブザーバビリティソリューションを使用すると、サービスが相互に作用したときに何が起こっているかを確認し、システムに関する有意義な洞察を得ることができます。

NetScaler は、マイクロサービスアーキテクチャのオブザーバビリティのニーズに対応する次のソリューションを提供します:

- NetScaler コンソールのサービスグラフと分析
- NetScaler オブザーバビリティエクスポーター

NetScaler コンソールのサービスグラフと分析

NetScaler Application Delivery Management（ADM） は、複数のインスタンスで実行する必要がある管理ジョブを企業全体で可視化し、自動化できる一元管理ソリューションです。

マイクロサービスアーキテクチャでは、単一のエンドユーザー要求が複数のマイクロサービスにまたがる可能性があるため、トラブルシューティングは困難です。

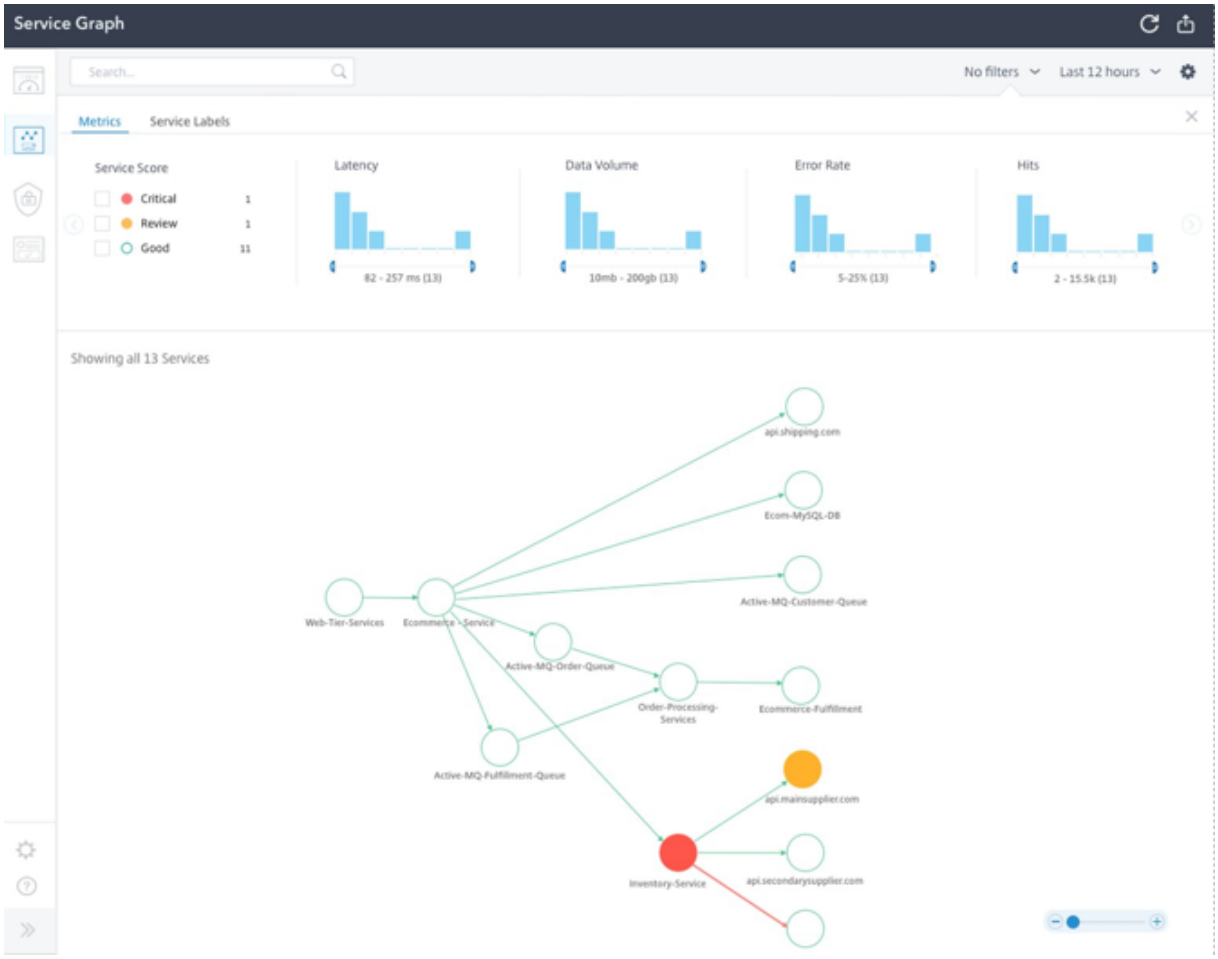
NetScaler Console のサービスグラフと分析は、マイクロサービス間の相互作用を可視化し、待ち時間や HTTP エラーなどのさまざまな指標に基づいて問題を特定して修正するのに役立ちます。

NetScaler Console は、NetScaler から収集されたメトリックとトランザクションログに基づいた高度な分析機能も提供します。

NetScaler コンソールソリューションには次の利点があります:

- コンテナ、オンプレミス、クラウドにまたがるアプリケーションを一元的に管理
- マイクロサービスの可観測性が向上し、トラブルシューティングが迅速になります。
- Canary の導入をサポート

次の図は、複数のマイクロサービスを含むアプリケーションのサンプルサービスグラフを示しています。



NetScaler Console のサービスグラフと分析の設定方法について詳しくは、サービスグラフのドキュメントを参照してください。

NetScaler オブザーバビリティエクスポーター

NetScaler Observability Exporter は、NetScaler からメトリックとトランザクションを収集し、サポートされているエンドポイントに適した形式（JSON、AVRO など）に変換するコンテナです。NetScaler オブザーバビリティエクスポーターによって収集されたデータを目的のエンドポイントにエクスポートできます。データを分析することで、NetScalers がプロキシするアプリケーションについて、マイクロサービスレベルで貴重な洞察を得ることができます。

分散型トレーシングのサポート

分散トレーサを使用すると、マイクロサービス間のデータフローを視覚化し、マイクロサービスアーキテクチャのボトルネックを特定するのに役立ちます。OpenTracing は、ディストリビューティッド（分散）トレーシングを設計および実装するための API の仕様および標準セットです 。

NetScaler オブザーバビリティエクスポーターは、NetScaler の分散トレースを実装しており、現在、分散トレーサーとして Zipkin をサポートしています。

Zipkin で Elasticsearch と Kibana を使用すると、トレース解析を強化できます。Elasticsearch は、トレースデータを長期的に保持します。Kibana では、ログメッセージを調べて視覚化するツールを提供することで、データについてより深い洞察を得ることができます。

トランザクション収集とストリーミングのサポート

NetScaler Observability Exporter は、トランザクションの収集とエンドポイントへのストリーミングをサポートしています。現在、NetScaler オブザーバビリティエクスポーターは、Elasticsearch と Kafka をトランザクションエンドポイントとしてサポートしています。

詳しくは、NetScaler オブザーバビリティエクスポーターのドキュメントを参照してください。

NetScaler Ingress Controller の YAML ファイル内のアノテーションを使用して分析を有効にする

Ingress または LoadBalancer 設定のサービスでスマートアノテーションとして定義されている分析プロファイルを使用して、分析を有効にできます。監視する必要がある特定のパラメータを定義するには、アプリケーションの Ingress またはサービス構成でパラメータを指定します。アノテーションを使用したアナリティクスの有効化の詳細については、「アノテーションを使用したアナリティクス」を参照してください。

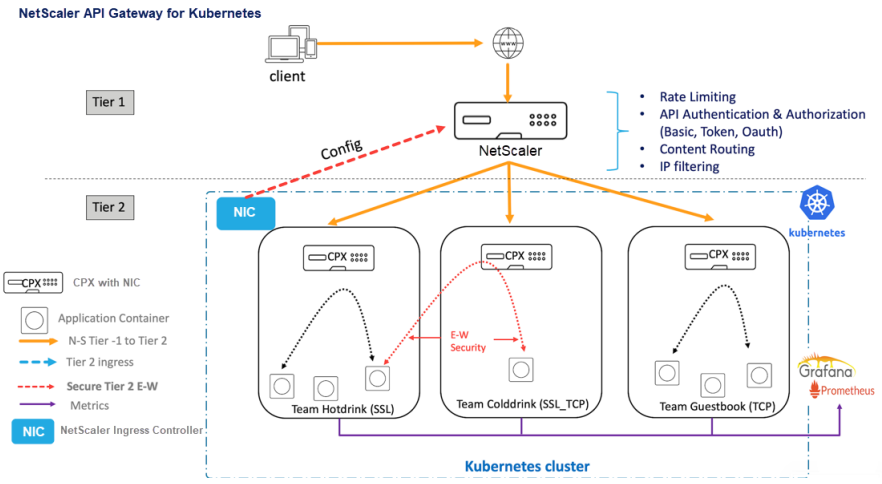
Kubernetes での API ゲートウェイ

August 15, 2023

API ゲートウェイは API の単一のエントリポイントとして機能し、システム内の複数の API やマイクロサービスへの安全で信頼性の高いアクセスを保証します。

NetScaler は、Kubernetes クラスターへの North-South API トラフィック用のエンタープライズグレードの API ゲートウェイを提供します。
API ゲートウェイは、NetScaler Ingress Controller と、オンプレミスまたはクラウド展開用のイングレスゲートウェイとしてデプロイされた NetScaler（NetScaler MPX、VPX、または CPX）を介して Kubernetes と統合されます。

次の図は、API ゲートウェイの 2 層トポロジを示しています。



Citrix が提供する API ゲートウェイを使用すると、次の機能を実行できます。

- 認証ポリシーを強制する
- サービスへのレートリミットアクセス
- 高度なコンテンツルーティング
- 書き換えポリシーとレスポンダーポリシーを使用した HTTP トランザクションの柔軟で包括的な変換
- Web アプリケーションファイアウォールポリシーを強制する

API Gateway の仕組み

API ゲートウェイは NetScaler Ingress ゲートウェイ上に構築され、カスタムリソース定義（CRD）などの Kubernetes API 拡張機能を使用します。CRD を使用すると、同じインスタンスで NetScaler ADC と API Gateway を自動的に構成できます。

NetScaler は、API ゲートウェイ用に次の CRD を提供します。

- 認証 CRD
- レートリミット CRD
- コンテンツルーティング CRD
- 書き換えとレスポンダ CRD
- WAF CRD

API ゲートウェイを使用する主な利点

Citrix が提供する APIGateway の主な利点は次のとおりです。

- NetScaler の高度なトラフィック管理と包括的なセキュリティ機能を使用します。
- 複数のネットワーク機能を Citrix Ingress Gateway の単一コンポーネントに統合することで、展開を最適化します。

- 複数のコンポーネントの導入に伴う運用の複雑さとコストを軽減します。
- 個別のコンポーネントを使用しながら複数ホップの TCP または TLS 復号化を減らすことで、アプリケーショントラフィックのパフォーマンスを向上させます。
- YAML またはヘルムチャートを直接使用することで、Kubernetes 環境へのデプロイと統合を簡素化します。

API ゲートウェイのデプロイ

CRD を使用して API ゲートウェイ機能を構成する方法の詳細については、NetScaler Ingress Controller ドキュメントを参照してください。

- 認証
- レート制限
- 高度なコンテンツルーティング
- 書き換えポリシーとレスポンダーポリシー
- Web アプリケーションファイアウォールポリシー

NetScaler コンソールを使用して **NetScaler** クラウドネイティブネットワークのトラブルシューティングを行います

March 21, 2024

概要

このドキュメントでは、NetScaler Console を使用して Kubernetes マイクロサービスアプリケーションを配信および監視する方法について説明します。また、CLI、サービスグラフ、トレースを使用して、プラットフォームと SRE チームがトラブルシューティングを行えるようにする方法についても説明します。

アプリケーションパフォーマンスとレイテンシーの概要

TLS 暗号化

TLS は、インターネット通信を保護するために設計された暗号化プロトコルです。TLS ハンドシェイクは、TLS 暗号化を使用する通信セッションを開始するプロセスです。TLS ハンドシェイク中、2 つの通信側はメッセージを交換して、相互の確認、相互の検証、使用する暗号化アルゴリズムの確立、セッションキーの合意を行います。TLS ハンドシェイクは、HTTPS の仕組みの基本的な部分です。

TLS と SSL ハンドシェイク

SSL (セキュア・ソケット・レイヤー) は、HTTP 用に開発されたオリジナルの暗号化プロトコルでした。TLS（トランスポート層セキュリティ）はしばらく前に SSL に取って代わりました。SSL ハンドシェイクは TLS ハンドシェイクと呼ばれるようになりましたが、「SSL」名は今でも広く使われています。

TLS ハンドシェイクはいつ発生しますか。

TLS ハンドシェイクは、ユーザーが HTTPS 経由で Web サイトに移動し、ブラウザーが最初に Web サイトのオリジンサーバーへのクエリを開始したときに実行されます。TLS ハンドシェイクは、API 呼び出しや DNS over HTTPS クエリなど、他の通信が HTTPS を使用する場合にも発生します。

TLS ハンドシェイクは、TCP ハンドシェイクを介して TCP 接続が開かれた後に発生します。

TLS ハンドシェイク中には何が起こりますか?

- TLS ハンドシェイク中、クライアントとサーバは共に次の処理を行います。
  - 使用する TLS のバージョン (TLS 1.0、1.2、1.3 など) を指定します。
  - 使用する暗号スイート (次のセクションを参照) を決定します。
  - サーバーの公開鍵と SSL 認証局のデジタル署名を使用して、サーバーの ID を認証します。
  - ハンドシェイクが完了したら、対称暗号化を使用するセッションキーを生成します。

TLS ハンドシェイクの手順を教えてください。

- TLS ハンドシェイクは、クライアントとサーバーによって交換される一連のデータグラム、つまりメッセージです。TLS ハンドシェイクには複数の手順が伴います。これは、クライアントとサーバーがハンドシェイクを完了し、さらに会話を可能にするために必要な情報を交換するためです。

TLS ハンドシェイク内の正確な手順は、使用する鍵交換アルゴリズムの種類と、両者がサポートする暗号スイートによって異なります。RSA 鍵交換アルゴリズムが最もよく使用されます。それは次のようになります。

1. 「client hello」メッセージ: クライアントは「hello」メッセージをサーバーに送信してハンドシェイクを開始します。このメッセージには、クライアントがサポートしている TLS バージョン、サポートされている暗号スイート、「クライアントランダム」と呼ばれるランダムなバイト文字列が含まれます。
2. 「server hello」メッセージ: クライアントの hello メッセージに応答して、サーバーは、サーバーの SSL 証明書、サーバーが選択した暗号スイート、および「server random」(サーバーによって生成されたもう 1 つのランダムなバイト文字列) を含むメッセージを送信します。
3. 認証: クライアントは、サーバーの SSL 証明書を、その証明書を発行した認証局と照合します。これにより、サーバーが本人であり、クライアントがドメインの実際の所有者とやり取りしていることが確認されます。
4. premaster secret: クライアントは、ランダムなバイト文字列「premaster secret」をもう 1 つ送信します。premaster シークレットは公開鍵で暗号化され、サーバーは秘密鍵でのみ復号できます。クライアントはサーバーの SSL 証明書から公開鍵を取得します。)
5. 使用した秘密鍵: サーバーはプレマスターシークレットを復号化します。
6. 作成されたセッションキー: クライアントとサーバーの両方が、クライアントランダム、サーバーランダム、およびプレマスターシークレットからセッションキーを生成します。彼らは同じ結果になるはずです。
7. Client is ready: クライアントは、セッションキーで暗号化された「完了」メッセージを送信します。

8. サーバー準備完了: サーバーは、セッションキーで暗号化された「完了」メッセージを送信します。
9. 安全な対称暗号化を実現: ハンドシェイクが完了し、セッションキーを使用して通信が継続されます。

すべての TLS ハンドシェイクは非対称暗号化 (公開鍵と秘密鍵) を使用しますが、セッション鍵を生成する過程で秘密鍵を使用するわけではありません。たとえば、エフェメラルな Diffie-Hellman ハンドシェイクは次のように処理されます。

1. Client hello: クライアントは、プロトコルバージョン、クライアントランダム、および暗号スイートのリストを含むクライアント hello メッセージを送信します。
2. Server hello: サーバーは、SSL 証明書、選択した暗号スイート、およびサーバーランダムで応答します。前のセクションで説明した RSA ハンドシェイクとは対照的に、このメッセージにはサーバに次の内容も含まれています（ステップ 3)。
3. サーバーのデジタル署名: サーバーは秘密鍵を使用して、クライアントランダム、サーバーランダム、および DH パラメーター * を暗号化します。この暗号化されたデータはサーバーのデジタル署名として機能し、SSL 証明書の公開キーと一致する秘密キーがサーバーにあることを証明します。
4. デジタル署名の確認: クライアントは、公開鍵を使用してサーバーのデジタル署名を復号化し、サーバーが秘密鍵を制御していること、および本人であることを確認します。Client DH パラメータ: クライアントは DH パラメータをサーバに送信します。
5. クライアントとサーバがプレマスターシークレットを計算する： RSA ハンドシェイクのように、クライアントがプレマスターシークレットを生成してサーバに送信する代わりに、クライアントとサーバは交換した DH パラメータを使用して、一致するプレマスターシークレットを個別に計算します。
6. 作成されたセッションキー: クライアントとサーバーは、RSA ハンドシェイクと同様に、プレマスターシークレット、クライアントランダム、およびサーバーランダムからセッションキーを計算するようになりました。

    - クライアントの準備完了:RSA
      ハンドシェイクと同じ
    - サーバは準備完了です
    - セキュアな対称暗号化を実現

    *DH パラメーター:DH は Diffie-Hellman の略です。Diffie-Hellman アルゴリズムは、指数計算を使用して同じ premaster シークレットに到達します。サーバーとクライアントはそれぞれ計算用のパラメーターを提供し、これらを組み合わせると、両側で異なる計算が行われ、結果は等しくなります。

エフェメラルな Diffie-Hellman ハンドシェイクと他の種類のハンドシェイクの対比、およびこれらがどのように前方秘匿性を実現するかについての詳細は、この TLS プロトコルのドキュメントを参照してください。

暗号スイートって何ですか?

- 暗号スイートは、セキュアな通信接続を確立するために使用する暗号化アルゴリズムのセットです。暗号化アルゴリズムとは、データをランダムに見せるためにデータに対して実行される一連の数学的演算です。さまざまな暗号スイートが広く使用されており、TLS ハンドシェイクの重要な部分は、そのハンドシェイクにどの暗号スイートを使用するかを合意することです。

開始するには、「リファレンス: TLS プロトコルのドキュメント」を参照してください。

NetScaler Application Delivery Management SSL ダッシュボード

NetScaler Application Delivery Management（ADM）により、証明書管理のあらゆる側面が合理化されるようになりました。1 つのコンソールから、使われていない、または期限切れが近い証明書のタブは閉じたまま、正しい発行者、キーの強度、および正しいアルゴリズムを確保する自動化されたポリシーを作成することができます。NetScaler Console SSL ダッシュボードとその機能を使い始めるには、SSL 証明書とは何か、NetScaler Console を使用して SSL 証明書を追跡する方法を理解する必要があります。

SSL トランザクションの一部であるセキュアソケットレイヤー (SSL) 証明書は、企業 (ドメイン) または個人を識別するデジタルデータフォーム (X509) です。この証明書には、サーバーとの安全なトランザクションを開始しようとするすべてのクライアントが確認できる公開キーコンポーネントが含まれます。対応する秘密キーは、Citrix Application Delivery Controller（ADC）アプライアンスに安全に配置され、非対称キー（または公開キー）の暗号化と復号化を完了するために使用されます。

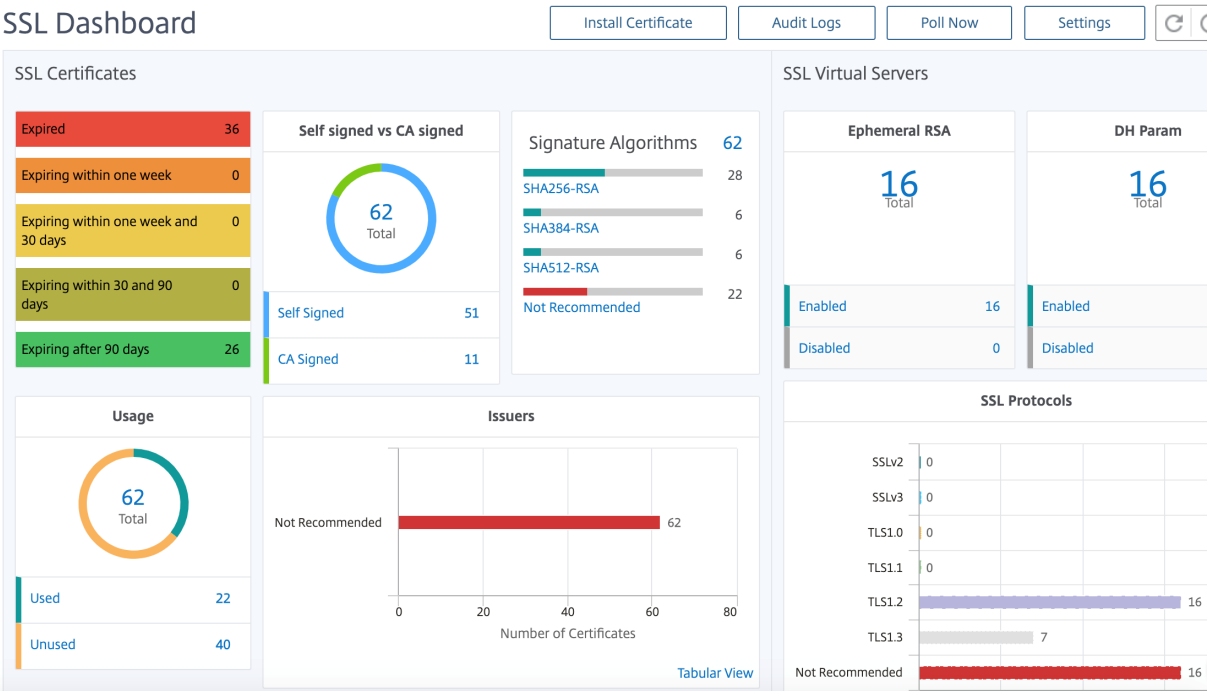
SSL 証明書およびキーは、次のいずれかの方法で入手できます。

- 認可された認証局 (CA) から
- NetScaler アプライアンス上で新しい SSL 証明書とキーを生成する

NetScaler コンソールでは、管理対象のすべての NetScaler インスタンスにインストールされている SSL 証明書を一元的に表示できます。SSL Dashboard では、証明書の発行者、キーの強度、署名アルゴリズム、期限切れまたは未使用の証明書などを追跡するのに役立つグラフを表示できます。また、仮想サーバーで実行されている SSL プロトコルの分布および各サーバーで有効化されているキーも確認できます。

さらに、証明書の有効期限が近づいたときに、証明書が間もなく期限切れになるという情報と、その証明書を使用している NetScaler インスタンスに関する情報が届くように通知を設定できます。

NetScaler インスタンスの証明書を CA 証明書にリンクできます。ただし、同じ CA 証明書にリンクする証明書のソースと発行元が同じであることを確認してください。証明書を CA 証明書にリンクしたら、それらのリンクを解除できます。



開始するには、SSL ダッシュボードのドキュメントを参照してください。

サード・パーティとの連携

アプリケーションのレイテンシーはミリ秒単位で測定され、使用するメトリクスに応じて 2 つのうちの 1 つを示すことができます。レイテンシを測定する一般的な方法は「ラウンドトリップ時間」(RTT) と呼ばれます。RTT は、データパケットがネットワーク上のある地点から別の地点に移動して、応答が送信元に返されるまでにかかる時間を計算します。もう 1 つは「最初のバイトまでの時間」(Time to first byte) (または TTFB) と呼ばれ、パケットがネットワーク上のある地点を出発してから宛先に到着するまでにかかる時間を記録します。RTT は、ネットワーク上の 1 つのポイントから実行でき、(TTFB のように) データ収集ソフトウェアを宛先ポイントにインストールする必要がないため、レイテンシの測定によく使用されます。

ADM サービスでは、アプリケーションの帯域幅の使用量とパフォーマンスをリアルタイムで監視することで、問題を簡単に特定し、潜在的な問題が顕在化してネットワーク上のユーザーに影響を及ぼす前に先制的に対処できます。このフローベースのソリューションは、インターフェイス、アプリケーション、カンバセーションごとに使用状況を追跡し、ネットワーク全体のアクティビティに関する詳細情報を提供します。

Splunk ツールを使う

インフラストラクチャとアプリケーションのパフォーマンスは相互に依存しています。全体像を把握するために、SignalFX はクラウドインフラストラクチャとその上で実行されるマイクロサービスとのシームレスな相関関係を提供します。メモリリーク、ノイズの多いネイバーコンテナー、その他のインフラストラクチャ関連の問題が原因でア

プリケーションが動作した場合、SignalFX から通知されます。全体像を把握するために、コンテキスト内で Splunk のログとイベントにアクセスすることで、より詳細なトラブルシューティングと根本原因の分析が可能になります。

SignalFX マイクロサービス APM と Splunk によるトラブルシューティングの詳細については、DevOps 向け Splunk の情報をご覧ください 。

MongoDB サポート

MongoDB は、柔軟な JSON に似たドキュメントにデータを格納します。つまり、フィールドはドキュメントごとに異なり、データ構造は時間の経過とともに変化する可能性があります。

ドキュメントモデルはアプリケーションコード内のオブジェクトにマップされるため、データの操作が容易になります。

オンデマンドクエリ、インデックス作成、リアルタイム集約により、データにアクセスして分析するための強力な方法が提供されます。

MongoDB は中核をなす分散データベースであるため、高可用性、水平スケーリング、地理的分散が組み込まれており、使いやすくなっています。

MongoDB は、以下を実現するテクノロジー基盤により、最新のアプリケーションの要求を満たすように設計されています。

- ドキュメントデータモデル—データを操作する最適な方法を提供します。
- 分散システム設計—データを必要な場所にインテリジェントに配置できます。
- どこにいても自由に実行できる統合エクスペリエンスにより、将来を見据えた作業が可能になり、ベンダーロックインを排除できます。

これらの機能により、MongoDB に支えられたインテリジェントな運用データプラットフォームを構築できます。詳細については、MongoDB のドキュメントを参照してください。

Ingress トラフィックを TCP または UDP ベースのアプリケーションに負荷分散する方法

Kubernetes 環境では、Ingress は Kubernetes クラスターの外部から Kubernetes サービスへのアクセスを許可するオブジェクトです。標準の Kubernetes Ingress リソースは、すべてのトラフィックが HTTP ベースであり、TCP、TCP-SSL、UDP などの HTTP ベース以外のプロトコルには対応していないと想定しています。したがって、DNS、FTP、LDAP などの L7 プロトコルに基づく重要なアプリケーションは、標準の Kubernetes Ingress を使用して公開することはできません。

Kubernetes の標準ソリューションは、LoadBalancer タイプのサービスを作成することです。詳細については、「NetScaler のサービスタイプロードバランサー」 を参照してください。

2 番目のオプションは、Ingress オブジェクトに注釈を付けることです。NetScaler Ingress Controller を使用すると、TCP または UDP ベースの入力トラフィックの負荷分散が可能になります。Kubernetes Ingress リソース定義で以下のアノテーションを使用して 、TCP または UDP ベースの Ingress トラフィックの負荷を分散できます。

- ingress.citrix.com/insecure-service-type: このアノテーションにより、NetScaler のプロトコルとして TCP、UDP、または ANY による L4 負荷分散が可能になります。
- ingress.citrix.com/insecure-port: アノテーションは TCP ポートを構成します。このアノテーションは、非標準ポートでマイクロサービスアクセスが必要な場合に役立ちます。デフォルトでは、ポート 80 が設定されています。

詳細については、「Ingress トラフィックを TCP または UDP ベースのアプリケーションに負荷分散する方法」を参照してください。

TCP または UDP ベースのアプリケーションのパフォーマンスを監視し、改善する

アプリケーション開発者は、NetScaler のリッチモニター（TCP-ECV、UDP-ECV など）を使用して、TCP または UDP ベースのアプリケーションの状態を綿密に監視できます。ECV (拡張コンテンツ検証) モニターは、アプリケーションが予期したコンテンツを返しているかどうかを確認するのに役立ちます。

また、ソース IP などの永続化方法を使用することで、アプリケーションのパフォーマンスを向上させることができます。これらの NetScaler 機能は、Kubernetes のスマートアノテーションを通じて使用できます 。その一例を以下に挙げます。

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
    name: mongodb
    annotations:
        ingress.citrix.com/insecure-port:  “80”
        ingress.citrix.com/frontend-ip:  “192.168.1.1”
```



```
        ingress.citrix.com/csvserver:  '{
  "l2conn" :" on"  }
 '
        ingress.citrix.com/lbvserver:  '{
  "mongodb-svc" :{
  "lbmethod" :" SRCIPDESTIPHASH"  }
  }
 '
        ingress.citrix.com/monitor:  '{
  "mongodbsvc" :{
  "type" :" tcp-ecv"  }
  }
 '
Spec:
    rules:
    - host: mongodb.beverages.com
      http:
        paths:
        - path: /
          backend:
            serviceName: mongodb-svc
            servicePort: 80
<!--NeedCopy-->
```

NetScaler Application Delivery Management (ADM) サービス

NetScaler コンソールサービスには次の利点があります:

- 機敏性—運用、更新、使用が容易。NetScaler Console Service のサービスモデルはクラウド上で利用できるため、提供されている機能の運用、更新、使用が簡単です。更新の頻度と自動更新機能の組み合わせにより、NetScaler 展開が迅速に強化されます。
- タイム・ツ・バリューの短縮—ビジネス目標の達成を迅速化。従来のオンプレミス展開とは異なり、NetScaler コンソールサービスは数回クリックするだけで使用できます。インストールと設定の時間を節約できるだけでなく、潜在的なエラーに時間とリソースを浪費することもありません。
- マルチサイト管理—複数のサイトデータセンターにまたがるインスタンスを 1 つのガラスで管理できます。NetScaler コンソールサービスを使用すると、さまざまなタイプの展開環境にある NetScaler を管理および監視できます。オンプレミスとクラウドに導入された NetScalers をワンストップで管理できます。
- 運用効率—運用生産性を向上させる最適化および自動化された方法。NetScaler Console Service を使用すると、従来のハードウェア展開の保守とアップグレードにかかる時間、費用、リソースを節約できるため、運用コストが削減されます。

Kubernetes アプリケーションのサービスグラフ

NetScaler Console のクラウドネイティブアプリケーション機能のサービスグラフを使用すると、次のことが可能になります:

- エンド・ツー・エンドのアプリケーション全体のパフォーマンスを確保
- アプリケーションのさまざまなコンポーネントの相互依存性によって生じるボトルネックを特定
- アプリケーションのさまざまなコンポーネントの依存関係に関する洞察を集める
- Kubernetes クラスター内のサービスを監視する
- 問題のあるサービスを監視する
- パフォーマンスの問題に寄与する要因を確認する
- サービス HTTP トランザクションの詳細な可視性を表示
- HTTP、TCP、SSL メトリックの分析

NetScaler Console でこれらの指標を視覚化することで、問題の根本原因を分析し、必要なトラブルシューティングアクションをより迅速に実行できます。サービスグラフには、さまざまなコンポーネントサービス内のアプリケーションが表示されます。Kubernetes クラスター内で実行されるこれらのサービスは、アプリケーション内外のさまざまなコンポーネントと通信できます。

はじめに、「サービスグラフの設定」をご参照ください。

3 層 **Web** アプリケーションのサービスグラフ

アプリケーションダッシュボードのサービスグラフ機能を使用すると、次の項目を表示できます。

- アプリケーションの構成方法の詳細（コンテンツスイッチング仮想サーバーと負荷分散仮想サーバーを使用）
  - GSLB アプリケーションの場合、データセンター、ADC インスタンス、CS、および LB 仮想サーバーを表示できます。
- クライアントからサービスへのエンド・ツー・エンドのトランザクション
- クライアントがアプリケーションにアクセスしている場所
- クライアント要求が処理されるデータセンターの名前と、関連するデータセンター NetScaler メトリック（GSLB アプリケーションのみ）
- クライアント、サービス、仮想サーバーのメトリックの詳細
- エラーがクライアントまたはサービスからのものである場合
- 「緊急」、「レビュー」、「良好」などのサービスステータス。NetScaler Console は、サービスの応答時間とエラー数に基づいてサービスの状態を表示します。
  - 重大 **(**赤**)** -平均サービス応答時間が 200 ミリ秒を超え、エラーカウントが 0 より大きいことを示します。
  - **Review (**オレンジ**)** -平均サービス応答時間が 200 ミリ秒を超えるか、エラーカウントが 0 より大きいことを示します。
  - 良好 **(**緑**)** -エラーがなく、平均サービス応答時間が 200 ミリ秒未満であることを示します
- **Critical**、**Review**、**Good** などのクライアントのステータス。NetScaler Console は、クライアントのネットワーク遅延とエラー数に基づいてクライアントステータスを表示します。
  - **Critical (**赤**)**-平均クライアントネットワーク遅延が 200 ミリ秒を超え、エラーカウントが 0 より大きいことを示します

  - **Review (**オレンジ**)** -平均クライアントネットワーク遅延が 200 ミリ秒を超えるか、エラーカウントが 0 より大きいことを示します。
  - 良好 **(**緑**)** -エラーがなく、平均クライアントネットワーク遅延が 200 ミリ秒未満であることを示します。

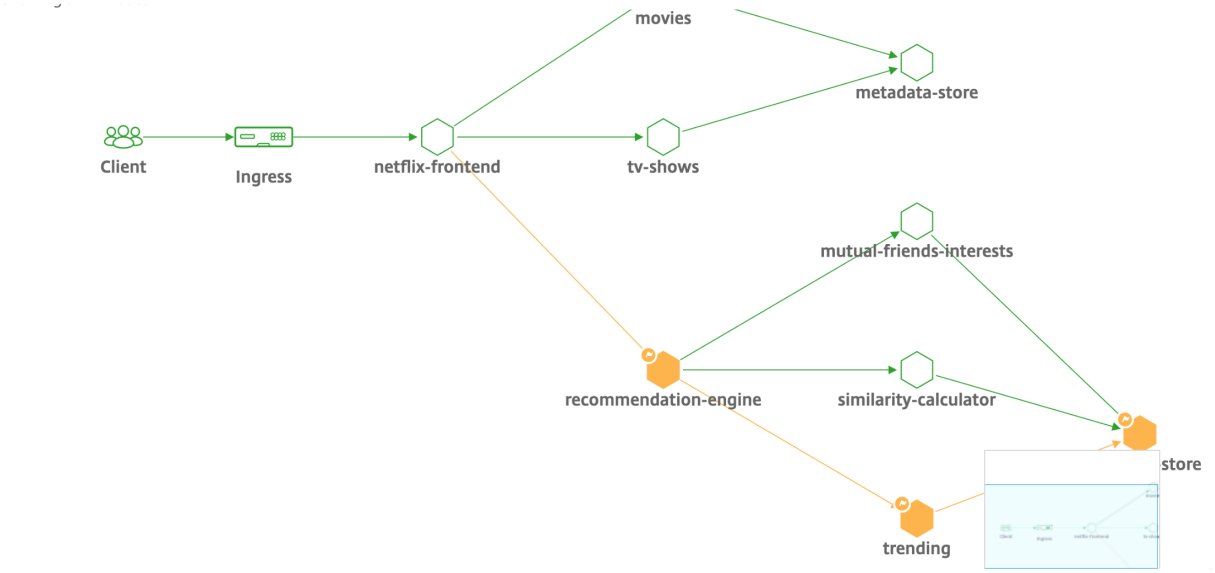
- クリティカル、レビュー、良好（**Good**）などの仮想サーバのステータス。NetScaler Console には、アプリスコアに基づいて仮想サーバーの状態が表示されます。

  - クリティカル **(**赤**)** -アプリのスコアが 40 未満になったことを示します
  - **Review (**オレンジ**)** -アプリのスコアが 40～75 の間であることを示します
  - **Good**（緑） -アプリのスコアが 75 を超えることを示します。

注意事項:

- サービスグラフには、負荷分散、コンテンツスイッチング、GSLB 仮想サーバーのみが表示されます。
- 仮想サーバがカスタムアプリケーションにバインドされていない場合、その詳細はそのアプリケーションのサービスグラフに表示されません。
- 仮想サーバと Web アプリケーションの間でアクティブなトランザクションが発生した場合にのみ、クライアントとサービスのメトリックをサービスグラフに表示できます。
- 仮想サーバーと Web アプリケーション間でアクティブなトランザクションを利用できない場合は、負荷分散、コンテンツスイッチング、GSLB 仮想サーバー、サービスなどの構成データに基づくサービスグラフにのみ詳細を表示できます。
- アプリケーション設定の更新がサービスグラフに反映されるまで 10 分かかる場合があります。

詳細については、「アプリケーション用サービスグラフ」を参照してください。



開始するには、Service Graph のドキュメントを参照してください。

NetScaler チーム向けのトラブルシューティング

それでは、NetScaler プラットフォームのトラブルシューティングで最もよく使われる属性と、これらのトラブルシューティング手法がマイクロサービストポロジーの Tier-1 展開にどのように適用されるかについて説明します。

NetScaler には、コマンドをリアルタイムで表示するコマンドラインインターフェイス（CLI）があり、ランタイム構成、静的、およびポリシー構成を決定するのに役立ちます。これは「**SHOW**」コマンドで簡単に行えます。

SHOW-ADC CLI オペレーションを実行します。

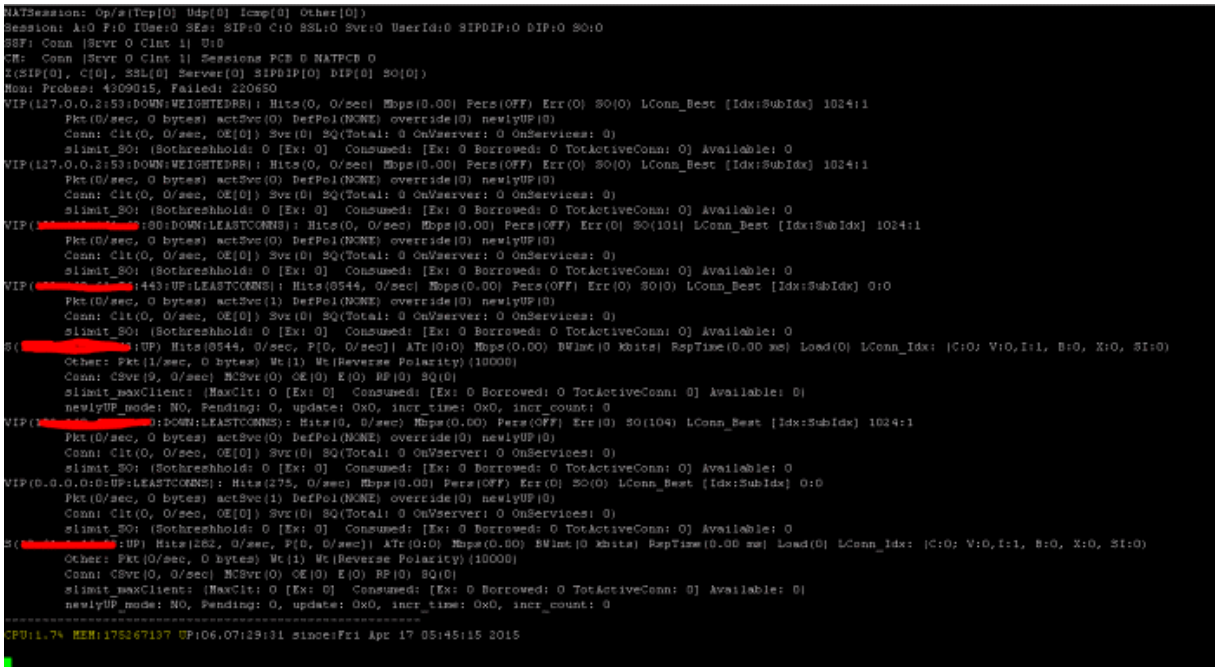
```
>Show running config (-summary -fullValues)

 Ability to search (grep command)
> “sh running config | -i grep vserver”

Check the version.
>Show license
   “sh license"
<!--NeedCopy-->
```

SSL 統計情報の表示

```
>Sh ssl
         System
         Frontend
         Backend
         Encryption
<!--NeedCopy-->
```



NetScaler には、7 秒のカウンター間隔に基づいてすべてのオブジェクトの統計を列挙するコマンドがあります。これは「**STAT**」コマンドによって容易になります。

NetScaler によるきめ細かな **L3-L7** テレメトリー

- システムレベル:ADC の CPU およびメモリ使用率。
- HTTP プロトコル:#Requests /レスポンス、GET/POST スプリット、N-S および E-W の HTTP エラー (サービスメッシュライトのみ、サイドカーはまもなく)。
- SSL: #Sessions および #Handshakes は、サービスメッシュ Lite 専用の N-S および E-W トラフィック用です。
- IP プロトコル:#Packets 受信/送信、#Bytes 受信/送信、#Truncated パケット、#IP アドレスルックアップ
- NetScaler AAA: #Active セッション
- インターフェイス:#Total マルチキャストパケット、#Total 転送バイト、および送受信された #Jumbo パケット
- 負荷分散仮想サーバーとコンテンツスイッチング仮想サーバー:#Packets、#Hits、#Bytes が受信/送信されました。

STAT-ADC CLI オペレーションを実行します。

```
1 >Statistics
2  “stat ssl”
3 <!--NeedCopy-->
```



```
> stat ns

System overview

Up since        Thu Apr 16 19:45:15 2015
Packet CPU usage (%)                1.60
Management CPU usage (%)            0.80
Memory usage (MB)                    165
InUse Memory (%)                   17.03
Last Transition time Th...015
System state                         UP
Master state                    Primary
# SSL cards UP                         0
# SSL cards present                    0

System Disks                   Used (%) Available
/flash Used (%)                       17      1168
/var Used (%)                         13     11246

Throughput Statistics                    Rate (/s)               Total
Megabits received                                2                288237
Megabits transmitted                             3                345685

TCP Connections                  Client   Server
All client connections              158      272
Established client connections      158      145

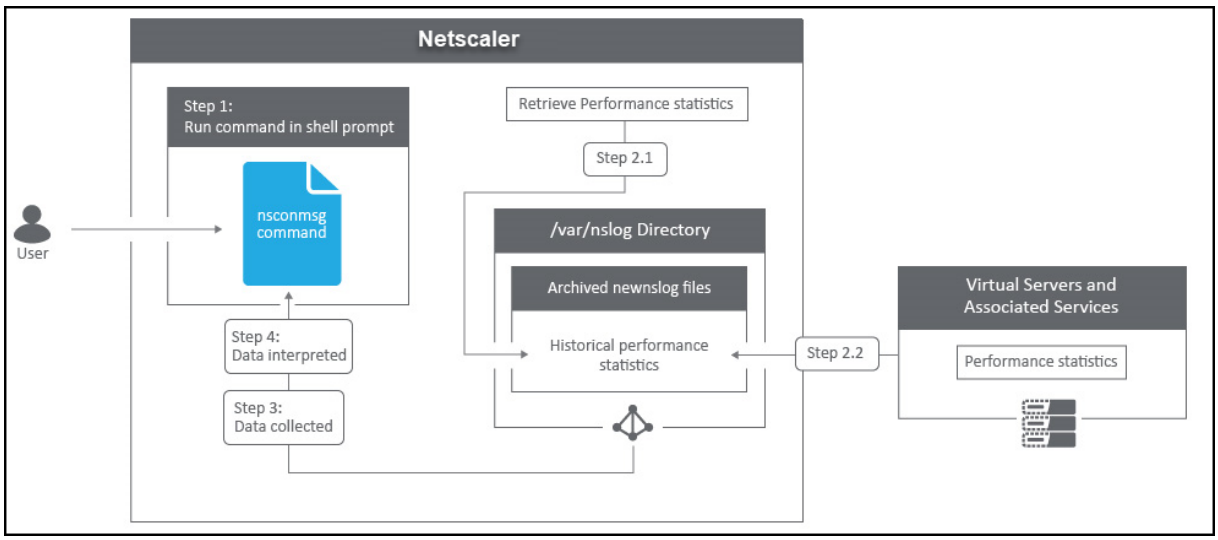
HTTP                                     Rate (/s)               Total
Total requests                                   0                191529
Total responses                                  0                263011
Request bytes received                        7007           1178810535
Response bytes received                     164477          12348432171

SSL                                      Rate (/s)               Total
```

NetScaler にはログアーカイブ構造があり、「**NSCONMSG** 」コマンドを使用して特定のエラーをトラブルシューティングする際に統計やカウンタを検索できます。

NSCONMSG -メインログファイル **(ns** データ形式**)**

```
1      Cd/var/nslog
2
3       “Mac Moves”
4      nsconmsg -d current -g nic_err
5  <!--NeedCopy-->
```

Nstcpdump

nstcpdumpは、低レベルのトラブルシューティングに使用できます。nstcpdumpは、nstraceより詳細な情報を収集しません。ADC CLI を開き、shellと入力します。フィルタはnstcpdumpと使用できますが、ADC リソース固有のフィルタとともには使用できません。ダンプ出力は CLI 画面内で直接表示できます。

CTRL+C —これらのキーを同時に押すと、nstcpdumpが停止します。

`nstcpdump.sh dst host x.x.x.x` —宛先ホストに送信されたトラフィックを表示します。

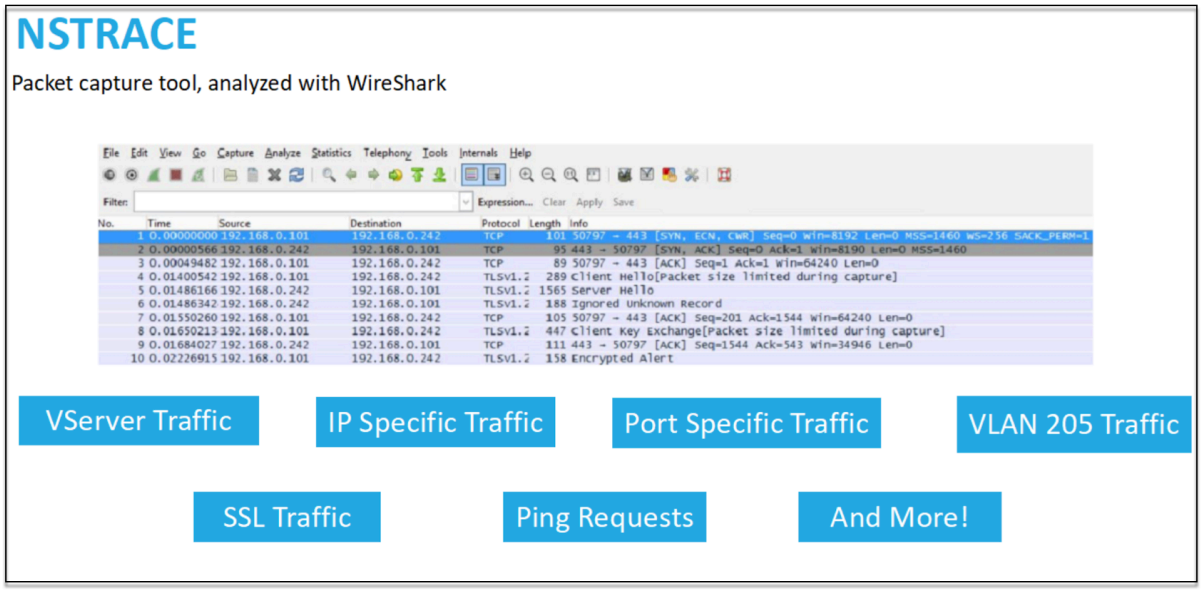
`nstcpdump.sh -n src host x.x.x.x` —指定されたホストからのトラフィックを表示し、IP アドレスを名前に変換しない (-n)。

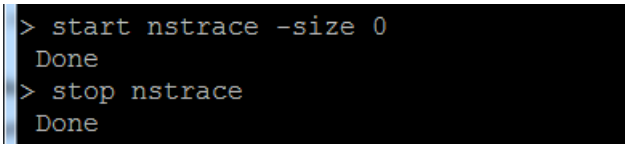
`nstcpdump.sh host x.x.x.x` —指定したホスト IP との間で送受信されるトラフィックを表示します。


```
root@Netscaler1# nstcpdump.sh -c 10 dst host 192.168.0.242
reading from file -, link-type EN10MB (Ethernet)
21:45:45.834700 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [S], seq 1702255264, win 8192, options [mss 1460,nop,wscale 8,nop,nop,sackOK],
length 0
21:45:45.836702 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [.], ack 748367253, win 64240, length 0
21:45:45.837202 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [P.], ack 1, win 64240, length 232
21:45:45.839203 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [.], ack 1544, win 64240, length 0
21:45:45.840244 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [P.], ack 1544, win 64240, length 342
21:45:45.847709 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [P.], ack 1619, win 64165, length 469
21:45:45.994744 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [P.], ack 2712, win 63072, length 581
21:45:46.002746 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [.], ack 7092, win 64240, length 0
21:45:46.003250 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [.], ack 15853, win 64240, length 0
21:45:46.009748 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [.], ack 30455, win 64240, length 0
```

NSTRACE -パケットトレースファイル

NSTRACE は、ネットワークをトラブルシューティングするための低レベルのパケットデバッグツールです。これにより、アナライザツールを使用してさらに分析できるキャプチャファイルを保存できます。一般的なツールは、ネットワークアナライザと Wireshark の 2 つです。





NSTRACE キャプチャファイルが ADC の /var/nstrace に作成されると、キャプチャファイルを Wireshark にインポートして、パケットキャプチャとネットワーク分析を行うことができます。

SYSCTL-詳細な **ADC** 情報**:** 説明、モデル、プラットフォーム、**CPU** など

```
    sysctl -a grep hw.physmem

    hw.physmem: 862306304
    netscaler.hw_physmem_mb: 822
<!--NeedCopy-->
```

aaad.debug-認証デバッグ情報のためにパイプをオープンする

```
process_radius Got RADIUS event
process_radius Received BAD_ACCESS_REJECT for: <username>
process_radius Sending reject.
send_reject_with_code Rejecting with error code 4001.
```

aaad.debug モジュールを使用した ADC または ADC ゲートウェイ経由での認証問題のトラブルシューティング方法の詳細については、aaad.debug のサポート記事を参照してください。

また、ADC のパフォーマンス統計やイベント・ログを直接取得することもできます。詳細については、ADC サポートドキュメントを参照してください。

SRE チームとプラットフォームチームのトラブルシューティング

Kubernetes トラフィックフロー

North/South:

- North/South トラフィックは、イングレス経由でユーザからクラスタに流れるトラフィックです。

East/West:

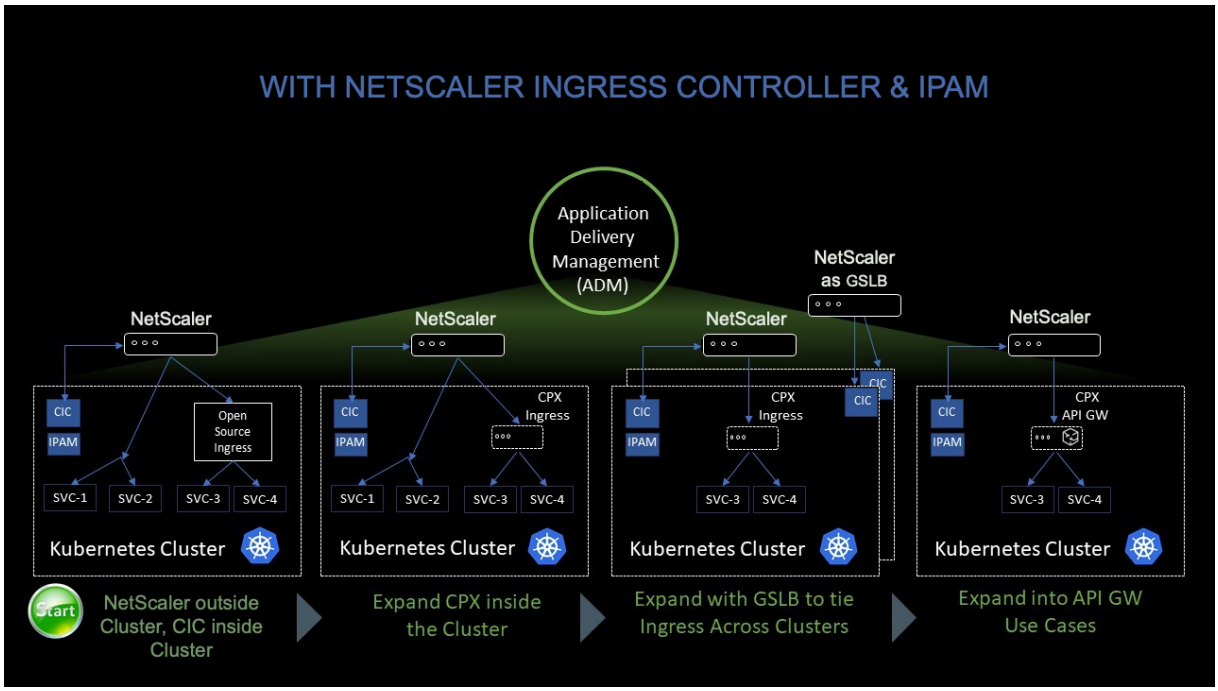
- East/West トラフィックは、Kubernetes クラスタの周りを流れるトラフィック (サービス間またはサービス間データストア) です。

NetScaler CPX が **Kubernetes** 環境で **east-west** トラフィックフローを負荷分散する方法

Kubernetes クラスターをデプロイしたら、ADM で Kubernetes 環境の詳細を指定して、クラスターを ADM と統合する必要があります。ADM は、サービス、エンドポイント、Ingress ルールなどの Kubernetes リソースの変更を監視します。

NetScaler CPX インスタンスを Kubernetes クラスターにデプロイすると、自動的に ADM に登録されます。登録プロセスの一環として、ADM は CPX インスタンスの IP アドレスと、NITRO REST API を使用してインスタンスに到達して構成できるポートについて学習します。

次の図は、Kubernetes クラスター内で NetScaler CPX が East-West トラフィックフローをどのように負荷分散するかを示しています。



この例の説明を次に示します。

Kubernetes クラスターのノード 1 とノード 2 には、フロントエンドサービスとバックエンドサービスのインスタンスが含まれています。NetScaler CPX インスタンスをノード 1 とノード 2 に展開すると、NetScaler CPX インスタンスは自動的に ADM に登録されます。ADM で Kubernetes クラスターの詳細を設定して、Kubernetes クラスターを ADM に手動で統合する必要があります。

クライアントがフロントエンドサービスを要求すると、Ingress リソースは、2 つのノード上にあるフロントエンドサービスのインスタンスの間で要求を負荷分散します。フロントエンドサービスのインスタンスがクラスター内のバックエンドサービスからの情報を必要とする場合、そのノードの NetScaler CPX インスタンスに要求が送信されます。その NetScaler CPX インスタンスは、クラスター内のバックエンドサービス間でリクエストの負荷分散を行い、East-West のトラフィックフローを実現します。

アプリケーション用の **ADM** サービスグラフ

NetScaler Console のサービスグラフ機能を使用すると、すべてのサービスをグラフィカルに監視できます。この機能は、詳細な分析と有用なメトリックも提供します。次のサービスグラフを表示できます。

- すべての NetScaler インスタンスで構成されたアプリケーション
- Kubernetes アプリケーション
- 3 層の Web アプリケーション

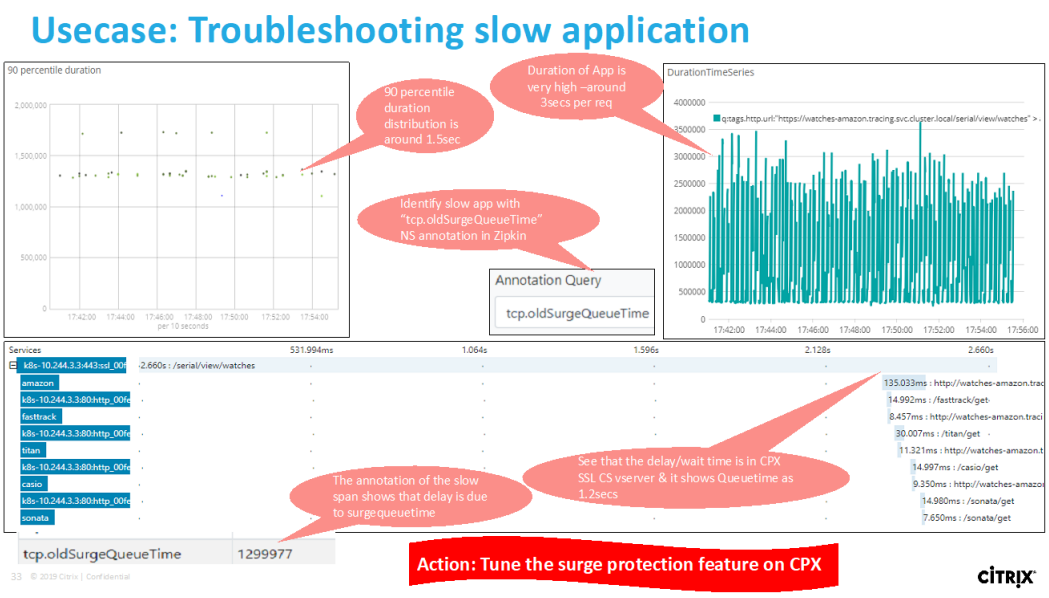
開始するには、サービスグラフの詳細を参照してください。

マイクロサービスアプリケーションカウンターの表示

サービスグラフには、Kubernetes クラスターに属するすべてのマイクロサービスアプリケーションも表示されます。ただし、サービスにマウスポインタを置くと、メトリクスの詳細が表示されます。

以下を表示できます:

- サービス名
- SSL、HTTP、TCP、SSL over HTTP、SSL などのサービスで使用されるプロトコル
- **Hits** —サービスによって受信されたヒットの総数
- サービス応答時間—サービスから取得した平均応答時間。
  (応答時間 = クライアント RTT + 要求の最後のバイト—要求の最初のバイト)
- エラー—4xx、5xx などのエラーの総数
- **Data Volume** —サービスによって処理されるデータの総量
- 名前空間—サービスの名前空間
- クラスター名—サービスがホストされているクラスター名
- **SSL** サーバーエラー—サービスからの SSL エラーの合計



これらの特定のカウンターとトランザクションログは、サポートされているさまざまなエンドポイントを使用して NetScaler Observability Exporter（COE）から抽出できます。COE の詳細については、次のセクションを参照してください。

NetScaler 統計用エクスポーター

これは、NetScaler 統計情報をスクレイピングし、HTTP 経由で Prometheus にエクスポートするシンプルなサーバーです。その後、Prometheus をデータソースとして Grafana に追加して、NetScaler の統計をグラフィカルに表示できます。

NetScaler インスタンスの統計情報とカウンターを監視するには、`citrix-adc-metric-exporter` コンテナまたはスクリプトとして実行できます。エクスポーターは、仮想サーバーへの総ヒット数、HTTP 要求レート、SSL 暗号化/復号化レートなど、NetScaler インスタンスから NetScaler 統計を収集し、Prometheus サーバーが統計情報を取得してタイムスタンプ付きで保存するまで保持します。その後、Grafana を Prometheus サーバーにポイントして、NetScaler 統計の分析に必要な統計情報の取得、プロット、アラームの設定、ヒートマップの作成、テーブルの生成などを行うことができます。

以下のセクションでは、図に示すような環境でエクスポータが動作するように設定する方法について詳しく説明します。エクスポーターがデフォルトでスクレイピングする NetScaler エンティティ/メトリックとその変更方法についても説明します。

NetScaler 用エクスポーターについて詳しくは、メトリクスエクスポーター GitHubを参照してください。

ADM サービス分散トレーシング

サービスグラフでは、分散トレーシングビューを使用して次の操作を実行できます。

- サービス全体のパフォーマンスを分析します。
- 選択したサービスとその相互依存サービス間の通信フローを視覚化します。
- エラーを示すサービスを特定し、エラーのあるサービスをトラブルシューティングする
- 選択したサービスと相互依存する各サービス間のトランザクション詳細を表示します。

ADM 分散トレーシングの前提条件

サービスのトレース情報を表示するには、次の操作を行う必要があります。

- East-West トラフィックを送信する間、アプリケーションが次のトレースヘッダーを保持していることを確認します。

- x-request-id
- x-b3-traceid
- x-b3-spanid
- x-b3-parentspanid
- x-b3-sampled
- x-b3-flags
- x-ot-span-context

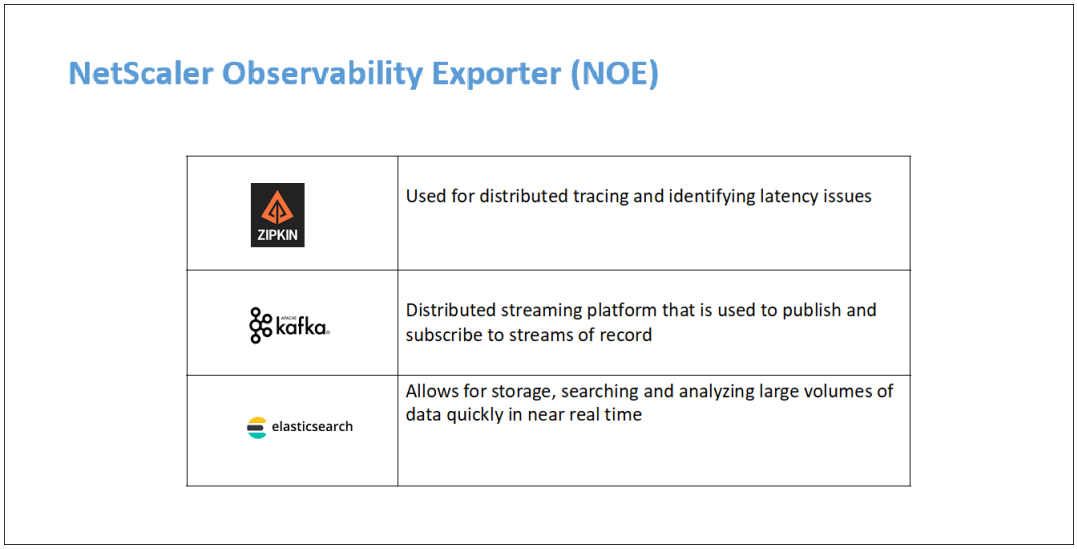
  - CPX YAML ファイルを NS_DISTRIBUTED_TRACING で更新し、値を「はい」に設定します。
    はじめに、「分散トレーシング」を参照してください。

NetScaler オブザーバビリティエクスポーター（**COE**）の解析

NetScaler Observability Exporter は、NetScaler からメトリックとトランザクションを収集し、サポートされているエンドポイントに適した形式（JSON、AVRO など）に変換するコンテナです。NetScaler Observability Exporter によって収集されたデータを目的のエンドポイントにエクスポートできます。エンドポイントにエクスポートされたデータを分析することで、NetScalers がプロキシするアプリケーションのマイクロサービスレベルで貴重な洞察を得ることができます。

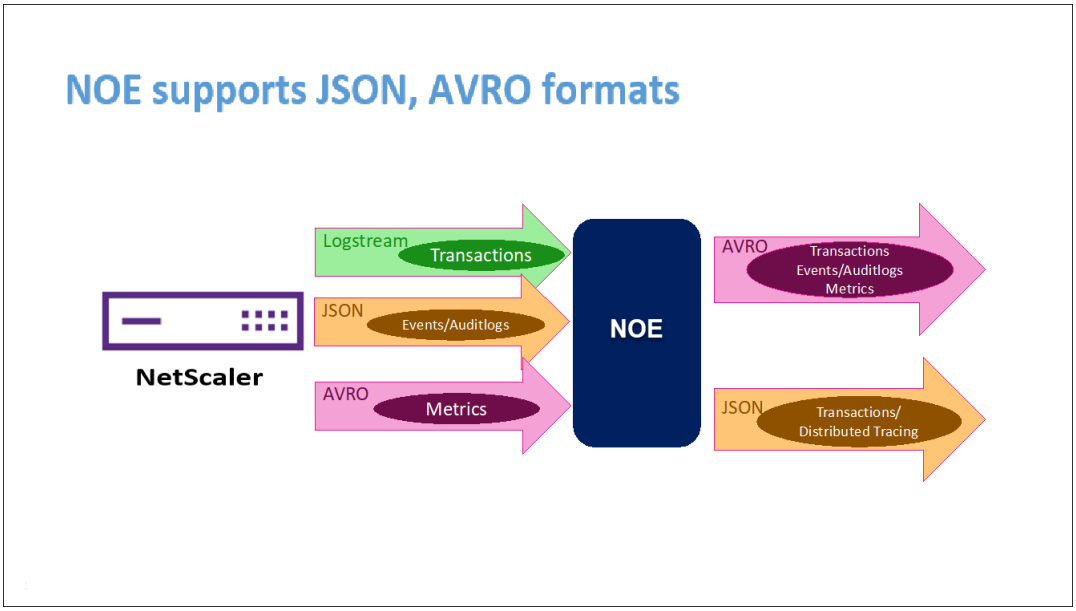
COE の詳細については、COE GitHubを参照してください。

Elasticsearch をトランザクションエンドポイントとする **COE**



Elasticsearch がトランザクションエンドポイントとして指定されている場合、NetScaler オブザーバビリティエクスポーターはデータを JSON 形式に変換します。Elasticsearch サーバーでは、NetScaler オブザーバビリティ

エクスポーターが各 ADC に対して 1 時間ごとに Elasticsearch インデックスを作成します。これらのインデックスは、データ、時間、ADC の UUID、および HTTP データのタイプ (http_event または http_error) に基づいています。次に、NetScaler オブザーバビリティエクスポーターは、各 ADC の Elastic 検索インデックスに JSON 形式でデータをアップロードします。通常のトランザクションはすべて http_event インデックスに配置され、異常は http_error インデックスに配置されます。



Zipkin による分散トレースのサポート

マイクロサービスアーキテクチャでは、1 つのエンドユーザー要求が複数のマイクロサービスにまたがる場合があるため、トランザクションの追跡やエラーの原因の修正が困難になります。このような場合、従来のパフォーマンス監視方法では、障害が発生した場所やパフォーマンスの低下の原因を正確に特定することはできません。リクエストを処理する各マイクロサービスに固有のデータポイントをキャプチャし、分析して有意義なインサイトを得る方法が必要です。

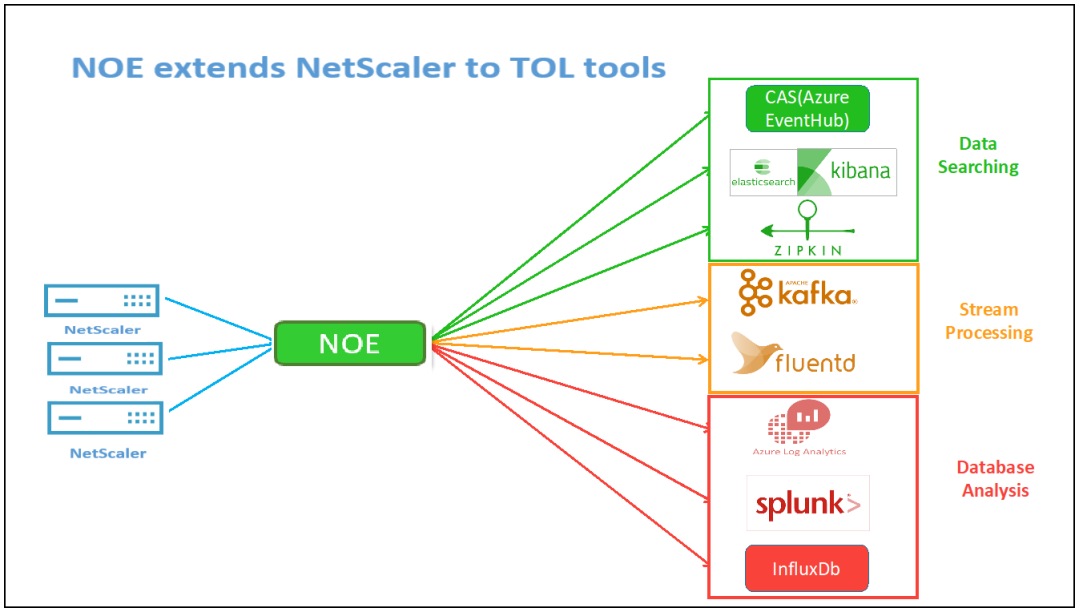
分散トレーシングは、トランザクションをエンドツーエンドで追跡し、複数のマイクロサービスにわたってトランザクションがどのように処理されているかを理解する方法を提供することで、この課題に対処します。

OpenTracing は、ディストリビューティッド（分散）トレーシングを設計および実装するための API の仕様および標準セットです 。分散トレーサを使用すると、マイクロサービス間のデータフローを視覚化し、マイクロサービスアーキテクチャのボトルネックを特定するのに役立ちます。

NetScaler オブザーバビリティエクスポーターは、NetScaler の分散トレースを実装しており、現在、分散トレーサーとして Zipkin をサポートしています 。

現在、NetScaler を使用してアプリケーションレベルでパフォーマンスを監視できます。NetScaler オブザーバビリティエクスポーターを NetScaler とともに使用すると、NetScaler CPX、MPX、または VPX によってプロキシされた各アプリケーションのマイクロサービスのトレースデータを取得できます。

はじめに、GitHub オブザーバビリティエクスポーターを参照してください。

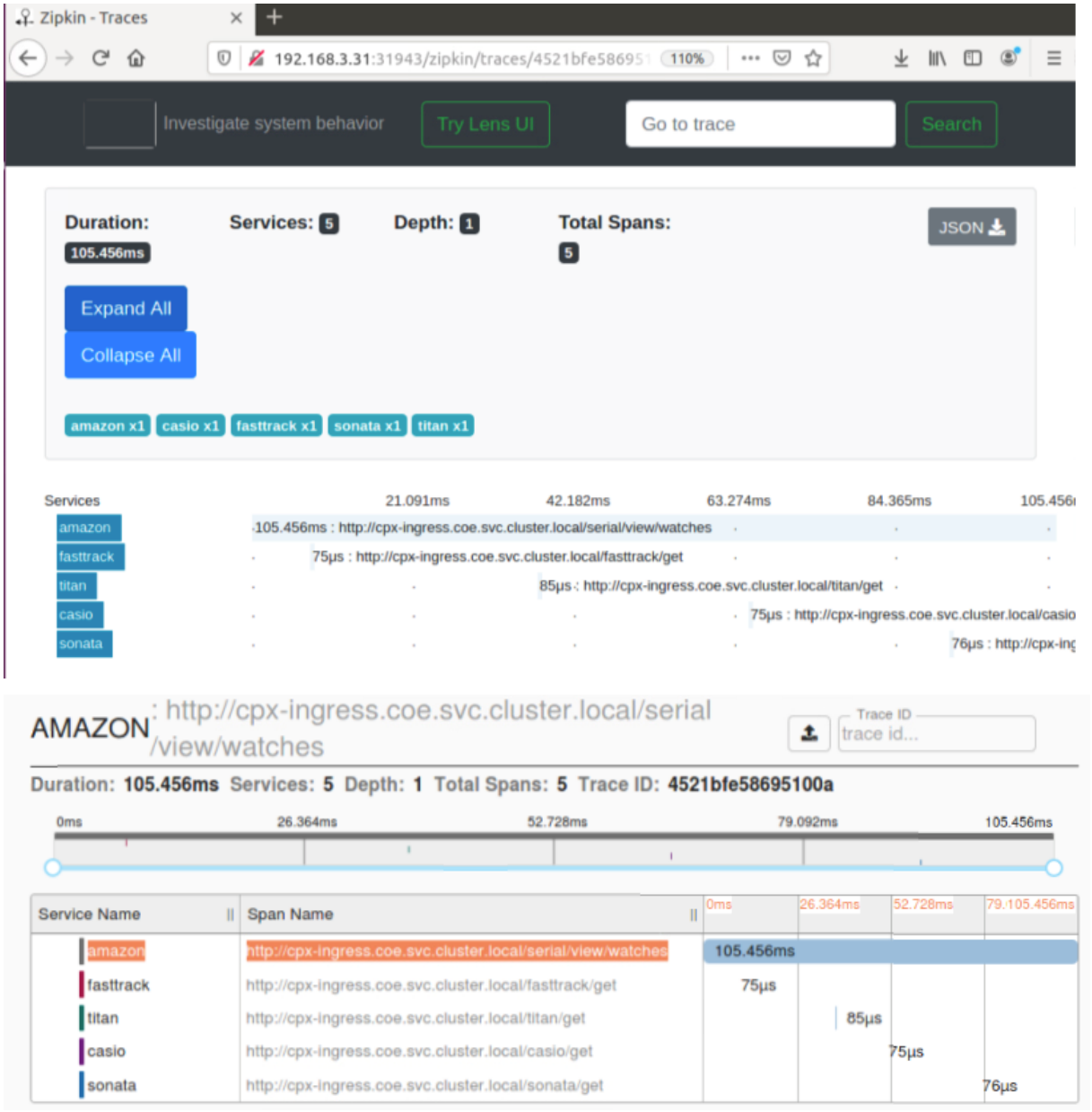


アプリケーションデバッグ用の **Zipkin**

Zipkin は、[Google の Dapper の論文に基づいたオープンソースの分散トレースシステムです](https://storage.googleapis.com/pub-tools-public-publication-data/pdf/36356.pdf)%E3%80%82Dapper%E3%81%AF%E3%80%81%E6%9C%AC%E7%95%AA%E7%92%B0%E5%A2%83%E3%81%A7%E3%81%AE%E5%88%86%E6%95%A3%E3%83%88%E3%83%AC%E3%83%BC%E3%82%B9%E3%81%AE%E3%81%9F%E3%82%81%E3%81%AEGoogle%E3%81%AE%E3%82%B7%E3%82%B9%E3%83%86%E3%83%A0%E3%81%A7%E3%81%99%E3%80%82Google%E3%81%AF%E3%81%93%E3%82%8C%E3%82%92%E5%BD%BC%E3%82%89%E3%81%AE%E8%AB%96%E6%96%87%E3%80%8CGoogle%E3%81%AE%E9%96%8B%E7%99%BA%E8%80%85%E3%81%AB%E8%A4%87%E9%9B%91%E3%81%AA%E5%88%86%E6%95%A3%E3%82%B7%E3%82%B9%E3%83%86%E3%83%A0%E3%81%AE%E6%8C%AF%E3%82%8B%E8%88%9E%E3%81%84%E3%81%AB%E9%96%A2%E3%81%99%E3%82%8B%E3%82%88%E3%82%8A%E5%A4%9A%E3%81%8F%E3%81%AE%E6%83%85%E5%A0%B1%E3%82%92%E6%8F%90%E4%BE%9B%E3%81%99%E3%82%8B%E3%81%9F%E3%82%81%E3%81%ABDapperを構築した」で説明している。トラブルシューティングを行う場合、特にシステムが複雑で分散している場合には、さまざまな角度からシステムを観察することが重要です。

次の Zipkin トレースデータは、Watches サンプルアプリケーションに関連する合計 5 つのスパンと 5 つのサービスを識別します。トレースデータには、5 つのマイクロサービスにまたがる特定のスパンデータが表示されます。

開始するには、Zipkinを参照してください。



最初のページ読み込みリクエストのアプリケーションレイテンシーを示す Zipkin スパンの例:

Services: amazon

Date Time	Relative Time	Annotation	Address
7/15/2020, 2:14:24 PM		Server Start	10.10.235.179:1719 (amazon)
7/15/2020, 2:14:24 PM	105.456ms	Server Finish	10.10.235.179:1719 (amazon)

Key	Value
component	py_zipkin
http.host	amazon:1719
http.method	GET
http.path	/serial/view/watches
http.url	http://cpx-ingress.coe.svc.cluster.local/serial/view/watches
Local Component	amazon
peer.address	10.10.235.190

データを見るための **Kibana**

Kibana は、Elasticsearch データを視覚化して Elastic Stack をナビゲートできるオープンなユーザーインターフェイスです。クエリのロードの追跡から、アプリ内でのリクエストの流れの把握まで、あらゆることを行えます。

アナリストでも管理者でも、Kibana は次の 3 つの重要な機能を提供することで、データをアクション可能にします。

- オープンソースの分析および可視化プラットフォーム。Kibana を使用して Elasticsearch データを探索し、美しいビジュアライゼーションとダッシュボードを構築できます。
- **Elastic** スタックを管理するための **UI**。セキュリティ設定の管理、ユーザーロールの割り当て、スナップショットの作成、データのロールアップなど、すべて Kibana UI から実行できます。
- **Elastic** のソリューションの一元化されたハブ。ログ分析からドキュメント検出、SIEM に至るまで、Kibana はこれらの機能やその他の機能にアクセスするためのポータルです。

Kibana は、Elasticsearch をデータソースとして使用するように設計されています。Elasticsearch は、Kibana を一番上に置き、データを保存して処理するエンジンだと考えてください。

Kibana はホームページから、データを追加するための次のオプションを提供します。

- ファイルデータビジュアライザーを使用してデータをインポートします。
- 組み込みのチュートリアルを使用して、Elasticsearch へのデータフローをセットアップします。データに関するチュートリアルがない場合は、Beats の概要に移動して、Beats ファミリーの他のデータシッパーについて学習します。
- サンプルデータセットを追加して、自分でデータを読み込まなくても Kibana を試乗できます。
- REST APIまたはクライアントライブラリを使用して、Elasticsearch にデータのインデックスを作成します。

Kibana はインデックスパターンを使用して 、どの Elasticsearch インデックスを調べるかを指示します。ファイルをアップロードしたり、組み込みのチュートリアルを実行したり、サンプルデータを追加したりすると、無料でインデックスパターンが得られ、探索を開始するのに適しています。独自のデータをロードする場合は、Stack Managementでインデックスパターンを作成できます。

ステップ 1: Logstash のインデックスパターンを構成する

ステップ 2: インデックスを選択し、入力するトラフィックを生成します。

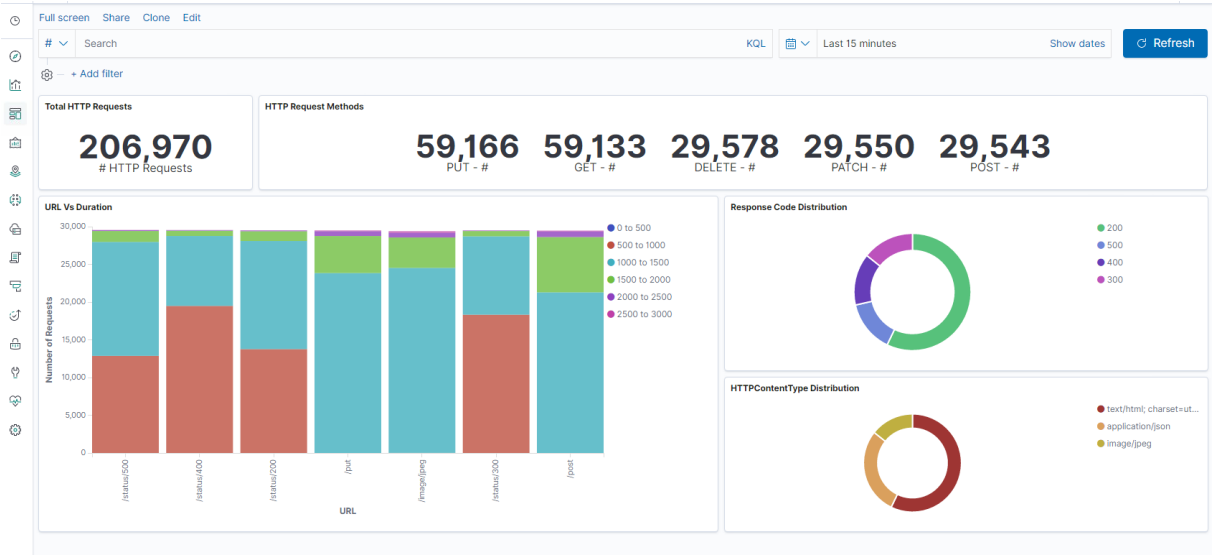
ステップ 3: ログフィードの非構造化データからアプリケーションを生成します。

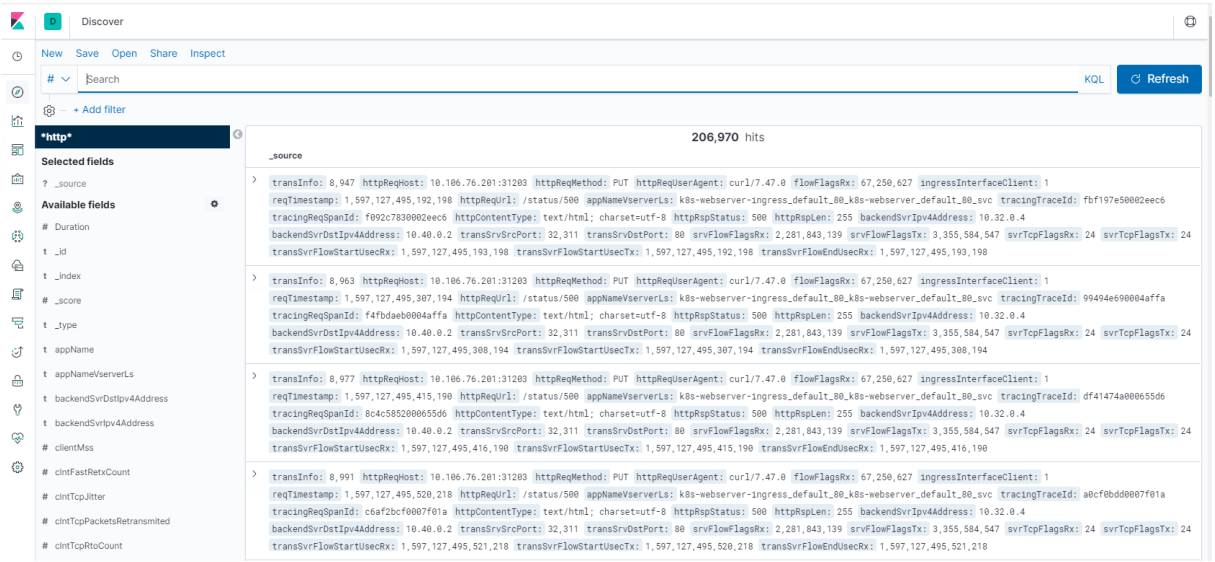
ステップ 4: Kibana は Logstash 入力をフォーマットしてレポートとダッシュボードを作成します。

- 時間範囲
- 表形式表示
- ヒット数はアプリケーションに基づきます。
  - 時刻 IP、エージェント、マシン.OS、レスポンスコード (200)、URL
  - 値によるフィルタリング

ステップ 5: 集計レポートでデータを視覚化します。

- チャート・レポート (円、グラフなど) での結果集計





NetScaler VPX インスタンスを展開する

March 20, 2024

> 注
>
> NetScaler または NetScaler Gateway をリリース 13.0 ビルド 61.xx 以降にインストールまたはアップグレードすると、NetScaler コンソールのサービス接続はデフォルトで有効になります。詳しくは、「データガバナンスと NetScalerConsole サービス接続」を参照してください。

NetScaler VPX 製品は、さまざまな仮想化およびクラウドプラットフォームでホストできる仮想アプライアンスです：

- Citrix Hypervisor
- VMware ESX
- Microsoft Hyper-V
- Linux KVM
- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform

詳細については、NetScaler VPX のデータシートを参照してください。

SDX アプライアンスでの NetScaler VPX インスタンスのプロビジョニングの詳細については、「NetScaler インスタンスのプロビジョニング」を参照してください。

NetScaler VPX 向け NetScalerApplication Delivery Management

NetScaler Application Delivery Management ソフトウェアは、管理者が企業全体の可視性を実現し、複数のインスタンスで実行する必要がある管理ジョブを自動化することで、運用を簡素化する一元管理ソリューションです。

NetScaler VPX インスタンスは、NetScaler Gateway、NetScaler SDX、NetScaler CPX、Citrix SD-WAN などの他のネットスケーラー製品に加えて管理および監視できます。Application Delivery Management ソフトウェアを使用すると、単一の統合コンソールからグローバルなアプリケーション配信インフラストラクチャ全体を管理、監視、トラブルシューティングできます。

詳細については、NetScaler Application Delivery Management ドキュメントを参照してください。

サポート・マトリックスと使用ガイドライン

March 21, 2024

このドキュメントでは、NetScaler VPX インスタンスでサポートされているさまざまなハイパーバイザーと機能について説明します。また、使用上のガイドラインと既知の制限事項についても説明します。

Citrix Hypervisor 上の VPX インスタンス

Citrix Hypervisor のバージョン	システム ID	VPX モデル
8.2 は 13.0 64.x 以降をサポート、8.0、7.6、7.1	450000	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G

VMware ESXi ハイパーバイザー上の **VPX** インスタンス

450010（システム ID）を備えた以下の VPX モデルは、表に記載されている VMware ESX のバージョンをサポートします。

VPX モデル**:VPX** 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G。

ESXi バージョン	ESXi のリリース日 (年/月/日)	ESXi ビルド番号	NetScaler VPX バージョン
ESXi 8.0 アップデート 2	2023/09/21	22380479	13.1-52.x およびそれ以降のビルド
ESXi 8.0 アップデート 1	2023/04/18	21495797	13.1-45.x およびそれ以降のビルド
ESXi 8.0c	2023/03/30	21493926	13.1-45.x およびそれ以降のビルド
ESXi 8.0	2022/10/11	20513097	13.1-42.x およびそれ以降のビルド
ESXi 7.0 アップデート 3	2023/09/28	22348816	13.1-51.x およびそれ以降のビルド
ESXi 7.0 アップデート 3	2023/07/06	21930508	13.1-49.x およびそれ以降のビルド
ESXi 7.0 アップデート 3 分	2023/05/03	21686933	13.1-48.x およびそれ以降のビルド
ESXi 7.0 アップデート i3	2022/12/08	20842708	13.1-37.x およびそれ以降のビルド
ESXi 7.0 アップデート 3f	2022/07/12	20036589	13.1-33.x およびそれ以降のビルド
ESXi 7.0 アップデート 3D	2022/03/29	19482537	13.1-27.x およびそれ以降のビルド
ESXi 7.0 アップデート 3c	2022/01/27	19193900	13.1-21.x およびそれ以降のビルド

ESXi バージョン	ESXi のリリース日 (年/月/日)	ESXi ビルド番号	NetScaler VPX バージョン
ESX 7.0 アップデート 2d	2021/09/14	18538813	13.1-9.x およびそれ以降のビルド
ESX 7.0 アップデート 2a	2021/04/29	17867351	13.1-4.x およびそれ以降のビルド

> 注:
>
> 各 ESXi パッチサポートは、前の表で指定されている NetScaler VPX バージョンで検証されており、NetScaler VPX 13.1 バージョンのすべての上位ビルドに適用されます。

Microsoft Hyper-V 上の VPX インスタンス

Hyper-V 版	システム ID	VPX モデル
2016, 2019	450020	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000

Nutanix AHV の VPX インスタンス

NetScaler VPX は、Citrix Ready パートナーシップを通じて Nutanix AHV でサポートされています。Citrix Ready は、ソフトウェアおよびハードウェアのベンダーが自社製品を開発し、デジタルワークスペース、ネットワーキング、および分析用の NetScaler テクノロジーと統合するのを支援するテクノロジーパートナープログラムです。

NetScaler VPX インスタンスを Nutanix AHV にデプロイする段階的な方法の詳細については、「Nutanix AHV への NetScaler VPX デプロイ」を参照してください。

サードパーティサポート**:**

NetScaler 環境での特定のサードパーティ（Nutanix AHV）の統合で問題が発生した場合は、サードパーティパートナー（Nutanix）に直接サポートインシデントをオープンしてください。

パートナーが問題が NetScaler にあると判断した場合、パートナーは NetScaler サポートに連絡してさらにサポートを受けることができます。問題が解決するまで、パートナーからの専任技術リソースが NetScaler サポートと協力します。

詳しくは、「Citrix Ready パートナープログラムに関するよくある質問」を参照してください。

汎用 KVM 上の VPX インスタンス

汎用 KVM バージョン	システム ID	VPX モデル
RHEL 7.4、RHEL 7.5 (NetScaler バージョン 12.1 50.x 以降)、RHEL 7.6、RHEL 8.0、Ubuntu 16.04、Ubuntu 18.04、RHV 4.2	450070	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G。VPX 25G, VPX 40G, VPX 100G

注意事項:

KVM ハイパーバイザーを使用するときは、次の点を考慮してください。

- VPX インスタンスは、表 1-4 に記載されている Hypervisor リリースバージョンに対して認定されており、バージョン内のパッチリリースには適していません。ただし、VPX インスタンスは、サポートされているバージョンのパッチリリースとシームレスに動作することが期待されます。そうでない場合は、トラブルシューティングとデバッグのためのサポートケースを記録します。
- RHEL 7.6 を使用する前に、KVM ホストで以下のステップを完了します。
  1. /etc/default/grub を編集して`"kvm_intel.preemption_timer=0"`を`GRUB_CMDLINE_LINUX`変数に追加します。
  2. コマンド`"# grub2-mkconfig -o /boot/grub2/grub.cfg"`で grub.cfg を再生成します。
  3. ホストマシンを再起動します。
- Ubuntu 18.04 を使用する前に、KVM ホストで以下のステップを完了してください。
  1. /etc/default/grub を編集して`"kvm_intel.preemption_timer=0"`を`GRUB_CMDLINE_LINUX`変数に追加します。
  2. コマンド`"# grub-mkconfig -o /boot/grub/grub.cfg "`で grub.cfg を再生成します。
  3. ホストマシンを再起動します。

AWS 上の **VPX** インスタンス

AWS バージョン	システム ID	VPX モデル
-	450040	VPX 10、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX BYOL、VPX 8000、VPX 10G、VPX 15G、VPX 25G は、EC2 インスタンスタイプ（C5、M5、および C5n）での BYOL でのみ使用できます。

> 注:
>
> VPX 25G オファリングは、AWS では 25G のスループットを提供しませんが、VPX 15G オファリングと比較してより高い SSL トランザクションレートを提供できます。

Azure 上の **VPX** インスタンス

Azure バージョン	システム ID	VPX モデル
-	450020	VPX 10、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX BYOL

VPX 機能マトリックス

	VPX on XenServer		VPX on VMware ESX				VPX on Microsoft Hyper-V	VPX on generic KVM			VPX on AWS	VPX on Azure	VPX on GCP
Features	**PV**	**SR-IOV**	**PV**	**SR-IOV**	**Emulated**	**PCI Passthrough**	**PV**	**PV**	**SR-IOV**	**PCI Passthrough**			
Multi-PE Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Clustering Support	Yes	Yes[1]	Yes	Yes[1]	Yes	Yes	Yes	Yes	Yes[1]	Yes	No	No	No
VLAN Tagging	Yes	Yes	Yes	Yes	Yes	Yes	Yes (only on 2012R2)	Yes	Yes	Yes	No	No	No
Detecting Link Events	No[2]	Yes[3]	No[2]	Yes[3]	No[2]	Yes[3]	No[2]	No[2]	Yes[3]	Yes[3]	No[2]	No[2]	No[2]
Interface Parameter Configuration	No	No	No	No	No	Yes	No	No	No	Yes	No	No	No
Static LA	Yes[2]	Yes[3]	Yes[2]	No	Yes[2]	Yes[3]	Yes[2]	Yes[2]	Yes[3]	Yes[3]	No	No	No
LACP	No	Yes[3]	Yes[2]	No	Yes[2]	Yes[3]	No	Yes[2]	Yes[3]	Yes[3]	No	No	No
Static CLAG	No	No	No	No	No	No	No	No	No	No	No	No	No
LACP CLAG	No	No	Yes[2]	No	Yes[2]	Yes[3]	No	Yes[2]	Yes[3]	Yes[3]	No	No	No
Hot-plug	No	No	No	No	No	No	No	No	No	No	Yes	No	No

前の表で使用されている上付き番号 (1、2、3) は、それぞれの番号付けで次の点を指します。

1. SRIOV では、バックプレーンではなく、クライアント側およびサーバ側インターフェイス用のクラスタリングサポートを利用できます。
2. インターフェイスダウンイベントは NetScaler VPX インスタンスには記録されません。
3. スタティック LA の場合、物理ステータスが DOWN のインターフェイスでトラフィックが送信される場合もあります。
4. LACP の場合、ピアデバイスは LACP タイムアウトメカニズムに基づいてインターフェイス DOWN イベントを認識します。
   - 短いタイムアウト：3 秒
   - 長いタイムアウト：90 秒
5. LACP では、VM 間でインターフェイスを共有しないでください。
6. ダイナミックルーティングの場合、リンクイベントが検出されないため、コンバージェンス時間はルーティングプロトコルによって異なります。
7. モニタ対象スタティックルート機能は、ルータの状態が VLAN ステータスに依存するため、モニタをスタティックルートにバインドしないと失敗します。VLAN ステータスは、リンクステータスによって異なります。
8. リンク障害がある場合、高可用性では部分的な障害検出は行われません。リンク障害があると、高可用性の分割脳の状態が発生する可能性があります。
   - VPX インスタンスからリンクイベント（無効/有効化、リセット）が生成された場合、リンクの物理ステータスは変わりません。静的 LA の場合、ピアによって開始されたトラフィックはすべてインスタンスでドロップされます。
   - VLAN タギング機能を機能させるには、次の手順を実行します。

   VMware ESX で、VMware ESX サーバの vSwitch でポートグループの VLAN ID を 1 ～4095 に設定します。VMware ESX サーバの vSwitch での VLAN ID の設定の詳細については、「VMware ESX Server 3 802.1Q VLAN ソリューション」を参照してください。

サポートされているブラウザー

オペレーティングシステム	ブラウザとバージョン
Windows 7	Internet Explorer-8, 9, 10, 11; Mozilla Firefox 3.6.25 以降; Google Chrome-15 以降
Windows 64 ビット	Internet Explorer-8、9; Google Chrome-15 以降
MAC	Mozilla Firefox-12 以降; Safari-5.1.3; Google Chrome-15 以降

VPX インスタンスの AMD プロセッササポート

NetScaler リリース 13.1 以降、VPX インスタンスは Intel プロセッサーと AMD プロセッサーの両方をサポートしています。VPX 仮想アプライアンスは、2 つ以上の仮想コアと 2 GB を超えるメモリを備えた任意のインスタンスタイプにデプロイできます。システム要件の詳細については、NetScaler VPX のデータシートを参照してください。

VPX プラットフォームと NIC マトリックステーブル

次の表は、VPX プラットフォームまたはクラウドでサポートされている NIC の一覧です。

	Mellanox CX-3	Mellanox CX-4	Mellanox CX-5	Intel 82599 SRIOV VF	Intel X710/X722/XL710 SRIOV VF	Intel X710/XL710 PCI-Passthrough Mode
VPX (ESXi)	いいえ	はい	いいえ	はい	いいえ	はい
VPX (Citrix Hypervisor)	-	-	-	はい	はい	いいえ
VPX (KVM)	いいえ	はい	はい	はい	はい	はい
VPX (Hyper-V)	-	-	-	いいえ	いいえ	いいえ
VPX (AWS)	-	-	-	はい	-	-
VPX (Azure)	はい	はい	はい	-	-	-
VPX (GCP)	-	-	-	-	-	-

使用ガイドライン

使用上のガイドラインに従ってください。

- VPX インスタンスは、サーバーのローカルディスクまたは SAN ベースのストレージボリュームにデプロイすることをお勧めします。

『**VMware vSphere 6.5** のパフォーマンスのベストプラクティス』ドキュメントの「**VMware ESXi CPU** に関する考慮事項」セクションを参照してください。ここに抽出があります:

- CPU/メモリの需要が高い仮想マシンを、オーバーコミットの多いホストまたはクラスターに配置することはお勧めしません。

- ほとんどの環境では、ESXi は、仮想マシンのパフォーマンスに影響を与えることなく、かなりのレベルの CPU オーバーコミットメントを許可します。ホストでは、そのホスト内の物理プロセッサコアの総数よりも多くの vCPU を実行できます。
- ESXi ホストが CPU 飽和状態になった場合、つまり、仮想マシンおよびホスト上のその他の負荷がホストにあるすべての CPU リソースを要求すると、レイテンシの影響を受けやすいワークロードがうまく動作しない可能性があります。この場合、一部の仮想マシンをパワーオフしたり、仮想マシンを別のホストに移行したりする (または DRS による自動移行を許可する) など、CPU の負荷を軽減することができます。
- 仮想マシンで ESXi Hypervisor の最新の機能セットを利用するには、最新のハードウェア互換性バージョンを使用することをお勧めします。ハードウェアと ESXi バージョンの互換性の詳細については、VMware のドキュメントを参照してください。
- NetScaler VPX は、レイテンシーに敏感で高性能な仮想アプライアンスです。期待されるパフォーマンスを実現するには、アプライアンスに vCPU 予約、メモリ予約、ホストでの vCPU ピン接続が必要です。また、ホスト上でハイパースレッディングを無効にする必要があります。ホストがこれらの要件を満たさない場合、高可用性フェイルオーバー、VPX インスタンス内の CPU スパイク、VPX CLI へのアクセスにおける低速化、ピットボスデーモンのクラッシュ、パケットドロップ、低スループットなどの問題が発生します。

Hypervisor は、次の 2 つの条件のいずれかが満たされると、過剰プロビジョニングと見なされます。

- ホストにプロビジョニングされた仮想コア (vCPU) の総数が、物理コア (pCPU) の総数を超えています。
- プロビジョニングされた仮想マシンの合計数は、pCPU の合計数よりも多くの vCPU を消費します。

  インスタンスが過剰プロビジョニングされている場合、ハイパーバイザーのスケジューリングオーバーヘッド、バグ、またはハイパーバイザーの制限により、ハイパーバイザーがインスタンスのリザーブドリソース（CPU、メモリなど）を保証しない場合があります。この動作により、NetScaler CPU リソースが不足し、使用ガイドラインの最初のポイントで説明されている問題が発生する可能性があります。管理者は、ホスト上でプロビジョニングされる vCPU の総数が pCPU の総数より少なくなるように、ホストのテナント数を減らすことをお勧めします。

  例

  ESX ハイパーバイザーの場合、`esxtop`コマンド出力で VPX vCPU の%RDY%パラメータが 0 より大きい場合、ESX ホストにスケジューリングオーバーヘッドがあると言われ、VPX インスタンスのレイテンシー関連の問題が発生する可能性があります。

  このような状況では、%RDY%が常に 0 に戻るように、ホストのテナンシーを減らします。または、ハイパーバイザーベンダーに連絡して、リソース予約が完了していない理由をトリアージします。

- ホットアドは、AWS 上の NetScaler を使用する PV および SRIOV インターフェイスでのみサポートされます。ENA インターフェイスを持つ VPX インスタンスはホットプラグをサポートしていないため、ホットプラグを試みるとインスタンスの動作が予測できない場合があります。
- AWS ウェブコンソールまたは AWS CLI インターフェイスを介したホット削除は、NetScaler の PV、SRIOV、および ENA インターフェイスではサポートされていません。ホット削除を試みると、インスタンスの動作が予測できなくなる可能性があります。

パケットエンジンの **CPU** 使用率を制御するコマンド

ハイパーバイザーおよびクラウド環境における VPX インスタンスのパケットエンジン（非管理）CPU 使用率の動作を制御するには、2 つのコマンド（`set ns vpxparam`および`show ns vpxparam`）を使用できます。

- `set ns vpxparam [-cpuyield (YES | NO | DEFAULT)] [-masterclockcpu1 (YES | NO)]`

  各 VM が、別の VM に割り当てられているが、使用されていない CPU リソースの使用を許可します。

  `Set ns vpxparam` パラメータ:

  -cpuyield: 割り当てられているが未使用の CPU リソースを解放または解放しません。

  - はい: 割り当てられているが未使用の CPU リソースを別の VM で使用できるようにします。
  - いいえ: 割り当てられている VM のすべての CPU リソースを予約します。このオプションは、ハイパーバイザーおよびクラウド環境で VPX CPU 使用率が高い割合を示します。
  - デフォルト: いいえ。

  > 注:
  >
  > すべての NetScaler VPX プラットフォームで、ホストシステム上の vCPU 使用率は 100% です。`set ns vpxparam -cpuyield YES`コマンドを入力して、この使用方法を上書きします。

  クラスタノードを「yield」に設定する場合は、CCO で次の追加設定を実行する必要があります。

  - クラスタが形成されると、すべてのノードに「yield=Default」が表示されます。
  - すでに「yield=Yes」に設定されたノードを使用してクラスターが形成されている場合、ノードは「DEFAULT」のイールドを使用してクラスターに追加されます。

  > 注:
  >
  > クラスタノードを「yield=YES」に設定する場合は、クラスタの形成後にのみ構成でき、クラスタが形成される前には設定できません。

  -masterclockcpu1: メインクロックソースを CPU0 (管理 CPU) から CPU1 に移動できます。このパラメータには、次のオプションがあります。

  - はい： 仮想マシンがメインクロックソースを CPU0 から CPU1 に移動できるようにします。
  - いいえ:VM はメインクロックソースに CPU0 を使用します。デフォルトでは、CPU0 がメインクロックソースです。

- `show ns vpxparam`

  現在の`vpxparam`設定を表示します。

その他の参考文献

- Citrix Ready 製品については、Citrix Ready Marketplaceにアクセスしてください
- Citrix Ready 製品サポートについては、よくある質問ページを参照してください。
- VMware ESX ハードウェアバージョンについては、VMware Tools のアップグレードを参照してください。

VMware ESX、Linux KVM、および Citrix Hypervisor で NetScaler ADC VPX のパフォーマンスを最適化する

August 29, 2023

NetScaler VPX のパフォーマンスは、ハイパーバイザー、割り当てられたシステムリソース、およびホスト構成によって大きく異なります。望ましいパフォーマンスを達成するには、まず VPX データシートの推奨事項に従ってから、このドキュメントに記載されているベストプラクティスを使用してさらに最適化します。

VMware ESX ハイパーバイザー上の NetScaler ADC VPX インスタンス

このセクションでは、構成可能なオプションと設定、および VMware ESX ハイパーバイザー上の NetScaler ADC VPX インスタンスの最適なパフォーマンスを実現するのに役立つその他の推奨事項について説明します。

- ESX ホストでの推奨構成
- E1000 ネットワークインターフェイスを備えた NetScaler ADC VPX
- VMXNET3 ネットワークインターフェイスを備えた NetScaler ADC VPX
- SR-IOV および PCI パススルーネットワークインターフェイスを備えた NetScaler ADC VPX

ESX ホストでの推奨構成

E1000、VMXNET3、SR-IOV、および PCI パススルーネットワークインターフェイスを備えた VPX で高いパフォーマンスを実現するには、次の推奨事項に従ってください。

- ESX ホストでプロビジョニングされる仮想 CPU (vCPU) の総数は、ESX ホストの物理 CPU (pCPU) の総数以下である必要があります。
- ESX ホストで良好な結果を得るには、非均一メモリアクセス (NUMA) アフィニティと CPU アフィニティを設定する必要があります。

  —Vmnic の NUMA アフィニティを見つけるには、ローカルまたはリモートでホストにログインし、次のように入力します。


```
#vsish -e get /net/pNics/vmnic7/properties | grep NUMA
Device NUMA Node: 0
<!--NeedCopy-->
```

- 仮想マシンの NUMA および vCPU アフィニティを設定するには、VMware のドキュメントを参照してください。

E1000 ネットワークインターフェイスを備えた **NetScaler ADC VPX**

VMware ESX ホストで次の設定を実行します。

- VMware ESX ホストで、1 つの物理 vSwitch から 2 つの vNIC を作成します。複数の vNIC により、ESX ホストに複数の Rx スレッドが作成されます。これにより、物理 NIC インターフェイスの Rx スループットが向上します。
- 作成した各 vNIC の vSwitch ポートグループレベルで VLAN を有効にします。
- vNIC 送信（Tx）スループットを向上させるには、vNIC ごとに ESX ホストで別の Tx スレッドを使用します。次の ESX コマンドを使用します。
  - ESX バージョン 5.5 の場合:

    ```
    esxcli system settings advanced set  -o /Net/NetTxWorldlet  -
        i
    <!--NeedCopy-->
    ```

  - ESX バージョン 6.0 以降の場合:

    ```
    esxcli system settings advanced set -o /Net/NetVMTxType  -i 1
    <!--NeedCopy-->
    ```

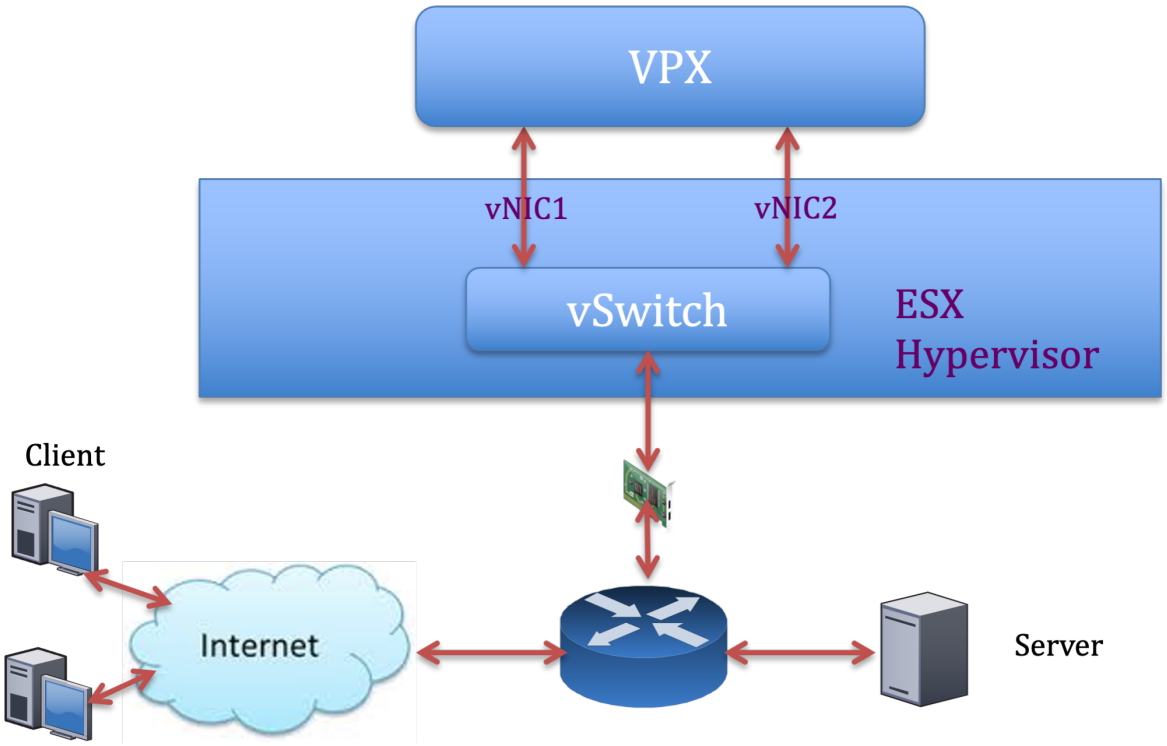
- vNIC Tx スループットをさらに高めるには、別の Tx 完了スレッドと、デバイス（NIC）キューごとの Rx スレッドを使用します。次の ESX コマンドを使用します。

```
esxcli system settings advanced set -o /Net/
   NetNetqRxQueueFeatPairEnable -i 0
<!--NeedCopy-->
```

> 注:
>
> VMware ESX ホストを再起動して、更新された設定を適用してください。

物理 **NIC** 展開ごとに **2** つの **vNIC**

次に、より優れたネットワークパフォーマンスを提供する展開の **pNIC** ごとに **2** つの **vNIC** モデルのトポロジおよび設定コマンドの例を示します。



NetScaler VPX 構成例:

前のサンプルトポロジに示した展開を実現するには、NetScaler VPX インスタンスで次の構成を実行します。

- クライアント側で、SNIP（1.1.1.2）をネットワークインターフェイス 1/1 にバインドし、VLAN タグモードを有効にします。

```
bind vlan 2 -ifnum 1/1  - tagged
bind vlan 2 -IPAddress 1.1.1.2 255.255.255.0
<!--NeedCopy-->
```

- サーバ側で、SNIP（2.2.2.2）をネットワークインターフェイス 1/1 にバインドし、VLAN タグモードを有効にします。

```
bind vlan 3 -ifnum 1/2  - tagged
bind vlan 3 -IPAddress 2.2.2.2 255.255.255.0
<!--NeedCopy-->
```

- HTTP 仮想サーバー (1.1.1.100) を追加し、サービス (2.2.2.100) にバインドします。

```
add lb vserver v1 HTTP 1.1.1.100 80 -persistenceType NONE -
    Listenpolicy None -cltTimeout 180
add service s1 2.2.2.100 HTTP 80 -gslb NONE -maxClient 0 -maxReq
    0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
    180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
bind lb vserver v1 s1
<!--NeedCopy-->
```


> 注:
>
> ルートテーブルに次の 2 つのエントリが含まれていることを確認します。
>
> - 1.1.1.0/24 SNIP 1.1.1.2 を指すゲートウェイを持つサブネット
> - 2.2.2.0/24 SNIP 2.2.2.2 を指すゲートウェイを持つサブネット

VMXNET3 ネットワークインターフェイスを備えた **NetScaler ADC VPX**

VMXNET3 ネットワークインターフェイスを使用した VPX で高いパフォーマンスを実現するには、VMware ESX ホストで次の設定を行います。

- 1 つの物理 vSwitch から 2 つの vNIC を作成します。複数の vNIC により、ESX ホストに複数の Rx スレッドが作成されます。これにより、物理 NIC インターフェイスの Rx スループットが向上します。
- 作成した各 vNIC の vSwitch ポートグループレベルで VLAN を有効にします。
- vNIC 送信（Tx）スループットを向上させるには、vNIC ごとに ESX ホストで別の Tx スレッドを使用します。次の ESX コマンドを使用します。
  - ESX バージョン 5.5 の場合:

```
esxcli system settings advanced set  -o /Net/NetTxWorldlet  -i
<!--NeedCopy-->
```

  - ESX バージョン 6.0 以降の場合:

```
 esxcli system settings advanced set -o /Net/NetVMTxType  -i 1
 <!--NeedCopy-->
```

VMware ESX ホストで、次の構成を実行します。

- VMware ESX ホストで、1 つの物理 vSwitch から 2 つの vNIC を作成します。複数の vNIC により、ESX ホストに複数の Tx スレッドと Rx スレッドが作成されます。これにより、物理 NIC インターフェイスの Tx スループットと Rx スループットが向上します。
- 作成した各 vNIC の vSwitch ポートグループレベルで VLAN を有効にします。
- vNIC の Tx スループットを向上させるには、デバイス（NIC）キューごとの Tx 完了スレッドと受信スレッドを別々に使用します。次のコマンドを使用します:

```
esxcli system settings advanced set -o /Net/
   NetNetqRxQueueFeatPairEnable -i 0
<!--NeedCopy-->
```

- 仮想マシンの構成に次の設定を追加して、vNIC ごとに 1 つの送信スレッドを使用するように仮想マシンを設定します。


```
ethernetX.ctxPerDev = "1"
<!--NeedCopy-->
```

- 仮想マシンの構成に次の設定を追加して、vNIC あたり最大 8 つの送信スレッドを使用するように仮想マシンを構成します。

```
ethernetX.ctxPerDev = "3"
<!--NeedCopy-->
```

> 注:
>
> vNIC あたりの送信スレッド数を増やすと、ESX ホストでより多くの CPU リソース (最大 8 つ) が必要になります。前述の設定を行う前に、十分な CPU リソースが使用可能であることを確認してください。

詳細については、vSphere の Telco および NFV
ワークロードのパフォーマンスチューニングのベストプラクティスを参照してください。

> 注:
>
> VMware ESX ホストを再起動して、更新された設定を適用してください。

VMXNET3 は、物理 **NIC** 展開ごとに **2** つの **vNIC** として設定できます。詳細については、「物理 NIC 展開ごとに 2 つの vNIC」を参照してください。

VMware ESX で **VMXNET3** デバイス用のマルチキューと **RSS** サポートを設定します　デフォルトでは、VMXNET3 デバイスは 8 つの Rx キューと Tx キューのみをサポートします。VPX の vCPU の数が 8 を超えると、VMXNET3 インターフェイスに設定されている Rx キューと Tx キューの数は、デフォルトで 1 に切り替わります。ESX の特定の構成を変更することで、VMXNET3 デバイス用に最大 19 個の Rx キューと Tx キューを設定できます。このオプションにより、パフォーマンスが向上し、VPX インスタンスの vCPU 間でパケットが均一に分散されます。

> 注:
>
> NetScaler リリース 13.1 ビルド 48.x 以降、NetScaler VPX は VMXNET3 デバイスの ESX 上で最大 19 個の Rx キューと Tx キューをサポートします。

前提条件:

ESX で VMXNET3 デバイス用に最大 19 個の Rx キューと Tx キューを構成するには、次の前提条件が満たされていることを確認してください。

- NetScaler VPX バージョンは 13.1 ビルド 48.X 以降です。
- NetScaler VPX は、VMware ESX 7.0 以降でサポートされているハードウェアバージョン 17 以降の仮想マシンで構成されます。

8 つ以上の **Rx** キューと **Tx** キューをサポートするように **VMXNET3** インターフェイスを設定します。

1. 仮想マシンの構成ファイル (.vmx) ファイルを開きます。
2. `ethernetX.maxTxQueues`および`ethernetX.maxRxQueues`の値を設定して Rx キューと TX キューの数を指定します（X は設定する仮想 NIC の数）。設定するキューの最大数は、仮想マシンの vCPU 数を超えてはいけません。

   > 注:
   >
   > キューの数を増やすと、ESX ホストのプロセッサオーバーヘッドも増加します。したがって、キューを増やす前に、ESX ホストに十分な CPU リソースがあることを確認してください。キューの数がパフォーマンスのボトルネックになっている場合は、サポートされるキューの最大数を増やすことができます。このような場合は、キューの数を徐々に増やすことをお勧めします。たとえば、8 から 12、次に 16 へ、そして 20 へ、というようになります。最大値まで直接上げるのではなく、各設定でパフォーマンスを評価してください。

SR-IOV および **PCI** パススルーネットワークインターフェイスを備えた **NetScaler ADC VPX**

SR-IOV および PCI パススルーネットワークインターフェイスを備えた VPX で高いパフォーマンスを実現するには、「ESX ホストでの推奨構成」を参照してください。

Linux-KVM プラットフォーム上の **NetScaler ADC VPX** インスタンス

このセクションでは、構成可能なオプションと設定、および Linux-KVM プラットフォーム上の NetScaler ADC VPX インスタンスの最適なパフォーマンスを達成するのに役立つその他の推奨事項について説明します。

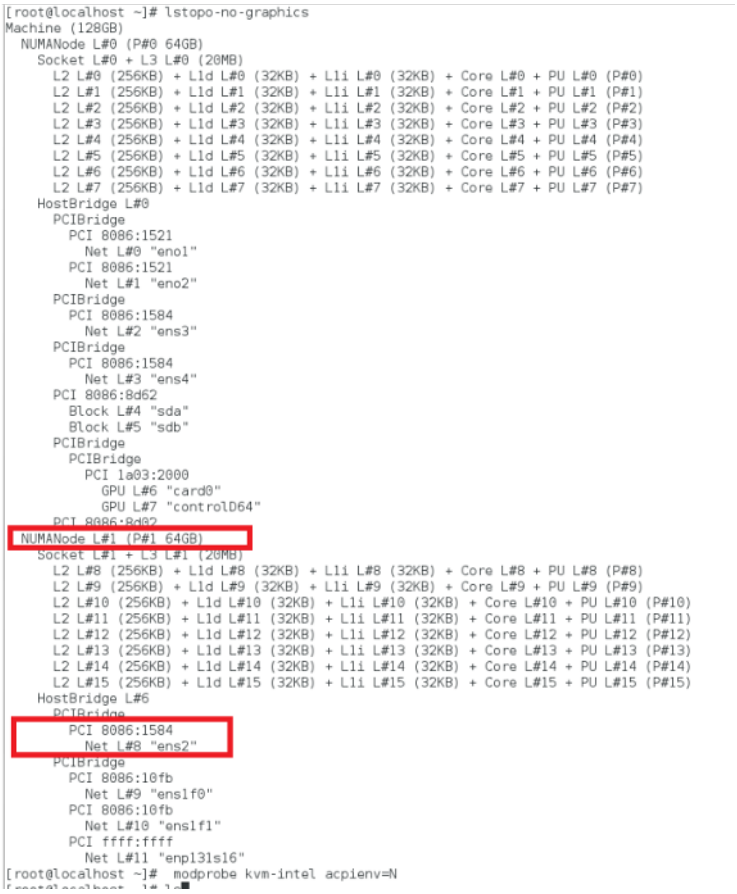
- KVM のパフォーマンス設定
- PV ネットワークインターフェイスを備えた NetScaler ADC VPX
- SR-IOV およびフォートビルの PCIe パススルーネットワークインターフェイスを備えた NetScaler ADC VPX

KVM のパフォーマンス設定

KVM ホストで次の設定を行います。

`lstopo`コマンドを使用して、**NIC** の **NUMA** ドメインを検索します。

VPX と CPU のメモリが同じ場所に固定されていることを確認します。
次の出力では、10G NIC「ens2」は NUMA ドメイン #1 に関連付けられています。



NUMA ドメインから **VPX** メモリを割り当てます。

numactlコマンドは、メモリの割り当て元の NUMA ドメインを示します。次の出力では、NUMA ノード #0 から約 10 GB の RAM が割り当てられています。

```
[root@localhost ~]# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 65429 MB
node 0 free: 55854 MB
node 1 cpus: 8 9 10 11 12 13 14 15
node 1 size: 65536 MB
node 1 free: 52388 MB
node distances:
node   0   1
  0:  10  21
  1:  21  10
[root@localhost ~]#
```

NUMA ノードマッピングを変更するには、次の手順に従います。

1. ホスト上の VPX の.xml を編集します。

```
/etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```


2. 次のタグを追加します。

```
<numatune>
<memory mode="strict" nodeset="1"/>    This is the NUMA domain
   name
</numatune>
<!--NeedCopy-->
```

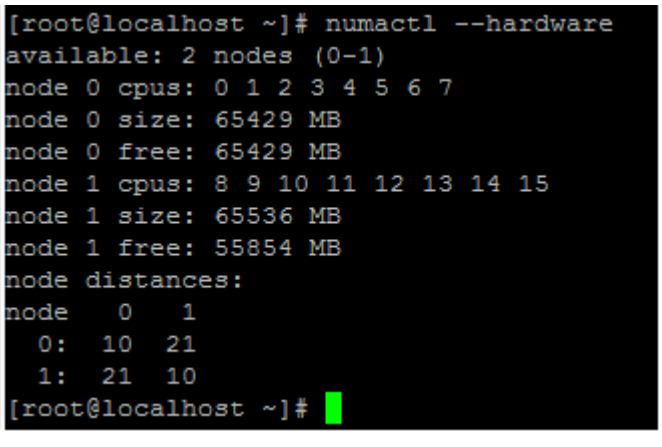
3. VPX をシャットダウンします。

4. 次のコマンドを実行します:

```
virsh define /etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```

このコマンドは、NUMA ノードマッピングを使用して VM の構成情報を更新します。

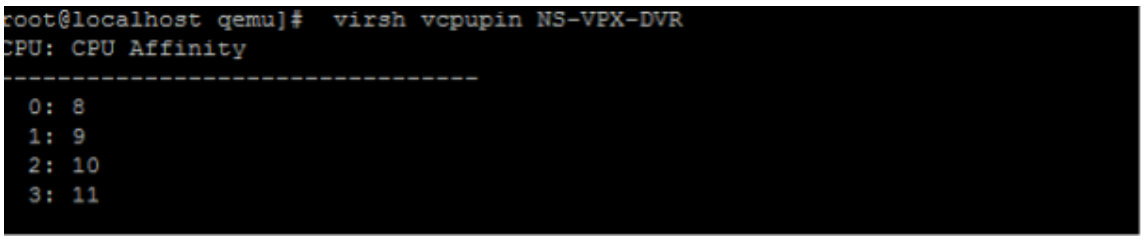
5. VPX の電源をオンにします。次に、ホスト上の`numactl -hardware`コマンド出力を確認して、VPX の更新されたメモリ割り当てを確認します。



VPX の **vCPU** を物理コアにピン留めします。

- VPX の vCPU から pCPU へのマッピングを表示するには、次のコマンドを入力します。

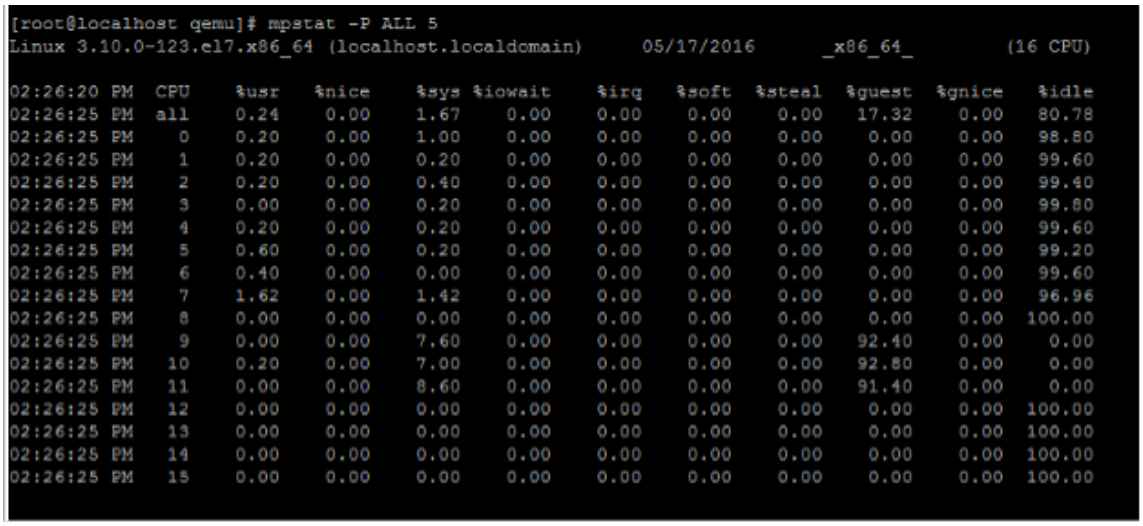
```
virsh vcpupin <VPX name>
<!--NeedCopy-->
```



vCPU 0 ～4 は物理コア 8 ～11 にマッピングされます。

- 現在の pCPU 使用率を表示するには、次のコマンドを入力します。


```
mpstat -P ALL 5
<!--NeedCopy-->
```



```
[root@localhost qemu]# mpstat -P ALL 5
Linux 3.10.0-123.el7.x86_64 (localhost.localdomain)     05/17/2016      _x86_64_        (16 CPU)

02:26:20 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
02:26:25 PM  all    0.24    0.00    1.67    0.00    0.00    0.00    0.00   17.32    0.00   80.78
02:26:25 PM    0    0.20    0.00    1.00    0.00    0.00    0.00    0.00    0.00    0.00   98.80
02:26:25 PM    1    0.20    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    2    0.20    0.00    0.40    0.00    0.00    0.00    0.00    0.00    0.00   99.40
02:26:25 PM    3    0.00    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.80
02:26:25 PM    4    0.20    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    5    0.60    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.20
02:26:25 PM    6    0.40    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    7    1.62    0.00    1.42    0.00    0.00    0.00    0.00    0.00    0.00   96.96
02:26:25 PM    8    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM    9    0.00    0.00    7.60    0.00    0.00    0.00    0.00   92.40    0.00    0.00
02:26:25 PM   10    0.20    0.00    7.00    0.00    0.00    0.00    0.00   92.80    0.00    0.00
02:26:25 PM   11    0.00    0.00    8.60    0.00    0.00    0.00    0.00   91.40    0.00    0.00
02:26:25 PM   12    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   13    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   14    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   15    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
```

この出力では、8 は管理 CPU、9 ～11 はパケットエンジンです。

- vCPU を pCPU 固定に変更するには、2 つのオプションがあります。
  - 次のコマンドを使用して、VPX の起動後に実行時に変更します。

```
virsh vcpupin <VPX name> <vCPU id>  <pCPU number>
virsh vcpupin NetScaler-VPX-XML 0 8
virsh vcpupin NetScaler-VPX-XML 1 9
virsh vcpupin NetScaler-VPX-XML 2 10
virsh vcpupin NetScaler-VPX-XML 3 11
<!--NeedCopy-->
```

  - VPX に静的な変更を加えるには、前と同じように次のタグを付けて `.xml` ファイルを編集します。

1. ホスト上の VPX の.xml ファイルを編集します。

```
/etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```

2. 次のタグを追加します。

```
<vcpu placement='static' cpuset='8-11'>4</vcpu>
    <cputune>
        <vcpupin vcpu='0' cpuset='8'/>
        <vcpupin vcpu='1' cpuset='9'/>
        <vcpupin vcpu='2' cpuset='10'/>
        <vcpupin vcpu='3' cpuset='11'/>
    </cputune>
<!--NeedCopy-->
```

3. VPX をシャットダウンします。

4. 次のコマンドを使用して、NUMA ノードマッピングを使用して VM の設定情報を更新します。

```
virsh define /etc/libvirt/qemu/ <VPX_name>.xml
<!--NeedCopy-->
```

5. VPX の電源をオンにします。次に、ホスト上の`virsh vcpupin <VPX name>`コマンド出力をチェックして、更新された CPU ピン接続を確認します。

ホスト割り込みオーバーヘッドを排除します。

- `kvm_stat`コマンドを使用して VM_EXITS を検出します。

  ハイパーバイザーレベルでは、ホスト割り込みは、VPX の仮想 CPU が固定されているのと同じ pCPU にマッピングされます。これにより、VPX 上の vCPU が定期的に追い出される可能性があります。

  ホストを実行している仮想マシンによって実行された VM の終了を確認するには、`kvm_stat`コマンドを使用します。

```
[root@localhost ~]# kvm_stat -1  | grep EXTERNAL
kvm_exit(EXTERNAL_INTERRUPT)  1728349 27738
[root@localhost ~]#
<!--NeedCopy-->
```

  1+M の順の値が大きいほど、問題があることを示します。

  単一の VM が存在する場合、期待値は 30 ～100 K です。これ以上の値は、同じ pCPU にマッピングされた 1 つ以上のホスト割り込みベクトルがあることを示している可能性があります。

- ホスト割り込みを検出し、ホスト割り込みを移行します。

  「/proc/interrupts」ファイルの`concatenate`コマンドを実行すると、すべてのホスト割り込みマッピングが表示されます。1 つ以上のアクティブな IRQ が同じ pCPU にマップされている場合、対応するカウンタが増分します。

  NetScaler VPX の pCPU と重複する割り込みを未使用の pCPU に移動します。

```
echo 0000000f > /proc/irq/55/smp_affinity
0000000f - - > it is a bitmap, LSBs indicates that IRQ 55 can
   only be scheduled on pCPUs 0  -  3
<!--NeedCopy-->
```

- IRQ バランスを無効にします。

  IRQ バランスデーモンを無効にして、その場で再スケジュールが実行されないようにします。

```
service irqbalance stop
service irqbalance show - To check the status
service irqbalance start - Enable if needed
<!--NeedCopy-->
```

  必ず`kvm_stat`コマンドを実行して、カウンタの数が多くないことを確認します。

PV ネットワークインターフェイスを備えた **NetScaler ADC VPX**

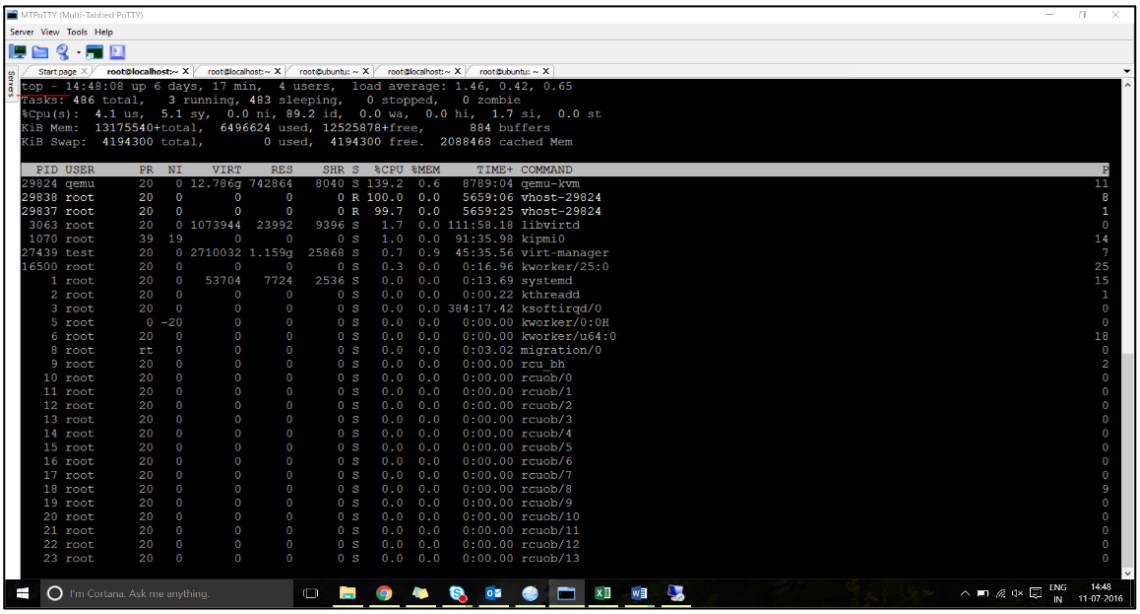
準仮想化（PV）、SR-IOV、および PCIe パススルーネットワークインターフェイスは、物理 NIC ごとに **2** つの **vNIC** 展開として設定できます。詳細については、「物理 NIC 展開ごとに 2 つの vNIC」を参照してください。

PV (virtio) インターフェイスの最適なパフォーマンスを得るには、次の手順に従います。

- PCIe スロット/NIC が関連付けられている NUMA ドメインを識別します。
- VPX のメモリと vCPU は、同じ NUMA ドメインに固定する必要があります。
- 仮想ホストスレッドは、同じ NUMA ドメイン内の CPU にバインドする必要があります。

仮想ホストスレッドを対応する **CPU** にバインドします。

1. トラフィックが開始されたら、ホストで`top`コマンドを実行します。



2. 仮想ホストプロセス (`vhost-<pid-of-qemu>`という名前) アフィニティを識別します。
3. 次のコマンドを使用して、前に特定した NUMA ドメインの物理コアに vHost プロセスをバインドします。

```
taskset -pc <core-id> <process-id>
<!--NeedCopy-->
```

例:

```
taskset -pc 12 29838
<!--NeedCopy-->
```

4. NUMA ドメインに対応するプロセッサコアは、次のコマンドで識別できます。

```
[root@localhost ~]# virsh capabilities | grep cpu
<cpu>
    </cpu>
```



```
        <cpus num='8'>
            <cpu id='0' socket_id='0' core_id='0' siblings='0'/>
            <cpu id='1' socket_id='0' core_id='1' siblings='1'/>
            <cpu id='2' socket_id='0' core_id='2' siblings='2'/>
            <cpu id='3' socket_id='0' core_id='3' siblings='3'/>
            <cpu id='4' socket_id='0' core_id='4' siblings='4'/>
            <cpu id='5' socket_id='0' core_id='5' siblings='5'/>
            <cpu id='6' socket_id='0' core_id='6' siblings='6'/>
            <cpu id='7' socket_id='0' core_id='7' siblings='7'/>
        </cpus>

        <cpus num='8'>
        <cpu id='8' socket_id='1' core_id='0' siblings='8'/>
        <cpu id='9' socket_id='1' core_id='1' siblings='9'/>
        <cpu id='10' socket_id='1' core_id='2' siblings='10'/>
        <cpu id='11' socket_id='1' core_id='3' siblings='11'/>
        <cpu id='12' socket_id='1' core_id='4' siblings='12'/>
        <cpu id='13' socket_id='1' core_id='5' siblings='13'/>
        <cpu id='14' socket_id='1' core_id='6' siblings='14'/>
        <cpu id='15' socket_id='1' core_id='7' siblings='15'/>
        </cpus>

    <cpuselection/>
    <cpuselection/>

<!--NeedCopy-->
```

QEMU プロセスを対応する物理コアにバインドします。

1. QEMU プロセスが実行されている物理コアを特定します。詳細については、前述の出力を参照してください。
2. 次のコマンドを使用して、vCPU をバインドするのと同じ物理コアに QEMU プロセスをバインドします。

```
taskset -pc 8-11 29824
<!--NeedCopy-->
```

SR-IOV およびフォートビルの **PCIe** パススルーネットワークインターフェイスを備えた **NetScaler ADC VPX**

SR-IOV および Fortville PCIe パススルーネットワークインターフェイスのパフォーマンスを最適化するには、次の手順を実行します。

- PCIe スロット/NIC が関連付けられている NUMA ドメインを識別します。
- VPX のメモリと vCPU は、同じ NUMA ドメインに固定する必要があります。

Linux KVM の **vCPU** およびメモリピンニング用のサンプル **VPX XML** ファイル:

```
    <domain type='kvm'>
        <name>NetScaler-VPX</name>
        <uuid>138f7782-1cd3-484b-8b6d-7604f35b14f4</uuid>
        <memory unit='KiB'>8097152</memory>
```



```
        <currentMemory unit='KiB'>8097152</currentMemory>
        <vcpu placement='static'>4</vcpu>

    <cputune>
        <vcpupin vcpu='0' cpuset='8'/>
        <vcpupin vcpu='1' cpuset='9'/>
        <vcpupin vcpu='2' cpuset='10'/>
        <vcpupin vcpu='3' cpuset='11'/>
    </cputune>

    <numatune>
    <memory mode='strict' nodeset='1'/>
    </numatune>

    </domain>
<!--NeedCopy-->
```

Citrix Hypervisor 上の NetScaler ADC VPX インスタンス

このセクションでは、構成可能なオプションと設定、および Citrix Hypervisors 上の NetScaler ADC VPX インスタンスの最適なパフォーマンスを達成するのに役立つその他の推奨事項について説明します。

- Citrix Hypervisor のパフォーマンス設定
- SR-IOV ネットワークインターフェイスを備えた NetScaler ADC VPX
- 準仮想化インターフェイスを備えた NetScaler ADC VPX

Citrix Hypervisor のパフォーマンス設定

「**xl**」コマンドを使用して **NIC** の **NUMA** ドメインを見つけます。

```
xl info -n
<!--NeedCopy-->
```

VPX の **vCPU** を物理コアにピン留めします。

```
xl vcpu-pin <Netsclaer VM Name>  <vCPU id>  <physical CPU id>
<!--NeedCopy-->
```

vCPU のバインドをチェックします。

```
xl vcpu-list
<!--NeedCopy-->
```

8 個を超える仮想 **CPU** を **NetScaler ADC** 仮想マシンに割り当てます。

8 個を超える仮想 CPU を構成するには、Citrix Hypervisor コンソールから次のコマンドを実行します。

```
xe vm-param-set uuid=your_vms_uuid VCPUs-max=16
```



```
xe vm-param-set uuid=your_vms_uuid VCPUs-at-startup=16
<!--NeedCopy-->
```

SR-IOV ネットワークインターフェイスを備えた **NetScaler ADC VPX**

SR-IOV ネットワークインターフェイスの最適なパフォーマンスを得るには、次の手順を実行します。

- PCIe スロットまたは NIC が接続されている NUMA ドメインを特定します。
- VPX のメモリと vCPU を同じ NUMA ドメインに固定します。
- ドメイン 0 vCPU を残りの CPU にバインドします。

準仮想化インターフェイスを備えた **NetScaler ADC VPX**

最適なパフォーマンスを得るには、他の PV 環境と同様に、pNIC ごとに 2 つの vNIC、および pNIC 構成ごとに 1 つの vNIC を推奨します。

準仮想化 (netfront) インターフェイスの最適なパフォーマンスを実現するには、次の手順を実行します。

- PCIe スロットまたは NIC が接続されている NUMA ドメインを特定します。
- VPX のメモリと vCPU を同じ NUMA ドメインに固定します。
- ドメイン 0 vCPU を同じ NUMA ドメインの残りの CPU にバインドします。
- 仮想 NIC のホスト Rx/Tx スレッドをドメイン 0 vCPU に固定します。

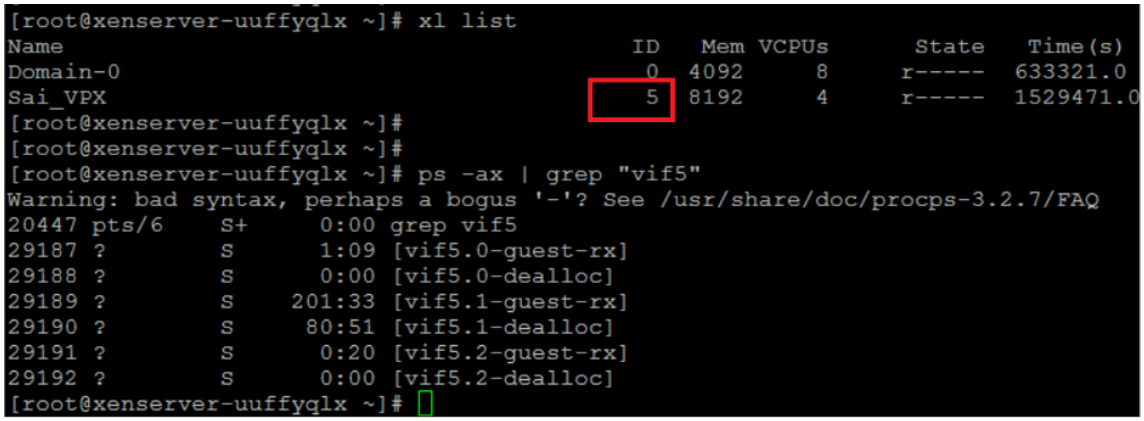
ホストスレッドをドメイン **0 vCPU** にピン留めします。

1. Citrix Hypervisor ホストシェルで`xl list`コマンドを使用して、VPX の Xen-ID を検索します。
2. 次のコマンドを使用して、ホストスレッドを識別します。

   ```
   ps -ax | grep vif <Xen-ID>
   <!--NeedCopy-->
   ```

   次の例では、これらの値は次のことを示しています。

   - **vif5.0** -XenCenter（管理インターフェイス）で VPX に割り当てられた最初のインターフェイスのスレッド。
   - **vif5.1** -VPX などに割り当てられた 2 番目のインターフェイスのスレッド。



3. 次のコマンドを使用して、スレッドをドメイン 0 vCPU に固定します。

```
taskset  - pc <core-id> <process-id>
<!--NeedCopy-->
```

例:

```
taskset -pc 1 29189
<!--NeedCopy-->
```

NetScaler VPX 構成をクラウドで NetScaler アプライアンスの最初の起動時に適用する

February 15, 2024

NetScaler VPX 構成は、クラウド環境での NetScaler アプライアンスの最初の起動時に適用できます。このステージは、このドキュメントでプレブートステージとして取り上げられています。したがって、ADC プールライセンスなどの特定のケースでは、特定の VPX インスタンスがはるかに短時間で起動されます。この機能は、Microsoft Azure、Google Cloud Platform、および AWS クラウドで使用できます。

ユーザーデータとは何ですか

クラウド環境で VPX インスタンスをプロビジョニングする場合、ユーザーデータをインスタンスに渡すオプションがあります。ユーザーデータを使用すると、一般的な自動設定タスクの実行、インスタンスの起動動作のカスタマイズ、インスタンスの起動後にスクリプトを実行できます。最初の起動時に、NetScaler VPX インスタンスは次のタスクを実行します。

- ユーザーデータを読み取ります。
- ユーザーデータで提供される構成を解釈します。
- 新しく追加された構成をブート時に適用します。

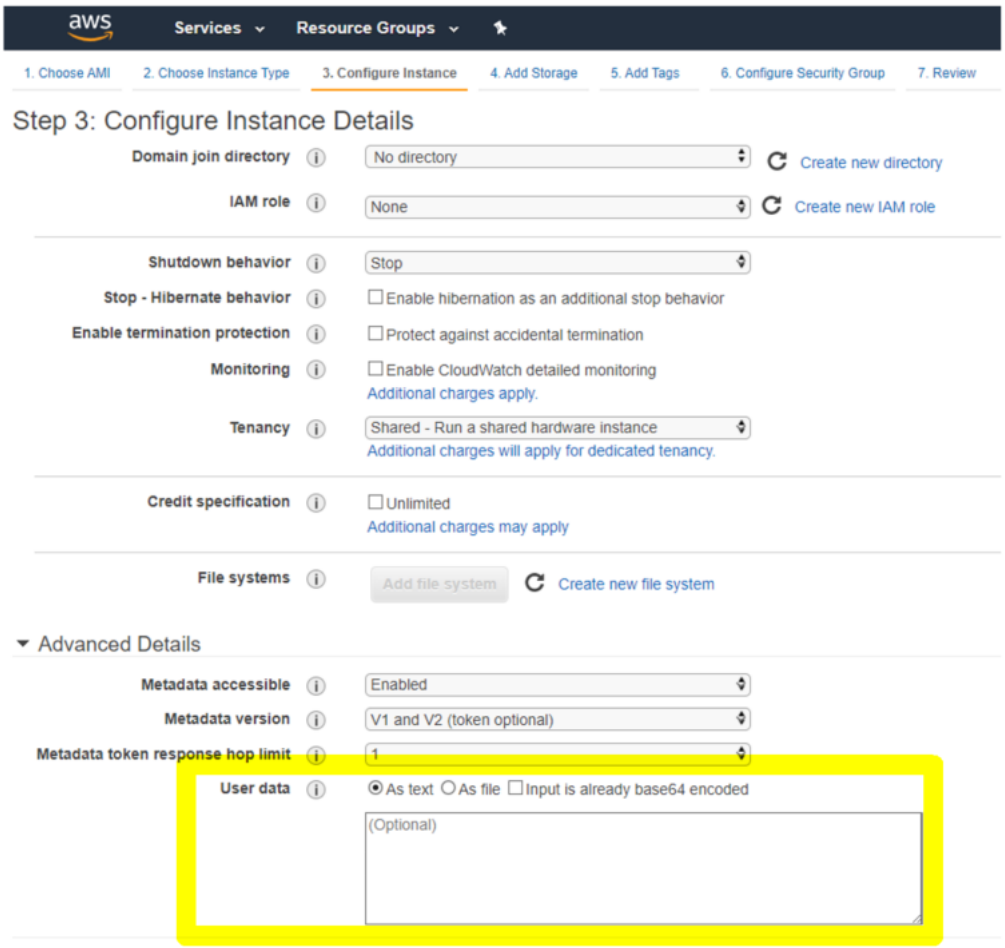
クラウドインスタンスでプレブートユーザーデータを提供する方法

プレブートユーザーデータを XML 形式でクラウドインスタンスに提供できます。クラウドによって、ユーザーデータを提供するためのインターフェースが異なります。

AWS コンソールを使用してプレブートユーザーデータを提供する

AWS コンソールを使用して NetScaler VPX インスタンスをプロビジョニングする場合は、[インスタンスの詳細の構成] **>** [詳細の詳細] に移動し、[ユーザーデータ] フィールドにプレブートユーザーデータ構成を指定します。

各手順の詳細については、「AWS ウェブコンソールを使用して AWS に NetScaler VPX インスタンスをデプロイする」を参照してください。
詳細については、インスタンスの起動に関するAWS ドキュメントを参照してください。



> 注:
>
> プリブートユーザーデータ機能の AWS IMDSv2 専用モードは、NetScaler VPX リリース 13.1.48.x 以降のリリースでサポートされています。

AWS CLI を使用してプレブートユーザーデータを提供する

AWS CLI で次のコマンドを入力します。

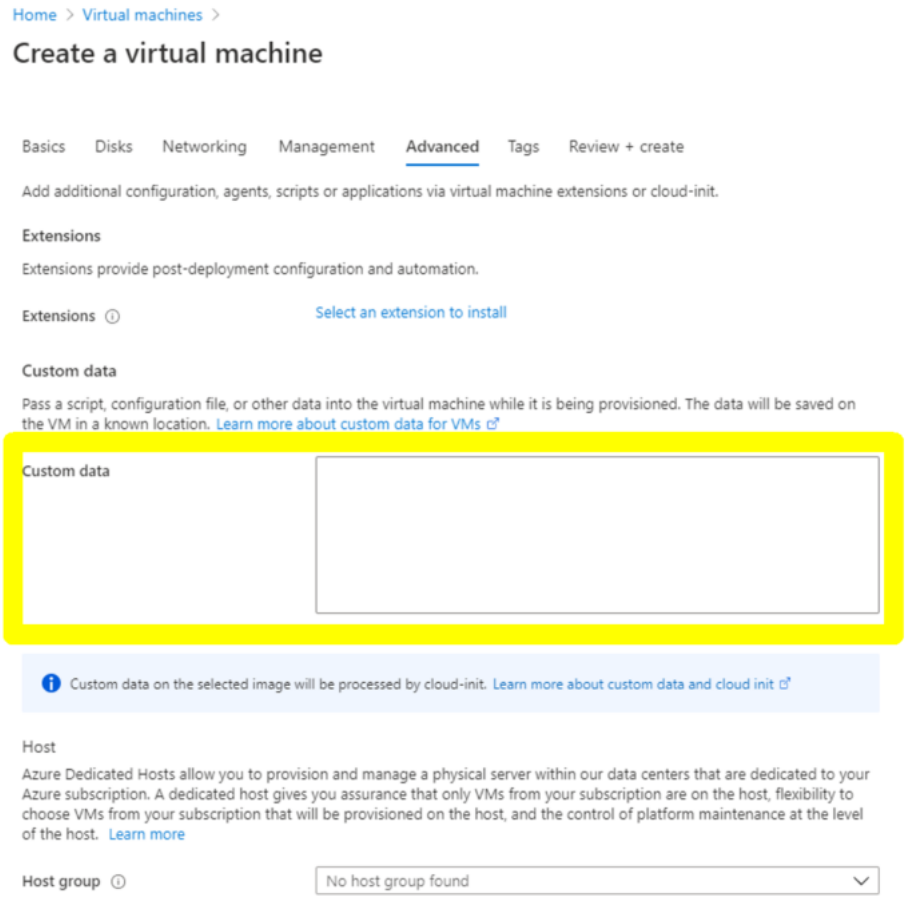
```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
    --key-name MyKeyPair \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --user-data file://my_script.txt
<!--NeedCopy-->
```

詳細については、インスタンスの実行に関するAWS ドキュメントを参照してください。

詳細については、インスタンスユーザーデータの使用に関するAWS ドキュメントを参照してください。

Azure コンソールを使用してプリブートユーザーデータを提供する

Azure コンソールを使用して NetScaler VPX インスタンスをプロビジョニングする場合は、[仮想マシンの作成] > [詳細設定] タブに移動します。[カスタムデータ] フィールドに、プリブートユーザーデータの構成を指定します。



Azure CLI を使用してプリブートユーザーデータを提供する

Azure CLI で次のコマンドを入力します。

```
az vm create \
  --resource-group myResourceGroup \
  --name MyVm \
  --image debian \
  --custom-data MyCloudInitScript.txt \
<!--NeedCopy-->
```

例**:**

```
az vm create --resource-group MyResourceGroup -name MyVm --image debian
     --custom-data MyCloudInitScript.txt
<!--NeedCopy-->
```

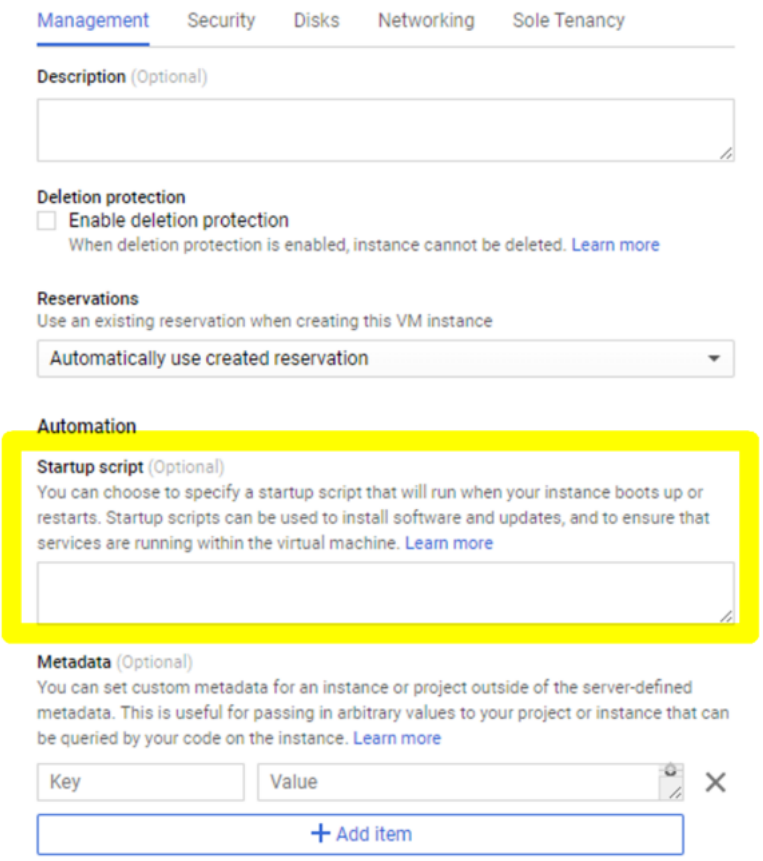
カスタムデータまたはプリブート設定をファイルとして「—custom-data」パラメータに渡すことができます。この例では、ファイル名は **MyCloudInitScript.txt** です。

詳細については、Azure CLI のドキュメントを参照してください。

GCP コンソールを使用してプレブートユーザーデータを提供する

GCP コンソールを使用して NetScaler VPX インスタンスをプロビジョニングする場合は、インスタンスのプロパティを入力します。管理、セキュリティ、ディスク、ネットワーキング、単独テナンシを展開します。[管理] タブに移動します。[自動化] セクションで、[スタートアップスクリプト] フィールドにプリブートユーザーデータ設定を指定します。

GCP を使用した VPX インスタンスの作成の詳細については、「Google Cloud Platform での NetScaler VPX インスタンスの展開」を参照してください。



gcloud CLI を使用してプレブートユーザーデータを提供する

GCP CLI で次のコマンドを入力します。

```
gcloud compute instances create INSTANCE_NAMES --metadata-from-file=
    startup-script=LOCAL_FILE_PATH
<!--NeedCopy-->
```

metadata-from-file -に格納されているファイルから値またはユーザーデータを読み取ります。.

詳細については、gcloud CLI ドキュメントを参照してください。

プレブートユーザーデータ形式

プレブートユーザーデータは XML 形式でクラウドインスタンスに提供する必要があります。起動時にクラウドインフラストラクチャを介して提供される NetScaler プレブートユーザーデータは、次の 4 つのセクションで構成されます。

- NetScaler 構成は<NS-CONFIG>タグで表されます。

- <NS-BOOTSTRAP>タグで表される NetScaler をカスタムブートストラップします。
- <NS-SCRIPTS>タグで表される NetScaler にユーザースクリプトを保存する。
- <NS-LICENSE-CONFIG>タグで表されるプールライセンス構成。

前の 4 つのセクションは、ADC のプレブート構成内で任意の順序で提供できます。
プリブートユーザーデータを提供しながら、次のセクションに示す書式に厳密に従うようにしてください。

> 注:
>
> 次の例に示すように、プレブートユーザーデータ構成全体を<NS-PRE-BOOT-CONFIG>タグで囲む必要があります。

例 **1**:

```
<NS-PRE-BOOT-CONFIG>
      <NS-CONFIG>           </NS-CONFIG>
      <NS-BOOTSTRAP>        </NS-BOOTSTRAP>
      <NS-SCRIPTS>          </NS-SCRIPTS>
      <NS-LICENSE-CONFIG>   </NS-LICENSE-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

例 **2**:

```
<NS-PRE-BOOT-CONFIG>
    <NS-LICENSE-CONFIG> </NS-LICENSE-CONFIG>
    <NS-SCRIPTS>        </NS-SCRIPTS>
    <NS-BOOTSTRAP>      </NS-BOOTSTRAP>
    <NS-CONFIG>         </NS-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

<NS-CONFIG>タグを使用して、プレブート段階で VPX インスタンスに適用する必要がある特定の NetScaler VPX 構成を指定します。

> 注:
>
> <NS-CONFIG>セクションには、有効な ADC CLI コマンドが必要です。CLI は、構文エラーまたは形式について検証されません。

NetScaler 構成

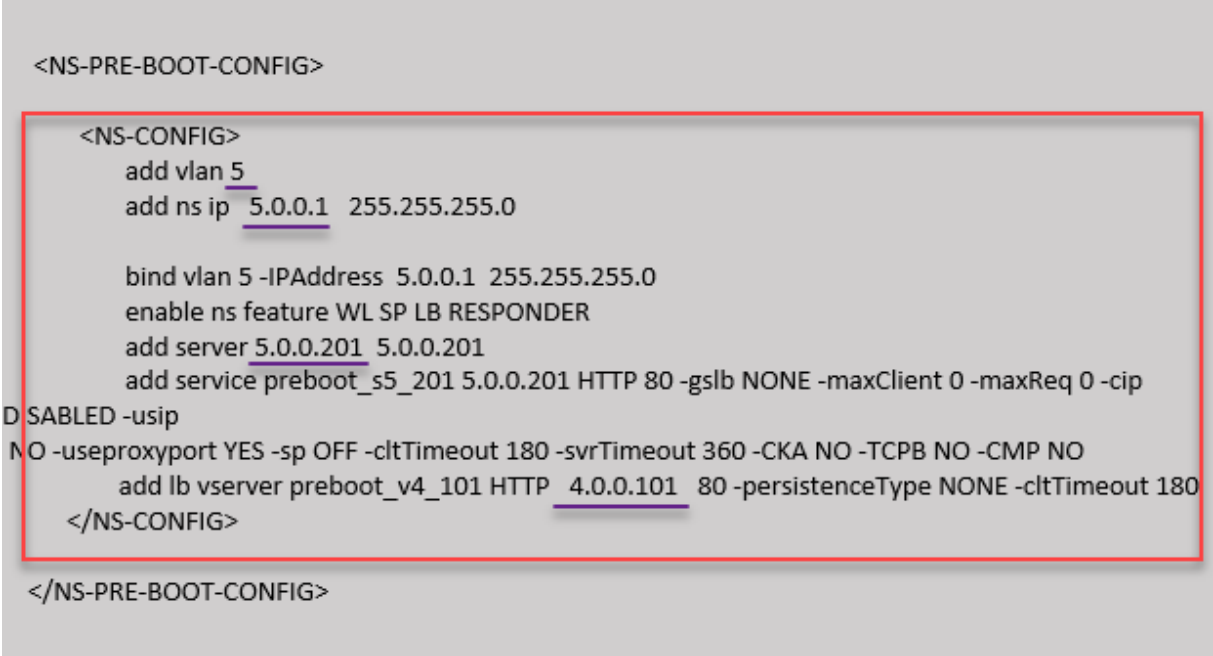
<NS-CONFIG>タグを使用して、プレブート段階で VPX インスタンスに適用する必要がある特定の NetScaler VPX 構成を指定します。

> 注:

> <NS-CONFIG>セクションには、有効な ADC CLI コマンドが必要です。CLI は、構文エラーまたは形式について検証されません。

例:

次の例では、<NS-CONFIG>セクションに設定の詳細を示します。ID「5」の VLAN が設定され、SNIP（5.0.0.1）にバインドされます。負荷分散仮想サーバー (4.0.0.101) も構成されています。



前のスクリーンショットに示した設定をここからコピーできます。

```
<NS-PRE-BOOT-CONFIG>
     <NS-CONFIG>
         add vlan 5
         add ns ip 5.0.0.1 255.255.255.0
         bind vlan 5 -IPAddress 5.0.0.1 255.255.255.0
         enable ns feature WL SP LB RESPONDER
         add server 5.0.0.201 5.0.0.201
         add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip
 NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -
    TCPB NO -CMP NO
         add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
            persistenceType NONE -cltTimeout 180
     </NS-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

NetScaler VPX インスタンスは、次の図に示すように、<NS-CONFIG>セクションに適用された構成を表示します。


```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  -------
1)      10.160.0.72      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      5.0.0.1          0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:48/64
        Interfaces : 1/1 1/2 LO/1

2)      VLAN ID: 5      VLAN Alias Name:
        IPs :
             5.0.0.1    Mask: 255.255.255.0

3)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 0/1
        IPs :
             10.160.0.72        Mask: 255.255.240.0
 Done
```

```
> sh server
1)      Name:          5.0.0.201       State:ENABLED
        IPAddress:        5.0.0.201
2)      Name:    169.254.169.254       State:ENABLED
        IPAddress: 169.254.169.254
 Done
> stat service

Service(s) Summary
                           IP  port         Type        State    Req/s
preb...5_201        5.0.0.201    80         HTTP         DOWN      0/s
gcpl...vice0 169.254.169.254    53          DNS           UP      0/s
 Done
> sh service preboot_s5_201
        preboot_s5_201 (5.0.0.201:80) - HTTP
        State: DOWN
        Last state change was at Tue Dec 29 07:18:28 2020
        Time since last state change: 0 days, 00:05:02.820
        Server Name: 5.0.0.201
        Server ID : None        Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Monitoring Owner: 0
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED
        Monitor Connection Close : NONE
        Appflow logging: ENABLED
        Process Local: DISABLED
```


ユーザースクリプト

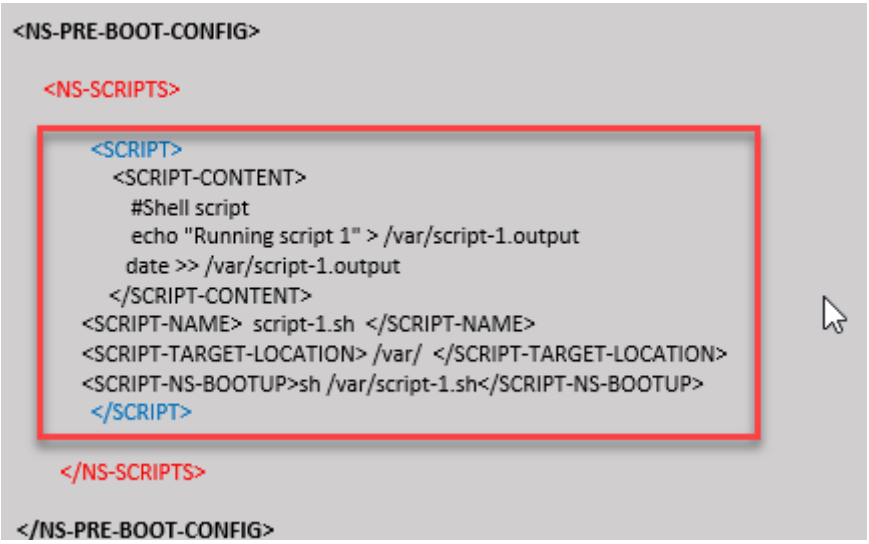
<NS-SCRIPTS>タグを使用して、NetScaler VPX インスタンスに保存して実行する必要があるスクリプトを指定します。

<NS-SCRIPTS>タグには多数のスクリプトを含めることができます。各スクリプトは<SCRIPT>タグ内に含める必要があります。
各<SCRIPT>セクションは 1 つのスクリプトに対応し、次のサブタグを使用してスクリプトの詳細をすべて含みます。

- **<SCRIPT-NAME>:** 保存する必要のあるスクリプトファイルの名前を示します。
- **<SCRIPT-CONTENT>:** 保存する必要のあるファイルの内容を示します。
- **<SCRIPT-TARGET-LOCATION>:** このファイルを保存する必要がある指定されたターゲットの場所を示します。ターゲットの場所が指定されていない場合、デフォルトでは、ファイルまたはスクリプトは「/nsconfig」ディレクトリに保存されます。
- **<SCRIPT-NS-BOOTUP>:** スクリプトの実行に使用するコマンドを指定します。
  - <SCRIPT-NS-BOOTUP>セクションを使用する場合、セクションで提供されるコマンドは「/nsconfig/nsafter.sh」に保存され、コマンドは「nsafter.sh」実行の一部としてパケットエンジンが起動した後に実行されます。
  - <SCRIPT-NS-BOOTUP>セクションを使用しない場合、スクリプトファイルは指定したターゲットの場所に保存されます。

例 **1**:

この例では、<NS-SCRIPTS>タグには script-1.sh というスクリプトの詳細が 1 つだけ含まれています。「script-1.sh」スクリプトは「/var」ディレクトリに保存されます。スクリプトは指定された内容で読み込まれ、パケットエンジンの起動後に「sh /var/script-1.sh」コマンドで実行されます。



前のスクリーンショットに示した設定をここからコピーできます。


```
<NS-PRE-BOOT-CONFIG>
    <NS-SCRIPTS>
    <SCRIPT>
            <SCRIPT-CONTENT>
                #Shell script
                echo "Running script 1" > /var/script-1.output
                date >> /var/script-1.output
            </SCRIPT-CONTENT>

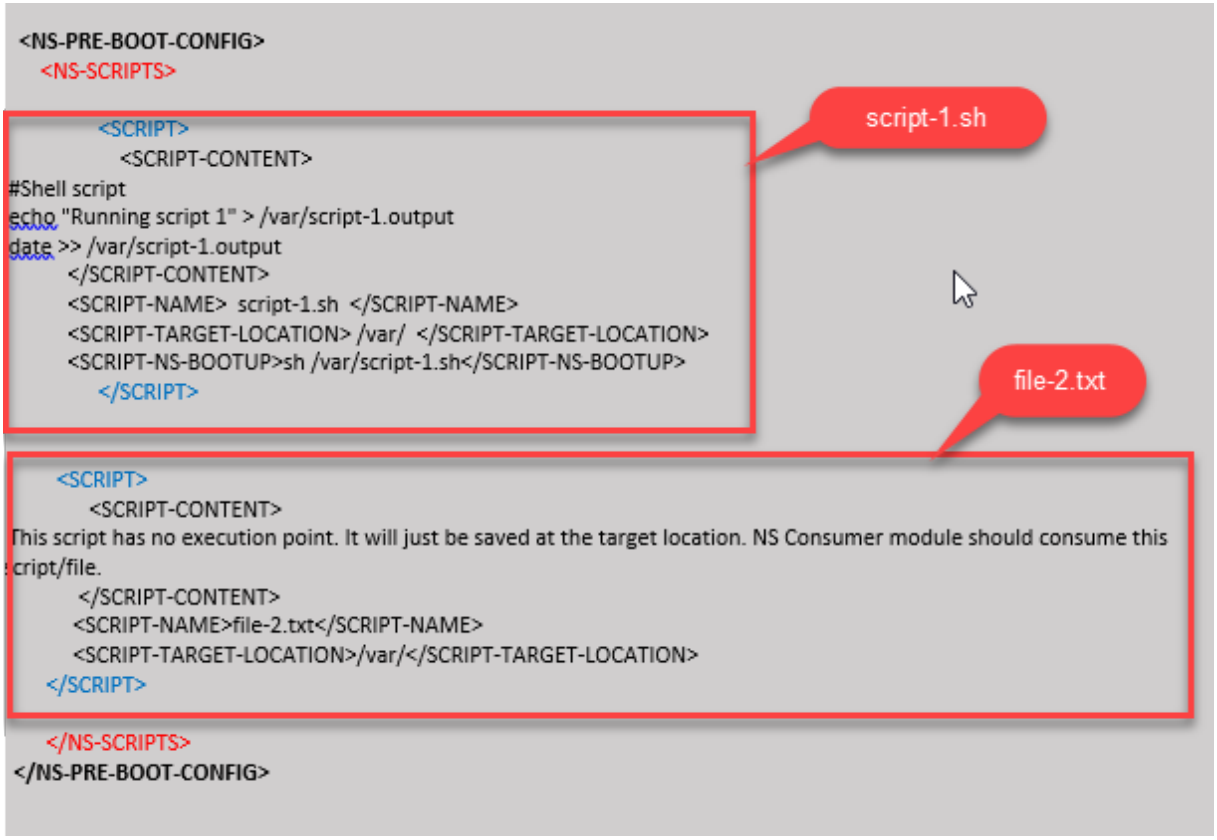
                <SCRIPT-NAME> script-1.sh </SCRIPT-NAME>
                <SCRIPT-TARGET-LOCATION> /var/ </SCRIPT-TARGET-LOCATION
                    >
                <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP
                    >
        </SCRIPT>
    </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

次のスナップショットでは、「script-1.sh」スクリプトが「/var/」ディレクトリに保存されていることを確認できます。「Script-1.sh」スクリプトが実行され、出力ファイルが適切に作成されます。

例 **2**:

次の例では、<NS-SCRIPTS>タグに 2 つのスクリプトの詳細が含まれています。

- 最初のスクリプトは「script-1.sh」として「/var」ディレクトリに保存されます。スクリプトは指定された内容で読み込まれ、パケットエンジンの起動後にコマンド「sh /var/script-1.sh」で実行されます。
- 2 番目のスクリプトは「file-2.txt」として「/var」ディレクトリに保存されます。このファイルには、指定されたコンテンツが入力されます。しかし、ブートアップ実行コマンド<SCRIPT-NS-BOOTUP>が提供されていないため、実行されません。



前のスクリーンショットに示した設定をここからコピーできます。

```
<NS-PRE-BOOT-CONFIG>
    <NS-SCRIPTS>
        <SCRIPT>
            <SCRIPT-CONTENT>
               #Shell script
               echo "Running script 1" > /var/script-1.output
               date >> /var/script-1.output
            </SCRIPT-CONTENT>

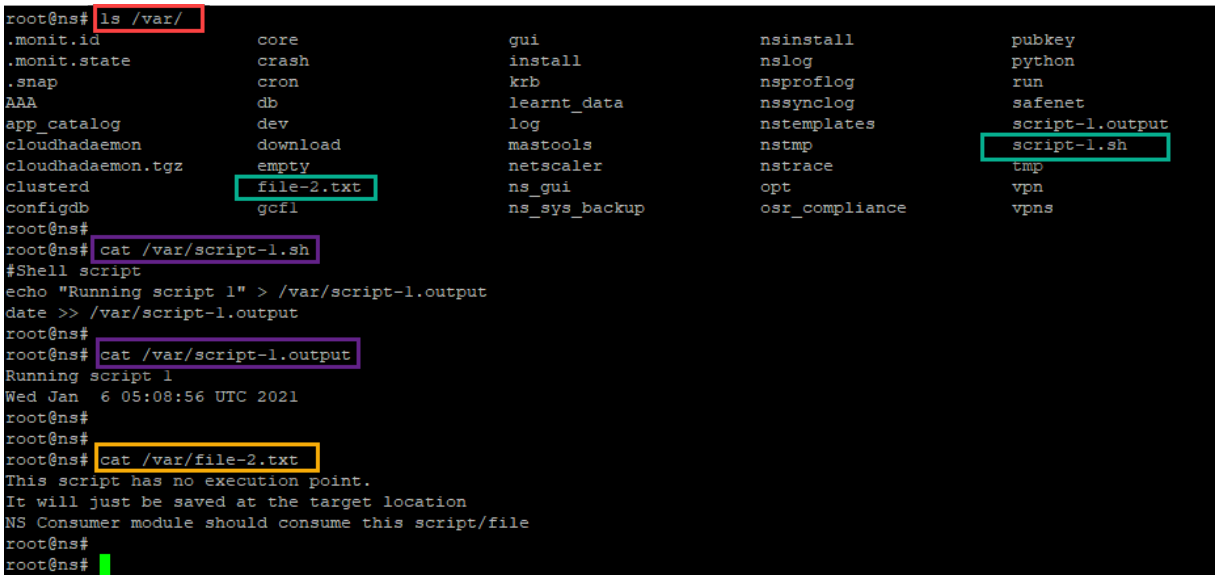
            <SCRIPT-NAME>  script-1.sh  </SCRIPT-NAME>
            <SCRIPT-TARGET-LOCATION> /var/  </SCRIPT-TARGET-LOCATION>
            <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP>
            </SCRIPT>

        <SCRIPT>
            <SCRIPT-CONTENT>
                This script has no execution point.
                It will just be saved at the target location
                NS Consumer module should consume this script/file
            </SCRIPT-CONTENT>
            <SCRIPT-NAME>file-2.txt</SCRIPT-NAME>
            <SCRIPT-TARGET-LOCATION>/var/</SCRIPT-TARGET-LOCATION>
        </SCRIPT>
    </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
```



```
26  <!--NeedCopy-->
```

次のスナップショットでは、script-1.sh と file-2.txt が「/var/」ディレクトリに作成されていることを確認できます。Script-1.sh が実行され、出力ファイルが適切に作成されます。



ライセンス

VPX インスタンスの起動中に NetScaler プールライセンスを適用するには、<NS-LICENSE-CONFIG>タグを使用します。<NS-LICENSE-CONFIG>セクション内の<LICENSE-COMMANDS>タグを使用して、プールされたライセンスコマンドを指定します。これらのコマンドは構文的に有効である必要があります。

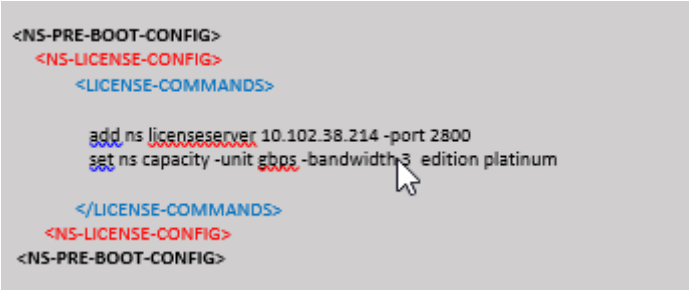
標準のプールライセンスコマンドを使用して、<LICENSE-COMMANDS>セクションで、ライセンスタイプ、容量、ライセンスサーバーなどのプールされたライセンスの詳細を指定できます。詳細については、「NetScaler プール容量ライセンスの構成」を参照してください。

<NS-LICENSE-CONFIG>を適用した後、VPX は起動時に要求されたエディションを起動し、VPX はライセンスサーバから構成されたライセンスをチェックアウトしようとします。

- ライセンスのチェックアウトが成功すると、構成された帯域幅が VPX に適用されます。
- ライセンスのチェックアウトに失敗した場合、約 10 ～12 分以内にライセンスはライセンスサーバから取得されません。その結果、システムがリブートし、ライセンスなしの状態になります。

例:

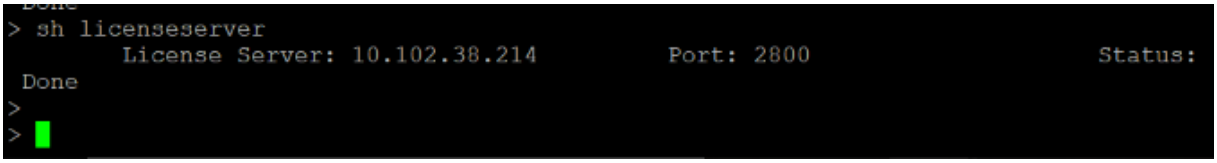
次の例では、<NS-LICENSE-CONFIG>を適用した後、VPX は起動時にプレミアムエディションを起動し、VPX はライセンスサーバ（10.102.38.214）から構成されたライセンスをチェックアウトしようとします。



前のスクリーンショットに示した設定をここからコピーできます。

```
<NS-PRE-BOOT-CONFIG>
   <NS-LICENSE-CONFIG>
        <LICENSE-COMMANDS>
            add ns licenseserver 10.102.38.214 -port 2800
            set ns capacity -unit gbps -bandwidth 3  edition platinum
        </LICENSE-COMMANDS>
    </NS-LICENSE-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

次の図に示すように、「ライセンスサーバーの表示」コマンドを実行し、ライセンスサーバー（10.102.38.214）が VPX に追加されていることを確認します。



ブートストラッピング

<NS-BOOTSTRAP>タグを使用して、カスタムブートストラップ情報を指定します。<NS-BOOTSTRAP>セクション内では、<SKIP-DEFAULT-BOOTSTRAP>タグと<NEW-BOOTSTRAP-SEQUENCE>タグを使用できます。このセクションでは、デフォルトのブートストラップを回避するかどうかを NetScaler アプライアンスに通知します。デフォルトのブートストラップが回避される場合、このセクションでは、新しいブートストラップシーケンスを提供するオプションを提供します。

デフォルトのブートストラップ構成

NetScaler アプライアンスのデフォルトのブートストラップ構成は、次のインターフェイスの割り当てに従います。

- **Eth0** -特定の NSIP アドレスを持つ管理インターフェイス。
- **Eth1** -特定の VIP アドレスを持つクライアント向けインターフェイス。
- **Eth2** -特定の SNIP アドレスを持つサーバー側インターフェイス。

ブートストラップ構成をカスタマイズする

デフォルトのブートストラップシーケンスをスキップして、NetScaler VPX インスタンスに新しいブートストラップシーケンスを指定することができます。<NS-BOOTSTRAP>タグを使用して、カスタムブートストラップ情報を指定します。たとえば、管理インターフェイス（NSIP）、クライアント側インターフェイス（VIP）、およびサーバー側インターフェイス（SNIP）が常に特定の順序で提供されるデフォルトのブートストラップを変更できます。

次の表に、<SKIP-DEFAULT-BOOTSTRAP>および<NEW-BOOTSTRAP-SEQUENCE>タグで許可されるさまざまな値を使用したブートストラップ動作を示します。

SKIP-DEFAULT-BOOTSTRAP	NEW-BOOTSTRAP-SEQUENCE	ブートストラップ動作
はい	はい	デフォルトのブートストラップ動作はスキップされ、<NS-BOOTSTRAP>セクションで提供される新しいカスタムブートストラップシーケンスが実行されます。
はい	いいえ	デフォルトのブートストラップ動作はスキップされます。「<NS-CONFIG>」セクションに記載されているブートストラップコマンドが実行されます。

ブートストラップ構成は、次の 3 つの方法でカスタマイズできます。

- インターフェイスの詳細のみを入力します。
- IP アドレスとサブネットマスクとともにインターフェイスの詳細を指定します。
- <NS-CONFIG>セクションにブートストラップ関連のコマンドを入力します。

方法 **1:** インターフェイスの詳細のみを指定してカスタムブートストラップ

管理インターフェイス、クライアント向けインターフェイス、およびサーバ側インターフェイスは指定しますが、その IP アドレスとサブネットマスクは指定しません。IP アドレスとサブネットマスクは、クラウドインフラストラクチャのクエリによって設定されます。

AWS のカスタムブートストラップの例

次の例に示すように、カスタムブートストラップシーケンスを指定します。詳細については、「クラウドインスタンスでプレブートユーザーデータを提供する方法」を参照してください。Eth1 インターフェイスは管理インターフェイス

(NSIP)、クライアントインターフェイス（VIP）として Eth0 インターフェイス、サーバインターフェイス（SNIP）として Eth2 インターフェイスが割り当てられます。<NS-BOOTSTRAP>セクションには、インターフェイスの詳細のみが含まれ、IP アドレスとサブネットマスクの詳細は含まれません。

```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

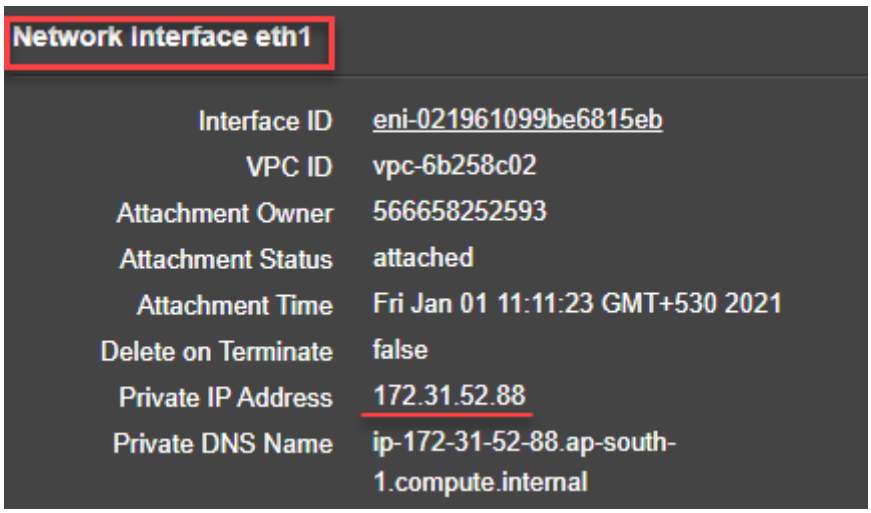
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

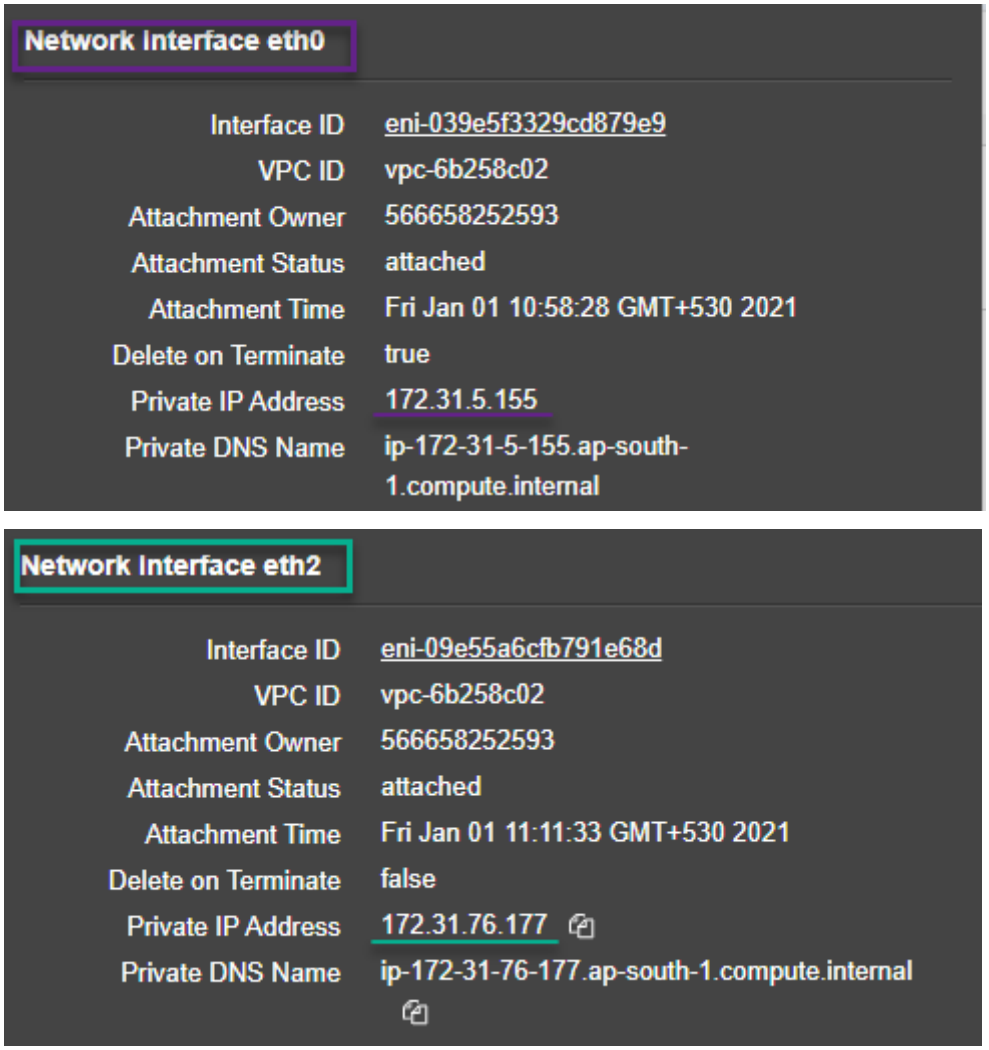
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOT STRAP>
</NS-PRE-BOOT-CONFIG>
```

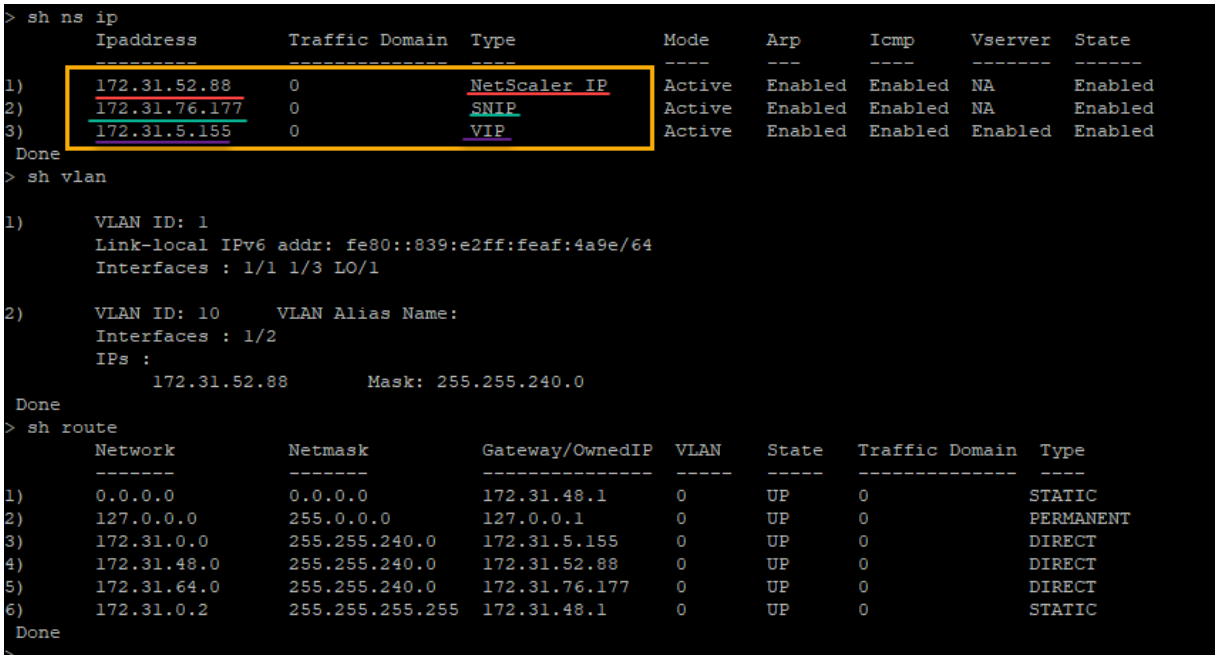
VM インスタンスの作成後、AWS ポータルで、ネットワークインターフェイスのプロパティを次のように確認できます。

1. [**AWS Portal] > [EC2** インスタンス] に移動し、カスタムブートストラップ情報を指定して作成したインスタンスを選択します。
2. [説明] タブでは、次の図に示すように、各ネットワークインタフェースのプロパティを確認できます。





ADC CLI で`show nsip`コマンドを実行し、ADC アプライアンスの初回起動時に NetScalerVPX インスタンスに適用されるネットワークインターフェイスを確認できます。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.31.76.177    0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.31.5.155     0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.0.0       255.255.240.0    172.31.5.155     0       UP      0               DIRECT
4)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
5)      172.31.64.0      255.255.240.0    172.31.76.177    0       UP      0               DIRECT
6)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
```

Azure のカスタムブートストラップの例

次の例に示すように、カスタムブートストラップシーケンスを指定します。詳細については、「クラウドインスタンスでプレブートユーザーデータを提供する方法」を参照してください。Eth2 インターフェイスは、管理インターフェイス（NSIP）として、Eth1 インターフェイスをクライアントインターフェイス（VIP）として、Eth0 インターフェイスをサーバインターフェイス（SNIP）として割り当てます。<NS-BOOTSTRAP>セクションには、インターフェイスの詳細のみが含まれ、IP アドレスとサブネットマスクの詳細は含まれません。


```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

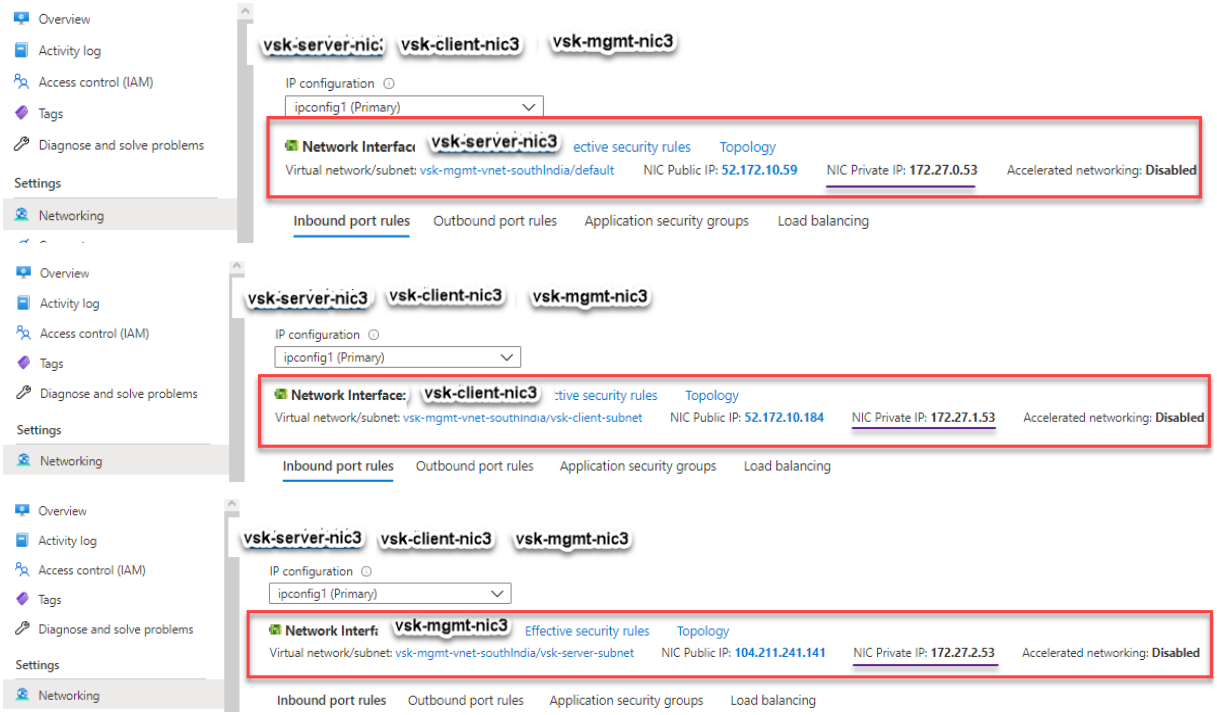
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

NetScaler VPX インスタンスが 3 つのネットワークインターフェイスで作成されていることがわかります。**Azure Portal > VM** インスタンス **>** ネットワークに移動し、次の図に示すように 3 つの NIC のネットワークプロパティを確認します。



ADC CLI で「show nsip」コマンドを実行し、<NS−BOOTSTRAP>セクションで指定された新しいブートストラ

ップシーケンスが適用されることを確認できます。「show route」コマンドを実行して、サブネットマスクを確認できます。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.53      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.53      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.27.1.53      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:c26c/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.53        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.53      0       UP      0               DIRECT
4)      172.27.1.0       255.255.255.0    172.27.1.53      0       UP      0               DIRECT
5)      172.27.2.0       255.255.255.0    172.27.2.53      0       UP      0               DIRECT
6)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
7)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
8)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
>
```

GCP のカスタムブートストラップの例

次の例に示すように、カスタムブートストラップシーケンスを指定します。詳細については、「クラウドインスタンスでプレブートユーザーデータを提供する方法」を参照してください。Eth1 インターフェイスは管理インターフェイス（NSIP）、クライアントインターフェイス（VIP）として Eth0 インターフェイス、サーバインターフェイス（SNIP）として Eth2 インターフェイスが割り当てられます。<NS-BOOTSTRAP>セクションには、インターフェイスの詳細のみが含まれ、IP アドレスとサブネットマスクの詳細は含まれません。


```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

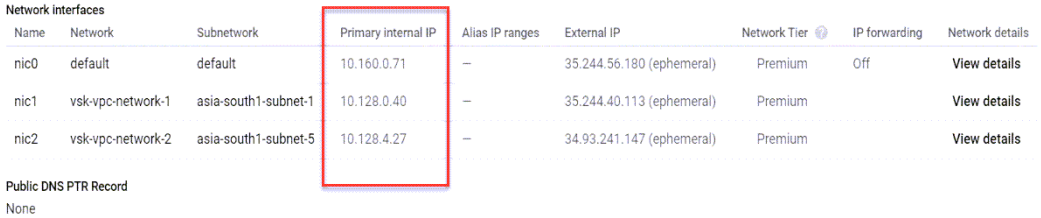
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

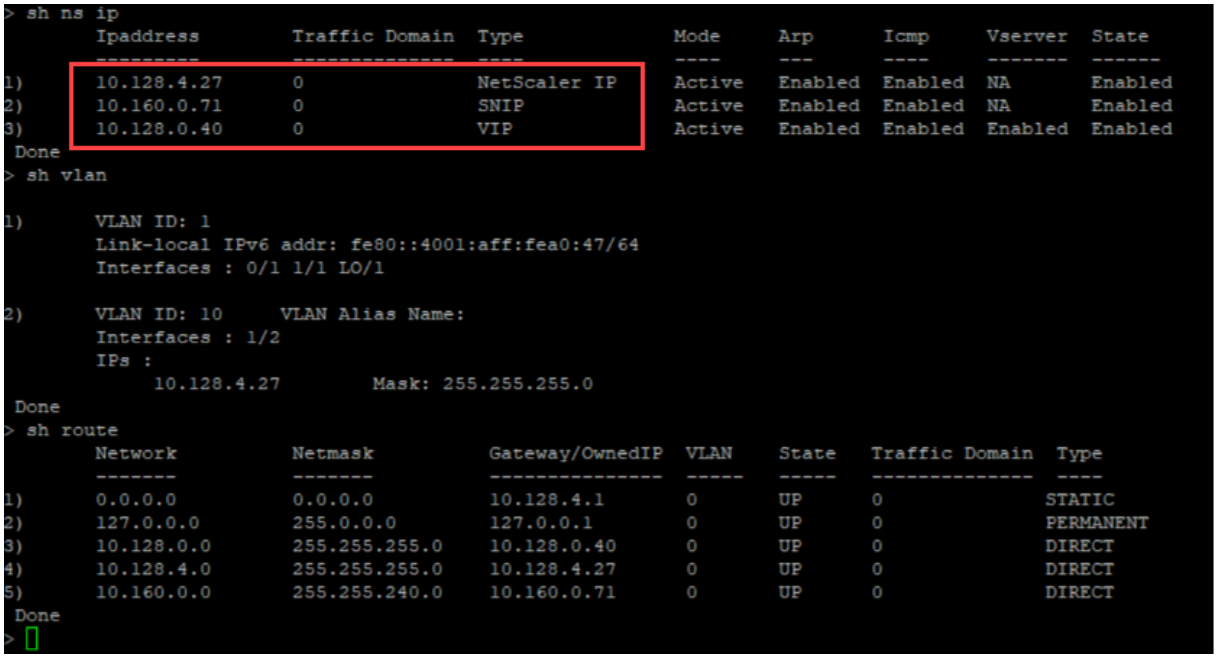
    </NS-BOOT STRAP>
</NS-PRE-BOOT-CONFIG>
```

GCP ポータルで VM インスタンスを作成した後、ネットワークインターフェイスのプロパティを次のように確認できます。

1. カスタムブートストラップ情報を指定して、作成したインスタンスを選択します。
2. ネットワークインターフェイスのプロパティに移動し、NIC の詳細を次のように確認します。



ADC CLI で`show nsip`コマンドを実行し、ADC アプライアンスの初回起動時に NetScalerVPX インスタンスに適用されるネットワークインターフェイスを確認できます。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  -------
1)      10.128.4.27      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.160.0.71      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.128.0.40      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:47/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             10.128.4.27        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.4.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.40      0       UP      0               DIRECT
4)      10.128.4.0       255.255.255.0    10.128.4.27      0       UP      0               DIRECT
5)      10.160.0.0       255.255.240.0    10.160.0.71      0       UP      0               DIRECT
 Done
>
```

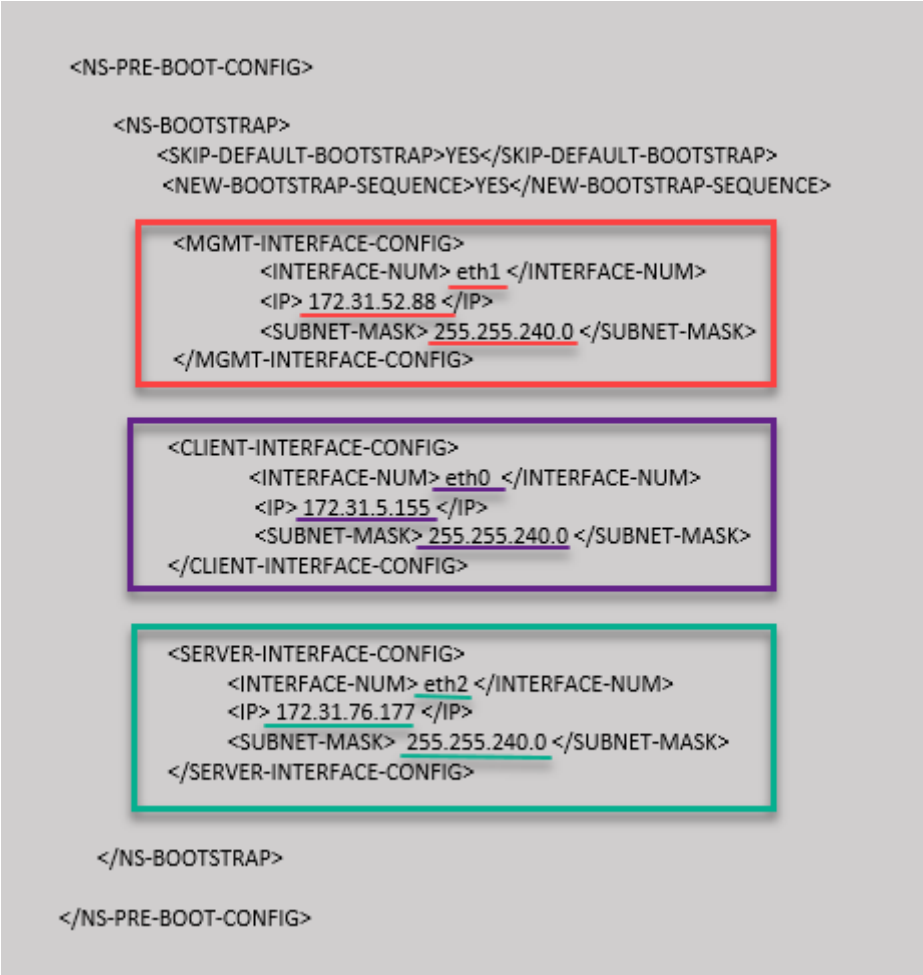
方法 **2:** インターフェイス、**IP** アドレス、およびサブネットマスクを指定してカスタムブートストラップ

管理インターフェイス、クライアント向けインターフェイス、およびサーバ向けインターフェイスと IP アドレスとサブネットマスクを指定します。

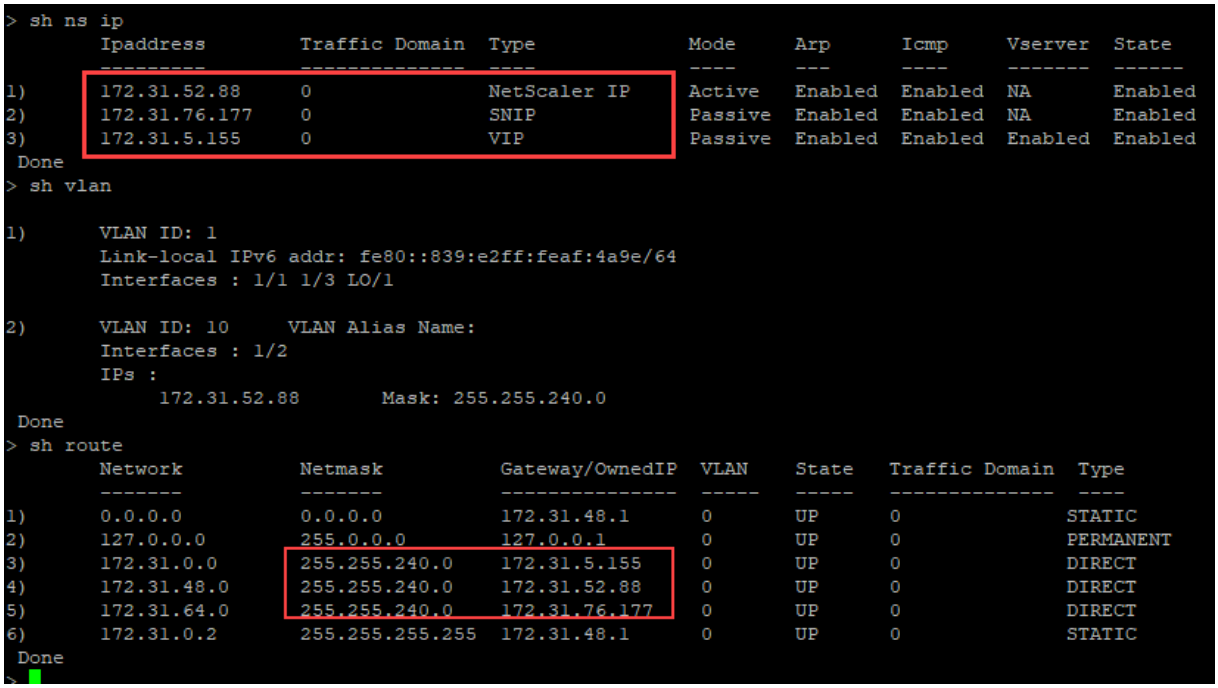
AWS のカスタムブートストラップの例

次の例では、デフォルトのブートストラップをスキップして、NetScaler アプライアンスの新しいブートストラップシーケンスを実行します。新しいブートストラップシーケンスでは、次の詳細を指定します。

- 管理インターフェイス：インターフェイス-Eth1、NSIP-172.31.52.88、およびサブネットマスク-255.255.240.0
- クライアント側インターフェイス：インターフェイス-Eth0、VIP-172.31.5.155、およびサブネットマスク-255.255.240.0。
- サーバー側インターフェイス：インターフェイス-Eth2、SNIP-172.31.76.177、サブネットマスク-255.255.240.0。



ADC CLI で`show nsip`コマンドを実行し、<NS-BOOTSTRAP>セクションで指定された新しいブートストラップシーケンスが適用されていることを確認できます。「show route」コマンドを実行して、サブネットマスクを確認できます。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  -----
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.31.76.177    0               SNIP             Passive  Enabled  Enabled  NA       Enabled
3)      172.31.5.155     0               VIP              Passive  Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.0.0       255.255.240.0    172.31.5.155     0       UP      0               DIRECT
4)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
5)      172.31.64.0      255.255.240.0    172.31.76.177    0       UP      0               DIRECT
6)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
>
```

Azure のカスタムブートストラップの例

次の例では、ADC の新しいブートストラップシーケンスが記述され、デフォルトのブートストラップがスキップされます。インターフェイスの詳細と IP アドレスとサブネットマスクを次のように指定します。

- 管理インターフェイス（eth2）、NSIP（172.27.2.53）、およびサブネットマスク（255.255.255.0）
- クライアント側インターフェイス（eth1）、VIP（172.27.1.53）、およびサブネットマスク（255.255.255.0）
- サーバー側インターフェイス (eth0)、SNIP (172.27.0.53)、およびサブネットマスク (255.255.255.0)


```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 172.27.2.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>

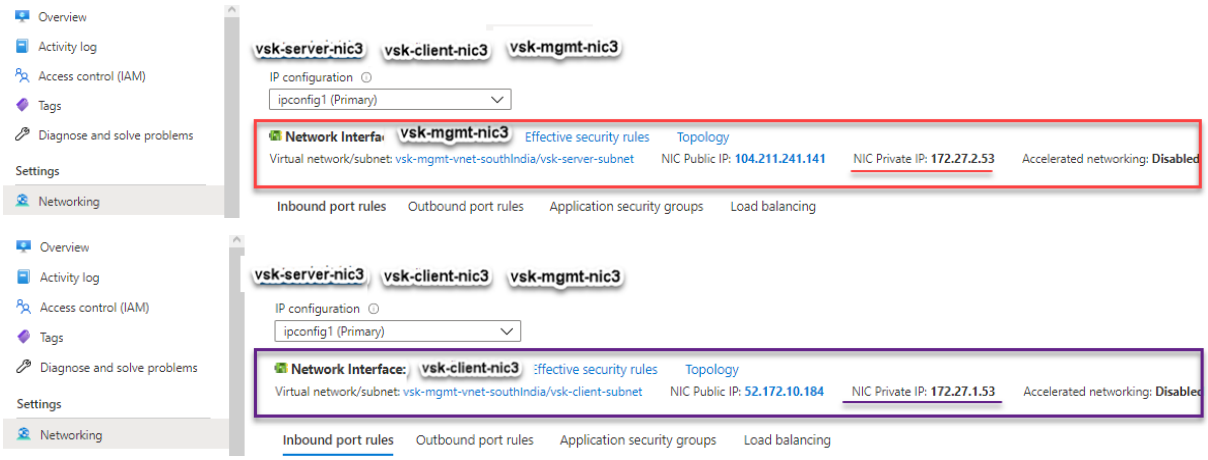
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
            <IP> 172.27.1.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
            </CLIENT-INTERFACE-CONFIG>

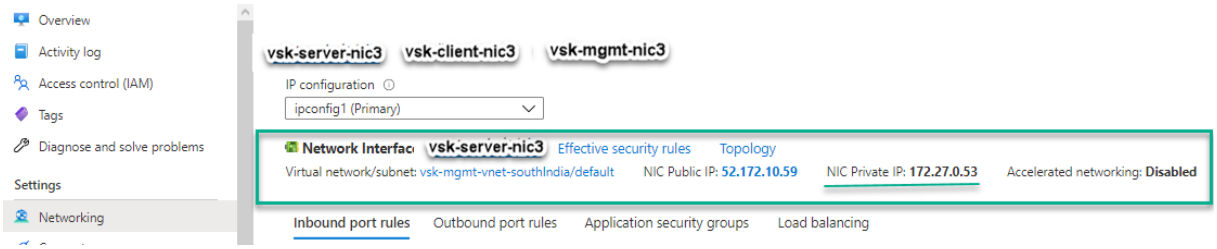
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
            <IP> 172.27.0.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

NetScaler VPX インスタンスが 3 つのネットワークインターフェイスで作成されていることがわかります。**Azure Portal > VM** インスタンス **>** ネットワークに移動し、次の図に示すように 3 つの NIC のネットワークプロパティを確認します。





ADC CLI で`show nsip`コマンドを実行し、<NS-BOOTSTRAP>セクションで指定された新しいブートストラップシーケンスが適用されていることを確認できます。「show route」コマンドを実行して、サブネットマスクを確認できます。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.53      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.53      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.27.1.53      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:c26c/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.53        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.53      0       UP      0               DIRECT
4)      172.27.1.0       255.255.255.0    172.27.1.53      0       UP      0               DIRECT
5)      172.27.2.0       255.255.255.0    172.27.2.53      0       UP      0               DIRECT
6)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
7)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
8)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
```

GCP のカスタムブートストラップの例

次の例では、ADC の新しいブートストラップシーケンスが記述され、デフォルトのブートストラップがスキップされます。インターフェイスの詳細と IP アドレスとサブネットマスクを次のように指定します。

- 管理インターフェイス（eth2）、NSIP（10.128.4.31）、およびサブネットマスク（255.255.255.0）
- クライアント側インターフェイス（eth1）、VIP（10.128.0.43）、およびサブネットマスク（255.255.255.0）
- サーバ側インターフェイス（eth0）、SNIP（10.160.0.75）、およびサブネットマスク（255.255.255.0）


```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>
        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 10.128.4.31 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
            <IP> 10.128.0.43 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </CLIENT-INTERFACE-CONFIG>
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
            <IP> 10.160.0.75 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

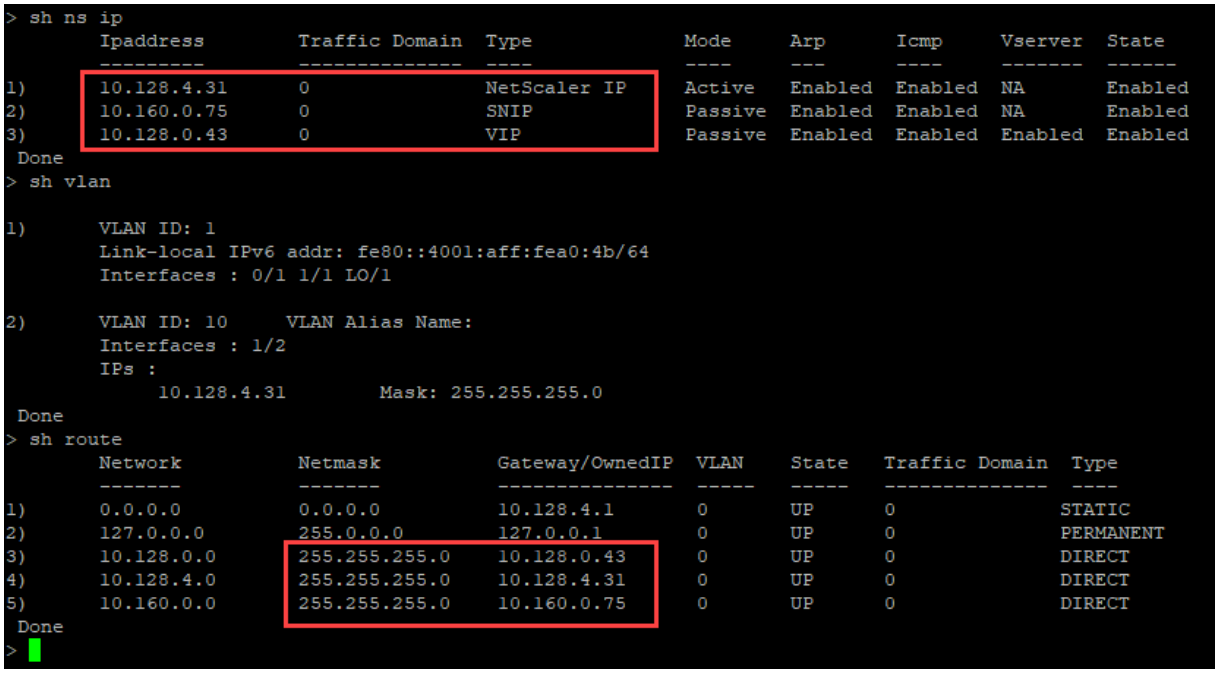
カスタムブートストラップを使用して GCP ポータルで VM インスタンスを作成した後、ネットワークインターフェイスのプロパティを次のように確認できます。

1. カスタムブートストラップ情報を指定して、作成したインスタンスを選択します。
2. ネットワークインターフェイスのプロパティに移動し、NIC の詳細を次のように確認します。

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	vsk-defnw-st-ip1 (10.160.0.75)	–	34.93.216.90 (ephemeral)	Premium	Off	View details
nic1	vsk-vpc-network-1	asia-south1-subnet-1	vsk-vpc-nw1-st-ip1 (10.128.0.43)	–	35.244.40.113 (ephemeral)	Premium		View details
nic2	vsk-vpc-network-2	asia-south1-subnet-5	vsk-nw2-st-ip-1 (10.128.4.31)	–	34.93.202.214 (ephemeral)	Premium		View details

ADC CLI で`show nsip`コマンドを実行し、<NS-BOOTSTRAP>セクションで指定された新しいブートストラップシーケンスが適用されていることを確認できます。「show route」コマンドを実行して、サブネットマスクを確認できます。



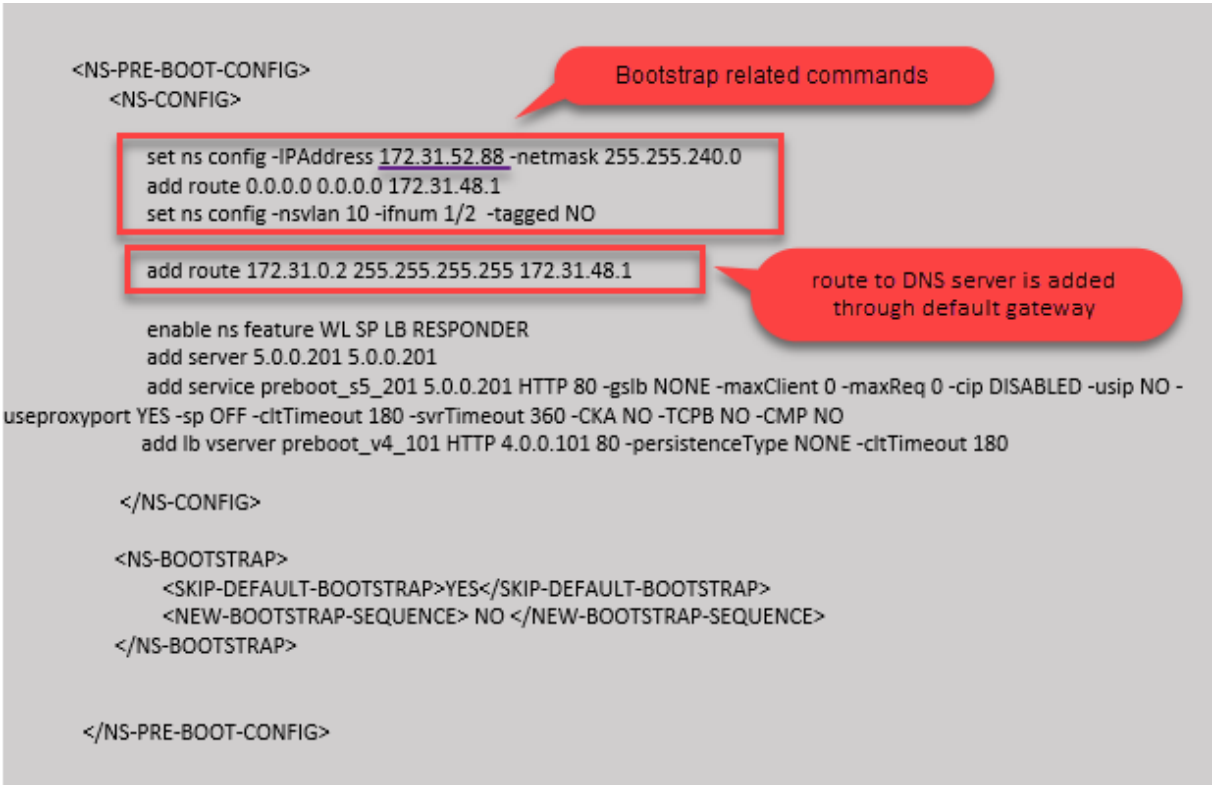
方法 **3: <NS-CONFIG>** セクションにブートストラップ関連のコマンドを指定して、カスタムブートストラップ

ブートストラップ関連のコマンドについては、<NS-CONFIG>セクションを参照してください。<NS-BOOTSTRAP>セクションでブートストラップコマンドを実行するには、<NS-CONFIG>セクションで<NEW-BOOTSTRAP-SEQUENCE>を「No」と指定する必要があります。NSIP、デフォルトルート、および NSVLAN を割り当てるコマンドも指定する必要があります。さらに、使用するクラウドに関連するコマンドも提供します。

カスタムブートストラップを提供する前に、クラウドインフラストラクチャが特定のインターフェイス構成をサポートしていることを確認してください。

AWS のカスタムブートストラップの例

この例では、ブートストラップ関連のコマンドを<NS-CONFIG>セクションで提供しています。<NS-BOOTSTRAP>セクションは、デフォルトのブートストラップがスキップされ、<NS-CONFIG>セクションで提供されるカスタムブートストラップ情報が実行されることを示します。NSIP の作成、デフォルトルートの追加、および NSVLAN の追加を行うコマンドも指定する必要があります。



前のスクリーンショットに示した設定をここからコピーできます。

```
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>

        set ns config -IPAddress 172.31.52.88 -netmask 255.255.240.0
        add route 0.0.0.0 0.0.0.0 172.31.48.1
        set ns config -nsvlan 10 -ifnum 1/2  -tagged NO
        add route 172.31.0.2 255.255.255.255 172.31.48.1

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
           maxClient 0 -maxReq 0 -cip DISABLED -usip NO - useproxyport
           YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
            -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
           persistenceType NONE -cltTimeout 180

    </NS-CONFIG>

    <NS-BOOTSTRAP>
     <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
     <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
    </NS-BOOTSTRAP>

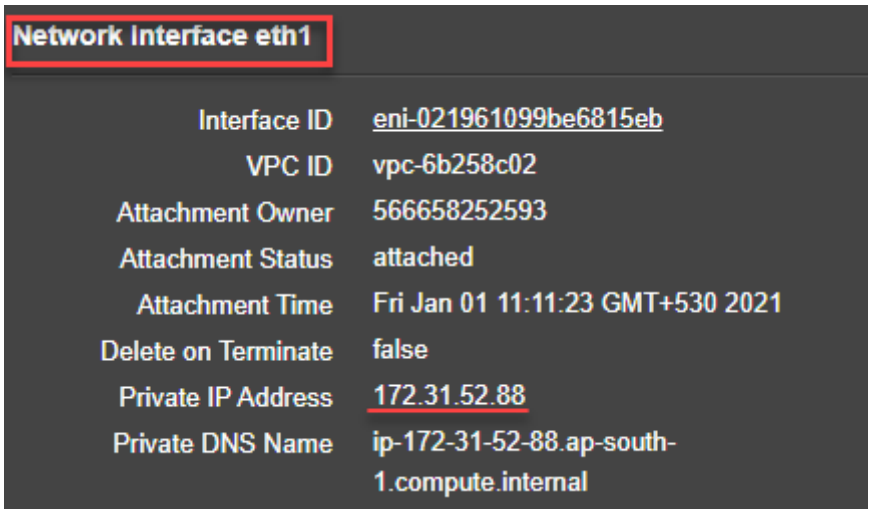

</NS-PRE-BOOT-CONFIG>
```

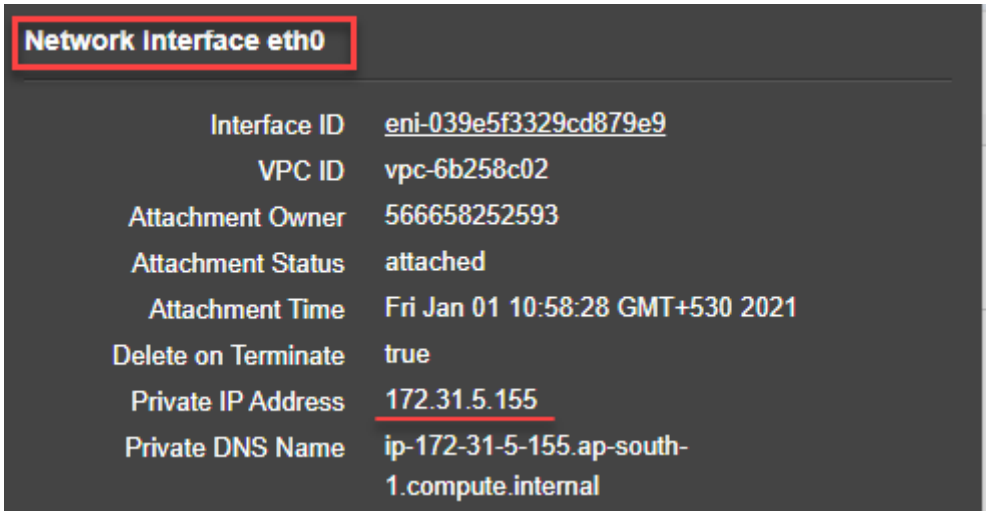


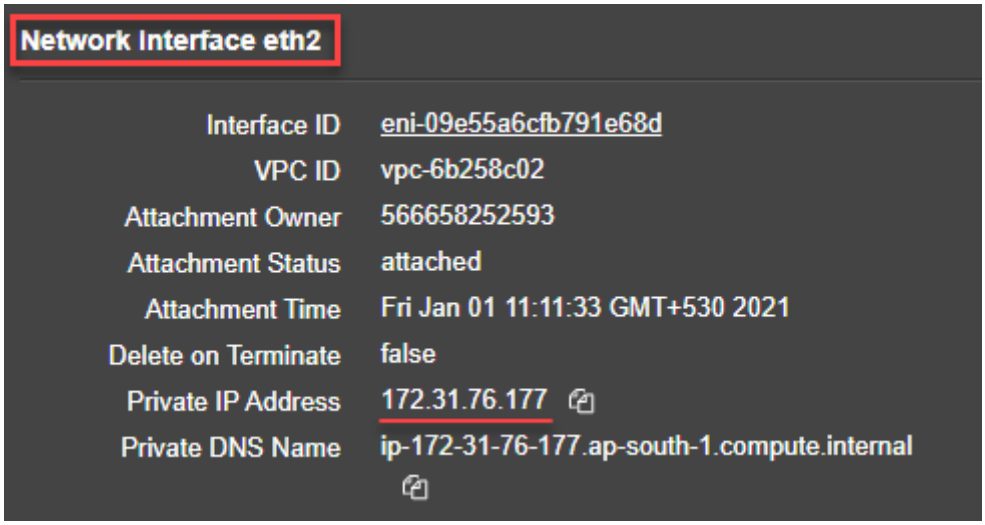
```
<!--NeedCopy-->
```

VM インスタンスの作成後、AWS ポータルで、ネットワークインターフェイスのプロパティを次のように確認できます。

1. [**AWS Portal] > [EC2** インスタンス] に移動し、カスタムブートストラップ情報を指定して作成したインスタンスを選択します。
2. [説明] タブでは、次の図に示すように、各ネットワークインタフェースのプロパティを確認できます。







ADC CLI で`show nsip`コマンドを実行し、ADC アプライアンスの初回起動時に NetScalerVPX インスタンスに適用されるネットワークインターフェイスを確認できます。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

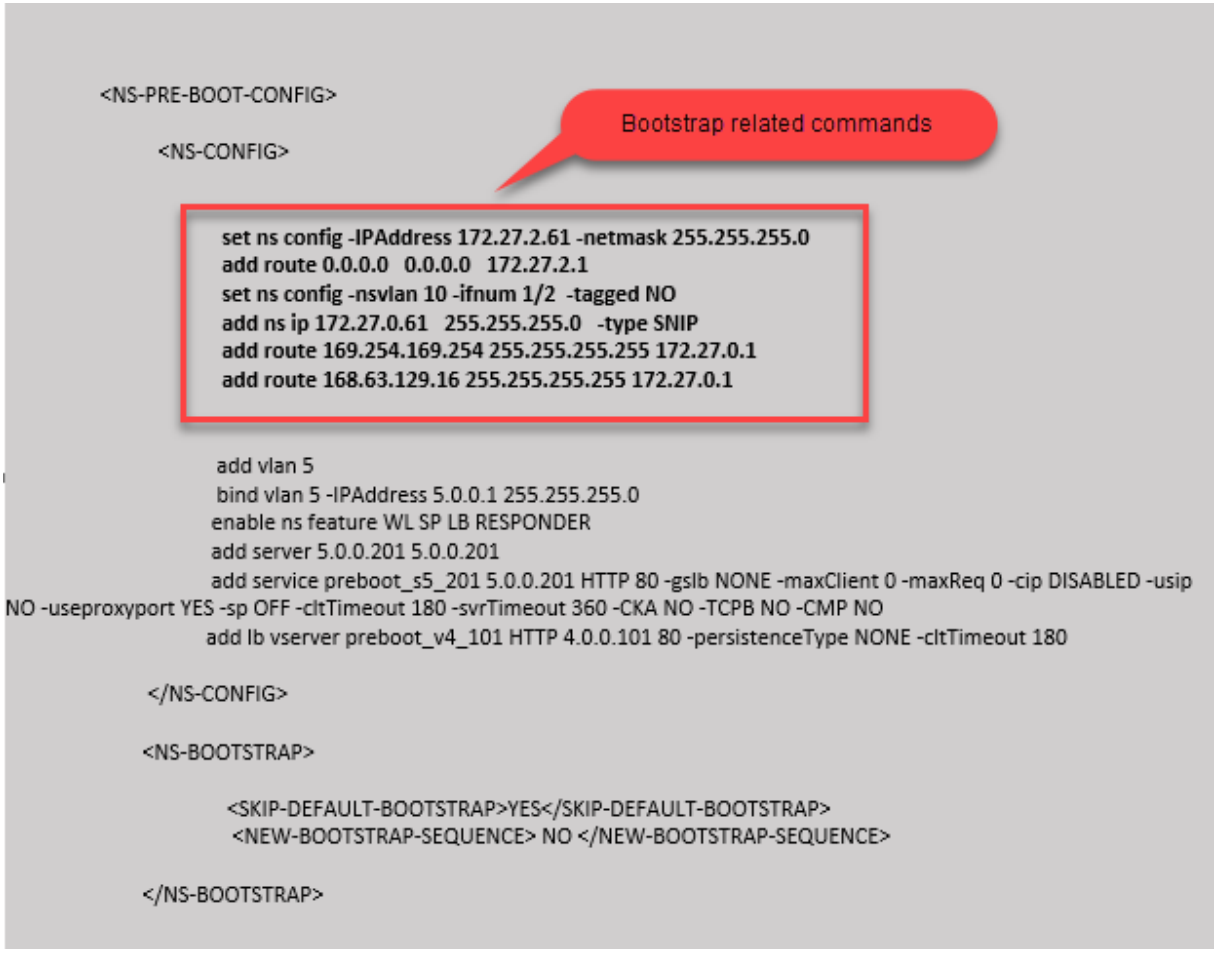
2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
4)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
>
```

Azure のカスタムブートストラップの例

この例では、ブートストラップ関連のコマンドを<NS-CONFIG>セクションで提供しています。<NS-BOOTSTRAP>セクションは、デフォルトのブートストラップがスキップされ、<NS-CONFIG>セクションで提供されるカスタムブートストラップ情報が実行されることを示します。

> 注:
>
> Azure クラウドの場合、インスタンスメタデータサーバー (IMDS) と DNS サーバーはプライマリインターフェイス (Eth0) を介してのみアクセスできます。したがって、Eth0 インターフェイスが管理インターフェイス

（NSIP）として使用されない場合、Eth0 インターフェイスは、少なくとも IMDS または DNS アクセスを動作させるには、SNIP として設定する必要があります。Eth0 のゲートウェイを経由する IMDS エンドポイント（169.254.169.254）および DNS エンドポイント（168.63.129.16）へのルートも追加する必要があります。




```
<NS-PRE-BOOT-CONFIG>

   <NS-CONFIG>

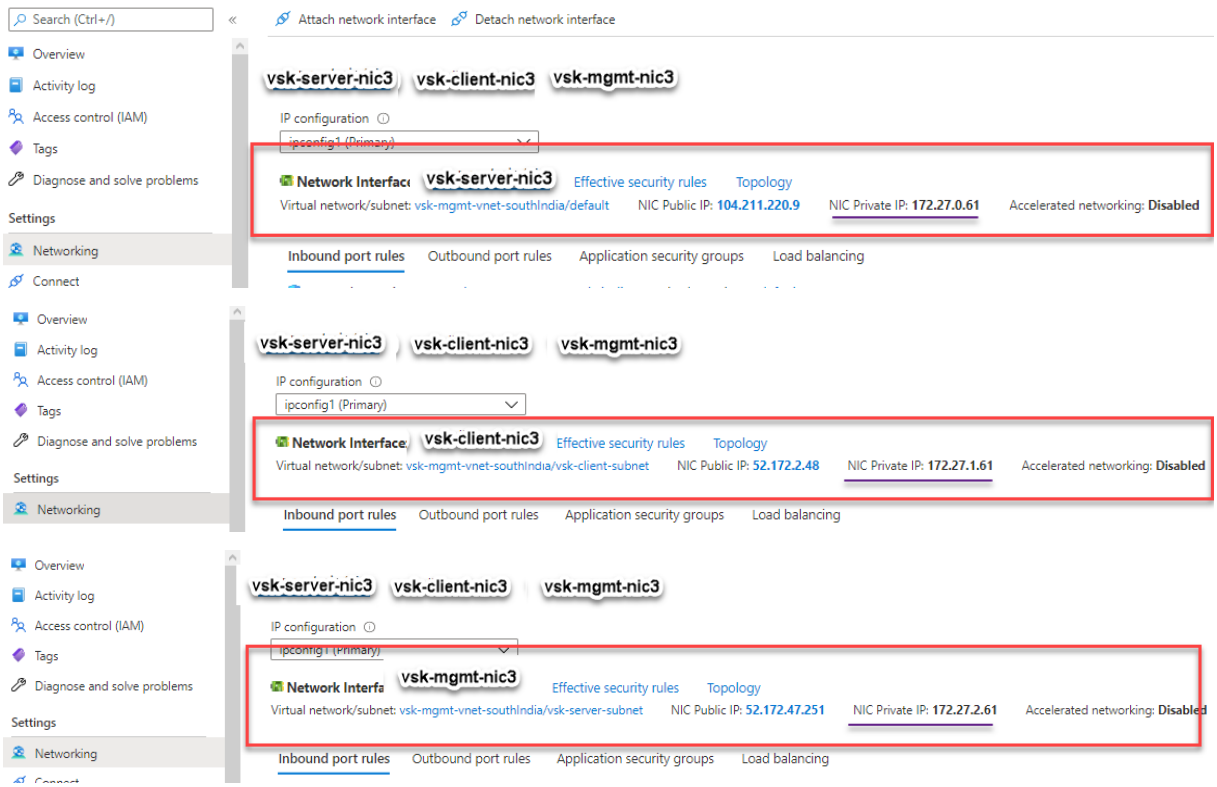
        set ns config -IPAddress 172.27.2.61 -netmask 255.255.255.0
        add route 0.0.0.0   0.0.0.0   172.27.2.1
        set ns config -nsvlan 10 -ifnum 1/2  -tagged NO
        add ns ip 172.27.0.61   255.255.255.0   -type SNIP
        add route 169.254.169.254 255.255.255.255 172.27.0.1
        add route 168.63.129.16 255.255.255.255 172.27.0.1

        add vlan 5
        bind vlan 5 -IPAddress 5.0.0.1 255.255.255.0
        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
           maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport
           YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
```

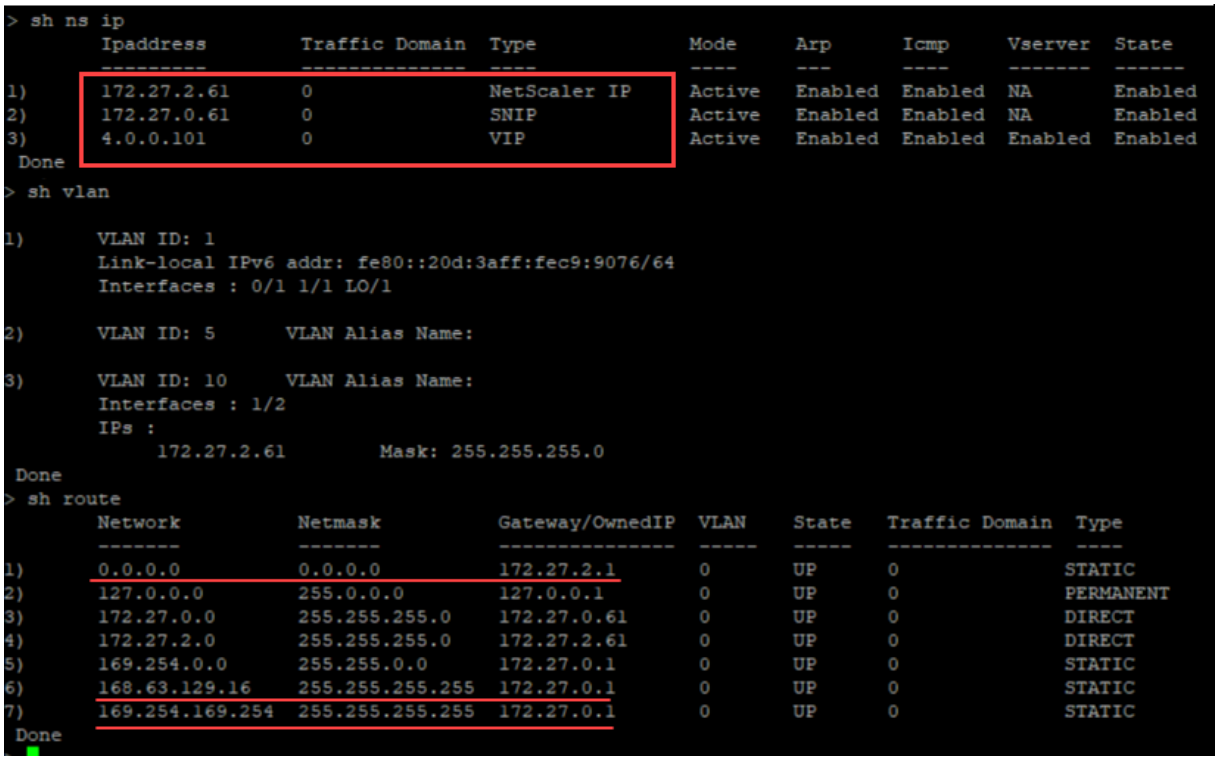


```
                -CMP NO
17          add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
               persistenceType NONE -cltTimeout 180
18
19      </NS-CONFIG>
20
21      <NS-BOOTSTRAP>
22
23      <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
24      <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
25
26      </NS-BOOTSTRAP>
27
28  </NS-PRE-BOOT-CONFIG>
29  <!--NeedCopy-->
```

NetScaler VPX インスタンスが 3 つのネットワークインターフェイスで作成されていることがわかります。**Azure Portal > VM** インスタンス **>** ネットワークに移動し、次の図に示すように 3 つの NIC のネットワークプロパティを確認します。



ADC CLI で`show nsip`コマンドを実行し、<NS-BOOTSTRAP>セクションで指定された新しいブートストラップシーケンスが適用されていることを確認できます。「show route」コマンドを実行して、サブネットマスクを確認できます。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode       Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----       ---      ----     -------  -----
1)      172.27.2.61      0               NetScaler IP     Active     Enabled  Enabled  NA       Enabled
2)      172.27.0.61      0               SNIP             Active     Enabled  Enabled  NA       Enabled
3)      4.0.0.101        0               VIP              Active     Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

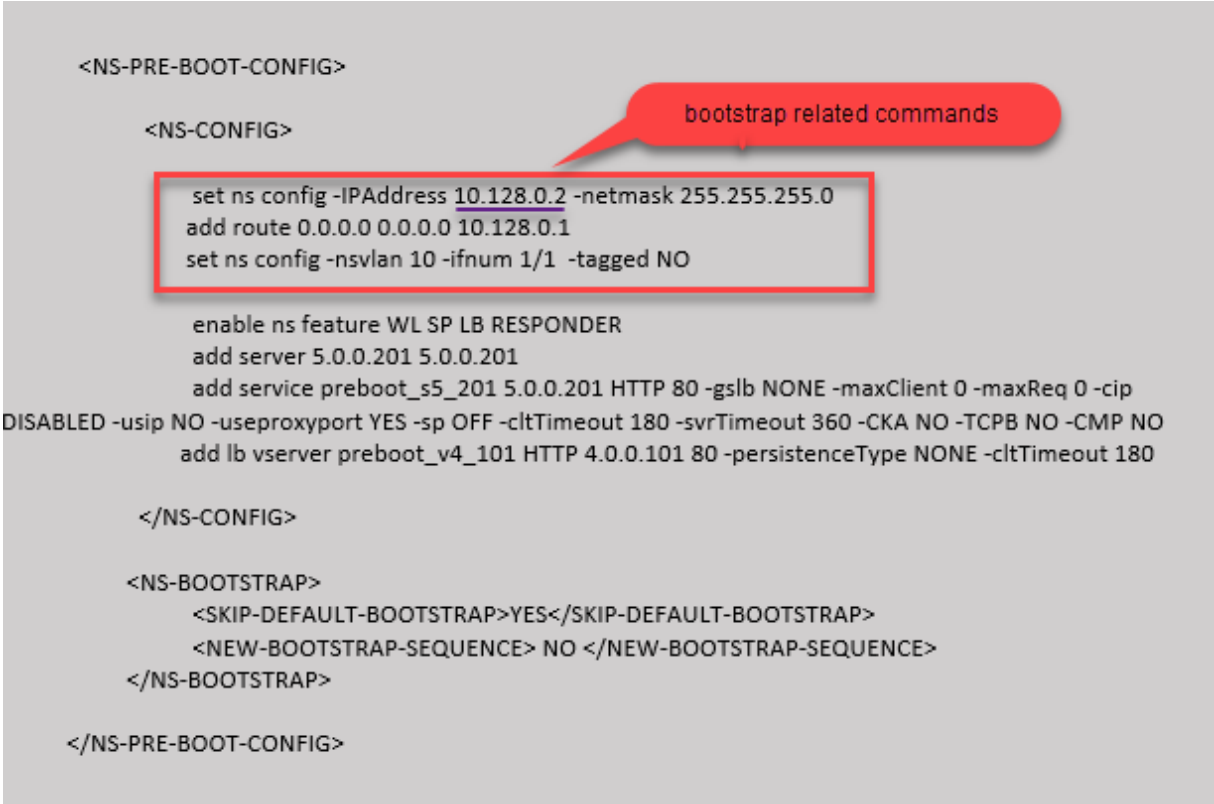
1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:9076/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 5      VLAN Alias Name:

3)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.61        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.61      0       UP      0               DIRECT
4)      172.27.2.0       255.255.255.0    172.27.2.61      0       UP      0               DIRECT
5)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
6)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
7)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
```

GCP のカスタムブートストラップの例

この例では、ブートストラップ関連のコマンドを<NS-CONFIG>セクションで提供しています。<NS-BOOTSTRAP>セクションは、デフォルトのブートストラップがスキップされ、<NS-CONFIG>セクションで提供されるカスタムブートストラップ情報が適用されることを示します。



前のスクリーンショットに示した設定をここからコピーできます。

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>

        set ns config -IPAddress 10.128.0.2 -netmask 255.255.255.0
        add route 0.0.0.0 0.0.0.0 10.128.0.1
        set ns config -nsvlan 10 -ifnum 1/1  -tagged NO

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport
            YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
             -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
            persistenceType NONE -cltTimeout 180

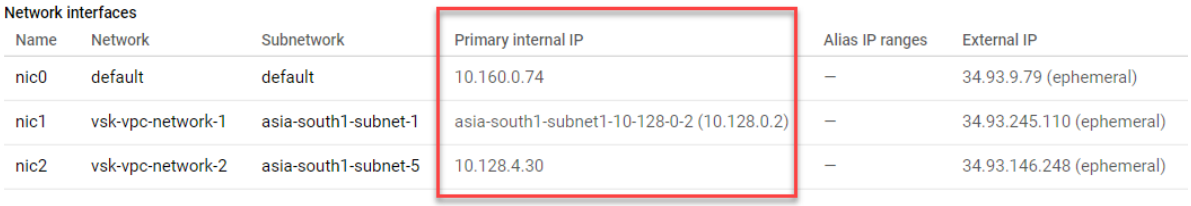
    </NS-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```


カスタムブートストラップを使用して GCP ポータルで VM インスタンスを作成した後、ネットワークインターフェイスのプロパティを次のように確認できます。

1. カスタムブートストラップ情報を指定して、作成したインスタンスを選択します。
2. [Network Interface] プロパティに移動し、図に示すように NIC の詳細を確認します。



ADC CLI で`show nsip`コマンドを実行し、**ADC** アプライアンスの最初の起動時に前の<NS-CONFIG>セクションで説明した設定が適用されていることを確認できます。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.128.0.2       0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:4a/64
        Interfaces : 0/1 1/2 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/1
        IPs :
             10.128.0.2         Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.0.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.2       0       UP      0               DIRECT
 Done
```

AWS および Azure での NIC のアタッチとデタッチによる影響

AWS と Azure には、ネットワークインターフェイスをインスタンスにアタッチし、ネットワークインターフェイスをインスタンスからデタッチするオプションが用意されています。インターフェイスをアタッチまたはデタッチすると、インターフェイスの位置が変わることがあります。そのため、Citrix では NetScaler VPX インスタンスからインターフェイスをデタッチしないことをお勧めします。カスタムブートストラップが構成されているときにインターフェイスをデタッチまたは接続すると、NetScaler VPX インスタンスは、管理インターフェイスの位置で新しく使用可能なインターフェイスのプライマリ IP を NSIP として再割り当てします。デタッチしたインターフェイスの後に使用可能なインターフェイスがない場合は、最初のインターフェイスが NetScaler VPX インスタンスの管理インターフェイスになります。

たとえば、NetScaler VPX インスタンスは、Eth0（SNIP）、Eth1（NSIP）、および Eth2（VIP）の 3 つのインターフェイスで起動されます。管理インターフェイスであるインスタンスから Eth1 インターフェイスをデタッチすると、ADC は次の使用可能なインターフェイス（Eth2）を管理インターフェイスとして設定します。そのため、NetScaler VPX インスタンスには引き続き Eth2 インターフェイスのプライマリ IP を介してアクセスされます。Eth2 も使用できない場合は、残りのインターフェイス（Eth0）が管理インターフェイスになります。そのため、NetScaler VPX インスタンスには引き続きアクセスできます。

Eth0 (SNIP)、Eth1 (VIP)、Eth2 (NSIP) の異なるインターフェイスの割り当てを考えてみましょう。Eth2（NSIP）をデタッチすると、Eth2 の後に新しいインターフェイスが使用できないため、最初のインターフェイス（Eth0）が管理インターフェイスになります。

パブリッククラウドプラットフォームでの **SSL-TPS** パフォーマンスを向上させる

August 15, 2023

パケットエンジン (PE) の重みを均等に分散することで、AWS と GCP クラウドで SSL-TPS のパフォーマンスを向上させることができます。この機能を有効にすると、HTTP スループットが約 10 ～12% わずかに低下する可能性があります。

AWS および GCP クラウドでは、10～16 個の vCPU を持つ NetScaler ADC VPX インスタンスでは、PE の重みがデフォルトで均等に分散されるため、パフォーマンスの向上は見られません。

> 注:
>
> Azure クラウドでは、PE の重みはデフォルトで均等に分散されます。この機能によって Azure インスタンスのパフォーマンスは向上しません。

NetScaler CLI を使用して **PE** モードを構成する

PE モードを設定したら、設定変更を有効にするためにシステムをリブートする必要があります。

コマンドプロンプトで入力します。

```
set cpuparam pemode [CPUBOUND | Default]
<!--NeedCopy-->
```

PE モードが CPUBOUND に設定されている場合、PE の重みは均等に分散されます。
PE モードが DEFAULT に設定されている場合、PE の重みはデフォルト値に設定されます。

> 注:
>
> このコマンドはノード固有です。高可用性またはクラスタセットアップでは、各ノードでコマンドを実行する

> 必要があります。CLIP でコマンドを実行すると、次のエラーが発生します。
>
> ```
> Operation not permitted on CLIP
> ```

設定されている PE モードの状態を表示するには、次のコマンドを実行します。

```
show cpuparam
<!--NeedCopy-->
```

例:

```
> show cpuparam
    Pemode:  CPUBOUND
 Done
<!--NeedCopy-->
```

クラウド内の **NetScaler ADC** アプライアンスの初回起動時に **PE** モード構成を適用する

クラウド内の NetScaler ADC アプライアンスの初回起動時に PE モード構成を適用するには、カスタムスクリプトを使用して`/nsconfig/.cpubound.conf`ファイルを作成する必要があります。詳しくは、「クラウドでの NetScaler ADC アプライアンスの初回起動時に NetScaler ADC VPX 構成を適用する」を参照してください。

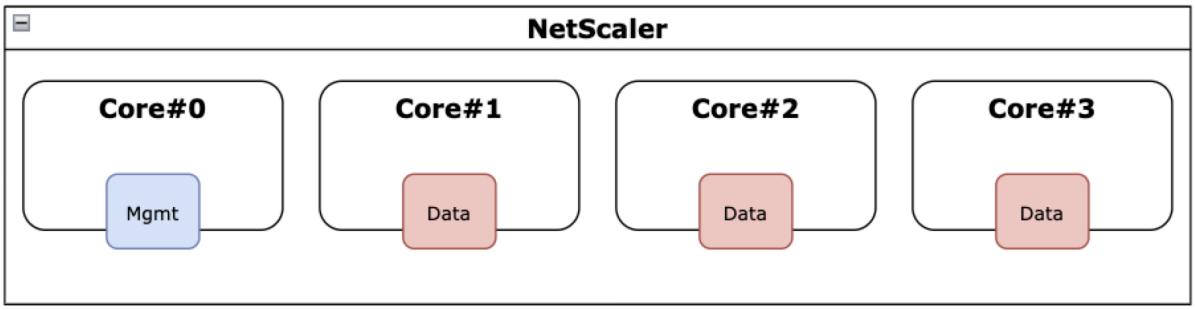
パブリッククラウド上の **NetScaler VPX** 同時マルチスレッドを構成する

February 15, 2024

NetScaler は、管理とデータプレーン機能にさまざまな専用コアを使用します。通常、1 つのコアが管理プレーン機能に割り当てられます。使用可能な残りのコアはデータプレーン機能に割り当てられます。

以下の画像は、4 コアの NetScaler VPX を簡略化した図を示しています。

図 1: 4 コアシステムでの NetScaler 管理とデータプレーンワークロード

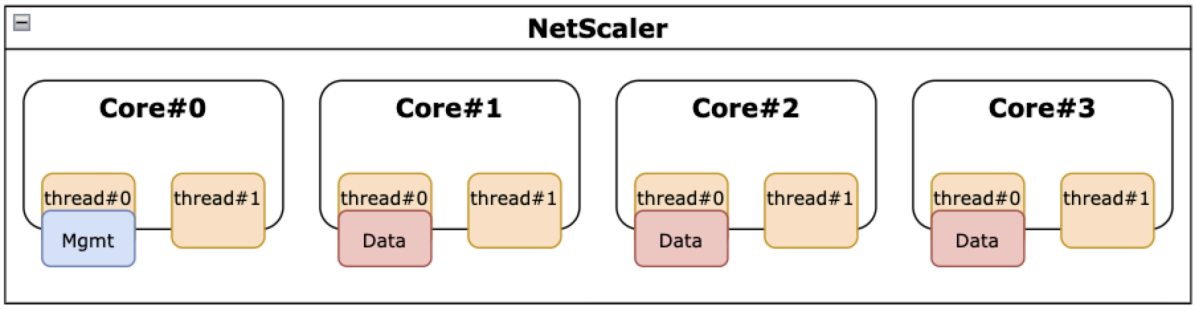


上の図は、使用可能なコア全体にわたる NetScaler 機能の分布を示していますが、基盤となるハードウェアを必ずしも正確に表しているわけではありません。最新の x86 CPU のほとんどは、Intel ハイパースレッディング（HT）ま

たは AMD 同時マルチスレッディング（SMT）として商業的に知られている機能により、物理コアあたり 2 つの論理コアを備えています。

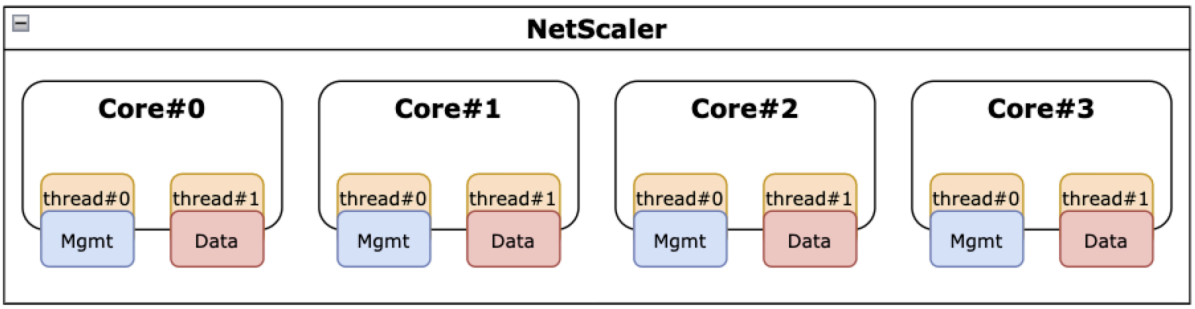
次の画像は、SMT が無効になっている最新の CPU 上で実行されている NetScaler VPX を示しています。各 CPU コアは、一般にスレッドと呼ばれる 2 つ以上の論理 CPU に分割されます。各スレッドには、それぞれ独自の複製リソースセットとパーティション化されたリソースの一部があり、兄弟スレッドと共有リソースをめぐって競合します。

図 2：SMT を無効にした 4 コア/8 スレッドシステムでの NetScaler 管理とデータプレーンワークロード



次の画像は、SMT が有効になっている最新の CPU 上で実行されている NetScaler VPX を示しています。

図 3：SMT が有効になっている 4 コアシステムでの NetScaler 管理とデータプレーンワークロード



SMT を有効にすると、次の点で NetScaler のパフォーマンスが向上します：

- すべての物理コアでデータプレーン機能を実行しています。
- 管理プレーン機能を兄弟スレッドに移動します。
- 管理プレーン機能がデータプレーン機能のパフォーマンスを損なうことを防ぐために、柔軟なリソース制限メカニズムを導入しました。

SMT サポートマトリックス

SMT をサポートする VPX プラットフォーム、クラウドインスタンスタイプ、NetScaler のバージョンを次の表に示します。

VPX プラットフォーム	インスタンスタイプ	NetScaler VPX バージョン
AWS	M5, m5n, c5, c5n	13.1-48.x およびそれ以降

> 注:
>
> SMT 機能を有効にすると、サポートされているタイプの NetScaler VPX パフォーマンスが向上します。

制限事項

SMT 機能により、NetScaler アプライアンスが利用できる仮想 CPU の数が実質的に倍増します。NetScaler アプライアンスがライセンス制限を使用できるようにするには、ライセンス制限を考慮する必要があります。

たとえば、図 3 に示されている NetScaler VPX について考えてみます。スループットベースのライセンスを使用する場合、8 個の vCPU を有効にするには、SMT 機能を備えた 10 Gbps 以上のライセンスが必要です。以前は、4 つの vCPU を有効にするには 1 Gbps のライセンスで十分でした。vCPU ライセンスを使用する場合、正常に動作するためには、2 倍の数の vCPU のライセンスをチェックアウトするように NetScaler VPX を構成する必要があります。このトピックに関する詳細なガイダンスについては、NetScaler テクニカルサポートにお問い合わせください。

SMT を設定して下さい

SMT 機能を有効にする前に、プラットフォームがこの機能をサポートしていることを確認してください。前のセクションのサポートマトリックスの表を参照してください。

SMT 機能を有効にするには、次の手順に従います:

1. 「/nsconfig」ディレクトリの下に名前 `.smt_handling` の空のファイルを作成します。
2. 現在の設定を保存します。
3. NetScaler VPX インスタンスを再起動します。

```
nscli> shell touch /nsconfig/.smt_handling
  Done
nscli> reboot
Are you sure you want to restart NetScaler (Y/N)? [N]:Y
Done
<!--NeedCopy-->
```

4. 再起動後、NetScaler は機能が使用可能で有効であることを示します。

```
smt_handling is set to  “1”

> shell sysctl -a | grep smt_handling
netscaler.smt_handling_platform: 1
netscaler.smt_handling: 1
<!--NeedCopy-->
```


SMT 機能を無効にするには、次の手順に従います:

1. .smt_handling ファイルを削除します。
2. NetScaler VPX インスタンスを再起動します。

```
shell rm -f /nsconfig/.smt_handling
  Done

reboot

Are you sure you want to restart NetScaler (Y/N)? [N]:Y
Done
<!--NeedCopy-->
```

3. 再起動後、NetScaler は機能が使用可能だが無効になっていることを示します。

```
> shell sysctl -a | grep smt_handling
netscaler.smt_handling_platform: 1
netscaler.smt_handling: 0
<!--NeedCopy-->
```

トラブルシューティング

sysctl シェルコマンドを実行して、SMT 機能のステータスを確認します。

```
```
> shell sysctl -a | grep smt_handling
>
<!--NeedCopy--> ```
```

このコマンドは、次の出力のいずれかを返すことができます。

- SMT 機能がありません。

  sysctlコマンドは出力を返しません。

- SMT 機能はサポートされていません。

  SMT 機能は次のいずれかの理由でサポートされていません:

  - お使いの NetScaler VPX は 13.1-48.x または 14.1-12.x より古いです。
  - お使いのクラウドは SMT をサポートしていません。
  - お使いの VM インスタンスタイプは SMT をサポートしていません。たとえば、vCPU 数が 8 を超えています。

    ```
    > shell sysctl -a | grep smt_handling
    netscaler.smt_handling_platform: 0  (indicates not supported)
    netscaler.smt_handling: 0  (indicates not enabled)
    <!--NeedCopy-->
    ```


    ```

- SMT 機能はサポートされていますが、有効になっていません。

```
> shell sysctl -a | grep smt_handling
netscaler.smt_handling_platform: 1 (available)
netscaler.smt_handling: 0 (not enabled)
<!--NeedCopy-->
```

NetScaler VPX インスタンスをベアメタルサーバーにインストールする

August 15, 2023

ベアメタルとは、クラウド環境に完全に統合された、物理的に分離された完全専用の物理サーバーです。シングルテナントサーバーとも呼ばれます。シングルテナントを使用すると、ノイズの多いネイバー効果を回避できます。ベアメタルでは、自分が唯一のユーザーなので、ノイズの多いネイバー効果は発生しません。

ハイパーバイザーがインストールされたベアメタルサーバーは、サーバー上に仮想マシンを作成するための管理スイートを提供します。ハイパーバイザーはアプリケーションをネイティブに実行しません。その目的は、ワークロードを個別の仮想マシンに仮想化して、仮想化の柔軟性と信頼性を高めることです。

NetScaler VPX インスタンスをベアメタルサーバーにインストールするための前提条件

ベアメタルサーバーは、それぞれのハイパーバイザーのすべてのシステム要件を満たすクラウドベンダーから入手する必要があります。

NetScaler VPX インスタンスをベアメタルサーバーにインストールします

NetScaler VPX インスタンスをベアメタルサーバーにインストールするには、まずクラウドベンダーから十分なシステムリソースを備えたベアメタルサーバーを入手する必要があります。そのベアメタルサーバーでは、NetScaler VPX インスタンスを展開する前に、Linux KVM、VMware ESX、Citrix Hypervisor、Microsoft Hyper-V などのサポートされているハイパーバイザーのいずれかをインストールして構成する必要があります。

NetScaler VPX インスタンスでサポートされているさまざまなハイパーバイザーと機能のリストの詳細については、「サポートマトリックスと使用ガイドライン」を参照してください。

さまざまなハイパーバイザーに NetScaler ADC VPX インスタンスをインストールする方法の詳細については、それぞれのドキュメントを参照してください。

- **Citrix Hypervisor**: Citrix Hypervisor への NetScaler ADC VPX インスタンスのインストールを参照してください。
- **VMware ESX**: VMware ESX への NetScaler ADC VPX インスタンスのインストールを参照してください。

- **Microsoft Hyper-V**: Microsoft Hyper-V サーバーへの NetScaler ADC VPX インスタンスのインストールを参照してください。
- **Linux KVM** プラットフォーム: Linux-KVM プラットフォームへの NetScaler ADC VPX インスタンスのインストールを参照してください。

Citrix Hypervisor に NetScaler ADC VPX インスタンスをインストールする

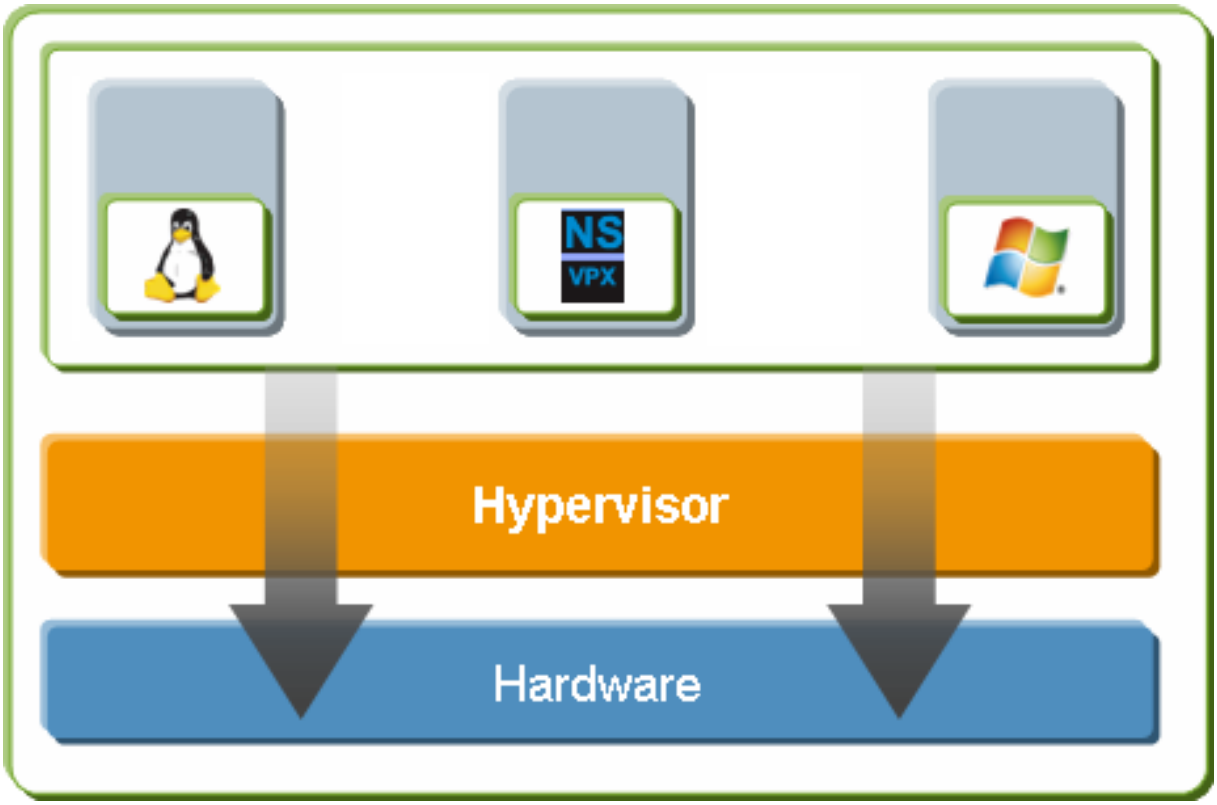
August 15, 2023

Citrix Hypervisor に VPX インスタンスをインストールするには、まず適切なシステムリソースを持つマシンに Hypervisor をインストールする必要があります。NetScaler VPX インスタンスのインストールを実行するには、Citrix XenCenter を使用します。Citrix XenCenter は、ネットワーク経由で Hypervisor ホストに接続できるリモートマシンにインストールする必要があります。

Hypervisor の詳細については、Citrix Hypervisor のドキュメントを参照してください。

次の図は、Hypervisor 上の NetScaler ADC VPX インスタンスのベアメタルソリューションアーキテクチャを示しています。

フィギュア。Citrix Hypervisor 上の NetScaler ADC VPX インスタンス



Hypervisor に **NetScaler ADC VPX** インスタンスをインストールするための前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- 最小要件を満たすハードウェアに Hypervisor バージョン 6.0 以降をインストールします。
- 最小システム要件を満たす管理ワークステーションに XenCenter をインストールします。
- 仮想アプライアンスのライセンスファイルを取得します。仮想アプライアンスライセンスの詳細については、『NetScaler ライセンスガイド』を参照してください。

Hypervisor のハードウェア要件

次の表は、NetScaler VPX インスタンスを実行するハイパーバイザープラットフォームの最小ハードウェア要件を示しています。

テーブル **1**. nCore VPX インスタンスを実行する Hypervisor の最小システム要件

Component	条件
CPU	仮想化アシスト (Intel-VT) が有効になっている 64 ビット x86 CPU が 2 つ以上あります。NetScaler VPX インスタンスを実行するには、Hypervisor ホストで仮想化のハードウェアサポートを有効にする必要があります。仮想化サポートの BIOS オプションが無効になっていないことを確認してください。詳細については、BIOS のドキュメントを参照してください。
RAM	3 GB
ディスク領域	40 GB のディスク容量を持つローカル接続ストレージ (PATA、SATA、SCSI)。注:Hypervisor のインストールでは、Hypervisor ホストコントールドメインに 4 GB のパーティションが作成されます。残りのスペースは、NetScaler VPX インスタンスやその他の仮想マシンに使用できます。
NIC	1 Gbps NIC × 1、推奨: 2 つの 1 Gbps NIC

Hypervisor のインストールについては、http://support.citrix.com/product/xens/の Hypervisor のドキュメントを参照してください。

次の表に、Hypervisor が各 nCore VPX 仮想アプライアンスに提供する必要がある仮想コンピューティングリソースを示します。

テーブル **2**. nCore VPX インスタンスの実行に必要な最小仮想コンピューティングリソース

Component	条件
メモリ	2 GB
仮想 CPU（VCPU）	2
仮想ネットワークインターフェイス	2

> 注:
>
> NetScaler VPX インスタンスを本番環境で使用する場合、スケジューリング動作とネットワーク遅延を改善するために、（仮想マシンプロパティの）CPU 優先度を最高レベルに設定することを Citrix では推奨しています。

XenCenter のシステム要件

XenCenter は、Windows のクライアントアプリケーションです。Hypervisor ホストと同じマシンでは実行できません。最小システム要件と XenCenter のインストールについて詳しくは、Hypervisor に関する次のドキュメントを参照してください。

- システム要件
- インストール

XenCenter を使用して **NetScaler VPX** インスタンスをハイパーバイザーにインストールする

Hypervisor と XenCenter をインストールして構成したら、XenCenter を使用して Hypervisor に仮想アプライアンスをインストールできます。インストールできる仮想アプライアンスの数は、Hypervisor を実行しているハードウェアで使用可能なメモリの量によって異なります。

XenCenter を使用して Hypervisor に NetScaler ADC VPX インスタンスをインストールするには、次の手順に従います。

1. ワークステーションで **XenCenter** を起動します。
2. ［サーバー］メニューの ［追加］を選択します。
3. [新規サーバーの追加] ダイアログボックスのホスト名テキストボックスに、接続するハイパーバイザーの IP アドレスまたは DNS 名を入力します。
4. ［**User Name**］ボックスと［**Password**］ボックスに管理者資格情報を入力して、［**Connect**］をクリックします。Hypervisor 名がナビゲーションペインに表示され、Hypervisor が接続されていることを示します。
5. ナビゲーションペインで、NetScaler VPX インスタンスをインストールする Hypervisor の名前をクリックします。
6. ［**VM**］メニューの［**Import**］を選択します。

7. インポートダイアログボックスのインポートファイル名で、NetScaler VPX インスタンス.xvaイメージファイルを保存した場所を参照します。[エクスポートされた仮想マシン] オプションが選択されていることを確認し、[次へ] をクリックします。
8. 仮想アプライアンスをインストールするハイパーバイザーを選択し、[次へ] をクリックします。
9. 仮想アプライアンスを保存するローカルストレージリポジトリを選択して［Import］をクリックし、インポート処理を開始します。
10. 必要に応じて、仮想ネットワークインターフェイスを追加、変更、または削除できます。完了したら［Next］をクリックします。
11. [完了] をクリックしてインポートプロセスを完了します。

    注: インポートプロセスのステータスを表示するには、[ログ] タブをクリックします。
12. 別の仮想アプライアンスをインストールする場合は、手順 5 〜11 を繰り返します。

> 注:
>
> VPX インスタンスの初期構成後、アプライアンスを最新のソフトウェアリリースにアップグレードする場合は、「システムソフトウェアのアップグレードまたはダウングレード」を参照してください。

シングルルート I/O 仮想化 (SR-IOV) ネットワークインターフェイスを使用するように VPX インスタンスを構成する

August 15, 2023

NetScaler VPX インスタンスを Citrix Hypervisor にインストールして構成したら、SR-IOV ネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

次の NIC がサポートされています。

- Intel 82599 10G
- Intel X710 10G
- Intel XL710 40G

制限事項

Citrix Hypervisor は、SR-IOV インターフェイスの一部の機能をサポートしていません。Intel 82599、Intel X710、および Intel XL710 NIC の制限は、次のセクションに記載されています。

Intel 82599 NIC の制限事項

Intel 82599 NIC は次の機能をサポートしていません。

- L2 モード切り替え
- クラスタリング
- 管理パーティション化［共有 VLAN モード］
- 高可用性［アクティブ/アクティブモード］
- ジャンボフレーム
- クラスター環境の IPv6 プロトコル

Intel X710 10G および **Intel XL710 40G NIC** の制限事項

Intel X710 10 G と Intel XL710 40G NIC には次の制限があります。

- L2 モードの切り替えはサポートされていません。
- 管理パーティショニング（共有 VLAN モード）はサポートされていません。
- クラスタでは、XL710 NIC がデータインターフェイスとして使用されている場合、ジャンボフレームはサポートされません。
- インターフェイスが切断され、再接続されると、インターフェイスリストの順序が変わります。
- スピート、デュプレックス、オートネゴシエーションなどの Interface パラメーター構成はサポートされません。
- Intel X710 10 G と Intel XL710 40G NIC の両方で、インターフェイスは 40/x インターフェイスとして表示されます。
- VPX インスタンスでサポートできる Intel X710/XL710 SR-IOV インターフェイスは 16 個までです。

> 注:
>
> Intel X710 10G および Intel XL710 40G NIC が IPv6 をサポートするには、Citrix Hypervisor ホストで次のコマンドを入力して、仮想機能（VF）のトラストモードを有効にします。
>
> ```
> # ip link set <PNIC> <VF> trust on
> ```
>
> 例:
>
> ```
> # ip link set ens785f1 vf 0 trust on
> ```

Intel 82599 NIC の前提条件

Citrix Hypervisor ホストで、次のことを確認してください。

- Intel 82599 NIC (NIC) をホストに追加します。

- **/etc/modprobe.d/blacklist.conf** ファイルに次のエントリを追加して、`ixgbevf`ドライバを一覧表示することを禁止します。

  blacklist ixgbevf

- **/etc/modprobe.d/blacklist.conf** ファイルで、以下のエントリを追加して、SR-IOV Virtual Functions（VF）を有効にします。

  options ixgbe max_vfs=*<number_of_VFs>*

  ここで、*<number_VFs>* は、作成する SR-IOV VF の数です。

- SR-IOV サーバーが BIOS で有効になっていることを確認します。

> 注:
>
> IXGBE ドライバーのバージョン 3.22.3 をお勧めします。

Citrix Hypervisor ホストを使用して、**Intel 82599 SR-IOV VF** を **NetScaler VPX** インスタンスに割り当てます

Intel 82599 SR-IOV VF を NetScaler VPX インスタンスに割り当てるには、次の手順に従います。

1. Citrix Hypervisor ホストで、次のコマンドを使用して SR-IOV VF を NetScaler ADC VPX インスタンスに割り当てます。

   xe host-call-plugin plugin=iovirt host-uuid=*<Xen host UUID>* **fn=assign_free_vf args:uuid**=*<NetScaler VM UUID>* **args:ethdev**=*<interface name>* **args:mac=***<Mac addr>*

   各項目の意味は次のとおりです:

   - *<Xen host UUID>*Citrix Hypervisor の UUID です。
   - *<NetScaler VM UUID>* は、NetScaler VPX インスタンスの UUID です。
   - *<interface name>* は SR-IOV VF のインターフェイスです。
   - *<MAC address >* は SR-IOV VF の MAC アドレスです。

   > 注
   >
   > args: mac= パラメータで使用する MAC アドレスを指定します。指定しない場合、`iovirt`スクリプトはランダムに MAC アドレスを生成して割り当てます。また、リンクアグリゲーションモードで SR-IOV VF を使用する場合は、必ず MAC アドレスを 00:00:00:00:00 と指定します。

2. NetScaler VPX インスタンスを起動します。

Citrix Hypervisor ホストを使用して、**Intel 82599 SR-IOV VF** を **NetScaler VPX** インスタンスに割り当て解除します

正しくない SR-IOV VF を割り当てた場合、または割り当てられた SR-IOV VF を変更する場合は、SR-IOV VF を NetScaler ADC VPX インスタンスに割り当て解除して再割り当てする必要があります。

NetScaler VPX インスタンス に割り当てられた SR-IOV ネットワークインターフェイスの割り当てを解除するには、次の手順に従います。

1. Citrix Hypervisor ホストで、次のコマンドを使用して SR-IOV VF を NetScaler ADC VPX インスタンスに割り当て、NetScaler VPX インスタンスを再起動します。

   xe host-call-plugin plugin=iovirt **host-uuid**=<*Xen_host_UUID*> **fn**=unassign_all **args:uuid**=<*Netscaler_VM_U*

   各項目の意味は次のとおりです:

   - <*Xen_host_UUID*>-Citrix Hypervisor ホストの UUID。
   - <*Netscaler_VM_UUID*>-NetScaler VPX インスタンスの UUID

2. NetScaler VPX インスタンスを起動します。

Citrix Hypervisor ホストを使用して、**Intel X710/XL710 SR-IOV VF** を **NetScaler VPX** インスタンスに割り当てます

Intel X710/XL710 SR-IOV VF を NetScaler VPX インスタンスに割り当てるには、次の手順に従います。

1. Citrix Hypervisor ホストで次のコマンドを実行して、ネットワークを作成します。

```
xe network-create name-label=<network-name>
<!--NeedCopy-->
```

   例:

```
xe network-create name-label=SR-IOV-NIC-18 8ee59b73-7319-6998-cd69
   -b9fa3e8d7503
<!--NeedCopy-->
```

2. SR-IOV ネットワークを構成する NIC の PIF ユニバーサル一意識別子 (UUID) を決定します。

```
xe pif-list

             uuid ( RO)  : e2874343-f1de-1fa7-8fef-98547c348783
             device ( RO): eth18
currently-attached ( RO): true
               VLAN ( RO): -1
       network-uuid ( RO): f865bd85-44dd-b865-ab65-dcd6ae28c16e
<!--NeedCopy-->
```


3. ネットワークを SR-IOV ネットワークとして設定します。次のコマンドは、新しく作成された SR-IOV ネットワークの UUID も返します。

```
xe network-sriov-create network-uuid=<network-uuid> pif-uuid=<
    physical-pif-uuid>
<!--NeedCopy-->
```

例:

```
xe network-sriov-create network-uuid=8ee59b73-7319-6998-cd69-
    b9fa3e8d7503 pif-uuid=e2874343-f1de-1fa7-8fef-98547
    c3487831629b44f-832a-084e-d67d-5d6d314d5e0f
<!--NeedCopy-->
```

SR-IOV ネットワークパラメーターの詳細情報を取得するには、次のコマンドを実行します。

```
[root@citrix-XS82-TOPO ~]# xe network-sriov-param-list uuid=1629
    b44f-832a-084e-d67d-5d6d314d5e0f

               uuid ( RO): 1629b44f-832a-084e-d67d-5d6d314d5e0f
      physical-PIF ( RO): e2874343-f1de-1fa7-8fef-98547c348783
       logical-PIF ( RO): 85d52771-5814-c62d-45fa-f37b536144ff
   requires-reboot ( RO): false
remaining-capacity ( RO): 32
<!--NeedCopy-->
```

4. 仮想インターフェイス (VIF) を作成し、ターゲット VM にアタッチします。

```
xe vif-create device=0 mac=b2:61:fc:ae:00:1d network-uuid=8ee59b73
    -7319-6998-cd69-b9fa3e8d7503 vm-uuid=b507e8a6-f5ca-18eb-561d
    -308218a9dd68
3e1e2e58-b2ad-6dc0-61d4-1d149c9c6466
<!--NeedCopy-->
```

> 注:VM の NIC インデックス番号は 0 で始まる必要があります。

VM UUID を見つけるには、次のコマンドを使用します。

```
[root@citrix-XS82-TOPO ~]# xe vm-list
uuid ( RO): b507e8a6-f5ca-18eb-561d-308218a9dd68
 name-label ( RW): sai-vpx-1
power-state ( RO): halted
<!--NeedCopy-->
```

Citrix Hypervisor ホストを使用して **NetScaler** インスタンスから **Intel X710/XL710 SR-IOV VF** を削除します

NetScaler VPX インスタンスから Intel X710/XL710 SR-IOV VF を削除するには、次の手順に従います。

1. 破棄する VIF の UUID をコピーします。

2. VIF を破棄するには、Citrix Hypervisor ホスト上で以下のコマンドを実行します。

```
xe vif-destroy uuid=<vif-uuid>
<!--NeedCopy-->
```

例:

```
[root@citrix-XS82-TOPO ~]# xe vif-destroy uuid=3e1e2e58-b2ad-6dc0
    -61d4-1d149c9c6466
<!--NeedCopy-->
```

SR-IOV インターフェイスでのリンクアグリゲーションの設定

SR-IOV 仮想機能 (VF) をリンクアグリゲーションモードで使用するには、作成した仮想機能のスプーフィングチェックを無効にする必要があります。

Citrix Hypervisor ホストで、次のコマンドを使用してスプーフィングチェックを無効にします。

ip リンクセット ** <interface_name>**vf** ** <VF_id>**spoofchk** オフ

各項目の意味は次のとおりです:

- <interface_name> は、インターフェイス名です。
- <VF_id> は、仮想機能 ID です。

作成したすべての仮想機能のスプーフィングチェックを無効にした後、NetScaler VPX インスタンスを再起動し、リンクアグリゲーションを設定します。手順については、「リンク集約の設定」を参照してください。

> 重要
>
> SR-IOV VF を NetScaler ADC VPX インスタンスに割り当てるときは、VF の MAC アドレス 00:00:00:00:00:00 を必ず指定してください。

SR-IOV インターフェイスで VLAN を設定します

SR-IOV 仮想機能に VLAN を設定できます。手順については、「VLAN の設定」を参照してください。

> 重要
>
> Citrix Hypervisor ホストに VF インターフェイスの VLAN 設定が含まれていないことを確認してください。

VMware ESX に NetScaler ADC VPX インスタンスをインストールする

August 15, 2023

NetScaler VPX インスタンスを VMware ESX にインストールする前に、VMware ESX サーバーが適切なシステムリソースを備えたマシンにインストールされていることを確認してください。NetScaler VPX インスタンスを VMware ESXi にインストールするには、VMware vSphere クライアントを使用します。これらのクライアントソフトウェアは、ネットワーク経由で VMware ESX に接続できるリモートマシンにインストールする必要があります。

このセクションでは、以下のトピックについて説明します。

- 前提条件
- VMware ESX に NetScaler ADC VPX インスタンスをインストールする

> 重要:
>
> NetScaler VPX インスタンスで標準の VMware Tools をインストールしたり、VMware Tools バージョンをアップグレードしたりすることはできません。NetScaler VPX インスタンス用の VMware ツールは、NetScaler ADC ソフトウェアリリースの一部として提供されます。

前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- 最小要件を満たすハードウェアに VMware ESX をインストールします。
- 最小システム要件を満たす管理用のワークステーションに VMware Client をインストールします。
- NetScaler VPX アプライアンスセットアップファイルをダウンロードします。
- 仮想スイッチを作成し、物理 NIC を仮想スイッチに接続します。
- ポートグループを追加し、仮想スイッチに接続します。
- ポートグループを VM に接続します。
- VPX ライセンスファイルを入手します。NetScaler VPX インスタンスライセンスの詳細については、「ライセンスの概要」を参照してください。

VMware ESX のハードウェア要件

次の表では、NetScaler VPX nCore 仮想アプライアンスを実行している VMware ESX サーバーの最小システム要件について説明します。

表 1. NetScaler VPX インスタンスを実行する VMware ESX サーバーの最小システム要件

Component	条件
CPU	仮想化アシスト (Intel-VT) が有効になっている 64 ビット x86 CPU が 2 つ以上あります。NetScaler VPX インスタンスを実行するには、仮想化のハードウェアサポートが VMware ESX ホストで有効になっている必要があります。仮想化サポートの BIOS オプションが無効になっていないことを確認します。詳しくは、BIOS のドキュメントを参照してください。NetScaler 13.1 リリース以降、VMware ESXi ハイパーバイザー上の NetScaler VPX インスタンスは AMD プロセッサをサポートしています。
RAM	2 ギガバイトの VPX 重要な展開では、システムがメモリに制約のある環境で動作するため、VPX に 2 GB の RAM を使用することはお勧めしません。これにより、スケール、パフォーマンス、または安定性に関連する問題が発生する可能性があります。推奨されるのは 4 GB の RAM または 8 GB の RAM です。
ディスク領域	ESXi をセットアップするための VMware の最小サーバ要件よりも 20 GB 多くなっています。サーバの最小要件については、VMware のマニュアルを参照してください。
ネットワーク	1 Gbps NIC (NIC) 1 つ、1 Gbps NIC を 2 つ推奨

VMware ESX のインストールについては、http://www.vmware.com/を参照してください。

SR-IOV ネットワークインターフェイスまたは PCI パススルーをサポートするには、次のプロセッサと設定が有効になっていることを確認してください。

- Intel VT をサポートする Intel ・プロセッサー
- AMD-V をサポートする AMD プロセッサー
- 入出力メモリ管理ユニット (IOMMU) または SR-IOV が BIOS で有効になっています

SR-IOV モードでは次の NIC がサポートされます。

- Mellanox ConnectX-4 NIC（Citrix ADC リリース 13.1-42.x 以降）
- Intel 82599 NIC

次の表に、VMware ESX サーバが各 VPX nCore 仮想アプライアンスに対して提供する必要がある仮想コンピューティングリソースを示します。

表 2. NetScaler VPX インスタンスの実行に必要な最小限の仮想コンピューティングリソース

Component	条件
メモリ	4 GB
仮想 CPU（VCPU）	2
仮想ネットワークインターフェイス	ESX では、VPX ハードウェアをバージョン 7 以上にアップグレードすると、最大 10 個の仮想ネットワークインターフェイスをインストールできます。
ディスク領域	20 GB

> 注:
>
> ハイパーバイザーに必要なディスク領域は含まれません。

VPX 仮想アプライアンスを本番環境で使用するには、完全なメモリ割り当てを予約する必要があります。少なくとも ESX の 1 つの CPU コアの速度に等しい CPU サイクル (MHz) を予約する必要があります。

VMware vSphere クライアントのシステム要件

VMware vSphere Client は、Windows および Linux の各オペレーティングシステムで実行できるクライアントアプリケーションです。VMware ESX サーバと同じマシンでは実行できません。次の表は、最小システム要件を示しています。

表 3. VMware vSphere クライアントインストールの最小システム要件

Component	条件
オペレーティングシステム	VMware からの詳細な要件については、http://kb.vmware.com/で「vSphere 互換性マトリックス」PDF ファイルを検索してください。
CPU	750 MHz。1 ギガヘルツ (GHz) 以上推奨
RAM	1 GB。2 GB 推奨
NIC (NIC)	100Mbps 以上の NIC。

OVF ツール **1.0** のシステム要件

Open Virtualization Format Tool（OVF Tool）は、Windows および Linux システムで実行できるクライアントアプリケーションです。VMware ESX サーバと同じマシンでは実行できません。次の表は、最小システム要件を示しています。

表 4. OVF ツールのインストールに必要な最小システム要件

Component	条件
オペレーティングシステム	VMware からの詳細な要件については、http://kb.vmware.com/で『OVF ツールユーザーガイド』の PDF ファイルを検索してください。
CPU	最低 750MHz、1GHz 以上推奨
RAM	最小 1 GB、推奨 2 GB
NIC (NIC)	100Mbps 以上の NIC。

OVF のインストールについては、http://kb.vmware.com/で『OVF ツールユーザーガイド』の PDF ファイルを検索してください。

NetScaler VPX セットアップファイルのダウンロード

VMware ESX 用の NetScaler VPX インスタンスセットアップパッケージは、オープン仮想マシン（OVF）フォーマット標準に準拠しています。これらのファイルは、Citrix Web サイトからダウンロードできます。ログオンするには、Citrix アカウントが必要です。Citrix アカウントをお持ちでない場合は、http://www.citrix.comのホームページにアクセスし、［新規ユーザー］リンクをクリックし、指示に従って Citrix アカウントを作成します。

ログオンしたら、Citrix のホームページから次のパスをナビゲートします。

Citrix.com > ダウンロード **> NetScaler >** 仮想アプライアンス。

次のファイルを、ESX サーバーと同じネットワーク上のワークステーションにコピーします。3 つのファイルをすべて同じフォルダーにコピーします。

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk（たとえば、nsvpx-esx-13.0-71.44_nc_64-disk1.vmdk）
- NSVPX-ESX-<release number>-<build number>.ovf（たとえば、nsvpx-ESX-13.0-71.44_nc_64.ovf）
- NSVPX-ESX-<release number>-<build number>.mf（たとえば、nsvpx-esx-13.0-71.44_nc_64.mf）

VMware ESX に **NetScaler ADC VPX** インスタンスをインストールする

VMware ESX をインストールして構成したら、VMware vSphere Client を使用して VMware ESX サーバーに仮想アプライアンスをインストールします。インストールできる仮想アプライアンスの数は、VMware ESX を実行するハードウェアで使用可能なメモリの量によって決まります。

VMware vSphere クライアントを使用して NetScaler VPX インスタンスを VMware ESX にインストールするには、以下の手順に従ってください。

1. ワークステーション上で VMware vSphere Client を起動します。
2. [**IP address / Name**] テキストボックスに、接続する VMware ESX サーバーの IP アドレスを入力します。
3. [ユーザー名] テキストボックスと [パスワード] テキストボックスに、管理者の認証情報を入力し、[ログイン] をクリックします。
4. [**File**] メニューの [**Deploy OVF Template**] を選択します。
5. [**OVF** テンプレートのデプロイ] ダイアログボックスの [ファイルからデプロイ] で、NetScaler VPX インスタンスセットアップファイルを保存した場所を参照し、.ovf ファイルを選択して、[次へ] をクリックします。
6. 仮想アプライアンス OVF テンプレートに示されるネットワークを、ESX ホストで構成したネットワークにマップします。[**Next**] をクリックして、VMware ESX への仮想アプライアンスのインストールを開始します。インストールが完了すると、ポップアップウィンドウによって正常にインストールされたことが通知されます。
7. これで、NetScaler VPX インスタンスを起動する準備ができました。ナビゲーションペインで、インストールした NetScaler ADC VPX インスタンスを選択し、右クリックメニューから [パワーオン] を選択します。
8. 仮想マシンを起動したら、コンソールから NetScaler ADC IP、ネットマスク、およびゲートウェイアドレスを設定します。設定が完了したら、コンソールで [**Save and Quit**] オプションを選択します。
9. 別の仮想アプライアンスをインストールするには、ステップ 6 からステップ 8 までを繰り返します。

> 注:
>
> デフォルトでは、NetScaler VPX インスタンスは E1000 ネットワークインターフェイスを使用します。
>
> インストール後、vSphere クライアントまたは vSphere Web Client を使用して VMware ESX 上の仮想アプライアンスを管理できます。
>
> VLAN タギング機能を動作させるには、VMware ESX サーバの vSwitch で、ポートグループの VLAN ID を [すべて] (4095) に設定します。VMware ESX サーバの vSwitch での VLAN ID の設定の詳細については、http://www.vmware.com/pdf/esx3_vlan_wp.pdfを参照してください。

VMware vMotion を使用して **NetScaler ADC VPX** インスタンスを移行する

VMware vSphere vMotion を使用して、NetScaler VPX インスタンスを移行できます。

使用上のガイドラインに従ってください。

- VMware は、PCI パススルーおよび SR-IOV インターフェイスで構成された仮想マシンでは vMotion 機能をサポートしていません。
- サポートされているインターフェイスは、E1000 と VMXNET3 です。VPX インスタンスで vMotion を使用するには、サポートされているインターフェイスでインスタンスが設定されていることを確認します。
- VMware vMotion を使用してインスタンスを移行する方法の詳細については、VMware のドキュメントを参照してください。

VMXNET3 ネットワークインターフェイスを使用するように NetScaler VPX インスタンスを構成する

August 15, 2023

VMware ESX に NetScaler VPX インスタンスをインストールして構成したら、VMware vSphere Web クライアントを使用して、VMXNET3 ネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

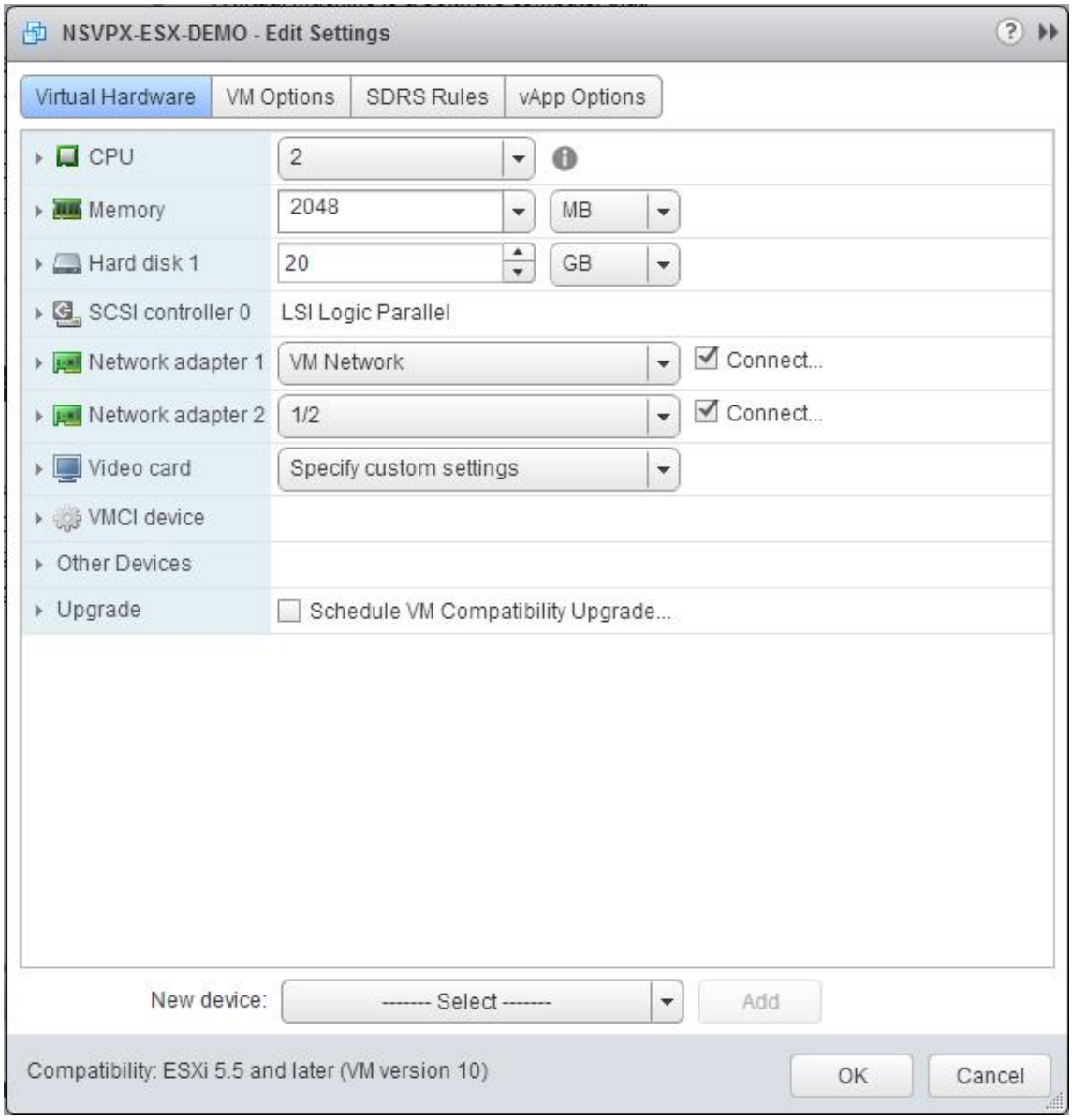
VMware vSphere Web クライアントを使用して VMXNET3 ネットワークインターフェイスを使用するように NetScaler VPX インスタンスを構成するには:

1. vSphere Web クライアントで、ホストとクラスタを選択します。
2. 次のように、NetScaler VPX インスタンスの互換性設定を ESX にアップグレードします。

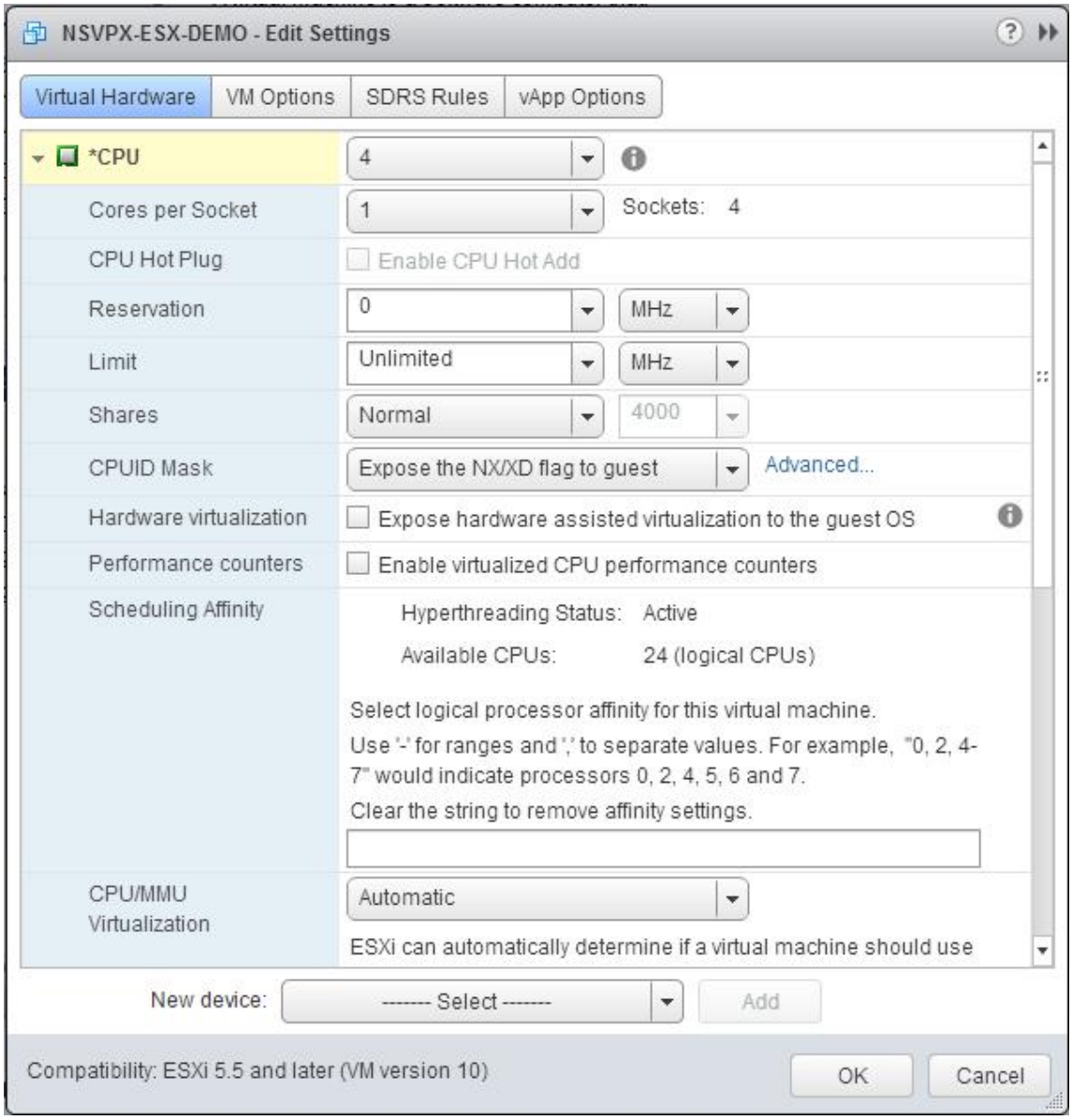
   a. NetScaler VPX インスタンスの電源を切ります。

   b. NetScaler VPX インスタンスを右クリックし、「互換性」>「仮想マシンの互換性のアップグレード」を選択します。

   c. 「仮想マシンの互換性の設定」ダイアログボックスで、「互換性」ドロップダウンリストから「ESXi 5.5 以降」を選択し、「OK」をクリックします。
3. NetScaler VPX インスタンスを右クリックし、[設定の編集] をクリックします。



4. [<virtual_appliance> 設定の編集] ダイアログボックスで、[CPU] セクションをクリックします。



5. [CPU] セクションで、以下を更新します。

   - CPU の数
   - ソケット数
   - 予約
   - 上限
   - 共有

   値を次のように設定します:

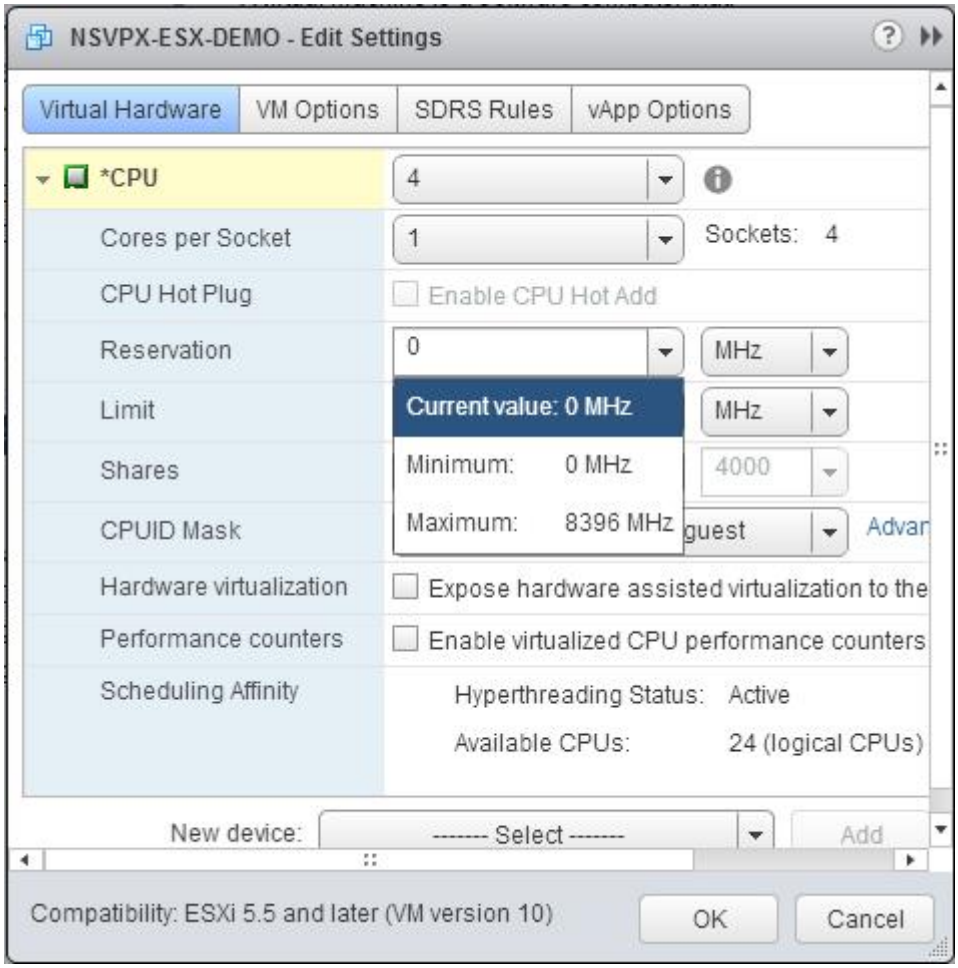
   a. CPU ドロップダウンリストで、仮想アプライアンスに割り当てる CPU の数を選択します。

   b.「ソケットあたりのコア数」ドロップダウンリストで、ソケット数を選択します。

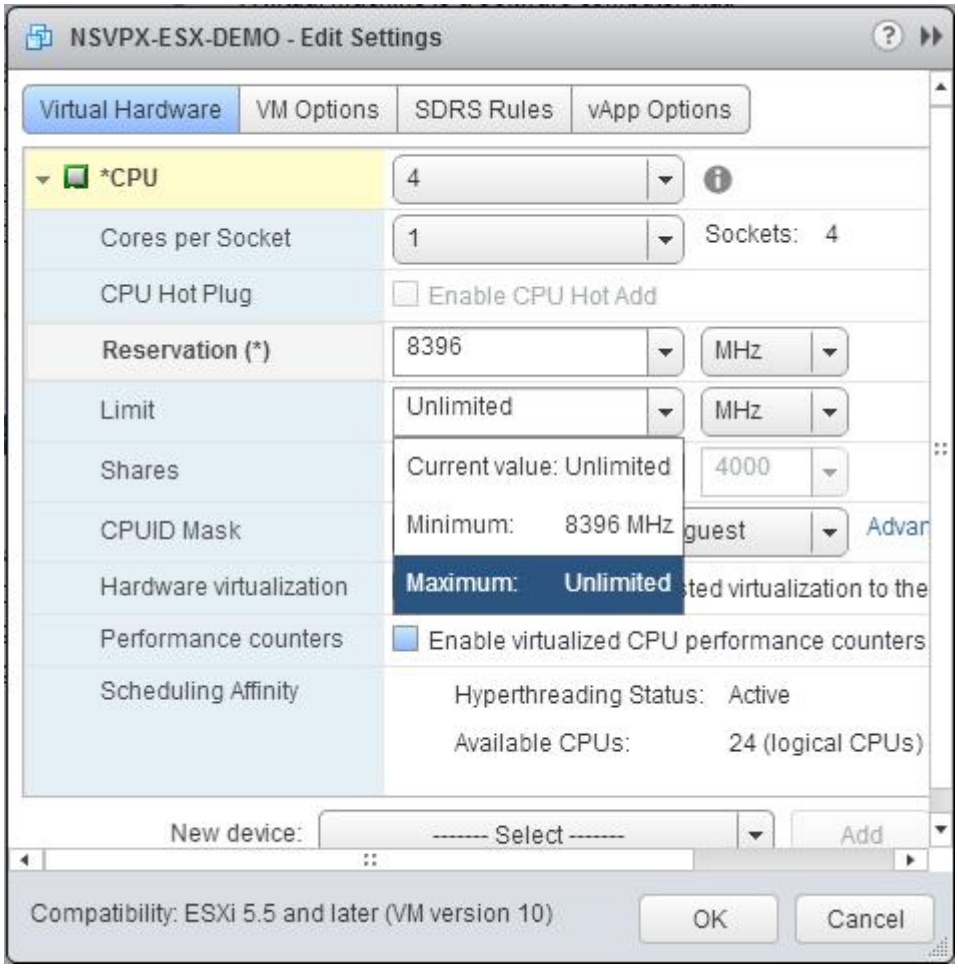
c. （オプション）[CPU ホットプラグ] フィールドで、[CPU ホットアドを有効にする] チェックボックスをオンまたはオフにします。

注：Citrix では、デフォルト（無効）をそのまま使用することをお勧めします。

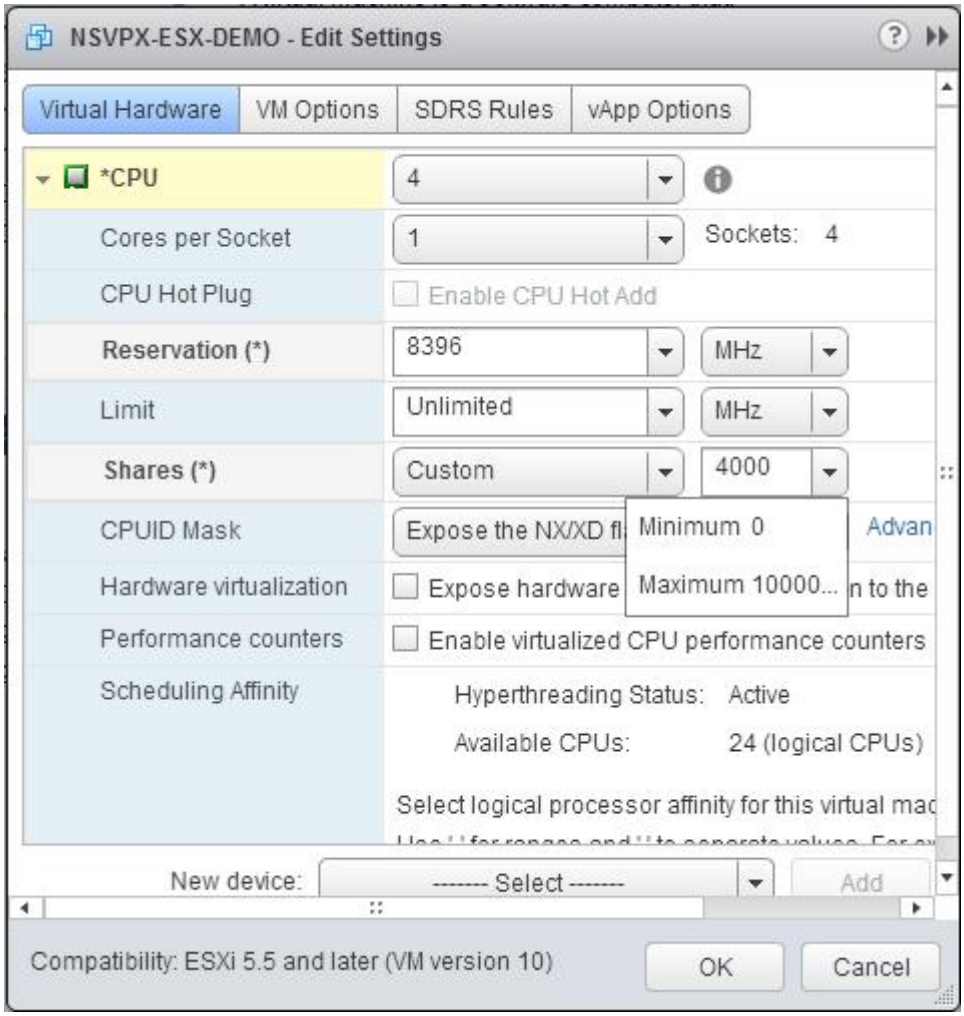
d. [予約] ドロップダウンリストで、最大値として表示される番号を選択します。



e. [制限] ドロップダウンリストで、最大値として表示されている数値を選択します。



f. [共有] ドロップダウンリストで、[カスタム] を選択し、最大値として表示される数を指定します。



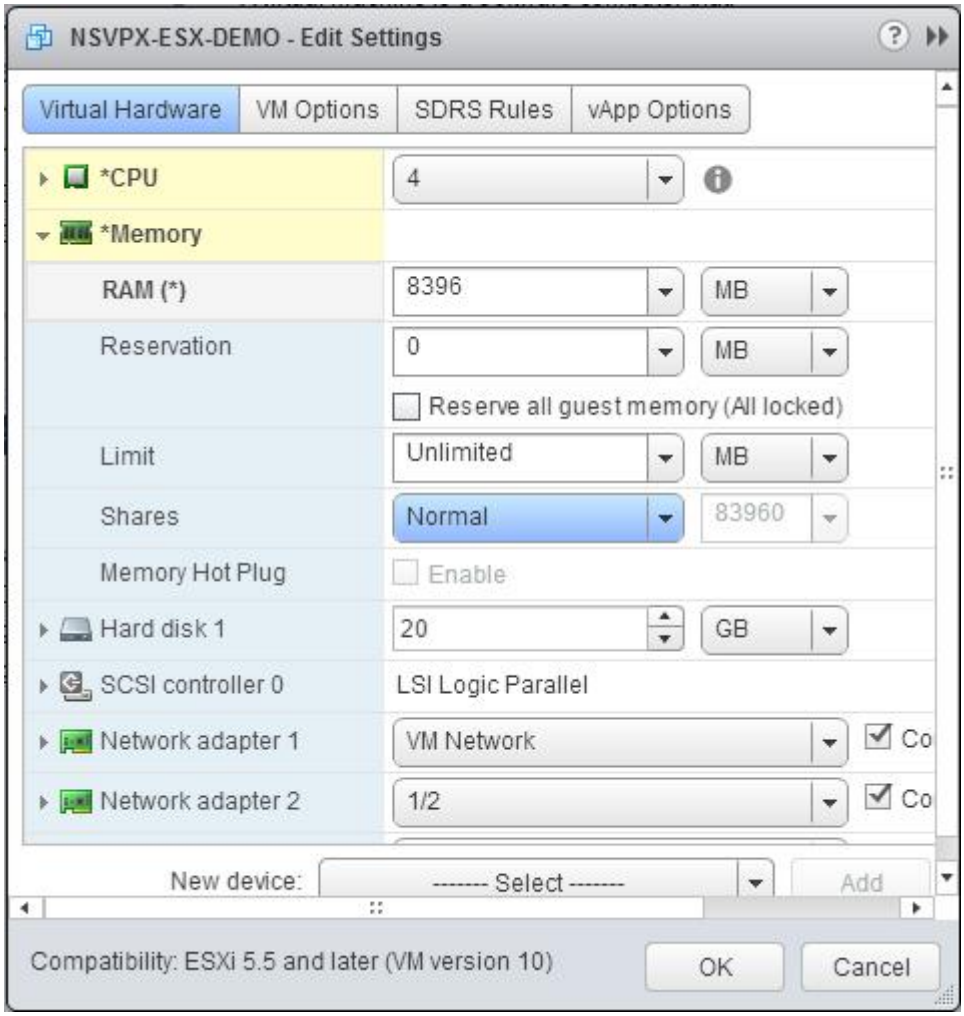
6. [メモリ] セクションで、次の項目を更新します。

   - メモリのサイズ
   - 予約
   - 上限
   - 共有

   値を次のように設定します:

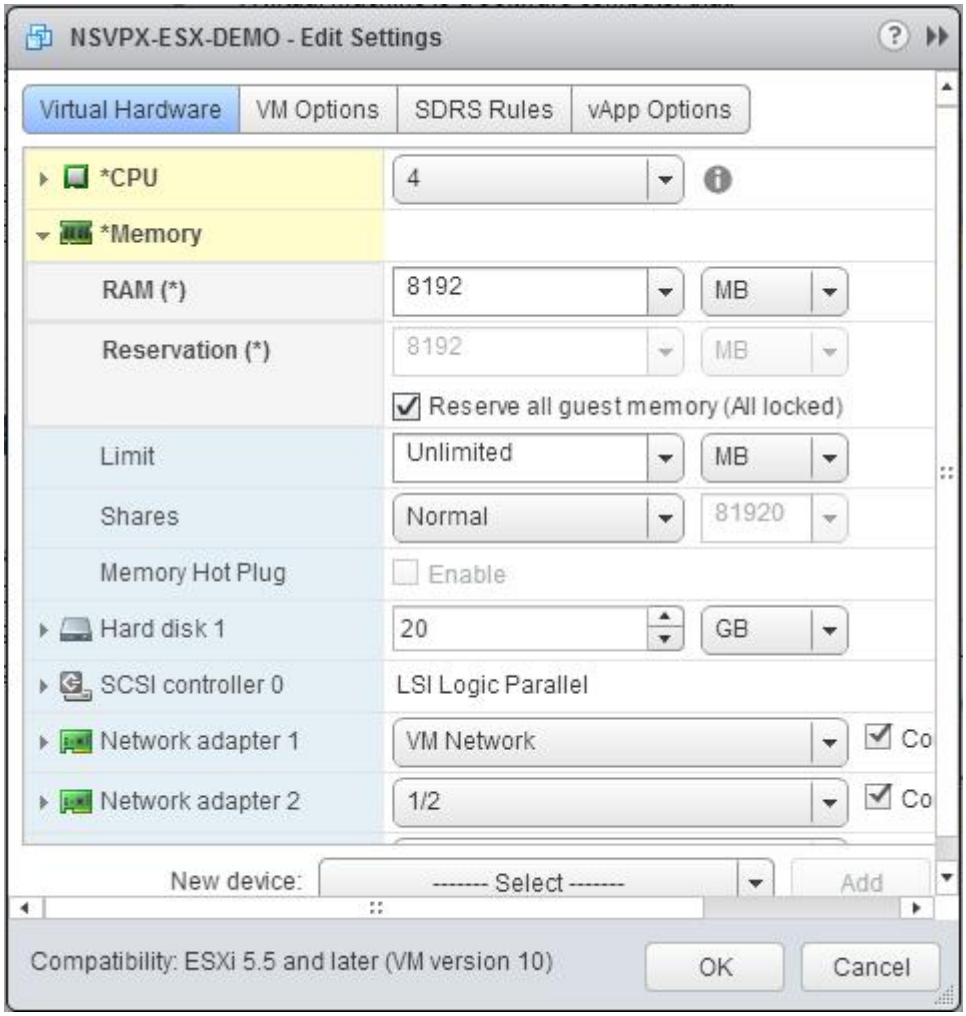
   a. [RAM] ドロップダウンリストで、RAM のサイズを選択します。これは vCPU の数 x 2 GB でなければなりません。たとえば、vCPU の数が 4 の場合、RAM は 4 x 2 GB = 8 GB でなければなりません。

   注: NetScaler VPX アプライアンスの Advanced エディションまたは Premium エディションでは、各 vCPU に 4GB の RAM を割り当ててください。たとえば、vCPU の数が 4 の場合、RAM = 4×4GB = 16GB になります。

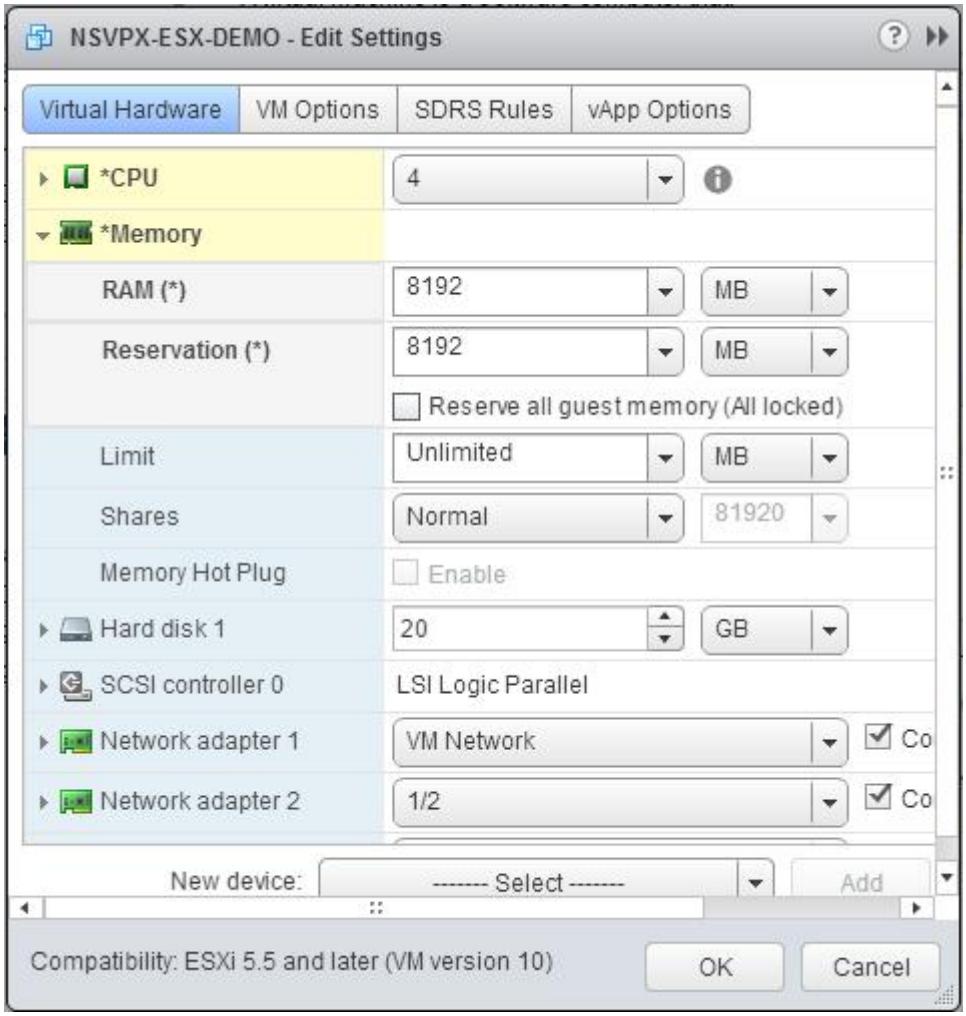


b. [予約] ドロップダウンリストで、メモリ予約の値を入力し、[すべてのゲストメモリを予約する (すべてのロック済み)] チェックボックスをオンにします。メモリ予約は vCPU の数 x 2 GB である必要があります。たとえば、vCPU の数が 4 の場合、メモリ予約は 4 x 2 GB = 8 GB である必要があります。

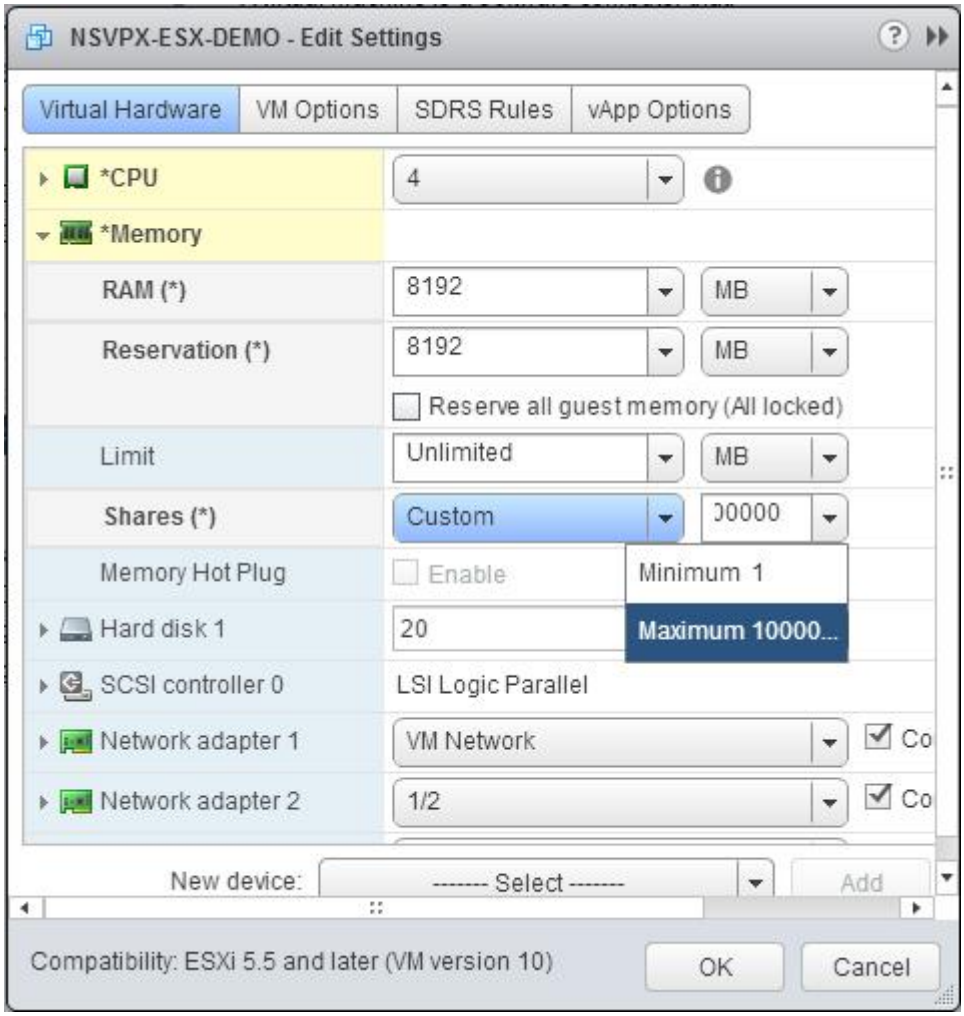
注: NetScaler VPX アプライアンスの Advanced エディションまたは Premium エディションでは、各 vCPU に 4GB の RAM を割り当ててください。たとえば、vCPU の数が 4 の場合、RAM = 4×4GB = 16GB になります。



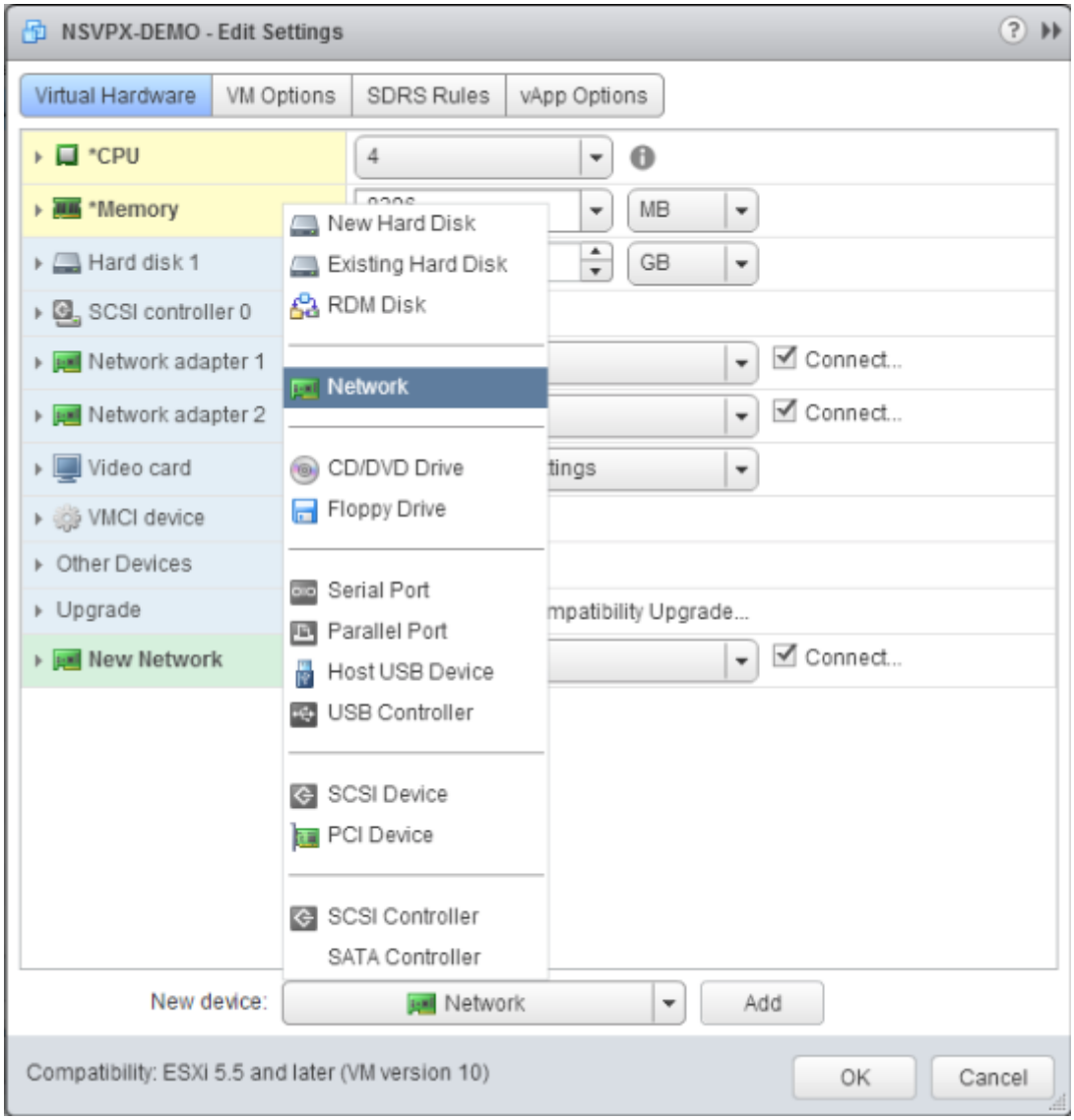
c. [制限] ドロップダウンリストで、最大値として表示されている数値を選択します。



d.「共有」ドロップダウンリストで、「カスタム」を選択し、最大値として表示される数値を選択します。



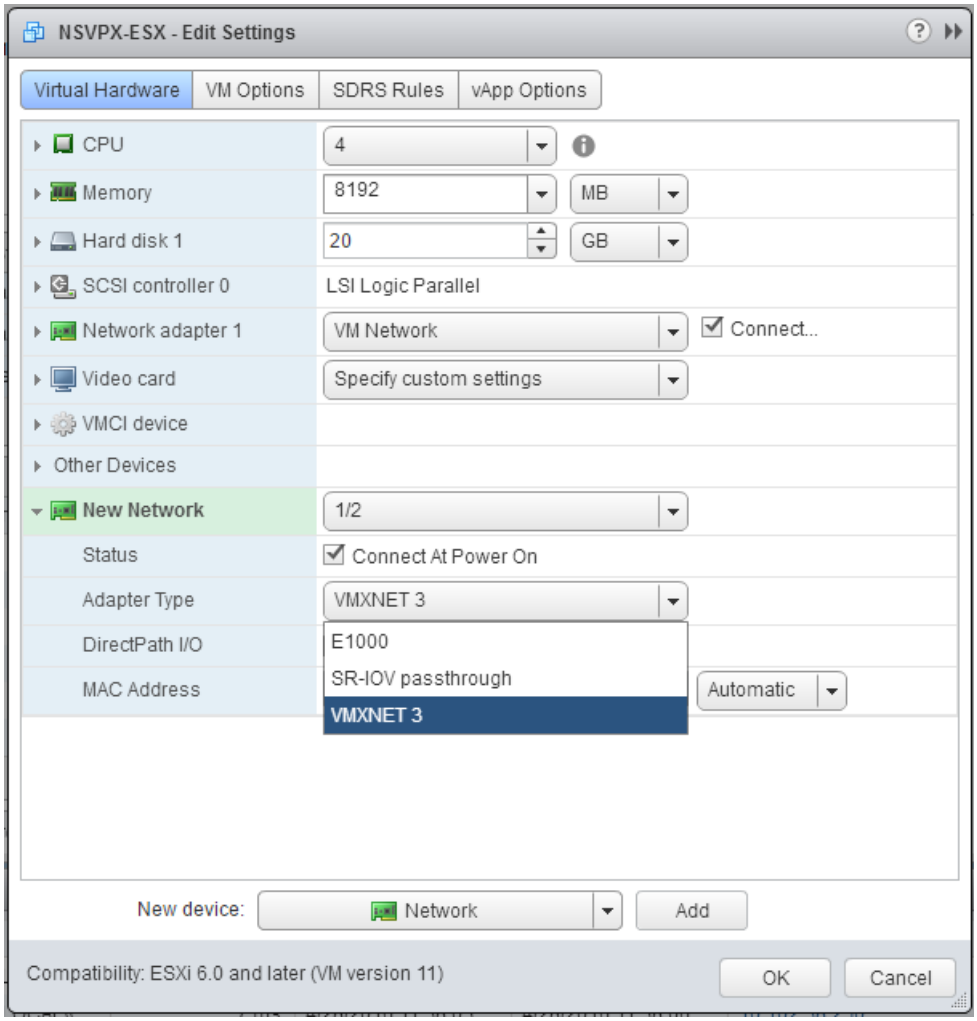
7. VMXNET3 ネットワークインターフェイスを追加します。[新しいデバイス] ドロップダウンリストから [ネットワーク] を選択し、[追加] をクリックします。



8. [New Network] セクションのドロップダウンリストからネットワークインターフェイスを選択し、次の操作を行います。

   a. ［Adapter Type］ボックスの一覧で、［VMXNET3］を選択します。

   > 重要
   >
   > デフォルトの E1000 ネットワークインターフェイスと VMXNET3 は共存できないため、E1000 ネットワークインターフェイスの削除を確認して、VMXNET3（0/1）を管理インターフェイスとして使用します。



9. 「OK」をクリックします。
10. NetScaler VPX インスタンスをパワーオンします。
11. NetScaler VPX インスタンスの電源が入ったら、次のコマンドを使用して構成を確認できます。

    > インターフェイスの概要を表示

出力には、設定したすべてのインターフェイスが表示されている必要があります。

```
> show interface summary
-------------------------------------------------------------------------

      Interface  MTU        MAC                Suffix
-------------------------------------------------------------------------

1     0/1        1500       00:0c:29:89:1d:0e  NetScaler Vir...rface,
    VMXNET3
2     1/1        9000       00:0c:29:89:1d:18  NetScaler Vir...rface,
    VMXNET3
3     1/2        9000       00:0c:29:89:1d:22  NetScaler Vir...rface,
    VMXNET3
```



```
8  4     LO/1        9000        00:0c:29:89:1d:0e    Netscaler Loopback
       interface
```

> 注
>
> VMXNET3 インターフェイスを追加して NetScaler ADC VPX アプライアンスを再起動すると、VMware ESX ハイパーバイザーによって NIC が VPX アプライアンスに提示される順序が変更されることがあります。そのため、ネットワークアダプター 1 が常に 0/1 のままであるとは限らず、その結果 VPX アプライアンスに対する管理接続が失われることがあります。この問題を回避するには、ネットワークアダプターの仮想ネットワークを変更します。
>
> これは VMware ESX ハイパーバイザの制限です。

SR-IOV ネットワークインターフェイスの使用を NetScaler ADC VPX インスタンスで構成する

August 15, 2023

VMware ESX に NetScaler ADC VPX インスタンスをインストールして構成した後、VMware vSphere Web クライアントを使用して、シングルルート I/O 仮想化（SR-IOV）ネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

制限事項

SR-IOV ネットワークインターフェイスで構成された NetScaler VPX には、以下の制限事項があります。

- 次の機能は、ESX VPX 上の Intel 82599 10G NIC を使用する SR-IOV インターフェイスではサポートされていません。
  - L2 モード切り替え
  - スタティックリンク集約および LACP
  - クラスタリング
  - 管理パーティション化［共有 VLAN モード］
  - 高可用性［アクティブ/アクティブモード］
  - ジャンボフレーム
  - IPv6
- KVM VPX 上の Intel 82599 10G NIC を搭載した SR-IOV インターフェイスでは、次の機能はサポートされていません。
  - スタティックリンク集約および LACP

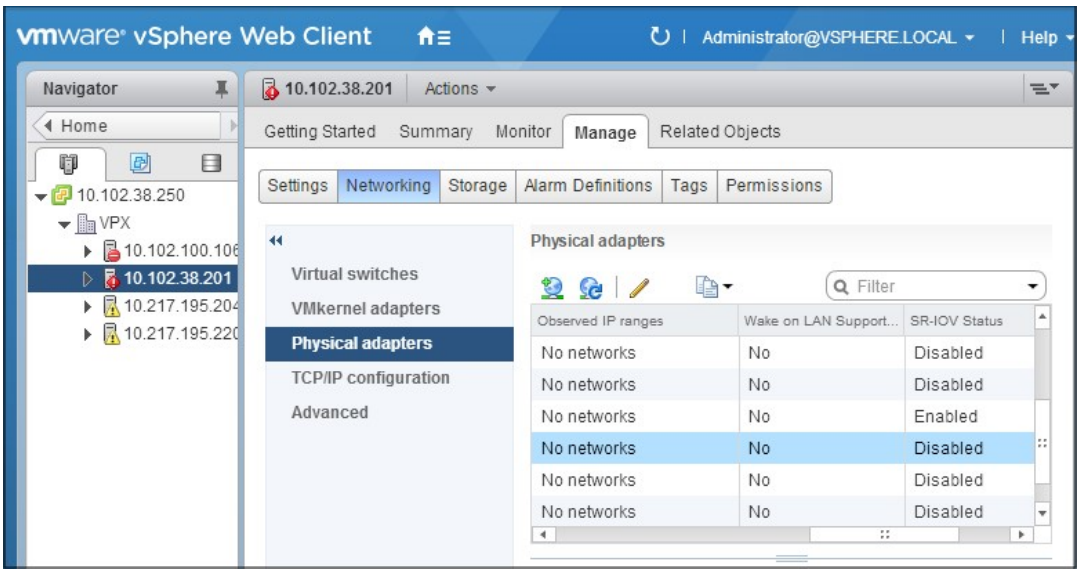
- L2 モード切り替え
- クラスタリング
- 管理パーティション化［共有 VLAN モード］
- 高可用性 [アクティブ–アクティブモード]
- ジャンボフレーム
- IPv6
- `ip link`コマンドによる SR-IOV VF インターフェイスのハイパーバイザでの VLAN 設定はサポートされていません

前提要件

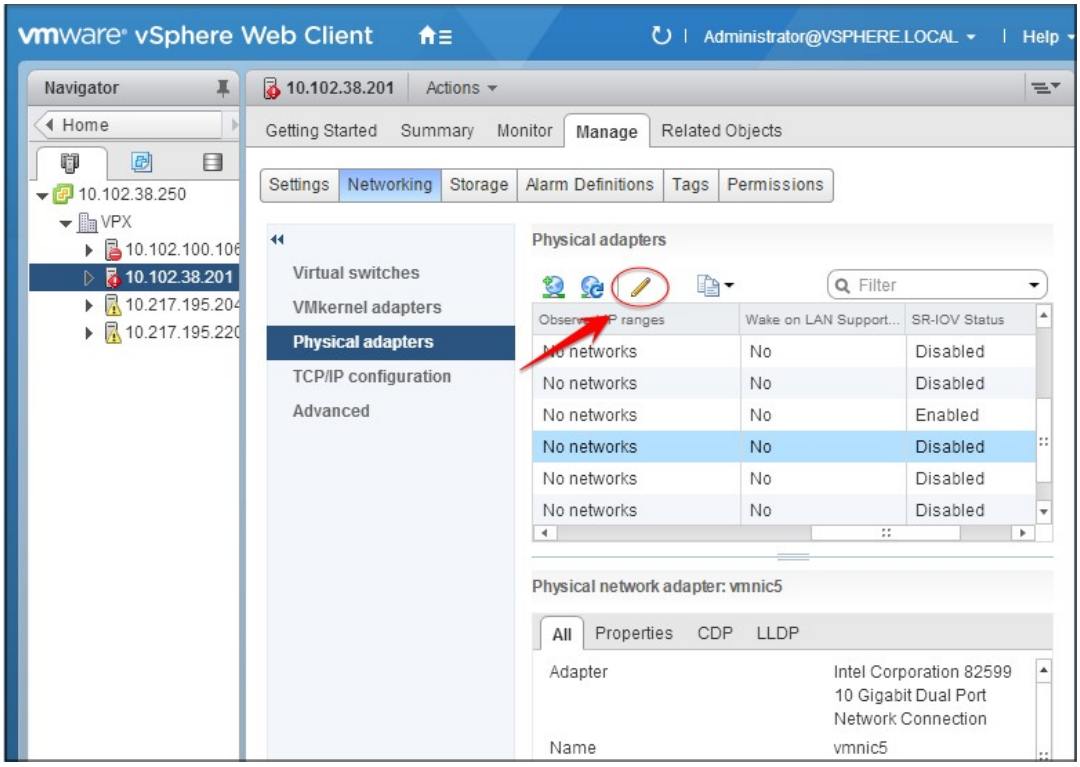
- 必ず ESX ホストに次のいずれかの NIC を追加してください。
  - Intel 82599 NIC、IXGBE ドライバーバージョン 3.7.13.7.14iov 以降が推奨されます。
  - Mellanox ConnectX-4 NIC
- ホスト物理アダプタで SR-IOV を有効にします。

  以下の手順に従って、ホスト物理アダプタで SR-IOV を有効にします。

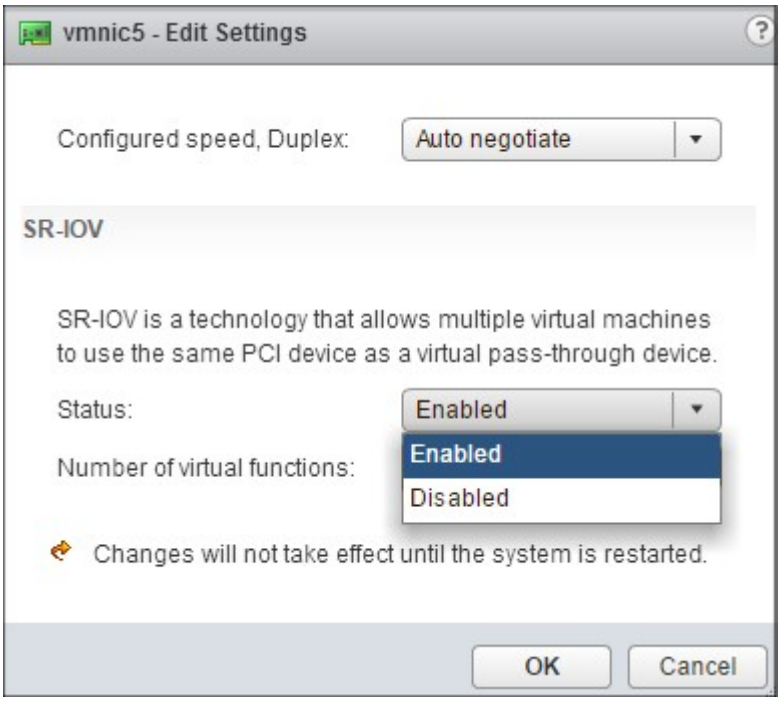
  1. vSphere Web クライアントで、ホストに移動します。
  2. [管理**] > [**ネットワーク] タブで、[物理アダプター] を選択します。[SR-IOV Status］フィールドに、物理アダプターが SR-IOV をサポートしているかどうかが表示されます。

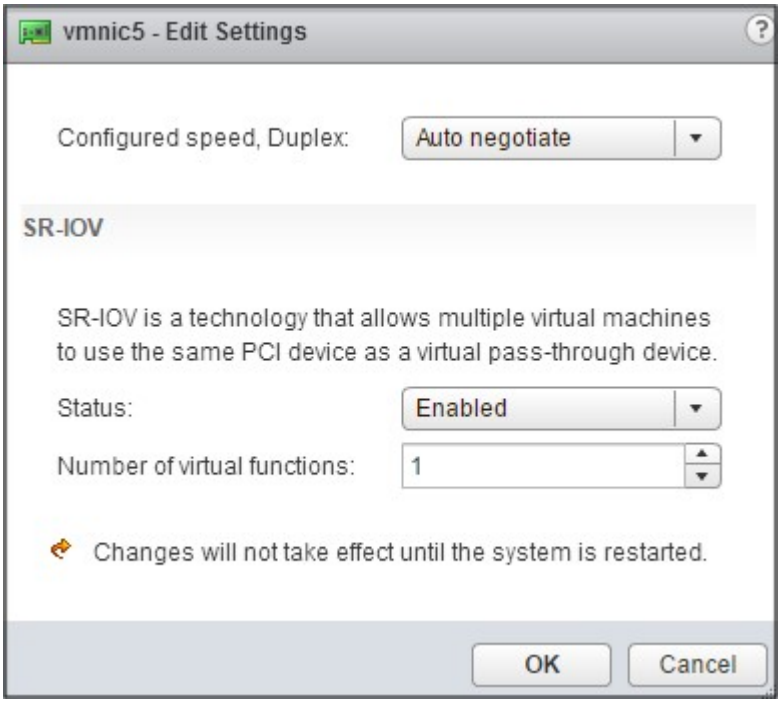
  3. 物理アダプタを選択し、鉛筆アイコンをクリックして [設定の編集] ダイアログボックスを開きます。



4.「SR-IOV」で、「ステータス」ドロップダウンリストから「有効」を選択します。



5. [仮想関数の数] フィールドに、アダプタに対して構成する仮想関数の数を入力します。



6. [**OK**] をクリックします。
7. ホストを再起動します。

- 分散仮想スイッチ (DVS) と`Portgroups`を作成します。手順については、VMware のドキュメントを参照してください。

> 注
>
> Citrix は、DVS および`Portgroups`でのみ SR-IOV 構成を認定しています。

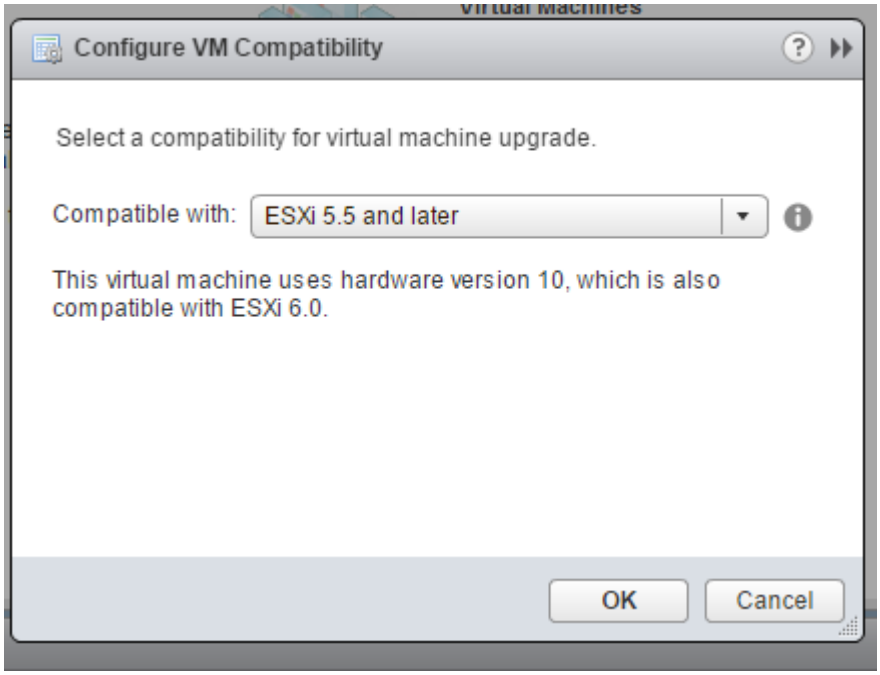
VMware vSphere Web クライアントを使用して **SR-IOV** ネットワークインターフェイスを使用するように **NetScaler ADC VPX** インスタンスを構成するには:

1. vSphere Web クライアントで、ホストとクラスタを選択します。
2. 次のように、NetScaler VPX インスタンスの互換性設定を ESX 5.5 以降にアップグレードします。

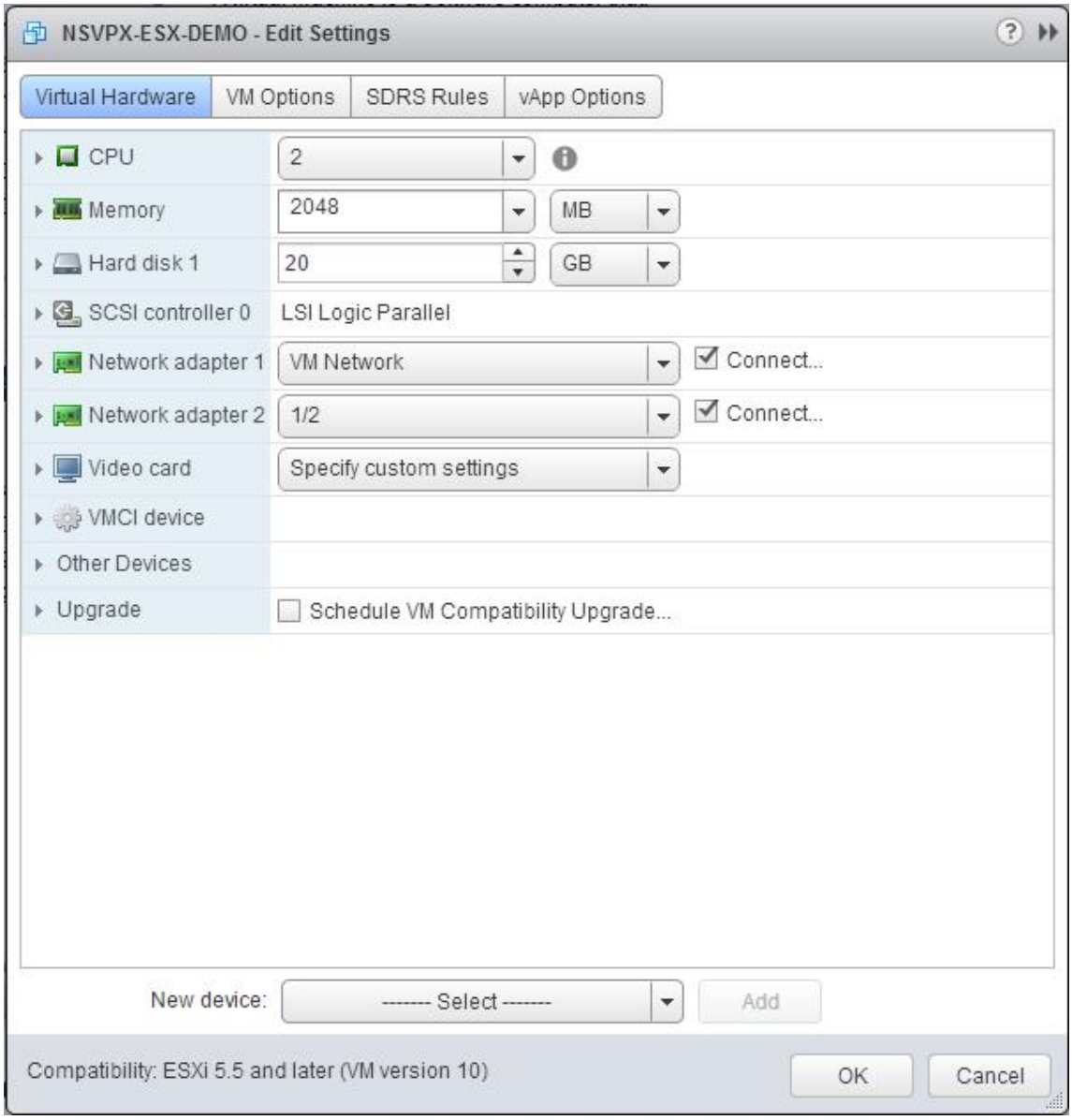
   a. NetScaler VPX インスタンスの電源を切ります。

   b. NetScaler VPX インスタンスを右クリックし、「互換性」**>**「仮想マシンの互換性のアップグレード」を選択します。

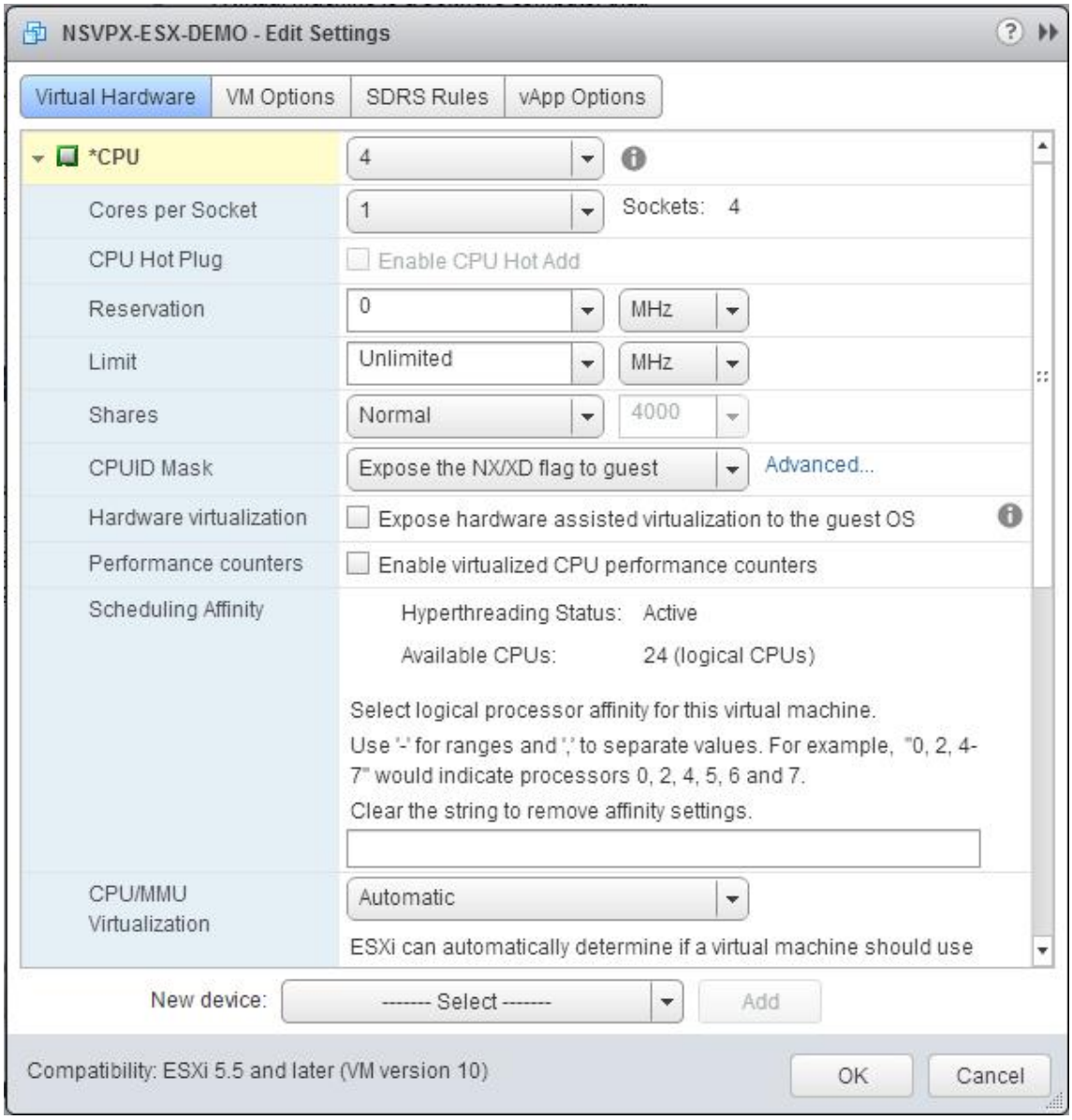
   c. ****「**** 仮想マシンの互換性の設定」ダイアログボックスで、「互換性」ドロップダウンリストから「**ESXi 5.5** 以降」を選択し、「OK」をクリックします。**



3. NetScaler VPX インスタンスを右クリックし、**[設定の編集]** をクリックします。



4. [**<virtual_appliance> 設定の編集] ダイアログボックスで、[CPU**] セクションをクリックします。



5. [**CPU**] セクションで、次の設定を更新します。

   - CPU の数
   - ソケット数
   - 予約
   - 上限
   - 共有

   値を次のように設定します:

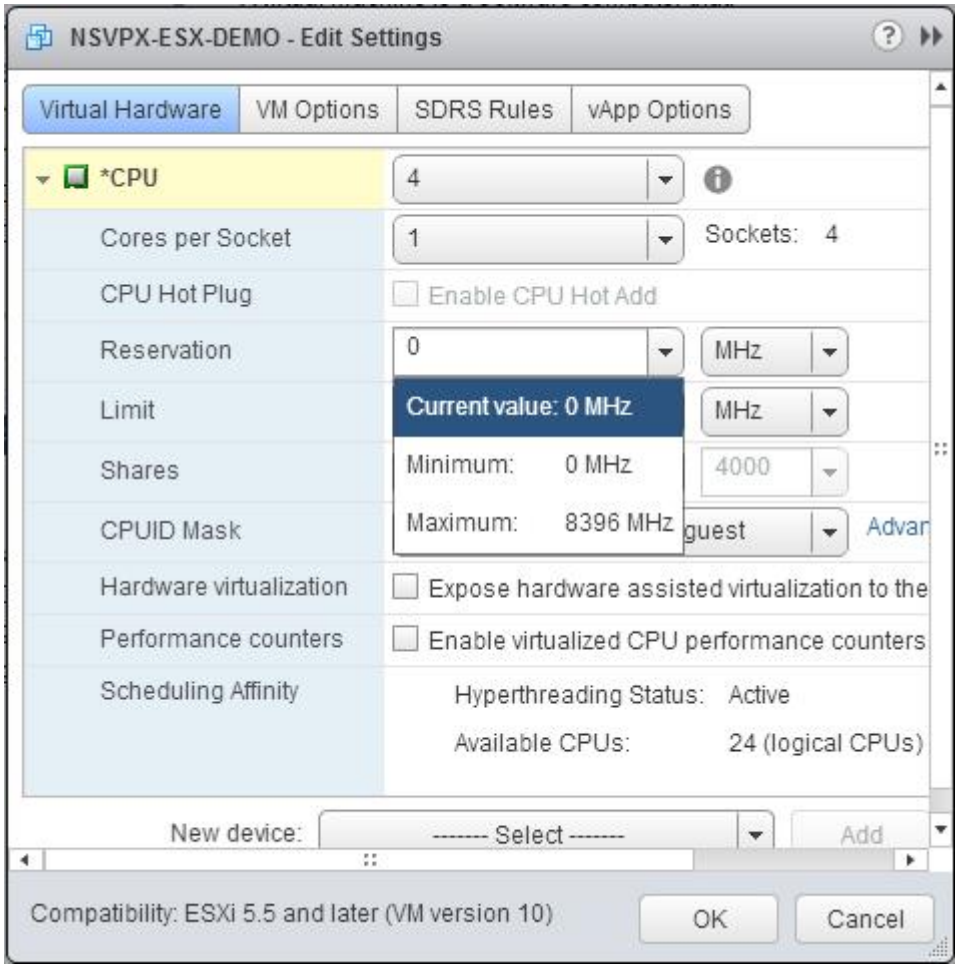
   a. **CPU** ドロップダウンリストで、仮想アプライアンスに割り当てる CPU の数を選択します。

   b.「ソケットあたりのコア数」ドロップダウンリストで、ソケット数を選択します。

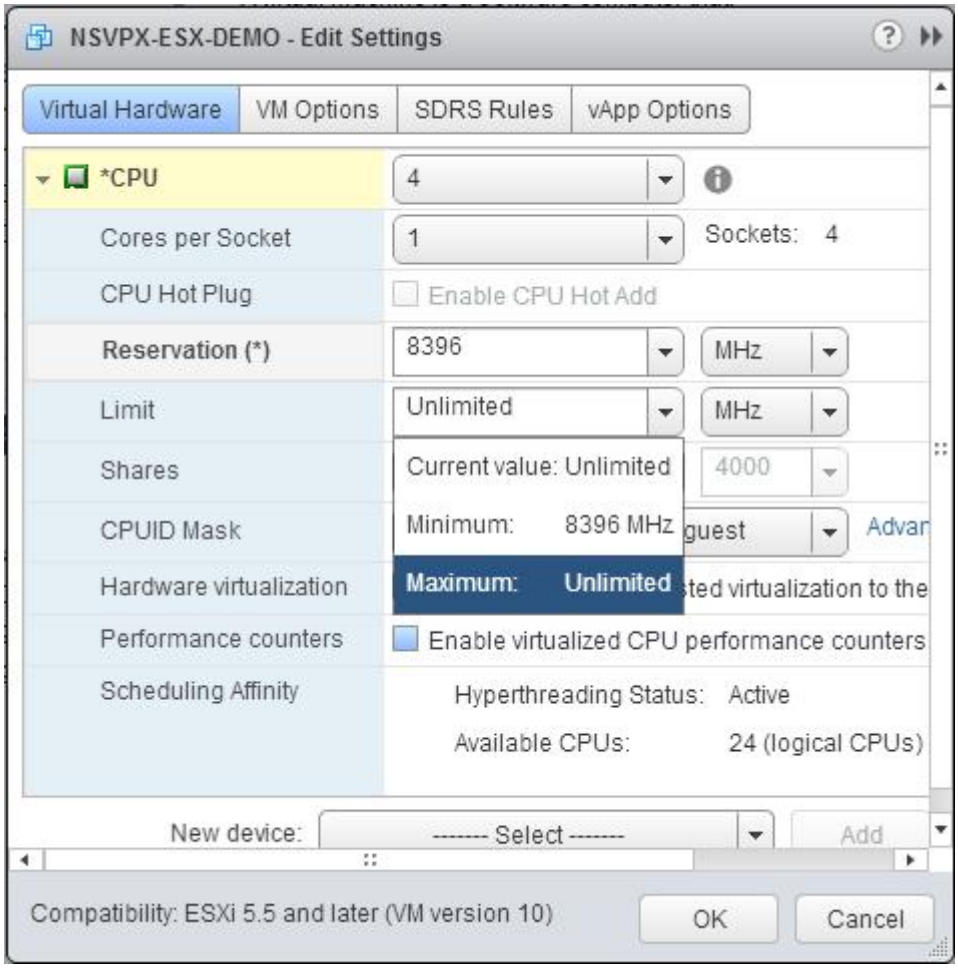
c. (オプション)「**CPU** ホットプラグ」フィールドで、「**CPU** ホットアドを有効にする」チェックボックスをオンまたはオフにします。

注： Citrix では、デフォルト（無効）をそのまま使用することをお勧めします。

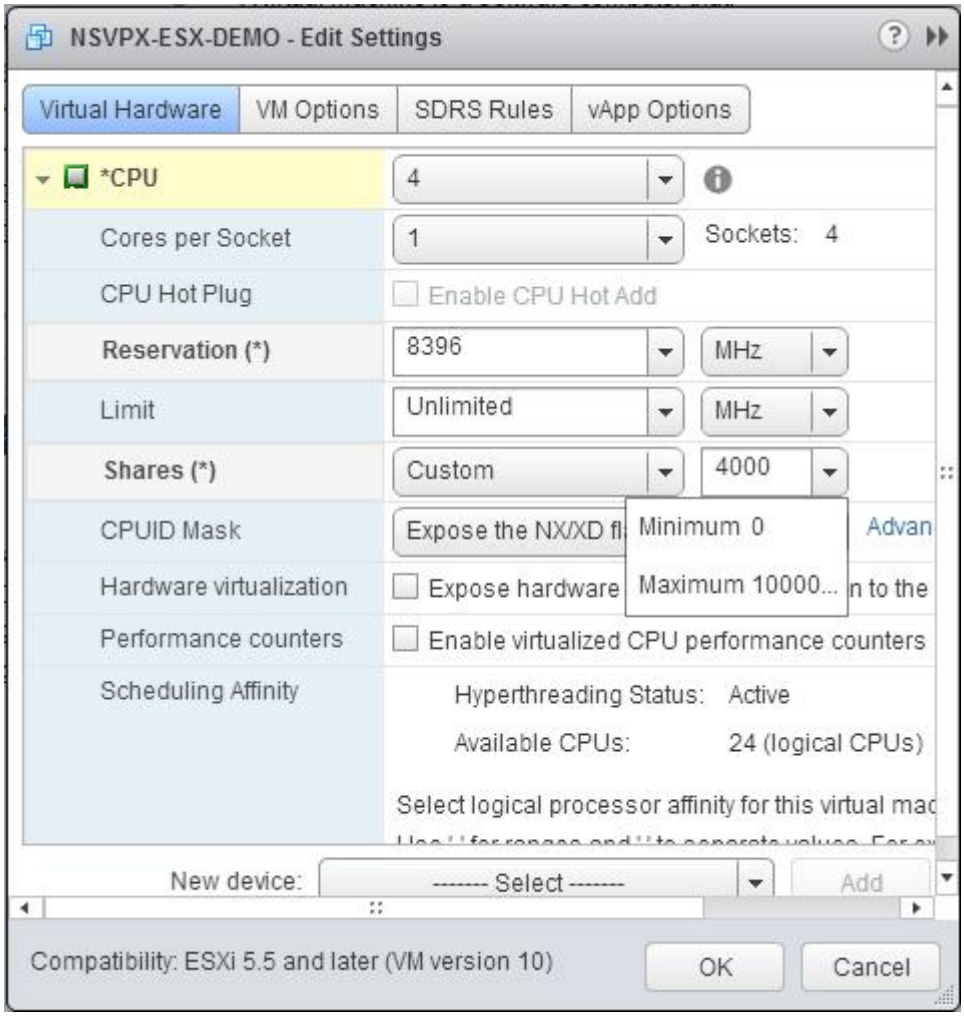
d. [予約] ドロップダウンリストで、最大値として表示される番号を選択します。



e. [制限] ドロップダウンリストで、最大値として表示されている数値を選択します。



f. [共有] ドロップダウンリストで、[カスタム] を選択し、最大値として表示される数を指定します。



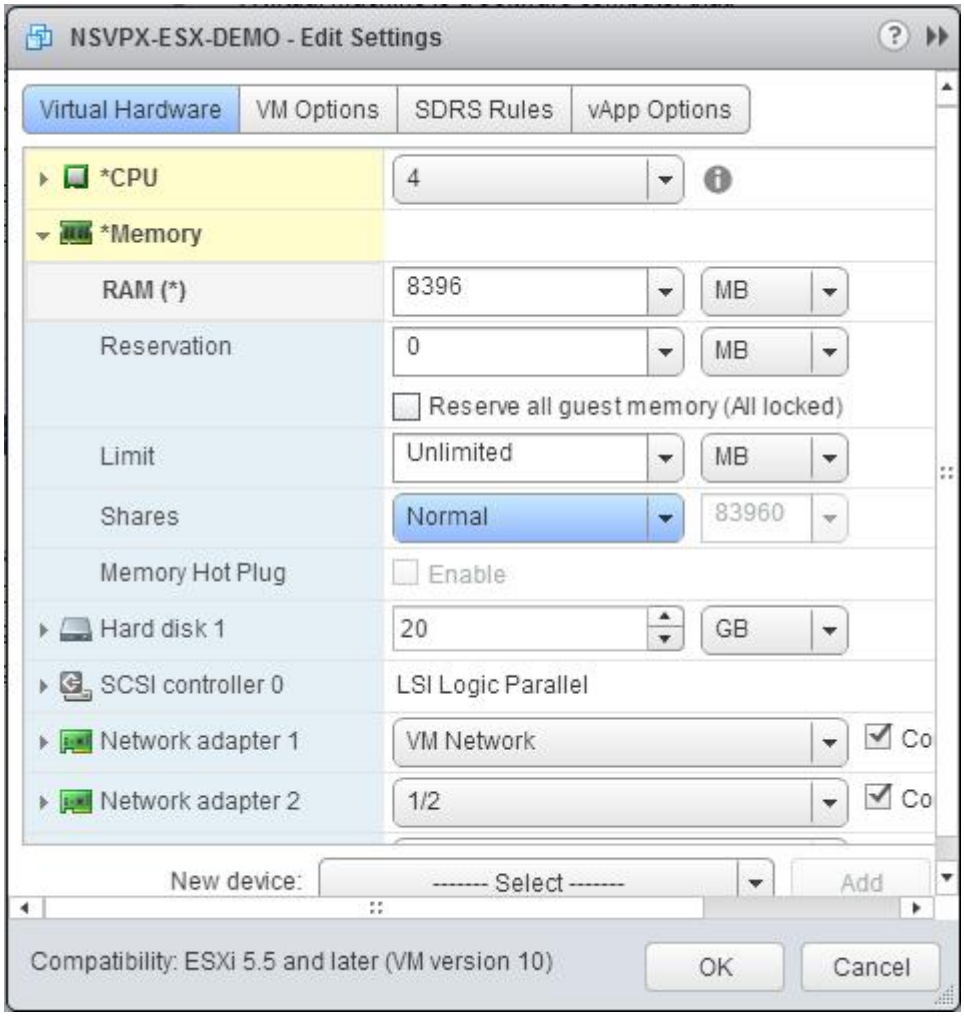
6. [メモリ] セクションで、次の設定を更新します。

   - メモリのサイズ
   - 予約
   - 上限
   - 共有

   値を次のように設定します:

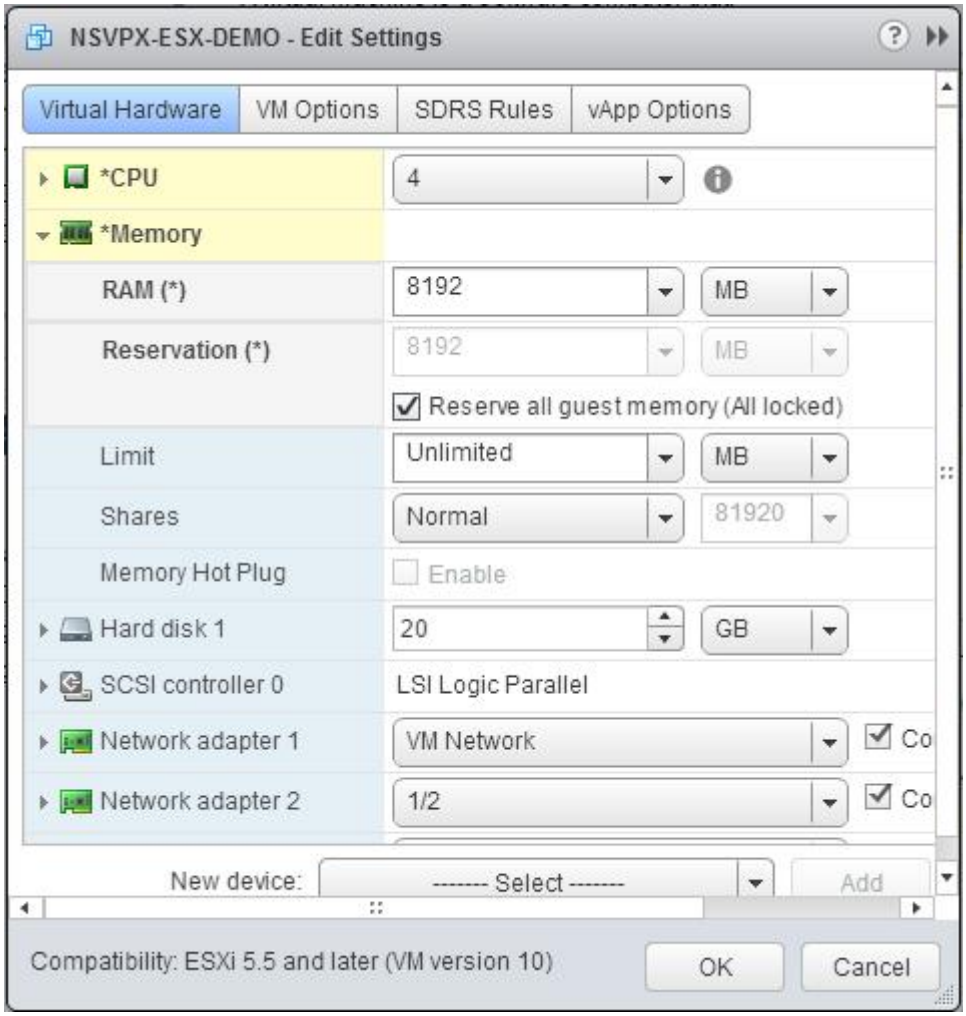
   a. [**RAM**] ドロップダウンリストで、RAM のサイズを選択します。これは vCPU の数 x 2 GB でなければなりません。たとえば、vCPU の数が 4 の場合、RAM = 4×2GB = 8GB になります。

   注： NetScaler VPX アプライアンスの高度なエディションまたはプレミアムエディションの場合は、各 vCPU に 4 GB の RAM を割り当ててください。たとえば、vCPU の数が 4 の場合、RAM = 4×4GB = 16GB になります。

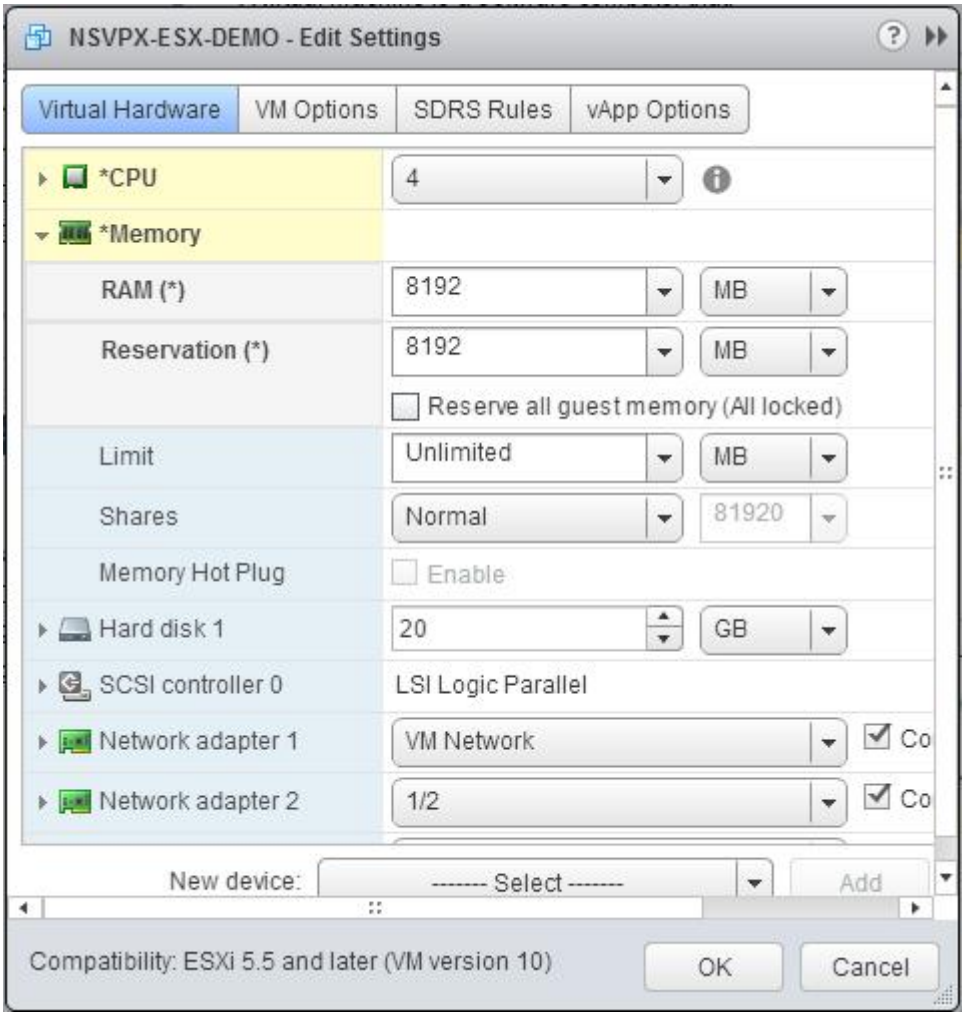


b. [予約] ドロップダウンリストで、メモリ予約の値を入力し、[すべてのゲストメモリを予約する **(**すべてのロック済み**)**] チェックボックスをオンにします。メモリ予約は vCPU の数 x 2 GB である必要があります。たとえば、vCPU の数が 4 の場合、メモリ予約は 4 x 2 GB = 8 GB である必要があります。

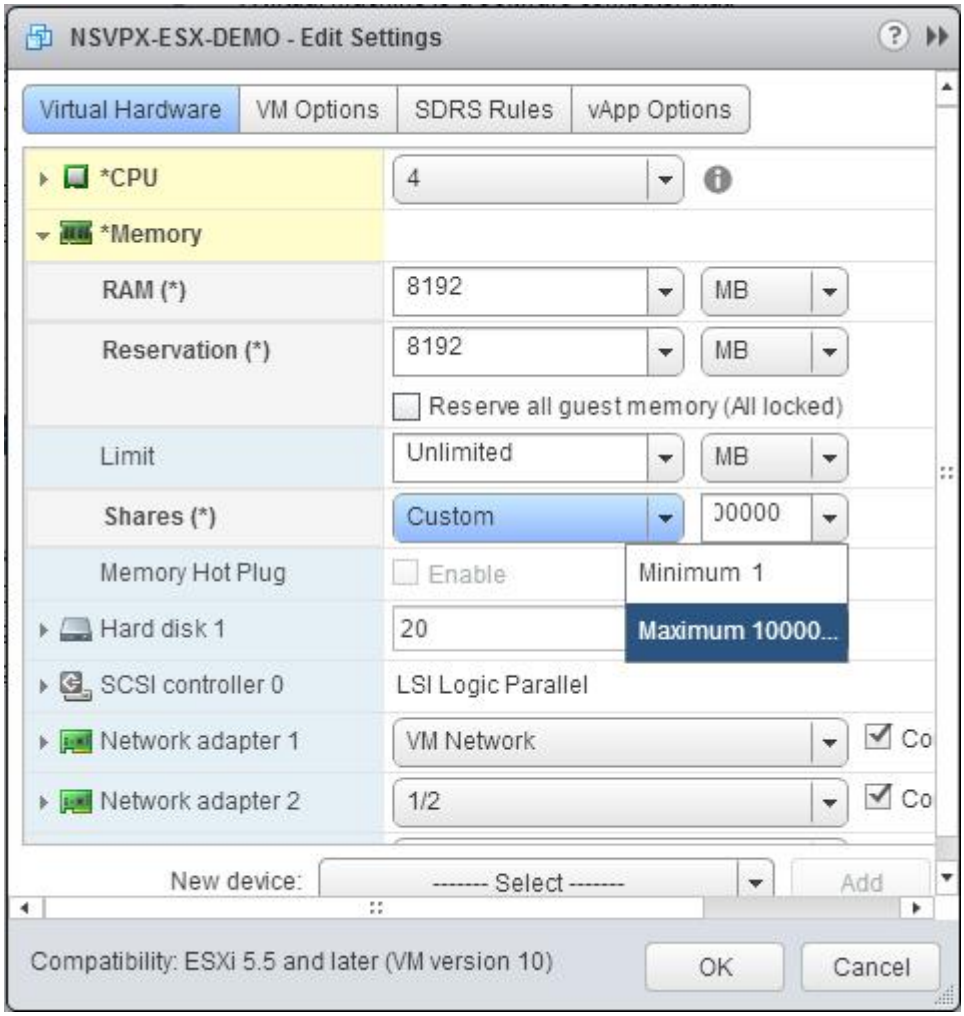
注： NetScaler VPX アプライアンスの高度なエディションまたはプレミアムエディションの場合は、各 vCPU に 4 GB の RAM を割り当ててください。たとえば、vCPU の数が 4 の場合、RAM = 4×4GB = 16GB になります。



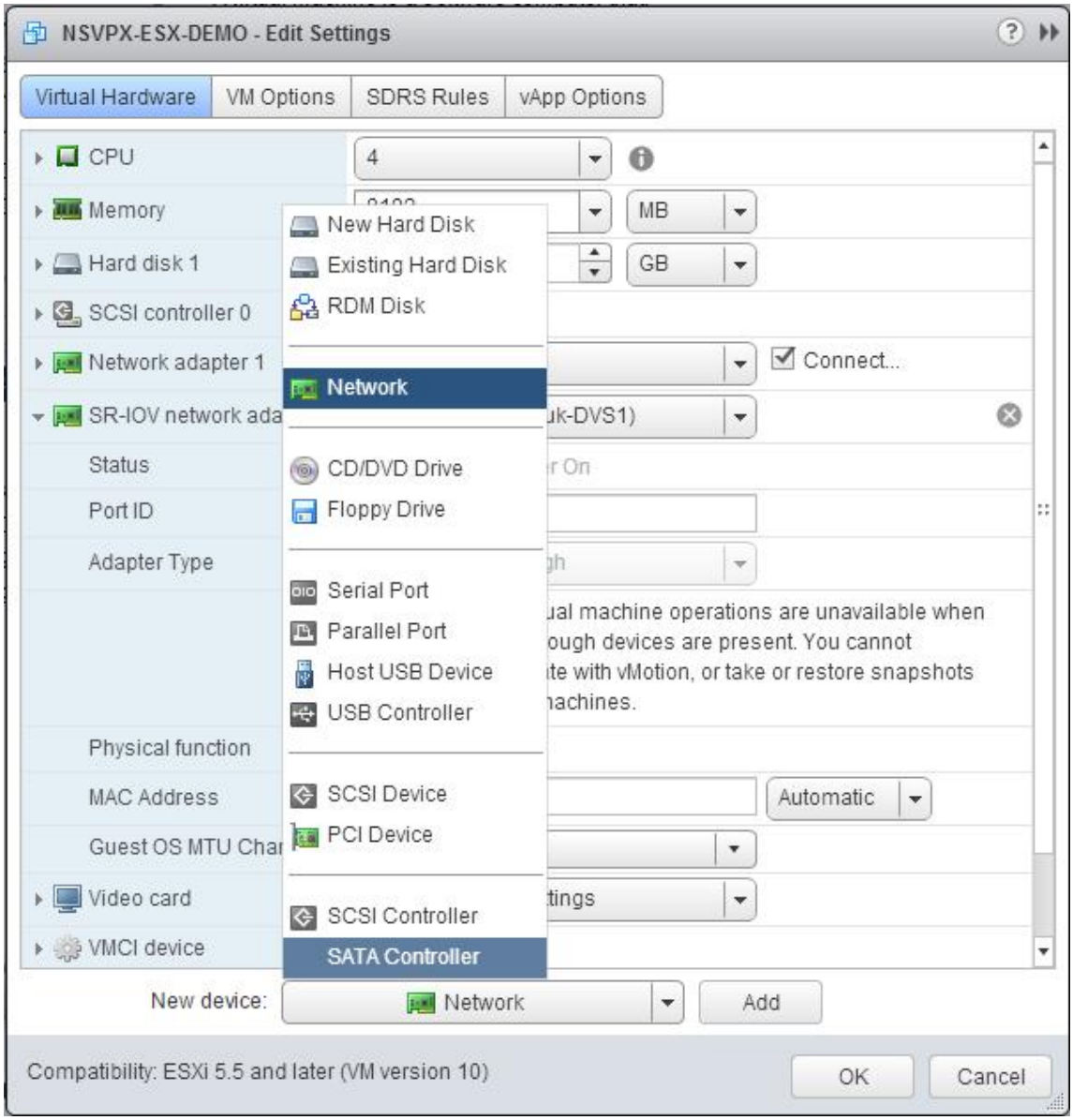
c. [制限】ドロップダウンリストで、最大値として表示されている数値を選択します。



d. [共有] ドロップダウンリストで、[カスタム] を選択し、最大値として表示される数値を選択します。

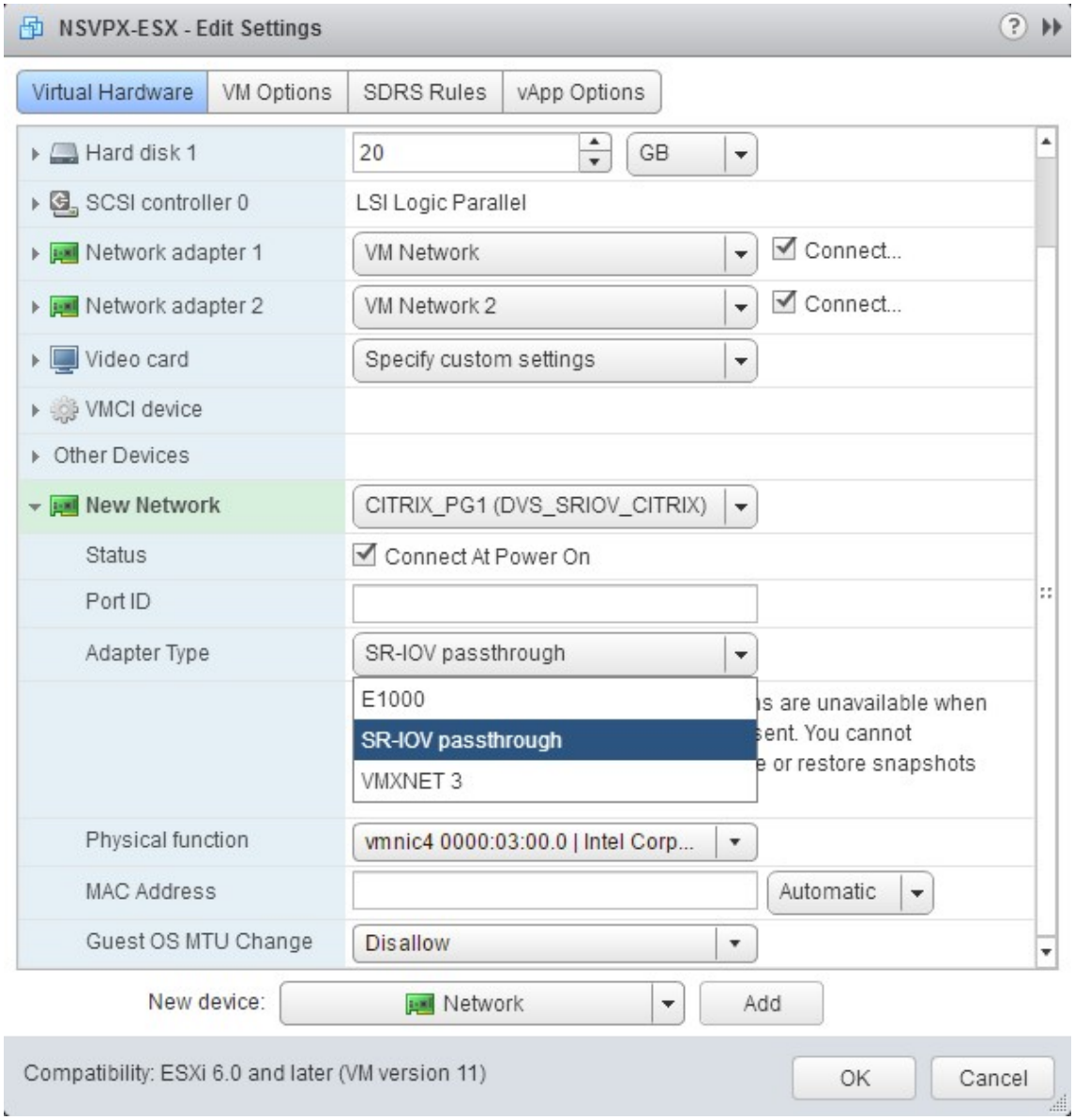


7. SR-IOV ネットワークインターフェイスを追加します。[新しいデバイス] ドロップダウンリストから [ネットワーク] を選択し、[追加] をクリックします。

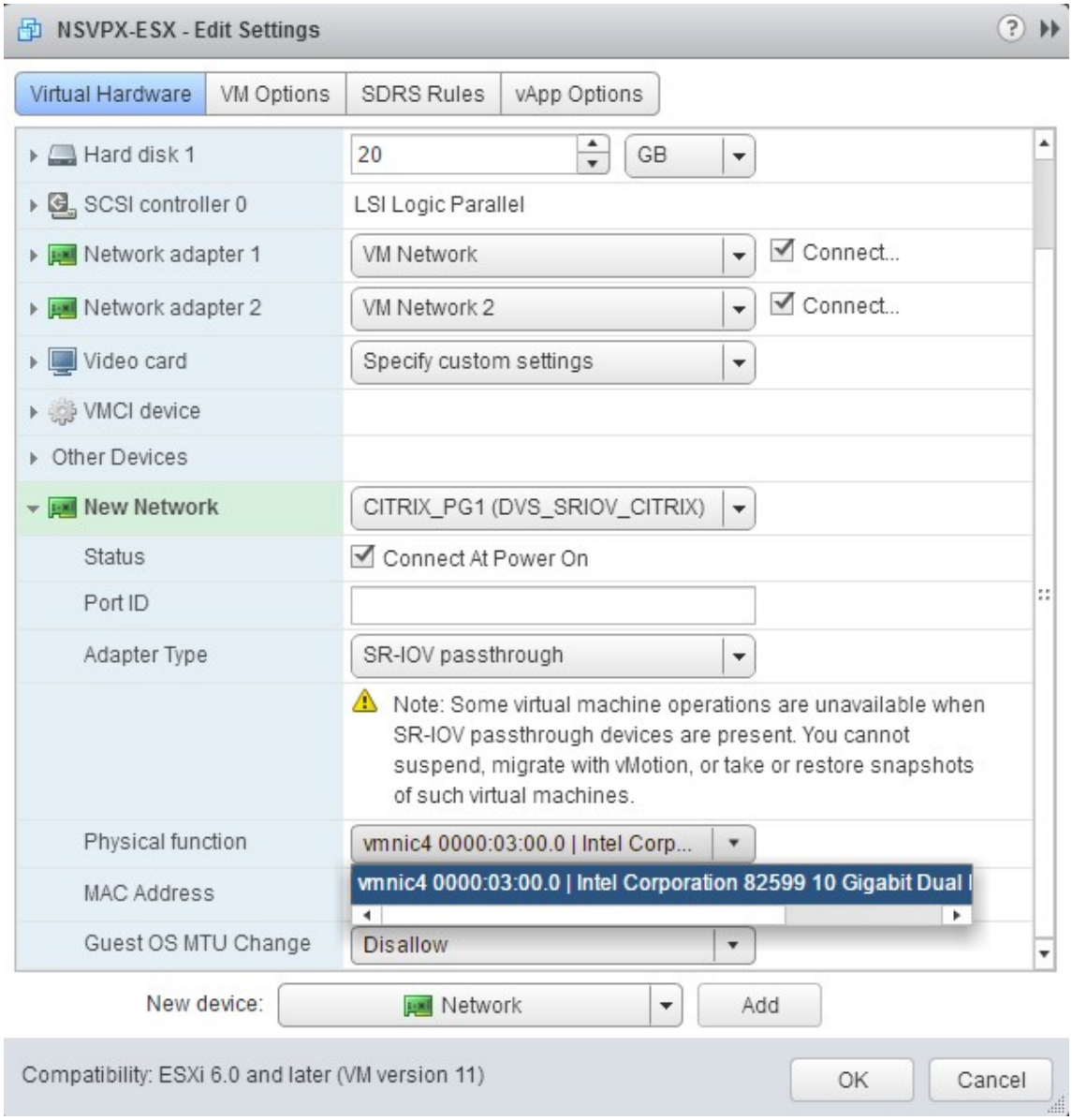


8. [新しいネットワーク] セクションで。ドロップダウンリストから、作成したPortgroupを選択し、次の操作を行います。

   a. [アダプタタイプ] ドロップダウンリストで、[**SR-IOV** パススルー] を選択します。



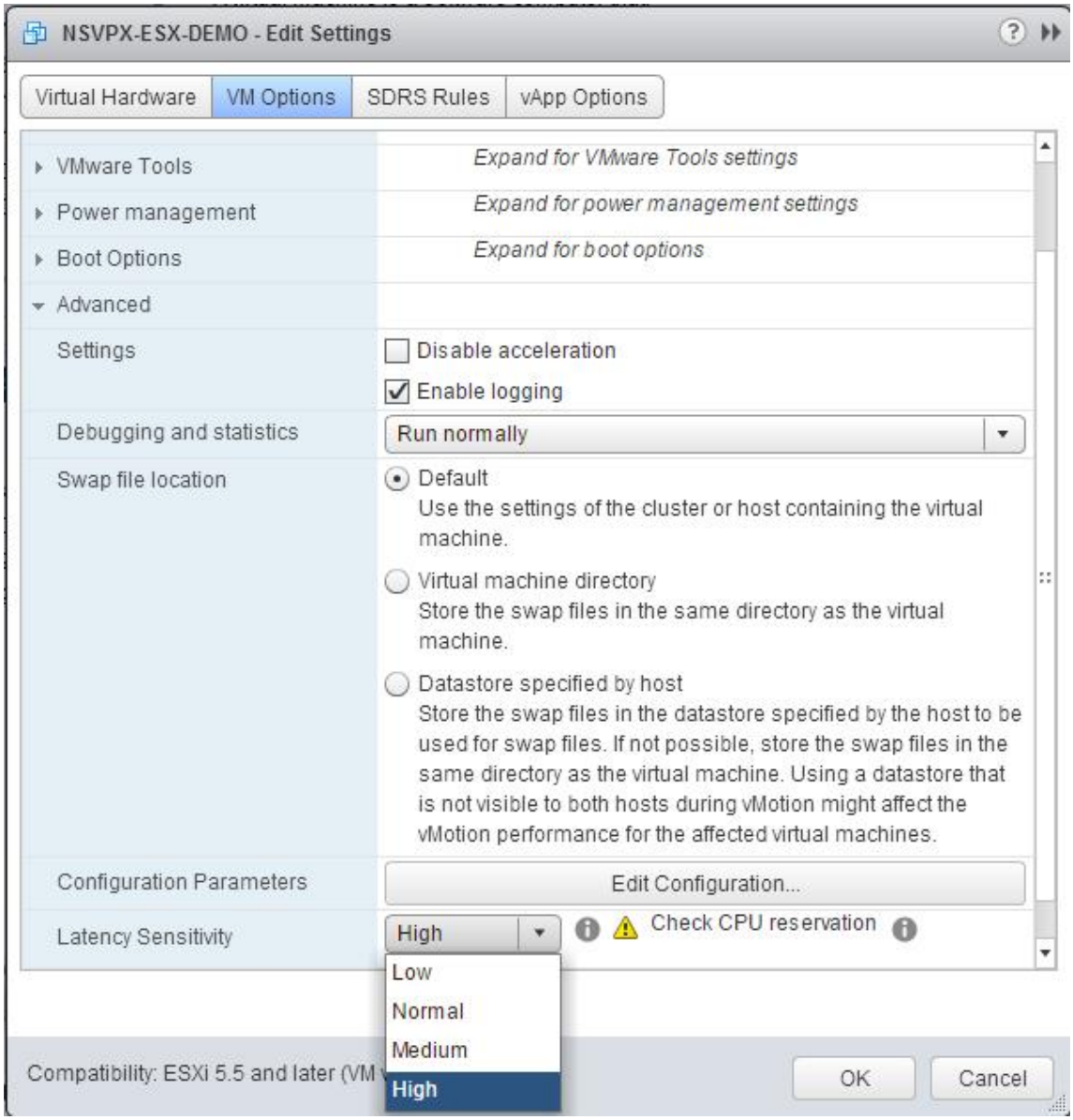
b. [物理機能] ドロップダウンリストで、`Portgroup`にマップされている物理アダプタを選択します。



c. ゲスト **OS** の **MTU** 変更ドロップダウンリストで、「許可」を選択します 。

9. [-設定の編集 <virtual_appliance>] ダイアログボックスで、[仮想マシンオプション] タブをクリックします。

10. [仮想マシンオプション] タブで、[詳細設定] セクションを選択します。[遅延感度] ドロップダウンリストから、[高] を選択します。



11. [**OK**] をクリックします。
12. NetScaler VPX インスタンスをパワーオンします。
13. NetScaler VPX インスタンスの電源が入ったら、次のコマンドを使用して構成を確認できます。

    > インターフェイスの概要を表示

出力には、設定したすべてのインターフェイスが表示されている必要があります。

```
> show interface summary
-------------------------------------------------------------------------------

      Interface  MTU        MAC                Suffix
-------------------------------------------------------------------------------
```



```
1     0/1         1500        00:0c:29:1b:81:0b    NetScaler Virtual
   Interface
2     10/1        1500        00:50:56:9f:0c:6f    Intel 82599 10G VF
   Interface
3     10/2        1500        00:50:56:9f:5c:1e    Intel 82599 10G VF
   Interface
4     10/3        1500        00:50:56:9f:02:1b    Intel 82599 10G VF
   Interface
5     10/4        1500        00:50:56:9f:5a:1d    Intel 82599 10G VF
   Interface
6     10/5        1500        00:50:56:9f:4e:0b    Intel 82599 10G VF
   Interface
7     LO/1        1500        00:0c:29:1b:81:0b    Netscaler Loopback
   interface
 Done
> show inter 10/1
1)      Interface 10/1 (Intel 82599 10G VF Interface) #1
        flags=0xe460 <ENABLED, UP, UP, HAMON, 802.1q>
        MTU=1500, native vlan=55, MAC=00:50:56:9f:0c:6f, uptime 0
           h21m53s
        Actual: media FIBER, speed 10000, duplex FULL, fctl NONE,
           throughput 10000
        LLDP Mode: NONE,                LR Priority: 1024

        RX: Pkts(838020742) Bytes(860888485431) Errs(0) Drops(2527)
           Stalls(0)
        TX: Pkts(838149954) Bytes(860895860507) Errs(0) Drops(0) Stalls
           (0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
 Done
```

E1000 から SR-IOV または VMXNET3 ネットワークインターフェイスへの NetScaler VPX の移行

August 15, 2023

2018 年 5 月 24 日

E1000 ネットワークインターフェイスを使用する既存の NetScaler VPX インスタンスを、SR-IOV または VMXNET3 ネットワークインターフェイスを使用するように構成できます。

SR-IOV ネットワークインターフェイスを使用するように既存の NetScaler ADC VPX インスタンスを構成するには、「SR-IOV ネットワークインターフェイスを使用するように NetScaler ADC VPX インスタンスを構成する」を参照してください。

VMXNET3 ネットワークインターフェイスを使用するように既存の NetScaler ADC VPX インスタンスを構成するには、VMXNET3 ネットワークインターフェイスを使用するように NetScaler ADC VPX インスタンスを構成するを

参照してください。

PCI パススルーネットワークインターフェイスを使用するように NetScaler VPX インスタンスを構成する

August 15, 2023

概要

VMware ESX Server に NetScaler VPX インスタンスをインストールして構成したら、vSphere Web Client を使用して、PCI パススルーネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

PCI パススルー機能では、ゲスト仮想マシンからホストに接続された物理 PCI および PCIe デバイスに直接アクセスできます。

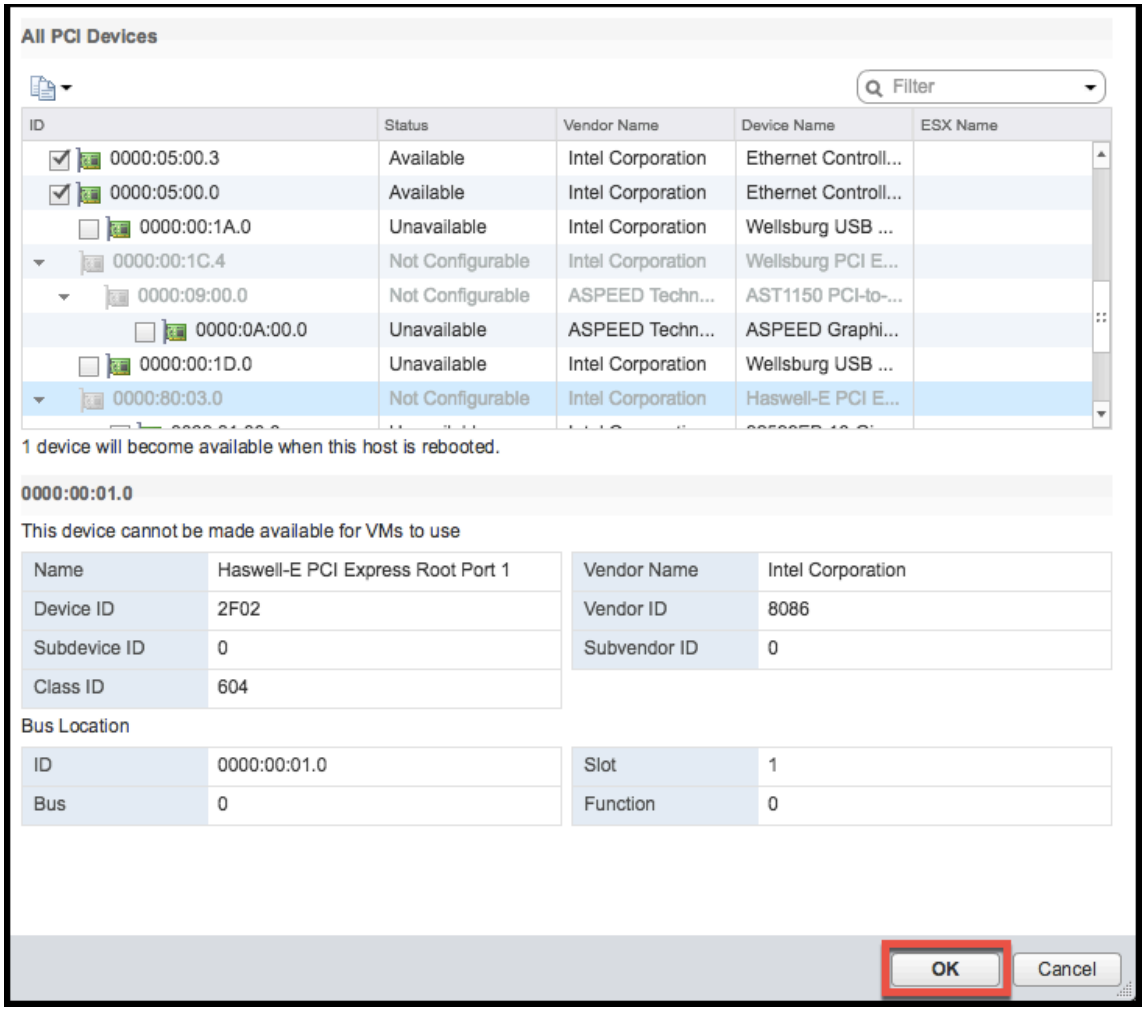
前提条件

- ホストの Intel XL710 NIC のファームウェアバージョンは、5.04 です。
- ホストに接続され構成されている PCI パススルーデバイス
- サポートされている NIC:
  - Intel X710 10G NIC
  - Intel XL710 デュアルポート 40G NIC
  - Intel XL710 シングルポート 40G NIC

ホスト上のパススルー・デバイスの構成

仮想マシンでパススルー PCI デバイスを構成する前に、ホストマシン上でそれを構成する必要があります。ホストでパススルーデバイスを構成するには次の手順を実行します。

1. vSphere Web クライアントのナビゲーターパネルからホストを選択します。
2. 管理 > 設定 > **PCI** デバイスをクリックします。すべての利用可能なパススルーデバイスが表示されます。
3. 設定するデバイスを右クリックし、[編集] をクリックします。
4. **[PCI** デバイスの可用性の編集] ウィンドウが表示されます。
5. パススルーに使用するデバイスを選択し、[**OK**] をクリックします。

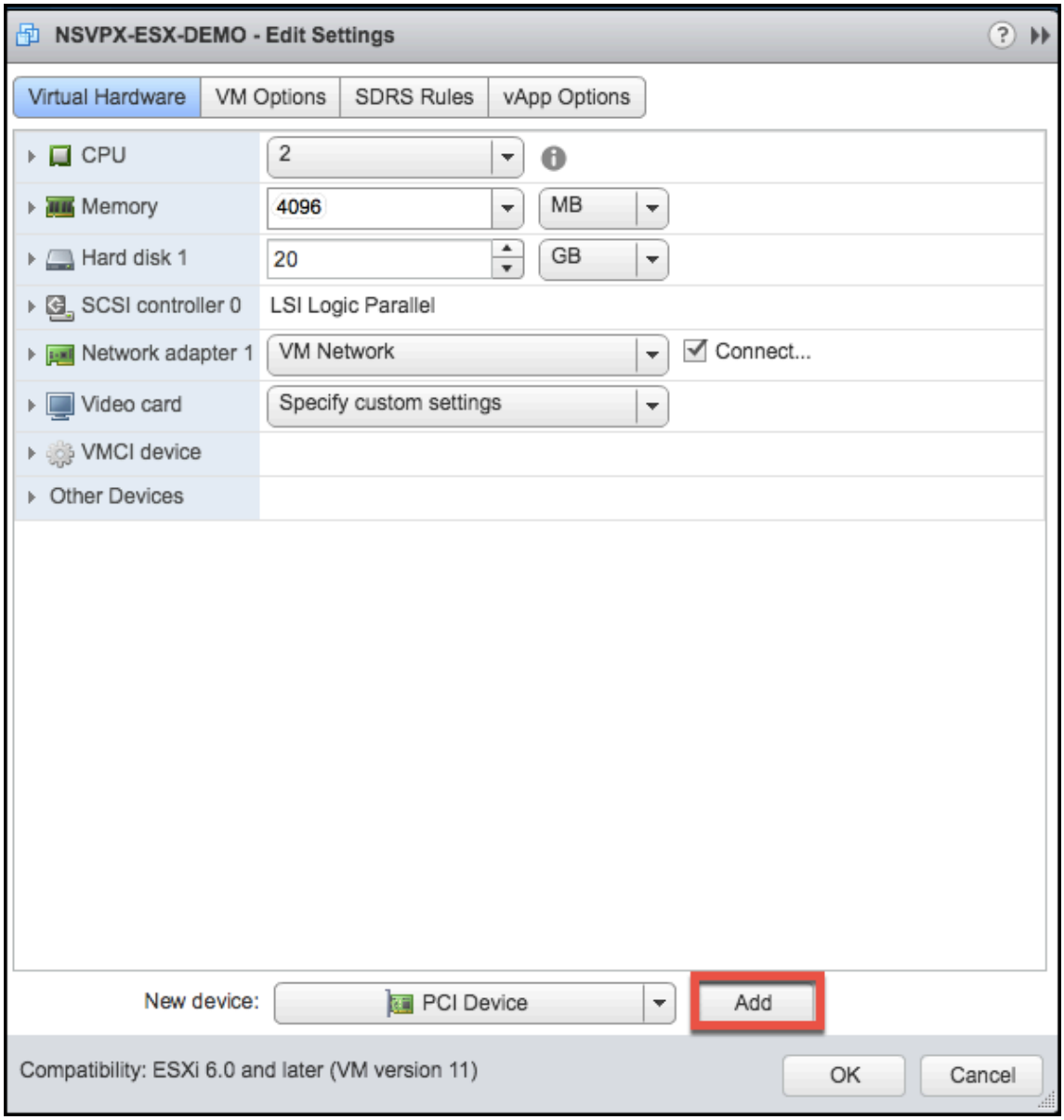


6. ホストマシンを再起動します。

NetScaler VPX インスタンスでパススルーデバイスを構成する

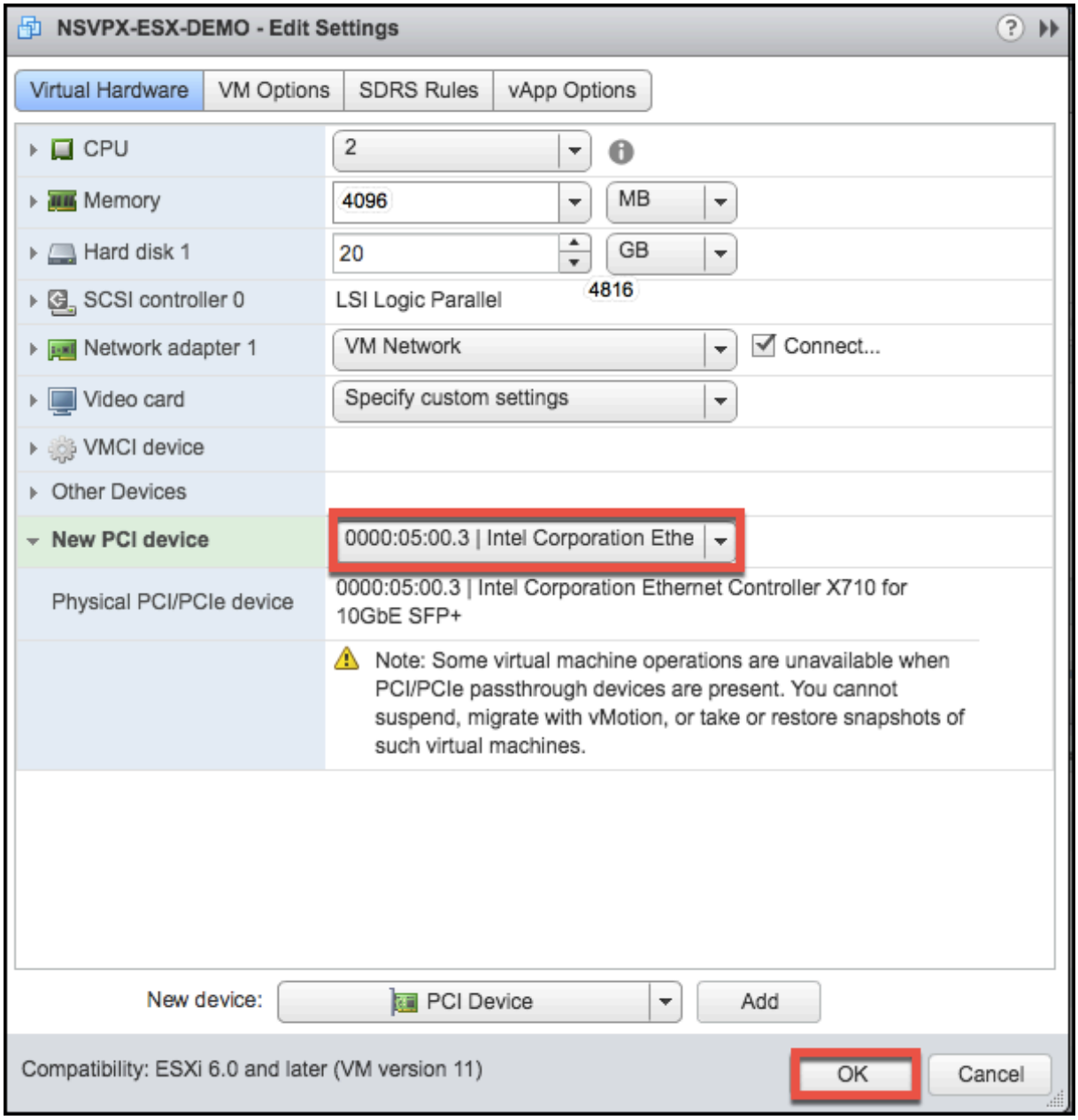
次の手順に従って、NetScaler VPX インスタンスでパススルー PCI デバイスを構成します。

1. 仮想マシンの電源を切ります。
2. 仮想マシンを右クリックして、[設定の編集] を選択します。
3. [仮想ハードウェア] タブで、[新しい ****** デバイス**]** ドロップダウンメニューから **[PCI** デバイス**]** を選択し、**[**追加 ******] をクリックします。



4. [**New PCI device**] を展開し、ドロップダウンリストから仮想マシンに接続するパススルーデバイスを選択し、[**OK**] をクリックします。

> 注
>
> VMXNET3 ネットワークインターフェイスと PCI パススルーネットワークインターフェイスは共存できません。



1. ゲスト仮想マシンの電源を入れます。

PCI パススルーネットワークインターフェイスを使用するように NetScaler VPX を構成する手順を完了しました。

VMware ESX ハイパーバイザーでの NetScaler ADC アプライアンスの初回起動時に NetScaler ADC VPX の構成を適用する

August 15, 2023

NetScaler VPX 構成は、VMware ESX ハイパーバイザーでの NetScaler ADC アプライアンスの初回起動時に適用できます。そのため、特定のセットアップまたは VPX インスタンスが起動される時間が大幅に短縮されることがあります。

プレブートユーザーデータとその形式について詳しくは、クラウドでの NetScaler ADC アプライアンスの初回起動時に NetScaler ADC VPX 構成を適用するを参照してください。

> 注:
>
> ESX でプレブートユーザーデータを使用してブートストラップするには、<NS-CONFIG> セクションでデフォルトゲートウェイ設定を渡す必要があります。<NS-CONFIG>タグの内容の詳細については、[Sample-<NS-CONFIG>-section] (apply-preboot-userdata-on-esx-vpx.html #sample-<ns-config>-section) を参照してください。

サンプル**<NS-CONFIG>**セクション**:**

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0  10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP> 10.102.38.216 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

ESX ハイパーバイザーでプレブートユーザーデータを提供する方法

Web クライアントまたは vSphere クライアントから ESX ハイパーバイザーのプレブートユーザーデータを提供するには、次の 2 つの方法があります。

- CD/DVD ISO を使用する
- OVF プロパティを使用する

CD/DVD ISO を使用したユーザーデータの提供

VMware vSphere クライアントを使用すると、CD/DVD ドライブを使用して ISO イメージとしてユーザーデータを VM に注入できます。

CD/DVD ISO を使用してユーザーデータを提供するには、次の手順に従います。

1. プレブートユーザーデータコンテンツを含むファイル名userdataでファイルを作成します。<NS-CONFIG>タグの内容について詳しくは、「サンプル」<NS-CONFIG> セクションを参照してください。

   > 注: ファイル名は、userdataとして厳密に使用する必要があります。

2. userdataファイルをフォルダーに保存し、フォルダーを使用して ISO イメージを構築します。

   userdataファイルを使用して ISO イメージを構築するには、次の 2 つの方法があります。

   - PowerISO などの任意の画像処理ツールを使用します。
   - Linux mkisofs でコマンドを使う。

   次の設定例は、Linux mkisofs でコマンドを使用して ISO イメージを生成する方法を示しています。

```
root@ubuntu:~/sai/14jul2021# ls -l total 4
drwxr-xr-x 2 root root 4096 Jul 14 12:32 esx_preboot_userdata
root@ubuntu:~/sai/14jul2021#
root@ubuntu:~/sai/14jul2021# ls -l esx_preboot_userdata/total 4
-rw-r--r-- 1 root root 3016 Jul 14 12:32 userdata
root@ubuntu:~/sai/14jul2021# mkisofs -o esx_preboot_userdata.iso
    ./esx_preboot_userdata
I: -input-charset not specified, using utf-8 (detected in locale
    settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 112
Path table size(bytes): 10
Max brk space used 0
176 extents written (0 MB)
root@ubuntu:~/sai/14jul2021# ls -lh
total 356K
drwxr-xr-x 2 root root 4.0K Jul 14 12:32 esx_preboot_userdata
-rw-r--r-- 1 root root 352K Jul 14 12:34 esx_preboot_userdata.iso

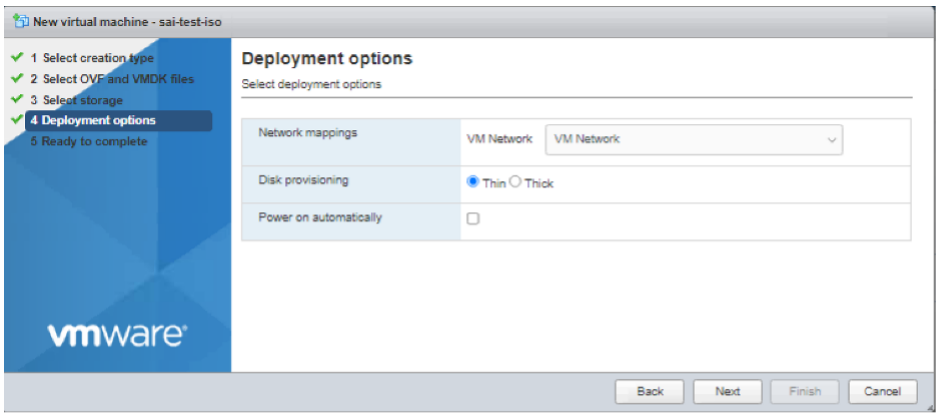
root@ubuntu:~/sai# ls preboot_userdata_155_193 userdata
root@ubuntu:~/sai# mkisofs -o preboot_userdata_155_193.iso ./
    preboot_userdata_155_193
I: -input-charset not specified, using utf-8 (detected in locale
    settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 112
Path table size(bytes): 10
Max brk space used 0
```



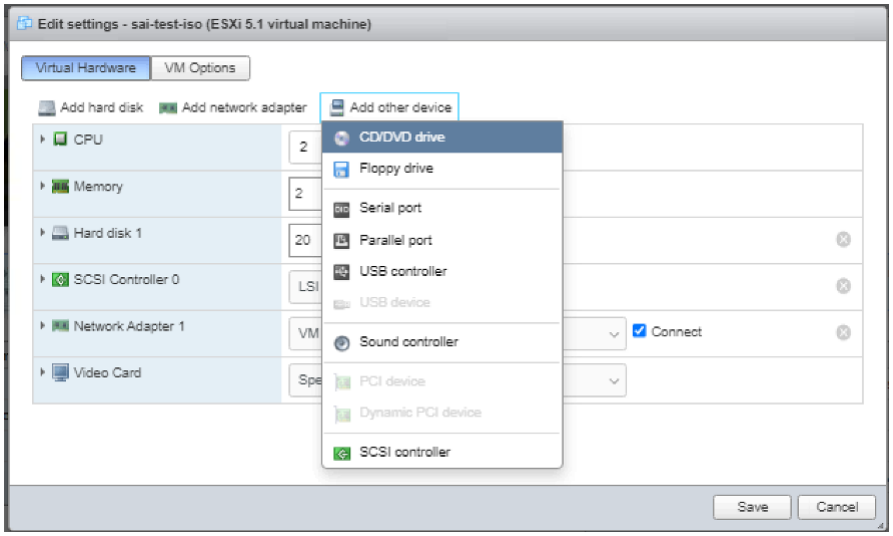
```
176 extents written (0 MB)

<!--NeedCopy-->
```

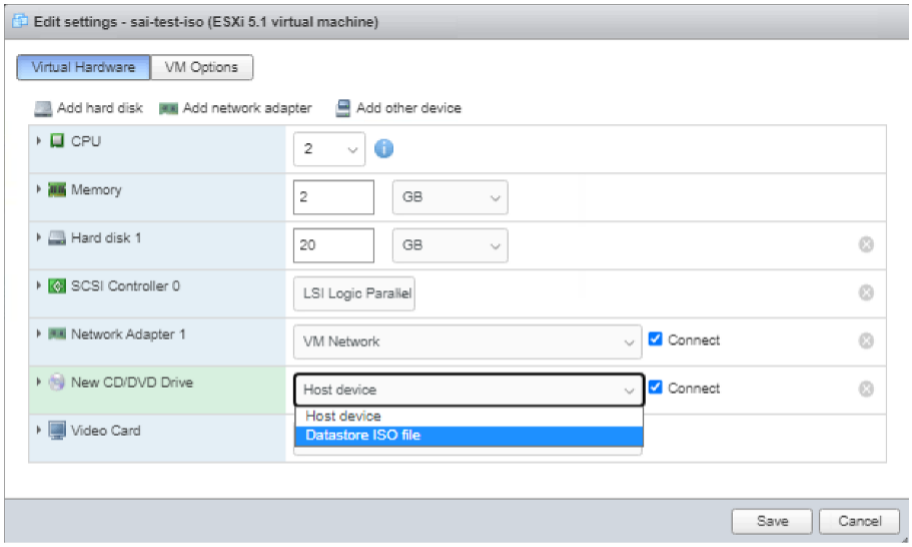
3. 標準の展開プロセスを使用して NetScaler ADC VPX インスタンスをプロビジョニングし、仮想マシンを作成します。ただし、VM の電源を自動的にオンにしないでください。



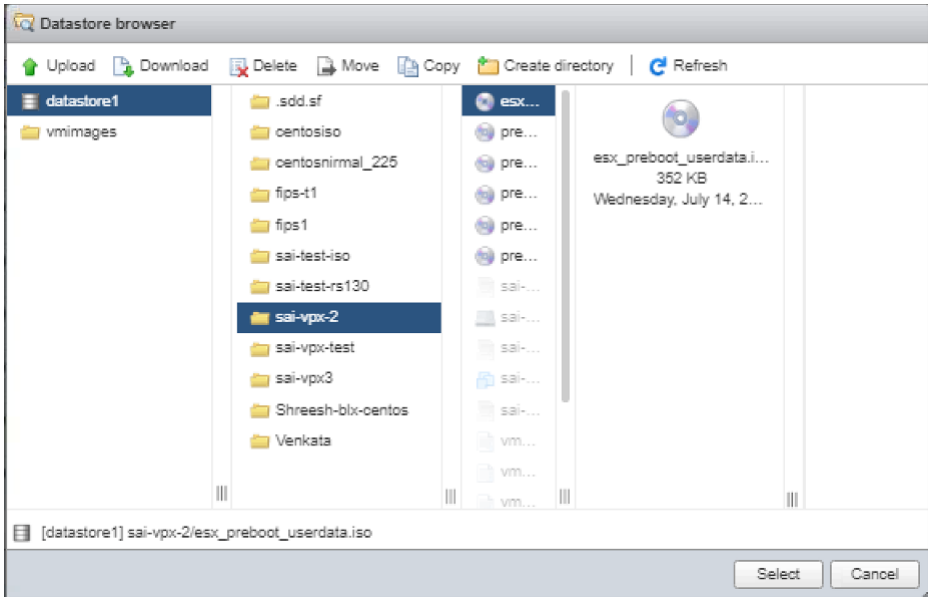
4. VM が正常に作成されたら、ISO ファイルを CD/DVD ドライブとして VM に接続します。



5. [新規 **CD/DVD** ドライブ] に移動し、ドロップダウンメニューから [データストア **ISO** ファイル] を選択します。



6. vSphere Client でデータストアを選択します。



7. VM の電源を入れます。

ESX Web クライアントの **OVF** プロパティを使用してユーザーデータを提供

OVF プロパティを使用してユーザーデータを提供する手順は、次のとおりです。

1. ユーザーデータコンテンツを含むファイルを作成します。


```
root@ubuntu:~/sai/14jul2021# cat esx_userdata.xml
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0 10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP>    10.102.38.219 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

2. ユーザーデータコンテンツを Base64 エンコーディングでエンコードします。Base64 エンコーディングは、次の 2 つの方法を使用して実行できます。

   - Linux では、以下のコマンドを使用します。

     ```
     base64 <userdata-filename>  >   <outuput-file>
     <!--NeedCopy-->
     ```

     例:

     ```
     base64 esx_userdata.xml  > esx_userdata_b64
     <!--NeedCopy-->
     ```

     ```
     root@ubuntu:~/sai/14jul2021# base64 esx_userdata.xml > esx_userdata_b64
     root@ubuntu:~/sai/14jul2021#
     root@ubuntu:~/sai/14jul2021# cat esx_userdata_b64
     PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+CglhZGQgcm91dGUgMC4wLjAuMCAw
     LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KCiAgICA8TlMtQk9PVFNUUkFQPgog
     ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1CT09U
     U1RSQVA+CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT
     VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAgICAg
     ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGgwIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAgICAg
     ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgPFNVQk5FVC1N
     QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
     RS1DT05GSUc+CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg==
     ```

   - オンラインツールを使用して、Base64 Encode や Decode などのユーザーデータコンテンツをエンコードします。

3. ESX ハイパーバイザー上の NetScaler ADC **VPX** インスタンスの **OVF** テンプレートに製品セクションを含めます。

   サンプル製品セクション:

   ```
   <ProductSection>

     <Info>Information about the installed software</Info>
     <Product>NSVPX-VSK Template</Product>
   ```



```
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>

  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
     userConfigurable="true" ovf:value="">

    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
<!--NeedCopy-->
```

4. Product セクションで、base64 でエンコードされたユーザーデータを guestinfo.userdata の ovf:value プロパティとして提供します。

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>
  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
     userConfigurable="true"
    ovf:value="PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+
       CglhZGQgcm91dGUgMC4wLjAuMCAw
    LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KCiAgICA8TlMtQk9PVFNUUkFQ

    ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1C

    U1RSQVA+
       CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT

    VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAg

    ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGggIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAg

    ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg

    ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgICAgICAg

    QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+
       CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
    RS1DT05GSUc+
       CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg
       ==">

    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
<!--NeedCopy-->
```


5. 変更した OVF テンプレートを [Product] セクションで仮想マシン展開に使用します。

```
Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
 Done
> sh ns ver
        NetScaler NS13.0: Build 83.9005.nc, Date: Jul 13 2021, 02:56:05   (64-bit)
 Done
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  S
tate
        ---------        --------------  ----             ----     ---      ----     -------  -
-----
1)      10.102.38.219    0               NetScaler IP     Active   Enabled  Enabled  NA       E
nabled
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State    Traffic Domain  Type
        -------          -------          ---------------  -----   -----    --------------  ----
1)      0.0.0.0          0.0.0.0          10.102.38.1      0       UP       0               STATI
C
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP       0               PERMA
NENT
3)      10.102.38.0      255.255.255.0    10.102.38.219    0       UP       0               DIREC
T
 Done
```

ESX vSphere クライアントの OVF プロパティを使用してユーザーデータを提供

ESX vSphere クライアントから OVF プロパティを使用してユーザーデータを提供するには、次の手順に従います。

1. ユーザーデータコンテンツを含むファイルを作成します。

```
root@ubuntu:~/sai/14jul2021# cat esx_userdata.xml
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0 10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP>    10.102.38.219 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

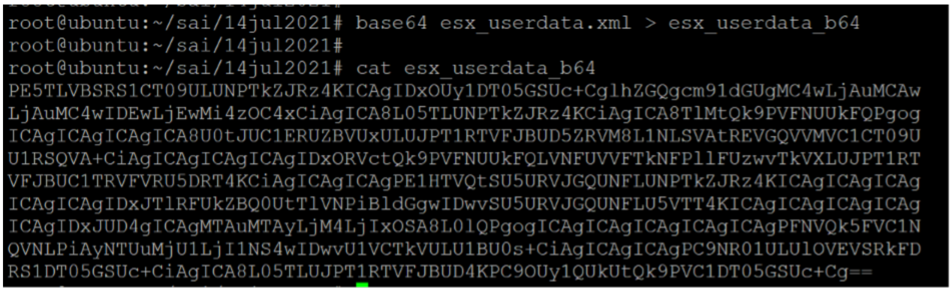
2. ユーザーデータコンテンツを Base64 エンコーディングでエンコードします。Base64 エンコーディングは、次の 2 つの方法を使用して実行できます。

   - Linux では、以下のコマンドを使用します。

```
base64 <userdata-filename>  >   <outuput-file>
<!--NeedCopy-->
```


例:

```
base64 esx_userdata.xml  > esx_userdata_b64
<!--NeedCopy-->
```



- オンラインツールを使用して、Base64 Encode や Decode などのユーザーデータコンテンツをエンコードします。

3. ESX ハイパーバイザー上の NetScaler ADC **VPX** インスタンスの **OVF** テンプレートに製品セクションを含めます。

   サンプル製品セクション:

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>

  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
     userConfigurable="true" ovf:value="">

    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
<!--NeedCopy-->
```

4. Product セクションで、base64 でエンコードされたユーザーデータを`guestinfo.userdata`の`ovf:value`プロパティとして提供します。

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.Citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>
  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
     userConfigurable="true"
```

```
9       ovf:value="PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+
            CglhZGQgcm91dGUgMC4wLjAuMCAw
10      LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KCiAgICA8TlMtQk9PVFNUUkFQ
11      ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1C
12      U1RSQVA+
            CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT
13      VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAg
14      ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGgwIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAg
15      ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgPFNVQk5F
16      QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+
            CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
17      RS1DT05GSUc+
            CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg
            ==">
18
19      <Label>Userdata</Label>
20      <Description> Userdata for ESX VPX </Description>
21    </Property>
22
23  </ProductSection>
24  <!--NeedCopy-->
```

5. 次のように `ovf:transport="com.vmware.guestInfo"` 、プロパティを仮想ハードウェアセクションに追加します。

```
1  <VirtualHardwareSection  ovf:transport="com.vmware.guestInfo">
2  <!--NeedCopy-->
```

6. 変更した OVF テンプレートを [Product] セクションで仮想マシン展開に使用します。

```
Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
 Done
> sh ns ver
        NetScaler NS13.0: Build 83.9005.nc, Date: Jul 13 2021, 02:56:05   (64-bit)
 Done
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  S
tate
        ---------        --------------  ----             ----     ---      ----     -------  -
-----
1)      10.102.38.219    0               NetScaler IP     Active   Enabled  Enabled  NA       E
nabled
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.102.38.1      0       UP      0               STATI
C
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMA
NENT
3)      10.102.38.0      255.255.255.0    10.102.38.219    0       UP      0               DIREC
T
 Done
```


AWS の VMware クラウドに NetScaler ADC VPX インスタンスをインストールする

August 15, 2023

AWS 上の VMware クラウド (VMC) を使用すると、必要な数の ESX ホストを使用してクラウドソフトウェア定義データセンター (SDDC) を AWS 上に作成できます。AWS 上の VMC は、NetScaler VPX デプロイをサポートしています。VMC は、オンプレミスの vCenter と同じユーザー・インタフェースを提供します。ESX ベースの NetScaler ADC VPX デプロイメントと同じように機能します。

前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- 1 つの VMware SDDC が少なくとも 1 つのホストに存在している必要があります。
- NetScaler VPX アプライアンスセットアップファイルをダウンロードします。
- 仮想マシンが接続する適切なネットワークセグメントを VMware SDDC 上に作成します。
- VPX ライセンスファイルを入手します。NetScaler VPX インスタンスライセンスの詳細については、http://support.citrix.com/article/ctx131110の『*NetScaler VPX* ライセンスガイド』を参照してください。

VMware クラウドのハードウェア要件

次の表に、VMware SDDC が各 VPX nCore 仮想アプライアンスに対して提供する必要がある仮想コンピューティングリソースを示します。

表 1. NetScaler VPX インスタンスの実行に必要な最小限の仮想コンピューティングリソース

Component	条件
メモリ	2 GB
仮想 CPU（VCPU）	2
仮想ネットワークインターフェイス	VMware SDDC では、VPX ハードウェアをバージョン 7 以降にアップグレードした場合、最大 10 個の仮想ネットワークインターフェイスをインストールできます。
ディスク領域	20 GB

> 注
>
> ハイパーバイザーに必要なディスク領域は含まれません。

VPX 仮想アプライアンスを実稼働で使用するには、フルメモリ割り当てを予約する必要があります。

OVF ツール 1.0 のシステム要件

Open Virtualization Format Tool（OVF Tool）は、Windows および Linux システムで実行できるクライアントアプリケーションです。次の表は、最小システム要件を示しています。

表 2. OVF ツールのインストールに必要な最小システム要件

Component	条件
オペレーティングシステム	VMware からの詳細な要件については、http://kb.vmware.com/で『OVF ツールユーザーガイド』の PDF ファイルを検索してください。
CPU	最低 750MHz、1GHz 以上推奨
RAM	最小 1 GB、推奨 2 GB
NIC	100Mbps 以上の NIC。

OVF のインストールについては、http://kb.vmware.com/で『OVF ツールユーザーガイド』の PDF ファイルを検索してください。

NetScaler VPX セットアップファイルのダウンロード

VMware ESX 用の NetScaler VPX インスタンスセットアップパッケージは、オープン仮想マシン（OVF）フォーマット標準に準拠しています。これらのファイルは、Citrix Web サイトからダウンロードできます。ログオンするには、Citrix アカウントが必要です。Citrix アカウントをお持ちでない場合は、http://www.citrix.comのホームページにアクセスしてください。［新しいユーザー］リンクをクリックし、指示に従って新しい Citrix アカウントを作成します。

ログオンしたら、Citrix のホームページから次のパスをナビゲートします。

Citrix.com > ダウンロード **> NetScaler >** 仮想アプライアンス。

次のファイルを、ESX サーバーと同じネットワーク上のワークステーションにコピーします。3 つのファイルをすべて同じフォルダーにコピーします。

- NSVPX-ESX-<release number>--disk1.vmdk <build number> (たとえば、nsvpx-ESX-13.0-79.64-disk1.vmdk)
- NSVPX-ESX-<release number>-.ovf <build number> (たとえば、nsvpx-ESX-13.0-79.64.OVF)
- NSVPX-ESX-<release number>-.mf <build number> (たとえば、nsvpx-ESX-13.0-79.64.mf)

VMware クラウドへの **NetScaler ADC VPX** インスタンスのインストール

VMware SDDC をインストールして設定したら、SDDC を使用して VMware クラウドに仮想アプライアンスをインストールできます。インストールできる仮想アプライアンスの数は、SDDC で使用可能なメモリの量によって異なります。

NetScaler VPX インスタンスを VMware クラウドにインストールするには、次の手順に従います。

1. ワークステーションで VMware SDDC を開きます。
2. [ユーザー名] テキストボックスと [パスワード] テキストボックスに、管理者の認証情報を入力し、[ログイン] をクリックします。
3. ［**File**］メニューの［**Deploy OVF Template**］を選択します。
4. ［**OVF** テンプレートのデプロイ］ダイアログボックスの［ファイルからデプロイ］で、NetScaler VPX インスタンスセットアップファイルを保存した場所を参照し、.ovf ファイルを選択して、［次へ］をクリックします。

> 注：デフォルトでは、NetScaler VPX インスタンスは E1000 ネットワークインターフェイスを使用します。VMXNET3 インターフェイスで ADC を展開するには、E1000 ではなく VMXNET3 インターフェイスを使用するように OVF を変更します。

1. 仮想アプライアンス OVF テンプレートに表示されるネットワークを、VMware SDDC で設定したネットワークにマッピングします。[次へ] をクリックして VMware SDDC への仮想アプライアンスのインストールを開始します。
2. これで、NetScaler VPX インスタンスを起動する準備ができました。ナビゲーションペインで、インストールした NetScaler ADC VPX インスタンスを選択し、右クリックメニューから［パワーオン］を選択します。コンソールポートをエミュレートするには、［**Console**］タブをクリックします。
3. 別の仮想アプライアンスをインストールする場合は、ステップ 6 から繰り返します。
4. 管理ネットワークとして選択した同じセグメントから管理 IP アドレスを指定します。ゲートウェイには同じサブネットが使用されます。
5. VMware SDDC では、ネットワークセグメントに属するすべてのプライベート IP アドレスに対して NAT ルールとファイアウォールルールを明示的に作成する必要があります。

Microsoft Hyper-V サーバーに **NetScaler VPX** インスタンスをインストールします

August 15, 2023

NetScaler VPX インスタンスを Microsoft Windows Server にインストールするには、まず、十分なシステムリソースを備えたマシンに、Hyper-V ロールを有効にした Windows Server をインストールする必要があります。Hyper-V の役割をインストールするときは、仮想ネットワークを作成するために Hyper-V で使用されるサーバー上の NIC を必ず指定してください。一部の NIC は、ホスト用に確保できます。Hyper-V マネージャーを使用して NetScaler VPX インスタンスのインストールを実行します。

Hyper-V 用の NetScaler ADC VPX インスタンスは、仮想ハードディスク（VHD）形式で配信されます。CPU、ネットワークインターフェイス、ハードディスクのサイズと形式などの要素について、デフォルト構成が格納されています。NetScaler VPX インスタンスをインストールしたら、仮想アプライアンスにネットワークアダプタを構成し、仮想 NIC を追加してから、NetScaler IP アドレス、サブネットマスク、およびゲートウェイを割り当てて、仮想アプライアンスの基本構成を完了できます。

VPX インスタンスの初期構成後、アプライアンスを最新のソフトウェアリリースにアップグレードする場合は、NetScaler VPX スタンドアロンアプライアンスのアップグレードを参照してください。

> 注
>
> ISIS（Intermediate System-to-Intermediate System）プロトコルは、Hyper-V 2012 プラットフォーム上でホストされる NetScaler VPX 仮想アプライアンスではサポートされません。

NetScaler VPX インスタンスを Microsoft サーバーにインストールするための前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- Windows サーバーで Hyper-V ロールを有効にします。詳しくは、http://technet.microsoft.com/en-us/library/ee344837(WS.10).aspxを参照してください。
- 仮想アプライアンスセットアップファイルをダウンロードします。
- NetScaler VPX インスタンスのライセンスファイルを取得します。NetScaler VPX インスタンスライセンスの詳細については、http://support.citrix.com/article/ctx131110の『*NetScaler VPX* ライセンスガイド』を参照してください。

Microsoft のサーバハードウェア要件

次の表は、Microsoft サーバーの最小システム要件を示しています。

表 1. Microsoft サーバーの最小システム要件

Component	条件
CPU	1.4GHz 64 ビットプロセッサ
RAM	8 GB
ディスク領域	32GB 以上

次の表は、
各 NetScaler ADC VPX インスタンスの仮想コンピューティングリソースを示しています。

表 2. NetScaler VPX インスタンスの実行に必要な最小限の仮想コンピューティングリソース

Component	条件
RAM	4 GB
仮想 CPU	2
ディスク領域	20 GB
仮想ネットワーク インターフェイス	1

NetScaler VPX セットアップファイルをダウンロードする

Hyper-V 用の NetScaler ADC VPX インスタンスは、仮想ハードディスク（VHD）形式で配信されます。これらのファイルは、Citrix Web サイトからダウンロードできます。ログインするには Citrix アカウントが必要です。Citrix アカウントをお持ちでない場合は、http://www.citrix.comのホームページにアクセスし、［サインイン］**>**［マイアカウント］**>**［**Citrix** アカウントの作成］の順にクリックし、手順に従って Citrix アカウントを作成します。

NetScaler VPX インスタンスのセットアップファイルをダウンロードするには、次の手順に従います。

1. Web ブラウザーで、http://www.citrix.com/に移動します。
2. ユーザー名とパスワードを使用してサインインします。
3. ［**Downloads**］をクリックします。
4. 「製品の選択」ドロップダウンメニューで、「**NetScaler**（**NetScaler ADC**）」を選択します。
5. 「**NetScaler** リリース **X.X**」**>**「仮想アプライアンス」で、「**NetScaler VPX** リリース **X.X**」をクリックします。
6. 圧縮ファイルをサーバーにダウンロードします。

NetScaler VPX インスタンスを **Microsoft** のサーバーにインストールします

Microsoft Server で Hyper-V ロールを有効にし、仮想アプライアンスファイルを抽出したら、Hyper-V Manager を使用して NetScaler ADC VPX インスタンスをインストールできます。仮想マシンをインポートしてから仮想 NIC を構成し、Hyper-V によって作成された仮想ネットワークに関連付ける必要があります。

最大 8 つの仮想 NIC を構成できます。物理 NIC が DOWN になっても、同じホスト（サーバー）上の他の仮想アプライアンスと通信できるため、仮想アプライアンスでは仮想 NIC は UP と見なされます。

> 注
>
> 仮想アプライアンスの実行中は、設定を変更することができません。仮想アプライアンスをシャットダウンしてから変更を行います。

Hyper-V マネージャーを使用して **NetScaler VPX** インスタンスを **Microsoft** サーバーにインストールするには:

1. Hyper-V マネージャーを起動するには、[スタート] ボタンをクリックし、[管理ツール] をポイントして、[**Hyper-V** マネージャー] をクリックします。
2. ナビゲーションペインの **Hyper-V Manage**r で、NetScaler VPX インスタンスをインストールするサーバーを選択します。
3. [アクション] メニューで、[仮想マシンのインポート] をクリックします。
4. [仮想マシンのインポート] ダイアログボックスの [場所] で、NetScaler VPX インスタンスのソフトウェアファイルを含むフォルダーのパスを指定し、[仮想マシンをコピー（新しい一意の **ID** を作成）] を選択します。このフォルダーは、Snapshots フォルダー、Virtual Hard Disks フォルダー、および Virtual Machines フォルダーを格納する親フォルダーです。
5. 注：圧縮ファイルを受け取った場合は、ファイルをフォルダーに抽出したことを確認してからフォルダーのパスを指定してください。
6. **[**インポート**]** をクリックします。
7. インポートした仮想アプライアンスが [仮想マシン] の下に表示されていることを確認します。
8. 別の仮想アプライアンスをインストールするには、手順 **2** ～**6** を繰り返します。

> 重要
>
> 手順 **4** で、必ずファイルを別のフォルダーに解凍してください。

Hyper-V で **NetScaler ADC VPX** インスタンスを自動プロビジョニングする

NetScaler VPX インスタンスの自動プロビジョニングはオプションです。自動プロビジョニングを実行しない場合は、NetScaler 仮想アプライアンスによって IP アドレスなどを構成するためのオプションが提供されます。

Hyper-V で NetScaler ADC VPX インスタンスを自動プロビジョニングするには、次の手順に従います。

1. 例に示されている説明に従い、XML ファイルを使用して ISO9660 準拠の ISO イメージを作成します。xml ファイルの名前が **userdata** であることを確認してください。

   XML ファイルから ISO ファイルを作成するには、以下を使用します。

   • PowerISO などの任意の画像処理ツール。

   • Linux の `mkisofs`コマンド。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<Environment xmlns:oe=`"http://schemas.dmtf.org/ovf/environment/1`
    "

xmlns:xsi=`"http://www.w3.org/2001/XMLSchema-instance`"

oe:id=""

```



```
xmlns=`"http://schemas.dmtf.org/ovf/environment/1`">

<PlatformSection>

<Kind>HYPER-V</Kind>

<Version>2013.1</Version>

<Vendor>CITRIX</Vendor>

<Locale>en</Locale>

</PlatformSection>

<PropertySection>

<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"
   />

<Property oe:key="com.citrix.netscaler.platform" oe:value="NS1000V
   "/>

<Property oe:key="com.citrix.netscaler.orch_env" oe:value="cisco-
   orch-env"/>

<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="
   10.102.100.122"/>

<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="
   255.255.255.128"/>

<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="
   10.102.100.67"/></PropertySection>

</Environment>
<!--NeedCopy-->
```

2. ISO イメージを Hyper-V Server にコピーします。
3. インポートした仮想アプライアンスを選択し、[アクション] メニューで [設定] を選択します。仮想アプライアンスを選択し、右クリックして [設定] を選択することもできます。選択した仮想アプライアンスの [設定] ウィンドウが表示されます。
4. 「設定」ウィンドウの「ハードウェア」セクションで、「**IDE Controller**」をクリックします。
5. 右側のウィンドウペインで、「**DVD Drive** 」を選択し、「追加」をクリックします。DVD ドライブは、左側のウィンドウペインの **IDE** コントローラセクションに追加されます。
6. 手順 5 で追加した **DVD** ドライブを選択します。右側のウィンドウペインで [イメージファイル] ラジオボタンを選択し、[参照] をクリックして、手順 2 で Hyper-V サーバにコピーした ISO イメージを選択します。
7. ［適用］をクリックします。

> 注
>
> 次の場合、仮想アプライアンスインスタンスはデフォルトの IP アドレスで起動します。
>
> - DVD ドライブがアタッチされているのに、ISO ファイルが提供されていない。
> - ISO ファイルにはユーザーデータファイルは含まれていません。
> - ユーザーデータのファイル名または形式が正しくありません。

NetScaler VPX インスタンスで仮想 NIC を構成するには、次の手順に従います。

1. インポートした仮想アプライアンスを選択し、[アクション] メニューで [設定] を選択します。
2. [設定] ダイアログボックスの左ペインの <virtual appliance name>[ハードウェアの追加] をクリックします。
3. 右ペインで、デバイスのリストから［ネットワークアダプター］を選択します。
4. ［追加］をクリックします。
5. 左ペインに［**Network Adapter** (not connected)］が表示されていることを確認します。
6. 左ペインでネットワークアダプターを選択します。
7. 右側のペインの [ネットワーク] メニューから、アダプターを接続する仮想ネットワークを選択します。
8. 使用する他のネットワークアダプタの仮想ネットワークを選択するには、手順 **6** と **7** を繰り返します。
9. ［適用］をクリックしてから、［**OK**］をクリックします。

NetScaler VPX インスタンスを構成するには:

1. 前にインストールした仮想アプライアンスを右クリックし、［開始］を選択します。
2. 仮想アプライアンスをダブルクリックして、コンソールにアクセスします。
3. 仮想アプライアンスの NetScaler IP アドレス、サブネットマスク、およびゲートウェイを入力します。

仮想アプライアンスの基本構成が完了しました。Web ブラウザーで IP アドレスを入力して、仮想アプライアンスにアクセスします。

> 注
>
> 仮想マシン（VM）テンプレートを使用して、SCVMM を使用して NetScaler ADC VPX インスタンスをプロビジョニングすることもできます。
>
> NetScaler VPX インスタンスで Microsoft Hyper-V NIC チーミングソリューションを使用する場合、詳細については、記事 CTX224494 を参照してください。

Linux-KVM プラットフォームへの NetScaler ADC VPX インスタンスのインストール

August 15, 2023

Linux-KVM プラットフォーム用の NetScaler ADC VPX を設定するには、グラフィカル仮想マシンマネージャ（仮想マネージャ）アプリケーションを使用できます。Linux-KVM コマンドラインを使用する場合は、`virsh` プログラムを使用できます。

KVM Module および QEMU のような仮想化ツールを使って、適切なハードウェアにホスト Linux オペレーティングシステムをインストールする必要があります。ハイパーバイザー上で展開できる仮想マシン（VM）の数はアプリケーション要件および選択されたハードウェアにより異なります。

NetScaler VPX インスタンスをプロビジョニングしたら、より多くのインターフェイスを追加できます。

制限事項と使用ガイドライン

一般的な推奨事項

予測できない動作を回避するには、次の推奨事項を適用します。

- VPX 仮想マシンに関連付けられている VNet インターフェイスの MTU を変更しないでください。インターフェイスモードや CPU などの構成パラメータを変更する前に、VPX VM をシャットダウンします。
- VPX VM を強制的にシャットダウンしないでください。つまり、**Force off** コマンドは使用しないでください。
- ホスト Linux 上で指定された任意の構成は、Linux ディストリビューションの設定によってそのまま維持されたり、維持されなかったりします。これらの構成を維持するよう選択して、ホスト Linux オペレーティングシステムのリブートにおける一貫した動作を確保できます。
- NetScaler パッケージはプロビジョニングされた各 NetScaler VPX インスタンスに対して一意である必要があります。

制限事項

- KVM で実行される VPX インスタンスのライブマイグレーションはサポートされていません。

Linux-KVM プラットフォームに **NetScaler ADC VPX** インスタンスをインストールするための前提条件

August 15, 2023

NetScaler VPX インスタンスで実行されている Linux-KVM サーバーの最小システム要件を確認します。

CPU 要件:

- Intel VT-X プロセッサーに含まれるハードウェア仮想化機能を備えた 64 ビット x86 プロセッサー。

CPU が Linux ホストをサポートしているかどうかをテストするには、ホスト Linux シェルプロンプトで次のコマンドを入力します。

```
\*.egrep '^flags.\*(vmx|svm)' /proc/cpuinfo*
<!--NeedCopy-->
```

前の拡張機能の **BIOS** 設定が無効になっている場合は、BIOS でそれらを有効にする必要があります。

- ホスト Linux に 2 つ以上の CPU コアを指定します。
- プロセッサ速度に対する特定の推奨設定はありませんが、速度が速ければ速いほど VM アプリケーションのパフォーマンスはよくなります。

メモリ **(RAM)** 要件:

ホスト Linux カーネルに対して 4GB 以上。VM が必要とするメモリを追加します。

ハードディスク要件:

ホスト Linux カーネルおよび VM 要件の領域を計算します。単一の NetScaler VPX VM は 20GB のディスク領域を必要とします。

ソフトウェア要件

使用されるホストカーネルは、リリース 2.6.20 以降で、すべての仮想化ツールがある 64 ビットの Linux カーネルである必要があります。3.6.11-4 以降といったより新しいカーネルを推奨します。

Red Hat、CentOS、Fedora などの多くの Linux ディストリビューションでは、カーネルのバージョンと関連する仮想化ツールのテストが行われています。

ゲスト **VM** のハードウェア要件

NetScaler VPX でサポートされるハードディスクの種類は IDE と virtIO です。ハードディスクの種類は、NetScaler パッケージに含まれる XML ファイルで構成されています。

ネットワーク要件

NetScaler VPX は、VirtIO 準仮想化、SR-IOV、および PCI パススルーネットワークインターフェイスをサポートします。

サポートされるネットワークインターフェースの詳細については、以下を参照してください。

- 仮想マシンマネージャーを使用して NetScaler VPX インスタンスをプロビジョニングします
- SR-IOV ネットワークインターフェースを使用するように NetScaler VPX インスタンスを構成する
- PCI パススルーネットワークインターフェイスを使用するように NetScaler VPX インスタンスを構成する

ソースインターフェイスおよびモード

ソースデバイスの種類は、Bridge または MacVTap のいずれかにできます。MacvTap では、VEPA モード、ブリッジ、プライベート、パススルーの 4 つのモードが可能です。次のように、使用できるインターフェイスのタイプとサポートされているトラフィックタイプを確認します。

ブリッジ:

- Linux Bridge。
- 正しい設定を選択したり、`IPtable` サービスを無効にしたりしないと、ホスト Linux の`Ebtables`および`iptables`設定によってブリッジのトラフィックがフィルタリングされることがあります。

MacV タップ **(VEPA** モード**)**:

- パフォーマンスは Bridge より向上します。
- 同じ下位デバイスからのインターフェイスを VM 間で共有できます。
- 仮想マシン間通信（同じ
- 下位のデバイスは、アップストリームスイッチまたはダウンストリームスイッチが VEPA モードをサポートしている場合にのみ可能です。

MacVTap (プライベートモード**)**:

- パフォーマンスは Bridge より向上します。
- 同じ下位デバイスからのインターフェイスを VM 間で共有できます。
- 同じ下位デバイスを使った内部 VM 通信を実行できません。

MacVTap (ブリッジモード):

- Bridge と比べてよい。
- 同じ下位デバイスのインターフェイスは、VM 間で共有できます。
- 下位のデバイスリンクがアップしている場合は、同じ下位デバイスを使用する VM 間通信が可能です。

MacVTap (パススルーモード**)**:

- Bridge と比べてよい。
- 同じ下位デバイスのインターフェイスは、仮想マシン間で共有できません。
- 1 つの VM のみ、下位デバイスを使用できます。

> 注:VPX インスタンスで最適なパフォーマンスを得るには、ソースインターフェイスで`gro`および`lro`機能がオフになっていることを確認してください。

送信元インターフェイスのプロパティ

ソースインターフェイスの Generic-receive-offload（`gro`）および大規模受信オフロード（`lro`）機能をオフにします。`gro`および`lro`機能をオフにするには、ホスト Linux シェルプロンプトで次のコマンドを実行します。


```
ethtool -K eth6 gro off
ethool -K eth6 lro off
```

例:

```
    [root@localhost ~]# ethtool -K eth6

                        Offload parameters for eth6:

                                            rx-checksumming: on

                                            tx-checksumming: on

                        scatter-gather: on

                        tcp-segmentation-offload: on

                        udp-fragmentation-offload: off

                        generic-segmentation-offload: on

                        generic-receive-offload: off

                        large-receive-offload: off

                        rx-vlan-offload: on

                        tx-vlan-offload: on

                        ntuple-filters: off

                        receive-hashing: on

    [root@localhost ~]#
<!--NeedCopy-->
```

例:

次の例のように、ホスト Linux ブリッジをソースデバイスとして使用する場合、ホストとゲスト VM を接続する仮想インターフェイスである VNet インターフェイスで`lro`機能をオフにする必要があります。

```
    [root@localhost ~]# brctl show eth6_br

    bridge name     bridge  id               STP enabled interfaces

    eth6_br         8000.00e0ed1861ae          no         eth6

                                                          vnet0

                                                          vnet2

    [root@localhost ~]#
<!--NeedCopy-->
```


上記の例では、2 つの仮想インターフェイスは eth6_br から派生し、vnet0 および vnet2 として表されます。次のコマンドを実行して、これらのインターフェイスのgro機能とlro機能をオフにします。

```
    ethtool -K vnet0 gro off
            ethtool -K vnet2 gro off
            ethtool -K vnet0 lro off
                    ethtool -K vnet2 lro off
<!--NeedCopy-->
```

無差別モード

次の機能を動作させるには、無差別モードを有効にする必要があります。

- L2 モード
- マルチキャストトラフィック処理
- ブロードキャスト
- IPV6 トラフィック
- 仮想 MAC
- 動的ルーティング

次のコマンドを使用して、無差別モードを有効にします。

```
[root@localhost ~]# ifconfig eth6 promisc
[root@localhost ~]# ifconfig eth6
eth6       Link encap:Ethernet  HWaddr 78:2b:cb:51:54:a3
           inet6 addr: fe80::7a2b:cbff:fe51:54a3/64 Scope:Link
           UP BROADCAST RUNNING PROMISC MULTICAST  MTU:9000  Metric:1
           RX packets:142961 errors:0 dropped:0 overruns:0 frame:0
           TX packets:2895843 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:14330008 (14.3 MB)  TX bytes:1019416071 (1.0 GB)

[root@localhost ~]#
<!--NeedCopy-->
```

必要なモジュール

ネットワークパフォーマンスを向上させるには、Linux ホストに vhost_net モジュールが存在することを確認してください。vhost_net モジュールの存在を確認するには、Linux ホストで次のコマンドを実行します。

```
lsmod | grep "vhost_net"
<!--NeedCopy-->
```

vhost_net がまだ実行されていない場合は、次のコマンドを入力して実行します。

```
modprobe vhost_net
<!--NeedCopy-->
```


OpenStack を使用して NetScaler ADC VPX インスタンスをプロビジョニングする

August 15, 2023

OpenStack 環境で NetScaler ADC VPX インスタンスをプロビジョニングするには、**Nova** ブートコマンド（OpenStack CLI）または Horizon（OpenStack ダッシュボード）を使用します。

VPX インスタンスをプロビジョニングします。オプションで、コンフィグドライブのデータを使用します。「構成ドライブ」とは、インスタンスの起動時に CD-ROM デバイスとしてアタッチされる特殊な構成ドライブを指します。この構成ドライブは、管理 IP アドレス、ネットワークマスク、デフォルトゲートウェイなど、ネットワーク構成を渡すためや、顧客スクリプトを注入するために使用できます。

NetScaler アプライアンスでは、デフォルトの認証方式はパスワードベースです。現在、OpenStack 環境上の NetScaler ADC VPX インスタンスでは、SSH キーペア認証メカニズムがサポートされています。

キーペア (公開鍵と秘密キー) は、公開鍵暗号化メカニズムを使用する前に生成されます。Horizon、Windows 用 Puttygen.exe、Linux 環境用`ssh-keygen` など、さまざまなメカニズムを使用して、キーペアを生成できます。キーペアの生成について詳しくは、それぞれの方式のオンラインドキュメントを参照してください。

キーペアが利用可能になったら、権限のあるユーザーがアクセスできる安全な場所に秘密鍵をコピーします。OpenStack では、Horizon または Nova ブートコマンドを使用して、VPX インスタンスにパブリックキーをデプロイできます。OpenStack を使用して VPX インスタンスをプロビジョニングすると、まず特定の BIOS 文字列を読み取って、インスタンスが OpenStack 環境で起動していることを検出します。この文字列は「OpenStack Foundation」であり、Red Hat Linux ディストリビューションの場合は、/etc/nova/release に保存されます。これは、KVM ハイパーバイザープラットフォームに基づくすべての OpenStack 実装で利用できる標準的なメカニズムです。ドライブには特定の OpenStack ラベルが必要です。

ネットワ ― ク構成、カスタムスクリプト、および SSH キーペアが提供されている場合は、構成ドライブが検出されると、インスタンスがそれらを読み取ろうとします。

ユーザーデータファイル

NetScaler VPX インスタンスは、ユーザーデータファイルとも呼ばれるカスタマイズされた OVF ファイルを使用して、ネットワーク構成、カスタムスクリプトを注入します。このファイルは、構成ドライブの一部として提供されます。次に、カスタマイズされた OVF ファイルの例を示します。

```
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1"
xmlns:cs="http://schemas.citrix.com/openstack">
<PlatformSection>
<Kind></Kind>
```

 
```

```
<Version>2016.1</Version>
<Vendor>VPX</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>
<Property oe:key="com.citrix.netscaler.platform" oe:value="NSVPX"/>
<Property oe:key="com.citrix.netscaler.orch_env" oe:value="openstack-orch-env"/>
<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.2.22"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="255.255.255.0"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.1.2.1"/>
</PropertySection>
 <cs:ScriptSection>
   <cs:Version>1.0</cs:Version>
     <ScriptSettingSection xmlns="http://schemas.citrix.com/openstack" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
      <Scripts>
        <Script>
              <Type>shell</Type>
              <Parameter>X Y</Parameter>
             <Parameter>Z</Parameter>
             <BootScript>before</BootScript>
               <Text>
                     #!/bin/bash
                     echo "Hi, how are you" $1 $2 >> /var/sample.txt
               </Text>
        </Script>
        <Script>
              <Type>python</Type>
              <BootScript>after</BootScript>
               <Text>
                    #!/bin/python
print("Hello");
               </Text>
        </Script>
  <Script>
               <Type>perl</Type>
               <BootScript>before</BootScript>
                <Text>
                    !/usr/bin/perl
my $name = "VPX";
print "Hello, World $name !\n" ;
                </Text>
        </Script>
        <Script>
             <Type>nscli</Type>
             <BootScript>after</BootScript>
             <Text>
              add vlan 33
bind vlan 33 -ifnum 1/2
```



```
59                  </Text>
60              </Script>
61           </Scripts>
62       </ScriptSettingSection>
63    </cs:ScriptSection>
64  </Environment>
65  <!--NeedCopy--> ```
```

前述の OVF ファイルでは、NetScaler のネットワーク構成には「PropertySection」が使用され、すべてのスクリプトを囲むには <cs:ScriptSection> が使用されています。<Scripts></Scripts> タグはすべてのスクリプトをまとめるのに使われます。各スクリプトは <Script></Script> タグの間に定義されます。各スクリプトタグには、従属するフィールドやタグがあります。

a)<Type>: スクリプトタイプの値を指定します。指定可能な値: Shell/Perl/Python/NSLCI（NetScaler CLI スクリプトの場合）

b)<Parameter>: スクリプトにパラメータを提供します。各スクリプトには複数の <Parameter> タグを付けることができます。

c)<BootScript>: スクリプト実行ポイントを指定します。このタグに指定できる値: 前/後。「before」は、PE がアップする前にスクリプトを実行することを指定します。「after」は、PE が起動した後にスクリプトが実行されることを指定します。

d)<Text>: スクリプトの内容を貼り付けます。

> 注
>
> 現在、VPX インスタンスはスクリプトのサニタイズを処理しません。管理者は、スクリプトの有効性を確認する必要があります。
>
> すべてのセクションを表示する必要はありません。空の「PropertySection」を使用して最初のブート時に実行するスクリプトのみを定義するか、ネットワーク構成のみを定義するには空の「PropertySection」を使用します。
>
> OVF ファイル（ユーザーデータファイル）の必要なセクションが入力されたら、そのファイルを使用して VPX インスタンスをプロビジョニングします。

ネットワーク構成

ネットワーク構成の一部として、VPX インスタンスは以下を読み込みます。

- 管理 IP アドレス
- ネットワークマスク
- デフォルトゲートウェイ

パラメーターは、正常に読み取られると、インスタンスをリモートで管理できるように NetScaler 構成に移入されます。パラメーターが読み取られない場合、または構成ドライブが存在しない場合は、インスタンスが以下のデフォルトの処理を実行します。

- DHCP から IP アドレス情報を取得する。
- DHCP で障害が発生するか、タイムアウトした場合、インスタンスはデフォルトのネットワーク設定（192.168.100.1/16）で起動します。

カスタマースクリプト

VPX インスタンスでは、初期プロビジョニング中にカスタムスクリプトを実行できます。アプライアンスは、シェル、Perl、Python、および NetScaler ADC CLI コマンドタイプのスクリプトをサポートしています。

SSH キーペア認証

VPX インスタンスは、構成ドライブ内でインスタンスメタデータの一部として使用できる公開鍵を「authorized_keys」ファイルにコピーします。これにより、ユーザーが秘密キーを使用してインスタンスにアクセスできるようになります。

> 注
>
> SSH キーが提供されると、デフォルトの認証情報（nsroot/nsroot）は機能しなくなります。パスワードベースのアクセスが必要な場合は、それぞれの SSH プライベートキーでログオンし、手動でパスワードを設定します。

はじめに

OpenStack 環境で VPX インスタンスをプロビジョニングする前に、.tgz ファイルから.qcow2ファイルを抽出してビルドします。

qcow2 イメージからの OpenStack イメージ。次の手順を実行します：

1. 次のコマンドを入力して、.tqzファイルから.qcow2ファイルを抽出します。

```
tar xvzf <TAR file>
tar xvzf <NSVPX-KVM-12.0-26.2_nc.tgz>
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
```

2. 次のコマンドを入力して、手順 1 で抽出した.qcoz2ファイルを使用して OpenStack イメージをビルドします。

```
openstack image create --container-format bare --property
    hw_disk_bus=ide --disk-format qcow2 --file <path to qcow2 file>
     --public <name of the OpenStack image>

glance image-create --name="NS-VPX-12-0-26-2" --property
    hw_disk_bus=ide --ispublic=
true --container-format=bare --disk-format=qcow2< NSVPX-KVM
    -12.0-26.2_nc.qcow2
```


図 **1**: 次の図に、glance image-create コマンドの出力例を示します。

```
+------------------+--------------------------------------------------------+
| Field            | Value                                                  |
+------------------+--------------------------------------------------------+
| checksum         | 154ade3fc7dca7d1706b1d03d7d97552                       |
| container_format | bare                                                   |
| created_at       | 2017-03-13T08:52:31Z                                   |
| disk_format      | qcow2                                                  |
| file             | /v2/images/322c1e0f-cce8-4b7b-b53e-bd8152c388ed/file   |
| id               | 322c1e0f-cce8-4b7b-b53e-bd8152c388ed                   |
| min_disk         | 0                                                      |
| min_ram          | 0                                                      |
| name             | VPX-KVM-12.0-26.2                                      |
| owner            | 58d17d81df5d4406afbb4fdab3a58d79                       |
| properties       | hw_disk_bus='ide'                                      |
| protected        | False                                                  |
| schema           | /v2/schemas/image                                      |
| size             | 784338944                                              |
| status           | active                                                 |
| updated_at       | 2017-03-13T08:52:43Z                                   |
| virtual_size     | None                                                   |
| visibility       | public                                                 |
+------------------+--------------------------------------------------------+
```

VPX インスタンスのプロビジョニング

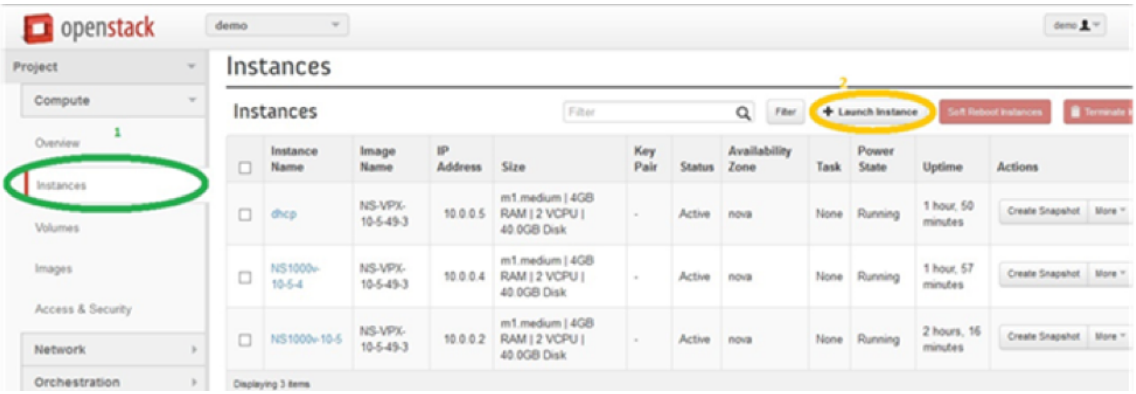
いずれかのオプションを使用して、VPX インスタンスを 2 つの方法でプロビジョニングできます。

- Horizon（OpenStack ダッシュボード）
- Nova boot コマンド（OpenStack CLI）

OpenStack ダッシュボードを使用して **VPX** インスタンスをプロビジョニングします

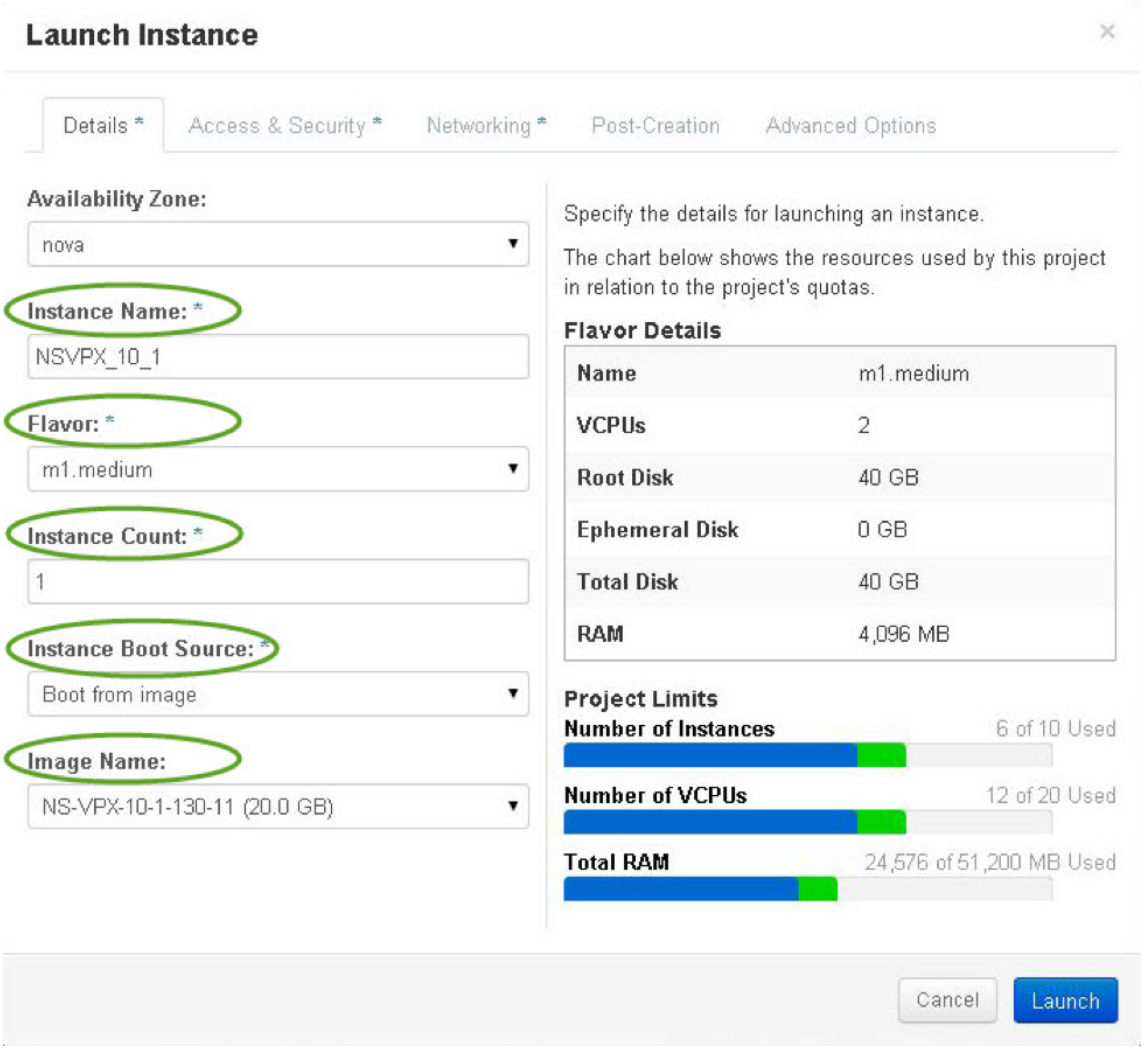
次の手順に従って、Horizon を使用して VPX インスタンスをプロビジョニングします。

1. OpenStack ダッシュボードにログオンします。
2. ダッシュボードの左側にあるプロジェクトパネルで、「インスタンス」を選択します。
3. [インスタンス] パネルで、[インスタンスの起動] をクリックして、[インスタンスの起動] ウィザードを開きます。



4. ［Launch Instance］ウィザードで、以下の情報を指定します。

   a) Instance Name - インスタンス名
   b) Flavor - インスタンスのフレーバー（種類）
   c) Instance Count - インスタンスの数
   d) Instance Boot Source - インスタンスの起動ソース
   e) イメージ名



5. 次の手順を実行して、Horizon を介して新しいキーペアか既存のキーペアを展開します。

   a) 既存のキーペアがない場合は、既存の方式を使用してキーを作成します。既存のキーがある場合は、この手順はスキップします。

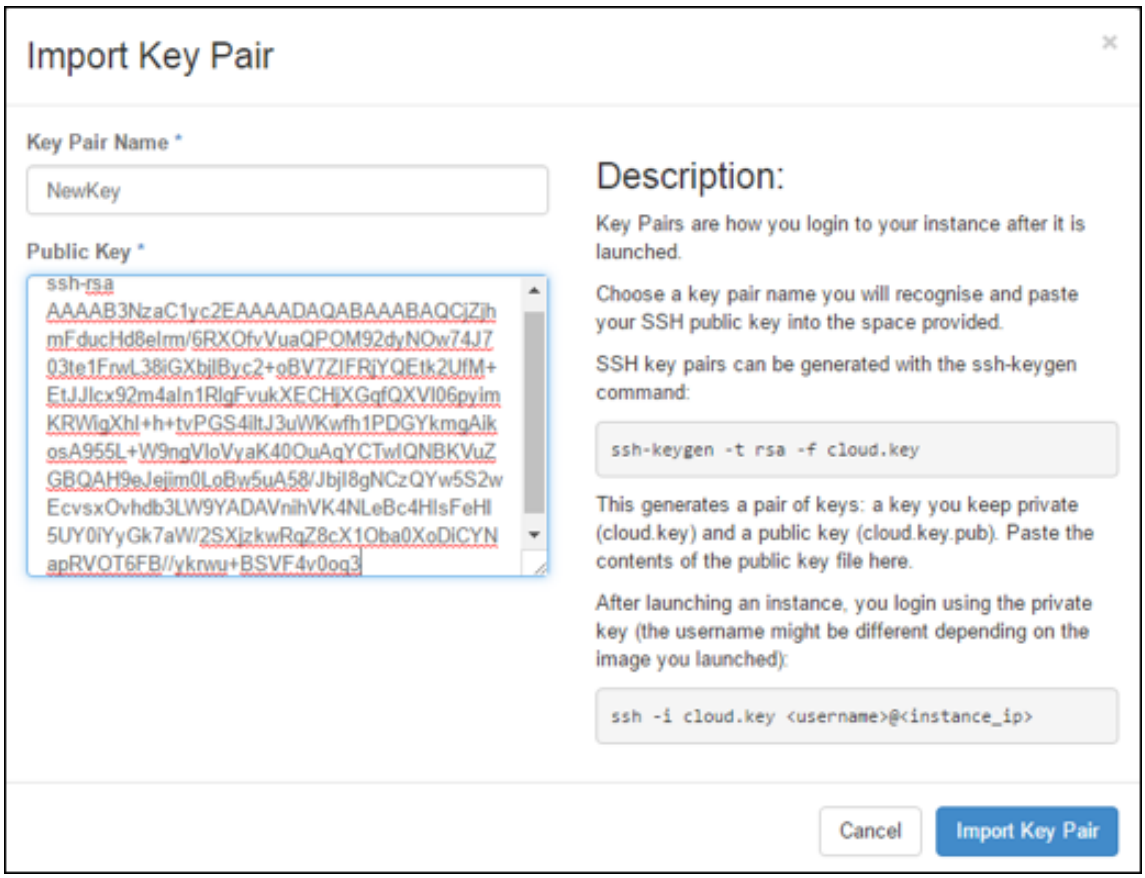
   b) 公開キーの内容をコピーします。

   c) [**Horizon] > [**インスタンス**] > [**新しいインスタンスの作成**]** に移動します。

   d)「アクセスとセキュリティ」をクリックします。

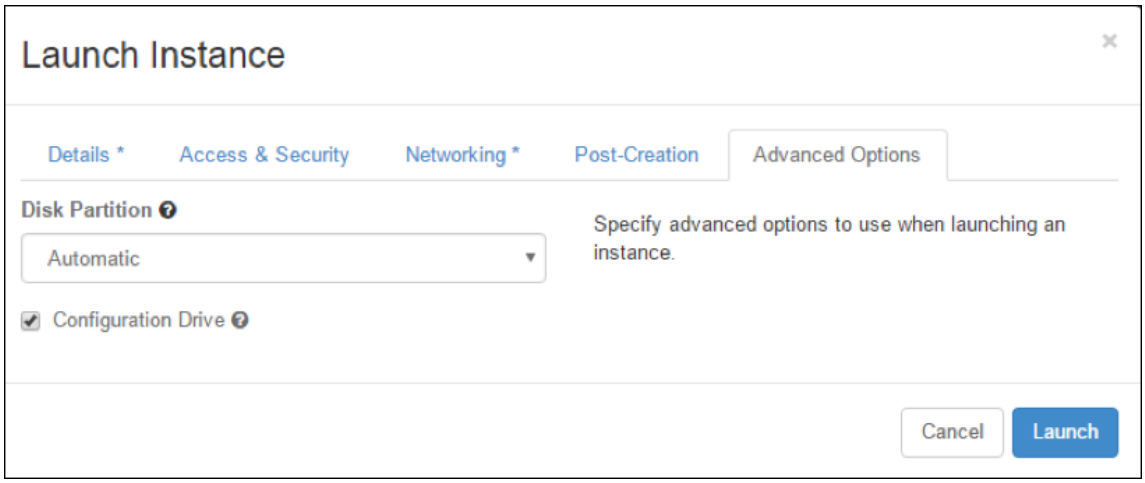
   e) キーペアドロップダウンメニューの横にある **+** 記号をクリックし、表示されているパラメータの値を入力します。

   f）公開鍵の内容を 公開鍵 ボックスに貼り付け、鍵に名前を付け、[鍵 ペアのインポート] をクリックします。



6. ウィザードの [ポスト作成] タブをクリックします。[カスタマイズスクリプト] で、ユーザーデータファイルのコンテンツを追加します。ユーザーデータファイルには、VPX インスタンスの IP アドレス、ネットマスクとゲートウェイの詳細、およびカスタマースクリプトが含まれます。

7. キーペアを選択またはインポートした後、config-drive オプションをチェックし、**Launch** をクリックします。



OpenStack CLI を使用して **VPX** インスタンスをプロビジョニングする

OpenStack CLI を使用して VPX インスタンスをプロビジョニングするには、以下の手順に従ってください。

1. qcow2 からイメージを作成するには、次のコマンドを入力します。

   ```
   openstack image create --container-format bare --property hw_disk_bus
   =ide --diskformat qcow2 --file NSVPX-OpenStack.qcow2 --public VPX
   -ToT-Image
   ```

2. インスタンスを作成するイメージを選択するには、次のコマンドを入力します。

   ```
   openstack image list | more
   ```

3. 特定のフレーバーのインスタンスを作成するには、次のコマンドを入力して、リストからフレーバー ID/名前を選択します。

   ```
   openstack flavor list
   ```

4. NIC を特定のネットワークに接続するには、次のコマンドを入力してネットワークリストからネットワーク ID を選択します。

   ```
   openstack network list
   ```

5. インスタンスを作成するには、次のコマンドを入力します。

   ```
   1 openstack server create --flavor FLAVOR_ID --image IMAGE_ID --key-
       name KEY_NAME
   2 --user-data USER_DATA_FILE_PATH --config-drive True --nic net-id=
       net-uuid
   3 INSTANCE_NAME
   4 openstack server create --image VPX-ToT-Image --flavor m1.medium
       --user-data
   5 ovf.xml --config-drive True --nic net-id=2734911b-ee2b-48d0-a1b6-3
       efd44b761b9
   6 VPX-ToT
   ```

図 **2**: 次の図は、出力例を示しています。

Field	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	
OS-EXT-SRV-ATTR:host	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
OS-EXT-SRV-ATTR:instance_name	instance-000001c2
OS-EXT-STS:power_state	0
OS-EXT-STS:task_state	scheduling
OS-EXT-STS:vm_state	building
OS-SRV-USG:launched_at	None
OS-SRV-USG:terminated_at	None
accessIPv4	
accessIPv6	
addresses	
adminPass	pFVvMtq7N8Z6
config_drive	True
created	2017-03-13T10:32:59Z
flavor	m1.medium (3)
hostId	
id	a1fe991e-3604-43a0-9dd6-59fa0f3749df
image	VPX-ToT-Image (f0c2f9d1-08f2-4b2e-9943-2ee6bc2edbc7)
key_name	None
name	VPX-ToT
os-extended-volumes:volumes_attached	[]
progress	0
project_id	58d17d81df5d4406afbb4fdab3a58d79
properties	
security_groups	[{u'name': u'default'}]
status	BUILD
updated	2017-03-13T10:33:00Z
user_id	a6347b33916b4eb1b1f76360a9c8f935

仮想マシンマネージャーを使用して **NetScaler VPX** インスタンスをプロビジョニングします

August 15, 2023

Virtual Machine Manager は、VM ゲストを管理するためのデスクトップツールです。これによって新しい VM ゲストおよびさまざまな種類のストレージを作成し、仮想ネットワークを管理できます。組み込み VNC ビューアーにより VM ゲストのグラフィカルコンソールにアクセスして、ローカルまたはリモートでパフォーマンス統計を閲覧できます。

優先 Linux ディストリビューションをインストールした後、KVM 仮想化を有効にして、仮想マシンのプロビジョニングを処理できます。

仮想マシンマネージャーを使用して NetScaler VPX インスタンスをプロビジョニングする場合、2 つのオプションがあります。

- 手動で IP アドレス、ゲートウェイ、およびネットマスクを入力する
- IP アドレス、ゲートウェイ、ネットマスクを自動的に割り当てる (自動プロビジョニング)

NetScaler VPX インスタンスのプロビジョニングには、次の 2 種類のイメージを使用できます。

- RAW
- QCOW2

NetScaler VPX RAW イメージを QCOW2 イメージに変換して、NetScaler VPX インスタンスをプロビジョニングできます。RAW イメージを QCOW2 イメージに変換するには、次のコマンドを入力します。

```
qemu-img convert -O qcow2 original-image.raw image-converted.qcow
```

次に例を示します:

```
qemu-img convert -O qcow2 NSVPX-KVM-11.1-12.5_nc.raw NSVPX-KVM-11.1-12.5
_nc.qcow
```

KVM での一般的な NetScaler VPX 展開には、次の手順があります。

- NetScaler VPX インスタンスを自動プロビジョニングするための前提条件の確認
- RAW イメージを使用した NetScaler VPX インスタンスのプロビジョニング
- QCOW2 イメージを使用した NetScaler VPX インスタンスの Provisioning
- Virtual Machine Manager を使用した VPX インスタンスへのインターフェイスの追加

NetScaler VPX インスタンスの自動プロビジョニングの前提条件を確認する

自動プロビジョニングはオプション機能であり、CDROM ドライブからのデータの使用を伴います。この機能が有効になっている場合は、初期セットアップ時に、NetScaler VPX インスタンスの管理 IP アドレス、ネットワークマスク、およびデフォルトゲートウェイを入力する必要があります。

VPX インスタンスを自動プロビジョニングする前に、次のタスクを完了する必要があります。

1. カスタマイズされたオープン仮想化形式 (OVF) XML ファイルまたはユーザーデータファイルを作成します。
2. オンラインアプリケーション（たとえば、PowerISO）を使用して、OVF ファイルを ISO イメージに変換します。
3. セキュアコピー (SCP) ベースのツールを使用して、ISO イメージを KVM ホストにマウントします。

サンプル **OVF XML** ファイル:

次に、OVF XML ファイルの内容の例を示します。このファイルをサンプルとして使用して、ファイルを作成することができます。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<Environment xmlns:oe="`http://schemas.dmtf.org/ovf/environment/1"`

xmlns:xsi="`http://www.w3.org/2001/XMLSchema-instance"`

oe:id=""

xmlns="`http://schemas.dmtf.org/ovf/environment/1"`

xmlns:cs="`http://schemas.citrix.com/openstack">`

<PlatformSection>

```



```
<Kind></Kind>

<Version>2016.1</Version>

<Vendor>VPX</Vendor>

<Locale>en</Locale>

</PlatformSection>

<PropertySection>

<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>

<Property oe:key="com.citrix.netscaler.platform" oe:value="NSVPX"/>

<Property oe:key="com.citrix.netscaler.orch_env" oe:value="KVM"/>

<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.2.22"/>

<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="
   255.255.255.0"/>

<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.1.2.1
   "/>

</PropertySection>

</Environment>
<!--NeedCopy-->
```

前述の OVF XML ファイルでは、NetScaler ネットワーク構成に「PropertySection」が使用されています。ファイルを作成するときには、この例の最後で強調表示されている、パラメーターの値を指定します。

- 管理 IP アドレス
- ネットマスク
- Gateway

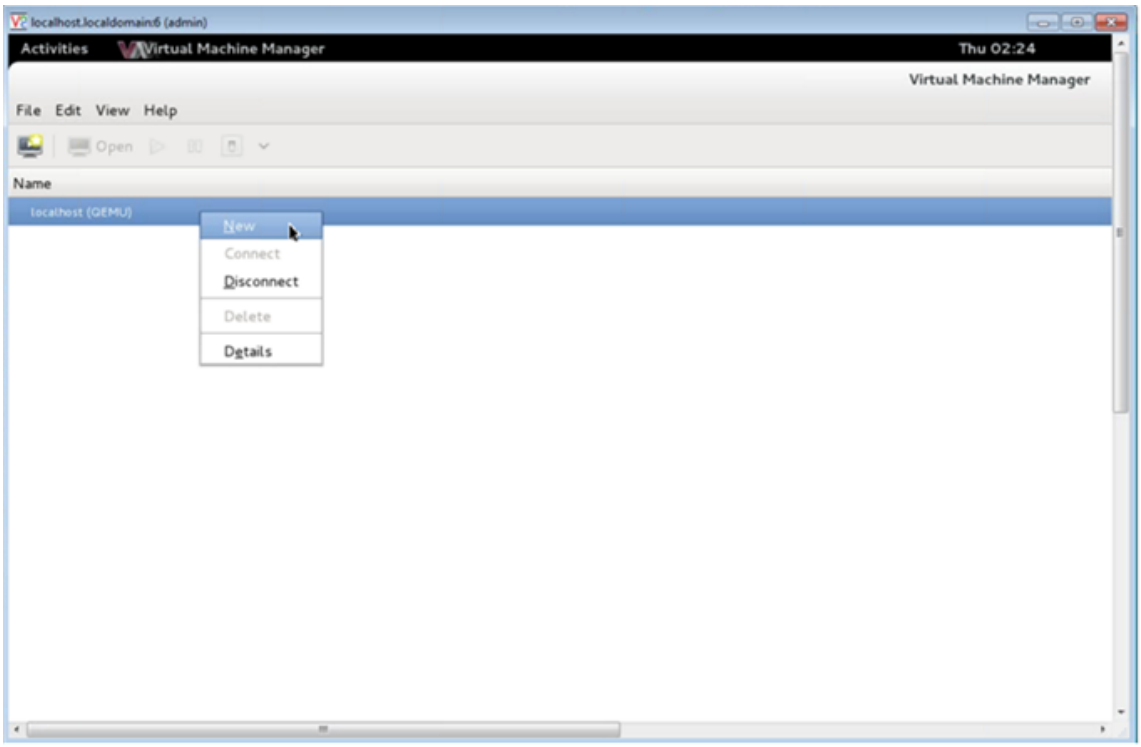
> 重要
>
> OVF ファイルが適切に XML 形式になっていない場合、VPX インスタンスにはファイルに指定されている値ではなく、デフォルトのネットワーク構成が割り当てられます。

RAW イメージを使用して NetScaler VPX インスタンスをプロビジョニングします

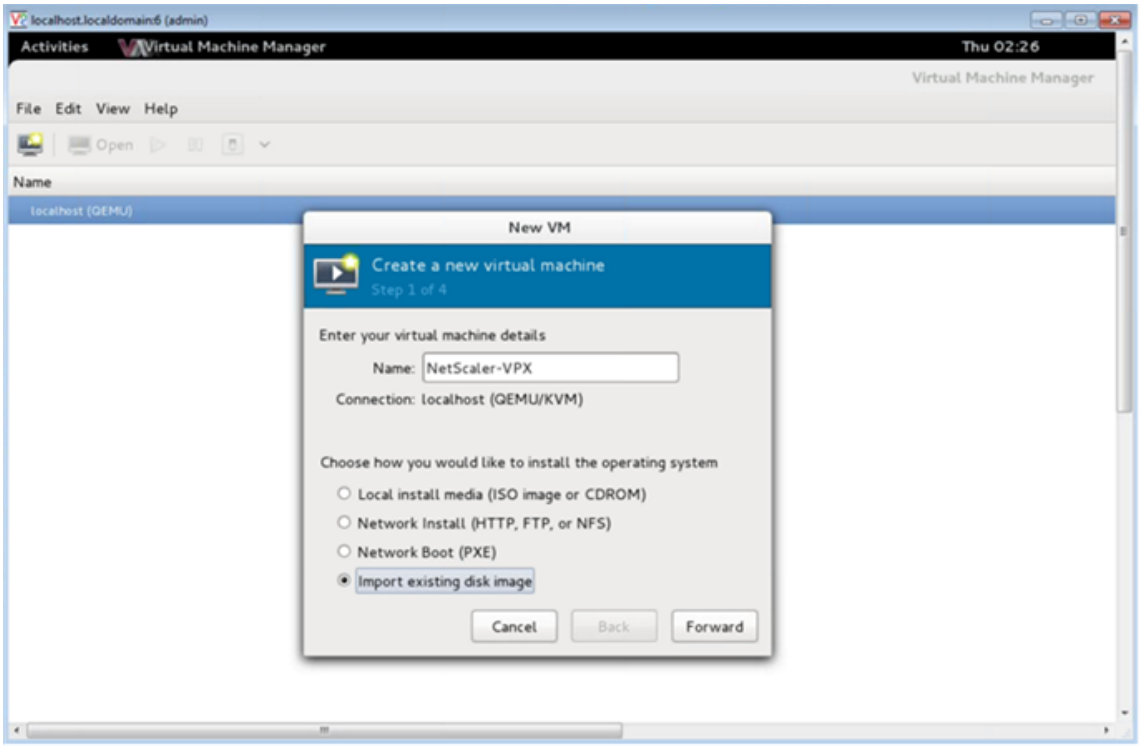
Virtual Machine Manager では、RAW イメージを使用して NetScaler VPX インスタンスをプロビジョニングできます。

仮想マシンマネージャーを使用して NetScaler VPX インスタンスをプロビジョニングするには、次の手順に従います。

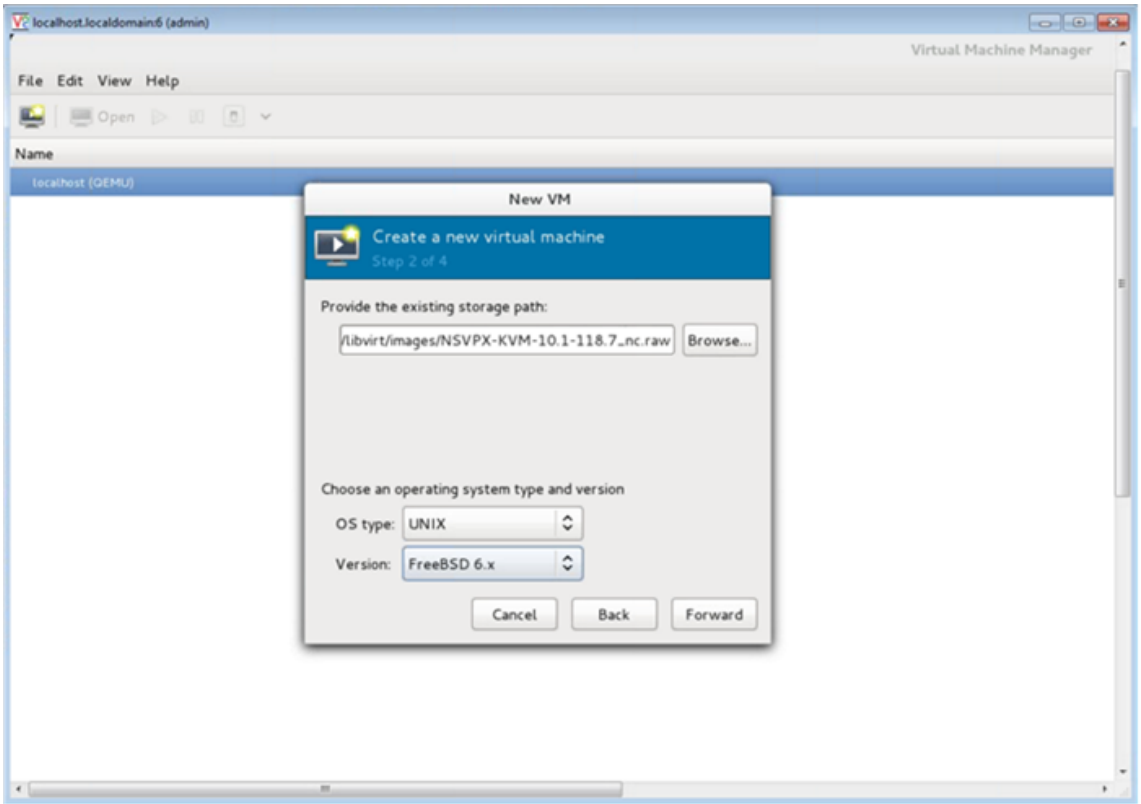
1. 仮想マシンマネージャー (アプリケーション > システムツール > バーチャルマシンマネージャー) を開き、[認証] ウィンドウにログオン資格情報を入力します。
2.  、ローカルホスト（**QEMU**）を右クリックして、新しい NetScaler ADC VPX インスタンスを作成します。



3. 名前テキストボックスに、新しい仮想マシンの名前 (たとえば、NetScaler-VPX) を入力します。
4. ［新規 **VM** ］ウィンドウの［オペレーティングシステムのインストール方法を選択］で［既存のディスクイメージをインポートする］を選択し、［転送］をクリックします。

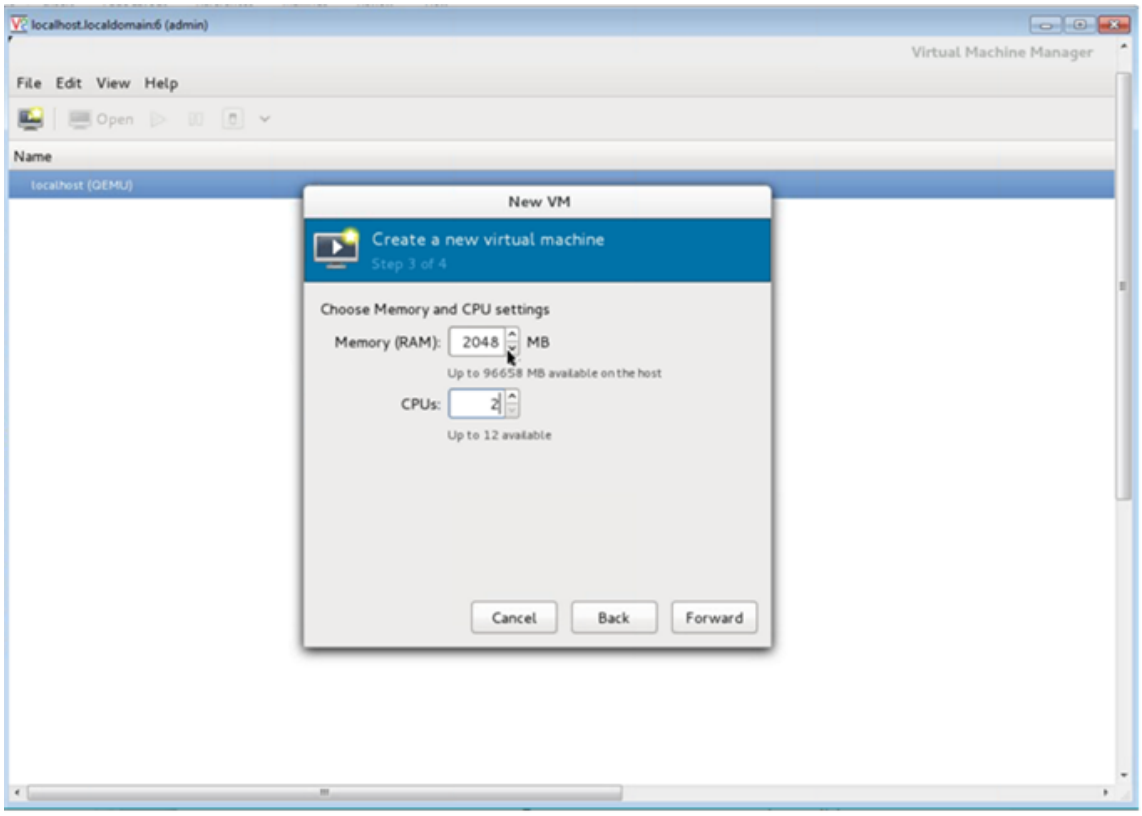


5. 「既存のストレージパスを指定」フィールドで、画像へのパスをナビゲートします。オペレーティングシステム種類に UNIX、バージョンとして FreeBSD 6.x を選択します。次に、「進む」をクリックします。

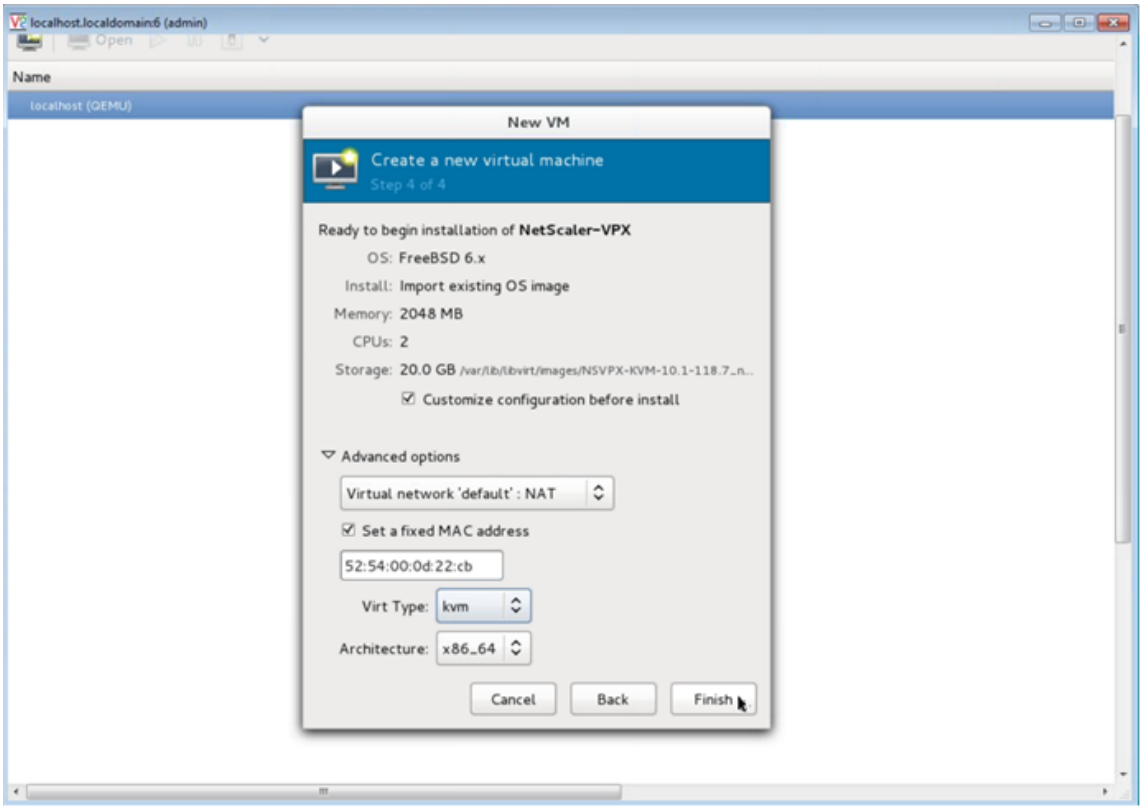


6. 「メモリと **CPU** の設定を選択」で次の設定を選択し、「転送」をクリックします。

   - メモリ (RAM) –2048MB
   - CPU –2

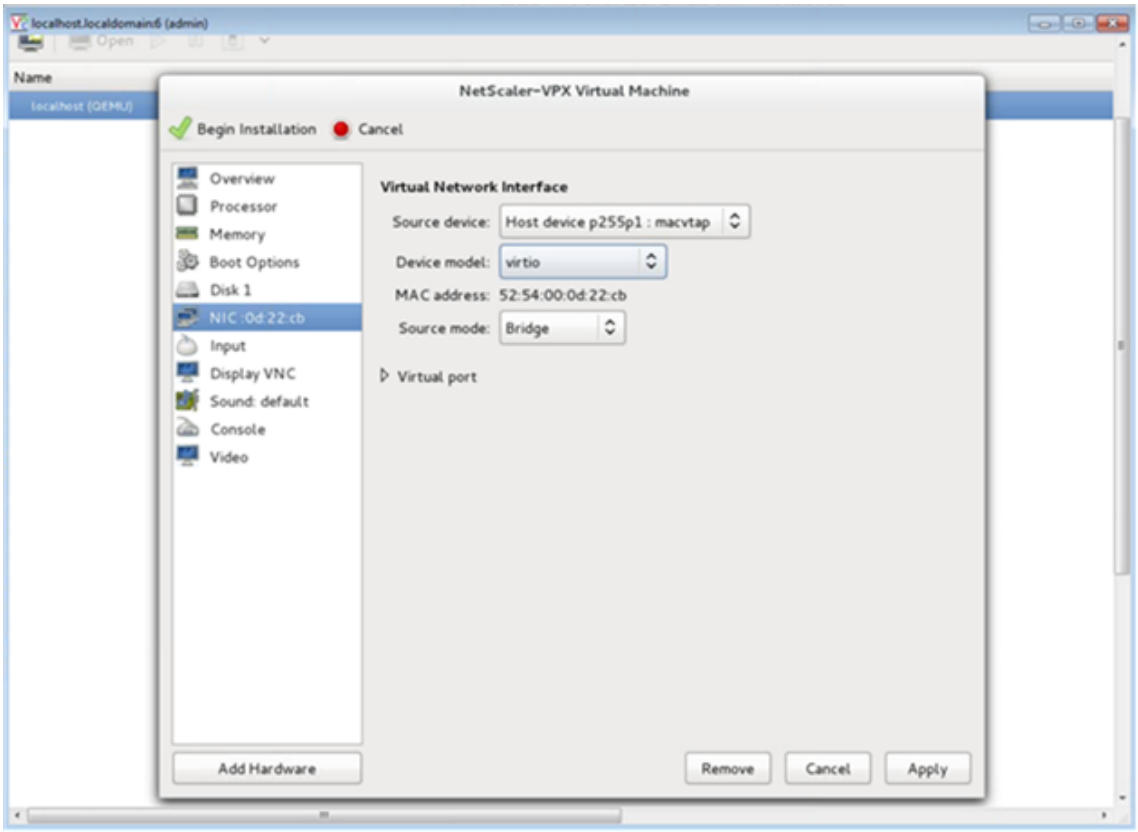


7. [インストール前に構成をカスタマイズする] チェックボックスをオンにします。オプションで、[詳細オプション**]** で MAC アドレスをカスタマイズできます。選択した **Virt** タイプが KVM で、選択されたアーキテクチャが x86_64 であることを確認します。[完了] をクリックします。



8. NIC を選択し、次の構成を指定します。

   - ソースデバイス：`ethX macvtap` またはブリッジ
   - デバイスモデル—`virtio`
   - ソースモード - Bridge



9. [適用] をクリックします。
10. VPX インスタンスを自動プロビジョニングする場合は、このドキュメントの「**CDROM** ドライブを接続して自動 **Provisioning** を有効にする」セクションを参照してください。それ以外の場合は、[インストレーションを開始**]** をクリックします。KVM で NetScaler ADC VPX をプロビジョニングしたら、インターフェイスを追加できます。

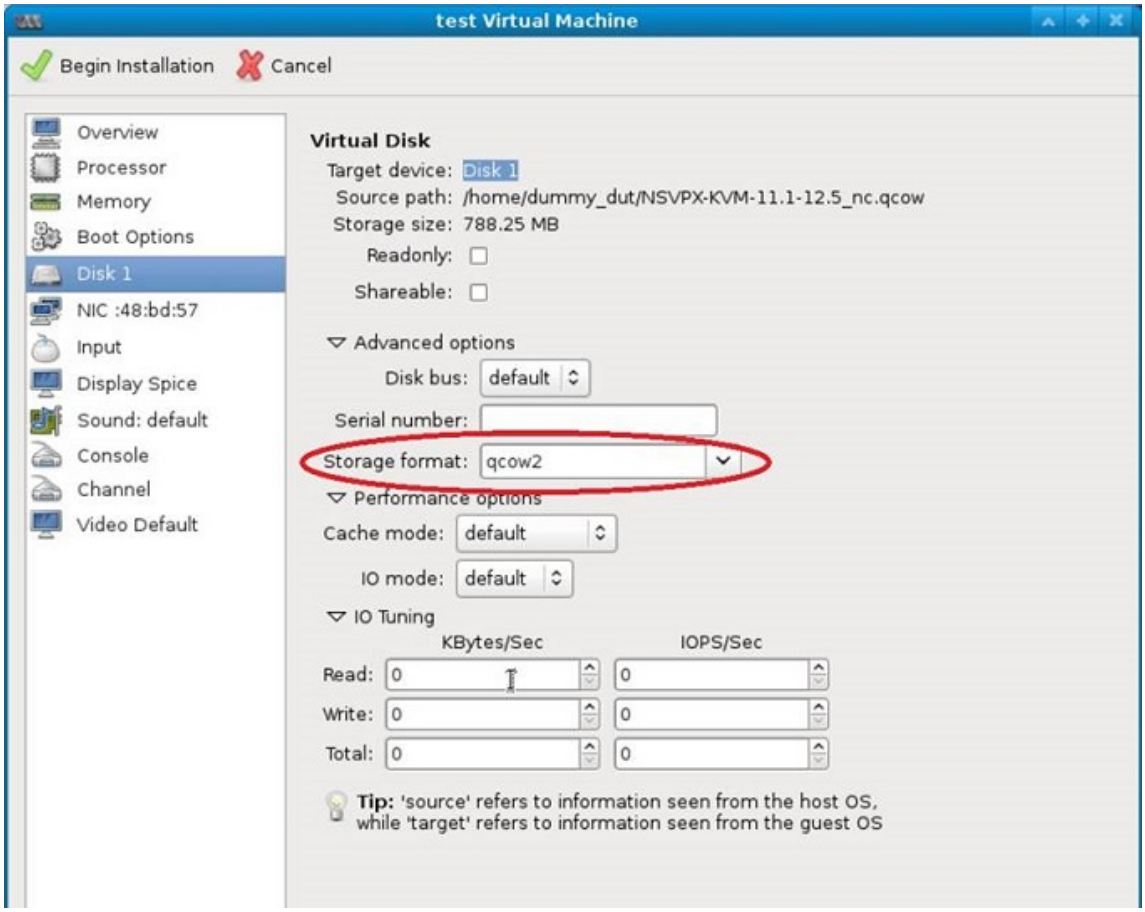
QCOW2 イメージを使用して **NetScaler ADC VPX** インスタンスをプロビジョニングする

仮想マシンマネージャーを使用すると、QCOW2 イメージを使用して NetScaler VPX インスタンスをプロビジョニングできます。

QCOW2 イメージを使用して NetScaler ADC VPX インスタンスをプロビジョニングするには、次の手順に従います。

1. RAW イメージを使用した NetScaler ADC ****VPX** インスタンスのプロビジョニングの手順 **1**～ステップ **8** に従います **。

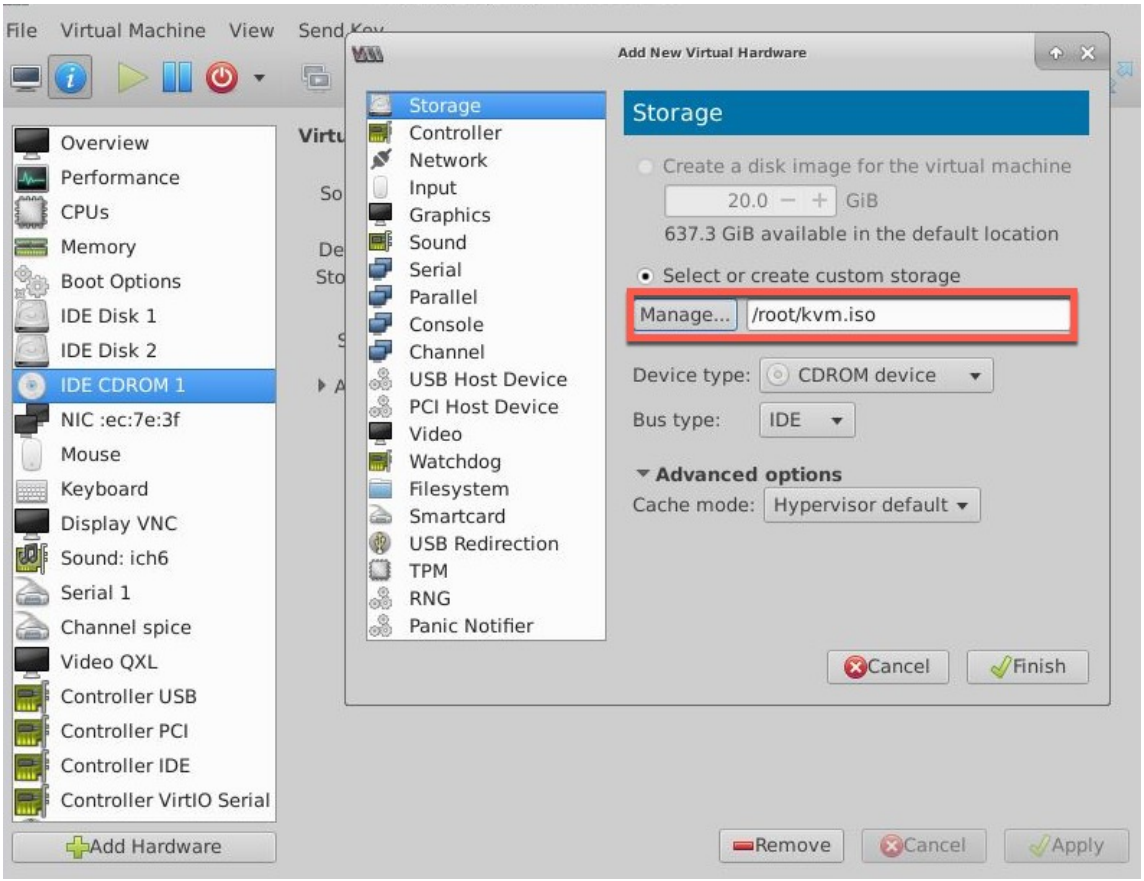
   注：ステップ **5** で **qcow2** イメージを選択することを確認してください。
2. **Disk 1** を選択し、[詳細オプション] をクリックします。
3. [ストレージ形式] ドロップダウンリストから [**qcow2**] を選択します。



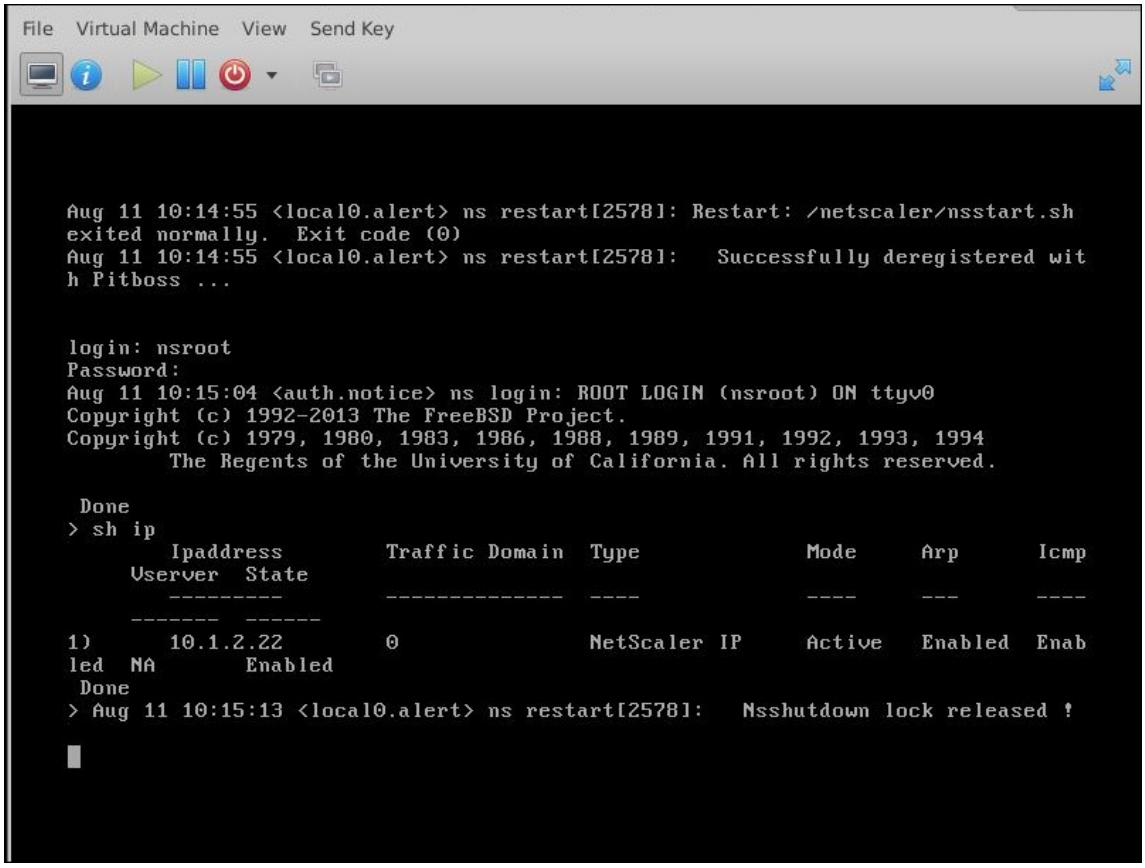
4. [**Apply**] をクリックし、次に [**Begin Installation**] をクリックします。KVM で NetScaler ADC VPX をプロビジョニングしたら、インターフェイスを追加できます。

CD-ROM ドライブを接続して自動プロビジョニングを有効にする

1. [ハードウェアの追加] **>** [記憶域] **>** [デバイスの種類] **>** [**CD-ROM** デバイス] をクリックします。
2. [管理] をクリックし、[NetScaler VPX インスタンスの自動プロビジョニングの前提条件] セクションでマウントした正しい ISO ファイルを選択し、[完了] をクリックします。NetScaler VPX インスタンスの [Resources] の下に新しい CDROM が作成されます。



3. VPX インスタンスの電源をオンにすると、スクリーンショットの例で示すように、OVF ファイルで提供されているネットワーク構成を使用して自動プロビジョニングが行われます。



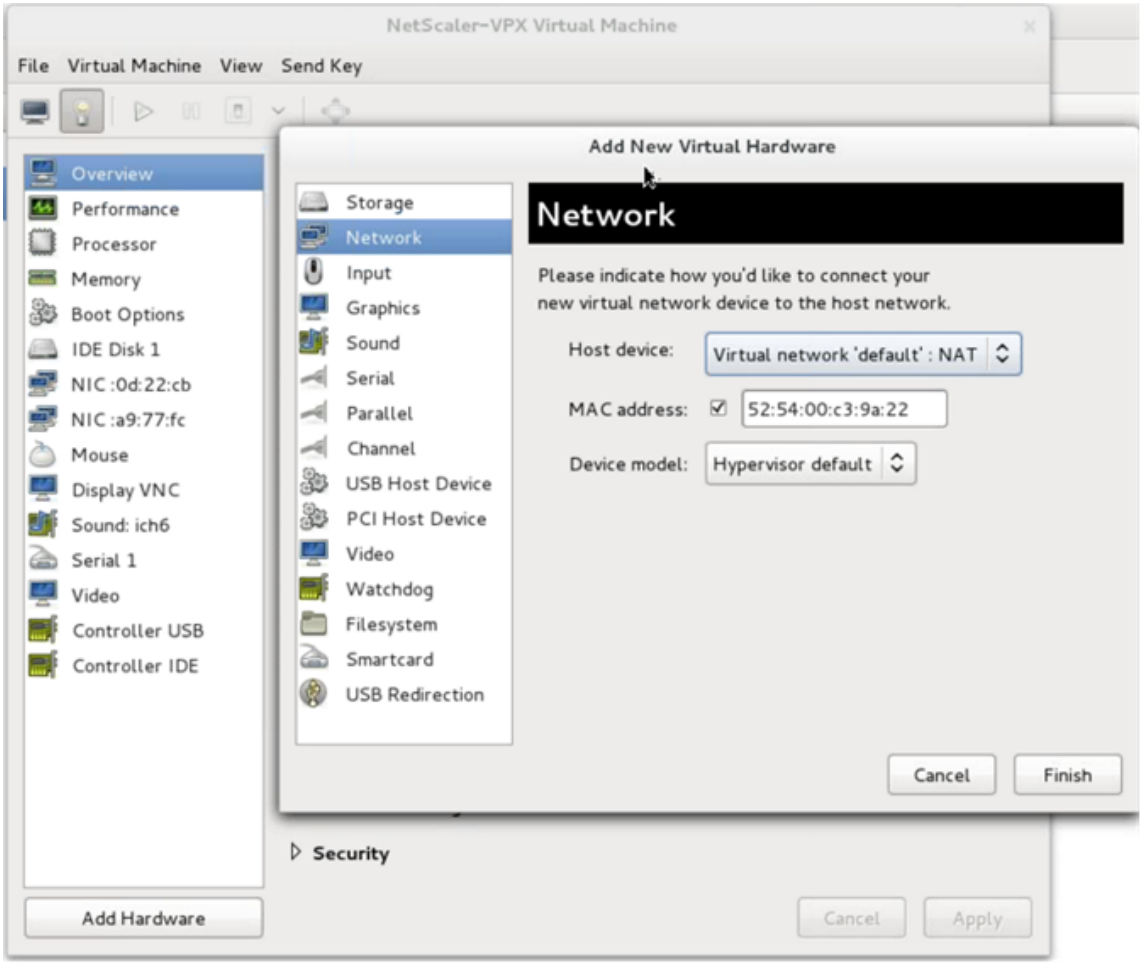
4. 自動プロビジョニングが失敗した場合、インスタンスはデフォルトの IP アドレス（192.168.100.1）で起動します。その場合は、初期設定を手動で完了する必要があります。詳細については、「ADC を初めて構成する」を参照してください。

仮想マシンマネージャーを使用して、**NetScaler VPX** インスタンスにインターフェイスを追加する

KVM で NetScaler VPX インスタンスをプロビジョニングしたら、インターフェイスを追加できます。

インターフェイスを追加するには、次の手順を実行します。

1. KVM の上で動作している NetScaler VPX インスタンスをシャットダウンします。
2. VPX インスタンスを右クリックし、ポップアップメニューから［**Open**］を選択します。
3.  、仮想ハードウェアの詳細を表示します。
4. [ハードウェアの追加] をクリックします。［**Add New Virtual Hardware**］ウィンドウで、ナビゲーションメニューから［**Network**］を選択します。



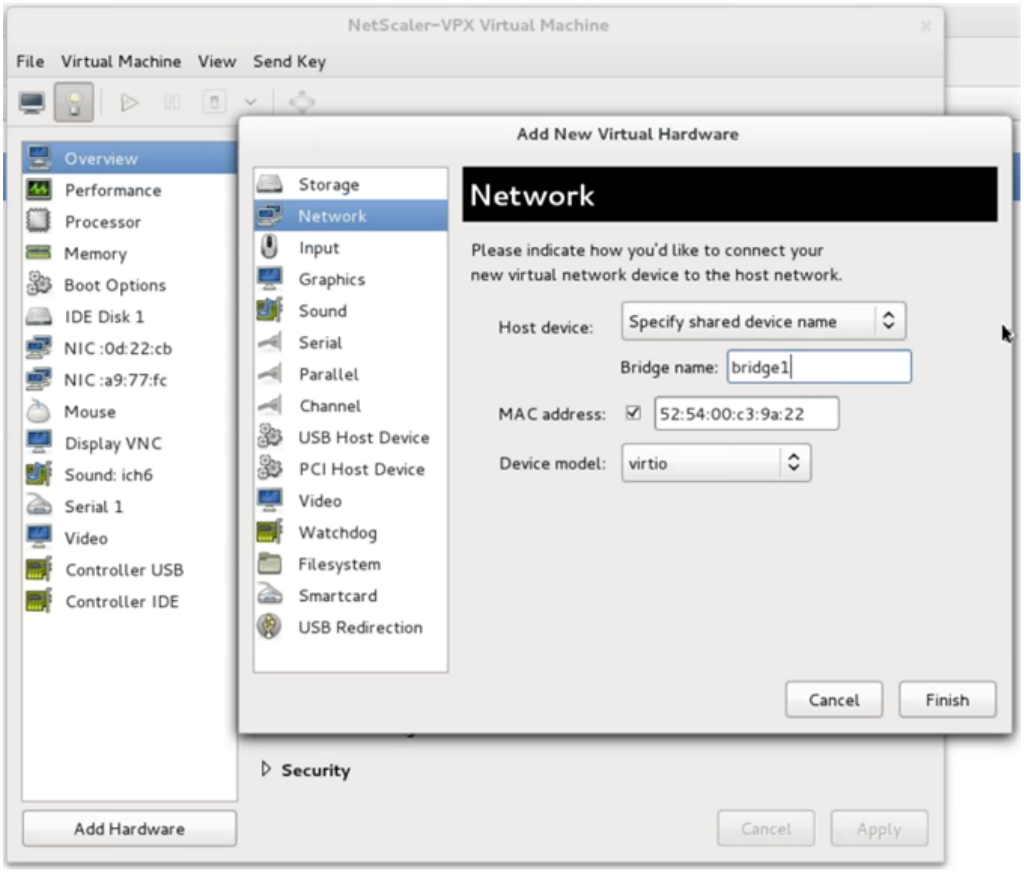
5. [**Host Device**] フィールドで、物理インターフェイスの種類を選択します。ホストデバイスの種類は、Bridge または MacVTap のいずれかにできます。macvTap の場合、VEPA モード、ブリッジ、プライベート、パススルーの 4 つのモードが可能です。

   a) Bridge の場合

      i. Host device - [Specify shared device name] オプションを選択します。

      ii. KVM ホストで構成される Bridge 名を指定します。

      注：KVM ホストで Linux Bridge が構成され、Bridge に物理インターフェイスが結合されて、Bridge が UP 状態になっている必要があります。



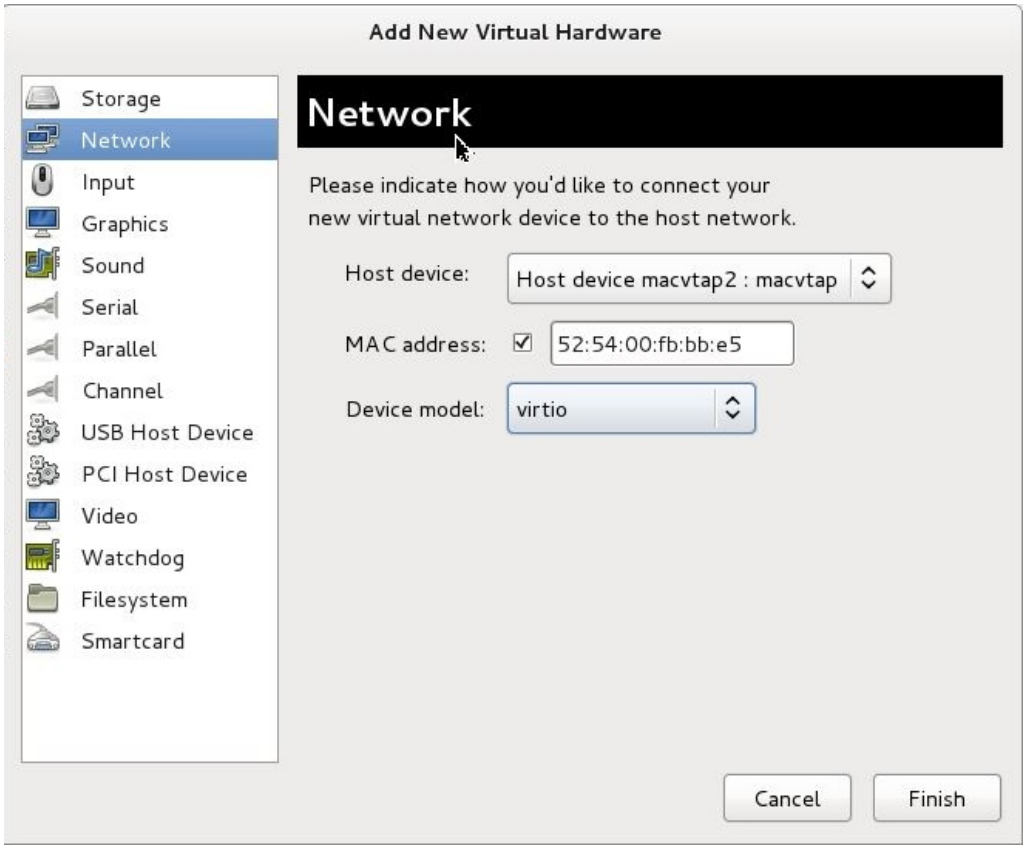
iii. デバイスモデル—`virtio`。

iv. [完了] をクリックします。

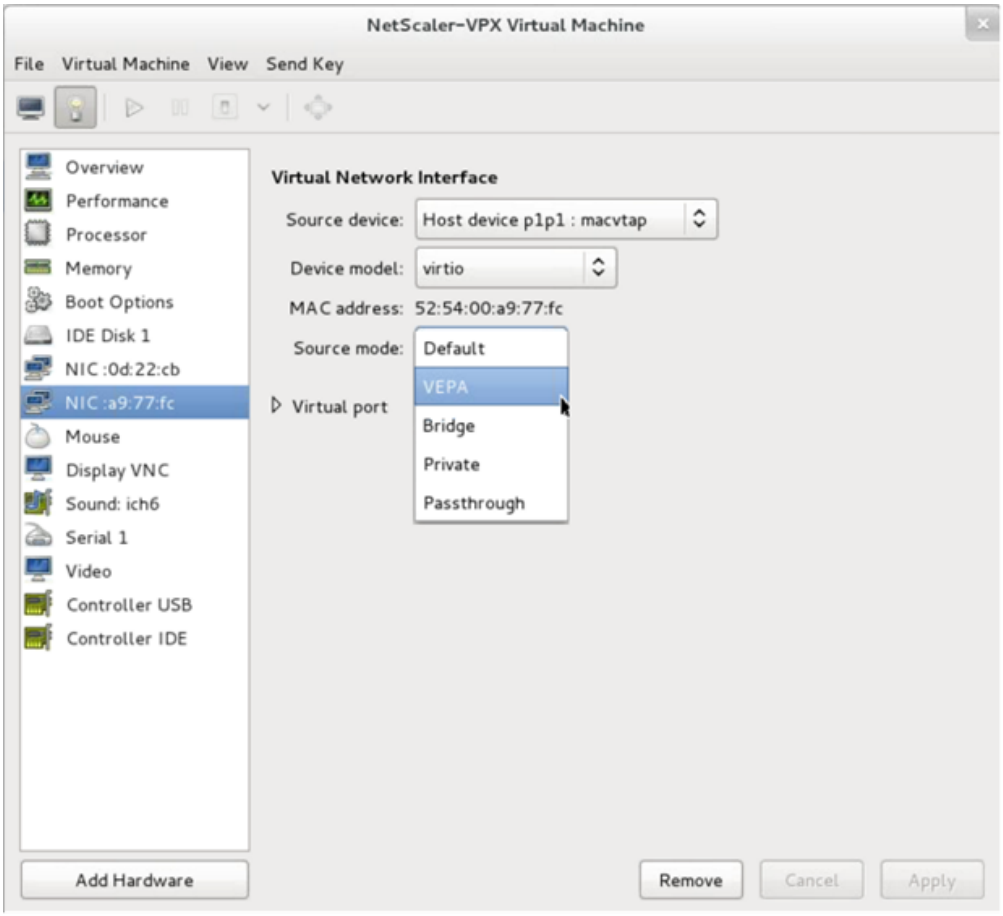
b) MacVTap 用

i. Host device - メニューからの物理インターフェイス

ii. デバイスモデル—`virtio`。



iii. ［完了］をクリックします。ナビゲーションペインで新しく追加された NIC を見ることができます。



iv. 新しく追加された NIC を選択して、この NIC の Source モードを選択します。利用可能なモードは VEPA、Bridge、Private、および Passthrough です。インターフェイスとモードについて詳しくは、「ソースインターフェイスおよびモード」を参照してください

v. [適用] をクリックします。

6. VPX インスタンスを自動プロビジョニングする場合は、このドキュメントの「自動プロビジョニングを有効にするための構成ドライブの追加」セクションを参照してください。それ以外の場合は、VPX インスタンスをパワーオンして初期構成を手動で完了します。

> 重要
>
> スピード、デュプレックス、オートネゴシエーションなどの Interface パラメーター構成はサポートされません。

SR-IOV ネットワークインターフェースを使用するように NetScaler VPX インスタンスを構成する

August 15, 2023

Linux-KVM プラットフォームで実行される NetScaler VPX インスタンスは、次の NIC でシングルルート I/O 仮想化（SR-IOV）を使用して構成できます。

- Intel 82599 10G
- Intel X710 10G
- Intel XL710 40G
- Intel X722 10G

ここでは、次の操作の方法について説明します。

- SR-IOV ネットワークインターフェイスを使用するように NetScaler VPX インスタンスを構成する
- SR-IOV インターフェイスで静的 LA/LACP を構成する
- SR-IOV インターフェイスで VLAN を構成する

制限事項

Intel 82599、X710、XL710、および X722 NIC を使用する際は、制限事項に留意してください。次の機能はサポートされません。

Intel 82599 NIC の制限事項:

- L2 モード切り替え
- 管理パーティション化（共有 VLAN モード）
- 高可用性（アクティブ/アクティブモード）
- ジャンボフレーム。
- IPv6: SR-IOV インターフェイスが 1 つ以上ある場合は、VPX インスタンスで最大 30 個までの一意の IPv6 アドレスのみを設定できます。
- `ip link`コマンドによる SRIOV VF インターフェイスのハイパーバイザでの VLAN 設定はサポートされていません。
- スピート、デュプレックス、オートネゴシエーションなどの Interface パラメーター構成はサポートされません。

Intel X710 10G、**Intel XL710 40G**、および **Intel X722 10G NIC** の制限事項:

- L2 モード切り替え
- 管理パーティション化（共有 VLAN モード）

- クラスタでは、XL710 NIC がデータインターフェイスとして使用されている場合、ジャンボフレームはサポートされません。
- インターフェイスが切断され、再接続されると、インターフェイスリストの順序が変わります。
- スピート、デュプレックス、オートネゴシエーションなどの Interface パラメーター構成はサポートされません。
- Intel X710 10G、Intel XL710 40G、および Intel X722 10G NIC のインターフェイス名は 40/X
- VPX インスタンスでは、最大 16 の Intel XL710/X722 SRIOV または PCI パススルーインターフェイスをサポートできます。

> 注**:** Intel X710 10G、Intel XL710 40G、および Intel X722 10G NIC が IPv6 をサポートするには、KVM ホストで次のコマンドを入力して、仮想機能 (VF) のトラストモードを有効にする必要があります。
>
> ```
> # ip link set <PNIC> <VF> trust on
> ```
>
> 例:
>
> ```
> # ip link set ens785f1 vf 0 trust on
> ```

前提条件

SR-IOV ネットワークインターフェイスを使用するように NetScaler VPX インスタンスを構成する前に、次の前提条件となるタスクを完了してください。対応するタスクを完了する方法の詳細については、「NIC」列を参照してください。

タスク	Intel 82599 NIC	Intel X710、XL710、X722 NIC
1. NIC を KVM ホストに追加します。	-	-
1. 最新の Intel ドライバーをダウンロードしてインストールします。	IXGBE ドライバー	I40E ドライバー
1. KVM ホスト上のドライバをブロックリストします。	/etc/modprobe.d/blacklist.conf ファイルに次のエントリを追加します。`blacklist ixgbevf`。IXGBE ドライバーのバージョン 4.3.15 を使用します（推奨）。	/etc/modprobe.d/blacklist.conf ファイルに次のエントリを追加します。`blacklist i40evf`。i40e ドライバーのバージョン 2.0.26 を使用します（推奨）。

タスク	Intel 82599 NIC	Intel X710、XL710、X722 NIC
1. KVM ホストで SR-IOV 仮想機能 (VF) を有効にします。次の 2 つの列の両方のコマンドで、number_of_VFs = 作成する仮想 VF の数。device_name = インターフェイス名です。	以前のバージョンのカーネル 3.8 を使用している場合は、次のエントリを /etc/modprobe.d/ixgbe ファイルに追加し、KVM ホストを再起動します。options ixgbe max_vfs =<number_of_VFs>。カーネル 3.8 以降を使用している場合は、次のコマンドを使用して VF を作成します。echo <number_of_VFs> > /sys/class/net/<device_name>/device/sriov_numvfs。図 1 の例を参照してください。	以前のバージョンのカーネル 3.8 を使用している場合は、/etc/modprobe.d/i40e.conf ファイルに次のエントリを追加し、KVM ホストを再起動します。options i40e max_vfs =<number_of_VFs> カーネル 3.8 以降を使用している場合は、次のコマンドを使用して VF を作成します。echo<number_of_VFs > > /sys/class/net/<device_name>/device/sriov_numvfs。図 2 の例を参照してください。
1. rc.local ファイルに VF を作成するために使用したコマンドを追加して、VF を永続的にします。	図 3 の例を参照してください。	図 3 の例を参照してください。

> 重要
>
> SR-IOV VF を作成するときは、VF に MAC アドレスを割り当てないようにしてください。

図 **1**: Intel 82599 10G NIC の KVM ホストで SR-IOV VF を有効にしてください。

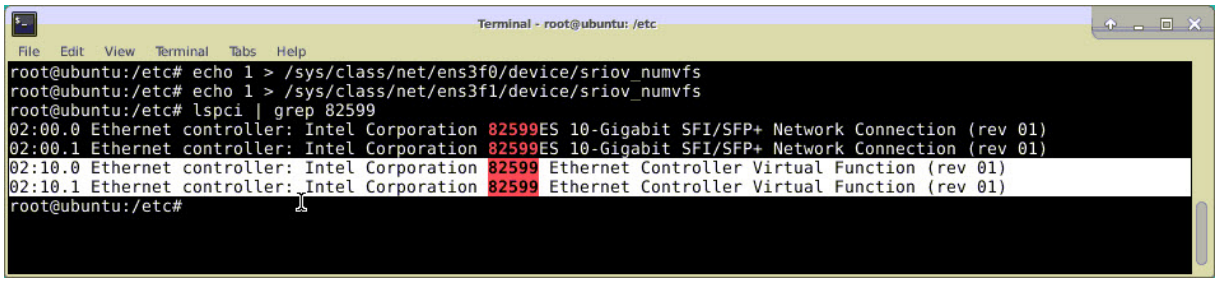


図 **2:** Intel X710 10G および XL710 40G NIC の KVM ホストで SR-IOV VF を有効にしてください。

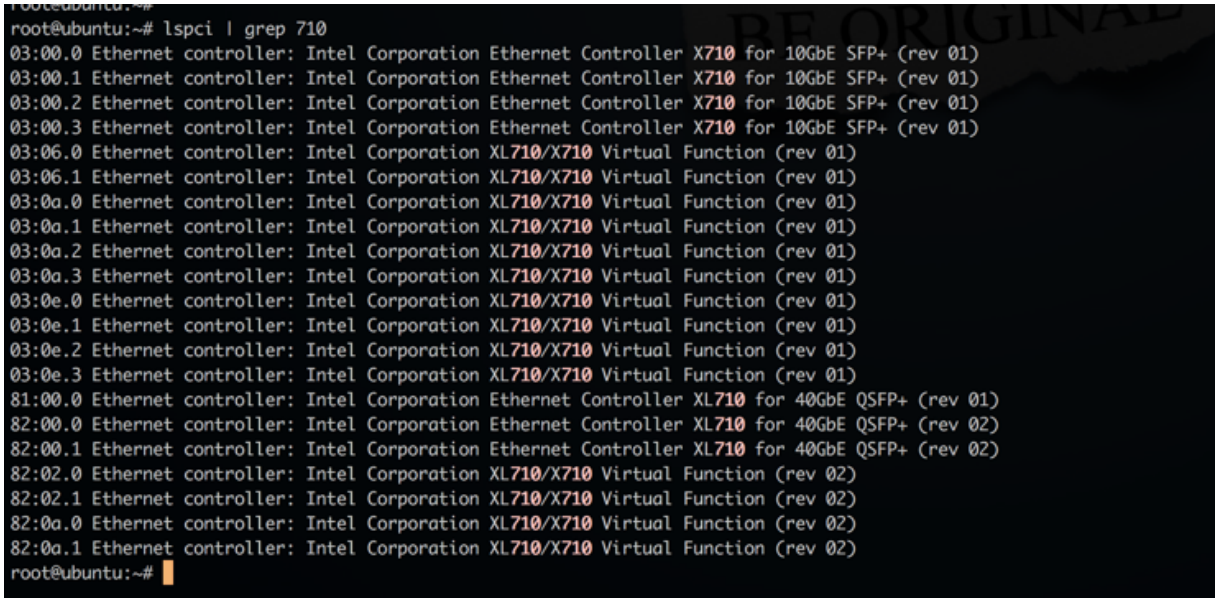


図 **3: Intel X722** 10G NIC の KVM ホストで SR-IOV VF を有効にしてください。

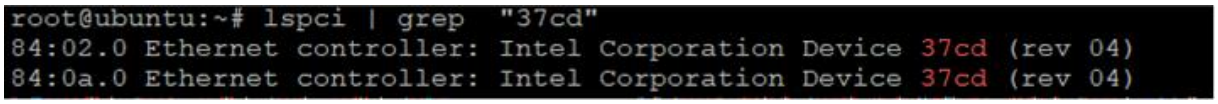
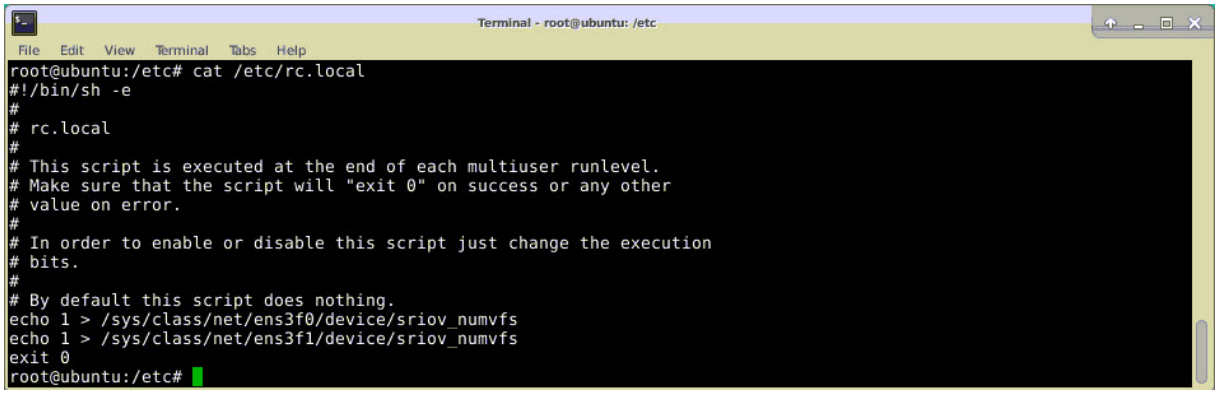


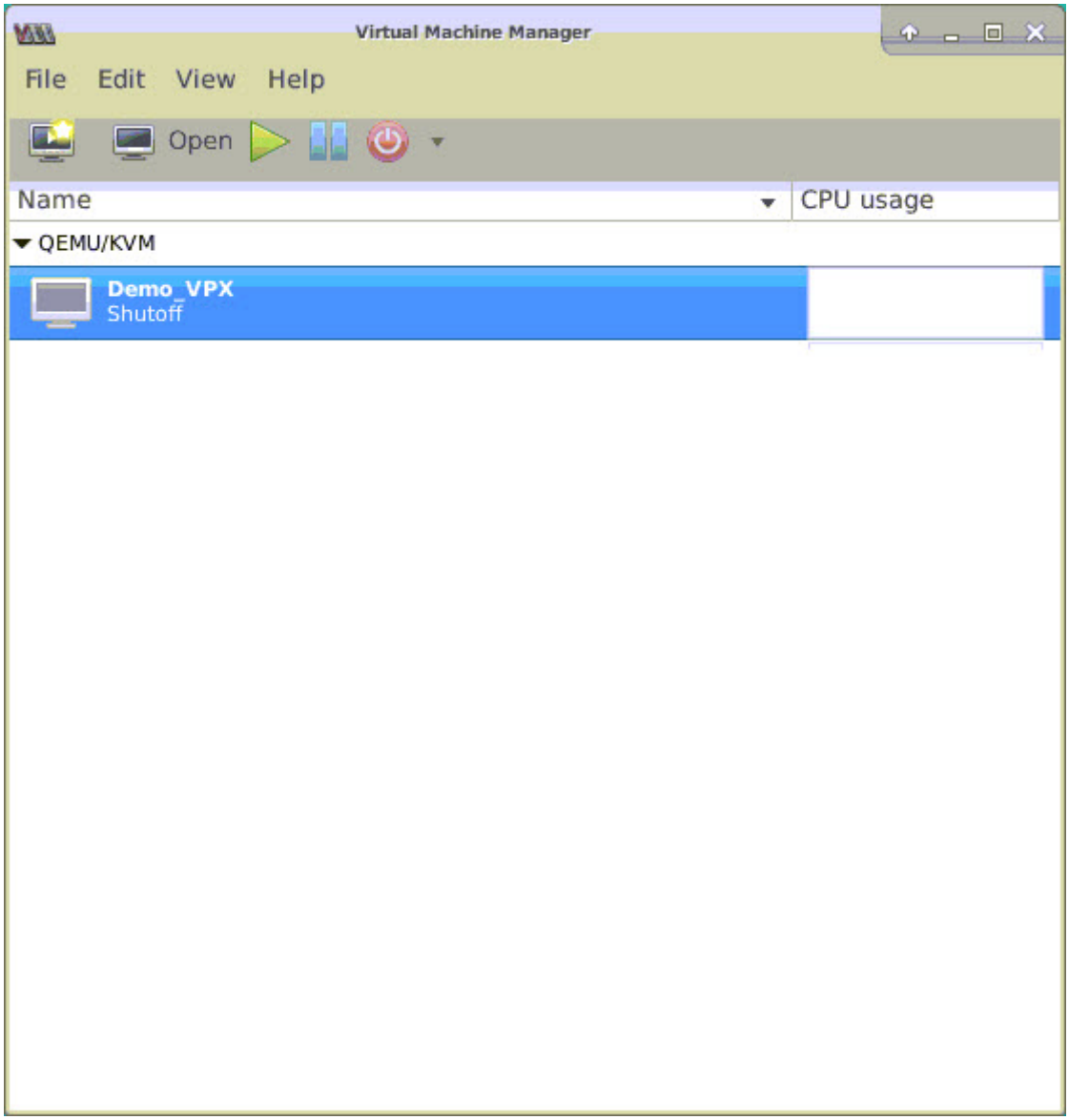
図 **4**: VF を永続的にする



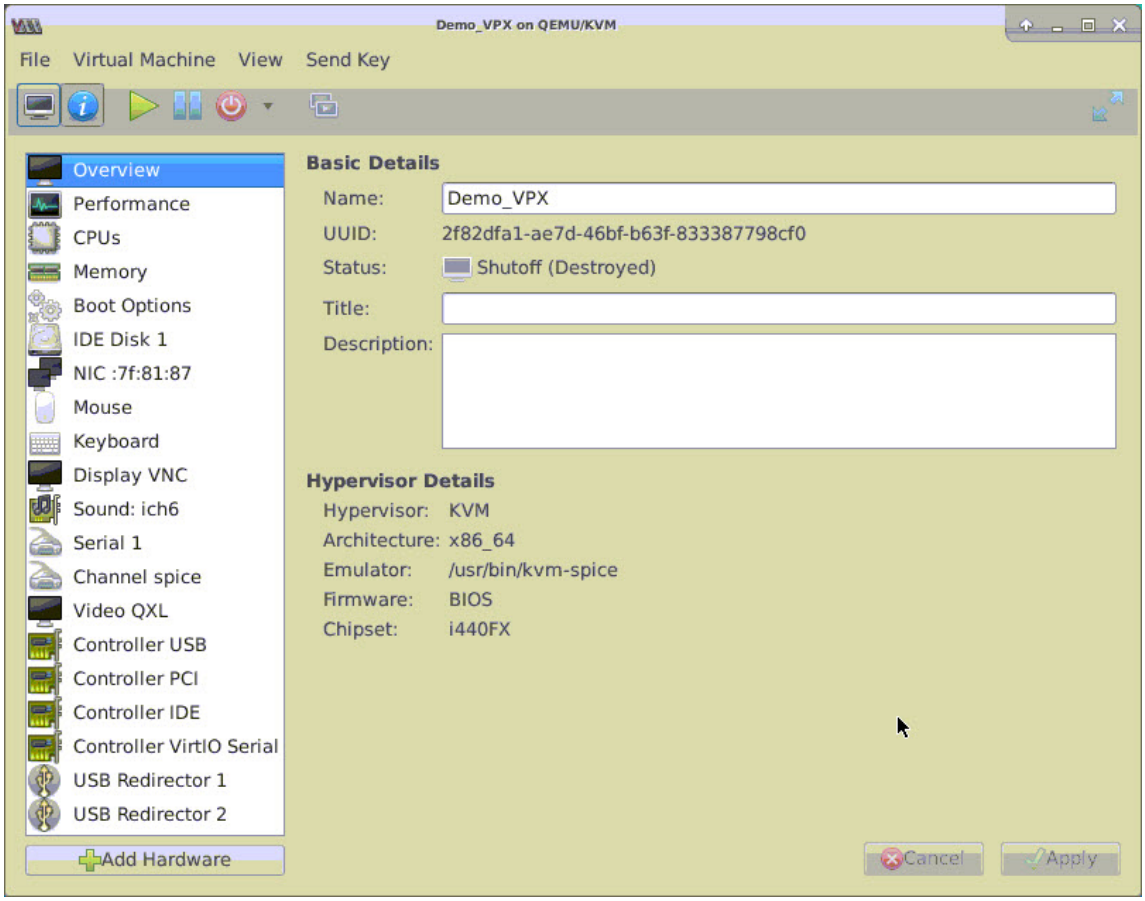
SR-IOV ネットワークインターフェイスの使用を **NetScaler ADC VPX** インスタンスで構成する

仮想マシンマネージャを使用して SR-IOV ネットワークインターフェイスを使用するように NetScaler ADC VPX インスタンスを構成するには、次の手順を実行します。

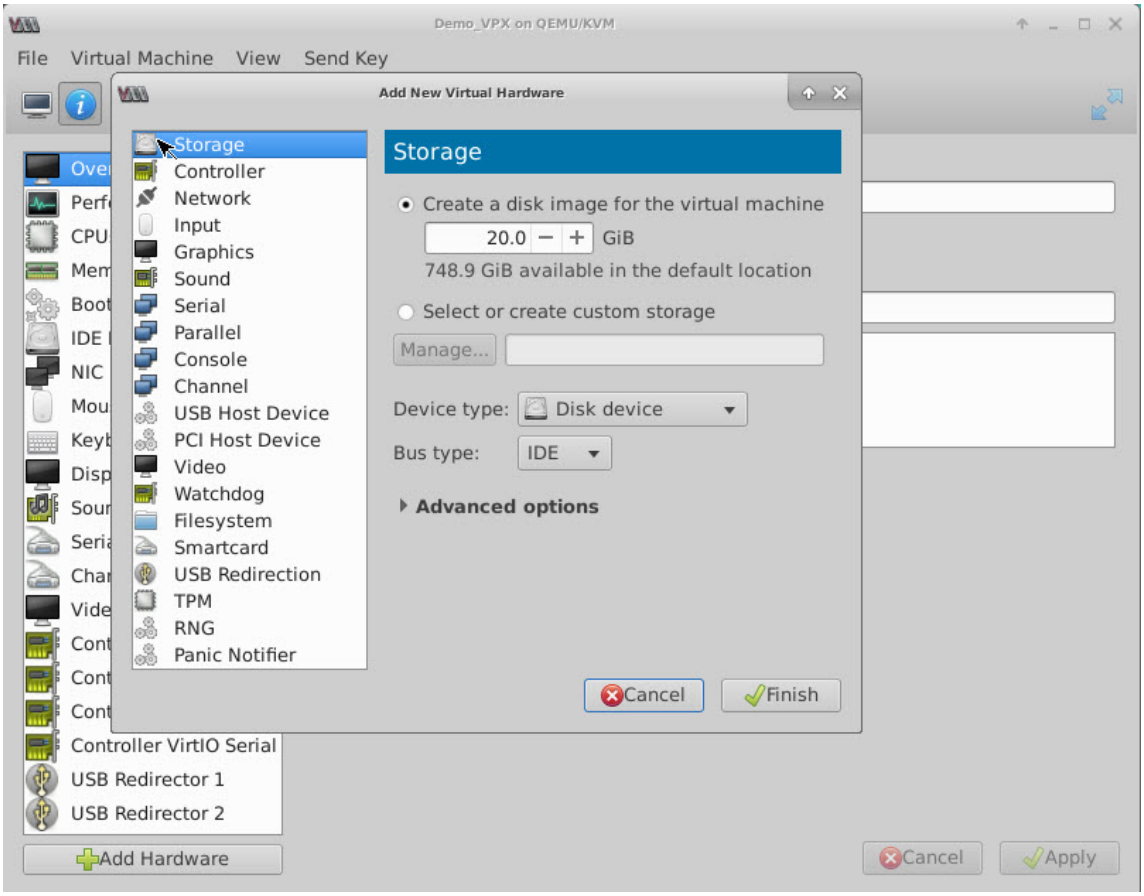
1. NetScaler VPX インスタンスの電源を切ります。
2. NetScaler VPX インスタンスを選択して、［Open］をクリックします。



3. <virtual machine on KVM> ウィンドウで、**i** アイコンを選択します。



4. [ハードウェアの追加] を選択します。



5. [新しい仮想ハードウェアの追加] ダイアログボックスで、次の操作を行います。

   a) PCI ホストデバイスを選択します。

   b) [Host Device] セクションで、作成した VF を選択して、[Finish] をクリックします。

図 **4**: Intel 82599 10G NIC の VF

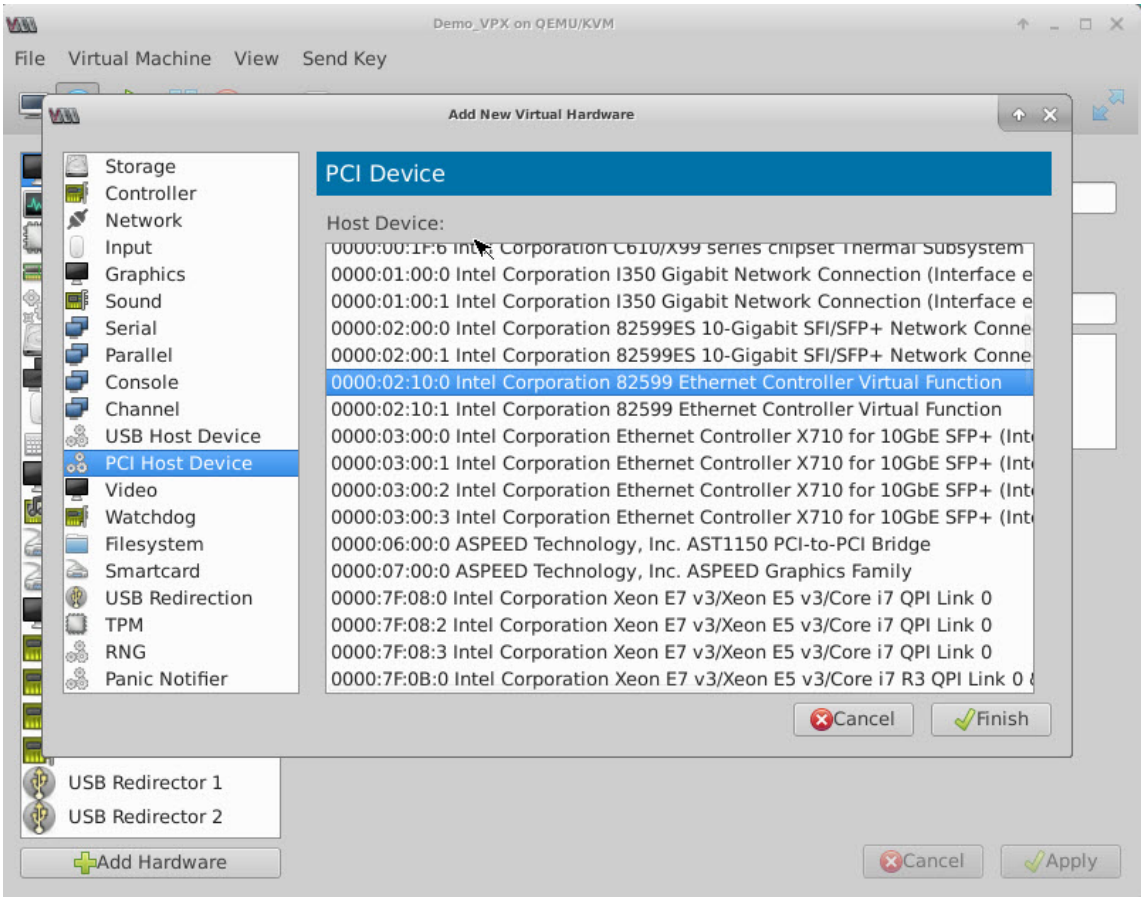


図 **5**: Intel XL710 40G NIC の VF

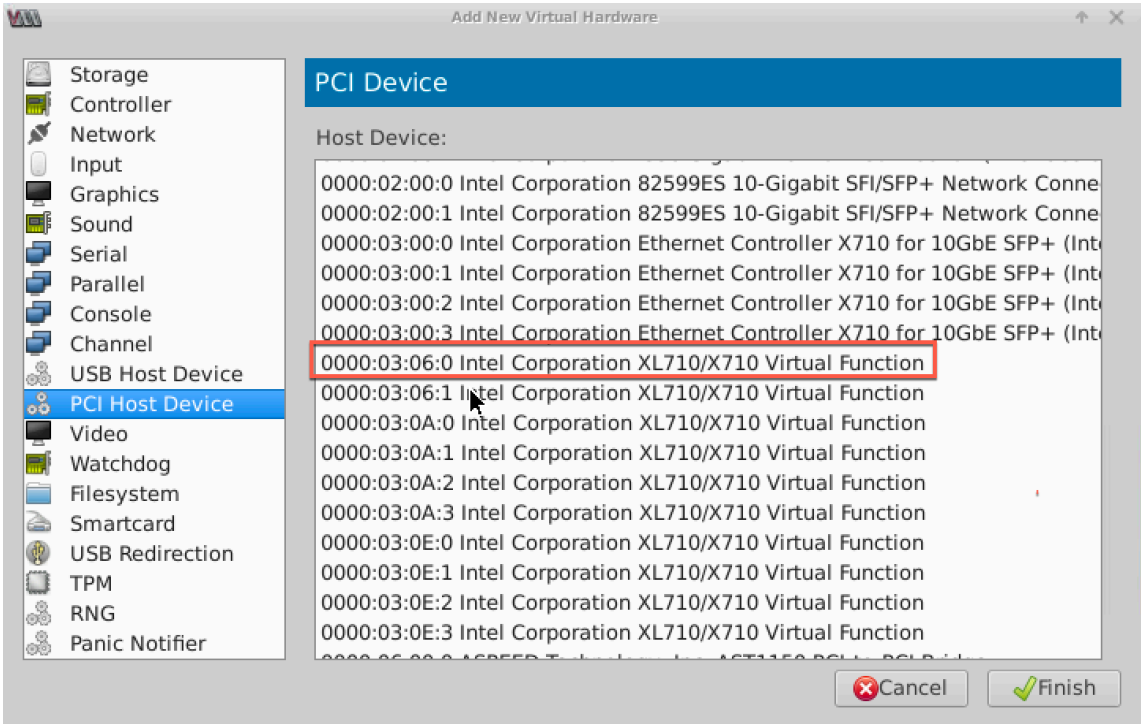
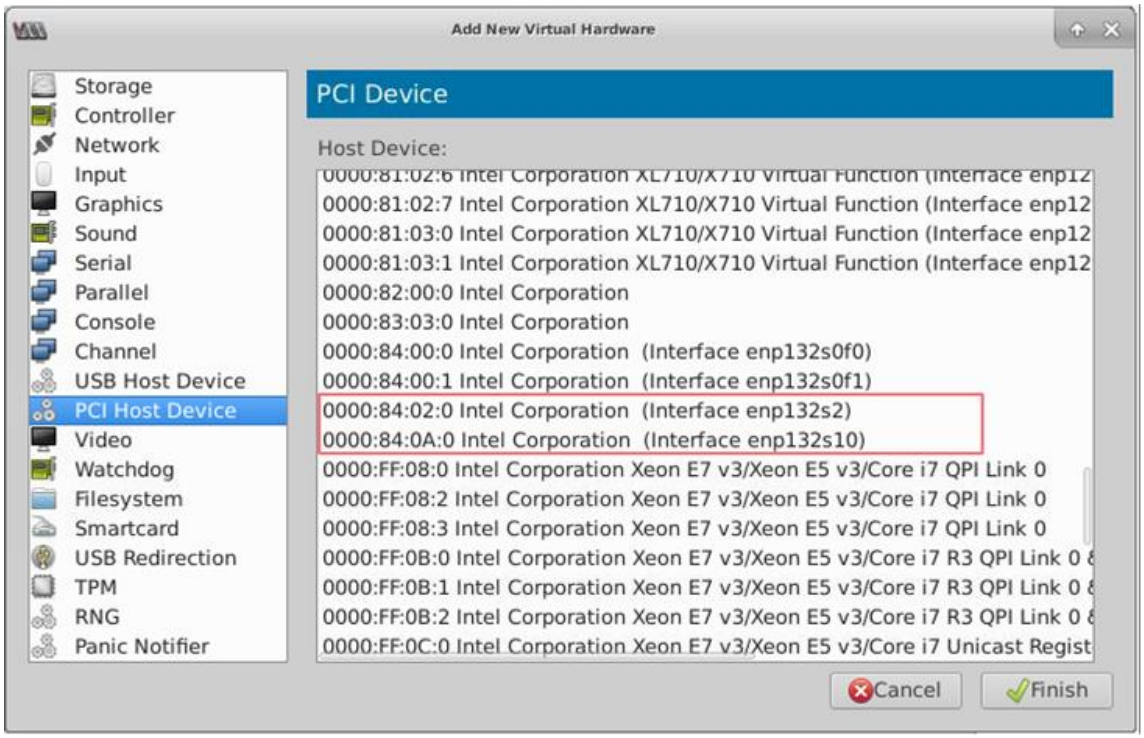


図 **6**: Intel X722 10G NIC の VF



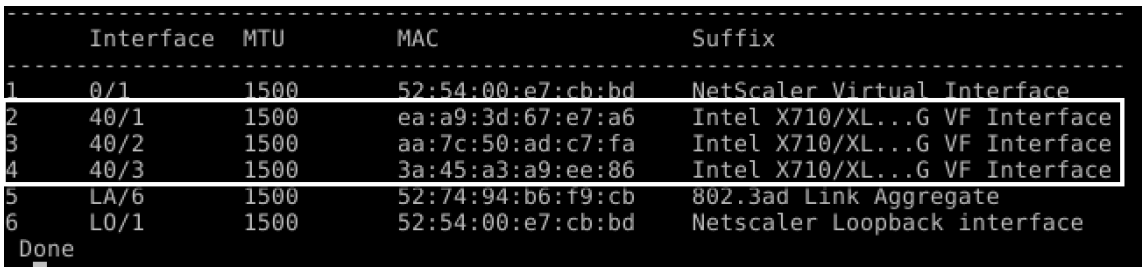
6. 手順 4 と 5 を繰り返し、作成した VF を追加します。
7. NetScaler VPX インスタンスをパワーオンします。
8. NetScaler VPX インスタンスがパワーオンしたら、次のコマンドを使用して構成を確認します。

```
1 show interface summary
2 <!--NeedCopy-->
```

構成したすべてのインターフェイスが出力に表示されます。

図 **6**: Intel 82599 NIC の出力の概要

フィギュア **7**. Intel X710 および XL710 NIC の出力の概要。



SR-IOV インターフェイスにスタティック **LA/LACP** を設定します

> 重要
>
> SR-IOV VF を作成する場合、MAC アドレスを VF に割り当てていないことを確認してください。

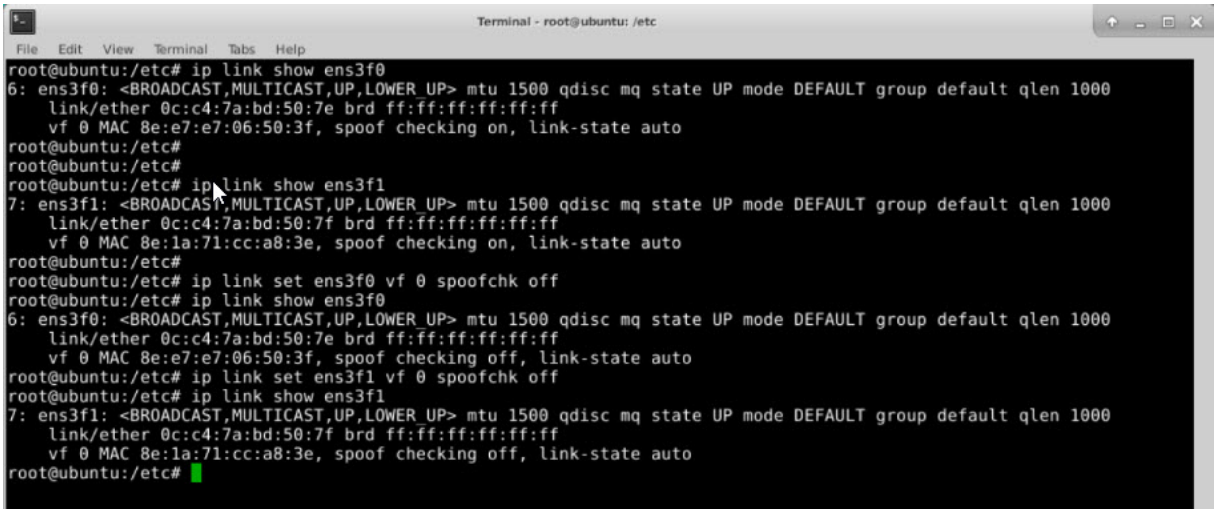
リンクアグリゲーションモードで、SR-IOV VF を使用するには、作成した VF のなりすましチェックを無効にします。KVM ホストでなりすましチェックを無効にするには、以下のコマンドを使用します。

*ip link set \<**interface**_name\> vf \<VF_id\> spoofchk off*

各項目の意味は次のとおりです:

- Interface_name –インターフェイス名です。
- VF_id –Virtual Function ID です。

例:



作成したすべての VF のなりすましチェックを無効にします。NetScaler VPX インスタンスを再起動し、リンクアグリゲーションを構成します。詳細な手順については、リンク集約の設定を参照してください。

SR-IOV インターフェイスで VLAN を構成する

SR-IOV VF で VLAN を構成できます。詳細な手順については 、VLAN の設定を参照してください。

> 重要
>
> KVM ホストに VF インターフェイスの VLAN 設定が含まれていないことを確認してください。

PCI パススルーネットワークインターフェイスを使用するように NetScaler VPX インスタンスを構成する

August 15, 2023

Linux-KVM プラットフォームに NetScaler VPX インスタンスをインストールして構成したら、仮想マシンマネージャーを使用して、PCI パススルーネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

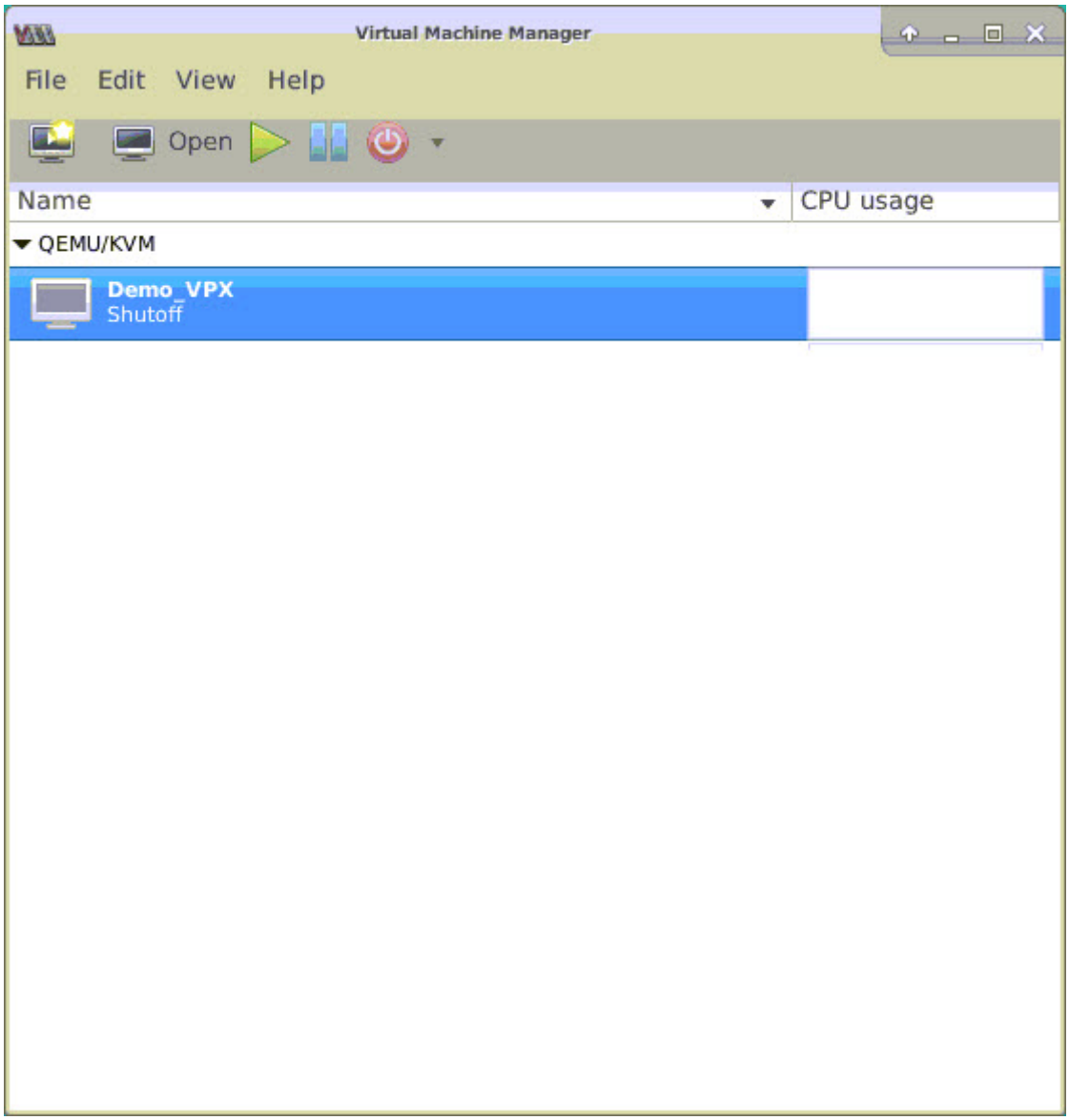
前提条件

- KVM ホスト上の Intel XL710 NIC (NIC) のファームウェア・バージョンは 5.04 です。
- KVM ホストは、IOMMU（Input–Output Memory Management Unit）と Intel VT をサポートし、これらは KVM ホストの BIOS で有効になっています。KVM ホストで **IOMMU** を有効にするには、**/boot/grub2/grub.cfg** ファイルに次のエントリを追加します。

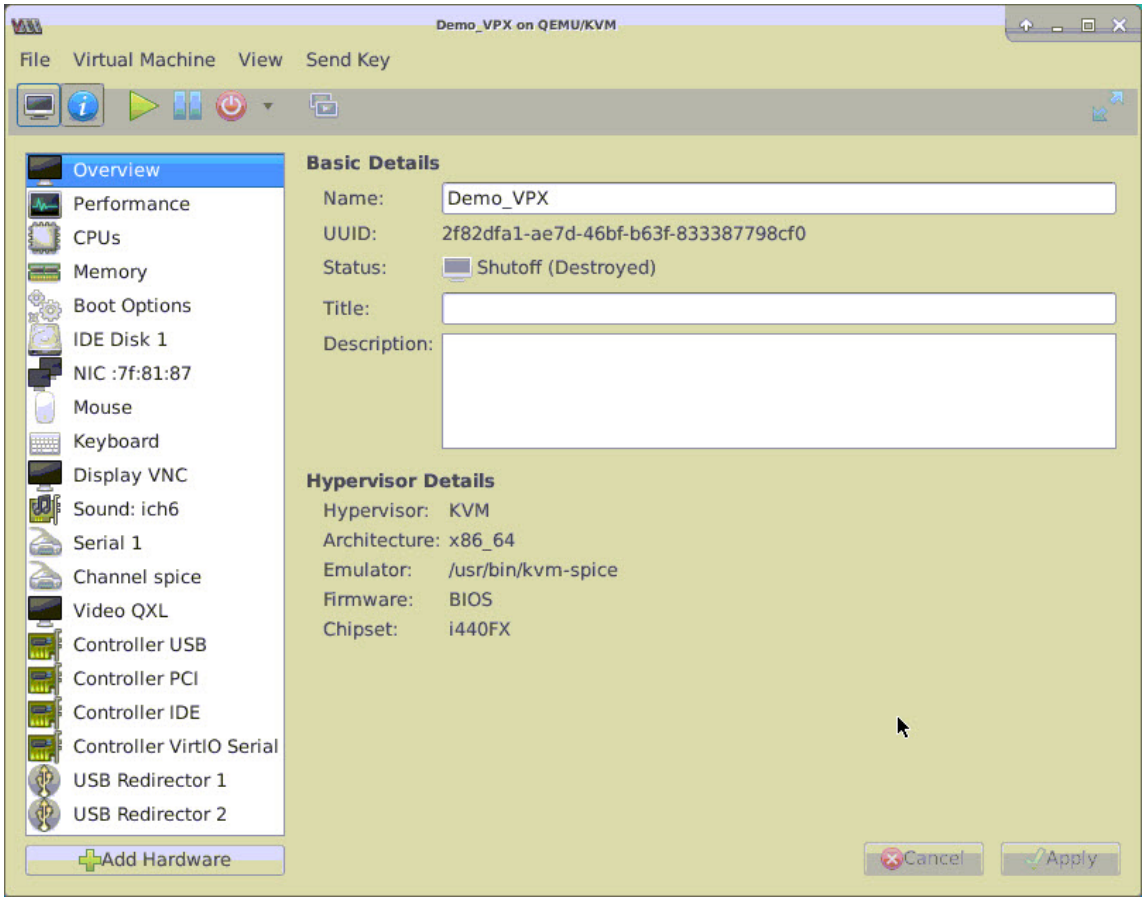
- 次のコマンドを実行して KVM ホストを再起動します。**grub2-MKConfig –o /boot/grub2/grub.cfg**

仮想マシンマネージャーを使用して **PCI** パススルーネットワークインターフェイスを使用するように **NetScaler ADC VPX** インスタンスを構成するには:

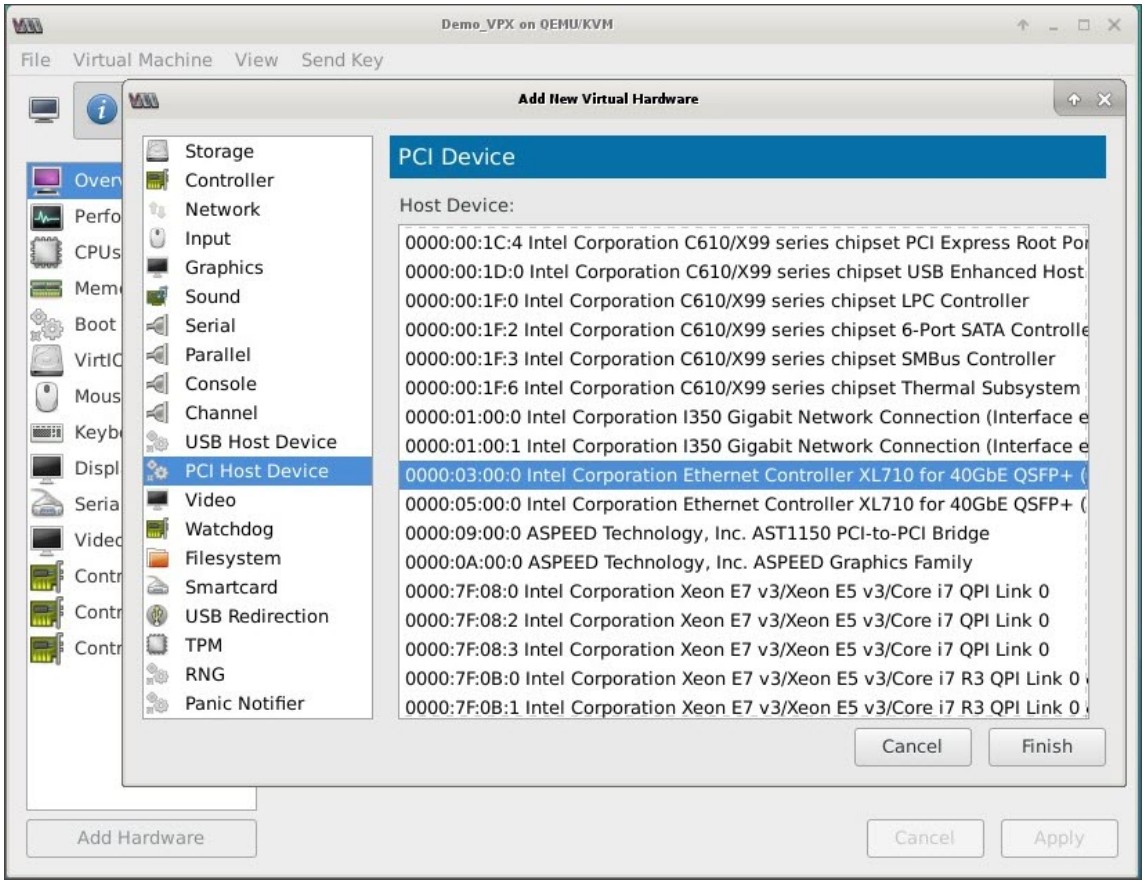
1. NetScaler VPX インスタンスの電源を切ります。
2. NetScaler VPX インスタンスを選択し、［開く］をクリックします。



3. **KVM>** の **virtual_machine** ウィンドウで、**i** アイコンをクリックします。



4. [ハードウェアの追加] をクリックします。
5. [新しい仮想ハードウェアの追加] ダイアログボックスで、次の操作を行います。
   a. **PCI** ホストデバイスを選択します。
   b.「ホストデバイス」セクションで、Intel XL710 の物理機能を選択します。
   c.［完了］をクリックします。

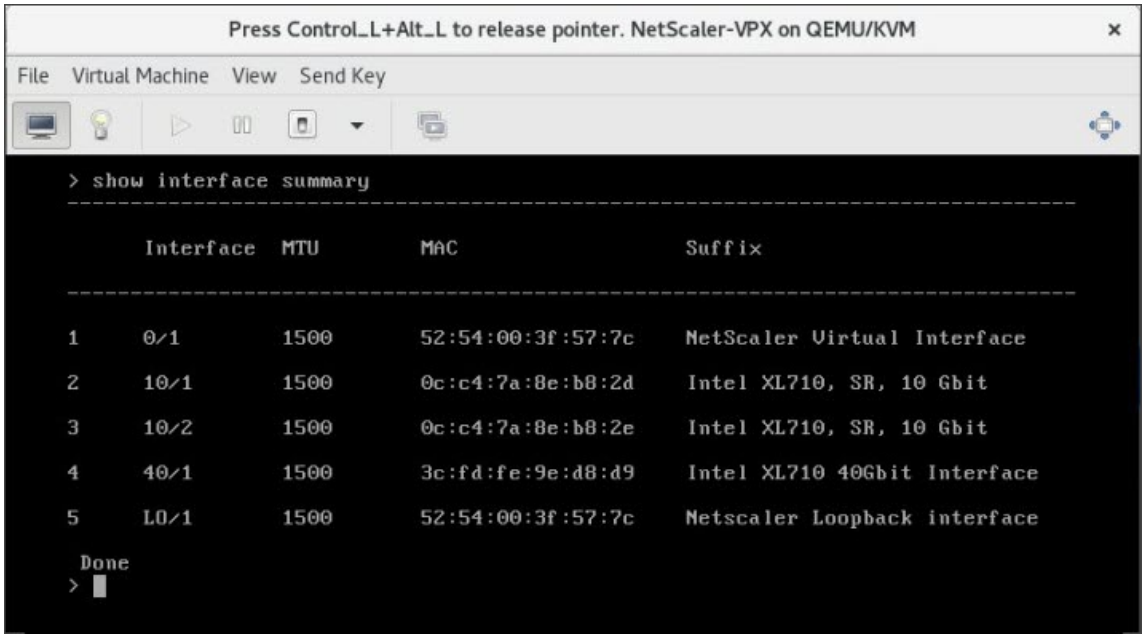


6. 手順 **4** と **5** を繰り返して、Intel XL710 物理関数を追加します。
7. NetScaler VPX インスタンスをパワーオンします。
8. NetScaler VPX インスタンスの電源が入ったら、次のコマンドを使用して構成を確認できます。

```
COMMAND

> show interface summary
```

出力には、設定したすべてのインターフェイスが表示されている必要があります。



virsh プログラムを使用して **NetScaler ADC VPX** インスタンスをプロビジョニングする

August 15, 2023

`virsh`プログラムは VM ゲストを管理するためのコマンドラインツールです。その機能性は Virtual Machine Manager に似ています。これにより VM Guest の状態（開始、停止、一時停止など）を変更でき、新しい Guests およびデバイスをセットアップして、既存の構成を編集できます。`virsh`プログラムは、VM ゲスト管理操作のスクリプト作成にも役立ちます。

`virsh`プログラムを使用して NetScaler ADC VPX をプロビジョニングするには、次の手順に従います。

1. tar コマンドを使用して、NetScaler VPX パッケージを解凍します。nsvpx-kvm-*_nc.tgz パッケージには、次のコンポーネントが含まれています。

   - VPX 属性 [NSVPX-KVM-*_nc.xml] を指定するドメイン XML ファイル
   - NS-VM ディスクイメージ [Checksum.txt] のチェックサム
   - NS-VM Disk Image [NSVPX-KVM-*_nc.raw]

   例:

```
tar -xvzf NSVPX-KVM-10.1-117_nc.tgz
NSVPX-KVM-10.1-117_nc.xml
NSVPX-KVM-10.1-117_nc.raw
checksum.txt
<!--NeedCopy-->
```


2. <DomainName>NSVPX-KVM- * _nc.xml XML ファイルを-NSVPX-KVM- *_nc.xml という名前のファイルにコピーします。<DomainName> は仮想マシンの名前でもあります。例:

```
cp NSVPX-KVM-10.1-117_nc.xml NetScaler-VPX-NSVPX-KVM-10.1-117_nc.xml
<!--NeedCopy-->
```

3. -NSVPX-KVM <DomainName>- * _nc.xml ファイルを編集して、以下のパラメータを指定します。

   - name - 名前を指定します。
   - Mac：MAC アドレスを指定します。
     注：ドメイン名と MAC アドレスは一意である必要があります。
   - source file：ディスクイメージの絶対ソースパスを指定します。ファイルパスは絶対パスである必要があります。RAW イメージファイルまたは QCOW2 イメージファイルのパスを指定することができます。

     RAW イメージファイルを指定する場合は、次の例のようにディスクイメージソースパスを指定します。

     例:

     ```
     <name>NetScaler-VPX</name>
     <mac address='52:54:00:29:74:b3'/>
     <source file='/root/NSVPX-KVM-10.1-117_nc.raw'/>
     <!--NeedCopy-->
     ```

     次の例に示すように、絶対 QCOW2 ディスクイメージソースパスを指定し、ドライバタイプを **qcow2** と定義します。

     例:

     ```
     <name>NetScaler-VPX</name>
     <mac address='52:54:00:29:74:b3'/>
     <driver name ='qemu' type='qcow2'/>
     <source file='/root/NSVPX-KVM-10.1-117_nc.qcow'/>*
     <!--NeedCopy-->
     ```

4. -NSVPX-KVM <DomainName>- * _nc.xml ファイルを編集して、ネットワークの詳細を設定します。

   - source dev - インターフェースを指定します。
   - mode - モードを指定します。デフォルトのインターフェイスは **Macvtap** ブリッジです。

   例: モード:macvTap Bridge ターゲットインターフェイスを`ethx`に設定し、モードをブリッジモデルタイプ`virtio`に設定

   ```
   <interface type='direct'>
       <mac address='52:54:00:29:74:b3'/>
       <source dev='eth0' mode='bridge'/>
       <target dev='macvtap0'/>
       <model type='virtio'/>
       <alias name='net0'/>
   ```



```
        <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
           function='0x0'/>
      </interface>
<!--NeedCopy-->
```

ここで、eth0 は仮想マシンに接続された物理インターフェイスです。

5. <DomainName> <DomainName> 次のコマンドを使用して、-NSVPX-KVM- * _nc.xml ファイル内の仮想マシン属性を定義します。virshは <DomainName>-NSVPX-KVM-*_nc.xml を定義:

```
virsh define NS-VPX-NSVPX-KVM-10.1-117_nc.xml
<!--NeedCopy-->
```

次のコマンドを入力して VM を起動します。virsh \<DomainUUID\ >] 例 start\ [\ <DomainName\ >

```
virsh start NetScaler-VPX
<!--NeedCopy-->
```

---

virsh コンソールコンソールから \<DomainUUID\ > \<DomainID\ >] 例 ゲスト VM を接続します\ [\ <DomainName\ >

```
virsh console NetScaler-VPX
<!--NeedCopy-->
```

virsh プログラムを使用して **NetScaler ADC VPX** インスタンスにインターフェイスを追加する

KVM 上で NetScaler VPX をプロビジョニングした後、追加のインターフェイスを付加できます。

インターフェイスを追加するには、次の手順を実行します。

1. KVM の上で動作している NetScaler VPX インスタンスをシャットダウンします。

<DomainName\ > 次のコマンドを使用し \<DomainUUID\ >] て-NSVPX-KVM-\ *\ _nc.xml ファイルを編集します:edit\ [\ virsh <DomainName\ >

2.

3. -NSVPX-KVM <DomainName>- * _nc.xml ファイルに、次のパラメータを追加します。

a) **MacVTap** 用

- Interface type - インターフェイスの種類として「direct」を指定します。
- MAC アドレス：MAC アドレスを指定し、MAC アドレスがインターフェイス全体で一意であることを確認します。
- source dev - インターフェイス名を指定します。
- mode-モードを指定します。サポートされているモードは、ブリッジ、VEPA、プライベート、パススルーです。
- モデルタイプ-モデルタイプを次のように指定します。`virtio`

例:

モード：MacVTap Pass-through

ターゲットインタフェースを
`ethx`、モードを
ブリッジ、モデルタイプを次のように設定します。
`virtio`

```
<interface type='direct'>
      <mac address='52:54:00:29:74:b3'/>
      <source dev='eth1' mode='passthrough'/>
      <model type='virtio'/>
 </interface>
<!--NeedCopy-->
```

ここで eth1 は仮想マシンに接続された物理インターフェイスです。

b) ブリッジモード用

注：KVM ホストで Linux Bridge が構成され、Bridge に物理インターフェイスが結合されて、Bridge が UP 状態になっている必要があります。

- Interface type - インターフェイスの種類として「bridge」を指定します。
- MAC アドレス：MAC アドレスを指定し、MAC アドレスがインターフェイス全体で一意であることを確認します。
- source dev - ブリッジ名を指定します。
- モデルタイプ-モデルタイプを次のように指定します。`virtio`

例：Bridge Mode

```
<interface type='bridge'>
      <mac address='52:54:00:2d:43:a4'/>
      <source bridge='br0'/>
      <model type='virtio'/>
 </interface>
<!--NeedCopy-->
```


NetScaler VPX ゲスト仮想マシンの管理

August 15, 2023

仮想マシンマネージャと `virsh`プログラムを使用して、仮想マシンゲストの起動または停止、新しいゲストとデバイスの設定、既存構成の編集、仮想ネットワークコンピューティング (VNC) によるグラフィカルコンソールへの接続などの管理タスクを実行できます。

仮想マシンマネージャーを使用して VPX ゲスト仮想マシンを管理する

- VM ゲストを一覧表示する

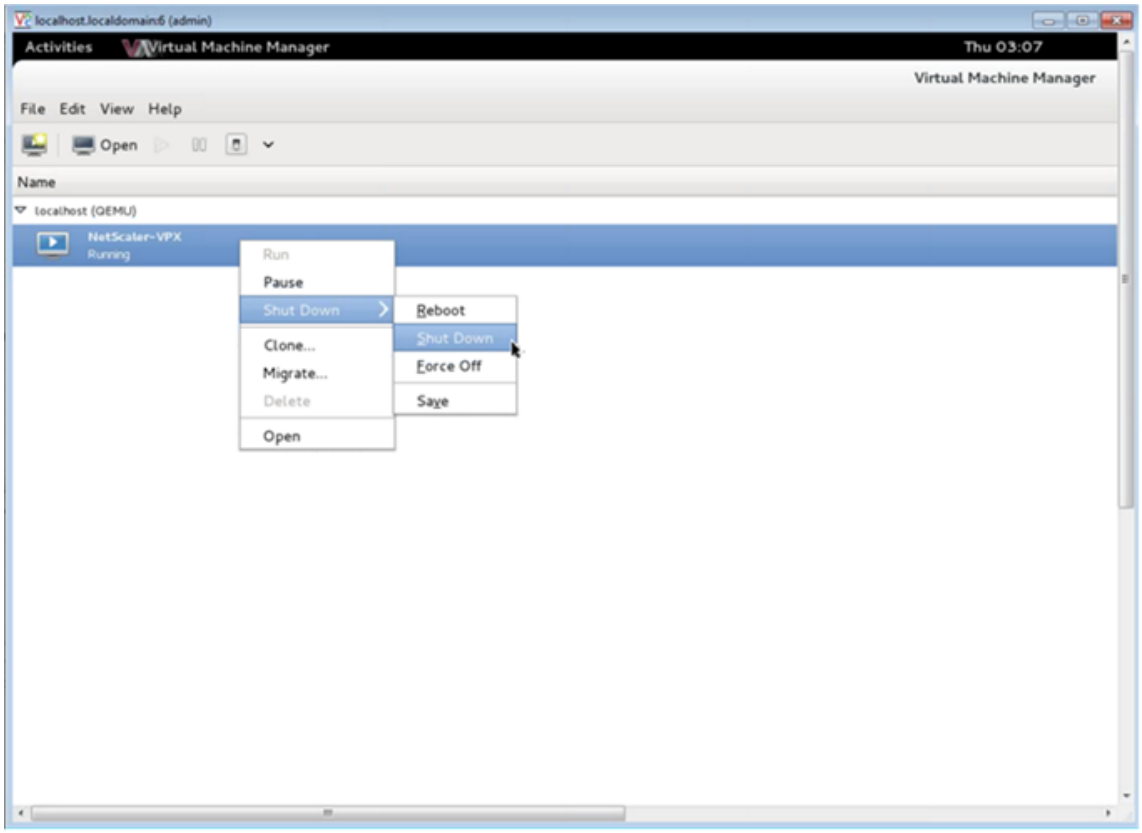
  Virtual Machine Manager のメインウィンドウには、接続される各 VM ホストサーバのすべての VM Guests の一覧が表示されます。各仮想マシンゲストエントリには、仮想マシンの名前と、アイコンに表示されるステータス（実行中、一時停止、またはシャットオフ）が含まれます。

- グラフィカルコンソールを開く

  VM Guest に対してグラフィカルコンソールを開いて、VNC 接続介して物理的ホストと通信するようにマシンと相互通信できます。Virtual Machine Manager でグラフィカルコンソールを開くには、VM Guest エントリーを右クリックして、ポップアップメニューで［オープン］オプションを選択します。

- ゲストの起動とシャットダウン

  Virtual Machine Manager から VM Guest を開始または停止できます。VM の状態を変更するには、VM Guest エントリーを右クリックして、ポップアップメニューで［Run］または［Shut Down］オプションのいずれかを選択します。



- ゲストを再起動

  Virtual Machine Manager から VM Guest を再起動できます。VM を再起動するには、VM Guest エントリーを右クリックして、ポップアップメニューで［Shut Down］>［Reboot］を選択します。。

- ゲストを削除する

  デフォルトでは、VM Guest を削除すると XML 構成が消去されます。また、ゲストのストレージファイルを削除できます。これを実行して、完全にそうすることはゲストを消します。

  1. Virtual Machine Manager で、VM Guest エントリーを右クリックします。
  2. ポップアップメニューで［Delete from］を選択します。確認用のウィンドウが開きます。
     注: 削除オプションは、VM ゲストがシャットダウンされている場合にのみ有効になります。
  3. ［削除］をクリックします。
  4. 完全にゲストを消去するには、［Delete Associated Storage Files］チェックボックスをオンにして、関連付けられた.raw ファイルを削除します。

`virsh` プログラムを使用して **NetScaler ADC VPX** ゲスト仮想マシンを管理する

- VM ゲストとその現在の状態を一覧表示します。

  ゲストに関する情報を表示するために`virsh`を使用するには


```
virsh list --all
```

コマンド出力はすべてのドメインとその状態を表示します。出力例:

```
Id Name                 State
----------------------------------
  0 Domain-0           running
  1 Domain-1           paused
  2 Domain-2           inactive
  3 Domain-3           crashed
<!--NeedCopy-->
```

- virshコンソールを開きます。

  ゲスト仮想マシンをコンソールから接続します。

  ```
  virsh console [<DomainID> | <DomainName> | <DomainUUID>]
  ```

  例:

  ```
  virsh console NetScaler-VPX
  ```

- ゲストを起動してシャットダウンします。

  Guest は DomainName または Domain-UUID を使って開始できます。

  ```
  virsh start [<DomainName> | <DomainUUID>]
  ```

  例:

  ```
  virsh start NetScaler-VPX
  ```

  ゲストをシャットダウンするには:

  ```
  virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
  ```

  例:

  ```
  virsh shutdown NetScaler-VPX
  ```

- ゲストを再起動

  ```
  virsh reboot [<DomainID> | <DomainName> | <DomainUUID>]
  ```

  例:

  ```
  virsh reboot NetScaler-VPX
  ```

  ゲストを削除する

  ゲスト仮想マシンを削除するには、削除コマンドを実行する前に、ゲストをシャットダウンして-NSVPX-KVM <DomainName>- * _nc.xml を定義解除する必要があります。

  ```
  virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
  virsh undefine [<DomainName> | <DomainUUID>]
  <!--NeedCopy-->
  ```


例:

```
virsh shutdown NetScaler-VPX
virsh undefine NetScaler-VPX
<!--NeedCopy-->
```

> 注:delete コマンドでは、手動で削除する必要があるディスクイメージファイルは削除されません。

OpenStack 上で **SR-IOV** を使用して **NetScaler VPX** インスタンスをプロビジョニングします

August 15, 2023

OpenStack で、シングルルート I/O 仮想化（Single-Root I/O Virtualization：SR-IOV）テクノロジを使用する高パフォーマンスの NetScaler VPX インスタンスを展開できます。

OpenStack で、3 つの手順で、SR-IOV テクノロジを使用する NetScaler VPX インスタンスを展開できます。

- ホスト上で SR-IOV Virtual Functions（VF）を有効にします。
- VF を構成し、OpenStack で使用できるようにします。
- OpenStack で NetScaler VPX をプロビジョニングします。

前提条件

次のことを確実にします。

- Intel 82599 NIC (NIC) をホストに追加します。
- 最新の IXGBE ドライバーをダウンロードしてインストールします。
- ホスト上の IXGBEVF ドライバをブロックリストします。/etc/modprobe.d/blacklist.conf ファイルに次のエントリを追加します。ブロックリスト `ixgbevf`

> 注
>
> `ixgbe`ドライバーのバージョンは 5.0.4 以上でなければなりません。

ホストで **SR-IOV VF** を有効にする

SR-IOV VF を有効にするには、次のいずれかの手順を実行します。

- 3.8 より前のカーネルバージョンを使用している場合は、/etc/modprobe.d/ixgbe ファイルに次のエントリを追加し、ホストを再起動します。options ixgbe max_vfs= <number_of_VFs>

- カーネル 3.8 以降のバージョンを使用している場合、以下のコマンドを使用して VF を作成します。

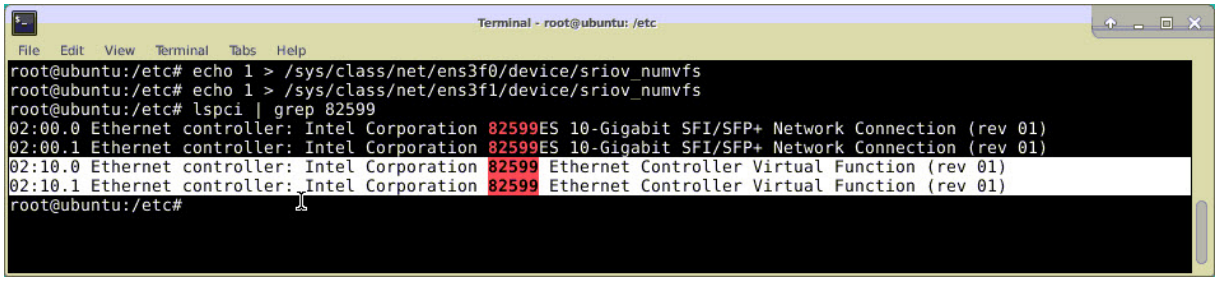
```
1     echo <number_of_VFs> > /sys/class/net/<device_name>/device/
         sriov_numvfs
2  <!--NeedCopy-->
```

各項目の意味は次のとおりです:

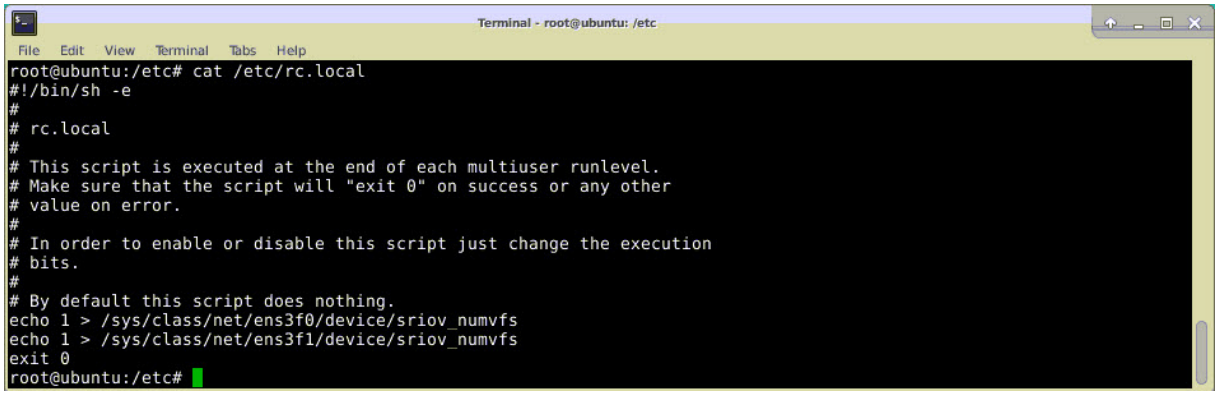
- number_of_VFs は、作成する Virtual Function の数です。
- device_name はインターフェイス名です。

> 重要
>
> SR-IOV VF を作成する場合、MAC アドレスを VF に割り当てないようにしてください。

次に、作成している 4 つの VF の例を示します。



VF を永続的にし、VF の作成に使用したコマンドを **rc.local** ファイルに追加します。rc.local ファイルの内容を示す例を次に示します。



詳細については、この Intel SR-IOV 構成ガイドを参照してください。

OpenStack で VF を設定して利用できるようにする

OpenStack で SR-IOV を設定するには、以下のリンクに記載されている手順に従ってください:. https://wiki.openstack.org/wiki/SR-IOV-Passthrough-For-Networking

OpenStack で NetScaler ADC VPX インスタンスをプロビジョニングする

OpenStack CLI を使用して、OpenStack 環境で NetScaler ADC VPX インスタンスをプロビジョニングできます。

VPX インスタンスをプロビジョニングします。オプションで、コンフィグドライブのデータを使用します。「構成ドライブ」とは、インスタンスの起動時にアタッチされる特殊な構成ドライブを指します。この構成ドライブを使用して、インスタンスのネットワーク設定を構成する前に、管理 IP アドレス、ネットワークマスク、デフォルトゲートウェイなどのネットワーク構成情報をインスタンスに渡すことができます。

OpenStack が VPX インスタンスをプロビジョニングする場合、まず OpenStack を示す特定の BIOS 文字列（OpenStack ファウンデーション）を読み取ることによって、インスタンスが OpenStack 環境で起動していることを検出します。Red Hat Linux ディストリビューションの場合、この文字列は/etc/nova/release に保存されます。これは、KVM ハイパーバイザープラットフォームに基づくすべての OpenStack 実装で使用できる標準メカニズムです。ドライブには特定の OpenStack ラベルが必要です。構成ドライブが検出されると、インスタンスは nova boot コマンドで指定されたファイル名から次の情報を読み取ろうとします。以下の手順では、このファイルを「userdata.txt」と呼びます。

- 管理 IP アドレス
- ネットワークマスク
- デフォルトゲートウェイ

パラメーターが正しく読み取られると、それらの値が NetScaler スタックに適用されます。これにより、インスタンスをリモートから管理できるようになります。パラメーターが読み取られない場合、または構成ドライブが存在しない場合は、インスタンスが以下のデフォルトの処理を実行します。

- DHCP から IP アドレス情報を取得する。
- DHCP から情報を取得できない場合は、デフォルトのネットワーク構成として 192.168.100.1/16 を使用する。

CLI を使用して OpenStack 上の NetScaler VPX インスタンスをプロビジョニングします

OpenStack CLI を使用して、OpenStack 環境で VPX インスタンスをプロビジョニングできます。次に、OpenStack で NetScaler ADC VPX インスタンスをプロビジョニングする手順の概要を示します。

1. .tgz ファイルから .qcow2ファイルを抽出する
2. qcow2 イメージから OpenStack イメージを作成する
3. VPX インスタンスの Provisioning

OpenStack 環境で VPX インスタンスをプロビジョニングするには、次の手順を実行します。

1. 次のコマンドを入力して、 .tqzファイルからqcow2ファイルを抽出します。


```
tar xvzf <TAR file>
tar xvzf NSVPX-KVM-12.0-26.2_nc.tgz
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
<!--NeedCopy-->
```

2. 次のコマンドを入力して、手順 1 で抽出した.qcoz2ファイルを使用して OpenStack イメージをビルドします。

```
glance image-create --name="<name of the OpenStack image>" --
    property hw_disk_bus=ide --is-public=true --container-format=
    bare --disk-format=qcow2< <name of the qcow2 file>

glance image-create --name="NS-VPX-12-0-26-2" --property
    hw_disk_bus=ide --is-public= true --container-format=bare --
    disk-format=qcow2< NSVPX-KVM-12.0-26.2_nc.qcow2
<!--NeedCopy-->
```

下図は、glance image-create コマンドの出力例です。

```
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 735dae4ea6e46e39ed3f0acfba02e755     |
| container_format | bare                                 |
| created_at       | 2017-02-16T10:03:29Z                 |
| disk_format      | qcow2                                |
| hw_disk_bus      | ide                                  |
| id               | aeaa13e9-b49b-411c-ab54-c61820a8e2f3 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | NSVPX-KVM-12.0-26.2                  |
| owner            | 06c41a73b32f4b48af55359fd7d3502c     |
| protected        | False                                |
| size             | 717946880                            |
| status           | active                               |
| tags             | []                                   |
| updated_at       | 2017-02-16T10:03:38Z                 |
| virtual_size     | None                                 |
| visibility       | private                              |
+------------------+--------------------------------------+
```

3. OpenStack イメージが作成されたら、NetScaler VPX インスタンスをプロビジョニングします。

```
nova boot --image NSVPX-KVM-12.0-26.2 --config-drive=true --
    userdata
./userdata.txt --flavor m1. medium --nic net-id=3b258725-eaae-
```



```
455e-a5de-371d6d1f349f --nic port-id=218ba819-9f55-4991-adb6-
02086a6bdee2 NSVPX-10
<!--NeedCopy-->
```

前述のコマンドでは、userdata.txt は、VPX インスタンスの IP アドレス、ネットマスク、デフォルトゲートウェイなどの詳細を含むファイルです。ユーザーデータファイルは、ユーザーカスタマイズ可能なファイルです。NSVPX-KVM-12.0-26.2 は、プロビジョニングする仮想アプライアンスの名前です。—NIC port-id=218ba819-9f55-4991-adb6-02086a6bdee2 は OpenStack VF です。

次の図に、nova boot コマンドの出力例を示します。

```
+--------------------------------------+--------------------------------------------------------------+
| Property                             | Value                                                        |
+--------------------------------------+--------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                       |
| OS-EXT-AZ:availability_zone          |                                                              |
| OS-EXT-SRV-ATTR:host                 | -                                                            |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | -                                                            |
| OS-EXT-SRV-ATTR:instance_name        | instance-0000003c                                            |
| OS-EXT-STS:power_state               | 0                                                            |
| OS-EXT-STS:task_state                | scheduling                                                   |
| OS-EXT-STS:vm_state                  | building                                                     |
| OS-SRV-USG:launched_at               | -                                                            |
| OS-SRV-USG:terminated_at             | -                                                            |
| accessIPv4                           |                                                              |
| accessIPv6                           |                                                              |
| adminPass                            | 43EjPdM5shLz                                                 |
| config_drive                         | True                                                         |
| created                              | 2017-02-20T11:53:37Z                                         |
| flavor                               | m1.medium (3)                                                |
| hostId                               |                                                              |
| id                                   | 6b9f6968-aab9-463c-b619-d58c73db3187                         |
| image                                | NSVPX-KVM-12.0-26.2 (a5478b8a-8435-48d1-b4a0-1494e2c8f8b1)   |
| key_name                             | -                                                            |
| metadata                             | {}                                                           |
| name                                 | NSVPX-10                                                     |
| os-extended-volumes:volumes_attached | []                                                           |
| progress                             | 0                                                            |
| security_groups                      | default                                                      |
| status                               | BUILD                                                        |
| tenant_id                            | 06c41a73b32f4b48af55359fd7d3502c                             |
| updated                              | 2017-02-20T11:53:38Z                                         |
| user_id                              | 418524f7101b4f0389ecbb36da9916b5                             |
+--------------------------------------+--------------------------------------------------------------+
```

次の図は、userdata.txt ファイルのサンプルです。<PropertySection></PropertySection> タグ内の値はユーザー設定可能な値で、IP アドレス、ネットマスク、デフォルトゲートウェイなどの情報を保持します。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1">
<PlatformSection>
<Kind>NOVA</Kind>
<Version>2013.1</Version>
<Vendor>Openstack</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"
```



```
      />
14  <Property oe:key="com.citrix.netscaler.platform" oe:value="vpx"/>
15  citrix.com 4
16  <Property oe:key="com.citrix.netscaler.orch_env"
17  oe:value="openstack-orch-env"/>
18  <Property oe:key="com.citrix.netscaler.mgmt.ip"
19  oe:value="10.1.0.100"/>
20  <Property oe:key="com.citrix.netscaler.mgmt.netmask"
21  oe:value="255.255.0.0"/>
22  <Property oe:key="com.citrix.netscaler.mgmt.gateway"
23  oe:value="10.1.0.1"/>
24  </PropertySection>
25  </Environment>
26  <!--NeedCopy-->
```

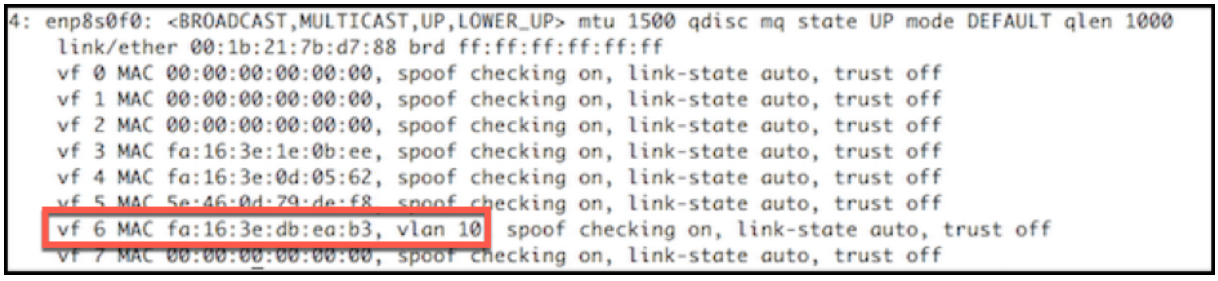
サポートされているその他の構成：ホストからの **SR-IOV VF** 上の **VLAN** の作成と削除

SR-IOV VF 上の VLAN を作成するには、次のコマンドを入力します。

```
ip link show enp8s0f0 vf 6 vlan 10
```

前述のコマンドでは、「enp8s0f0」は物理機能の名前です。

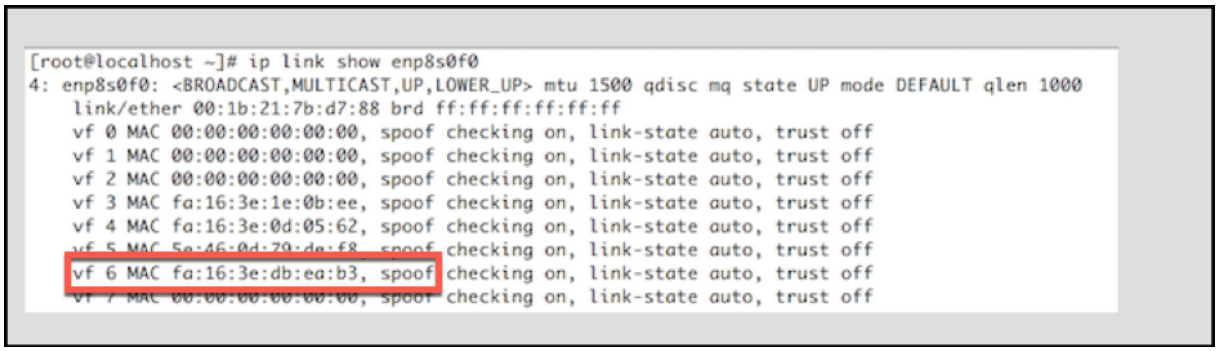
例：vf 6 で作成された VLAN 10



SR-IOV VF 上の VLAN を削除するには、次のコマンドを入力します。

```
ip link show enp8s0f0 vf 6 vlan 0
```

例：VLAN 10、vf 6 から削除された



これらの手順により、SRIOV テクノロジを使用する NetScaler VPX インスタンスを OpenStack 上で展開する方法が完了します。

KVM 上の NetScaler VPX インスタンスが OVS DPDK ベースのホストインターフェイスを使用するように構成する

August 15, 2023

KVM（Fedora と RHOS）で実行されている NetScaler VPX インスタンスを Open vSwitch（OVS）と Data Plane Development Kit（DPDK）を使用するように構成して、ネットワークパフォーマンスを向上させることができます。このドキュメントでは、KVM ホスト上の OVS-DPDK によって公開されるvhost-user ポートで動作するように NetScaler ADC VPX インスタンスを構成する方法について説明します。

OVS は、オープンソースの Apache 2.0 ライセンスでライセンスされている多層仮想スイッチです。DPDK は、高速パケット処理のためのライブラリとドライバのセットです。

以下のバージョンの Fedora、RHOS、OVS、および DPDK は、NetScaler VPX インスタンスを設定するために認定されています。

Fedora	RHOS
Fedora 25	RHOS 7.4
OVS 2.7.0	OVS 2.6.1
DPDK 16.11.12	DPDK 16.11.12

前提条件

DPDK をインストールする前に、ホストに 1GB の巨大なページがあることを確認してください。

詳細については、この DPDK システム要件ドキュメントを参照してください。OVS DPDK ベースのホストインターフェイスを使用するように KVM で NetScaler ADC VPX インスタンスを構成するために必要な手順の概要は次のとおりです。

- DPDK をインストールします。
- OVS を構築し、インストールします。
- OVS ブリッジを作成します。
- OVS ブリッジに物理インターフェイスを接続します。
- OVS データパスにvhost-userポートを接続します。
- OVS-DPDK ベースのvhost-userポートで KVM-VPX をプロビジョニングします

DPDK のインストール

DPDK をインストールするには、この Open vSwitch with DPDK ドキュメントに記載されている指示に従ってください。

OVS のビルドとインストール

OVS のダウンロードページから OVS をダウンロードします。次に、DPDK データパスを使用して OVS をビルドおよびインストールします。「Open vSwitch のインストール」ドキュメントに記載されている手順に従います。

詳細については、「DPDK 入門ガイド for Linux」を参照してください。

OVS ブリッジの作成

必要に応じて、Fedora コマンドか RHOS コマンドを入力して、OVS ブリッジを作成します。

Fedora コマンド:

```
> $OVS_DIR/utilities/ovs-vsctl add-br ovs-br0 -- set bridge ovs-br0
    datapath_type=netdev
<!--NeedCopy-->
```

RHOS コマンド:

```
ovs-vsctl add-br ovs-br0 -- set bridge ovs-br0 datapath_type=netdev
<!--NeedCopy-->
```

物理インターフェースを OVS ブリッジに接続します

ポートを DPDK にバインドし、次の Fedora または RHOS コマンドを入力して OVS ブリッジに接続します。

Fedora コマンド:

```
> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 dpdk0 -- set Interface
    dpdk0 type=dpdk  options:dpdk-devargs=0000:03:00.0

> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 dpdk1 -- set Interface
    dpdk1 type=dpdk  options:dpdk-devargs=0000:03:00.1
<!--NeedCopy-->
```

RHOS コマンド:

```
ovs-vsctl add-port ovs-br0 dpdk0 -- set Interface dpdk0 type=dpdk
    options:dpdk-devargs=0000:03:00.0


```



```
ovs-vsctl add-port ovs-br0 dpdk1 -- set Interface dpdk1 type=dpdk
    options:dpdk-devargs=0000:03:00.1
<!--NeedCopy-->
```

オプションの一部として表示される dpdk-devargs は、それぞれの物理 NIC の PCI BDF を指定します。

OVS データパスに vhost-user ポートを接続する

OVS データパスにvhost-userポートを接続するには、次の Fedora または RHOS コマンドを入力します。

Fedora コマンド:

```
> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 vhost-user1 -- set
    Interface vhost-user1 type=dpdkvhostuser -- set Interface vhost-
    user1  mtu_request=9000

> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 vhost-user2 -- set
    Interface vhost-user2 type=dpdkvhostuser -- set Interface vhost-
    user2  mtu_request=9000

chmod g+w  /usr/local/var/run/openvswitch/vhost*
<!--NeedCopy-->
```

RHOS コマンド:

```
ovs-vsctl add-port ovs-br0 vhost-user1 -- set Interface vhost-user1
    type=dpdkvhostuser -- set Interface vhost-user1  mtu_request=9000

ovs-vsctl add-port ovs-br0 vhost-user2 -- set Interface vhost-user2
    type=dpdkvhostuser -- set Interface vhost-user2  mtu_request=9000

chmod g+w /var/run/openvswitch/vhost*
<!--NeedCopy-->
```

OVS-DPDK ベースの vhost-user ポートを持つ KVM-VPX のプロビジョニング

次の QEMU コマンドを使用して、CLI からのみ OVS-DPDK ベースのvhost-userポートを持つ Fedora KVM 上の VPX インスタンスをプロビジョニングできます。

Fedora コマンド:

```
qemu-system-x86_64 -name KVM-VPX -cpu host -enable-kvm -m 4096M \

-object memory-backend-file,id=mem,size=4096M,mem-path=/dev/hugepages,
    share=on -numa node,memdev=mem \

-mem-prealloc -smp sockets=1,cores=2 -drive file=<absolute-path-to-disc
    -image-file>,if=none,id=drive-ide0-0-0,format=<disc-image-format> \

```



```
-device ide-drive,bus=ide.0,unit=0,drive=drive-ide0-0-0,id=ide0-0-0,
    bootindex=1 \

-netdev type=tap,id=hostnet0,script=no,downscript=no,vhost=on \

-device virtio-net-pci,netdev=hostnet0,id=net0,mac=52:54:00:3c:d1:ae,
    bus=pci.0,addr=0x3 \

-chardev socket,id=char0,path=</usr/local/var/run/openvswitch/vhost-
    user1> \

-netdev type=vhost-user,id=mynet1,chardev=char0,vhostforce -device
    virtio-net-pci,mac=00:00:00:00:00:01,netdev=mynet1,mrg_rxbuf=on \

-chardev socket,id=char1,path=</usr/local/var/run/openvswitch/vhost-
    user2> \

-netdev type=vhost-user,id=mynet2,chardev=char1,vhostforce -device
    virtio-net

pci,mac=00:00:00:00:00:02,netdev=mynet2,mrg_rxbuf=on \

--nographic
<!--NeedCopy-->
```

RHOS の場合は、次のサンプル XML ファイルを使用して、virshを使用して NetScaler ADC VPX インスタンスをプロビジョニングします。

```
<domain type='kvm'>

  <name>dpdk-vpx1</name>

  <uuid>aedb844b-f6bc-48e6-a4c6-36577f2d68d6</uuid>

  <memory unit='KiB'>16777216</memory>

  <currentMemory unit='KiB'>16777216</currentMemory>

  <memoryBacking>

    <hugepages>

      <page size='1048576' unit='KiB'/>

    </hugepages>

  </memoryBacking>

  <vcpu placement='static'>6</vcpu>

  <cputune>

    <shares>4096</shares>
```



```
<vcpupin vcpu='0' cpuset='0'/>
<vcpupin vcpu='1' cpuset='2'/>
<vcpupin vcpu='2' cpuset='4'/>
<vcpupin vcpu='3' cpuset='6'/>
<emulatorpin cpuset='0,2,4,6'/>
</cputune>
<numatune>
<memory mode='strict' nodeset='0'/>
</numatune>
<resource>
<partition>/machine</partition>
</resource>
<os>
<type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
<boot dev='hd'/>
</os>
<features>
<acpi/>
<apic/>
</features>
<cpu mode='custom' match='minimum' check='full'>
<model fallback='allow'>Haswell-noTSX</model>
<vendor>Intel</vendor>
<topology sockets='1' cores='6' threads='1'/>
<feature policy='require' name='ss'/>
<feature policy='require' name='pcid'/>
```



```
    <feature policy='require' name='hypervisor'/>

    <feature policy='require' name='arat'/>

<domain type='kvm'>

  <name>dpdk-vpx1</name>

  <uuid>aedb844b-f6bc-48e6-a4c6-36577f2d68d6</uuid>

  <memory unit='KiB'>16777216</memory>

  <currentMemory unit='KiB'>16777216</currentMemory>

  <memoryBacking>

    <hugepages>

      <page size='1048576' unit='KiB'/>

    </hugepages>

  </memoryBacking>

  <vcpu placement='static'>6</vcpu>

  <cputune>

    <shares>4096</shares>

    <vcpupin vcpu='0' cpuset='0'/>

    <vcpupin vcpu='1' cpuset='2'/>

    <vcpupin vcpu='2' cpuset='4'/>

    <vcpupin vcpu='3' cpuset='6'/>

    <emulatorpin cpuset='0,2,4,6'/>

  </cputune>

  <numatune>

    <memory mode='strict' nodeset='0'/>

  </numatune>

  <resource>

    <partition>/machine</partition>

  </resource>
```



```
132
133   <os>
134
135     <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
136
137     <boot dev='hd'/>
138
139   </os>
140
141   <features>
142
143     <acpi/>
144
145     <apic/>
146
147   </features>
148
149   <cpu mode='custom' match='minimum' check='full'>
150
151     <model fallback='allow'>Haswell-noTSX</model>
152
153     <vendor>Intel</vendor>
154
155     <topology sockets='1' cores='6' threads='1'/>
156
157     <feature policy='require' name='ss'/>
158
159     <feature policy='require' name='pcid'/>
160
161     <feature policy='require' name='hypervisor'/>
162
163     <feature policy='require' name='arat'/>
164
165     <feature policy='require' name='tsc_adjust'/>
166
167     <feature policy='require' name='xsaveopt'/>
168
169     <feature policy='require' name='pdpe1gb'/>
170
171     <numa>
172
173       <cell id='0' cpus='0-5' memory='16777216' unit='KiB' memAccess='
          shared'/>
174
175     </numa>
176
177   </cpu>
178
179   <clock offset='utc'/>
180
181   <on_poweroff>destroy</on_poweroff>
182
183   <on_reboot>restart</on_reboot>
```



```
184
185    <on_crash>destroy</on_crash>
186
187    <devices>
188
189      <emulator>/usr/libexec/qemu-kvm</emulator>
190
191      <disk type='file' device='disk'>
192
193        <driver name='qemu' type='qcow2' cache='none'/>
194
195        <source file='/home/NSVPX-KVM-12.0-52.18_nc.qcow2'/>
196
197        <target dev='vda' bus='virtio'/>
198
199        <address type='pci' domain='0x0000' bus='0x00' slot='0x07'
             function='0x0'/>
200
201      </disk>
202
203      <controller type='ide' index='0'>
204
205        <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
             function='0x1'/>
206
207      </controller>
208
209      <controller type='usb' index='0' model='piix3-uhci'>
210
211        <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
             function='0x2'/>
212
213      </controller>
214
215      <controller type='pci' index='0' model='pci-root'/>
216
217      <interface type='direct'>
218
219        <mac address='52:54:00:bb:ac:05'/>
220
221        <source dev='enp129s0f0' mode='bridge'/>
222
223        <model type='virtio'/>
224
225        <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
             function='0x0'/>
226
227      </interface>
228
229      <interface type='vhostuser'>
230
231        <mac address='52:54:00:55:55:56'/>
232
```



```
233         <source type='unix' path='/var/run/openvswitch/vhost-user1' mode=
               'client'/>
234
235         <model type='virtio'/>
236
237         <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
               function='0x0'/>
238
239       </interface>
240
241       <interface type='vhostuser'>
242
243         <mac address='52:54:00:2a:32:64'/>
244
245         <source type='unix' path='/var/run/openvswitch/vhost-user2' mode=
               'client'/>
246
247         <model type='virtio'/>
248
249         <address type='pci' domain='0x0000' bus='0x00' slot='0x05'
               function='0x0'/>
250
251       </interface>
252
253       <interface type='vhostuser'>
254
255         <mac address='52:54:00:2a:32:74'/>
256
257         <source type='unix' path='/var/run/openvswitch/vhost-user3' mode=
               'client'/>
258
259         <model type='virtio'/>
260
261         <address type='pci' domain='0x0000' bus='0x00' slot='0x06'
               function='0x0'/>
262
263       </interface>
264
265       <interface type='vhostuser'>
266
267         <mac address='52:54:00:2a:32:84'/>
268
269         <source type='unix' path='/var/run/openvswitch/vhost-user4' mode=
               'client'/>
270
271        <model type='virtio'/>
272
273         <address type='pci' domain='0x0000' bus='0x00' slot='0x09'
               function='0x0'/>
274
275       </interface>
276
277       <serial type='pty'>
```



```
278
279       <target port='0'/>
280
281     </serial>
282
283     <console type='pty'>
284
285       <target type='serial' port='0'/>
286
287     </console>
288
289     <input type='mouse' bus='ps2'/>
290
291     <input type='keyboard' bus='ps2'/>
292
293     <graphics type='vnc' port='-1' autoport='yes'>
294
295       <listen type='address'/>
296
297     </graphics>
298
299     <video>
300
301       <model type='cirrus' vram='16384' heads='1' primary='yes'/>
302
303       <address type='pci' domain='0x0000' bus='0x00' slot='0x02'
              function='0x0'/>
304
305     </video>
306
307     <memballoon model='virtio'>
308
309       <address type='pci' domain='0x0000' bus='0x00' slot='0x08'
              function='0x0'/>
310
311     </memballoon>
312
313   </devices>
314
315 </domain
316 <!--NeedCopy-->
```

注意事項

XML ファイルでは、サンプルファイルに示されているように、hugepage サイズは 1GB である必要があります。

```
1 <memoryBacking>
2
3     <hugepages>
4
5       <page size='1048576' unit='KiB'/>
6
```



```
    </hugepages>
<!--NeedCopy-->
```

また、サンプルファイルでは、vhost-user1 は ovs-br0 にバインドされたvhostユーザーポートです。

```
<interface type='vhostuser'>

      <mac address='52:54:00:55:55:56'/>

      <source type='unix' path='/var/run/openvswitch/vhost-user1' mode=
         'client'/>

      <model type='virtio'/>

      <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
         function='0x0'/>

    </interface>
<!--NeedCopy-->
```

NetScaler VPX インスタンスを起動するには、virsh コマンドの使用を開始します。

KVM ハイパーバイザーでの NetScaler ADC アプライアンスの初回起動時に NetScaler ADC VPX の構成を適用する

August 15, 2023

NetScaler ADC アプライアンスの初回起動時に、KVM ハイパーバイザーに NetScaler ADC VPX 構成を適用できます。したがって、VPX インスタンスでのお客様のセットアップは、はるかに短時間で構成できます。

プレブートユーザーデータとその形式について詳しくは、「クラウドでの NetScaler ADC アプライアンスの初回起動時に NetScaler ADC VPX 構成を適用する」を参照してください。

> 注:
>
> KVM Hypervisor でプレブートユーザーデータを使用してブートストラップするには、デフォルトのゲートウェイ設定を<NS-CONFIG>セクションに渡す必要があります。<NS-CONFIG> タグの内容について詳しくは、次の「サンプル」<NS-CONFIG> セクションを参照してください。

サンプル**<NS-CONFIG>**セクション**:**

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0  10.102.38.1
    </NS-CONFIG>

```



```
    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP> 10.102.38.216 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

KVM ハイパーバイザーでプリブートユーザーデータを提供する方法

KVM ハイパーバイザー上のプリブートユーザーデータは、CD-ROM デバイスを使用して接続された ISO ファイルを介して提供できます。

CD-ROM ISO ファイルを使用したユーザーデータの提供

バーチャルマシンマネージャー (VMM) を使用すると、CDROM デバイスを使用して ISO イメージとしてバーチャルマシン (VM) にユーザーデータを挿入できます。KVM は、VM ホストサーバー上の物理ドライブに直接アクセスするか、ISO イメージにアクセスして VM ゲストの CD-ROM をサポートします。

CD-ROM ISO ファイルを使用してユーザーデータを指定するには、次の手順に従います。

1. プレブートユーザーデータコンテンツを含むファイル名`userdata`でファイルを作成します。

   > 注: ファイル名は、`userdata`として厳密に使用する必要があります。

2. `userdata`ファイルをフォルダーに保存し、フォルダーを使用して ISO イメージを構築します。

   `userdata`ファイルを使用して ISO イメージを構築するには、次の 2 つの方法があります。

   - PowerISO などの任意の画像処理ツールを使用します。
   - Linux `mkisofs` でコマンドを使う。

   次の設定例は、Linux `mkisofs` でコマンドを使用して ISO イメージを生成する方法を示しています。

```
root@ubuntu:~/sai/19oct# ls -lh
total 4.0K
-rw-r--r-- 1 root root 1.1K Oct 19 16:25 userdata
root@ubuntu:~/sai/19oct#
root@ubuntu:~/sai/19oct# mkisofs -o kvm-userdata.iso userdata
I: -input-charset not specified, using utf-8 (detected in locale
   settings)
Total translation table size: 0
```



```
Total rockridge attributes bytes: 0
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 0
175 extents written (0 MB)
root@ubuntu:~/sai/19oct#
root@ubuntu:~/sai/19oct# ls -lh
total 356K
-rw-r--r-- 1 root root 350K Oct 19 16:25 kvm-userdata.iso
-rw-r--r-- 1 root root 1.1K Oct 19 16:25 userdata
<!--NeedCopy-->
```

3. 標準の展開プロセスを使用して NetScaler ADC VPX インスタンスをプロビジョニングし、仮想マシンを作成します。ただし、VM の電源を自動的にオンにしないでください。
4. 仮想マシンマネージャーで CD-ROM デバイスを追加するには、次の手順に従います。
   a) 仮想マシンマネージャで VM ゲストエントリをダブルクリックしてコンソールを開き、[表示] **> [**詳細**]** の順に選択して **[**詳細**]** ビューに切り替えます。
   b) [ハードウェアの追加**] > [**ストレージ**] > [**デバイスの種類**] > [CDROM** デバイス] をクリックします。
   c) [管理] をクリックして正しい ISO ファイルを選択し、[完了] をクリックします。NetScaler VPX インスタンスの「リソース」の下に新しい CDROM が作成されます。
5. VM の電源を入れます。

AWS での NetScaler VPX

March 20, 2024

NetScaler VPX インスタンスは、Amazon Web Services (AWS) で起動できます。NetScaler VPX アプライアンスは、AWS マーケットプレイスで Amazon Machine Image (AMI) として利用できます。AWS 上の NetScaler VPX インスタンスを使用すると、AWS のクラウドコンピューティング機能を使用したり、NetScaler の負荷分散機能とトラフィック管理機能をビジネスニーズに合わせて使用したりできます。VPX インスタンスは、物理 NetScaler アプライアンスのすべてのトラフィック管理機能をサポートし、スタンドアロンインスタンスまたは HA ペアとして展開できます。VPX の機能の詳細については、VPX のデータシートを参照してください。

はじめに

VPX のデプロイを開始する前に、次の情報を理解しておく必要があります。

- AWS の用語
- AWS-VPX のサポートマトリックス
- 制限事項と使用ガイドライン

- Prerequisites
- AWS 上の NetScaler VPX インスタンスの仕組み

AWS で NetScaler VPX インスタンスを展開する

AWS では、VPX インスタンスで次のデプロイタイプがサポートされています。

- Standalone
- 高可用性 (アクティブ-パッシブ)
  - 同一ゾーン内での高可用性
  - Elastic IP を使用した異なるゾーンでの高可用性
  - プライベート IP を使用して、異なるゾーン間で高可用性
- アクティブ-アクティブ GSLB
- ADM を使用した自動スケーリング (アクティブ-アクティブ)

ハイブリッド展開

- NetScaler を AWS アウトポストにデプロイ
- AWS の VMC に NetScaler をデプロイする

ライセンス

AWS 上の NetScaler VPX インスタンスにはライセンスが必要です。AWS で実行されている NetScaler VPX インスタンスでは、次のライセンスオプションを使用できます。

- 無料 (無制限)
- 毎時
- 年次
- BYOL
- 無料トライアル（AWS マーケットプレイスでは、すべての NetScaler VPX-AWS サブスクリプションを 21 日間無料で提供）

自動化

- NetScaler コンソール: スマートデプロイ
- AWS クイックスタート:AWS 上のウェブアプリケーション用 NetScaler VPX
- GitHub CFT: AWS デプロイ用の NetScaler テンプレートとスクリプト

- GitHub Ansible: AWS デプロイ用の NetScaler テンプレートとスクリプト
- GitHub Terraform: AWS デプロイ用の NetScaler テンプレートとスクリプト
- AWS パターンライブラリ (PL): NetScaler VPX

ブログ

- NetScaler on AWS が顧客によるアプリケーションの安全な配信をどのように支援するか
- NetScaler と AWS によるハイブリッドクラウドでのアプリケーション配信
- Citrix は AWS ネットワーキングコンピテンシーパートナーです
- NetScaler: いつでもパブリッククラウドに対応
- NetScaler を使用してパブリッククラウドで簡単にスケールアウトまたはスケールインできます
- Citrix、AWS Outposts で ADC のデプロイメントの選択肢を拡大
- NetScaler と Amazon VPC イングレスルーティングの使用
- Citrix は、AWS での選択肢、パフォーマンス、シンプルなデプロイメントを提供します
- NetScaler Web App Firewall のセキュリティ—現在 AWS Marketplace で公開中
- Aria Systems が AWS で NetScaler Web App Firewall を使用する方法

ビデオ

- ADM によるパブリッククラウドの NetScaler 導入の簡素化
- すぐに使用できるテラフォームスクリプトを使用して AWS で NetScaler VPX を Provisioning および構成する
- クラウドフォーメーションテンプレートを使用して NetScaler HA を AWS にデプロイ
- AWS クイックスタートを使用してアベイラビリティーゾーン全体に NetScaler HA
- AWS で NetScaler をデプロイする方法
- ADM を使用した NetScaler オートスケール
- AWS または AWS 自動スケーリンググループでのバックエンドサーバーの自動スケーリングをサポートする NetScaler

お客様のケーススタディ

- テクノロジーソリューション-Xenit AB

- Citrix と AWS クラウドとのより良いビジネス方法—Aria
- NetScaler と AWS の優位性をご覧ください
- Rain for Rent-お客様事件

解決方法

- NetScaler を使用して AWS にデジタル広告プラットフォームをデプロイする
- NetScaler による AWS でのクリックストリーム分析の強化

サポート

- サポートケースを開く
- NetScaler サブスクリプションサービスについては、「AWS での VPX インスタンスのトラブルシューティング」を参照してください。サポートケースを提出するには、AWS アカウント番号とサポート PIN コードを見つけて、NetScaler サポートに連絡してください。
- NetScaler カスタマーライセンスサービスまたは BYOL の場合は、有効なサポートおよびメンテナンス契約を結んでいることを確認してください。契約を結んでいない場合は、NetScaler の担当者にお問い合わせください。

その他の参考資料

- AWS オンデマンドウェビナー-AWS 上の NetScaler
- AWS での NetScaler VPX デプロイガイド
- SC2S/シークレットリージョンでの VPX Amazon Machine Image (AMI) の作成
- AWS 上の NetScaler
- NetScaler VPX データシート
- AWS Marketplace の NetScaler
- NetScaler は、AWS ネットワーキングパートナーソリューション（ロードバランサー）の一部です。
- AWS 上の VMware クラウド向け NetScaler
- AWS に関するよくある質問

AWS の用語

August 15, 2023

このセクションでは、よく使用される AWS の用語と語句のリストについて説明します。詳細については、「AWS 用語集」を参照してください。

用語	定義
Amazon マシンイメージ（AMI）	マシンイメージ。クラウド内の仮想サーバーであるインスタンスを起動するのに必要な情報を提供します。
Elastic Block Store	AWS クラウドで Amazon EC2 インスタンスと一緒に使用される、永続ブロックストレージボリュームを提供します。
Simple Storage Service（S3）	Internet 用のストレージ。Web 規模のコンピューティングを開発者が簡単に実施できるように設計されています。
Elastic Compute Cloud（EC2）	クラウドで、安全でサイズ変更できる処理能力を提供する Web サービスです。Web 規模のクラウドコンピューティングを開発者が簡単に実施できるように設計されています。
Elastic Load Balancing（ELB）	複数のアベイラビリティゾーンで、複数の EC2 インスタンスにまたがる受信アプリケーショントラフィックを分散します。これによってアプリケーションのフォールトトレランスが増加します。
エラスティックネットワークインターフェイス (ENI)	仮想プライベートクラウド (VPC) のインスタンスにアタッチできる仮想ネットワークインターフェイス。
Elastic IP（EIP）アドレス	Amazon EC2 または Amazon VPC で割り当てられ、インスタンスにアタッチされた、静的パブリック IPv4 アドレスです。Elastic IP アドレスは特定のインスタンスではなく、お使いのアカウントに関連しています。ニーズの変化に応じて、割り当て、アタッチ、デタッチ、および解放が簡単にできるため、Elastic（融通が利く）と呼ばれています。
インスタンスタイプ	Amazon EC2 では、さまざまなユースケースに対応できるよう最適化された幅広い種類のインスタンスを提供してます。インスタンスタイプを構成する CPU、メモリ、ストレージ、およびネットワーク機能の組み合わせはさまざまで、アプリケーションに合わせて最適なリソースの組み合わせを柔軟に選択できます。

用語	定義
Identity and Access Management（IAM）	AWS で ID が実行できること、または実行できないことを決定する許可ポリシーを持つ AWS の ID。IAM ロールを使うことで EC2 インスタンス上で実行されるアプリケーションが、AWS リソースに安全にアクセスできるようになります。高可用性セットアップで VPX インスタンスを展開する場合、IAM ロールは必須です。
インターネットゲートウェイ	ネットワークをインターネットに接続します。VPC 外部の IP アドレスのトラフィックをインターネットゲートウェイにルーティングできます。
キーペア	身元を電子的に証明するために使用する一連の資格情報。キーペアはプライベートキーとパブリックキーで構成されます。
ルートテーブル	関連付けられているサブネットからのトラフィックを制御するための一連のルーティング規則。1 つのルートテーブルに対して複数のサブネットを関連付けることができますが、各サブネットは一度に 1 つのルートテーブルにしか関連付けることができません。
セキュリティグループ	あるインスタンスに対して許可されている、名前が付けられた一連の受信方向のネットワーク接続。
サブネット	EC2 インスタンスをアタッチできる VPC の IP アドレス範囲の一部分。セキュリティと運用上の必要に応じて、サブネットを作成し、インスタンスをグループ分けできます。
Virtual Private Cloud（VPC）	定義した仮想ネットワーク内で AWS リソースを起動できる、AWS クラウドの論理的に隔離されたセクションをプロビジョニングする Web サービス。
Auto Scaling	ユーザー定義のポリシー、スケジュール、ヘルスチェックに基づいて Amazon EC2 インスタンスを自動的に起動または終了するウェブサービス。
クラウドの形成	関連する AWS リソースを 1 つの単位として一緒に作成および削除するテンプレートを書き込んだり変更したりするサービス。

AWS-VPX のサポートマトリックス

March 20, 2024

次の表に、サポートされている VPX モデルと AWS リージョン、インスタンスタイプ、およびサービスを示します。

表 1: AWS でサポートされている VPX モデル

サポートされている VPX モデル
NetScaler VPX スタンダード/アドバンスト/プレミアムエディション-200 Mbps
NetScaler VPX スタンダード/アドバンスト/プレミアムエディション-1000 Mbps
NetScaler VPX スタンダード/アドバンスト/プレミアムエディション-3 Gbps
NetScaler VPX スタンダード/アドバンスト/プレミアムエディション-5 Gbps
NetScaler VPX スタンダード/アドバンスト/プレミアム-10 Mbps
NetScaler VPX エクスプレス-20 Mbps
NetScaler VPX-カスタマーライセンス
NetScaler（旧 NetScaler）VPX FIPS-カスタマーライセンス

表: サポートされている 2 つの AWS リージョン

サポートされている AWS リージョン
米国西部 (オレゴン)
米国西部 (北カリフォルニア)
米国東部 (オハイオ)
米国東部 (バージニア北部)
アジアパシフィック (ムンバイ)
アジアパシフィック (ソウル)
アジアパシフィック (シンガポール)
アジアパシフィック (シドニー)
アジアパシフィック (東京)
アジアパシフィック (香港)
アジアパシフィック (大阪)

サポートされている AWS リージョン
アジアパシフィック (ジャカルタ)
アジアパシフィック (ハイデラバード)
カナダ (中部)
EU (フランクフルト)
欧州 (アイルランド)
欧州 (ロンドン)
欧州 (パリ)
欧州 (ミラノ)
南米 (サンパウロ)
AWS GovCloud (米国東部)
AWS GovCloud (米国西部)
AWS トップシークレット (C2S)
中東 (バーレーン)
アフリカ (ケープタウン)
C2S

> 注:
>
> AWS 香港リージョンでは、NetScaler VPX サポートは BYOL ライセンスでのみ利用可能です。

表 3: サポートされている AWS インスタンスタイプ

サポートされる AWS インスタンスタイプ
c4.large、c4.xlarge、c4.2xlarge、c4.4xlarge、c4.8xlarge
c5.large、c5.xlarge、c5.2xlarge、c5.4xlarge、c5.9xlarge、c5.18xlarge、c5.24xlarge
c5n.large、c5n.xlarge、c5n.2xlarge、c5n.4xlarge、c5n.9xlarge、c5n.18xlarge
奥行 2.XL サイズ、D2.2xL サイズ、D2.4xL サイズ、D2.8xL サイズ
m3.large, m3.xlarge, m3.2xlarge
m4.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m4.10xlarge, m4.16xlarge
m5.large, m5.xlarge, m5.2xlarge, m5.4xlarge, m5.8xlarge, m5.12xlarge, m5.16xlarge, m5.24xlarge

サポートされる AWS インスタンスタイプ
m5a.large, m5a.xlarge, m5a.2xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge
m6i.large, m6i.xlarge, m6i.2xlarge, m6i.4xlarge, m6i.8xlarge, m6i.12xlarge, m6i.16xlarge, m6i.24xlarge, m6i.32xlarge
r7iz.large、r7iz.xlarge、r7iz.2xlarge、r7iz.4xlarge、r7iz.8xlarge、r7iz.12xlarge、r7iz.16xlarge、r7iz.32xlarge
t2.medium, t2.large, t2.xlarge, t2.2xlarge
t3a.medium, t3a.large, t3a.xlarge, t3a.2xlarge

表 4: サポートされる AWS サービス

サポートされている AWS サービス
EC2: ADC インスタンスを起動します。
ラムダ： CFT からの NetScaler VPX インスタンスのプロビジョニング中に、NetScaler VPX NITRO API を呼び出します。
VPC と **VPC** イングレスルーティング**:** VPC は、ADC を起動できる分離されたネットワークを作成します。VPC 入力ルーティング
Route53: NetScaler Autoscale e ソリューション内のすべての NetScaler VPX ノードにトラフィックを分散します。
ELB: NetScaler Autoscale e ソリューション内のすべての NetScaler VPX ノードにトラフィックを分散します。
Cloudwatch： NetScaler VPX インスタンスのパフォーマンスとシステムパラメーターを監視します。
AWS Autoscaling: バックエンドサーバーの自動スケーリングに使用されます。
クラウドの形成： CloudFormation テンプレートは、NetScaler VPX インスタンスをデプロイするために使用されます。
Simple Queue Service (SQS): バックエンド自動スケーリングでスケールアップおよびスケールダウンイベントを監視します。
簡易通知サービス **(SNS):** バックエンド自動スケーリングでスケールアップおよびスケールダウンイベントを監視します。
ID とアクセス管理 **(IAM):** AWS のサービスとリソースへのアクセスを提供します。
AWS Outposts: AWS Outposts で NetScaler VPX インスタンスをプロビジョニングします。

Citrix では、次の AWS インスタンスタイプを推奨します。

- マーケットプレイスエディションまたは帯域幅ベースのプールライセンス用の M5 および C5n シリーズ
- vCPU ベースのプールライセンス用の C5n シリーズ

AWS マーケットプレイスでの VPX オファリング	AWS インスタンスの推奨事項
VPX 10、VPX エクスプレス 20、VPX 200	M5.xLarge
VPX 1000、VPX 3G、VPX 5G	M5.2xLarge

Citrix では、スループットに基づいて以下の AWS インスタンスタイプを推奨します。

プールライセンス付き VPX（帯域幅ライセンス）	AWS インスタンスの推奨事項
VPX 8G	C5n.4xLarge
VPX 10G、VPX 15G、VPX 25G	C5n.9xLarge

> 注:
>
> VPX 25G サービスでは、AWS で期待される 25G のスループットは得られませんが、SSL トランザクションレートは高くなります。

5G を超えるスループットを実現するには、次の手順を実行します:

- **AWS** マーケットプレイスで提供されている **NetScaler VPX-**カスタマーライセンス（**BYOL**）サービスを選択してください。
- NetScaler GUI または CLI で［プールライセンス（帯域幅ライセンス)］を選択します。

1 秒あたりのパケット数、SSL トランザクションレートなどのさまざまなメトリックに基づいてインスタンスを判断するには、Citrix 連絡先に連絡してガイダンスを求めてください。vCPU ベースのプールライセンスとサイジングのガイダンスについては、NetScaler サポートにお問い合わせください。

制限事項と使用ガイドライン

August 15, 2023

NetScaler VPX インスタンスを AWS にデプロイする際には、以下の制限事項と使用上のガイドラインが適用されます。

- 開始する前に、AWS での NetScaler ADC VPX インスタンスのデプロイの AWS 用語のセクションをお読みください。
- クラスタリング機能は、VPX ではサポートされていません。

- 高可用性セットアップを効果的に機能させるには、専用の NAT デバイスを管理インターフェイスに関連付けるか、EIP を NSIP に関連付けます。NAT について詳しくは、AWS ドキュメントの「NAT Instances」を参照してください。
- データトラフィックおよび管理トラフィックは、異なるサブネットに属する ENI で分離する必要があります。
- 管理 ENI には NSIP アドレスのみが必要です。
- セキュリティ上の理由により、EIP を NSIP に関連付ける代わりに NAT インスタンスを使用する場合は、VPC レベルでルーティングを適切に変更する必要があります。VPC レベルのルーティングの変更手順については、AWS ドキュメントの「シナリオ 2: パブリックサブネットとプライベートサブネットを持つ VPC」を参照してください。
- VPX インスタンスは、ある EC2 インスタンスタイプから別のインスタンスタイプへ（たとえば、m3.large から m3.xlarge へ）移動できます。
- AWS 上の VPX のストレージオプションについては、EBS は耐久性があり、インスタンスからデタッチした後でもデータが利用可能になるため、EBS をお勧めします。
- VPX への ENI の動的追加はサポートされていません。VPX インスタンスを再起動して更新を適用します。スタンドアロンインスタンスまたは HA インスタンスを停止し、新しい ENI を接続してからインスタンスを再起動することをお勧めします。
- 1 つの ENI に複数の IP アドレスを割り当てることができます。ENI あたりの IP アドレスの最大数は EC2 インスタンスタイプによって決まります。Elastic Network Interfacesの「インスタンスタイプごとのネットワークインターフェイスごとの IP アドレス」のセクションを参照してください。IP アドレスを ENI に割り当てる前に、AWS で割り当てる必要があります。詳細については、「Elastic ネットワークインターフェイス」を参照してください。
- NetScaler VPX インターフェイスでは、インターフェイスの有効化および無効化コマンドは使用しないことをお勧めします。
- NetScaler`set ha node \<NODE\_ID\> -haStatus STAYPRIMARY` と`set ha node \<NODE\_ID\> -haStatus STAYSECONDARY` コマンドはデフォルトで無効になっています。
- IPv6 は VPX ではサポートされていません。
- AWS の制限により、次の機能はサポートされていません。
  - GARP（Gratuitous ARP）
  - L2 モード
  - タグ付き VLAN
  - 動的ルーティング
  - 仮想 MAC
- RNAT が機能するには、送信元/宛先チェック が無効になっていることを確認します。詳細については、Elastic Network Interfacesの「ソース/デスティネーションチェックの変更」を参照してください。

- AWS での NetScaler VPX デプロイメントでは、一部の AWS リージョンで AWS インフラストラクチャが AWS API 呼び出しを解決できない場合があります。これは、API 呼び出しが NetScaler VPX インスタンスの非管理インターフェイスを介して発行された場合に発生します。
  回避策として、API 呼び出しを管理インターフェイスにのみ制限してください。これを行うには、VPX インスタンスに NSVLAN を作成し、適切なコマンドを使用して管理インターフェイスを NSVLAN にバインドします。
  例:
  ```
  set ns config -nsvlan <vlan id> -ifnum 1/1 -tagged NO
  save config
  ```
  プロンプトで VPX インスタンスを再起動します。nsvlanの設定の詳細については、NSVLAN の設定を参照してください。
- AWS コンソールでは、実際の使用量がはるかに低い場合でも、監視タブに表示される **VPX** インスタンスの **vCPU** 使用率が高い場合があります（最大 100%）。実際の vCPU 使用率を確認するには、「すべての **CloudWatch** メトリックスを表示」に移動します。詳細については、「Amazon CloudWatch を使用してインスタンスを監視する」を参照してください。

前提条件

August 29, 2023

AWS で VPX インスタンスを作成する前に、次のものがあることを確認してください。

- **AWS** アカウント:AWS 仮想プライベートクラウド (VPC) で NetScaler VPX AMI を起動します。AWS アカウントは www.aws.amazon.com で無料で作成できます。
- **AWS ID** およびアクセス管理（**IAM**）ユーザーアカウント: ユーザーの AWS サービスおよびリソースへのアクセスを安全にコントロールします。IAM ユーザーアカウントの作成方法の詳細については、「IAM ユーザーの作成 (コンソール)」を参照してください。IAM ロールは、スタンドアロンデプロイと高可用性デプロイの両方で必須です。

  AWS アカウントに関連付けられた IAM ロールには、さまざまなシナリオで次の IAM アクセス権限が必要です。

  同じ **AWS** ゾーン内の **IPv4** アドレスと **HA** ペア**:**

  ```
  "ec2:DescribeInstances",
  "ec2:AssignPrivateIpAddresses",
  "iam:SimulatePrincipalPolicy",
  "iam:GetRole",
  "ec2:CreateTags"
  <!--NeedCopy-->
  ```

  同じ **AWS** ゾーン内の **IPv6** アドレスと **HA** ペア**:**


```
"ec2:DescribeInstances",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
<!--NeedCopy-->
```

同じ **AWS** ゾーン内の **IPv4** と **IPv6** の両方のアドレスとの **HA** ペア**:**

```
"ec2:DescribeInstances",
"ec2:AssignPrivateIpAddresses",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
<!--NeedCopy-->
```

異なる **AWS** ゾーンにまたがる **Elastic IP** アドレスを持つ **HA**:

```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
<!--NeedCopy-->
```

異なる **AWS** ゾーンのプライベート **IP** アドレスを持つ **HA** ペア**:**

```
"ec2:DescribeInstances",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
<!--NeedCopy-->
```

異なる **AWS** ゾーンのプライベート **IP** アドレスと **Elastic IP** アドレスの両方を持つ **HA** ペア**:**

```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
```



```
"iam:GetRole",
"ec2:CreateTags"
<!--NeedCopy-->
```

AWS バックエンドの自動スケーリング**:**

```
"ec2:DescribeInstances",
"autoscaling:*",
"sns:CreateTopic",
"sns:DeleteTopic",
"sns:ListTopics",
"sns:Subscribe",
"sqs:CreateQueue",
"sqs:ListQueues",
"sqs:DeleteMessage",
"sqs:GetQueueAttributes",
"sqs:SetQueueAttributes",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
<!--NeedCopy-->
```

> 注:
>
> - 前述の機能を組み合わせて使用する場合は、各機能に IAM アクセス権限を組み合わせて使用します。
> - Citrix CloudFormation テンプレートを使用すると、IAM ロールが自動的に作成されます。このテンプレートでは、作成済みの IAM ロールを選択することはできません。
> - GUI から VPX インスタンスにログオンすると、IAM ロールに必要な権限を設定するよう求めるプロンプトが表示されます。権限をすでに構成している場合は、このプロンプトを無視してください。

- **AWS CLI**: ターミナルプログラムから AWS マネジメントコンソールが提供するすべての機能を使用する。詳細については、AWS CLI ユーザーガイドを参照してください。また、ネットワークインターフェイスの種類を SR-IOV に変更するには、AWS CLI も必要です。
- **Elastic Network Adapter (ENA)**: M5、C5 インスタンスなどの ENA ドライバー対応インスタンスタイプの場合、ファームウェアバージョンは 13.0 以降である必要があります。
- NetScaler VPX の EC2 インスタンスでインスタンスメタデータサービス（IMDS）を構成する必要があります。IMDSv1 と IMDSv2 は、実行中の AWS EC2 インスタンスからインスタンスメタデータにアクセスするための 2 つのモードです。IMDSv2 は IMDSv1 よりも安全です。インスタンスを両方の方法（デフォルトオプション）を使用するように構成することも、IMDSv2 モードのみを使用するように構成することもできます（IMDSv1 を無効にする）。Citrix ADC VPX は、NetScaler VPX リリース 13.1.48.x 以降の IMDSv2 専用モードをサポートしています。

NetScaler VPX インスタンスで AWS IAM ロールを設定します

August 16, 2023

Amazon EC2 インスタンスで実行されるアプリケーションには、AWS API リクエストに AWS 認証情報を含める必要があります。AWS 認証情報を Amazon EC2 インスタンス内に直接保存し、そのインスタンス内のアプリケーションがそれらの認証情報を使用できるようにすることができます。ただし、認証情報を管理し、認証情報が各インスタンスに安全に渡されるようにし、認証情報をローテーションするときに各 Amazon EC2 インスタンスを更新する必要があります。それは多くの追加作業です。

代わりに、Amazon EC2 インスタンスで実行されるアプリケーションの一時的な認証情報を管理するには、ID とアクセス管理 (IAM) ロールを使用することができ、また使用する必要があります。ロールを使用すると、長期にわたる認証情報 (ユーザー名、パスワード、アクセスキーなど) を Amazon EC2 インスタンスに配布する必要はありません。代わりに、ロールはアプリケーションが他の AWS リソースを呼び出すときに使用できる一時的なアクセス権限を提供します。Amazon EC2 インスタンスを起動するときに、インスタンスに関連付ける IAM ロールを指定します。インスタンスで実行されるアプリケーションは、ロールが提供した一時的な認証情報を使用して API リクエストに署名できます。

AWS アカウントに関連付けられた IAM ロールには、さまざまなシナリオで次の IAM アクセス権限が必要です。

同じ **AWS** ゾーン内の **IPv4** アドレスと **HA** ペア**:**

```
"ec2:DescribeInstances",
"ec2:AssignPrivateIpAddresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

同じ **AWS** ゾーン内の **IPv6** アドレスと **HA** ペア**:**

```
"ec2:DescribeInstances",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

同じ **AWS** ゾーン内の **IPv4** と **IPv6** の両方のアドレスとの **HA** ペア**:**

```
"ec2:DescribeInstances",
"ec2:AssignPrivateIpAddresses",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

異なる **AWS** ゾーンにまたがる **Elastic IP** アドレスを持つ **HA**:


```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

異なる **AWS** ゾーンのプライベート **IP** アドレスを持つ **HA** ペア**:**

```
"ec2:DescribeInstances",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

異なる **AWS** ゾーンのプライベート **IP** アドレスと **Elastic IP** アドレスの両方を持つ **HA** ペア**:**

```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

AWS バックエンドの自動スケーリング**:**

```
"ec2:DescribeInstances",
"autoscaling:*",
"sns:CreateTopic",
"sns:DeleteTopic",
"sns:ListTopics",
"sns:Subscribe",
"sqs:CreateQueue",
"sqs:ListQueues",
"sqs:DeleteMessage",
"sqs:GetQueueAttributes",
"sqs:SetQueueAttributes",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

注意事項:

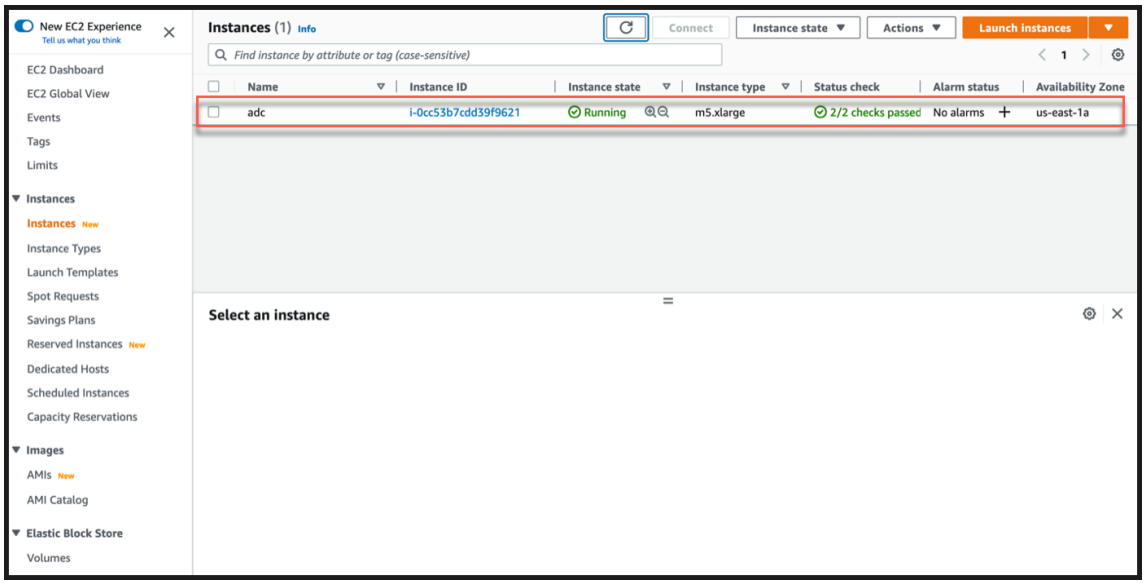
- 前述の機能を組み合わせて使用する場合は、各機能に IAM アクセス権限を組み合わせて使用します。
- Citrix CloudFormation テンプレートを使用すると、IAM ロールが自動的に作成されます。このテンプレートでは、作成済みの IAM ロールを選択することはできません。
- GUI から VPX インスタンスにログオンすると、IAM ロールに必要な権限を設定するよう求めるプロンプトが表示されます。権限をすでに構成している場合は、このプロンプトを無視してください。
- IAM ロールは、スタンドアロンデプロイと高可用性デプロイの両方で必須です。

IAM 役割を作成する

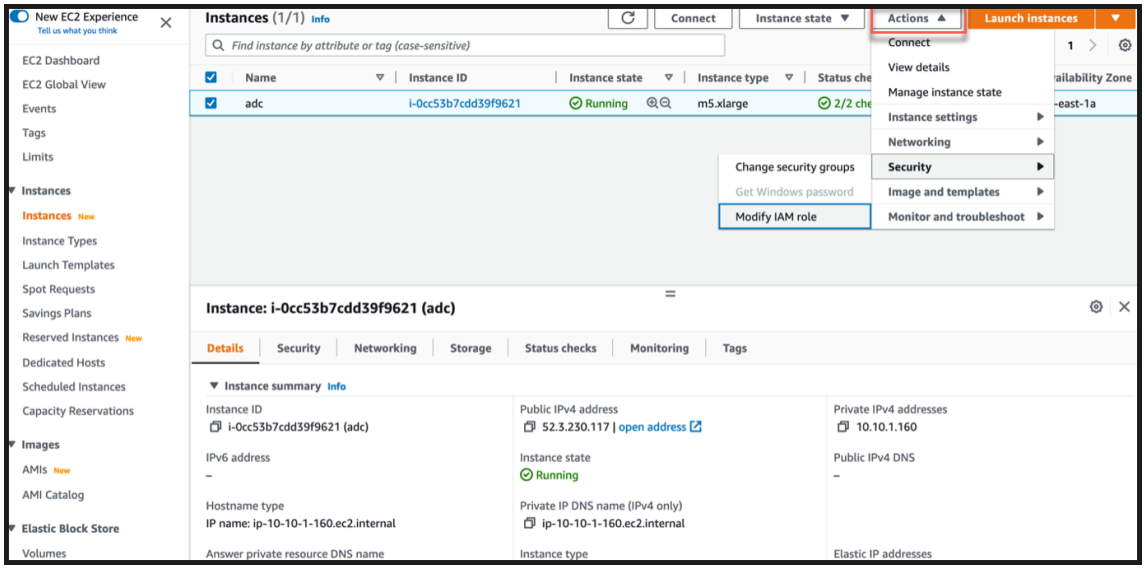
この手順では、AWS バックエンド自動スケーリング機能の IAM ロールを作成する方法について説明します。

> 注:
>
> 同じ手順に従って、他の機能に対応する任意の IAM ロールを作成できます。

1. EC2 用 AWS マネジメントコンソールにログインします。
2. EC2 インスタンスページに移動し、ADC インスタンスを選択します。

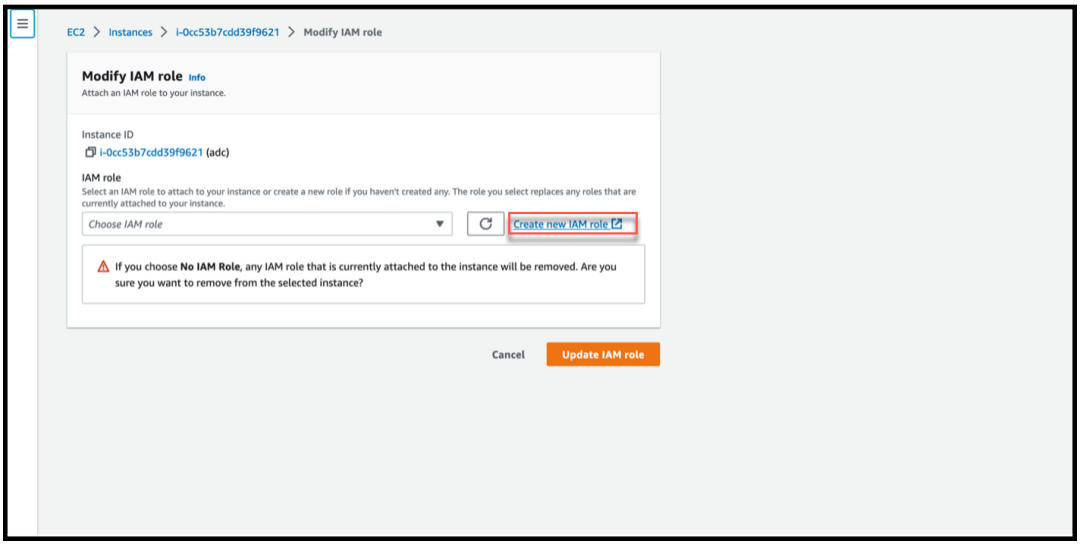


3. [アクション] **>** **[**セキュリティ**] > [IAM** ロールの変更**]** に移動します。

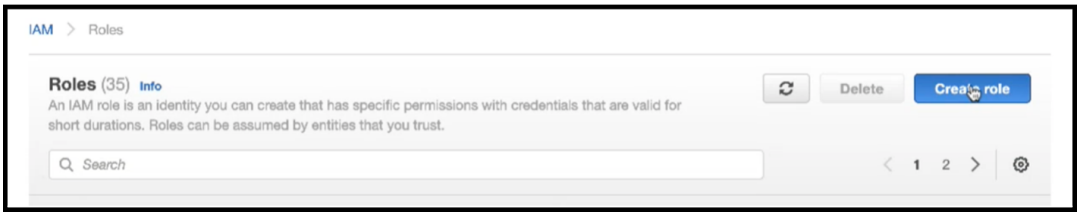


4. **IAM** ロールの変更ページでは、既存の IAM ロールを選択するか、IAM ロールを作成できます。
5. IAM ロールを作成するには、次の手順に従います。

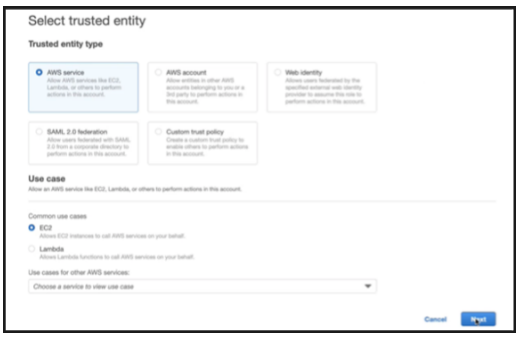
   a)「**IAM** ロールの変更」ページで、「新しい **IAM** ロールを作成」をクリックします。



   b)「ロール」ページで、「ロールを作成」をクリックします。



   c) [信頼できるエンティティタイプ] で [**AWS service**] を選択し、[一般的な使用例] で [**EC2**] を選択し、[次へ] をクリックします。



d)「権限の追加」ページで、「ポリシーの作成」をクリックします。

e) **JSON** タブをクリックして JSON エディターを開きます。

f) JSON エディターで、すべてを削除し、使用したい機能の IAM 権限を貼り付けます。

たとえば、AWS バックエンド自動スケーリング機能用の次の IAM アクセス権限を貼り付けます。

```
{

    "Version": "2012-10-17",
    "Statement": [
        {

            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "autoscaling:\*",
```



```
                    "sns:CreateTopic",
                    "sns:DeleteTopic",
                    "sns:ListTopics",
                    "sns:Subscribe",
                    "sqs:CreateQueue",
                    "sqs:ListQueues",
                    "sqs:DeleteMessage",
                    "sqs:GetQueueAttributes",
                    "sqs:SetQueueAttributes",
                    "iam:SimulatePrincipalPolicy",
                    "iam:GetRole"
                ],
                "Resource": "\*"
            }

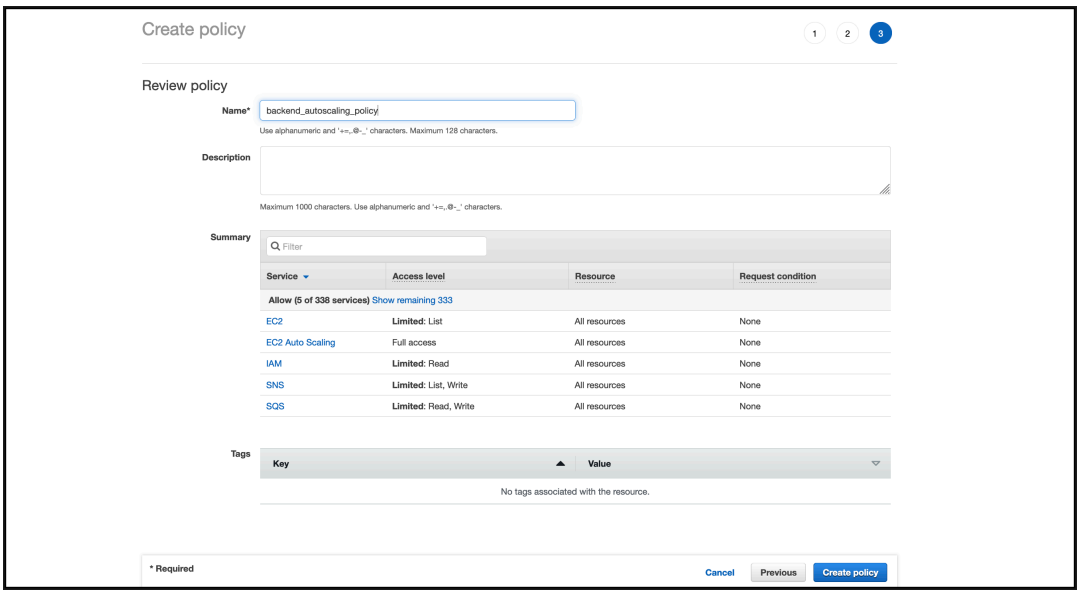
        ]
    }


<!--NeedCopy-->
```

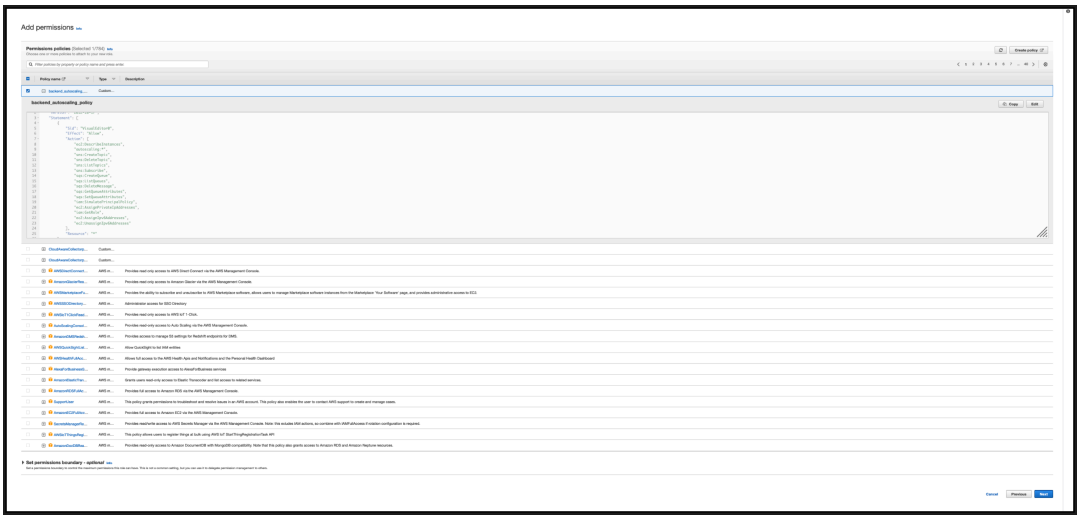
指定する「バージョン」キーと値のペアが、AWS によって自動的に生成されるものと同じであることを確認してください。

g) [次へ**:** 確認] をクリックします。

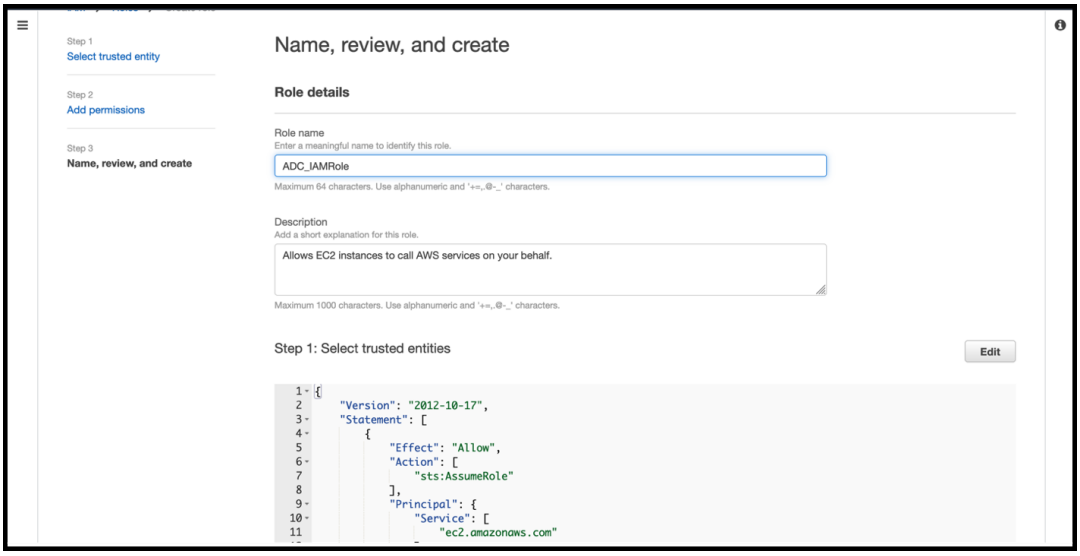
h) [ポリシーの確認] タブで、ポリシーに有効な名前を付けて、[ポリシーの作成] をクリックします。



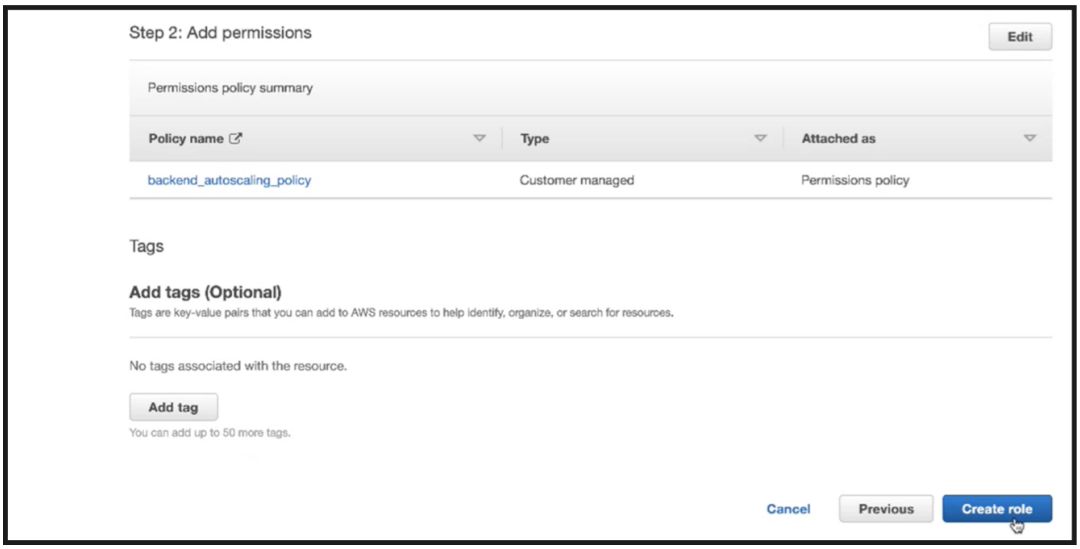
i) **ID** アクセス管理ページで、作成したポリシー名をクリックします。ポリシーを展開して JSON 全体を確認し、[次へ] をクリックします。



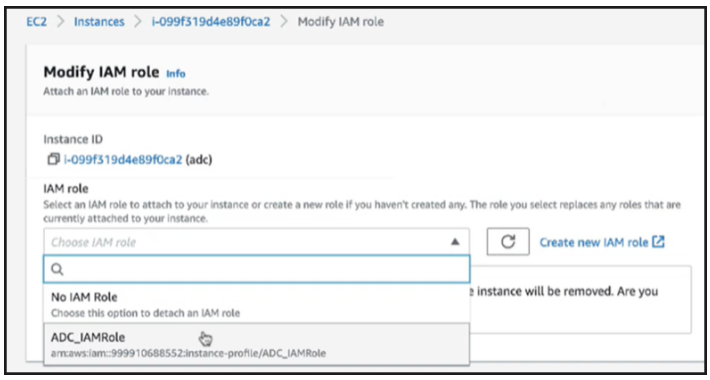
j)「名前、レビュー、作成」ページで、ロールに有効な名前を付けます。



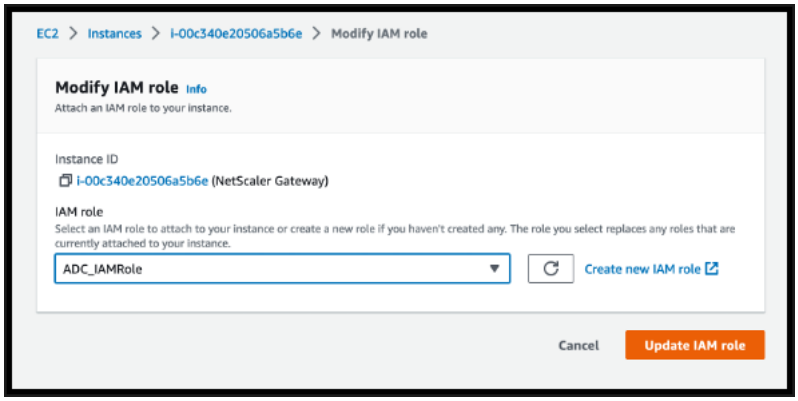
k)「ロールを作成」をクリックします。



6. 手順 1、2、3 を繰り返します。[更新] ボタンを選択し、ドロップダウンメニューを選択すると、作成したロールが表示されます。



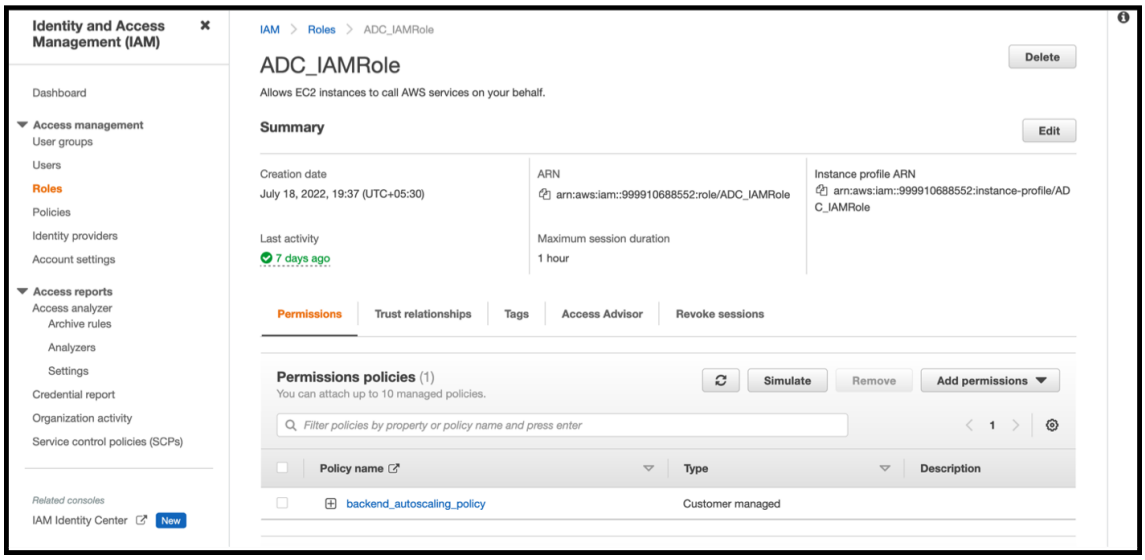
7.「**IAM** ロールを更新」をクリックします。



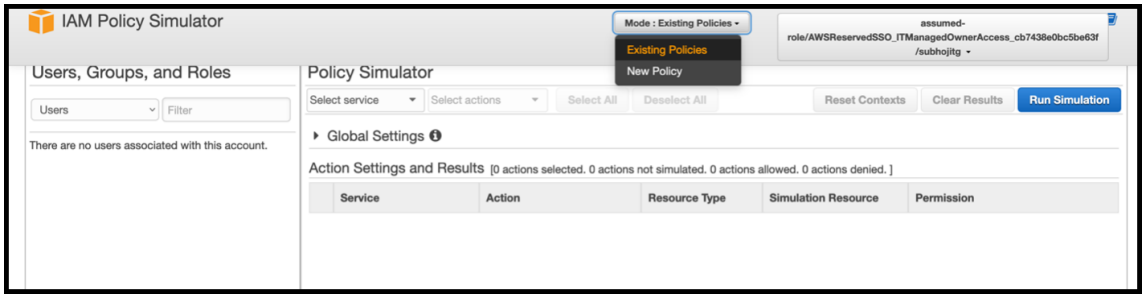
IAM ポリシーシミュレーターで **IAM** ポリシーをテストする

IAM ポリシーシミュレーターは、IAM アクセスコントロールポリシーを実稼働環境に導入する前にその効果をテストできるツールです。権限の確認とトラブルシューティングが簡単になります。

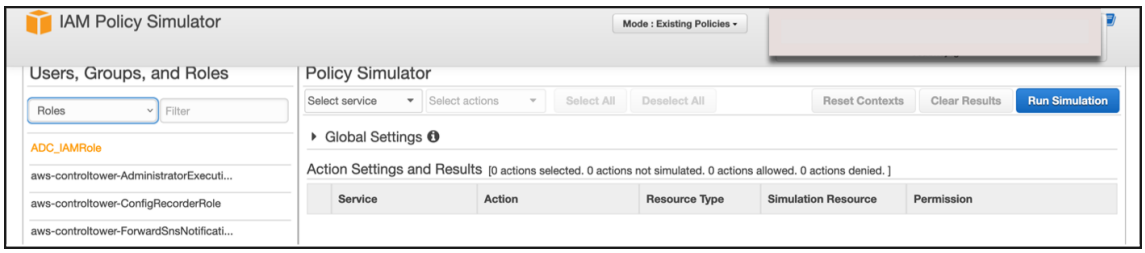
1. **IAM** ページで、テストする IAM ロールを選択し、「**Simulate**」をクリックします。次の例では、「ADC_IAMRole」が IAM ロールです。



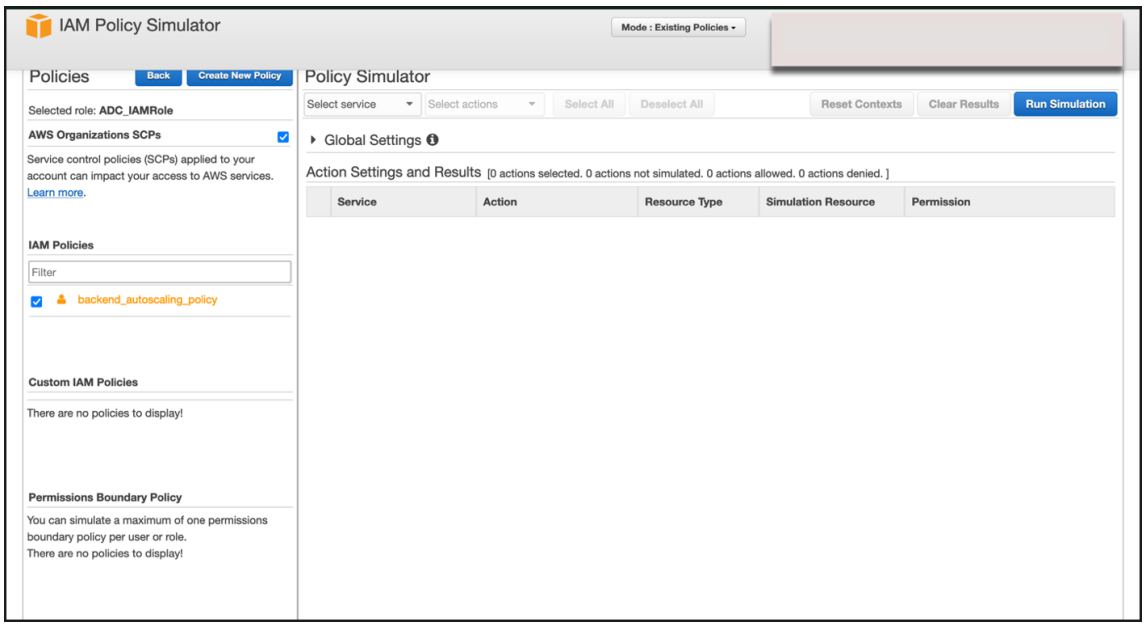
2. **IAM** ポリシーシミュレーターコンソールで、「モード」として「既存のポリシー」を選択します。



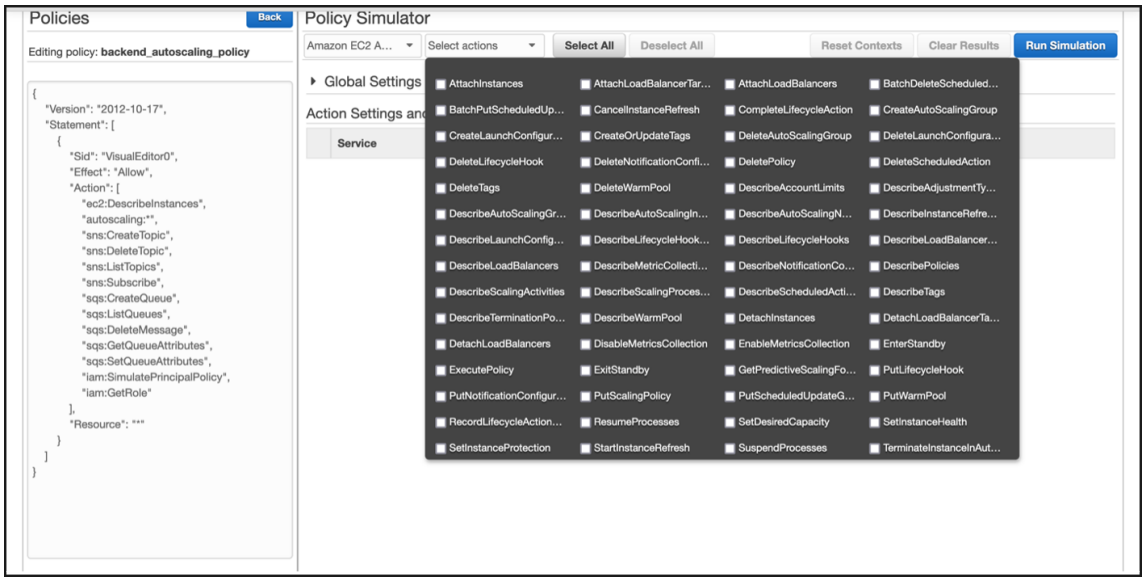
3. [ユーザー、グループ、ロール] タブで、ドロップダウンメニューから [ロール] を選択し、既存のロールを選択します。



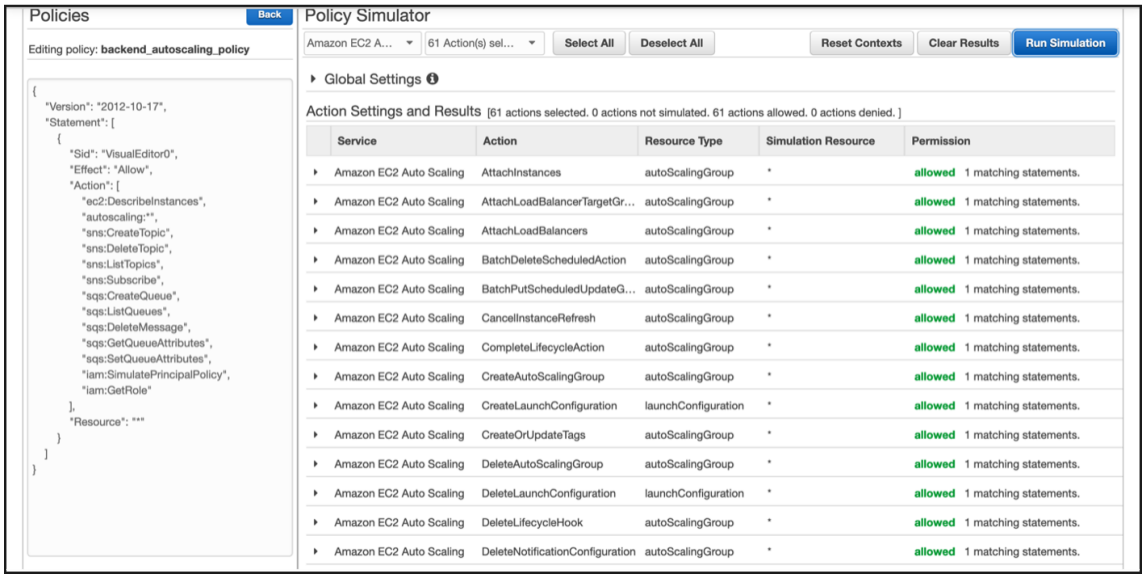
4. 既存のロールを選択したら、その下にある既存のポリシーを選択します。



5. ポリシーを選択すると、画面の左側に正確な JSON が表示されます。[アクションの選択] ドロップダウンメニューで目的のアクションを選択します。



6. [シミュレーションを実行] をクリックします。



詳細については、AWS IAM ドキュメントを参照してください。

その他の参考資料

IAM ロールを使用して Amazon EC2 インスタンスで実行されているアプリケーションにアクセス権限を付与する

AWS 上の NetScaler VPX インスタンスの仕組み

August 15, 2023

NetScaler VPX インスタンスは AWS マーケットプレイスで AMI として入手でき、AWS VPC 内で EC2 インスタンスとして起動することもできます。NetScaler VPX AMI インスタンスには、少なくとも 2 つの仮想 CPU と 2 GB のメモリが必要です。また、AWS VPC 内で起動される EC2 インスタンスは、複数のインターフェイス、インターフェイスごとに複数の IP アドレス、VPX 構成に必要なパブリックおよびプライベート IP アドレスも提供できます。各 VPX インスタンスには、少なくとも 3 つの IP サブネットが必要です。

- 管理サブネット
- クライアント側サブネット（VIP）
- バックエンド向けサブネット (SNIP、MIP など)

AWS での標準の VPX インスタンスのインストールには、3 つのネットワークインターフェイスをお勧めします。

現在、AWS では、AWS VPC 内で実行しているインスタンスでのみ、マルチ IP 機能を使用できます。VPC 内の VPX インスタンスを使用して、EC2 インスタンスで実行しているサーバーの負荷を分散できます。Amazon VPC を使用

すれば、独自の IP アドレス範囲、サブネット、ルートテーブル、ネットワークゲートウェイなどを含めて、仮想ネットワーク環境を作成および管理できます。

注: デフォルトでは、AWS アカウントごとに AWS リージョンごとに最大 5 つの VPC インスタンスを作成できます。Amazon のリクエストフォームを送信することで、VPC の上限を引き上げることをリクエストできます。
http://aws.amazon.com/contact-us/vpc-request

図 **1**: AWS アーキテクチャでの NetScaler VPX インスタンスデプロイのサンプル

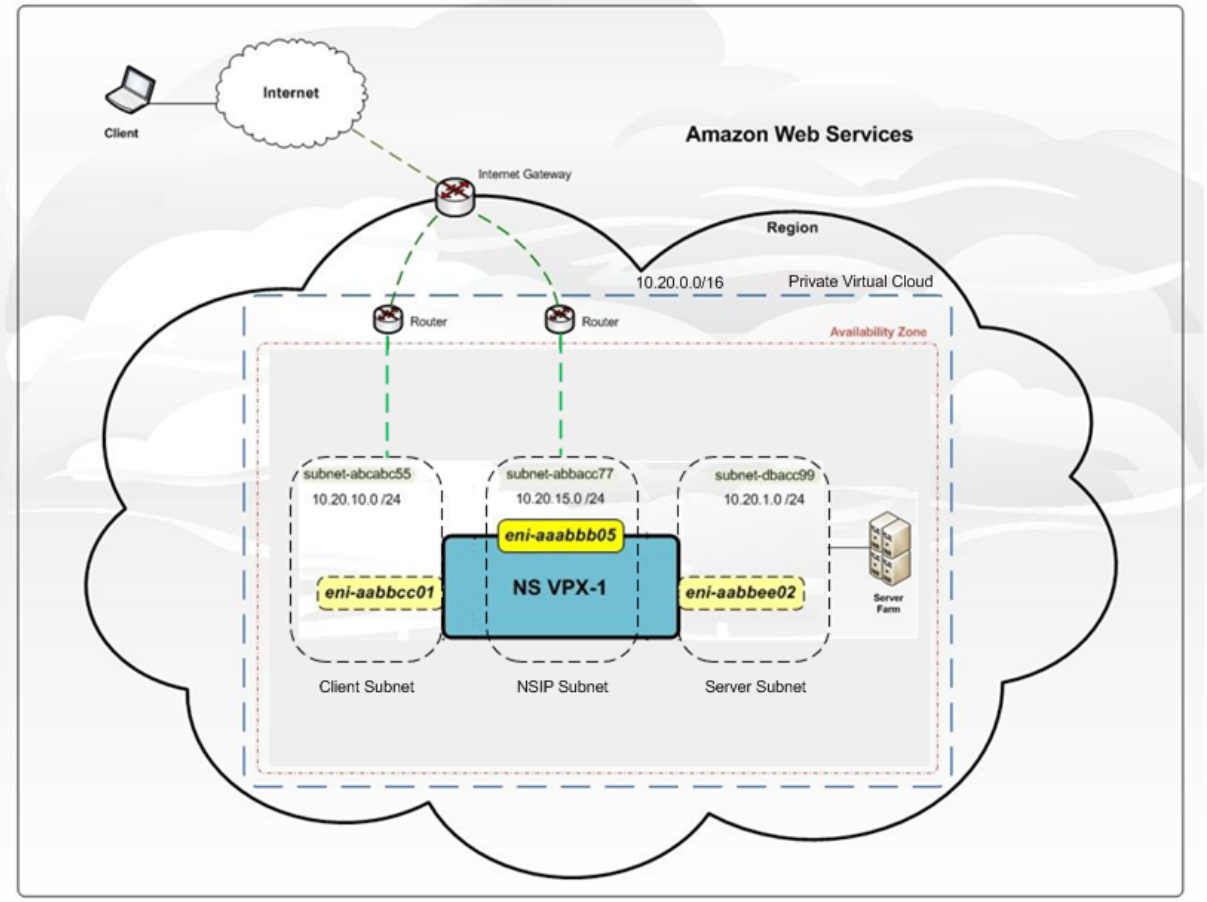


図 1 は、
NetScaler VPX がデプロイされた AWS VPC のシンプルなトポロジーを示しています。AWS VPC は、以下の要素で構成されています。

1. VPC からの送受信トラフィックをルーティングするための単一のインターネットゲートウェイ。
2. インターネットゲートウェイとインターネット間のネットワーク接続。
3. 3 つのサブネット（管理、クライアント、サーバー用に 1 つずつ）。
4. インターネットゲートウェイと 2 つのサブネット（管理用とクライアント用）間のネットワーク接続。
5. VPC 内にデプロイされたスタンドアロンの NetScaler VPX インスタンス。VPX インスタンスには、各サブネットに 1 つずつ接続された ENI が 3 つあります。

NetScaler VPX スタンドアロンインスタンスを AWS にデプロイする

August 15, 2023

NetScaler VPX スタンドアロンインスタンスは、次のオプションを使用して AWS にデプロイできます。

- AWS ウェブコンソール
- Citrix が作成した CloudFormation テンプレート
- AWS CLI

このトピックでは、NetScaler VPX インスタンスを AWS にデプロイする手順について説明します。

デプロイを開始する前に、以下のトピックをお読みください。

- Prerequisites
- 制限事項と使用上のガイドライン

AWS ウェブコンソールを使用して NetScaler VPX インスタンスを AWS にデプロイします

AWS Web コンソールを使用して、AWS で NetScaler VPX インスタンスを展開できます。展開のプロセスには、次の手順が含まれます。

1. キーペアの作成
2. 仮想プライベートクラウド（VPC）の作成
3. サブネットをさらに追加する
4. セキュリティグループとセキュリティルールの作成
5. ルートテーブルの追加
6. インターネットゲートウェイを作成する
7. NetScaler VPX インスタンスを作成する
8. ネットワークインターフェースをさらに作成してアタッチする
9. エラスティック IP の管理 NIC へのアタッチ
10. VPX インスタンスに接続する

ステップ **1:** キーペアを作成します。

Amazon EC2 は、キーペアを使用してログオン情報を暗号化および復号します。インスタンスにログオンするには、キーペアを作成し、インスタンスを起動するときにキーペアの名前を指定し、インスタンスに接続するときにプライベートキーを指定する必要があります。

AWS Launch Instance ウィザードを使用してインスタンスを確認し、起動すると、既存のキーペアを使用するか、新しいキーペアを作成するように求められます。キーペアの作成方法の詳細については、「Amazon EC2 キーペア」を参照してください。

ステップ **2: VPC** を作成します。

NetScaler VPC インスタンスは AWS VPC 内で展開されます。VPC では、AWS アカウント専用の仮想ネットワークを定義できます。AWS VPC の詳細については、「Amazon VPC の使用開始」を参照してください。

NetScaler VPX インスタンスに対する VPC の作成中は、次の点に留意してください。

- AWS アベイラビリティーゾーンに AWS VPC を作成するには、単一のパブリックサブネットのみのオプションで VPC を使用します。
- Citrix では、以下のタイプのサブネットを少なくとも **3** つ作成することをお勧めします。
  - 管理トラフィック用の 1 つのサブネット。このサブネットに管理 IP (NSIP) を配置します。デフォルトでは、エラスティックネットワークインターフェース (ENI) eth0 が管理 IP に使用されます。
  - クライアントアクセス（ユーザーから NetScaler ADC VPX）トラフィック用の 1 つ以上のサブネット。クライアントが NetScaler ADC 負荷分散仮想サーバーに割り当てられた 1 つ以上の仮想 IP（VIP）アドレスに接続します。
  - サーバーアクセス（VPX からサーバーへ）トラフィック用の 1 つ以上のサブネット。サーバーはこのサブネットを介して VPX 所有のサブネット IP（SNIP）アドレスに接続します。NetScaler 負荷分散と仮想サーバー、仮想 IP アドレス（VIP）、サブネット IP アドレス（SNIP）の詳細については、以下を参照してください。
  - すべてのサブネットは、同じアベイラビリティーゾーンに存在する必要があります。

ステップ **3:** サブネットを追加します。

VPC ウィザードを使った場合、作成されたサブネットは 1 つのみです。要件に応じて、さらにサブネットを作成することもできます。サブネットをさらに作成する方法について詳しくは、「VPC へのサブネットの追加」を参照してください。

ステップ **4:** セキュリティグループとセキュリティルールを作成します。

受信トラフィックと送信トラフィックを制御するには、セキュリティグループを作成し、そのグループに規則を追加します。グループを作成してルールを追加する方法の詳細については、「VPC のセキュリティグループ」を参照してください。

NetScaler VPX インスタンスの場合、EC2 ウィザードはデフォルトのセキュリティグループを提供します。このセキュリティグループは、AWS マーケットプレイスによって生成され、Citrix が推奨する設定に基づいています。ただし、要件に応じてさらにセキュリティグループを作成できます。

> 注
>
> ポート 22、80、443 をセキュリティグループでそれぞれ SSH、HTTP、HTTPS アクセス用に開きます。

ステップ **5:** ルートテーブルを追加します。

ルートテーブルには、ネットワークトラフィックの経路を判断する際に使用される、ルートと呼ばれる一連のルールが含まれます。VPC の各サブネットはルートテーブルに関連付ける必要があります。ルートテーブルの作成方法について詳しくは、「ルートテーブル」を参照してください。

ステップ **6**: インターネット **Gateway** を作成します。

インターネットゲートウェイには 2 つの目的があります。1 つは、インターネットでルーティング可能なトラフィックのターゲットを VPC ルートテーブルに提供すること、もう 1 つはパブリック IPv4 アドレスが割り当てられたインスタンスに対してネットワークアドレス変換 (NAT) を実行することです。

インターネットトラフィックに対して、インターネットゲートウェイを作成します。インターネットゲートウェイの作成方法の詳細については、「インターネットゲートウェイをアタッチする」を参照してください。

ステップ **7**: **AWS EC2** サービスを使用して **NetScaler ADC VPX** インスタンスを作成します。

AWS EC2 サービスを使って NetScaler VPX インスタンスを作成するには、次の手順に従います。

1. AWS ダッシュボードから、**[**コンピューティング**] > [EC2] > [**インスタンスの起動**] > [AWS** マーケットプレイス] に移動します。

   Launch Instance をクリックする前に、**Launch Instance** の下に表示される注記を確認して、リージョンが正しいことを確認してください。

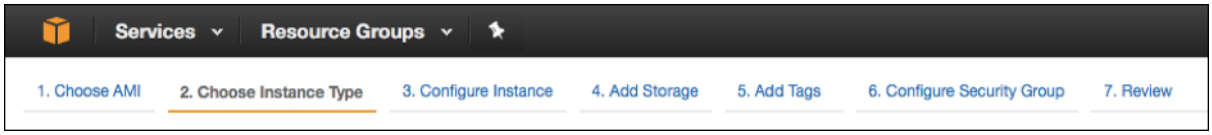
   

2. ［Search AWS Marketplace］バーで、「NetScaler VPX」と入力して検索します。
3. 展開するバージョンを選択し、[**Select**] をクリックします。NetScaler VPX バージョンでは、次のオプションがあります。

   - ライセンスバージョン
   - NetScaler VPX Express アプライアンス（これは無料の仮想アプライアンスで、NetScaler 12.0 56.20 から入手できます。）
   - 自分のデバイスを持参

Launch Instance ウィザードが起動します。ウィザードに従って、インスタンスを作成します。このウィザードでは、次のことを求められます。

- インスタンスの種類の選択
- インスタンスの構成
- ストレージの追加
- タグの追加
- セキュリティグループの構成
- 確認



ステップ **8:** ネットワークインターフェースをさらに作成してアタッチします。

VIP と SNIP 用に 2 つのネットワークインターフェイスを作成します。ネットワークインターフェイスの作成方法の詳細については、「ネットワークインターフェイスの作成」を参照してください。

ネットワークインターフェイスを作成したら、VPX インスタンスにアタッチする必要があります。インターフェイスを接続する前に、VPX インスタンスをシャットダウンし、インターフェイスを接続し、インスタンスの電源をオンにします。ネットワークインターフェイスの接続方法の詳細については、「インスタンスの起動時にネットワークインターフェイスをアタッチする」セクションを参照してください。

ステップ **9: Elastic IP** を割り当てて関連付けます。

EC2 インスタンスにパブリック IP アドレスを割り当てた場合、そのアドレスはインスタンスが停止されるまで割り当てられたままになります。その後、アドレスはプールに解放されます。インスタンスを再起動すると、新しいパブリック IP アドレスが割り当てられます。

対照的に、エラスティック IP（EIP）アドレスの場合は、インスタンスから割り当てが解除されるまで割り当ての状態が維持されます。

管理 NIC のエラスティック IP を割り当てて、関連付けます。Elastic IP アドレスを割り当てて関連付ける方法の詳細については、以下のトピックを参照してください。

- Elastic IP アドレスの割り当て
- Elastic IP アドレスを実行中のインスタンスに関連付ける

これらのステップで、AWS に NetScaler VPX インスタンスを作成する手順が完了します。インスタンスの準備が完了するまで数分かかる場合があります。インスタンスがステータスチェックに合格したことを確認します。この情報は、「インスタンス」ページの「ステータスチェック」列で確認できます。

ステップ **10: VPX** インスタンスに接続します。

VPX インスタンスを作成したら、GUI と SSH クライアントを使用してインスタンスを接続します。

- GUI

NetScaler VPX インスタンスにアクセスするためのデフォルト管理者の資格情報は以下のとおりです。

ユーザー名: `nsroot`

パスワード:ns root アカウントのデフォルトパスワードは、NetScaler VPX インスタンスの AWS インスタンス ID に設定されます。初めてログオンすると、セキュリティ上の理由からパスワードを変更するように求められます。パスワードを変更したら、設定を保存する必要があります。設定が保存されずにインスタンスが再起動する場合は、デフォルトのパスワードでログオンする必要があります。プロンプトでパスワードを再度変更します。

- SSH クライアント

AWS マネジメントコンソールから、**NetScaler VPX** インスタンスを選択して **[接続]** をクリックします。「インスタンスへの接続」ページの指示に従ってください。

AWS ウェブコンソールを使用して NetScaler ADC VPX スタンドアロンインスタンスを AWS にデプロイする方法の詳細については、以下を参照してください。

- シナリオ：スタンドアロンインスタンス
- Citrix CloudFormation テンプレートを使用して AWS 上の NetScaler VPX インスタンスを構成する方法

Citrix の **CloudFormation** テンプレートを使用して **NetScaler VPX** インスタンスを構成する

Citrix が提供する CloudFormation テンプレートを使用して、VPX インスタンスの起動を自動化できます。このテンプレートには、単一の NetScaler VPX インスタンスを起動したり、NetScaler VPX インスタンスのペアを使用して高可用性環境を作成したりする機能があります。

テンプレートは AWS Marketplace または GitHub から起動できます。

CloudFormation テンプレートには既存の VPC 環境が必要で、3 つのエラスティックネットワークインターフェース（ENI）を備えた VPX インスタンスを起動します。CloudFormation テンプレートを開始する前に、以下の要件を満たしていることを確認してください。

- AWS 仮想プライベートクラウド（VPC）
- VPC 内の 3 つのサブネット (1 つは管理用、1 つはクライアントトラフィック用、もう 1 つはバックエンドサーバー用)
- インスタンスへの SSH アクセスを有効にする EC2 キーペア
- UDP 3003、TCP 3009—3010、HTTP、SSH ポートが開いているセキュリティグループ

前提条件を満たす方法の詳細については、「AWS ウェブコンソールを使用して AWS に NetScaler ADC VPX インスタンスをデプロイする」セクションまたは AWS のドキュメントを参照してください。

このビデオでは 、AWS Marketplace で利用可能な Citrix CloudFormation テンプレートを使用して、NetScaler VPX スタンドアロンインスタンスを構成して起動する方法について説明します。

さらに、GitHub で利用可能な Citrix CloudFormation テンプレートを使用して、NetScaler VPX Express のスタンドアロンインスタンスを構成して起動します。

https://github.com/citrix/citrix-adc-aws-cloudformation/tree/master/templates/standalone/

IAM ロールはスタンドアロンデプロイでは必須ではありません。ただし、Citrix では、将来の必要に備えて、必要な権限を持つ IAM ロールを作成してインスタンスにアタッチすることを推奨しています。IAM ロールにより、スタンドアロンインスタンスは、必要に応じて SR-IOV を使用して高可用性ノードに簡単に変換されます。

必要な権限の詳細については、「SR-IOV ネットワークインターフェイスを使用するための NetScaler ADC VPX インスタンスの構成」を参照してください。

> 注:
> AWS ウェブコンソールを使用して NetScaler ADC VPX インスタンスを AWS にデプロイする場合、CloudWatch サービスはデフォルトで有効になります。Citrix CloudFormation テンプレートを使用して NetScaler ADC VPX インスタンスを展開する場合、デフォルトのオプションは「はい」です。CloudWatch サービスを無効にする場合は、[いいえ] を選択します。詳細については、「Amazon CloudWatch を使用したインスタンスのモニタリング」を参照してください。

AWS CLI を使用して **NetScaler ADC VPX** インスタンスを構成する

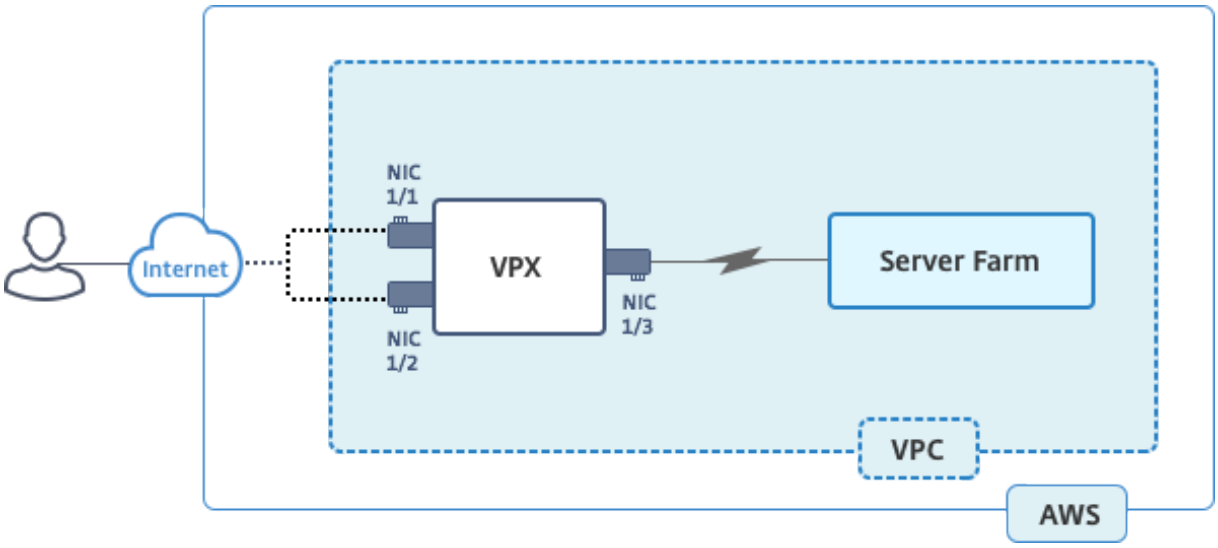
AWS CLI を使用してインスタンスを起動できます。詳細については、AWS コマンドラインインターフェイスのドキュメントを参照してください。

シナリオ：スタンドアロンインスタンス

August 15, 2023

このシナリオでは、AWS GUI を使用して NetScaler VPX スタンドアロン EC2 インスタンスを AWS にデプロイする方法を示しています。3 つの NIC を持つスタンドアロン VPX インスタンスを作成します。インスタンスは、負荷分散仮想サーバーとして構成されており、バックエンドサーバー（サーバーファーム）と通信します。この設定では、インスタンスとバックエンドサーバー間、およびパブリックインターネット上のインスタンスと外部ホスト間の必要な通信ルートを設定します。

VPX インスタンスをデプロイする手順の詳細については、「NetScaler VPX スタンドアロンインスタンスを AWS にデプロイする」を参照してください。

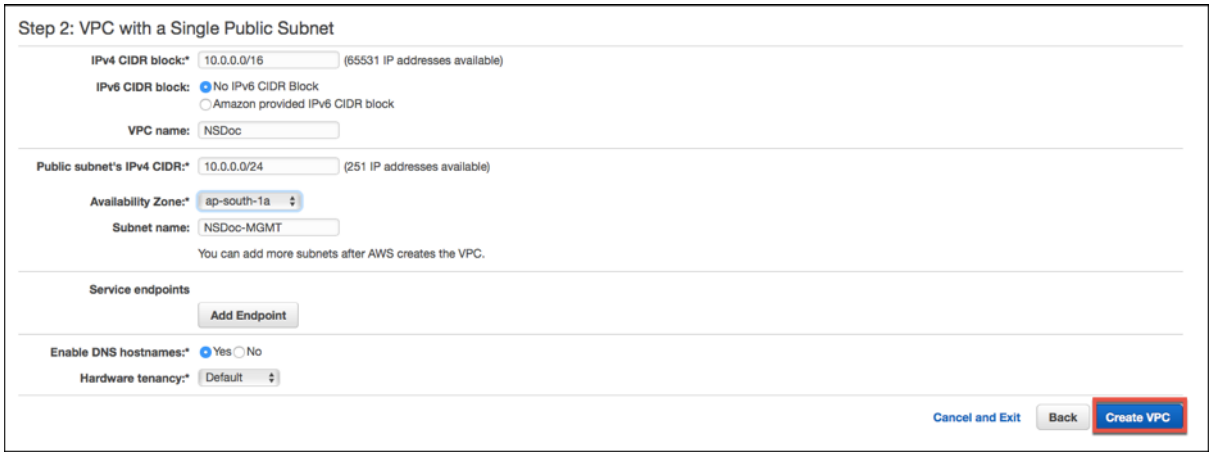


3 つの NIC を作成します。各 NIC は、IP アドレスのペア（パブリックとプライベート）を使用して構成できます。NIC は、次の目的に役立ちます。

NIC	目的	関連付けられている
eth0	NSIP（管理トラフィックを処理する）	パブリック IP アドレスとプライベート IP アドレス
eth1	クライアント側のトラフィック（VIP）をサービスする	パブリック IP アドレスとプライベート IP アドレス
eth2	バックエンド・サーバ（SNIP）との通信	パブリック IP アドレス (プライベート IP アドレスは必須ではありません)

ステップ 1: VPC を作成します。

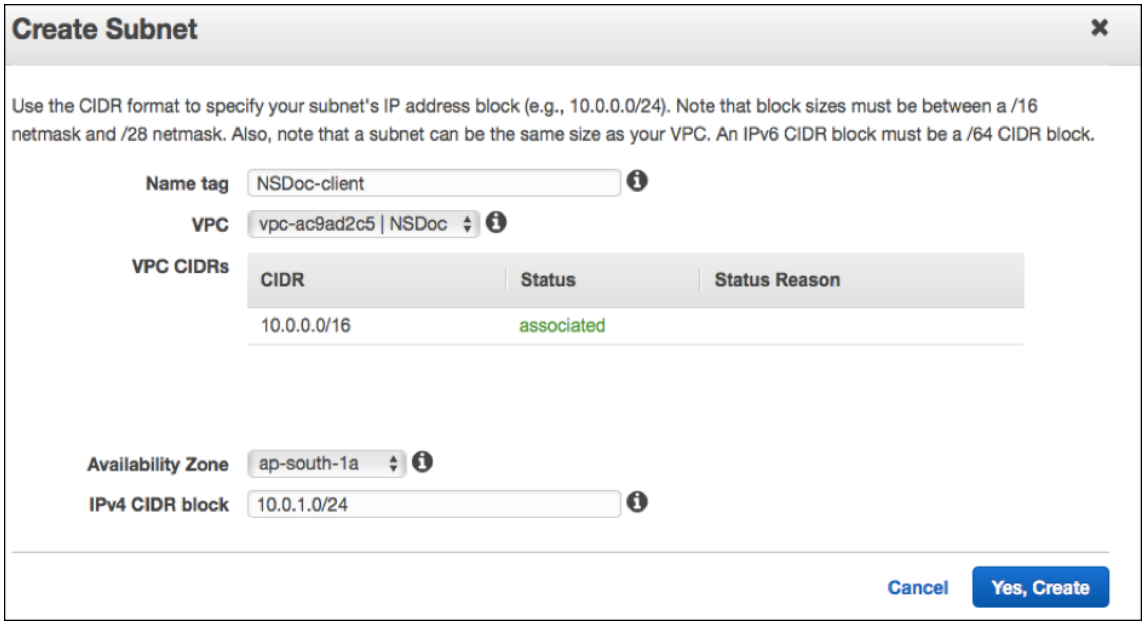
1. AWS ウェブコンソールにログインし、[ネットワークとコンテンツ配信] > [**VPC**] に移動します。[**VPC** ウィザードの起動] をクリックします。
2. **1** つのパブリックサブネットを持つ **VPC** を選択し、**[**選択**]** をクリックします。
3. このシナリオでは、IP CIDR ブロックを 10.0.0.0/16 に設定します。
4. VPC の名前を指定します。
5. パブリックサブネットを 10.0.0.0/24 に設定します。(これは管理ネットワークです)。
6. アベイラビリティ ゾーンを選択してください。
7. サブネットの名前を付けます。
8. [**VPC** の作成] をクリックします。



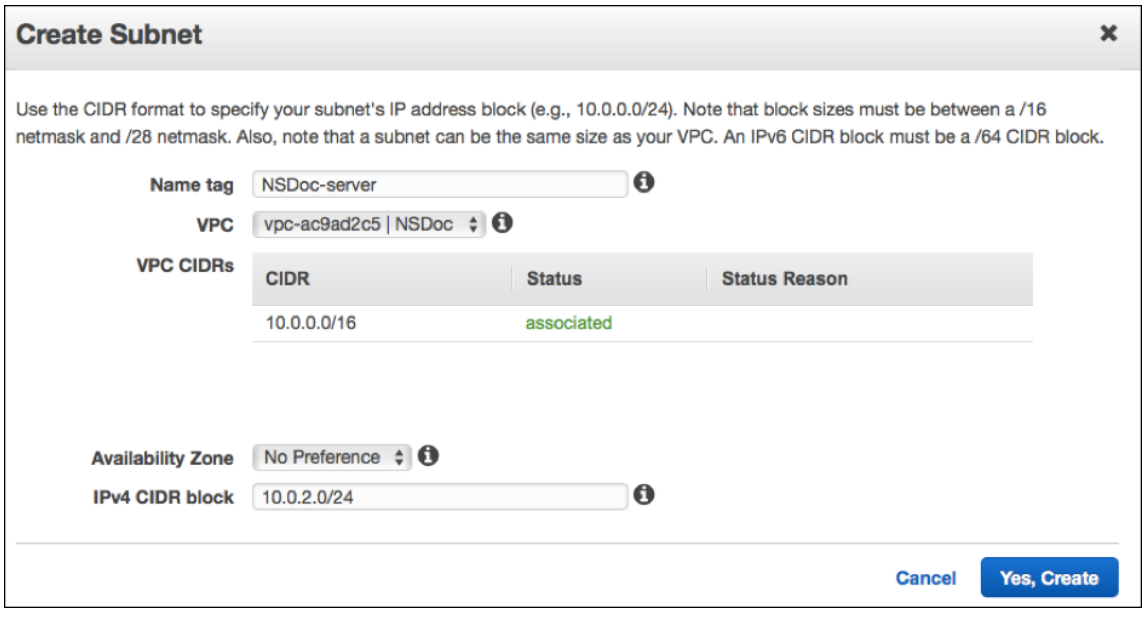
ステップ 2: 追加のサブネットを作成します。

1. https://console.aws.amazon.com/vpc/で Amazon VPC コンソールを開きます。
2. 次の詳細を入力したら、ナビゲーションペインで [Subnets]、[Create Subnet] を選択します。
   - 名前タグ: サブネットの名前を指定します。

- VPC: サブネットを作成する対象の VPC を選択します。
- アベイラビリティーゾーン: ステップ 1 で VPC を作成したアベイラビリティーゾーンを選択します。
- IPv4 CIDR ブロック: サブネットの IPv4 CIDR ブロックを指定します。このシナリオでは、10.0.1.0/24 を選択します。



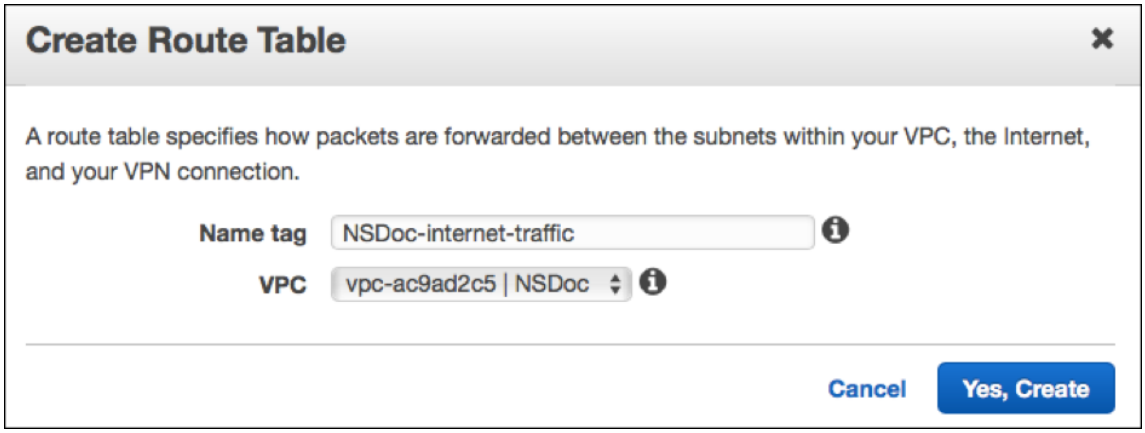
3. 手順を繰り返して、バックエンドサーバー用のサブネットをもう 1 つ作成します。



ステップ 3: ルートテーブルを作成します。

1. https://console.aws.amazon.com/vpc/で Amazon VPC コンソールを開きます。
2. ナビゲーションペインで、「ルートテーブル」>「ルートテーブルの作成」を選択します。
3. ルートテーブルの作成ウィンドウで、名前を追加し、ステップ 1 で作成した VPC を選択します。

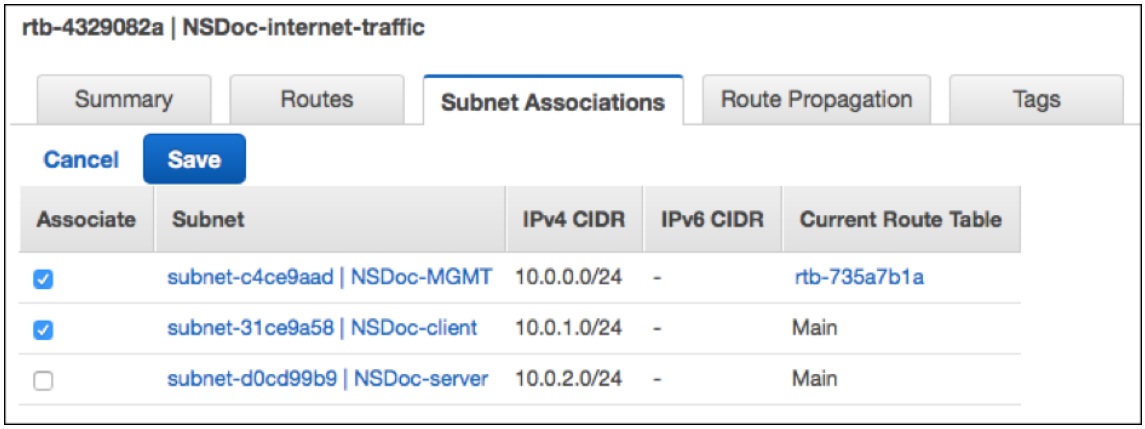
4. [**Yes, Create**] をクリックします。



ルートテーブルは、この VPC 用に作成したすべてのサブネットに割り当てられます。これにより、あるサブネット内のインスタンスからのトラフィックのルーティングが別のサブネットのインスタンスに到達できるようになります。

5. [サブネットの関連付け] をクリックし、[編集] をクリックします。

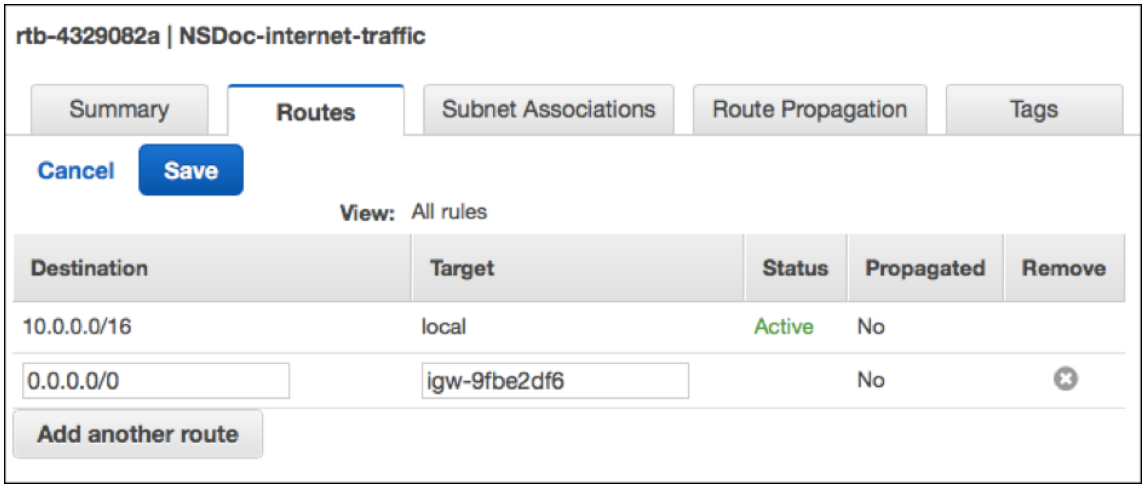
6. 管理サブネットとクライアントサブネットをクリックし、[保存] をクリックします。これにより、インターネットトラフィック専用のルートテーブルが作成されます。



7. [ルート**] > [**編集**] > [**別のルートを追加] をクリックします。

8. [宛先] フィールドに 0.0.0.0/0 を追加し、[ターゲット] フィールドをクリックして、VPC <xxxx> ウィザードが自動的に作成したインターネットゲートウェイの igw- を選択します。

9. [保存] をクリックします。

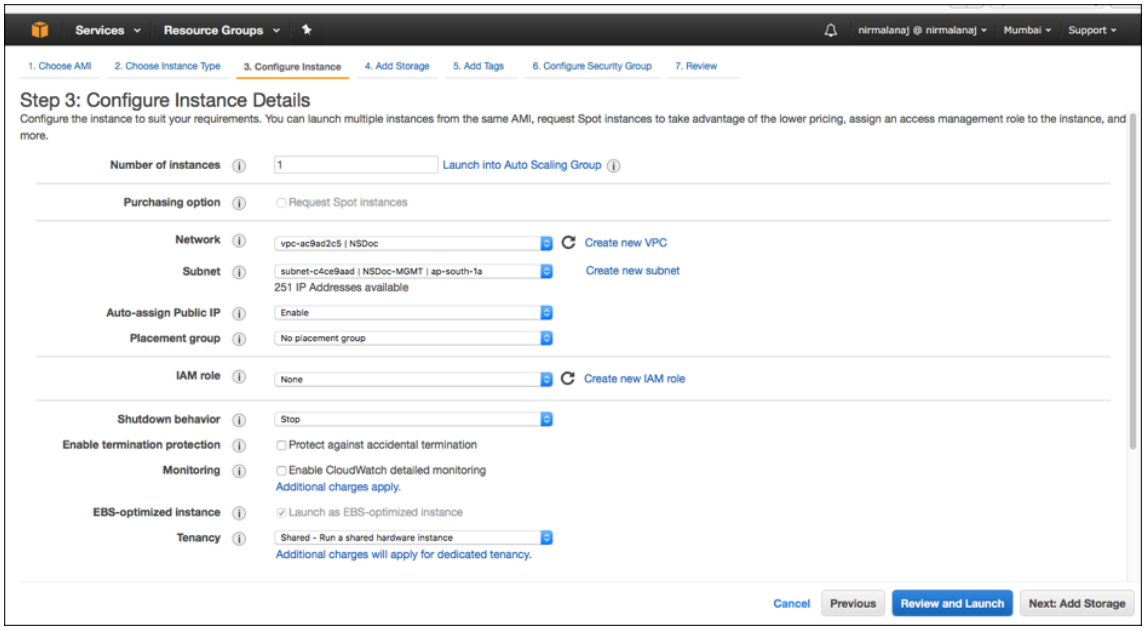


10. 手順に従って、サーバー側トラフィックのルートテーブルを作成します。

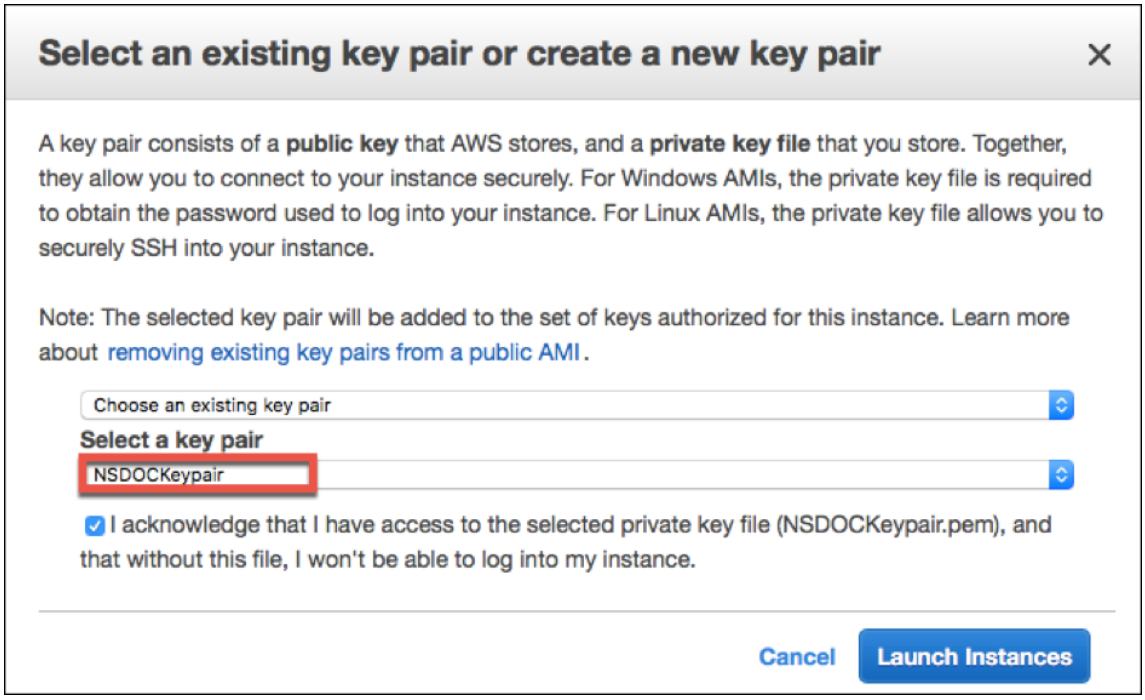
ステップ 4: NetScaler VPX インスタンスを作成します。

1. AWS マネジメントコンソールにログインし、**Compute** の下の **EC2** をクリックします。
2. AWS Marketplace クリックします。AWS Marketplace 検索バーに「NetScaler VPX」と入力し、Enter キーを押します。使用可能な NetScaler VPX エディションが表示されます。
3. 「選択」をクリックして、目的の NetScaler VPX エディションを選択します。EC2 インスタンスウィザードが起動します。
4. インスタンスタイプの選択ページで、**m4** を選択します。**Xlarge** (推奨) をクリックし、[次へ**:** インスタンスの詳細の設定] をクリックします。
5. 「インスタンスの詳細の設定」ページで、以下を選択し、「次へ: ストレージの追加」をクリックします。

    - インスタンス数:1
    - ネットワーク: ステップ 1 で作成した VPC
    - サブネット: 管理サブネット
    - パブリック IP の自動割り当て: 有効



6. 「ストレージの追加」ページで、デフォルトオプションを選択し、「次へ: タグの追加」をクリックします。
7. 「タグの追加」ページで、インスタンスの名前を追加し、「次へ: セキュリティグループの設定」をクリックします。
8. 「セキュリティグループの設定」ページで、デフォルトオプション（AWS Marketplace によって生成され、Citrix Systems が推奨する設定に基づく）を選択し、「確認して起動」>「起動」をクリックします。
9. 既存のキーペアを選択するか、新しいキーペアを作成するように求められます。「キーペアの選択」ドロップダウンリストから、前提条件として作成したキーペアを選択します（「前提条件」セクションを参照）。
10. チェックボックスをオンにしてキーペアを確認し、[Launch Instances] をクリックします。



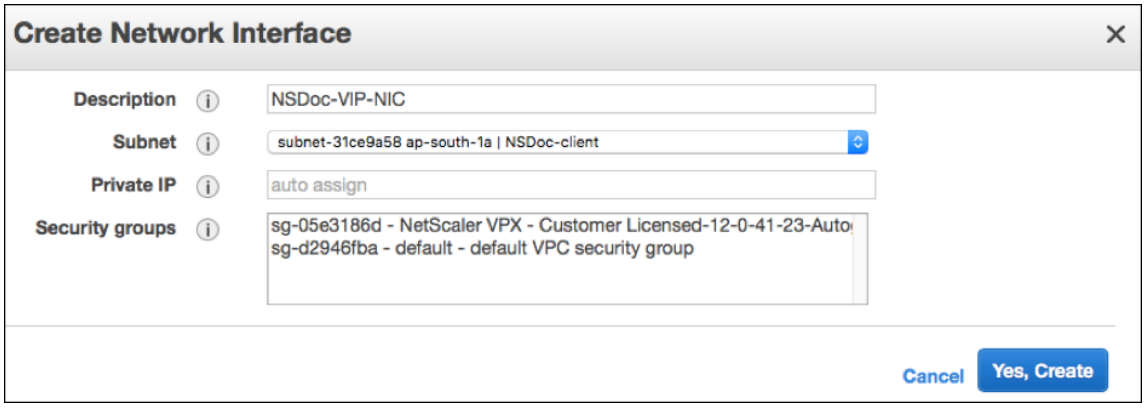
インスタンス起動ウィザードには起動ステータスが表示され、インスタンスが完全に起動するとインスタンスリストに表示されます。

インスタンスをチェックし、AWS コンソールで [EC2] > [実行中のインスタンス] をクリックします。インスタンスを選択して名前を追加します。インスタンスステートが実行中で、ステータスチェックが完了していることを確認します。

ステップ 5: ネットワークインターフェースをさらに作成してアタッチします。

VPC を作成したときには、その VPC に関連付けられたネットワークインターフェースは 1 つだけです。次に、VPC に VIP と SNIP の 2 つのネットワークインターフェイスを追加します。

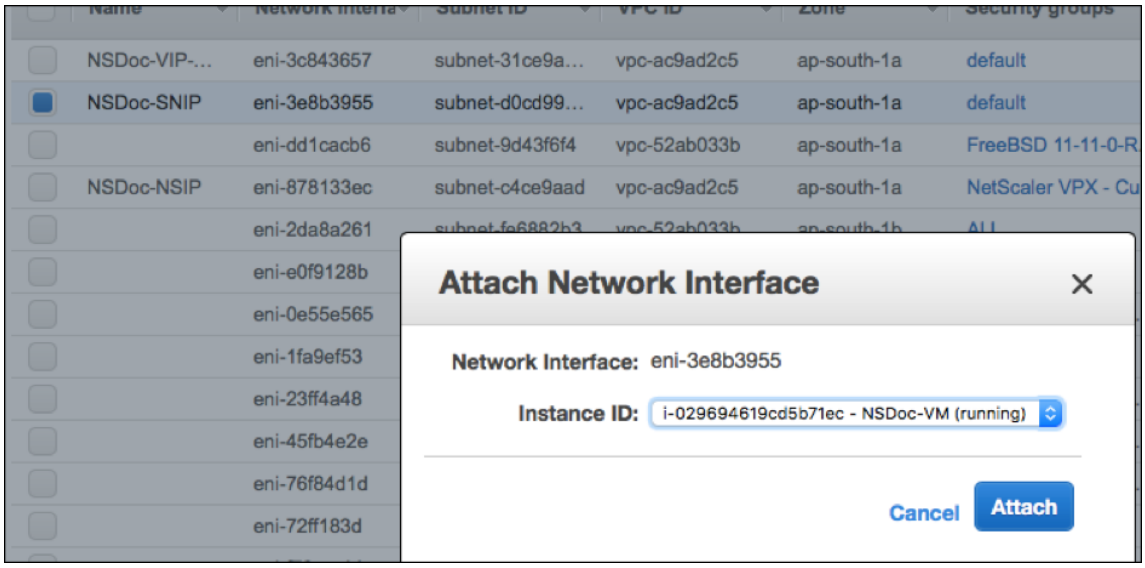
1. https://console.aws.amazon.com/ec2/で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[ネットワークインターフェイス] を選択します。
3. [ネットワークインターフェースの作成] を選択します。
4. [説明] に、わかりやすい名前を入力します。
5. [サブネット] には、VIP 用に以前に作成したサブネットを選択します。
6. プライベート IP については、デフォルトオプションのままにします。
7. 「セキュリティグループ」で、グループを選択します。
8. [**Yes, Create**] をクリックします。



9. ネットワークインターフェースを作成したら、インターフェースに名前を追加します。
10. 手順を繰り返して、サーバー側トラフィック用のネットワークインターフェースを作成します。

ネットワークインターフェースを接続します。

1. ナビゲーションペインで、[ネットワークインターフェイス] を選択します。
2. ネットワークインターフェースを選択し、[接続] を選択します。
3. 「ネットワークインターフェースをアタッチ」ダイアログで、インスタンスを選択して「アタッチ」を選択します。



ステップ 6: エラスティック IP を NSIP に接続します。

1. AWS マネジメントコンソールから、[ネットワークとセキュリティ] **> [Elastic IP**] に移動します。
2. 接続できる無料の EIP を確認してください。存在しない場合は、[新しいアドレスを割り当て] をクリックします。
3. 新しく割り当てられた IP アドレスを選択し、[アクション] **> [**アソシエイトアドレス] を選択します。

4. ネットワークインターフェースのラジオボタンをクリックします。
5. ネットワークインターフェースのドロップダウンリストから、管理 NIC を選択します。
6. プライベート **IP** ドロップダウンメニューから、AWS で生成された IP アドレスを選択します。
7.「再関連付け」チェックボックスを選択します。
8.「関連付け」をクリックします。

VPX インスタンスにアクセスします。

スタンドアロンの NetScaler VPX インスタンスを 3 つの NIC で構成したら、VPX インスタンスにログオンして NetScaler 側の構成を完了します。次のオプションの使用:

- GUI: 管理 NIC のパブリック IP をブラウザに入力します。`nsroot` ユーザー名として、インスタンス ID (i-0c1ffe1d987817522) をパスワードとして使用してログオンします。

> 注:
>
> 初めてログオンすると、セキュリティ上の理由からパスワードを変更するように求められます。パスワードを変更したら、設定を保存する必要があります。設定が保存されずにインスタンスが再起動する場合は、デフォルトのパスワードでログオンする必要があります。プロンプトが表示されたらパスワードをもう一度変更し、設定を保存します。

- SSH: SSH クライアントを開き、次のように入力します。

```
ssh -i \<location of your private key\> ns root@\<public DNS of the instance\>
```

パブリック DNS を見つけるには、インスタンスをクリックして [接続] をクリックします。

関連情報:

- NetScaler ADC が所有する IP アドレス（NSIP、VIP、および SNIP）を構成するには、NetScaler ADC 所有の IP アドレスの構成を参照してください。
- NetScaler VPX アプライアンスの BYOL バージョンを構成しました。詳細については、VPX ライセンスガイド（http://support.citrix.com/article/CTX122426）

NetScaler VPX ライセンスをダウンロードする

October 25, 2023

AWS マーケットプレイスから NetScaler ADC VPX-カスタマーライセンスインスタンスを起動した後、ライセンスが必要です。VPX ライセンスの詳細については、ライセンスの概要を参照してください。

次の操作を実行する必要があります。

1. Citrix Web サイト内のライセンスポータルを使用して、有効なライセンスを生成します。
2. ライセンスをインスタンスにアップロードします。

有料 マーケットプレイスインスタンスの場合は、ライセンスをインストールする必要はありません。正しい機能セットとパフォーマンスが自動的にアクティブ化されます。

モデル番号が VPX5000 より大きい NetScaler VPX インスタンスを使う場合は、ネットワークのスループットはインスタンスのライセンスに規定されているのと同じではないことがあります。ただし、SSL スループットや 1 秒あたりの SSL トランザクションといった他の機能は改善されている場合があります。

`c4.8xlarge`インスタンスタイプでは 5 Gbps のネットワーク帯域幅が観測されます。

AWS サブスクリプションを BYOL に移行する方法

このセクションでは、AWS サブスクリプションから独自のライセンス (BYOL) に移行する手順、およびその逆について説明します。

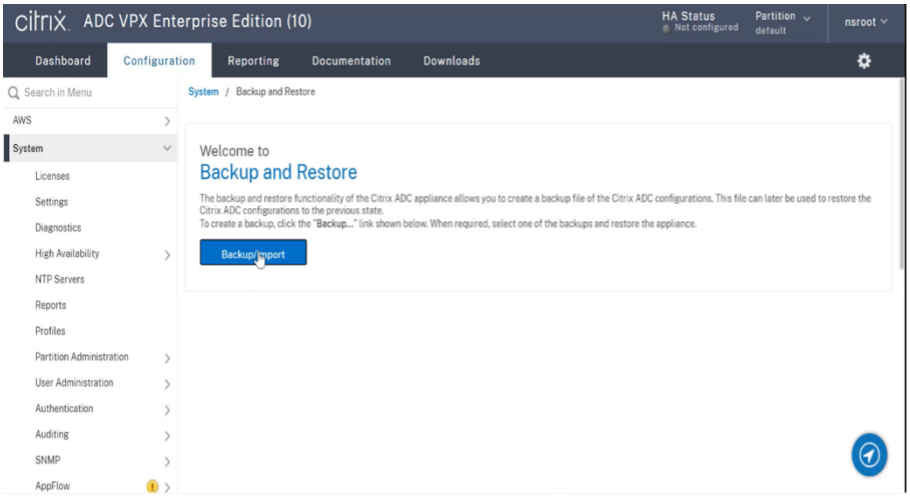
AWS サブスクリプションを BYOL に移行するには、次の手順を実行します。

> 注
>
> ** ステップ **2** とステップ **3** は **Citrix** ADC VPX インスタンスで実行され、その他の手順はすべて AWS ポータルで実行されます。

1. NetScaler VPX-同じセキュリティグループ、IAM ロール、サブネットを持つ古い EC2 インスタンスと同じアベイラビリティーゾーンで、カスタマーライセンスを使用して BYOL EC2 インスタンスを作成します。新しい EC2 インスタンスには ENI インターフェイスが 1 つだけ必要です。
2. NetScaler GUI を使用して古い EC2 インスタンスのデータをバックアップするには、次の手順に従います。

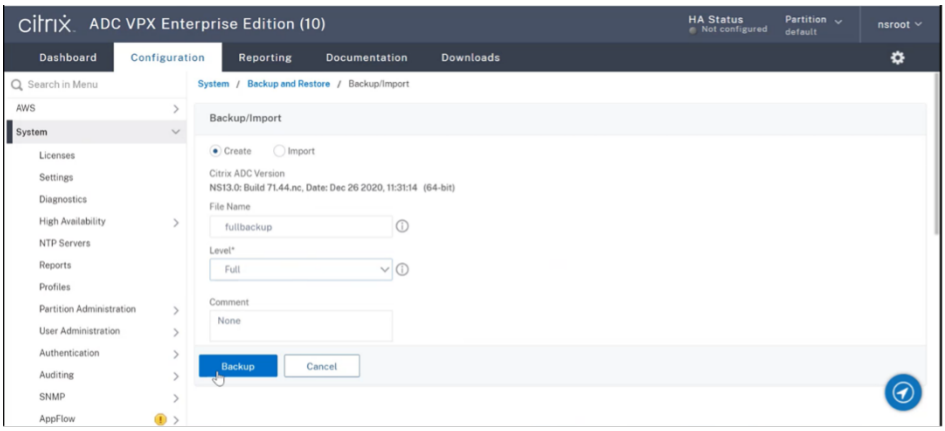
   a) [システム] **>** [バックアップと復元] に移動します。

   b) [ようこそ] ページで、[バックアップ/インポート] をクリックしてプロセスを開始します。

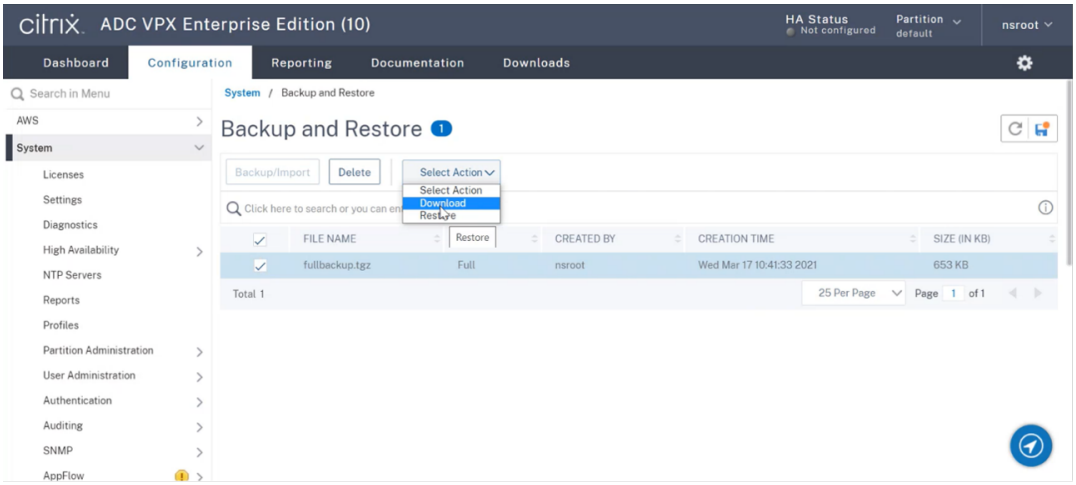


c) [バックアップ/インポート] ページで、次の詳細を入力します。

- **Name**： バックアップファイルの名前。
- **Level**： バックアップレベルを「フル」として選択します。
- [コメント]: バックアップの簡単な説明を入力します。



d) [バックアップ] をクリックします。バックアップが完了したら、ファイルを選択してローカルマシンにダウンロードできます。

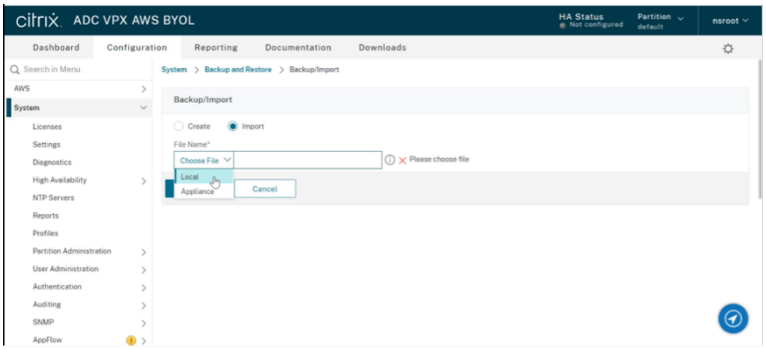


3. NetScaler GUI を使用して新しい EC2 インスタンスにデータを復元するには、次の手順に従います。

   a) [システム] **>** [バックアップと復元] に移動します。

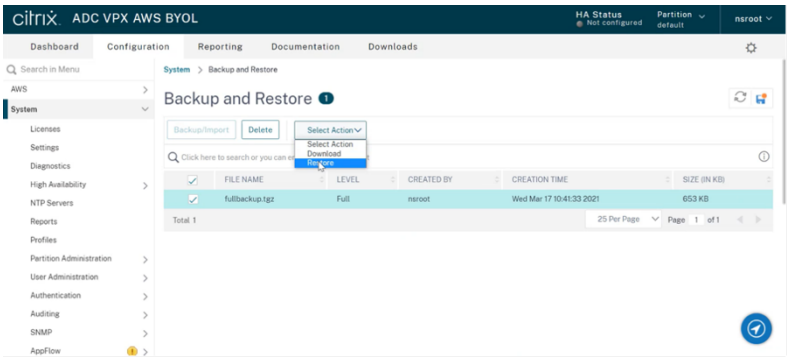
   b) [バックアップ/インポート] をクリックして、プロセスを開始します。

   c) [インポート] オプションを選択し、バックアップファイルをアップロードします。

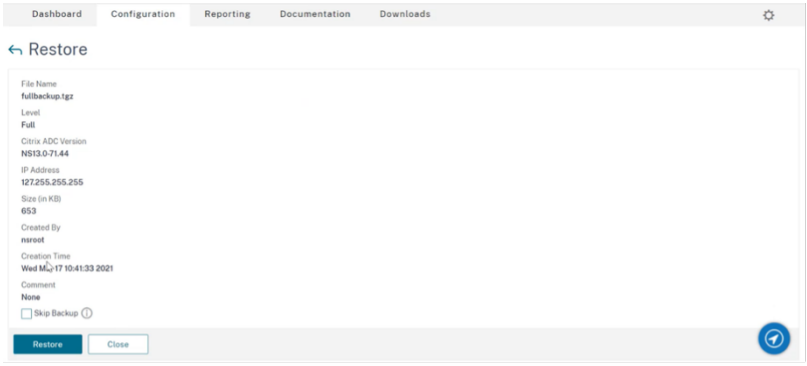


   d) ファイルを選択します。

   e) [アクションの選択] ドロップダウンメニューから、[復元] を選択します。



   f) [復元] ページで、ファイルの詳細を確認し、[復元] をクリックします。



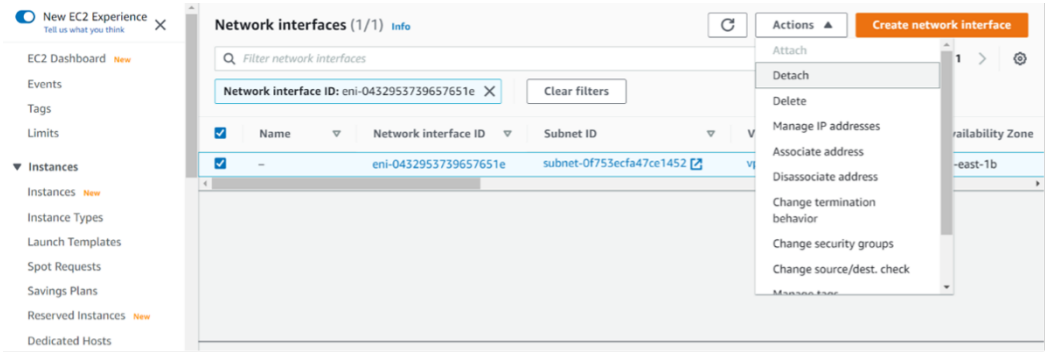
g) 復元後、EC2 インスタンスを再起動します。

4. 古い EC2 インスタンスから新しい EC2 インスタンスに、すべてのインターフェイス（NSIP アドレスがバインドされている管理インターフェイスを除く）を移動します。ネットワークインターフェイスを別の EC2 インスタンスに移動するには、次の手順を実行します。

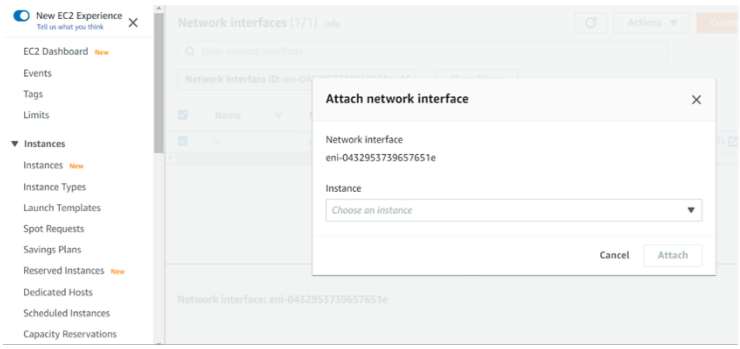
   a) **AWS** ポータルで、古い EC2 インスタンスと新しい EC2 インスタンスの両方を停止します。

   b) [ネットワークインターフェイス] に移動し、古い EC2 インスタンスにアタッチされたネットワークインターフェイスを選択します。

   c) [アクション] **> [**デタッチ**]** をクリックして **EC2** インスタンスをデタッチします。



   d) [アクション] **> [Attach]** の順にクリックして、ネットワークインターフェイスを新しい **EC2** インスタンスにアタッチします。ネットワークインターフェイスをアタッチする必要がある EC2 インスタンス名を入力します。



e) 接続されている他のすべてのインターフェイスについて、ステップ **1** からステップ **4** を実行します。シーケンスに従い、インターフェイスの順序を維持するようにしてください。つまり、まずインターフェイス 2 をデタッチして接続し、次にインターフェイス 3 をデタッチして接続します。

5. 古い EC2 インスタンスから管理インターフェイスをデタッチすることはできません。したがって、古い EC2 インスタンスの管理インターフェイス（プライマリネットワークインターフェイス）上のすべてのセカンダリ IP アドレス（存在する場合）を新しい EC2 インスタンスに移動します。IP アドレスをあるインターフェイスから別のインターフェイスに移動するには、次の手順を実行します。

   a) **AWS** ポータルで、古い EC2 インスタンスと新しい EC2 インスタンスの両方が **Stop** 状態であることを確認します。

   b) [ネットワークインターフェイス] に移動し、古い EC2 インスタンスにアタッチされた管理ネットワークインターフェイスを選択します。

   c) [アクション] **> [IP** アドレスの管理] の順にクリックし、割り当てられているすべてのセカンダリ IP アドレス（存在する場合）を書き留めます。

   d) 新しい EC2 インスタンスの管理ネットワークインターフェイスまたはプライマリインターフェイスに移動します。

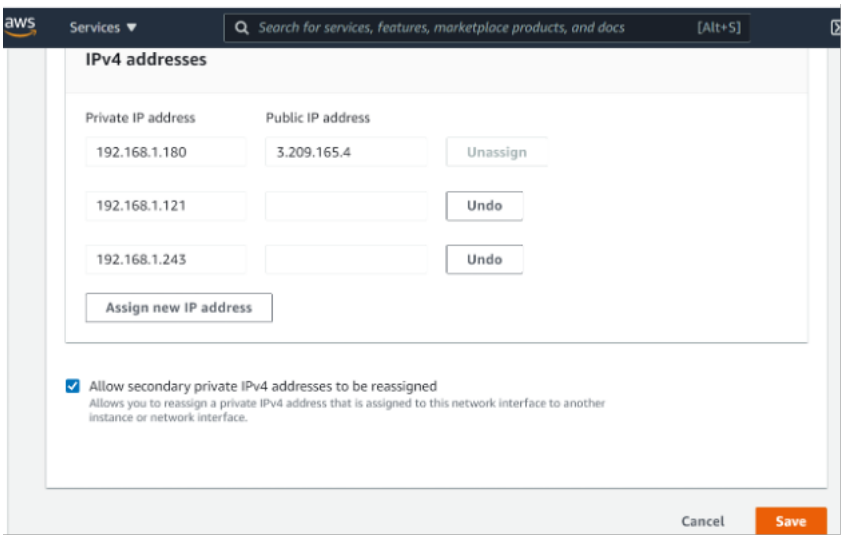
   e) [アクション] **> [IP** アドレスの管理] の順にクリックします。

   f) [**IPv4** アドレス] で、[新しい **IP** アドレスを割り当て] をクリックします。

   g) ステップ **3** で説明する IP アドレスを入力します。

   h) [セカンダリプライベート **IP** アドレスの再割り当てを許可する] チェックボックスをオンにします。

   i) [保存] をクリックします。



6. 新しい EC2 インスタンスを起動し、設定を確認します。すべての設定が移動されたら、要件に従って古い EC2 インスタンスを削除または保持できます。

7. 古い EC2 インスタンスの NSIP アドレスに EIP アドレスがアタッチされている場合は、古いインスタンスの NSIP アドレスを新しいインスタンスの NSIP アドレスに移動します。
8. 古いインスタンスに戻す場合は、古いインスタンスと新しいインスタンスの逆の方法で同じ手順を実行します。
9. サブスクリプションインスタンスから BYOL インスタンスに移行した後、ライセンスが必要です。ライセンスをインストールするには、次の手順に従います。
    - Citrix Web サイトのライセンスポータルを使用して、有効なライセンスを生成します。
    - ライセンスをインスタンスにアップロードします。詳細については、VPX ADC-新しいライセンスをインストールするを参照してください。

> 注
>
> BYOL インスタンスをサブスクリプションインスタンス (有料マーケットプレイスインスタンス) に移動する場合、ライセンスをインストールする必要はありません。正しい機能セットとパフォーマンスが自動的にアクティブ化されます。

制限事項

管理インターフェイスを新しい EC2 インスタンスに移動することはできません。したがって、管理インターフェイスを手動で構成することをお勧めします。詳細については、前の手順の手順 **5** を参照してください。新しい EC2 インスタンスは、古い EC2 インスタンスの正確なレプリカで作成されますが、新しい IP アドレスは NSIP アドレスだけです。

異なる可用性ゾーンでの負荷分散サーバー

August 15, 2023

VPX インスタンスを使用して、同じアベイラビリティーゾーンまたは以下の場所で稼働しているサーバーの負荷分散を行うことができます。

- 同じ AWS VPC 内の異なるアベイラビリティーゾーン (AZ)
- 別の AWS リージョン
- VPC 内の AWS EC2

VPX インスタンスが、その VPX インスタンスが存在する AWS VPC の外部で実行されているサーバーの負荷を分散できるようにするには、次のように、EIP を使用してインターネットゲートウェイを介してトラフィックをルーティングするようにインスタンスを設定します。

1. NetScaler CLI または GUI を使用して、NetScaler VPX インスタンスで SNIP を構成します。
2. サーバー側のトラフィック用にパブリックサブネットを作成することで、トラフィックが AZ からルーティングされるようにします。

3. AWS GUI コンソールを使用して、インターネット Gateway ルートをルーティングテーブルに追加します。
4. 更新したルーティングテーブルをサーバー側のサブネットに関連付けます。
5. NetScaler SNIP アドレスにマップされているサーバー側のプライベート IP アドレスに EIP を関連付けます。

AWS での高可用性の機能

August 15, 2023

AWS 上の 2 つの NetScaler ADC VPX インスタンスを高可用性（HA）アクティブ/パッシブペアとして構成できます。1 つのインスタンスをプライマリノードとして、もう 1 つのインスタンスをセカンダリノードとして設定すると、1 次ノードは接続を受け入れ、サーバーを管理します。セカンダリノードはプライマリを監視します。何らかの理由で 1 次ノードが接続を受け入れることができない場合は、2 次ノードが引き継ぎます。

AWS では、VPX インスタンスで次のデプロイタイプがサポートされています。

- 同一ゾーン内での高可用性
- 異なるゾーン間の高可用性

> 注
>
> 高可用性を機能させるには、両方の NetScaler ADC VPX インスタンスに IAM ロールがアタッチされ、Elastic IP（EIP）アドレスが NSIP に割り当てられていることを確認してください。NSIP が NAT インスタンスを介してインターネットにアクセスできる場合は、NSIP に EIP を割り当てる必要はありません。

同じゾーン内の高可用性

同じゾーン内の高可用性展開では、両方の VPX インスタンスのネットワーク構成が類似している必要があります。

次の 2 つのルールに従います。

ルール 1. 1 つの VPX インスタンスの NIC は、他の VPX の対応する NIC と同じサブネットにある必要があります。どちらのインスタンスにも次のものが必要です。

- 同じサブネット (管理サブネットと呼ばれる) 上の管理インターフェイス
- 同じサブネット (クライアントサブネットと呼ばれる) 上のクライアントインターフェイス
- 同じサブネット (サーバーサブネットと呼ばれる) 上のサーバーインターフェイス

ルール 2。両方のインスタンスの管理 NIC、クライアント NIC、およびサーバ NIC のシーケンスが同じである必要があります。
たとえば、次のシナリオはサポートされていません。

VPX インスタンス 1

NIC 0: 管理
NIC 1: クライアント
NIC 2: サーバ

VPX インスタンス 2

NIC 0: 管理

NIC 1: サーバ

NIC 2: クライアント

このシナリオでは、インスタンス 1 の NIC 1 はクライアントサブネットにあり、インスタンス 2 の NIC 1 はサーバーサブネットにあります。HA が機能するには、両方のインスタンスの NIC 1 がクライアントサブネットまたはサーバーサブネット内にある必要があります。

13.0 41.xx から、フェールオーバー後にプライマリ HA ノードの NIC（クライアント側およびサーバ側の NIC）に接続されたセカンダリプライベート IP アドレスをセカンダリの HA ノードに移行することで、高可用性を実現できます。この展開は、以下のように管理されます。

- 両方の VPX インスタンスは、NIC 列挙に従って NIC の数とサブネットマッピングが同じです。
- 各 VPX NIC には、管理 IP アドレスに対応する最初の NIC を除き、追加のプライベート IP アドレスが 1 つあります。追加のプライベート IP アドレスは、AWS ウェブコンソールでプライマリプライベート IP アドレスとして表示されます。このドキュメントでは、この余分な IP アドレスをダミー IP アドレスと呼んでいます)。
- ダミー IP アドレスは、NetScaler インスタンスで VIP および SNIP として構成しないでください。
- 必要に応じて、その他のセカンダリプライベート IP アドレスを作成し、VIP および SNIP として設定する必要があります。
- フェールオーバー時に、新しいプライマリノードは設定された SNIP および VIP を検索し、前のプライマリに接続されている NIC から新しいプライマリ上の対応する NIC に移動します。
- NetScaler インスタンスでは、HA が機能するためには IAM アクセス許可が必要です。各インスタンスに追加された IAM ポリシーに次の IAM 権限を追加します。

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeNetworkInterfaces"
"ec2:AssignPrivateIpAddresses"
```

> 注:`unassignPrivateIpAddress` は必須ではありません。

この方法は従来の方法よりも高速です。古い方法では、HA はプライマリノードの AWS Elastic ネットワークインターフェイスからセカンダリノードへの移行に依存します。

従来の方法では、次のポリシーが必要です。


```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeAddresses"
"ec2:AssociateAddress"
"ec2:DisassociateAddress"
```

詳細については、「AWS に高可用性ペアをデプロイする」を参照してください。

異なるゾーン間の高可用性

独立ネットワーク構成（INC）モードでは、2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンに 2 つの NetScaler ADC VPX インスタンスを高可用性アクティブ/パッシブのペアとして構成できます。フェイルオーバー時に、プライマリインスタンスの VIP の EIP (Elastic IP) がセカンダリに移行し、セカンダリが新しいプライマリとして引き継がれます。フェイルオーバープロセスでは、AWS API は以下を実行します。

- `IPSets`が接続されている仮想サーバーをチェックします。
- 仮想サーバーがリッスンしている 2 つの IP アドレスから、パブリック IP が関連付けられている IP アドレスを検索します。1 つは仮想サーバに直接接続され、もう 1 つは IP セットを介して接続されます。
- パブリック IP (EIP) を、新しいプライマリ VIP に属するプライベート IP に再関連付けします。

異なるゾーン間の HA には、次のポリシーが必要です。

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeAddresses"
"ec2:AssociateAddress"
"ec2:DisassociateAddress"
```

詳細については、「AWS アベイラビリティーゾーン全体の高可用性」を参照してください。

展開を開始する前に

AWS で HA のデプロイを開始する前に、次のドキュメントをお読みください。

- Prerequisites
- 制限事項と使用ガイドライン
- AWS で NetScaler ADC VPX インスタンスを展開する
- 高可用性

トラブルシューティング

AWS クラウド上の NetScaler ADC VPX インスタンスの HA フェイルオーバー中の障害をトラブルシューティングするには、`/var/log/`の場所に保存されている`cloud-ha-daemon.log`ファイルを確認してください。

同じ **AWS** 可用性ゾーンに **VPX HA** ペアを展開する

August 15, 2023

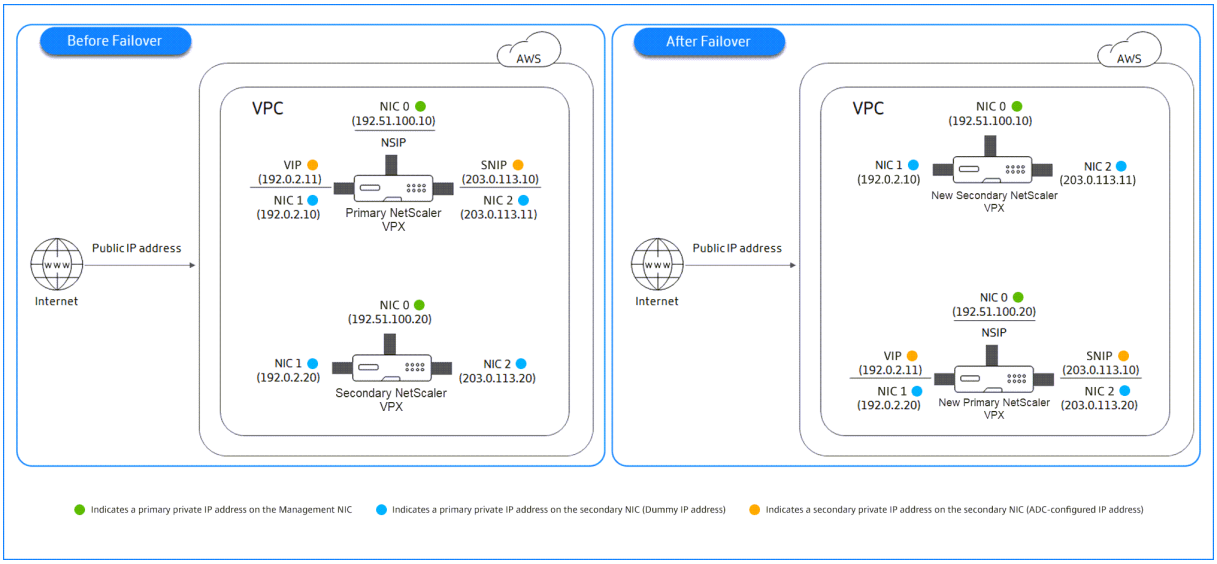
> 注:
>
> NetScaler ADC リリース 13.1 ビルド 27.x 以降、同じ AWS アベイラビリティーゾーン内の VPX HA ペアは IPv6 アドレスをサポートします。

両方の VPX インスタンスが同じサブネット上にある同じ AWS ゾーンで、AWS 上の 2 つの NetScaler ADC VPX インスタンスを HA ペアとして構成できます。HA は、フェイルオーバー後に、プライマリ HA ノードの NIC（クライアント側およびサーバ側 NIC）に接続されているセカンダリプライベート IP アドレスをセカンダリ HA ノードに移行することで実現されます。セカンダリプライベート IP アドレスに関連付けられているすべての Elastic IP アドレスも移行されます。

NetScaler VPX HA ペアは、同じ AWS アベイラビリティーゾーンで IPv4 アドレスと IPv6 アドレスの両方をサポートします。

次の図は、セカンダリプライベート IP アドレスを移行する HA フェールオーバーのシナリオを示しています。

図 1：プライベート IP マイグレーションを使用した、AWS 上の NetScaler ADC VPX HA ペア



ドキュメントを開始する前に、次のドキュメントをお読みください。

- Prerequisites
- 制限事項と使用ガイドライン
- AWS で NetScaler ADC VPX インスタンスを展開する
- 高可用性

VPX HA ペアを同じゾーンにデプロイする方法

VPX HA ペアを同じゾーンにデプロイする手順の概要を次に示します。

1. AWS で 2 つの VPX インスタンスを作成し、それぞれに NIC を 3 つ作成します。
2. AWS セカンダリプライベート IP アドレスをプライマリノードの VIP および SNIP に割り当てる
3. AWS セカンダリプライベート IP アドレスを使用してプライマリノードで VIP と SNIP を設定する
4. 両方のノードで HA を構成する

手順 **1.** 同じ **VPC** を使用して、それぞれ **3** つの **NIC**（イーサネット **0**、イーサネット **1**、イーサネット **2**）を持つ **2** つの **VPX** インスタンス（プライマリノードとセカンダリノード）を作成します

AWS ウェブコンソールを使用して、NetScaler VPX インスタンスを AWS にデプロイするに記載されている手順に従います。

手順 **2.** プライマリノードで、イーサネット **1 (**クライアント **IP** または **VIP)** とイーサネット **2 (**バックエンドサーバー **IP** または **SNIP)** にプライベート **IP** アドレスを割り当てます

AWS コンソールは、設定された NIC にプライマリプライベート IP アドレスを自動的に割り当てます。VIP と SNIP には、セカンダリプライベート IP アドレスと呼ばれる、より多くのプライベート IP アドレスを割り当てます。

プライベート IP アドレスをネットワークインターフェイスに割り当てるには、次の手順に従います。

1. https://console.aws.amazon.com/ec2/で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[**Network Interfaces**] を選択し、インスタンスにアタッチされているネットワークインターフェイスを選択します。
3. アクション **> IP** アドレスの管理を選択します。
4. 要件に基づいて [**IPv4** アドレス] または [**IPv6** アドレス] を選択します。
5. IPv4 アドレスの場合:

    a) [新しい **IP** を割り当てる] を選択します。
    b) インスタンスのサブネット範囲内にある特定の IPv4 アドレスを入力するか、フィールドを空白のままにして Amazon が IP アドレスを選択するようにします。
    c)（オプション）セカンダリプライベート **IP** アドレスが既に別のネットワークインターフェイスに割り当てられている場合に、そのアドレスを再割り当てできるようにするには、**[Allow reassign]** を選択します。

6. IPv6 アドレスの場合:

    a) [新しい **IP** を割り当てる] を選択します。
    b) インスタンスのサブネット範囲内にある特定の IPv6 アドレスを入力するか、フィールドを空白のままにして Amazon が IP アドレスを選択できるようにします。

c)（オプション）プライマリまたはセカンダリプライベート **IP** アドレスが既に別のネットワークインターフェイスに割り当てられている場合に、そのアドレスを再割り当てできるようにするには、**[Allow reassign]** を選択します。

7. はい **>** 更新を選択します。

インスタンスの説明の下に、割り当てられたプライベート IP アドレスが表示されます。

> 注:
>
> IPv4 HA ペア展開では、インターフェイスにセカンダリ IPv4 アドレスのみを割り当て、それらを VIP アドレスおよび SNIP アドレスとして使用できます。ただし、IPv6 HA ペア展開では、インターフェイスでプライマリ IPv6 アドレスまたはセカンダリ IPv6 アドレスを割り当て、それらを VIP アドレスおよび SNIP アドレスとして使用できます。

手順 **3.** セカンダリプライベート **IP** アドレスを使用して、プライマリノードで **VIP** と **SNIP** を構成します

SSH を使用してプライマリノードにアクセスします。ssh クライアントを開き、次のように入力します。

```
ssh -i <location of your private key> nsroot@<public DNS of the
   instance>
<!--NeedCopy-->
```

次に、VIP と SNIP を設定します。

VIP の場合は、次のように入力します。

```
add ns ip <IPAddress> <netmask> -type <type>
<!--NeedCopy-->
```

[SNIP] に、次のように入力します。

```
add ns ip <IPAddress> <netmask> -type SNIP
<!--NeedCopy-->
```

`save config`を入力して保存します。

設定された IP アドレスを表示するには、次のコマンドを入力します。

```
show ns ip
<!--NeedCopy-->
```

詳しくは、次のトピックを参照してください:

- 仮想 IP（VIP）アドレスの構成と管理
- NSIP アドレスの構成

ステップ **4:** 両方のインスタンスで **HA** を設定する

プライマリノードでシェルクライアントを開き、次のコマンドを入力します。

```
add ha node <id> <private IP address of the management NIC of the
    secondary node>
<!--NeedCopy-->
```

セカンダリノードで、次のコマンドを入力します。

```
add ha node <id> < private IP address of the management NIC of the
    primary node >
<!--NeedCopy-->
```

`save config` と入力して、設定を保存します。

構成された HA ノードを表示するには、`show ha node`と入力します。

フェイルオーバー時に、前のプライマリノードで VIP および SNIP として構成されたセカンダリプライベート IP アドレスは、新しいプライマリノードに移行されます。

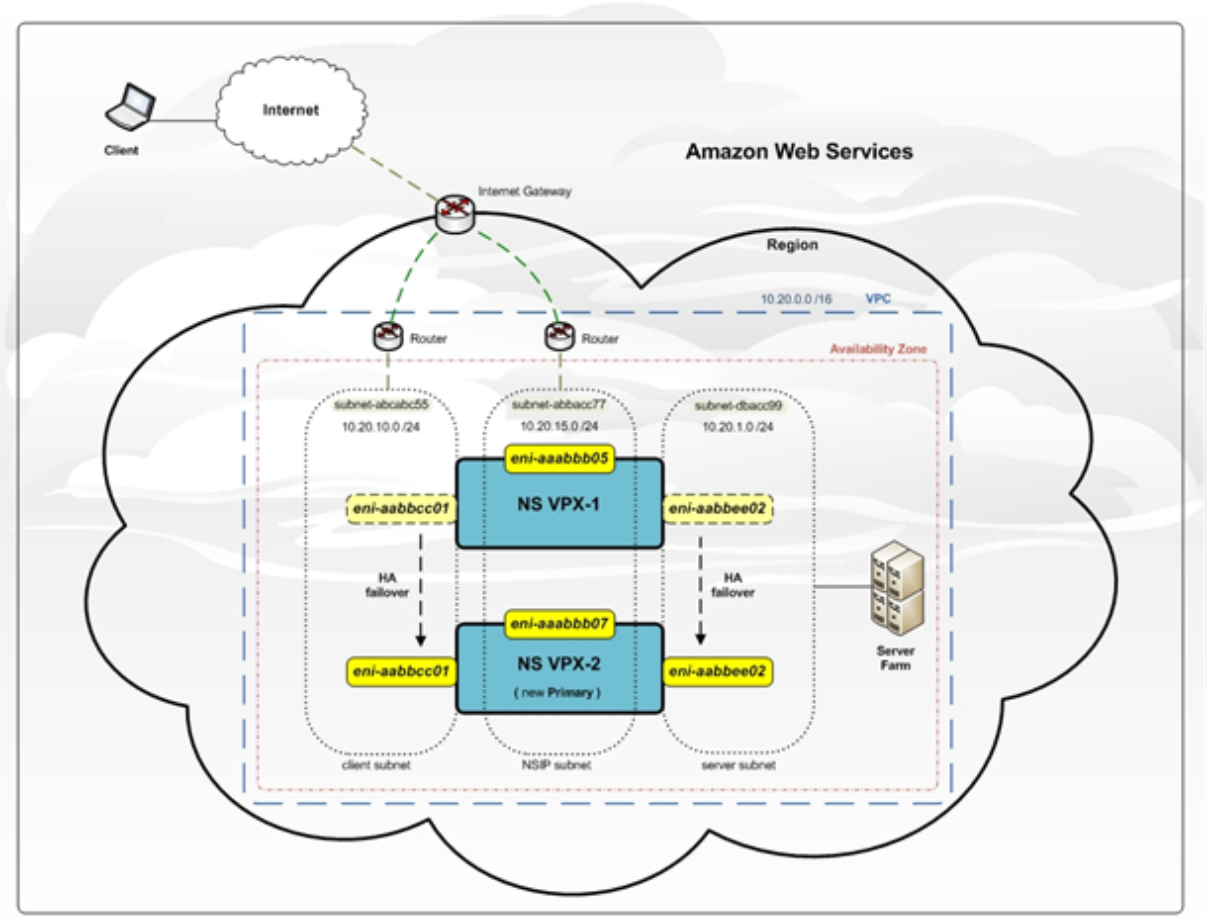
ノードでフェイルオーバーを強制するには、`force HA`failover と入力します。

VPX HA ペアをデプロイするレガシーメソッド

13.0 41.x より前のリリースでは、AWS Elastic Network Interface（ENI）の移行により、同じゾーン内の HA が実現されていました。ただし、このメソッドは徐々に非推奨になっています。

次の図は、AWS 上の NetScaler VPX インスタンスの HA デプロイメントアーキテクチャの例を示しています。

図 1：AWS での NetScaler VPX HA ペア（ENI 移行を使用）



次のいずれかのオプションを使用して、2 つの VPX インスタンスを HA ペアとして AWS にデプロイできます。

- AWS マネジメントコンソールを使用して IAM ロールを持つインスタンスを手動で作成し、そのインスタンスに HA を設定します。
- または、Citrix CloudFormation テンプレートを使用して高可用性展開を自動化することもできます。

CloudFormation テンプレートは、HA ペアの作成に必要なステップ数を大幅に減らし、IAM ロールを自動的に作成します。このセクションでは、Citrix CloudFormation テンプレートを使用して NetScaler ADC VPX HA（アクティブ-パッシブ）ペアをデプロイする方法を示します。

2 つの NetScaler ADC VPX インスタンスを HA ペアとして展開する場合は、次の点に注意してください。

注意事項

- AWS の HA では、プライマリノードに少なくとも 2 つの ENI（1 つは管理用、もう 1 つはデータトラフィック用）があり、セカンダリノードには管理 ENI が 1 つ必要です。ただし、セキュリティ上の理由から、プライマリノードに 3 つの ENI を作成します。この設定では、プライベートネットワークとパブリックネットワークを分離できるためです（推奨）。

- セカンダリノードの ENI インターフェイスは常に 1 つで（管理用）、プライマリノードには最大 4 つの ENI を設定できます。
- 高可用性ペアの各 VPX インスタンスの NSIP アドレスは、インスタンスのデフォルト ENI で構成する必要があります。
- Amazon では、AWS ではブロードキャスト/マルチキャストパケットは許可されていません。その結果、HA セットアップでは、プライマリ VPX インスタンスに障害が発生すると、データプレーン ENI がプライマリ VPX インスタンスからセカンダリ VPX インスタンスに移行されます。
- デフォルト (管理) ENI を別の VPX インスタンスに移動することはできないため、クライアントとサーバーのトラフィック (データプレーントラフィック) にはデフォルトの ENI を使用しないでください。
- /var/log/ns.log の AWSCONFIG IOCTL NSAPI_HOTPLUG_INTF 成功出力 0 というメッセージは、2 つのデータ ENI がセカンダリインスタンス（新しいプライマリ）に正常にアタッチされたことを示しています。
- AWS の ENI デタッチ/アタッチにより、フェールオーバーには最大 20 秒かかる場合があります。
- フェールオーバー後、障害が発生したインスタンスは必ず再起動します。
- ハートビートパケットは、管理インターフェイスでのみ受信されます。
- プライマリ VPX インスタンスとセカンダリ VPX インスタンスの設定ファイルは、`nsroot` パスワードを含めて同期されます。セカンダリノードの`nsroot` パスワードは、HA 構成の同期後にプライマリノードのパスワードに設定されます。
- AWS API サーバーにアクセスするには、VPX インスタンスにパブリック IP アドレスが割り当てられているか、VPC のインターネットゲートウェイを指す VPC サブネットレベルでルーティングが正しく設定されている必要があります。
- ネームサーバー/DNS サーバーが VPC レベルの DHCP オプションにより構成されている必要があります。
- Citrix CloudFormation テンプレートでは、異なるアベイラビリティーゾーン間で HA セットアップは作成されません。
- Citrix CloudFormation テンプレートでは、INC モードは作成されません。
- AWS デバッグメッセージは、VPX インスタンスのログファイル /var/log/ns.log にあります。

Citrix CloudFormation テンプレートを使用して高可用性ペアをデプロイする

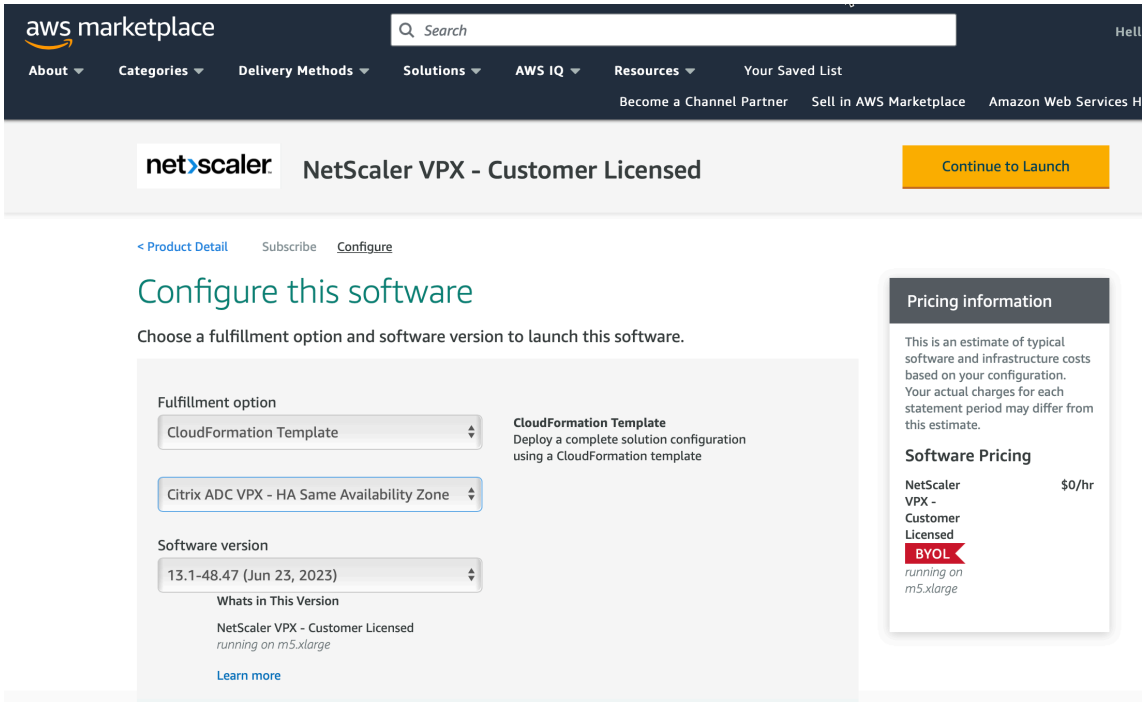
CloudFormation テンプレートを開始する前に、次の要件を満たしていることを確認してください。

- VPC
- VPC 内の 3 つのサブネット
- UDP 3003、TCP 3009—3010、HTTP、SSH ポートが開いているセキュリティグループ
- キーペア
- インターネットゲートウェイを作成する
- クライアントネットワークと管理ネットワークのルートテーブルを編集して、インターネットゲートウェイを指すようにする

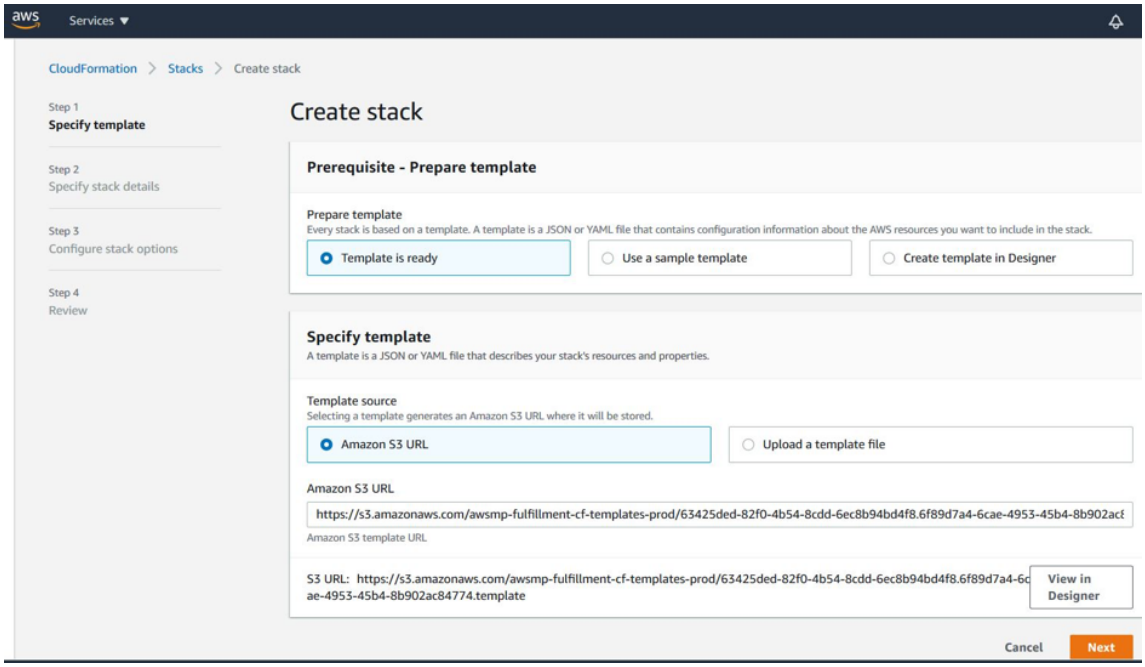
> 注
>
> Citrix CloudFormation テンプレートは、IAM ロールを自動的に作成します。既存の IAM ロールはテンプレートには表示されません。

Citrix CloudFormation テンプレートを起動するには、次の手順に従います。

1. AWS 認証情報を使用して AWS マーケットプレイスにログオンします 。
2. 検索フィールドに「**NetScaler VPX**」と入力して NetScaler AMI を検索し、[**Go**] をクリックします。
3. 検索結果ページで、目的の NetScaler ADC VPX 製品をクリックします。
4. 「価格設定」タブをクリックして、「価格情報」に移動します。
5. リージョンとフルフィルメントオプションを「**NetScaler VPX-**カスタマーライセンス」として選択します。
6. [続行] をクリックして購読します。
7. [購読] ページで詳細を確認し、[構成に進む] をクリックします。
8. **CloudFormation** テンプレートとして [配信方法] を選択します。
9. 必要な CloudFormation テンプレートを選択します。
10. [****** ソフトウェアのバージョンとリージョン] を選択し、[****** 続行] をクリックして起動します。

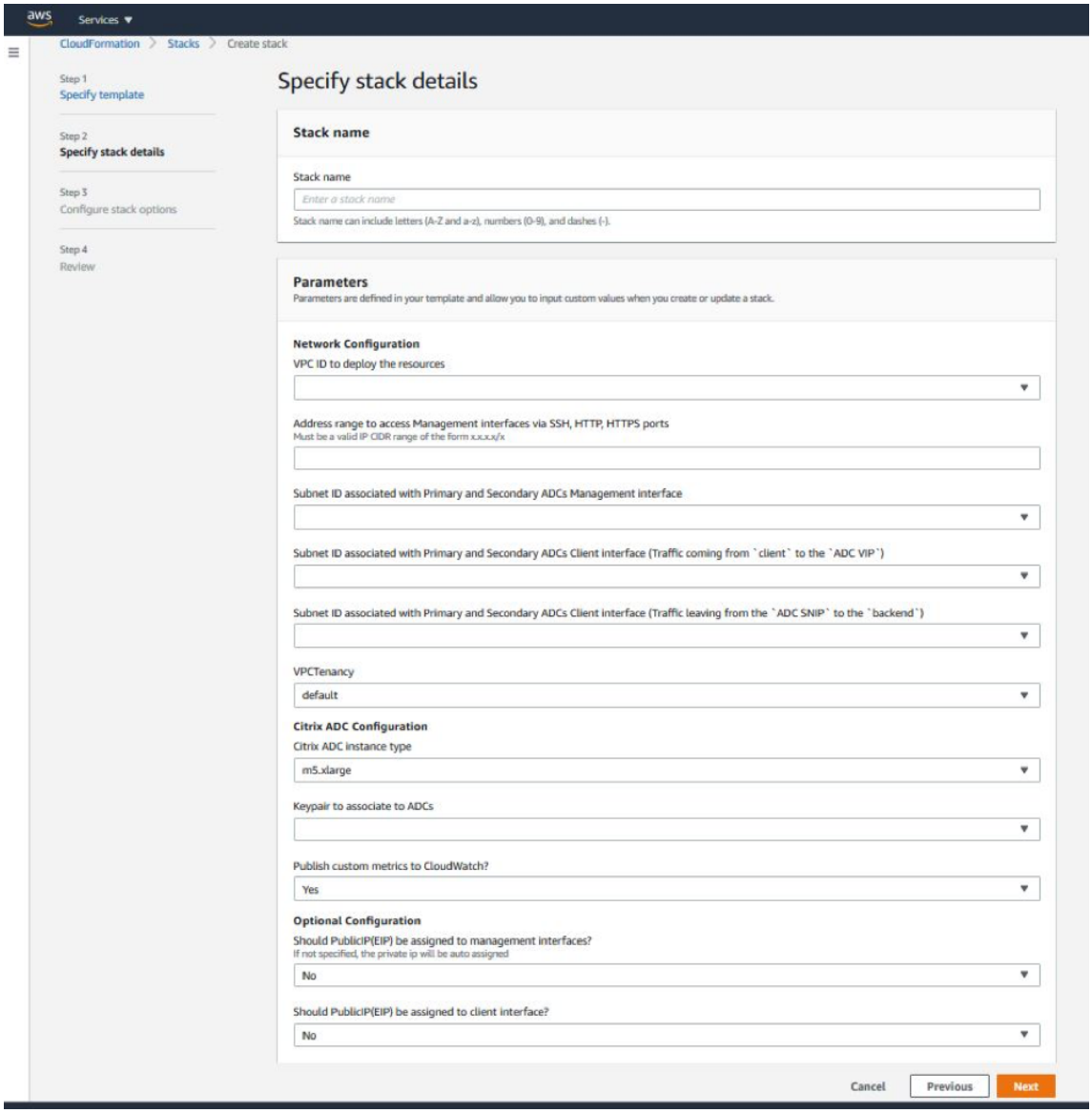


11. [アクションの選択] で、[**CloudFormation** の起動] を選択し、[起動] をクリックします。
    [スタックの作成] ページが表示されます。
12. [次へ] をクリックします。

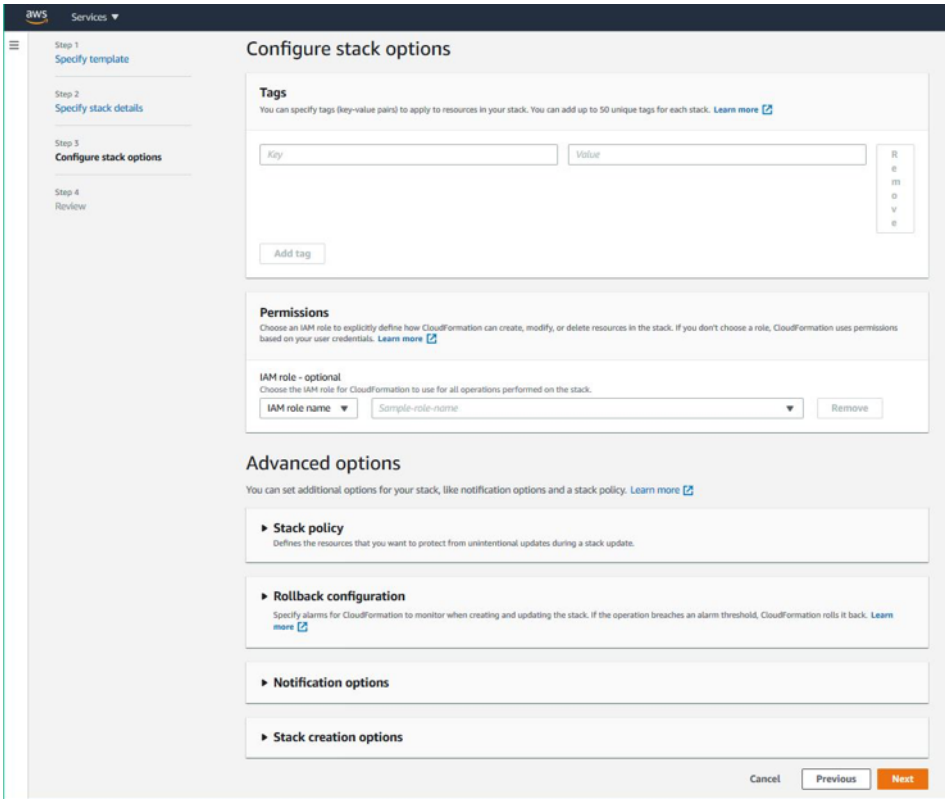


13. **[**スタックの詳細を指定] ページが表示されます。次の詳細を入力します。

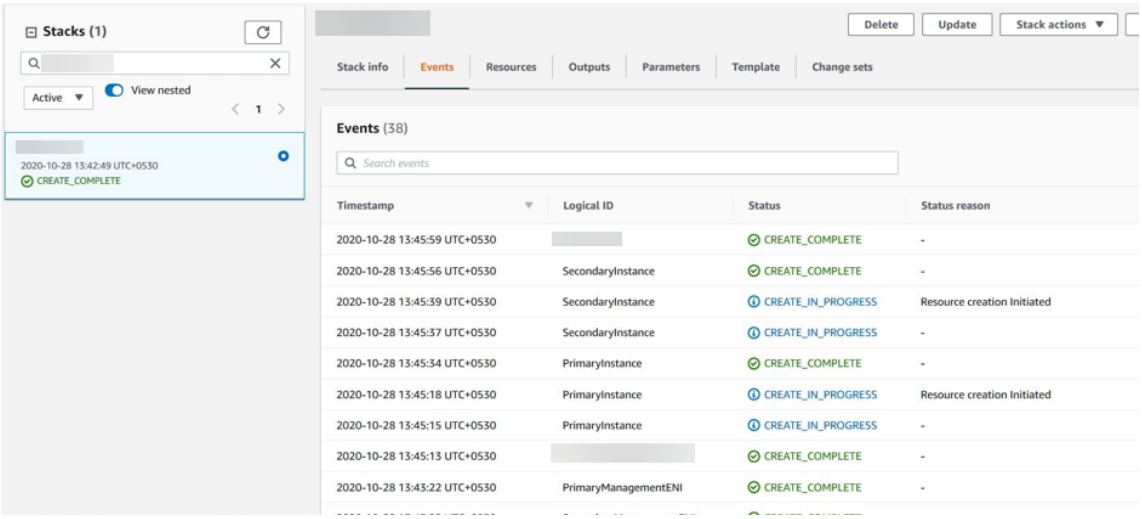
    - スタック名を入力します。名前は 25 文字以内である必要があります。
    - [ネットワーク構成] で、次の操作を実行します。
        - [管理サブネットワーク]、[クライアントサブネットワーク**]**、および **[**サーバーサブネットワーク**]** を選択します。[VPC ID] で選択した VPC 内で作成した正しいサブネットワークを選択していることを確認します。
        - プライマリ管理 **IP**、セカンダリ管理 **IP**、クライアント **IP**、およびサーバ **IP** を追加します。IP アドレスは、それぞれのサブネットワークの同じサブネットに属している必要があります。または、テンプレートに IP アドレスが自動的に割り当てられるようにすることもできます。
        - **vpcTenancy** で [デフォルト] を選択します。
    - **NetScaler** 構成で、以下を実行します。
        - [インスタンスタイプ] で [**m5.xlarge**] を選択します。
        - [Key Pair] のメニューから、作成済みのキーペアを選択します。
        - デフォルトでは、カスタムメトリクスを **CloudWatch** にパブリッシュしますか? オプションが [はい**]** に設定されています。このオプションを無効にするには、[いいえ] を選択します。CloudWatch メトリクスの詳細については、「Amazon CloudWatch を使用してインスタンスを監視する」を参照してください。
    - [オプション構成] で、次の操作を行います。
        - デフォルトでは、**PublicIP**（**EIP**）を管理インターフェイスに割り当てる必要がありますか オプションが [いいえ] に設定されています。
        - デフォルトでは、**PublicIP (EIP)** をクライアントインターフェイスに割り当てる必要がありますか オプションが [いいえ] に設定されています。



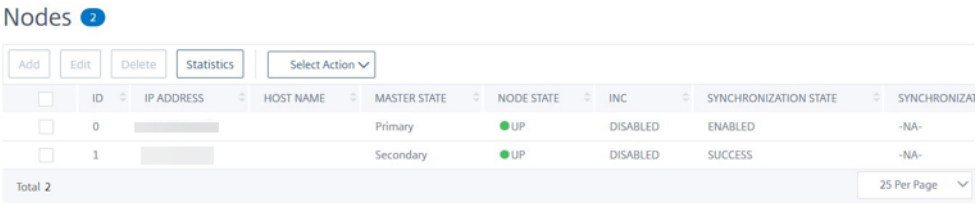
14. ［次へ］をクリックします。
15. **［スタックオプションの設定］** ページが表示されます。これはオプションのページです。



16. ［次へ］をクリックします。
17. [オプション**]** ページが表示されます。(これはオプションのページです)。［次へ］をクリックします。
18. [**Review**] ページが表示されます。しばらくして、設定を確認し、必要に応じて変更を加えます。
19. [**AWS CloudFormation** が **IAM** リソースを作成する可能性があることを承認します**]** を選択します。チェックボックスをオンにし、[スタックを作成] をクリックします。
20. **CREATE-IN-PROGRESS** が表示されます。ステータスが **CREATE-COMPLETE** になるまで待ちます。ステータスが **COMPLETE** に変更されない場合は、[**Events**] タブで失敗の原因を確認し、適切な構成でインスタンスを再作成します。



21. IAM リソースが作成されたら、[**EC2** マネジメントコンソール] **>** [インスタンス] に移動します。IAM ロールで作成された 2 つの VPX インスタンスがあります。プライマリノードとセカンダリノードは、それぞれ 3 つのプライベート IP アドレスと 3 つのネットワークインターフェイスを使用して作成されます。
22. ユーザー名`nsroot`とインスタンス ID をパスワードとしてプライマリノードにログオンします。GUI から、[システム] **>** [高可用性] **>** [ノード] に移動します。NetScaler VPX は、CloudFormation テンプレートによって HA ペアで既に構成されています。
23. NetScaler VPX HA ペアが表示されます。



Amazon CloudWatch を使用してインスタンスをモニタリングする

Amazon CloudWatch サービスを使用して、CPU とメモリの使用率、スループットなど、一連の NetScaler VPX メトリクスを監視できます。CloudWatch は AWS で実行されるリソースとアプリケーションをリアルタイムでモニタリングします。AWS マネジメントコンソールを使用して、Amazon CloudWatch ダッシュボードにアクセスできます。詳細については、「Amazon CloudWatch」を参照してください。

注意事項

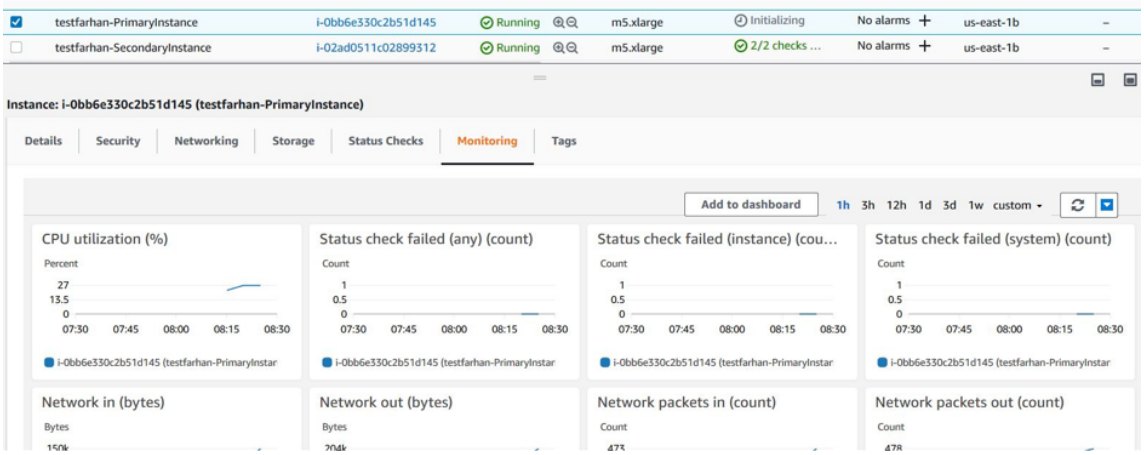
- AWS ウェブコンソールを使用して AWS に NetScaler VPX インスタンスをデプロイすると、CloudWatch サービスはデフォルトで有効になります。

- Citrix CloudFormation テンプレートを使用して NetScaler ADC VPX インスタンスをデプロイする場合、デフォルトのオプションは「はい」です。CloudWatch サービスを無効にする場合は、[いいえ] を選択します。
- メトリクスは、CPU (管理およびパケット CPU 使用率)、メモリ、およびスループット (インバウンドとアウトバウンド) で使用できます。

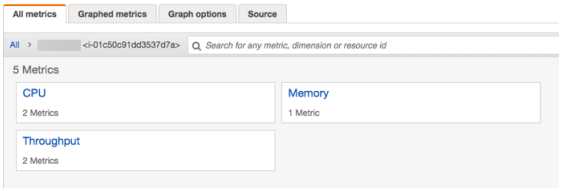
CloudWatch メトリクスの表示方法

インスタンスの CloudWatch メトリックスを表示するには、次の手順に従います。

1. **AWS** マネジメントコンソール **> EC2 >** インスタンスにログオンします。
2. インスタンスを選択します。
3. [監視] をクリックします。
4. [**CloudWatch** メトリックスをすべて表示] をクリックします。

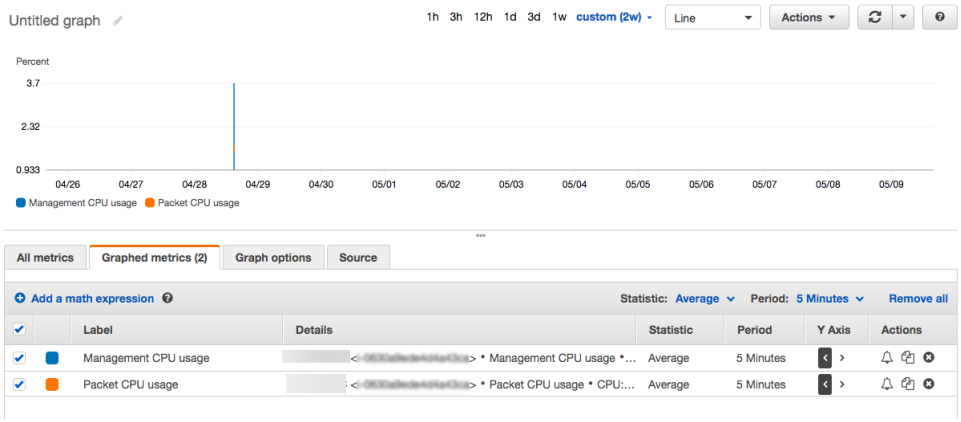


5. [すべてのメトリックス] で、インスタンス ID をクリックします。



6. 表示するメトリクスをクリックし、期間（分、時間、日、週、月）を設定します。
7. グラフ化されたメトリクスをクリックして、使用量の統計を表示します。グラフをカスタマイズするには、**[**グラフ**]** オプションを使用します。

フィギュア。CPU 使用率に関するグラフ化されたメトリック



高可用性セットアップでの **SR-IOV** の設定

高可用性セットアップでの SR-IOV インターフェイスのサポートは、NetScaler ADC リリース 12.0 57.19 以降から利用できます。SR-IOV を構成する方法の詳細については、「SR-IOV ネットワークインターフェイスを使用するように NetScaler ADC VPX インスタンスを構成する」を参照してください。

関連情報

AWS での高可用性の機能

さまざまな **AWS** 可用性ゾーンにわたる高可用性

March 20, 2024

独立ネットワーク構成（INC）モードでは、2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンに 2 つの NetScaler VPX インスタンスを高可用性アクティブ/パッシブのペアとして構成できます。何らかの理由で 1 次ノードが接続を受け入れることができない場合は、2 次ノードが引き継ぎます。

高可用性の詳細については、「高可用性」を参照してください。INC の詳細については、「異なるサブネットでの高可用性ノードの設定」を参照してください。

注意事項

- 配置を開始する前に、次のドキュメントをお読みください。
  - AWS の用語
  - Prerequisites
  - 制限事項と使用ガイドライン

- VPX 高可用性ペアは、異なるサブネットの同じアベイラビリティーゾーンに存在することも、2 つの異なる AWS アベイラビリティーゾーンに存在することもできます。
- 管理（NSIP）、クライアントトラフィック（VIP）、バックエンドサーバー（SNIP）には異なるサブネットを使用することをお勧めします。
- フェイルオーバーを機能させるには、独立ネットワーク構成 (INC) モードで高可用性を設定する必要があります。
- 2 つのインスタンスでは、ハートビートに使用される UDP トラフィック用にポート 3003 が開いている必要があります。
- 残りの API が機能するように、両方のノードの管理サブネットは、内部 NAT を介してインターネットまたは AWS API サーバーにアクセスできる必要があります。
- IAM ロールには、パブリック IP または Elastic IP (EIP) 移行用の E2 アクセス権限と、プライベート IP 移行用の EC2 ルートテーブルのアクセス許可が必要です。

次の方法で AWS アベイラビリティーゾーン間で高可用性をデプロイできます。

- エラスティック IP アドレスの使用
- プライベート IP アドレスの使用

その他の参考資料

AWS 向け NetScaler アプリケーション配信管理（ADM）の詳細については、「AWS への NetScaler コンソールエージェントのインストール」を参照してください。

異なる **AWS** ゾーンにエラスティック **IP** アドレスを使用して **VPX** 高可用性ペアを展開する

August 15, 2023

INC モードで Elastic IP（EIP）アドレスを使用して、2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンに 2 つの NetScaler VPX インスタンスを設定できます。

高可用性の詳細については、「高可用性」を参照してください。INC の詳細については、「異なるサブネットでの高可用性ノードの設定」を参照してください。

異なる **AWS** ゾーンにわたる **EIP** アドレスを持つ **HA** のしくみ

フェールオーバー時には、プライマリインスタンスの VIP の EIP がセカンダリに移行し、セカンダリが新しいプライマリとして引き継がれます。フェイルオーバープロセスでは、AWS API は以下を行います。

1. `IPSets`が接続されている仮想サーバーをチェックします。

2. 仮想サーバーがリッスンしている 2 つの IP アドレスから、パブリック IP が関連付けられている IP アドレスを検索します。1 つは仮想サーバに直接接続され、もう 1 つは IP セットを介して接続されます。
3. パブリック IP (EIP) を、新しいプライマリ VIP に属するプライベート IP に再関連付けします。

> 注
>
> EIP を使用する際に、サービス拒否 (DoS) などの攻撃からネットワークを保護するために、AWS でセキュリティグループを作成して IP アクセスを制限できます。高可用性を実現するために、展開に従って EIP からプライベート IP 移動ソリューションに切り替えることができます。

異なる **AWS** ゾーン間でエラスティック **IP** アドレスを使用して **VPX** 高可用性ペアをデプロイする方法

VPX ペアを 2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンにデプロイする手順の概要を以下に示します。

1. Amazon 仮想プライベートクラウドを作成します。
2. 2 つの VPX インスタンスを、2 つの異なるアベイラビリティーゾーン、または同じゾーンで異なるサブネットにデプロイします。
3. 高可用性を構成する

    a) 両方のインスタンスで INC モードで高可用性をセットアップします。
    b) 両方のインスタンスに IP セットを追加します 。
    c) 両方のインスタンスの IP セットを VIP にバインドします。
    d) プライマリ・インスタンスに仮想サーバを追加します。

ステップ 1 と 2 では、AWS コンソールを使用します。手順 3 では、NetScaler VPX GUI または CLI を使用します。

手順 **1.** Amazon 仮想プライベートクラウド (VPC) を作成します。

手順 **2.** 2 つの VPX インスタンスを 2 つの異なるアベイラビリティーゾーン、または同じゾーンで異なるサブネットにデプロイします。プライマリ VPX の VIP に EIP を接続します。

VPC を作成して AWS に VPX インスタンスをデプロイする方法の詳細については、「AWS への NetScaler ADC VPX スタンドアロンインスタンスのデプロイ 」および「シナリオ: スタンドアロンインスタンス」を参照してください。

手順 **3.** 高可用性を構成します。NetScaler VPX CLI または GUI を使用して、高可用性をセットアップできます。

CLI を使用した高可用性の設定

1. 両方のインスタンスで INC モードで高可用性をセットアップします。

    プライマリノードで、次の操作を行います。

    ```
    add ha node 1 <sec_ip> -inc ENABLED
    ```


セカンダリノード:

```
add ha node 1 <prim_ip> -inc ENABLED
```

<sec_ip>はセカンダリノードの管理 NIC のプライベート IP アドレスを指します。

<prim_ip>はプライマリノードの管理 NIC のプライベート IP アドレスを指します

2. 両方のインスタンスに IP セットを追加します。

   両方のインスタンスで以下のコマンドを入力します。

   ```
   add ipset <ipsetname>
   ```

3. IP セットを両方のインスタンスの VIP セットにバインドします。

   両方のインスタンスで以下のコマンドを入力します。

   ```
   add ns ip <secondary vip> <subnet> -type VIP
   ```

   ```
   bind ipset <ipsetname> <secondary VIP>
   ```

   > 注
   >
   > IP セットは、プライマリ VIP またはセカンダリ VIP にバインドできます。ただし、IP セットをプライマリ VIP にバインドする場合は、セカンダリ VIP を使用して仮想サーバに追加し、逆にセカンダリ VIP を使用します。

4. プライマリ・インスタンスに仮想サーバを追加します。

   次のコマンドを入力します。

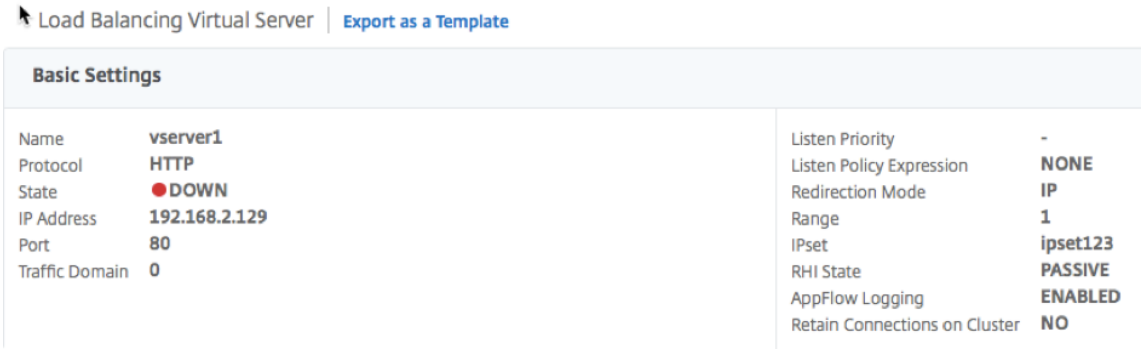
   ```
   add <server_type> vserver <vserver_name> <protocol> <primary_vip>
    <port> -ipset \<ipset_name>
   ```

GUI を使用した高可用性の構成

1. 両方のインスタンスで INC モードで高可用性をセットアップする
2. ユーザー名nsrootとインスタンス ID をパスワードとしてプライマリノードにログオンします。
3. GUI から、[構成] > [システム] > [高可用性] に移動します。[追加] をクリックします。
4. リモートノード **IP** アドレスフィールドに、2 次ノードの管理 NIC のプライベート IP アドレスを追加します。
5. [セルフノードで **NIC (**独立ネットワーク構成**)** モードをオンにする] を選択します。
6. [リモートシステムログイン認証情報] で、セカンダリノードのユーザー名とパスワードを追加し、[作成] をクリックします。
7. セカンダリノードで手順を繰り返します。
8. IP セットを追加し、IP セットを両方のインスタンスの VIP セットにバインドします。

9. GUI から、[システム] > [ネットワーク] > [**IP**] > [追加] に移動します。
10. [IP アドレス]、[ネットマスク]、[IP タイプ (仮想 IP)] に必要な値を追加し、[作成] をクリックします。
11. [システム] > [ネットワーク] > [**IP** セット] > **[**追加**]** に移動します。IP セット名を追加し、[**Insert**] をクリックします。
12. [IPv4] ページで、仮想 IP を選択し、[挿入] をクリックします。[**Create**] をクリックして IP セットを作成します。
13. プライマリ・インスタンスに仮想サーバを追加する

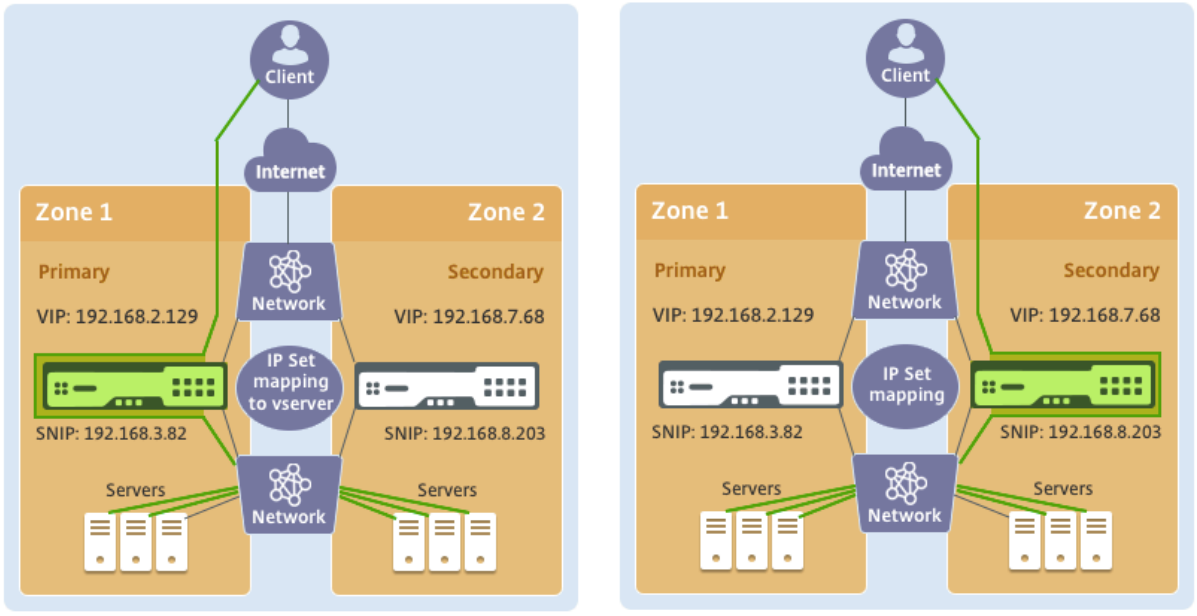
    GUI から、[構成**]**> [トラフィック管理] > [仮想サーバ**] > [**追加**]** に移動します。



シナリオ

このシナリオでは、1 つの VPC が作成されます。その VPC では、2 つのアベイラビリティーゾーンに 2 つの VPX インスタンスが作成されます。各インスタンスには、管理用、クライアント用、バックエンドサーバー用の 3 つのサブネットがあります。EIP はプライマリノードの VIP に接続されます。

図：この図は、AWS での INC モードでの NetScaler ADC VPX の高可用性セットアップを示しています



このシナリオでは、CLI を使用して高可用性を設定します。

1. 両方のインスタンスで INC モードで高可用性をセットアップします。

   プライマリノードとセカンダリノードで次のコマンドを入力します。

   プライマリで**:**

   ```
   add ha node 1 192.168.6.82 -inc enabled
   ```

   ここで、192.168.6.82 は、セカンダリノードの管理 NIC のプライベート IP アドレスを指します。

   セカンダリの場合**:**

   ```
   add ha node 1 192.168.1.108 -inc enabled
   ```

   ここで、192.168.1.108 は、プライマリノードの管理 NIC のプライベート IP アドレスを指します。

2. IP セットを追加し、IP セットを両方のインスタンスの VIP にバインドします。

   プライマリで:

   ```
   add ipset ipset123
   add ns ip 192.168.7.68 255.255.255.0 -type VIP
   bindipset ipset123 192.168.7.68
   ```

   セカンダリの場合:

   ```
   add ipset ipset123
   add ns ip 192.168.7.68 255.255.255.0 -type VIP
   bind ipset ipset123 192.168.7.68
   ```

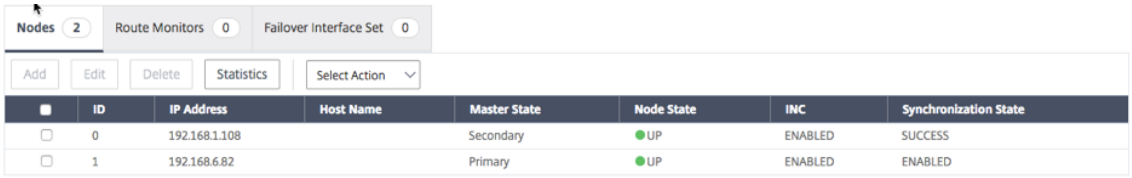

3. プライマリ・インスタンスに仮想サーバを追加します。

   以下のコマンドを実行します。

   ```
   add lbvserver vserver1 http 192.168.2.129 80 -ipset ipset123
   ```

4. 構成を保存します。

5. 強制フェールオーバーの後、セカンダリは新しいプライマリになります。



異なる **AWS** ゾーンにプライベート **IP** アドレスを使用して **VPX** 高可用性ペアを展開する

October 25, 2023

INC モードのプライベート IP アドレスを使用して、2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンで 2 つの NetScaler ADC VPX インスタンスを設定できます。このソリューションは、Elastic IP アドレスを持つ既存のマルチゾーン VPX 高可用性ペアと簡単に統合できます。したがって、両方のソリューションを一緒に使用できます。

高可用性の詳細については、「高可用性」を参照してください。INC の詳細については、「異なるサブネットでの高可用性ノードの設定」を参照してください。

> 注:
>
> このデプロイは、NetScaler リリース 13.0 ビルド 67.39 以降でサポートされています。このデプロイは AWS Transit Gateway と互換性があります。

AWS 非共有 VPC を使用したプライベート IP アドレスを使用した高可用性ペア

前提条件

AWS アカウントに関連付けられた IAM ロールに次の IAM アクセス権限があることを確認します。

```
{

    "Version": "2012-10-17",
    "Statement": [
        {

            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeAddresses",
                "ec2:AssociateAddress",
                "ec2:DisassociateAddress",
                "ec2:DescribeRouteTables",
                "ec2:DeleteRoute",
                "ec2:CreateRoute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "iam:SimulatePrincipalPolicy",
                "iam:GetRole"
            ],
            "Resource": "*",
            "Effect": "Allow"
         }

    ]
 }


<!--NeedCopy-->
```

AWS 非共有 VPC を使用して、プライベート IP アドレスを持つ VPX HA ペアをデプロイする

次に、プライベート IP アドレスを使用して 2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンに VPX ペアをデプロイする手順の概要を示します。

1. Amazon 仮想プライベートクラウドを作成します。
2. 2 つの異なるアベイラビリティーゾーンに 2 つの VPX インスタンスをデプロイします。
3. 高可用性を構成する

    a) 両方のインスタンスで INC モードで高可用性をセットアップします。
    b) クライアントインターフェイスを指すそれぞれのルートテーブルを VPC に追加します。
    c) プライマリ・インスタンスに仮想サーバを追加します。

ステップ 1、2、および 3b では、AWS コンソールを使用します。手順 3a と 3c では、NetScaler VPX GUI または CLI を使用します。

手順 **1.** Amazon 仮想プライベートクラウド (VPC) を作成します。

手順 **2**: 同じ数の ENI (ネットワークインターフェイス) を持つ 2 つの異なるアベイラビリティーゾーンに 2 つの VPX インスタンスをデプロイします。

VPC を作成して AWS に VPX インスタンスをデプロイする方法の詳細については、「AWS への NetScaler ADC VPX スタンドアロンインスタンスのデプロイ 」および「シナリオ: スタンドアロンインスタンス」を参照してください。

手順 **3.** Amazon VPC サブネットと重複しないサブネットを選択して、ADC VIP アドレスを設定します。VPC が 192.168.0.0/16 の場合、ADC VIP アドレスを設定するには、次の IP アドレス範囲から任意のサブネットを選択できます。

- 0.0.0.0-192.167.0.0
- 192.169.0.0-254.255.255.0

この例では、10.10.10.0/24 サブネットを選択し、このサブネットに VIP を作成しました。VPC サブネット以外の任意のサブネットを選択できます (192.168.0.0/16).

ステップ **4:** VPC ルートテーブルから、プライマリノードのクライアントインターフェイス (VIP) を指すルートを追加します。

AWS CLI から、次のコマンドを入力します。

```
aws ec2 create-route --route-table-id rtb-2272532 --destination-cidr-
    block 10.10.10.0/24 --gateway-id <eni-client-primary>
<!--NeedCopy-->
```

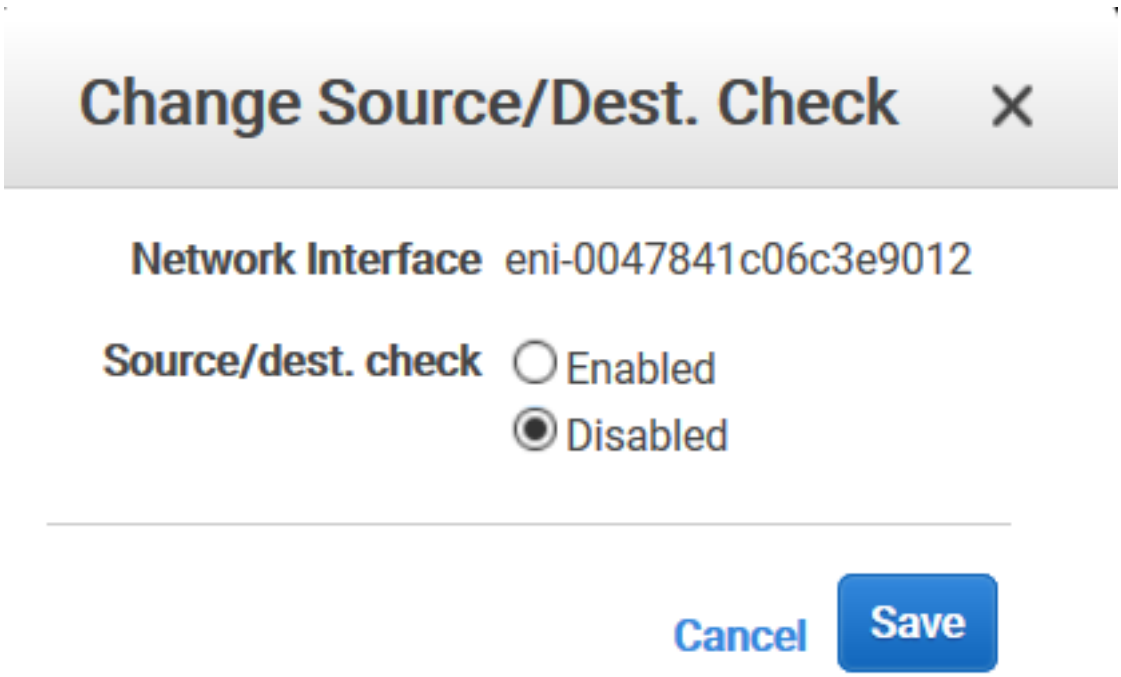
AWS GUI から、次の手順を実行してルートを追加します。

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、ルートテーブルを選択し、ルートテーブルを選択します。
3. [アクション] を選択し、[ルートの編集] をクリックします。
4. ルートを追加するには、[**Add route**] を選択します。[宛先**]** に、宛先 CIDR ブロック、単一の IP アドレス、またはプレフィックスリストの ID を入力します。ゲートウェイ ID には、プライマリノードのクライアントインターフェイスの ENI を選択します。

> 注:
>
> プライマリ・インスタンスのクライアント ENI で **Source/Dest Check** を無効にする必要があります。

コンソールを使用してネットワークインターフェイスの source/destination チェックを無効にするには、次の手順を実行します。

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[ネットワークインターフェイス] を選択します。
3. プライマリクライアントインターフェイスのネットワークインターフェイスを選択し、[アクション] を選択し、[ソース/デストを変更] をクリックします。を確認してください。
4. ダイアログボックスで、[無効] を選択し、[保存] をクリックします。



手順 **5.** 高可用性を構成します。NetScaler VPX CLI または GUI を使用して、高可用性をセットアップできます。

CLI を使用した高可用性の設定

1. 両方のインスタンスで INC モードで高可用性をセットアップします。

   プライマリノードで、次の操作を行います。

   ```
   1 add ha node 1 <sec_ip> -inc ENABLED
   2 <!--NeedCopy-->
   ```

   セカンダリノード:

   ```
   1 add ha node 1 <prim_ip> -inc ENABLED
   2 <!--NeedCopy-->
   ```

   <sec_ip> セカンダリノードの管理 NIC のプライベート IP アドレスを参照します。

   <prim_ip> プライマリノードの管理 NIC のプライベート IP アドレスを参照します。

2. プライマリ・インスタンスに仮想サーバを追加します。選択したサブネット（10.10.10.0/24 など）から追加する必要があります。

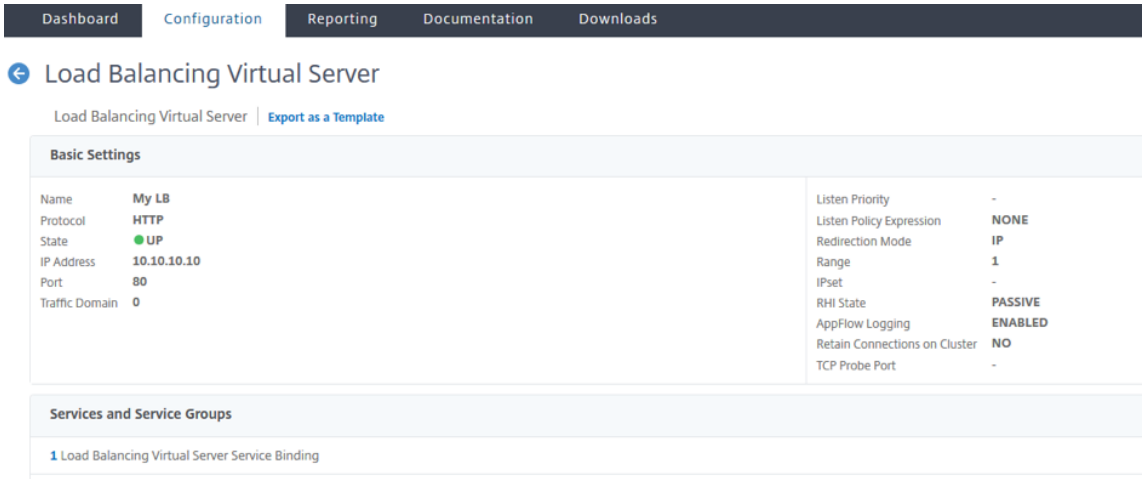
   次のコマンドを入力します。

   ```
   1 add \<server\_type\> vserver \<vserver\_name\> \<protocol\> \<
       primary\_vip\> \<port\>
   2 <!--NeedCopy-->
   ```

GUI を使用した高可用性の構成

1. 両方のインスタンスで INC モードで高可用性をセットアップする
2. ユーザー名nsrootとインスタンス ID をパスワードとしてプライマリノードにログオンします。
3. [構成] > [システム] > [高可用性] に移動し、[追加] をクリックします。
4. リモートノード **IP** アドレスフィールドに、2 次ノードの管理 NIC のプライベート IP アドレスを追加します。
5. [セルフノードで **NIC (**独立ネットワーク構成**)** モードをオンにする] を選択します。
6. [リモートシステムログイン認証情報] で、セカンダリノードのユーザー名とパスワードを追加し、[作成] をクリックします。
7. セカンダリノードで手順を繰り返します。
8. プライマリ・インスタンスに仮想サーバを追加する

   [設定] > [トラフィック管理] > [仮想サーバー**] > [**追加**]** に移動します。



AWS 共有 VPC を使用して、プライベート IP アドレスを持つ VPX HA ペアをデプロイする

AWS 共有 VPC モデルでは、VPC を所有するアカウント (所有者) が 1 つ以上のサブネットを他のアカウント (参加者) と共有します。そのため、VPC 所有者アカウントと参加者アカウントがあります。サブネットが共有されると、参加者は共有されたサブネット内のアプリケーションリソースを表示、作成、変更、および削除できます。参加者は、他の参加者または VPC 所有者に属するリソースを表示、変更、削除することはできません。

AWS 共有 VPC の詳細については、AWS ドキュメントを参照してください。

> 注:
>
> AWS 共有 VPC を使用してプライベート IP アドレスを持つ VPX HA ペアをデプロイする設定手順は、AWS 非共有 VPC を使用してプライベート IP アドレスを持つ VPX HA ペアをデプロイすると同じですが 、次の例外があります。
>
> - クライアントインターフェイスを指す VPC 内のルートテーブルは、*VPC* 所有者アカウントから追加する必要があります。

前提条件

- AWS 参加者アカウントの NetScaler VPX インスタンスに関連付けられている IAM ロールに次の IAM 権限があることを確認してください。

```
"Version": "2012-10-17",
    "Statement": [
        {

            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:DisassociateAddress",
                "iam:GetRole",
```



```
                    "iam:SimulatePrincipalPolicy",
                    "ec2:DescribeInstances",
                    "ec2:DescribeAddresses",
                    "ec2:ModifyNetworkInterfaceAttribute",
                     “ec2:AssociateAddress” ,
                    "sts:AssumeRole"
            ],
                "Resource": "*"
             }

        ]
    }

<!--NeedCopy-->
```

> 注記:
>
> **AssumeRole** を使用すると、NetScaler VPX インスタンスは、VPC 所有者アカウントによって作成されたクロスアカウント IAM ロールを引き継ぐことができます。

- VPC 所有者アカウントが、クロスアカウント IAM ロールを使用して、参加者アカウントに次の IAM アクセス権限を付与していることを確認します。

```
{

    "Version": "2012-10-17",
    "Statement": [
        {

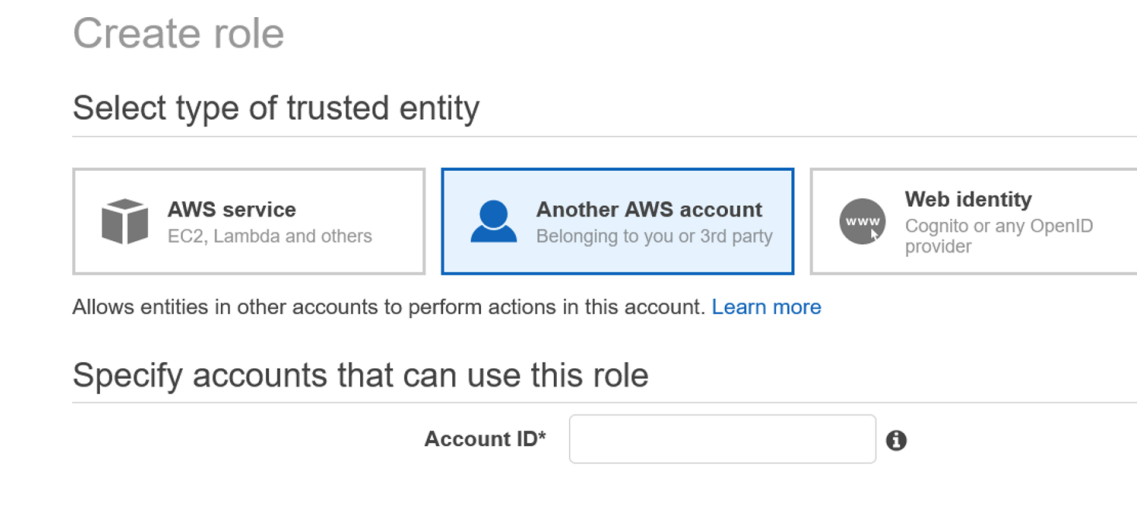
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateRoute",
                "ec2:DeleteRoute",
                "ec2:DescribeRouteTables"
            ],
            "Resource": "*"
         }

    ]
 }

<!--NeedCopy-->
```

クロスアカウント **IAM** ロールの作成

1. AWS ウェブコンソールにログインします。
2. [**IAM**] タブで [ロール] に移動し、[**** ロールの作成 **] を選択します。
3. [別の **AWS** アカウント] を選択します。



4. 管理者アクセス権を付与する参加者アカウントの 12 桁のアカウント ID 番号を入力します。

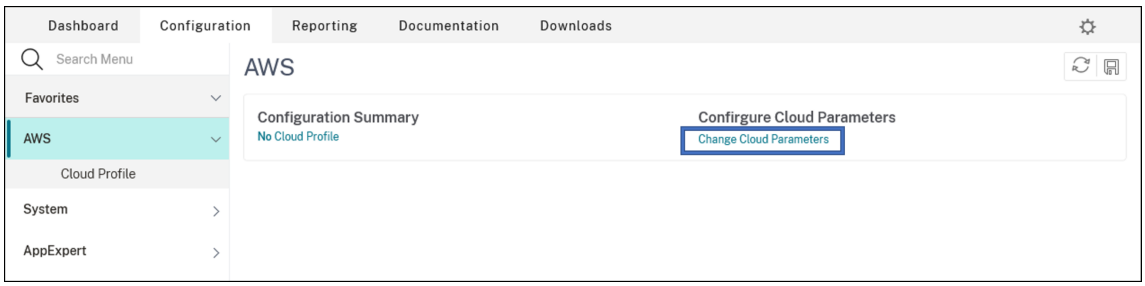
NetScaler CLI を使用してクロスアカウント **IAM** ロールを設定する

次のコマンドを実行すると、NetScaler VPX インスタンスが VPC 所有者アカウントに存在するクロスアカウント IAM ロールを引き継ぐことができます。

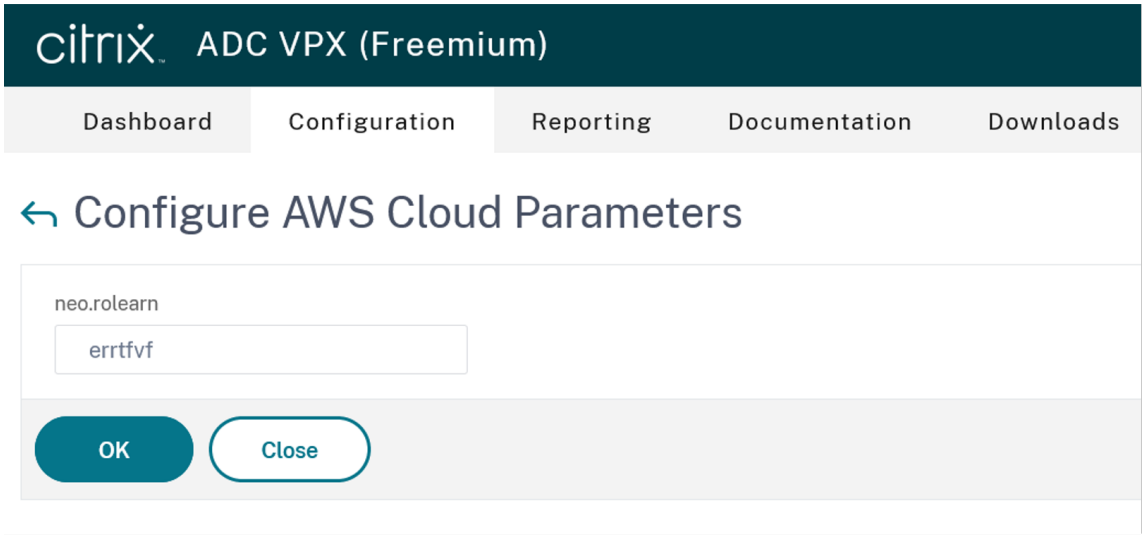
```
set cloud awsParam -roleARN <string>
<!--NeedCopy-->
```

NetScaler GUI を使用してクロスアカウント **IAM** ロールを設定

1. NetScaler アプライアンスにサインインし、[構成] **> [AWS] > [**クラウドパラメータの変更] に移動します。



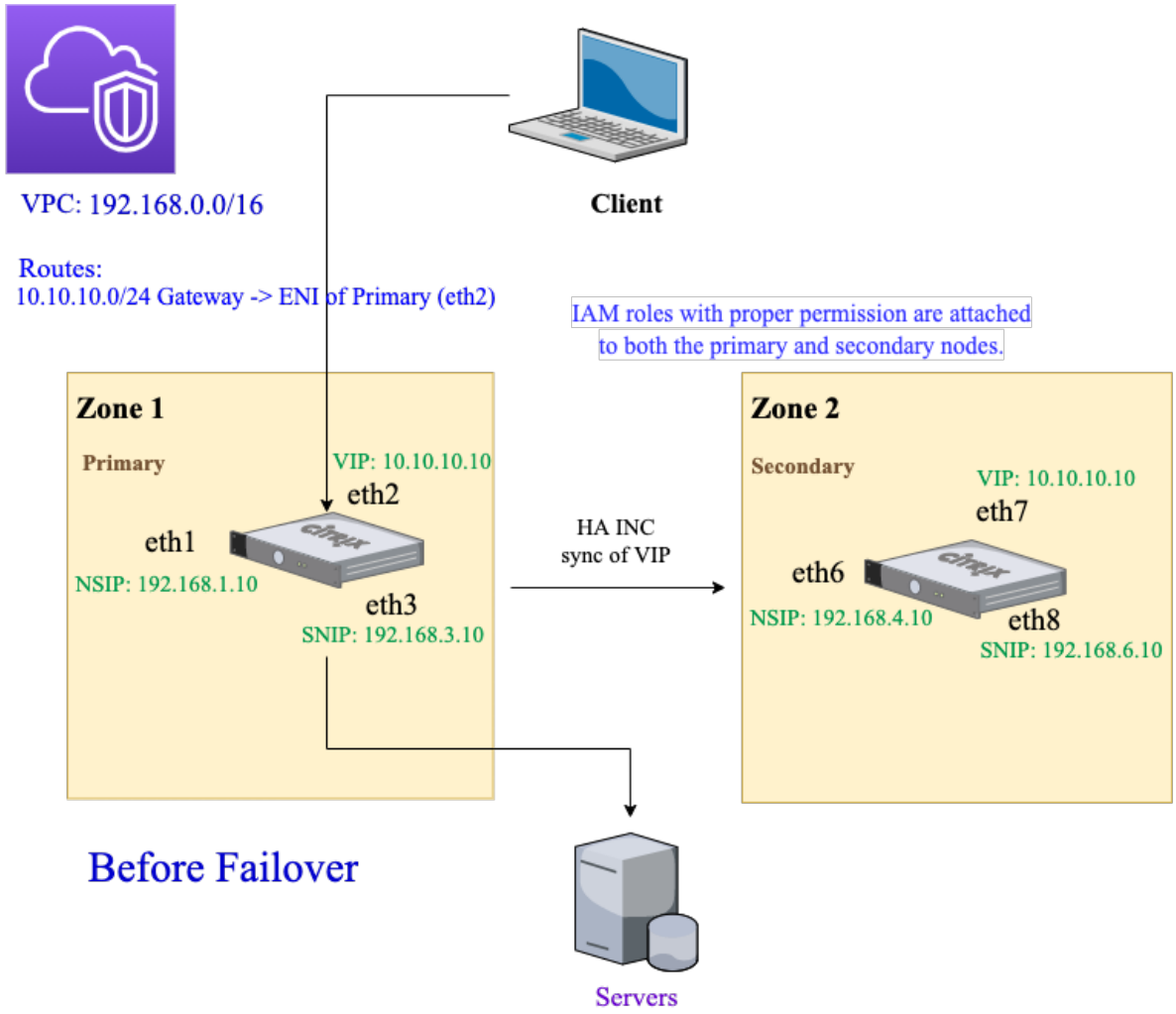
2. **[AWS** クラウドパラメータの設定] ページで、[**roLearn**] フィールドに値を入力します。

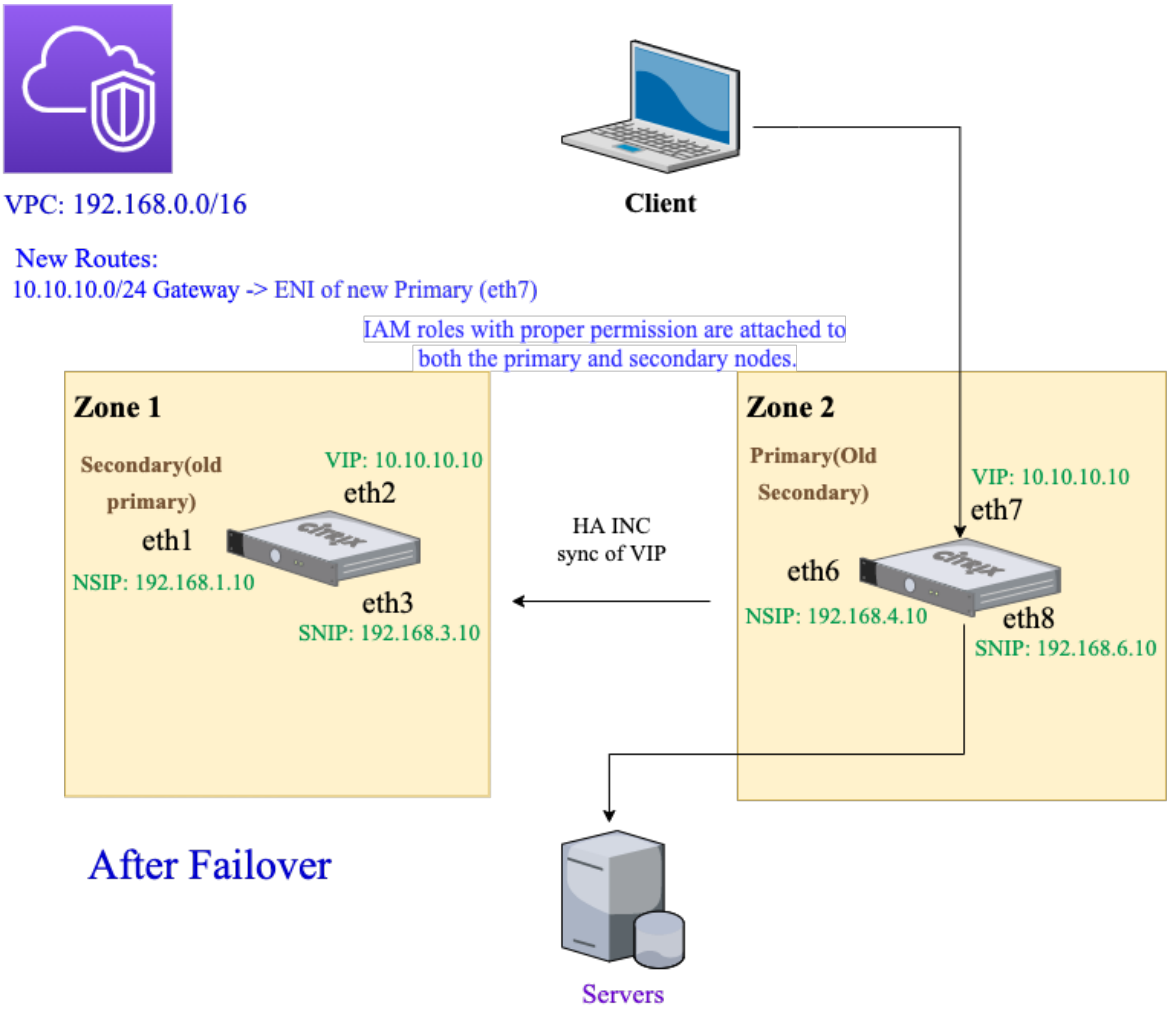


シナリオ

このシナリオでは、1 つの VPC が作成されます。その VPC では、2 つのアベイラビリティーゾーンに 2 つの VPX インスタンスが作成されます。各インスタンスには、管理用、クライアント用、バックエンドサーバー用の 3 つのサブネットがあります。

次の図は、AWS 上の INC モードでの NetScaler VPX 高可用性セットアップを示しています。VPC の一部ではないカスタムサブネット 10.10.10.10 が VIP として使用されます。したがって、10.10.10.10 サブネットはアベイラビリティーゾーン全体で使用できます。





このシナリオでは、CLI を使用して高可用性を設定します。

1. 両方のインスタンスで INC モードで高可用性をセットアップします。

   プライマリノードとセカンダリノードで次のコマンドを入力します。

   プライマリノードで、次の操作を行います。

   ```
   1 add ha node 1 192.168.4.10 -inc enabled
   2 <!--NeedCopy-->
   ```

   ここで、192.168.4.10 は、セカンダリノードの管理 NIC のプライベート IP アドレスを指します。

   セカンダリノード:

   ```
   1 add ha node 1 192.168.1.10 -inc enabled
   2 <!--NeedCopy-->
   ```

   ここで、192.168.1.10 は、プライマリノードの管理 NIC のプライベート IP アドレスを指します。

2. プライマリ・インスタンスに仮想サーバを追加します。

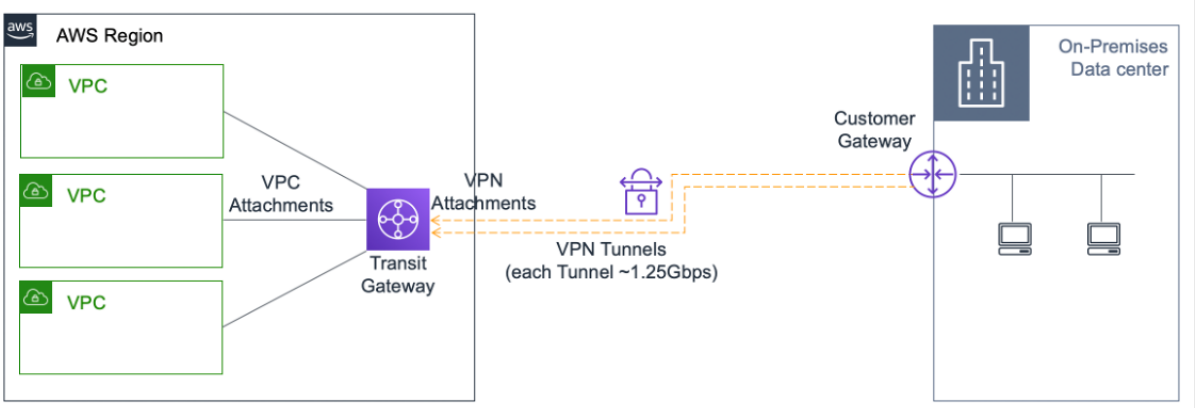
次のコマンドを入力します。

```
1 add lbvserver vserver1 http 10.10.10.10 80
2 <!--NeedCopy-->
```

3. 構成を保存します。
4. 強制フェールオーバーの後:
   - セカンダリインスタンスが新しいプライマリインスタンスになります。
   - プライマリ ENI を指す VPC ルートは、セカンダリクライアント ENI に移行します。
   - クライアントトラフィックは、新しいプライマリインスタンスに再開されます。

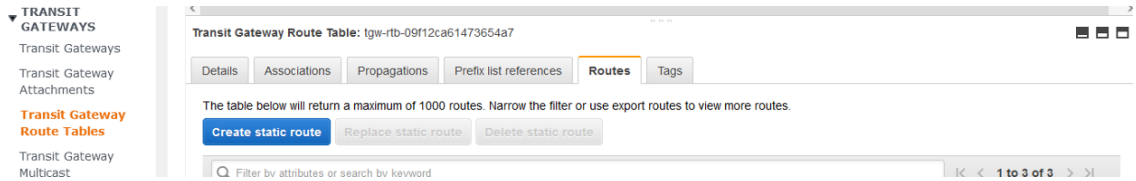
HA プライベート IP ソリューションの AWS Transit Gateway の設定

AWS Transit Gateway は、AWS VPC、リージョン、およびオンプレミスネットワーク全体で、内部ネットワーク内でプライベート VIP サブネットをルーティング可能にする必要があります。VPC は AWS Transit Gateway に接続する必要があります。AWS Transit Gateway ルートテーブル内の VIP サブネットまたは IP プールの静的ルートが作成され、VPC をポイントします。

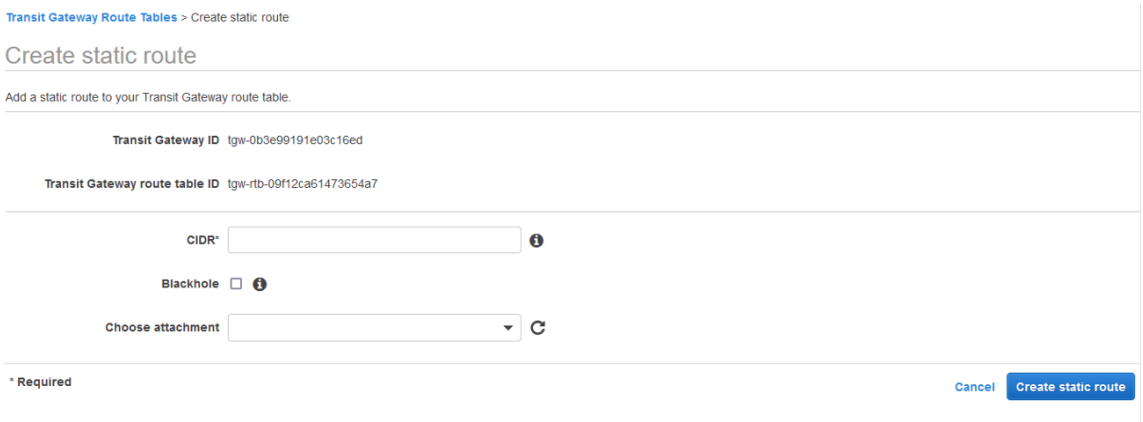


AWS Transit Gateway を設定するには、次の手順に従います。

1. Amazon VPC コンソールを開きます。
2. ナビゲーションペインで、[**Transit Gateway** ートテーブル] を選択します。
3. [ルート] タブを選択し、[静的ルートの作成] をクリックします。



4. CIDR がプライベート VIPS サブネットを指し、アタッチメントが NetScaler VPX のある VPC を指す静的ルートを作成します。



5. [スタティックルートの作成] をクリックし、[閉じる] を選択します。

トラブルシューティング

マルチゾーン HA で HA プライベート IP ソリューションを設定する際に問題が発生した場合は、トラブルシューティングのために次の重要なポイントを確認してください。

- プライマリノードとセカンダリノードの両方に同じ IAM 権限セットがあります。
- INC モードは、プライマリノードとセカンダリノードの両方で有効になっています。
- プライマリノードとセカンダリノードの両方に同じ数のインターフェイスがあります。
- インスタンスを作成する際は、両方のノードで同じ順序でインターフェースをアタッチします。プライマリノードで、クライアントインターフェースが最初に接続され、サーバーインターフェースが次に接続されている場合。次に、セカンダリノードでも同じ順序に従います。不一致がある場合は、正しい順序でインターフェイスを取り外して再接続します。
- トラフィックが流れない場合は、「送信元/宛先」を確認してください。プライマリノードのクライアントインターフェイスで「Check」が初めて無効になります。
- cloudhadaemon コマンド (`ps -aux | grep cloudha`) がシェルで実行されていることを確認します。
- NetScaler ファームウェアのバージョンが 13.0 ビルド 70.x 以降であることを確認してください。
- フェイルオーバープロセスの問題については、/var/log/cloud-ha-daemon.log にあるログファイルを確認してください。

AWS Outpost で NetScaler VPX インスタンスを展開する

December 8, 2023

AWS Outposts は、お客様のサイトにデプロイされている AWS のコンピューティングおよびストレージ容量のプールです。Outposts は、オンプレミスの場所に AWS のインフラストラクチャとサービスを提供します。AWS は

AWS リージョンの一部としてこの容量を運用、監視、管理します。オンプレミスと AWS クラウドで同じ NetScaler VPX インスタンス、AWS API、ツール、およびインフラストラクチャを使用して、一貫したハイブリッドエクスペリエンスを実現できます。

Outposts にサブネットを作成し、EC2 インスタンス、EBS ボリューム、ECS クラスター、RDS インスタンスなどの AWS リソースを作成するときに指定できます。Outposts サブネット内のインスタンスは、プライベート IP アドレスを使用して AWS リージョンの他のインスタンスと通信します。これらはすべて、同じ Amazon 仮想プライベートクラウド（VPC）内にあります。

詳細については、AWS Outposts ユーザーガイドを参照してください。

AWS アウトポストの仕組み

AWS Outposts は、お客様のアウトポストと AWS リージョンの間で常時接続された状態で稼働するように設計されています。リージョンとオンプレミス環境のローカルワークロードにこの接続を実現するには、Outpost をオンプレミスネットワークに接続する必要があります。オンプレミスネットワークは、リージョンとインターネットへの WAN アクセスを提供する必要があります。また、インターネットは、オンプレミスのワークロードやアプリケーションが存在するローカルネットワークへの LAN または WAN アクセスを提供する必要があります。

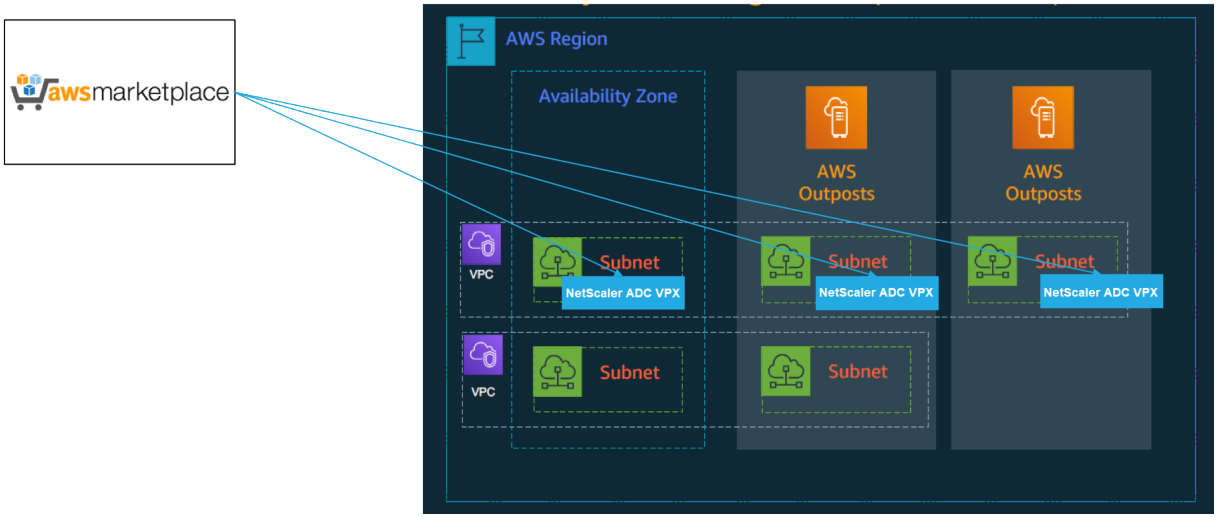
前提条件

- サイトに AWS Outposts をインストールする必要があります。
- AWS Outposts のコンピューティングおよびストレージ容量が使用可能である必要があります。

AWS Outposts の注文方法の詳細については、次の AWS ドキュメントを参照してください。
https://aws.amazon.com/blogs/aws/aws-outposts-now-available-order-your-racks-today/

AWS ウェブコンソールを使用して NetScaler VPX インスタンスを AWS アウトポストにデプロイする

次の図は、アウトポストへの NetScaler VPX インスタンスの簡単な展開を示しています。AWS Marketplace にある NetScaler AMI は、アウトポストにもデプロイされています。



AWS ウェブコンソールにログインし、次の手順を実行して、NetScaler VPX EC2 インスタンスを AWS アウトポストにデプロイします。

1. キーペアを作成します。
2. 仮想プライベートクラウド (VPC) を作成します。
3. サブネットをさらに追加します。
4. セキュリティグループとセキュリティルールを作成します。
5. ルートテーブルを追加します。
6. インターネットゲートウェイを作成します。
7. AWS EC2 サービスを使用して NetScaler VPX インスタンスを作成します。
   AWS ダッシュボードから、[コンピュート**] > [EC2] > [**インスタンスの起動**] > [AWS Marketplace**] に移動します。
8. ネットワークインターフェースをさらに作成して接続します。
9. エラスティック IP を管理用 NIC に接続します。
10. VPX インスタンスに接続します。

各手順の詳細については、「AWS ウェブコンソールを使用して AWS に NetScaler VPX インスタンスをデプロイする」を参照してください。

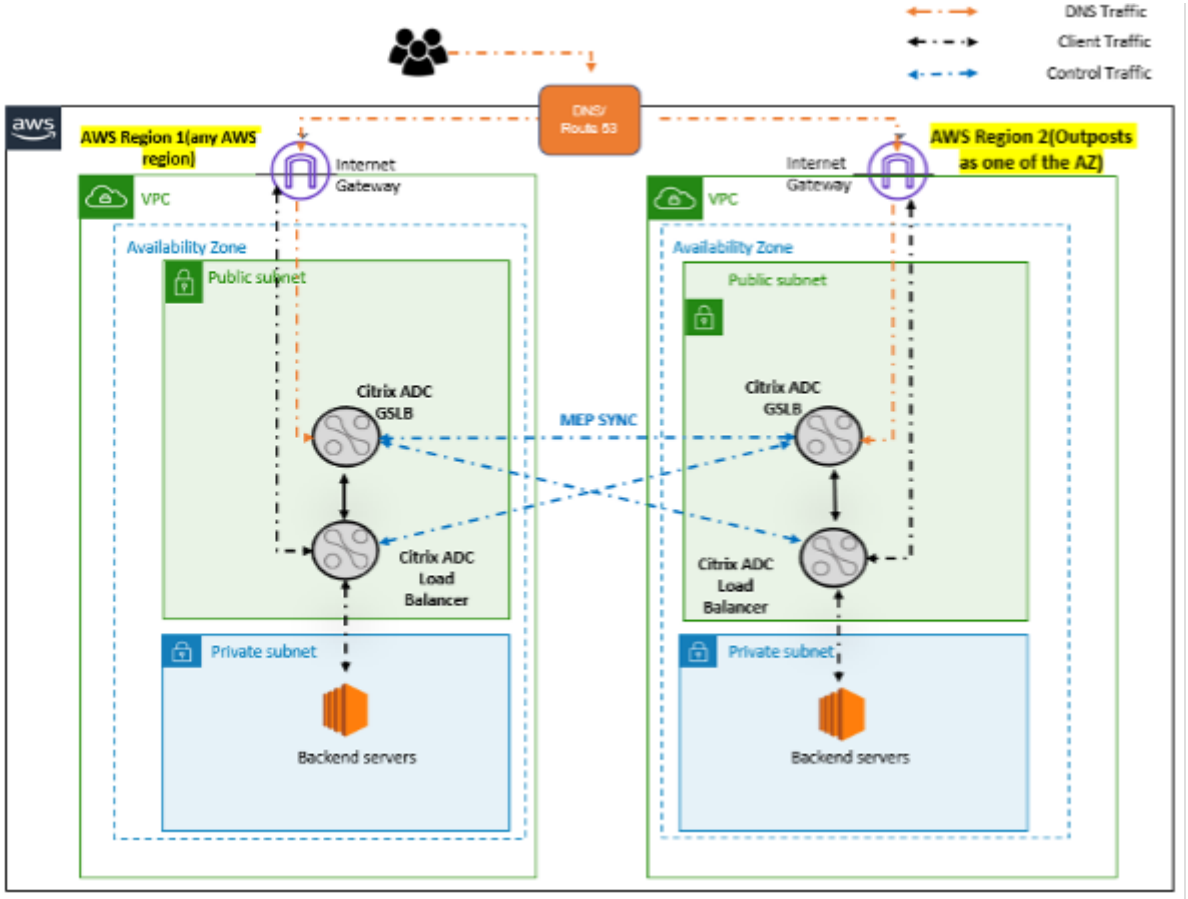
同じアベイラビリティーゾーンのデプロイ内での高可用性については、「AWS に高可用性ペアをデプロイする」を参照してください。

AWS アウトポストを使用してハイブリッドクラウドに NetScaler VPX インスタンスをデプロイする

NetScaler VPX インスタンスは、AWS のアウトポストを含む AWS 環境のハイブリッドクラウドにデプロイできます。NetScaler グローバルサーバー負荷分散（GSLB）ソリューションを使用すると、アプリ配信メカニズムを簡素化できます。GSLB ソリューションは、AWS リージョンと AWS Outposts インフラストラクチャを使用して構築されたハイブリッドクラウド内の複数のデータセンターにアプリケーショントラフィックを分散します。

NetScaler GSLB は、さまざまなユースケースに対応するために、アクティブ-アクティブとアクティブ-パッシブの両方の展開タイプをサポートしています。これらの柔軟な導入オプションとアプリケーション配信メカニズムに加えて、NetScaler は、アプリケーションが AWS Cloud にネイティブにデプロイされているか、AWS Outposts にネイティブにデプロイされているかに関係なく、ネットワークとアプリケーションのポートフォリオ全体を保護します。

次の図は、AWS とのハイブリッドクラウドにおける NetScaler アプライアンスによるアプリケーション配信を示しています。



アクティブ-アクティブ展開では、NetScaler は分散環境全体でトラフィックをグローバルに誘導します。環境内のすべてのサイトは、メトリクス交換プロトコル (MEP) を通じて、リソースの可用性と状態に関するメトリックを交換します。NetScaler アプライアンスは、この情報を使用してサイト間のトラフィックを負荷分散し、GSLB 構成で指定された定義された方法（ラウンドロビン、最小接続、および静的近接性）によって決定された最も適切な GSLB サイトにクライアント要求を送信します。

アクティブ-アクティブ GSLB デプロイメントは次の目的で使用できます。

- すべてのノードがアクティブな状態で、リソース使用率を最適化します。
- リクエストを個々のユーザーに最も近いサイトに誘導することで、ユーザーエクスペリエンスを向上させます。
- ユーザーが定義したペースでアプリケーションをクラウドに移行します。

アクティブ/パッシブ GSLB デプロイメントは次の用途に使用できます。

- 障害回復
- クラウドバースト

参照ドキュメント

- AWS で NetScaler VPX インスタンスを展開する
- AWS ウェブコンソールを使用して NetScaler VPX インスタンスを AWS アウトポストにデプロイする
- NetScaler VPX インスタンスで GSLB を構成する

NetScaler Web App Firewall を使用して **AWS API** ゲートウェイを保護

August 15, 2023

NetScaler アプライアンスを AWS API Gateway の前にデプロイし、API ゲートウェイを外部の脅威から保護できます。NetScaler Web App Firewall（WAF）は、OWASP の上位 10 件の脅威とゼロデイ攻撃から API を保護できます。NetScaler Web App Firewall は、すべての ADC フォームファクターで単一のコードベースを使用します。そのため、あらゆる環境にわたってセキュリティポリシーを一貫して適用し、適用することができます。NetScaler Web App Firewall は導入が簡単で、単一のライセンスとして利用できます。NetScaler Web App Firewall には次の機能があります。

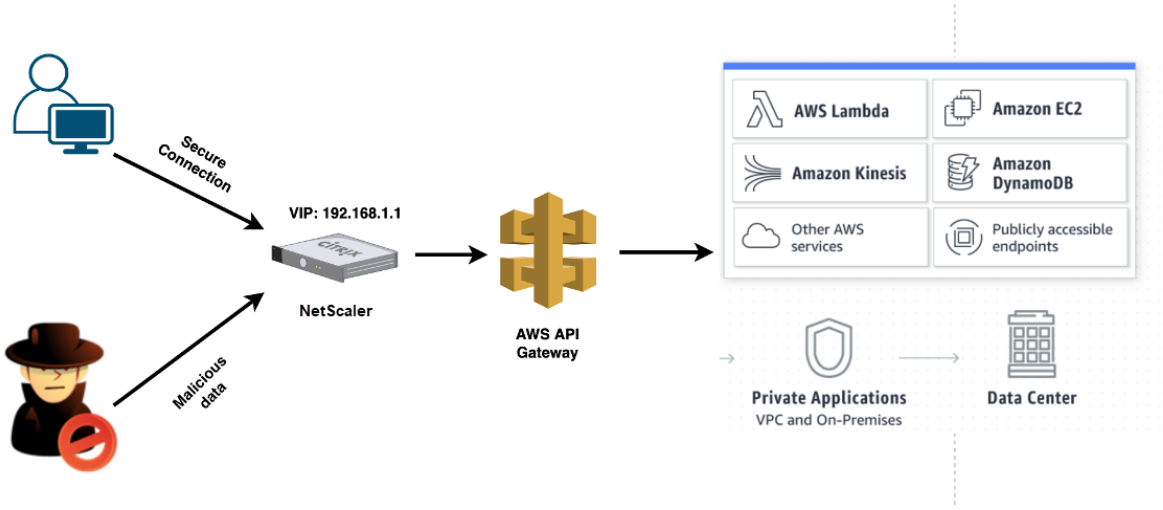
- 構成の簡素化
- ボットの管理
- 総合的な可視性
- 複数のソースからのデータを照合し、統一された画面にデータを表示する

API ゲートウェイ保護に加えて、他の NetScaler ADC 機能も使用できます。詳しくは、NetScaler のドキュメントを参照してください。データセンターのフェイルオーバーを回避し、シャットダウン時間を最小限に抑えるだけでなく、アベイラビリティーゾーン内またはアベイラビリティーゾーン間で ADC を高可用性に設定できます。Autoscale 機能でクラスタリングを使用または構成することもできます。

以前、AWS API Gateway は、その背後にあるアプリケーションを保護するために必要な保護をサポートしていませんでした。Web アプリケーションファイアウォール (WAF) 保護がなければ、API はセキュリティ上の脅威にさらされがちでした。

AWS API ゲートウェイの前に **NetScaler ADC** アプライアンスをデプロイする

次の例では、NetScaler アプライアンスが AWS API ゲートウェイの前にデプロイされています。



AWS Lambda サービスに対する本物の API リクエストがあるとします。このリクエストは、Amazon API Gateway のドキュメントに記載されているどの APIサービスにも適用できます。上の図に示すように、トラフィックフローは次のようになります。

1. クライアントが AWS Lambda 関数 (XYZ) にリクエストを送信します。このクライアント要求は、NetScaler ADC 仮想サーバー（192.168.1.1）に送信されます。
2. 仮想サーバはパケットを検査し、悪意のあるコンテンツがないかチェックします。
3. NetScaler ADC アプライアンスは、書き換えポリシーをトリガーして、クライアント要求のホスト名と URL を変更します。たとえば、`https://restapi.citrix.com/default/LamdaFunctionXYZ`を`https://citrix.execute-api.<region>.amazonaws.com/default/LambdaFunctionXYZ`に変更するとします。
4. NetScaler アプライアンスは、このリクエストを AWS API ゲートウェイに転送します。
5. AWS API Gateway はさらに Lambda サービスにリクエストを送信し、Lambda 関数「XYZ」を呼び出します。
6. 同時に、攻撃者が悪意のあるコンテンツを含む API リクエストを送信すると、その悪意のあるリクエストは NetScaler ADC アプライアンスに到達します。
7. NetScaler ADC アプライアンスはパケットを検査し、構成されたアクションに基づいてパケットをドロップします。

WAF を有効にして NetScaler ADC アプライアンスを構成する

NetScaler ADC アプライアンスで WAF を有効にするには、次の手順を実行します。

1. コンテンツスイッチまたは負荷分散仮想サーバーを追加します。仮想サーバーの IP アドレスが 192.168.1.1 で、ドメイン名（restapi.citrix.com）に解決されるとします。
2. NetScaler 仮想サーバーで WAF ポリシーを有効にします。詳細については、「Web App Firewall の構成」を参照してください。

3. 書き換えポリシーを有効にして、ドメイン名を変更します。たとえば、「restapi.citrix.com」ドメイン名のロードバランサーへの受信リクエストを、「citrix.execute-api」のバックエンド AWS API ゲートウェイに書き換えるように変更するとします。<region>.amazonaws” ドメイン名。
4. NetScaler ADC アプライアンスで L3 モードを有効にして、プロキシとして機能させます。次のコマンドを使用します:

```
enable ns mode L3
<!--NeedCopy-->
```

前の例のステップ 3 で、Web サイト管理者が NetScaler ADC アプライアンスで「restapi.citrix.com」ドメイン名を「citrix.execute-api」に置き換えることを望んでいるとします。<region>.amazonaws.com” と入力し、URL に「デフォルト/ラムダ/XYZ」を付けます。

次の手順では、書き換え機能を使用してクライアント要求のホスト名と URL を変更する方法について説明します。

1. SSH を使用して NetScaler ADC アプライアンスにログオンします。
2. 書き換えアクションを追加する。

```
add rewrite action rewrite_host_hdr_act replace "HTTP.REQ.HEADER("
    Host")" ""citrix.execute-api.<region>.amazonaws.com""

add rewrite action rewrite_url_act replace HTTP.REQ.URL.
    PATH_AND_QUERY ""/default/lambda/XYZ""
<!--NeedCopy-->
```

3. 書き換えアクションの書き換えポリシーを追加します。

```
add rewrite policy rewrite_host_hdr_pol "HTTP.REQ.HEADER("Host").
    CONTAINS("restapi.citrix.com") "rewrite_host_hdr_act

add rewrite policy rewrite_url_pol "HTTP.REQ.HEADER("Host").
    CONTAINS("restapi.citrix.com") "rewrite_url_act
<!--NeedCopy-->
```

4. 書き換えポリシーを仮想サーバにバインドします。

```
bind lb vserver LB_API_Gateway -policyName rewrite_host_hdr_pol -
    priority 10 -gotoPriorityExpression 20 -type REQUEST

bind lb vserver LB_API_Gateway -policyName rewrite_url_pol -
    priority 20 -gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

詳しくは、「NetScaler ADC アプライアンスのクライアント要求でホスト名と URL を変更するように書き換えを構成する」を参照してください。

NetScaler の機能と機能

NetScaler ADC アプライアンスは、展開を保護するだけでなく、ユーザーの要件に基づいて要求を強化することもできます。NetScaler ADC アプライアンスには、次の主要な機能があります。

- **API** ゲートウェイの負荷分散：複数の API ゲートウェイがある場合は、NetScaler ADC アプライアンスを使用して複数の API ゲートウェイを負荷分散し、API リクエストの動作を定義できます。
  - さまざまな負荷分散方式を使用できます。たとえば、Least 接続メソッドは API Gateway 制限のオーバーロードを回避し、Custom load メソッドは特定の API ゲートウェイの特定の負荷を維持するなどです。詳細については、「負荷分散アルゴリズム」を参照してください。
  - SSL オフロードは、トラフィックを中断することなく設定されます。
  - [送信元 IP (USIP) を使用] モードを有効にすると、クライアント IP アドレスが保持されます。
  - ユーザー定義の SSL 設定: 独自の署名証明書とアルゴリズムを使用して、独自の SSL 仮想サーバーを作成できます。
  - バックアップ仮想サーバー:API ゲートウェイにアクセスできない場合は、追加のアクションのためにリクエストをバックアップ仮想サーバーに送信できます。
  - 他にも多くの負荷分散機能を使用できます。詳しくは、「NetScaler ADC アプライアンスのトラフィックの負荷分散」を参照してください。
- 認証、承認、監査**:** LDAP、SAML、RADIUS などの独自の認証方法を定義し、API リクエストの承認と監査を行うことができます。
- レスポンダー**:** シャットダウン時に API リクエストを他の API Gateway にリダイレクトできます。
- レート制限**:** レート制限機能を設定して、API ゲートウェイの過負荷を回避できます。
- 可用性の向上： NetScaler アプライアンスを高可用性セットアップまたはクラスターセットアップで構成して、AWS API トラフィックの可用性を高めることができます。
- **REST API:** REST API をサポートします。REST API は、クラウド本番環境での作業の自動化に使用できます。
- データの監視**:** 参照用にデータを監視し、ログに記録します。

NetScaler アプライアンスにはさらに多くの機能があり、AWS API ゲートウェイと統合できます。詳しくは、NetScaler のドキュメントを参照してください。

バックエンドの **AWS Autoscaling** サービスを追加する

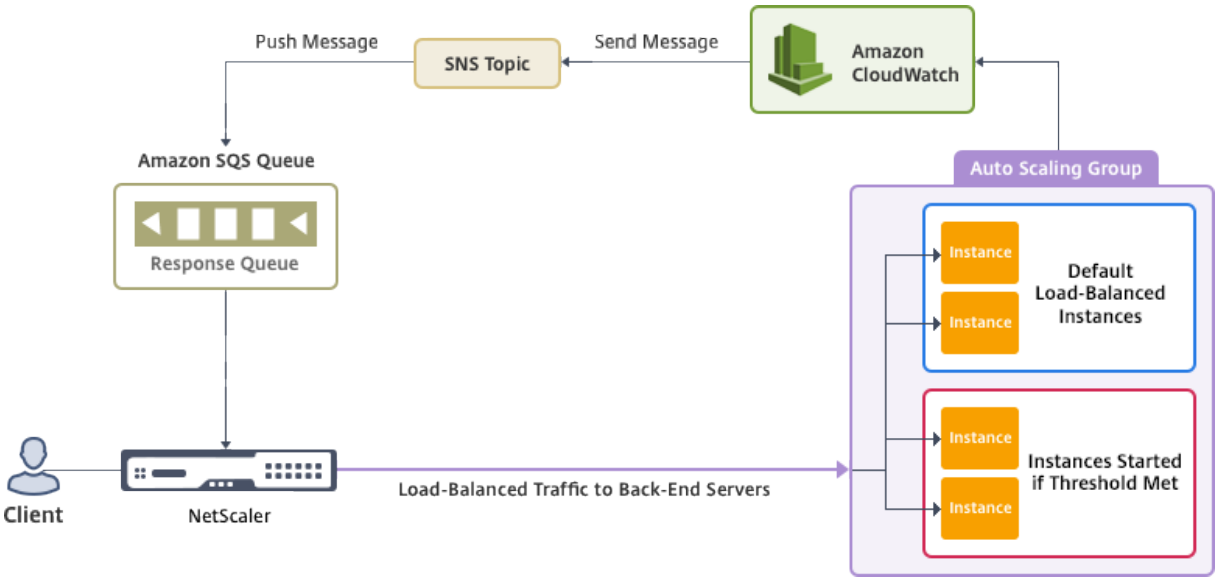
October 25, 2023

クラウドでアプリケーションを効率的にホストすると、アプリケーションの必要に応じて、リソースを簡単にコスト効率よく管理できます。需要の増大に対応するには、ネットワークリソースをスケールアップする必要があります。アイドル状態のリソースに不要なコストを費やさないようにするには、需要が低下しているかどうかに関わらず、スケールダウンが必要です。特定の時間に必要な数のインスタンスのみをデプロイしてアプリケーションの実行コストを最小限に抑えるには、トラフィック、メモリ、CPU の使用率などを常に監視する必要があります。しかし、トラフィックを手動で監視するのは煩雑です。アプリケーション環境を動的にスケールアップまたはスケールダウンするには、トラフィックの監視プロセスを自動化し、必要に応じてリソースを増減する必要があります。

AWS Auto Scaling サービスと統合され、NetScaler VPX インスタンスには次の利点があります。

- 負荷分散と管理: 需要に応じてサーバーをスケールアップおよびスケールダウンするように自動構成します。VPX インスタンスはバックエンドサブネット内の Autoscale e グループを自動検出し、ユーザーが Autoscale e グループを選択して負荷を分散できるようにします。これはすべて、VPX インスタンスの仮想 IP アドレスとサブネット IP アドレスを自動構成することによって行われます。
- 高可用性: 複数のアベイラビリティーゾーンにまたがる Autoscale e グループを検出し、サーバーの負荷を分散します。
- ネットワークの可用性の向上:VPX インスタンスは以下をサポートします。
  - VPC ピアを使用した異なる VPC のバックエンドサーバー
  - 同じ配置グループのバックエンドサーバー
  - 異なるアベイラビリティゾーンのバックエンドサーバー
- 正常な接続終了**:** グレースフルタイムアウト機能を使用して、スケールダウンアクティビティが発生してもクライアント接続が失われないように、Autoscale サーバーを正常に削除します。

図:NetScaler VPX インスタンスによる AWS オートスケーリングサービス



この図は、AWS オートスケーリングサービスが NetScaler ADC VPX インスタンス（負荷分散仮想サーバー）とどのように互換性があるかを示しています。詳しくは、次の AWS のトピックを参照してください。

- オートスケーリンググループ
- CloudWatch
- Simple Notification Service（SNS）
- シンプルキューサービス (Amazon SQS)

はじめに

NetScaler VPX インスタンスで自動スケーリングの使用を開始する前に、次のタスクを完了する必要があります。

1. 次のトピックをお読みください。

   - Prerequisites
   - 制限事項と使用上のガイドライン

2. 要件に応じて、AWS で NetScaler VPX インスタンスを作成します。

   - NetScaler VPX スタンドアロンインスタンスの作成方法の詳細については、「AWS への NetScaler ADC VPX スタンドアロンインスタンスのデプロイ 」および「シナリオ：スタンドアロンインスタンス」を参照してください。
   - VPX インスタンスを HA モードでデプロイする方法の詳細については、「AWS に高可用性ペアをデプロイする」を参照してください。

   > 注:
   >
   > Citrix では、AWS で Citrix ADC VPX インスタンスを作成する場合は CloudFormation テンプレートを使用することをお勧めします。
   >
   > 管理用（NSIP）、クライアント側 LB 仮想サーバー（VIP）用、サブネット IP（NSIP）用の 3 つのインターフェイスを作成することをお勧めします。

3. AWS Autoscale グループを作成します。既存の自動スケーリング設定がない場合は、次のことを行う必要があります。

   a) 起動構成を作成する

   b) 自動スケーリンググループの作成

   c) オートスケーリンググループの検証

   詳しくは、http://docs.aws.amazon.com/autoscaling/latest/userguide/GettingStartedTutorial.htmlを参照してください。

4. AWS Autoscale グループでは、少なくとも 1 つのスケールダウンポリシーを指定する必要があります。NetScaler VPX インスタンスはステップスケーリングポリシーのみをサポートします。シンプルスケーリングポリシーとターゲットトラッキングスケーリングポリシーは、Autoscale e グループではサポートされていません。

AWS 自動スケーリングサービスを NetScaler VPX インスタンスに追加する

GUI を使用して、ワンクリックで VPX インスタンスに自動スケーリングサービスを追加できます。次の手順を実行して、VPX インスタンスに自動スケーリングサービスを追加します。

1. nsrootの認証情報を使用して VPX インスタンスにログオンします。
2. NetScaler VPX インスタンスに初めてログオンすると、デフォルトの Cloud Profile ページが表示されます。ドロップダウンメニューから AWS AutoScaling グループを選択し、[作成] をクリックしてクラウドプロファイルを作成します。後でクラウドプロファイルを作成する場合は、「スキップ」をクリックします。

   クラウドプロファイルの作成時に留意すべきポイント**:** デフォルトでは、CloudFormation テンプレートによって以下の IAM ロールが作成され、アタッチされます。

```
{


    "Version": "2012-10-17",
    "Statement": [

     {


            "Action": [

                "ec2:DescribeInstances",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DetachNetworkInterface",
                "ec2:AttachNetworkInterface",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:RebootInstances",
                "autoscaling:\*",
                "sns:\*",
                "sqs:*"

             “iam: SimulatePrincipalPolicy”
             “iam: GetRole”

            ],

            "Resource": "*",
            "Effect": "Allow"

        }


    ]

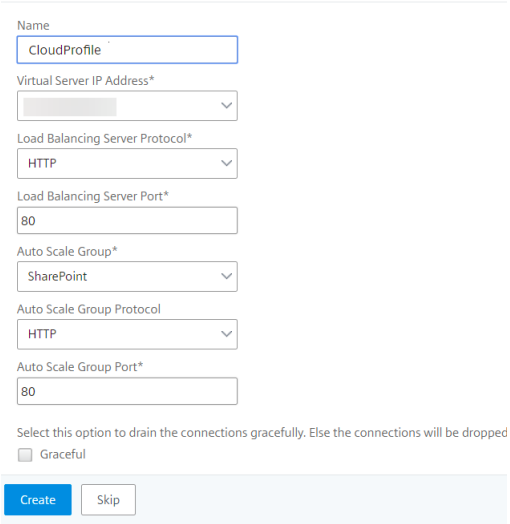
 }

<!--NeedCopy-->
```

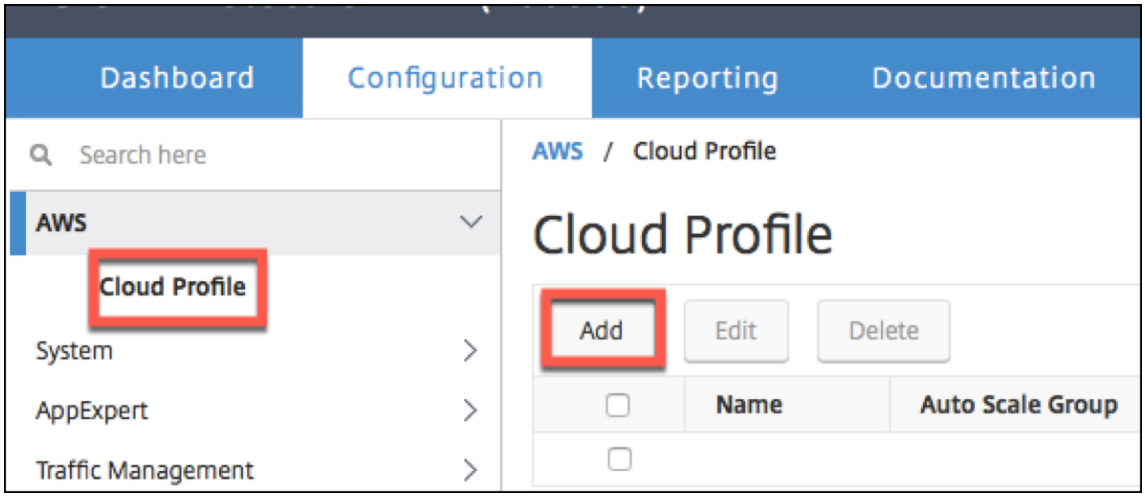

インスタンスの IAM ロールに適切な権限があることを確認します。

- 仮想サーバーの IP アドレスは、VPX インスタンスで使用可能な空き IP アドレスから自動入力されます。https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/MultipleIP.html#ManageMultipleIP
- Autoscale グループは、AWS アカウントに設定されている Autoscale グループから事前入力されます。http://docs.aws.amazon.com/autoscaling/latest/userguide/AutoScalingGroup.html。
- 自動スケーリンググループのプロトコルとポートを選択する際には、サーバーがそれらのプロトコルとポートをリッスンしていることを確認し、サービスグループに適切なモニターをバインドします。デフォルトでは、TCP モニターが使用されます。
- SSL プロトコルタイプ Auto Scaling の場合、クラウドプロファイルを作成すると、証明書がないために負荷分散仮想サーバーまたはサービスグループがダウンします。証明書は、仮想サーバまたはサービスグループに手動でバインドできます。
- Autoscale e サーバーを正常に削除するには、「グレースフルタイムアウト」オプションを選択します。このオプションが選択されていない場合、Autoscale e グループのサーバーは負荷が下がった直後に削除され、接続されている既存のクライアントのサービスが中断される可能性があります。Graceful を選択してタイムアウトを発生させるのはスケールダウンの場合です。VPX インスタンスはサーバーをすぐには削除しませんが、サーバーの 1 つを正常に削除するようにマークします。この間、インスタンスではこのサーバーへの新しい接続は許可されません。既存の接続は、タイムアウトが発生するまで処理され、タイムアウト後に VPX インスタンスがサーバーを削除します。

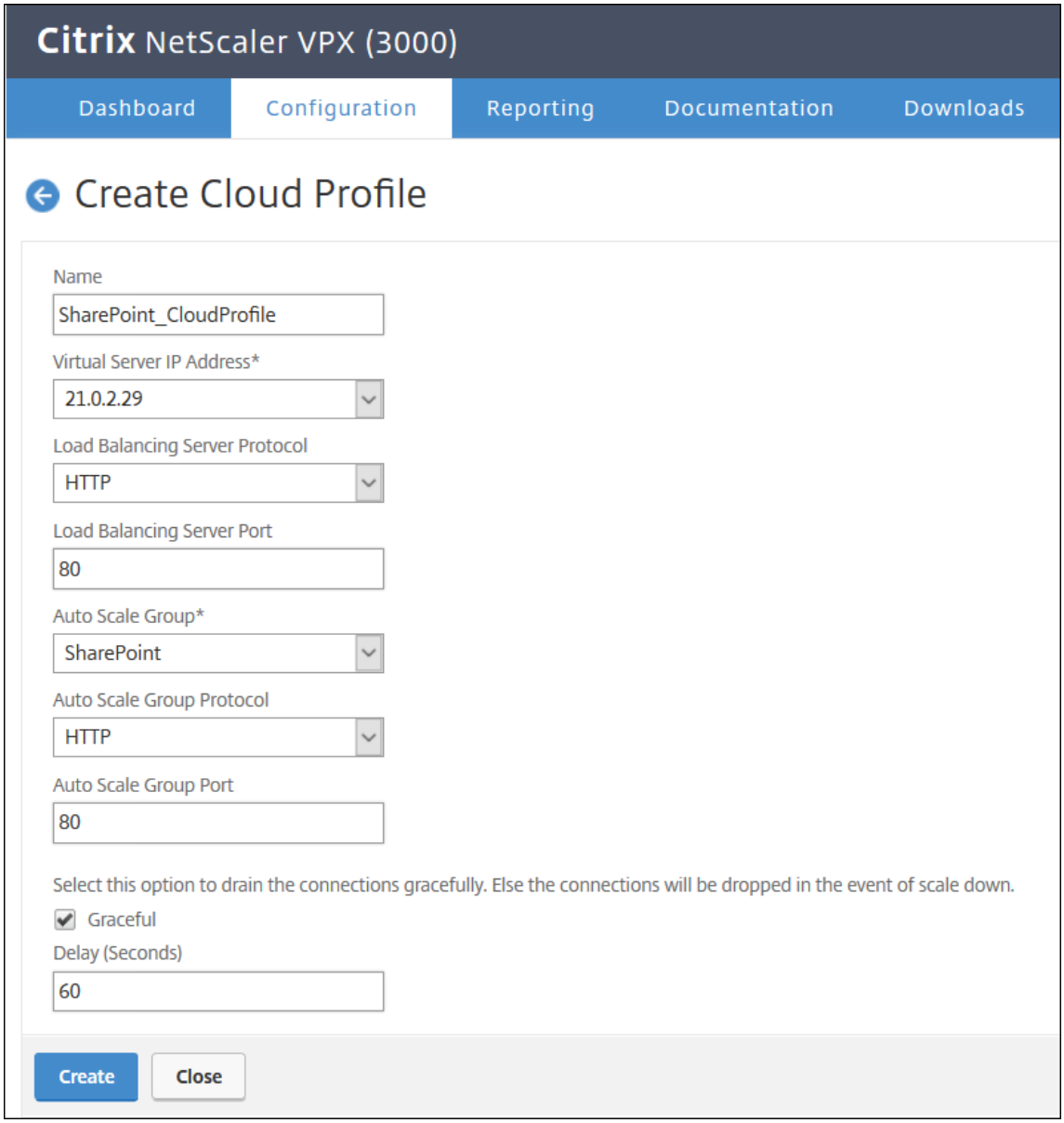
図: デフォルトのクラウドプロファイルページ



3. 初めてログオンした後、クラウドプロファイルを作成する場合は、GUI で [システム] **> [AWS] > [**クラウドプロファイル] に移動し、[追加] をクリックします。

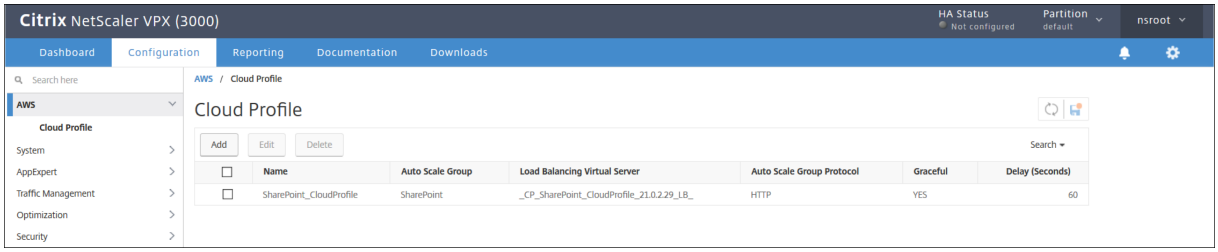


クラウドプロファイルの作成設定ページが表示されます。



Cloud Profile は、NetScaler 負荷分散仮想サーバーと、メンバーを自動スケーリンググループのサーバーとするサービスグループを作成します。バックエンドサーバーは、VPX インスタンスで構成された SNIP を介して到達可能である必要があります。

> 注:
>
> NetScaler リリース 13.1-42.x 以降では、AWS の同じ自動スケーリンググループ（ASG）を使用して、（異なるポートを使用して）サービスごとに異なるクラウドプロファイルを作成できます。したがって、NetScaler VPX インスタンスは、パブリッククラウド内の同じ自動スケーリンググループを持つ複数のサービスをサポートします。

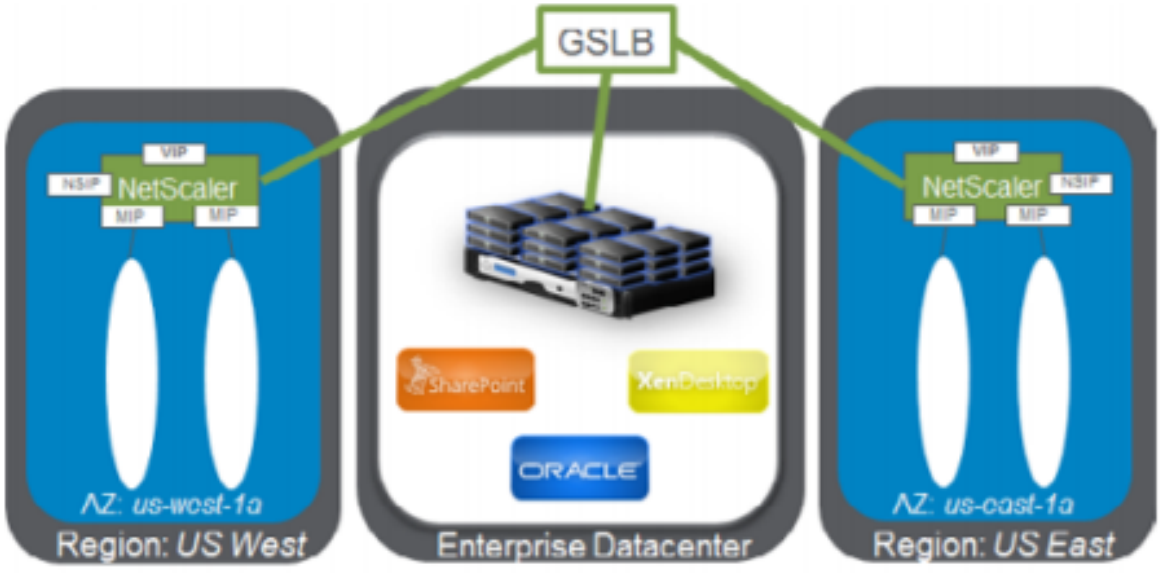


> 注
>
> **AWS** コンソールでオートスケール関連の情報を表示するには、**[**EC2**] > [ダッシュボード] > [**自動スケーリング**]> [Auto Scaling** Group] に移動します。**

NetScaler GSLB を AWS に展開

April 15, 2024

GSLB for NetScaler on AWS を設定するには、基本的に、NetScaler が属する VPC の外部にあるサーバー（別のアベイラビリティリージョンの別の VPC 内やオンプレミスのデータセンターなど）にトラフィックを負荷分散するように NetScaler を構成する必要があります。



データベース管理システムの概要

クラウドロードバランサーに DBS（ドメインベースサービス）を使用する NetScaler GSLB サポートにより、クラウドロードバランサーソリューションを使用して動的クラウドサービスを自動的に検出できます。この構成により、

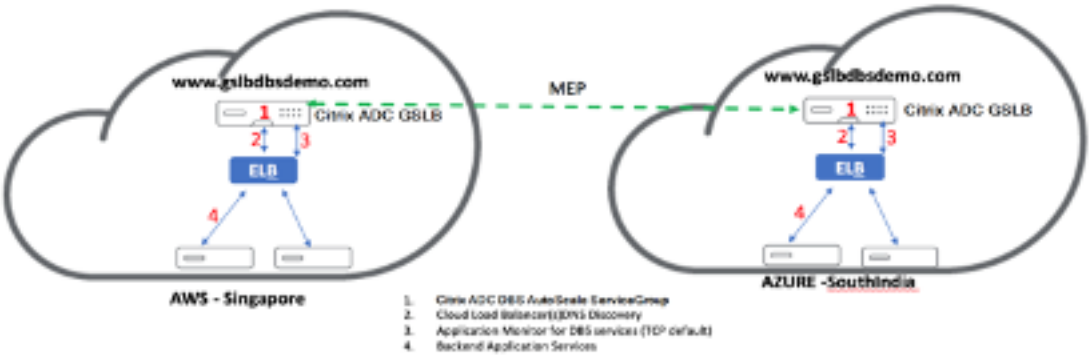
NetScaler はアクティブ-アクティブ環境でグローバルサーバー負荷分散ドメイン名ベースサービス（GSLB DBS）を実装できます。DBS では、DNS 検出から AWS 環境のバックエンドリソースを拡張できます。

このセクションでは、AWS AutoScaling 環境における NetScaler 間の統合について説明します。このドキュメントの最後のセクションでは、AWS リージョンに固有の 2 つの異なる可用性ゾーン（AZ）にまたがる NetScaler ADC の HA ペアを設定する機能について詳しく説明します。

ELB によるドメイン名ベースのサービス (DBS)

GSLB DBS は、ユーザー Elastic Load Balancer (ELB) の FQDN を利用して、AWS 内で作成および削除されるバックエンドサーバーを含むように GSLB サービスグループを動的に更新します。AWS のバックエンドサーバーまたはインスタンスは、ネットワーク需要または CPU 使用率に基づいてスケーリングするように設定できます。この機能を構成するには、NetScaler を ELB にポイントして、AWS 内でインスタンスが作成および削除されるたびに NetScaler を手動で更新しなくても、AWS 内のさまざまなサーバーに動的にルーティングできます。GSLB サービスグループの NetScaler DBS 機能は、DNS 対応サービス検出を使用して、AutoScale グループで識別される DBS 名前空間のメンバーサービスリソースを決定します。

クラウドロードバランサーを搭載した NetScaler GSLB DBS autoScale コンポーネント:



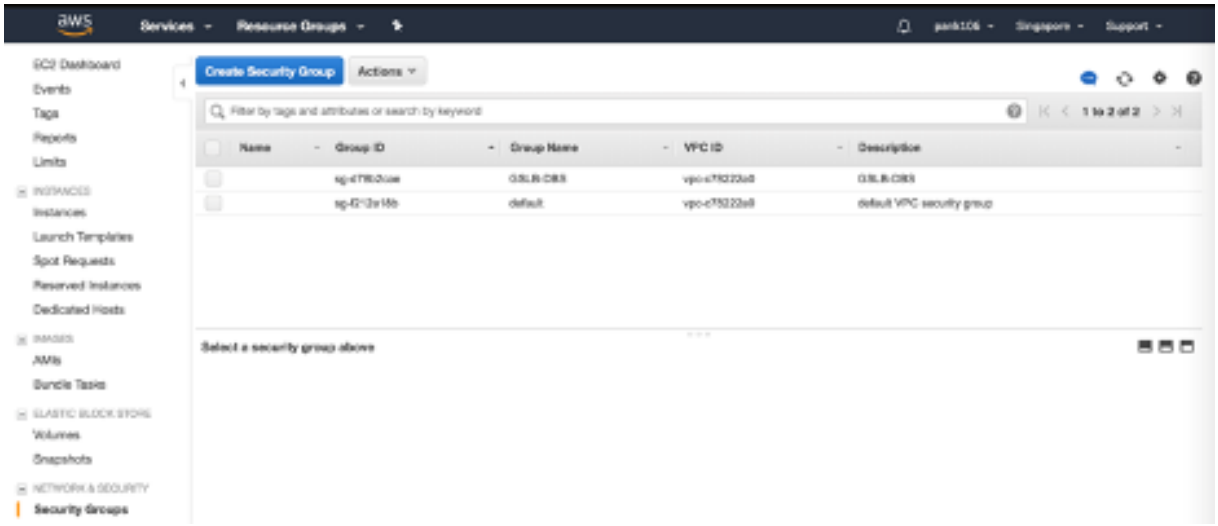
AWS コンポーネントの設定

セキュリティグループ

> 注:
>
> ELB、NetScaler GSLB インスタンス、Linux インスタンスには、それぞれ必要なルールセットが異なるため、異なるセキュリティグループを作成することをお勧めします。この例では、簡潔にするために、統合セキュリティグループ設定があります。

仮想ファイアウォールが適切に設定されていることを確認するには、「VPC のセキュリティグループ」を参照してください。

1. ユーザー **AWS** リソースグループにログインし、[**EC2**] **>** [ネットワークとセキュリティ] **>** [セキュリティグループ] に移動します。



1. [セキュリティグループの作成] をクリックし、名前と説明を入力します。このセキュリティグループには、NetScaler と Linux のバックエンド Web サーバーが含まれます。

1. 次のスクリーンショットから受信ポートルールを追加します。

> 注:
>
> きめ細かなセキュリティ強化には、ソース IP アクセスを制限することが推奨されます。詳細については、「Web サーバールール」を参照してください。

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
SSH	TCP	22	0.0.0.0/0	
DNS (UDP)	UDP	53	0.0.0.0/0	
DNS (UDP)	UDP	53	::/0	
Custom TCP Rule	TCP	3389	0.0.0.0/0	
Custom TCP Rule	TCP	3389	::/0	
All ICMP - IPv4	All	N/A	0.0.0.0/0	
All ICMP - IPv4	All	N/A	::/0	
Custom TCP Rule	TCP	5985	0.0.0.0/0	
Custom TCP Rule	TCP	5985	::/0	
Custom TCP Rule	TCP	3008 - 3011	0.0.0.0/0	
Custom TCP Rule	TCP	3008 - 3011	::/0	

2. Amazon Linux バックエンドウェブサービス

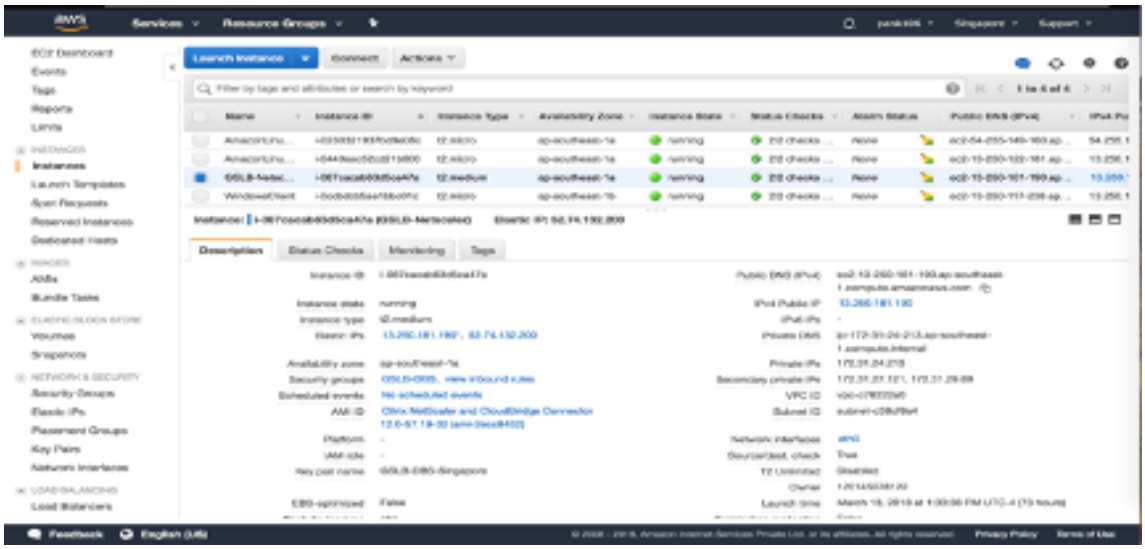
    a) ユーザー **AWS** リソースグループにログインし、[**EC2] > [**インスタンス] に移動します。

    a) 以下の詳細を使用して [インスタンスを起動] をクリック し、**Amazon Linux** インスタンスを設定し ます。

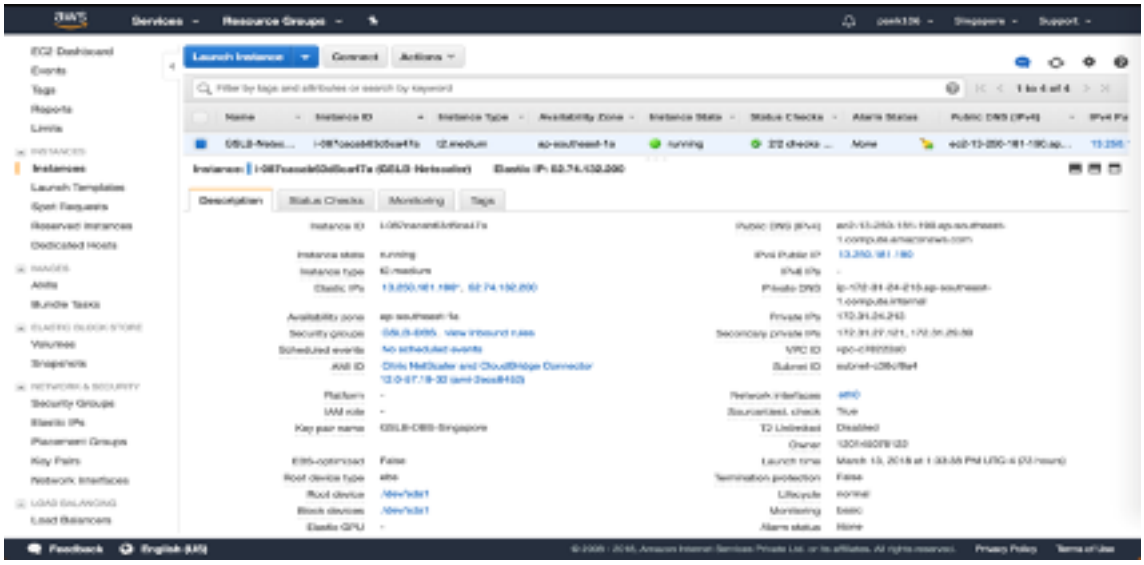
    このインスタンスでのウェブサーバーまたはバックエンドサービスの設定に関する詳細を入力します。

3. NetScaler の構成

    a) ユーザー **AWS** リソースグループにログインし、[**EC2] > [**インスタンス] に移動します。



    a) [**Launch Instance**] をクリックし、次の詳細を使用して **Amazon AMI** インスタンスを設定します。



4. エラスティック IP 設定

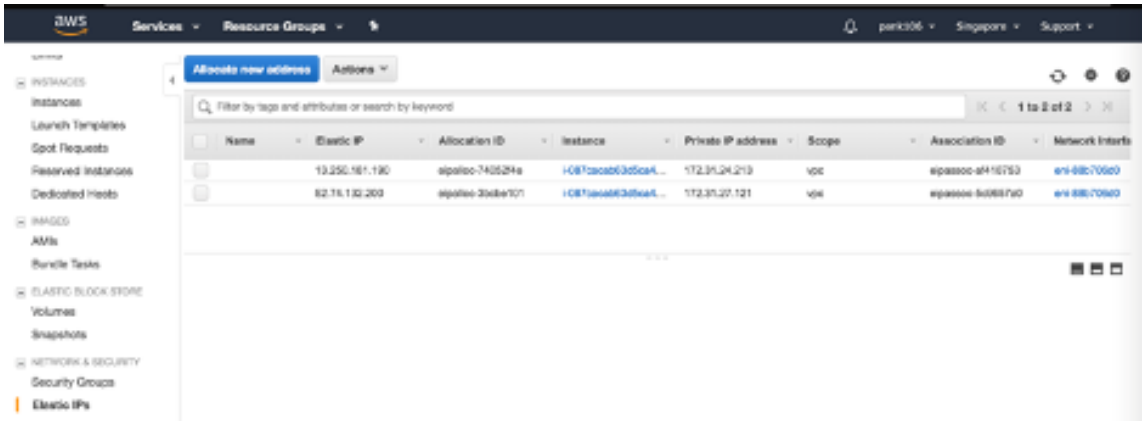
> 注:
>
> NetScaler は、NSIP 用のパブリック IP を持たないことで、コストを削減するために必要に応じて単一の Elastic IP で実行することもできます。代わりに、GSLB サイト IP と ADNS IP に加えて、ボックスへの管理アクセスをカバーできる Elastic IP を SNIP に添付します。

a) ユーザー **AWS** リソースグループにログインし、[**EC2**] **>** [ネットワークとセキュリティ] **>** [**Elastic IP**] に移動します。

b) [新しいアドレスを割り当て] をクリックして Elastic IP アドレスを作成します。

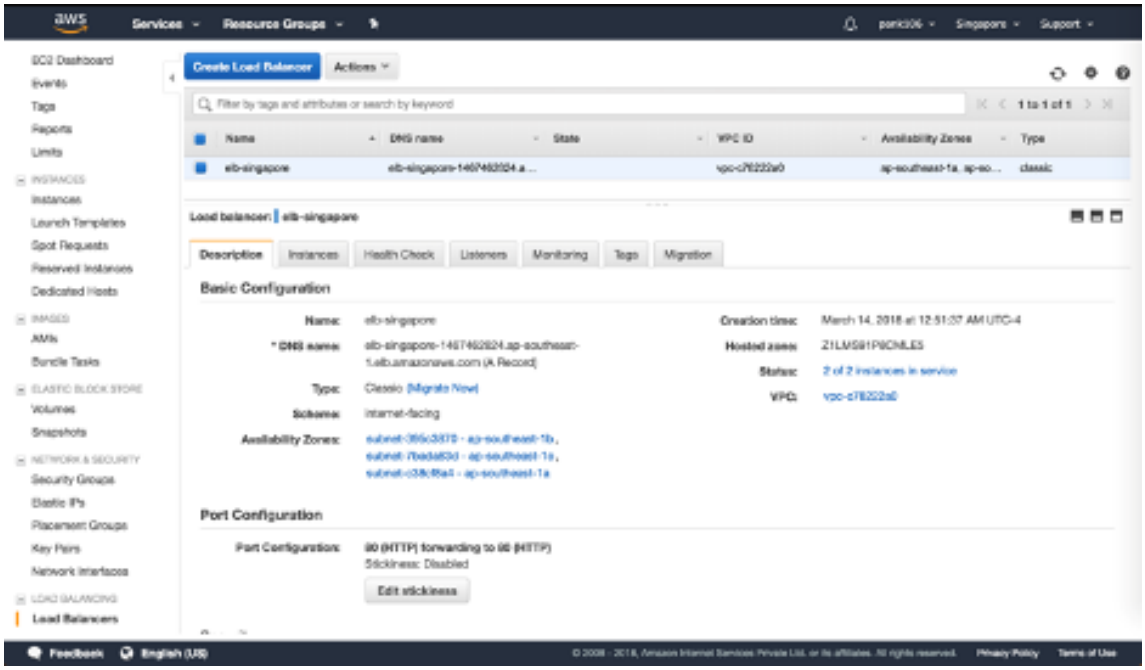
c) AWS 内で NetScaler インスタンスを実行しているユーザーを指すように Elastic IP を設定します。

d) 2 つ目の Elastic IP を構成し、NetScaler インスタンスを実行しているユーザーに再度割り当てます。



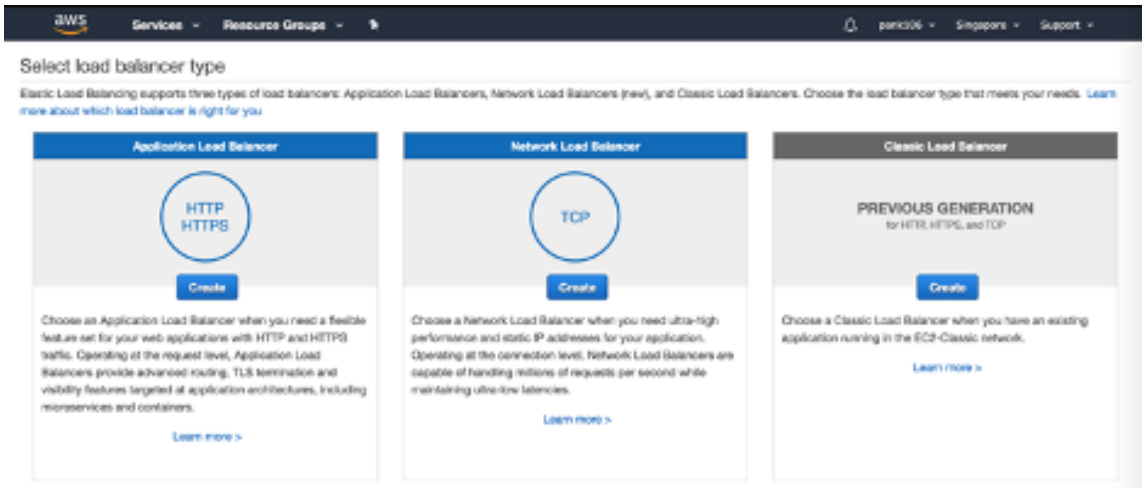
5. 伸縮自在なロードバランサー

a) ユーザー **AWS** リソースグループにログインし、[**EC2**] **>** [負荷分散] **>** [ロードバランサー] に移動します。



a) [**Create Load Balancer**] をクリックして、クラシックロードバランサーを設定します。

ユーザー Elastic Load Balancers を使用すると、ユーザーはバックエンド Amazon Linux インスタンスの負荷を分散できると同時に、需要に基づいてスピンアップされる他のインスタンスの負荷を分散することもできます。



グローバルサーバー負荷分散ドメイン名ベースのサービスの設定

トラフィック管理構成については、「NetScaler GSLB ドメインベースサービスの構成」を参照してください。

デプロイメントの種類

3 NIC 導入

- 一般的な展開
  - GSLB StyleBook
  - ADM と
  - GSLB (ドメイン登録ありの Route53)
  - ライセンス-プール/マーケットプレイス
- 使用例
  - 3 つの NIC 展開は、データと管理トラフィックの実際の分離を実現するために使用されます。
  - 3 つの NIC を配備することで、ADC の規模と性能も向上します。
  - 3 つの NIC 配置は、スループットが通常 1 Gbps 以上で、3 つの NIC 配置が推奨されるネットワークアプリケーションで使用されます。

CFT デプロイメント

お客様は、デプロイをカスタマイズする場合や、デプロイを自動化する場合、CloudFormation テンプレートを使用してデプロイします。

展開手順

デプロイの手順は次のとおりです:

1. GSLB 用の 3 つの NIC デプロイメント
2. ライセンス
3. 展開オプション

GSLB 用の **3** つの **NIC** デプロイメント　NetScaler VPX インスタンスは、AWS Marketplace では Amazon マシンイメージ（AMI）として入手でき、AWS VPC 内のエラスティックコンピューティングクラウド（EC2）インスタンスとして起動できます。NetScaler VPX でサポートされる AMI として許可されている最小 EC2 インスタンスタイプは m4.large です。NetScaler VPX AMI インスタンスには、最低 2 つの仮想 CPU と 2 GB のメモリが必要です。また、AWS VPC 内で起動される EC2 インスタンスは、複数のインターフェイス、インターフェイスごとに複数の IP アドレス、VPX 構成に必要なパブリックおよびプライベート IP アドレスも提供できます。各 VPX インスタンスには、少なくとも 3 つの IP サブネットが必要です。

- 管理サブネット

- クライアント側サブネット（VIP）
- バックエンド向けサブネット (SNIP)

NetScaler では、AWS に標準 VPX インスタンスをインストールする場合、3 つのネットワークインターフェイスを推奨しています。

現在、AWS では、AWS VPC 内で実行しているインスタンスでのみ、マルチ IP 機能を使用できます。VPC 内の VPX インスタンスを使用して、EC2 インスタンスで実行しているサーバーの負荷を分散できます。Amazon VPC を使用すると、ユーザーは、独自の IP アドレス範囲、サブネット、ルートテーブル、ネットワークゲートウェイなどの仮想ネットワーク環境を作成および制御できます。

注:

デフォルトでは、ユーザーは AWS アカウントごとに AWS リージョンごとに最大 5 つの VPC インスタンスを作成できます。ユーザーは、Amazon のリクエストフォーム「Amazon VPC リクエスト」を送信することで、VPC 制限の引き上げをリクエストできます。

ライセンス　AWS 上の NetScaler VPX インスタンスにはライセンスが必要です。AWS で実行されている NetScaler VPX インスタンスでは、次のライセンスオプションを使用できます。

- 無料 (無制限)
- 毎時
- 年次

自分のライセンスを持参する

無料トライアル（すべての NetScaler VPX-AWS サブスクリプションは、AWS Marketplace で 21 日間無料）。

展開オプション　ユーザーは NetScaler VPX スタンドアロンインスタンスを AWS にデプロイできます。
詳しくは、「NetScaler VPX スタンドアロンインスタンスを AWS にデプロイする」を参照してください

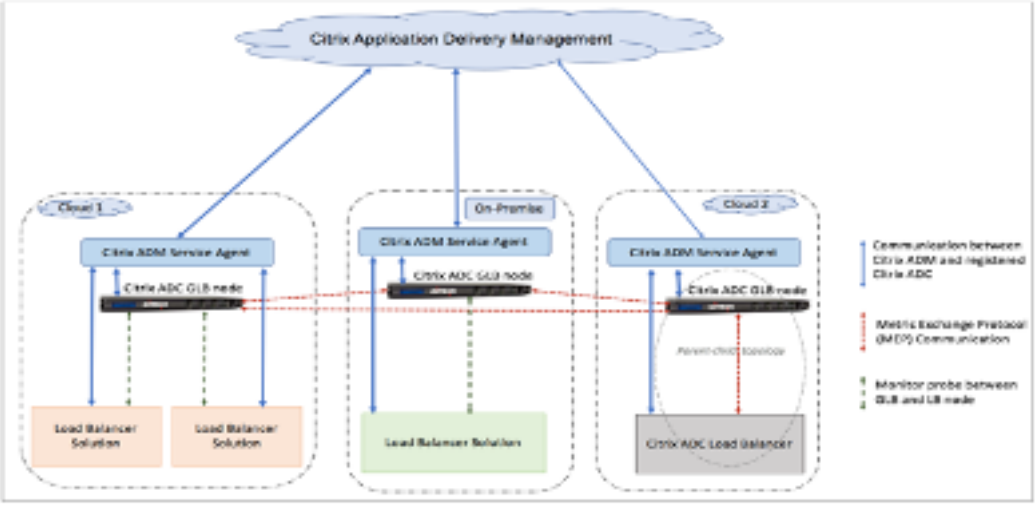
ハイブリッドおよびマルチクラウド展開向けの **NetScaler** グローバル負荷分散

NetScaler ハイブリッドおよびマルチクラウドのグローバルサーバー負荷分散（GSLB）ソリューションにより、ユーザーはハイブリッドクラウド、複数のクラウド、およびオンプレミス展開の複数のデータセンターにアプリケーショントラフィックを分散できます。NetScaler のハイブリッドおよびマルチクラウド GSLB ソリューションにより、ユーザーは既存の設定を変更することなく、ハイブリッドまたはマルチクラウド環境で負荷分散設定を管理できます。また、ユーザーがオンプレミス環境を使用している場合は、クラウドに完全に移行する前に、NetScaler ハイブリッドおよびマルチクラウドの GSLB ソリューションを使用して一部のサービスをクラウドでテストできます。たとえば、ユーザーはトラフィックのごく一部しかクラウドにルーティングできず、トラフィックのほとんどをオンプレミスで処理できます。また、NetScaler のハイブリッドおよびマルチクラウド GSLB ソリューションにより、ユーザーは地理的に離れた場所にある NetScaler インスタンスを単一の統合コンソールから管理および監視できます。

ハイブリッドおよびマルチクラウドアーキテクチャは、「ベンダーロックイン」を回避し、さまざまなインフラストラクチャを使用してユーザーパートナーや顧客のニーズを満たすことで、企業全体のパフォーマンスを向上させることもできます。複数のクラウドアーキテクチャにより、ユーザーは使用した分だけ支払う必要があるため、インフラストラクチャのコストをより適切に管理できます。また、オンデマンドでインフラストラクチャを使用するようになったため、ユーザーはアプリケーションをより適切に拡張できます。また、クラウド間ですばやく切り替えて、各プロバイダーの最高のサービスを活用することもできます。

NetScaler ハイブリッドおよびマルチクラウドの **GSLB** ソリューションのアーキテクチャ

次の図は、NetScaler ハイブリッドおよびマルチクラウド GSLB 機能のアーキテクチャを示しています。



NetScaler GSLB ノードは DNS の名前解決を処理します。これらの GSLB ノードはいずれも、任意のクライアントロケーションから DNS リクエストを受信できます。DNS リクエストを受信する GSLB ノードは、設定された負荷分散方法で選択されたロードバランサー仮想サーバーの IP アドレスを返します。メトリクス（サイト、ネットワーク、およびパーシスタンスメトリック）は、独自の NetScaler プロトコルであるメトリック交換プロトコル（MEP）を使用して GSLB ノード間で交換されます。MEP プロトコルの詳細については、「メトリクス交換プロトコルを構成する」を参照してください。

GSLB ノードに設定されたモニターは、同じデータセンター内の負荷分散仮想サーバーのヘルスステータスを監視します。親子トポロジでは、GSLB ノードと NetScaler ノード間のメトリックは MEP を使用して交換されます。ただし、親子トポロジーでは、GSLB と NetScaler LB ノード間のモニタープローブの構成はオプションです。

NetScaler エージェントは、NetScaler Console とユーザーデータセンターの管理対象インスタンス間の通信を可能にします。NetScaler エージェントとそのインストール方法の詳細については、「はじめに」を参照し てください。

> 注:
>
> このドキュメントでは、次の前提条件を定めています。

- ユーザーが既存の負荷分散設定を持っている場合は、起動して実行中です。
- SNIP アドレスまたは GSLB サイトの IP アドレスは、NetScaler GSLB ノードごとに構成されています。この IP アドレスは、他のデータセンターとメトリックスを交換するときに、データセンターのソース IP アドレスとして使用されます。
- 各 NetScaler GSLB インスタンスには、DNS トラフィックを受信するように ADNS または ADNS-TCP サービスが構成されています。
- 必要なファイアウォールとセキュリティグループは、クラウドサービスプロバイダーで設定されます。

セキュリティグループの設定

ユーザーは、クラウドサービスプロバイダーで必要なファイアウォール/セキュリティグループ構成を設定する必要があります。AWS のセキュリティ機能の詳細については、AWS/Documentation/Amazon VPC/ユーザーガイド/セキュリティを参照してください。

また、GSLB ノードでは、ユーザーは MEP トラフィック交換用の ADNS サービス/DNS サーバーの IP アドレス用にポート 53 を開き、GSLB サイトの IP アドレス用にポート 3009 を開く必要があります。負荷分散ノードでは、ユーザーはアプリケーショントラフィックを受信するために適切なポートを開く必要があります。たとえば、ユーザーは HTTP トラフィックを受信するためにポート 80 を開き、HTTPS トラフィックを受信するためにポート 443 を開く必要があります。ポート 443 を開いて、NetScaler エージェントと NetScaler コンソール間の NITRO 通信を行います。

動的ラウンドトリップタイム GSLB 方式では、ユーザーはポート 53 を開いて、設定されている LDNS プローブタイプに応じて UDP および TCP プローブを許可する必要があります。UDP または TCP プローブは SNIP の 1 つを使用して開始されるため、この設定はサーバー側のサブネットにバインドされたセキュリティグループに対して行う必要があります。

NetScaler ハイブリッドおよびマルチクラウド **GSLB** ソリューションの機能

このセクションでは、NetScaler ハイブリッドおよびマルチクラウド GSLB ソリューションの機能の一部について説明します。

他の負荷分散ソリューションとの互換性

NetScaler のハイブリッドおよびマルチクラウド GSLB ソリューションは、NetScaler ロードバランサー、NGINX、HAProxy、その他のサードパーティ製ロードバランサーなど、さまざまな負荷分散ソリューションをサポートしています。

> 注:
>
> NetScaler 以外の負荷分散ソリューションは、近接ベースおよび非メトリックベースの GSLB メソッドが使用

> され、親子トポロジが構成されていない場合にのみサポートされます。

GSLB の方式

NetScaler のハイブリッドおよびマルチクラウド GSLB ソリューションは、以下の GSLB メソッドをサポートしています。

- メトリックベースの GSLB メソッド。メトリックベースの GSLB メソッドは、メトリック交換プロトコルを介して他の NetScaler ノードからメトリックを収集します。
  - 最小接続: クライアント要求は、アクティブな接続数が最も少ないロードバランサーにルーティングされます。
  - 最小帯域幅: クライアント要求は、現在最も少ない量のトラフィックを処理しているロードバランサーにルーティングされます。
  -
- 非メトリックベースの GSLB メソッド
  - ラウンドロビン: クライアントリクエストは、ロードバランサーのリストの上部にあるロードバランサーの IP アドレスにルーティングされます。その後、そのロードバランサーはリストの一番下に移動します。
  - ソース IP ハッシュ: このメソッドは、クライアント IP アドレスのハッシュ値を使用してロードバランサーを選択します。
- 近接ベースの GSLB メソッド
  - 静的近接: クライアントリクエストは、クライアント IP アドレスに最も近いロードバランサーにルーティングされます。
  - ラウンドトリップ時間 (RTT): この方法では、RTT 値 (クライアントのローカル DNS サーバーとデータセンター間の接続における遅延時間) を使用して、最もパフォーマンスの高いロードバランサーの IP アドレスを選択します。

負荷分散方法の詳細については、「負荷分散アルゴリズム」を参照してください。

GSLB トポロジ

NetScaler のハイブリッドおよびマルチクラウド GSLB ソリューションは、アクティブ/パッシブトポロジーと親子トポロジーをサポートします。

- アクティブ/パッシブトポロジ： 障害点からの保護により、災害復旧を実現し、アプリケーションの継続的な可用性を確保します。プライマリデータセンターがダウンすると、パッシブデータセンターは運用可能になります。GSLB アクティブ/パッシブトポロジの詳細については、「GSLB をディザスタリカバリ用に設定する」を参照してください。

- 親子トポロジ—お客様がメトリックベースの GSLB メソッドを使用して GSLB ノードと LB ノードを構成していて、LB ノードが別の NetScaler インスタンスに展開されている場合に使用できます。親子トポロジでは、LB ノード（子サイト）は NetScaler アプライアンスである必要があります。親サイトと子サイト間のメトリックの交換はメトリック交換プロトコル（MEP）を介して行われます。

親子トポロジの詳細については、「MEP プロトコルを使用した親子トポロジの配置」を参照してください。

IPv6 サポート

NetScaler のハイブリッドおよびマルチクラウド GSLB ソリューションは IPv6 もサポートしています。

監視

NetScaler のハイブリッドおよびマルチクラウド GSLB ソリューションは、安全な接続を有効にするオプションを備えた組み込みモニターをサポートしています。ただし、LB 構成と GSLB 構成が同じ NetScaler インスタンス上にある場合、または親子トポロジーが使用されている場合、モニターの構成は任意です。

永続性

NetScaler のハイブリッドおよびマルチクラウド GSLB ソリューションは以下をサポートします:

- ソース IP ベースの永続性セッション。これにより、設定されたタイムアウトウィンドウ内に到達した場合に、同じクライアントからの複数の要求が同じサービスに送信されます。クライアントが別の要求を送信する前にタイムアウト値が期限切れになると、セッションは破棄され、構成された負荷分散アルゴリズムを使用して、クライアントの次の要求に対して新しいサーバーが選択されます。
- スピルオーバーパーシステンス。プライマリへの負荷がしきい値を下回った後も、バックアップ仮想サーバは受信した要求を処理し続けます。詳細は、「スピルオーバーの設定」を参照してください。
- サイトパーシステンスにより、GSLB ノードがクライアントリクエストを処理するデータセンターを選択し、選択したデータセンターの IP アドレスを以降のすべての DNS リクエストに転送します。構成された永続性がダウンしているサイトに適用される場合、GSLB ノードは GSLB メソッドを使用して新しいサイトを選択し、新しいサイトはクライアントからのその後の要求に対して永続的になります。

NetScaler コンソールスタイルブックを使用した構成

お客様は、NetScaler Console のデフォルトのマルチクラウド GSLB StyleBook を使用して、ハイブリッドおよびマルチクラウドの GSLB 構成で NetScaler インスタンスを構成できます。

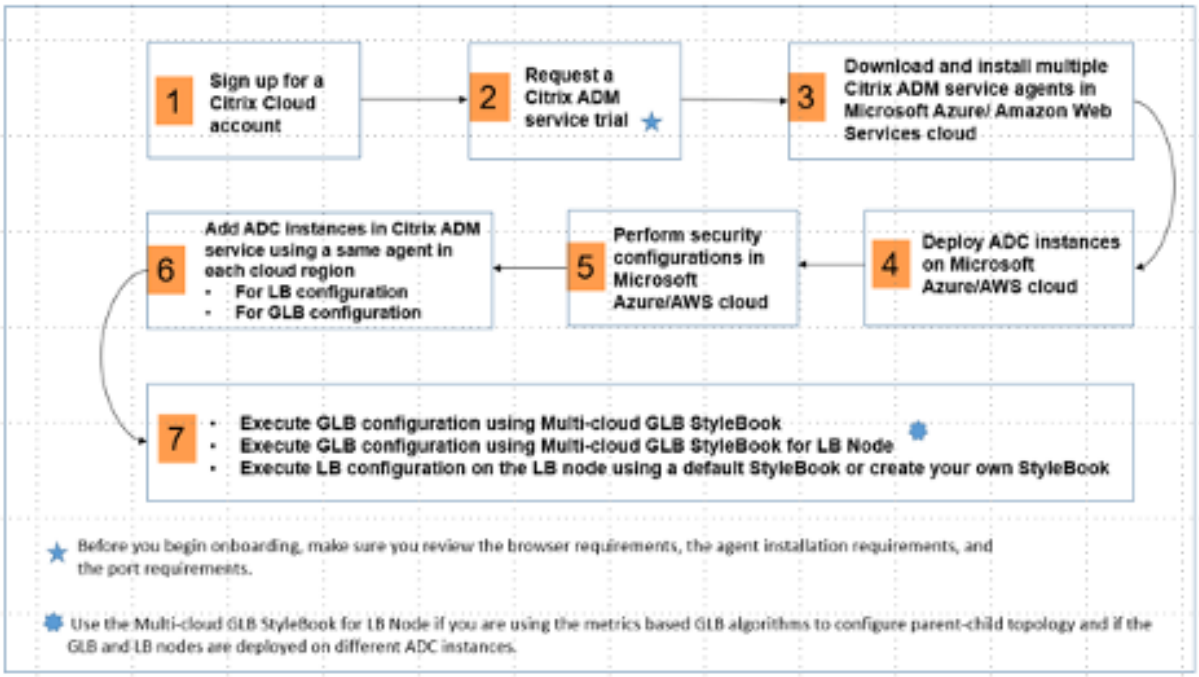
お客様は、LB ノード StyleBook 用のデフォルトのマルチクラウド GSLB StyleBook を使用して、アプリケーショントラフィックを処理する親子トポロジの子サイトである NetScaler 負荷分散ノードを構成できます。この

StyleBook は、ユーザーが親子トポロジーで LB ノードを構成する場合にのみ使用してください。ただし、各 LB ノードは、この StyleBook を使用して個別に設定する必要があります。

NetScaler ハイブリッドおよびマルチクラウド **GSLB** ソリューション構成のワークフロー

お客様は、NetScaler Console で付属のマルチクラウド GSLB StyleBook を使用して、ハイブリッドおよびマルチクラウドの GSLB 構成で NetScaler インスタンスを構成できます。

次の図は、NetScaler ハイブリッドおよびマルチクラウド GSLB ソリューションを構成するためのワークフローを示しています。ワークフロー図の手順については、図の後で詳しく説明します。



クラウド管理者として次のタスクを実行します:

1. NetScaler ラウドアカウントにサインアップしてください。

   NetScaler Console の使用を開始するには、NetScaler Cloud の企業アカウントを作成するか、社内の誰かが作成した既存のアカウントに参加します。

2. ユーザーが NetScaler Cloud にログオンした後、**NetScaler** アプリケーション配信管理 タイルの 「管理**」をクリック して、ADM サービスを初めて設定します。

3. 複数の NetScaler Console サービスエージェントをダウンロードしてインストールします。

   ユーザーは、NetScaler Console サービスエージェントをネットワーク環境にインストールして構成し、NetScaler Console とデータセンターまたはクラウド内の管理対象インスタンス間の通信を有効にする必要があります。各リージョンにエージェントをインストールして、管理対象インスタンスで LB と GSLB の設定を構成できるようにします。LB 構成と GSLB 構成では、1 つのエージェントを共有できます。上記の 3 つのタスクの詳細については、「はじめに」を参照してください。

4. Microsoft AWS クラウド/オンプレミスのデータセンターにロードバランサーをデプロイします。

   ユーザーがクラウドとオンプレミスにデプロイするロードバランサーのタイプに応じて、それに応じてプロビジョニングします。たとえば、ユーザーは Amazon Web Services（AWS）の仮想プライベートクラウドとオンプレミスのデータセンターに NetScaler VPX インスタンスをプロビジョニングできます。仮想マシンを作成して他のリソースを構成することにより、NetScaler インスタンスがスタンドアロンモードで LB または GSLB ノードとして機能するように構成します。NetScaler VPX インスタンスを展開する方法の詳細については、次のドキュメントを参照してください：

   - AWS 上の NetScaler VPX。
   - NetScaler VPX スタンドアロンインスタンスを構成します。

5. セキュリティ設定を実行します。

   ARM と AWS でネットワークセキュリティグループとネットワーク ACL を設定し、ユーザーインスタンスとサブネットのインバウンドトラフィックとアウトバウンドトラフィックを制御します。

6. NetScaler コンソールに NetScaler インスタンスを追加します。

   NetScaler インスタンスは、ユーザーが NetScaler Console から検出、管理、監視したいネットワークアプライアンスまたは仮想アプライアンスです。これらのインスタンスを管理および監視するには、ユーザーはインスタンスをサービスに追加し、LB（ユーザーが NetScaler for LB を使用している場合）と GSLB インスタンスの両方を登録する必要があります。NetScaler コンソールに NetScaler インスタンスを追加する方法について詳しくは、「はじめに」を参照し てください。

7. デフォルトの NetScaler コンソールスタイルブックを使用して GSLB 構成と LB 構成を実装します。

   - マルチクラウド GSLBStyleBook を使用して、選択した GSLB NetScaler インスタンスで GSLB 構成を実行します。
   - 負荷分散設定を実装します。（管理対象インスタンスに LB 設定がすでにある場合、ユーザーはこのステップをスキップできます）。ユーザーは、次の 2 つの方法のいずれかで NetScaler インスタンスにロードバランサーを構成できます。
   - アプリケーションの負荷分散のためにインスタンスを手動で設定します。インスタンスを手動で設定する方法の詳細については、「基本的な負荷分散の設定」を参照してください。
   - StyleBook を使用してください。ユーザーは、NetScaler コンソールスタイルブック（HTTP/SSL 負荷分散スタイルブックまたは HTTP/SSL 負荷分散（モニター付き）スタイルブック）のいずれかを使用して、選択した NetScaler インスタンスにロードバランサー構成を作成できます。ユーザーは独自の StyleBook を作成することもできます。StyleBooks について詳しくは、StyleBooks を参照してください。

8. 以下のいずれかの場合に、LB ノード用のマルチクラウド GSLB StyleBook を使用して GSLB 親子トポロジを設定します：

- ユーザーがメトリックベースの GSLB アルゴリズム（最小パケット、最小接続、最小帯域幅）を使用して GSLB ノードと LB ノードを構成していて、LB ノードが別の NetScaler インスタンスに展開されている場合。
- サイトの永続性が必要な場合。

StyleBook を使用して **NetScaler LB** ノードで **GSLB** を構成する

お客様は、メトリックベースの **GSLB** アルゴリズム（最小パケット、最小接続、最小帯域幅）を使用して **GSLB** ノード と **LB** ノードを構成していて、**LB** ノードが別の **NetScaler** インスタンスに展開されている場合、**LB** ノード用のマルチクラウド **GSLB StyleBook** を使用できます。

ユーザーはこの StyleBook を使用して、既存の親サイトに対してさらに多くの子サイトを構成することもできます。この StyleBook は、一度に 1 つの子サイトを構成します。したがって、この StyleBook から子サイトと同じ数の構成（構成パック）を作成します。StyleBook は子サイトに GSLB 設定を適用します。ユーザーは最大 1024 の子サイトを構成できます。

> 注**:** 親 サイトを設定するには
>
> 、マルチクラウド GSLB StyleBook を使用してください。

この StyleBook では、次の前提条件があります:

- SNIP アドレスまたは GSLB サイトの IP アドレスが設定されています。
- 必要なファイアウォールとセキュリティグループは、クラウドサービスプロバイダーで設定されます。

マルチクラウド **GSLB StyleBook for LB** ノードを使用して親子トポロジの子サイトを構成する

1. アプリケーション **>** 構成 **>** 新規作成に移動します。
2. [アプリケーション] > [構成] に移動し 、[新規作成] をクリック します。

   StyleBook は、この StyleBook で定義されているすべてのパラメータの値を入力できるユーザー・インタフェース・ページとして表示されます。

   > 注**:**
   >
   > このドキュメントでは、データセンターとサイトという用語は同じ意味で使用されています。

3. 次のパラメーターを設定します:
   - アプリケーション名。子サイトを作成する GSLB サイトにデプロイされている GSLB アプリケーションの名前を入力します。
   - プロトコル。ドロップダウンリストボックスから、デプロイされたアプリケーションのアプリケーションプロトコルを選択します。

- **LB** ヘルスチェック (オプション)
- ヘルスチェックの種類。ドロップダウンリストボックスから、サイト上のアプリケーションを表すロードバランサー VIP アドレスの正常性のチェックに使用するプローブのタイプを選択します。
- セキュアモード。(オプション) SSL ベースのヘルスチェックが必要な場合は、[はい] を選択してこのパラメーターを有効にします。
- **HTTP** リクエスト。（オプション）ユーザがヘルスチェックタイプとして HTTP を選択した場合は、VIP アドレスのプローブに使用される完全な HTTP 要求を入力します。
- **HTTP** ステータス応答コードのリスト。（オプション）ユーザがヘルスチェックタイプとして HTTP を選択した場合は、VIP が正常であるときに HTTP 要求への応答で予想される HTTP ステータスコードのリストを入力します。

4. 親サイトを構成しています。
   - 子サイト（LB ノード）を作成する親サイト（GSLB ノード）の詳細を指定します。
     - サイト名。親サイトの名前を入力します。
     - サイト **IP** アドレス。親サイトが他のサイトとメトリックを交換するときにソース IP アドレスとして使用する IP アドレスを入力します。この IP アドレスは、各サイトの GSLB ノードですでに設定されていることを前提としています。
     - サイトのパブリック **IP** アドレス。(オプション) メトリックの交換に使用される親サイトのパブリック IP アドレスを入力します (そのサイトの IP アドレスが NAT の場合)。
5. 子サイトを構成しています。
   - 子サイトの詳細を入力します。
     - サイト名。サイトの名前を入力します。
     - サイト **IP** アドレス。子サイトの IP アドレスを入力します。ここでは、子サイトとして構成されている NetScaler ノードのプライベート IP アドレスまたは SNIP を使用します。
     - サイトのパブリック **IP** アドレス。(オプション) メトリックの交換に使用される子サイトのパブリック IP アドレスを入力します (そのサイトの IP アドレスが NAT の場合)。
6. アクティブな GSLB サービスの設定 (オプション)
   - LB 仮想サーバーの IP アドレスがパブリック IP アドレスでない場合にのみ、アクティブな GSLB サービスを構成します。このセクションでは、ユーザーがアプリケーションがデプロイされているサイトのローカル GSLB サービスのリストを設定できます。
     - サービス **IP**。このサイトの負荷分散仮想サーバーの IP アドレスを入力します。
     - サービスのパブリック **IP** アドレス。仮想 IP アドレスがプライベートで、パブリック IP アドレスが NAT に設定されている場合は、パブリック IP アドレスを指定します。

- サービスポート。このサイトの GSLB サービスのポートを入力します。
- サイト名。GSLB サービスがあるサイトの名前を入力します。

7.「ターゲットインスタンス 」をクリック し、GSLB 構成を展開する各サイトの GSLB インスタンスとして構成された NetScaler インスタンスを選択します。

8.「作成 」をクリック して、選択した NetScaler インスタンス（LB ノード）に LB 構成を作成します。ユーザーは、[ドライラン] をクリックして、ターゲットインスタンスに作成されるオブジェクトを確認することもできます。ユーザーが作成した StyleBook 構成は、構成ページの構成リストに表示されます。ユーザーは、NetScaler コンソール GUI を使用してこの構成を確認、更新、または削除できます。

CloudFormation テンプレートの展開

NetScaler VPX は、AWS Marketplace で Amazon マシンイメージ（AMI）として入手できます。CloudFormation テンプレートを使用して AWS で NetScaler VPX をプロビジョニングする前に、AWS ユーザーは条件に同意し、AWS Marketplace 製品に登録する必要があります。マーケットプレイスで販売されている NetScaler VPX の各エディションでは、この手順が必要です。

CloudFormation リポジトリ内の各テンプレートには、テンプレートの使用法とアーキテクチャを説明するドキュメントが併置されています。テンプレートは、NetScaler VPX 推奨導入アーキテクチャを体系化したり、ユーザーに NetScaler を紹介したり、特定の機能、エディション、またはオプションを実演したりすることを目的としています。ユーザーは、特定の制作およびテストのニーズに合わせてテンプレートを再利用、変更、または拡張できます。ほとんどのテンプレートには、IAM ロールを作成する権限に加えて、完全な EC2 権限が必要です。

CloudFormation テンプレートには、NetScaler VPX の特定のリリース（リリース 12.0-56.20 など）とエディション（たとえば、NetScaler VPX プラチナエディション-10 Mbps）または NetScaler BYOL に固有の AMI ID が含まれています。CloudFormation テンプレートで別のバージョン/エディションの NetScaler VPX を使用するには、ユーザーがテンプレートを編集して AMI ID を置き換える必要があります。

最新の NetScaler AWS-AMI-ID はここにあります：NetScaler AWS CloudFormation マスター。

CFT スリー **NIC** デプロイメント

このテンプレートは、2 つの可用性ゾーンに 3 つのサブネット (管理、クライアント、サーバー) を持つ VPC をデプロイします。パブリックサブネットにデフォルトルートを持つインターネットゲートウェイをデプロイします。また、このテンプレートは、NetScaler の 2 つのインスタンスを持つ可用性ゾーン間で HA ペアを作成します。プライマリの 3 つの VPC サブネット（管理、クライアント、サーバー）に関連付けられた 3 つの ENI と、セカンダリの 3 つの VPC サブネット（管理、クライアント、サーバー）に関連付けられた 3 つの ENI です。この CFT によって作成されるすべてのリソース名には、スタック名の tagName が接頭辞として付けられます。

CloudFormation テンプレートの出力には以下が含まれます。

- primaryCitrixADCManagementURL-プライマリ VPX の管理 GUI への HTTPS URL（自己署名証明書を使用）
- PrimaryCitrixADCManagementUrl2-プライマリ VPX の管理 GUI への HTTP URL
- primaryCitrixADCInstanceId-新しく作成されたプライマリ VPX インスタンスのインスタンス ID
- primaryCitrixADCPublicVIP-VIP に関連付けられているプライマリ VPX インスタンスの Elastic IP アドレス
- PrimaryCitrixADCPrivateNSIP-プライマリ VPX の管理に使用されるプライベート IP (NS IP)
- PrimaryCitrixADCPublicNSIP-プライマリ VPX の管理に使用されるパブリック IP (NS IP)
- PrimaryCitrixADCPrivateVIP-VIP に関連付けられているプライマリ VPX インスタンスのプライベート IP アドレス
- PrimaryCitrixADCSnip-SNIP に関連付けられているプライマリ VPX インスタンスのプライベート IP アドレス
- SecondaryCitrixADCManagementURL-セカンダリ VPX の管理 GUI への HTTPS URL（自己署名証明書を使用）
- セカンダリ CitrixADCManagementUrl2-セカンダリ VPX の管理 GUI への HTTP URL
- secondaryCitrixADCInstanceId-新しく作成されたセカンダリ VPX インスタンスのインスタンス ID
- SecondaryCitrixADCPrivateNSIP-セカンダリ VPX の管理に使用されるプライベート IP (NS IP)
- SecondaryCitrixADCPublicNSIP-セカンダリ VPX の管理に使用されるパブリック IP (NS IP)
- secondaryCitrixADCPrivateVIP-VIP に関連付けられているセカンダリ VPX インスタンスのプライベート IP アドレス
- SecondaryCitrixADCSnip-SNIP に関連付けられているセカンダリ VPX インスタンスのプライベート IP アドレス
- SecurityGroup-VPX が属するセキュリティグループ ID

CFT に入力を提供する場合、CFT のあらゆる パラメーターに対して*は、それが必須フィールドであることを意味します。たとえば、`VPC ID*` は必須フィールドです。

次の前提条件が満たされている必要があります。CloudFormation テンプレートには、通常の EC2 の完全な権限を超えて、IAM ロールを作成するための十分な権限が必要です。また、このテンプレートのユーザーは、この CloudFormation テンプレートを使用する前に、条件に同意して AWS Marketplace 製品に登録する必要があります。

以下も存在しているはずです。

- キー ペア
- 3 つの未割り当て EIP

- 一次管理
- クライアント VIP
- 二次管理

AWS での NetScaler VPX インスタンスのプロビジョニングの詳細については、AWS での NetScaler VPX インスタンスのプロビジョニングを参照してください。

StyleBook を使用して GSLB を設定する方法については、「StyleBook を使用して GSLB を設定する」を参照してください。

前提条件

AWS で VPX インスタンスを作成する前に、ユーザーは以下があることを確認する必要があります:

- Amazon Web Services（AWS）仮想プライベートクラウド（VPC）で NetScaler VPX AMI を起動するための AWS アカウント。ユーザーは Amazonで無料で AWS アカウントを作成できます。
- ユーザーの AWS サービスとリソースへのアクセスを安全に制御するための AWS ID およびアクセス管理 (IAM) ユーザーアカウント。IAM ユーザーアカウントの作成方法の詳細については、トピック「IAM ユーザーの作成 (コンソール)」を参照してください。

IAM ロールは、スタンドアロンデプロイと高可用性デプロイの両方で必須です。IAM ロールには次の権限が必要です:

- ec2:DescribeInstances
- ec2:DescribeNetworkInterfaces
- ec2:DetachNetworkInterface
- ec2:AttachNetworkInterface
- ec2:StartInstances
- ec2:StopInstances
- ec2:RebootInstances
- ec2:DescribeAddresses
- ec2:AssociateAddress
- ec2:DisassociateAddress
- autoscaling:*
- sns:*
- sqs:*
- iam:SimulatePrincipalPolicy
- iam:GetRole

NetScaler CloudFormation テンプレートを使用すると、IAM ロールが自動的に作成されます。このテンプレートでは、作成済みの IAM ロールを選択することはできません。

> 注:
>
> ユーザーが GUI を使用して VPX インスタンスにログオンすると、IAM ロールに必要な権限を構成するように求めるプロンプトが表示されます。権限がすでに設定されている場合は、プロンプトを無視します。

- AWS CLI は、ターミナルプログラムから AWS マネジメントコンソールが提供するすべての機能を使用するために必要です。詳細については、「AWS コマンドラインインターフェイスとは?」を参照してください。。また、ネットワークインターフェイスの種類を SR-IOV に変更するには、AWS CLI が必要です。

GSLB の前提条件

NetScaler GSLB サービスグループの前提条件には、セキュリティグループ、Linux ウェブサーバー、AWS 内の NetScaler ADC、エラスティック IP、およびエラスティックロードバランサーに関する知識と能力を備えた、正常に機能している AWS 環境が含まれます。

GSLB DBS サービスの統合には、AWS ELB ロードバランサーインスタンス用の NetScaler バージョン 12.0.57 が必要です。

現在サポートされている VPX モデルと AWS リージョン、インスタンスタイプ、サービスに関する最新情報については、VPX-AWS サポートマトリックスを参照してください。

そのほかの参照先

ハイブリッドおよびマルチクラウド展開用の NetScaler コンソール GSLB。

AWS への NetScaler Web App Firewall デプロイ

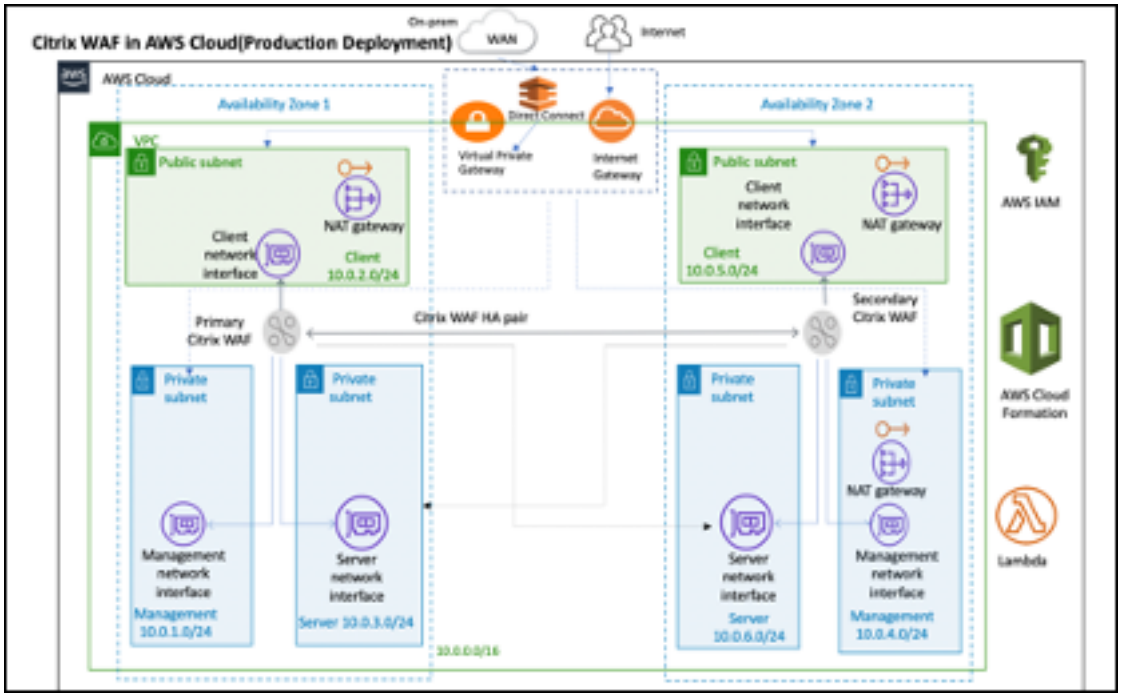
April 15, 2024

NetScaler Web App Firewall は、レイヤー 3 ネットワークデバイスとして、または顧客サーバーと顧客ユーザー間のレイヤー 2 ネットワークブリッジとして、通常は顧客企業のルーターまたはファイアウォールの背後に設置できます。NetScaler Web App Firewall は、Web サーバーとハブ間のトラフィックを傍受できる場所にインストールするか、ユーザーがそれらの Web サーバーにアクセスする際に経由するスイッチを使用する必要があります。次に、ユーザーは、要求を Web サーバーに直接送信するのではなく Web アプリケーションファイアウォールに送信し、ユーザーに直接応答するのではなく Web アプリケーションファイアウォールに応答するようにネットワークを構成します。Web アプリケーションファイアウォールは、内部ルールセットとユーザーの追加と変更の両方を使用して、トラフィックを最終的な宛先に転送する前にフィルタリングします。有害であると検出したアクティビティをブロックまたはレンダリングし、残りのトラフィックを Web サーバに転送します。上の図は、フィルタリングプロセスの概要を示しています。

詳しくは、「NetScaler Web App Firewall 仕組み」を参照してください。

本番環境への導入のための **AWS** 上の **NetScaler Web App Firewall** アーキテクチャ

この画像は、AWS クラウドに NetScaler Web App **Firewall** 環境を構築するデフォルトパラメータ付きの仮想プライベートクラウド（VPC）を示しています。



実稼働環境では、NetScaler Web App Firewall 環境用に次のパラメーターが設定されます:

- このアーキテクチャは、AWS CloudFormation テンプレートと AWS クイックスタートガイドの使用を前提としています。このガイドは、GitHub/AWS-QuickStart/QuickStart-Citrix-ADC-VPXにあります。
- 2 つの可用性ゾーンにまたがる VPC。AWS のベストプラクティスに従って、2 つのパブリックサブネットと 4 つのプライベートサブネットで構成され、/16 クラスレスドメイン間ルーティング (CIDR) ブロック (65,536 個のプライベート IP アドレスを持つネットワーク) を持つ AWS 上の独自の仮想ネットワークを提供します。。*
- NetScaler Web App Firewall の 2 つのインスタンス (プライマリとセカンダリ)。各可用性ゾーンに 1 つずつ。
- ネットワークインターフェイス (管理、クライアント、サーバー) ごとに 1 つずつ、関連付けられたインスタンスのトラフィックを制御する仮想ファイアウォールとして機能する **3** つのセキュリティグループ。
- インスタンスごとに **3** つのサブネット。1 つは管理用、1 つはクライアント用、もう 1 つはバックエンドサーバー用です。
- VPC にアタッチされたインターネットゲートウェイ、およびインターネットへのアクセスを許可するためにパブリックサブネットに関連付けられた Public Subnets ルートテーブル。このゲートウェイは、Web App Firewall ホストがトラフィックを送受信するために使用されます。インターネットゲートウェイの詳細については、「インターネットゲートウェイ」を参照してください。*

- **5** つのルートテーブル-プライマリとセカンダリ Web App Firewall の両方のクライアントサブネットに関連付けられた 1 つのパブリックルートテーブル。残りの 4 つのルートテーブルは、4 つのプライベートサブネット (プライマリおよびセカンダリ Web App Firewall の管理サブネットとサーバー側サブネット) のそれぞれにリンクしています。。*
- Web App Firewall の AWS Lambda は次のことを処理します:
  - HA モードの各可用性ゾーンに 2 つの Web App Firewall を設定する
  - サンプルの Web アプリケーションファイアウォールプロファイルを作成し、この構成を Web App Firewall に関してプッシュします
- AWS Identity and Access Management (IAM) は、ユーザーの AWS サービスとリソースへのアクセスを安全に制御します。デフォルトでは、CloudFormation テンプレート (CFT) によって必要な IAM ロールが作成されます。ただし、ユーザーは NetScaler ADC インスタンスに独自の IAM ロールを提供できます。
- パブリックサブネットでは、パブリックサブネット内のリソースへのアウトバウンドインターネットアクセスを許可する 2 つのマネージドネットワークアドレス変換 (NAT) ゲートウェイ。

> 注:
>
> NetScaler Web App Firewall を既存の VPC に展開する CFT Web App Firewall テンプレートは、アスタリスクでマークされたコンポーネントをスキップし、ユーザーに既存の VPC 構成の入力を求めます。
>
> バックエンドサーバーは CFT によって展開されません。

コストとライセンス

ユーザーは、AWS デプロイの実行中に使用される AWS サービスの費用を負担します。このデプロイに使用できる AWS CloudFormation テンプレートには、ユーザーが必要に応じてカスタマイズできる設定パラメータが含まれています。インスタンスタイプなど、これらの設定の一部は、デプロイのコストに影響します。コストの見積もりについては、ユーザーが使用している各 AWS サービスの料金表ページを参照してください。価格は変更される場合があります。

AWS 上の NetScaler Web App Firewall にはライセンスが必要です。NetScaler Web App Firewall のライセンスを取得するには、ユーザーはライセンスキーを S3 バケットに配置し、展開を起動するときにその場所を指定する必要があります。

> 注:
>
> ユーザーが自分のライセンス使用 (BYOL) ライセンスモデルを選択するときは、AppFlow 機能が有効になっていることを確認する必要があります。BYOL ライセンスの詳細については、「AWS Marketplace/CitrixVPX-カスタマーライセンス」を参照してください。

AWS で実行されている Citrix ADC Web App Firewall では、次のライセンスオプションを使用できます。ユーザーは、スループットなどの 1 つの要素に基づいて AMI (Amazon マシンイメージ) を選択できます。

- ライセンスモデル：従量制（本番ライセンスの場合は PAYG）または自分のライセンスの使用（BYOL、カスタマーライセンス AMI-NetScaler ADC プール容量）。NetScaler ADC プールキャパシティの詳細については、「NetScaler ADC プールキャパシティ」を参照してください。
  - BYOL には、次の 3 つのライセンスモードがあります：
    * NetScaler プールキャパシティの構成：Citrix ADC プールキャパシティの構成
    * NetScaler VPX チェックインおよびチェックアウトライセンス (CICO): Citrix ADCVPX チェックインおよびチェックアウトライセンス

      > ヒント：
      >
      > ユーザーが VPX-200、VPX-1000、VPX-3000、VPX-5000、または VPX-8000 のアプリケーションプラットフォームタイプで CICO ライセンスを選択した場合は、NetScaler Console ライセンスサーバーに同じスループットライセンスがあることを確認する必要があります。

    * NetScaler 仮想 CPU ライセンス: NetScaler 仮想 CPU ライセンス

> 注：
>
> ユーザーが VPX インスタンスの帯域幅を動的に変更したい場合は、BYOL オプションを選択する必要があります。たとえば、**NetScaler Console** からライセンスを割り当てることができる **NetScaler** プールキャパシティを選択するか、再起動せずにオンデマンドでインスタンスの最小容量と最大容量に従って **NetScaler** からライセンスをチェックアウトできます。再起動は、ユーザーがライセンスエディションを変更する場合のみ必要です。

- スループット:200 Mbps または 1 Gbps
- バンドル: プレミアム

展開オプション

この導入ガイドには、次の 2 つの展開オプションがあります：

- 最初のオプションは、クイックスタートガイド形式と次のオプションを使用して展開することです：
  - **NetScaler Web App Firewall** を新しい **VPC** に展開します（エンドツーエンド展開）。このオプションは、VPC、サブネット、セキュリティグループ、およびその他のインフラストラクチャコンポーネントで構成される新しい AWS 環境を構築し、その新しい VPC に NetScaler Web App Firewall をデプロイします。
  - 既存の **VPC** に **NetScaler Web App Firewall** を展開します。このオプションは、ユーザーの既存の AWS インフラストラクチャに NetScaler Web App Firewall をプロビジョニングします。
- 2 つ目のオプションは、NetScaler Console を使用して Web App Firewall StyleBook を使用して展開することです

AWS クイックスタート

ステップ **1:** ユーザー **AWS** アカウントにサインインする

- AWS のユーザーアカウントにサインインする:Amazon アカウントの作成 (必要な場合) または Amazon アカウントにサインインするために必要なアクセス権限を持つ IAM (アイデンティティおよびアクセス管理) ユーザーロールを持つ AWS 。
- ナビゲーションバーのリージョンセレクターを使用して、ユーザーが AWS 可用性ゾーン全体に高可用性をデプロイする AWS リージョンを選択します。
- ユーザー AWS アカウントが正しく設定されていることを確認してください。詳細については、このドキュメントの「技術要件」セクションを参照してください。

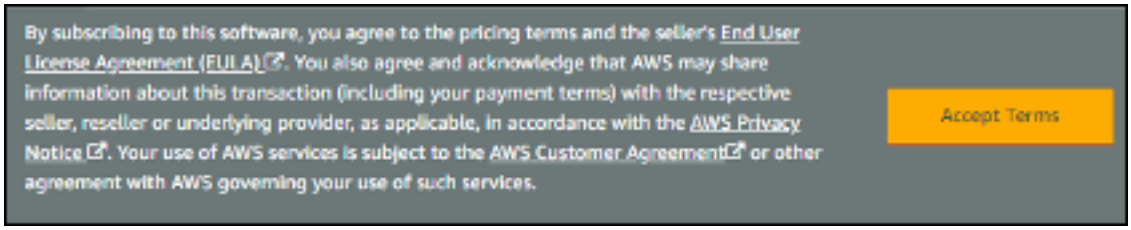
ステップ **2: NetScaler Web App Firewall AMI** にサブスクライブする

- このデプロイには、AWS Marketplace にある NetScaler Web App Firewall 用の AMI へのサブスクリプションが必要です。
- ユーザー AWS アカウントにサインインします。
- 次の表のいずれかのリンクを選択して、NetScaler Web App Firewall オファリングのページを開きます。
  - ユーザーは、以下のステップ 3 でクイックスタートガイドを起動して NetScaler Web App Firewall を展開するときに、**NetScaler Web App Firewall** イメージパラメーターを使用して、AMI サブスクリプションに一致するバンドルとスループットオプションを選択します。次のリストは、AMI オプションと対応するパラメーター設定を示しています。この VPX AMI インスタンスには 2 つ以上の仮想 CPU と 2GB 以上のメモリが必要です。

> 注:
>
> AMI ID を取得するには、GitHub の「AWS Marketplace 上の NetScaler 製品: AWS Marketplace 上の Citrix 製品」ページを参照してください。

- AWS Marketplace AMI
  - NetScaler Web App Firewall (Web App Firewall)-200 Mbps: Citrix Web App Firewall (Web アプリケーションファイアウォール)-200 Mbps
  - NetScaler Web App Firewall (Web App Firewall)-1000 Mbps: Citrix Web App Firewall (Web アプリケーションファイアウォール)-1000 Mbps
- AMI ページで、[購読を続ける] を選択します。

- ソフトウェアの使用に関する契約条件を確認し、[**Accept**Terms] を選択します。



> 注:
>
> ユーザーは確認ページを受け取り、アカウント所有者に確認メールが送信されます。サブスクリプションの詳細な手順については、AWS Marketplace ドキュメントの「はじめに」の「はじめに」を参照してください。

- サブスクリプションプロセスが完了したら、それ以上アクションを行わずに AWS Marketplace を終了します。AWS Marketplace からソフトウェアをプロビジョニングしないでください。ユーザーは、クイックスタートガイドを使用して AMI をデプロイします。

ステップ **3: AWS** クイックスタートを起動する

- ユーザー AWS アカウントにサインインし、次のいずれかのオプションを選択して AWS CloudFormation テンプレートを起動します。オプションの選択に関するヘルプについては、このガイドの前半の「展開オプション」を参照してください。
  - 以下のいずれかの AWS CloudFormation テンプレートを使用して、AWS 上の新しい VPC に NetScaler VPX をデプロイします:

    * Citrix/Citrix-ADC-AWS-CloudFormation /テンプレート/高可用性/クロス可用性ゾーン
    * Citrix/Citrix-ADC-AWS-CloudFormation /テンプレート/高可用性/同一可用性ゾーン
  - AWS-Quickstart/QuickStart-Citrix-ADC-ウェブアプリファイアウォールは次の場所にある AWS クイックスタートテンプレートを使用して、AWS 上の新規または既存の VPC に NetScaler Web App Firewall をデプロイします。

> 重要:
>
> ユーザーが NetScaler Web App Firewall を既存の VPC に展開する場合、VPC が 2 つの可用性ゾーンにまたがり、ワークロードインスタンスの各可用性ゾーンに 1 つのパブリックサブネットと 2 つのプライベートサブネットがあり、サブネットが共有されていないことを確認する必要があります。このデプロイガイドは共有サブネットをサポートしていません。共有 VPC の操作: 共有 VPCの操作を参照してください。これらのサブネットでは、インスタンスがインターネットに公開されずにパッケージやソフトウェアをダウンロードできるように、ルートテーブルに NAT Gateway が必要です。NAT Gateway の詳細については、「NAT Gateway」を参照してください。サブネットが重複しないようにサブネットを構成します。

また、ユーザーは、DHCP オプションのドメイン名オプションが、Amazon VPC ドキュメントの「DHCP オプションセット:DHCP オプションセット」で説明されているとおりに構成されていることを確認する必要があります。 ユーザーは、クイックスタートガイドを起動したときに VPC 設定の入力を求められます。

- 各デプロイが完了するまでに約 15 分かかります。
- ナビゲーションバーの右上隅に表示される AWS リージョンを確認し、必要に応じて変更します。ここで、Citrix Web App Firewall のネットワークインフラストラクチャが構築されます。テンプレートは、デフォルトで米国東部 (オハイオ) リージョンで起動されます。

> 注:
>
> このデプロイには NetScaler Web App Firewall が含まれていますが、これは現在すべての AWS リージョンでサポートされているわけではありません。サポートされているリージョンの最新リストについては、「AWS サービスエンドポイント: AWS サービスエンドポイント」を参照してください。

- **[テンプレートの選択]** ページで、テンプレート URL のデフォルト設定を保持し、[次へ] を選択します。
- **[詳細の指定]** ページで、ユーザーの都合に合わせてスタック名を指定します。テンプレートのパラメータを確認します。入力が必要なパラメータの値を指定します。その他すべてのパラメータについては、デフォルト設定を確認し、必要に応じてカスタマイズします。
- 次の表では、パラメータがカテゴリ別にリストされ、デプロイオプションについて個別に説明されています:
- NetScaler Web App Firewall を新規または既存の VPC に展開するためのパラメータ（展開オプション 1）
- パラメータのレビューとカスタマイズが完了したら、[次へ] を選択する必要があります。

NetScaler Web App Firewall を新しい **VPC** に展開するためのパラメーター

VPC ネットワーク設定　このデプロイに関する参考情報については、AWS-QuickStart/QuickStart-Citrix-ADC-WebApp Firewall/Templates という CFT テンプレートを参照してください。

パラメータラベル **(**名前**)**	デフォルト	説明
プライマリアベイラビリティゾーン **(primaryAvailabilityZone)**	入力が必要	プライマリ NetScaler Web App Firewall 展開の可用性ゾーン
セカンダリ可用性ゾーン **(**セカンダリ可用性ゾーン**)**	入力が必要	NetScaler Web App Firewall セカンダリ展開の可用性ゾーン
VPC CIDR (VPCCIDR)	10.0.0.0/16	VPC の CIDR ブロック。x.x.x.x/x 形式の有効な IP CIDR 範囲である必要があります。
リモート **SSH CIDR IP (**管理**)** (restrictedsSHCIDR)	入力が必要	EC2 インスタンスに SSH できる IP アドレス範囲 (ポート:22)。
リモート **HTTP CIDR IP (**クライアント**)** (制限付き WebAppCIDR)	0.0.0.0/0	EC2 インスタンスに HTTP できる IP アドレス範囲 (ポート:80)
リモート **HTTP CIDR IP (**クライアント**)** (制限付き WebAppCIDR)	0.0.0.0/0	EC2 インスタンスに HTTP できる IP アドレス範囲 (ポート:80)
プライマリ管理プライベートサブネット **CIDR** (プライマリ管理プライベートサブネット CIDR)	10.0.1.0/24	可用性ゾーン 1 にあるプライマリ管理サブネットの CIDR ブロック。
プライマリ管理プライベート **IP** (プライマリ管理プライベート IP)	—	プライマリ管理サブネット CIDR からプライマリ管理 ENI（最後のオクテットは 5 から 254 の間でなければならない）に割り当てられたプライベート IP。
プライマリクライアントパブリックサブネット **CIDR** (primary ClientPublicSubnetCIDR)	10.0.2.0/24	可用性ゾーン 1 にあるプライマリクライアントサブネットの CIDR ブロック。
プライマリクライアントプライベート **IP** (プライマリクライアントプライベート IP)	—	プライマリクライアントサブネット CIDR のプライマリクライアント IP からプライマリクライアント ENI（最後のオクテットは 5～254 でなければなりません）に割り当てられたプライベート IP。

パラメータラベル **(**名前**)**	デフォルト	説明
プライマリサーバプライベートサブネット **CIDR** (プライマリサーバプライベートサブネット CIDR)	10.0.3.0/24	可用性ゾーン 1 にあるプライマリサーバーの CIDR ブロック。
プライマリサーバプライベート **IP** (プライマリサーバプライベート IP)	—	プライマリサーバサブネット CIDR からプライマリサーバ ENI（最後のオクテットは 5 〜254）に割り当てられたプライベート IP。
セカンダリ管理プライベートサブネット **CIDR** (セカンダリ管理プライベートサブネット CIDR)	10.0.4.0/24	可用性ゾーン 2 にあるセカンダリ管理サブネットの CIDR ブロック。
セカンダリ管理プライベート **IP** (セカンダリ管理プライベート IP)	—	セカンダリ管理 ENI に割り当てられたプライベート IP（最後のオクテットは 5 〜254 である必要があります）。セカンダリ管理サブネット CIDR からセカンダリ管理 IP を割り当てます。
セカンダリクライアントパブリックサブネット **CIDR** (セカンダリ ClientPublicSubnetCIDR)	10.0.5.0/24	可用性ゾーン 2 にあるセカンダリクライアントサブネットの CIDR ブロック。
セカンダリクライアントプライベート **IP** (セカンダリクライアントプライベート IP)	—	セカンダリクライアント ENI に割り当てられたプライベート IP（最後のオクテットは 5 〜254 である必要があります）。セカンダリ・クライアント・サブネット CIDR からセカンダリ・クライアント IP を割り当てます。
セカンダリサーバプライベートサブネット **CIDR** (セカンダリサーバプライベートサブネット CIDR)	10.0.6.0/24	可用性ゾーン 2 にあるセカンダリサーバーサブネットの CIDR ブロック。
セカンダリサーバプライベート **IP** (セカンダリサーバプライベート IP)	—	セカンダリサーバ ENI に割り当てられたプライベート IP（最後のオクテットは 5 〜254 である必要があります）。セカンダリサーバサブネット CIDR からセカンダリサーバ IP を割り当てます。

パラメータラベル (名前)	デフォルト	説明
VPC テナンシー属性 (VPCTenancy)	デフォルト	VPC に起動されたインスタンスの許可されたテナンシー。単一の顧客専用の EC2 インスタンスを起動するには、[Dedicated tenancy] を選択します。

踏み台ホスト設定

パラメータラベル (名前)	デフォルト	説明
踏み台ホストが必要 (LinuxBastionHostEIP)	いいえ	デフォルトでは、踏み台ホストは設定されません。ただし、ユーザーがサンドボックスデプロイを選択する場合は、メニューから [はい] を選択します。これにより、プライベートサブネットとパブリックサブネットのコンポーネントへのアクセス権をユーザーに付与する EIP を使用して、パブリックサブネットに **Linux** 踏み台ホストがデプロイされます。

NetScaler Web App Firewall 構成

パラメータラベル (名前)	デフォルト	説明
キーペア名 (keyPairName)	入力が必要	公開鍵と秘密鍵のペア。これにより、ユーザーは起動後にユーザーインスタンスに安全に接続できます。これは、ユーザーが希望する AWS リージョンで作成したキーペアです。「技術要件」セクションを参照してください。

パラメータラベル (名前)	デフォルト	説明
NetScaler インスタンスタイプ (CitrixADCInstanceType)	m4.xlarge	ADC インスタンスに使用する EC2 インスタンスタイプ。選択したインスタンスタイプが AWS Marketplace で利用可能なインスタンスタイプと一致していることを確認してください。一致しない場合、CFT が失敗する可能性があります。
NetScaler ADC AMI ID (Citrix ADC ImageID)	—	NetScaler Web App Firewall 導入に使用される AWS Marketplace AMI。これは、ステップ 2 でサブスクライブした AMI ユーザーと一致する必要があります。
NetScaler ADC VPX IAM ロール (`iam:GetRole`)	—	このテンプレート: AWS-QuickStart/Quickstart-Citrix-ADC-VPX/テンプレートは 、NetScaler VPX に必要な IAM ロールとインスタンスプロファイルを作成します。空のままにすると、CFT は必要な IAM ロールを作成します。
クライアント・パブリック **IP (EIP)** (クライアント・パブリック IP)	いいえ	ユーザーがユーザーのクライアントネットワークインターフェイスにパブリック EIP を割り当てる場合は、[はい] を選択します。それ以外の場合、展開後でも、ユーザーは必要に応じて後で割り当てることができます。

プールライセンス設定

パラメータラベル (名前)	デフォルト	説明
NetScaler コンソールプールライセンス	いいえ	ライセンスに BYOL オプションを選択する場合は、リストから [はい] を選択します。これにより、ユーザーはすでに購入したライセンスをアップロードできます。ユーザーは作業を開始する前に、NetScaler ADC プールキャパシティを構成して、NetScaler Console のプールライセンスが利用可能であることを確認する必要があります。「NetScaler プール容量の構成」を参照してください。
アクセス可能な **NetScaler** コンソール/**NetScaler** コンソールエージェント **IP**	入力が必要	カスタマーライセンスオプションでは、ユーザーが NetScaler Console をオンプレミスで展開するか、エージェントをクラウドに展開するかにかかわらず、入力パラメーターとして使用できる NetScaler Console IP にアクセスできることを確認してください。
ライセンスモード	オプション	ユーザーは 3 つのライセンスモードから選択できます: NetScaler プールキャパシティを設定します。詳しくは、「Citrix ADC プール容量の構成」を参照してください ライセンスモードが Pooled-Licensing の場合のみ、このフィールドが表示されます。これは、BYOL ADC が作成された後に割り当てられるライセンスの初期帯域幅を Mbps 単位で割り当てます。10 Mbps の倍数でなければなりません。
ライセンス帯域幅 **(Mbps)**	0 メガビット/秒	NetScaler VPX チェックインおよびチェックアウトライセンス (CICO)。詳しくは、「Citrix ADC VPX チェックインおよびチェックアウトライセンス」を参照してください NetScaler 仮想 CPU ライセンス。詳しくは、「Citrix ADC 仮想 CPU ライセンス」を参照してください
ライセンスエディション	Premium	プールキャパシティライセンスモードのライセンスエディションはプレミアムです。

パラメータラベル (名前)	デフォルト	説明
アプライアンスプラットフォームタイプ	オプション	ユーザが CICO ライセンスモードを選択する場合のみ、必要なアプライアンスプラットフォームタイプを選択してください。ユーザーには、VPX-200、VPX-1000、VPX-3000、VPX-5000、VPX-8000 などのオプションが表示されます。
ライセンスエディション	Premium	vCPU ベースのライセンスのライセンスエディションはプレミアムです。

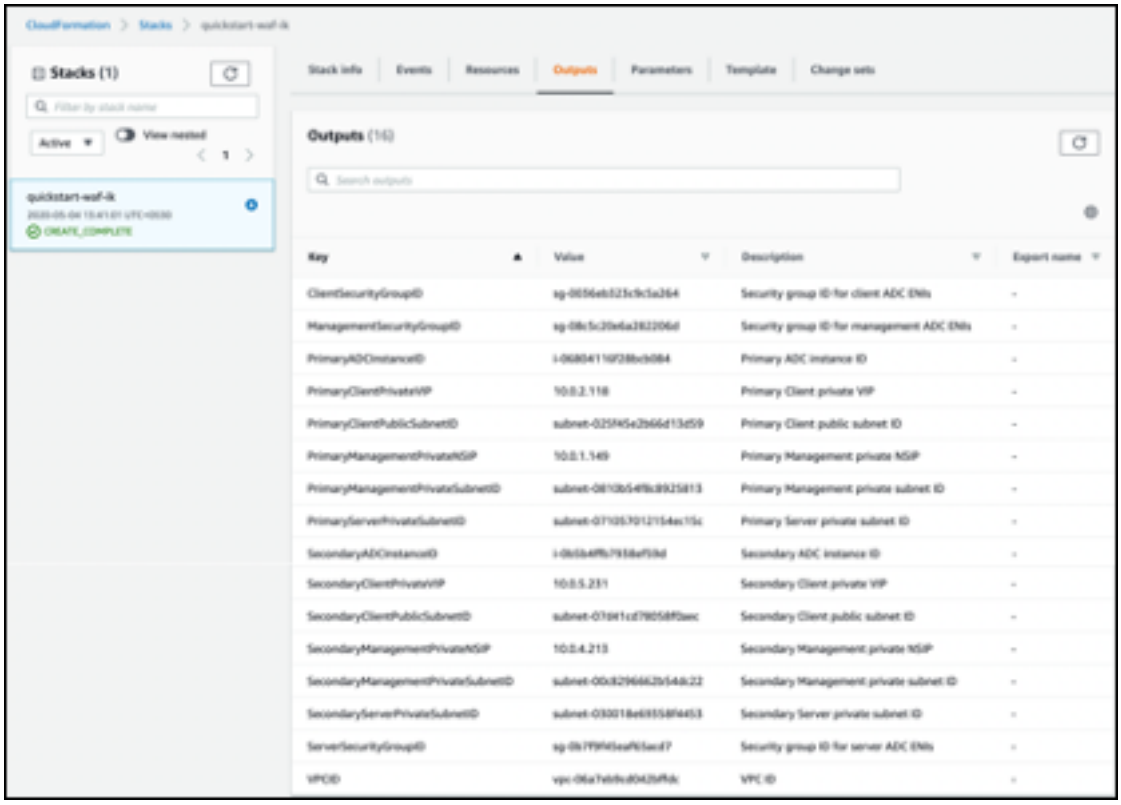
AWS クイックスタート設定

> 注:
>
> 独自の配置プロジェクト用にクイックスタートガイドテンプレートをカスタマイズする場合を除き、次の 2 つのパラメータの既定の設定を維持することをお勧めします。これらのパラメーターの設定を変更すると、コード参照が自動的に更新され、新しいクイックスタートガイドの場所が示されます。詳細については、AWS クイックスタートガイド寄稿者ガイドを参照してください。AWS Quick Starts/Option 1-クイックスタートを採用する。

パラメータラベル (名前)	デフォルト	説明
クイックスタートガイド **S3** バケット名 (qss3BucketName)	`aws-quickstart`	ユーザーがクイックスタートガイドアセットのコピー用に作成した S3 バケット（ユーザーがクイックスタートガイドを自分で使用するためにカスタマイズまたは拡張することを決定した場合）。バケット名には、数字、小文字、大文字、ハイフンを含めることができますが、ハイフンで開始または終了することはできません。

パラメータラベル **(**名前**)**	デフォルト	説明
クイックスタートガイド **S3** キープレフィックス (qss3KeyPrefix)	クイックスタート-citrix-adc-vpx/	[オブジェクトキー] と メタデータ: [オブジェクトキーとメタデータ] の S3 キー名プレフィックスは、クイックスタートガイドアセットのユーザーコピー用のフォルダーをシミュレートするために使用されます。これは、ユーザーがクイックスタートガイドを自分用にカスタマイズまたは拡張することを決定した場合に、クイックスタートガイドアセットのユーザーコピー用のフォルダーをシミュレートするために使用されます。この接頭辞には、数字、小文字、大文字、ハイフン、およびスラッシュを含めることができます。

- **[**オプション] ページで、ユーザーはスタック内のリソースにリソースタグまたはキーと値のペアを指定し、詳細オプションを設定できます。リソースタグの詳細については、「リソースタグ」を参照してください。AWS CloudFormation スタックオプションの設定の詳細については、「AWS CloudFormation スタックオプションの設定」を参照してください。完了したら、[次へ] を選択する必要があります。
- [**Review**] ページで、テンプレートの設定を確認して確認します。[**Capabilities**] で、2 つのチェックボックスをオンにして、テンプレートが IAM リソースを作成し、マクロを自動展開する機能が必要になる可能性があることを確認します。
- [**Create**] を選択してスタックをデプロイします。
- スタックのステータスを監視します。ステータスが **CREATE_COMPLETE** の場合、**NetScaler Web App Firewall** インスタンスは準備完了です。
- スタックの [出力] タブに表示される URL を使用して、作成されたリソースを表示します。



ステップ **4:** デプロイメントをテストする

このデプロイでは、** インスタンスをプライマリとセカンダリと呼びます **。各インスタンスには、それぞれ異なる IP アドレスが関連付けられています。クイックスタートが正常に展開されると、トラフィックは可用性ゾーン 1 で構成されたプライマリ NetScaler Web App Firewall インスタンスを経由します。フェイルオーバー状態で、プライマリインスタンスがクライアントの要求に応答しない場合、セカンダリの Web App Firewall インスタンスが引き継ぎます。

プライマリインスタンスの仮想 IP アドレスの Elastic IP アドレスがセカンダリインスタンスに移行され、セカンダリインスタンスが新しいプライマリインスタンスとして引き継がれます。

フェイルオーバープロセスでは、NetScaler Web App Firewall は次の処理を行います:

- NetScaler Web App Firewall は、IP セットが接続されている仮想サーバーをチェックします。
- NetScaler Web App Firewall は、仮想サーバーがリッスンしている 2 つの IP アドレスから、関連するパブリック IP アドレスを持つ IP アドレスを見つけます。1 つは仮想サーバに直接接続され、もう 1 つは IP セットを介して接続されます。
- NetScaler Web App Firewall は、パブリック Elastic IP アドレスを、新しいプライマリ仮想 IP アドレスに属するプライベート IP アドレスに再関連付けます。

デプロイを検証するには、以下を実行します:

- プライマリインスタンスに接続する

たとえば、プロキシサーバー、ジャンプホスト（AWS で実行されている Linux/Windows/FW インスタンス、または踏み台ホスト）、またはその VPC に到達可能な別のデバイス、またはオンプレミス接続を扱う場合は直接接続などです。

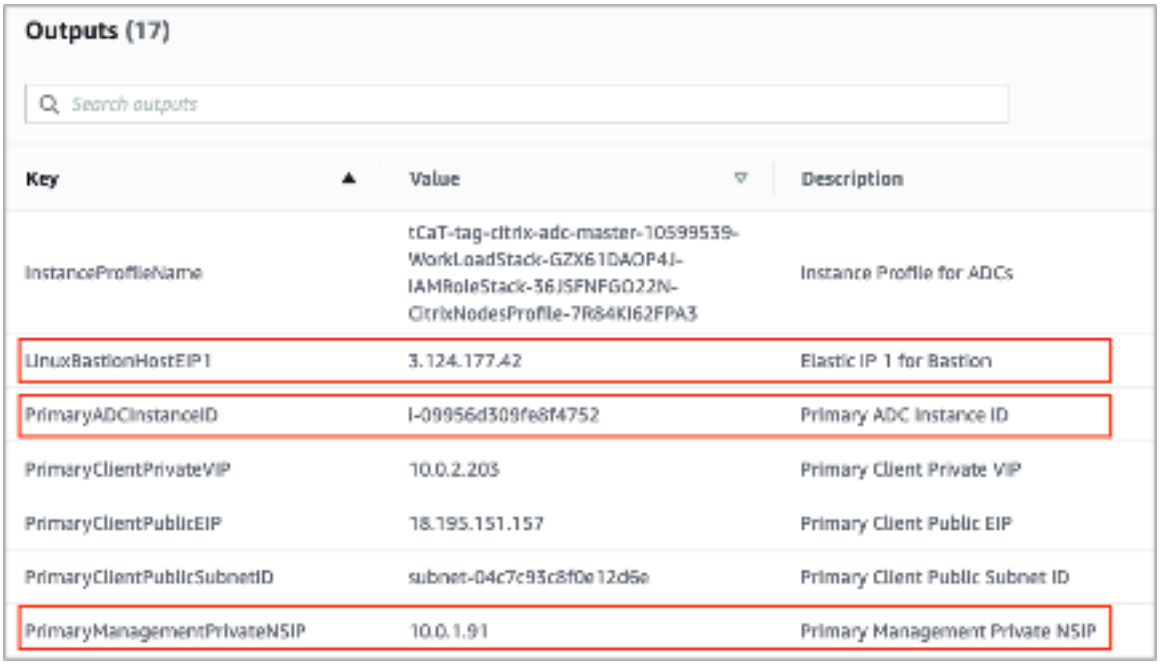
- トリガーアクションを実行してフェイルオーバーを強制し、セカンダリインスタンスが引き継ぐかどうかを確認します。

> ヒント：
>
> NetScaler Web App Firewall に関する構成をさらに検証するには、プライマリ **NetScaler**Web App Firewall インスタンスに接続した後に次のコマンドを実行します。
>
> ```
> Sh appfw profile QS-Profile
> ```

踏み台ホストを使用して **NetScaler Web App Firewall HA** ペアに接続

ユーザーがサンドボックス展開を選択している場合（たとえば、CFT の一部としてユーザーが踏み台ホストの設定を選択する場合）、パブリックサブネットにデプロイされた Linux 踏み台ホストは、Web App Firewall インターフェイスにアクセスするように構成されます。

ここでサインインしてアクセスする AWS CloudFormation コンソールで、サインインしてマスタースタックを選択し、[出力] タブで **LinuxBastionHosteIP1** の値を見つけます。



- **PrivateManagementPrivatenSIP** と **primaryADCInstanceId** キーの値は、ADC に SSH 接続するための後のステップで使用します。

- [サービス] を選択します。
- [コンピュート] タブで [**EC2**] を選択します。
  - リソースで、実行中のインスタンスを選択します。
  - プライマリ Web App Firewall インスタンスの [説明] タブで、**IPv4** パブリック **IP** アドレスを書き留めます。ユーザーは SSH コマンドを構築するためにその IP アドレスが必要です。

- キーをユーザーキーチェーンに保存するには、次のコマンドを実行します。`ssh-add -K [your-key-pair].pem`

Linux では、ユーザーは-K フラグを省略する必要があるかもしれません。

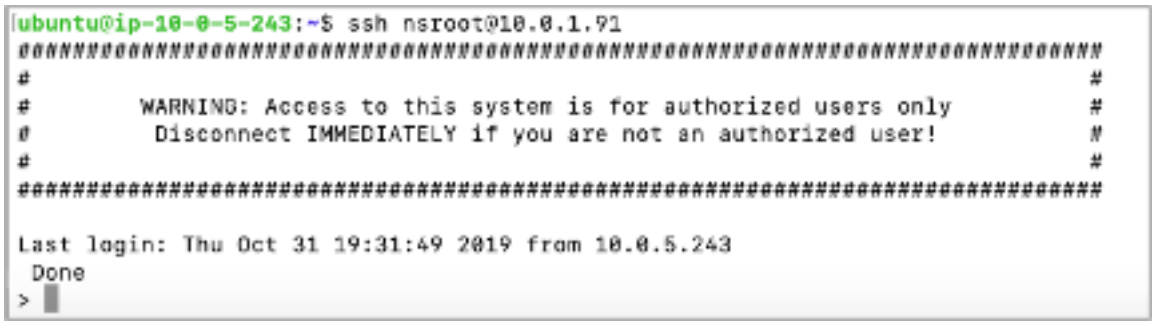
- ユーザーがステップ 1 でメモした **LinuxBastionHosteIP1** の値を使用して、次のコマンドを使用して踏み台ホストにログインします。

```
ssh -A ubuntu@[LinuxBastionHostEIP1]
```

- 踏み台ホストから、ユーザーは SSH を使用してプライマリ Web App Firewall インスタンスに接続できます。

```
ssh nsroot@[Primary Management Private NSIP]
```

パスワード:[プライマリ ADC インスタンス ID]



これで、ユーザーは NetScaler Web App Firewall プライマリインスタンスに接続されました。使用可能なコマンドを確認するには、help コマンドを実行します。現在の HA 設定を表示するには、show HA node コマンドを実行します。

NetScaler コンソール

NetScaler アプリケーション配信管理サービスは、オンプレミスまたはクラウドに展開されている NetScaler MPX、NetScaler VPX、NetScaler Gateway、NetScaler Secure Web Gateway、NetScaler SDX、NetScaler ADC CPX、NetScaler SD-WAN アプライアンスなどの NetScaler 展開を管理するための簡単でスケーラブルなソリューションを提供します。

NetScaler Console Service ドキュメントには、サービスの開始方法、サービスでサポートされている機能のリスト、およびこのサービスソリューションに固有の構成に関する情報が含まれています。

詳しくは、「NetScaler コンソールの概要」を参照してください。

NetScaler コンソールを使用して AWS に NetScaler VPX インスタンスをデプロイする

顧客がアプリケーションをクラウドに移行すると、アプリケーションの一部であるコンポーネントが増え、分散性が高まり、動的に管理する必要があります。

詳細については、「AWS での NetScaler VPX インスタンスの Provisioning」を参照してください。

NetScaler Web App Firewall と OWASP トップ 10 —2017

オープン Web アプリケーションセキュリティプロジェクト：OWASP は 、Web アプリケーションセキュリティのための 2017 年の OWASP トップ 10 をリリースしました。このリストは、最も一般的な Web アプリケーションの脆弱性を説明しており、Web セキュリティを評価するための優れた出発点です。ここでは、これらの欠陥を軽減するために NetScaler Web App Firewall（Web App Firewall）を構成する方法について詳しく説明します。Web App Firewall は、NetScaler （プレミアムエディション）の統合モジュールとしてだけでなく、さまざまなアプライアンスでも利用できます。

詳しくは、「NetScaler Web App Firewall」および「OWASP トップ 10-2021」を参照してください。

OWASP Top 10 の完全なドキュメントは OWASP Top Tenで入手できます。

署名は、ユーザーがユーザーアプリケーションの保護を最適化するのに役立つ次の展開オプションを提供します：

- ネガティブセキュリティモデル：ネガティブセキュリティモデルでは、ユーザーは事前に構成された豊富なシグネチャルールを使用して、パターンマッチングの機能を適用して攻撃を検出し、アプリケーションの脆弱性から保護します。ユーザーは望まないものだけをブロックし、残りを許可します。ユーザーは、ユーザーアプリケーションの特定のセキュリティニーズに基づいて独自の署名ルールを追加して、独自のカスタマイズされたセキュリティソリューションを設計できます。
- ハイブリッドセキュリティモデル: 署名の使用に加えて、ユーザーはポジティブセキュリティチェックを使用して、ユーザーアプリケーションに最適な構成を作成できます。署名を使用してユーザーが望まないものをブロックし、肯定的なセキュリティチェックを使用して許可されているものを強制します。

署名を使用してユーザーアプリケーションを保護するには、ユーザーは署名オブジェクトを使用するように 1 つ以上のプロファイルを構成する必要があります。ハイブリッドセキュリティ構成では、ユーザー署名オブジェクトの SQL インジェクションおよびクロスサイトスクリプティングパターン、および SQL 変換ルールは、シグニチャルールだけでなく、Web アプリケーションファイアウォールプロファイルで設定されたポジティブセキュリティチェックによっても使用されます。署名オブジェクト。

Web アプリケーションファイアウォールは、ユーザ保護された Web サイトおよび Web サービスへのトラフィックを調べ、シグネチャと一致するトラフィックを検出します。一致は、ルール内のすべてのパターンがトラフィックに一致する場合にのみトリガーされます。一致が発生すると、ルールに対して指定されたアクションが呼び出されます。要求がブロックされると、ユーザーはエラーページまたはエラーオブジェクトを表示できます。ログメッセージは、ユーザーがユーザーアプリケーションに対して開始されている攻撃を特定するのに役立ちます。ユーザーが統計を有効にすると、Web アプリケーションファイアウォールは、Web アプリケーションファイアウォールの署名またはセキュリティチェックに一致する要求に関するデータを保持します。

トラフィックがシグニチャとポジティブセキュリティチェックの両方に一致する場合、2 つのアクションのうち、より厳しい制限が適用されます。たとえば、ブロックアクションが無効になっているシグネチャルールにリクエストが一致し、アクションがブロックされている SQL Injection ポジティブセキュリティチェックにも一致する場合、リクエストはブロックされます。この場合、署名違反は [ブロックされていない] として記録される可能性がありますが、要求は SQL インジェクションチェックによってブロックされます。

カスタマイズ: 必要に応じて、ユーザーは独自のルールを署名オブジェクトに追加できます。ユーザーは SQL/XSS パターンをカスタマイズすることもできます。ユーザーアプリケーションの特定のセキュリティニーズに基づいて独自の署名ルールを追加するオプションにより、ユーザーは独自のカスタマイズされたセキュリティソリューションを柔軟に設計できます。ユーザーは望まないものだけをブロックし、残りを許可します。特定の高速一致パターンを指定の場所に配置すると、処理オーバーヘッドを大幅に削減してパフォーマンスを最適化できます。ユーザーは、SQL インジェクションおよびクロスサイトスクリプティングパターンを追加、変更、または削除できます。組み込みの正規表現と式エディタは、ユーザーがユーザーパターンを構成し、その正確性を検証するのに役立ちます。

NetScaler Web App Firewall

Web App Firewall は、最新のアプリケーションに最先端の保護を提供するエンタープライズグレードのソリューションです。NetScaler Web App Firewall は、Web サイト、Web アプリケーション、API などの一般公開資産に対する脅威を軽減します。NetScaler Web App Firewall には、IP レピュテーションベースのフィルタリング、ボット対策、OWASP トップ 10 アプリケーション脅威対策、レイヤー 7 DDoS 保護などが含まれています。また、認証を強制するオプション、強力な SSL/TLS 暗号、TLS 1.3、レート制限、および書き換えポリシーも含まれています。NetScaler Web App Firewall は、基本的な保護機能と高度な Web App Firewall 保護の両方を使用して、比類のない使いやすさでアプリケーションを包括的に保護します。起動して実行するのはほんの数分です。さらに、NetScaler Web App Firewall は動的プロファイリングと呼ばれる自動学習モデルを使用することで、ユーザーの貴重な時間を節約できます。Web App Firewall は、保護されたアプリケーションの仕組みを自動的に学習することで、開発者がアプリケーションをデプロイしたり変更したりしても、アプリケーションに適応します。NetScaler Web App Firewall は、PCI-DSS、HIPAA などを含むすべての主要な規制基準や機関へのコンプライアンスに役立

ちます。CloudFormation テンプレートを使えば、これまでになく簡単に立ち上げてすぐに実行できます。Auto Scaling を使用すると、トラフィックが拡大しても、ユーザーはアプリケーションを保護したまま安心できます。

Web App Firewall 導入戦略

Web アプリケーションファイアウォールを展開するための最初のステップは、最大限のセキュリティ保護が必要なアプリケーションまたは特定のデータ、脆弱性の低いアプリケーション、およびセキュリティ検査を安全に回避できるものを評価することです。これにより、ユーザーは最適な構成を考案し、トラフィックを分離するための適切なポリシーとバインドポイントを設計できます。たとえば、ユーザーは、画像、MP3 ファイル、ムービーなどの静的な Web コンテンツに対する要求のセキュリティ検査をバイパスするポリシーを構成し、動的コンテンツのリクエストに高度なセキュリティチェックを適用する別のポリシーを構成することができます。ユーザーは複数のポリシーとプロファイルを使用して、同じアプリケーションの異なるコンテンツを保護できます。

次のステップは、展開のベースラインを設定することです。まず、仮想サーバーを作成し、そのサーバーを介してテストトラフィックを実行して、ユーザーシステムを通過するトラフィックの速度と量を把握します。

次に、Web App Firewall をデプロイします。NetScaler コンソールと Web App Firewall StyleBook を使用して、Web App Firewall を構成します。詳細については、このガイドの下の「StyleBook」セクションを参照してください。

Web アプリファイアウォールを Web App Firewall StyleBook で展開および構成したら、次のステップとして役立つ次のステップは、NetScaler ADC Web App Firewall と OWASP トップ 10 を実装することです。

最後に、Web App Firewall 保護のうち 3 つは、一般的な種類の Web 攻撃に対して特に効果的であるため、他のどの保護よりも一般的に使用されています。したがって、これらは初期展開時に実装する必要があります。

詳しくは、「Azure への NetScaler Web App Firewall 展開」を参照してください。

NetScaler コンソール

NetScaler コンソールは、オンプレミスまたはクラウドに展開されている NetScaler ADC MPX、NetScaler ADC VPX、NetScaler Gateway、NetScaler Secure Web Gateway、NetScaler ADC SDX、NetScaler ADC CPX、NetScaler SD-WAN アプライアンスなどの NetScaler ADC 展開を管理するためのスケーラブルなソリューションを提供します。

NetScaler コンソールのアプリケーション分析 および管理機能

NetScaler コンソールでサポートされている機能は、アプリセキュリティにおける NetScaler Console の役割にとって重要です。

機能の詳細については、「機能と解決策」を参照してください。

前提条件

AWS で VPX インスタンスを作成する前に、ユーザーは前提条件が満たされていることを確認する必要があります。詳細については、「前提条件」を参照してください:

制限事項と使用ガイドライン

AWS に Citrix ADC VPX インスタンスをデプロイする場合、「制限と使用ガイドライン」に記載されている制限および使用ガイドラインが適用されます 。

技術的要件

ユーザーがクイックスタートガイドを起動してデプロイを開始する前に、次のリソーステーブルで指定されているようにユーザーアカウントを設定する必要があります。そうしないと、デプロイが失敗する可能性があります。

リソース

必要に応じて、ユーザー Amazon アカウントにサインインし、次のリソースのサービス制限の引き上げをリクエストします。AWS/Sign in。これらのリソースを使用する既存の展開がすでにあり、この展開で既定の制限を超える可能性があると思われる場合は、これを実行する必要があります。デフォルトの制限については、AWS ドキュメント「AWS サービスクォータ」の「AWS サービスクォータ」を参照してください。

AWS Trusted Advisor (こちらを参照): AWS/Sign inは、一部のサービスのいくつかの側面の使用状況と制限を表示するサービス制限チェックを提供します。

リソース	この展開では、
VPC	1
エラスティック IP アドレス	0/1 (踏み台ホスト用)
IAM セキュリティグループ	3
IAM ロール	1
サブネット	6 (3/アベイラビリティゾーン)
インターネットゲートウェイ	1
ルートテーブル	5
Web App Firewall VPX インスタンス	2
踏み台ホスト	0/1
NAT ゲートウェイ	2

領域

AWS 上の NetScaler Web App Firewall は、現在すべての AWS リージョンでサポートされているわけではありません。サポートされているリージョンの最新リストについては、AWS ドキュメント「AWS サービスエンドポイント」の「AWS サービスエンドポイント」を参照してください。

AWS リージョンの詳細と、クラウドインフラストラクチャが重要な理由については、「グローバルインフラストラクチャ」を参照してください。

キー ペア

クイックスタートガイドを使用して、ユーザーがデプロイする予定のリージョンのユーザー AWS アカウントに、少なくとも 1 つの Amazon EC2 キーペアが存在することを確認します。キーペア名を書き留めます。ユーザーは、展開中にこの情報の入力を求められます。キーペアを作成するには、AWS ドキュメント「Amazon EC2 キーペアと Linux インスタンス」の Amazon EC2 キーペアと Linux インスタンスの指示に従います。

ユーザーがテストまたは概念実証の目的でクイックスタートガイドをデプロイする場合は、本番インスタンスですでに使用されているキーペアを指定する代わりに、新しいキーペアを作成することをお勧めします。

参照ドキュメント

- HTML SQL インジェクションチェック
- XML SQL インジェクションチェック
- コマンドラインを使用して HTML クロスサイトスクリプティングチェックを設定する
- XML クロスサイトスクリプティングチェック
- コマンドラインによるバッファオーバーフローセキュリティチェックの設定
- 署名オブジェクトの追加または削除
- 署名オブジェクトの設定または変更
- 署名オブジェクトの更新
- Snort 規則の統合
- ボットの検出
- Microsoft Azure で NetScaler VPX インスタンスを展開する

SR-IOV ネットワークインターフェイスの使用を NetScaler ADC VPX インスタンスで構成する

August 15, 2023

> 注
>
> 高可用性セットアップでの SR-IOV インターフェイスのサポートは、NetScaler ADC リリース 12.0 57.19 以降から利用できます。

AWS で NetScaler ADC VPX インスタンスを作成した後、AWS CLI を使用して、SR-IOV ネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

NetScaler VPX 3G および 5G の NetScaler ADC VPX AWS マーケットプレイスエディションを除き、すべての NetScaler ADC VPX モデルでは、ネットワークインターフェイスのデフォルト構成で SR-IOV が有効になっていません。

設定を開始する前に、次のトピックをお読みください。

- Prerequisites
- 制限事項および使用上のガイドライン

このセクションでは、以下のトピックについて説明します。

- インターフェイスタイプを SR-IOV に変更
- 高可用性セットアップでの SR-IOV の設定

インターフェイスタイプを SR-IOV に変更

show interface summary コマンドを実行すると、ネットワークインターフェイスのデフォルト設定を確認できます。

例 **1**: 次の CLI スクリーンキャプチャは、NetScaler VPX AWS Marketplace エディションの 3G および 5G で SR-IOV がデフォルトで有効になっているネットワークインターフェイスの構成を示しています。

```
> show interface summary
-------------------------------------------------------------------------------
      Interface  MTU        MAC                Suffix
-------------------------------------------------------------------------------
1       1/1      1500       0a:1e:2e:17:a2:37  Intel 82599 10G VF Interface
2       LO/1     1500       0a:1e:2e:17:a2:37  Netscaler Loopback interface
 Done
```

例 **2:** 次の CLI 画面キャプチャは、SR-IOV が有効になっていないネットワークインターフェイスのデフォルト設定を示しています。


```
 Done
> sh int s
-------------------------------------------------------------------------------
      Interface  MTU        MAC                Suffix
-------------------------------------------------------------------------------
1     1/1        1500       12:fc:04:c5:d0:12  NetScaler Virtual Interface
2     LO/1       1500       12:fc:04:c5:d0:12  Netscaler Loopback interface
 Done
>
```

インターフェイスの種類を SR-IOV に変更する方法の詳細については、http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sriov-networking.htmlを参照してください

インターフェイスタイプを **SR-IOV** に変更するには

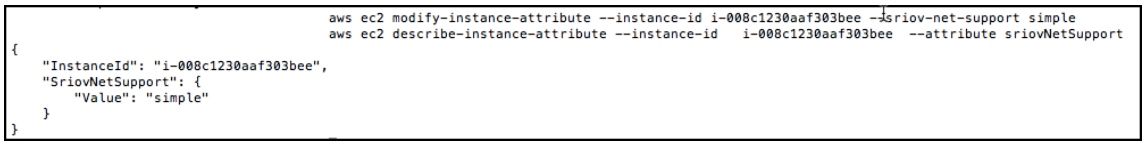
1. AWS の上で動作している NetScaler VPX インスタンスをシャットダウンします。
2. ネットワークインターフェイスで SR-IOV を有効にするには、次のコマンドを AWS CLI に入力します。

   ```
   $ aws ec2 modify-instance-attribute --instance-id \<instance\\_id
   \> --sriov-net-support simple
   ```

3. SR-IOV が有効にされたかどうか確認するには、次のコマンドを AWS CLI に入力します。

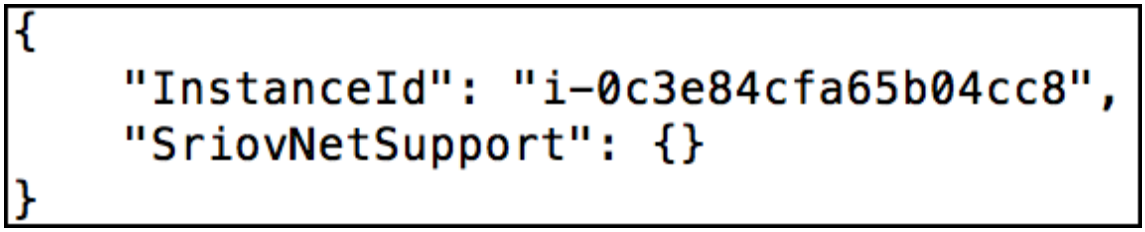
   ```
   $ aws ec2 describe-instance-attribute --instance-id \<instance\\
   _id\> --attribute sriovNetSupport
   ```

   例 **3:** AWS CLI を使用して、ネットワークインターフェイスの種類が SR-IOV に変更されました。

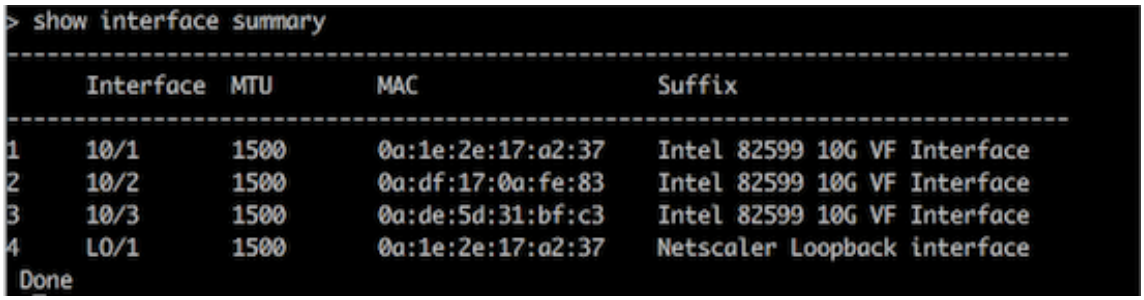
   SR-IOV が有効になっていない場合、SriovNetSupport の値は存在しません。

   例 **4:** 次の例では、SR-IOV サポートが有効になっていません。

4. VPX インスタンスの電源を入れます。ネットワークインターフェイスの変更されたステータスを確認するには、CLI で「show interface summary」と入力します。

例 **5:** 次の画面キャプチャは、SR-IOV が有効になっているネットワークインターフェイスを示しています。インターフェイス 10/1、10/2、10/3 で SR-IOV が有効にされています。



これらの手順では、SR-IOV ネットワークインターフェイスを使用するように VPX インスタンスを構成する手順を完了します。

高可用性セットアップでの **SR-IOV** の設定

NetScaler リリース 12.0 ビルド 57.19 以降の SR-IOV インターフェイスでは、高可用性がサポートされています。

高可用性セットアップを手動で展開した場合、または NetScaler バージョン 12.0 56.20 以前の Citrix CloudFormation テンプレートを使用して展開した場合、高可用性セットアップにアタッチされた IAM ロールには次の権限が必要です。

- ec2:DescribeInstances
- ec2:DescribeNetworkInterfaces
- ec2:DetachNetworkInterface
- ec2:AttachNetworkInterface
- ec2:StartInstances
- ec2:StopInstances
- ec2:RebootInstances
- autoscaling:*
- サンズ:*
- sqs:*
- IAM: プリンシパルポリシーのシミュレーション
- IAM:GetRole

デフォルトでは、NetScaler ADC バージョン 12.0 57.19 用の Citrix CloudFormation テンプレートによって、必要な権限が IAM ロールに自動的に追加されます。

> 注
>
> SR-IOV インターフェイスを使用したハイアベイラビリティのセットアップには、約 100 秒のダウンタイムが発生します。

関連リソース:

IAM ロールの詳細については、AWS ドキュメントを参照してください。

AWS ENA での拡張ネットワークの使用を NetScaler ADC VPX インスタンスで構成する

August 15, 2023

AWS で NetScaler ADC VPX インスタンスを作成した後、AWS CLI を使用して、AWS Elastic Network Adapter（ENA）を使用した拡張ネットワーキングを使用するように仮想アプライアンスを構成できます。

AWS ENA と組み合わせると、拡張ネットワーキングは、より高い帯域幅、高いパケット/秒（PPS）パフォーマンス、一貫して低いインスタンス間レイテンシーを提供します。

設定を開始する前に、次のトピックをお読みください。

- Prerequisites
- 制限事項および使用上のガイドライン

ENA 対応インスタンスでは、次の HA 構成がサポートされています。

- プライベート IP アドレスは同じアベイラビリティーゾーン内で移動できます。
- Elastic IP アドレスは、アベイラビリティーゾーン間で移動できます。

AWS 上の NetScaler VPX インスタンスのアップグレード

August 15, 2023

AWS 上で動作する NetScaler VPX の EC2 インスタンスの種類、スループット、ソフトウェアエディション、およびシステムソフトウェアをアップグレードすることができます。一部のアップグレード方法では、高可用性構成を使用してダウンタイムを最小限に抑えることができます。

> 注:
>
> - NetScaler VPX AMI 用の NetScaler ソフトウェアの Release 10.1.e-124.1308.e 以降（ユーティリティライセンスおよびカスターマーライセンスを含む）では、M1 および M2 のインスタンスファミリをサポートしません。
> - VPX インスタンスのサポートが変更されたため、10.1.e-124 以降のリリースから 10.1.123.x またはそれ以前のリリースへのダウングレードはサポートされていません。
> - ほとんどのアップグレードでは新規の AMI を起動する必要はなく、現在の NetScaler AMI インスタン

> ス上でアップグレードできます。新規の NetScaler AMI インスタンスへのアップグレードを行う場合は、高可用性構成を使用してください。

AWS 上の NetScaler VPX インスタンスの EC2 インスタンスタイプを変更する

Release 10.1.e-124.1308.e 以降が動作する NetScaler VPX インスタンスでは、AWS コンソールで EC2 インスタンスの種類を変更できます。次の手順に従います。

1. VPX インスタンスを停止します。
2. AWS コンソールで EC2 インスタンスの種類を変更します。
3. インスタンスを起動します。

上記の手順は、Release 10.1.e-124.1308.e よりも前の NetScaler VPX インスタンスでも使用できます。ただし、EC2 インスタンスの種類を M3 に変更することはできません。その場合は、NetScaler ADC ソフトウェアを 10.1.e-124 以降のリリースにアップグレードするには、まず標準の NetScaler ADC アップグレード手順（）に従って、上記の手順を実行する必要があります。

AWS での NetScaler VPX インスタンスのスループットまたはソフトウェアエディションのアップグレード

ソフトウェアエディション（Standard エディションから Premium エディションへのアップグレードなど）またはスループット（たとえば、200 Mbps から 1000 Mbps へのアップグレードなど）をアップグレードするには、インスタンスのライセンスによって異なります。

カスタマーライセンスの使用 (自分のライセンスの持ち込み)

カスタマーライセンスを使用している場合は、Citrix の Web サイトから新しいライセンスを購入してダウンロードし、VPX インスタンスにライセンスをインストールできます。Citrix Web サイトからのライセンスのダウンロードとインストールの詳細については、VPX ライセンスガイドを参照してください。

ユーティリティライセンス (時間単位のユーティリティライセンス) を使用する

AWS では課金ベースのインスタンスの直接アップグレードがサポートされていません。課金ベースの NetScaler VPX インスタンスのソフトウェアエディションやスループットをアップグレードする場合は、適切なライセンスおよびキャパシティの新規の AMI を起動して、古いインスタンスから構成を移行します。これは、このページの「Citrix ADC 高可用性構成を使用した新しい NetScaler ADC AMI インスタンスへのアップグレード」の説明に従って、NetScaler ADC 高可用性構成を使用して実現できます 。

AWS での NetScaler VPX インスタンスのシステムソフトウェアのアップグレード

10.1.e-124.1308.e 以降のリリースを実行している VPX インスタンスをアップグレードする必要がある場合は、「Citrix ADC アプライアンスのアップグレードとダウングレード」の標準の NetScaler ADC アップグレード手順に従ってください。

10.1.e-124.1308.e より古いリリースを実行している VPX インスタンスを 10.1.e-124.1308.e 以降のリリースにアップグレードする必要がある場合は、まずシステムソフトウェアをアップグレードしてから、インスタンスタイプを次のように M3 に変更します。

1. VPX インスタンスを停止します。
2. AWS コンソールで EC2 インスタンスの種類を変更します。
3. インスタンスを起動します。

NetScaler の高可用性構成を使用して新しい NetScaler AMI インスタンスにアップグレードする

高可用性構成を使用して新しい NetScaler AMI インスタンスにアップグレードするには、以下のタスクを実行します。

- AWS Marketplace で、EC2 インスタンスの種類、ソフトウェアエディション、スループット、またはソフトウェアリリースを指定して新しいインスタンスを作成します。
- 古いインスタンス（アップグレード前）と新しいインスタンスとの間に高可用性を構成します。これにより、古いインスタンスの構成内容が新しいインスタンスに同期されます。
- 古いインスタンスから新しいインスタンスへの強制高可用性フェールオーバーを実行します。これにより、新しいインスタンスがプライマリノードとして設定され、新しいトラフィックを受信し始めます。
- 古いインスタンスを停止して、再構成するか AWS から削除します。

前提条件と考慮すべき点

- AWS 上の 2 つの NetScaler ADC VPX インスタンス間で高可用性がどのように機能するかを理解してください。AWS 上の 2 つの NetScaler ADC VPX インスタンス間の高可用性構成の詳細については、「AWS での高可用性ペアのデプロイ」を参照してください。
- 新しいインスタンスは、古いインスタンスと同じアベイラビリティゾーン内に作成し、同じセキュリティグループおよびサブネットが設定されている必要があります。
- 高可用性のセットアップでは、両インスタンスのユーザーの AWS IAM (Identity and Access Management) アカウントに関連付けられたアクセスキーと秘密キーが必要です。正しいキー情報を使用して VPX インスタンスを作成しないと、高可用性のセットアップに失敗します。VPX インスタンスの IAM アカウントの作成の詳細については、「前提条件」を参照してください。
  - 新しいインスタンスを作成するには EC2 コンソールを使用する必要があります。AWS の 1-Click 起動は使用できません。これは、アクセスが許可されず、秘密キーを入力できないためです。

- 新しいインスタンスには ENI インターフェイスが 1 つだけ必要です。

高可用性構成を使用して NetScaler ADC VPX インスタンスをアップグレードするには、次の手順に従います。

1. 古いインスタンスと新しいインスタンスの間で高可用性を構成します。2 つの NetScaler ADC VPX インスタンス間で高可用性を構成するには、各インスタンスのコマンドプロンプトで次のように入力します。

   - `add ha node <nodeID> <IPaddress of the node to be added>`
   - `save config`

   例:

   古いインスタンスのコマンドプロンプトで、次のように入力します。

   ```
   1 add ha node 30 192.0.2.30
   2 Done
   3 <!--NeedCopy-->
   ```

   新しいインスタンスのコマンドプロンプトで、次のように入力します。

   ```
   1 add ha node 10 192.0.2.10
   2 Done
   3 <!--NeedCopy-->
   ```

   以下の点に注意してください:

   - この高可用性セットアップで、古いインスタンスがプライマリノードで新しいインスタンスがセカンダリノードになります。
   - NSIP アドレスは古いインスタンスから新しいインスタンスにコピーされません。このため、アップグレード完了時に新しいインスタンスには異なる管理 IP アドレスが設定されます。
   - 新しいインスタンスの`nsroot`アカウントパスワードは、HA 同期後に古いインスタンスのアカウントパスワードに設定されます。

   AWS 上の 2 つの NetScaler ADC VPX インスタンス間の高可用性構成の詳細については、「AWS での高可用性ペアのデプロイ」を参照してください。

2. HA フェールオーバーを強制します。高可用性構成でフェイルオーバーを強制するには、いずれかのインスタンスのコマンドプロンプトで次のように入力します。

   ```
   1 force HA failover
   2 <!--NeedCopy-->
   ```

   強制フェールオーバーにより、古いインスタンスの ENI が新しいインスタンスに移行され、トラフィックが新しいインスタンス（新しいプライマリノード）に流れます。また、古いインスタンス（新しいセカンダリノード）が再起動します。

   次の警告メッセージが表示されたら、N を入力して操作を中止します。

   ```
   1 [WARNING]:Force Failover may cause configuration loss, peer health
         not optimum. Reason(s):
   ```



```
2 HA version mismatch
3 HA heartbeats not seen on some interfaces
4 Please confirm whether you want force-failover (Y/N)?
5 <!--NeedCopy-->
```

この警告メッセージは、2 つの VPX インスタンスのシステムソフトウェアで高可用性がサポートされていない場合に表示されます。このため、強制フェールオーバー時に古いインスタンスの構成情報が新しいインスタンスに同期されません。

この問題の回避策は次のとおりです。

a) 古いインスタンスの NetScaler シェルプロンプトで、次のコマンドを実行して構成ファイル（ns.conf）のバックアップを作成します。

```
copy /nsconfig/ns.conf to /nsconfig/ns.conf.bkp
```

b) バックアップ構成ファイル（ns.conf.bkp）から次の行を削除します。

- `set ns config -IPAddress <IP> -netmask <MASK>`

例: `set ns config -IPAddress 192.0.2.10 -netmask 255.255.255.0`

c) 古いインスタンスのバックアップ構成ファイル (ns.conf.bkp) を新しいインスタンスの /nsconfig ディレクトリにコピーします。

d) 新しいインスタンスの NetScaler ADC シェルプロンプトで、次のコマンドを入力して、古いインスタンスの構成ファイル（ns.conf.bkp）を新しいインスタンスにロードします。

- `batch -f /nsconfig/ns.conf.bkp`

e) 新しいインスタンスに設定を保存します。

- `save conifg`

f) いずれかのノードのコマンドプロンプトで、次のコマンドを入力してフェールオーバーを強制し、強制フェールオーバー操作を確認する警告メッセージに Y を入力します。

- `force ha failover`

例:

```
1          > force ha failover
2
3 WARNING]:Force Failover may cause configuration loss, peer health
     not optimum.
4          Reason(s):
5          HA version mismatch
6          HA heartbeats not seen on some interfaces
7          Please confirm whether you want force-failover (Y/N)? Y
8 <!--NeedCopy-->
```


3. HA 構成を削除して、2 つのインスタンスが HA 構成に含まれないようにします。これを行うには、まずセカンダリノードの高可用性構成を削除して、次にプライマリノードの高可用性を削除します。

   2 つの NetScaler VPX インスタンス間の高可用性構成を削除するには、各インスタンスのコマンドプロンプトで以下のコマンドを実行します。

```
1      > remove ha node \<nodeID\>
2      >  save config
3  <!--NeedCopy-->
```

   AWS 上の 2 つの VPX インスタンス間の高可用性設定の詳細については、「AWS に高可用性ペアをデプロイする」を参照してください。

   例:

古いインスタンス (新しいセカンダリノード) のコマンドプロンプトで、次のように入力します。

```
1          > remove ha node 30
2            Done
3          > save config
4            Done
5  <!--NeedCopy-->
```

新しいインスタンス (新しいプライマリノード) のコマンドプロンプトで、次のように入力します。

```
1          > remove ha node 10
2            Done
3          > save config
4            Done
5  <!--NeedCopy-->
```

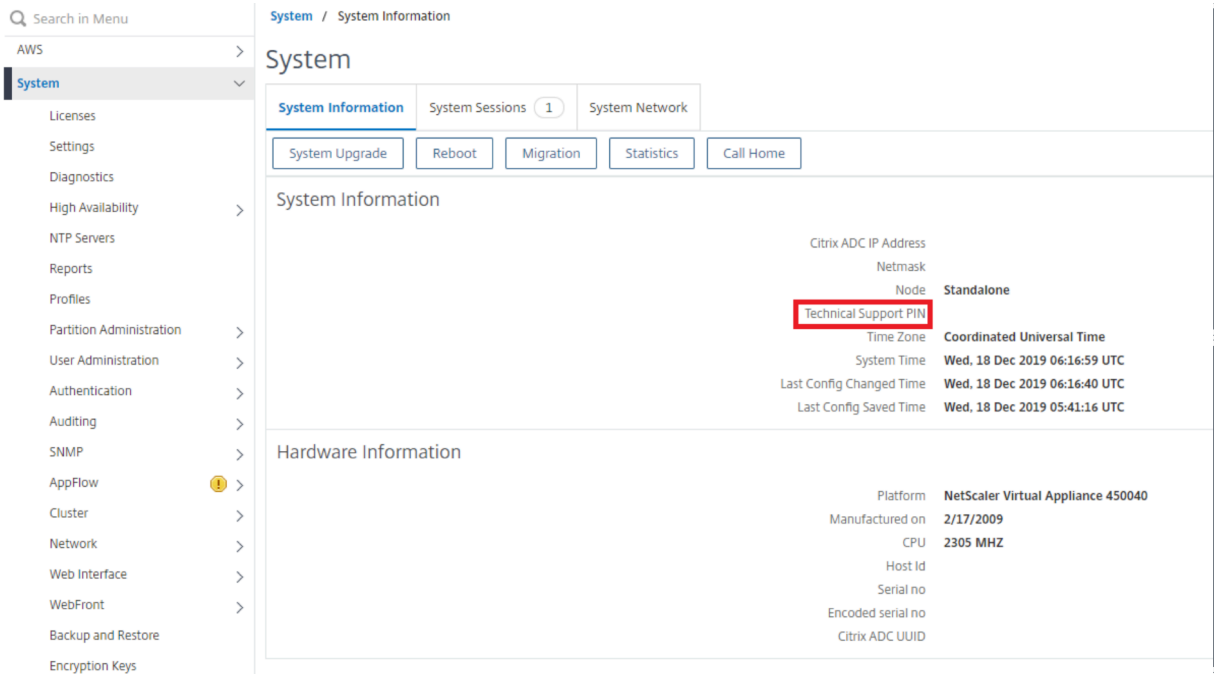
AWS での VPX インスタンスのトラブルシューティング

August 15, 2023

Amazon では、NetScaler VPX インスタンスへのコンソールアクセスを提供していません。トラブルシューティングを行うには、AWS GUI を使用してアクティビティログを表示する必要があります。デバッグできるのは、ネットワークが接続されている場合だけです。インスタンスのシステムログを表示するには、インスタンスを右クリックして [System Log] を選択します。

NetScaler は、AWS マーケットプレイスでライセンスされた NetScaler VPX インスタンス（時間単位の料金がかかるユーティリティライセンス）を AWS 上でサポートします。サポートケースを提出するには、AWS アカウント番号とサポート PIN コードを見つけて、NetScaler サポートに連絡してください。また、名前とメールアドレスの入力を求められます。サポート PIN を見つけるには、VPX GUI にログオンし、システムページに移動します。

サポート PIN を示すシステムページの例を次に示します。



AWS に関するよくある質問

March 20, 2024

- **NetScaler VPX** インスタンスは **AWS** の暗号化されたボリュームをサポートしていますか**?**

  暗号化と復号化はハイパーバイザーレベルで行われるため、どのインスタンスでもシームレスに機能します。暗号化されたボリュームの詳細については、次の AWS ドキュメントを参照してください。

  https://docs.aws.amazon.com/kms/latest/developerguide/services-ebs.html

- **AWS** で **NetScaler VPX** インスタンスをプロビジョニングする最も良い方法は何ですか**?**

  NetScaler VPX インスタンスは、次のいずれかの方法で AWS にプロビジョニングできます。

  - AWS マーケットプレイスの AWS CloudFormation テンプレート (CFT)
  - NetScaler コンソール
  - AWS クイックスタート
  - GitHub の Citrix AWS CFT
  - GitHub の Citrix Terraform スクリプト
  - GitHub の Citrix Ansible プレイブック
  - AWS EC2 起動ワークフロー

  使用するオートメーションツールに基づいて、一覧表示されたオプションのいずれかを選択できます。

  オプションの詳細については、「AWS での NetScaler VPX」を参照してください。

- **AWS** で **NetScaler VPX** インスタンスをアップグレードするには**?**

  AWS で NetScaler VPX インスタンスをアップグレードするには、「AWS での NetScaler VPX インスタンスのアップグレード」の手順に従って、システムソフトウェアをアップグレードするか、新しい NetScaler VPX Amazon Machine Image（AMI）にアップグレードします。

  NetScaler VPX インスタンスをアップグレードする推奨される方法は、ジョブを使用した NetScaler インスタンスのアップグレードの手順に従って、ADM サービスを使用することです。

- **AWS** での **NetScaler VPX** の **HA** フェイルオーバー時間はどれくらいですか**?**

  - AWS アベイラビリティーゾーン内での NetScaler VPX の高可用性フェイルオーバーには約 3 秒かかります。
  - AWS アベイラビリティーゾーン全体の NetScaler VPX の HA フェイルオーバーには、約 5 秒かかります。

- テクニカルサポート **PIN** を提供する **NetScaler VPX** マーケットプレイスのサブスクリプションをご利用のお客様には、どのレベルのサポートが提供されますか**?**

  デフォルトでは、テクニカルサポート PIN を提供するお客様には「ソフトウェアの選択」サービスが提供されます。

- **Elastic IP** デプロイを使用した異なるゾーンでの高可用性では 、アプリケーションごとに複数の **IPSet** を作成する必要がありますか。

  はい。複数の VIP が複数の EIP にマッピングされた複数のアプリケーションがある場合は、複数の IPSet が必要です。したがって、HA フェールオーバー中に、EIP のすべてのプライマリ VIP マッピングがセカンダリ（新しいプライマリ）VIP に変更されます。

- 異なるゾーン展開で高可用性で **INC** モードが有効になるのはなぜですか。

  アベイラビリティーゾーン全体の HA ペアは、異なるネットワークにあります。HA 同期の場合、ネットワーク構成を同期してはいけません。これは、HA ペアで INC モードを有効にすることによって実現されます。

- アベイラビリティーゾーンが同じ **VPC** 内にある場合、あるアベイラビリティーゾーンの **HA** ノードは、別のアベイラビリティーゾーンのバックエンドサーバーと通信できますか。

  はい。同じ VPC の異なるアベイラビリティーゾーンにあるサブネットには、SNIP 経由でバックエンドサーバーのサブネットを指す追加のルートを追加することで到達できます。たとえば、AZ1 の ADC の SNIP サブネットが 192.168.3.0/24、AZ2 のバックエンドサーバーのサブネットが 192.168.6.0/24 の場合、AZ1 に存在する NetScaler アプライアンスに 192.168.6.0 255.255.0 192.168.3.1 としてルートを追加する必要があります。

- **Elastic IP** を使用した異なるゾーン間での高可用性とプライベート **IP**デプロイを使用した異なるゾーン間での高可用性は一緒に機能しますか 。

  はい、両方の設定を同じ HA ペアに適用できます。

- プライベート **IP** デプロイを使用した異なるゾーン間の高可用性で 、**VPC** 内に複数のルートテーブルを持つ複数のサブネットがある場合、**HA** ペアのセカンダリノードは、**HA** フェールオーバー中にチェックされるルートテーブルについてどのように認識しますか。

  セカンダリノードはプライマリ NIC を認識し、VPC 内のすべてのルートテーブルを検索します。

- **AWS** で **VPX** のデフォルトイメージを使用する場合の`/var`パーティションのサイズはどれくらいですか? ディスク容量を増やすには?

  ディスクイメージを小さく保つために、ルートディスクのサイズは 20 GB に制限されています。

  `/var/core/`または`/var/crash/`ディレクトリ領域を増やす場合は、追加のディスクを接続します。`/var`サイズを大きくするには、現在のところ、重要なコンテンツを新しいディスクにコピーした後、追加のディスクを接続して`/var`にシンボリックリンクを作成する必要があります。

- **vCPU** にアクティブ化され、割り当てられるパケットエンジンは何台ありますか。

  パケットエンジン（PE）は、ライセンスされた vCPU の数によって制限されます。NetScaler デーモンは特定の vCPU に固定されず、非 PE vCPU で実行される可能性があります。AWS によると、C5.9xLarge は 72 GB のメモリを持つ 36vCPU インスタンスです。プールライセンスでは、NetScaler VPX インスタンスが最大数の PE でデプロイされます。この場合、19 PE がコア 1 ～19 で実行されます。ただし、ADC 管理プロセスは CPU 20～31 から実行されます。

- **ADC** の適切な **AWS** インスタンスを決定するには?

  1. スループット、PPS、SSL 要件、平均パケットサイズなどのユースケースと要件を理解します。
  2. VPX 帯域幅オファリングや vCPU ベースのライセンスなど、要件を満たす適切な ADC 製品とライセンスを選択します。
  3. 選択したオファリングに基づいて、AWS インスタンスを決定します。

  例:

  5 Gbps ライセンスでは、5 つのデータパケットエンジンが有効になります。したがって、vCPU 要件は 6（管理の場合は 5+1）です。ただし、6 つの vCPU インスタンスは利用できません。したがって、5 Gbps の帯域幅をサポートするネットワークを選択すれば、8 vCPU はそのスループットに到達するのに十分です。たとえば、5 Gbps のライセンスの最大 PE 割り当てを有効にするには、5 Gbps 帯域幅ライセンスに m5.2xlarge を選択する必要があります。ただし、スループットによって制限されない vCPU ライセンスを使用すると、m5.xlarge インスタンス自体を使用して 5 Gbps のスループットが得られる可能性があります。

Instance Size	vCPU	Memory (GiB)	Instance Storage (GiB)	Network Bandwidth (Gbps)	EBS Bandwidth (Mbps)
m5.large	2	8	EBS-Only	Up to 10	Up to 4,750
m5.xlarge	4	16	EBS-Only	Up to 10	Up to 4,750
m5.2xlarge	8	32	EBS-Only	Up to 10	Up to 4,750
m5.4xlarge	16	64	EBS-Only	Up to 10	4,750

- **AWS** の **ADC** には **3** つの **NIC 3** サブネットのデプロイメントが必須ですか?

  `Three NICs-three subnets`は、管理、クライアント、およびサーバーネットワーク用の推奨展開です。この展開により、トラフィックの分離と VPX パフォーマンスが向上します。2 つの NIC-2 サブネット、および 1 つの nic-One サブネットは、他の使用可能なオプションです。Citrix では、2 つの NIC（1 つのサブネットデプロイメント）など、複数の NIC が AWS でサブネットを共有することはお勧めしません。非対称ルーティングなどのネットワークの問題につながる可能性があるためです。詳細については、「AWS でネットワークインターフェイスを設定するためのベストプラクティス」を参照してください。

Microsoft Azure で NetScaler VPX インスタンスを展開する

March 21, 2024

NetScaler VPX インスタンスを Microsoft Azure Resource Manager（ARM）にデプロイすると、次の機能セットの両方を使用してビジネスニーズを満たすことができます。

- Azure クラウドコンピューティング機能
- NetScaler の負荷分散とトラフィック管理機能

NetScaler VPX インスタンスは、スタンドアロンインスタンスとして、またはアクティブ/スタンバイモードの高可用性ペアとして ARM にデプロイできます。

NetScaler VPX インスタンスを Microsoft Azure にデプロイするには、次の 2 つの方法があります。

- Azure マーケットプレイスを通じて。NetScaler VPX 仮想アプライアンスは、Microsoft Azure Marketplace でイメージとして使用することができます。
- GitHub で入手可能な NetScaler Azure Resource Manager (ARM) json テンプレートを使用する。詳細については、NetScaler ソリューションテンプレートの GitHub リポジトリを参照してください。

Microsoft Azure スタックは、ローカルデータセンターに Microsoft Azure パブリッククラウドサービスを提供し、組織がハイブリッドクラウドを構築できるようにするハードウェアとソフトウェアの統合プラットフォームです。NetScaler VPX インスタンスを Microsoft Azure スタックにデプロイできるようになりました。

前提条件

NetScaler VPX インスタンスを Azure にデプロイする前に、ある程度の前提知識が必要です。

- Azure の用語とネットワークの詳細に精通しています。詳細については、Azure の用語を参照してください。
- NetScaler アプライアンスに関する知識。NetScaler アプライアンスの詳細については、「NetScaler」を参照してください。
- NetScaler ネットワークに関する知識。ネットワーキング トピックを参照してください。

NetScaler VPX インスタンスが Azure 上で動作する仕組み

オンプレミス展開では、NetScaler VPX インスタンスは、少なくとも次の 3 つの IP アドレスを必要とします。

- 管理 IP アドレス。NSIP アドレスと呼ばれます。
- サーバーファームとやり取りするためのサブネット IP（SNIP）アドレス
- クライアント要求を受け付ける仮想サーバー IP（VIP）アドレス

詳しくは、「Microsoft Azure 上の NetScaler VPX インスタンスのネットワークアーキテクチャ」を参照してください。

> 注
>
> NetScaler VPX インスタンスは Intel プロセッサと AMD プロセッサの両方をサポートします。VPX 仮想アプライアンスは、2 つ以上の仮想コアと 2 GB を超えるメモリを備えた任意のインスタンスタイプにデプロイできます。システム要件の詳細については、NetScaler VPX のデータシートを参照してください。

Azure 環境では、次の 3 つの方法で Azure 上に NetScaler VPX インスタンスをプロビジョニングできます。

- マルチ NIC マルチ IP アーキテクチャ
- シングル NIC マルチ IP アーキテクチャ
- 単一の NIC シングル IP

ニーズに応じて、これらのサポートされているアーキテクチャタイプのいずれかを使用できます。

マルチ NIC マルチ IP アーキテクチャ

このデプロイタイプでは、VPX インスタンスに複数のネットワークインターフェイス (NIC) をアタッチできます。NIC には、静的または動的パブリック IP アドレスとプライベート IP アドレスを 1 つ以上割り当てることができます。

詳細については、次のユースケースを参照してください:

- 複数の IP アドレスと NIC を使用して高可用性設定を構成する
- PowerShell コマンドを使用して、複数の IP アドレスと NIC を使用した高可用性設定を構成する

> 注
>
> Azure 環境での MAC の移動やインターフェイスのミュートを避けるため、NetScaler VPX インスタンスのデータインターフェイス（タグなし）ごとに VLAN を作成し、NIC のプライマリ IP を Azure でバインドすることをお勧めします。詳細については、CTX224626 の記事を参照してください。

シングル **NIC** マルチ **IP** アーキテクチャ

この導入タイプでは、1 つのネットワークインターフェイス (NIC) が複数の IP 構成 (静的または動的なパブリック IP アドレスとプライベート IP アドレスが割り当てられる) に関連付けられます。
詳細については、次のユースケースを参照してください:

- NetScaler VPX スタンドアロンインスタンスに複数の IP アドレスを構成する
- PowerShell コマンドを使用して、NetScaler VPX スタンドアロンインスタンスに複数の IP アドレスを構成する

単一の **NIC** シングル **IP**

この導入タイプでは、NSIP、SNIP、VIP の機能を実行するために使用される単一の IP アドレスに関連付けられた 1 つのネットワークインターフェイス（NIC）を使用します。

詳しくは、「NetScaler VPX スタンドアロンインスタンスの構成」を参照してください。

> 注
>
> 単一 IP モードは Azure 展開環境でのみ使用することができます。このモードは、オンプレミス、AWS、またはその他の種類の展開にある NetScaler VPX インスタンスでは使用できません。

NetScaler VPX ライセンス

Azure 上の NetScaler VPX インスタンスにはライセンスが必要です。Azure 上で実行される NetScaler VPX インスタンスでは、次のライセンスオプションを使用できます。

- サブスクリプションベースのライセンス：NetScaler VPX アプライアンスは、Azure Marketplace で有料インスタンスとして利用できます。サブスクリプションベースのライセンスは、従量課金制のオプションです。ユーザーは時間単位で課金されます。

  > 注
  >
  > サブスクリプションベースのライセンスインスタンスの場合、サブスクリプションの請求は、特定のライセンスモデルのライセンス期間を通じて適用されます。クラウドの制限により、Azure はサブスクリプションに適用されるライセンスモデルの変更または削除をサポートしていません。サブスクリプションライセンスを変更または削除するには、既存の ADC VM を削除し、必要なライセンスを使用して新しい ADC VM を再作成します。

  NetScaler は、サブスクリプションベースのライセンスインスタンスのテクニカルサポートを提供します。サポートケースを提出するには、「NetScaler on Azure のサポート—時間単位のサブスクリプションライセンス」を参照してください。

- 自分のライセンスを持参 **(BYOL)**: 自分のライセンス (BYOL) を持ち込む場合は、http://support.citrix.com/article/CTX122426にある VPX ライセンスガイドを参照してください。次の操作を実行する必要があります:
  - NetScaler Web サイトのライセンスポータルを使用して、有効なライセンスを生成します。
  - ライセンスをインスタンスにアップロードします。

  > 注
  >
  > Azure スタック環境では、**BYOL** が唯一のライセンスオプションです。

- **NetScaler VPX** チェックイン/チェックアウトライセンス：詳細については、「NetScalerVPX チェックイン/チェックアウトライセンス」を参照してください。

  NetScaler リリース 12.0 56.20 以降、オンプレミスおよびクラウド展開用の NetScaler VPX Express にはライセンスファイルは必要ありません。NetScaler VPX Express の詳細については、Citrix ADC ライセンスの概要の「NetScaler VPX Express ライセンス」セクションを参照してください。

Azure Marketplace では、次の VPX モデルとライセンスタイプを使用できます。

VPX モデル	ライセンスの種類	推奨インスタンス		
		VPX 1 NIC/2 NIC	**VPX 3 NIC**	**VPX** 最大 **8** 個の **NIC**
VPX10	スタンダード、アドバンス、プレミアム	Standard_D2s_v4	Standard_DS3_v2	Standard_DS4_v2
VPX200	スタンダード、アドバンス、プレミアム	Standard_D2s_v4	Standard_DS3_v2	Standard_DS4_v2
VPX1000	スタンダード、アドバンス、プレミアム	Standard_D4s_v4	Standard_DS3_v2	Standard_DS4_v2
VPX3000	スタンダード、アドバンス、プレミアム	Standard_D4s_v4	Standard_D8ds_v5	Standard_DS4_v2
VPX5000	スタンダード、アドバンス、プレミアム	Standard_D8ds_v5	Standard_D8ds_v5	Standard_DS4_v2
VPX8000	スタンダード、アドバンス、プレミアム	Standard_D8ds_v5	Standard_D8ds_v5	Standard_DS4_v2
VPX10000	スタンダード、アドバンス、プレミアム	Standard_D16s_v4	Standard_D16s_v4	Standard_D16s_v4

注意事項:

- 次の VPX モデルで最適なパフォーマンスを得るには、NetScaler VPX インスタンスで Azure アクセラレーテッドネットワーキングを有効にする必要があります。

  - VPX1000
  - VPX3000
  - VPX5000
  - VPX8000
  - VPX10000

  高速ネットワークの構成について詳しくは、「Azure 高速ネットワークを使用するように NetScaler VPX インスタンスを構成する」を参照してください。

- VPX8000 および VPX10000 のライセンスは、BYOL としてのみご利用いただけます。
- Azure Marketplace から購入したサブスクリプションベースの時間単位ライセンスに関係なく、まれに、Azure にデプロイされた NetScaler VPX インスタンスにデフォルトの NetScaler ライセンスが付与されることがあります。これは、Azure インスタンスメタデータサービス (IMDS) の問題が原因で発生します。
- NetScaler VPX インスタンスの構成を変更する前に、ウォームリスタートを実行して、正しい NetScaler VPX ライセンスを有効にします。

Azure における **NetScaler VPX** インスタンスの **IPv6** サポート

リリース 13.1-21.x 以降、NetScaler VPX スタンドアロンインスタンスは Azure の IPv6 アドレスをサポートします。IPv6 アドレスは、Azure クラウドの NetScaler VPX スタンドアロンインスタンスで VIP アドレスと SNIP アドレスとして構成できます。

Azure で IPv6 を有効にする方法については、次の Azure ドキュメントを参照してください。

- Azure 仮想ネットワークの IPv6 とは何ですか?
- Azure 仮想ネットワークの IPv4 アプリケーションに IPv6 を追加する-Azure CLI
- 住所の種類

NetScaler アプライアンスが IPv6 をサポートする方法については、「インターネットプロトコルバージョン 6」を参照してください。

IPv6 の制限事項**:**

- NetScaler の IPv6 環境では現在、Azure バックエンドの自動スケーリングはサポートされていません。
- IPv6 は NetScaler VPX HA 展開ではサポートされていません。

制限事項

NetScaler VPX 負荷分散ソリューションを ARM で実行すると、次の制限があります。

- Azure アーキテクチャでは、以下の NetScaler 機能をサポートしていません。

  - Gratuitous ARP (GARP)
  - L2 モード
  - タグ付き VLAN
  - 動的ルーティング
  - 仮想 MAC
  - USIP
  - クラスタリング

> 注:
>
> NetScaler Application Delivery Management（ADM）Autoscale 機能（クラウド展開）では、ADC インスタンスはすべてのライセンスでクラスタリングをサポートします。詳細については、「NetScaler コンソールを使用した Microsoft Azure での NetScaler VPX の自動スケーリング」を参照してください。

- NetScaler VPX 仮想マシンを任意のタイミングでシャットダウンし、一時的に割り当てを解除しなければならないことが予想される場合は、仮想マシンの作成時に静的内部 IP アドレスを割り当てます。静的内部 IP アドレスを割り当てないと、Azure が再起動のたびに異なる IP アドレスを仮想マシンに割り当てる可能性があり、仮想マシンにアクセスできなくなる場合があります。
- Azure 環境では、VPX 10、VPX 200、VPX 1000、VPX 3000、および VPX 5000 の NetScaler VPX モデルのみがサポートされます。詳しくは、NetScaler VPX のデータシートを参照してください。

  モデル番号が VPX 3000 以降の NetScaler VPX インスタンスを使う場合、ネットワークスループットはインスタンスのライセンスに規定されているのと同じではないことがあります。ただし、SSL スループットや SSL トランザクション/秒など、その他の機能が向上する可能性があります。
- 仮想マシンのプロビジョニング中に Azure によって生成されたデプロイ ID は、ARM のユーザーには表示されません。展開 ID を使用して ARM で NetScaler VPX アプライアンスを展開することはできません。
- NetScaler VPX インスタンスは、初期化時に 20 Mbps のスループットと標準エディションの機能をサポートします。
- 高速ネットワークが有効になっている Azure 上の NetScaler VPX インスタンスは、パフォーマンスが向上します。Azure 高速ネットワークは、リリース 13.0 ビルド 76.x 以降の NetScaler VPX インスタンスでサポートされています。NetScaler VPX で高速ネットワークを有効にするには、高速ネットワークをサポートする Azure インスタンスタイプを使用することをお勧めします。
- Citrix Virtual Apps and Desktops の導入では、VPX インスタンス上の VPN 仮想サーバーを次のモードで構成できます:

  - 基本モード。`ICAOnly` VPN 仮想サーバーパラメーターが ON に設定されます。基本モードは、ライセンスされていない NetScaler VPX インスタンスでも完全に動作します。
  - SmartAccess モード。`ICAOnly` VPN 仮想サーバーパラメーターが OFF に設定されています。SmartAccess モードは、ライセンスのない NetScaler VPX インスタンス上の 5 人の NetScaler AAA セッションユーザーに対してのみ機能します。

> 注:
>
> SmartControl 機能を設定するには、NetScaler VPX インスタンスにプレミアムライセンスを適用する必要があります。

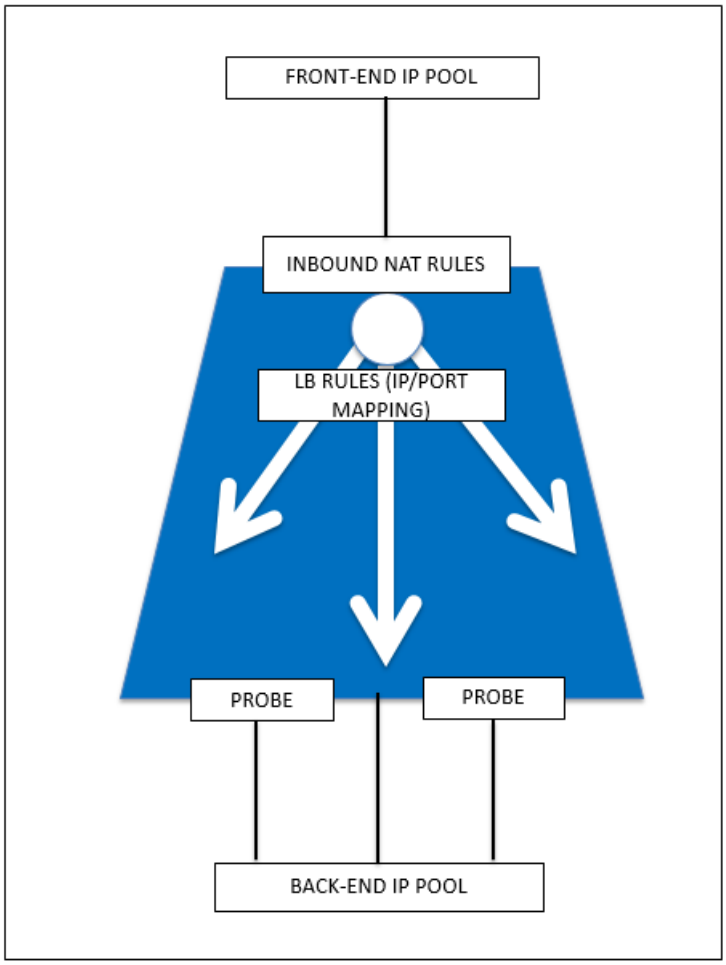
Azure の用語

August 15, 2023

NetScaler VPX Azure のドキュメントで使用されている Azure 用語の一部を以下に示します。

1. Azure ロードバランサー—Azure ロードバランサーは、ネットワーク内のコンピューター間で着信トラフィックを分散するリソースです。トラフィックは、ロードバランサーセット内に定義された仮想マシンに分配されます。ロードバランサーには、外部ロードバランサー、インターネットに接続するロードバランサー、または内部ロードバランサーがあります。
2. Azure Resource Manager (ARM) —ARM は Azure のサービスの新しい管理フレームワークです。Azure Load Balancer は、ARM ベースの API およびツールを使用して管理されます。
3. バックエンドアドレスプール—負荷が分散される仮想マシンの NIC (NIC) に関連付けられた IP アドレスです。
4. BLOB-バイナリラージオブジェクト—Azure ストレージに格納できるファイルまたはイメージのようなバイナリオブジェクト。
5. フロントエンド IP 構成—Azure ロードバランサーには、仮想 IP (VIP) とも呼ばれるフロントエンド IP アドレスを 1 つ以上含めることができます。これらの IP アドレスがトラフィックの入口として使用されます。
6. インスタンスレベルのパブリック IP (ILPIP) —ILPIP は、仮想マシンやロールインスタンスが存在するクラウドサービスではなく、仮想マシンまたはロールインスタンスに直接割り当てることができるパブリック IP アドレスです。これは、クラウドサービスに割り当てられた VIP（仮想 IP）に代わるものではありません。これは、仮想マシンまたはロールインスタンスに直接接続するために使用できる追加の IP アドレスです。

   注：以前には、ILPIP は PIP（「パブリック IP」の略）と呼ばれていました。
7. インバウンド NAT ルール—ロードバランサーのパブリックポートをバックエンドアドレスプール内の特定の仮想マシンのポートにマッピングするルールが含まれます。
8. IP-Config：個々の NIC に関連付けられた IP アドレスのペア（パブリック IP とプライベート IP）として定義できます。IP-Config では、パブリック IP アドレスが NULL の場合があります。各 NIC には、最大 255 までの IP 構成を関連付けることができます。
9. 負荷分散ルール：特定のフロントエンド IP とポートの組み合わせを、バックエンド IP アドレスとポートの組み合わせのセットにマップする規則プロパティ。ロードバランサーリソースの単一の定義を使用して複数のロードバランサー規則を定義でき、その各規則は、フロントエンド IP およびポートと、仮想マシンに関連付けられたバックエンド IP およびポートの組み合わせを示します。



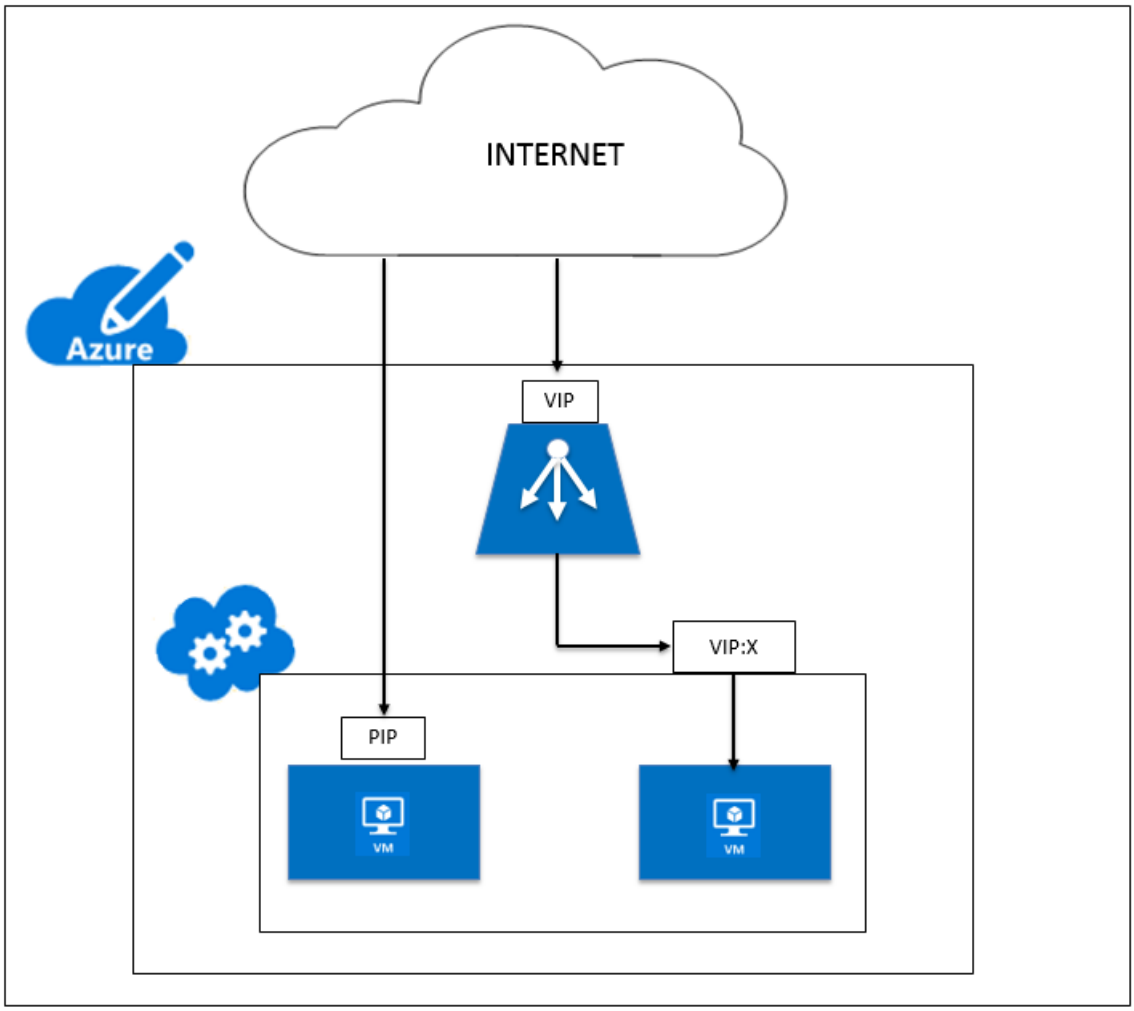
10. ネットワークセキュリティグループ—仮想ネットワーク内の仮想マシンインスタンスへのネットワークトラフィックを許可または拒否するアクセス制御リスト (ACL) ルールのリストが含まれます。NSG は、サブネット、またはそのサブネット内の個々の仮想マシンインスタンスに関連付けることができます。ネットワークセキュリティグループがサブネットに関連付けられている場合、ACL ルールはそのサブネット内のすべての仮想マシンインスタンスに適用されます。さらに、ネットワークセキュリティグループをその仮想マシンに直接関連付けることで、個々の仮想マシンへのトラフィックをさらに制限できます。

11. プライベート IP アドレス—Azure 仮想ネットワーク内の通信に使用され、VPN Gateway を使用してネットワークを Azure に拡張するときのオンプレミスネットワークで使用されます。プライベート IP アドレスを使用すると、Azure リソースは、VPN ゲートウェイまたは ExpressRoute 回路を経由して、インターネットで到達できる IP アドレスを使用せずに、仮想ネットワークまたはオンプレミスネットワーク内の他のリソースと通信できます。Azure Resource Manager 展開モデルでは、プライベート IP アドレスは次の種類の Azure リソースに関連付けられます - 仮想マシン、内部ロードバランサー（ILB)、およびアプリケーションゲートウェイ。

12. プローブ—これには、バックエンドアドレスプール内の仮想マシンインスタンスの可用性をチェックするために使用されるヘルスプローブが含まれます。個別の仮想マシンが一定時間ヘルスプローブに応答しない場合、それはトラフィック供用から除外されます。プローブを使用すると、仮想インスタンスのヘルスを追跡できま

す。ヘルスプローブが失敗した場合、仮想インスタンスはローテーションから自動的に除外されます。

13. パブリック IP アドレス (PIP) —PIP は、Azure のパブリック向けサービスを含むインターネットとの通信に使用され、仮想マシン、インターネット向けロードバランサー、VPN ゲートウェイ、およびアプリケーションゲートウェイに関連付けられます。
14. リージョン-国境を越えず、1 つ以上のデータセンターを含む地理内のエリア。価格設定、地域サービスおよびタイプは、リージョンレベルで公開されます。リージョンは通常、（最大で数百マイル離れた）別のリージョンと対にされ、リージョンペアを形成します。障害回復シナリオおよび高可用性シナリオでは、リージョンペアをメカニズムとして使用できます。また、一般に場所とも呼ばれます。
15. リソースグループ-リソースマネージャーのコンテナには、アプリケーションの関連リソースが格納されます。リソースグループには、アプリケーションのリソースをすべて含めることも、論理的にグループにまとめられたリソースだけを含めることもできます。
16. ストレージアカウント—Azure ストレージアカウントを使用すると、Azure Storage の Azure BLOB、キュー、テーブル、およびファイルサービスにアクセスできます。ストレージアカウントは、Azure ストレージデータオブジェクトに一意の名前空間を提供します。
17. 仮想マシン—オペレーティングシステムを実行する物理コンピュータのソフトウェア実装。同じハードウェア上で複数の仮想マシンを同時に実行できます。Azure には、いろいろなサイズの仮想マシンが用意されています。
18. 仮想ネットワーク-Azure 仮想ネットワークは、クラウド内の独自のネットワークを表します。それはサブスクリプション専用の Azure クラウドの論理的隔離です。IP アドレスブロック、DNS 設定、セキュリティポリシー、およびこのネットワーク内のルートテーブルを全面的に制御できます。さらに VNet のサブネットに分割したり、Azure IaaS 仮想マシンおよびクラウドサービス（PaaS ロールインスタンス）を起動したりすることもできます。また、Azure で利用できる接続性オプションの 1 つを使用して、仮想ネットワークをオンプレミスネットワークに接続できます。本質的には、Azure が提供するエンタープライズスケールの利点を持つ IP アドレスブロック上の全面的なコントロールを使用して、ネットワークを Azure に拡張できます。



Microsoft Azure 上の NetScaler ADC VPX インスタンスのネットワークアーキテクチャ

August 15, 2023

Azure Resource Manager (ARM) では、NetScaler VPX 仮想マシン (VM) は仮想ネットワークに存在します。仮想ネットワークの特定のサブネットに単一のネットワークインターフェイスを作成し、VPX インスタンスに接続できます。ネットワークセキュリティグループを使用して、Azure 仮想ネットワークの VPX インスタンスとの間のネットワークトラフィックをフィルタリングできます。ネットワークセキュリティグループには、VPX インスタンスへのインバウンドネットワークトラフィックまたは VPX インスタンスからのアウトバウンドネットワークトラフィックを許可または拒否するセキュリティルールが含まれています。詳細については、「セキュリティグループ」を参照してください。

ネットワークセキュリティグループは、NetScaler VPX インスタンスへの要求をフィルタリングし、VPX インスタ

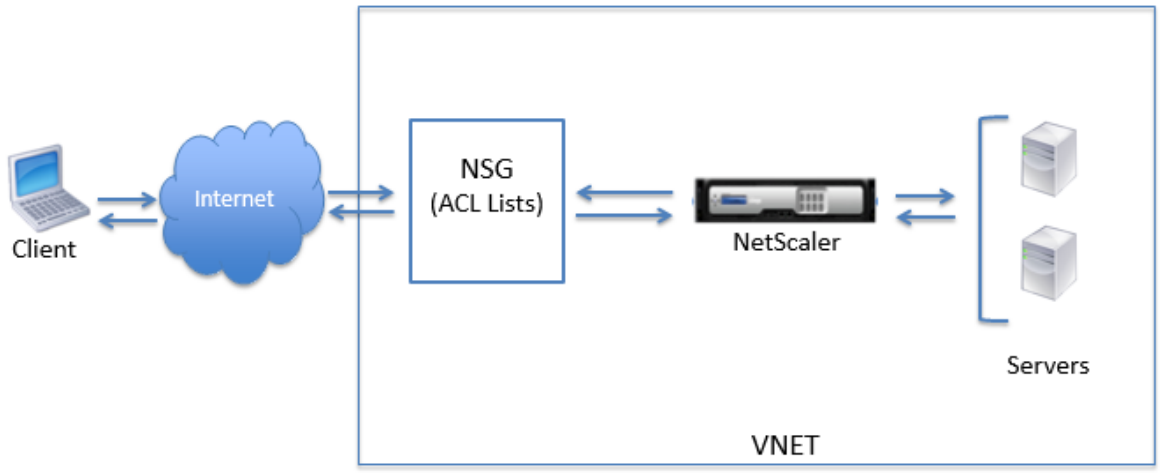
ンスはそれらをサーバーに送信します。サーバーからの応答は、逆の順序で同じパスをたどります。ネットワークセキュリティグループは、単一の VPX VM をフィルタリングするように構成することも、サブネットと仮想ネットワークを使用して複数の VPX インスタンスを展開する場合のトラフィックをフィルタリングすることもできます。

NIC には、ネットワーク構成の詳細（仮想ネットワーク、サブネット、内部 IP アドレス、パブリック IP アドレスなど）が含まれます。

ARM では、単一の NIC と 1 つの IP アドレスでデプロイされた仮想マシンにアクセスするために使用される、次の IP アドレスを知っておくとよいでしょう。

- パブリック IP（PIP）アドレスは、NetScaler VM の仮想 NIC 上で直接構成されるインターネット側 IP アドレスです。これにより、外部ネットワークから VM に直接アクセスできます。
- NetScaler IP（NSIP とも呼ばれる）アドレスは、仮想マシン上で構成された内部 IP アドレスです。これはルーティング不可能です。
- 仮想 IP アドレス（VIP）は、NSIP とポート番号を使用して構成されます。クライアントは PIP アドレスから NetScaler サービスにアクセスし、要求が NetScaler VPX VM または Azure ロードバランサーの NIC に到達すると、VIP が内部 IP（NSIP）および内部ポート番号に変換されます。
- 内部 IP アドレスは、仮想ネットワークのアドレス空間プールにある、VM のプライベート内部 IP アドレスです。この IP アドレスは、外部ネットワークから到達できません。この IP アドレスは、静的に設定しない限り、デフォルトで動的です。インターネットからのトラフィックは、ネットワークセキュリティグループで作成されたルールに従って、このアドレスにルーティングされます。ネットワークセキュリティグループは NIC と統合して、仮想マシンで設定されたサービスに応じて、適切なタイプのトラフィックを NIC の適切なポートに選択的に送信します。

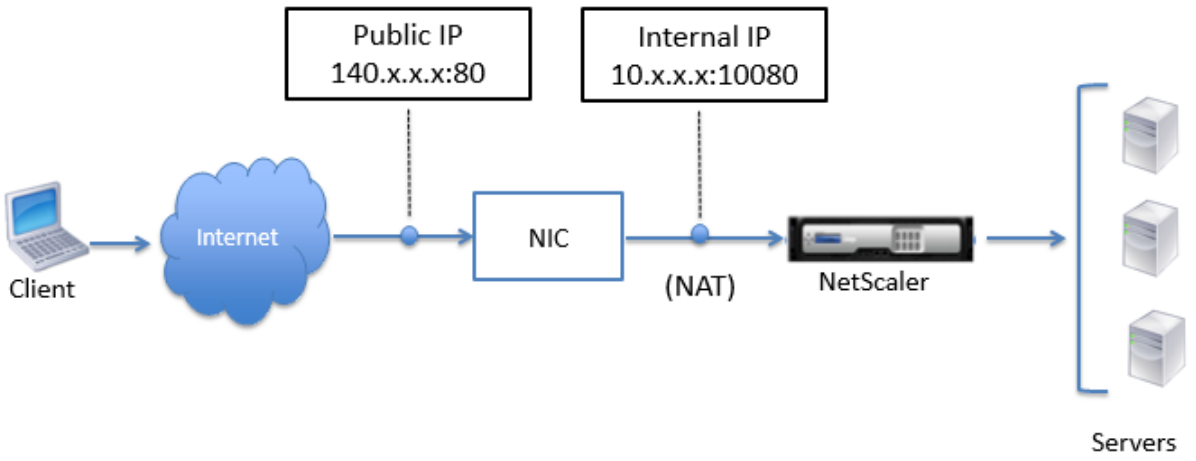
以下の図は、ARM でプロビジョニングされた NetScaler VPX インスタンスを介したクライアントからサーバーへのトラフィックフローを示しています。



ネットワークアドレス変換によるトラフィックフロー

NetScaler VPX インスタンス（インスタンスレベル）のパブリック IP（PIP）アドレスをリクエストすることもできます。この直接 PIP を VM レベルで使用する場合、ネットワークトラフィックを傍受する受信および送信規則を定義する必要はありません。インターネットからの着信要求が VM で直接受信されます。Azure はネットワークアドレス変換 (NAT) を実行し、トラフィックを VPX インスタンスの内部 IP アドレスに転送します。

以下の図は、Azure がネットワークアドレス変換を実行し、NetScaler 内部 IP アドレスをマップする方法を示しています。



この例では、ネットワークセキュリティグループに割り当てられたパブリック IP は 140.x.x.x で、内部 IP アドレスは 10.x.x.x です。インバウンドルールとアウトバウンドルールが定義されている場合、パブリック HTTP ポート 80 はクライアントリクエストを受信するポートとして定義され、対応するプライベートポート 10080 は NetScaler ADC VPX インスタンスがリッスンするポートとして定義されます。クライアント要求はパブリック IP アドレス 140.x.x.x で受信されます。Azure がネットワークアドレス変換を実行して、PIP を内部 IP アドレス 10.x.x.x（ポート 10080）にマップし、クライアント要求を転送します。

> 注
>
> 高可用性における NetScaler VPX VM は、自身に定義された受信規則により負荷分散トラフィックを制御する、外部または内部のロードバランサーによって制御されます。外部トラフィックは最初にこれらのロードバランサーによって傍受され、トラフィックはロードバランサーで定義されたバックエンドプール、NAT ルール、ヘルスプローブを含むロードバランシングルールに従って転送されます。

ポートの使用に関する注意事項

NetScaler VPX インスタンスの作成中または仮想マシンのプロビジョニング後に、ネットワークセキュリティグループでより多くのインバウンドルールとアウトバウンドルールを構成できます。各受信および送信規則は、パブリックポートおよびプライベートポートに関連付けられています。

ネットワークセキュリティグループルールを設定する前に、使用できるポート番号に関する次のガイドラインに注意してください。

1. NetScaler VPX インスタンスは次のポートを予約します。インターネットからのリクエストにパブリック IP アドレスを使用する場合、これらをプライベートポートとして定義することはできません。

   ポート 21, 22, 80, 443, 8080, 67, 161, 179, 500, 520, 3003, 3008, 3009, 3010, 3011, 4001, 5061, 9000, 7000.

   ただし、VIP などのインターネットに直接接続するサービスで標準ポート（ポート 443 など）を使用する場合は、ネットワークセキュリティグループを使用してポートマッピングを作成する必要があります。これにより、標準ポートがこの VIP サービス用に NetScaler で構成された別のポートにマップされます。

   たとえば、VIP サービスが VPX インスタンスのポート 8443 で実行されているが、パブリックポート 443 にマッピングされているとします。したがって、ユーザーがパブリック IP を介してポート 443 にアクセスすると、要求はプライベートポート 8443 に送信されます。

2. パブリック IP アドレスでは、ポートマッピングが動的に解放される、パッシブ FTP や ALG のようなプロトコルをサポートしていません。

3. 高可用性は、Azure ロードバランサーで構成された PIP ではなく、VPX インスタンスに関連付けられたパブリック IP アドレス（PIP）を使用するトラフィックでは機能しません。

> 注
>
> Azure Resource Manager では、NetScaler VPX インスタンスにはパブリック IP アドレス（PIP）と内部 IP アドレスの 2 つの IP アドレスが関連付けられています。外部トラフィックは PIP に接続しますが、内部 IP アドレスまたは NSIP はルーティング不可能です。VPX で VIP を設定するには、内部 IP アドレスと使用可能な任意の空きポートを使用します。VIP の構成に PIP を使用してはいけません。

NetScaler VPX スタンドアロンインスタンスを構成する

October 25, 2023

仮想マシンを作成して他のリソースを構成することにより、スタンドアロンモードで Azure Resource Manager（ARM）ポータルで単一の NetScaler VPX インスタンスをプロビジョニングできます。

はじめに

次の項目があることを確認します。

- Microsoft Azure ユーザーアカウント
- Microsoft Azure Resource Manager へのアクセス

- Microsoft Azure SDK
- Microsoft Azure PowerShell

Microsoft Azure ポータルのページで 、ユーザー名とパスワードを指定して Azure Resource Manager ポータルにログオンします。

> 注
>
> ARM ポータルで、1 つのペインでオプションをクリックすると、右側に新しいペインが開きます。ペイン間を移動してデバイスを構成します。

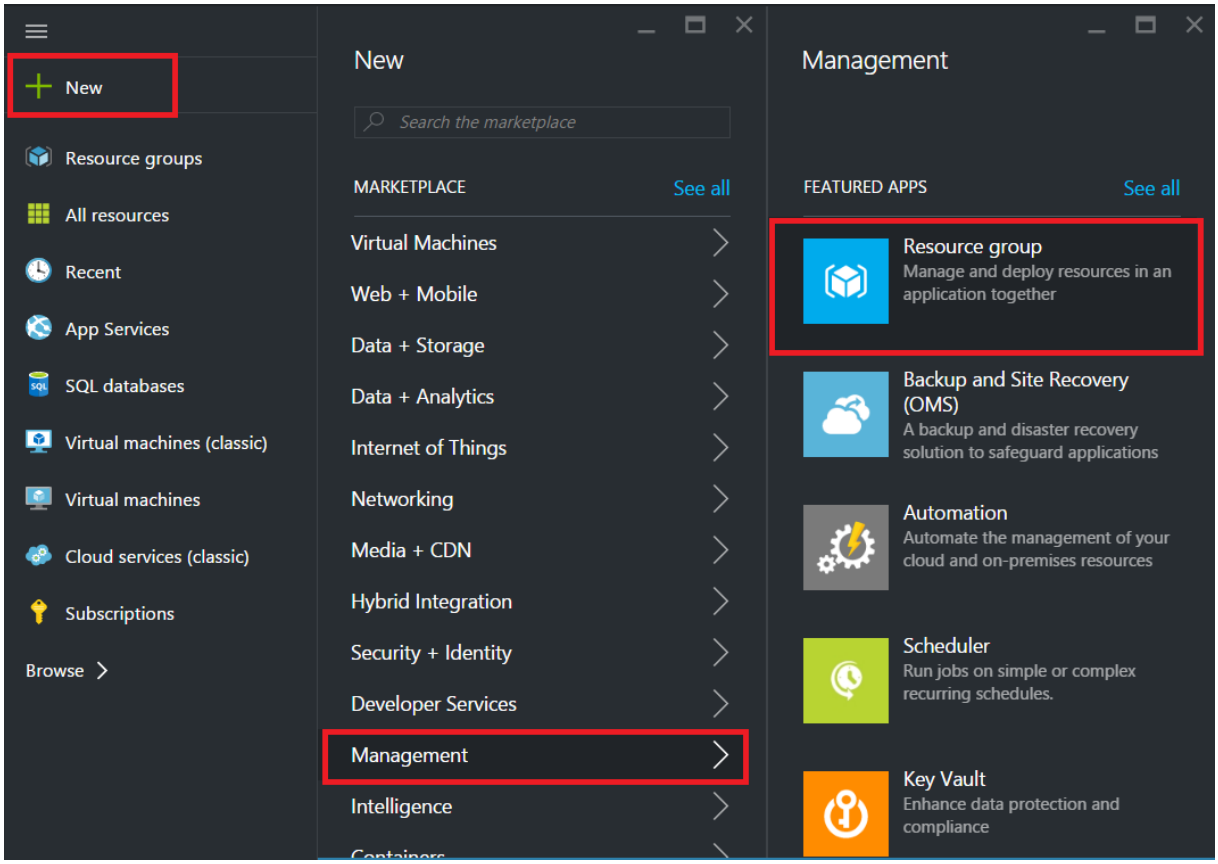
設定手順の概要

1. リソースグループの構成
2. ネットワークセキュリティグループの構成
3. 仮想ネットワークインターフェイスとそのサブネットの構成
4. ストレージアカウントの構成
5. 可用性セットの構成
6. NetScaler VPX インスタンスを構成します。

リソースグループの構成

すべてのリソースのコンテナとなる新しいリソースグループを作成します。リソースグループを使用して、リソースをグループとして展開、管理、および監視します。

1. 新規 > 管理 > リソースグループをクリックします。
2. リソースグループペインで、次の詳細を入力します。
    - リソースグループ名
    - リソースグループの場所
3. ［作成］をクリックします。



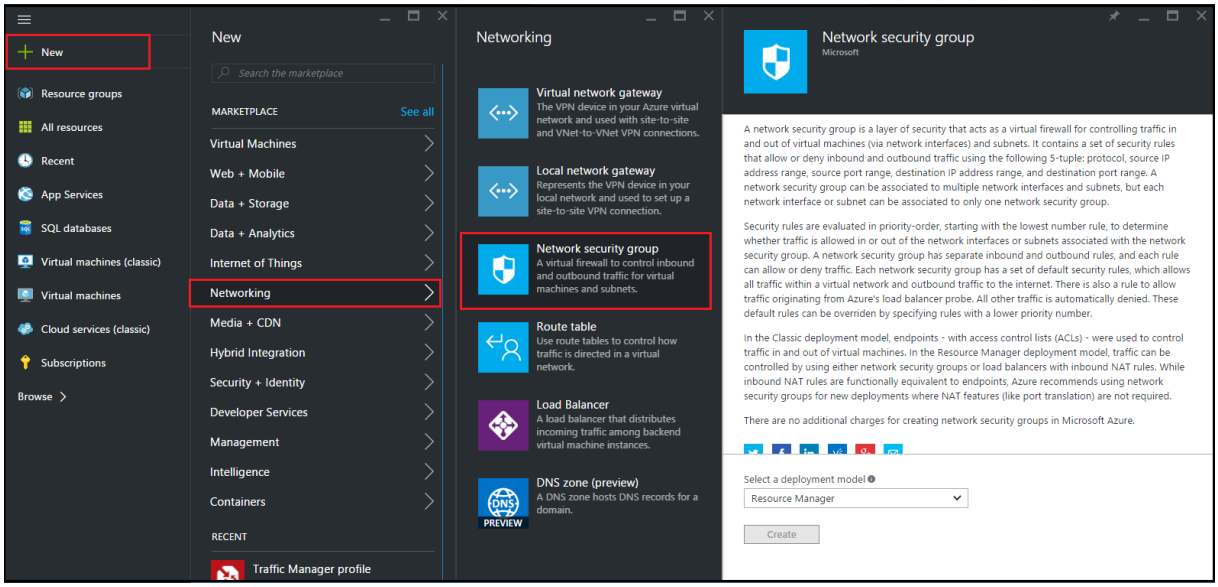
ネットワークセキュリティグループの構成

仮想ネットワーク内の着信トラフィックと発信トラフィックを制御するインバウンドルールとアウトバウンドルールを割り当てるネットワークセキュリティグループを作成します。ネットワークセキュリティグループを使用すると、単一の仮想マシンのセキュリティルールを定義したり、仮想ネットワークサブネットのセキュリティルールを定義したりできます。

1. [新規] > [ネットワーク] > [ネットワークセキュリティグループ] をクリックします。
2. [ネットワークセキュリティグループの作成] ペインで、次の詳細を入力し、[作成] をクリックします。

    - Name - セキュリティグループの名前を入力します
    - Resource group - ボックスの一覧からリソースグループを選択します

> 注
>
> 正しい場所を選択していることを確認します。場所が異なれば、ボックスの一覧に表示されるリソースの一覧も異なります。

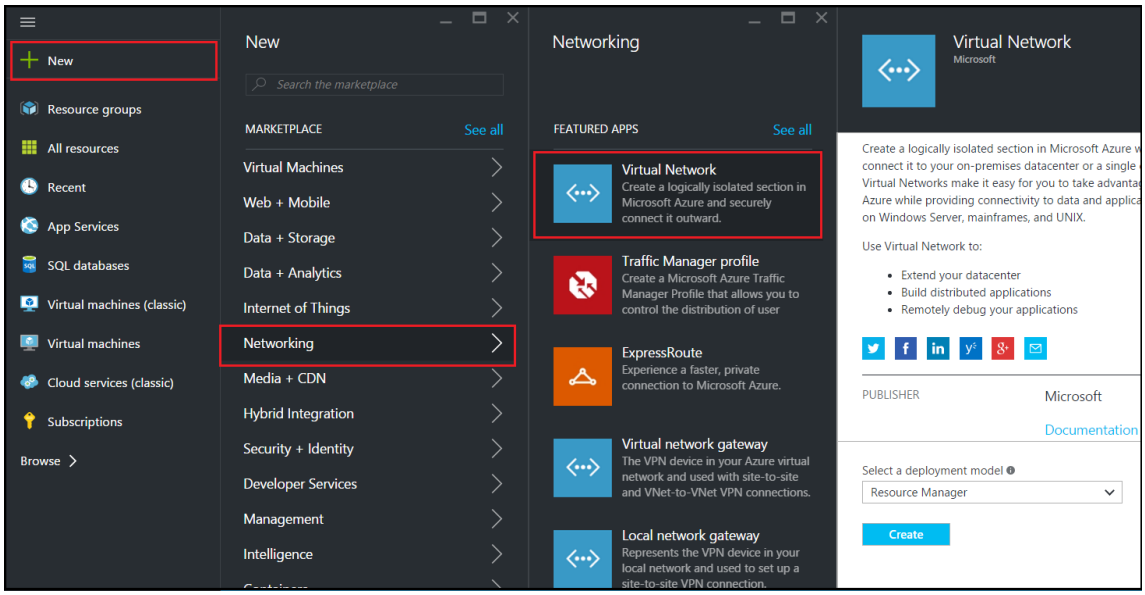


仮想ネットワークとサブネットを構成する

ARM の仮想ネットワークは、サービスのセキュリティを強化し、隔離するものです。同じ仮想ネットワークに属する VM およびサービスは、互いにアクセスできます。

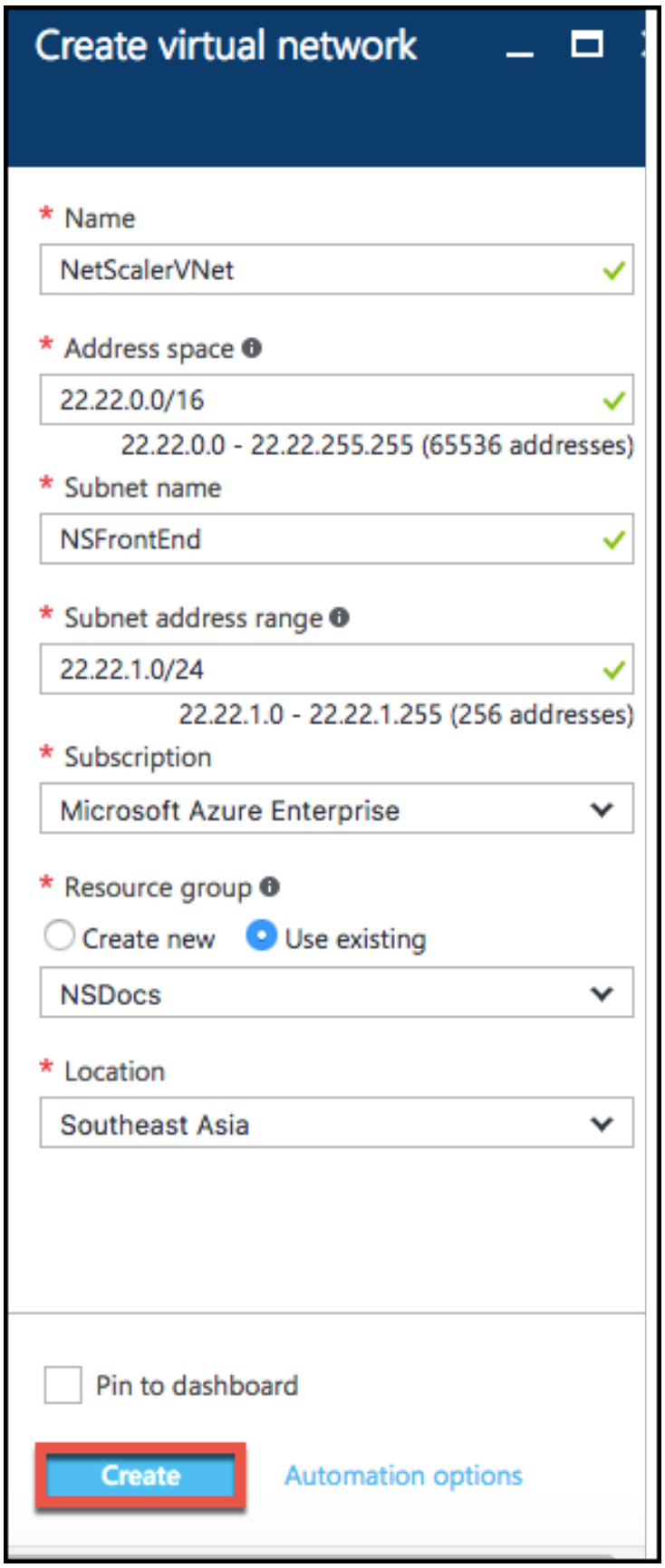
仮想ネットワークとサブネットを作成する手順は次のとおりです。

1. 新規 > ネットワーク > 仮想ネットワークをクリックします。
2. [仮想ネットワーク] ペインで、展開モードが [リソースマネージャ] であることを確認し、[作成] をクリックします。



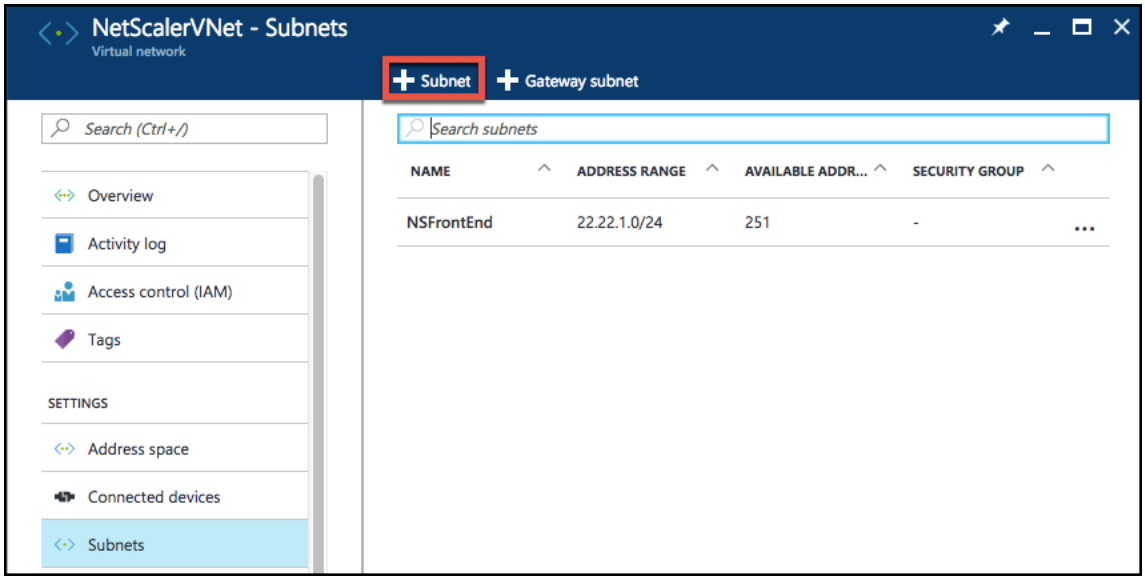
3. 仮想ネットワークの作成 ウィンドウで、次の値を入力し、作成 をクリックします。

- 仮想ネットワークの名前
- Address space - 仮想ネットワークの予約済 IP アドレスブロックを入力します
- サブネット-最初のサブネットの名前を入力します (2 つ目のサブネットはこのステップの後半で作成します)
- Subnet address range - サブネットの予約済 IP アドレスブロックを入力します
- Resource group - ボックスの一覧から以前に作成したリソースグループを選択します。



2 番目のサブネットを設定する

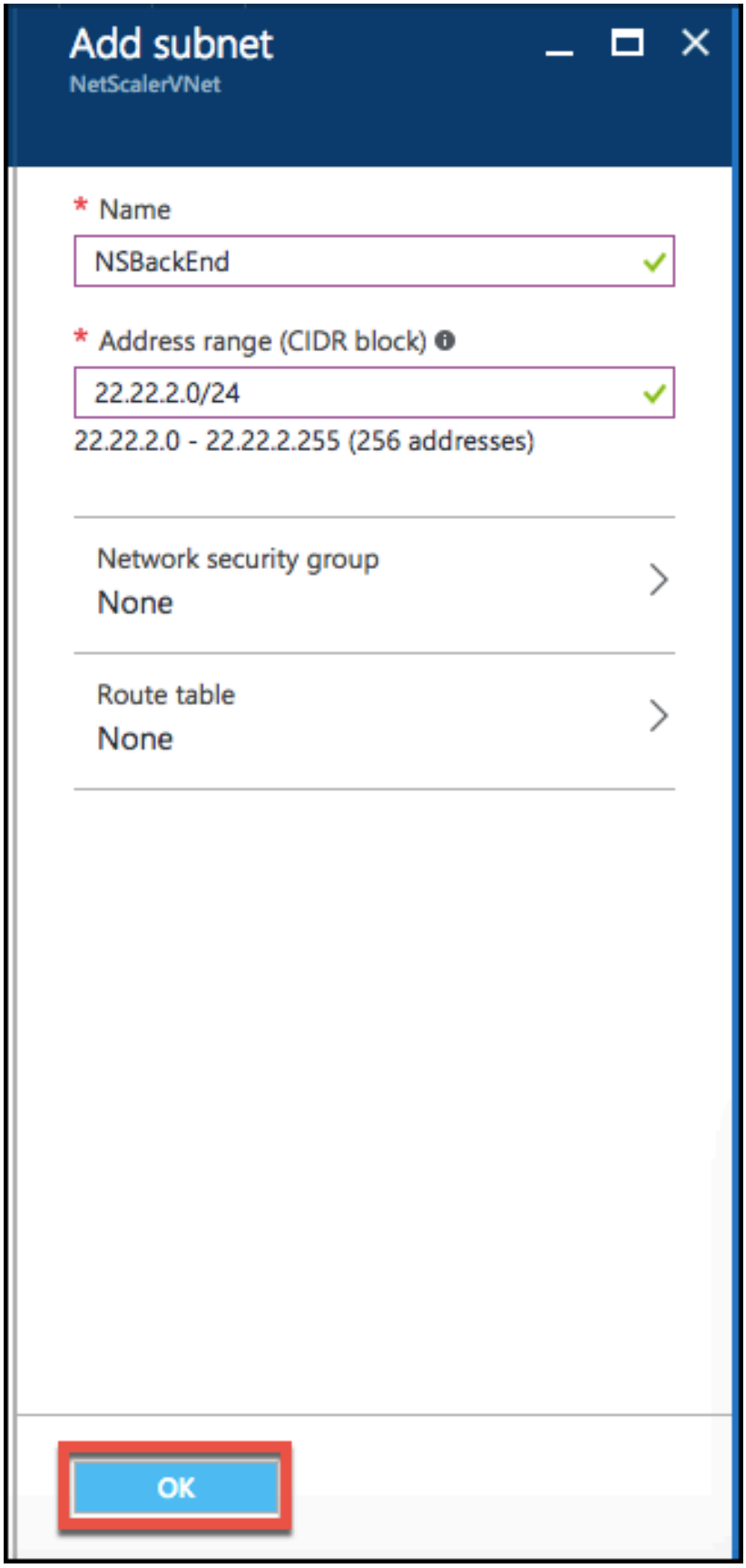
1. [すべてのリソース] ペインから新しく作成した仮想ネットワークを選択し、[設定] ペインで [サブネット] をクリックします。



2. **+Subnet** をクリックし、次の詳細を入力して 2 番目のサブネットを作成します。

   - 2 番目のサブネットの名前
   - Address range - サブネットの予約済 IP アドレスブロックを入力します
   - ネットワークセキュリティグループ-ドロップダウンリストからネットワークセキュリティグループを選択します。

3. ［作成］をクリックします。



ストレージアカウントの構成

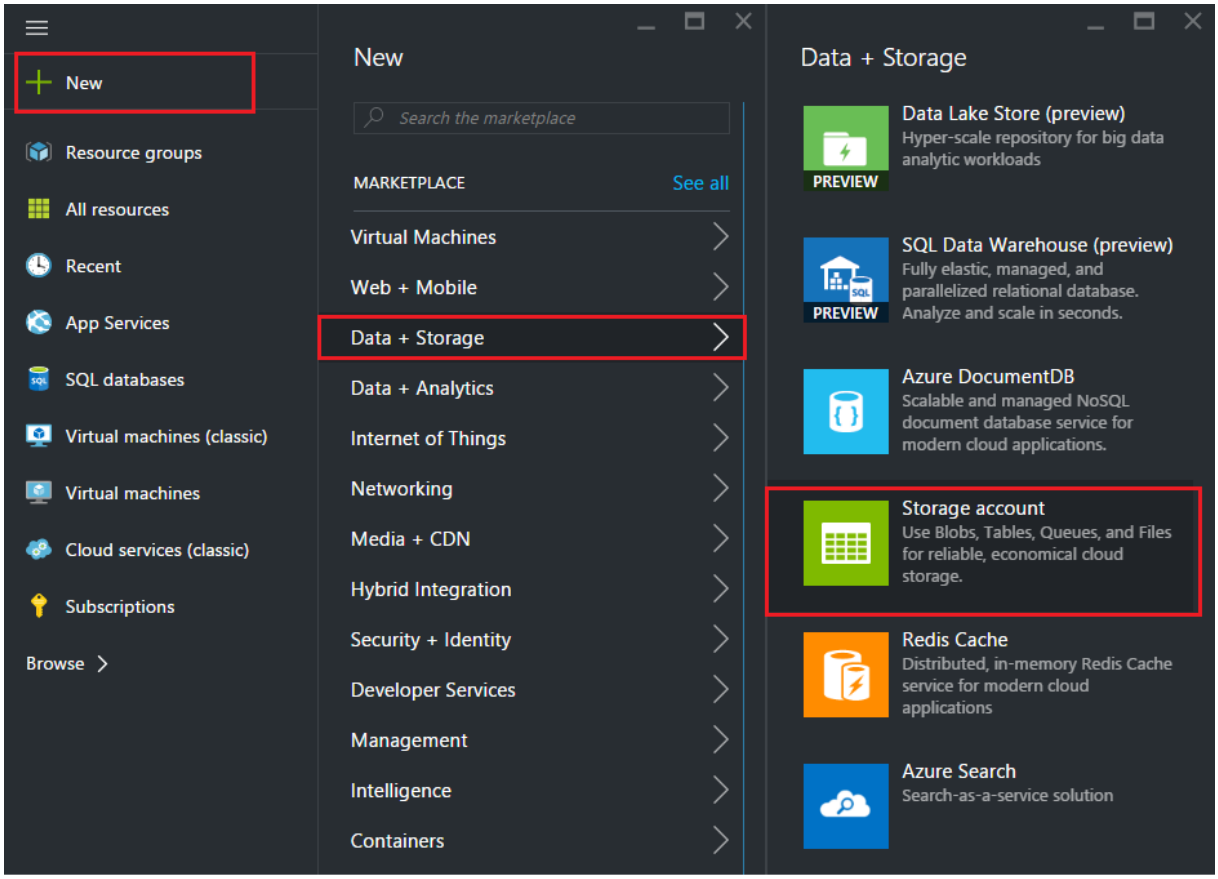
ARM IaaS インフラストラクチャストレージには、BLOB、表、キューおよびファイルの形式でデータを保存できるすべてのサービスが含まれます。ARM では、これらの形式のストレージデータを使用してアプリケーションを作成することもできます。

ストレージアカウントを作成してすべてのデータを保存します。

1. [**+** 新規] > [データ**] + [**ストレージ] > [ストレージアカウント] をクリックします。
2. ストレージアカウントの作成ペインで、次の詳細を入力します。

    - アカウントの名前
    - デプロイモード- 必ずリソースマネージャーを選択してください
    - アカウントの種類-ドロップダウンリストから「汎用」を選択します
    - レプリケーション-ドロップダウンリストから「ローカル冗長ストレージ」を選択します
    - Resource group - ボックスの一覧から新しく作成したリソースグループを選択します

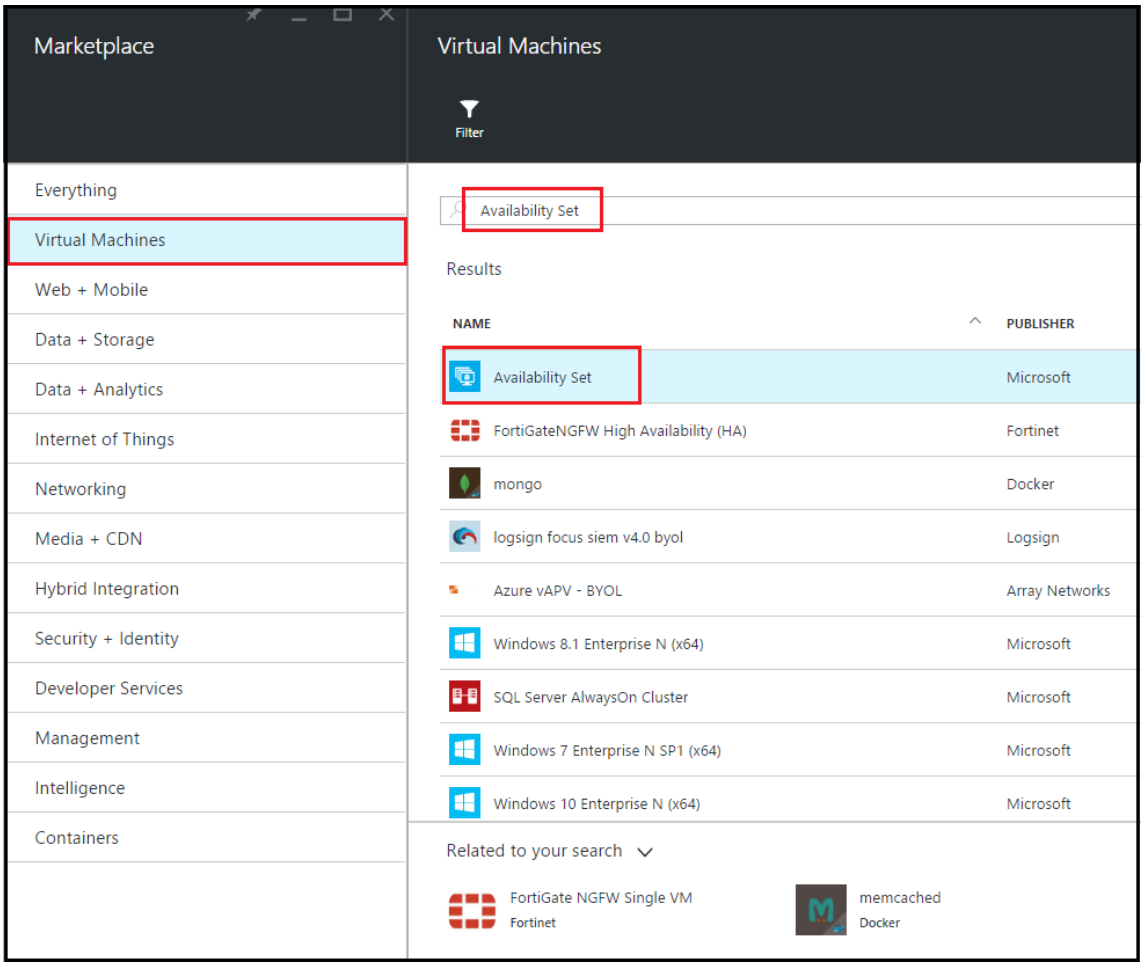
3. ［作成］をクリックします。



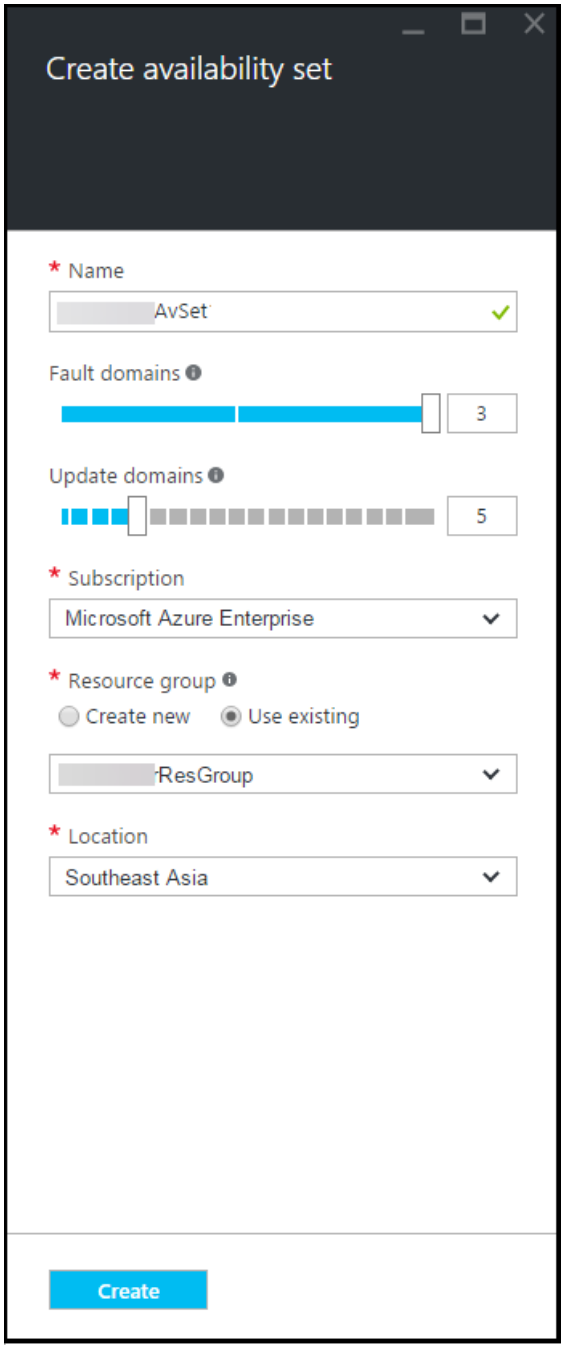
可用性セットの構成

可用性セットにより、計画的または計画外のメンテナンスが発生した場合でも、少なくとも 1 つの VM が稼働し続けることが保証されます。同じ可用性セットに属する 2 台以上の VM は、異なるフォールトドメインに配置されて、サービスの冗長性を確保します。

1. [+ 新規] をクリックします。
2. MARKETPLACE ペインで「すべて表示」をクリックし、「仮想マシン」をクリックします。
3. 可用性セットを検索し、表示されたリストから [可用性セット エンティティ] を選択します。



4. [作成] をクリックし、 [可用性セットの作成] ウィンドウで、次の詳細を入力します。
   - セットの名前
   - Resource group - ボックスの一覧から新しく作成したリソースグループを選択します
5. [作成] をクリックします。



NetScaler VPX インスタンスの構成

仮想ネットワークに NetScaler VPX のインスタンスを作成します。Azure Marketplace から NetScaler VPX イメージを取得し、Azure Resource Manager ポータルを使用して NetScaler VPX インスタンスを作成します。

NetScaler VPX インスタンスの作成を開始する前に、インスタンスが存在する必要なサブネットを持つ仮想ネットワークが作成されていることを確認してください。仮想マシンのプロビジョニング時に仮想ネットワークを作成することもできますが、柔軟性に欠けるため別のサブネットを作成することはできません。仮想ネットワークの作成につ

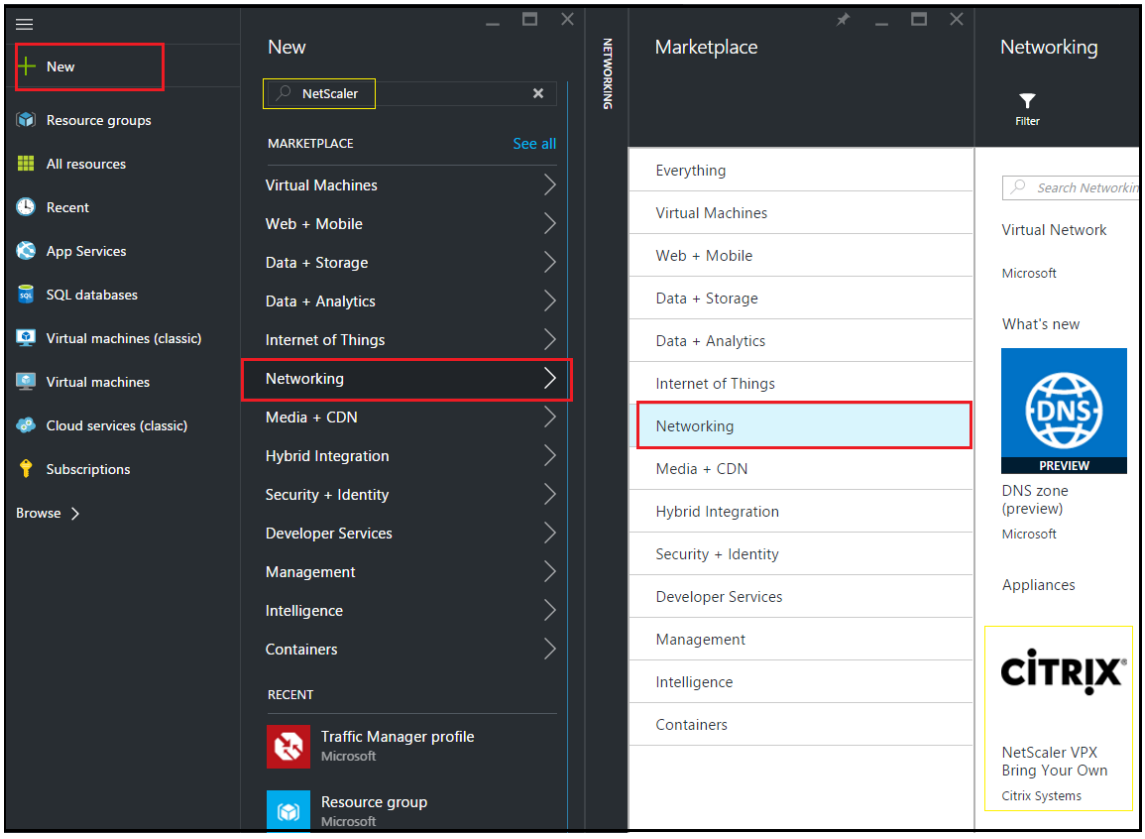
いては、http://azure.microsoft.com/en-us/documentation/articles/create-virtual-network/を参照してください。

オプションで、仮想マシンがインターネットリソースにアクセスできるようにする DNS サーバと VPN 接続を設定します。

> 注
>
> プロビジョニング時にネットワーク情報を利用できるよう、NetScaler VPX VM をプロビジョニングする前に、リソースグループ、ネットワークセキュリティグループ、仮想ネットワークおよび他のエンティティを作成することをお勧めします。

1. [**+** 新規] > [ネットワーク] をクリックします。
2. [すべて表示] をクリックし、[ネットワーク] ペインで [**NetScaler 13.0**] をクリックします。
3. ソフトウェアプランのリストから「**NetScaler 13.0 VPX Bring Your Own License** 」を選択します。

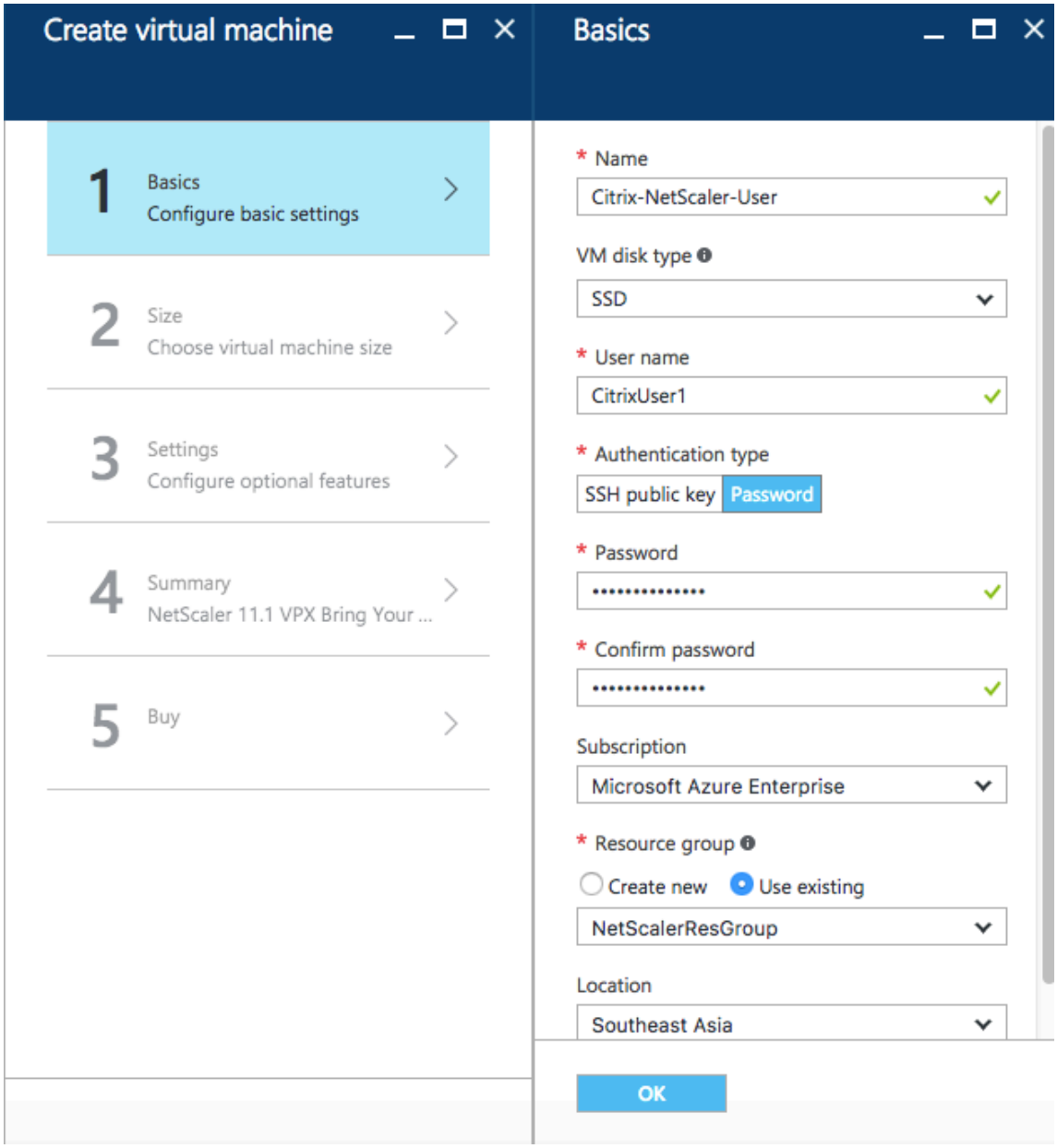
   ARM ポータルでエンティティをすばやく見つける方法として、Azure Marketplace 検索ボックスにエンティティの名前を入力して を押すこともできます <Enter>。検索ボックスに「NetScaler 」と入力して、NetScaler イメージを検索します。



> 注

> 最新のイメージを選択するようにしてください。NetScaler ADC イメージの名前にリリース番号が含まれている場合があります。

4. [**NetScaler VPX** 自分のライセンスを持参] ページのドロップダウンリストから [リソースマネージャー] を選択し、[作成] をクリックします。



5. [仮想マシンの作成] ウィンドウで、各セクションで必要な値を指定して、仮想マシンを作成します。各セクションで「**OK**」をクリックして設定を保存します。

ベーシック:

- Name - NetScaler VPX インスタンスの名前を指定します
- VM disk type - ボックスの一覧から SSD （デフォルト値）または HDD を選択します。

- User name and Password - 作成したリソースグループのリソースにアクセスするためのユーザー名およびパスワードを指定します
- Authentication Type - ［SSH Public Key］または［Password］を選択します
- Resource group - ボックスの一覧から作成したリソースグループを選択します。

ここではリソースグループを作成できますが、Azure Resource Manager で［Resource groups］からリソースグループを作成して、そのグループをボックスの一覧から選択することをお勧めします

> 注
>
> Azure スタック環境では、基本パラメータに加えて、次のパラメータを指定します。
>
> - Azure スタックドメイン
> - Azure スタックテナント (オプション)
> - Azure クライアント (オプション)
> - Azure クライアントシークレット (オプション)

サイズ:

基本設定で選択した仮想マシンのディスクタイプ、SDD、または HDD に応じて、ディスクサイズが表示されます。

- 必要に応じてディスクサイズを選択し、[選択] をクリックします。

設定:

- デフォルトのディスクタイプ（[Standard]）を選択します
- Storage account - ストレージアカウントを選択します
- Virtual network - 仮想ネットワークを選択します
- Subnet - サブネットアドレスを設定します
- Public IP address - IP アドレス割り当ての種類を選択します
- Network security group - 作成したセキュリティグループを選択します。セキュリティグループで、受信規則および送信規則が構成されていることを確認します。
- アベイラビリティセット-ドロップダウンメニューボックスからアベイラビリティセットを選択します

要約:

構成設定が検証され、［Summary］ページに検証の結果が表示されます。検証が失敗すると、［Summary］ページに障害の理由が表示されます。個別のセクションに戻り、必要に応じて変更します。検証に合格したら、「**OK**」をクリックします。

購入:

購入ページでオファーの詳細と法的条件を確認し、「購入」をクリックします。

高可用性導入では、同じ可用性セットおよび同じリソースグループに NetScaler VPX 独立したインスタンスを 2 つ作成して、アクティブ/スタンバイ構成で展開します。

NetScaler VPX スタンドアロンインスタンスに複数の IP アドレスを構成する

August 16, 2023

このセクションでは、Azure Resource Manager（ARM）で複数の IP アドレスを使用してスタンドアロン NetScaler ADC VPX インスタンスを構成する方法について説明します。VPX インスタンスには 1 つ以上の NIC を接続でき、各 NIC には 1 つ以上の静的または動的なパブリックおよびプライベート IP アドレスを割り当てることができます。複数の IP アドレスを NSIP、VIP、SNIP などとして割り当てることができます。

詳細については、Azure のドキュメント「Azure ポータルを使用して仮想マシンに複数の IP アドレスを割り当てる」を参照してください。

PowerShell コマンドを使用する場合は、「PowerShell コマンドを使用してスタンドアロンモードで NetScaler ADC VPX インスタンスの複数の IP アドレスを構成する」を参照してください。

使用例

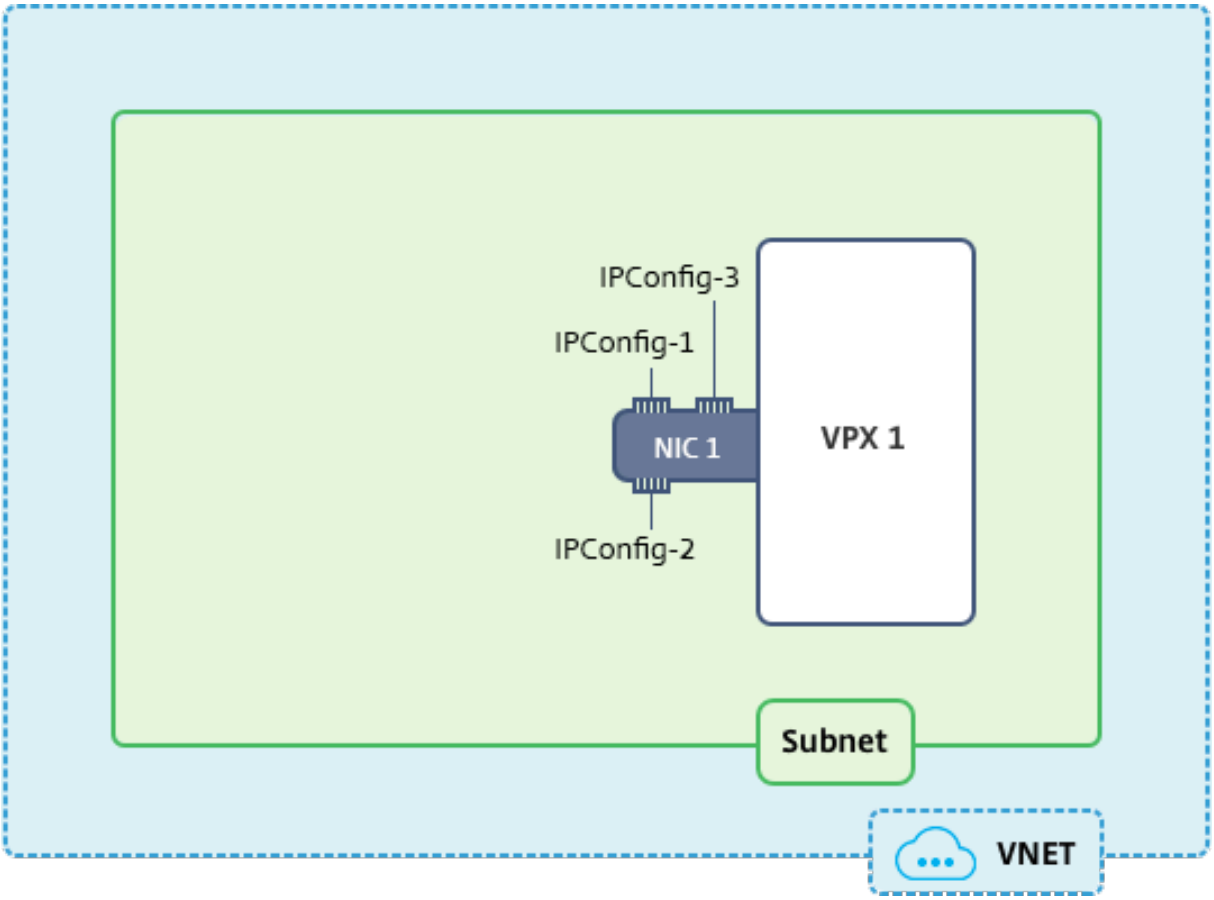
この使用例では、スタンドアロンの NetScaler ADC VPX アプライアンスは、仮想ネットワーク（VNET）に接続された単一の NIC で構成されます。NIC は、表に示すように、3 つの IP 構成（ipconfig）に関連付けられ、各サーバは異なる目的で使用されます。

IP コンフィグ	関連付けられている	目的
ipconfig1	静的パブリック IP アドレス; 静的プライベート IP アドレス	管理トラフィックを提供する
ipconfig2	静的パブリック IP アドレス; 静的プライベートアドレス	クライアント側のトラフィックを提供する
ipconfig3	静的プライベート IP アドレス	バックエンドサーバーと通信する

> 注
>
> `IPConfig-3`はパブリック IP アドレスに関連付けられていません。

図: トポロジ

次の図はこの使用例を視覚的に示しています。



> 注
>
> マルチ NIC、マルチ IP Azure NetScaler VPX 展開では、プライマリ（最初の）NIC のプライマリ（最初）`IPConfig` に関連付けられたプライベート IP が、アプライアンスの管理 NSIP として自動的に追加されます。`IPConfigs`に関連付けられた残りのプライベート IP アドレスは、必要に応じて、`add ns ip` コマンドを使用して VIP または SNIP として VPX インスタンスに追加する必要があります。

はじめに

始める前に、次のリンクに示す手順に従って VPX インスタンスを作成します。

NetScaler VPX スタンドアロンインスタンスを構成する

このユースケースでは、NSDoc0330VM VPX インスタンスが作成されます。

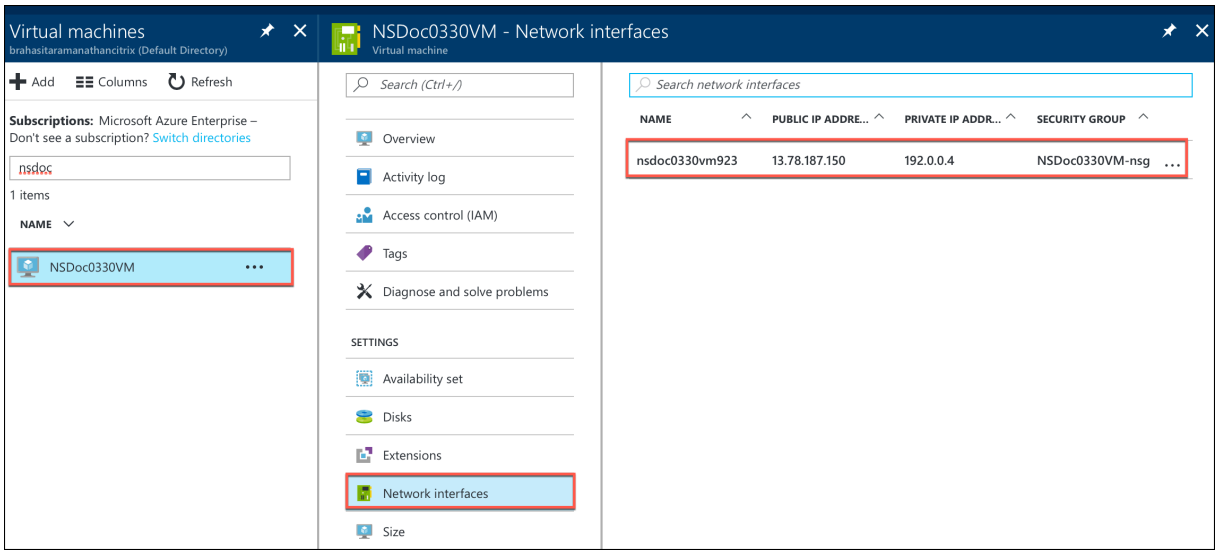
スタンドアロンモードで、**NetScaler VPX** インスタンスに対して複数 **IP** アドレスを構成する手順。

スタンドアロンモードの NetScaler VPX アプライアンスに複数の IP アドレスを構成するには:

1. VM への IP アドレス追加
2. NetScaler が所有する IP アドレスを構成する

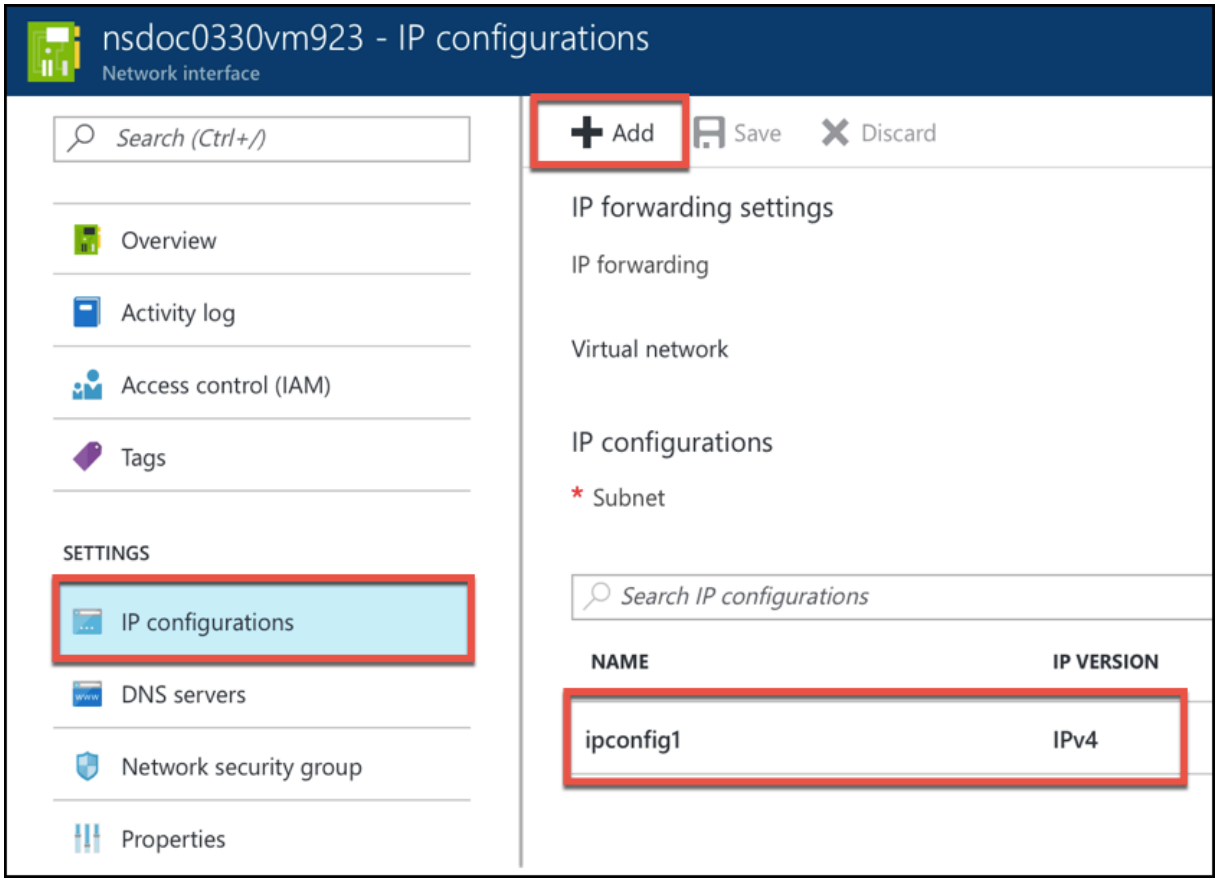
ステップ **1**: VM に IP アドレスを追加する

1. ポータルで [その他のサービス] をクリックし、フィルターボックスに「仮想マシン」と入力し、[仮想マシン] をクリックします。
2. 仮想マシンのブレードで、IP アドレスを追加する仮想マシンをクリックします。表示される仮想マシンブレードの [ネットワーク インターフェイス] をクリックし、ネットワークインターフェイスを選択します。



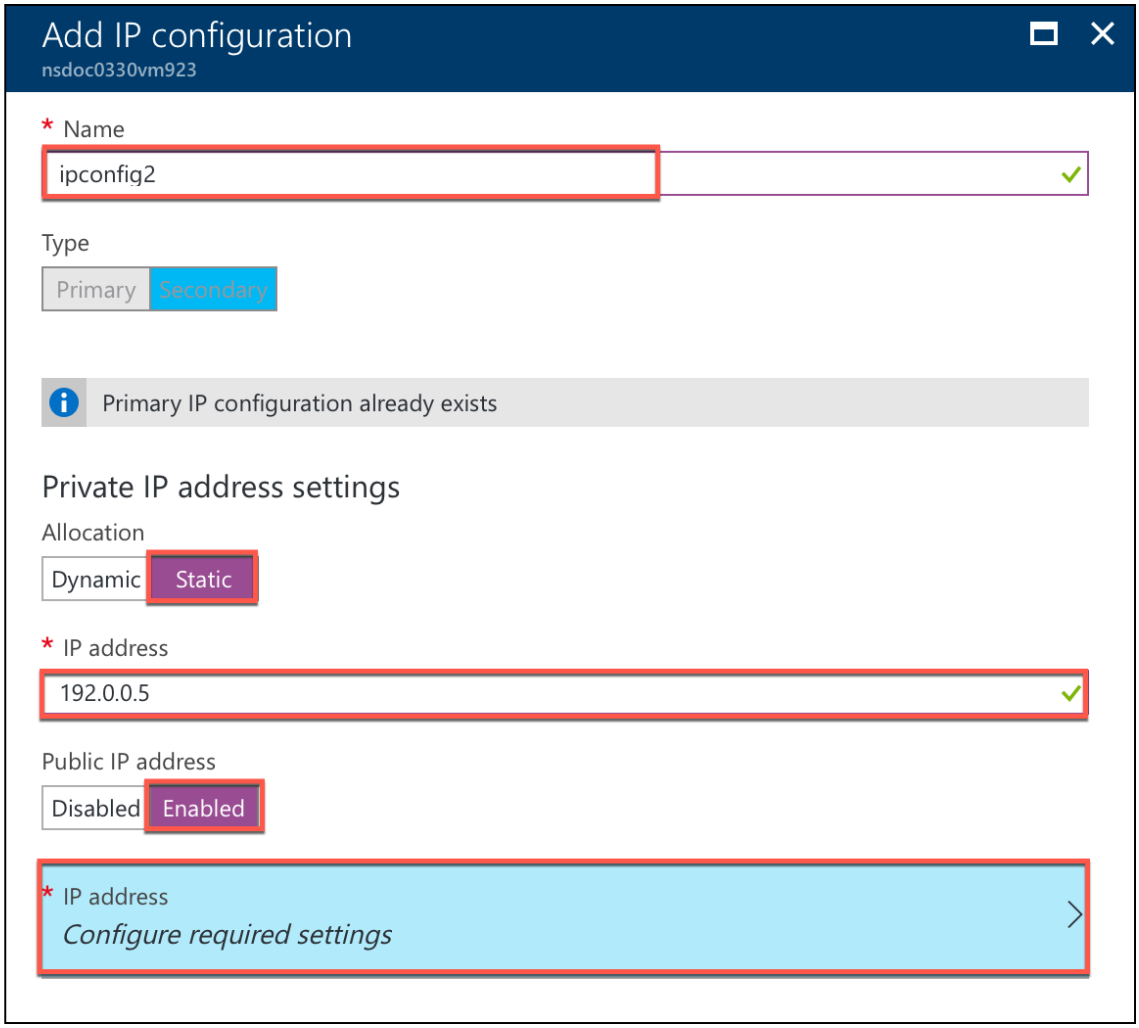
選択した NIC のブレードで、[**IP** 構成] をクリックします。仮想マシンの作成時に割り当てられた既存の IP 構成、**ipconfig1** が表示されます。この使用例では、ipconfig1 に割り当てられている IP アドレスが静的アドレスであることを確認します。次に、さらに 2 つの IP 構成、ipconfig2（VIP）と ipconfig3（SNIP）を作成します。

さらに`ipconfigs`を作成するには、**Add** を作成します。



[**IP** 構成の追加] ウィンドウで、[名前] を入力し、割り当て方法として [静的] を指定し、IP アドレス (このユースケースでは 192.0.0.5) を入力し、[パブリック **IP** アドレス] を有効にします。

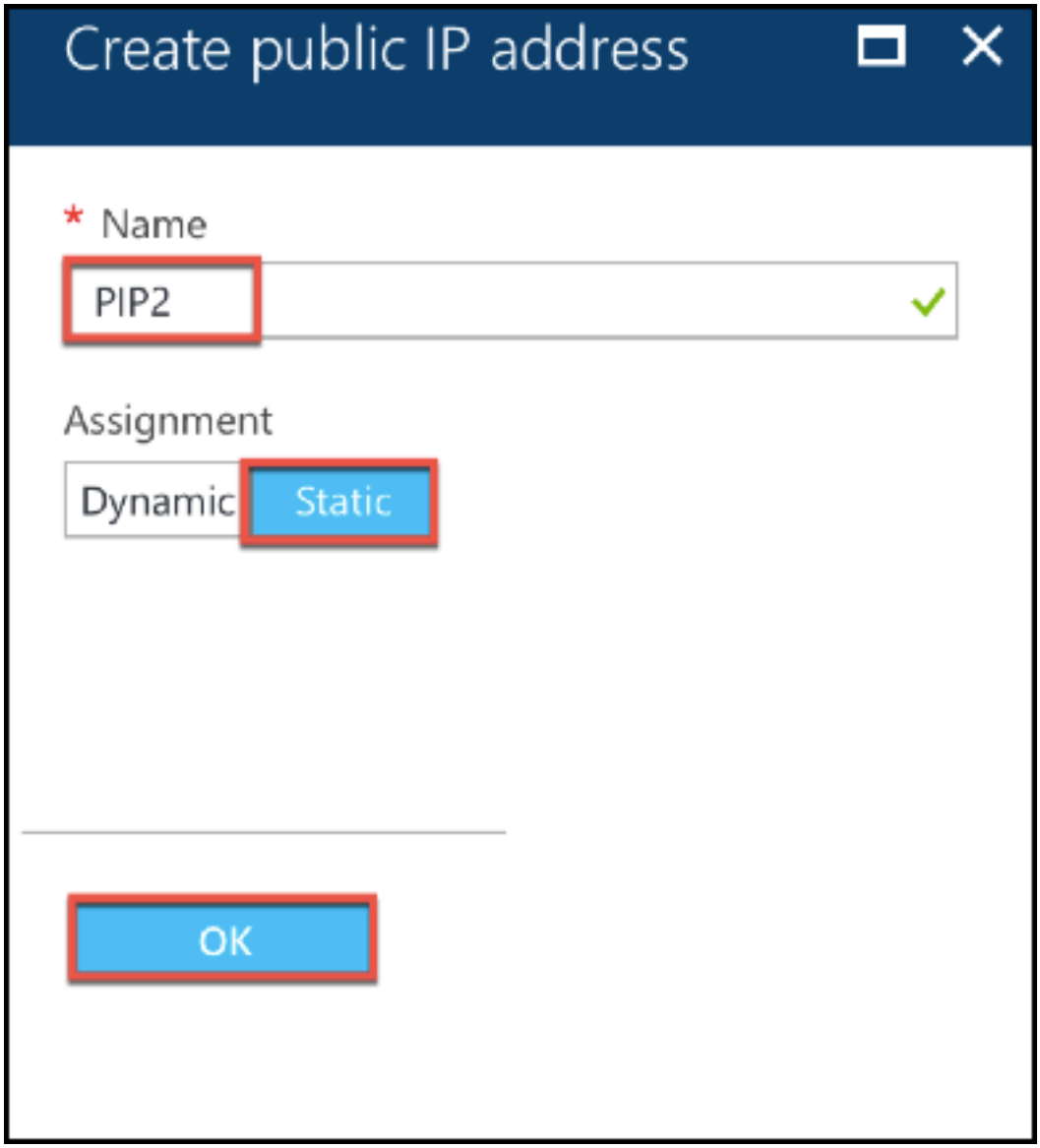
> 注
>
> 静的なプライベート IP アドレスを追加する前に、IP アドレスの可用性をチェックし、その IP アドレスが、NIC の接続先と同じサブネットに属していることを確認します。



次に、[必要な設定の構成] をクリックして ipconfig2 の静的パブリック IP アドレスを作成します。

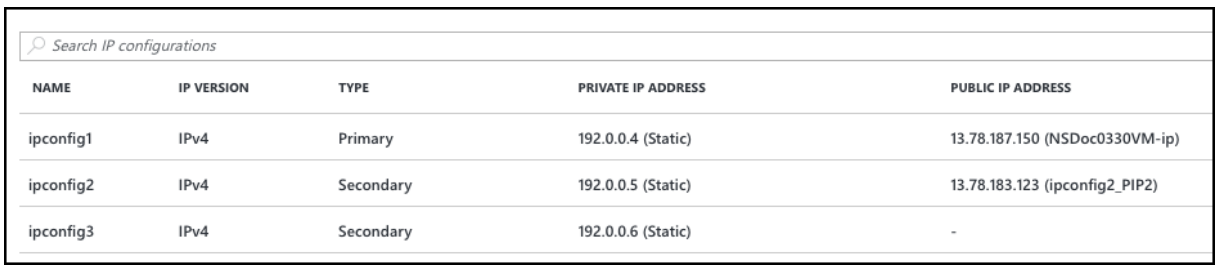
デフォルトでは、パブリック IP アドレスは動的なアドレスです。VM に常に同じパブリック IP アドレスを使用させるために、静的なパブリック IP アドレスを作成します。

「パブリック IP アドレスの作成」ブレードで「名前」を追加し、「割り当て」で「静的」をクリックします。[**OK**] をクリックします。



> 注
>
> 割り当て方式を静的に設定している場合でも、パブリック IP リソースに割り当てられる実際の IP アドレスを指定することはできません。代わりに、リソースが作成された Azure の場所で利用可能な IP アドレスプールから割り当てられます。

手順に従って、もう 1 つの IP 構成 ipconfig3 を追加します。パブリック IP アドレスは必須ではありません。



手順 **2**: **NetScaler** 固有の **IP** アドレスの構成

GUI または`add ns ip`コマンドを使用して、NetScaler が所有する IP アドレスを設定します。詳細については、「NetScaler ADC 所有の IP アドレスの構成」を参照してください。

複数の **IP** アドレスと **NIC** を使用して高可用性設定を構成する

December 8, 2023

Microsoft Azure デプロイメントでは、Azure ロードバランサー（ALB）を使用して、2 つの NetScaler VPX インスタンスの高可用性構成を実現します。これは、ALB でヘルスプローブを構成することによって実現されます。ALB は、プライマリインスタンスとセカンダリインスタンスの両方に 5 秒ごとにヘルスプローブを送信することで、各 VPX インスタンスを監視します。

この設定では、プライマリノードだけがヘルスプローブに応答し、セカンダリノードは応答しません。プライマリがヘルスプローブに応答を送信すると、ALB はインスタンスへのデータトラフィックの送信を開始します。プライマリインスタンスが 2 回連続してヘルスプローブに失敗した場合、ALB はトラフィックをそのインスタンスにリダイレクトしません。フェイルオーバー時は、新しいプライマリがヘルスプローブへの応答を開始し、ALB はそのプライマリにトラフィックをリダイレクトします。標準の VPX 高可用性フェイルオーバー時間は 3 秒です。トラフィックスイッチングにかかる合計フェールオーバー時間は、最大 13 秒です。

Azure のアクティブ/パッシブ高可用性（HA）セットアップで、複数の NIC を持つ一対の NetScaler VPX インスタンスを展開できます。各 NIC に複数の IP アドレスを設定できます。

マルチ NIC の高可用性導入では、次のオプションを使用できます。

- Azure 可用性セットを使用した高可用性
- Azure アベイラビリティーゾーンを使用した高可用性

Azure アベイラビリティセットとアベイラビリティーゾーンの詳細については、Azure のドキュメント「Linux 仮想マシンの可用性の管理」を参照してください。

可用性セットを使用した高可用性

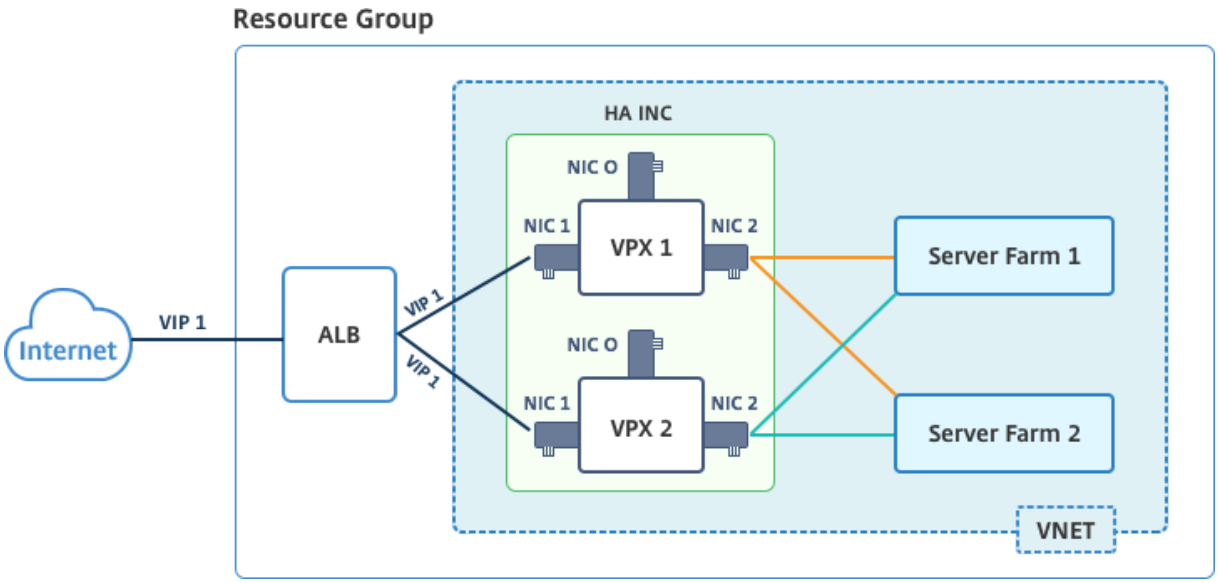
可用性セットを使用した高可用性セットアップは、次の要件を満たす必要があります。

- HA Independent Network Configuration（INC）構成
- Direct Server Return（DSR）モードの Azure Load Balancer（ALB）

すべてのトラフィックはプライマリノードを経由します。セカンダリノードは、プライマリノードが失敗するまでスタンバイモードを維持します。

> 注
>
> Azure クラウド上の NetScaler VPX 高可用性デプロイが機能するには、2 つの VPX ノード間で移動できるフローティングパブリック IP（PIP）が必要です。Azure ロードバランサー (ALB) は、フローティング PIP を提供します。このフローティング PIP は、フェールオーバーが発生した場合に自動的に第 2 ノードに移動されます。

図:Azure 可用性セットを使用した高可用性デプロイアーキテクチャの例



アクティブ/パッシブ展開では、ALB フロントエンドパブリック IP (PIP) アドレスが各 VPX ノードに VIP アドレスとして追加されます。HA-INC 構成では、これらの VIP アドレスはフローティングされ、SNIP アドレスはインスタンス固有のアドレスとなります。

VPX ペアをアクティブ/パッシブ高可用性モードでデプロイするには、次の 2 つの方法があります。

- **NetScaler VPX** 標準高可用性テンプレート: このオプションを使用して、3 つのサブネットと 6 つの NIC のデフォルトオプションで HA ペアを構成します。
- **Windows PowerShell** コマンド: このオプションを使用して、サブネットと NIC の要件に応じて HA ペアを構成します。

このトピックでは、Citrix テンプレートを使用してアクティブ/パッシブ HA セットアップで VPX ペアを展開する方法について説明します。PowerShell コマンドを使用する場合は、PowerShell コマンドを使用して複数の IP アドレスと NIC を使用した HA セットアップの構成を参照してください。

NetScaler の高可用性テンプレートを使用して **HA-INC** ノードを構成する

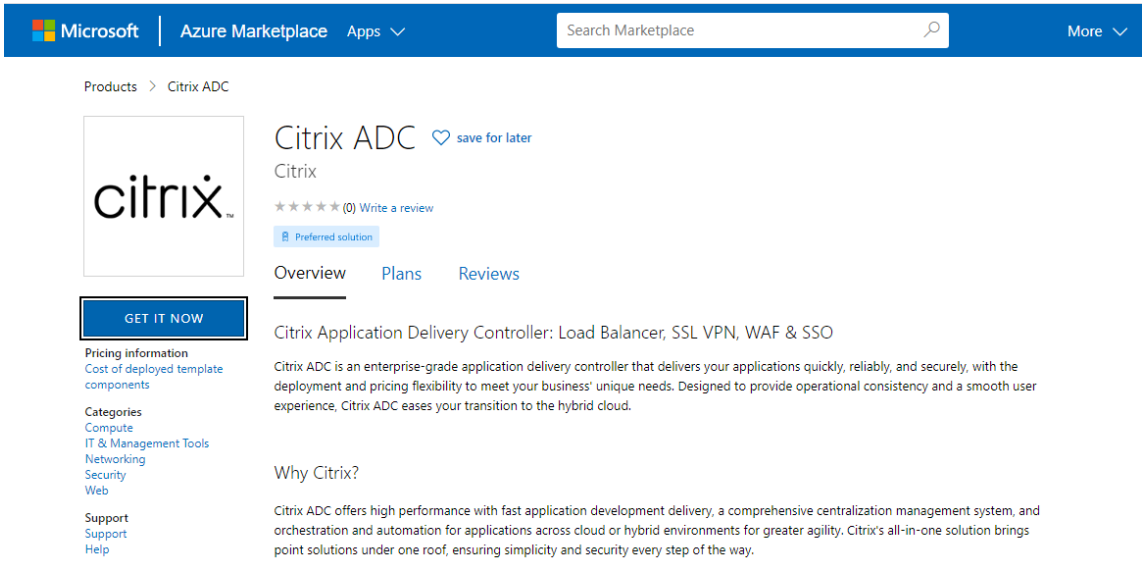
標準テンプレートを使用すると、一対の VPX インスタンスを HA-INC モードで迅速かつ効率的にデプロイできます。このテンプレートでは、3 つのサブネットと 6 つの NIC を持つ 2 つのノードが作成されます。サブネットは管理、ク

ライアント、サーバー側のトラフィック用です。各サブネットには、両方の VPX インスタンスに対して 2 つの NIC があります。

NetScaler HA ペアテンプレートは、Azure マーケットプレイスで入手できます。

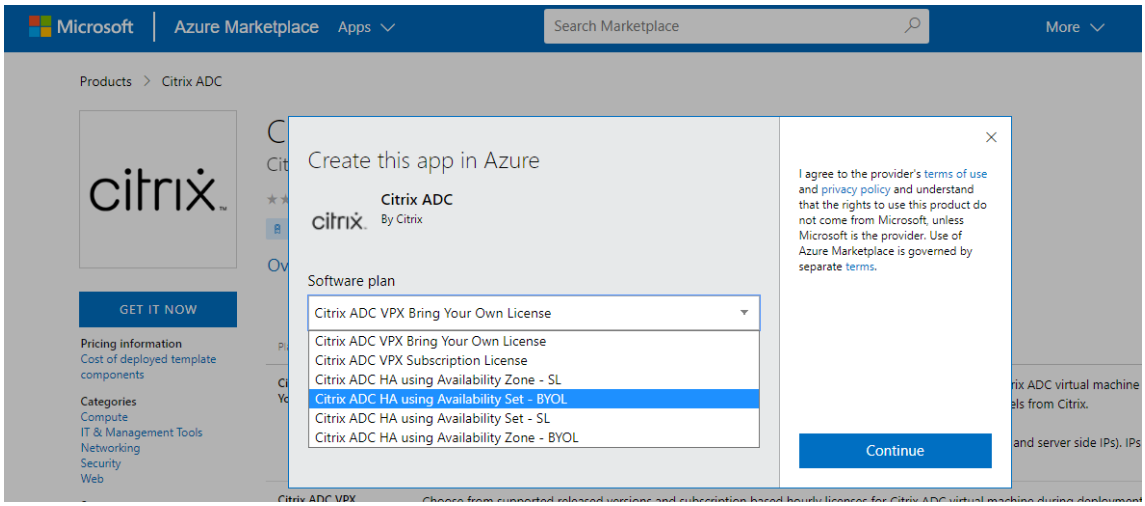
次の手順を実行して、Azure 可用性セットを使用して、テンプレートを起動し、高可用性 VPX ペアをデプロイします。

1. Azure Marketplace から **NetScaler** を検索します。

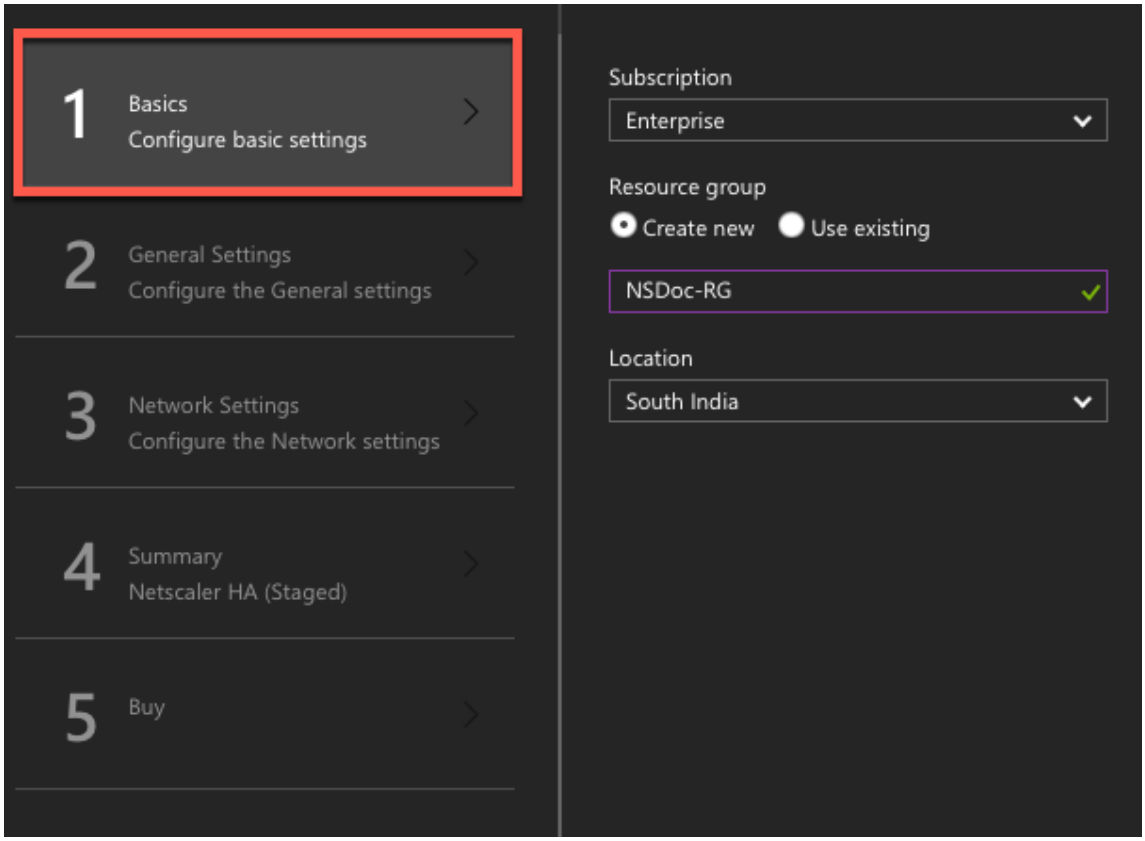


2. [今すぐ入手] をクリックします。

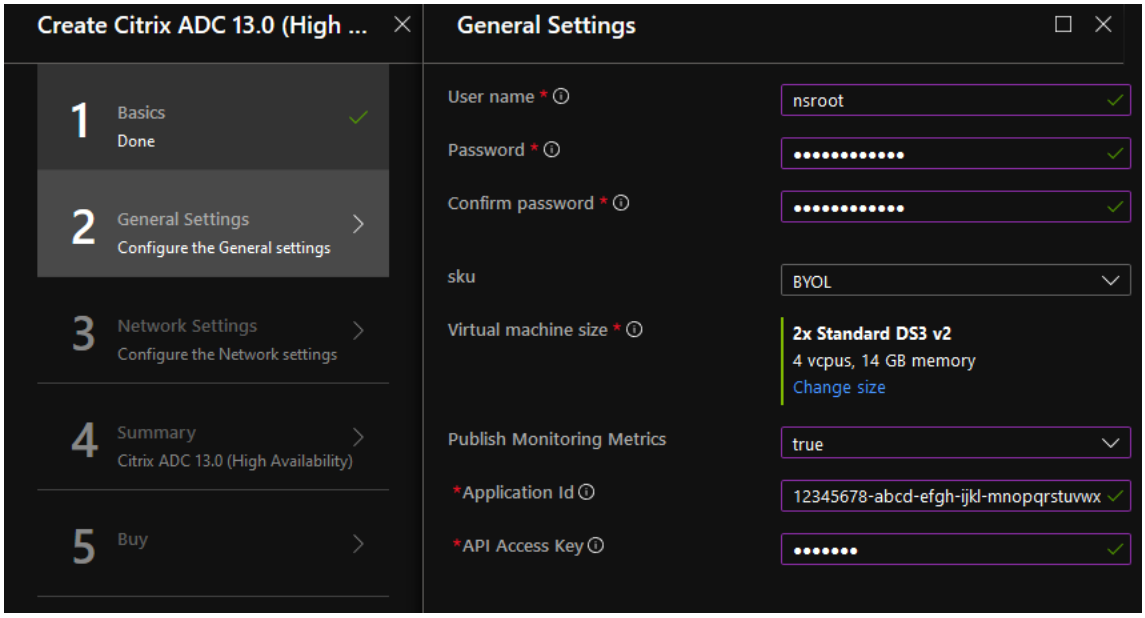
3. 必要な HA 導入とライセンスを選択し、[続行] をクリックします。



4. [基本] ページが表示されます。リソースグループを作成し、**OK** を選択します。



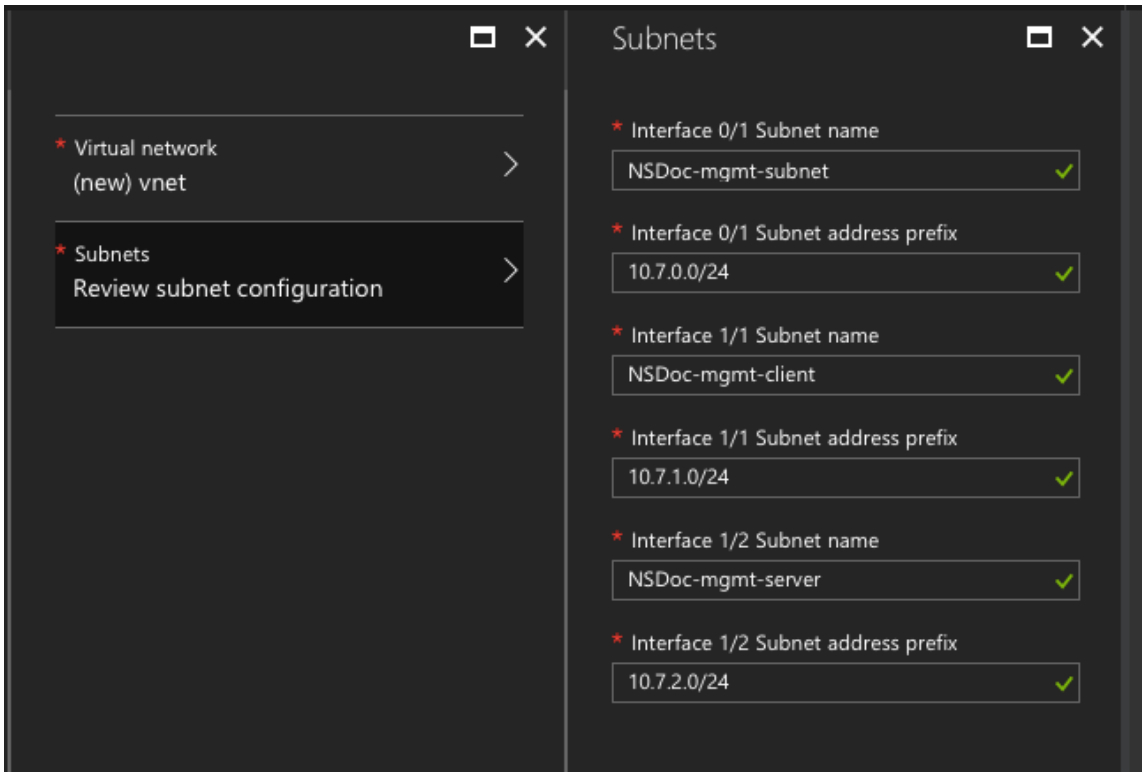
5. [一般設定] ページが表示されます。詳細を入力して「**OK**」を選択します。



> 注:
>
> デフォルトでは、** 監視指標の公開オプションは **false** に設定されています。このオプションを有効にする場合は、****true** を選択してください。

> リソースにアクセスできる AzureActive Directory（ADD）アプリケーションとサービスプリンシパルを作成します。新しく作成された AAD アプリケーションにコントリビュータロールを割り当てます。詳細については、「ポータルを使用してリソースにアクセスできる Azure Active Directory アプリケーションおよびサービスプリンシパルを作成する」を参照してください。

6. [ネットワーク設定] ページが表示されます。VNet とサブネットの構成を確認し、必要な設定を編集して、[**OK**] を選択します。

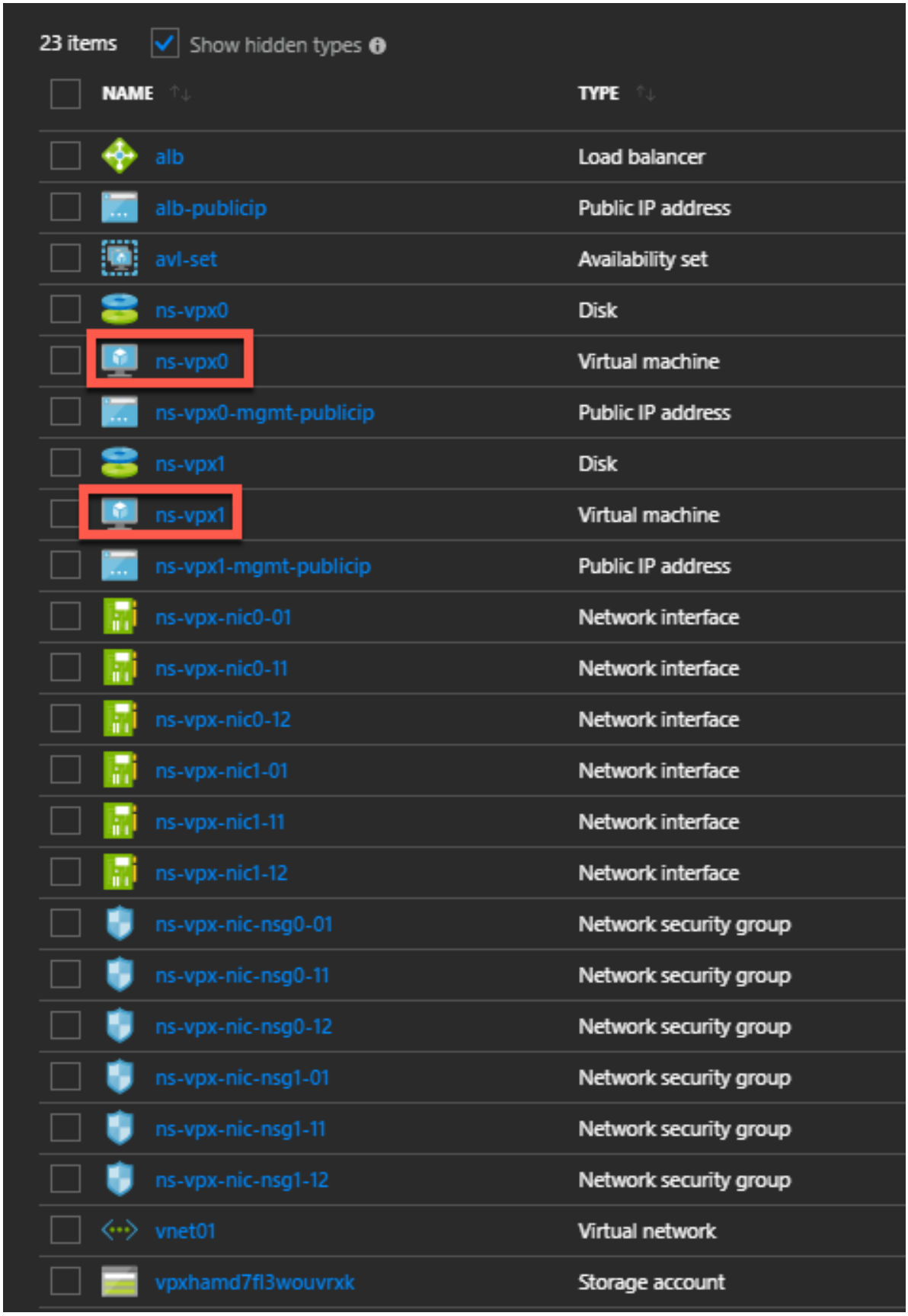


7. ［概要］ページが開きます。構成を確認し、適宜編集します。［**OK**］を選択して確定します。

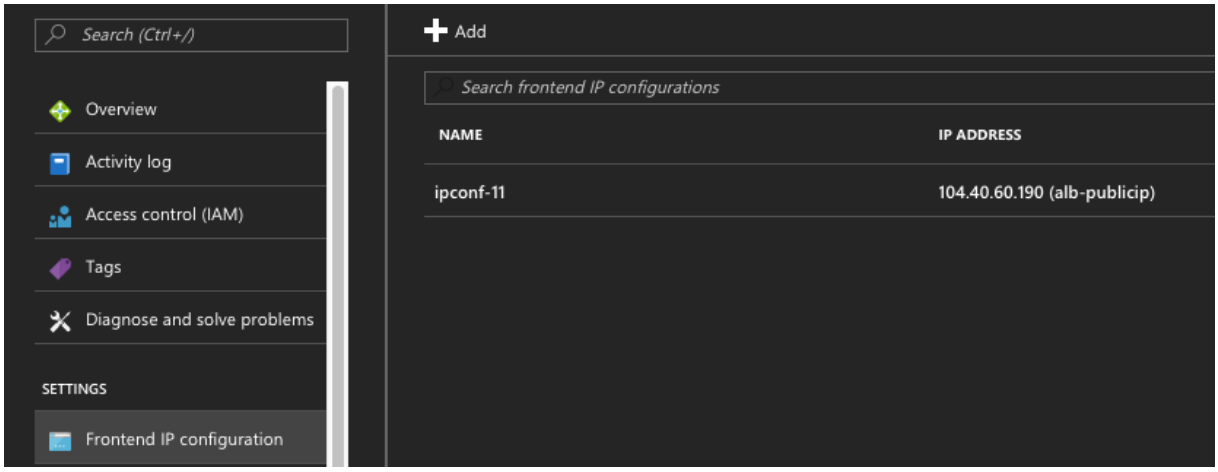
8. 「購入」ページが表示されます。[購入] を選択してデプロイを完了します。

必要な構成で Azure リソースグループが作成されるまで時間がかかることがあります。完了後、Azure ポータルでリソースグループを選択し、LB ルール、バックエンドプール、ヘルスプローブなどの構成の詳細を表示します。高可用性ペアは、ns-vpx0 および ns-vpx1 と表示されます。

追加のセキュリティルールやポートを作成するなど、HA セットアップでさらに変更が必要な場合は、Azure Portal から実行できます。



次に、プライマリノードで **ALB** のフロントエンドパブリック **IP**（**PIP**）アドレスを使用して負荷分散仮想サーバーを構成する必要があります。ALB PIP を検索するには、ALB > フロントエンド IP 設定を選択します。



負荷分散仮想サーバーの構成方法の詳細については、「リソース」セクションを参照してください。

リソース:

次のリンクには、HA の導入と仮想サーバの設定に関する追加情報が表示されます。

- 異なるサブネットでの高可用性ノードの構成
- 基本的な負荷分散を設定する

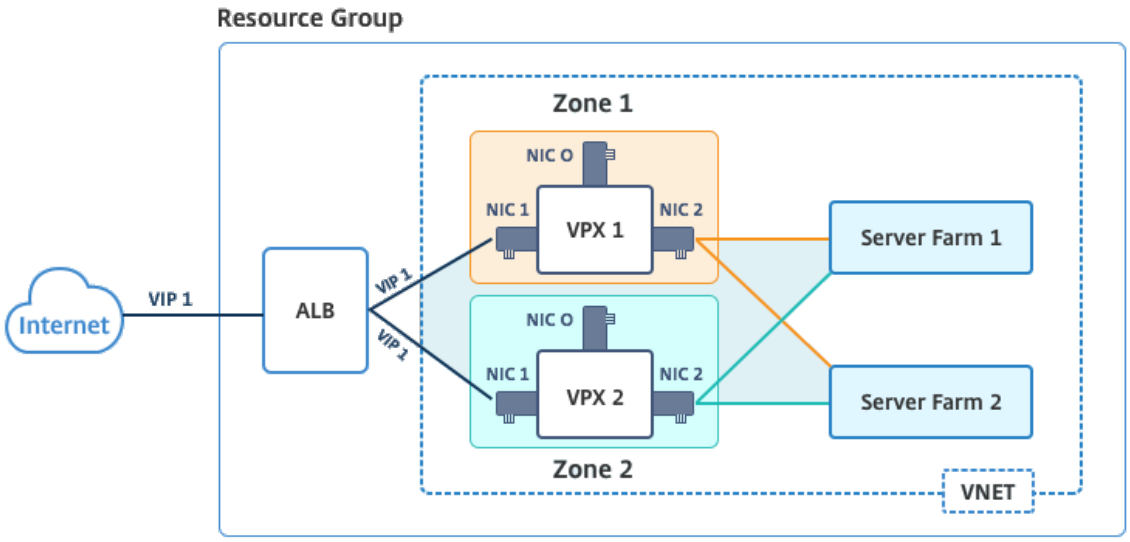
関連リソース:

- PowerShell コマンドを使用して、複数の IP アドレスと NIC を使用した高可用性設定を構成する
- Azure でのアクティブスタンバイ HA デプロイメントでの GSLB の構成

アベイラビリティーゾーンを使用した高可用性

Azure アベイラビリティーゾーンは、Azure リージョン内の障害分離された場所であり、冗長な電源、冷却、ネットワーキングを提供し、回復力を高めます。特定の Azure リージョンだけがアベイラビリティーゾーンをサポートします。詳細については、Azure のドキュメント [Azure のアベイラビリティーゾーンとは何ですか] を参照してください。

図:Azure アベイラビリティーゾーンを使用した高可用性デプロイアーキテクチャの例



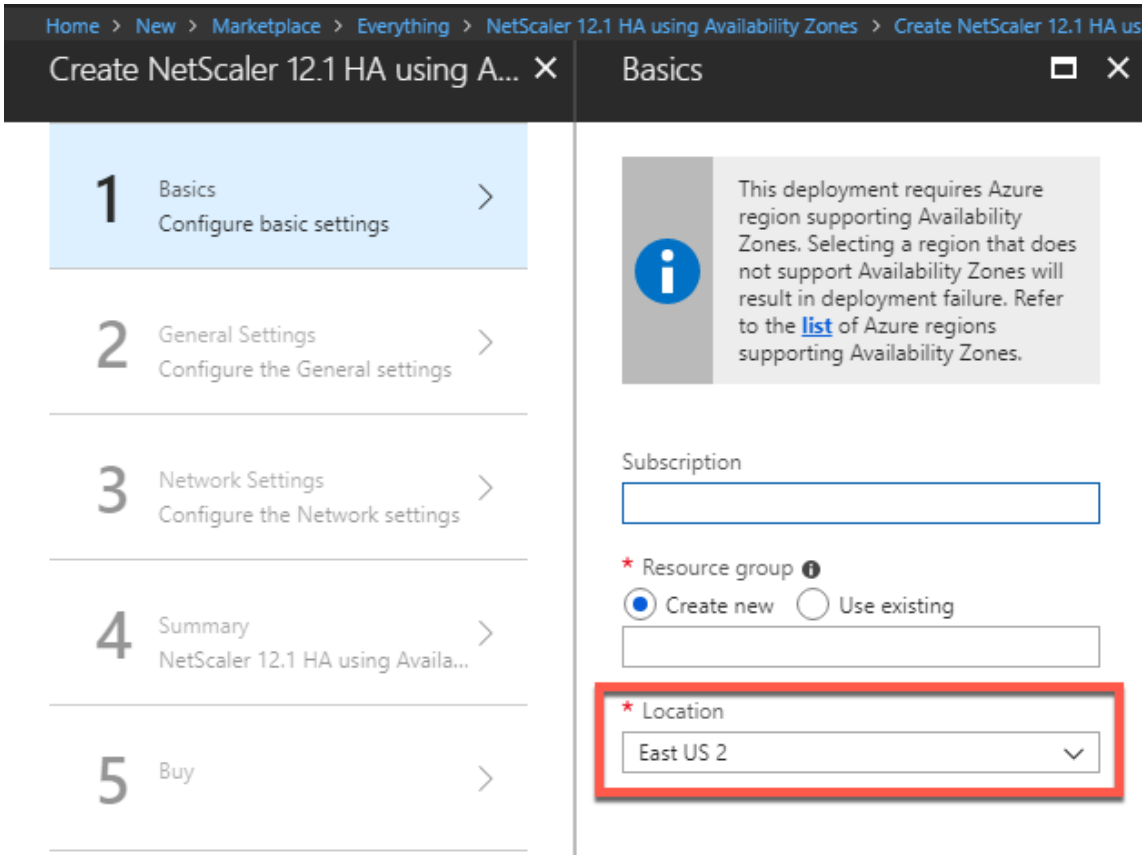
Azure Marketplace e で入手可能な「アベイラビリティーゾーンを使用した NetScaler 13.0 HA」というテンプレートを使用して、VPX ペアを高可用性モードでデプロイできます。

Azure アベイラビリティーゾーンを使用してテンプレートを起動し、高可用性 VPX ペアをデプロイするには、次の手順を実行します。

1. Azure Marketplace から、Citrix ソリューションテンプレートを選択して開始します。

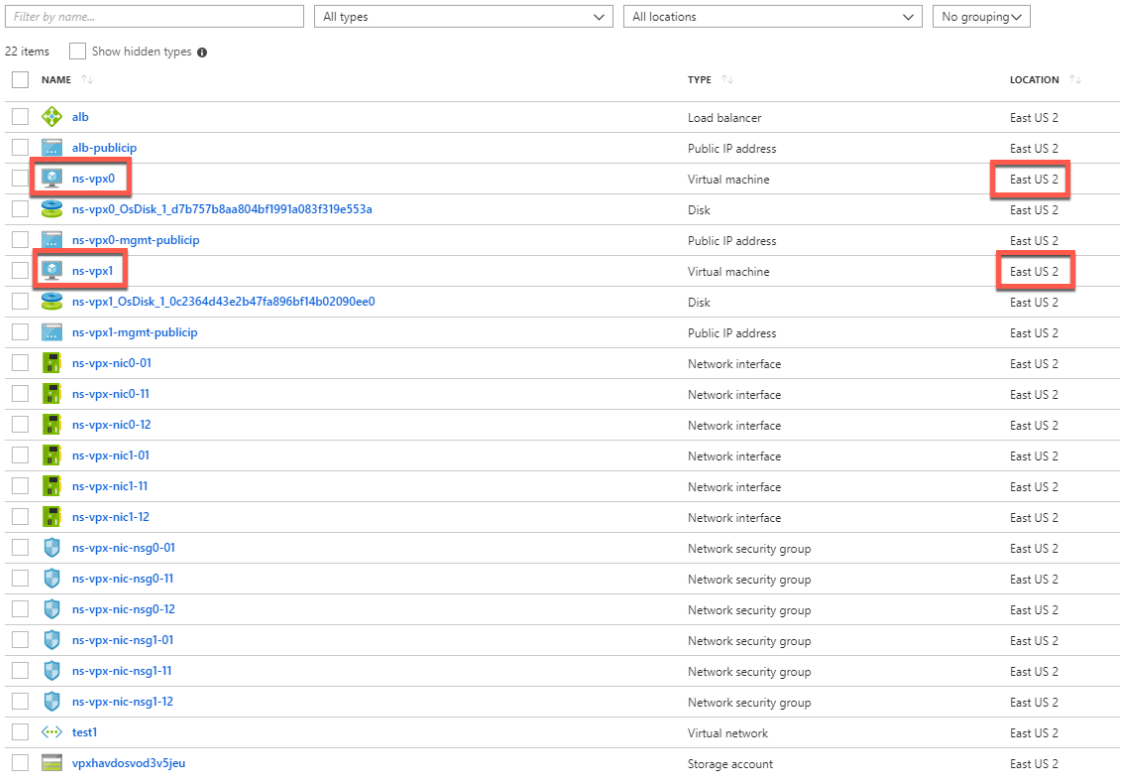
2. デプロイメント・タイプがリソース・マネージャーであることを確認し、「作成」を選択します。
3. [基本] ページが表示されます。詳細を入力し、[**OK**] をクリックします。

> 注: アベイラビリティーゾーンをサポートする Azure リージョンを選択してください。アベイラビリティーゾーンをサポートするリージョンの詳細については、Azure のドキュメントを参照してください。Azure のアベイラビリティーゾーンは何ですか。



4. [一般設定] ページが表示されます。詳細を入力して「**OK**」を選択します。
5. [ネットワーク設定] ページが表示されます。VNet とサブネットの構成を確認し、必要な設定を編集して、[**OK]** を選択します。
6. [概要] ページが開きます。構成を確認し、適宜編集します。[**OK**] を選択して確定します。
7. 「購入」ページが表示されます。[購入] を選択してデプロイを完了します。

   必要な構成で Azure リソースグループが作成されるまで時間がかかることがあります。完了後、リソースグループを選択すると、LB ルール、バックエンドプール、ヘルスプローブなどの構成の詳細が Azure ポータルに表示されます。高可用性ペアは、ns-vpx0 および ns-vpx1 と表示されます。また、「場所」列にも場所が表示されます。



追加のセキュリティルールやポートを作成するなど、HA セットアップでさらに変更が必要な場合は、Azure Portal から実行できます。

Azure モニターのメトリックを使用してインスタンスを監視する

Azure モニターデータプラットフォームのメトリックを使用して、CPU、メモリ使用率、スループットなどの一連の NetScaler VPX リソースを監視できます。メトリックサービスは、Azure 上で稼働する NetScaler VPX リソースをリアルタイムで監視します。メトリックスエクスプローラーを使用して、収集されたデータにアクセスできます。詳細については、「Azure Monitor メトリックスの概要」を参照してください。

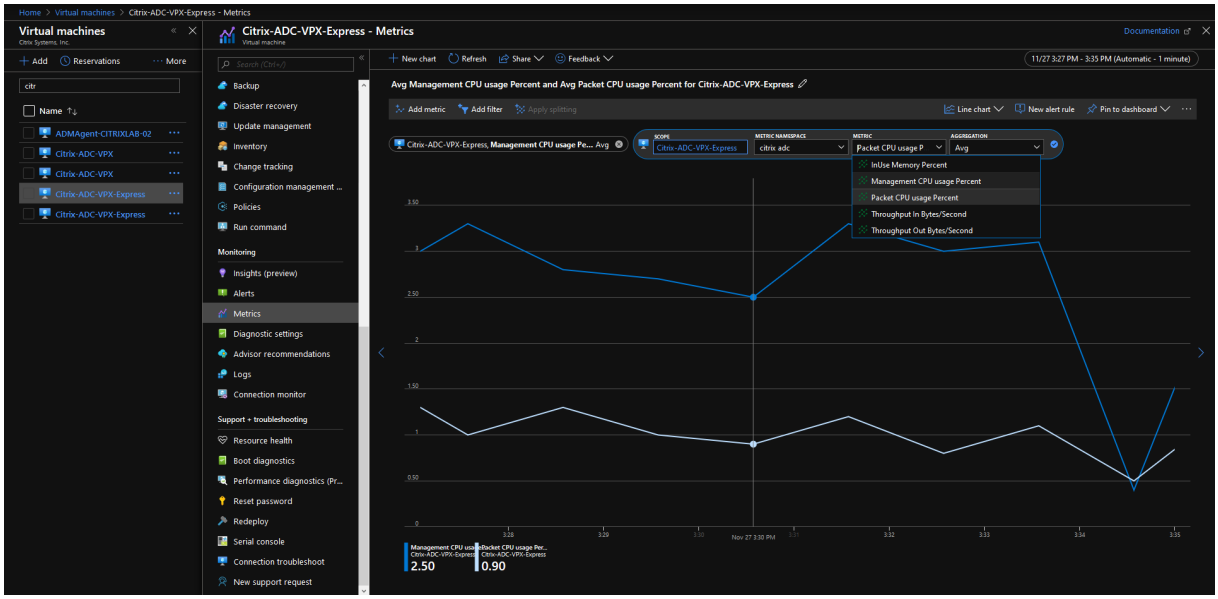
注意事項

- Azure Marketplace オファーを使用して NetScaler VPX インスタンスを Azure にデプロイすると、メトリックスサービスはデフォルトで無効になります。
- メトリックスサービスは Azure CLI ではサポートされていません。
- メトリクスは、CPU (管理およびパケット CPU 使用率)、メモリ、およびスループット (インバウンドとアウトバウンド) で使用できます。

Azure モニターでメトリックを表示する方法

インスタンスの Azure モニターでメトリックスを表示するには、次の手順を実行します。

1. **Azure Portal > Virtual Machines** にログオンします。
2. プライマリノードとなる仮想マシンを選択します。
3. モニタリングセクションで、メトリクスをクリックします。
4. メトリック名前空間のドロップダウンメニューから、**NetScaler** をクリックします。
5.「指標」ドロップダウンメニューの「すべての指標」で、表示したい指標をクリックします。
6. [指標を追加] をクリックすると、同じグラフに別の指標が表示されます。チャートオプションを使用してチャートをカスタマイズします。



PowerShell コマンドを使用して、複数の **IP** アドレスと **NIC** を使用した高可用性設定を構成する

August 15, 2023

Azure のアクティブ/パッシブ高可用性（HA）セットアップで、複数の NIC を持つ一対の NetScaler ADC VPX インスタンスを展開できます。各 NIC に複数の IP アドレスを設定できます。

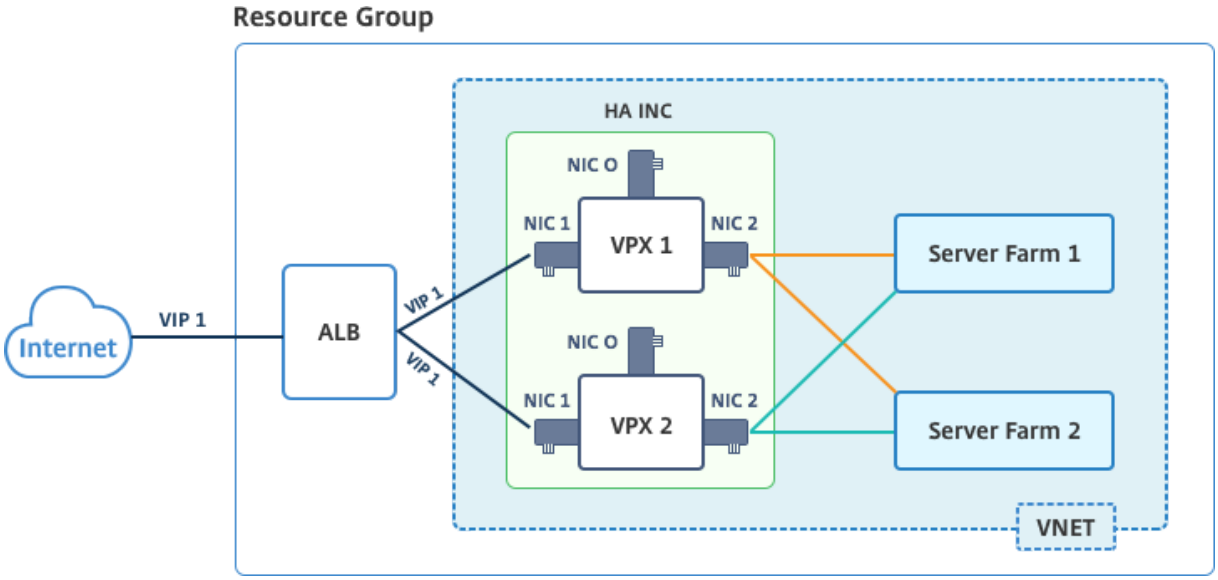
アクティブ/パッシブ展開では以下が必要です。

- HA Independent Network Configuration（INC）構成
- Direct Server Return（DSR）モードの Azure Load Balancer（ALB）

すべてのトラフィックはプライマリノードを経由します。セカンダリノードは、プライマリノードが失敗するまでスタンバイモードを維持します。

> 注
>
> Azure クラウド上の NetScaler ADC VPX 高可用性展開を機能させるには、2 つの高可用性ノード間で移動できるフローティングパブリック IP（PIP）が必要です。Azure ロードバランサー (ALB) は、フローティング PIP を提供します。このフローティング PIP は、フェールオーバーが発生した場合に自動的に第 2 ノードに移動されます。

図: アクティブ-パッシブ展開アーキテクチャの例



アクティブ/パッシブ展開では、ALB フローティングパブリック IP（PIP）アドレスが各 VPX ノードに VIP アドレスとして追加されます。HA-INC 構成では、これらの VIP アドレスはフローティングされ、SNIP アドレスはインスタンス固有のアドレスとなります。

ALB は 5 秒ごとにヘルスプローブを送信して各 VPX インスタンスを監視し、定期的にヘルスプローブ応答を送信するトラフィックのみをそのインスタンスにリダイレクトします。そのため、HA セットアップでは、プライマリノードがヘルスプローブに応答し、セカンダリノードは応答しません。プライマリインスタンスが 2 つの連続したヘルスプローブを見逃した場合、ALB はそのインスタンスにトラフィックをリダイレクトしません。フェイルオーバー時は、新しいプライマリがヘルスプローブへの応答を開始し、ALB はそのプライマリにトラフィックをリダイレクトします。標準の VPX 高可用性フェイルオーバー時間は 3 秒です。トラフィック切り替えにかかる合計フェイルオーバー時間は、最大で 13 秒になる可能性があります。

VPX ペアをアクティブ/パッシブ HA セットアップに導入するには、次の 2 つの方法があります。

- **NetScaler VPX** 標準高可用性テンプレート: このオプションを使用して、3 つのサブネットと 6 つの NIC のデフォルトオプションで HA ペアを構成します。
- **Windows PowerShell** コマンド: このオプションを使用して、サブネットと NIC の要件に応じて HA ペアを構成します。

このトピックでは、PowerShell コマンドを使用してアクティブ/パッシブ HA セットアップで VPX ペアを展開する方法について説明します。NetScaler VPX 標準 HA テンプレートを使用する場合は、複数の IP アドレスと NIC を使用した HA セットアップの構成を参照してください。

PowerShell コマンドを使用して HA-INC ノードを構成する

シナリオ:HA-INC PowerShell の展開

このシナリオでは、表に示されているトポロジを使用して NetScaler VPX ペアをデプロイします。各 VPX インスタンスには 3 つの NIC が含まれており、各 NIC は異なるサブネットにデプロイされます。各 NIC には IP 構成が割り当てられます。

ALB	VPX1	VPX2
ALB はパブリック IP 3 (pip3) に関連付けられています	管理 IP は IPConfig1 で設定されます。これには 1 つのパブリック IP (pip1) と 1 つのプライベート IP (12.5.2.24) が含まれます。nic1; Mgmtsubnet=12.5.2.0/24	管理 IP は IPConfig5 で設定されます。これには 1 つのパブリック IP (pip3) と 1 つのプライベート IP (12.5.2.26) が含まれます。nic4; Mgmtsubnet=12.5.2.0/24
LB ルールとポートは HTTP (80)、SSL (443)、ヘルスプローブ (9000) に設定されています	クライアント側の IP は IPConfig3 で設定されます。これには 1 つのプライベート IP (12.5.1.27)、nic2、frontendSubet=12.5.1.0/24 が含まれます	クライアント側の IP は IPConfig7 で設定されます。これには 1 つのプライベート IP (12.5.1.28)、nic5、frontendSubet=12.5.1.0/24 の 1 つのプライベート IP が含まれます
-	サーバー側の IP は IPConfig4 で設定されます。これには 1 つのプライベート IP (12.5.3.24)、nic3、backendSubnet=12.5.3.0/24 が含まれます	サーバー側の IP は IPConfig8 で設定されます。これには 1 つのプライベート IP (12.5.3.28)、nic6、backendSubnet=12.5.3.0/24 が含まれます
-	NSG のルールとポートは SSH (22)、HTTP (80)、HTTPS (443) です	-

パラメータ設定

このシナリオでは、次のパラメータ設定が使用されます。

$locName = “East Asia”

$rgName = “MulitIP-MultiNIC-RG”

$nicName1= “VM1-NIC1”

$nicName2 = “VM1-NIC2”

$nicName3= “VM1-NIC3”

$nicName4 = “VM2-NIC1”

$nicName5= “VM2-NIC2”

$nicName6 = “VM2-NIC3”

$vNetName = “Azure-MultiIP-ALB-vnet”

$vNetAddressRange= “12.5.0.0/16”

$frontEndSubnetName= “frontEndSubnet”

$frontEndSubnetRange= “12.5.1.0/24”

$mgmtSubnetName= “mgmtSubnet”

$mgmtSubnetRange= “12.5.2.0/24”

$backEndSubnetName = “backEndSubnet”

$backEndSubnetRange = “12.5.3.0/24”

$prmStorageAccountName = “multiipmultinicbstorage”

$avSetName = “multiple-avSet”

$vmSize= “Standard_DS4_V2”

$publisher = “Citrix”

$offer = “netscalervpx-120”

$sku = “netscalerbyol”

$version=” latest”

$pubIPName1=” VPX1MGMT”

$pubIPName2=” VPX2MGMT”

$pubIPName3=” ALBPIP”

$domName1=” vpx1dns”

$domName2=” vpx2dns”

$domName3=” vpxalbdns”

$vmNamePrefix=” VPXMultiIPALB”

$osDiskSuffix1=” osmultiipalbdiskdb1”

$osDiskSuffix2=” osmultiipalbdiskdb2”

$lbName= “MultiIPALB”

$frontEndConfigName1= “FrontEndIP”

$backendPoolName1= “BackendPoolHttp”

$lbRuleName1= “LBRuleHttp”

$healthProbeName= “HealthProbe”

$nsgName=” NSG-MultiIP-ALB”

$rule1Name=” Inbound-HTTP”

$rule2Name=” Inbound-HTTPS”

$rule3Name=” Inbound-SSH”

展開を完了するには、PowerShell コマンドを使用して次の手順を完了します。

1. リソースグループ、ストレージアカウント、高可用性セットの作成
2. ネットワークセキュリティグループの作成と規則の追加
3. 仮想ネットワークと 3 つのサブネットの作成
4. パブリック IP アドレスの作成
5. VPX1 の IP 構成の作成
6. VPX2 の IP 構成の作成
7. VPX1 の NIC の作成
8. VPX2 の NIC の作成
9. VPX1 の作成
10. VPX2 の作成
11. ALB の作成

リソースグループ、ストレージアカウント、および可用性セットを作成します。

```
New-AzureRmResourceGroup -Name $rgName -Location $locName


$prmStorageAccount=New-AzureRMStorageAccount -Name
    $prmStorageAccountName -ResourceGroupName $rgName -Type Standard_LRS
     -Location $locName


$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
    $rgName -Location $locName
```

ネットワークセキュリティグループを作成し、ルールを追加します。

```
$rule1 = New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -
    Description "Allow HTTP" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 101

```



```
3
4  -SourceAddressPrefix Internet -SourcePortRange * -
       DestinationAddressPrefix * -DestinationPortRange 80
5
6
7  $rule2 = New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -
       Description "Allow HTTPS" -Access Allow -Protocol Tcp -Direction
       Inbound -Priority 110
8
9
10 -SourceAddressPrefix Internet -SourcePortRange * -
       DestinationAddressPrefix * -DestinationPortRange 443
11
12
13 $rule3 = New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name -
       Description "Allow SSH" -Access Allow -Protocol Tcp -Direction
       Inbound -Priority 120
14
15
16 -SourceAddressPrefix Internet -SourcePortRange * -
       DestinationAddressPrefix * -DestinationPortRange 22
17
18
19 $nsg = New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName -
       Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,$rule3
```

仮想ネットワークと **3** つのサブネットを作成します。

```
1  $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       $frontEndSubnetName -AddressPrefix $frontEndSubnetRange (this
       parameter value should be as per your requirement)
2
3
4  $mgmtSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name $mgmtSubnetName
        -AddressPrefix $mgmtSubnetRange
5
6
7  $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       $backEndSubnetName -AddressPrefix $backEndSubnetRange
8
9
10 $vnet =New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
       $rgName -Location $locName -AddressPrefix $vNetAddressRange -Subnet
       $frontendSubnet,$backendSubnet, $mgmtSubnet
11
12
13 $subnetName ="frontEndSubnet"
14
15
16 \$subnet1=\$vnet.Subnets|?{
17  \$\_.Name -eq \$subnetName }
18
19
```



```
20
21 $subnetName="backEndSubnet"
22
23
24 \$subnet2=\$vnet.Subnets|?{
25  \$\_.Name -eq \$subnetName }
26
27
28
29 $subnetName="mgmtSubnet"
30
31
32 \$subnet3=\$vnet.Subnets|?{
33  \$\_.Name -eq \$subnetName }
```

パブリック **IP** アドレスを作成します。

```
1 $pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
     $rgName -DomainNameLabel $domName1 -Location $locName -
     AllocationMethod Dynamic
2
3 $pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
     $rgName -DomainNameLabel $domName2 -Location $locName -
     AllocationMethod Dynamic
4
5 $pip3=New-AzureRmPublicIpAddress -Name $pubIPName3 -ResourceGroupName
     $rgName -DomainNameLabel $domName3 -Location $locName -
     AllocationMethod Dynamic
```

VPX1 の **IP** 構成を作成します。

```
1  $IpConfigName1 = "IPConfig1"
2
3
4  $IPAddress = "12.5.2.24"
5
6
7  $IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
      Subnet $subnet3 -PrivateIpAddress $IPAddress -PublicIpAddress $pip1
      -Primary
8
9
10 $IPConfigName3="IPConfig-3"
11
12
13 $IPAddress="12.5.1.27"
14
15
16 $IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
      Subnet $subnet1 -PrivateIpAddress $IPAddress -Primary
17
18
19 $IPConfigName4 = "IPConfig-4"
```



```
20
21
22 $IPAddress = "12.5.3.24"
23
24
25 $IPConfig4 = New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -
       Subnet $subnet2 -PrivateIpAddress $IPAddress -Primary
```

VPX2 の **IP** 構成を作成します。

```
1  $IpConfigName5 = "IPConfig5"
2
3
4  $IPAddress="12.5.2.26"
5
6
7  $IPConfig5=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName5 -
       Subnet $subnet3 -PrivateIpAddress $IPAddress -PublicIpAddress $pip2
       -Primary
8
9
10 $IPConfigName7="IPConfig-7"
11
12
13 $IPAddress="12.5.1.28"
14
15
16 $IPConfig7=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName7 -
       Subnet $subnet1 -PrivateIpAddress $IPAddress -Primary
17
18
19 $IPConfigName8="IPConfig-8"
20
21
22 $IPAddress="12.5.3.28"
23
24
25 $IPConfig8=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName8 -
       Subnet $subnet2 -PrivateIpAddress $IPAddress -Primary
```

VPX1 用の **NIC** を作成します。

```
1  $nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName
       $rgName -Location $locName -IpConfiguration $IpConfig1 -
       NetworkSecurityGroupId $nsg.Id
2
3
4  $nic2=New-AzureRmNetworkInterface -Name $nicName2 -ResourceGroupName
       $rgName -Location $locName -IpConfiguration $IpConfig3 -
       NetworkSecurityGroupId $nsg.Id
5
6
7  $nic3=New-AzureRmNetworkInterface -Name $nicName3 -ResourceGroupName
```



```
    $rgName -Location $locName -IpConfiguration $IpConfig4 -
    NetworkSecurityGroupId $nsg.Id
```

VPX2 用の **NIC** を作成します。

```
1 $nic4=New-AzureRmNetworkInterface -Name $nicName4 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig5 -
    NetworkSecurityGroupId $nsg.Id
2
3
4 $nic5=New-AzureRmNetworkInterface -Name $nicName5 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig7 -
    NetworkSecurityGroupId $nsg.Id
5
6
7 $nic6=New-AzureRmNetworkInterface -Name $nicName6 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig8 -
    NetworkSecurityGroupId $nsg.Id
```

VPX1 を作成します。

この手順には、次の下位手順が含まれています。

- VM 設定オブジェクトの作成
- 資格情報、OS、イメージの設定
- NIC の追加
- OS ディスクの指定と VM の作成

```
1  $suffixNumber = 1
2
3  $vmName=$vmNamePrefix + $suffixNumber
4
5  $vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
       AvailabilitySetId $avSet.Id
6
7  $cred=Get-Credential -Message "Type the name and password for VPX
       login."
8
9  $vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
       ComputerName $vmName -Credential $cred
10
11 $vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
       $publisher -Offer $offer -Skus $sku -Version $version
12
13 $vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.
       Id -Primary
14
15 $vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.
       Id
16
```



```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.Id

$osDiskName=$vmName + "-" + $osDiskSuffix1

$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/" + $osDiskName + ".vhd"

$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri $osVhdUri -CreateOption fromImage

Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -Name $sku

New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location $locName
```

VPX2 を作成します。

```
```
$suffixNumber=2

$vmName=$vmNamePrefix + $suffixNumber

$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avSet.Id

$cred=Get-Credential -Message "Type the name and password for VPX login."

$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -ComputerName $vmName -Credential $cred

$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName $publisher -Offer $offer -Skus $sku -Version $version

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic4.Id -Primary

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic5.Id

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic6.Id

$osDiskName=$vmName + "-" + $osDiskSuffix2
```

 
```

```
30
31
32 $osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
       + $osDiskName + ".vhd"
33
34
35 $vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
       $osVhdUri -CreateOption fromImage
36
37
38 Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
       Name $sku
39
40
41 New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
       $locName
42 <!--NeedCopy--> ```
```

NIC に割り当てられたプライベート IP アドレスとパブリック IP アドレスを表示するには、次のコマンドを入力します。

```
1  ```
2  $nic1.IPConfig
3
4
5  $nic2.IPConfig
6
7
8  $nic3.IPConfig
9
10
11 $nic4.IPConfig
12
13
14 $nic5.IPConfig
15
16
17 $nic6.IPConfig
18 <!--NeedCopy--> ```
```

Azure の負荷分散 **(ALB)** を作成します。

この手順には、次の下位手順が含まれています。

- フロントエンド IP 構成を作成する
- ヘルスプローブの作成
- バックエンドアドレスプールの作成
- 負荷分散規則（HTTP および SSL）の作成
- フロントエンド IP 設定、バックエンドアドレスプール、および LB ルールを使用して ALB を作成します。

- IP 構成をバックエンドプールに関連付ける

```
$frontEndIP1=New-AzureRmLoadBalancerFrontendIpConfig -Name
$frontEndConfigName1 -PublicIpAddress $pip3

$healthProbe=New-AzureRmLoadBalancerProbeConfig -Name $healthProbeName
 -Protocol Tcp -Port 9000 -IntervalInSeconds 5 -ProbeCount 2

$beAddressPool1=New-AzureRmLoadBalancerBackendAddressPoolConfig -
Name $backendPoolName1

$lbRule1=New-AzureRmLoadBalancerRuleConfig -Name $lbRuleName1
 -FrontendIpConfiguration $frontEndIP1 -BackendAddressPool
$beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort
80 -BackendPort 80 -EnableFloatingIP

$lb=New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name
$lbName -Location $locName -FrontendIpConfiguration $frontEndIP1
-LoadBalancingRule $lbRule1 -BackendAddressPool $beAddressPool1 -
Probe $healthProbe

$nic2.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($lb
.BackendAddressPools[0])

$nic5.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($lb
.BackendAddressPools[0])

$lb=$lb |Set-AzureRmLoadBalancer

$nic2=$nic2 | Set-AzureRmNetworkInterface

$nic5=$nic5 | Set-AzureRmNetworkInterface
```

NetScaler VPX ペアを正常に展開したら、各 VPX インスタンスにログオンして HA-INC、SNIP アドレス、および VIP アドレスを構成します。

1. 次のコマンドを入力して HA ノードを追加します。

   ```
   add ha node 1 PeerNodeNSIP -inc Enabled
   ```

2. クライアント側 NIC のプライベート IP アドレスを VPX1（NIC2）および VPX2（NIC5）の SNIP として追加する

   ```
   add nsip privateIPofNIC2 255.255.255.0 -type SNIP
   add nsip privateIPofNIC5 255.255.255.0 -type SNIP
   ```

3. ALB のフロントエンド IP アドレス (パブリック IP) を持つプライマリノードに負荷分散仮想サーバーを追加します。

   ```
   add lb virtual server v1 HTTP FrontEndIPofALB 80
   ```


関連リソース:

Azure でのアクティブスタンバイ HA デプロイメントでの GSLB の構成

フローティング **IP** 無効モードの **ALB** を使用して **Azure** に **NetScaler ADC** 高可用性ペアをデプロイする

October 25, 2023

Azure のアクティブ/パッシブ高可用性（HA）セットアップで、複数の NIC を持つ一対の NetScaler ADC VPX インスタンスを展開できます。各 NIC には多数の IP アドレスを含めることができます。

アクティブ/パッシブ展開では以下が必要です。

- HA Independent Network Configuration（INC）構成
- Azure Load Balancer (ALB) には次の機能があります。
  - フローティング IP 対応モードまたはダイレクトサーバーリターン (DSR) モード
  - フローティング IP 無効モード

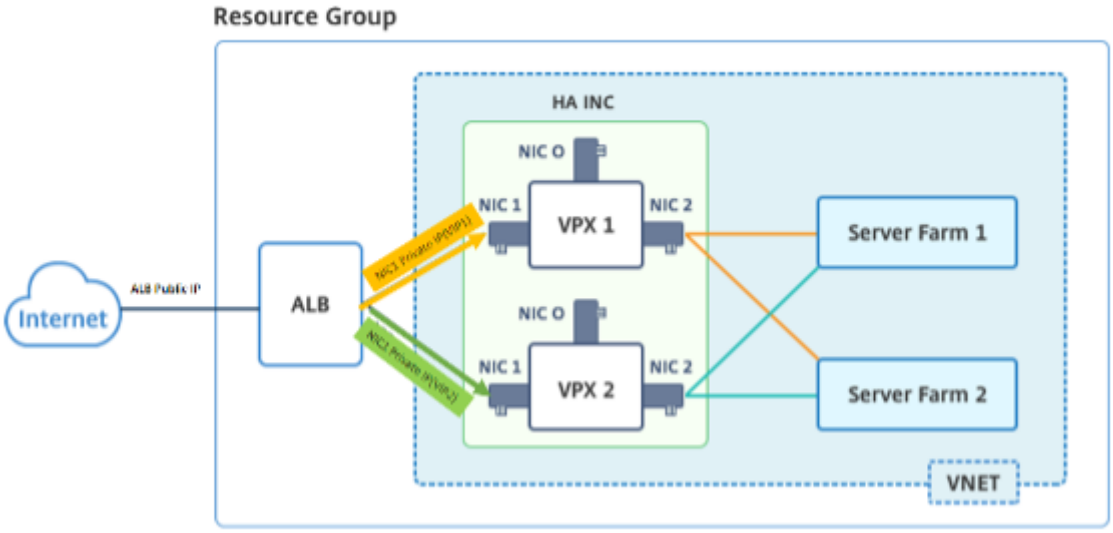
ALB Floating IP オプションの詳細については、Azure のドキュメントを参照してください。

ALB Floating IP が有効になっている Azure のアクティブ/パッシブ HA セットアップで VPX ペアをデプロイする場合は、「PowerShell コマンドを使用して複数の IP アドレスと NIC を使用して高可用性セットアップを構成する」を参照してください。

フローティング **IP** 無効モードの **ALB** を使用した **HA** 導入アーキテクチャ

アクティブ/パッシブ展開では、各インスタンスのクライアントインターフェイスのプライベート IP アドレスが各 VPX インスタンスの VIP アドレスとして追加されます。HA-INC モードで、IP セットを使用して VIP アドレスを共有し、SNIP アドレスをインスタンス固有に設定します。すべてのトラフィックはプライマリインスタンスを通過します。セカンダリインスタンスは、プライマリインスタンスに障害が発生するまでスタンバイモードです。

図: アクティブ-パッシブ展開アーキテクチャの例



前提条件

NetScaler VPX インスタンスを Azure に展開する前に、次の情報を理解している必要があります。

- Azure の用語とネットワークの詳細。詳細については、「Azure の用語」を参照してください。
- NetScaler ADC アプライアンスの動作。詳しくは、NetScaler のドキュメントを参照してください。
- NetScaler ネットワーキング。詳細については、ADC ネットワークを参照してください。
- Azure ロードバランサーと負荷分散ルール設定。詳細については、Azure ALB のドキュメントを参照してください。

ALB フローティング **IP** を無効にして **VPX HA** ペアを **Azure** にデプロイする方法

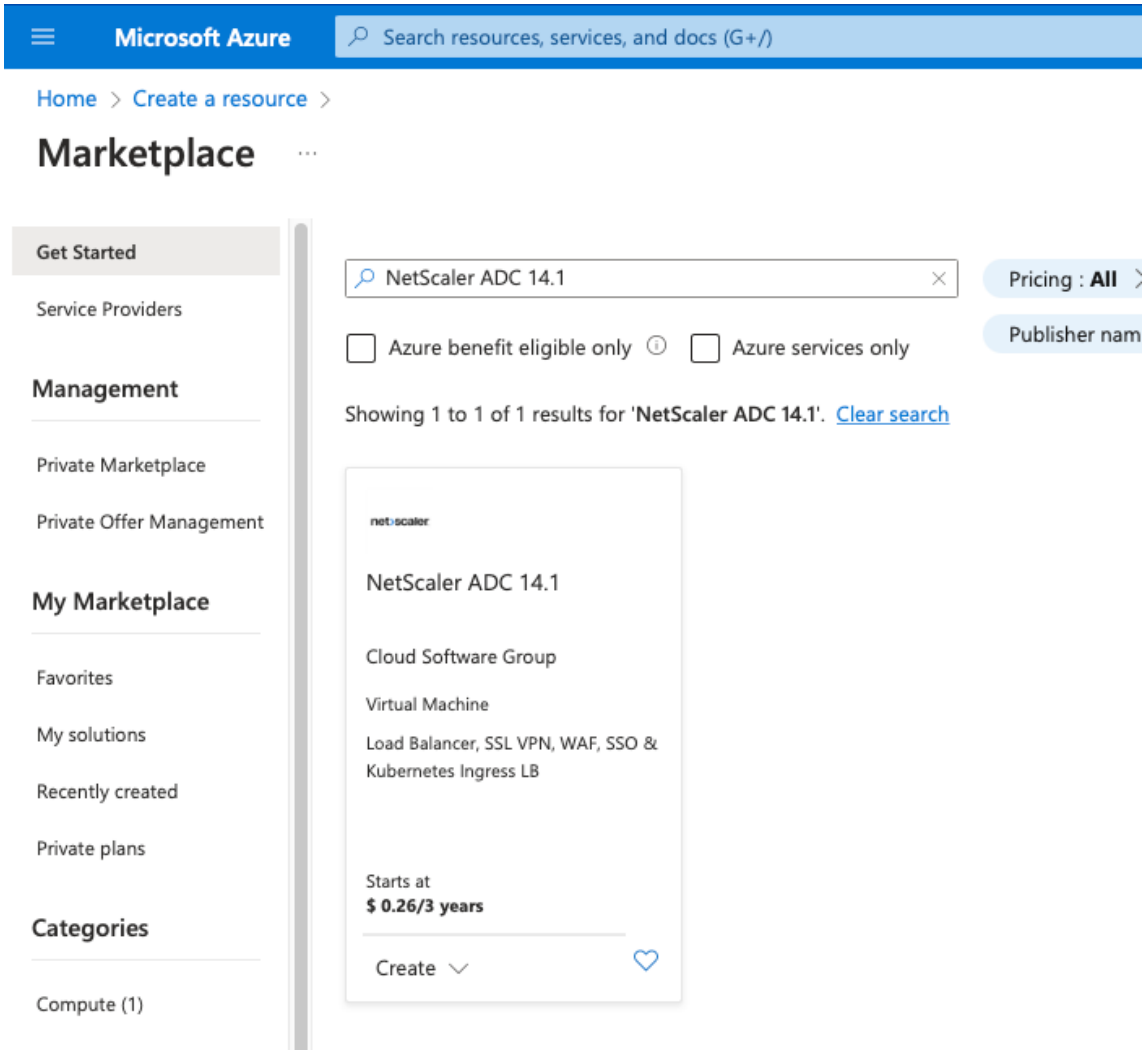
HA と ALB の導入手順の概要は次のとおりです。

1. Azure に 2 つの VPX インスタンス (プライマリインスタンスとセカンダリインスタンス) をデプロイします。
2. 両方のインスタンスにクライアントとサーバーの NIC を追加します。
3. フローティング IP モードが無効になっている負荷分散ルールを持つ ALB をデプロイします。
4. NetScaler GUI を使用して、両方のインスタンスで高可用性設定を構成します。

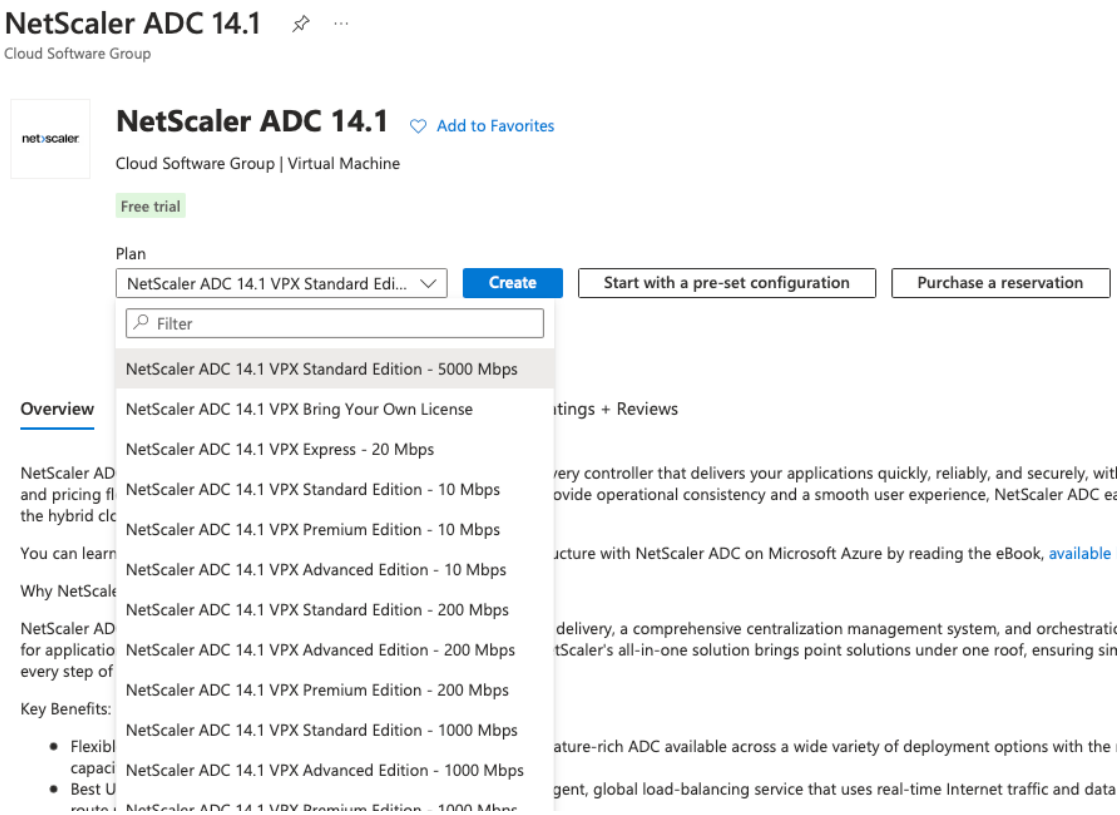
手順 **1. Azure** に **2** つの **VPX** インスタンスをデプロイします。

次の手順に従って、2 つの VPX インスタンスを作成します。

1. Azure Marketplace から NetScaler ADC バージョンを選択します（この例では、NetScaler ADC リリース 13.1 が使用されています）。

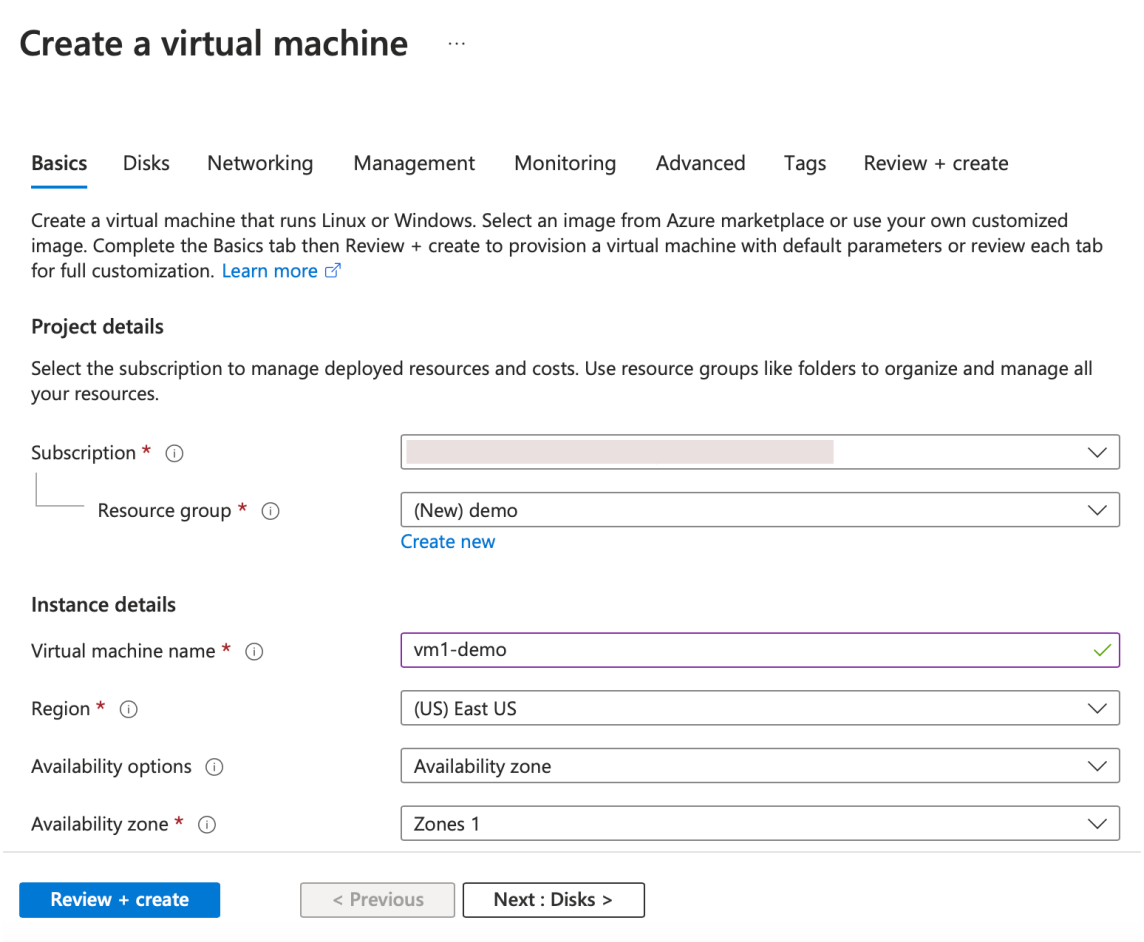


2. 必要な ADC ライセンスモードを選択し、[作成] をクリックします。



[仮想マシンの作成] ページが開きます。

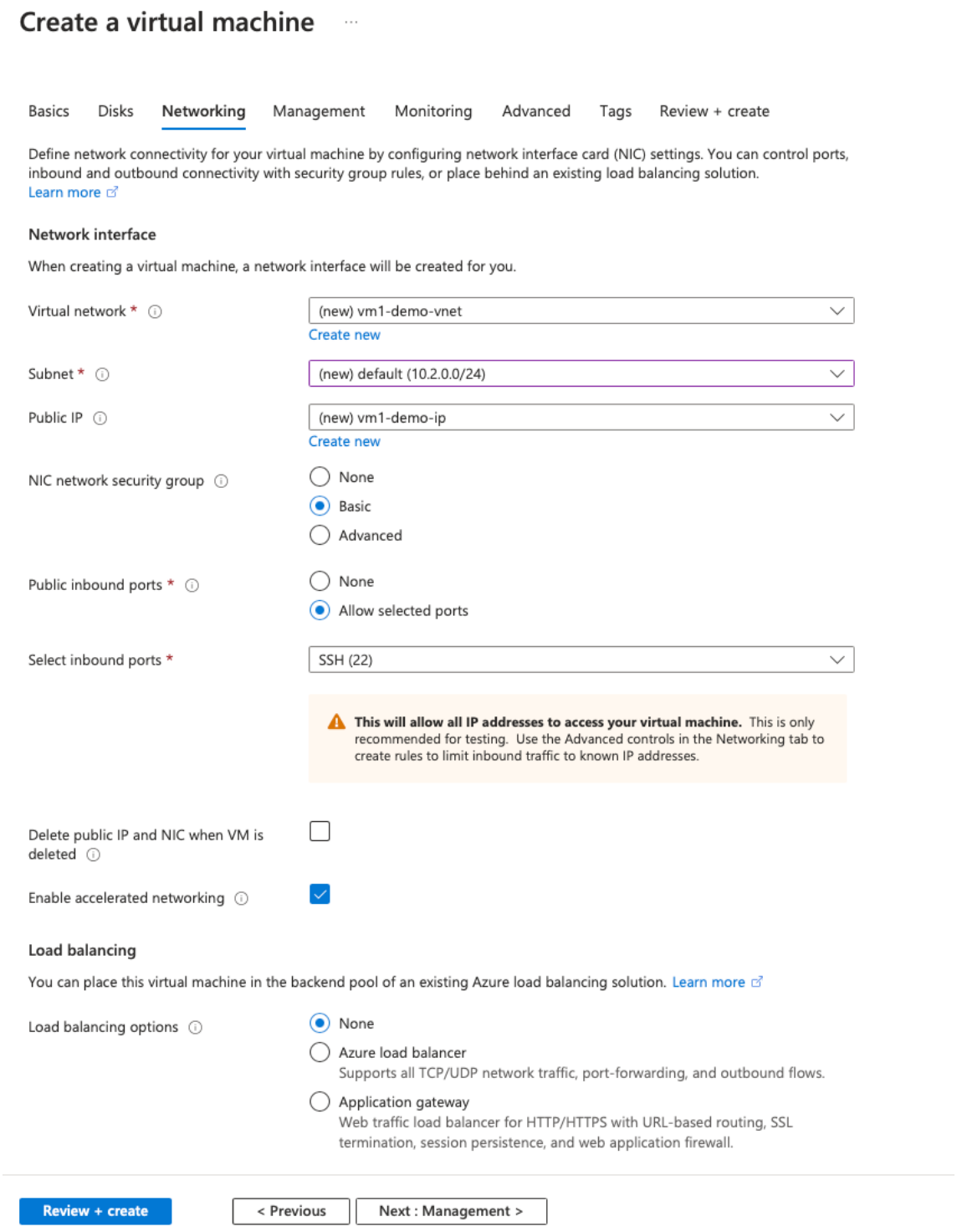
3. 導入を成功させるには、[基本]、[ディスク]、[ネットワーク]、[管理]、[監視]、[詳細]、[タグ] の各タブに必要な詳細情報を入力します。



4. [ネットワーク] タブで、管理、クライアント、サーバーの NIC の 3 つのサブネットを持つ新しい仮想ネットワークを作成します。それ以外の場合は、既存の仮想ネットワークを使用することもできます。管理 NIC は、VM の展開中に作成されます。クライアントとサーバーの NIC は、仮想マシンの作成後に作成および接続されます。NIC ネットワークセキュリティグループでは、次のいずれかを実行できます。

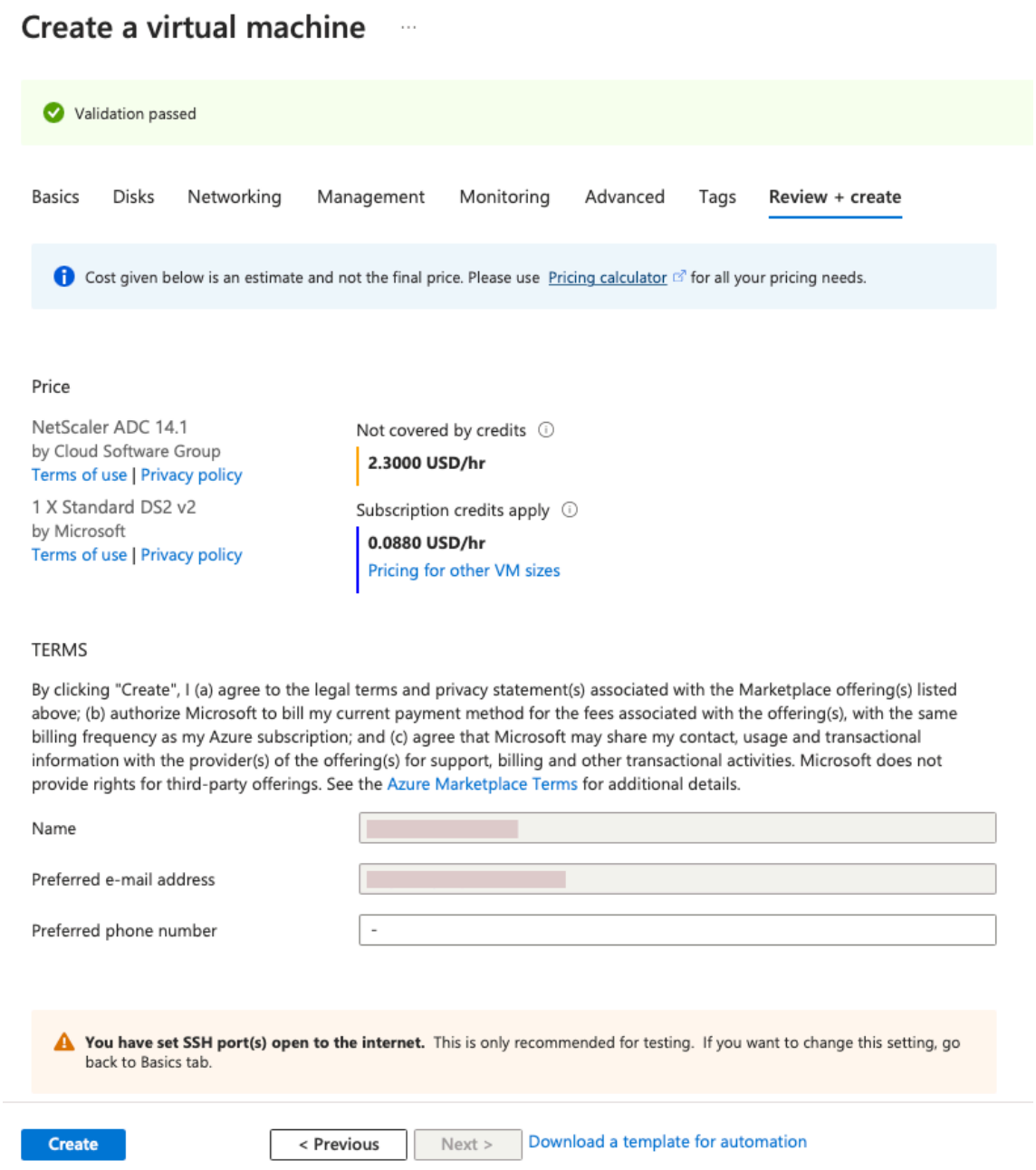
   - [詳細] を選択し、要件に合った既存のネットワークセキュリティグループを使用します。
   - [基本] を選択し、必要なポートを選択します。

   > 注:
   >
   > 仮想マシンのデプロイが完了した後に、ネットワークセキュリティグループの設定を変更することもできます。

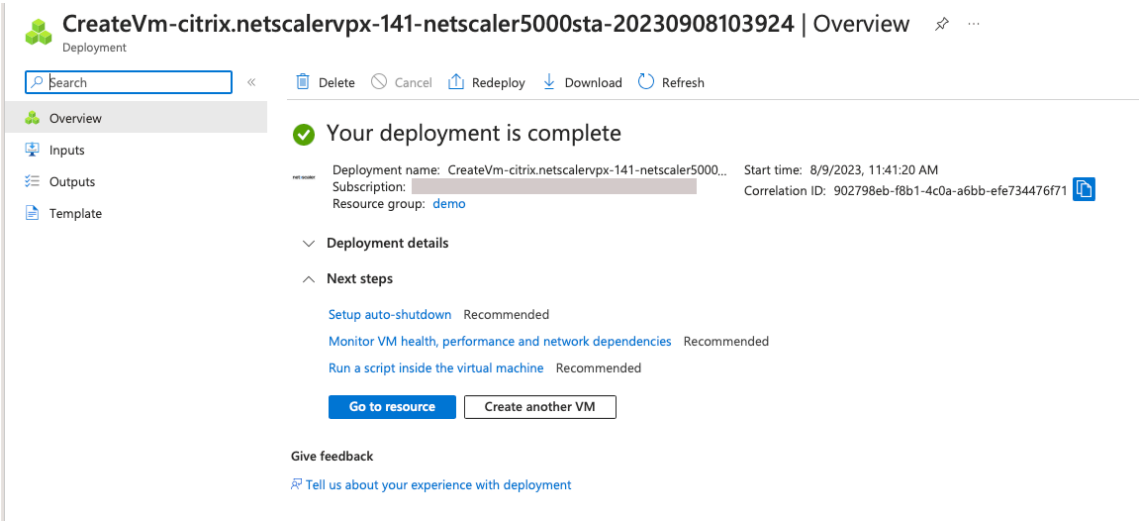


5. [次へ: ** 確認 + 作成 **] をクリックします。

   検証が成功したら、基本設定、仮想マシンの構成、ネットワーク、および追加設定を確認し、[作成] をクリックします。



6. デプロイが完了したら、[**Go to Resource**] をクリックして設定の詳細を表示します。



同様に、2 つ目の NetScaler ADC VPX インスタンスを展開します。

手順 **2.** クライアントとサーバーの **NIC** を両方のインスタンスに追加します。

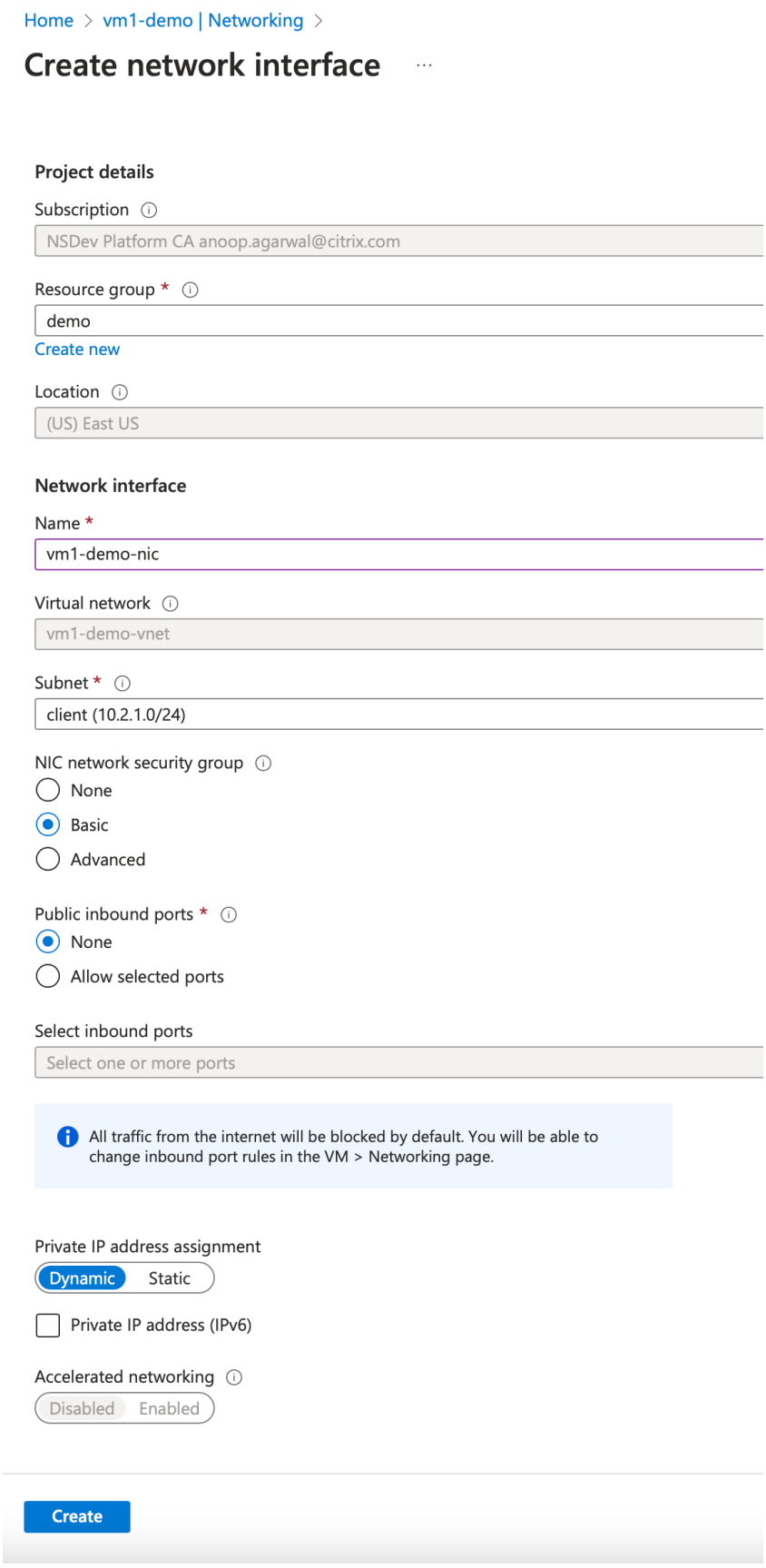
> 注:
>
> さらに NIC を接続するには、まず仮想マシンを停止する必要があります。Azure ポータルで、停止する VM を選択します。[概要] タブで、[停止] をクリックします。ステータスが [停止] と表示されるまで待ちます。

プライマリインスタンスにクライアント NIC を追加するには、次の手順に従います。

1. [ネットワーク**] > [**ネットワークインターフェイスの接続] に移動します。

   既存の NIC を選択するか、新しいインターフェイスを作成して接続できます。

2. NIC ネットワークセキュリティグループについては、[詳細] を選択して既存のネットワークセキュリティグループを使用するか、[基本] を選択して作成できます。



サーバ NIC を追加するには、クライアント NIC を追加する場合と同じ手順に従います。

NetScaler VPX インスタンスには、3 つの NIC（管理 NIC、クライアント NIC、およびサーバー NIC）がすべて接続されています。

前の手順を繰り返して、セカンダリインスタンスに NIC を追加します。

両方のインスタンスで NIC を作成してアタッチしたら、[**Overview] > [Start**] に移動して両方のインスタンスを再起動します。

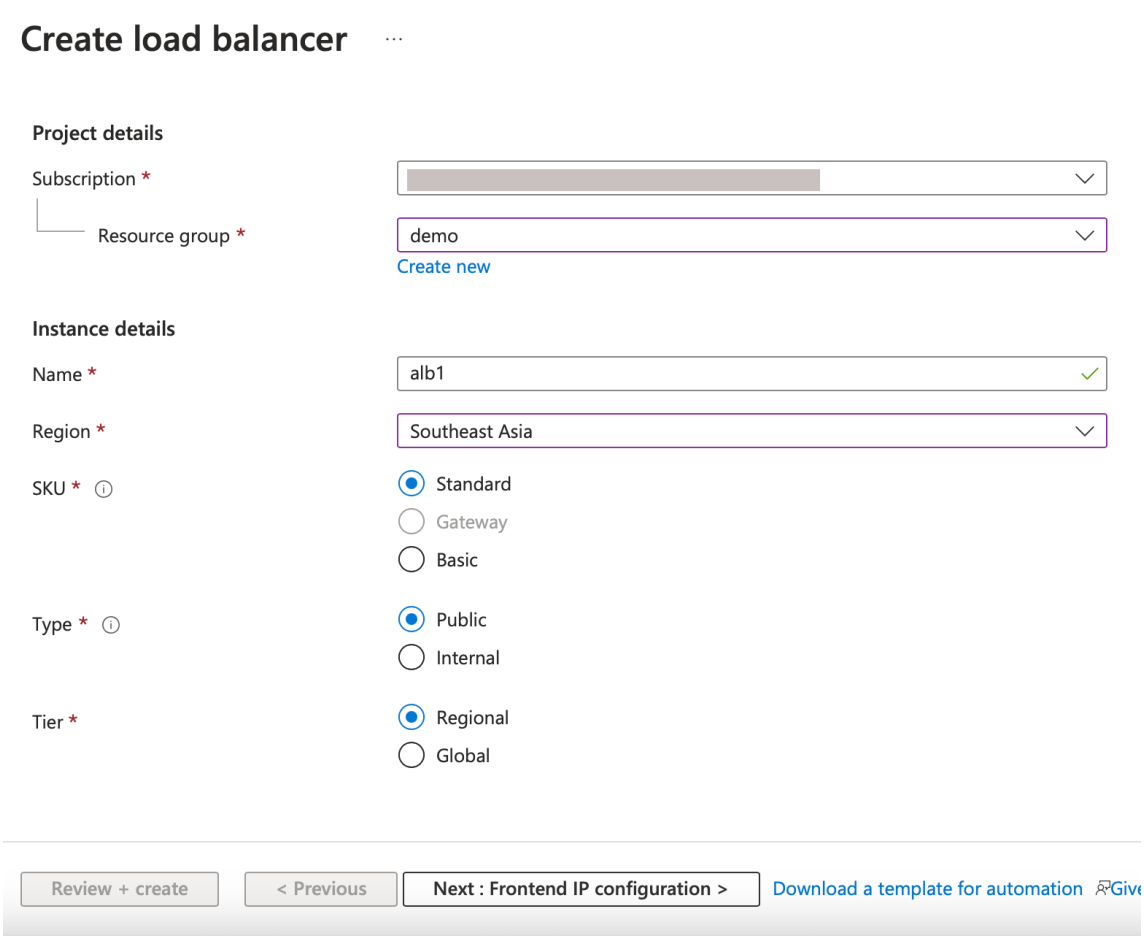
> 注:
>
> クライアント NIC インバウンドルールでは、ポートを通過するトラフィックを許可する必要があります。このルールは、後で NetScaler ADC VPX インスタンスの構成時に負荷分散仮想サーバーを作成するために使用されます。

手順 **3.** フローティング **IP** モードが無効になっている負荷分散ルールを持つ **ALB** をデプロイします。

ALB の設定を開始するには、次の手順に従います。

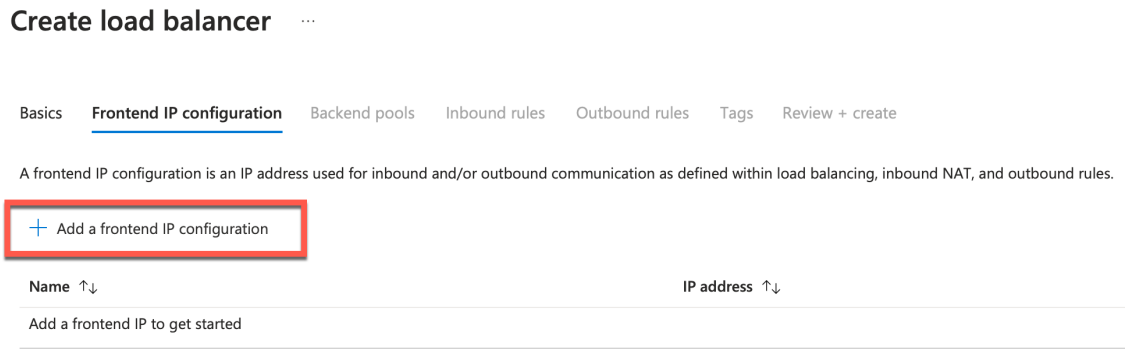
1. [ロードバランサー] ページに移動し、[作成] をクリックします。
2. **[**ロードバランサーの作成] ページで、必要に応じて詳細を入力します。

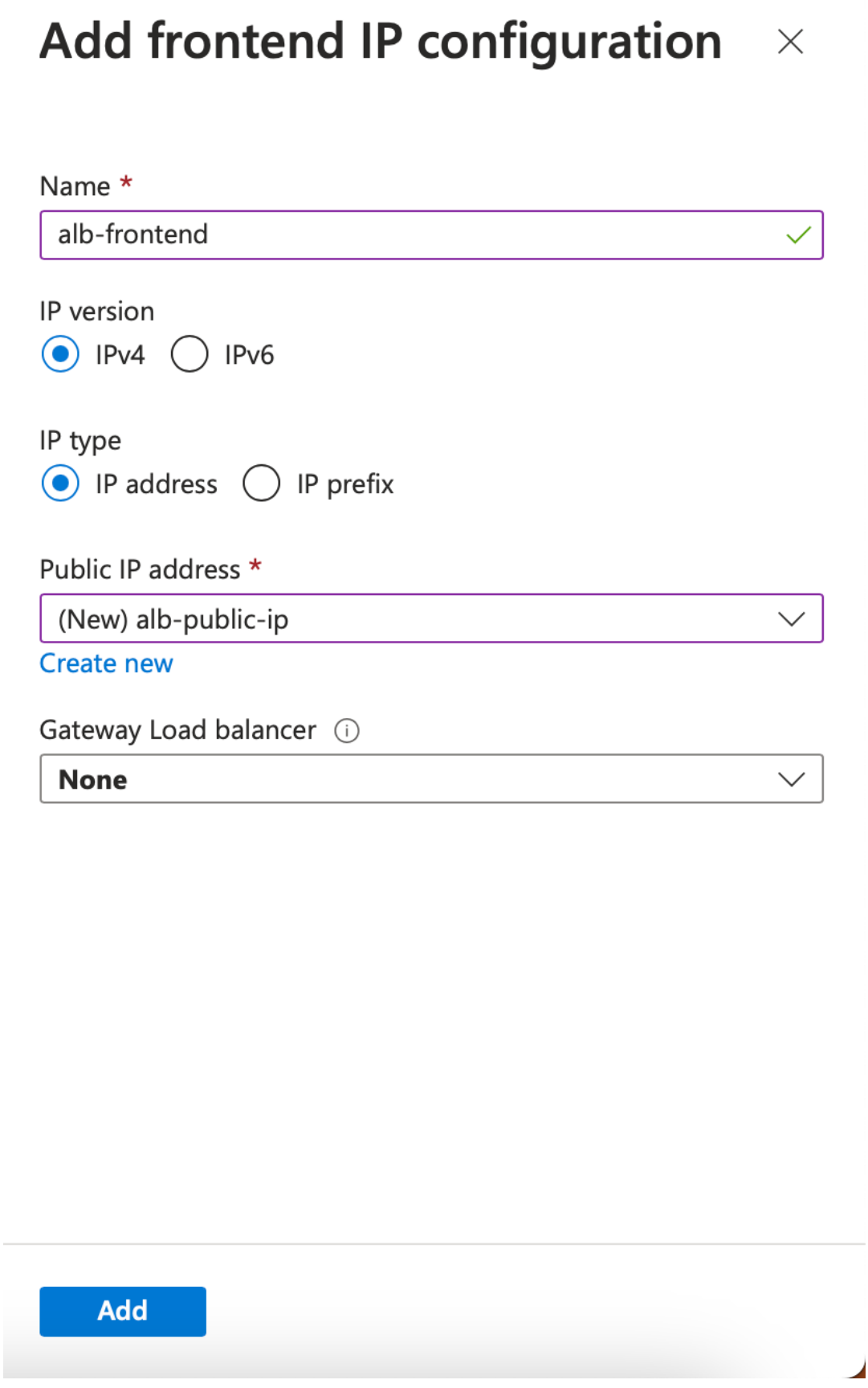
   次の例では、Standard SKU のリージョンパブリックロードバランサーをデプロイします。



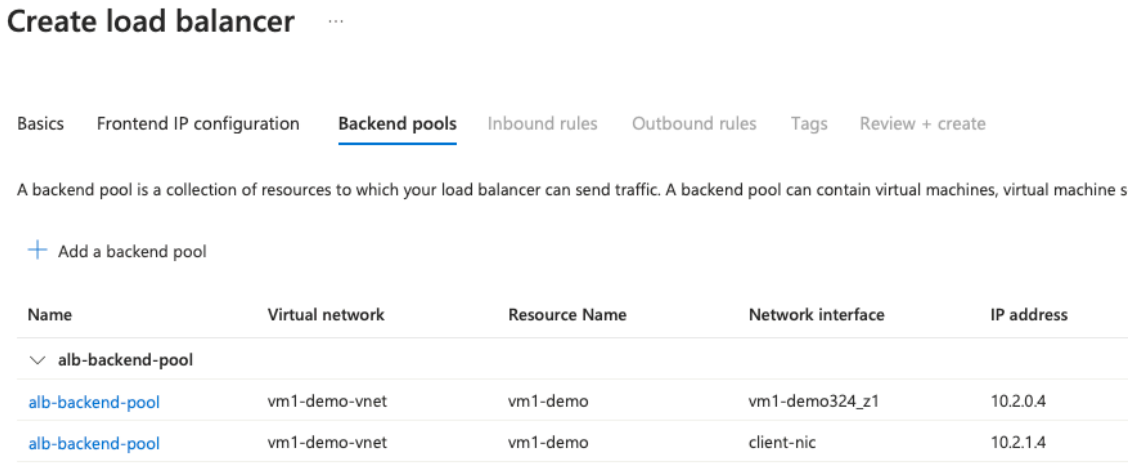
> 注:
>
> NetScaler ADC 仮想マシンに接続されているすべてのパブリック IP は、ALB の SKU と同じ SKU を持つ必要があります。ALB SKU の詳細については、Azure Load Balancer SKU のドキュメントを参照してください。

3. [フロントエンド **IP** 設定] タブで、IP アドレスを作成するか、既存の IP アドレスを使用します。

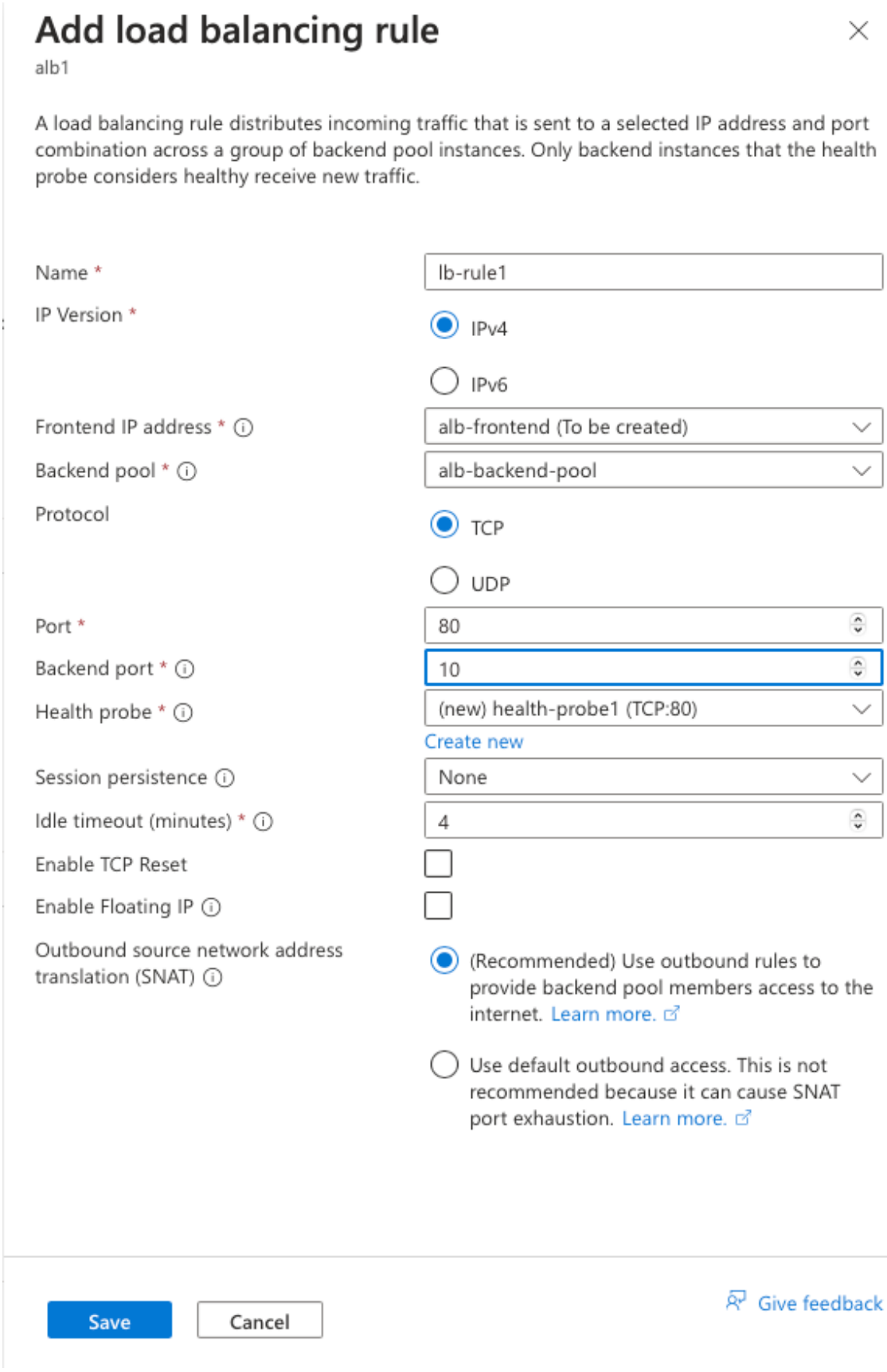




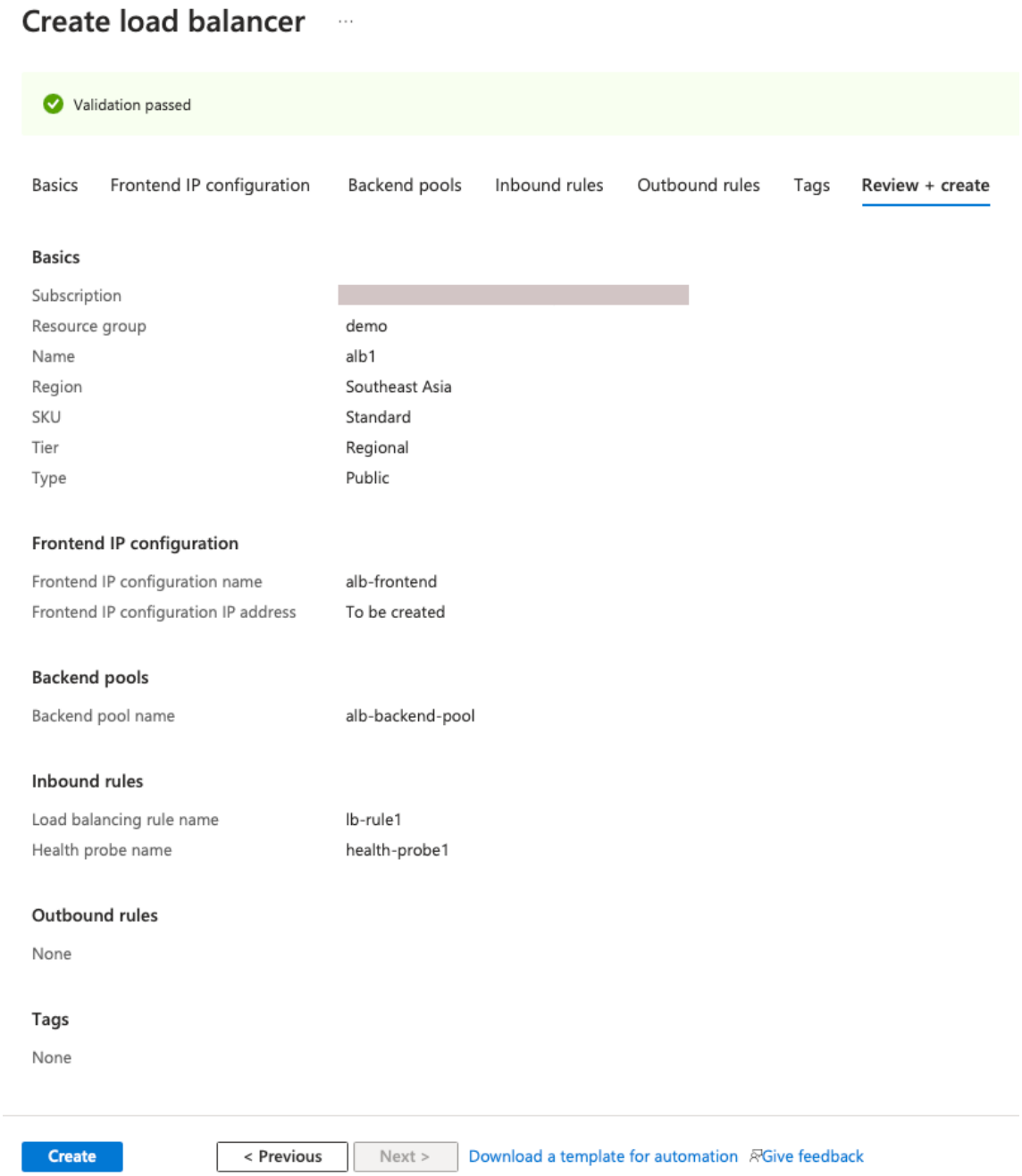
4. ［バックエンドプール］タブで、［NIC ベースのバックエンドプール構成］を選択し、両方の NetScaler ADC 仮想マシンのクライアント NIC を追加します。



5. [受信ルール] タブで、[負荷分散ルールの追加] をクリックし、前の手順で作成したフロントエンド IP アドレスとバックエンドプールを指定します。要件に基づいてプロトコルとポートを選択します。ヘルスプローブを作成するか、既存のヘルスプローブを使用します。フローティング IP オプションは [無効] に設定する必要があります。



6. [レビュー**] + [**作成] をクリックします。検証に合格したら、[作成] をクリックします。



ステップ **4: NetScaler GUI** を使用して、両方の **NetScaler ADC VPX** インスタンスの **HA** 設定を構成します。

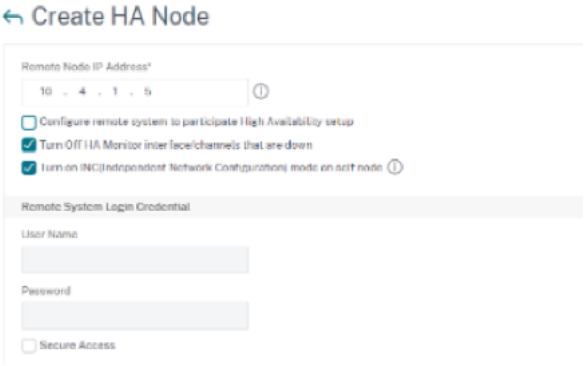
Azure で NetScaler ADC VPX インスタンスを作成したら、NetScaler GUI を使用して HA を構成できます。

手順 **1.** 両方のインスタンスで **INC** モードで高可用性をセットアップします。

プライマリ・インスタンスで、次の手順を実行します。

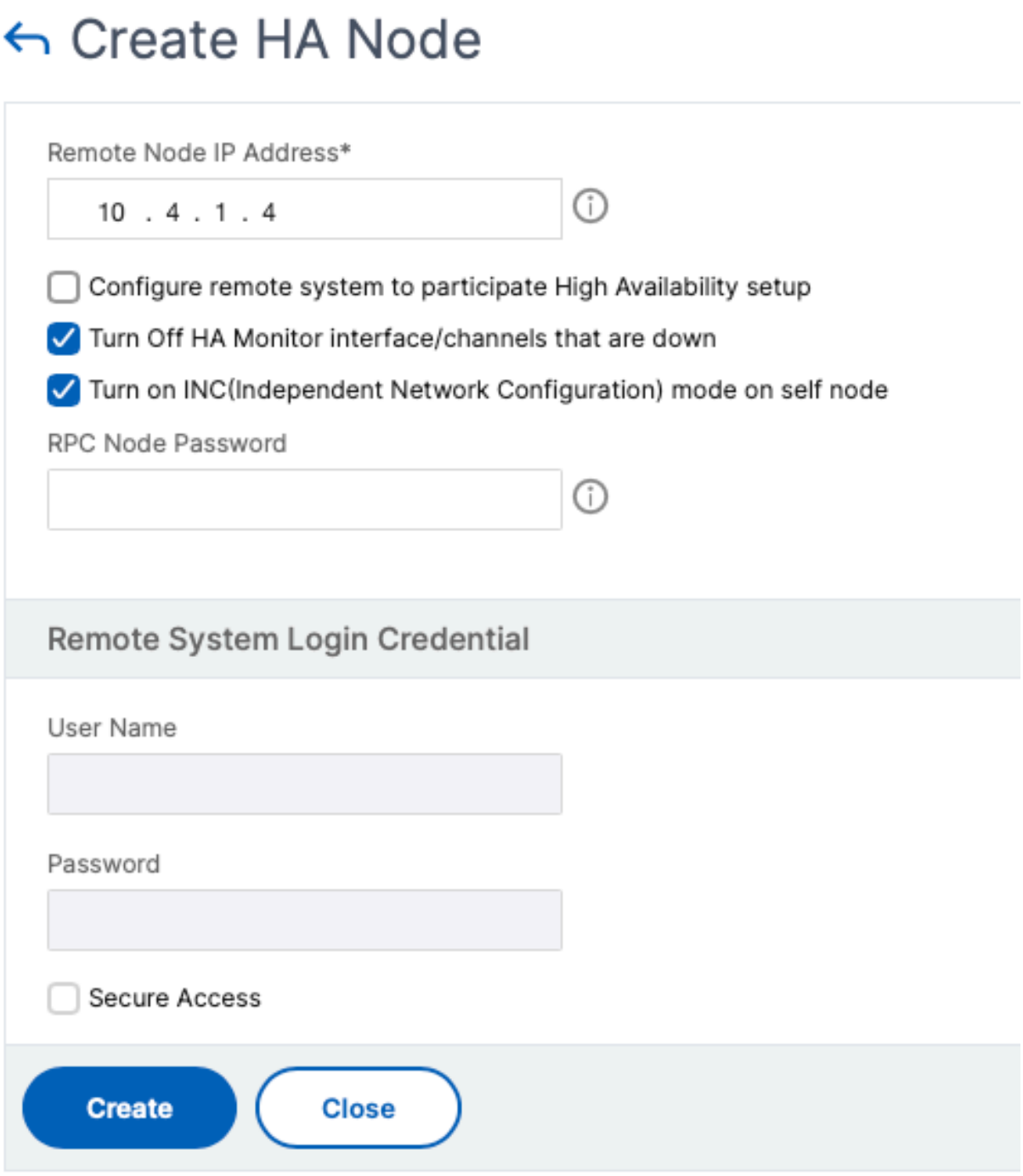
1. インスタンスのデプロイ時に指定したユーザー名`nsroot`とパスワードを使用して、インスタンスにログオンします。

2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノードの **IP** アドレス] フィールドに、セカンダリインスタンスの管理 NIC のプライベート IP アドレス（例：10.4.1.5）を入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. ［作成］をクリックします。



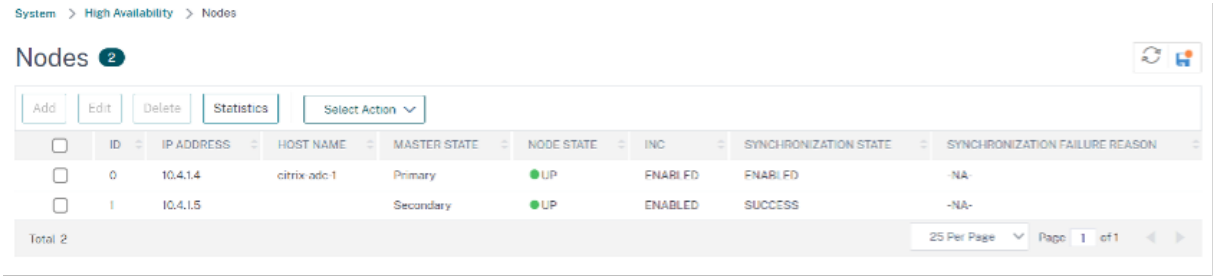
セカンダリインスタンスで、次の手順を実行します。

1. インスタンスのデプロイ時に指定したユーザー名nsrootとパスワードを使用して、インスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノードの **IP** アドレス] フィールドに、プライマリインスタンスの管理 NIC のプライベート IP アドレス（例：10.4.1.4）を入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. ［作成］をクリックします。



先に進む前に、セカンダリインスタンスの同期状態が [ノード] ページで [**SUCCESS**] と表示されていることを確認します。

> 注:
>
> これで、セカンダリインスタンスはプライマリインスタンスと同じログオン認証情報を持つようになりました。



手順 **2.** 両方のインスタンスに仮想 **IP** アドレスとサブネット **IP** アドレスを追加します。

プライマリ・インスタンスで、次の手順を実行します。

1. [システム] **>** [ネットワーク] **>** [**IP**] **>** [**IPv4**] に移動し、[追加] をクリックします。
2. 次の手順に従って、プライマリ VIP アドレスを追加します。

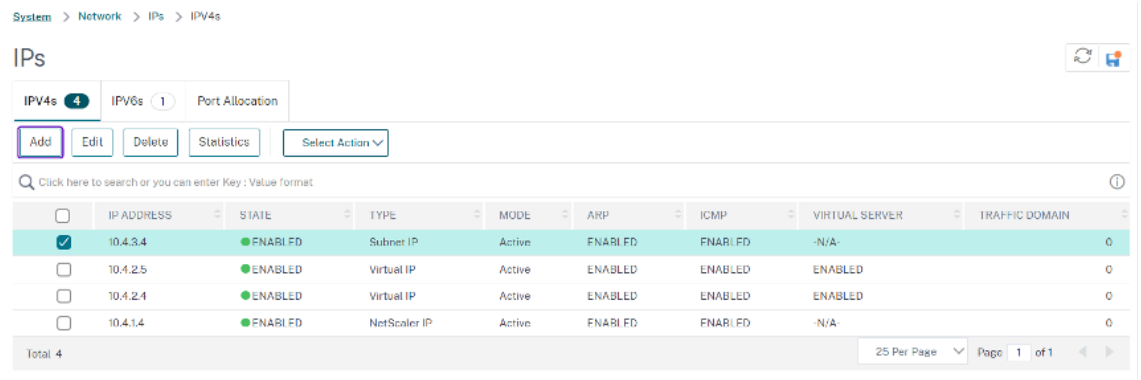
   a) プライマリインスタンスのクライアント NIC のプライベート IP アドレスと、VM インスタンスのクライアントサブネットに対して構成されたネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
   c) [作成] をクリックします。

3. 次の手順に従って、プライマリ SNIP アドレスを追加します。

   a) プライマリ・インスタンスのサーバ NIC の内部 IP アドレスと、プライマリ・インスタンスのサーバ・サブネットに対して構成されているネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
   c) [作成] をクリックします。

4. 次の手順に従って、セカンダリ VIP アドレスを追加します。

   a) セカンダリインスタンスのクライアント NIC の内部 IP アドレスと、VM インスタンスのクライアントサブネットに設定されているネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
   c) [作成] をクリックします。



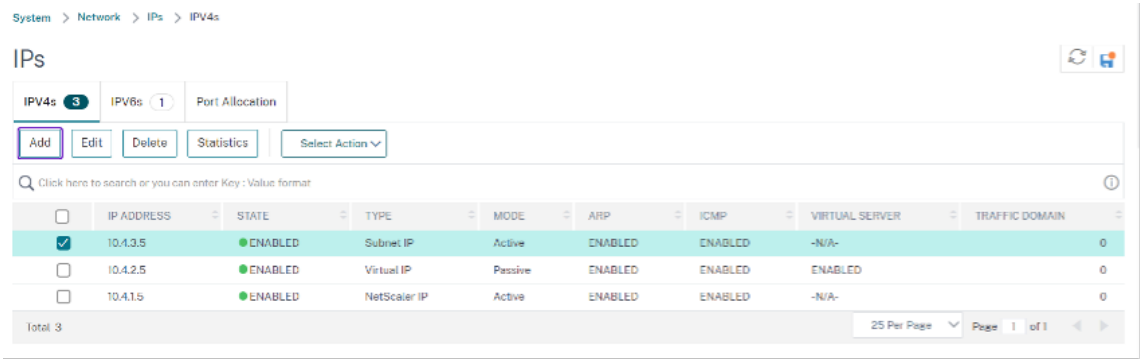
セカンダリインスタンスで、次の手順を実行します。

1. [システム] **>** [ネットワーク] **>** **[IP] > [IPv4]** に移動し、[追加] をクリックします。
2. 次の手順に従って、セカンダリ VIP アドレスを追加します。

   a) セカンダリインスタンスのクライアント NIC の内部 IP アドレスと、VM インスタンスのクライアントサブネットに設定されているネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。

3. 次の手順に従って、セカンダリ SNIP アドレスを追加します。

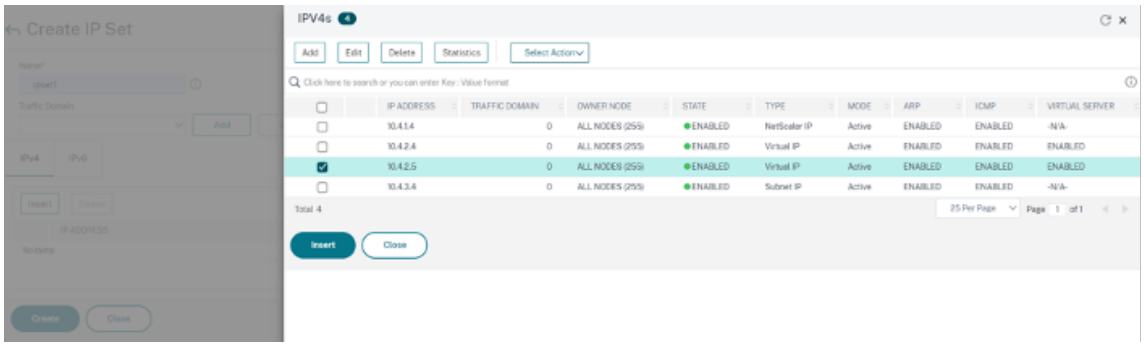
   a) セカンダリインスタンスのサーバ NIC の内部 IP アドレスと、セカンダリインスタンスのサーバサブネットに設定されているネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
   c) ［作成］をクリックします。



手順 **3. IP** セットを追加し、両方のインスタンスで **IP** セットをセカンダリ **VIP** にバインドします。

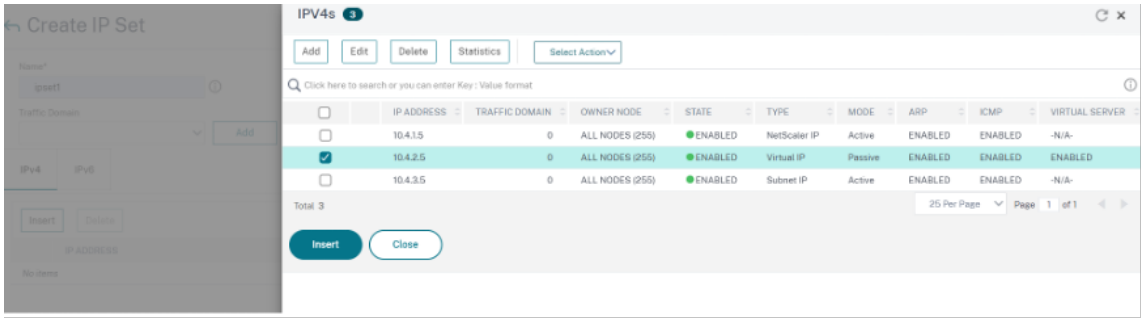
プライマリ・インスタンスで、次の手順を実行します。

1. **System > Network > IP Sets > Add** に移動します。
2. IP セット名を追加し、[**Insert**] をクリックします。
3. [**IPv4**] ページで、仮想 IP（セカンダリ VIP）を選択し、[挿入] をクリックします。
4. [**Create**] をクリックして IP セットを作成します。



セカンダリインスタンスで、次の手順を実行します。

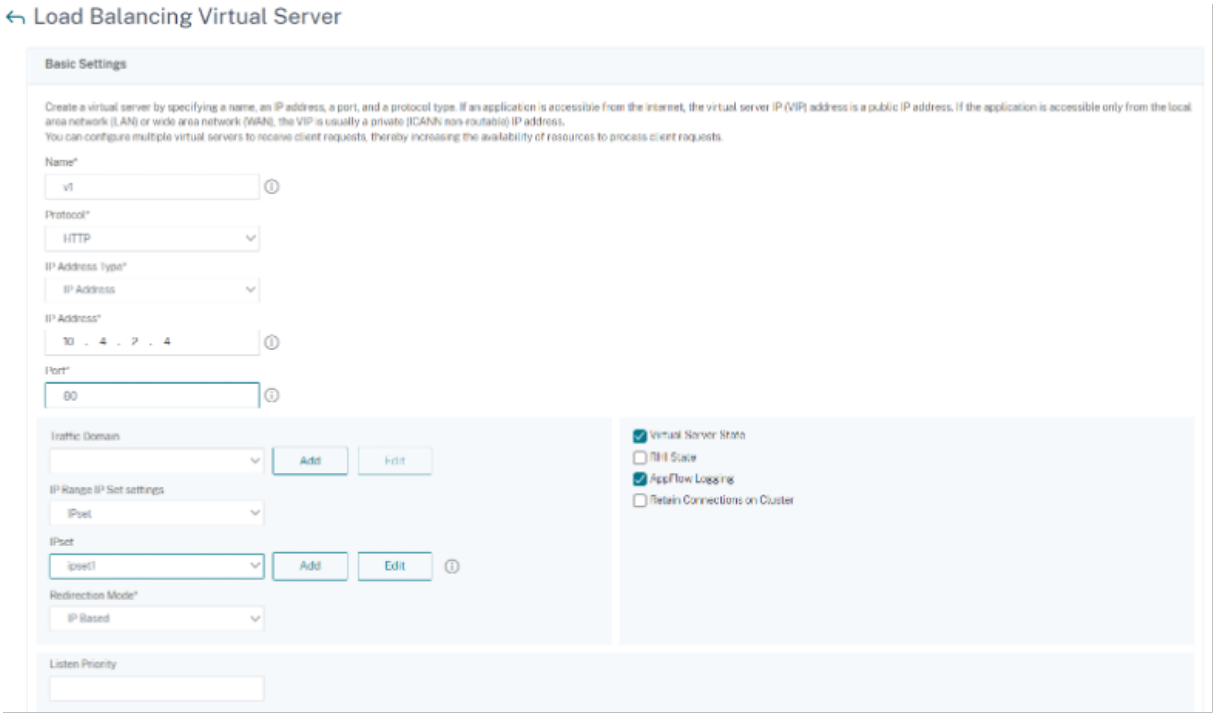
1. **System > Network > IP Sets > Add** に移動します。
2. IP セット名を追加し、[**Insert**] をクリックします。
3. [**IPv4s**] ページで、仮想 IP (セカンダリ VIP) を選択し、[挿入] をクリックします。
4. [**Create**] をクリックして IP セットを作成します。



> 注:
>
> IP セット名は、プライマリインスタンスとセカンダリインスタンスの両方で同じである必要があります。

ステップ **4:** プライマリインスタンスに負荷分散仮想サーバーを追加します。

1. [設定] **>** [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] **>** [追加] に移動します。
2. [名前]、[プロトコル]、[IP アドレスタイプ (IP アドレス)]、[IP アドレス] (プライマリ VIP)、および [ポート] に必要な値を追加します。
3. [詳細] クリックします。[**IP** 範囲 **IP** セット設定] に移動し、ドロップダウンメニューから [**IPSet**] を選択し、ステップ **3** で作成した IPSet を指定します。
4. **OK** をクリックして、負荷分散仮想サーバーを作成します。

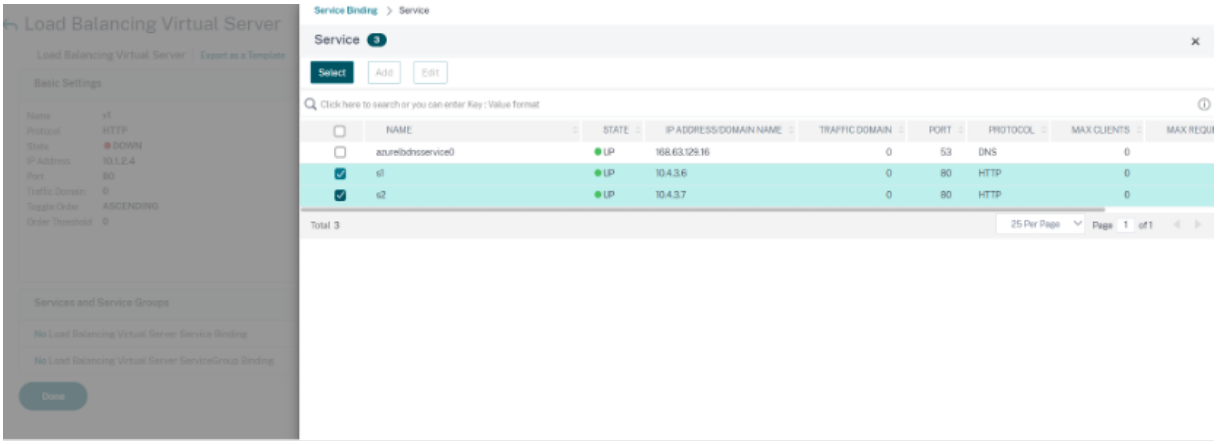


手順 **5.** プライマリインスタンスにサービスまたはサービスグループを追加します。

1. **[設定] > [**トラフィック管理**] > [**負荷分散**] > [**サービス**] >[**追加**]** に移動します。
2. サービス名、IP アドレス、プロトコル、およびポートに必要な値を追加し、[**OK**] をクリックします。

手順 **6**: サービスまたはサービスグループを、プライマリインスタンスの負荷分散仮想サーバにバインドします。

1. **[設定] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動します。
2. 手順 **4** で構成した負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [サービスとサービスグループ] タブで、[負荷分散仮想サーバーサービスバインドなし] をクリックします。
4. 手順 **5** で構成したサービスを選択し、[バインド] をクリックします。



手順 **7.** 構成を保存します。

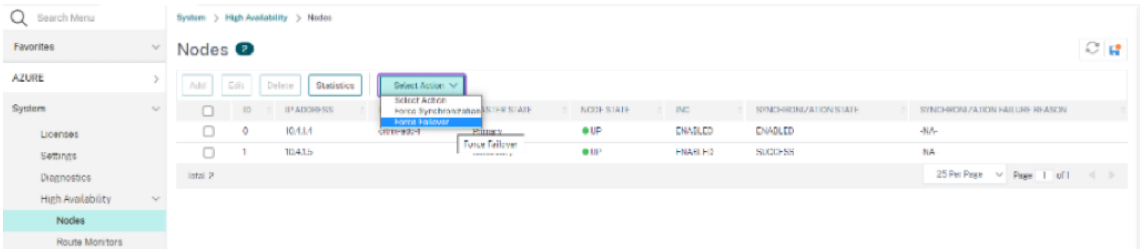
そうしないと、再起動後、または即時再起動が行われた場合、すべての設定が失われます。

手順 **8.** 設定を確認します。

フェイルオーバー後に ALB フロントエンド IP アドレスに到達できることを確認します。

1. ALB フロントエンド IP アドレスをコピーします。
2. IP アドレスをブラウザに貼り付けて、バックエンドサーバーに到達可能であることを確認します。
3. プライマリインスタンスで、フェイルオーバーを実行します。

   NetScaler GUI から、［構成］**>**［システム］**>**［高可用性］**>**［アクション］**>**［強制フェールオーバー］に移動します。



4. フェイルオーバー後、以前に使用した ALB フロントエンド IP を介してバックエンドサーバーにアクセスできることを確認してください。

Azure アクセラレーションネットワークを使用するように **NetScaler VPX** インスタンスを構成する

December 8, 2023

高速ネットワーキングにより、仮想マシンへのシングルルート I/O 仮想化 (SR-IOV) 仮想機能 (VF) NIC が有効になり、ネットワークのパフォーマンスが向上します。この機能は、信頼性の高いストリーミングと低い CPU 使用率でより高いスループットでデータを送受信する必要がある負荷の高いワークロードで使用できます。
NIC が高速ネットワーキングで有効になっている場合、Azure は NIC の既存のパラ仮想化 (PV) インターフェイスと SR-IOV VF インターフェイスをバンドルします。SR-IOV VF インターフェイスのサポートにより、NetScaler VPX インスタンスのスループットが有効になり、向上します。

高速ネットワーキングには、次の利点があります。

- 低レイテンシ
- 1 秒あたりのパケット数 (pps) のパフォーマンスが向上
- スループットの強化
- ジッタの低減
- CPU 使用率の低下

> 注
>
> Azure アクセラレーションネットワーキングは、リリース 13.0 ビルド 76.29 以降の NetScaler VPX インスタンスでサポートされています。

前提条件

- VM のサイズが Azure アクセラレーションネットワーキングの要件と一致していることを確認します。
- 任意の NIC で高速ネットワーキングを有効にする前に、VM（個別または可用性セット内）を停止します。

制限事項

高速ネットワーキングは、一部のインスタンスタイプでのみ有効にできます。詳細については、「サポートされるインスタンスタイプ」を参照してください。

高速ネットワーキングでサポートされる **NIC**

Azure では、ネットワークを高速化するために SR-IOV モードの Mellanox ConnectX3、ConnectX4、および ConnectX5 NIC が提供されています。

NetScaler VPX インターフェイスでアクセラレーテッドネットワーキングが有効になっている場合、Azure は ConnectX3、ConnectX4、または ConnectX5 インターフェイスのいずれかを NetScaler VPX アプライアンスの既存の PV インターフェイスにバンドルします。

> 注
>
> NetScaler VPX は、リリース 13.1 ビルド 37.x 以降の ConnectX5 NIC をサポートしています。

仮想マシンにインターフェイスをアタッチする前に高速ネットワークを有効にする方法の詳細については、「高速ネットワークを使用したネットワークインターフェイスの作成」を参照してください。

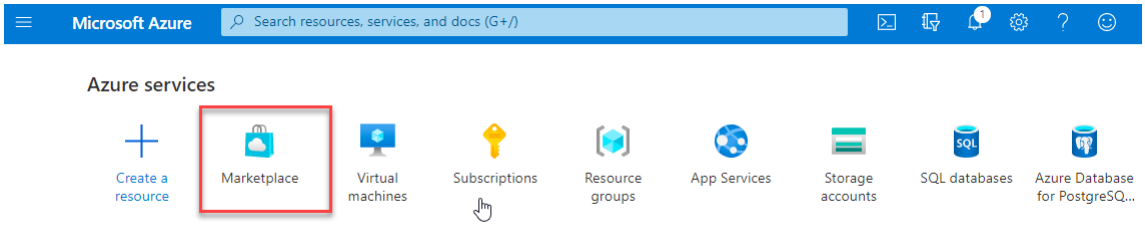
仮想マシンの既存のインターフェイスで高速ネットワーキングを有効にする方法の詳細については、「仮想マシンで既存のインターフェイスを有効にする」を参照してください。

Azure コンソールを使用して **NetScaler VPX** インスタンスで高速ネットワークを有効にする方法

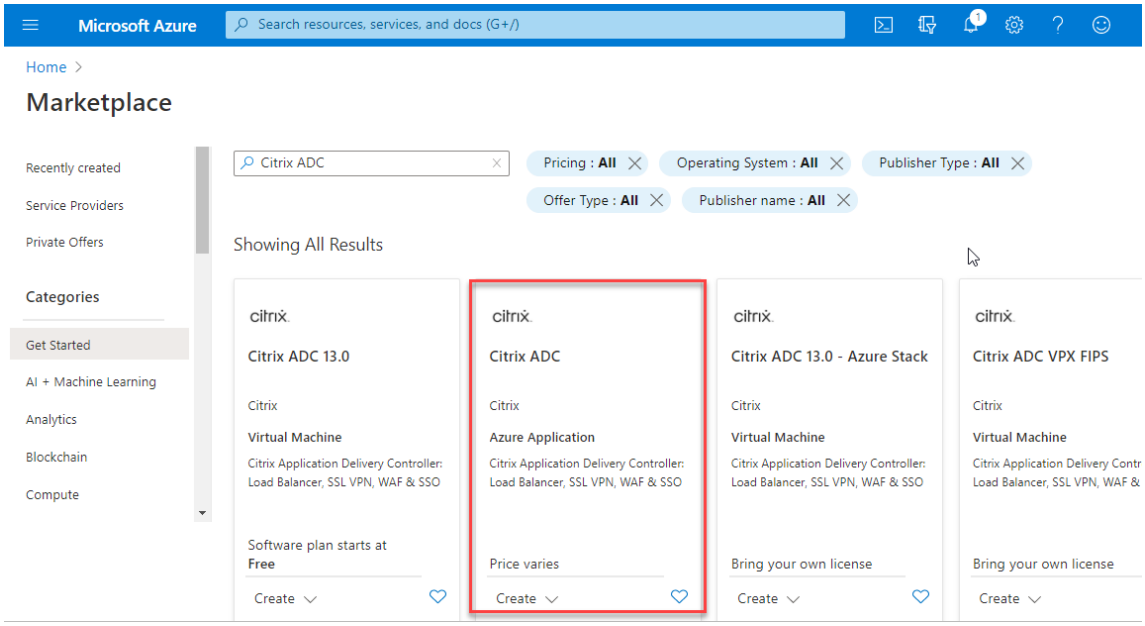
Azure コンソールまたは Azure PowerShell を使用して、特定のインターフェイスで高速ネットワークを有効にできます。

Azure のアベイラビリティセットまたはアベイラビリティーゾーンを使用して高速ネットワークを有効にするには、次の手順を実行します。

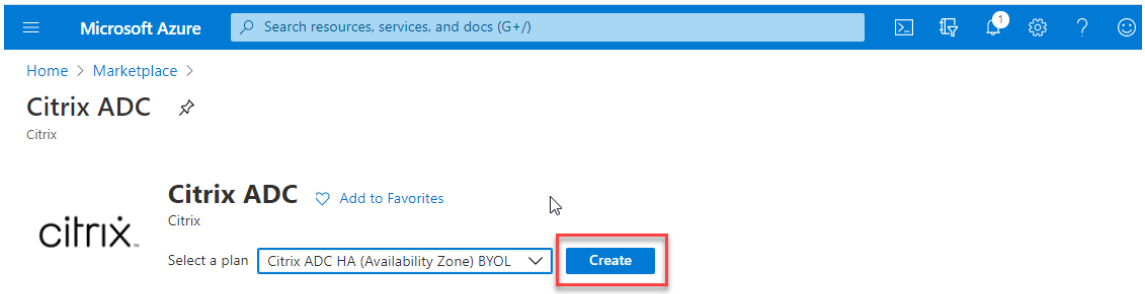
1. Azure ポータルにログインし、Azureマーケットプレイスにナビゲートします。



2. **Azure Marketplace** から **NetScaler** を検索してください。

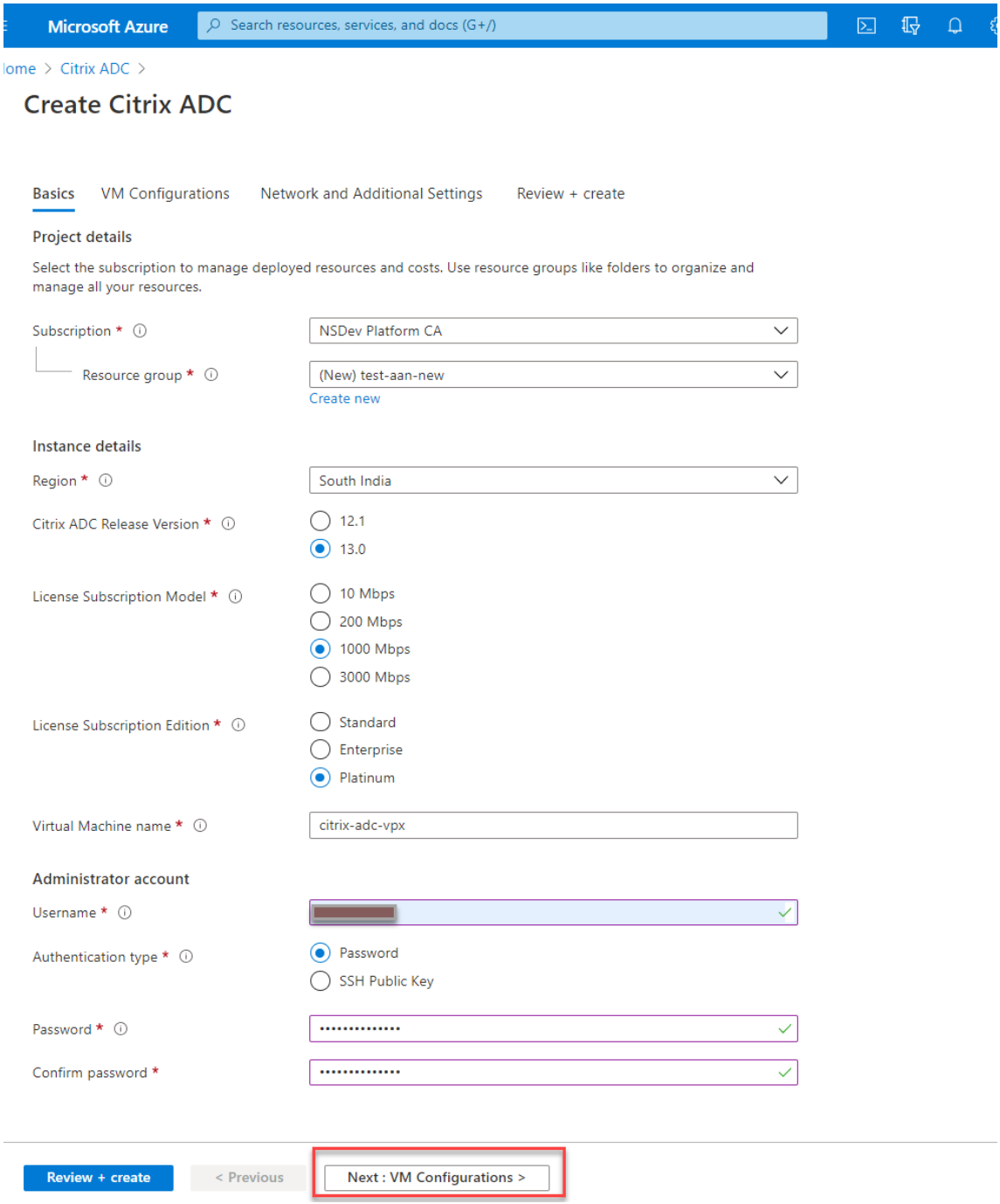


3. ライセンスとともに FIPS 以外の NetScaler プランを選択し、［作成］をクリックします。



［**NetScaler** の作成］ページが表示されます。

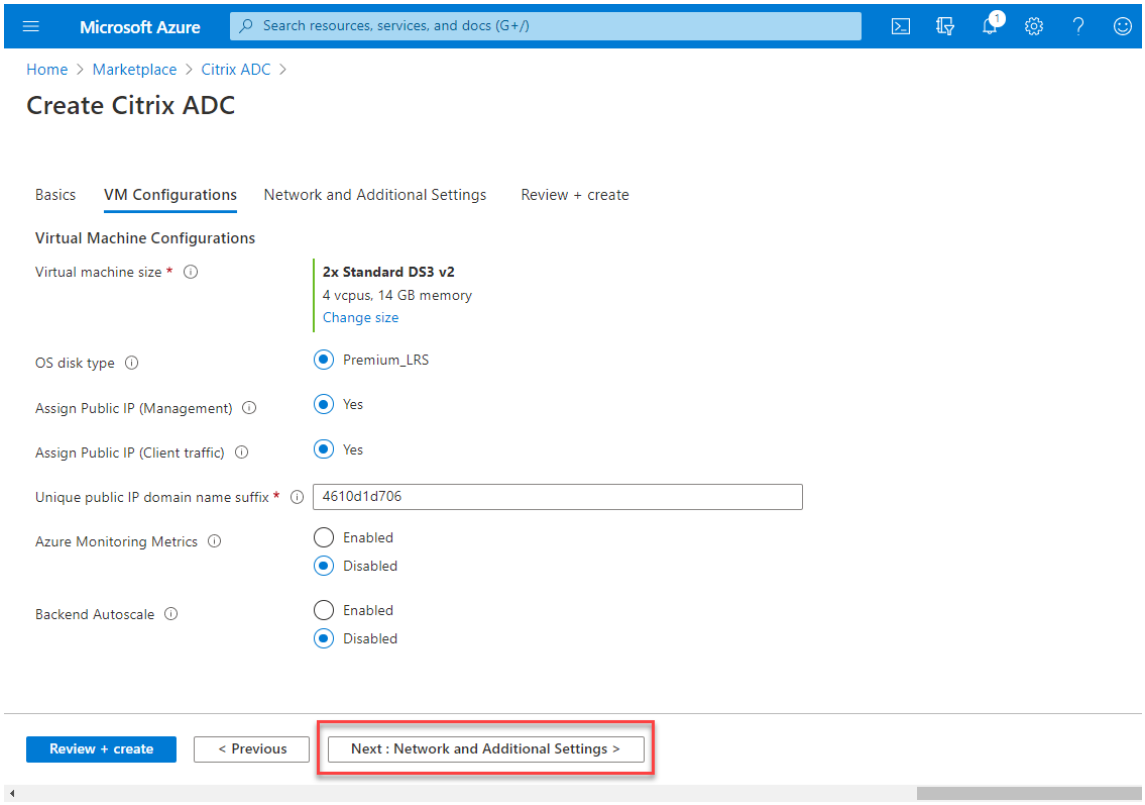
4. [基本] タブで、リソースグループを作成します。[パラメータ] タブで、地域、管理者ユーザー名、管理者パスワード、ライセンスタイプ（VM SKU）、およびその他のフィールドの詳細を入力します。



5. [次へ] をクリックします: VM 構成 > 。

   [**VM** 構成] ページで、次の手順を実行します。

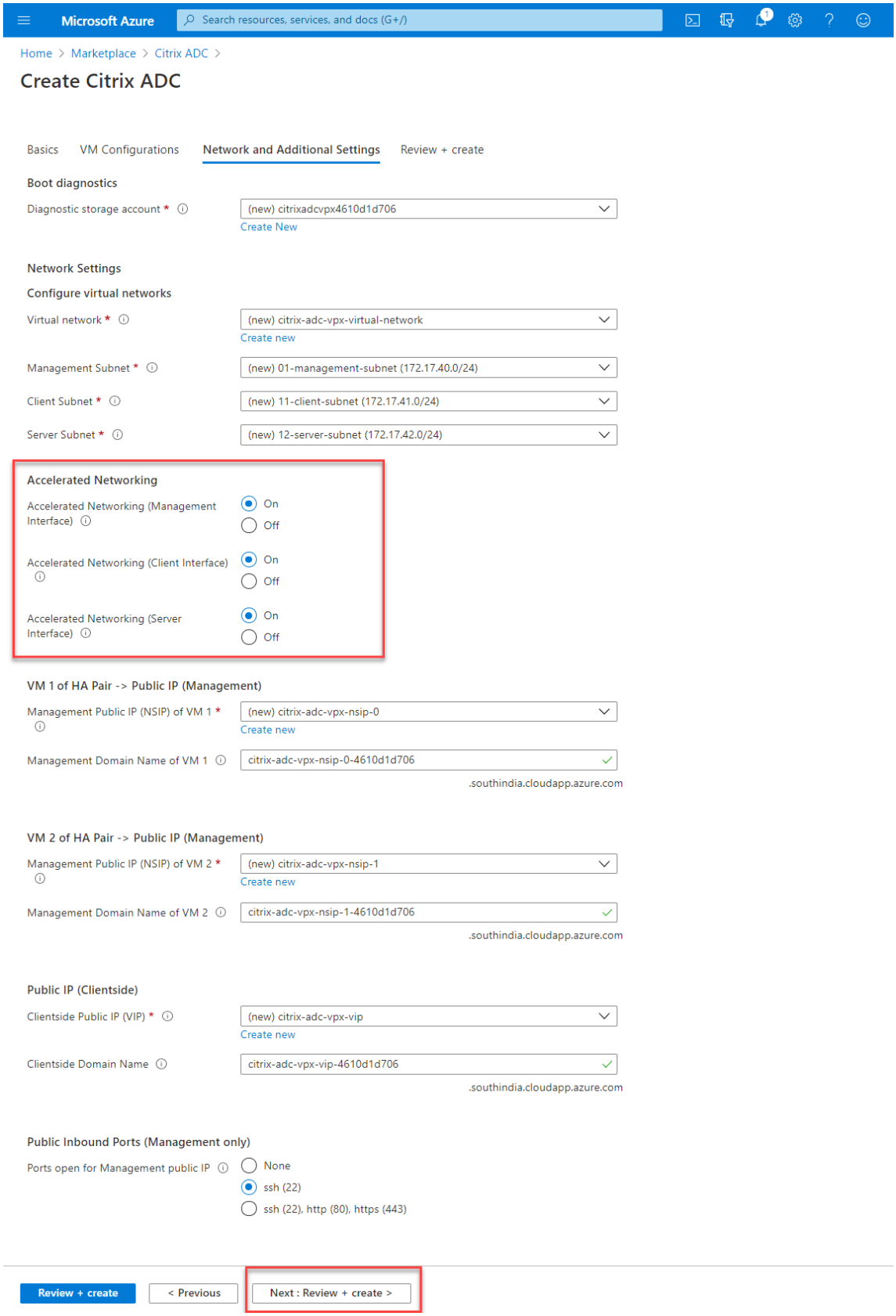
   a) パブリック IP ドメイン名のサフィックスを設定します。
   b) **Azure** モニタリングメトリクスを有効または無効にします。
   c) バックエンドオートスケールを有効または無効にします。



6. 次へ**:** ネットワークと追加設定をクリックします。

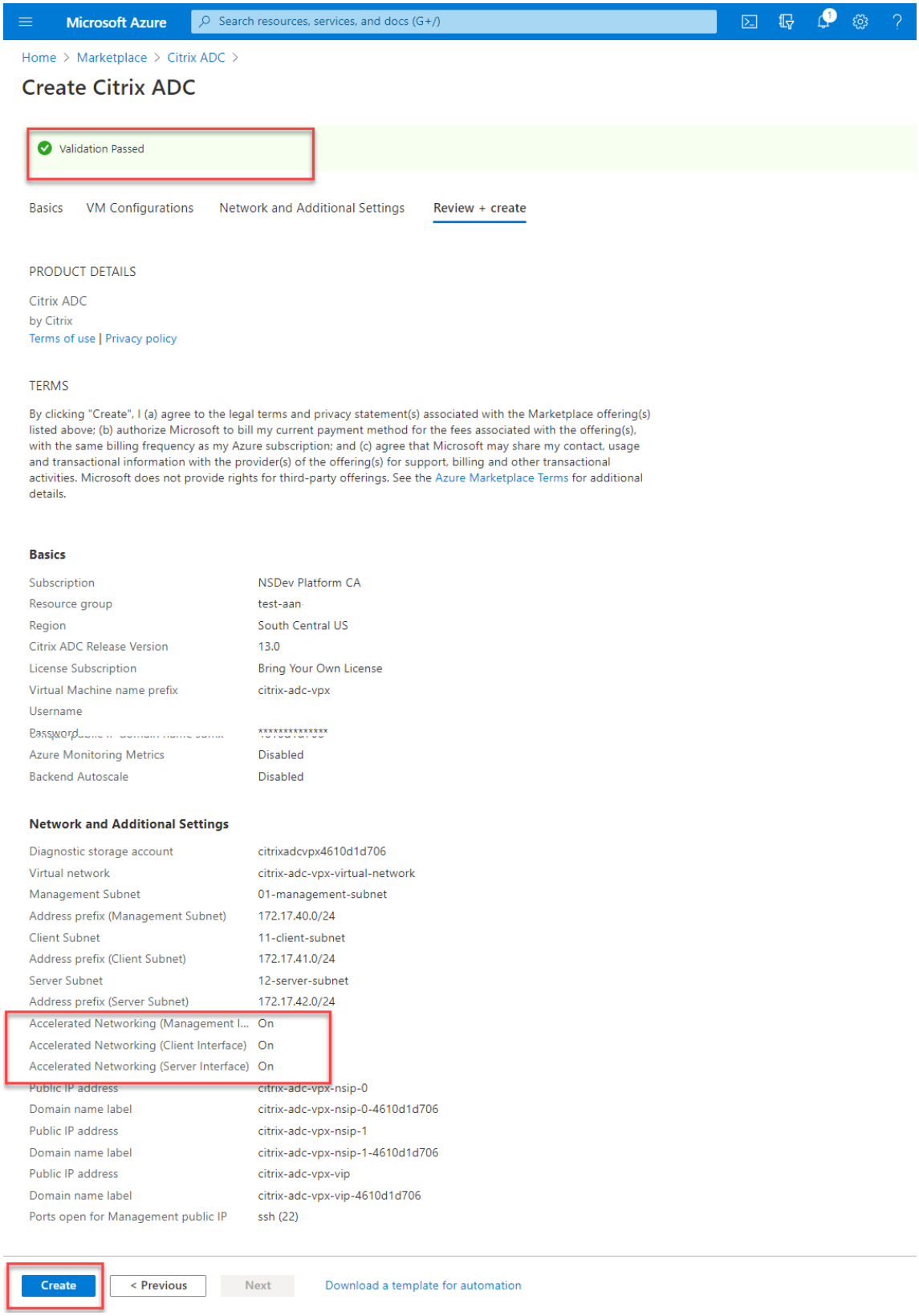
   [ネットワークとその他の設定**]** ページで、ブート診断アカウントを作成し、ネットワーク設定を構成します。

   [高速ネットワーキング] セクションには、管理インターフェイス、クライアントインターフェイス、およびサーバインターフェイスについて、アクセラレーションネットワーキングを個別に有効または無効にするオプションがあります。

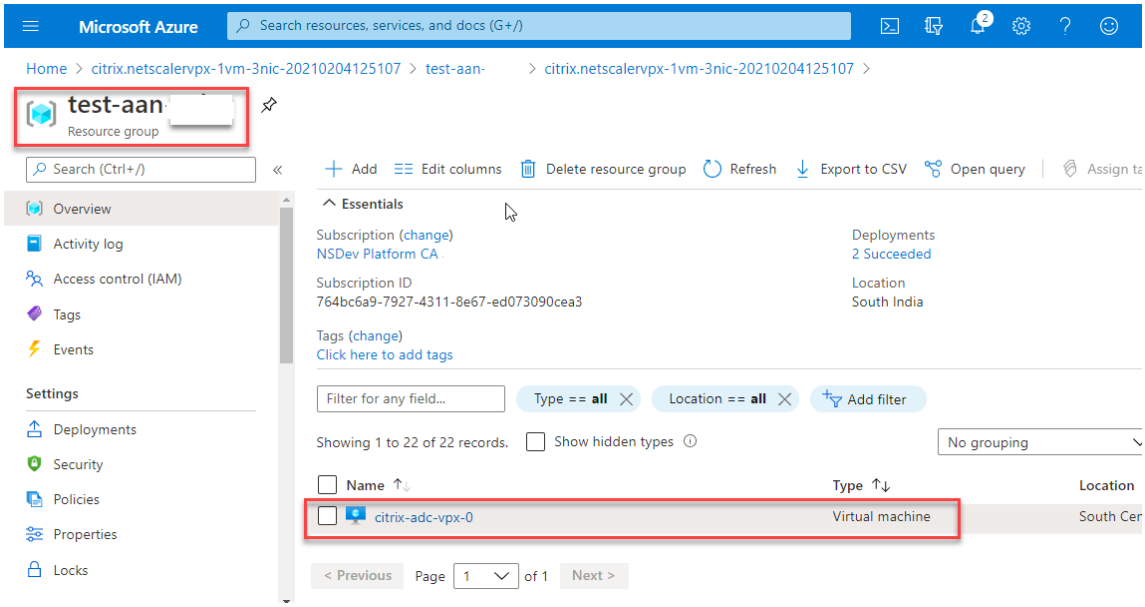


7. [** 次へ: レビュー + 作成 **] をクリックします。

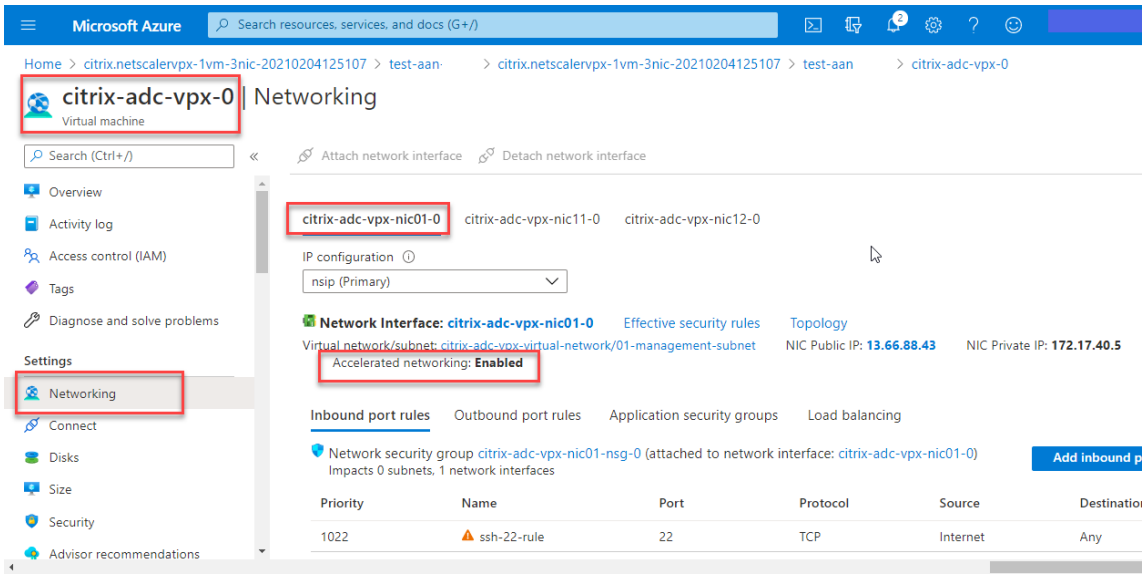
   検証が成功したら、基本設定、仮想マシンの構成、ネットワーク、および追加設定を確認し、[作成] をクリックします。Azure リソースグループが必要な構成で作成されるまでに時間がかかる場合があります。



8. デプロイが完了したら、リソースグループを選択して構成の詳細を表示します。



9. 高速ネットワーク構成を確認するには、[仮想マシン] **>** [ネットワーク] を選択します。アクセラレートネットワーキングのステータスは、NIC ごとに [有効] または [無効] と表示されます。



Azure PowerShell を使用して高速ネットワーキングを有効にする

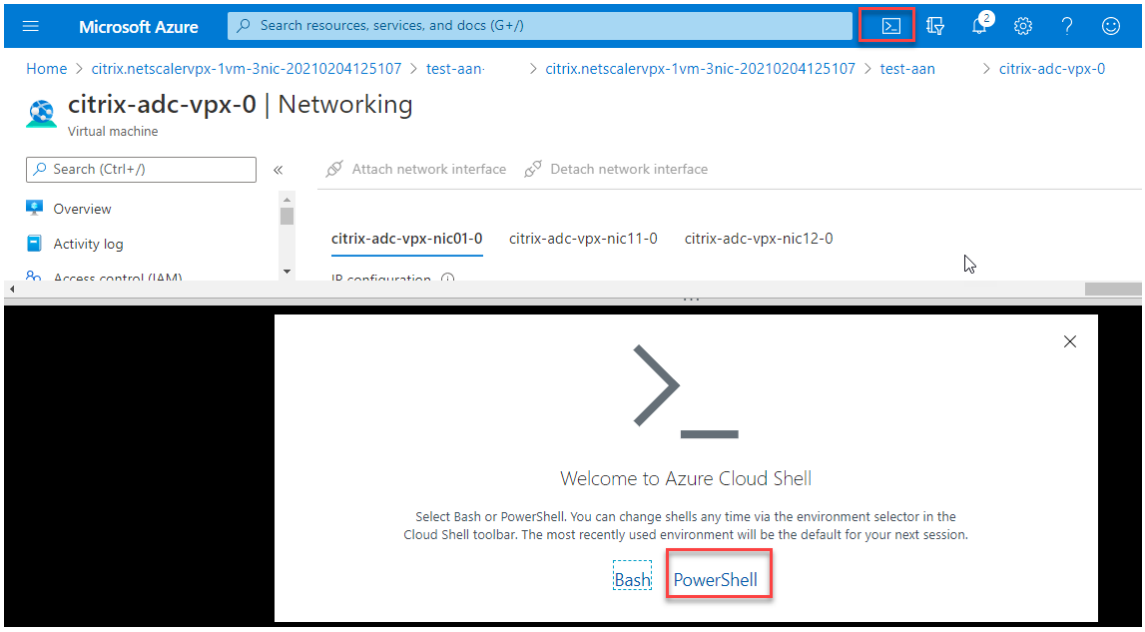
仮想マシンの作成後に高速ネットワークを有効にする必要がある場合は、Azure PowerShell を使用して有効化できます。

> 注:
>
> Azure PowerShell を使用した高速ネットワーキングを有効にする前に、仮想マシンを停止してください。

Azure PowerShell を使用して高速ネットワークを有効にするには、次の手順を実行します。

1. **Azure** ポータルに移動し、右上隅にある **PowerShell** アイコンをクリックします。

   > 注:
   >
   > Bash モードの場合は、PowerShell モードに切り替えます。



2. コマンドプロンプトで、次のコマンドを実行します:

```
az network nic update --name <nic-name> --accelerated-networking [true | false] --resource-group <resourcegroup-name>
<!--NeedCopy-->
```

アクセラレートネットワーキングパラメータは、次のいずれかの値を受け入れます。

- **True**: 指定した NIC で高速ネットワーキングを有効にします。
- **False**: 指定された NIC のアクセラレーションネットワーキングを無効にします。

特定の **NIC** で高速ネットワーキングを有効にするには、次の手順を実行します。

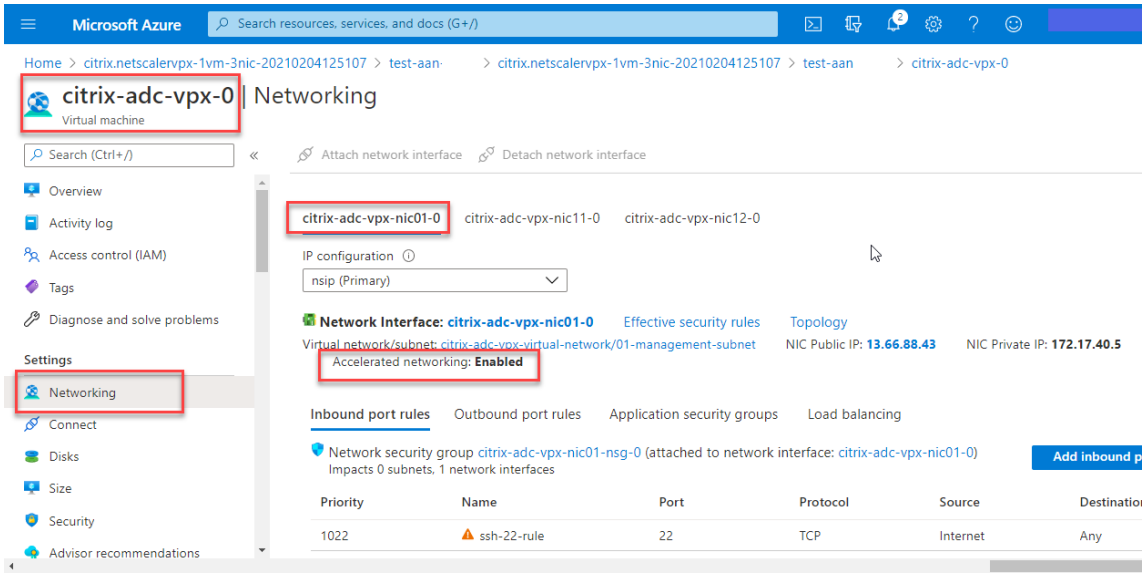
```
az network nic update --name citrix-adc-vpx-nic01-0 --accelerated-networking true --resource-group rsgp1-aan
<!--NeedCopy-->
```

特定の **NIC** で高速ネットワーキングを無効にするには、次の手順を実行します。

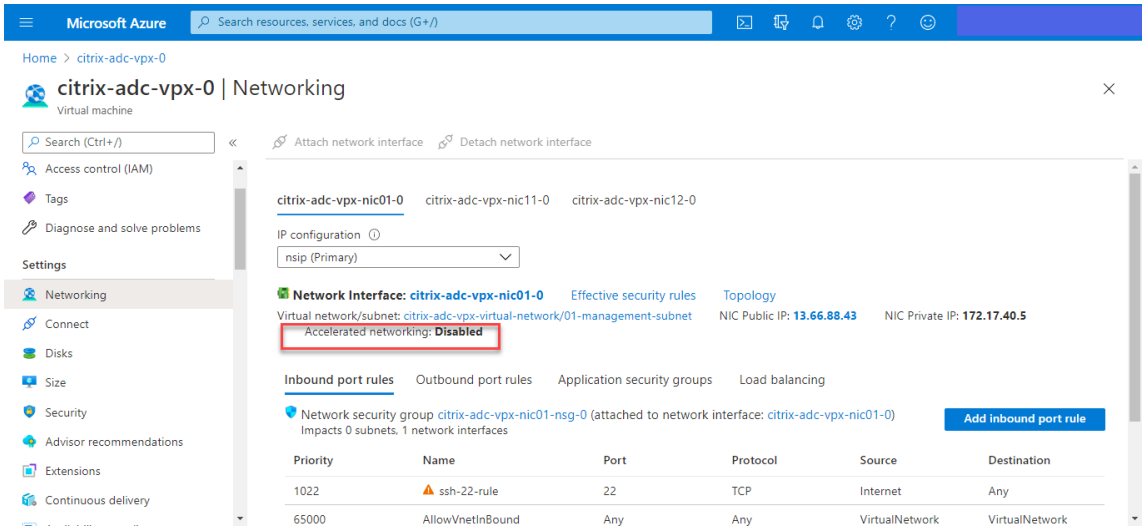
```
az network nic update --name citrix-adc-vpx-nic01-0 --accelerated-networking false --resource-group rsgp1-aan
<!--NeedCopy-->
```

3. デプロイが完了した後にアクセラレーテッドネットワーキングのステータスを確認するには、[**VM**] > [ネットワーク] に移動します。

次の例では、アクセラレートネットワーキングが有効になっていることを確認します。



次の例では、アクセラレートネットワーキングが無効になっていることがわかります。



NetScaler FreeBSD Shell を使用してインターフェイス上で高速ネットワークを検証するには

NetScaler FreeBSD シェルにログインし、次のコマンドを実行して高速ネットワークのステータスを確認できます。

ConnectX3 NIC の例**:**

次の例は、Mellanox ConnectX3 NIC の「ifconfig」コマンド出力を示しています。「50/n」は、Mellanox ConnectX3 NIC の VF インターフェイスを示します。0/1 と 1/1 は、NetScaler VPX インスタンスの PV インターフェイスを示します。PV インターフェイス（1/1）と CX3 VF インターフェイス（50/1）の両方が同じ MAC アドレス（00:22:48:1 c: 99:3 e）を持つことがわかります。これは、2 つのインターフェイスが一緒にバンドルされていることを示しま

す。

```
root@nvr-us-cx3# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 1500
        options=3<RXCSUM,TXCSUM>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
0/1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>
        ether 00:0d:3a:98:71:be
        inet 172.16.27.11 netmask 0xffffff00 broadcast 172.16.27.255
        inet6 fe80::20d:3aff:fe98:71be%0/1 prefixlen 64 autoconf scopeid 0x2
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active
1/1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>
        ether 00:22:48:1c:99:3e
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active
50/1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500

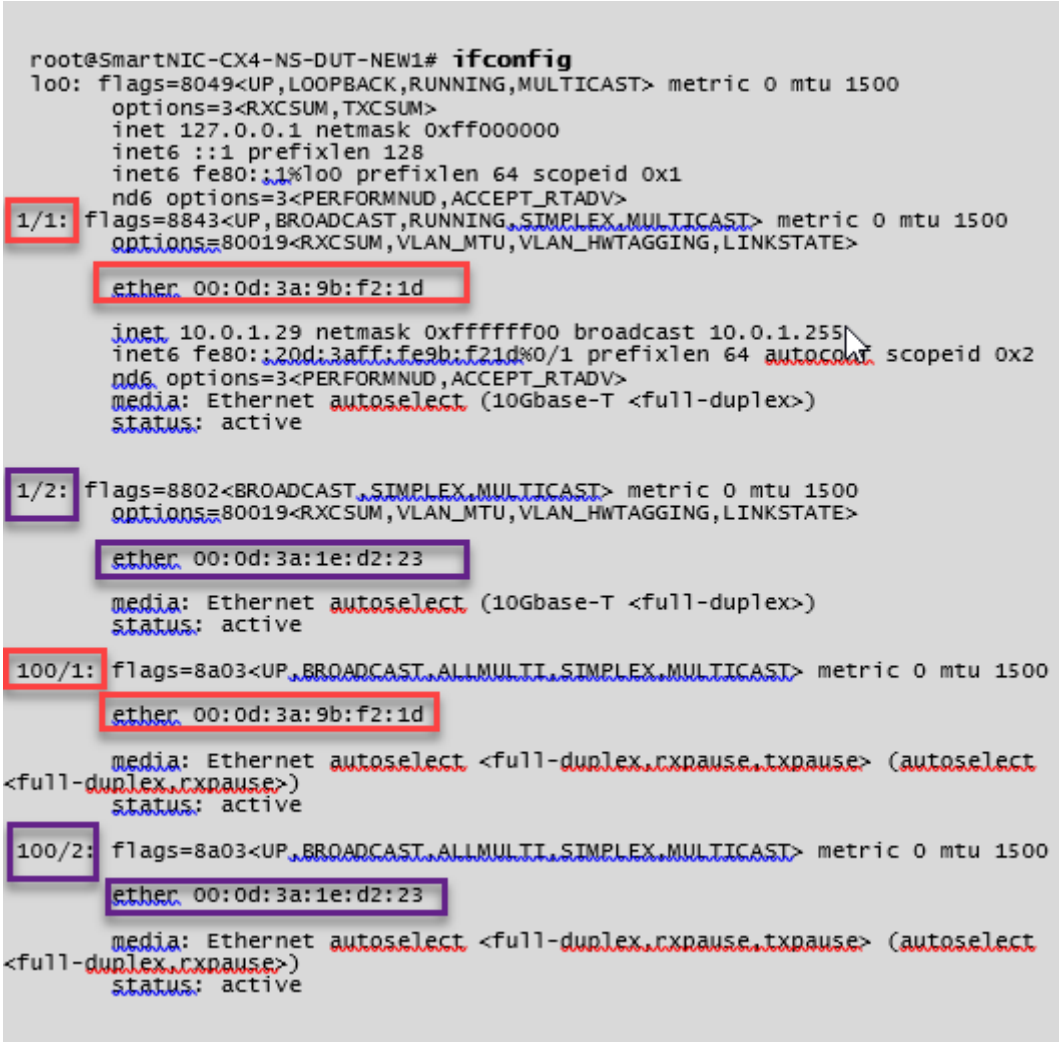
options=900b8<VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,VLAN_HWFILTER,LINKSTATE>
        ether 00:22:48:1c:99:3e
        media: Ethernet autoselect (<unknown subtype>)
        status: active
```

ConnectX4 NIC の例**:**

次の例は、Mellanox ConnectX4 NIC の「ifconfig」コマンド出力を示しています。「100/n」は、Mellanox ConnectX4 NIC の VF インターフェイスを示します。0/1、1/1、および 1/2 は、NetScaler VPX インスタンスの PV インターフェイスを示します。

PV インターフェイス（1/1）と CX4 VF インターフェイス（100/1）の両方が同じ MAC アドレス（00:0 d: 3a: 9b: f 2:1 d）を持つことがわかります。これは、2 つのインターフェイスが一緒にバンドルされていることを示します。同

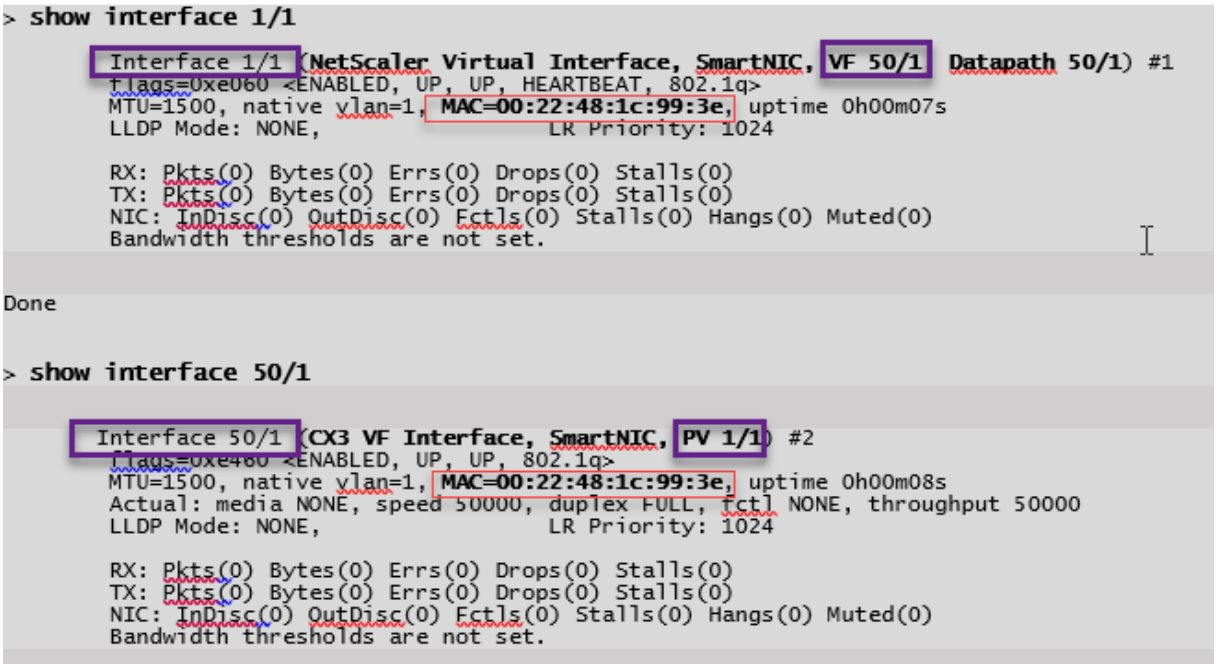
様に、PV インターフェイス（1/2）と CX4 VF インターフェイス（100/2）は同じ MAC アドレス（00:0 d: 3a: 1:D 2:23）を持ちます。



ADC CLI を使用してインターフェイスで高速ネットワーキングを検証するには

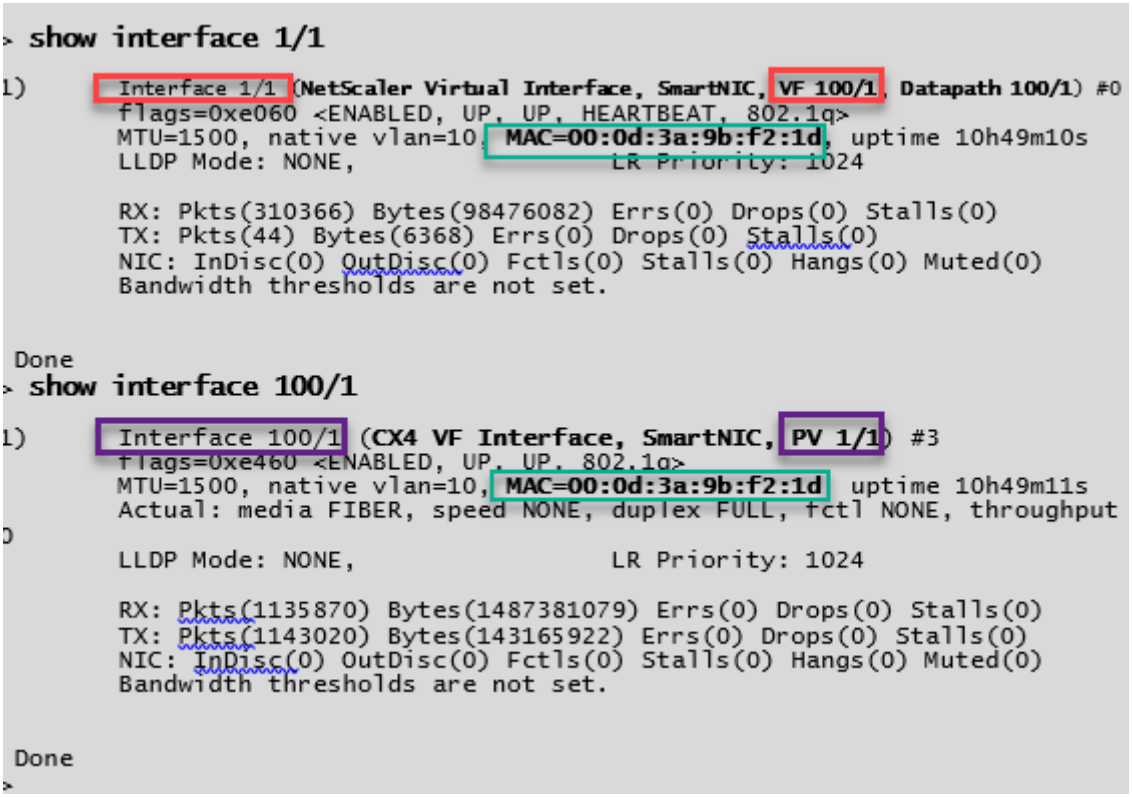
ConnectX3 NIC の例**:**

次の show interface コマンドの出力は、PV インターフェイス 1/1 が SR-IOV VF NIC である仮想機能 50/1 にバンドルされていることを示しています。1/1 と 50/1 の NIC の両方の MAC アドレスは同じです。高速ネットワーキングが有効になると、1/1 インターフェイスのデータは、ConnectX3 インターフェイスである 50/1 インターフェイスのデータパスを介して送信されます。PV インターフェイス (1/1) の「show interface」出力が VF (50/1) を指していることがわかります。同様に、VF インターフェイス (50/1) の「show interface」出力は PV インターフェイス (1/1) を指します。



ConnectX4 NIC の例**:**

次の show interface コマンドの出力は、PV インターフェイス 1/1 が SR-IOV VF NIC である仮想機能 100/1 にバンドルされていることを示しています。1/1 と 100/1 の NIC の両方の MAC アドレスは同じです。高速ネットワーキングが有効になると、1/1 インターフェイスのデータは、ConnectX4 インターフェイスである 100/1 インターフェイスのデータパスを介して送信されます。PV インターフェイス (1/1) の「show interface」出力が VF (100/1) を指すことがわかります。同様に、VF インターフェイス (100/1) の「show interface」出力は PV インターフェイス (1/1) を指します。



NetScaler での注意点

- PV インターフェイスは、必要なすべての操作のプライマリインターフェイスまたはメインインターフェイスと見なされます。設定は PV インターフェイスでのみ実行する必要があります。
- VF インターフェイスでのすべての「set」操作は、以下を除いてブロックされます。
  - インターフェイスを有効にする
  - インターフェイスを無効にする
  - インターフェイスをリセット
  - 統計をクリアする

  > 注:
  >
  > VF インターフェイスでは操作を実行しないことをお勧めします。

- `show interface`コマンドを使用して、PV インターフェイスと VF インターフェイスとのバインディングを確認できます。
- NetScaler リリース 13.1～33.x 以降、NetScaler VPX インスタンスは、Azure アクセラレーテッドネットワークでの動的な NIC の取り外しと、取り外された NIC の再接続をシームレスに処理できます。Azure は、ホストのメンテナンス作業のために、高速ネットワークの SR-IOV VF NIC を削除できます。NIC が Azure VM から削除されると、NetScaler VPX インスタンスのインターフェイスステータスが「リンクダウン」と表示さ

れ、トラフィックは仮想インターフェイスのみを経由します。取り外した NIC が再接続されると、VPX インスタンスは再接続された SR-IOV VF NIC を使用します。このプロセスはシームレスに行われ、設定は不要です。

PV インターフェースへの VLAN の構成

PV インターフェイスが VLAN にバインドされている場合、関連するアクセラレーション VF インターフェイスも PV インターフェイスと同じ VLAN にバインドされます。この例では、PV インターフェイス（1/1）は VLAN（20）にバインドされています。PV インターフェイス（1/1）にバンドルされている VF インターフェイス（100/1）も VLAN 20 にバインドされます。

例:

1. VLAN を作成します。

```
add vlan 20
<!--NeedCopy-->
```

2. VLAN を PV インターフェイスにバインドします。

```
bind vlan 20  -ifnum 1/1

show vlan

1)  VLAN ID: 1
    Link-local IPv6 addr: fe80::20d:3aff:fe9b:f21d/64
    Interfaces : LO/1

2)  VLAN ID: 10     VLAN Alias Name:
    Interfaces : 0/1 100/1
    IPs : 10.0.1.29  Mask: 255.255.255.0

3)  VLAN ID: 20     VLAN Alias Name:
    Interfaces :  1/1 100/2

<!--NeedCopy-->
```

> 注
>
> VLAN バインディング操作は、アクセラレーション VF インターフェイスでは許可されません。

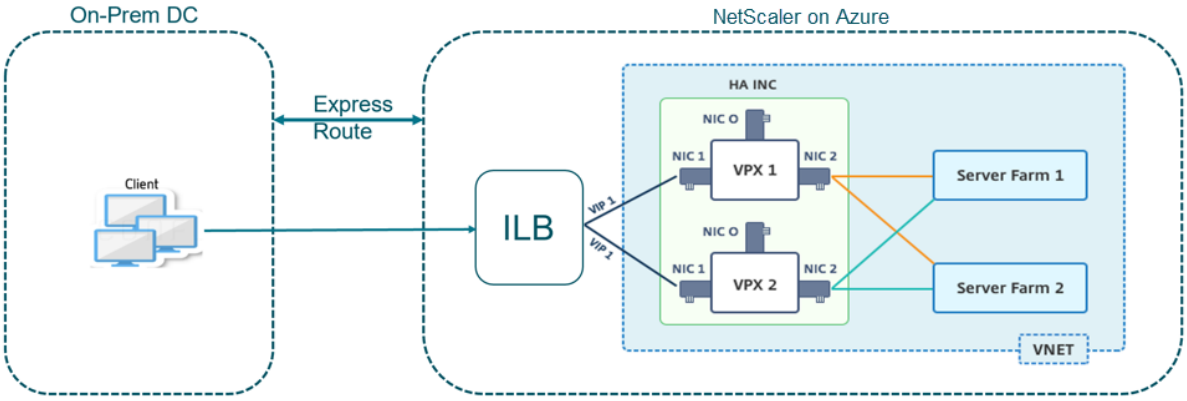
```
bind vlan 1 -ifnum 100/1
ERROR: Operation not permitted
<!--NeedCopy-->
```


Azure ILB で NetScaler 高可用性テンプレートを使用して HA-INC ノードを構成する

August 15, 2023

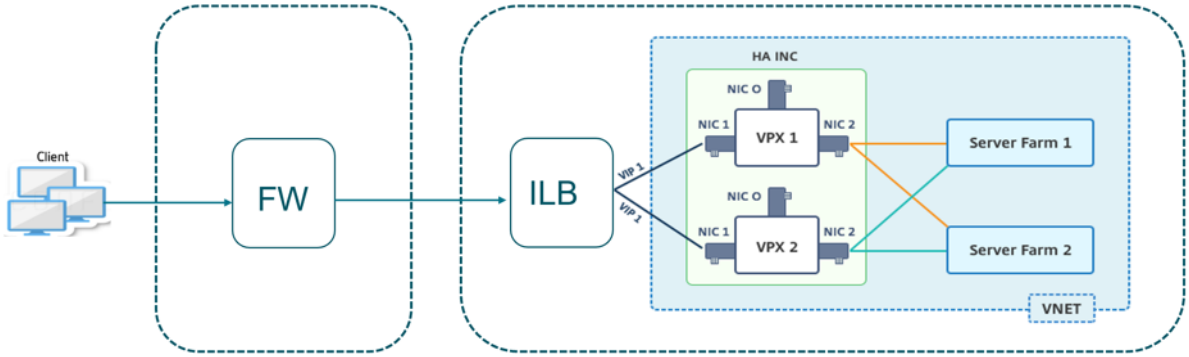
イントラネットアプリケーション用の標準テンプレートを使用すると、HA-INC モードで一対の VPX インスタンスを迅速かつ効率的にデプロイできます。Azure 内部ロードバランサー (ILB) は、図 1 に示すように、フロントエンドに内部 IP アドレスまたはプライベート IP アドレスを使用します。このテンプレートでは、3 つのサブネットと 6 つの NIC を持つ 2 つのノードが作成されます。サブネットは、管理、クライアント、およびサーバー側のトラフィック用で、各サブネットはデバイスごとに異なる NIC に属します。

図 1: 内部ネットワーク内のクライアント用の NetScaler ADC HA ペア



この展開は、図 2 に示すように、NetScaler HA ペアがファイアウォールの内側にある場合にも使用できます。パブリック IP アドレスはファイアウォールに属し、ILB のフロントエンド IP アドレスに NAT されます。

図 2: パブリック IP アドレスを持つファイアウォールと NetScaler ADC HA のペア

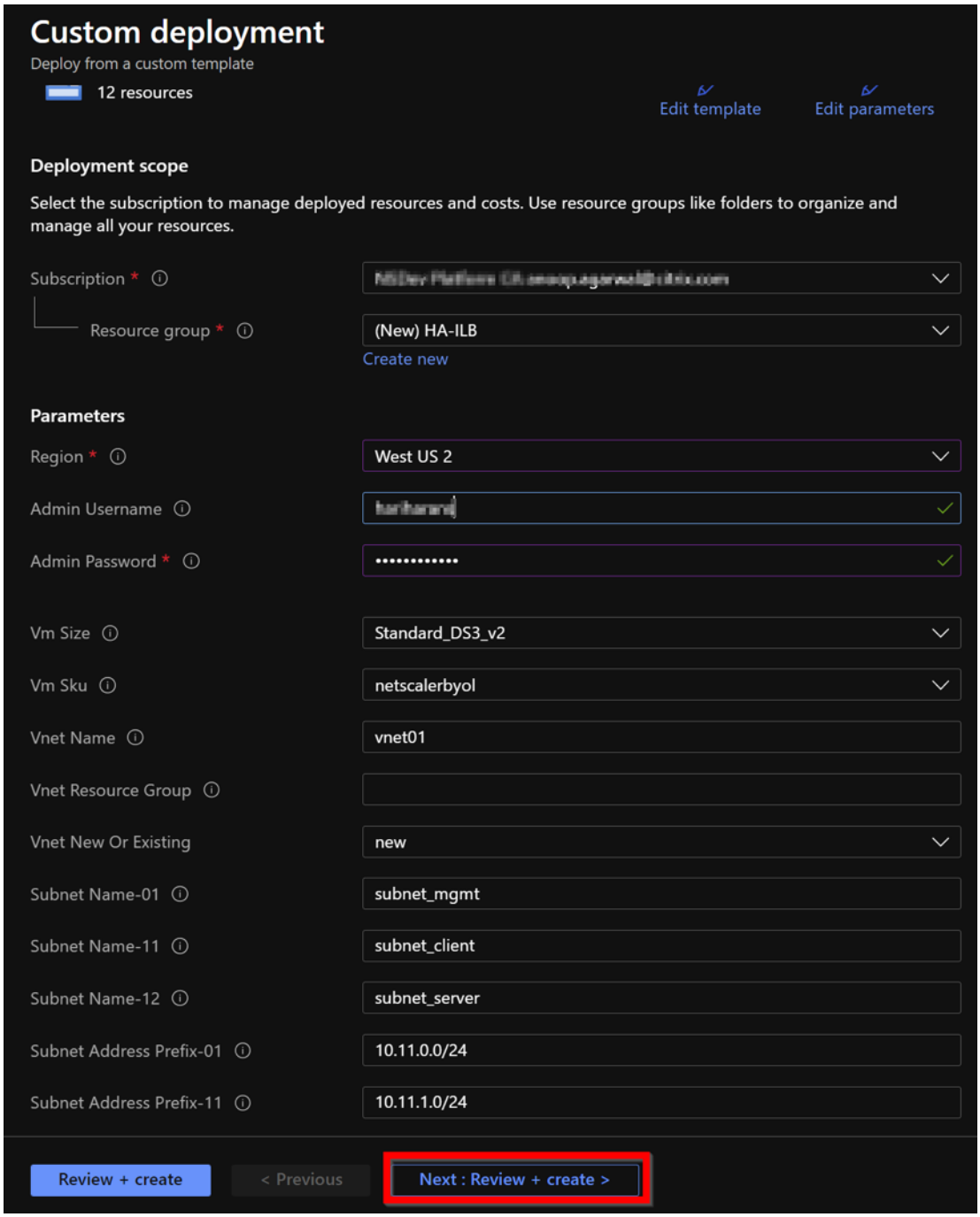


イントラネットアプリケーション用の NetScaler ADC HA ペアテンプレートは、Azure ポータルで入手できます。

次の手順を実行してテンプレートを起動し、Azure 可用性セットを使用して高可用性 VPX ペアをデプロイします。

1. Azure Portal から、[カスタム展開] ページに移動します。

2. [基本] ページが表示されます。リソースグループを作成します。[パラメータ] タブで、[リージョン]、[管理者ユーザー名]、[管理者パスワード]、[ライセンスタイプ] (VM sku)、およびその他のフィールドの詳細を入力します。

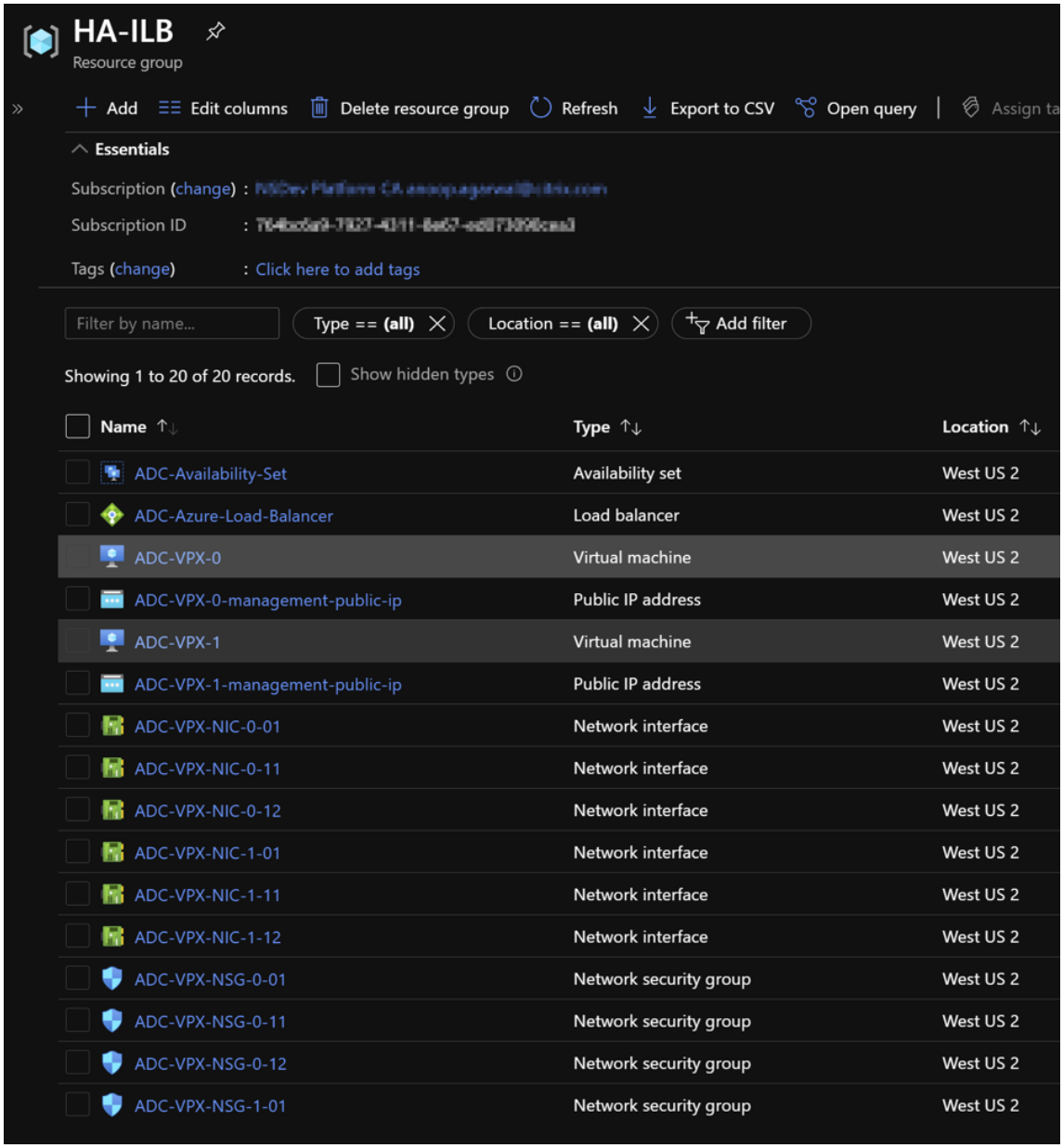


3. [** 次へ: レビュー + 作成 **] をクリックします。

   必要な構成で Azure リソースグループが作成されるまで時間がかかることがあります。完了後、Azure ポータルでリソースグループを選択して、LB ルール、バックエンドプール、ヘルスプローブなどの構成の詳細を確認します。高可用性ペアは ADC-VPX-0 と ADC-VPX-1 として表示されます。

追加のセキュリティルールやポートを作成するなど、HA セットアップでさらに変更が必要な場合は、Azure Portal から実行できます。

必要な構成が完了すると、次のリソースが作成されます。



4. **ADC-VPX-0** および **ADC-VPX-1** ノードにログオンして、次の設定を検証します。

   - 両方のノードの NSIP アドレスは管理サブネットに存在する必要があります。
   - プライマリ（ADC-VPX-0）ノードとセカンダリ（ADC-VPX-1）ノードには、2 つの SNIP アドレスが表示される必要があります。一方の SNIP（クライアントサブネット）は ILB プローブへの応答に使用され、もう 1 つの SNIP（サーバーサブネット）はバックエンドサーバー通信に使用されます。

> 注
>
> HA-INC モードでは、ADC-VPX-0VM と ADC-VPX-1VM の SNIP アドレスは、両方が同じである従来のオンプレミス ADC HA 展開とは異なり、同じサブネット内では異なります。
> VPX ペア SNIP が異なるサブネットにある場合、または VIP が SNIP と同じサブネット内にない場合に展開をサポートするには、Mac ベース転送（MBF）を有効にするか、各 VIP の静的ホストルートを各 VPX ノードに追加する必要があります。

プライマリノード（ADC-VPX-0）

```
> sh ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.11.0.5        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.11.1.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.11.3.4        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
>
>
```

```
> sh ha node
1)      Node ID:      0
        IP:       10.11.0.5 (ADC-VPX-0)
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:0:20:26 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.11.0.4
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

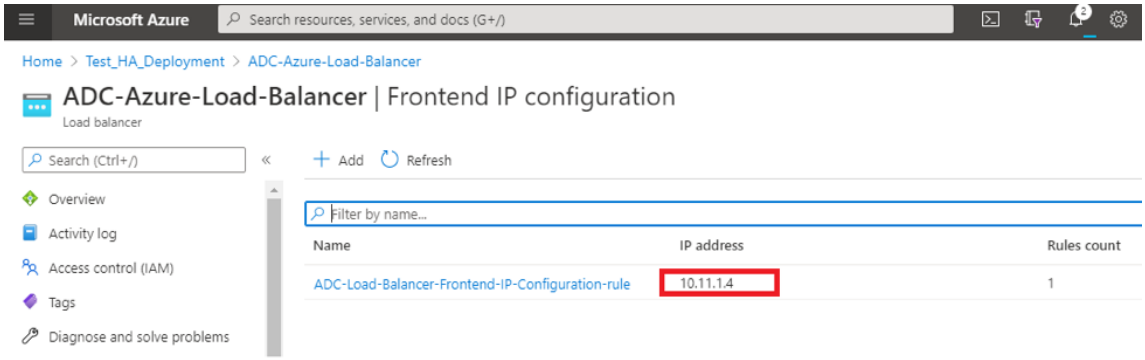
セカンダリノード（ADC-VPX-1）


```
> sh ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.11.0.4        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.11.1.6        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.11.3.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
>
>
```

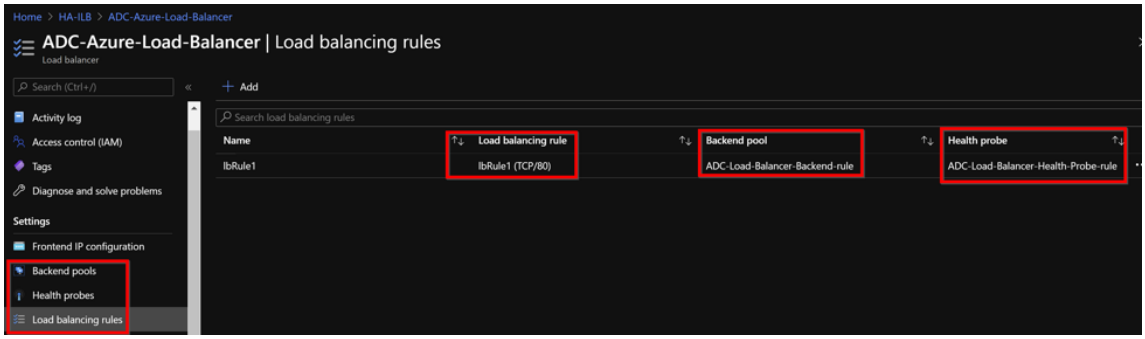
```
> sh ha node
1)      Node ID:      0
        IP:       10.11.0.4 (ADC-VPX-1)
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:0:24:18 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.11.0.5
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

5. プライマリノードとセカンダリノードが UP で、同期ステータスが **SUCCESS** になったら、プライマリノード (ADC-VPX-0) の負荷分散仮想サーバーまたはゲートウェイ仮想サーバーを、ADC Azure ロードバランサーのプライベートフローティング IP (FIP) アドレスで構成する必要があります。詳細については、「サンプル設定」セクションを参照してください。

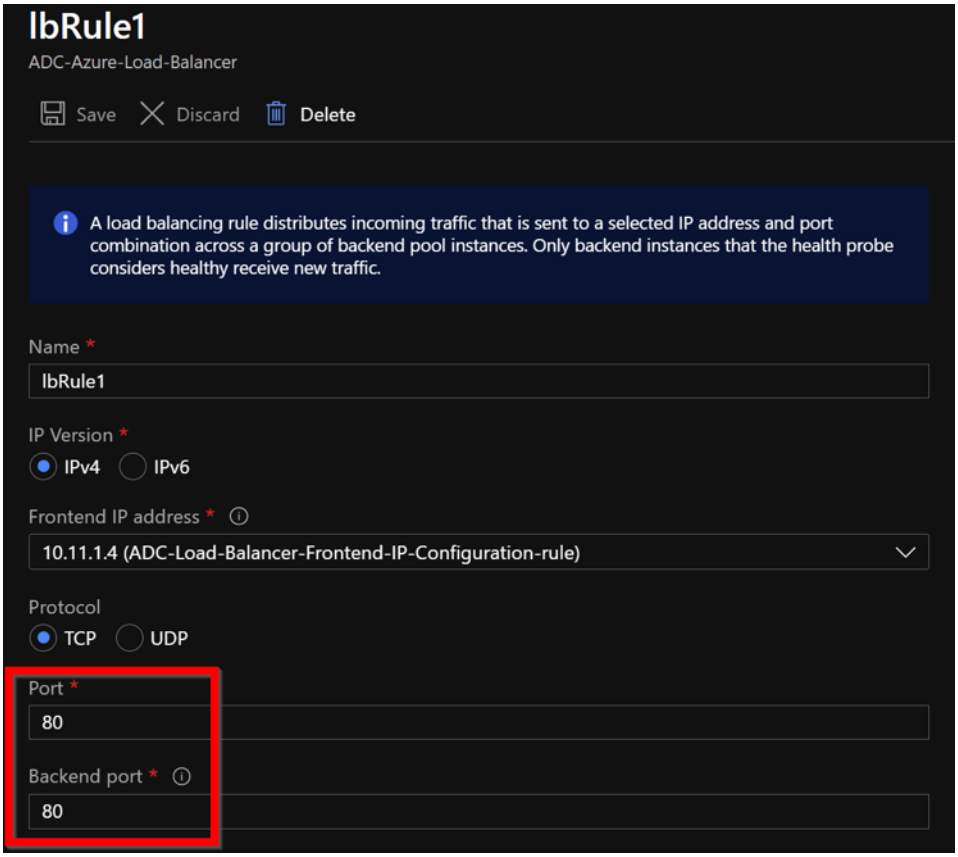
6. ADC Azure 負荷分散サーバーのプライベート IP アドレスを見つけるには、**Azure portal > ADC Azure Load Balancer > Frontend IP configuration** に移動します。



7. **Azure Load Balancer** の 構成ページで、ARM テンプレートの展開は、LB ルール、バックエンドプール、およびヘルスプローブの作成に役立ちます。

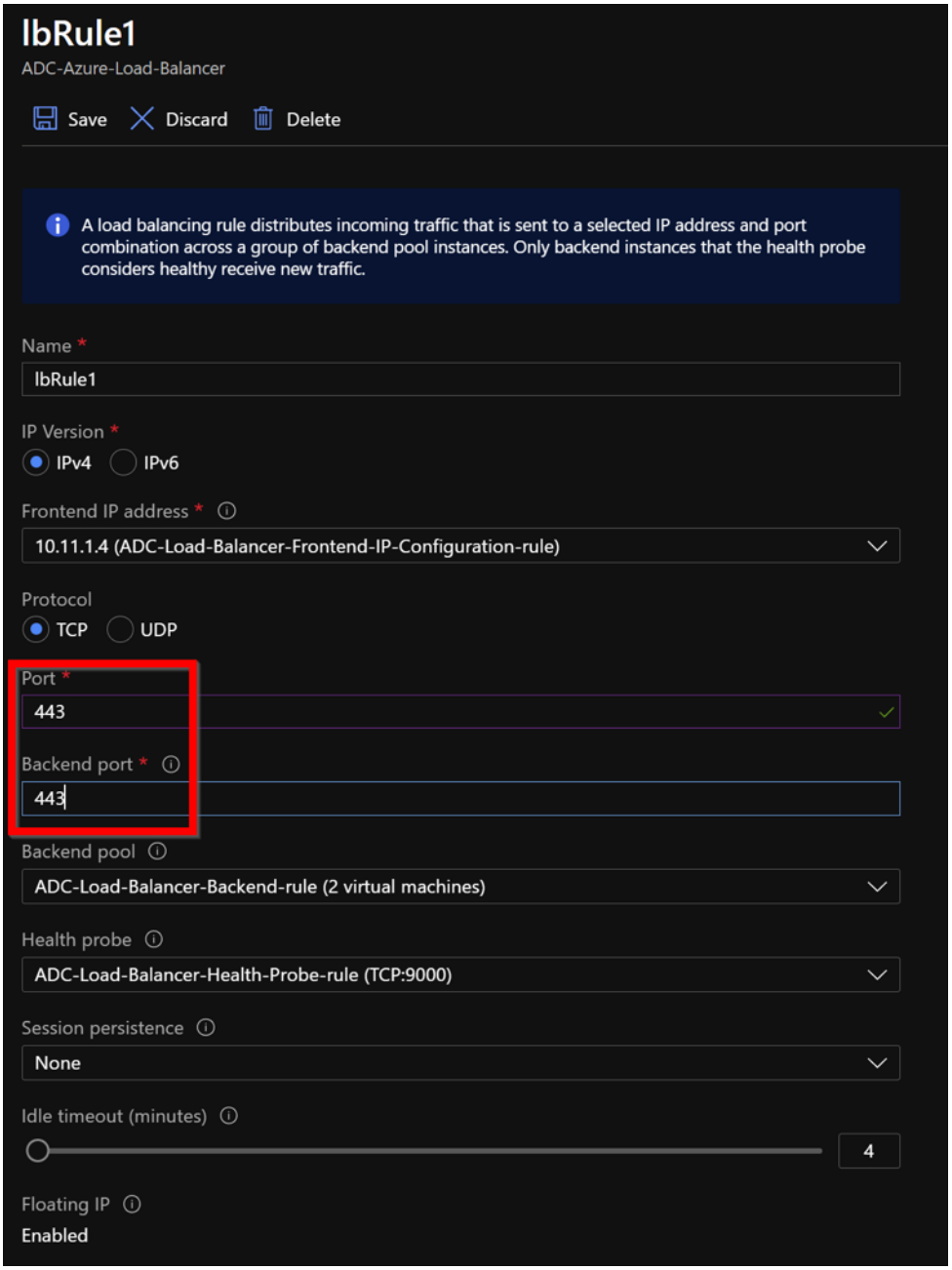


- LB ルール (lbrule1) はデフォルトでポート 80 を使用します。



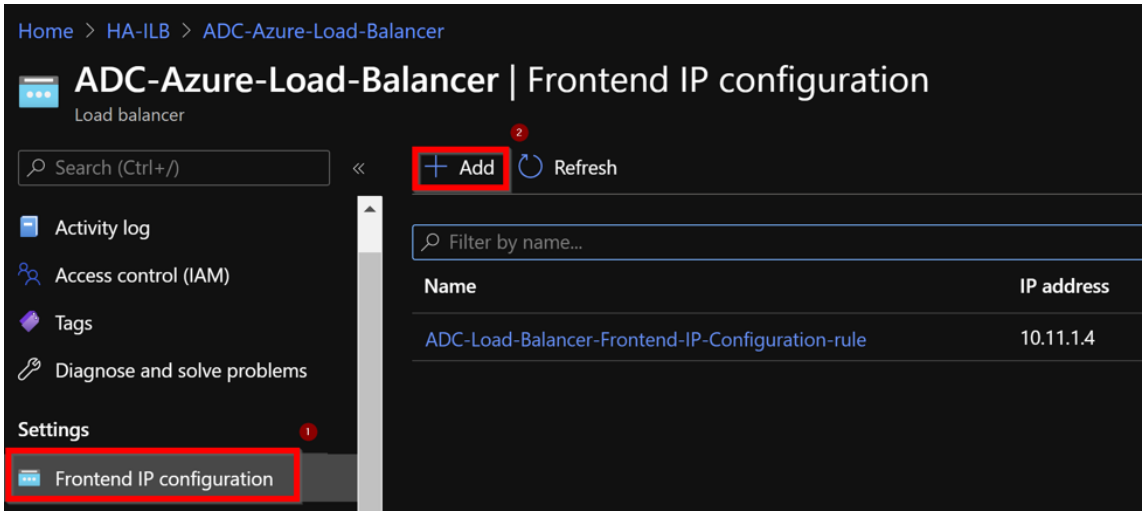
- ポート 443 を使用するようにルールを編集し、変更を保存します。

  > 注
  >
  > セキュリティを強化するため、LB 仮想サーバーまたはゲートウェイ仮想サーバーには SSL ポート 443 を使用することをお勧めします。

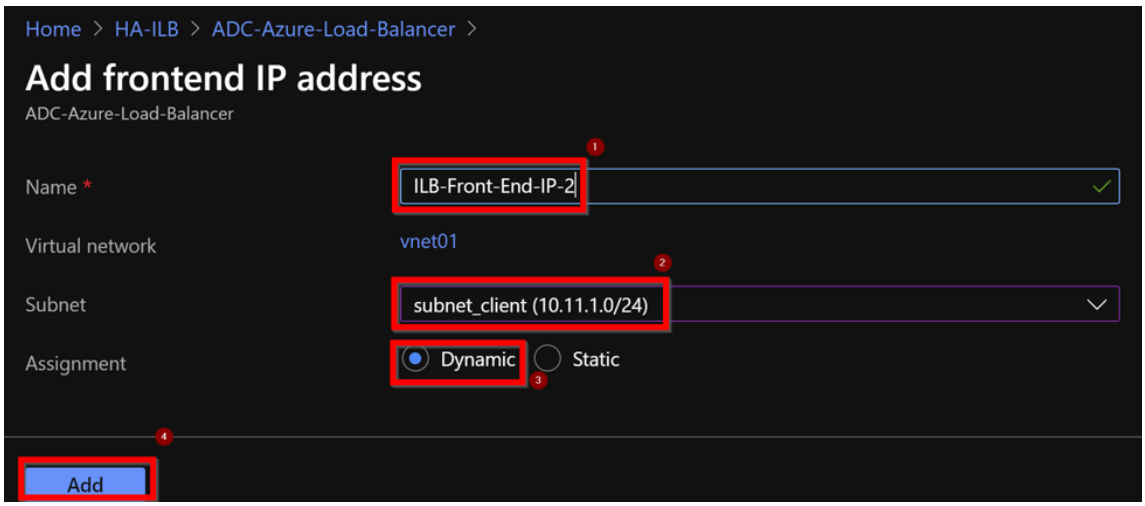


ADC に VIP アドレスを追加するには、次の手順に従います。

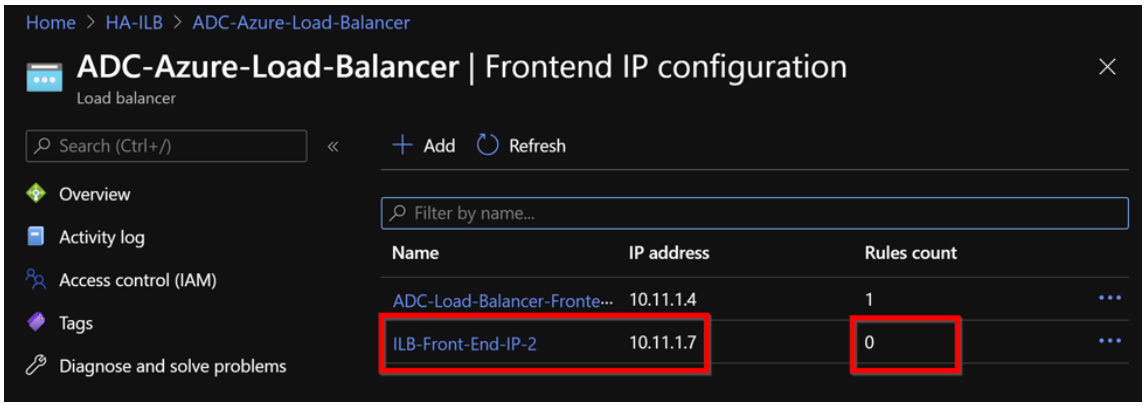
1. **Azure Load Balancer > Frontend IP** 構成に移動し、[追加] をクリックして新しい内部ロードバランサー IP アドレスを作成します。



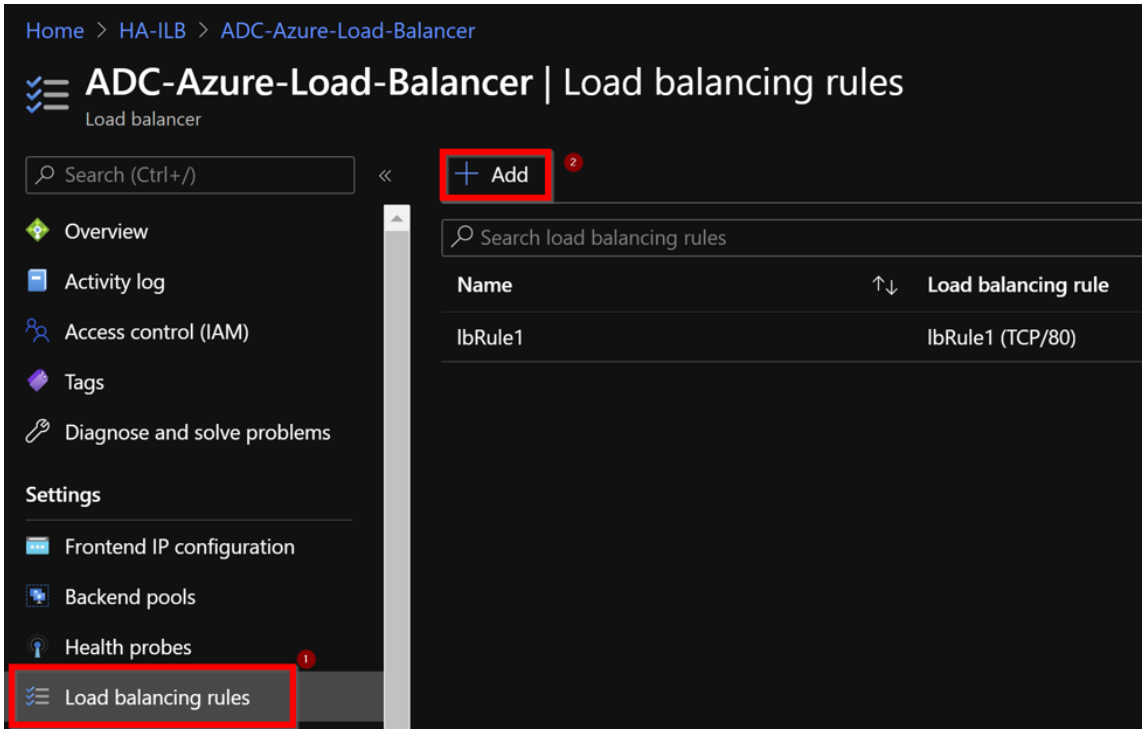
2. **[Add frontend IP address**] ページで、名前を入力し、クライアントサブネットを選択し、動的 IP アドレスまたは静的 IP アドレスを割り当てて、[**Add**] をクリックします。



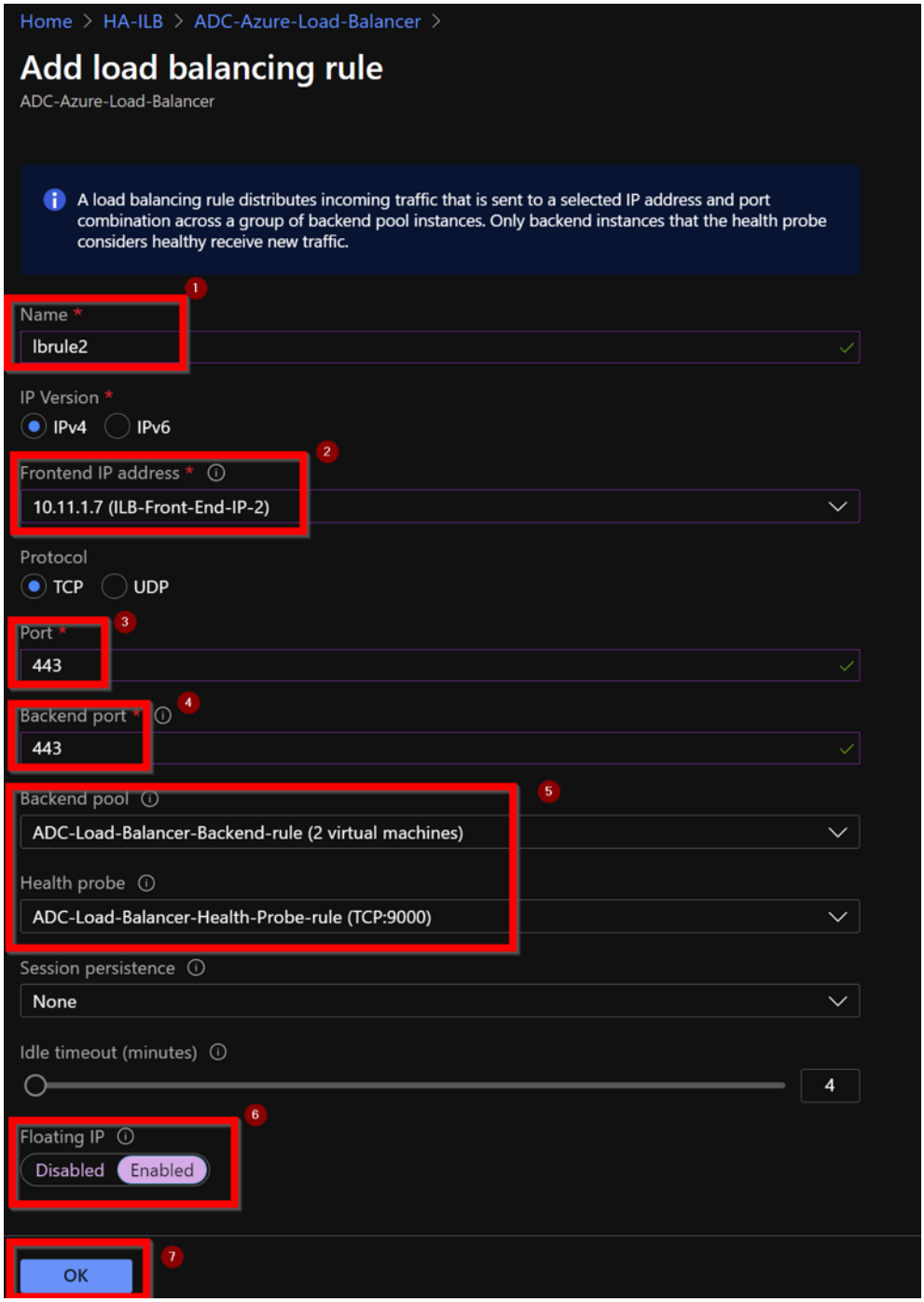
3. フロントエンド IP アドレスは作成されますが、LB ルールは関連付けられていません。新しい負荷分散ルールを作成し、フロントエンド IP アドレスに関連付けます。



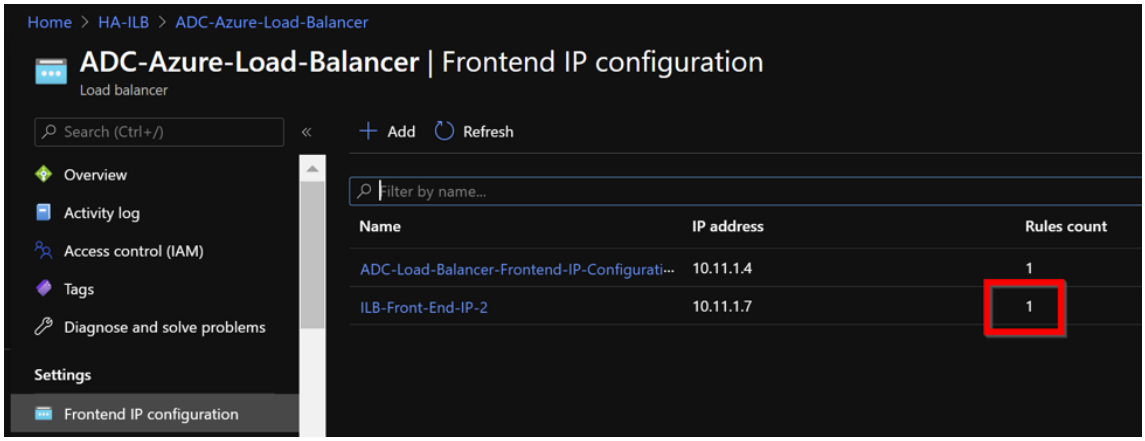
4. [**Azure** ロードバランサー] ページで、[負荷分散ルール] を選択し、[追加] をクリックします。



5. 新しいフロントエンド IP アドレスとポートを選択して、新しい LB ルールを作成します。[フローティング **IP**] フィールドは [有効] に設定する必要があります。



6. これで、フロントエンド **IP** 設定に、適用されている LB ルールが表示されます。



設定例

ゲートウェイ VPN 仮想サーバーと負荷分散仮想サーバーを設定するには、プライマリノード (ADC-VPX-0) で次のコマンドを実行します。設定はセカンダリノード（ADC-VPX-1）に自動的に同期されます。

ゲートウェイのサンプル構成

```
enable feature aaa LB SSL SSLVPN
enable ns mode MBF
add vpn vserver vpn_ssl SSL 10.11.1.4 443
add ssl certKey ckp -cert wild-cgwsanity.cer -key wild-cgwsanity.key
bind ssl vserver vpn_ssl -certkeyName ckp
<!--NeedCopy-->
```

負荷分散のサンプル構成

```
enable feature LB SSL
enable ns mode MBF
add lb vserver lb_vs1 SSL 10.11.1.7 443
bind ssl vserver lb_vs1 -certkeyName ckp
<!--NeedCopy-->
```

ILB の内部 IP アドレスに関連付けられている完全修飾ドメイン名 (FQDN) を使用して、負荷分散または VPN 仮想サーバーにアクセスできるようになりました。

負荷分散仮想サーバーの構成方法の詳細については、「リソース」セクションを参照してください。

リソース:

次のリンクには、HA の導入と仮想サーバの設定に関する追加情報が表示されます。

- 異なるサブネットに高可用性ノードを構成する
- 基本的な負荷分散を設定する

関連リソース:

- PowerShell コマンドを使用して、複数の IP アドレスと NIC を使用した高可用性設定を構成する

- Azure でのアクティブスタンバイ HA デプロイメントでの GSLB の構成

インターネット向けアプリケーション用の **NetScaler** 高可用性テンプレートを使用して **HA-INC** ノードを構成する

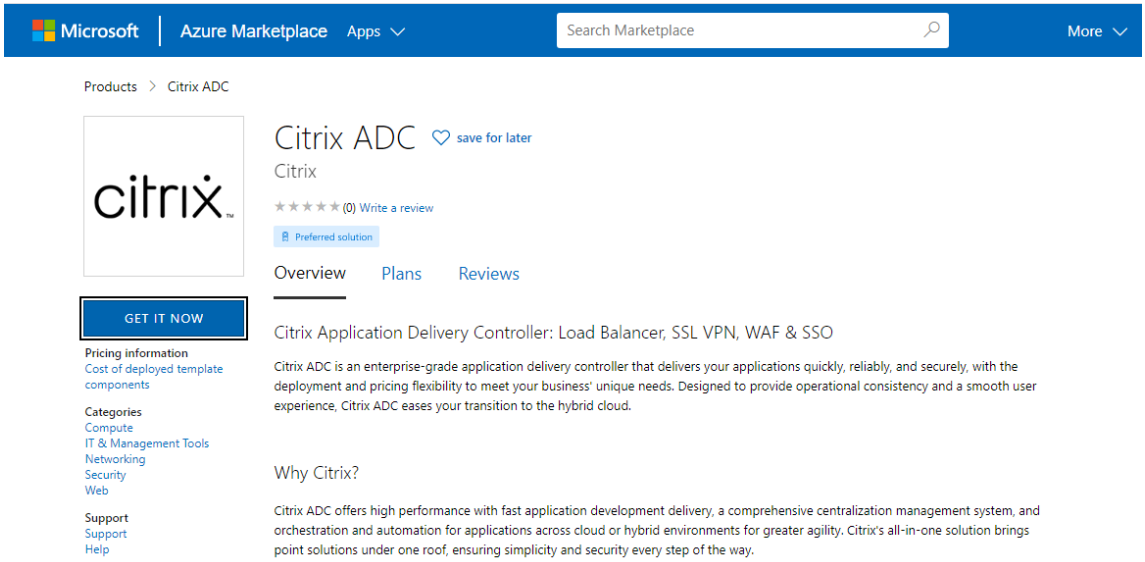
February 15, 2024

インターネット向けアプリケーションの標準テンプレートを使用すると、一対の VPX インスタンスを HA-INC モードで迅速かつ効率的にデプロイできます。Azure ロードバランサー (ALB) は、フロントエンドにパブリック IP アドレスを使用します。このテンプレートでは、3 つのサブネットと 6 つの NIC を持つ 2 つのノードが作成されます。サブネットは、管理、クライアント、およびサーバー側のトラフィック用です。各サブネットには、両方の VPX インスタンス用に 2 つの NIC があります。

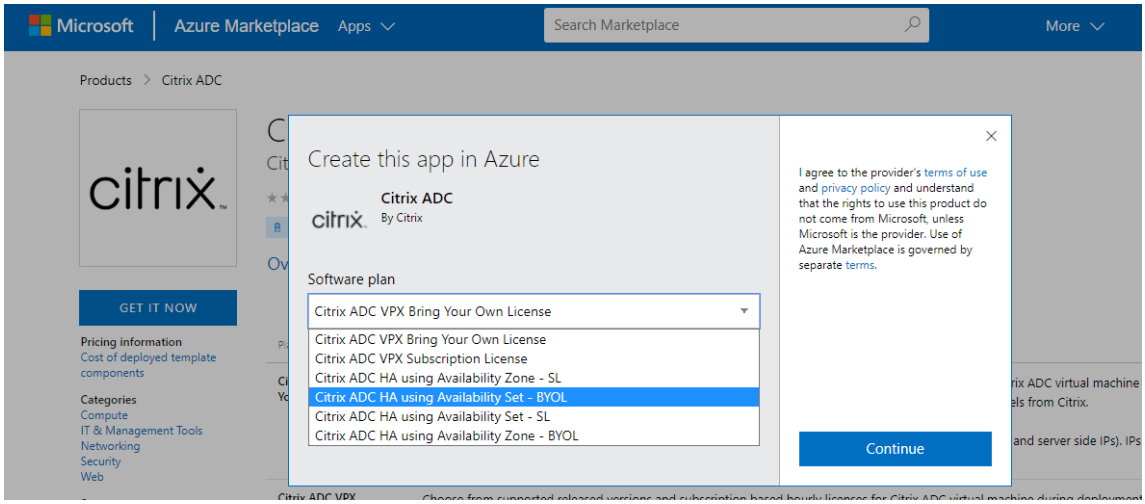
インターネット向けアプリケーションの NetScaler HA ペアテンプレートは、Azure Marketplace で入手できます。

次の手順を実行してテンプレートを起動し、Azure 可用性セットまたは可用性ゾーンを使用して高可用性 VPX ペアをデプロイします。

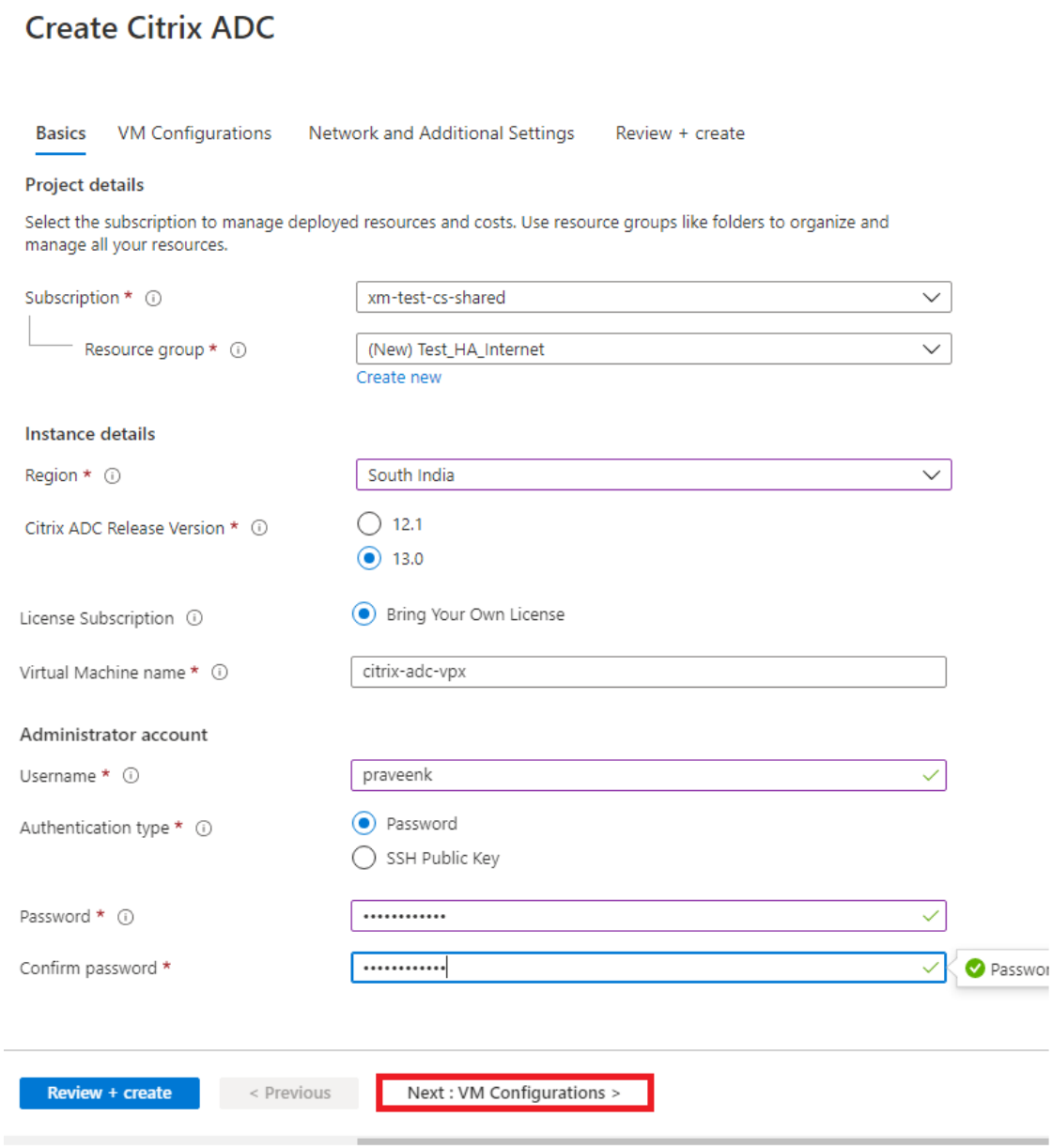
1. Azure Marketplace から **NetScaler** を検索してください。
2. [今すぐ入手] をクリックします。



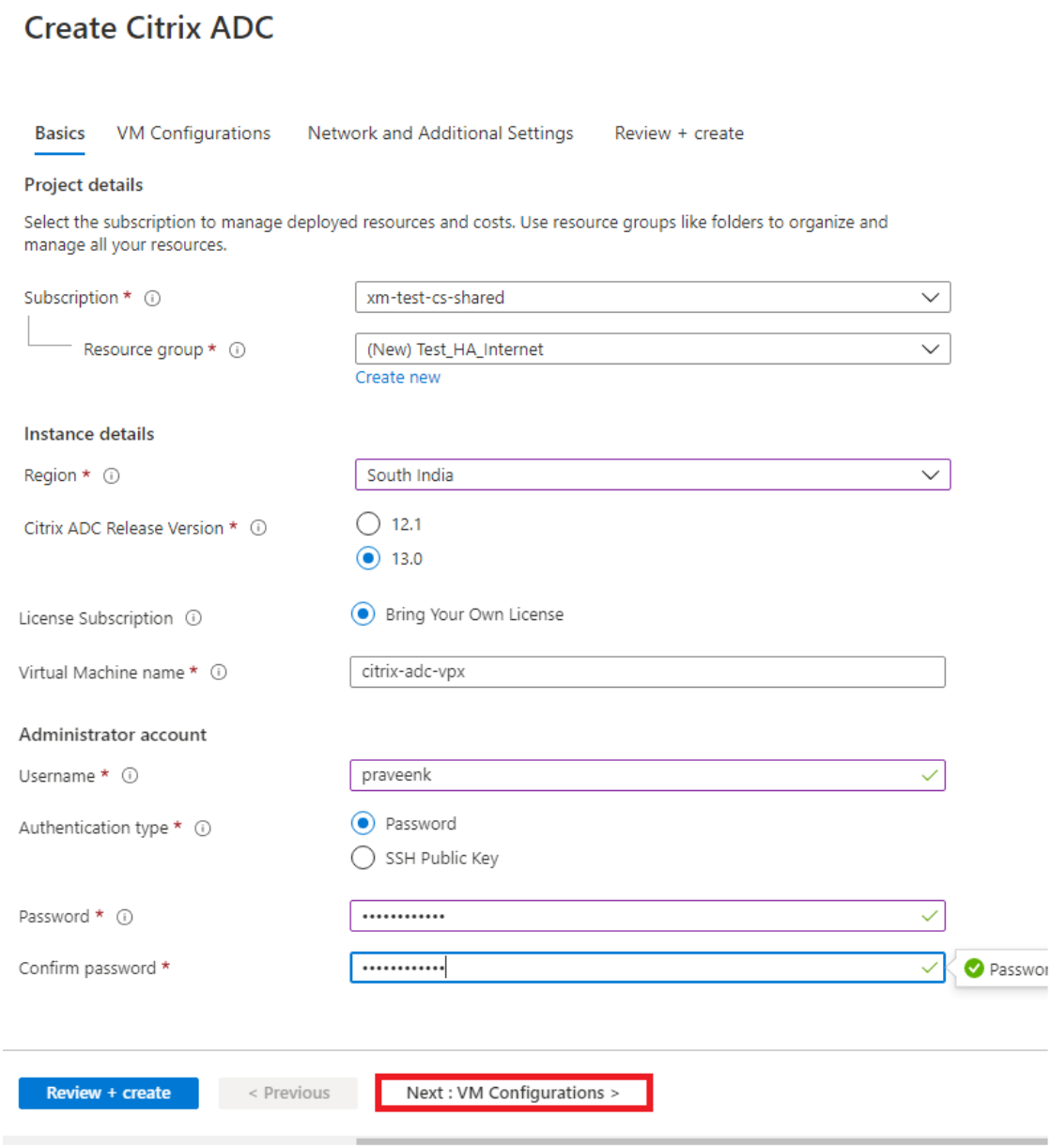
3. 必要な HA 導入とライセンスを選択し、[続行] をクリックします。



4. [基本] ページが表示されます。リソースグループを作成します。[パラメータ] タブで、地域、管理者ユーザー名、管理者パスワード、ライセンスタイプ（VM SKU）、およびその他のフィールドの詳細を入力します。



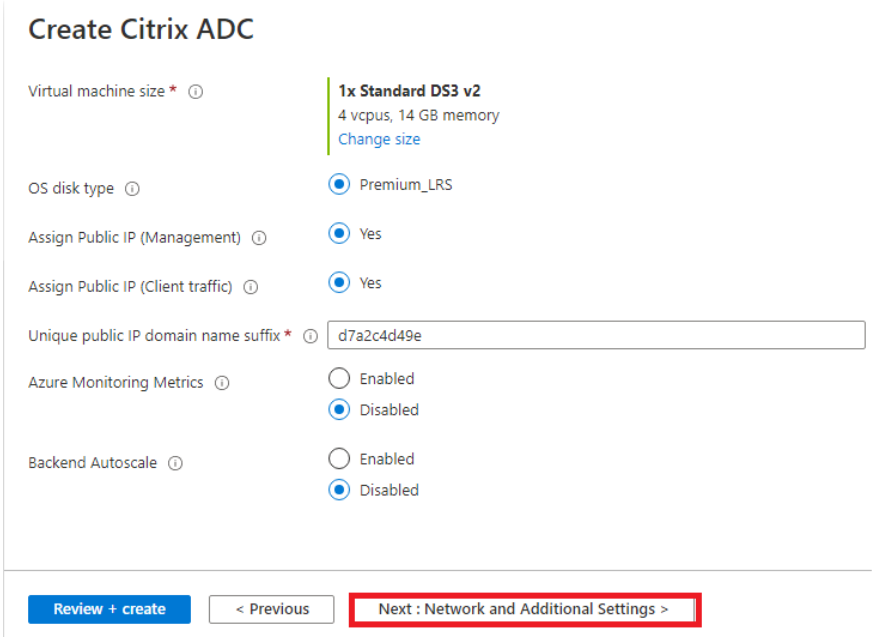
5. [次へ] をクリックします: VM 構成 > 。



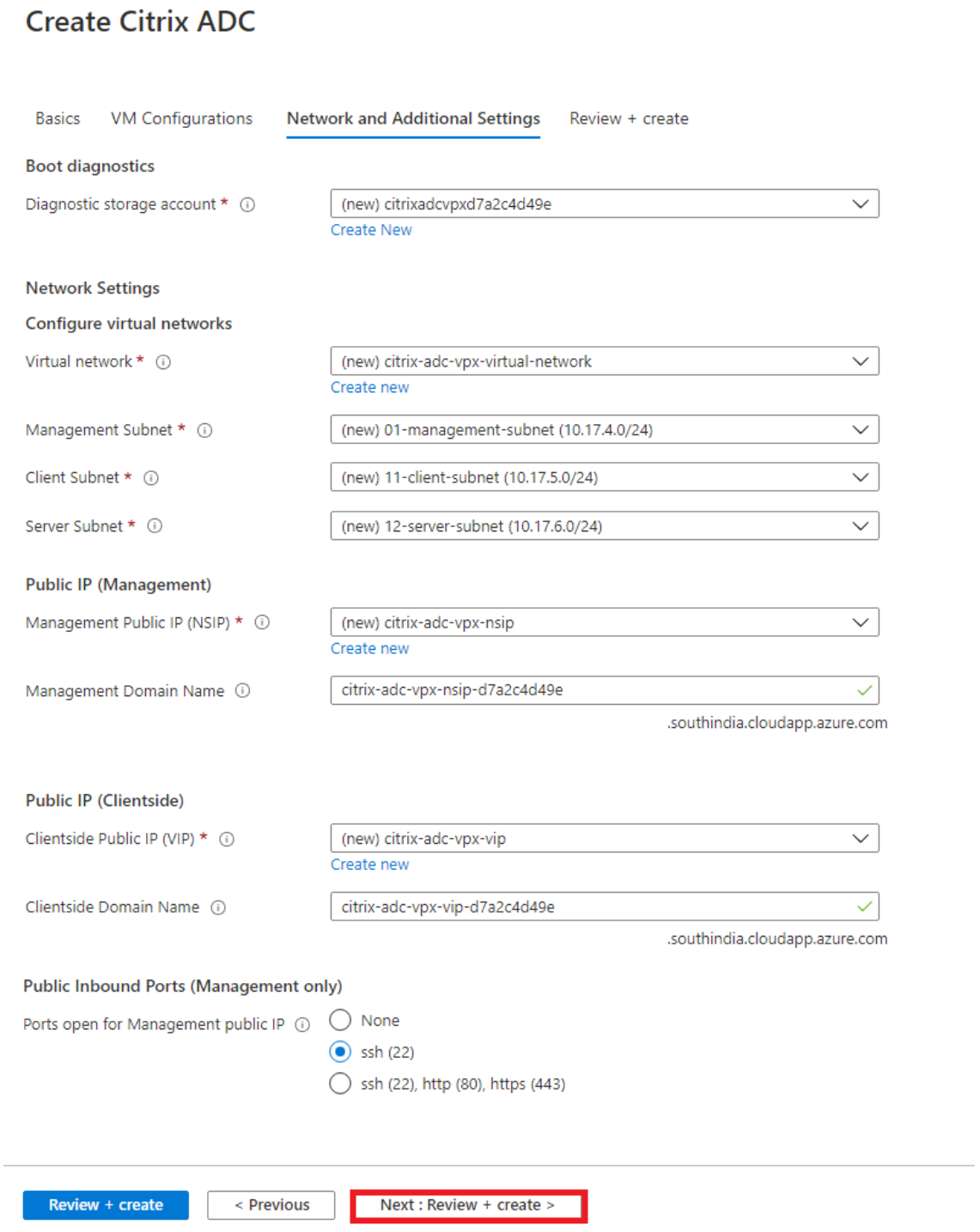
6. [**VM** 構成] ページで、次の手順を実行します。

   - パブリック IP ドメイン名サフィックスの設定
   - **Azure** 監視メトリクスを有効または無効にする
   - バックエンド **Autoscale** を有効または無効にする

7. [次へ**:** ネットワークとその他の設定] をクリックします。



8. [ネットワークとその他の設定] ページで、ブート診断アカウントを作成し、ネットワーク設定を行います。



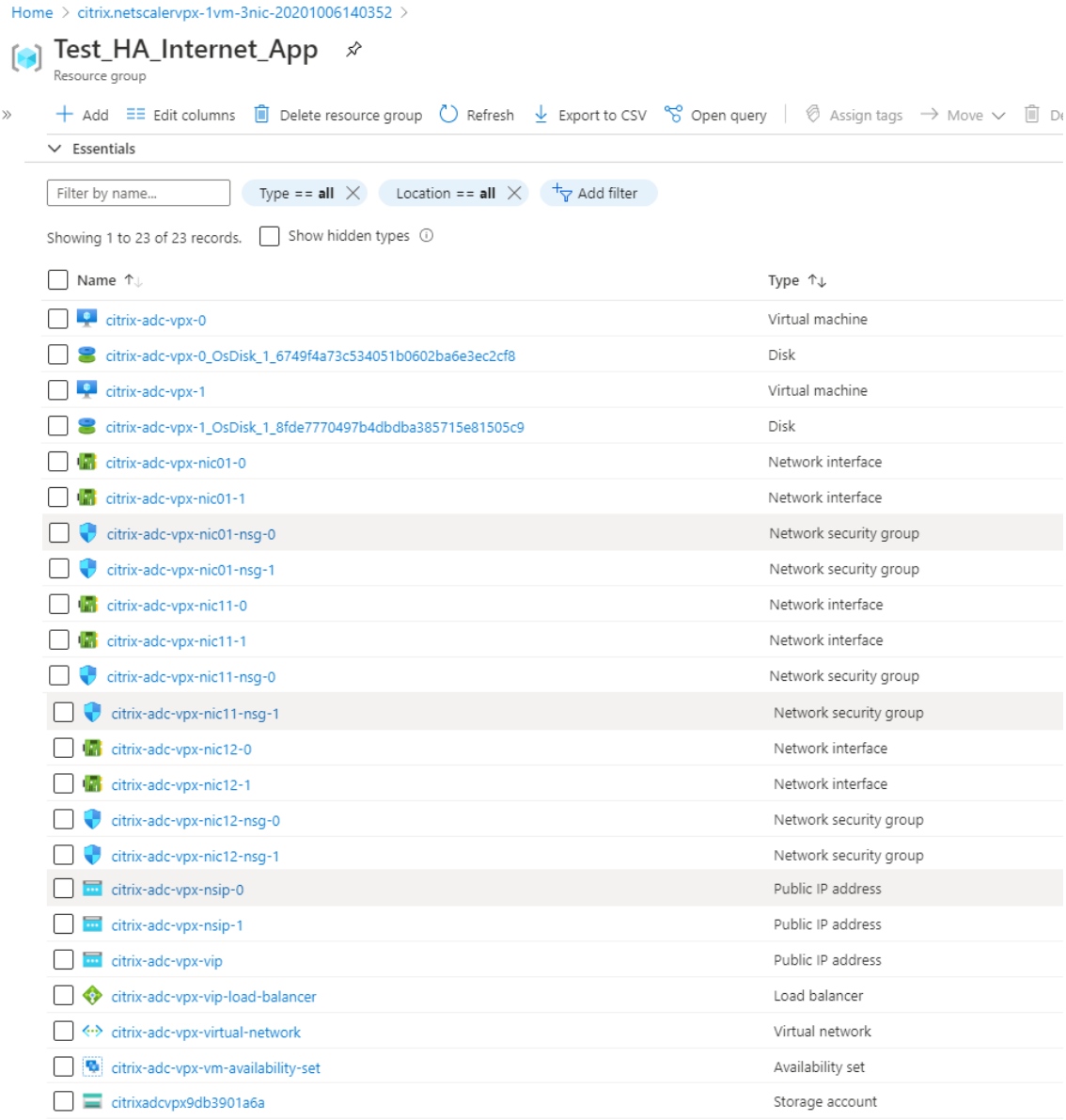
9. [** 次へ: レビュー + 作成 **] をクリックします。
10. 基本設定、VM 構成、ネットワーク、その他の設定を確認して、[作成] をクリックします。

    必要な構成で Azure リソースグループが作成されるまで時間がかかることがあります。完了したら、Azure

ポータルでリソースグループを選択すると、LB ルール、バックエンドプール、ヘルスプローブなどの構成の詳細が表示されます。高可用性ペアは、**citrix-adc-vpx-0** と **citrix-adc-vpx-1** として表示されます。

追加のセキュリティルールやポートを作成するなど、HA セットアップでさらに変更が必要な場合は、Azure Portal から実行できます。

必要な構成が完了すると、次のリソースが作成されます。



11. 次の構成を検証するには、citrix-adc-vpx-0 ノードと citrix-adc-vpx-1 ノードにログオンする必要があります。

    - 両方のノードの NSIP アドレスは管理サブネットに存在する必要があります。
    - プライマリ (citrix-adc-vpx-0) ノードとセカンダリ (citrix-adc-vpx-1) ノードには、2 つの SNIP アドレスが必要です。1 つの SNIP (クライアントサブネット) は ALB プローブへの応答に使用され、もう 1

つの SNIP (サーバーサブネット) はバックエンドサーバー通信に使用されます。

> 注
>
> HA-INC モードでは、citrix-adc-vpx-0 と citrix-adc-vpx-1 VM の SNIP アドレスは異なります。これは、両方が同じである従来のオンプレミス ADC 高可用性導入環境とは異なります。

プライマリノード (citrix-adc-vpx-0) で

```
> sh ip
        Ipaddress        Traffic Domain  Type            Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----            ----     ---      ----     -------  -------
1)      10.18.0.4        0               NetScaler IP    Active   Enabled  Enabled  NA       Enabled
2)      10.18.1.5        0               SNIP            Active   Enabled  Enabled  NA       Enabled
3)      10.18.2.4        0               SNIP            Active   Enabled  Enabled  NA       Enabled
 Done
```

```
> sh ha node
1)      Node ID:      0
        IP:       10.18.0.4 (ns-vpx0)
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:3:34:21 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.18.0.5
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
```

セカンダリノード (citrix-adc-vpx-1) 上

```
> show ip
        Ipaddress        Traffic Domain  Type            Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----            ----     ---      ----     -------  -------
1)      10.18.0.5        0               NetScaler IP    Active   Enabled  Enabled  NA       Enabled
2)      10.18.1.4        0               SNIP            Active   Enabled  Enabled  NA       Enabled
3)      10.18.2.5        0               SNIP            Active   Enabled  Enabled  NA       Enabled
 Done
>
```

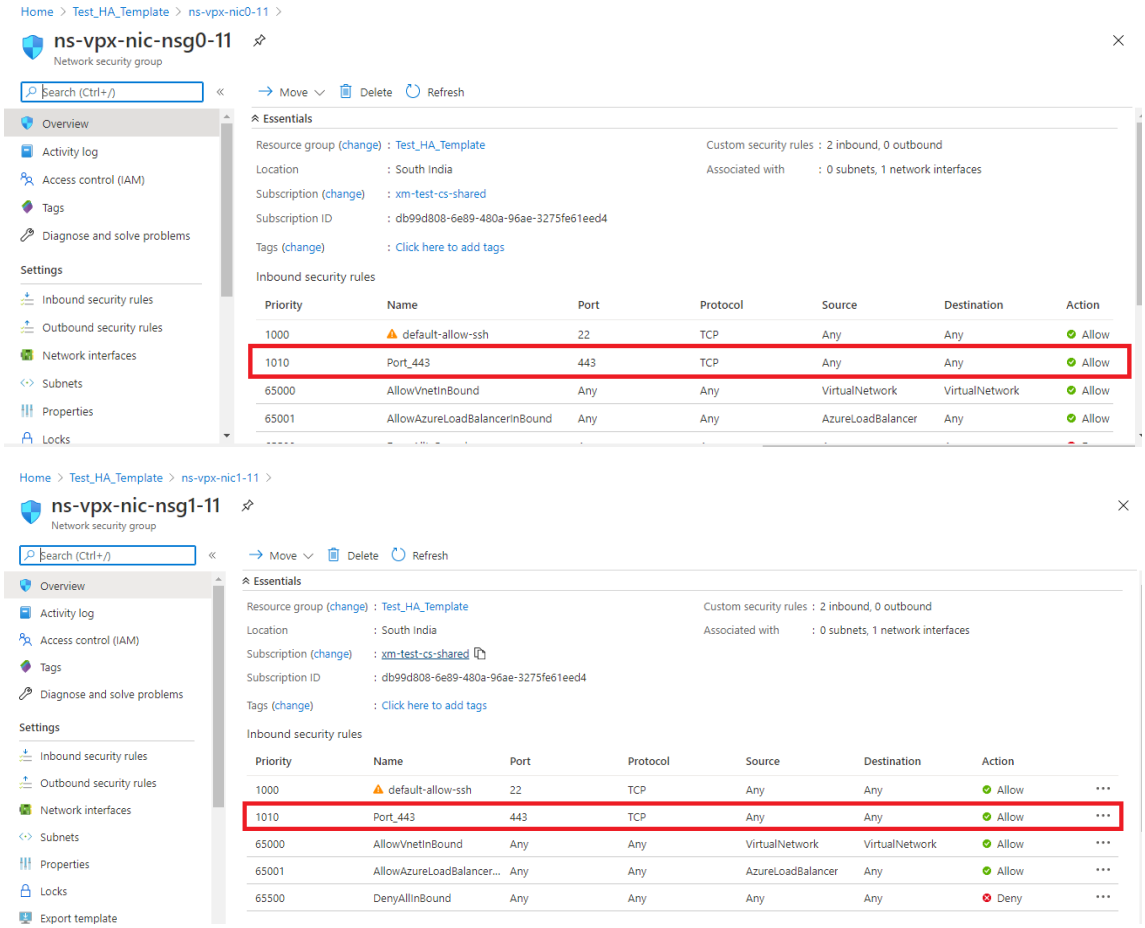


```
> sh ha node
1)      Node ID:      0
        IP:       10.18.0.5 (ns-vpx1)
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:3:23:51 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.18.0.4
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

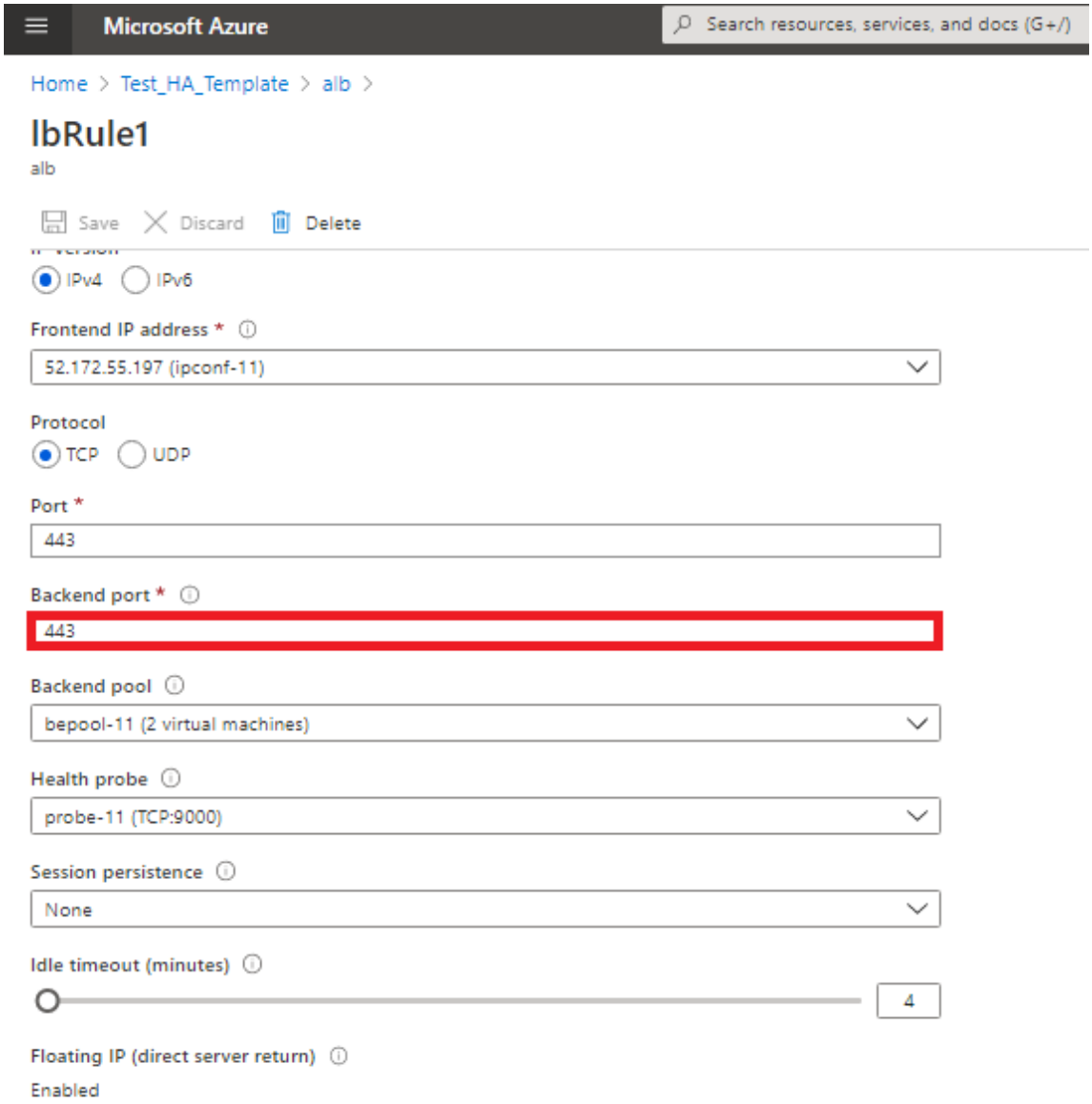
12. プライマリノードとセカンダリノードが UP になり、同期ステータスが **SUCCESS** になったら、ALB 仮想のパブリック IP アドレスを使用して、プライマリノード（citrix-adc-vpx-0）の負荷分散仮想サーバーまたはゲートウェイ仮想サーバーを構成する必要があります。サーバ。詳細については、「サンプル設定」セクションを参照してください。

13. ALB 仮想サーバーのパブリック IP アドレスを見つけるには、**Azure portal > Azure Load Balancer > Frontend IP configuration** に移動します。

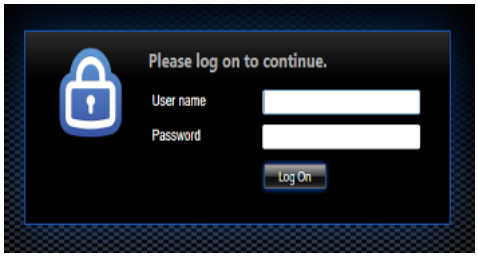
14. 仮想サーバーポート 443 のインバウンドセキュリティルールを、両方のクライアントインターフェイスのネットワークセキュリティグループに追加します。



15. アクセスする ALB ポートを設定し、指定したポートのインバウンドセキュリティルールを作成します。バックエンドポートは、負荷分散仮想サーバーポートまたは VPN 仮想サーバーポートです。



16. これで、ALB パブリック IP アドレスに関連付けられた完全修飾ドメイン名 (FQDN) を使用して、負荷分散仮想サーバーまたは VPN 仮想サーバーにアクセスできます。



設定例

ゲートウェイ VPN 仮想サーバーと負荷分散仮想サーバーを設定するには、プライマリノード (ADC-VPX-0) で次のコマンドを実行します。設定はセカンダリノード（ADC-VPX-1）に自動的に同期されます。

ゲートウェイのサンプル構成

```
enable feature aaa LB SSL SSLVPN
add ip 52.172.55.197 255.255.255.0 -type VIP
add vpn vserver vpn_ssl SSL 52.172.55.197 443
add ssl certKey ckp -cert cgwsanity.cer -key cgwsanity.key
bind ssl vserver vpn_ssl -certkeyName ckp
<!--NeedCopy-->
```

負荷分散のサンプル構成

```
enable feature LB SSL
enable ns mode MBF
add lb vserver lb_vs1 SSL 52.172.55.197 443
bind ssl vserver lb_vs1 -certkeyName ckp
<!--NeedCopy-->
```

ALB のパブリック IP アドレスに関連付けられた FQDN を使用して、負荷分散または VPN 仮想サーバーにアクセスできるようになりました。

負荷分散仮想サーバーを構成する方法の詳細については、「リソース」セクションを参照してください。

リソース:

次のリンクには、HA の導入と仮想サーバの設定に関する追加情報が表示されます。

- 仮想サーバーを作成する
- 基本的な負荷分散を設定する

Azure の外部ロードバランサーと内部ロードバランサーを同時に使用して高可用性設定を構成する

August 15, 2023

Azure の高可用性ペアは、外部ロードバランサーと内部ロードバランサーの両方を同時にサポートします。

Azure 外部ロードバランサーと内部ロードバランサーの両方を使用して高可用性ペアを構成するには、次の 2 つのオプションがあります。

- NetScaler ADC アプライアンス上で 2 つの LB 仮想サーバーを使用する。
- 1 つの LB 仮想サーバーと IP セットを使用する。単一の LB 仮想サーバは、IPSet によって定義された複数の IP にトラフィックを処理します。

外部ロードバランサーと内部ロードバランサーを同時に使用して Azure で高可用性ペアを構成するには、次の手順を実行します。

手順 1 と 2 については、Azure ポータルを使用します。手順 3 および 4 では、NetScaler VPX GUI または CLI を使用します。

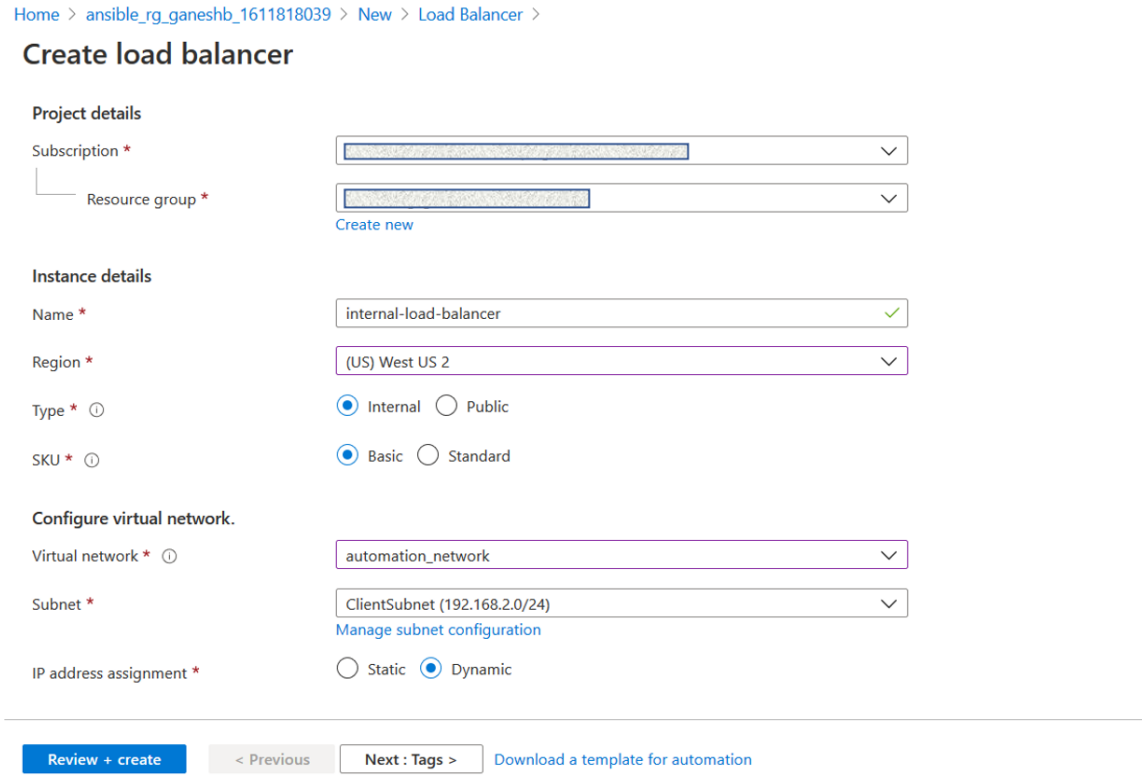
手順 **1.** Azure ロードバランサー (外部ロードバランサーまたは内部ロードバランサー) を構成します。

Azure 外部ロードバランサーを使用した高可用性セットアップの構成の詳細については、「複数の IP アドレスと NIC を使用した高可用性セットアップを構成する」を参照してください。

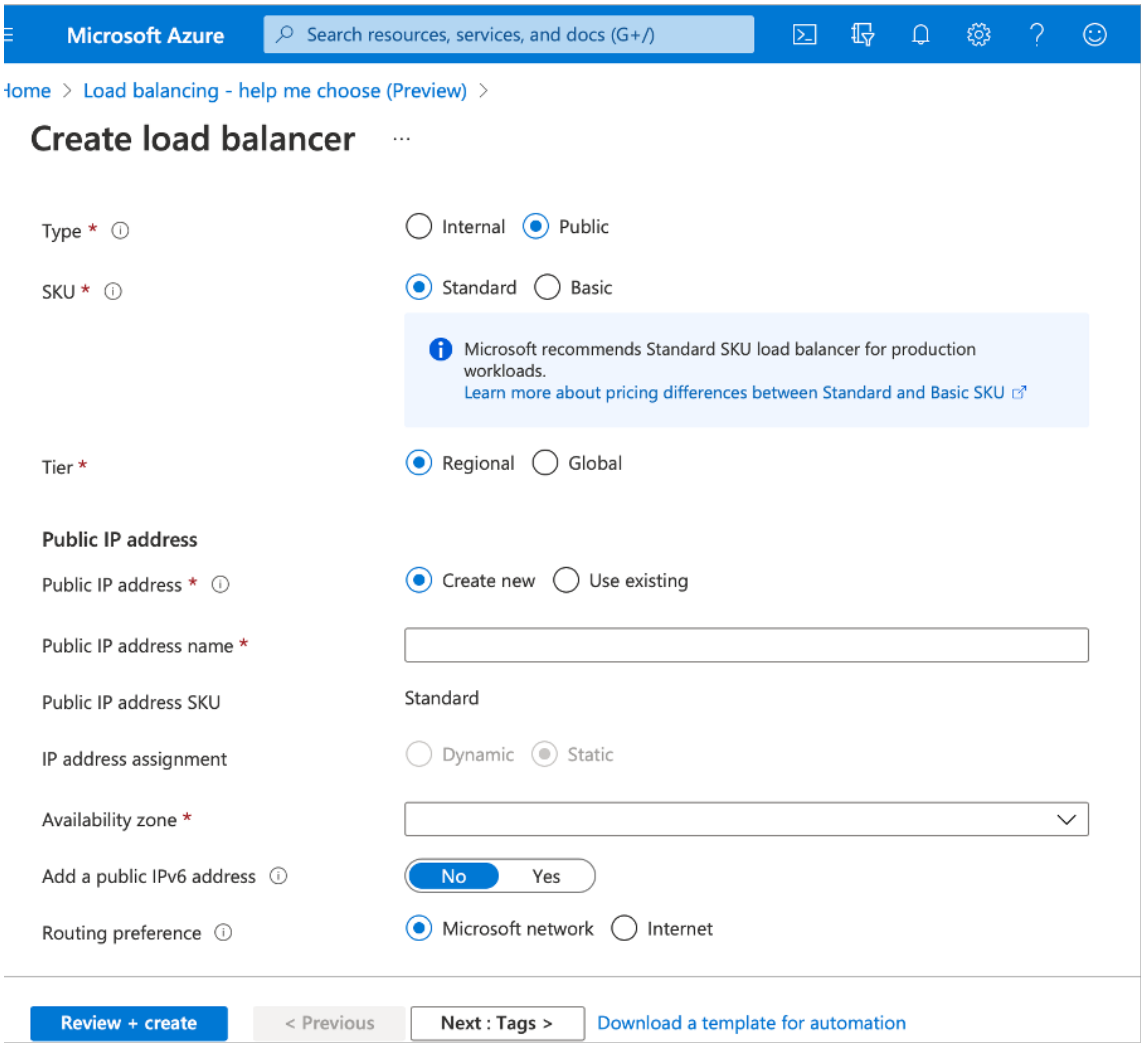
Azure 内部ロードバランサーによる高可用性セットアップの構成について詳しくは、「Azure ILB で NetScaler 高可用性テンプレートを使用して HA-INC ノードを構成する」を参照してください。

手順 **2.** リソースグループに追加のロードバランサー (ILB) を作成します。ステップ 1 では、外部ロードバランサーを作成した場合は、内部ロードバランサーを作成し、逆に作成します。

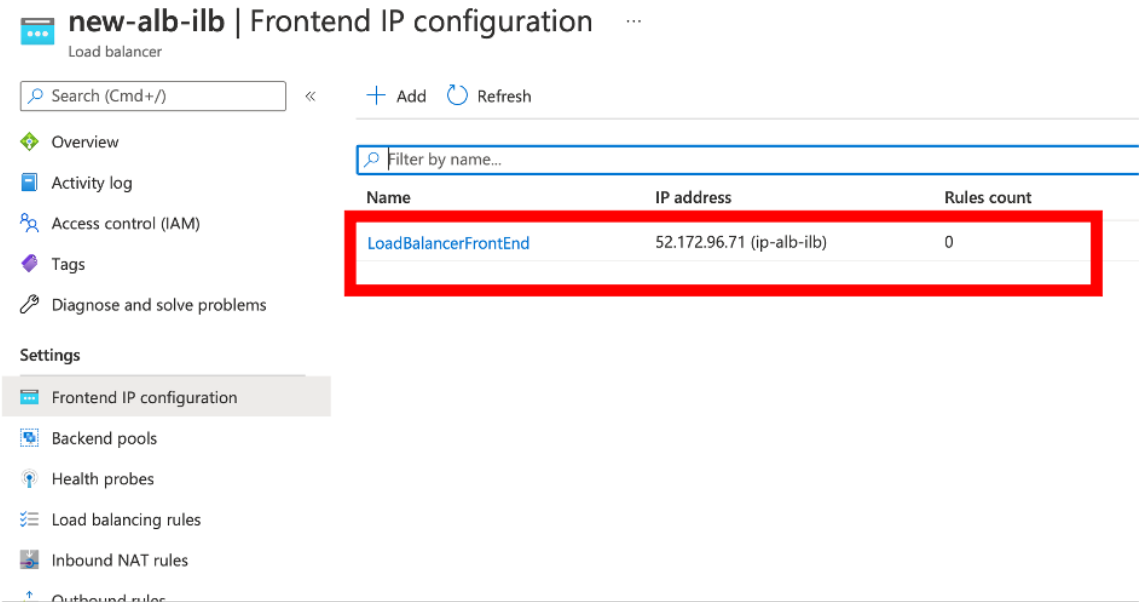
- 内部ロードバランサーを作成するには、ロードバランサのタイプを [内部] として選択します。[サブネット] フィールドで、NetScaler ADC クライアントサブネットを選択する必要があります。競合がない限り、そのサブネットに静的 IP アドレスを指定することもできます。それ以外の場合は、ダイナミック IP アドレスを選択します。



- 外部ロードバランサーを作成するには、ロードバランサの種類を [パブリック] として選択し、ここにパブリック IP アドレスを作成します。



1. Azure Load Balancer を作成したら、フロントエンド **IP** 設定に移動し、ここに示す IP アドレスを書き留めます。ステップ 3 のように ADC 負荷分散仮想サーバーを作成するときは、この IP アドレスを使用する必要があります。



2. **Azure Load Balancer** の設定ページでは、ARM テンプレートをデプロイすることで LB ルール、バックエンドプール、ヘルスプローブを作成できます。
3. 高可用性ペアのクライアント NIC を ILB のバックエンドプールに追加します。
4. ヘルスプローブの作成（TCP、9000 ポート）
5. 次の 2 つのロードバランシングルールを作成します。
    - ポート 80 の HTTP トラフィック（Webapp ユースケース）の 1 つの LB ルール。ルールでは、バックエンドポート 80 も使用する必要があります。作成したバックエンドプールとヘルスプローブを選択します。フローティング IP を有効にする必要があります。
    - ポート 443 の HTTPS または CVAD トラフィックに対する別の LB ルール。プロセスは HTTP トラフィックと同じです。

手順 **3.** NetScaler ADC アプライアンスのプライマリノードで、ILB の負荷分散仮想サーバーを作成します。

1. 負荷分散仮想サーバーを追加します。

```
add lb vserver <name> <serviceType> [<ILB Frontend IP address>] [<
   port>]
<!--NeedCopy-->
```

例**:**

```
add lb vserver vserver_name HTTP 52.172.96.71 80
<!--NeedCopy-->
```

> 注:
>
> ステップ 2 で作成した追加のロードバランサーに関連付けられた、ロードバランサーのフロントエンド

> IP アドレスを使用します。

2. サービスを負荷分散仮想サーバーにバインドします。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

詳細については、「基本的な負荷分散の設定」を参照してください。

ステップ **4:** ステップ 3 の代わりに、IPSet を使用して ILB の負荷分散仮想サーバーを作成できます。

1. 仮想サーバー IP (VIP) タイプの IP アドレスを追加します。

```
add nsip <ILB Frontend IP address> -type <type>
<!--NeedCopy-->
```

例:

```
add nsip 52.172.96.71 -type vip
<!--NeedCopy-->
```

2. プライマリノードとセカンダリノードの両方に IPSet を追加します。

```
add ipset <name>
<!--NeedCopy-->
```

例:

```
add ipset ipset1
<!--NeedCopy-->
```

3. IP アドレスを IP セットにバインドします。

```
bind ipset <name> <ILB Frontend IP address>
<!--NeedCopy-->
```

例:

```
bind ipset ipset1 52.172.96.71
<!--NeedCopy-->
```

4. 既存の LB 仮想サーバーを IPSet を使用するように設定します。

```
set lb vserver <vserver name> -ipset <ipset name>
<!--NeedCopy-->
```

例:


```
set lb vserver vserver_name -ipset ipset1
<!--NeedCopy-->
```

詳細については、「マルチ IP 仮想サーバーの構成」を参照してください。

Azure VMware ソリューションに NetScaler VPX インスタンスをインストールする

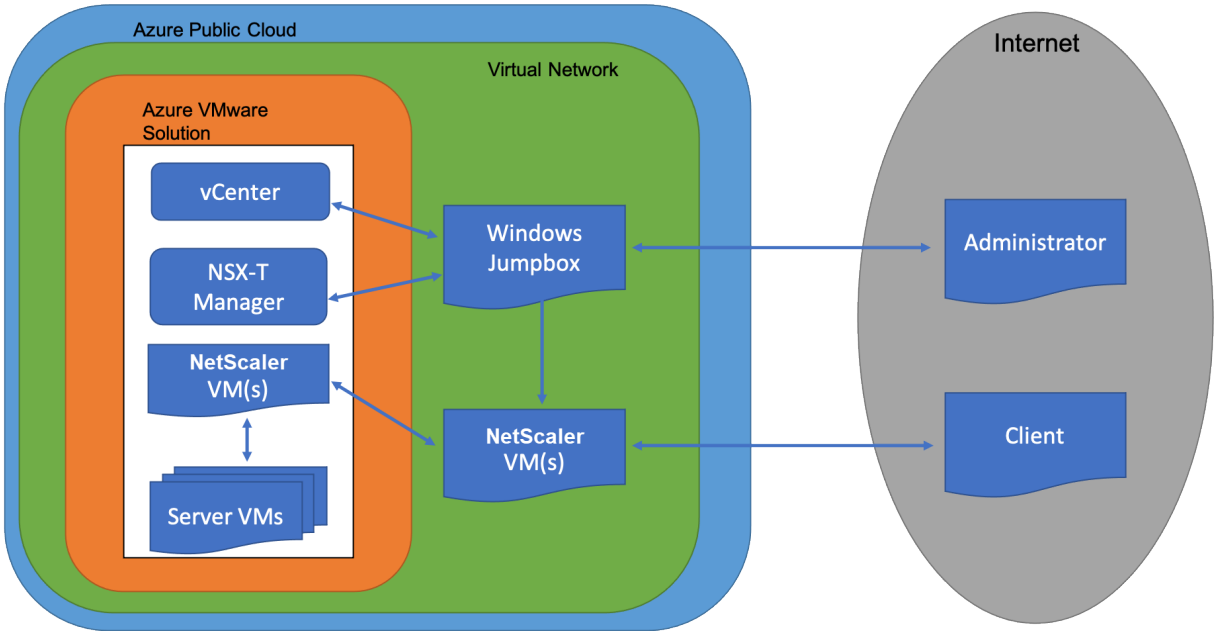
December 8, 2023

Azure VMware ソリューション (AVS) は、専用のベアメタル Azure インフラストラクチャから構築された vSphere クラスタを含むプライベートクラウドを提供します。最初のデプロイメントは最小で 3 台のホストですが、追加ホストは一度に 1 つずつ追加でき、クラスタごとに最大 16 台のホストを追加できます。プロビジョニングされたすべてのプライベートクラウドには、vCenter Server、vSAN、vSphere、NSX-T があります。

Azure 上の VMware クラウド (VMC) を使用すると、必要な数の ESX ホストを使用して Azure 上にクラウドソフトウェア定義データセンター (SDDC) を作成できます。Azure 上の VMC は、NetScaler VPX デプロイメントをサポートしています。VMC は、オンプレミスの vCenter と同じユーザー・インタフェースを提供します。これは、ESX ベースの NetScaler VPX 展開と同様に機能します。

次の図は、管理者またはクライアントがインターネット経由でアクセスできる Azure パブリッククラウド上の Azure VMware ソリューションを示しています。管理者は、Azure VMware ソリューションを使用して、ワークロードまたはサーバー仮想マシンを作成、管理、および構成できます。管理者は、Windows ジャンプボックスから AVS の Web ベースの vCenter および NSX-T マネージャにアクセスできます。vCenter を使用して Azure VMware Solution 内に NetScaler VPX インスタンス（スタンドアロンまたは高可用性ペア）とサーバー仮想マシンを作成し、NSX-T Manager を使用して対応するネットワークを管理できます。AVS 上の NetScaler VPX インスタンスは、オンプレミスの VMware ホストのクラスタと同様に機能します。AVS は、同じ仮想ネットワーク内に作成された Windows ジャンプボックスから管理されます。

クライアントは、ADC の VIP に接続することによってのみ AVS サービスにアクセスできます。Azure VMware ソリューション外の別の NetScaler VPX インスタンスは、同じ Azure 仮想ネットワーク内にある別の NetScaler VPX インスタンスは、Azure VMware ソリューション内の NetScaler VPX インスタンスの VIP をサービスとして追加するのに役立ちます。要件に応じて、インターネット上でサービスを提供するように NetScaler VPX インスタンスを構成できます。



前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- Azure VMware ソリューションとその前提条件の詳細については、Azure VMware ソリューションのドキュメントを参照してください。
- Azure VMware ソリューションのデプロイの詳細については、「Azure VMware ソリューションのプライベートクラウドをデプロイする」を参照してください。
- Azure VMware ソリューションにアクセスして管理するための Windows ジャンプボックス仮想マシンの作成の詳細については、「Azure VMware ソリューションのプライベートクラウドにアクセスする」を参照してください。
- Windows ジャンプボックス仮想マシンで、NetScaler VPX アプライアンスセットアップファイルをダウンロードします。
- 仮想マシンが接続する VMware SDDC 上に、適切な NSX-T ネットワークセグメントを作成します。詳細については、「Azure VMware ソリューションでのネットワークセグメントの追加」を参照してください。
- VPX ライセンスファイルを入手します。
- Azure VMware Solution プライベートクラウドに作成または移行された仮想マシン (VM) は、ネットワークセグメントに接続する必要があります。

VMware クラウドのハードウェア要件

次の表に、VMware SDDC が各 VPX nCore 仮想アプライアンスに対して提供する必要がある仮想コンピューティングリソースを示します。

表 1. NetScaler VPX インスタンスの実行に必要な最小限の仮想コンピューティングリソース

コンポーネント	条件
メモリ	2 GB
仮想 CPU（VCPU）	2
仮想ネットワークインターフェイス	VMware SDDC では、VPX ハードウェアをバージョン 7 以降にアップグレードした場合、最大 10 個の仮想ネットワークインターフェイスをインストールできます。
ディスク領域	20 GB

> 注:
>
> ハイパーバイザーに必要なディスク領域は含まれません。

VPX 仮想アプライアンスを実稼働で使用するには、フルメモリ割り当てを予約する必要があります。

OVF ツール 1.0 のシステム要件

Open Virtualization Format Tool（OVF Tool）は、Windows および Linux システムで実行できるクライアントアプリケーションです。次の表に、OVF ツールをインストールするためのシステム要件を示します。

表 2. OVF ツールのインストールに関するシステム要件

コンポーネント	条件
オペレーティングシステム	VMware からの詳細な要件については、http://kb.vmware.com/で『OVF ツールユーザーガイド』の PDF ファイルを検索してください。
CPU	最低 750MHz、1GHz 以上推奨
RAM	最小 1 GB、推奨 2 GB
NIC	100Mbps 以上の NIC。

OVF のインストールについては、http://kb.vmware.com/で『OVF ツールユーザーガイド』の PDF ファイルを検索してください。

NetScaler VPX セットアップファイルのダウンロード

VMware ESX 用の NetScaler VPX インスタンスセットアップパッケージは、オープン仮想マシン（OVF）フォーマット標準に準拠しています。これらのファイルは、Citrix Web サイトからダウンロードできます。ログオンするに

は、Citrix アカウントが必要です。Citrix アカウントをお持ちでない場合は、http://www.citrix.comのホームページにアクセスしてください。［新しいユーザー］リンクをクリックし、指示に従って新しい Citrix アカウントを作成します。

ログオンしたら、Citrix のホームページから次のパスをナビゲートします。

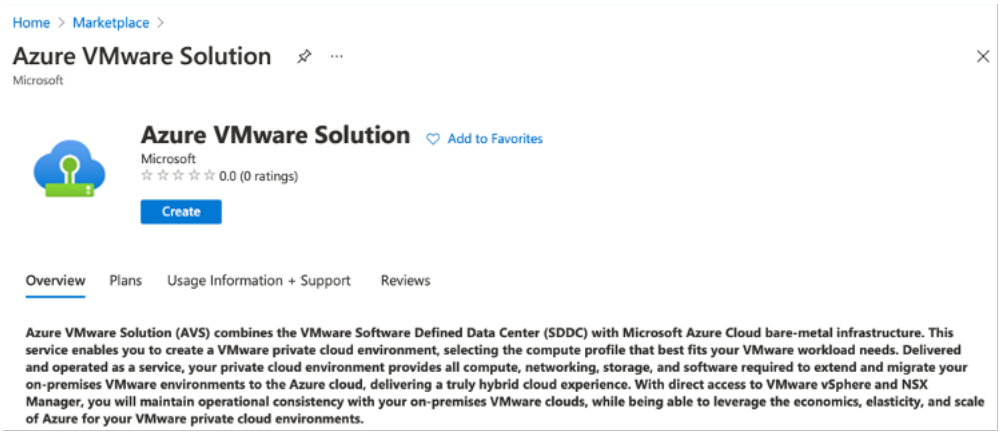
Citrix.com > ダウンロード **> NetScaler >** 仮想アプライアンス。

次のファイルを、ESX サーバーと同じネットワーク上のワークステーションにコピーします。3 つのファイルをすべて同じフォルダーにコピーします。

- NSVPX-ESX-<release number>--disk1.vmdk <build number> (たとえば、nsvpx-ESX-13.0-79.64-disk1.vmdk)
- NSVPX-ESX-<release number>-.ovf <build number> (たとえば、nsvpx-ESX-13.0-79.64.OVF)
- NSVPX-ESX-<release number>-.mf <build number> (たとえば、nsvpx-ESX-13.0-79.64.mf)

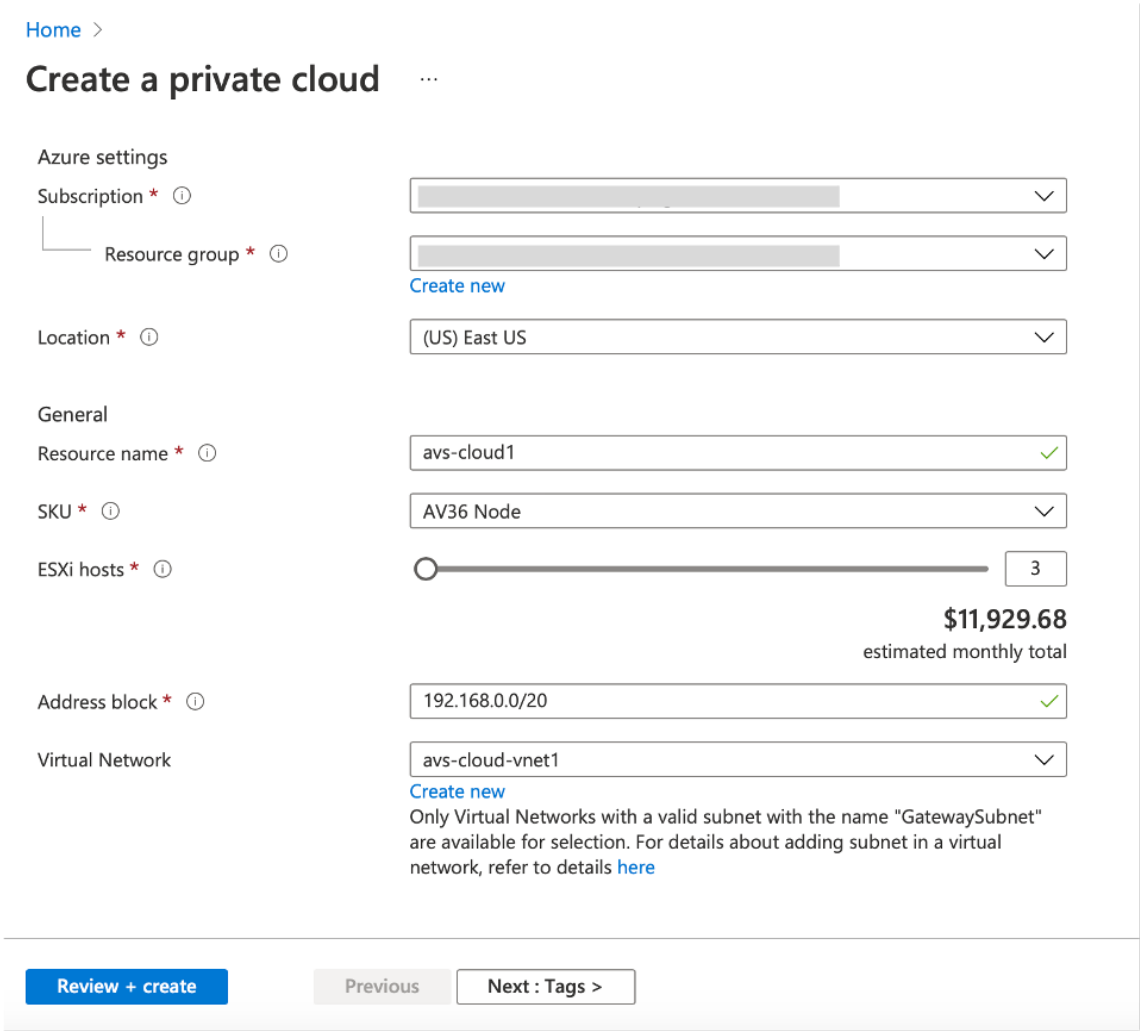
Azure VMware ソリューションをデプロイする

1. Microsoft Azure ポータルにログインし、Azureマーケットプレイスに移動します。
2. **Azure** マーケットプレイスから **AzureVMware** ソリューションを検索し、[作成] をクリックします。

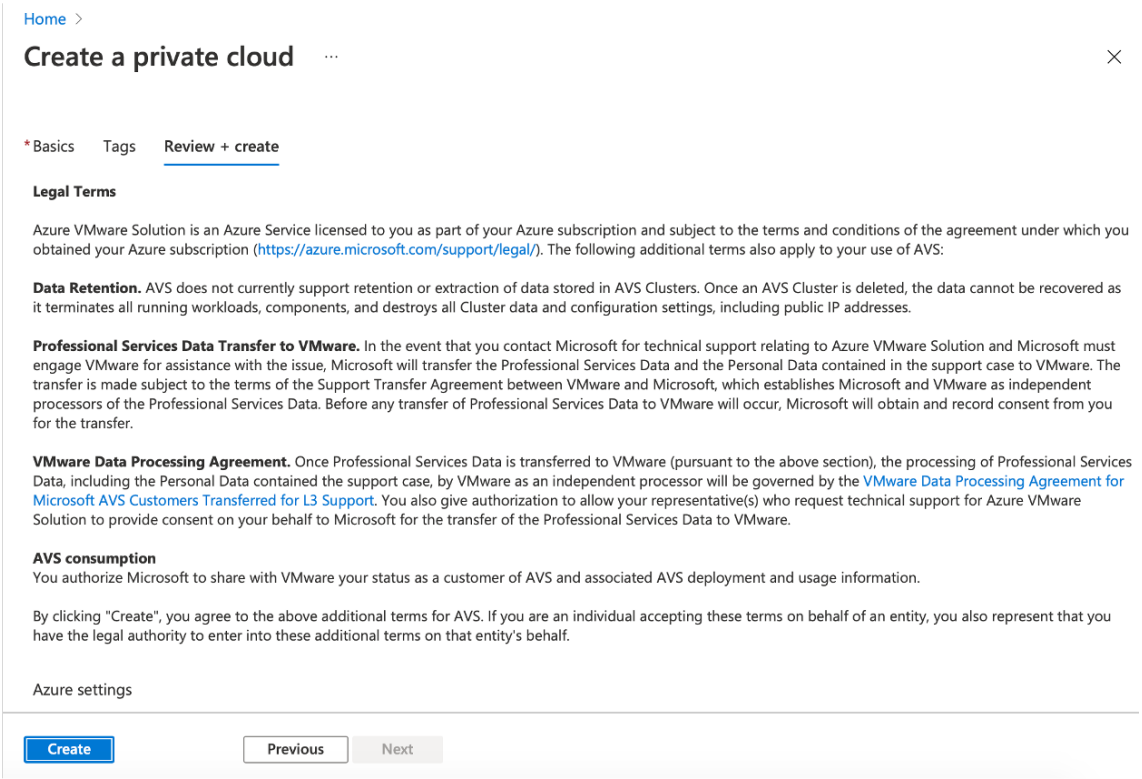


3. [プライベートクラウドの作成] ページで、次の詳細を入力します。

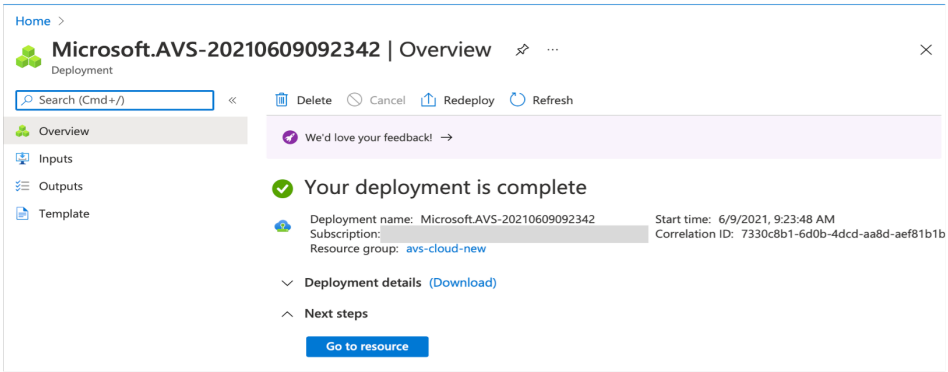
   - プライベートクラウドのデフォルトクラスタを作成するには、最低 3 つの ESXi ホストを選択します。
   - [**Address** ブロック] フィールドには、**/22** アドレス空間を使用します。
   - 仮想ネットワークの場合、CIDR 範囲が、オンプレミスまたはその他の Azure サブネット (仮想ネットワーク) またはゲートウェイサブネットと重複していないことを確認します。
   - ゲートウェイサブネットは、プライベートクラウドとの接続のルーティングを表現するために使用されます。



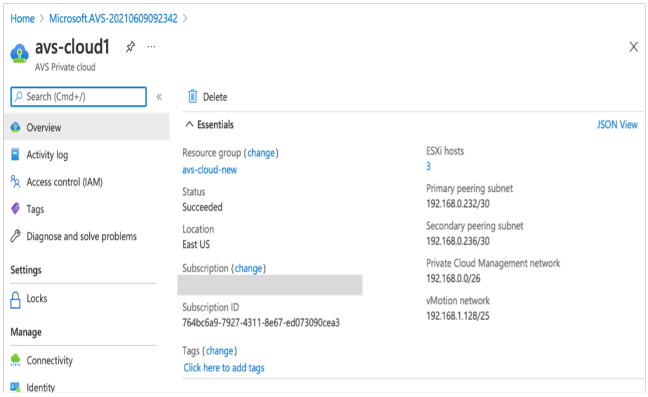
4. [レビュー**] + [**作成] をクリックします。
5. 設定を確認します。設定を変更する必要がある場合は、[前へ] をクリックします。



6. [作成] をクリックします。プライベートクラウドのプロビジョニングプロセスが開始されます。プライベートクラウドのプロビジョニングには最大 2 時間かかることがあります。



7. [リソースに移動] をクリックして、作成されたプライベートクラウドを確認します。



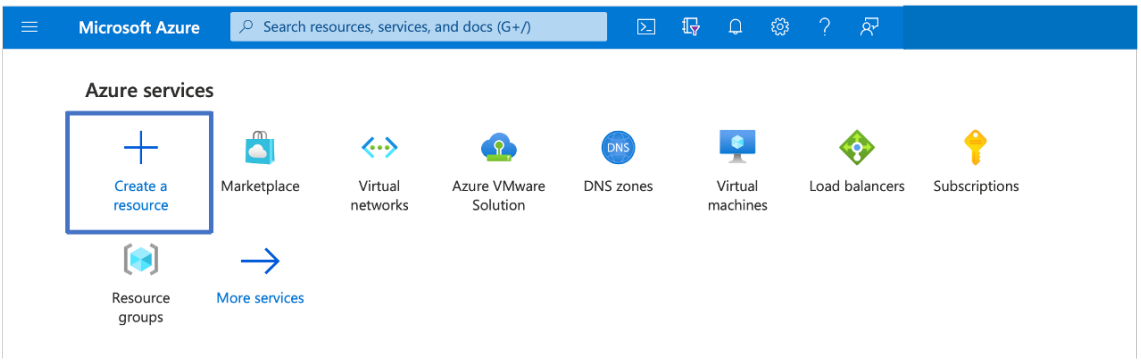
> 注
>
> このリソースにアクセスするには、Windows でジャンプボックスとして機能する仮想マシンが必要です。

Windows を実行している **Azure** 仮想マシンに接続する

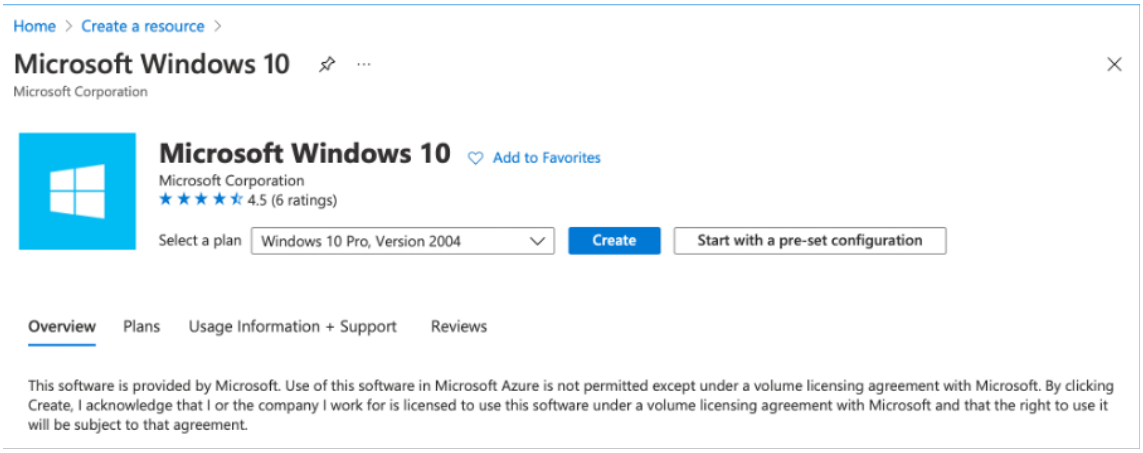
この手順では、Azure ポータルを使用して、Windows Server 2019 を実行する仮想マシン (VM) を Azure にデプロイする方法について説明します。VM の動作を確認するには、仮想マシンに RDP し、IIS Web サーバーをインストールします。

作成したプライベートクラウドにアクセスするには、同じ仮想ネットワーク内に Windows ジャンプボックスを作成する必要があります。

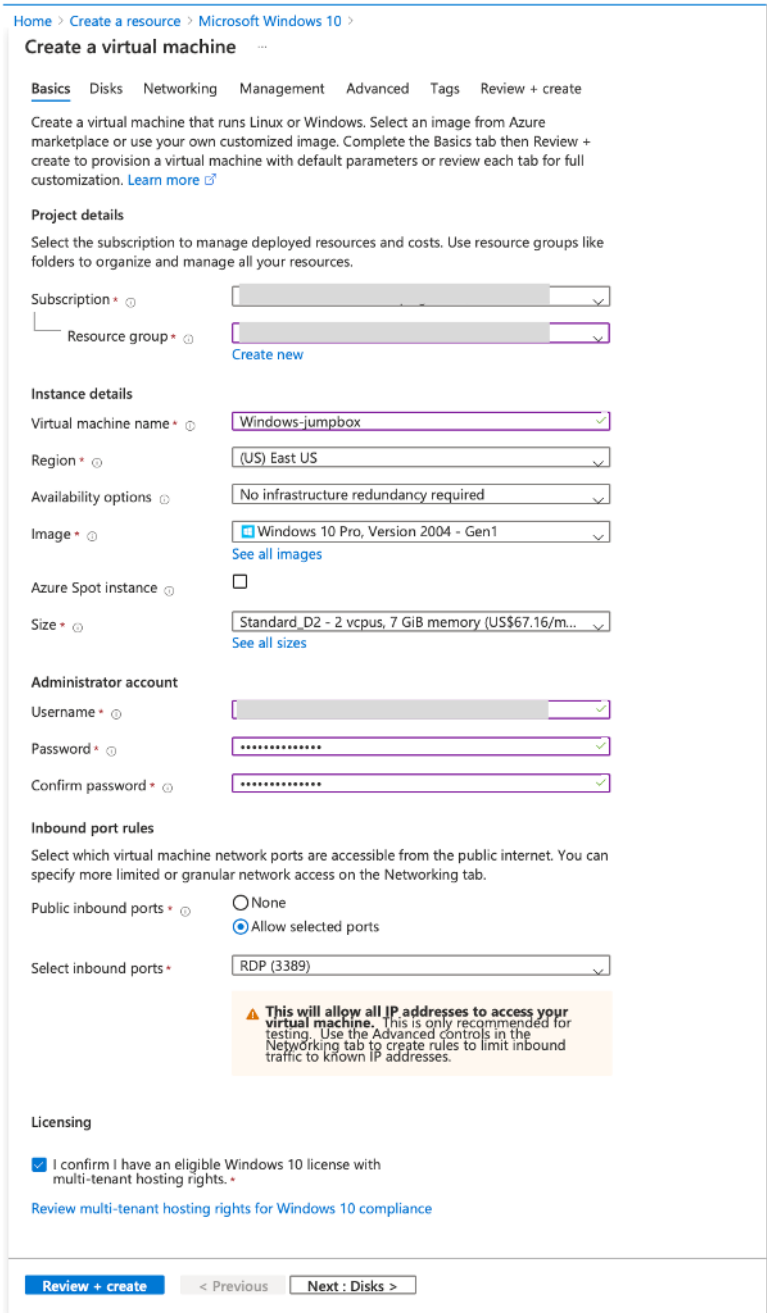
1. **Azure** ポータルに移動し、[リソースの作成] をクリックします。



2. **Microsoft Windows 10** を検索し、[作成] をクリックします。



3. Windows Server 2019 を実行する仮想マシン (VM) を作成します。[仮想マシンの作成] ページが表示されます。[基本] タブにすべての詳細を入力し、[ライセンス] チェックボックスをオンにします。残りのデフォルトのままにして、ページの下部にある [**Review + create**] ボタンを選択します。



4. 検証の実行後、ページの下部にある [作成] ボタンを選択します。
5. デプロイが完了したら、[リソースに移動] を選択します。
6. 作成した Windows 仮想マシンに移動します。Windows 仮想マシンのパブリック IP アドレスを使用し、RDP を使用して接続します。

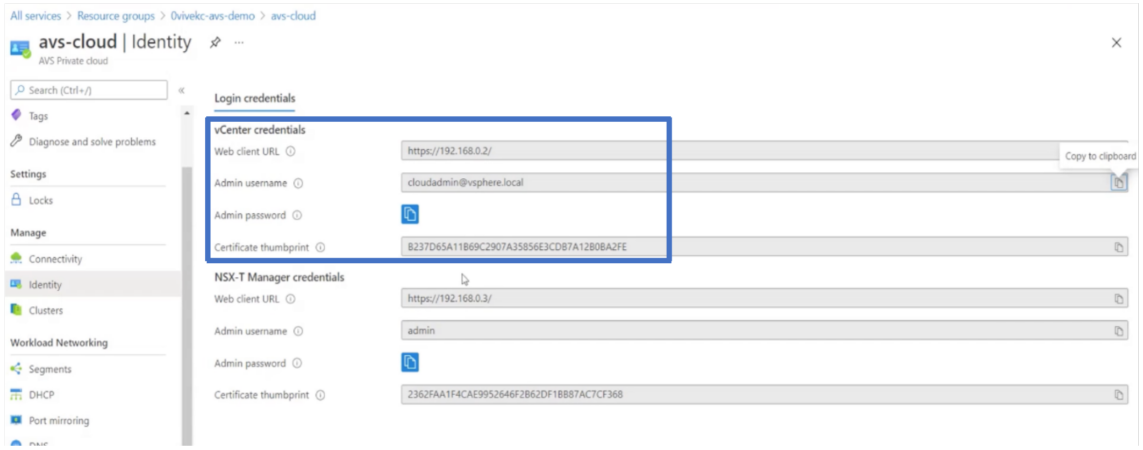
   Azure ポータルの [接続] ボタンを使用して、Windows デスクトップからリモートデスクトップ (RDP) セッションを開始します。まず仮想マシンに接続し、次にサインオンします。

   Mac から Windows 仮想マシンに接続するには、Microsoft リモートデスクトップなどの Mac 用 RDP クラ

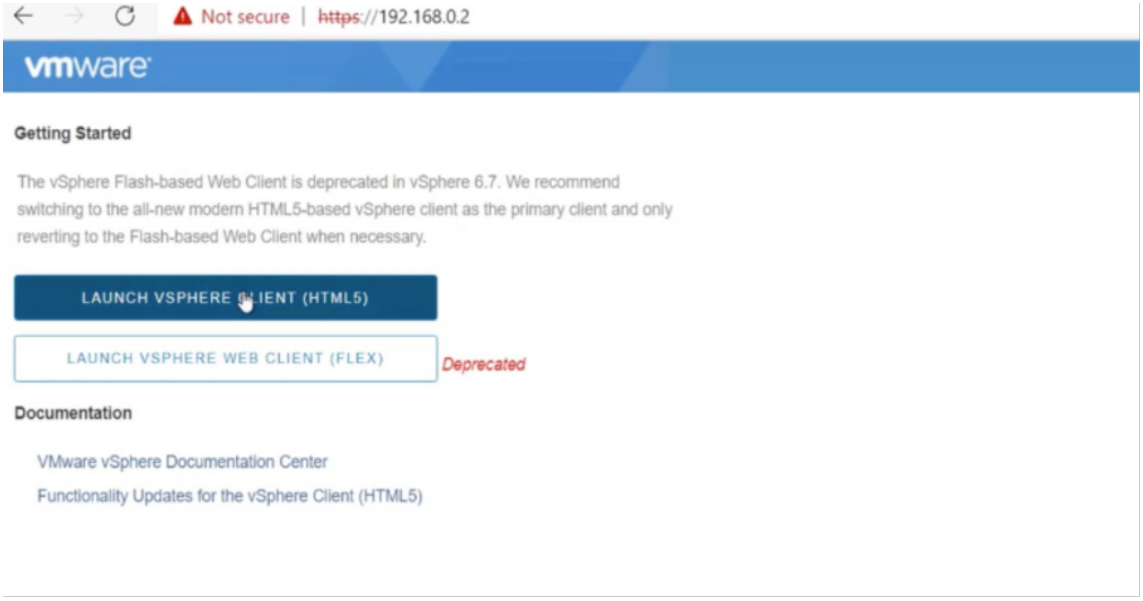
イアントをインストールする必要があります。詳細については、「Windows を実行する Azure 仮想マシンに接続してサインオンする方法」を参照してください。

プライベートクラウド **vCenter** ポータルにアクセスする

1. Azure VMware ソリューションのプライベートクラウドで、[管理] で [アイデンティティ] を選択します。vCenter の認証情報を書き留めます。

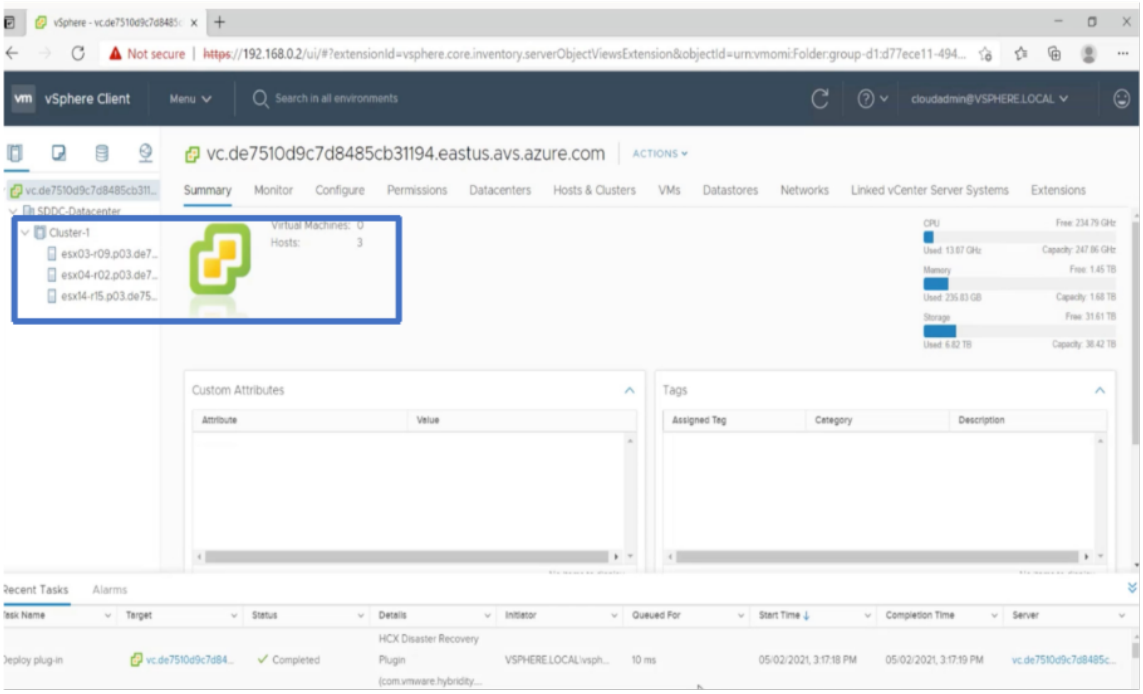


2. vCenter Web クライアントの URL を入力して、vSphere クライアントを起動します。



3. Azure VMware ソリューションプライベートクラウドの vCenter 認証情報を使用して VMware vSphere にログインします。

4. vSphere クライアントでは、Azure ポータルで作成した ESXi ホストを確認できます。



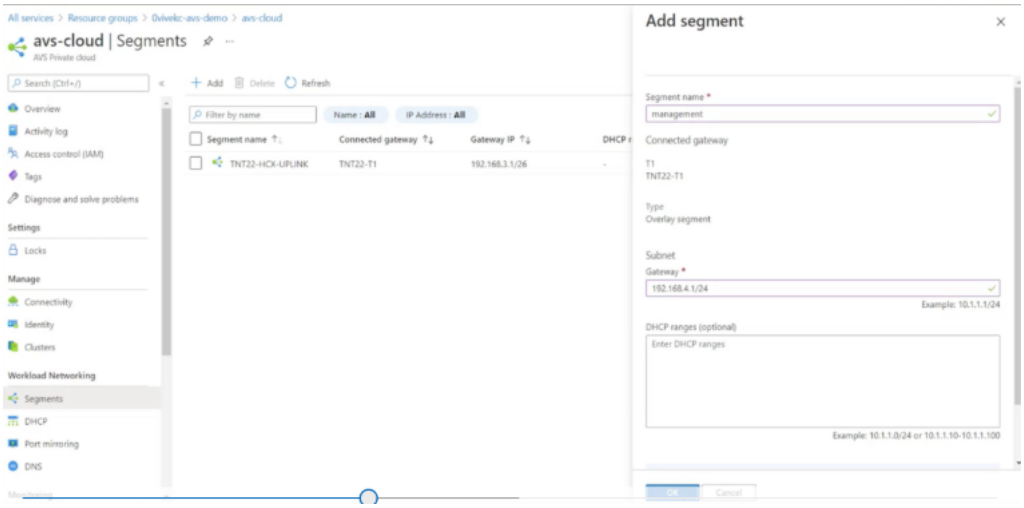
詳細については、「プライベートクラウド vCenter ポータルへのアクセス」を参照してください。

Azure ポータルで **NSX-T** セグメントを作成します

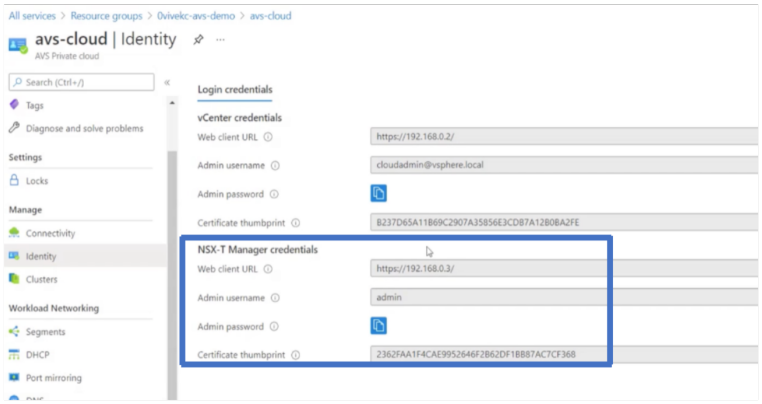
NSX-T セグメントは、Azure ポータルの Azure VMware ソリューションコンソールから作成および構成できます。これらのセグメントはデフォルトの Tier-1 ゲートウェイに接続され、これらのセグメントのワークロードは

East-West および North-South 接続を取得します。セグメントを作成すると、NSX-T Manager および vCenter に表示されます。

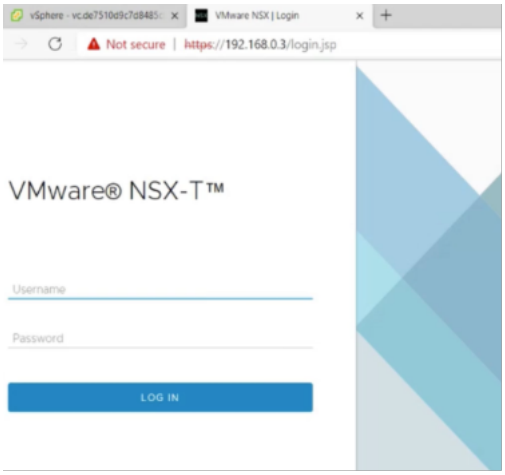
1. Azure VMware ソリューションのプライベートクラウドで、[ワークロードネットワーキング] で、[セグメント**] > [**追加] の順に選択します。新しい論理セグメントの詳細を入力し、「**OK**」を選択します。クライアント、管理、およびサーバーインターフェイスに対して 3 つの別々のセグメントを作成できます。



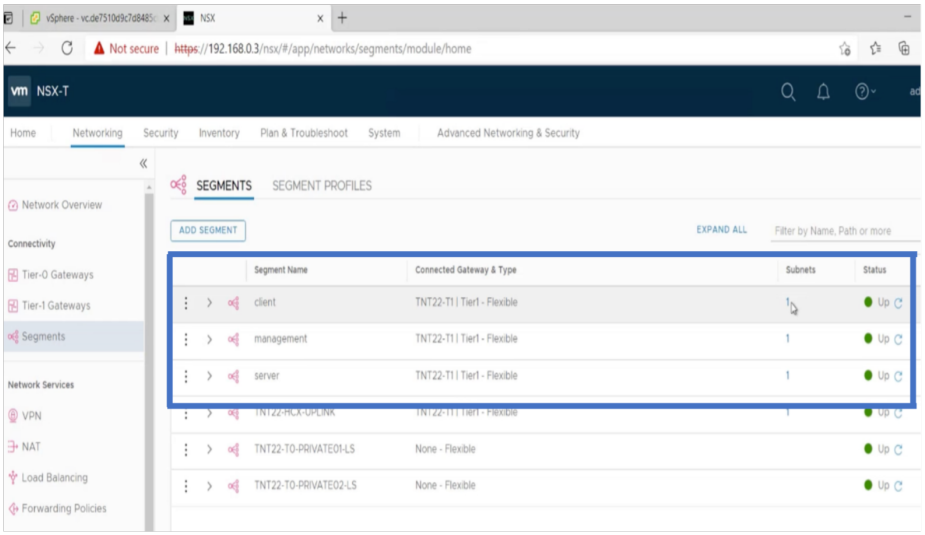
2. Azure VMware ソリューションのプライベートクラウドで、[管理] で [アイデンティティ] を選択します。NSX-T マネージャのクレデンシャルを書き留めます。



3. NSX-T Web クライアント URL を入力して VMware NSX-T マネージャを起動します。



4. NSX-T マネージャの [ネットワーク] **>** [セグメント] の下に、作成したすべてのセグメントが表示されます。サブネットを確認することもできます。



詳細については、「Azure ポータルで NSX-T セグメントを作成する」を参照してください。

VMware クラウドへの **NetScaler VPX** インスタンスのインストール

VMware ソフトウェア定義データセンター（SDDC）をインストールして構成したら、SDDC を使用して VMware クラウドに仮想アプライアンスをインストールできます。インストールできる仮想アプライアンスの数は、SDDC で使用可能なメモリの量によって異なります。

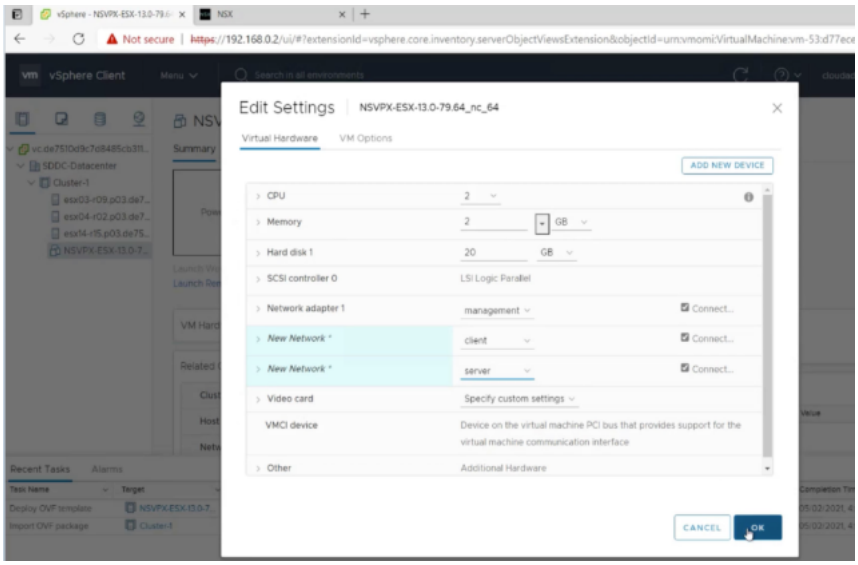
VMware クラウドに NetScaler VPX インスタンスをインストールするには、Windows ジャンプボックス仮想マシンで次の手順を実行します。

1. ESXi ホスト用の NetScaler VPX インスタンスセットアップファイルを、NetScaler ダウンロードサイトからダウンロードします。
2. Windows のジャンプボックスで VMware SDDC を開きます。

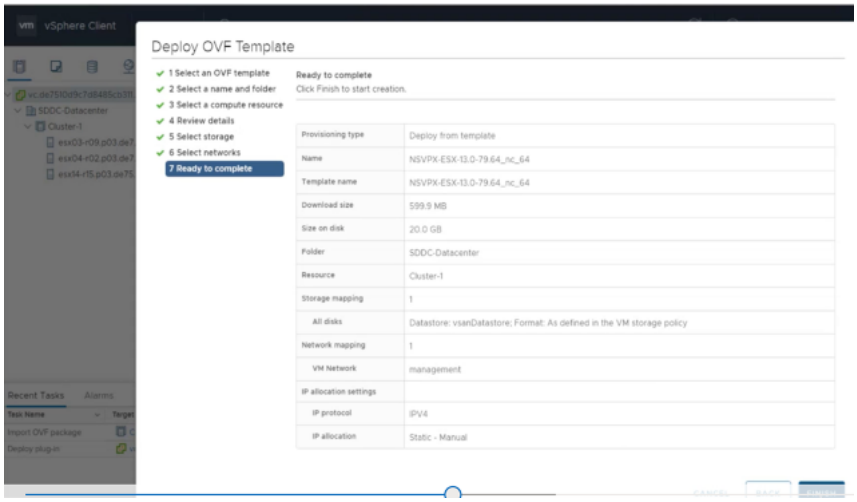
3. [ユーザー名] フィールドと [パスワード] フィールドに管理者の資格情報を入力し、[ログイン] をクリックします。
4. [**File**] メニューの [**Deploy OVF Template**] を選択します。
5. [**OVF** テンプレートのデプロイ] ダイアログボックスの [ファイルからの展開] フィールドで、NetScaler VPX インスタンスセットアップファイルを保存した場所を参照し、.ovf ファイルを選択し、[次へ] をクリックします。

   > 注
   >
   > デフォルトでは、NetScaler VPX インスタンスは E1000 ネットワークインターフェイスを使用します。VMXNET3 インターフェイスで ADC を展開するには、E1000 ではなく VMXNET3 インターフェイスを使用するように OVF を変更します。VMXNET3 インターフェイスの可用性は Azure インフラストラクチャによって制限され、Azure VMware ソリューションでは利用できない場合があります。

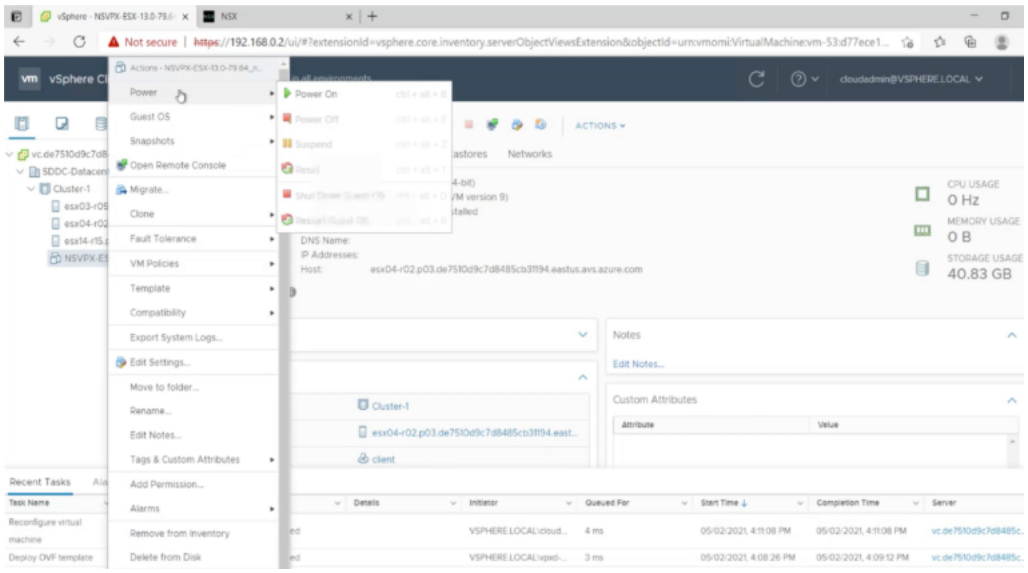
6. 仮想アプライアンス OVF テンプレートに表示されるネットワークを、VMware SDDC で設定したネットワークにマッピングします。[**OK**] をクリックします。



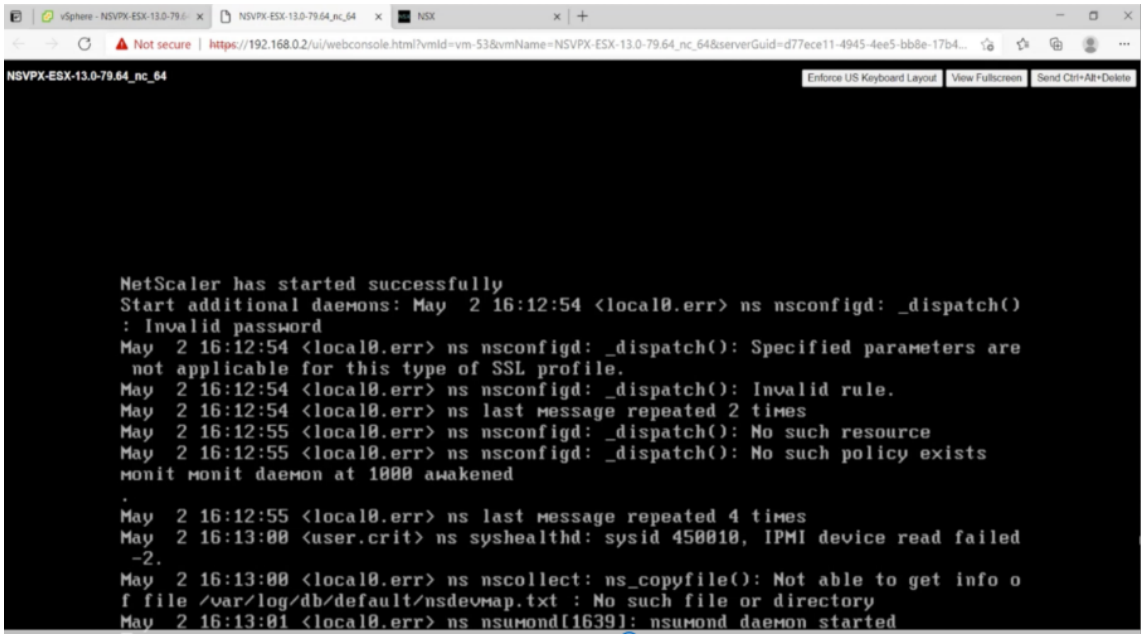
7. [完了] をクリックして VMware SDDC への仮想アプライアンスのインストールを開始します。



8. これで、NetScaler VPX インスタンスを起動する準備ができました。ナビゲーションペインで、インストールした NetScaler VPX インスタンスを選択し、右クリックメニューから［パワーオン］を選択します。コンソールポートをエミュレートするには、［**Console**］タブをクリックします。



9. これで、vSphere クライアントから NetScaler 仮想マシンに接続されています。



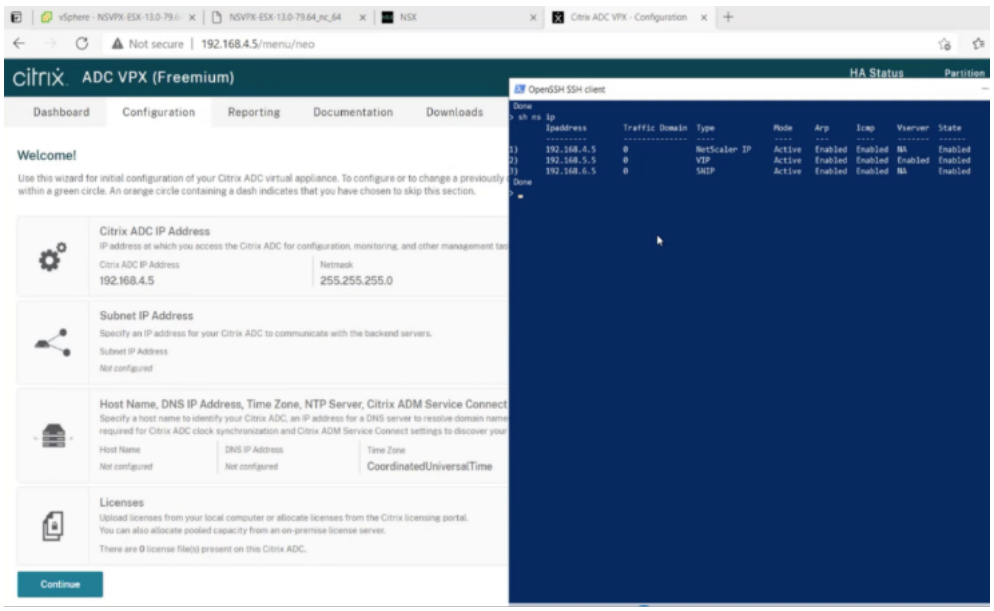
10. SSH キーを使用して NetScaler アプライアンスにアクセスするには、CLI で次のコマンドを入力します。

```
ssh nsroot@<management IP address>
<!--NeedCopy-->
```

例:

```
ssh nsroot@192.168.4.5
<!--NeedCopy-->
```

11. ADC の設定は、`show ns ip`コマンドを使用して確認できます。

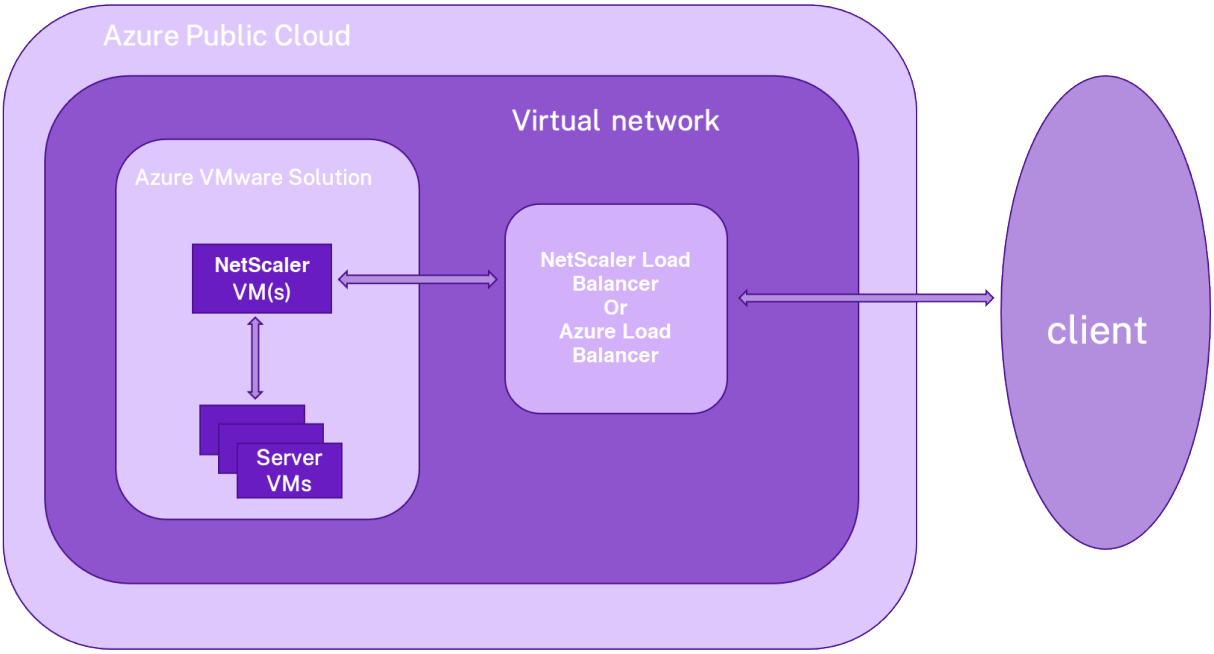


Azure VMware ソリューションでスタンドアロンの NetScaler ADC VPX インスタンスを構成する

August 15, 2023

インターネット向けアプリケーション用の Azure VMware ソリューション（AVS）上の NetScaler VPX スタンドアロンインスタンスを構成できます。

次の図は、Azure VMware ソリューション上の NetScaler VPX スタンドアロンインスタンスを示しています。クライアントは、AVS 内の NetScaler の仮想 IP（VIP）アドレスに接続することで AVS サービスにアクセスできます。これを実現するには、NetScaler ロードバランサーまたは Azure ロードバランサーインスタンスを AVS の外部で同じ Azure 仮想ネットワーク内にプロビジョニングします。AVS サービス内の NetScaler VPX インスタンスの VIP にアクセスするようにロードバランサーを構成します。



前提条件

仮想アプライアンスのインストールを開始する前に、次の Azure の前提条件をお読みください。

- Azure VMware ソリューションとその前提条件の詳細については、Azure VMware ソリューションのドキュメントを参照してください。
- Azure VMware ソリューションのデプロイの詳細については、「Azure VMware ソリューションのプライベートクラウドをデプロイする」を参照してください。
- Azure VMware ソリューションにアクセスして管理するための Windows ジャンプボックス仮想マシンの作成の詳細については、「Azure VMware ソリューションのプライベートクラウドへのアクセス」を参照してください。

- Windows ジャンプボックス仮想マシンで、NetScaler VPX アプライアンスセットアップファイルをダウンロードします。
- 仮想マシンが接続する VMware SDDC 上に、適切な NSX-T ネットワークセグメントを作成します。詳細については、「Azure VMware ソリューションでのネットワークセグメントの追加」を参照してください。
- NetScaler VPX インスタンスを VMware クラウドにインストールする方法の詳細については、「VMware クラウドに NetScaler VPX インスタンスをインストールする」を参照してください。

NetScaler ロードバランサーを使用して **AVS** 上の **NetScaler VPX** スタンドアロンインスタンスを構成します

次の手順に従って、NetScaler ロードバランサーを使用するインターネット向けアプリケーション用に AVS 上の NetScaler VPX スタンドアロンインスタンスを構成します。

1. NetScaler VPX インスタンスを Azure クラウドにデプロイします。詳しくは、「NetScaler VPX スタンドアロンインスタンスの構成」を参照してください。

   > 注:
   >
   > Azure VMware Cloud と同じ仮想ネットワークにデプロイされていることを確認します。

2. AVS にデプロイされた NetScaler VPX の VIP アドレスにアクセスするように NetScaler VPX インスタンスを構成します。

   a) 負荷分散仮想サーバーを追加します。

   ```
   add lb vserver <name> <serviceType> [<vip>] [<port>]
   <!--NeedCopy-->
   ```

   例:

   ```
   add lb vserver lb1 HTTPS 172.31.0.6 443
   <!--NeedCopy-->
   ```

   b) AVS にデプロイされた NetScaler VPX IP に接続するサービスを追加します。

   ```
   add service <name> <ip> <serviceType> <port>
   <!--NeedCopy-->
   ```

   例:

   ```
   add service webserver1 192.168.4.10 HTTP 80
   <!--NeedCopy-->
   ```

   c) サービスを負荷分散仮想サーバーにバインドします。

   ```
   bind lb vserver <name> <serviceName>
   <!--NeedCopy-->
   ```


例:

```
bind lb vserver lb1 webserver1
<!--NeedCopy-->
```

Azure ロードバランサーを使用して AVS 上の NetScaler VPX スタンドアロンインスタンスを構成する

以下の手順に従って、Azure ロードバランサーを使用するインターネット向けアプリケーション用に AVS 上の NetScaler VPX スタンドアロンインスタンスを構成します。

1. Azure クラウドで Azure ロードバランサーインスタンスを構成します。詳しくは、ロードバランサーの作成に関する Azure ドキュメントを参照してください。
2. AVS にデプロイされている NetScaler VPX インスタンスの VIP アドレスをバックエンドプールに追加します。

   次の Azure コマンドは、1 つのバックエンド IP アドレスを負荷分散バックエンドアドレスプールに追加します。

   ```
   az network lb address-pool address add
                                       --resource-group <Azure VMC
                                           Resource Group>
                                       --lb-name <LB Name>
                                       --pool-name <Backend pool name
                                           >
                                       --vnet <Azure VMC Vnet>
                                       --name <IP Address name>
                                       --ip-address <VIP of ADC in
                                           VMC>
   <!--NeedCopy-->
   ```

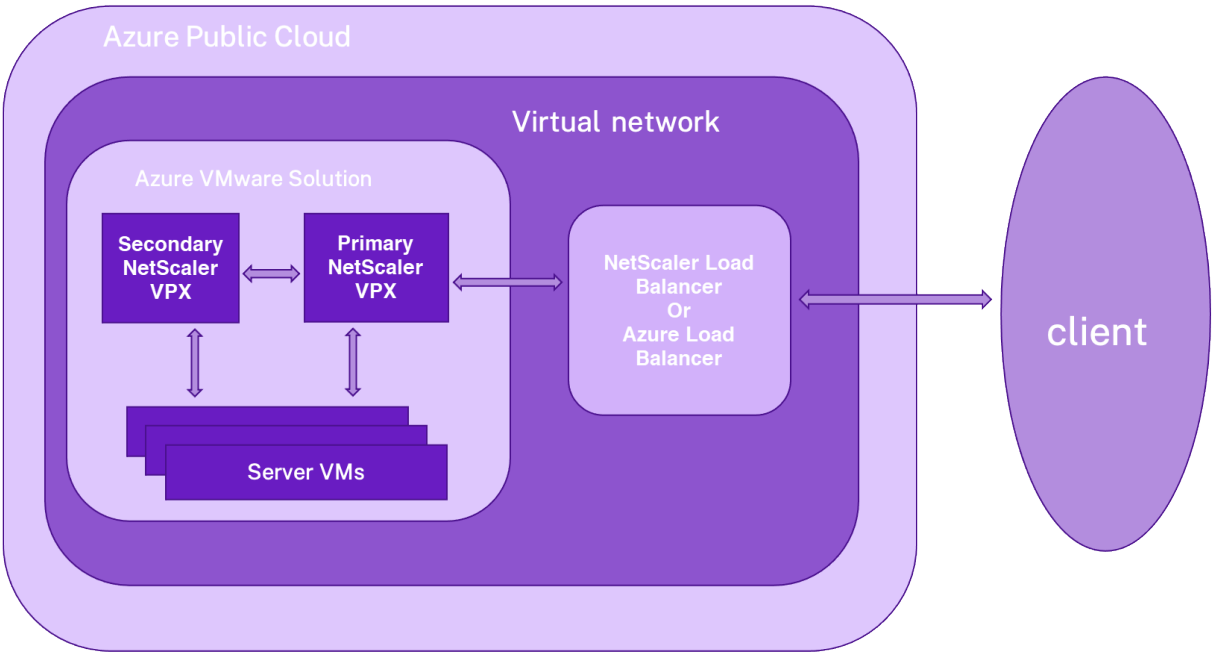
   > 注:
   >
   > Azure ロードバランサーが Azure VMware クラウドと同じ仮想ネットワークにデプロイされていることを確認します。

Azure VMware ソリューションで NetScaler ADC VPX の高可用性セットアップを構成する

August 15, 2023

インターネットに接続するアプリケーション用の Azure VMware ソリューション（AVS）で NetScaler VPX HA セットアップを構成できます。

次の図は、AVS 上の NetScaler VPX HA ペアを示しています。クライアントは、AVS 内のプライマリ ADC ノードの VIP に接続することで、AVS サービスにアクセスできます。これを実現するには、NetScaler ロードバランサーまたは Azure ロードバランサーインスタンスを AVS の外部で同じ Azure 仮想ネットワーク内にプロビジョニングします。AVS サービス内のプライマリ ADC ノードの VIP にアクセスするようにロードバランサを設定します。



前提条件

仮想アプライアンスのインストールを開始する前に、次の Azure の前提条件をお読みください。

- Azure VMware ソリューションとその前提条件の詳細については、Azure VMware ソリューションのドキュメントを参照してください。
- Azure VMware ソリューションのデプロイの詳細については、「Azure VMware ソリューションのプライベートクラウドをデプロイする」を参照してください。
- Azure VMware ソリューションにアクセスして管理するための Windows ジャンプボックス仮想マシンの作成の詳細については、「Azure VMware ソリューションのプライベートクラウドへのアクセス」を参照してください。
- Windows ジャンプボックス仮想マシンで、NetScaler VPX アプライアンスセットアップファイルをダウンロードします。
- 仮想マシンが接続する VMware SDDC 上に、適切な NSX-T ネットワークセグメントを作成します。詳細については、「Azure VMware Solution でのネットワークセグメントの追加」を参照してください。

構成の手順

以下の手順に従って、インターネット向けアプリケーションの AVS で NetScaler VPX 高可用性セットアップを構成します。

1. VMware クラウド上に 2 つの NetScaler VPX インスタンスを作成します。詳しくは、「VMware クラウドへの NetScaler VPX インスタンスのインストール」を参照してください。
2. NetScaler HA のセットアップを構成します。詳細については、「高可用性の構成」を参照してください。
3. インターネットに接続されたアプリケーションにアクセスできるように NetScaler HA セットアップを構成します。

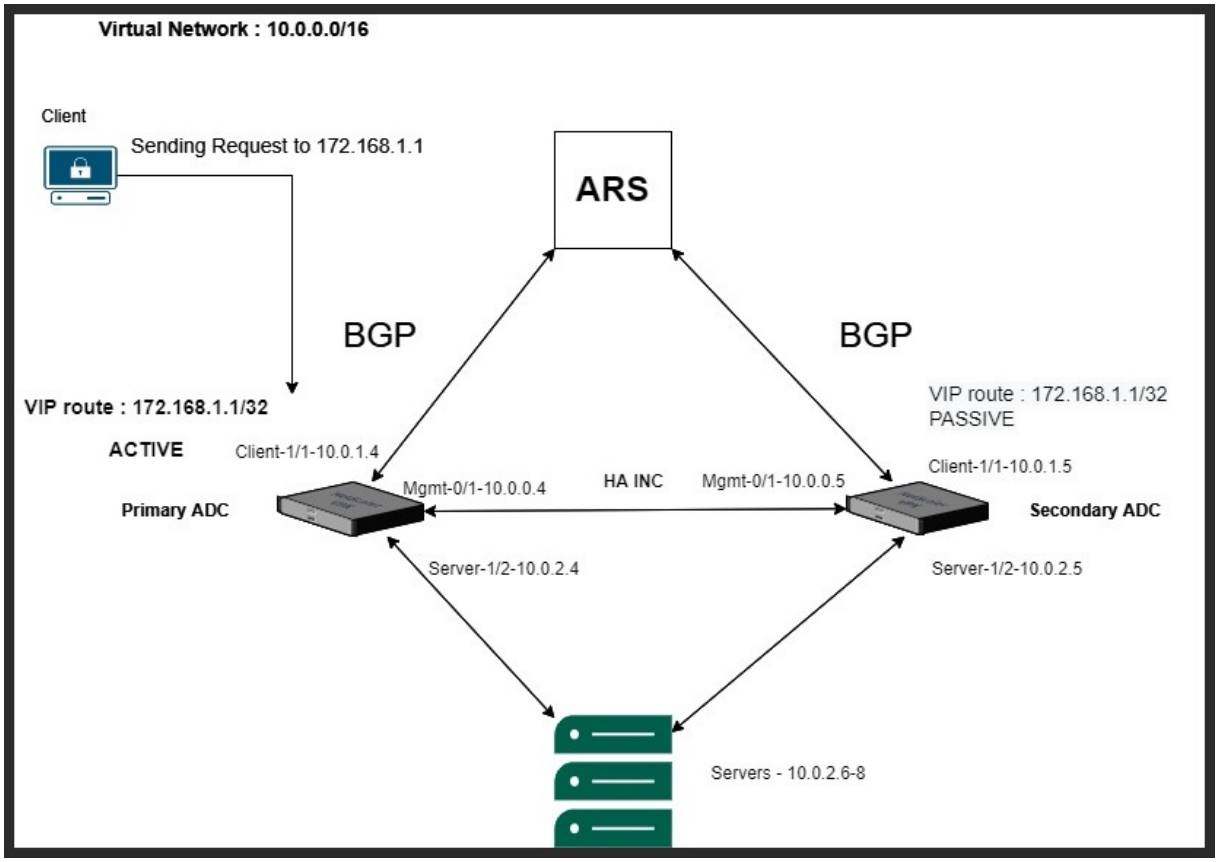
    - NetScaler ロードバランサーを使用して NetScaler VPX インスタンスを構成するには、「NetScaler ロードバランサーを使用して AVS 上の NetScaler VPX スタンドアロンインスタンスを構成する」を参照してください。
    - Azure ロードバランサーを使用して NetScaler VPX インスタンスを構成するには、「Azure ロードバランサーを使用して AVS 上の NetScaler VPX スタンドアロンインスタンスを構成する」を参照してください。

NetScaler VPX HA ペアで Azure ルートサーバーを構成する

August 15, 2023

NetScaler VPX インスタンスを使用して Azure ルートサーバーを構成し、BGP プロトコルを使用して仮想ネットワークで構成された VIP ルートを交換できます。NetScaler ADC は、スタンドアロンまたは HA-INC モードで展開し、BGP で構成できます。この展開では、ADC HA ペアの前に Azure ロードバランサー (ALB) は必要ありません。

次の図は、VPX HA トポロジが Azure ルートサーバーとどのように統合されるかを示しています。各 ADC インスタンスには、管理用、クライアントトラフィック用、サーバートラフィック用の 3 つのインターフェイスがあります。



トポロジ図では、次の IP アドレスを使用します。

プライマリ **ADC** インスタンスの **IP** 設定の例**:**

```
NSIP: 10.0.0.4/24
SNIP on 1/1: 10.0.1.4/24
SNIP on 1/2: 10.0.2.4/24
VIP: 172.168.1.1/32
<!--NeedCopy-->
```

セカンダリ **ADC** インスタンスの **IP** 設定の例**:**

```
NSIP: 10.0.0.5/24
SNIP on 1/1: 10.0.1.5/24
SNIP on 1/2: 10.0.2.5/24
VIP: 172.168.1.1/32
<!--NeedCopy-->
```

前提条件

NetScaler VPX インスタンスを Azure に展開する前に、次の情報を理解している必要があります。

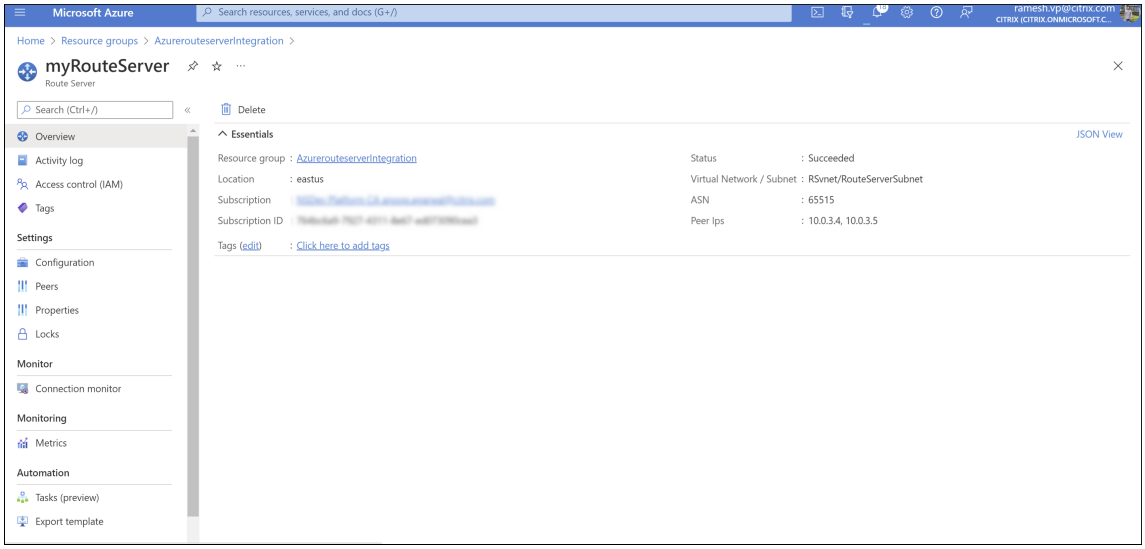
- Azure の用語とネットワークの詳細。詳細については、「Azure の用語」を参照してください。
- Azure ルートサーバーの概要。詳細については、「Azure Route Server とは」を参照してください。。

- NetScaler ADC アプライアンスの動作。詳しくは、NetScaler のドキュメントを参照してください。
- NetScaler ネットワーキング。詳細については、ADC ネットワークを参照してください。

NetScaler VPX HA ペアで Azure ルートサーバーを構成する方法

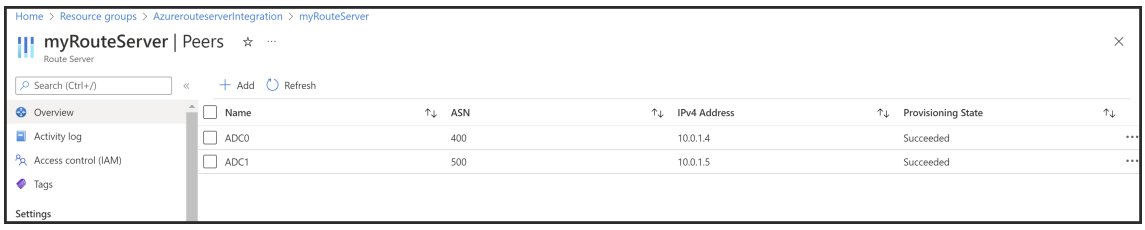
1. Azure Portal にルートサーバーを作成します。詳細については、「Azure portal を使用したルートサーバーの作成と構成」を参照してください。

   次の例では、サブネット 10.0.3.0/24 が Azure サーバーのデプロイに使用されます。ルートサーバーが作成されたら、ルートサーバーの IP アドレスを取得します (例:10.0.3.4、10.0.3.5)。



2. Azure Portal でネットワーク仮想アプライアンス (NVA) とのピアリングを設定します。NetScaler VPX インスタンスを NVA として追加します。詳細については、「NVA とのピアリングの設定」を参照してください。

   次の例では、1/1 インターフェイスの ADC SNIP（10.0.1.4 と 10.0.1.5）と ASN: 400 と 500 がピアの追加時に使用されます。



3. 高可用性構成用に 2 つの NetScaler ADC VPX インスタンスを追加します。

   次の手順を実行します:

   a) Azure に 2 つの VPX インスタンス (プライマリインスタンスとセカンダリインスタンス) をデプロイします。
   b) 両方のインスタンスにクライアントとサーバーの NIC を追加します。
   c) NetScaler GUI を使用して、両方のインスタンスで高可用性設定を構成します。

4. プライマリ ADC インスタンスで動的ルーティングを設定します。

   設定例:

```
enable ns mode L3 MBF USNIP SRADV DRADV  PMTUD
enable ns feature LB BGP
add ns ip 10.0.1.4 255.255.255.0 -vServer DISABLED -dynamicRouting
    ENABLED
VTYSH
configure terminal
router BGP 400
timers bgp 1 3
neighbor 10.0.3.4 remote-as 65515
neighbor 10.0.3.4 advertisement-interval 3
neighbor 10.0.3.4 fall-over  bfd
neighbor 10.0.3.5 remote-as 65515
neighbor 10.0.3.5 advertisement-interval 3
neighbor 10.0.3.5 fall-over  bfd
address-family ipv4
redistribute kernel
redistribute static
<!--NeedCopy-->
```

5. セカンダリ ADC インスタンスで動的ルーティングを設定します。

   設定例:

```
enable ns mode L3 MBF USNIP SRADV DRADV  PMTUD
enable ns feature LB BGP
add ns ip 10.0.1.5 255.255.255.0 -vServer DISABLED -dynamicRouting
    ENABLED
VTYSH
configure terminal
router BGP 500
timers bgp 1 3
neighbor 10.0.3.4 remote-as 65515
neighbor 10.0.3.4 advertisement-interval 3
neighbor 10.0.3.4 fall-over  bfd
neighbor 10.0.3.5 remote-as 65515
neighbor 10.0.3.5 advertisement-interval 3
neighbor 10.0.3.5 fall-over  bfd
address-family ipv4
redistribute kernel
redistribute static
<!--NeedCopy-->
```

6. VTY シェルインターフェイスで BGP コマンドを使用して確立された BGP ピアを確認します。詳細については、「BGP 設定の確認」を参照してください。

```
show ip bgp neighbors
<!--NeedCopy-->
```

7. プライマリ ADC インスタンスで LB 仮想サーバーを設定します。

設定例:

```
add ns ip 172.16.1.1 255.255.255.255 -type VIP -hostRoute ENABLED
add lbvserver v1 HTTP 172.16.1.1 80
add service s1 10.0.2.6 HTTP 80
bind lbvserver v1 s1
enable ns feature lb
<!--NeedCopy-->
```

NetScaler VPX インスタンスと同じ仮想ネットワーク内のクライアントが、LB 仮想サーバーにアクセスできるようになりました。この場合、NetScaler VPX インスタンスは VIP ルートを Azure ルートサーバーにアドバタイズします。

Azure の Autoscale 設定を追加する

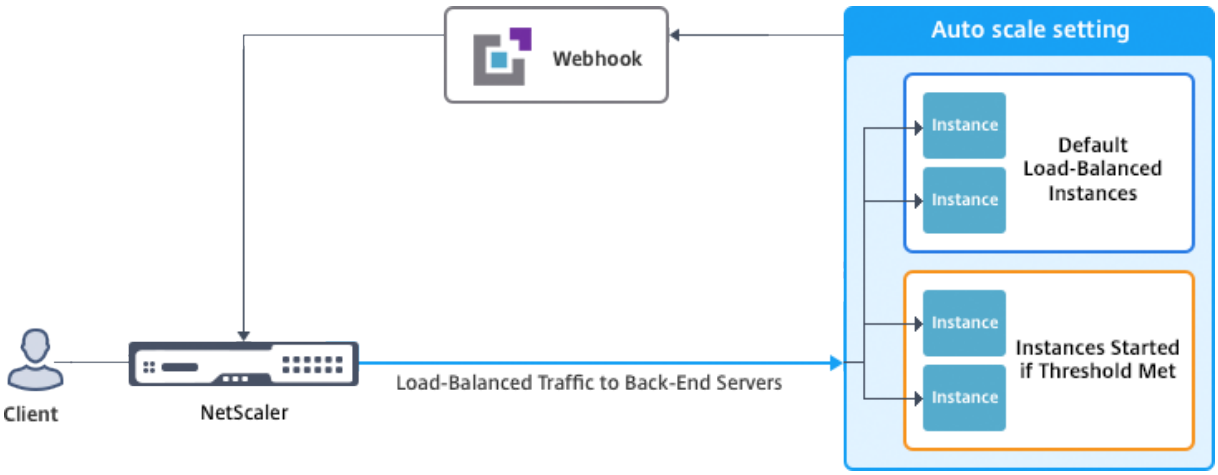
December 8, 2023

クラウドでアプリケーションを効率的にホストすると、アプリケーションの必要に応じて、リソースを簡単にコスト効率よく管理できます。需要の増大に対応するには、ネットワークリソースをスケールアップする必要があります。需要が収まるかどうかにかかわらず、アイドル状態のリソースの不必要なコストを避けるためにスケールダウンする必要があります。アプリケーションの実行コストを最小限に抑えるには、トラフィック、メモリ、CPU の使用などを常に監視する必要があります。しかし、トラフィックを手動で監視するのは煩雑です。アプリケーション環境を動的にスケールアップまたはスケールダウンするには、トラフィックの監視プロセスを自動化し、必要に応じてリソースを増減する必要があります。

Azure での VPX マルチ IP スタンドアロンおよび高可用性のデプロイには、Azure 仮想マシンスケールセット (VMSS) で Autoscale を使用できます。

Azure VMSS および AAutoscale 機能と統合された NetScaler VPX インスタンスには、次の利点があります。

- 負荷分散と管理: 需要に応じてサーバーをスケールアップおよびスケールダウンするように自動構成します。NetScaler VPX インスタンスは、VPX インスタンスが展開されているのと同じ仮想ネットワーク、または同じ Azure サブスクリプション内のピアリングされた仮想ネットワーク内の VMS AAutoscale e 設定を自動検出します。VMSS Autoscale 設定を選択して、負荷を分散できます。これは、VPX インスタンスで NetScaler 仮想 IP アドレスとサブネット IP アドレスを自動構成することによって行われます。
- 高可用性:Autoscale グループを検出し、サーバーの負荷を分散します。
- ネットワークの可用性の向上：VPX インスタンスは、異なる仮想ネットワーク（VNet）上のバックエンドサーバーをサポートします。



詳細については、次の Azure トピックを参照してください。

- 仮想マシンのスケールセットのドキュメント
- Microsoft Azure 仮想マシン、クラウドサービス、および Web アプリケーションの Autoscale の概要

はじめに

1. Azure 関連の使用に関するガイドラインを参照してください。詳細については、「Microsoft Azure での NetScaler VPX インスタンスの展開」を参照してください。
2. 要件（スタンドアロンまたは高可用性デプロイ）に応じて、Azure 上に 3 つのネットワークインターフェイスを使用して 1 つまたは複数の NetScaler VPX インスタンスを作成します。
3. VPX インスタンスの 0/1 インターフェイスのネットワークセキュリティグループで TCP 9001 ポートを開きます。VPX インスタンスは、このポートを使用してスケールアウトおよびスケールイン通知を受け取ります。
4. NetScaler VPX インスタンスが展開されている同じ仮想ネットワークに Azure VMSS を作成します。VMSS と NetScaler VPX インスタンスが異なる Azure 仮想ネットワークに展開されている場合、次の条件を満たす必要があります。

   - 両方の仮想ネットワークが同じ Azure サブスクリプションに含まれている必要があります。
   - 2 つの仮想ネットワークは、Azure の仮想ネットワークピアリング機能を使用して接続する必要があります。

   既存の VMSS 設定がない場合は、次のタスクを完了します。

   a）VMSS の作成

   b) VMSS でオートスケールを有効にする

   c) VMSS Autoscale 設定でスケールインおよびスケールアウトポリシーを作成する

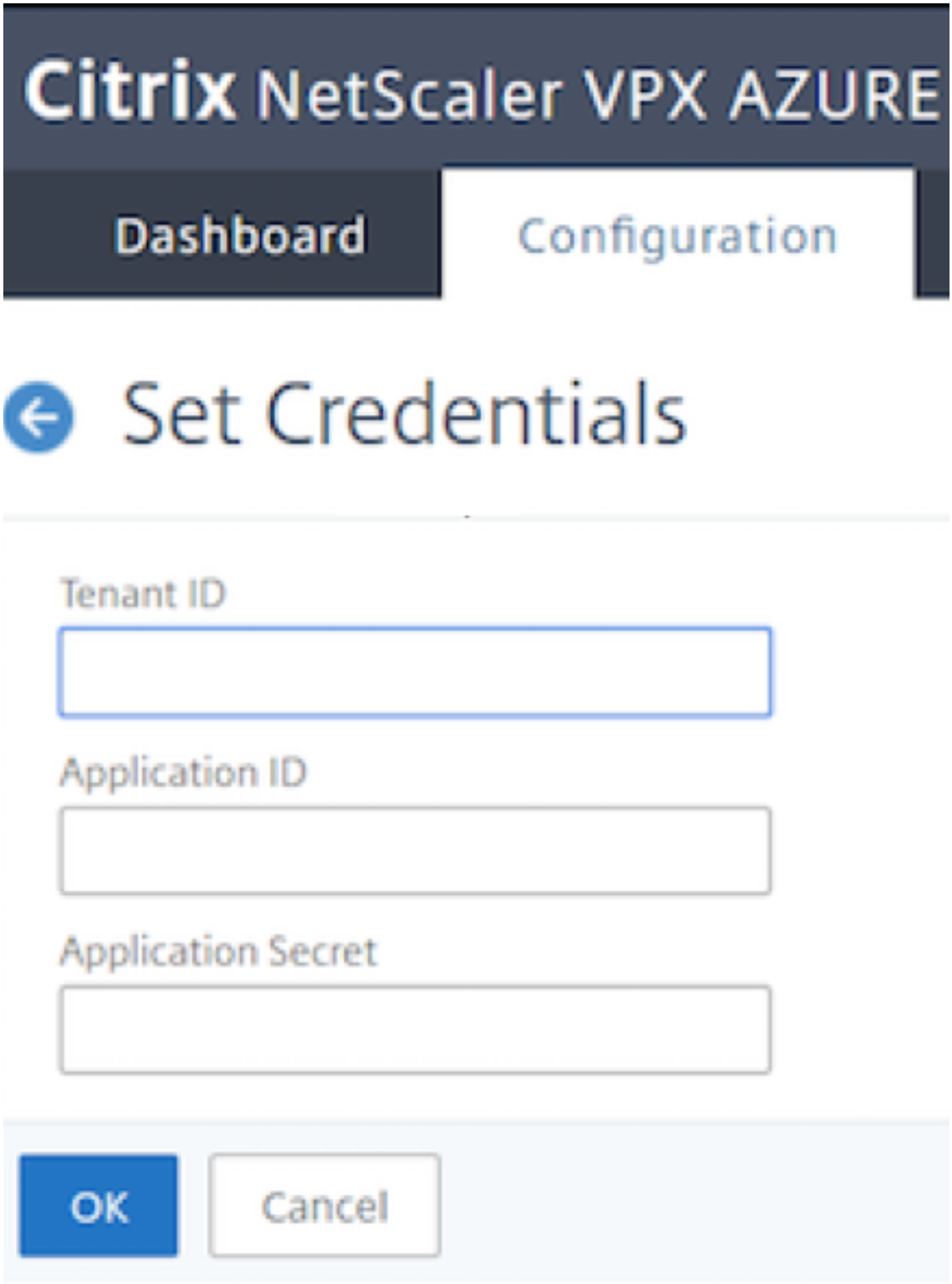
   詳細については、「Azure 仮想マシンのスケールセットを使用した Autoscale の概要」を参照してください。

5. リソースにアクセスできる AzureActive Directory（ADD）アプリケーションとサービスプリンシパルを作成します。新しく作成された AAD アプリケーションにコントリビュータロールを割り当てます。詳細については、「ポータルを使用してリソースにアクセスできる Azure Active Directory アプリケーションおよびサービスプリンシパルを作成する」を参照してください。

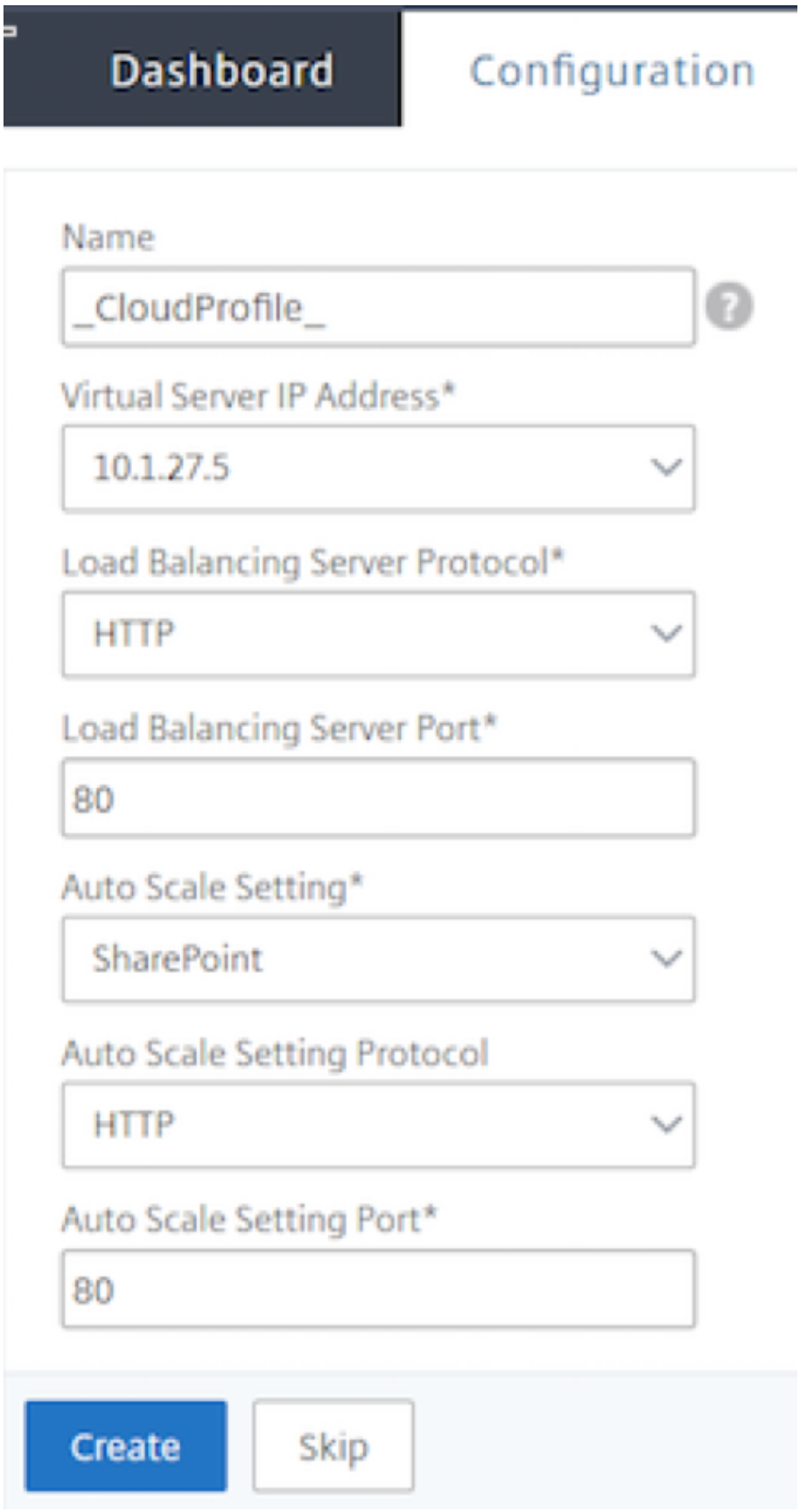
VMSS を NetScaler VPX インスタンスに追加する

GUI を使用して、ワンクリックで VPX インスタンスに Autoscale 設定を追加できます。VPX インスタンスに Autoscale 設定を追加するには、次の手順を実行します。

1. VPX インスタンスにログオンします。
2. NetScaler VPX インスタンスに初めてログオンすると、[認証情報の設定] ページが表示されます。Autoscale 機能を機能させるために必要な Azure 認証情報を追加します。



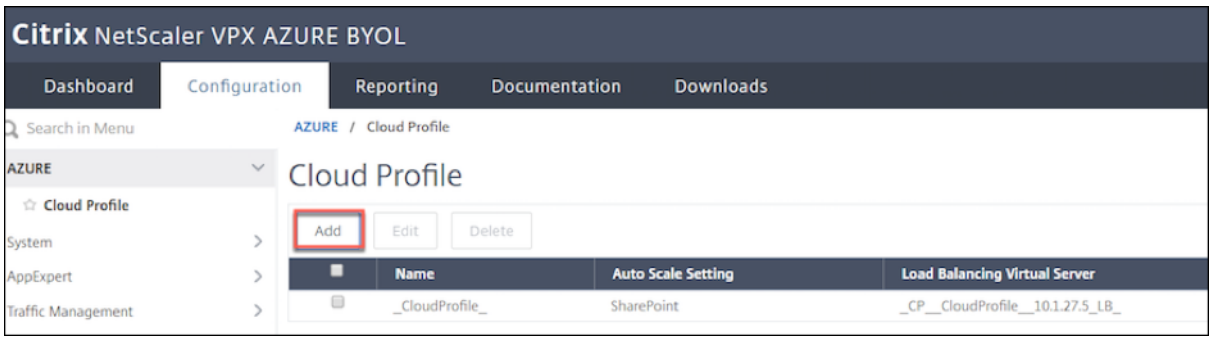
3. デフォルトのクラウドプロファイルページで、次の例に示すように詳細を入力し、[Create] をクリックします。



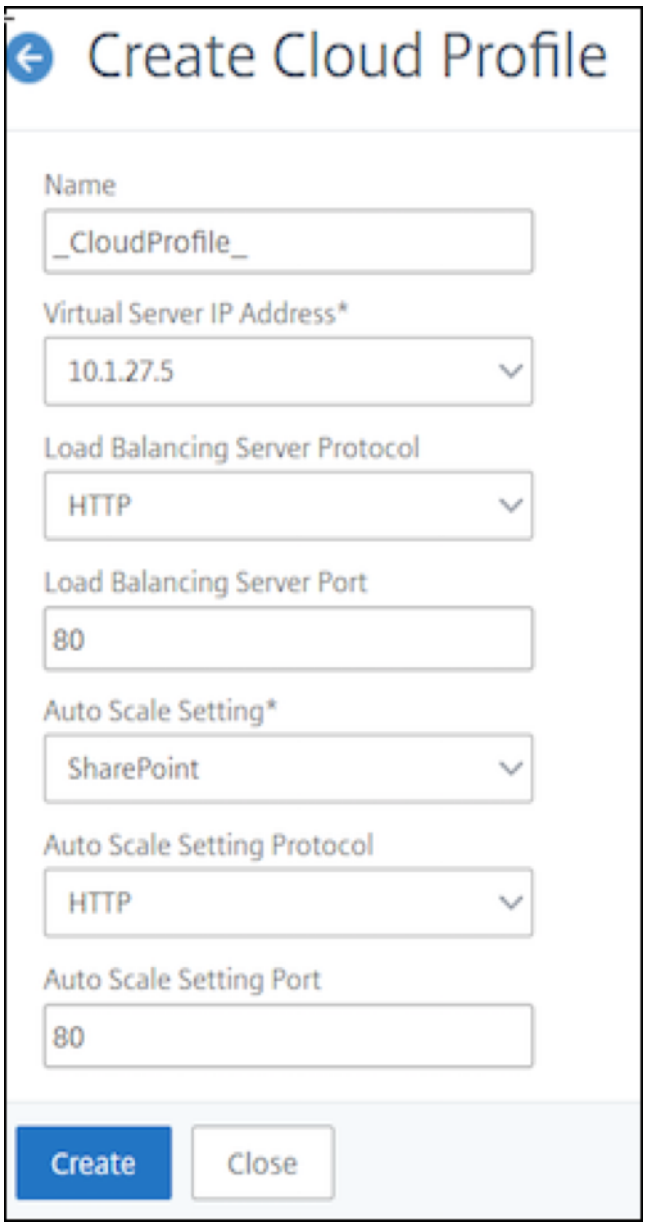
クラウドプロファイルの作成時に留意すべきポイント

- 仮想サーバーの IP アドレスは、VPX インスタンスで使用可能な空き IP アドレスから自動的に設定されます。詳細については、「Azure ポータルを使用して仮想マシンに複数の IP アドレスを割り当てる」を参照してください。
- Autoscale 設定は、同じ仮想ネットワークまたはピアリングされた仮想ネットワーク内の NetScaler VPX インスタンスに接続されている VMSS インスタンスから事前に入力されます。詳細については、「Azure 仮想マシンのスケールセットを使用した Autoscale の概要」を参照してください。
- Auto Scaling Group のプロトコルとポートを選択するときは、サーバーがプロトコルとポートをリッスンしていることを確認し、サービスグループで正しいモニターをバインドします。デフォルトでは、TCP モニターが使用されます。
- SSL プロトコルタイプが Autos Scaling の場合、クラウドプロファイルを作成した後、証明書がないために負荷分散仮想サーバーまたはサービスグループがダウンします。証明書は、仮想サーバまたはサービスグループに手動でバインドできます。

初回ログオン後、クラウドプロファイルを作成する場合は、GUI で [システム] **> [Azure] > [**クラウドプロファイル] に移動し、[追加] をクリックします。



クラウドプロファイルの作成設定ページが表示されます。



クラウドプロファイルは、NetScaler 負荷分散仮想サーバーと、Auto Scaling グループのサーバーとしてメンバー（サーバー）を持つサービスグループを作成します。バックエンドサーバーは、VPX インスタンスで構成された SNIP を介して到達可能である必要があります。

> 注:
>
> NetScaler リリース 13.1-42.x 以降では、Azure の同じ VMSS を使用して、（異なるポートを使用して）サービスごとに異なるクラウドプロファイルを作成できます。したがって、NetScaler VPX インスタンスは、パブリッククラウド内の同じ自動スケーリンググループを持つ複数のサービスをサポートします。

Azure Portal で自動スケール関連の情報を表示するには、[すべてのサービス] **>** [仮想マシンスケールセット] **>** [仮想マシンスケールセットの選択] **>** [スケーリング] の順に移動します。

NetScaler VPX 展開用の Azure タグ

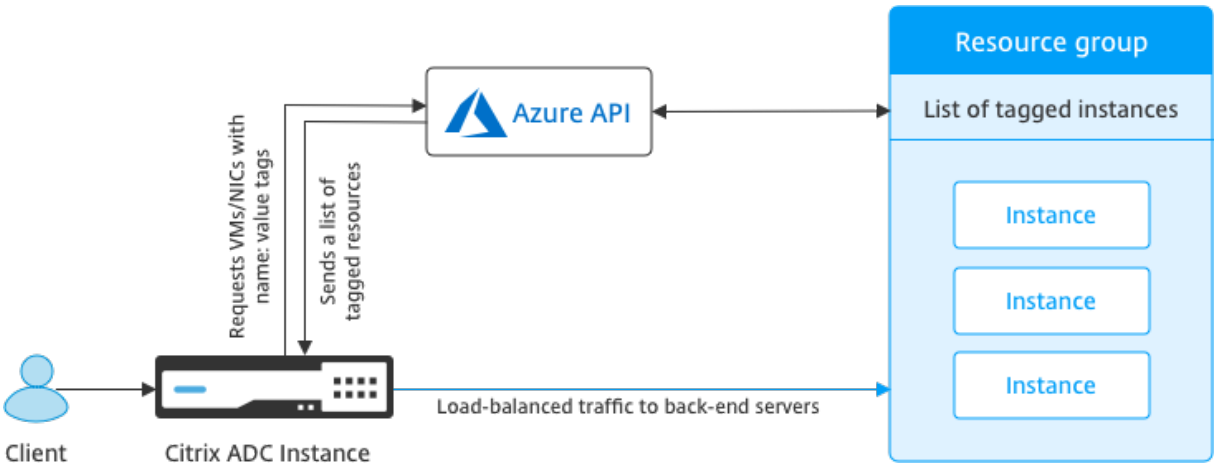
December 8, 2023

Azure クラウドポータルでは、名前: 値のペア (Dept: Finance など) でリソースにタグを付けて、リソースグループ間、およびポータル内でサブスクリプション間でリソースを分類して表示できます。タグ付けは、課金、管理、または自動化のためにリソースを整理する必要がある場合に役立ちます。

VPX デプロイにおける Azure タグの仕組み

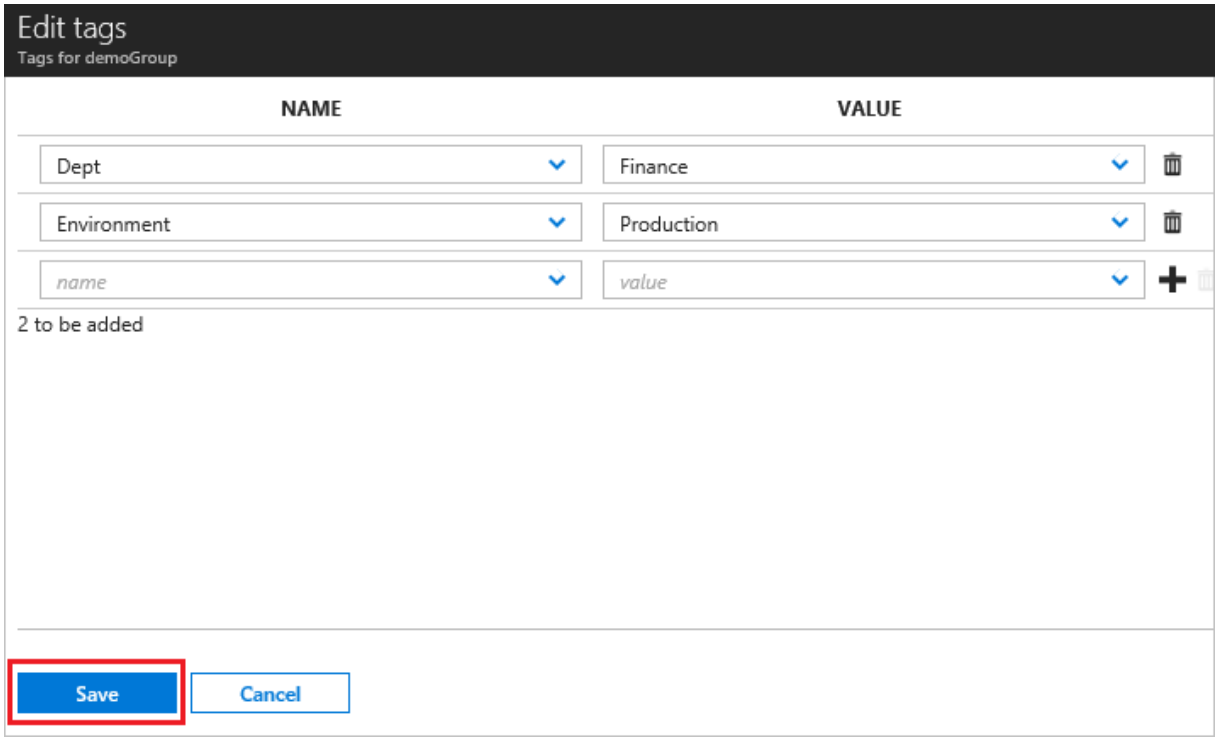
Azure Cloud にデプロイされた NetScaler VPX スタンドアロンおよび高可用性インスタンスの場合、Azure タグに関連付けられた負荷分散サービスグループを作成できるようになりました。VPX インスタンスは、Azure 仮想マシン（バックエンドサーバー）とネットワークインターフェイス（NIC）、またはその両方をそれぞれのタグで常に監視し、それに応じてサービスグループを更新します。

VPX インスタンスは、タグを使用してバックエンドサーバーの負荷分散を行うサービスグループを作成します。インスタンスは、特定のタグ名とタグ値でタグ付けされたすべてのリソースについて Azure API にクエリします。割り当てられたポーリング期間（デフォルトでは 60 秒）に応じて、VPX インスタンスは定期的に Azure API をポーリングし、VPX GUI で割り当てられたタグ名とタグ値を使用して利用可能なリソースを取得します。適切なタグが付いた VM または NIC が追加または削除されると、ADC はそれぞれの変更を検出し、VM または NIC の IP アドレスをサービスグループに自動的に追加または削除します。



はじめに

NetScaler 負荷分散サービスグループを作成する前に、Azure のサーバーにタグを追加します。タグは、仮想マシンまたは NIC に割り当てることができます。



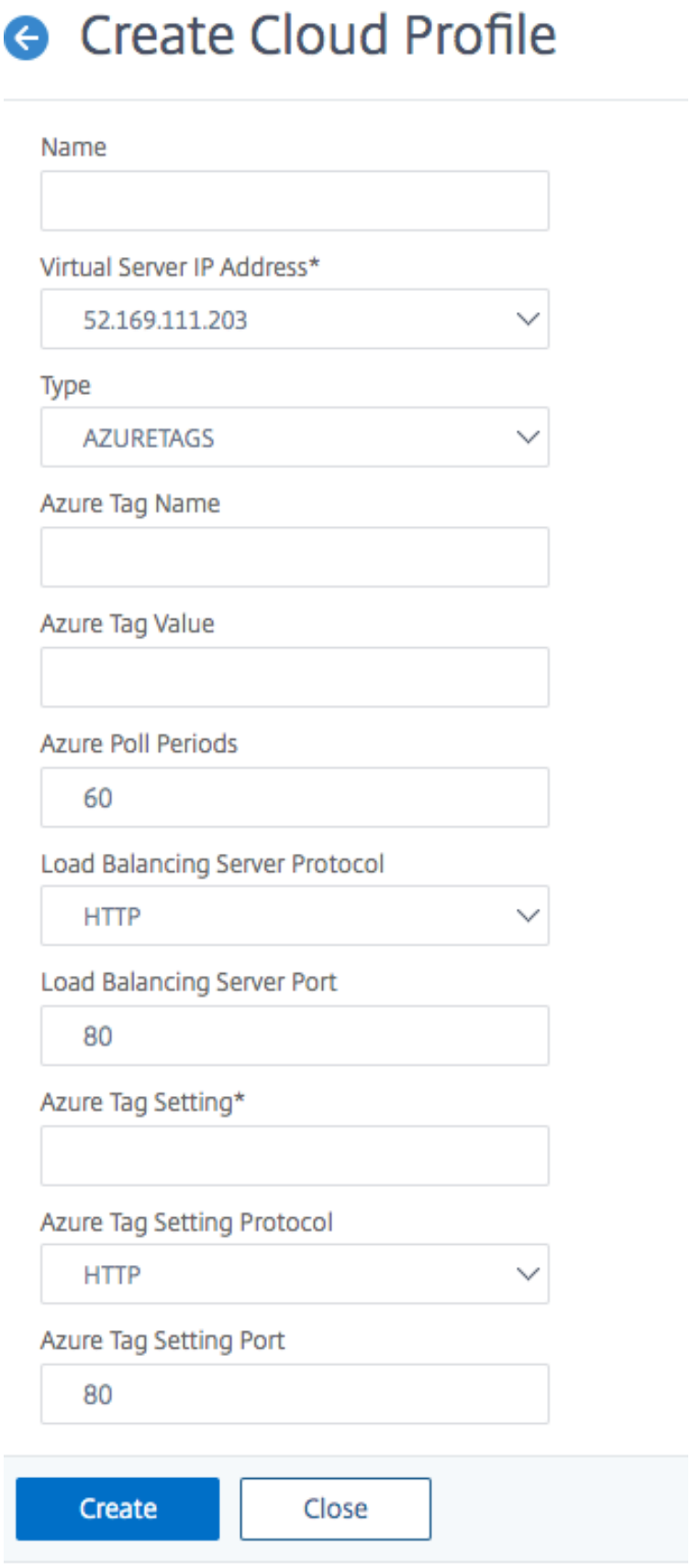
Azure タグの追加の詳細については、Microsoft ドキュメント「タグを使用して Azure リソースを整理する」を参照してください。

> 注:
> Azure タグ設定を追加する ADC CLI コマンドは、数字またはアルファベットのみで始まるタグ名とタグ値をサポートし、他のキーボード文字は使用できません。

VPX GUI を使用して **Azure** タグ設定を追加する方法

VPX GUI を使用して Azure タグクラウドプロファイルを VPX インスタンスに追加すると、インスタンスは指定されたタグを使用してバックエンドサーバーの負荷を分散できます。次の手順を実行します:

1. VPX GUI から、[構成**] > [Azure] > [**クラウドプロファイル**]** に移動します。
2. [追加] をクリックしてクラウドプロファイルを作成します。クラウドプロファイルウィンドウが開きます。



1. 次のフィールドに値を入力します。

   - 名前: プロフィールの名前を追加します
   - 仮想サーバーの IP アドレス: 仮想サーバーの IP アドレスは、VPX インスタンスで使用可能な空き IP アドレスから自動的に設定されます。詳細については、「Azure ポータルを使用して仮想マシンに複数の IP アドレスを割り当てる」を参照してください。
   - タイプ: メニューから「AZURETAGS」を選択します。
   - Azure タグ名:Azure ポータルで仮想マシンまたは NIC に割り当てた名前を入力します。
   - Azure タグ値:Azure ポータルの VM または NIC に割り当てた値を入力します。
   - Azure ポーリング期間: デフォルトでは、ポーリング間隔は最小値の 60 秒です。必要に応じて変更できます。
   - 負荷分散サーバープロトコル: ロードバランサーがリッスンするプロトコルを選択します。
   - 負荷分散サーバーポート: ロードバランサーが受信するポートを選択します。
   - Azure タグ設定: このクラウドプロファイル用に作成されるサービスグループの名前。
   - Azure タグ設定プロトコル: バックエンドサーバーがリッスンするプロトコルを選択します。
   - Azure タグ設定ポート: バックエンドサーバーがリッスンするポートを選択します。

2. [作成] をクリックします。

タグ付けされた仮想マシンまたは NIC に対して、ロードバランサー仮想サーバーとサービスグループが作成されます。ロードバランサー仮想サーバーを確認するには、VPX GUI から [** トラフィック管理] > [負荷分散] **[仮想サーバー] に移動します。

VPX CLI を使用して **Azure** タグ設定を追加する方法

NetScaler CLI で次のコマンドを入力して、Azure タグのクラウドプロファイルを作成します。

```
add cloud profile `<profile name>` -type azuretags -vServerName `<
    vserver name>` -serviceType HTTP -IPAddress `<vserver IP address>` -
    port 80 -serviceGroupName `<service group name>` -
    boundServiceGroupSvcType HTTP -vsvrbindsvcport 80 -azureTagName `<
    Azure tag specified on Azure portal>` -azureTagValue `<Azure value
    specified on the Azure portal>` -azurePollPeriod 60

<!--NeedCopy-->
```

> 重要
> : すべての設定を保存する必要があります。保存しないと、インスタンスの再起動後に設定が失われます。「`save config`」と入力します。

例 **1**:「mytagName/MyTagValue」ペアでタグ付けされたすべての Azure VM/NIC の HTTP トラフィックのクラウドプロファイルのサンプルコマンドを次に示します。


```
add cloud profile MyTagCloudProfile -type azuretags -vServerName
    MyTagVServer -serviceType HTTP -IPAddress 40.115.116.57 -port 80 -
    serviceGroupName MyTagsServiceGroup -boundServiceGroupSvcType HTTP -
    vsvrbindsvcport 80 -azureTagName myTagName -azureTagValue myTagValue
     -azurePollPeriod 60
Done
<!--NeedCopy-->
```

クラウドプロファイルを表示するには、次のように入力します `show cloudprofile`。

例 **2**: 次の CLI コマンドは、例 1 で新しく追加されたクラウドプロファイルに関する情報を出力します。

```
show cloudprofile
1)    Name:  MyTagCloudProfile Type:  azuretags         VServerName:
    MyTagVServer ServiceType:  HTTP       IPAddress:  52.178.209.133
    Port:  80               ServiceGroupName:  MyTagsServiceGroup
    BoundServiceGroupSvcType:  HTTP
       Vsvrbindsvcport:  80     AzureTagName:  myTagName AzureTagValue:
           myTagValue AzurePollPeriod:  60     GraceFul:  NO
          Delay:  60
<!--NeedCopy-->
```

クラウドプロファイルを削除するには、「rm cloud profile <`cloud profile name`>」と入力します。

例 **3**: 次のコマンドは、例 1 で作成したクラウドプロファイルを削除します。

```
> rm cloudprofile MyTagCloudProfile
  Done
<!--NeedCopy-->
```

トラブルシューティング

問題: ごくまれに、「rm cloud profile」CLI コマンドで、削除されたクラウドプロファイルに関連付けられているサービスグループおよびサーバーの削除に失敗することがあります。これは、削除されるクラウドプロファイルのポーリング期間が経過する秒前にコマンドが発行された場合に発生します。

解決方法: 残りのサービスグループごとに次の CLI コマンドを入力して、残りのサービスグループを手動で削除します。

```
#> rm servicegroup <serviceGroupName>

<!--NeedCopy-->
```

残りの各サーバに対して次の CLI コマンドを入力して、残りのサーバもそれぞれ削除します。

```
#> rm server <name>
<!--NeedCopy-->
```

問題:CLI を使用して VPX インスタンスに Azure タグ設定を追加すると、ウォームリブート後も HA ペアノードで rain_tags プロセスが実行され続けます。

解決方法: ウォームリブート後に、セカンダリノードでプロセスを手動で終了します。セカンダリ HA ノードの CLI からシェルプロンプトに出ます。

```
#> shell

<!--NeedCopy-->
```

rain_tags プロセスを強制終了するには、次のコマンドを使用します。

```
# PID=`ps -aux | grep rain_tags | awk '{
 print $2 }
 '`; kill -9 $PID

<!--NeedCopy-->
```

問題: バックエンドサーバーは正常であるにもかかわらず、VPX インスタンスから到達できず、DOWN として報告されることがあります。
解決方法: VPX インスタンスが、バックエンドサーバーに対応するタグ付き IP アドレスに到達できることを確認します。タグ付きの NIC の場合、これは NIC の IP アドレスです。タグ付きの VM の場合、これは仮想マシンのプライマリ IP アドレスです。VM/NIC が別の Azure VNet 上に存在する場合は、VNet ピアリングが有効になっていることを確認します。

NetScaler VPX インスタンスで GSLB を構成する

August 15, 2023

グローバルサーバー負荷分散（GSLB）用に構成された NetScaler ADC アプライアンスは、WAN の障害点から保護することにより、ディザスタリカバリとアプリケーションの継続的な可用性を提供します。GSLB は、クライアント要求を最も近い、または最もパフォーマンスの高いデータセンター、または停止が発生した場合に存続しているデータセンターに送信することにより、データセンター間で負荷を分散できます。

このセクションでは、Windows PowerShell コマンドを使用して、Microsoft Azure 環境の 2 つのサイトの VPX インスタンスで GSLB を有効にする方法について説明します。

> 注
>
> GSLB の詳細については、「グローバルサーバーの負荷分散」を参照してください。

Azure 上の NetScaler VPX インスタンスで GSLB を構成するには、次の 2 つのステップがあります。

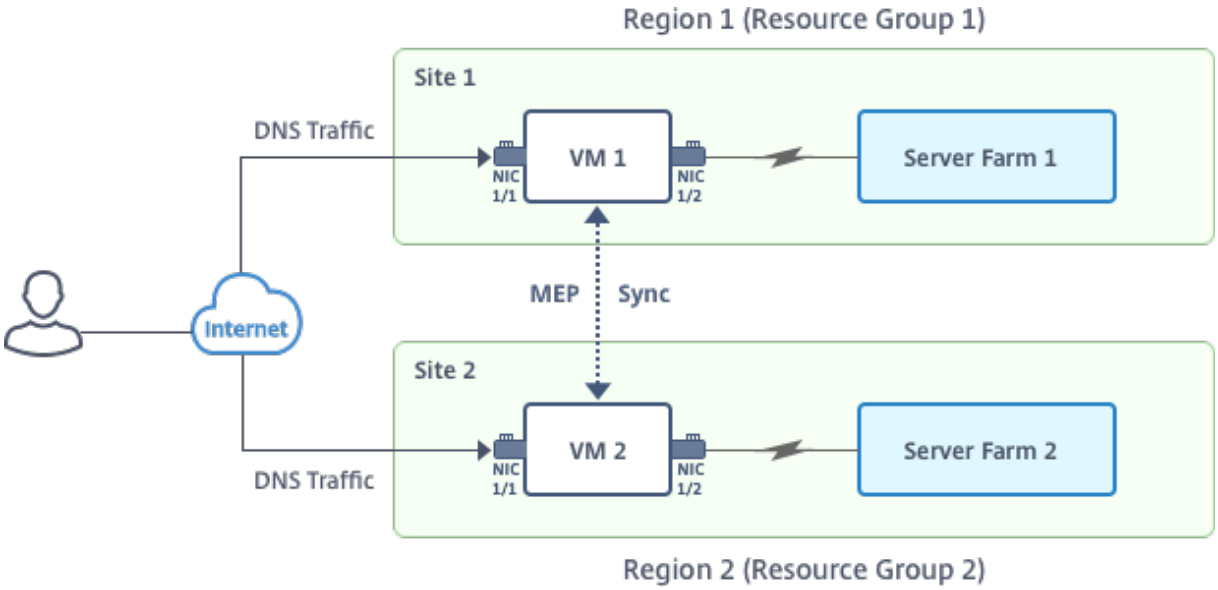
1. 各サイトに、複数の NIC と複数の IP アドレスを持つ VPX インスタンスを作成します。
2. VPX インスタンスで GSLB を有効にします。

> 注
>
> 複数の NIC および IP アドレスの構成の詳細については、「PowerShell コマンドを使用してスタンドアロンモードで NetScaler ADC VPX インスタンスの複数の IP アドレスを構成する」を参照してください。

シナリオ

このシナリオには、2 つのサイト（Site 1 と Site 2）が含まれています。各サイトの VM（VM1 と VM2）には、複数の NIC、複数の IP アドレス、および GSLB が構成されています。

フィギュア。2 つのサイト（Site 1 と Site 2）に実装された GSLB セットアップ



このシナリオでは、各 VM には 3 つの NIC（NIC 0/1、1/1、1/2）が設定されています。各 NIC に複数のプライベートおよびパブリック IP アドレスを設定できます。これらの NIC は次の目的で構成されています。

- NIC 0/1：管理トラフィックを提供する
- NIC 1/1：クライアント側のトラフィックを提供する
- NIC 1/2：バックエンドサーバーと通信する

このシナリオで各 NIC に設定された IP アドレスの詳細については、「IP 構成の詳細 」セクションを参照してください。

パラメーター

このドキュメントのこのシナリオのサンプルパラメーター設定は、次のとおりです。必要な場合は、異なる設定を使用できます。

```
$location="West Central US"

```



```
$vnetName="NSVPX-vnet"

$RGName="multiIP-RG"

$prmStorageAccountName="multiipstorageaccnt"

$avSetName="MultiIP-avset"

$vmSize="Standard_DS3_V2"

<!--NeedCopy-->
```

注**:** VPX インスタンスの最小要件は、2 つの vCPU と 2 GB RAM です。

```
$publisher="citrix"

$offer="netscalervpx111"

$sku="netscalerbyol"

$version="latest"

$vmNamePrefix="MultiIPVPX"

$nicNamePrefix="MultiipVPX"

$osDiskSuffix="osdiskdb"

$numberOfVMs=1

$ipAddressPrefix="10.0.0."

$ipAddressPrefix1="10.0.1."

$ipAddressPrefix2="10.0.2."

$pubIPName1="MultiIP-pip1"

$pubIPName2="MultiIP-pip2"

$IpConfigName1="IPConfig1"

$IPConfigName2="IPConfig-2"

$IPConfigName3="IPConfig-3"

$IPConfigName4="IPConfig-4"

$frontendSubnetName="default"

$backendSubnetName1="subnet_1"

$backendSubnetName2="subnet_2"
```



```
40
41 $suffixNumber=10
42 <!--NeedCopy-->
```

仮想マシンの作成

PowerShell コマンドを使用して、ステップ 1 ～10 に従って、複数の NIC と複数の IP アドレスを使用して VM1 を作成します。

1. リソースグループの作成
2. ストレージアカウントの作成
3. アベイラビリティセットの作成
4. 仮想ネットワークの作成
5. パブリック IP アドレスの作成
6. NIC の作成
7. VM 設定オブジェクトの作成
8. 認証情報を取得し、VM の OS プロパティを設定します
9. NIC の追加
10. OS ディスクの指定と VM の作成

すべての手順とコマンドを完了して VM1 を作成した後で、これらの手順を繰り返して VM2 固有のパラメーターで VM2 を作成します。

リソースグループの作成

```
1 New-AzureRMResourceGroup -Name $RGName -Location $location
2 <!--NeedCopy-->
```

ストレージアカウントの作成

```
1 $prmStorageAccount=New-AzureRMStorageAccount -Name
      $prmStorageAccountName -ResourceGroupName $RGName -Type Standard_LRS
       -Location $location
2 <!--NeedCopy-->
```


アベイラビリティセットの作成

```
$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
    $RGName -Location $location
<!--NeedCopy-->
```

仮想ネットワークの作成

1. サブネットを追加します。

```
$subnet1=New-AzureRmVirtualNetworkSubnetConfig -Name
    $frontendSubnetName -AddressPrefix "10.0.0.0/24"
$subnet2=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName1 -AddressPrefix "10.0.1.0/24"
$subnet3=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName2 -AddressPrefix "10.0.2.0/24"
<!--NeedCopy-->
```

2. 仮想ネットワークオブジェクトを追加します。

```
$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
     $RGName -Location $location -AddressPrefix 10.0.0.0/16 -Subnet
     $subnet1, $subnet2, $subnet3
<!--NeedCopy-->
```

3. サブネットを取得します。

```
$frontendSubnet=$vnet.Subnets|?{
 $\_.Name -eq $frontendSubnetName }

$backendSubnet1=$vnet.Subnets|?{
 $\_.Name -eq $backendSubnetName1 }

$backendSubnet2=$vnet.Subnets|?{
 $_.Name -eq $backendSubnetName2 }

<!--NeedCopy-->
```

パブリック **IP** アドレスの作成

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
    $RGName -Location $location -AllocationMethod Dynamic
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
    $RGName -Location $location -AllocationMethod Dynamic
<!--NeedCopy-->
```


NIC の作成

NIC 0/1 の作成

```
$nic1Name=$nicNamePrefix + $suffixNumber + "-Mgmnt"
$ipAddress1=$ipAddressPrefix + $suffixNumber
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    SubnetId $frontendSubnet.Id -PublicIpAddress $pip1 -PrivateIpAddress
     $ipAddress1 -Primary
$nic1=New-AzureRMNetworkInterface -Name $nic1Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig1
<!--NeedCopy-->
```

NIC 1/1 の作成

```
$nic2Name $nicNamePrefix + $suffixNumber + "-frontend"
$ipAddress2=$ipAddressPrefix1 + ($suffixNumber)
$ipAddress3=$ipAddressPrefix1 + ($suffixNumber + 1)
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -
    PublicIpAddress $pip2 -SubnetId $backendSubnet1.Id  -
    PrivateIpAddress $ipAddress2  -Primary
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    SubnetId $backendSubnet1.Id  -PrivateIpAddress $ipAddress3
nic2=New-AzureRMNetworkInterface -Name $nic2Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig2, $IpConfig3
<!--NeedCopy-->
```

NIC 1/2 の作成

```
$nic3Name=$nicNamePrefix + $suffixNumber + "-backend"
$ipAddress4=$ipAddressPrefix2 + ($suffixNumber)
$IPConfig4=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -
    SubnetId $backendSubnet2.Id -PrivateIpAddress $ipAddress4 -Primary
$nic3=New-AzureRMNetworkInterface -Name $nic3Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig4
<!--NeedCopy-->
```

VM 設定オブジェクトの作成

```
$vmName=$vmNamePrefix
$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avSet.Id
<!--NeedCopy-->
```

認証情報の取得と **OS** プロパティの設定

```
$cred=Get-Credential -Message "Type the name and password for VPX login
    ."
```



```
$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
    ComputerName $vmName -Credential $cred
$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
    $publisher -Offer $offer -Skus $sku -Version $version
<!--NeedCopy-->
```

NIC の追加

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -
    Primary
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.Id
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.Id
<!--NeedCopy-->
```

OS ディスクの指定と **VM** の作成

```
$osDiskName=$vmName + "-" + $osDiskSuffix
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
     +$osDiskName + ".vhd"
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
    $osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
    Name $sku
New-AzureRMVM -VM $vmConfig -ResourceGroupName $RGName -Location
    $location
<!--NeedCopy-->
```

> 注
>
> 「PowerShell コマンドを使用したマルチ NIC 仮想マシンの作成」に記載されている手順 1～10 を繰り返して、VM2 に固有のパラメータを使用して VM2 を作成します。

IP 構成の詳細

次の IP アドレスを使用します。

テーブル **1**. VM1 で使用する IP アドレス

NIC	プライベート IP	パブリック IP（PIP）	説明
0/1	10.0.0.10	PIP1	NSIP（管理 IP）として構成
1/1	10.0.1.10	PIP2	SNIP/GSLB サイト IP として設定されています

NIC	プライベート IP	パブリック IP（PIP）	説明
-	10.0.1.11	-	LB サーバ IP として設定されています。パブリック IP アドレスは必須ではありません
1/2	10.0.2.10	-	モニタプローブをサービスに送信するための SNIP として設定。パブリック IP は必須ではありません。

表 **2.** VM2 で使用する IP アドレス

NIC	内部 IP	パブリック IP（PIP）	説明
0/1	20.0.0.10	PIP4	NSIP（管理 IP）として構成
1/1	20.0.1.10	PIP5	SNIP/GSLB サイト IP として設定されています
-	20.0.1.11	-	LB サーバ IP として設定されています。パブリック IP アドレスは必須ではありません
1/2	20.0.2.10	-	モニタプローブをサービスに送信するための SNIP として設定。パブリック IP は必須ではありません。

このシナリオの構成例を次に示します。VM1 と VM2 の NetScaler VPX CLI で作成された IP アドレスと初期 LB 構成を示しています。

VM1 の設定例を次に示します。

```
add ns ip 10.0.1.10 255.255.255.0 -mgmtAccess ENABLED
Add nsip 10.0.2.10 255.255.255.0
add service svc1 10.0.1.10 ADNS 53
add lb vserver v1 HTTP 10.0.1.11 80
add service s1 10.0.2.120 http 80
Add service s2 10.0.2.121 http 80
Bind lb vs v1 s[1-2]
<!--NeedCopy-->
```

VM2 の設定例を次に示します。


```
add ns ip 20.0.1.10 255.255.255.0  -mgmtAccess ENABLED
Add nsip 20.0.2.10 255.255.255.0
add service svc1 20.0.1.10 ADNS 53
add lb vserver v1 HTTP 20.0.1.11 80
Add service s1 20.0.2.90 http 80
Add service s2 20.0.2.91 http 80
Bind lb vs v1 s[1-2]
<!--NeedCopy-->
```

GSLB サイトおよびその他の設定を構成する

次のトピックで説明するタスクを実行して、2 つの GSLB サイトとその他の必要な設定を構成します。

Global Server Load Balancing

詳細については、次のサポート記事https://support.citrix.com/article/CTX110348を参照してください。

VM1 および VM2 での GSLB 設定の例を次に示します。

```
enable ns feature LB GSLB
add gslb site site1 10.0.1.10 -publicIP PIP2
add gslb site site2 20.0.1.10 -publicIP PIP5
add gslb service site1_gslb_http_svc1 10.0.1.11 HTTP 80 -publicIP PIP3
    -publicPort 80 -siteName site1
add gslb service site2_gslb_http_svc1 20.0.1.11 HTTP 80 -publicIP PIP6
    -publicPort 80 -siteName site2
add gslb vserver gslb_http_vip1 HTTP
bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
<!--NeedCopy-->
```

Azure で実行されている NetScaler VPX インスタンスに GSLB を構成しました。

障害回復

災害（さいがん）とは、自然の災害、または人為的な出来事によって引き起こされる事業機能の突然の混乱である。災害はデータセンターの運用に影響を及ぼします。その後、災害現場で失われたリソースとデータを完全に再構築して復元する必要があります。データ消失やデータセンターのダウンタイムは重要であり、ビジネス継続性が低下します。

お客様が今日直面している課題の 1 つは、DR サイトをどこに置くかを決めることです。企業は、基盤となるインフラストラクチャやネットワーク障害に関係なく、一貫性とパフォーマンスを求めています。

多くの組織がクラウドへの移行を決定している理由として考えられるのは、次のとおりです。

- オンプレミスのデータセンターを持つことは非常に高価です。クラウドを使用することで、企業は自社のシステムを拡張する時間とリソースを解放できます。

- 自動オーケストレーションの多くは、より迅速なリカバリを可能にします
- 継続的なデータ保護や継続的なスナップショットを提供してデータを複製し、システム停止や攻撃から保護します。
- パブリッククラウドにすでに存在しているさまざまな種類のコンプライアンスやセキュリティ制御を顧客が必要とするユースケースをサポートします。これらにより、独自に構築するよりも、必要なコンプライアンスを簡単に達成できます。

GSLB 用に構成された NetScaler ADC は、トラフィックを最も負荷の少ないデータセンターまたは最もパフォーマンスの高いデータセンターに転送します。この構成は、アクティブ-アクティブ設定と呼ばれ、パフォーマンスが向上するだけでなく、セットアップの一部であるデータセンターがダウンした場合に、トラフィックを他のデータセンターにルーティングすることで、ディザスタリカバリを即座に実行できます。これにより、NetScaler はお客様の貴重な時間と費用を節約できます。

災害復旧のためのマルチ **NIC** マルチ **IP (3** つの **NIC)** の導入

お客様は、セキュリティ、冗長性、可用性、容量、およびスケーラビリティが重要な本番環境に導入する場合、3 つの NIC 導入を使用して導入する可能性があります。この展開方法では、複雑さと管理の容易さはユーザーにとって重大な問題ではありません。

ディザスタリカバリ用の単一 **NIC** マルチ **IP**（**1NIC**）導入

お客様は、以下の理由で非実稼働環境に導入する場合、NIC を 1 つにまとめて導入する可能性があります。

- テスト用の環境をセットアップするか、本番環境への導入前に新しい環境をステージングします。
- クラウドに迅速かつ効率的に直接デプロイします。
- 単一のサブネット構成のシンプルさを求めながら。

アクティブ/スタンバイの高可用性セットアップで **GSLB** を構成する

August 15, 2023

Azure のアクティブ/スタンバイ HA 展開には、次の 3 つの手順でグローバルサーバー負荷分散（GSLB）を構成できます。

1. 各 GSLB サイトで VPX HA ペアを作成します。HA ペアの作成方法については、「複数の IP アドレスと NIC を使用した高可用性設定の構成」を参照してください。

2. Azure Load Balancer（ALB）をフロントエンド IP アドレスと、GSLB および DNS トラフィックを許可する規則で構成します。

   この手順には、次の下位手順が含まれています。これらの下位手順の完了に使用する PowerShell コマンドについては、このセクションのシナリオを参照してください。

   a. GSLB サイトのフロントエンド`IPconfig`を作成します。

   b. HA 内のノードの NIC 1/1 の IP アドレスを持つバックエンドアドレスプールを作成します。

   c. 次のような負荷分散規則を作成します。

   ```
   TCP/3009  -  gslb communication
   TCP/3008 - gslb communication
   UDP/53 - DNS communication
   ```

   d. バックエンドアドレスプールと手順 c で作成した LB 規則を関連付けます。

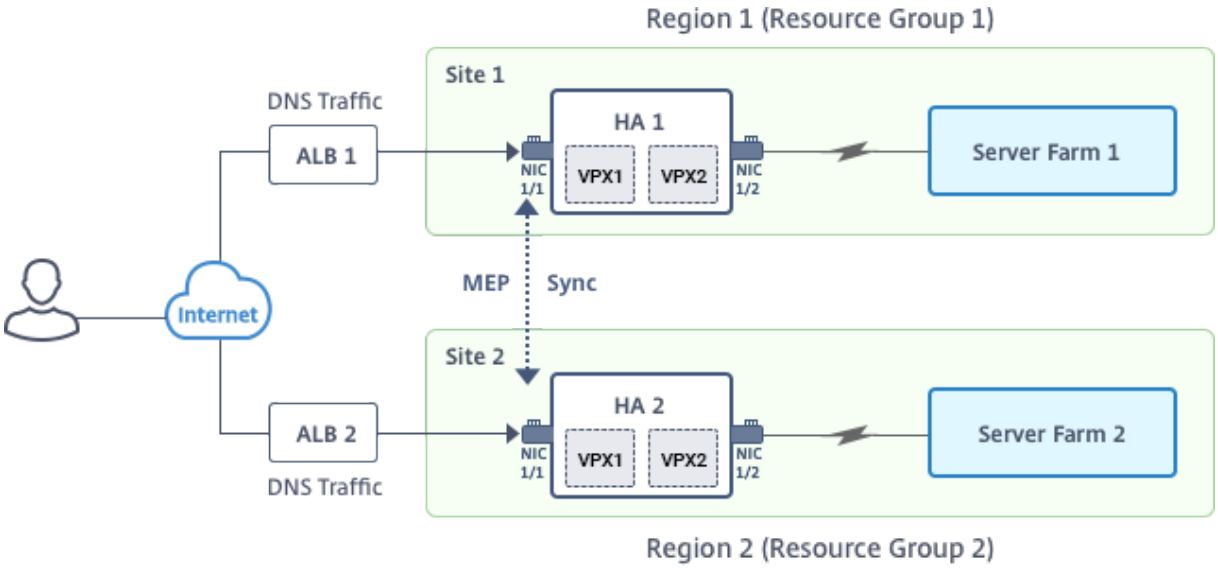
   e. 両方の HA ペアのノードの NIC 1/1 のネットワークセキュリティグループを更新して、TCP 3008、TCP 3009、および UDP 53 ポートのトラフィックを許可します。

3. 各 HA ペアで GSLB を有効にします。

シナリオ

このシナリオには、2 つのサイト（Site 1 と Site 2）が含まれています。各サイトの HA ペア（HA1 と HA2）には、複数の NIC、複数の IP アドレス、および GSLB が構成されています。

図: Azure でのアクティブ-スタンバイ HA デプロイメントでの GLSB



このシナリオでは、各 VM には 3 つの NIC（NIC 0/1、1/1、1/2）が設定されています。これらの NIC は次の目的で構成されています。

NIC 0/1：管理トラフィックを提供する

NIC 1/1：クライアント側のトラフィックを提供する

NIC 1/2：バックエンドサーバーと通信する

パラメーター設定

ALB のサンプルパラメーター設定は、次のとおりです。必要な場合は、異なる設定を使用できます。

```
$locName="South east Asia"

$rgName="MulitIP-MultiNIC-RG"

$pubIPName4="PIPFORGSLB1"

$domName4="vpxgslbdns"

$lbName="MultiIPALB"

$frontEndConfigName2="FrontEndIP2"

$backendPoolName1="BackendPoolHttp"

$lbRuleName2="LBRuleGSLB1"

$lbRuleName3="LBRuleGSLB2"

$lbRuleName4="LBRuleDNS"

$healthProbeName="HealthProbe"
```

フロントエンド **IP** アドレスとルールを使用して **ALB** を構成し、**GSLB** と **DNS** トラフィックを許可する

手順 1.**GSLB** サイト **IP** 用のパブリック **IP** を作成する

```
$pip4=New-AzureRmPublicIpAddress -Name $pubIPName4 -ResourceGroupName
    $rgName -DomainNameLabel $domName4 -Location $locName -
    AllocationMethod Dynamic


Get-AzureRmLoadBalancer -Name \$lbName -ResourceGroupName \$rgName |
    Add-AzureRmLoadBalancerFrontendIpConfig -Name \$frontEndConfigName2
    -PublicIpAddress \$pip4 | Set-AzureRmLoadBalancer
```

手順 2.**LB** ルールを作成し、既存の **ALB** を更新します。

```
$alb = get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName


```



```
$frontendipconfig2=Get-AzureRmLoadBalancerFrontendIpConfig -
    LoadBalancer $alb -Name $frontEndConfigName2


$backendPool=Get-AzureRmLoadBalancerBackendAddressPoolConfig -
    LoadBalancer $alb -Name $backendPoolName1


$healthprobe=Get-AzureRmLoadBalancerProbeConfig -LoadBalancer $alb -
    Name $healthProbeName


\$alb | Add-AzureRmLoadBalancerRuleConfig -Name \$lbRuleName2 -
    BackendAddressPool \$backendPool -FrontendIPConfiguration  \
    $frontendipconfig2 -Protocol \"Tcp\" -FrontendPort 3009 -BackendPort
     3009 -Probe \$healthprobe -EnableFloatingIP  | Set-
    AzureRmLoadBalancer


\$alb | Add-AzureRmLoadBalancerRuleConfig -Name \$lbRuleName3 -
    BackendAddressPool \$backendPool -FrontendIPConfiguration  \
    $frontendipconfig2 -Protocol \"Tcp\" -FrontendPort 3008 -BackendPort
     3008 -Probe \$healthprobe -EnableFloatingIP  | Set-
    AzureRmLoadBalancer


\$alb | Add-AzureRmLoadBalancerRuleConfig -Name \$lbRuleName4 -
    BackendAddressPool \$backendPool -FrontendIPConfiguration  \
    $frontendipconfig2 -Protocol \"Udp\" -FrontendPort 53 -BackendPort
    53 -Probe \$healthprobe -EnableFloatingIP  | Set-AzureRmLoadBalancer
```

各高可用性ペアで **GSLB** を有効にします

各 ALB（ALB 1 と ALB 2）で 2 つのフロントエンド IP アドレスを設定しました。1 つめの IP アドレスは LB 仮想サーバー、もう 1 つは GSLB サイトの IP です。

HA 1 には次のフロントエンド IP アドレスがあります。

- frontendiPOFalb1 (LB 仮想サーバー用)
- PIPFORGSLB1（GSLB IP）

HA 2 には次のフロントエンド IP アドレスがあります。

- frontendiPOFALB2 (LB 仮想サーバー用)
- PIPFORGSLB2（GSLB IP）

このシナリオでは、次のコマンドを使用します。

```
enable ns feature LB GSLB

```



```
add service dnssvc PIPFORGSLB1 ADNS 53

add gslb site site1 PIPFORGSLB1 -publicIP PIPFORGSLB1

add gslb site site2 PIPFORGSLB2 -publicIP PIPFORGSLB2

add gslb service site1_gslb_http_svc1 FrontEndIPofALB1 HTTP 80 -
   publicIP FrontEndIPofALB1 -publicPort 80 -siteName site1

add gslb service site2_gslb_http_svc1 FrontEndIPofALB2 HTTP 80 -
   publicIP FrontEndIPofALB2 -publicPort 80 -siteName site2

add gslb vserver gslb_http_vip1 HTTP

bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1

bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1

bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
```

関連リソース:

NetScaler VPX インスタンスで GSLB を構成する

Global Server Load Balancing

Azure に NetScaler GSLB を展開

April 15, 2024

需要が高まる中、地域の顧客にサービスを提供するオンプレミスデータセンターを運営している企業は、Azure クラウドを使用して世界中に規模を拡大してデプロイしたいと考えています。NetScaler をネットワーク管理者側で使用すると、GSLB StyleBook を使用してオンプレミスとクラウドの両方でアプリケーションを構成できます。NetScaler ADM を使用して同じ構成をクラウドに転送できます。GSLB との距離に応じて、オンプレミスリソースとクラウドリソースのいずれかにアクセスできます。これにより、世界のどこにいても、シームレスなエクスペリエンスを実現できます。

データベース管理システムの概要

NetScaler GSLB は、クラウドロードバランサーでのドメインベースサービス（DBS）の使用をサポートしています。これにより、クラウドロードバランサーソリューションを使用して動的クラウドサービスを自動検出できます。この構成により、NetScaler はアクティブ-アクティブ環境に GSLB DBS を実装できます。DBS では、DNS 検出から Microsoft Azure 環境のバックエンドリソースを拡張できます。このセクションでは、AzureAutoscale 環境における NetScaler 間の統合について説明します。

Azure ロードバランサー **(ALB)** を使用するドメイン名ベースのサービス

GLSB DBS は、ユーザー ALB の FQDN を利用して、Azure 内で作成および削除されるバックエンドサーバーを含むように GSLB サービスグループを動的に更新します。この機能を設定するには、ユーザーは NetScaler を ALB にポイントして、Azure 内のさまざまなサーバーに動的にルーティングします。これは、Azure 内でインスタンスが作成および削除されるたびに NetScaler を手動で更新しなくても実行できます。GSLB サービスグループ向けの NetScaler DBS 機能は、DNS 対応のサービス検出を使用して、Autoscale グループで識別される DBS 名前空間のメンバーサービスリソースを特定します。

次の図は、クラウドロードバランサーを使用する NetScaler GSLB DBS オートスケールコンポーネントを示しています。

Azure GSLB の前提条件

NetScaler GSLB サービスグループの前提条件には、セキュリティグループ、Linux ウェブサーバー、AWS 内の NetScaler アプライアンス、Elastic IP、および Elastic ロードバランサー（ELB）を設定するための知識と能力を備えた、正常に機能する Microsoft Azure 環境が含まれます。

- GSLB DBS サービスの統合には、Microsoft Azure ロードバランサーインスタンス用の NetScaler バージョン 12.0.57 が必要です。
- GSLB サービスグループエンティティ:NetScaler バージョン 12.0.57
- DBS 動的検出を使用した自動スケーリングをサポートする GSLB サービスグループが導入されました。
- DBS 機能コンポーネント (ドメインベースのサービス) は GSLB サービスグループにバインドする必要があります。

例:

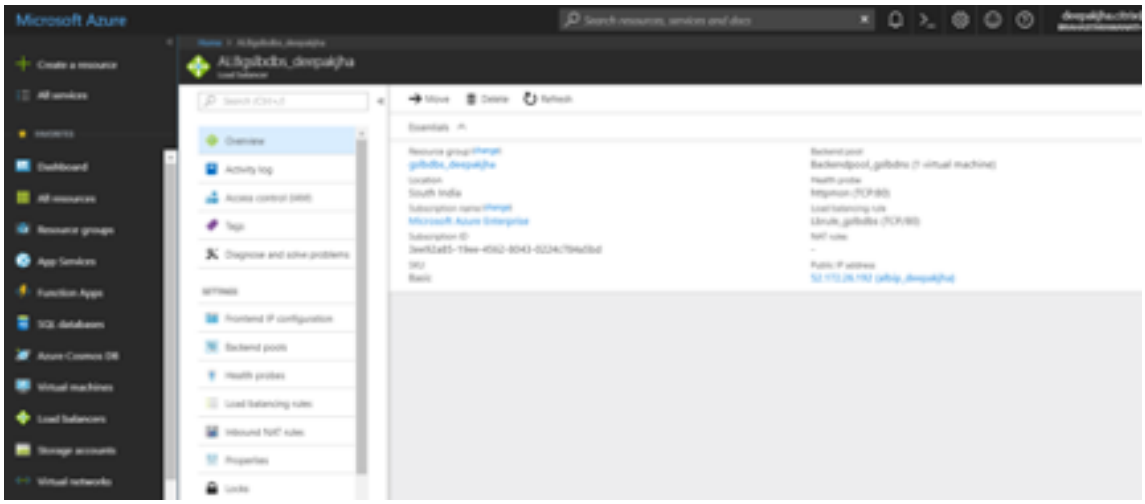
```
```
 > add server sydney_server LB-Sydney-xxxxxxxxxx.ap-southeast-2.elb.
    amazonaws.com
 > add gslb serviceGroup sydney_sg HTTP -autoscale DNS -siteName sydney
 > bind gslb serviceGroup sydney_sg sydney_server 80
```

 
```

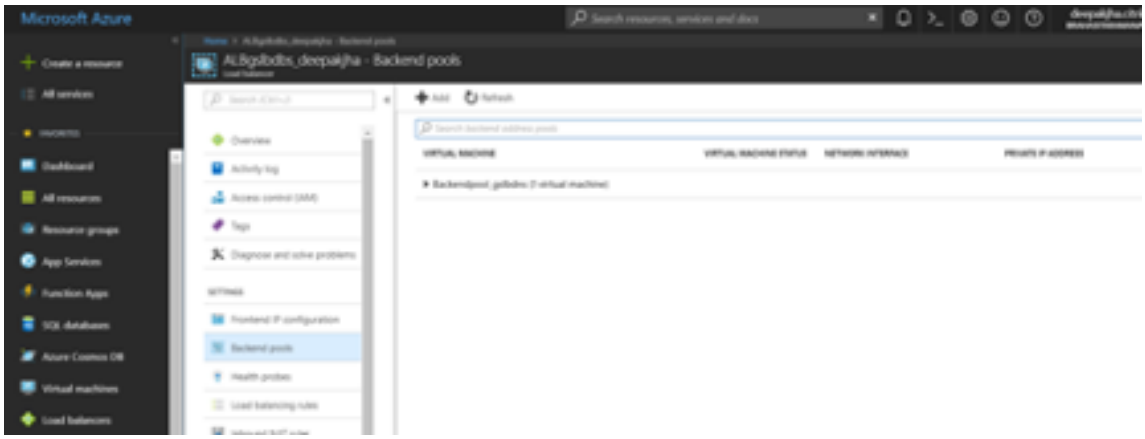
```
<!--NeedCopy--> ```
```

Azure コンポーネントの設定

1. ユーザー Azure Portal にログインし、NetScaler テンプレートから新しい仮想マシンを作成します。
2. Azure Load Balancer を作成します。



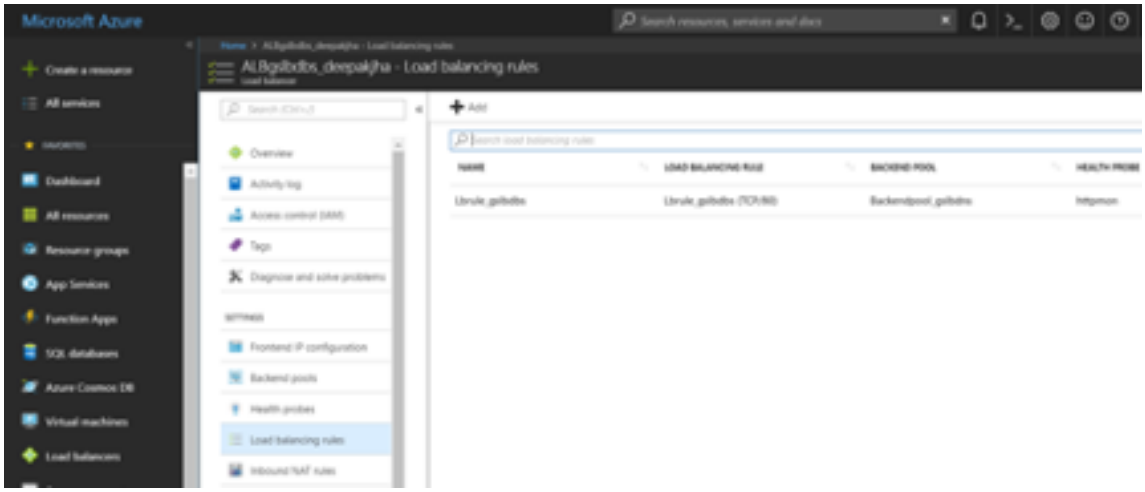
3. 作成した NetScaler バックエンドプールを追加します。



4. ポート 80 のヘルスプローブを作成します。

   ロードバランサーから作成されたフロントエンド IP を利用して負荷分散ルールを作成します。

   - プロトコル:TCP
   - バックエンドポート:80
   - バックエンドプール: 手順 1 で作成した NetScaler
   - ヘルスプローブ: ステップ 4 で作成
   - セッションの永続性: なし



NetScaler GSLB ドメインベースのサービスの設定

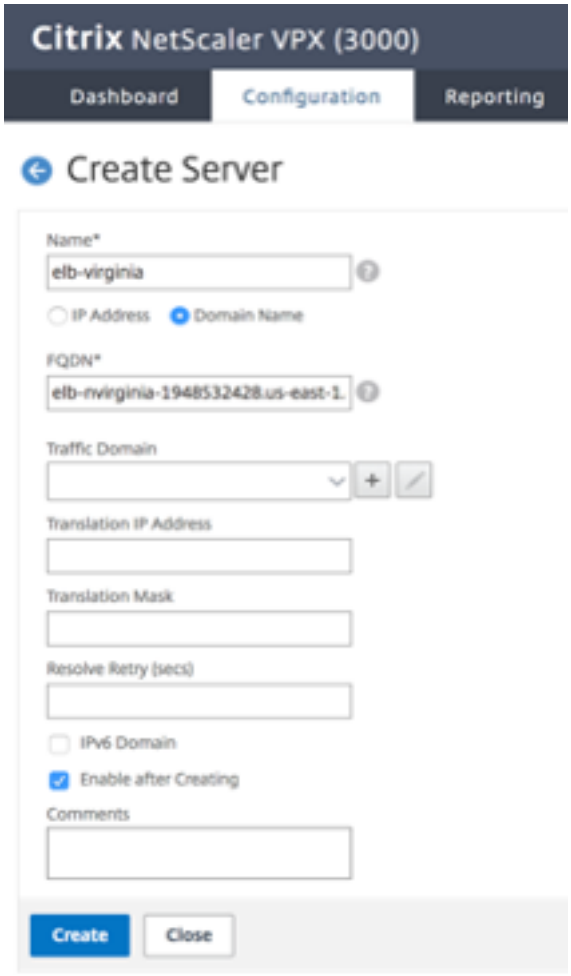
次の構成は、GSLB 対応環境で ADC を自動スケーリングするためのドメインベースのサービスを有効にするために必要なものをまとめたものです。

- トラフィック管理の設定
- GSLB 構成

トラフィック管理の設定

> 注:
>
> NetScaler をネームサーバーまたは DBS サービスグループの ELB/ALB ドメインの解決に使用する DNS 仮想サーバーのいずれかで構成する必要があります。ネームサーバーまたは DNS 仮想サーバーの詳細については、「DNS ネームサーバー」を参照してください。

1. [トラフィック管理] > [負荷分散] > [サーバー] に移動し ます。
2. [追加] をクリックしてサーバーを作成し、ALB の Azure の A レコード (ドメイン名) に対応する名前と FQDN を指定します。



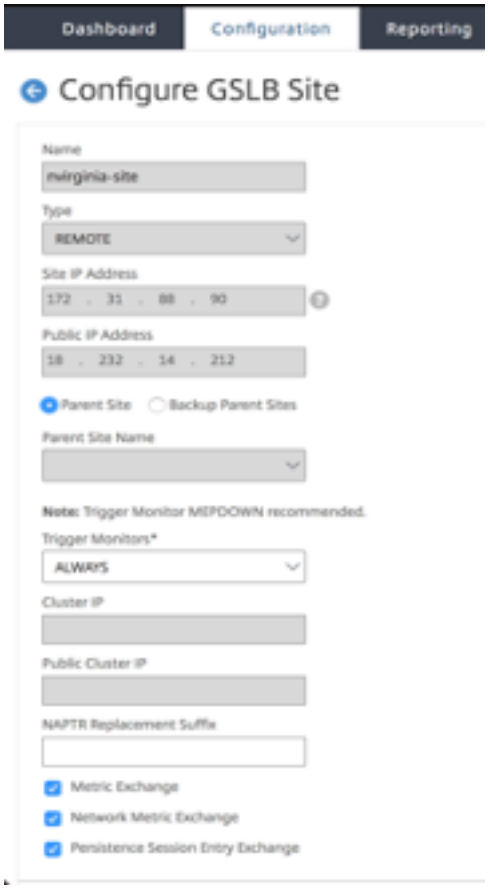
3. 手順 2 を繰り返して、Azure の 2 番目のリソースから 2 番目の ALB を追加します。

GSLB 構成

1. GSLB サイトを設定するには、[追加] ボタンをクリックします。
2. GSLB サイトを構成するための詳細を指定します。

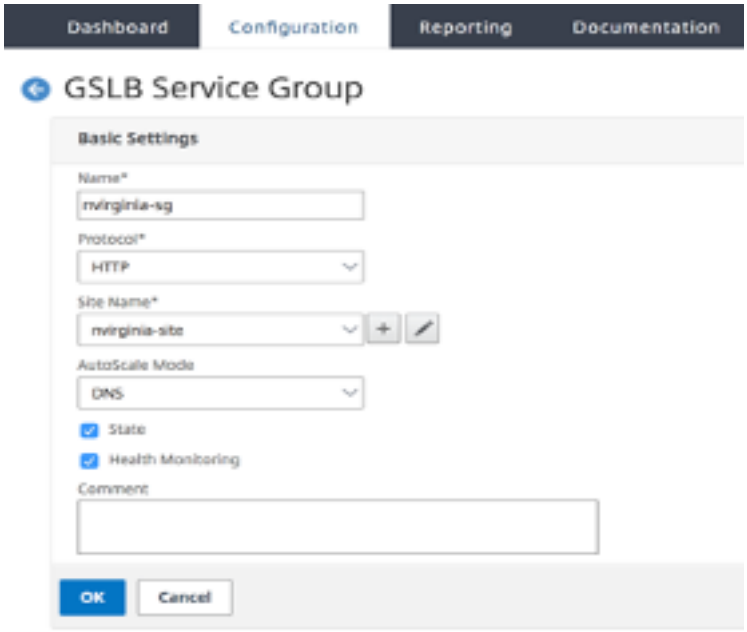
   サイトに名前を付けます。タイプは、サイトを構成する NetScaler に基づいてリモートまたはローカルとして構成されます。サイト IP アドレスは GSLB サイトの IP アドレスです。GSLB サイトは、この IP アドレスを使用して他の GSLB サイトと通信します。パブリック IP アドレスは、特定の IP アドレスが外部のファイアウォールまたは NAT デバイスでホストされているクラウドサービスを使用する場合に必要です。サイトは親サイトとして構成する必要があります。トリガーモニターが **ALWAYS** に設定されていることを確認します。また、下部にある [メトリック交換]、[ネットワークメトリック交換]、および **[**** パーシスタンスセッションエントリ交換 ****] の 3 つのボックスを必ずオンにしてください。

   トリガーモニターを **MEPDOWN** に設定することをお勧めします。詳細については、「GSLB サービスグループの設定」を参照してください。

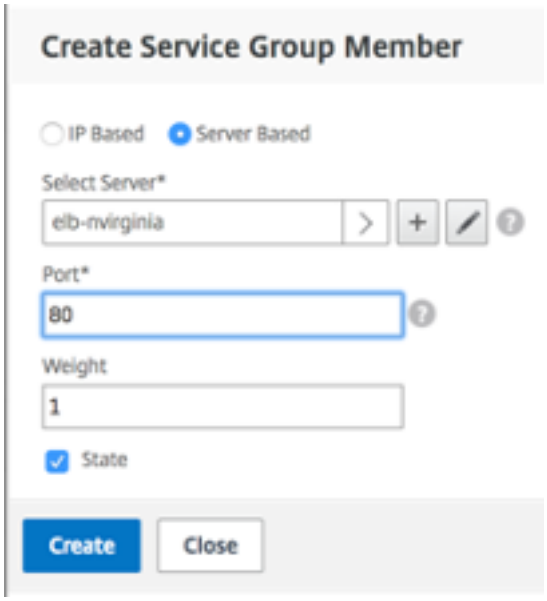


3. [**Create**] をクリックします。
4. [トラフィック管理**] > [GSLB] > [**サービスグループ] に移動します。
5. [追加] をクリックしてサービスグループを追加します。
6. 詳細を指定してサービスグループを設定します

   サービスグループに名前を付け、HTTP プロトコルを使用します。[サイト名] で、作成した各サイトを選択します。必ず自動スケールモードを DNS として設定し、状態およびヘルスモニタリングのチェックボックスをオフにします。**OK** をクリックしてサービスグループを作成します。



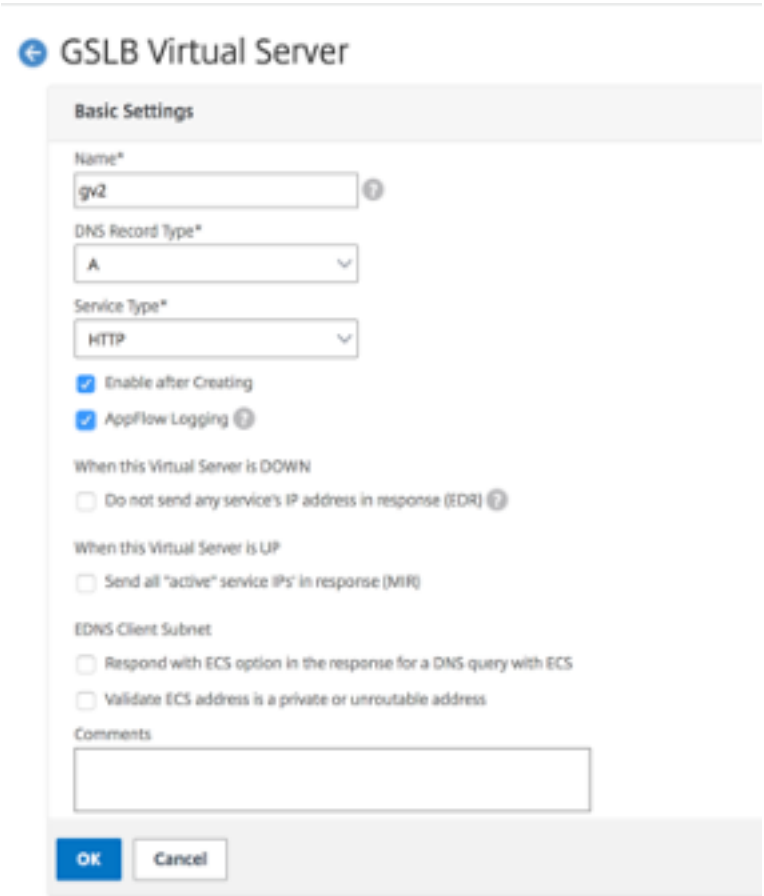
7. [サービスグループメンバー] をクリックし、[サーバーベース] を選択します。実行ガイドの開始時に設定した各 ELB を選択します。トラフィックがポート 80 を通過するように設定します。[**Create**] をクリックします。



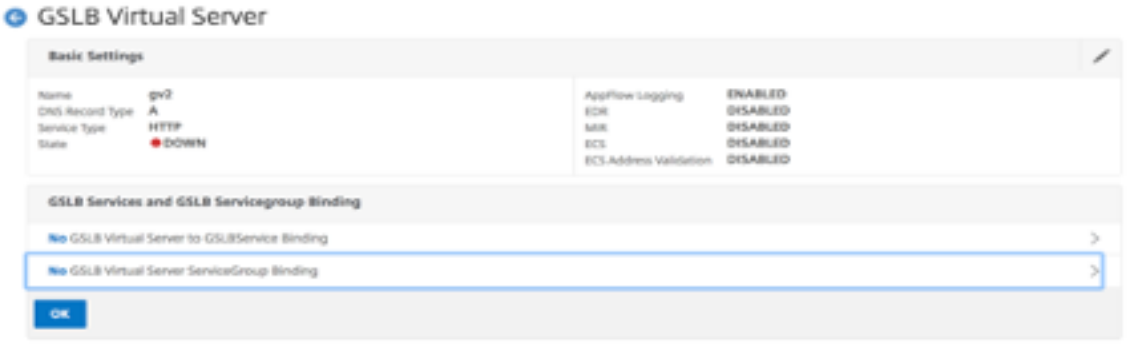
サービスグループメンバーバインディングには、ELB から受信する 2 つのインスタンスが入力されている必要があります。

8. 手順 5 と 6 を繰り返して、Azure の 2 番目のリソースロケーションのサービスグループを設定します。（これは同じ NetScaler GUI から実行できます）。
9. GSLB 仮想サーバーをセットアップします。[トラフィック管理] **> [GSLB] > [**仮想サーバー] に移動します。
10. [追加] をクリックして仮想サーバーを作成します。
11. 詳細を指定して GSLB 仮想サーバーを構成します。

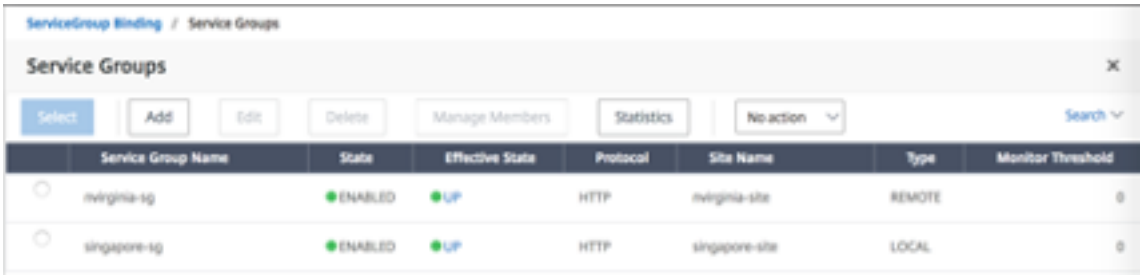
    サーバーの名前を指定し、DNS レコードタイプが A に設定され、サービスタイプが HTTP に設定され、AppFlow ロギングの作成後に有効にするチェックボックスをオンにします。**OK** をクリックして GSLB 仮想サーバーを作成します。



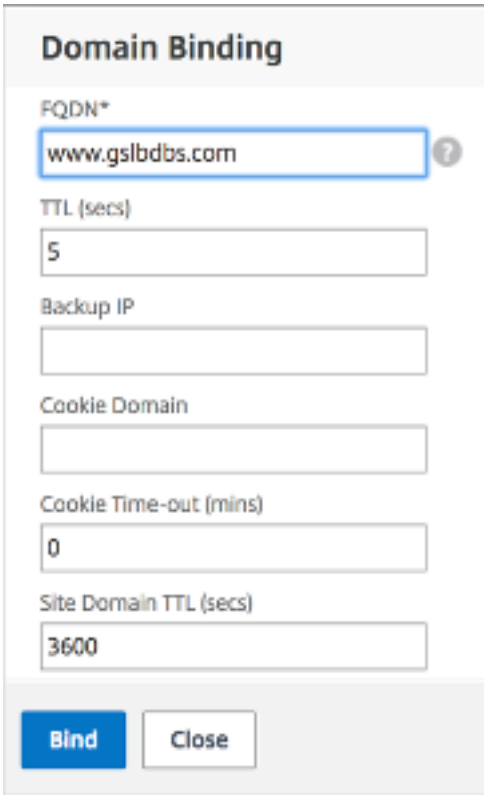
12. GSLB 仮想サーバーを作成したら、「**GSLB** 仮想サーバーサービスグループバインディングなし」をクリックします。



13. 「サービスグループバインディング」で 、「サービスグループ名 の選択」を使用して、前のステップで作成したサービスグループを選択して追加します。



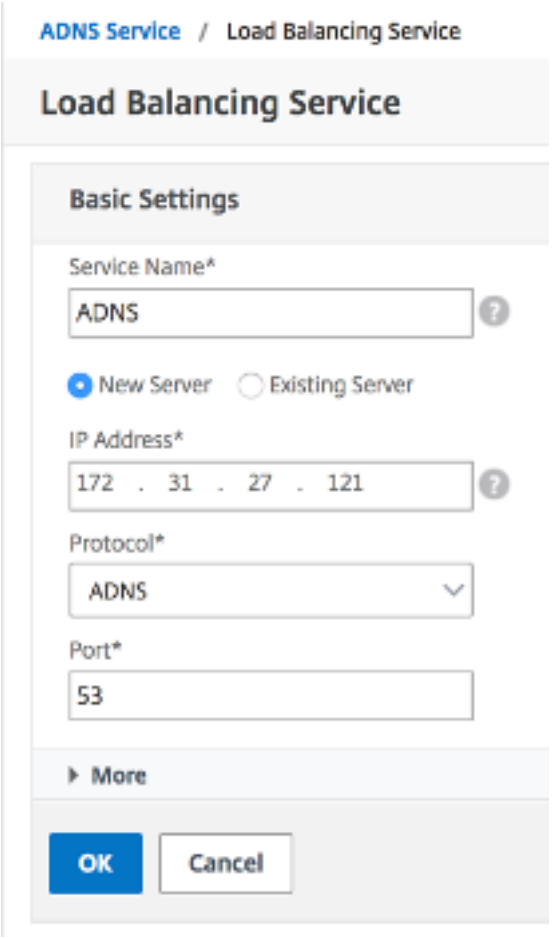
14. GSLB 仮想サーバードメインバインディングを設定するには、「**GSLB** 仮想サーバードメインバインディングなし」をクリックします。FQDN とバインドを設定します。他のパラメータはデフォルト設定のままにします。



15. [サービスなし] をクリックして **ADNS** サービスを設定します。

16. 詳細を指定して負荷分散サービスを設定します。

    サービス名を追加し、[新規サーバー] をクリックして、ADNS サーバーの **IP** アドレスを入力します。ユーザ ADNS がすでに設定されている場合、ユーザは [既存のサーバ] を選択し、ドロップダウンメニューからユーザ ADNS を選択できます。プロトコルが ADNS で、トラフィックがポート 53 を経由するように設定されていることを確認します。



17. 方法を [最小接続] に、[バックアップ方法] を [ラウンドロビン] に設定します。

18.「完了」をクリックし、ユーザーの GSLB 仮想サーバーが「Up」と表示されていることを確認します。

そのほかの参照先

ハイブリッドおよびマルチクラウド環境向けの NetScaler グローバル負荷分散

NetScaler Web App Firewall を Azure にデプロイする

April 15, 2024

NetScaler Web App Firewall は、最新のアプリケーションに最先端の保護を提供するエンタープライズグレードのソリューションです。NetScaler Web App Firewall は、Web サイト、Web アプリケーション、API などの一般公開資産に対する脅威を軽減します。NetScaler Web App Firewall には、IP レピュテーションベースのフィルタリング、ボット対策、OWASP トップ 10 アプリケーション脅威対策、レイヤー 7 DDoS 保護などが含まれています。また、認証を強制するオプション、強力な SSL/TLS 暗号、TLS 1.3、レート制限、および書き換えポリシーも含まれています。NetScaler Web App Firewall は、基本的な WAF 保護と高度な WAF 保護の両方を使用して、比類のない使いやすさでアプリケーションを包括的に保護します。起動して実行するのはほんの数分です。さらに、NetScaler Web App Firewall は動的プロファイリングと呼ばれる自動学習モデルを使用することで、ユーザーの貴重な時間を節約できます。NetScaler Web App Firewall は、保護対象アプリケーションの動作を自動的に学習することで、開発者がアプリケーションを展開したり変更したりしても、アプリケーションに適応します。NetScaler Web App Firewall は、PCI-DSS、HIPAA などを含むすべての主要な規制基準や機関へのコンプライアンスに役立ちます。CloudFormation テンプレートを使えば、これまでになく簡単に立ち上げてすぐに実行できます。Auto Scaling を使用すると、トラフィックが拡大しても、ユーザーはアプリケーションを保護したまま安心できます。

NetScaler Web App Firewall は、顧客サーバーと顧客ユーザー間のレイヤー 3 ネットワークデバイスまたはレイヤー 2 ネットワークブリッジとして、通常は顧客企業のルーターまたはファイアウォールの背後に設置できます。詳しくは、「NetScaler Web App Firewall 概要」を参照してください。

NetScaler Web App Firewall 導入戦略

1. Web アプリケーションファイアウォールの導入は、どのアプリケーションや特定のデータを最大限のセキュリティ保護が必要か、どのアプリケーションが脆弱性が低く、どのアプリケーションまたは特定のデータがセキュリティ検査を安全に回避できるかを評価することです。これにより、ユーザーは最適な構成を考案し、トラフィックを分離するための適切なポリシーとバインドポイントを設計できます。たとえば、ユーザーは、画像、MP3 ファイル、ムービーなどの静的な Web コンテンツに対する要求のセキュリティ検査をバイパスするポリシーを構成し、動的コンテンツのリクエストに高度なセキュリティチェックを適用する別のポリシーを構成することができます。ユーザーは複数のポリシーとプロファイルを使用して、同じアプリケーションの異なるコンテンツを保護できます。
2. 導入のベースラインとなるには、仮想サーバーを作成し、そのサーバーを通過するトラフィックをテストして、ユーザーシステムを流れるトラフィックの速度と量を把握します。

3. Web アプリケーションファイアウォールを展開します。NetScaler Console と Web アプリケーションファイアウォール StyleBook を使用して、Web アプリケーションファイアウォールを構成します。詳細については、このガイドの下の「StyleBook」セクションを参照してください。
4. NetScaler Web App Firewall と OWASP トップテンを実装します。

Web Application Firewall の 3 つの保護は、一般的な種類の Web 攻撃に対して特に効果的であるため、他のどの保護よりも一般的に使用されています。したがって、これらは初期展開時に実装する必要があります。これには、次の種類のアカウントがあります:

- **HTML** クロスサイトスクリプティング: スクリプトが配置されている Web サイトとは異なる Web サイトのコンテンツにアクセスまたは変更しようとするスクリプトのリクエストとレスポンスを調べます。このチェックは、このようなスクリプトを検出すると、要求または応答を宛先に転送する前にスクリプトを無害にするか、接続をブロックします。
- **HTML SQL** インジェクション: フォームフィールドデータを含むリクエストに SQL コマンドを SQL データベースに挿入しようとしていないかを調べます。このチェックは、挿入された SQL コードを検出すると、要求をブロックするか、または Web サーバーに要求を転送する前に、挿入された SQL コードを無害にします。

  > 注:
  >
  > 構成に次の条件が適用されるように Web App Firewall が正しく構成されていることを確認してください。
  >
  > ```
  > >*  ユーザーが HTML クロスサイトスクリプティングチェックまたは HTML SQL インジェクションチェック (あるいはその両方) を有効にする場合。
  > >
  > >*  ユーザー保護された Web サイトは、ファイルのアップロードを許可したり、大きな POST 本文データを格納できる Web フォームを含んだりできます。
  > ```

このケースを処理するための Web アプリケーションファイアウォールの構成の詳細については、「アプリケーションファイアウォールの構成:Web App Firewall の構成」を参照してください。

- バッファオーバーフロー: リクエストを調べて、Web サーバーでバッファオーバーフローを引き起こそうとする試みを検出します。

Web アプリケーションファイアウォールの設定

NetScaler Web App Firewall がすでに有効になっていて、正しく機能していることを確認します。Web アプリケーションファイアウォールスタイルブックを使用して NetScaler Web App Firewall を構成することをお勧めします。ほとんどのユーザーは、これが Web アプリケーションファイアウォールを構成する最も簡単な方法であり、間違いを防ぐように設計されています。GUI とコマンドラインインターフェースはどちらも、主に既存の構成を変更したり、詳細オプションを使用したりする経験のあるユーザーを対象としています。

SQL インジェクション

NetScaler Web App Firewall HTML SQL インジェクションチェックは、ユーザーアプリケーションのセキュリティを侵害する可能性のある不正な SQL コードの注入に対する特別な防御策を提供します。NetScaler Web App Firewall は、1) POST 本文、2) ヘッダー、3) クッキーの 3 つの場所で、注入された SQL コードのリクエストペイロードを調べます。詳細については、「HTML SQL インジェクションチェック」を参照してください。

クロスサイトスクリプティング

HTML クロスサイトスクリプティング (クロスサイトスクリプティング) チェックでは、クロスサイトスクリプティング攻撃の可能性について、ユーザーリクエストのヘッダーと POST 本文の両方を調べます。クロスサイトスクリプトが見つかった場合は、攻撃を無害化するようにリクエストを変更（変換）するか、リクエストをブロックします。詳細については、「HTML クロスサイトスクリプティングチェック」を参照してください。

バッファオーバーフローチェック

バッファオーバーフローチェックは、Web サーバ上でバッファオーバーフローを引き起こす試みを検出します。Web アプリケーションファイアウォールが URL、Cookie、またはヘッダーが設定された長さよりも長いことを検出すると、バッファオーバーフローを引き起こす可能性があるため、要求をブロックします。詳細については、「バッファオーバーフローチェック」を参照してください。

仮想パッチ/署名

シグネチャは、既知の攻撃からユーザーの Web サイトを保護するタスクを簡略化するために、特定の設定可能なルールを提供します。シグニチャは、オペレーティングシステム、Web サーバー、Web サイト、XML ベースの Web サービス、またはその他のリソースに対する既知の攻撃のコンポーネントであるパターンを表します。事前設定された豊富な組み込みルールやネイティブルールは、パターンマッチングの力を利用して攻撃を検出し、アプリケーションの脆弱性から保護する、使いやすいセキュリティソリューションを提供します。詳細については、「署名」を参照してください。

NetScaler Web App Firewall は、署名の自動更新と手動更新の両方をサポートしています。
また、署名の自動更新を有効にして最新の状態に保つことをお勧めします。

これらの署名ファイルは AWS 環境でホストされているため、最新の署名ファイルを取得するには、ネットワークファイアウォールから NetScaler IP アドレスへのアウトバウンドアクセスを許可することが重要です。リアルタイムトラフィックの処理中に NetScaler の署名を更新しても影響はありません。

アプリケーション・セキュリティ分析

アプリケーションセキュリティダッシュボードは、ユーザーアプリケーションのセキュリティ状態の全体像を提供します。たとえば、セキュリティ違反、署名違反、脅威インデックスなどの主要なセキュリティメトリックが表示されます。アプリケーションセキュリティダッシュボードには、検出された NetScaler の syn 攻撃、スモールウィンドウ攻撃、DNS フラッド攻撃などの攻撃関連情報も表示されます。

> 注:
>
> アプリケーションセキュリティダッシュボードのメトリックを表示するには、ユーザーが監視したい NetScaler インスタンスで AppFlow for Security Insight を有効にする必要があります。

アプリケーションセキュリティダッシュボードで NetScaler インスタンスのセキュリティメトリックを表示するには:

1. 管理者の資格情報を使用して NetScaler コンソールにログインします。
2. [アプリケーション] **> [App Security Dashboard]** に移動し、[デバイス] リストからインスタンスの IP アドレスを選択します。

ユーザーは、グラフにプロットされたバブルをクリックすることで、Application Security Investigator で報告された不一致をさらに掘り下げることができます。

ADM での集中型学習

NetScaler Web App Firewall は、SQL インジェクションやクロスサイトスクリプティング（XSS）などの悪意のある攻撃からユーザーの Web アプリケーションを保護します。データ侵害を防ぎ、適切なセキュリティ保護を提供するために、ユーザーはトラフィックの脅威を監視し、攻撃に関する実用的なデータをリアルタイムで監視する必要があります。報告された攻撃は誤検知であり、例外として提供する必要がある場合があります。

NetScaler Console での集中学習は、WAF がユーザー Web アプリケーションの動作（通常のアクティビティ）を学習できるようにする反復型パターンフィルターです。エンジンは、モニタリングに基づいて、HTTP トラフィックに適用されるセキュリティチェックごとに推奨されるルールまたは例外のリストを生成します。

必要な緩和として手動で展開するよりも、学習エンジンを使用して緩和ルールを展開する方がはるかに簡単です。

学習機能を展開するには、ユーザーは最初にユーザー NetScaler で Web アプリケーションファイアウォールプロファイル（セキュリティ設定セット）を構成する必要があります。詳細については、「Web App Firewall プロファイルの作成」を参照してください。

NetScaler Console は、セキュリティチェックごとに例外（緩和）のリストを生成します。管理者は、NetScaler Console で例外の一覧を確認して、展開するかスキップするかを決定できます。

NetScaler コンソールの WAF 学習機能を使用すると、次のことが可能になります:

- 次のセキュリティチェックを使用して学習プロファイルを設定します。

- バッファオーバーフロー
- HTML クロスサイトスクリプティング

  > 注:
  >
  > クロスサイトスクリプトの場所の制限は FormField のみです。

- HTML SQL インジェクション

  > 注:
  >
  > HTML SQL インジェクションチェックを行うには、ユーザーが NetScaler で`set -sqlinjectionTransformSpecialChars ON`と`set -sqlinjectiontype sqlspclcharorkeywords`を構成する必要があります。

- NetScaler Console で緩和ルールを確認し、必要なアクション（展開またはスキップ）を実行することを決定します。
- メール、Slack、ServiceNow を通じて通知を受け取ることができます。
- ダッシュボードを使用してリラクゼーションの詳細を表示します。

NetScaler コンソールで WAF ラーニングを使用するには:

1. 学習プロファイルの設定: 学習プロファイルの設定
2. リラクゼーションルールを見る: リラクゼーションルールとアイドルルールを見る
3. WAF ラーニングダッシュボードを使用する: WAF ラーニングダッシュボードを表示する

StyleBook

StyleBooks は、ユーザーアプリケーションの複雑な NetScaler 構成を管理するタスクを簡素化します。StyleBook は、ユーザーが NetScaler 構成を作成および管理するために使用できるテンプレートです。ここでは、ユーザーは主に Web アプリケーションファイアウォールの展開に使用される StyleBook に関心があります。StyleBook について詳しくは、「StyleBook」を参照してください。

セキュリティインサイト分析

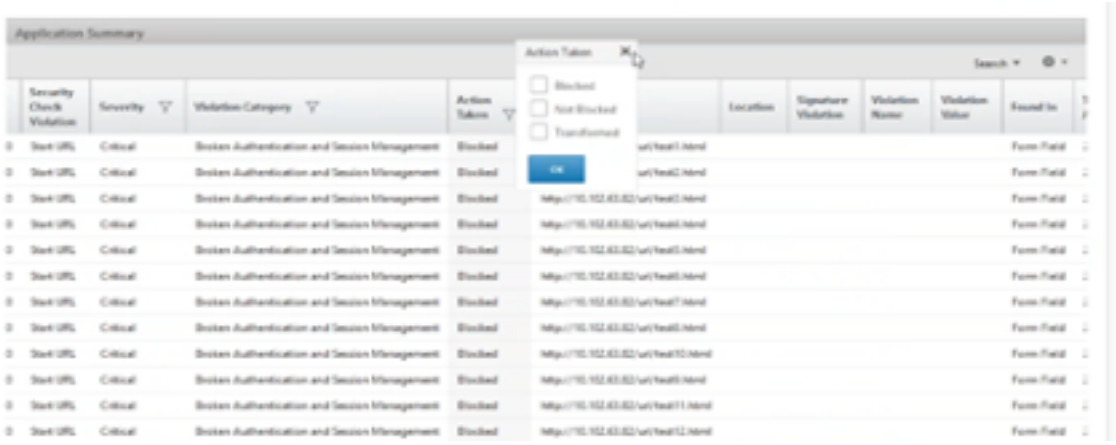
インターネットに接続している Web アプリケーションや Web サービスアプリケーションの攻撃に対する脆弱性が高まってきています。アプリケーションを攻撃から保護するために、ユーザーは過去、現在、差し迫った脅威の性質と範囲、攻撃に関するリアルタイムの実用的なデータ、および対策に関する推奨事項を可視化する必要があります。Security Insight は、ユーザーがユーザーアプリケーションのセキュリティ状態を評価し、ユーザーアプリケーションを保護するための是正措置を講じるのに役立つ、単一ペインのソリューションを提供します。
詳細については、「セキュリティインサイト」を参照してください。

セキュリティ侵害に関する詳細情報の取得

ユーザーは、アプリケーションに対する攻撃のリストを表示し、攻撃の種類と重大度、ADC インスタンスによって実行されたアクション、要求されたリソース、および攻撃元に関する洞察を得ることができます。

たとえば、ユーザーは、ブロックされた Microsoft Lync に対する攻撃の数、要求されたリソース、および送信元の IP アドレスを特定したい場合があります。

Security Insight ダッシュボードで、[**Lync**] > [合計違反数] をクリックします。テーブルで、[実行されたアクション] 列見出しのフィルタアイコンをクリックし、[ブロック] を選択します。



要求されたリソースについては、[**URL**] 列を確認してください。攻撃元については、「**Client IP** 」列を参照 してください。

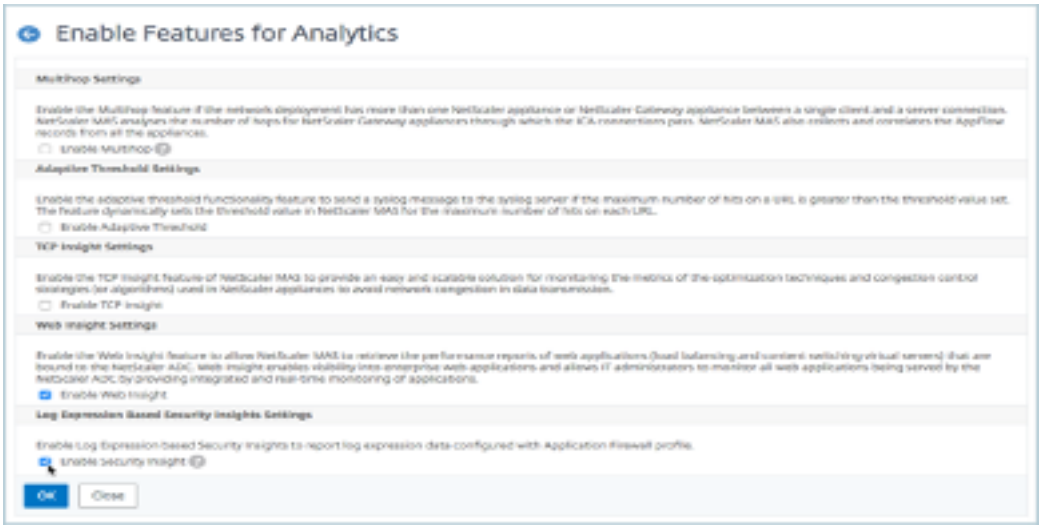
ログ式の詳細を表示する

NetScaler は、アプリケーションファイアウォールプロファイルで構成されたログ式を使用して、ユーザーエンタープライズ内のアプリケーションに対する攻撃に対処します。**Security Insight** では、ユーザーは ADC インスタンスが使用するログ式に対して返された値を表示できます。これらの値には、要求ヘッダー、要求本文などがあります。ログ式の値に加えて、ユーザーは、ADC インスタンスが攻撃のアクションを実行するために使用したアプリケーションファイアウォールプロファイルで定義されたログ式の名前とコメントを表示することもできます。

前提条件:

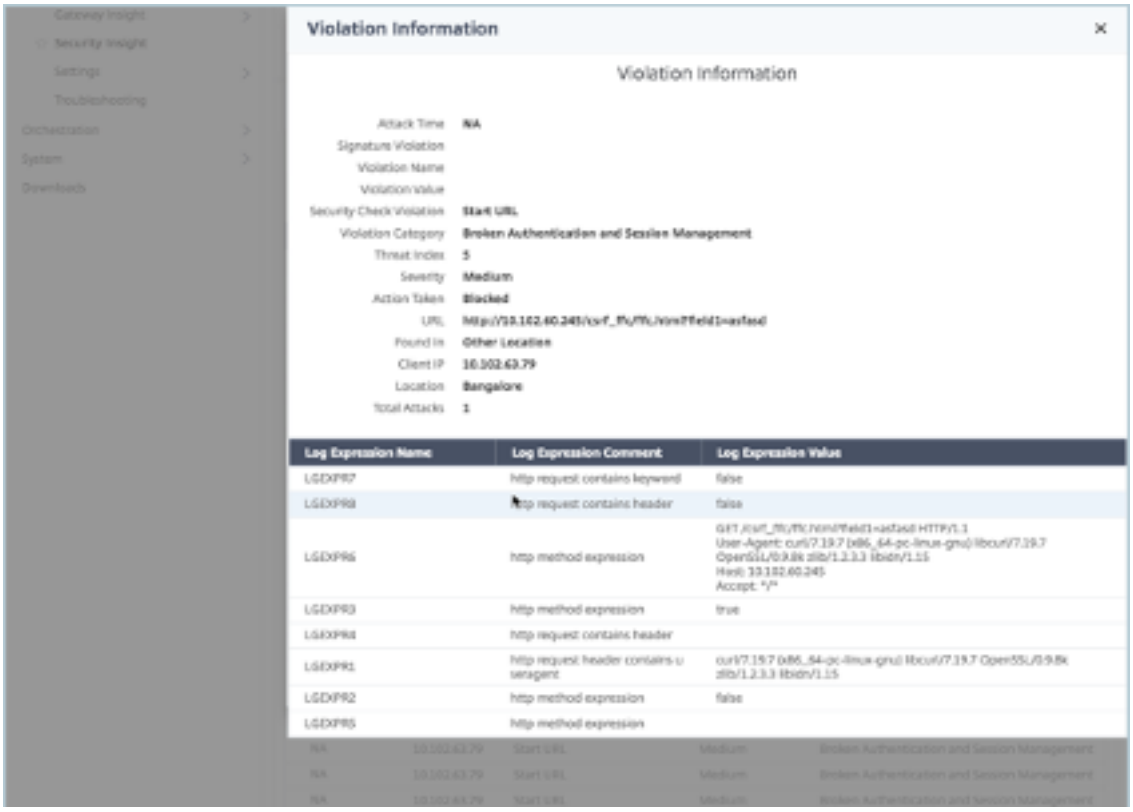
ユーザーが以下を行うことを確認します。

- アプリケーションファイアウォールプロファイルでログ式を設定します。詳しくは、「アプリケーションファイアウォール」を参照してください。
- NetScaler コンソールでログ式ベースのセキュリティインサイト設定を有効にします。以下を実行します:
  - [分析] > [設定] に移動し、[分析の機能を有効にする] をクリックします。
  - 「分析の機能を有効にする」ページで、「** ログ式ベースの **Security Insight** 設定」セクションで「**Security Insight** を有効にする」を選択し、「OK」** をクリックします。



たとえば、ユーザーエンタープライズ内の Microsoft Lync への攻撃に対して実行したアクションについて、ADC インスタンスから返されたログ式の値を表示したい場合があります。

Security Insight ダッシュボードで、[**Lync**] > [合計違反数] に移動します。[アプリケーションサマリ] テーブルで、URL をクリックして、[違反情報] ページに、ログ式の名前、コメント、アクションの ADC インスタンスによって返された値など、違反の完全な詳細を表示します。

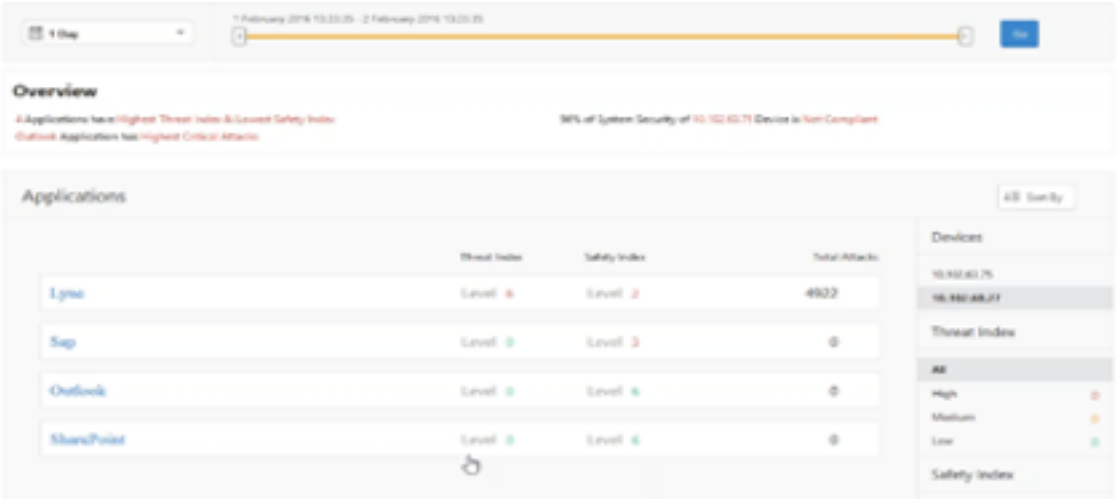


構成を展開する前に安全指数を決定してください。セキュリティ違反は、ユーザーがセキュリティ構成を ADC インスタンスに展開した後に発生しますが、ユーザーはセキュリティ構成を展開する前にその有効性を評価したい場合が

あります。

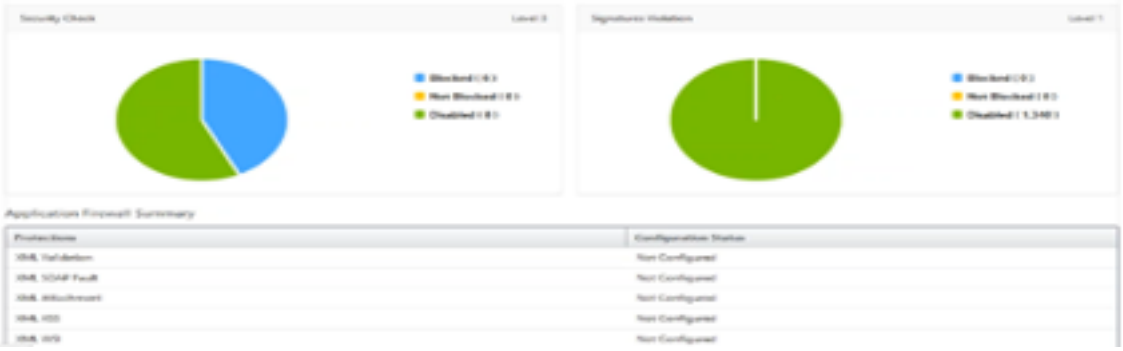
たとえば、ユーザーは、IP アドレスが 10.102.60.27 の ADC インスタンス上の SAP アプリケーションの構成の安全性指標を評価したい場合があります。

Security Insight ダッシュボードの [デバイス] で、ユーザーが設定した ADC インスタンスの IP アドレスをクリックします。ユーザーは、脅威インデックスと攻撃の総数の両方が 0 であることがわかります。脅威インデックスは、アプリケーションに対する攻撃の数と種類を直接反映しています。攻撃回数がゼロということは、アプリケーションがまったく脅威にさらされていないことを示しています。



Sap > 安全性指数 **>**SAP_Profile** をクリックし、表示される安全性指標情報を評価します。

アプリケーションファイアウォールの概要では、ユーザーはさまざまな保護設定の構成ステータスを表示できます。ログを記録する設定になっている場合や、構成されていない設定がある場合は、アプリケーションに割り当てられる安全性指数は低くなります。



セキュリティ違反

インターネットに公開されている Web アプリケーションは、攻撃に対して非常に脆弱になっています。NetScaler Console では、実行可能な違反の詳細を視覚化して、アプリケーションを攻撃から保護できます。

アプリケーションのセキュリティ違反の詳細を表示する

インターネットに公開されている Web アプリケーションは、攻撃に対して非常に脆弱になっています。NetScaler Console では、実行可能な違反の詳細を視覚化して、アプリケーションを攻撃から保護できます。「セキュリティ」 > 「**** セキュリティ違反 **** 」に 移動すると、単一ペインのソリューションで次のことが可能になります:

- ネットワーク、ボット、**WAF** などのカテゴリに基づいてアプリケーションセキュリティ違反にアクセスする
- アプリケーションを保護するための是正措置を講じる

NetScaler Console でセキュリティ違反を確認するには、以下を確認してください:

- ユーザーは NetScaler のプレミアムライセンス（WAF および BOT 違反用）を持っています。
- ユーザーは、負荷分散またはコンテンツスイッチ仮想サーバー（WAF および BOT 用）のライセンスを申請しました。詳細については、「仮想サーバーでのライセンスの管理」を参照してください。
- ユーザーはより多くの設定を有効にできます。詳しくは、NetScaler 製品ドキュメントの「セットアップ」セクション「セットアップ」に 記載さ れている手順を参照してください。

違反カテゴリ

NetScaler Console では、「すべての違反」に含まれる違反を確認できます:

設定する

違反については、メトリクスコレクターが有効になっているか どうか を確認してください。デフォルトでは、**NetScaler** の メトリックコレクターは有効になっています。詳細については、「インテリジェントアプリ分析の設

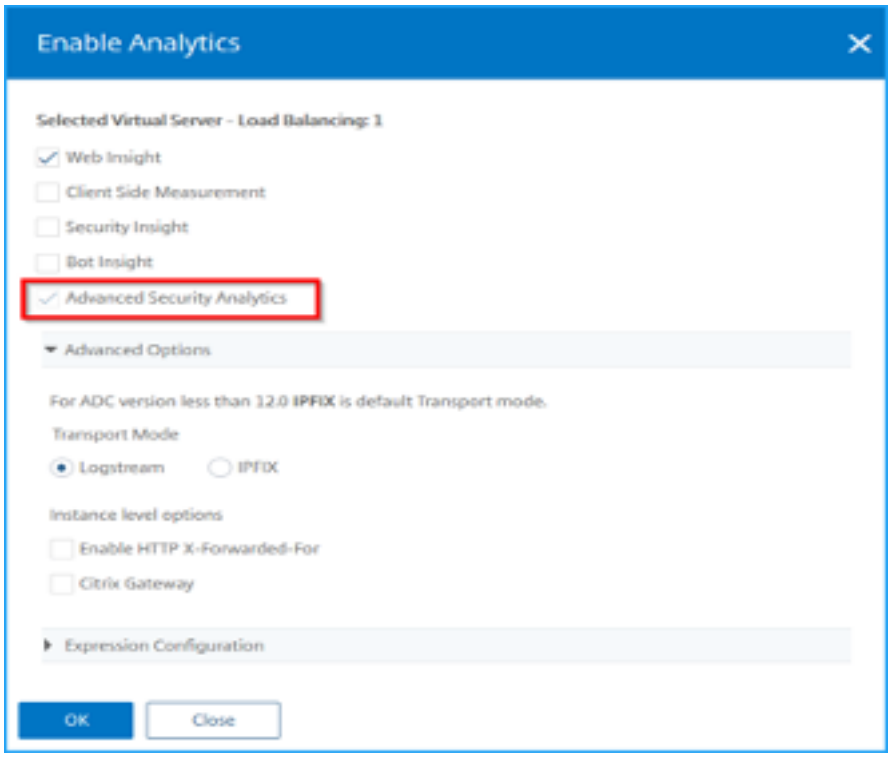
定」を参照してください。

高度なセキュリティ分析を有効にする

- [ネットワーク] > [インスタンス] > [**NetScaler**] に移動し、インスタンスタイプを選択します。たとえば、MPX。
- NetScaler インスタンスを選択し、［アクション の 選択］リストから［分析の設定］を選択 します。
- 仮想サーバーを選択し、「アナリティクスを有効にする」をクリック します。
- **[**アナリティクス を有効に する] ウィンドウで:
  - 「**Web** インサイト」を選択 します。ユーザーが Web Insight を選択すると、読み取り専用の [高度なセキュリティ分析] オプションが自動的に有効になります。

    > 注:
    >
    > 高度なセキュリティ分析 オプションは、プレミアムライセンスの ADC インスタンスにのみ表示されます。

  - トランスポート モード としてログストリームを選択
  - 式はデフォルトで true です
  - 「**OK**」をクリックします



ウェブトランザクション設定を有効にする

- [アナリティクス] >**[設定]** に 移動します。

「設定 」ページが表示されます。

- 「アナリティクスの機能を有効にする」をクリック します。
- [**Web** トランザクション設定**]** で、[すべて] を選択します。

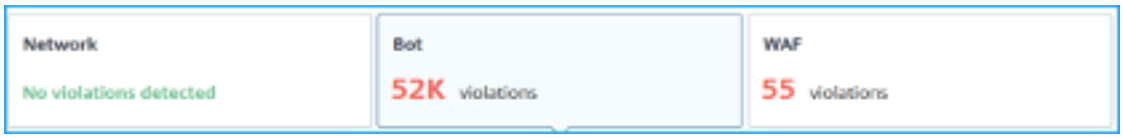
- [**OK**] をクリック します。

セキュリティ違反ダッシュボード

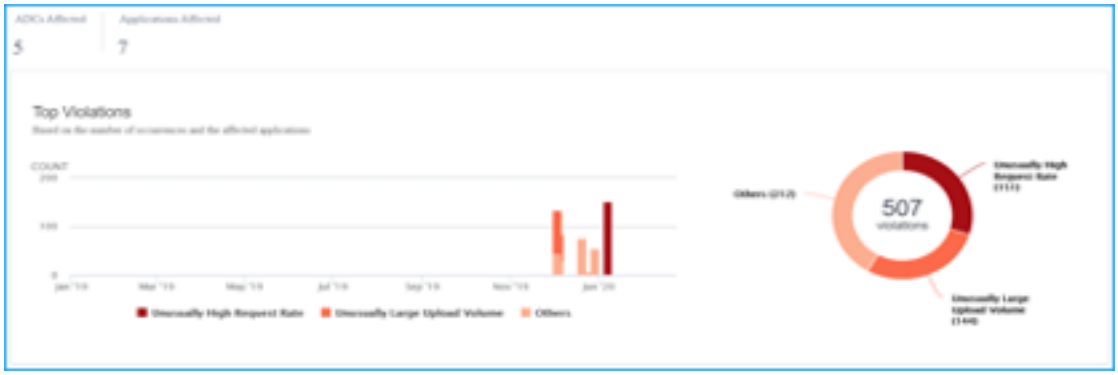
セキュリティ違反ダッシュボードでは、ユーザーは以下を表示できます。

- すべての NetScaler とアプリケーションで発生した違反の合計数。違反の合計は、選択した期間に基づいて表示されます。

- 各カテゴリの下での違反の合計数。



- 影響を受けた ADC の合計、影響を受けたアプリケーションの合計、および影響を受けたアプリケーションの合計数に基づいて、上位レベルの違反数が表示されます。



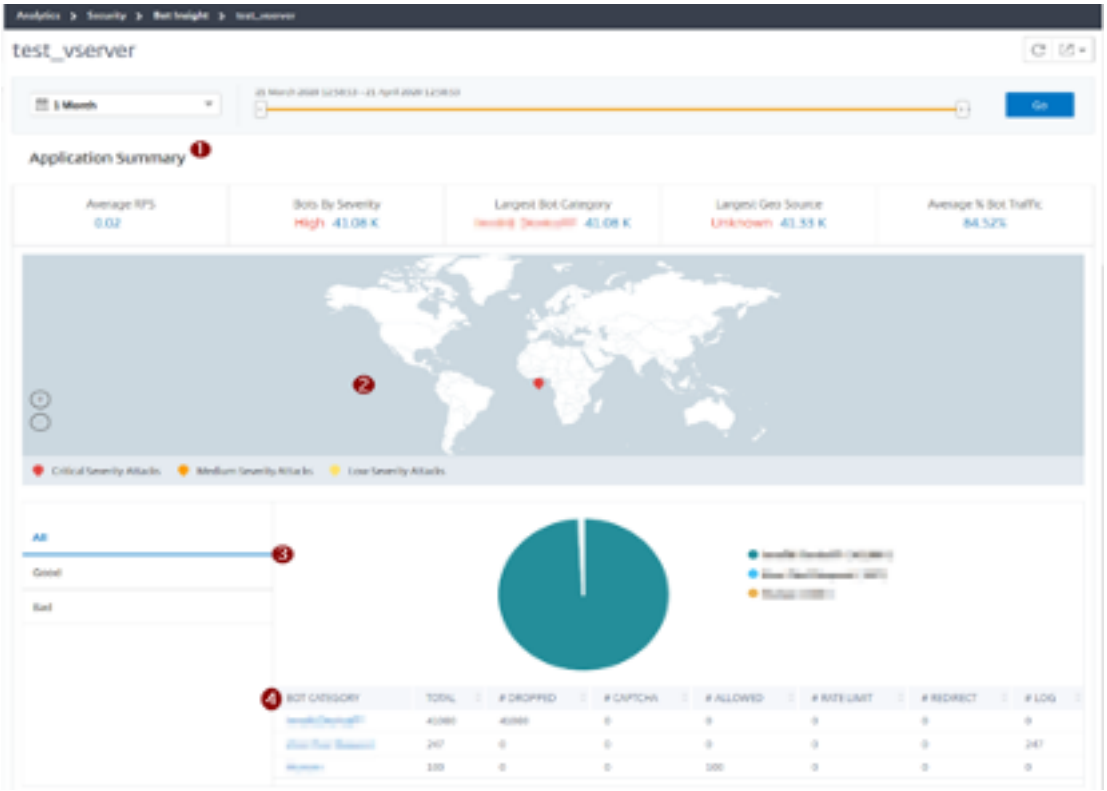
違反の詳細については、「すべての違反」を参照してください。

ボットの洞察

NetScaler でボットインサイトを設定します。詳細については、「Bot」を参照してください。

ボットを表示

仮想サーバーをクリックして、アプリケーションの概要を表示します



1. 次のようなアプリケーション概要の詳細を提供します。

- 平均 **RPS** —仮想サーバーで受信した 1 秒あたりの平均ボットトランザクションリクエスト (RPS) を示します。
- 重要度別のボット—重 大度に基づいて発生したボットトランザクションの数が最も多かったことを示します。重要度は、「緊急」、「高」、「中」、 「低」 に基づいて 分類されます。

たとえば、仮想サーバーに 11770 の高重要度ボットと 1550 の重大度ボットがある場合、NetScaler Console では、「重要度別のボット」に「クリティカル **1.55K**」と表示されます。

- 最大のボットカテゴリ —ボットカテゴリに基づいて発生したボット攻撃の数が最も多いことを示します。

たとえば、仮想サーバーにブロックリストに登録されているボットが 8000、許可リストに登録されているボットが 5000 個、レート制限超過ボットが 10,000 個ある場合、NetScaler Console では、「最大ボットカテゴリ」に「レート制限が **10K** を超えました」と表示されます。

- 最大のジオソース —地域に基づいて発生したボット攻撃の数が最も多いことを示します。

たとえば、仮想サーバーのサンタクララでのボット攻撃が 5000 件、ロンドンでのボット攻撃が 7000 件、バンガロールでのボット攻撃が 9000 件ある場合、NetScalerConsole では最大地理ソースの下にバンガロール **9K** と表示されます。

- 平均ボットトラフィック**%**—人間 のボット比率を示します。

2. マップビュー内の場所に基づいてボット攻撃の重大度を表示します
3. ボット攻撃の種類 (「良好」、「悪い」、「すべて」) を表示します
4. ボット攻撃の合計数と、対応する構成されたアクションを表示します。たとえば、次の設定があるとします。
   - IP アドレスの範囲 (192.140.14.9 ～192.140.14.254) をブロックリストボットとして選択し、これらの IP アドレス範囲のアクションとして [ドロップ] を選択します。
   - IP 範囲（192.140.15.4 から 192.140.15.254）をブロックリストボットとして指定し、これらの IP 範囲のアクションとしてログメッセージを作成するように選択されました

     このシナリオでは、NetScaler コンソールには以下が表示されます:

     - ブロックリストされたボットの総数
     - ドロップされたボットの総数
     - ログに記録されているボットの総数

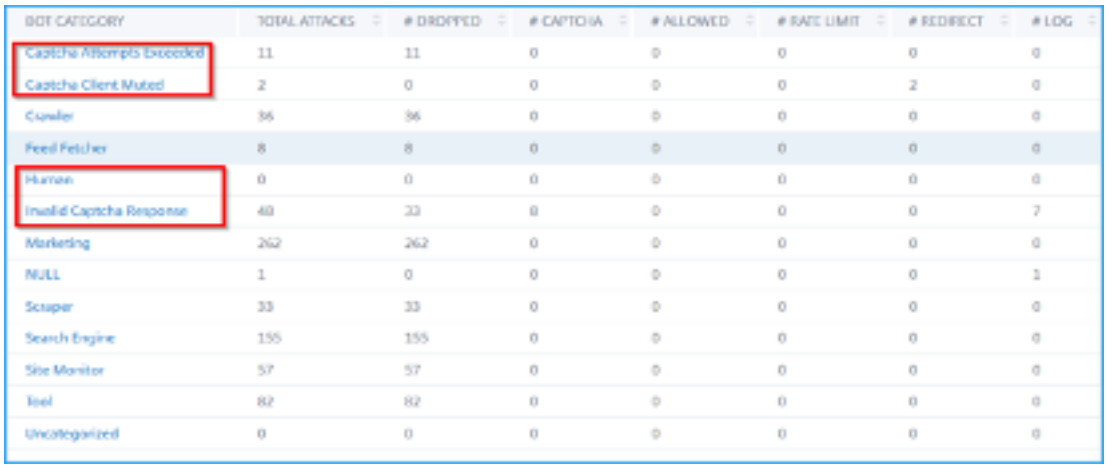
CAPTCHA ボットを表示する

ウェブページでは、CaptCha は、着信トラフィックが人間か自動化されたボットからのものかを識別するように設計されています。NetScaler Console で CAPTCHA アクティビティを表示するには、ユーザーは CAPTCHA を

NetScaler Console インスタンスの IP レピュテーションおよびデバイスフィンガープリント検出技術のボットアクションとして構成する必要があります。詳細については、「ボット管理の設定」を参照してください。

NetScaler コンソールがボットインサイトに表示するキャプチャアクティビティは次のとおりです:

- キャプチャ試行回数超過 —ログイン失敗後に行われた **CAPTCHA** の最大試行回数を示します
- **Captcha client muted**—CAPTCHA チャレンジで以前に不正なボットとして検出されたためにドロップまたはリダイレクトされたクライアント要求の数を示します。
- 人間 —人間のユーザーから実行されたキャプチャエントリを示します
- 無効なキャプチャ応答-NetScaler が CAPTCHA チャレンジを送信したときに、ボットまたは人間から受信した不正な CAPTCHA 応答の数を示します

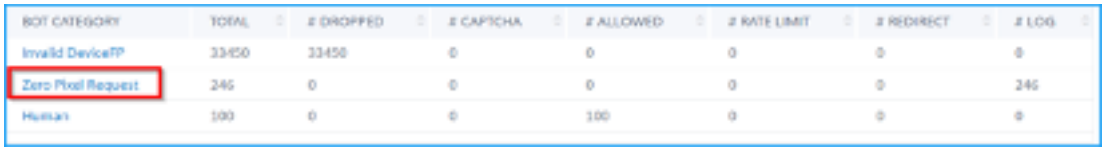


ボットトラップを見る

NetScaler コンソールでボットトラップを表示するには、NetScaler でボットトラップを構成する必要があります。詳細については、「ボット管理の設定」を参照してください。

ボットトラップを識別するために、スクリプトはウェブページで有効になっており、このスクリプトは人間には見えませんが、ボットには見えません。NetScaler Console は、ボットがこのスクリプトにアクセスすると、ボットトラップを識別して報告します。

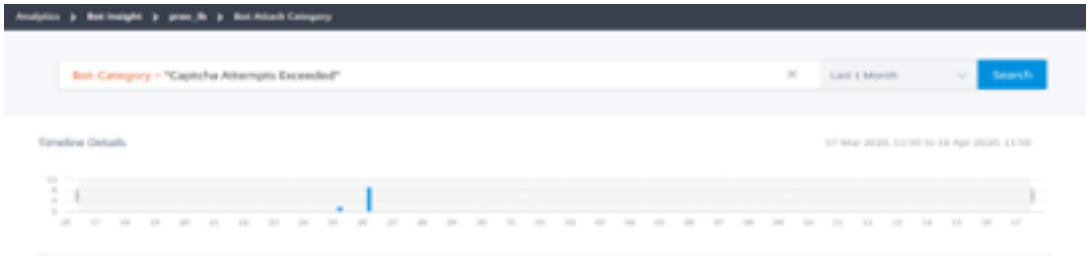
仮想サーバーをクリックし、「ゼロピクセルリクエスト」を選択 します



ボットの詳細の表示

詳細については、[ボットカテゴリ] の [ボット攻撃タイプ] をクリックします。

選択したキャプチャカテゴリの攻撃時間やボット攻撃の総数などの詳細が表示されます。



ユーザーは棒グラフをドラッグして、ボット攻撃で表示する特定の時間範囲を選択することもできます。

ボット攻撃の追加情報を取得するには、をクリックして展開します。

- インスタンス **IP** —NetScaler インスタンスの IP アドレスを示します。
- **Total Bots** —その特定の期間に発生したボット攻撃の合計数を示します。
- **HTTP** リクエスト **URL**： キャプチャレポート用に設定されている URL を示します。
- 国コード —ボット攻撃が発生した国を示します。
- 地域 —ボット攻撃が発生した地域を示します。
- プロファイル名 —構成中にユーザーが提供したプロファイル名を示します。

高度な検索

ユーザーは、検索テキストボックスと期間リストを使用して、ユーザーの要件に従ってボットの詳細を表示することもできます。ユーザーが検索ボックスをクリックすると、検索ボックスに次の検索候補のリストが表示されます。

- インスタンス **IP** —NetScaler インスタンスの IP アドレス。
- クライアント **IP** —クライアント IP アドレス。
- ボットタイプ—「 良い」 または「悪い」などのボットタイプ。
- 重要度 —ボット攻撃の重大度。
- アクション実行 —ドロップ、アクションなし、リダイレクトなど、ボット攻撃後に実行されたアクション。
- ボットカテゴリ—ブロック リスト、許可リスト、フィンガープリントなどのボット攻撃のカテゴリ。カテゴリに基づいて、ユーザーはボットアクションをそのカテゴリに関連付けることができます。
- ボット検出 —ユーザーが NetScaler で構成したボット検出タイプ（ブロックリスト、許可リストなど）。
- 場所 —ボット攻撃が発生した地域/国
- リクエスト **URL** —ボット攻撃の可能性がある URL

ユーザーは、ユーザー検索クエリで演算子を使用して、ユーザー検索の焦点を絞り込むこともできます。たとえば、ユーザーがすべての不良ボットを閲覧したい場合は次のようにします。

- 検索ボックスをクリックし、[ボットタイプ] を選択します
- 検索ボックスをもう一度クリックし、演算子 **=** を選択 します
- 検索ボックスをもう一度クリックし、[**Bad**] を選択します
- [検索] をクリックして結果を表示します

異常に高いリクエスト率

ユーザーは、アプリケーションとの間で送受信されるトラフィックを制御できます。ボット攻撃は、異常に高い要求率を実行する可能性があります。たとえば、ユーザーがアプリケーションを 100 リクエスト/分を許可するように設定し、ユーザーが 350 リクエストを監視した場合、ボット攻撃である可能性があります。

異常に高い要求率インジケーターを使用して、ユーザーはアプリケーションに受け取った異常な要求率を分析できます。

[イベントの詳細] で、ユーザーは以下を表示できます。

- 影響を受けるアプリケーション。複数のアプリケーションが違反の影響を受ける場合、ユーザーはリストからアプリケーションを選択することもできます。
- すべての違反を示すグラフ
- 違反の発生時刻
- 違反の検出メッセージ。受信した要求の合計と、予想された要求よりも受信した過剰な要求の割合を示します。
- 想定されるリクエストレートの許容範囲は、アプリケーションにより異なる

ボットの検出

NetScaler ボット管理システムは、さまざまな手法を使用して受信ボットトラフィックを検出します。この手法は、ボットタイプを検出するための検出ルールとして使用されます。

GUI によるボット管理の設定　ユーザーは、最初にアプライアンスで機能を有効にすることで、NetScaler ボット管理を構成できます。詳細については、「ボット検出」を参照してください。

IP レピュテーション

IP レピュテーションは、不要な要求を送信する IP アドレスを識別するツールです。IP レピュテーションリストを使用すると、レピュテーションの悪い IP アドレスからのリクエストを拒否できます。

GUI を使用して **IP** レピュテーションを設定する　この設定は、ボット IP レピュテーション機能の前提条件です。詳細については、「IP レピュテーション」を参照してください。

Bot シグネチャの自動更新　ボット静的シグニチャ手法では、シグニチャルックアップテーブルと良いボットと不良ボットのリストを使用します。詳しくは、「署名の自動更新」を参照してください。

NetScaler Web App Firewall と OWASP トップ 10—2021

オープンウェブアプリケーションセキュリティプロジェクト（OWAP）は、ウェブアプリケーションセキュリティに関する 2021 年の OWASP トップ 10 を発表しました。このリストは、最も一般的な Web アプリケーションの脆弱性を説明しており、Web セキュリティを評価するための優れた出発点です。このセクションでは、これらの欠陥を軽減するように NetScaler Web App Firewall を構成する方法について説明します。WAF は、NetScaler（プレミアムエディション）およびさまざまなアプライアンスの統合モジュールとして利用できます。

OWASP Top 10 の完全なドキュメントは OWASP Top Tenで入手できます。

OWASP トップ 10 2021	NetScaler Web App Firewall 機能
A 1:2021 壊れたアクセスコントロール	AAA、NetScaler の AAA モジュール内の認証セキュリティ機能、フォーム保護、Cookie 改ざん保護、StartURL、ClosureURL
A 2:2021-暗号化の失敗	クレジットカード保護、セーフコマース、クッキープロキシ、クッキー暗号化
A 3:2021-インジェクション	インジェクション攻撃防止 (SQL または OS コマンドインジェクション、XPath インジェクション、LDAP インジェクションなどのカスタムインジェクション)、シグネチャの自動更新機能
A 5:2021 セキュリティの設定ミス	WSI チェック、XML メッセージ検証、XML SOAP 障害フィルタリングチェックを含むこの保護
A 6:2021-脆弱性と古いコンポーネント	脆弱性スキャンレポート、アプリケーションファイアウォールテンプレート、およびカスタム署名
A 7:2021-識別と認証の失敗	AAA、クッキー改ざん防止、クッキープロキシ、クッキー暗号化、CSRF タギング、SSL の使用
A 8:2021 —ソフトウェアとデータの整合性に関する障害	XML セキュリティチェック、GWT コンテンツタイプ、カスタム署名、JSON および XML 用の Xpath
A 9:2021 —セキュリティロギングと監視の失敗	ユーザー設定可能なカスタムロギング、管理および分析システム

A 1:2021 壊れたアクセスコントロール

認証されたユーザーに許可される内容の制限は、多くの場合、適切に適用されません。攻撃者はこれらの欠陥を悪用して、他のユーザーのアカウントへのアクセス、機密ファイルの閲覧、他のユーザーのデータの変更、アクセス権の変更など、不正な機能やデータにアクセスできます。

NetScaler Web App Firewall 保護

- すべてのアプリケーショントラフィックの認証、認可、および監査をサポートする AAA 機能により、サイト管理者は ADC アプライアンスでアクセス制御を管理できます。
- ADC アプライアンスの AAA モジュール内の認証セキュリティ機能により、アプライアンスは、保護されているサーバ上のどのコンテンツに各ユーザにアクセスを許可するかを検証できます。
- フォームフィールドの一貫性: オブジェクト参照がフォームに非表示フィールドとして保存されている場合、フォームフィールドの一貫性を使用して、これらのフィールドが以降のリクエストで改ざんされていないことを検証できます。
- Cookie プロキシと Cookie の一貫性: クッキー値に保存されているオブジェクト参照は、これらの保護機能で検証できます。
- URL クロージャで URL チェックを開始: 事前定義された URL の許可リストへのユーザーアクセスを許可します。URL クロージャは、ユーザーセッション中に有効な応答に表示されるすべての URL のリストを作成し、そのセッション中に自動的にアクセスを許可します。

A 2:2021-暗号化の失敗

多くの Web アプリケーションと API は、財務、医療、PII などの機密データを適切に保護していません。攻撃者は、クレジットカード詐欺、個人情報の盗難、またはその他の犯罪を行うために、そのような保護が不十分なデータを盗んだり変更したりする可能性があります。機密データは、保存中または転送中の暗号化など、追加の保護なしで侵害される可能性があり、ブラウザと交換する場合は特別な予防措置が必要です。

NetScaler Web App Firewall 保護

- Web Application Firewall は、クレジットカード情報などの機密データの漏洩からアプリケーションを保護します。
- 機密データをセーフコマース保護のセーフオブジェクトとして設定して、露出を防ぐことができます。
- クッキー内の機密データは、クッキープロキシとクッキー暗号化によって保護できます。

A 3:2021-インジェクション

SQL、NoSQL、OS、LDAP インジェクションなどのインジェクションの欠陥は、信頼できないデータがコマンドまたはクエリの一部としてインタープリターに送信されるときに発生します。攻撃者の敵対的なデータは、通訳者を騙して意図しないコマンドを実行させたり、適切な許可なしにデータにアクセスさせたりする可能性があります。

XSS の欠陥は、アプリケーションが適切な検証やエスケープを行わずに信頼できないデータを新しい Web ページに含めたり、HTML または JavaScript を作成できるブラウザ API を使用してユーザー提供のデータで既存の Web ページを更新したりする場合に発生します。XSS を使用すると、攻撃者は被害者のブラウザでスクリプトを実行して、ユーザーセッションをハイジャックしたり、Web サイトを改ざんしたり、ユーザーを悪意のあるサイトにリダイレクトしたりできます。

NetScaler Web App Firewall 保護

- SQL インジェクション防止機能は、一般的なインジェクション攻撃から保護します。カスタム・インジェクション・パターンをアップロードして、XPath や LDAP を含むあらゆる種類のインジェクション攻撃から保護できます。これは HTML ペイロードと XML ペイロードの両方に適用されます。
- シグネチャの自動更新機能は、インジェクションシグネチャを最新の状態に保ちます。
- フィールドフォーマット保護機能を使用すると、管理者は任意のユーザーパラメータを正規表現に制限できます。たとえば、郵便番号フィールドには整数のみを含めることも、5 桁の整数を含めることもできます。
- フォームフィールドの一貫性では、送信された各ユーザーフォームをユーザーセッションフォームの署名と照合して検証し、すべてのフォーム要素の有効性を確認します。
- バッファオーバーフローチェックは、URL、ヘッダー、および Cookie が適切な制限内にあることを確認し、大きなスクリプトやコードを挿入する試みをブロックします。
- XSS 保護は、一般的な XSS 攻撃から保護します。カスタム XSS パターンをアップロードして、許可されているタグと属性のデフォルトリストを変更できます。ADC WAF は、許可された HTML 属性とタグのホワイトリストを使用して、XSS 攻撃を検出します。これは HTML ペイロードと XML ペイロードの両方に適用されます。
- ADC WAF は、OWASP XSS フィルター評価チートシートに記載されているすべての攻撃をブロックします。
- フィールドフォーマットチェックは、攻撃者が XSS 攻撃の可能性がある不適切な Web フォームデータを送信するのを防ぎます。
- フォームフィールドの一貫性。

A 5:2021-セキュリティの設定ミス

セキュリティの設定ミスは、最もよく見られる問題です。これは通常、安全でないデフォルト設定、不完全または即興の構成、オープンクラウドストレージ、誤って構成された HTTP ヘッダー、機密情報を含む詳細なエラーメッセー

ジの結果です。すべてのオペレーティングシステム、フレームワーク、ライブラリ、アプリケーションを安全に構成する必要があるだけでなく、パッチを適用してタイムリーにアップグレードする必要があります。

古いまたは構成が不十分な XML プロセッサの多くは、XML ドキュメント内の外部エンティティ参照を評価します。外部エンティティは、ファイル URI ハンドラー、内部ファイル共有、内部ポートスキャン、リモートコード実行、およびサービス拒否攻撃を使用して内部ファイルを開示するために使用できます。

NetScaler Web App Firewall 保護

- アプリケーションファイアウォールによって生成される PCI-DSS レポートには、ファイアウォールデバイスのセキュリティ設定が記録されます。
- スキャンツールからのレポートは、セキュリティ設定ミスを処理するために ADC WAF シグネチャに変換されます。
- NetScaler Web アプリケーションファイアウォールは、Cenzic、IBM AppScan（エンタープライズおよびスタンダード）、Qualys、TrendMicro、WhiteHat、およびカスタム脆弱性スキャンレポートをサポートしています。
- XML ベースのアプリケーション (クロスサイトスクリプティング、コマンドインジェクションなど) への攻撃に適応できる一般的なアプリケーションの脅威を検出してブロックすることに加えて。
- NetScaler Web App Firewall Web アプリケーションファイアウォールには、XML 固有のセキュリティ保護機能が豊富に含まれています。これには、SOAP メッセージと XML ペイロードを徹底的に検証するスキーマ検証や、悪意のある実行可能ファイルまたはウイルスを含む添付ファイルをブロックする強力な XML 添付ファイルチェックが含まれます。
- 自動トラフィック検査方法は、アクセスを獲得することを目的とした URL やフォームに対する XPath インジェクション攻撃をブロックします。
- NetScaler Web App Firewall Web アプリケーションファイアウォールは、外部エンティティ参照、再帰的拡張、過剰なネスト、長いまたは多くの属性や要素を含む悪意のあるメッセージなど、さまざまな DoS 攻撃も阻止します。

A 6:2021-脆弱で時代遅れのコンポーネント

ライブラリ、フレームワーク、その他のソフトウェアモジュールなどのコンポーネントは、アプリケーションと同じ権限で実行されます。脆弱なコンポーネントが悪用された場合、そのような攻撃は重大なデータ損失またはサーバーの乗っ取りを助長する可能性があります。既知の脆弱性を持つコンポーネントを使用するアプリケーションと API は、アプリケーションの防御を弱体化させ、さまざまな攻撃や影響を引き起こす可能性があります。

NetScaler Web App Firewall 保護

- サードパーティ製コンポーネントを最新の状態にしておくことをお勧めします。

- ADC シグネチャに変換された脆弱性スキャンレポートを使用して、これらのコンポーネントに仮想的にパッチを適用できます。
- これらの脆弱なコンポーネントに使用できるアプリケーションファイアウォールテンプレートを使用できます。
- カスタムシグネチャをファイアウォールにバインドして、これらのコンポーネントを保護できます。

A 7:2021 —認証が壊れています

認証とセッション管理に関連するアプリケーション機能は正しく実装されていないことが多く、攻撃者はパスワード、キー、またはセッショントークンを侵害したり、他の実装の欠陥を悪用して他のユーザーの ID を一時的または永続的に引き継ぐことができます。

NetScaler Web App Firewall 保護

- NetScaler AAA モジュールは、ユーザー認証を実行し、バックエンドアプリケーションにシングルサインオン機能を提供します。これは NetScaler AppExpert ポリシーエンジンに統合され、ユーザーおよびグループの情報に基づくカスタムポリシーを許可します。
- ファイアウォールは、SSL オフロードと URL 変換機能を使用して、サイトが安全なトランスポート層プロトコルを使用して、ネットワークスニッフィングによるセッショントークンの盗用を防ぐこともできます。
- Cookie プロキシと Cookie 暗号化を使用すると、クッキーの盗難を完全に軽減できます。

A 8:2021-ソフトウェアとデータの整合性の失敗

安全でないデシリアライゼーションは、多くの場合、リモートでコードが実行される原因になります。デシリアライゼーションの欠陥によってリモートコードが実行されない場合でも、リプレイ攻撃、インジェクション攻撃、権限昇格攻撃などの攻撃を実行するために使用できます。

NetScaler Web App Firewall 保護

- カスタム署名付きの JSON ペイロード検査。
- XML セキュリティ:XML サービス拒否 (XDoS)、XML SQL および Xpath インジェクション、クロスサイトスクリプティング、フォーマットチェック、WS-I 基本プロファイルコンプライアンス、XML 添付ファイルチェックから保護します。
- フィールドフォーマットチェックとクッキーコンシステンシーとフィールドコンシステンシーを使用できます。

A 9:2021-セキュリティロギングと監視の失敗

不十分なロギングと監視、およびインシデント対応との統合の欠落または非効率的な統合により、攻撃者はシステムをさらに攻撃し、永続性を維持し、より多くのシステムにピボットし、データを改ざん、抽出、または破壊することができます。ほとんどの侵害調査では、侵害を検出する時間は 200 日を超え、通常は内部のプロセスや監視ではなく外部の関係者によって検出されることが示されています。

NetScaler Web App Firewall 保護

- セキュリティチェックまたは署名に対してログアクションが有効になっている場合、生成されるログメッセージには、アプリケーションファイアウォールが Web サイトとアプリケーションを保護している間に監視した要求と応答に関する情報が提供されます。
- アプリケーションファイアウォールは、組み込みの ADC データベースを使用して、悪意のある要求の発信元である IP アドレスに対応する場所を識別する便利さを提供します。
- デフォルトフォーマット (PI) 式を使用すると、ログに含まれる情報を柔軟にカスタマイズできます。また、アプリケーションファイアウォールが生成するログメッセージにキャプチャする特定のデータを追加することもできます。
- アプリケーションファイアウォールは CEF ログをサポートしています。

参照ドキュメント

- HTML SQL インジェクションチェック
- XML SQL インジェクションチェック
- コマンドラインを使用して HTML クロスサイトスクリプティングチェックを設定する
- XML クロスサイトスクリプティングチェック
- コマンドラインによるバッファオーバーフローセキュリティチェックの設定
- 署名オブジェクトの追加または削除
- 署名オブジェクトの設定または変更
- 署名オブジェクトの更新
- Snort 規則の統合
- ボットの検出
- Microsoft Azure で NetScaler VPX インスタンスを展開する

NetScaler Gateway アプライアンスのアドレスプールのイントラネット IP を構成する

February 15, 2024

場合によっては、NetScaler Gateway プラグインを使用して接続するユーザーは、NetScaler Gateway アプライアンス用に一意の IP アドレスが必要です。グループのアドレスプール（IP プーリングとも呼ばれる）を有効にすると、NetScaler Gateway アプライアンスは一意の IP アドレスエイリアスを各ユーザーに割り当てることができます。アドレスプールは、イントラネット IP (IIP) アドレスを使用して構成します。

Azure にデプロイされた NetScaler Gateway アプライアンスでアドレスプールを構成するには、次の 2 ステップの手順に従います。

- アドレスプールで使用されるプライベート IP アドレスを Azure に登録する
- NetScaler Gateway アプライアンスでのアドレスプールの構成

Azure ポータルにプライベート IP アドレスを登録する

Azure では、複数の IP アドレスを持つ NetScaler ADC VPX インスタンスを展開できます。次の 2 つの方法で IP アドレスを VPX インスタンスに追加できます。

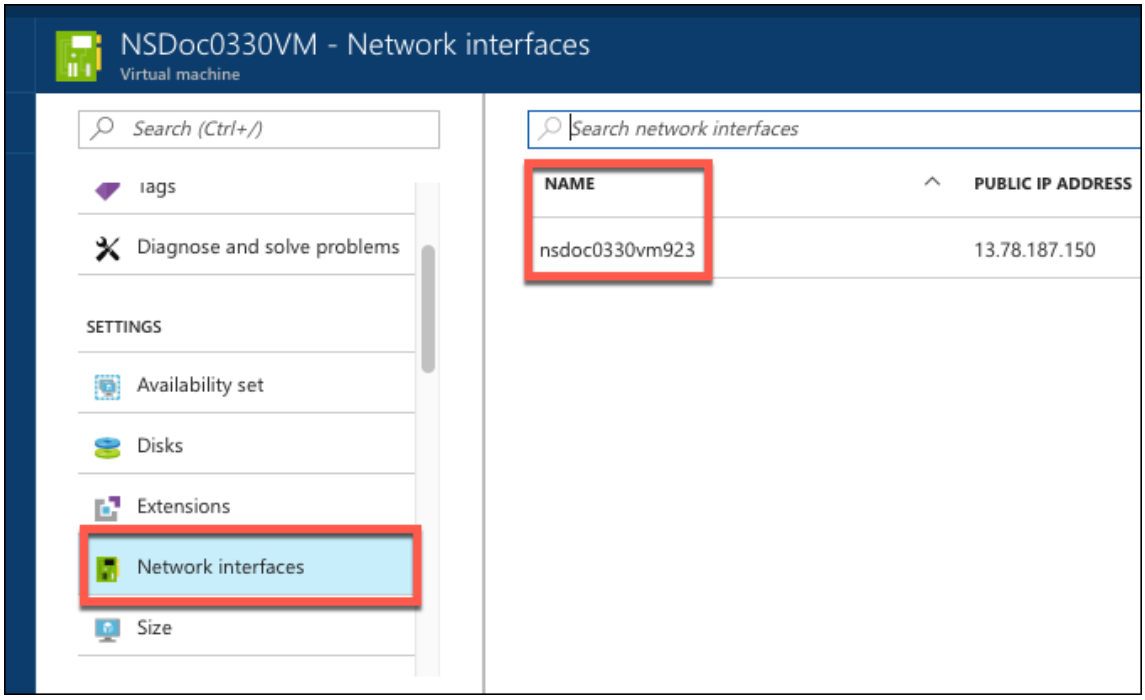
a. VPX インスタンスの Provisioning 中

VPX インスタンスのプロビジョニング中に複数の IP アドレスを追加する方法の詳細については、「NetScaler ADC スタンドアロンインスタンスの複数の IP アドレスを構成する」を参照してください。VPX インスタンスのプロビジョニング中に PowerShell コマンドを使用して IP アドレスを追加するには、PowerShell コマンドを使用してスタンドアロンモードで NetScaler ADC VPX インスタンスの複数の IP アドレスを構成するを参照してください。

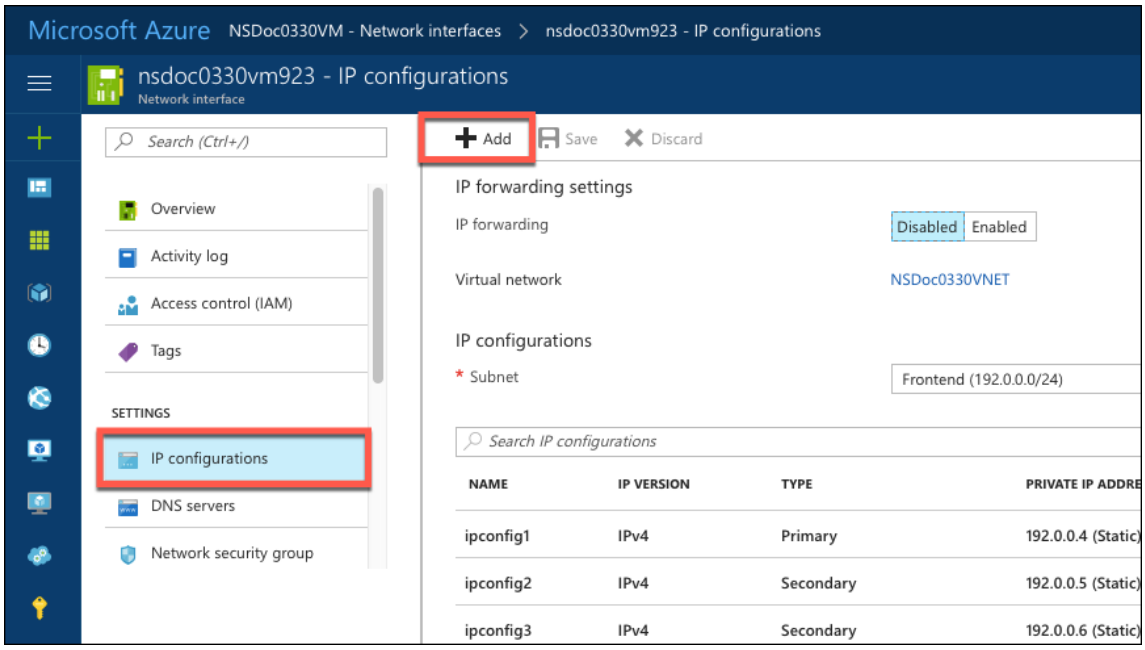
b. VPX インスタンスをプロビジョニング後

VPX インスタンスをプロビジョニングしたら、次の手順に従って Azure ポータルにプライベート IP アドレスを登録します。この IP アドレスは、NetScaler Gateway アプライアンスでアドレスプールとして構成します。

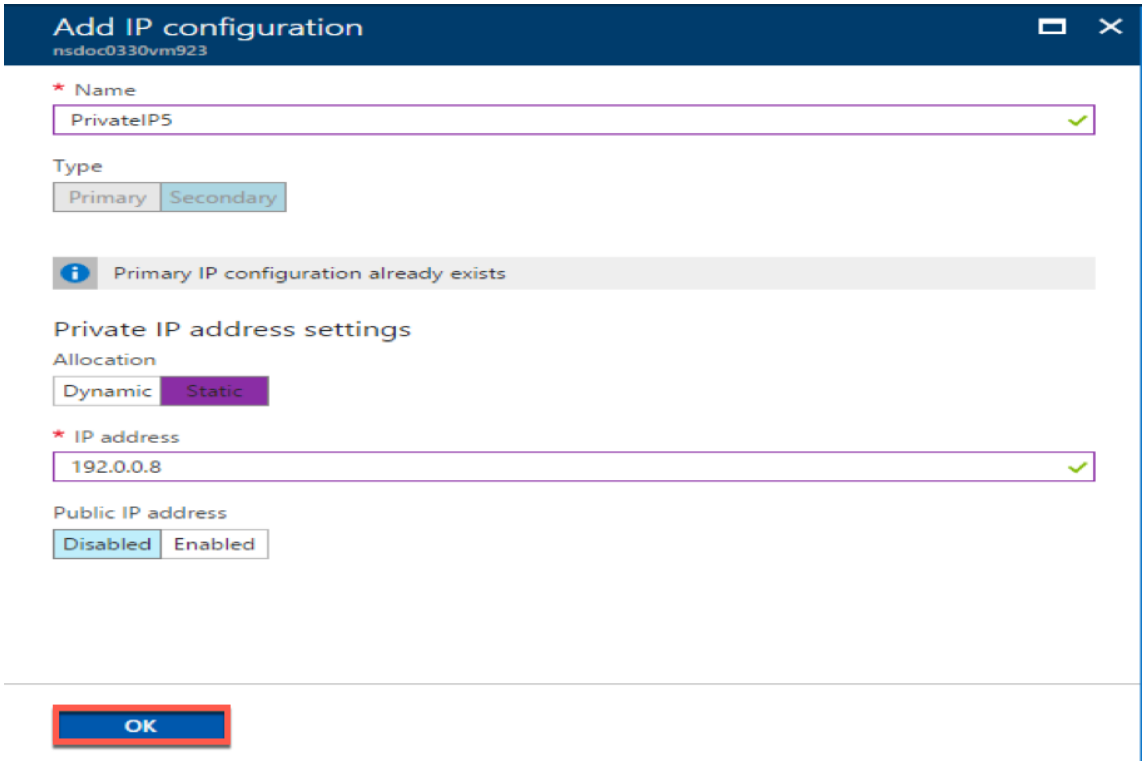
1. Azure Resource Manager（ARM）から、すでに作成されている NetScaler VPX インスタンス > ネットワークインターフェイスに移動します。登録する IIP が属しているサブネットにバインドされているネットワークインターフェイスを選択します。



2. [**IP** 構成] をクリックし、[追加] をクリックします。



3. 以下の例のように必要な詳細を入力し、[**OK**] をクリックします。



NetScaler Gateway アプライアンスでアドレスプールを構成する

NetScaler Gateway でアドレスプールを構成する方法の詳細については、「アドレスプールの構成」を参照してください。

制限:IIP アドレスの範囲をユーザーにバインドすることはできません。アドレスプールで使用されるすべての IIP アドレスを登録する必要があります。

PowerShell コマンドを使用して、**NetScaler VPX** スタンドアロンインスタンスに複数の **IP** アドレスを構成する

August 15, 2023

Azure 環境では、複数の NIC を設定して NetScaler VPX 仮想アプライアンスを展開できます。各 NIC に複数の IP アドレスを設定できます。このセクションでは、PowerShell コマンドを使用して、単一の NIC と複数の IP アドレスを使用して NetScaler ADC VPX インスタンスを展開する方法について説明します。複数 NIC と複数 IP の展開にも同じスクリプトを使用できます。

> 注
>
> このドキュメントでは、IP-config は、個々の NIC に関連付けられている IP アドレス、パブリック IP、プライベート IP のペアを指します。詳細については、Azure の用語のセクションを参照してください 。

使用例

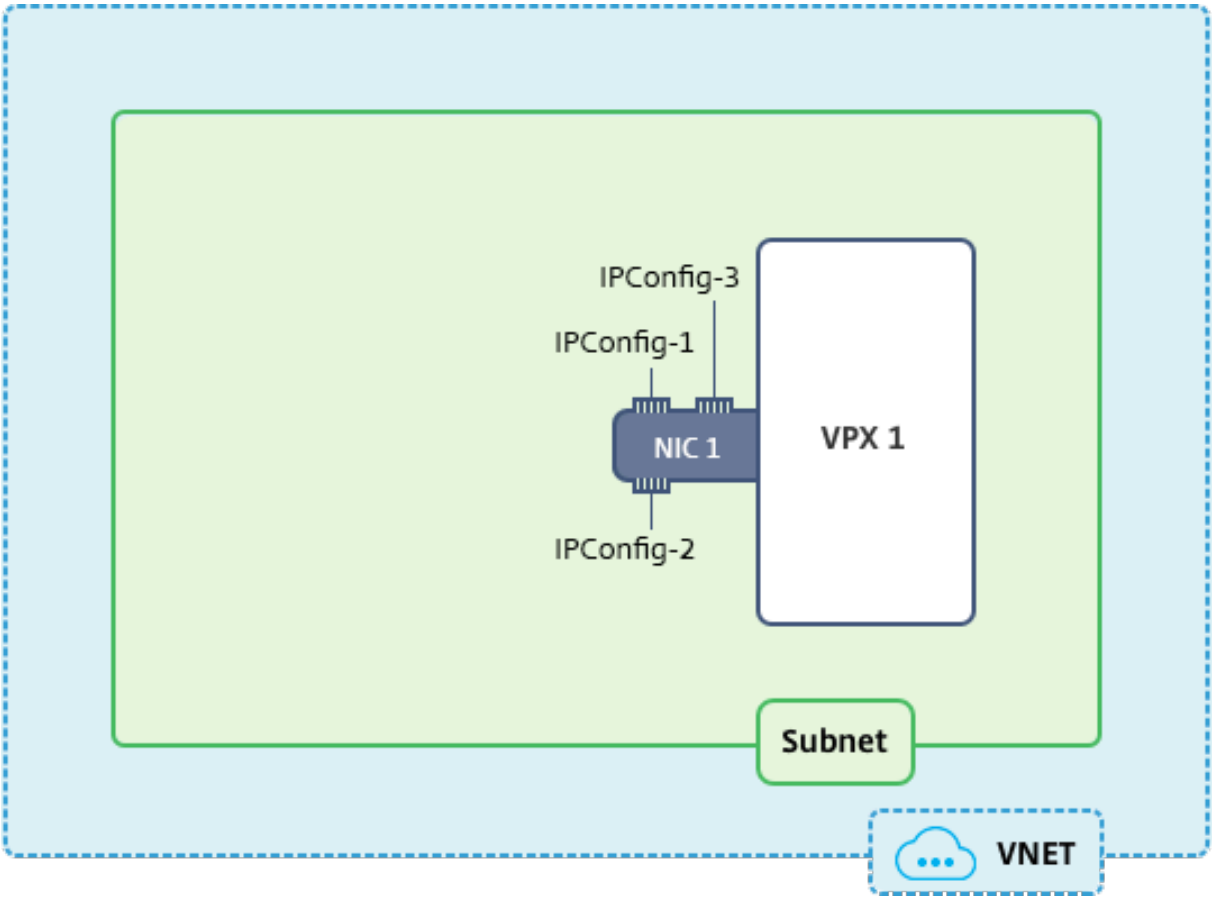
この使用例では、1 つの NIC が仮想ネットワーク（VNET）に接続されています。この NIC には、次の表に示す 3 つの IP 構成が関連付けられています。

IP コンフィグ	関連付けられている
IPConfig-1	静的パブリック IP アドレス; 静的プライベート IP アドレス
IPConfig-2	静的パブリック IP アドレス; 静的プライベートアドレス
IPConfig-3	静的プライベート IP アドレス

> 注
>
> IPConfig-3 は、パブリック IP アドレスに関連付けられていません。

図: トポロジ

次の図はこの使用例を視覚的に示しています。



> 注
>
> マルチ NIC、マルチ IP Azure NetScaler VPX 展開では、プライマリ（最初の）NIC のプライマリ（最初）`IPConfig`に関連付けられたプライベート IP アドレスが、アプライアンスの管理 NSIP アドレスとして自動的に追加されます。`IPConfigs`に関連付けられた残りのプライベート IP アドレスは、要件に応じて`add ns ip`コマンドを使用して、VPX インスタンスに VIP または SNIP として追加する必要があります。

スタンドアロンモードで NetScaler VPX 仮想アプライアンスに対して複数 IP アドレスを構成する場合に必要な手順の概要は次のとおりです。

1. リソースグループの作成
2. ストレージアカウントの作成
3. 可用性セットの作成
4. ネットワークサービスグループの作成
5. 仮想ネットワークの作成
6. パブリック IP アドレスの作成
7. IP 構成の割り当て
8. NIC の作成
9. NetScaler VPX インスタンスの作成
10. NIC 構成のチェック

11. VPX 側の構成のチェック

スクリプト

パラメーター

このドキュメントのこの使用例のサンプルパラメーター設定は、次のとおりです。必要な場合は、異なる設定を使用できます。

$locName=”westcentralus”

$rgName=”Azure-MultiIP”

$nicName1=”VM1-NIC1”

$vNetName=”Azure-MultiIP-vnet”

$vNetAddressRange=”11.6.0.0/16”

$frontEndSubnetName=”frontEndSubnet”

$frontEndSubnetRange=”11.6.1.0/24”

$prmStorageAccountName=”multiipstorage”

$avSetName=”multiip-avSet”

$vmsize= “Standard_DS4_v2”(このパラメータは最大 4 つの NIC を持つ仮想マシンを作成します。)

注**:** VPX インスタンスの最小要件は、2 つの vCPU と 2 GB RAM です。

$ 発行元 =「Citrix」

$offer=”netscalervpx110-6531”（異なる offer を使用できます）

$sku=”netscalerbyol”（offer によって、異なる SKU にすることができます）

$version=”latest”

$pubIPName1=”PIP1”

$pubIPName2=”PIP2”

$domName1=”multiipvpx1”

$domName2=”multiipvpx2”

$vmNamePrefix=”VPXMultiIP”

$osDiskSuffix=”osmultiipalbdiskdb1”

ネットワークセキュリティグループ **(NSG)** 関連情報:

$nsgName=”NSG-MultiIP”

$rule1Name=” Inbound-HTTP”

$rule2Name=” Inbound-HTTPS”

$rule3Name=” Inbound-SSH”

$IpConfigName1=” IPConfig1”

$IPConfigName2=” IPConfig-2”

$IPConfigName3=” IPConfig-3”

1. リソースグループの作成

```
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

2. ストレージアカウントの作成

```
$prmStorageAccount = New-AzureRMStorageAccount -Name $prmStorageAccountName
 -ResourceGroupName $rgName -Type Standard_LRS -Location $locName
```

3. 可用性セットの作成

```
$avSet = New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
 $rgName -Location $locName
```

4. ネットワークセキュリティグループの作成

1. 規則を追加します。トラフィックを処理するポートのネットワークセキュリティグループにルールを追加する必要があります。

   ```
   $rule1=New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -
   Description "Allow HTTP"-Access Allow -Protocol Tcp -Direction
   Inbound -Priority 101 -SourceAddressPrefix Internet -SourcePortRange
    * -DestinationAddressPrefix * -DestinationPortRange 80
   $rule2=New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -
   Description "Allow HTTPS"-Access Allow -Protocol Tcp -Direction
   Inbound -Priority 110 -SourceAddressPrefix Internet -SourcePortRange
    * -DestinationAddressPrefix * -DestinationPortRange 443
   $rule3=New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name
   -Description "Allow SSH"-Access Allow -Protocol Tcp -Direction
   Inbound -Priority 120 -SourceAddressPrefix Internet -SourcePortRange
    * -DestinationAddressPrefix * -DestinationPortRange 22
   ```


2. ネットワークセキュリティグループオブジェクトを作成します。

```
$nsg=New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName
-Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,
$rule3
```

5. 仮想ネットワークの作成

1. サブネットを追加します。

```
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
$frontEndSubnetName -AddressPrefix $frontEndSubnetRange
```

2. 仮想ネットワークオブジェクトを追加します。

```
$vnet=New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
 $rgName -Location $locName -AddressPrefix $vNetAddressRange -
Subnet $frontendSubnet
```

3. サブネットを取得します。

```
$subnetName="frontEndSubnet"
$subnet1=$vnet.Subnets|?{ $_.Name -eq $subnetName }
```

6. パブリック **IP** アドレスの作成

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
 $rgName -DomainNameLabel $domName1 -Location $locName -AllocationMethod
 Static
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
 $rgName -DomainNameLabel $domName2 -Location $locName -AllocationMethod
 Static
```

> 注
>
> 使用する前にドメイン名の可用性をチェックします。
>
> IP アドレスの割り当て方法は動的または静的にできます。

7. IP 構成の割り当て

この使用例では、IP アドレスを割り当てる前に次の点を検討します。

- IPConfig-1 が VPX1 の subnet1 に属していること
- IPConfig-2 が VPX1 の subnet 1 に属していること

- IPConfig-3 が VPX1 の subnet 1 に属していること

> 注
>
> 複数の IP 構成を 1 つの NIC に割り当てるときには、1 つの構成をプライマリとして割り当てる必要があります。

```
$IPAddress1="11.6.1.27"
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress1 -PublicIpAddress $pip1
     -Primary
$IPAddress2="11.6.1.28"
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress2 -PublicIpAddress $pip2
$IPAddress3="11.6.1.29"
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress3 -Primary
```

サブネットの要件に合う有効な IP アドレスを使用して、その可用性をチェックします。

8. NIC の作成

```
$nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName
 $rgName -Location $locName -IpConfiguration $IpConfig1,$IpConfig2,
$IPConfig3 -NetworkSecurityGroupId $nsg.Id
```

9. NetScaler VPX インスタンスの作成

1. 変数を初期化します。

   ```
   $suffixNumber = 1
   $vmName = $vmNamePrefix + $suffixNumber
   ```

2. VM Config オブジェクトを作成します。

   ```
   $vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
   AvailabilitySetId $avSet.Id
   ```

3. 資格情報、OS、イメージを設定します。

   ```
   $cred=Get-Credential -Message "Type the name and password for VPX
    login."
   $vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
   ComputerName $vmName -Credential $cred
   $vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
   $publisher -Offer $offer -Skus $sku -Version $version
   ```


4. NIC を追加します。

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.
Id -Primary
```

> 注
>
> マルチ NIC VPX 展開では、1 つの NIC がプライマリである必要があります。したがって、その NIC を VPX インスタンスに追加するときに「-Primary」を追加する必要があります。

5. OS ディスクを指定して、VM を作成します。

```
$osDiskName=$vmName + "-"+ $osDiskSuffix1
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString()+ "
vhds/"+ $osDiskName + ".vhd"
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -
VhdUri $osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product
$offer -Name $sku
New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
$locName
```

10. NIC 構成のチェック

VPX インスタンスの起動後、次のコマンドを使用して VPX NIC の `IPConfigs` に割り当てられている IP アドレスを確認できます。

```
$nic.IPConfig
```

11. VPX 側の構成のチェック

NetScaler VPX インスタンスが起動すると、`IPconfig` プライマリ NIC のプライマリに関連付けられたプライベート IP アドレスが NSIP アドレスとして追加されます。残りのプライベート IP アドレスは、要件に従って、VIP または SNIP アドレスとして追加する必要があります。次のコマンドを使用します。

```
add nsip <Private IPAddress><netmask> -type VIP/SNIP
```

スタンドアロンモードの NetScaler VPX インスタンスに対して複数 IP アドレスを構成しました。

Azure 展開の追加の PowerShell スクリプト

August 15, 2023

このセクションでは、Azure PowerShell で次の構成を実行できる PowerShell コマンドレットについて説明します。

- NetScaler VPX スタンドアロンインスタンスのプロビジョニング
- Azure 外部ロードバランサーを使用した高可用性セットアップで NetScaler VPX ペアをプロビジョニングします
- Azure 内部ロードバランサーを使用した高可用性セットアップで NetScaler VPX ペアをプロビジョニングします

PowerShell コマンドを使用して実行できる構成については、次のトピックも参照してください。

- PowerShell コマンドを使用して、複数の IP アドレスと NIC を使用した高可用性設定を構成する
- NetScaler VPX インスタンスで GSLB を構成する
- NetScaler のアクティブ/スタンバイ高可用性セットアップで GSLB を構成する
- PowerShell コマンドを使用して、スタンドアロンモードの NetScaler ADC VPX インスタンスで複数の IP アドレスを構成する
- スタンドアロン VPX インスタンス用に複数の Azure VIP を構成する

NetScaler VPX スタンドアロンインスタンスのプロビジョニング

1. リソースグループの作成

   リソースグループには、ソリューションのすべてのリソースを含めることも、グループとして管理するリソースのみを含めることもできます。ここで指定した場所は、そのリソースグループ内のリソースのデフォルトの場所です。ロードバランサーを作成する場合、すべてのコマンドで同じリソースグループを使用してください。

   ```
   $rgName="<resource group name>"
   $locName="<location name, such as West US>
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

   例えば:

   ```
   $rgName = "ARM-VPX"
   $locName = "West US"
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   <!--NeedCopy-->
   ```

2. ストレージアカウントの作成

   ストレージアカウントには、小文字と数字のみを含む一意の名前を選択する必要があります。

   ```
   $saName="<storage account name>"
   $saType="<storage account type>"、1 つ指定してください: Standard_LRS、Standard_GRS、Standard_RAGRS、またはPremium_LRS
   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
   ```


例えば:

```
$saName="vpxstorage"
$saType="Standard_LRS"
New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
    -Type $saType -Location $locName
<!--NeedCopy-->
```

3. アベイラビリティセットの作成

   可用性セットにより、メンテナンス時などのダウンタイム中でも仮想マシンを使用し続けることができます。可用性セットが構成されたロードバランサーでは、アプリケーションをいつでも使用できます。

   ```
   $avName="<availability set name>"

   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
   $rgName -Location $locName
   ```

4. 仮想ネットワークの作成

   以前に作成されたサブネットがない場合、少なくとも 1 つのサブネットを持つ新しい仮想ネットワークを追加します。

   ```
   $FrontendAddressPrefix="10.0.1.0/24"
   $BackendAddressPrefix="10.0.2.0/24"
   $vnetAddressPrefix="10.0.0.0/16"
   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
   frontendSubnet -AddressPrefix $FrontendAddressPrefix
   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
   backendSubnet -AddressPrefix $BackendAddressPrefix
   New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName
   $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
   Subnet $frontendSubnet,$backendSubnet
   ```

   例えば:

   ```
   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       frontendSubnet -AddressPrefix $FrontendAddressPrefix

   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       backendSubnet -AddressPrefix $BackendAddressPrefix

   New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName
       -Location $locName -AddressPrefix $vnetAddressPrefix -Subnet
       $frontendSubnet,$backendSubnet
   <!--NeedCopy-->
   ```

5. **NIC** を作成する

   NIC を作成し、それを NetScaler VPX インスタンスに関連付けます。上記の手順で作成されたフロントエン

ドサブネットは 0 でインデックス付けされ、バックエンドサブネットは 1 でインデックス付けされます。次の 3 つのいずれかの方法で NIC を作成します。

a) パブリック *IP* アドレスを持つ *NIC*

```
$nicName="<name of the NIC of the VM>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
 $rgName -Location $locName -AllocationMethod Dynamic

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
 $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex
].Id -PublicIpAddressId $pip.Id
```

b) パブリック *IP* アドレスと *DNS* ラベルが付けられた *NIC*

```
$nicName="<name of the NIC of the VM>"

$domName="<domain name label>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
 $rgName -DomainNameLabel $domName -Location $locName -AllocationMethod
 Dynamic
```

$domName を割り当てる前に、次のコマンドを使用して、それが利用できるかどうかを確認します。

```
Test-AzureRmDnsAvailability -DomainQualifiedName $domName -
Location $locName

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
 $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex
].Id -PublicIpAddressId $pip.Id
```

例えば:

```
1 $nicName="frontendNIC"
2
3 $domName="vpxazure"
4
5 $pip = New-AzureRmPublicIpAddress -Name $nicName -
     ResourceGroupName $rgName -DomainNameLabel $domName -Location
     $locName -AllocationMethod Dynamic
6
7 $nic = New-AzureRmNetworkInterface -Name $nicName -
     ResourceGroupName $rgName -Location $locName -SubnetId $vnet.
     Subnets[0].Id -PublicIpAddressId $pip.Id
8 <!--NeedCopy-->
```

c) 動的パブリックアドレスと静的プライベート *IP* アドレスを持つ *NIC*

仮想マシンに追加するプライベート（静的）IP アドレスが、指定したサブネットのアドレスと同じ範囲である必要があります。


```
$nicName="<name of the NIC of the VM>"

$staticIP="<available static IP address on the subnet>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
 $rgName -Location $locName -AllocationMethod Dynamic

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
 $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex
].Id -PublicIpAddressId $pip.Id -PrivateIpAddress $staticIP
```

6. 仮想オブジェクトの作成

```
$vmName="<VM name>"

$vmSize="<VM size string>"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
 $rgName

$vm=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId
 $avset.Id
```

7. **NetScaler VPX** イメージを取得

```
$pubName="<Image publisher name>"

$offerName="<Image offer name>"

$skuName="<Image SKU name>"

$cred=Get-Credential -Message "Type the name and password of the
local administrator account."
```

VPX へのログインに使用する資格情報を入力してください

```
$vm=Set-AzureRmVMOperatingSystem -VM $vm -Linux -ComputerName
$vmName -Credential $cred -Verbose

$vm=Set-AzureRmVMSourceImage -VM $vm -PublisherName $pubName -
Offer $offerName -Skus $skuName -Version "latest"

$vm=Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id
```

例えば:

```
$pubName="citrix"
```

次のコマンドを使用すると、Citrix からのすべてのオファーが表示されます。

```
1 Get-AzureRMVMImageOffer -Location $locName -Publisher $pubName |
     Select Offer
2
3 $offerName="netscalervpx110-6531"
```



```
<!--NeedCopy-->
```

次のコマンドは、特定のオファー名について発行元から提供される SKU を知るために使用します。

```
Get-AzureRMVMImageSku -Location $locName -Publisher $pubName -
Offer $offerName | Select Skus
```

8. 仮想マシンの作成

```
$diskName="<name identifier for the disk in Azure storage, such
as OSDisk>"
```

例えば:

```
$diskName="dynamic"

$pubName="citrix"

$offerName="netscalervpx110-6531"

$skuName="netscalerbyol"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName

$osDiskUri=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds/"
    + $diskName  + ".vhd"

$vm=Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri
     -CreateOption fromImage
<!--NeedCopy-->
```

MarketPlace に存在するイメージから VM を作成する場合、次のコマンドを使用して VM プランを指定します。

```
Set-AzureRmVMPlan -VM $vm -Publisher $pubName -Product $offerName
 -Name $skuName
```

```
New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
$vm
```

Azure 外部ロードバランサーを使用した高可用性セットアップで **NetScaler VPX** ペアをプロビジョニングします

Azure のユーザー資格情報を使用して AzureRmAccount にログオンします。

1. リソースグループの作成

   ここで指定した場所は、そのリソースグループ内のリソースのデフォルトの場所です。ロードバランサーを作成する場合、すべてのコマンドで同じリソースグループを使用してください。


```
$rgName="<resource group name>"

$locName="<location name, such as West US>"

New-AzureRmResourceGroup -Name $rgName -Location $locName
```

例えば:

```
1 $rgName = "ARM-LB-NS"
2
3 $locName = "West US"
4
5 New-AzureRmResourceGroup -Name $rgName -Location $locName
6 <!--NeedCopy-->
```

2. ストレージアカウントの作成

   ストレージアカウントには、小文字と数字のみを含む一意の名前を選択する必要があります。

   ```
   $saName="<storage account name>"

   $saType="<storage account type>"
   ```

   で次のいずれかを指定します: Standard_LRS、Standard_GRS、Standard_RAGRS、またはPremium_LRS

   ```
   New-AzureRmStorageAccount -Name $saName -ResourceGroupName
   $rgName -Type $saType -Location $locName
   ```

   例えば:

   ```
   1 $saName="vpxstorage"
   2
   3 $saType="Standard_LRS"
   4
   5 New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
       -Type $saType -Location $locName
   6 <!--NeedCopy-->
   ```

3. アベイラビリティセットの作成

   可用性セットが構成されたロードバランサーでは、アプリケーションをいつでも使用できます。

   ```
   $avName="<availability set name>"

   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
   $rgName -Location $locName
   ```

4. 仮想ネットワークの作成

   以前に作成されたサブネットがない場合、少なくとも 1 つのサブネットを持つ新しい仮想ネットワークを追加します。

   ```
   1 $vnetName = "LBVnet"
   2
   3 $FrontendAddressPrefix="10.0.1.0/24"
   ```



```
4
5  $BackendAddressPrefix="10.0.2.0/24"
6
7  $vnetAddressPrefix="10.0.0.0/16"
8
9  $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       frontendSubnet -AddressPrefix $FrontendAddressPrefix
10
11 $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       backendSubnet -AddressPrefix $BackendAddressPrefix
12
13 $vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
        $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
       Subnet $frontendSubnet,$backendSubnet
14 <!--NeedCopy-->
```

注: 必要に応じて AddressPrefix パラメータ値を選択してください。

フロントエンドおよびバックエンドサブネットを、前の手順で作成した仮想ネットワークに割り当てます。

フロントエンドサブネットが配列 VNet の最初の要素である場合、subnetId は $vnet.subnets [0] .Id でなければなりません。

フロントエンドサブネットが配列の 2 番目の要素である場合、subnetId は $vnet.subnets [1] .Id というようにする必要があります。

5. フロントエンド **IP** アドレスを構成し、バックエンドアドレスプールを作成する

   受信ロードバランサーネットワークトラフィック用のフロントエンド IP アドレスを構成し、負荷分散トラフィックを受信するバックエンドアドレスプールを作成します。

```
1  $pubName="PublicIp1"
2
3  $publicIP1 = New-AzureRmPublicIpAddress -Name $pubName -
       ResourceGroupName $rgName -Location $locName -AllocationMethod
       Static -DomainNameLabel nsvpx
4  <!--NeedCopy-->
```

   注:DomainNameLabel の値が使用可能かどうかを確認してください。

```
1  $FIPName = "ELBFIP"
2
3  $frontendIP1 = New-AzureRmLoadBalancerFrontendIpConfig -Name
       $FIPName -PublicIpAddress $publicIP1
4
5  $BEPool = "LB-backend-Pool"
6
7  $beaddresspool1= New-AzureRmLoadBalancerBackendAddressPoolConfig -
       Name $BEPool
8  <!--NeedCopy-->
```

6. ヘルスプローブの作成

ポート 9000、間隔 5 秒で TCP ヘルププローブを作成します。

```
$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name
    HealthProbe -Protocol Tcp -Port 9000 -IntervalInSeconds 5 -
    ProbeCount 2
<!--NeedCopy-->
```

7. 負荷分散ルールを作成する

   負荷分散するサービスごとに LB ルールを作成します。

   例えば:

   次の例を使用して、HTTP サービスの負荷分散を行うことができます。

```
$lbrule1 = New-AzureRmLoadBalancerRuleConfig -Name "HTTP-LB" -
    FrontendIpConfiguration $frontendIP1 -BackendAddressPool
    $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort
     80 -BackendPort 80
<!--NeedCopy-->
```

8. インバウンド **NAT** ルールの作成

   負荷分散していないサービスに対する NAT 規則を作成します。

   たとえば、NetScaler VPX インスタンスへの SSH アクセスを作成する場合などです。

   注:2 つの NAT ルールでプロトコル frontendPort-backendPort トリプレットが同じであってはなりません。

```
$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name SSH1     -FrontendIpConfiguration $frontendIP1 -Protocol
    TCP -FrontendPort 22 -BackendPort 22

$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name SSH2 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -
    FrontendPort 10022 -BackendPort 22
<!--NeedCopy-->
```

9. ロードバランサーエンティティの作成

   すべてのオブジェクト（NAT 規則、ロードバランサー規則、プローブ構成）を一度に追加してロードバランサーを作成します。

```
$lbName="ELB"

$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name
    $lbName -Location $locName -InboundNatRule $inboundNATRule1,
    $inboundNATRule2 -FrontendIpConfiguration $frontendIP1 -
    LoadBalancingRule $lbrule1 -BackendAddressPool $beAddressPool1
    -Probe $healthProbe
<!--NeedCopy-->
```


10. **NIC** を作成する

    2 つの NIC を作成し、各 NIC を各 VPX インスタンスに関連付けます

    a）NIC1 を VPX1 に

    例えば:

```
$nicName="NIC1"

$lbName="ELB"

$bePoolIndex=0

* Rule indexes starts from 0.

$natRuleIndex=0

$subnetIndex=0

* Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName
    $rgName

$nic1=New-AzureRmNetworkInterface -Name $nicName -
    ResourceGroupName $rgName -Location $locName -Subnet $vnet.
    Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.
    BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule
    $lb.InboundNatRules[$natRuleIndex]
<!--NeedCopy-->
```

    b）NIC2 を VPX2 に

    例えば:

```
$nicName="NIC2"

$lbName="ELB"

$bePoolIndex=0

$natRuleIndex=1

* Second Inbound NAT (SSH) rule we need to use

`$subnetIndex=0

* Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName
    $rgName

$nic2=New-AzureRmNetworkInterface -Name $nicName -
```



```
        ResourceGroupName $rgName -Location $locName -Subnet $vnet.
        Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.
        BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule
        $lb.InboundNatRules[$natRuleIndex]
<!--NeedCopy-->
```

11. **NetScaler VPX** インスタンスの作成

    2 つの NetScaler VPX インスタンスを、同じリソースグループおよび可用性セットの一部として作成し、外部ロードバランサーに割り当てます。

    a）NetScaler VPX インスタンス 1

    例えば:

```
$vmName="VPX1"

$vmSize="Standard_A3"

$pubName="citrix"

$offerName="netscalervpx110-6531"

$skuName="netscalerbyol"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
    $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avset.Id

$cred=Get-Credential -Message "Type Credentials which will be used
    to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName
    $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -
    Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $nic1.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/
    " + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri
    $osDiskUri1 -CreateOption fromImage

```



```
Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName
    -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
   $vm1
<!--NeedCopy-->
```

b）NetScaler VPX インスタンス 2

例えば:

```
$vmName="VPX2"

$vmSize="Standard_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
    $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
   AvailabilitySetId $avset.Id

$cred=Get-Credential -Message " Type Credentials which will be
   used to login to VPX instance "

$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName
   $vmName -Credential $cred -Verbose

$vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -
   Offer $offerName -Skus $skuName -Version "latest"

$vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $nic2.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
   Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/
   " + $diskName  + ".vhd"

$vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri
   $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName
    -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
   $vm2
<!--NeedCopy-->
```

12. 仮想マシンを構成する

両方の NetScaler VPX インスタンスが開始された場合、SSH プロトコル経由で両方の VPX インスタンスに

接続して仮想マシンを構成します。

a) アクティブ-アクティブ: 両方の NetScaler VPX インスタンスのコマンドラインで同じ構成コマンドセットを実行します。

b) アクティブ/パッシブ: このコマンドを両方の NetScaler VPX インスタンスのコマンドラインで実行します。

```
add ha node #nodeID <nsip of other NetScaler VPX>
```

アクティブ-パッシブモードでは、プライマリノードでのみ構成コマンドを実行します。

Azure 内部ロードバランサーを使用した高可用性セットアップで **NetScaler VPX** ペアをプロビジョニングします

Azure のユーザー資格情報を使用して AzureRmAccount にログオンします。

1. リソースグループの作成

   ここで指定した場所は、そのリソースグループ内のリソースのデフォルトの場所です。ロードバランサーを作成する場合、すべてのコマンドで同じリソースグループを使用してください。

   ```
   $rgName="\<resource group name\>"
   ```

   ```
   $locName="\<location name, such as West US\>"
   ```

   ```
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

   例えば:

   ```
   $rgName = "ARM-LB-NS"

   $locName = "West US"

   New-AzureRmResourceGroup -Name $rgName -Location $locName
   <!--NeedCopy-->
   ```

2. ストレージアカウントの作成

   ストレージアカウントには、小文字と数字のみを含む一意の名前を選択する必要があります。

   ```
   $saName="<storage account name>"
   ```

   ```
   $saType="<storage account type>"
   ```

   で次のいずれかを指定します: Standard_LRS、Standard_GRS、Standard_RAGRS、またはPremium_LRS

   ```
   New-AzureRmStorageAccount -Name $saName -ResourceGroupName
   $rgName -Type $saType -Location $locName
   ```

   例えば:


```
$saName="vpxstorage"

$saType="Standard_LRS"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
    -Type $saType -Location $locName
<!--NeedCopy-->
```

3. アベイラビリティセットの作成

   可用性セットが構成されたロードバランサーでは、アプリケーションをいつでも使用できます。

   ```
   $avName="<availability set name>"

   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
   $rgName -Location $locName
   ```

4. 仮想ネットワークの作成

   以前に作成されたサブネットがない場合、少なくとも 1 つのサブネットを持つ新しい仮想ネットワークを追加します。

   ```
   $vnetName = "LBVnet"

   $vnetAddressPrefix="10.0.0.0/16"

   $FrontendAddressPrefix="10.0.1.0/24"

   $BackendAddressPrefix="10.0.2.0/24"

   $vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
       $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
      Subnet $frontendSubnet,$backendSubnet`

   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
      frontendSubnet -AddressPrefix $FrontendAddressPrefix

   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
      backendSubnet -AddressPrefix $BackendAddressPrefix
   <!--NeedCopy-->
   ```

   注: 必要に応じて AddressPrefix パラメータ値を選択してください。

   フロントエンドおよびバックエンドサブネットを、前の手順で作成した仮想ネットワークに割り当てます。

   フロントエンドサブネットが配列 VNet の最初の要素である場合、subnetId は $vnet.subnets [0] .Id でなければなりません。

   フロントエンドサブネットが配列の 2 番目の要素である場合、subnetId は $vnet.subnets [1] .Id というようにする必要があります。

5. バックエンドアドレスプールの作成


```
$beaddresspool= New-AzureRmLoadBalancerBackendAddressPoolConfig -
Name "LB-backend"
```

6. **NAT** ルールを作成する

   負荷分散していないサービスに対する NAT 規則を作成します。

```
$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name "Inboundnatrule1" -FrontendIpConfiguration $frontendIP -
    Protocol TCP -FrontendPort 3441 -BackendPort 3389

$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name "RDP2" -FrontendIpConfiguration $frontendIP -Protocol TCP
    -FrontendPort 3442 -BackendPort 3389
<!--NeedCopy-->
```

   要件に応じて、フロントエンドポートおよびバックエンドポートを使用します。

7. ヘルスプローブの作成

   ポート 9000、間隔 5 秒で TCP ヘルププローブを作成します。

```
$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name "
    HealthProbe"  " -Protocol tcp -Port 9000 -IntervalInSeconds 5 -
    ProbeCount 2
<!--NeedCopy-->
```

8. 負荷分散ルールを作成する

   負荷分散するサービスごとに LB ルールを作成します。

   次に例を示します:

   次の例を使用して、HTTP サービスの負荷分散を行うことができます。

```
$lbrule = New-AzureRmLoadBalancerRuleConfig -Name "lbrule1" -
    FrontendIpConfiguration $frontendIP -BackendAddressPool
    $beAddressPool -Probe $healthProbe -Protocol Tcp -FrontendPort
    80 -BackendPort 80
<!--NeedCopy-->
```

   要件に応じて、フロントエンドポートおよびバックエンドポートを使用します。

9. ロードバランサーエンティティの作成

   すべてのオブジェクト（NAT 規則、ロードバランサー規則、プローブ構成）を一度に追加してロードバランサーを作成します。

```
$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgname -Name
    "InternalLB" -Location $locName -FrontendIpConfiguration
    $frontendIP -InboundNatRule $inboundNATRule1,$inboundNatRule2 -
    LoadBalancingRule $lbrule -BackendAddressPool $beAddressPool -
    Probe $healthProbe
<!--NeedCopy-->
```


10. **NIC** を作成する

    2 つの NIC を作成し、各 NIC を各 NetScaler VPX インスタンスに関連付けます

```
$backendnic1= New-AzureRmNetworkInterface -ResourceGroupName
    $rgName -Name lb-nic1-be -Location $locName -PrivateIpAddress
    10.0.2.6 -Subnet $backendSubnet -LoadBalancerBackendAddressPool
     $nrplb.BackendAddressPools[0] -LoadBalancerInboundNatRule
    $nrplb.InboundNatRules[0]
<!--NeedCopy-->
```

    この NIC は NetScaler VPX 1 用です。プライベート IP は、追加されたサブネットと同じサブネット内に存在する必要があります。

```
$backendnic2= New-AzureRmNetworkInterface -ResourceGroupName
    $rgName -Name lb-nic2-be -Location $locName -PrivateIpAddress
    10.0.2.7 -Subnet $backendSubnet -LoadBalancerBackendAddressPool
     $nrplb.BackendAddressPools[0] -LoadBalancerInboundNatRule
    $nrplb.InboundNatRules[1].
<!--NeedCopy-->
```

    この NIC は NetScaler ADC VPX 用です 2. `Private IPAddress`パラメーターには、要件に応じて任意のプライベート IP を設定できます。

11. **NetScaler VPX** インスタンスの作成

    同じリソースグループと可用性セットの一部である 2 つの VPX インスタンスを作成し、それを内部ロードバランサーにアタッチします。

    a）NetScaler VPX インスタンス 1

    例えば:

```
$vmName="VPX1"

$vmSize="Standard_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
     $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avset.Id

$cred=Get-Credential -Message "Type Credentials which will be used
     to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName
    $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -
    Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $backendnic1.Id
```



```
16
17 $diskName="dynamic"
18
19 $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
       Name $saName
20
21 $osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/
       " + $diskName  + ".vhd"
22
23 $vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri
       $osDiskUri1 -CreateOption fromImage
24
25 Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName
        -Name $skuName
26
27 New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
       $vm1
28 <!--NeedCopy-->
```

b）NetScaler VPX インスタンス 2

例えば:

```
1  $vmName="VPX2"
2
3  $vmSize="Standard_A3"
4
5  $avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
        $rgName
6
7  $vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
       AvailabilitySetId $avset.Id
8
9  $cred=Get-Credential -Message " Type Credentials which will be
       used to login to VPX instance "
10
11 $vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName
       $vmName -Credential $cred -Verbose
12
13 $vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -
       Offer $offerName -Skus $skuName -Version "latest"
14
15 $vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $backendnic2.Id
16
17 $diskName="dynamic"
18
19 $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
       Name $saName
20
21 $osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/
       " + $diskName  + ".vhd"
22
23 $vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri
       $osDiskUri1 -CreateOption fromImage
```



```
24
25 Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName
       -Name $skuName
26
27 New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
      $vm2
28 <!--NeedCopy-->
```

12. 仮想マシンを構成する

    両方の NetScaler VPX インスタンスが開始された場合、SSH プロトコル経由で両方の VPX インスタンスに接続して仮想マシンを構成します。

    a) アクティブ-アクティブ: 両方の NetScaler VPX インスタンスのコマンドラインで同じ構成コマンドセットを実行します。

    b) アクティブ/パッシブ: このコマンドを両方の NetScaler VPX インスタンスのコマンドラインで実行します。

    ```
    add ha node #nodeID <nsip of other NetScaler VPX>
    ```

    アクティブ-パッシブモードでは、プライマリノードでのみ構成コマンドを実行します。

Azure に関するよくある質問

August 15, 2023

- **Azure Marketplace** からインストールされた **NetScaler VPX** インスタンスのアップグレード手順は、オンプレミスアップグレード手順とは異なりますか?

  いいえ。標準の NetScaler VPX アップグレード手順を使用して、Microsoft Azure クラウド内の NetScaler VPX インスタンスを NetScaler VPX リリース 11.1 以降にアップグレードできます。GUI または CLI の手順を使用してアップグレードできます。新規インストールの場合は、Microsoft Azure クラウド用の NetScaler ADC VPX イメージを使用します。

  NetScaler VPX アップグレードビルドをダウンロードするには、「**NetScaler** ダウンロード」**>** 「****NetScaler** ファームウェア」に移動します。**

- **Azure** でホストされている **NetScaler ADC VPX** インスタンスで観察される **MAC** 移動とインターフェイスミュートを修正するにはどうすればよいですか?

  Azure マルチ NIC 環境では、デフォルトでは、すべてのデータインターフェイスに MAC 移動とインターフェイスのミュートが表示されることがあります。Azure 環境で MAC が移動したりインターフェイスがミュートされたりしないように、NetScaler VPX インスタンスのデータインターフェイス（タグなし）ごとに VLAN を作成し、NIC のプライマリ IP を Azure にバインドすることを Citrix では推奨しています。

  詳細については、CTX224626 の記事を参照してください。

Google Cloud Platform への NetScaler ADC VPX インスタンスのデプロイ

February 15, 2024

NetScaler VPX インスタンスを Google Cloud Platform（GCP）にデプロイできます。GCP の VPX インスタンスを使用すると、GCP クラウドコンピューティング機能を活用し、ビジネスニーズに合わせて Citrix の負荷分散機能とトラフィック管理機能を使用できます。VPX インスタンスを GCP にスタンドアロンインスタンスとしてデプロイできます。シングル NIC 構成とマルチ NIC 構成の両方がサポートされています。

サポートされる機能

Premium、Advanced、Standard のすべての機能は、使用されているライセンス/バージョンタイプに基づいて GCP でサポートされます。

制限事項

- IPv6 はサポートされていません。

ハードウェア要件

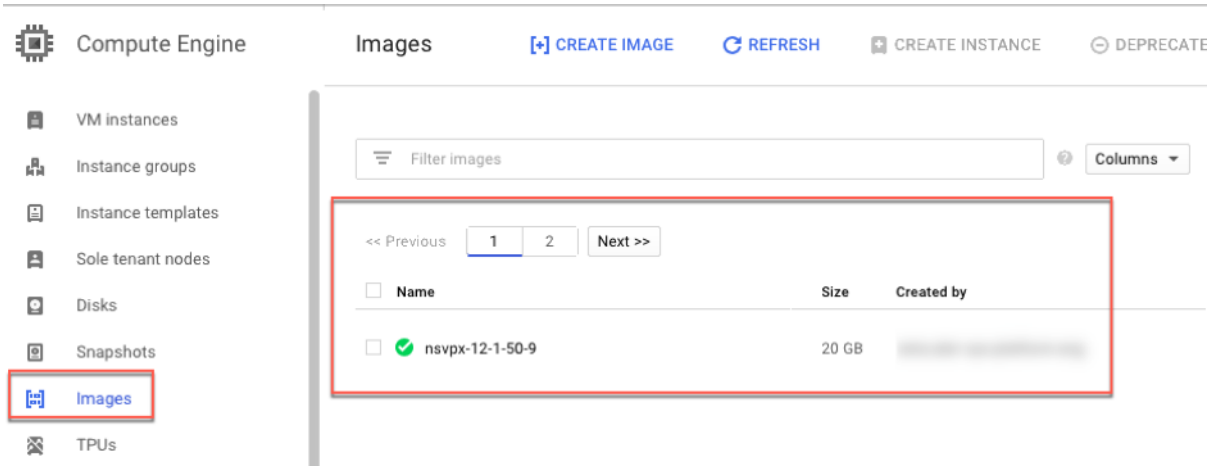
GCP の VPX インスタンスには、最低 2 つの vCPU と 4 GB の RAM が必要です。

前提条件

1. デバイスに「gcloud」ユーティリティをインストールします。このユーティリティは、次のリンクで見つけることができます。https://cloud.google.com/sdk/install
2. NSVPX-GCP イメージは NetScaler サイトからダウンロードします。
3. 「https://cloud.google.com/storage/docs/uploading-objects」の手順に従って、ファイル（nsvpx-GCP-12.1-50.9_nc_64.tar.GZ）を Google のストレージバケットにアップロードします。
4. gcloud ユーティリティで次のコマンドを実行して、イメージを作成します。

```
gcloud compute images create <IMAGE_NAME> --source-uri=gs://<
    STORAGE_BUCKET_NAME>/<FILE_NAME>.tar.gz --guest-os-features=
    MULTI_IP_SUBNET
<!--NeedCopy-->
```

画像が作成されるまでに少し時間がかかる場合があります。イメージが作成されると、GCP コンソールの [コンピュート] > [コンピュートエンジン] に表示されます。



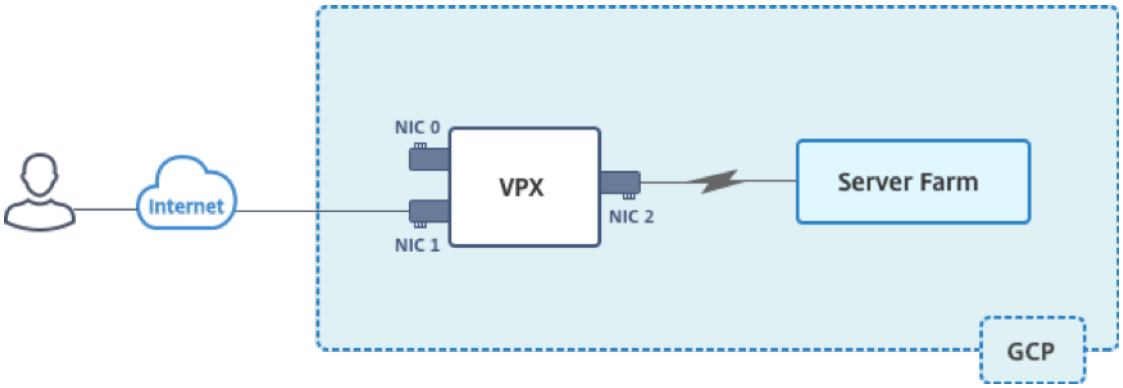
注意事項

デプロイを開始する前に、次の GCP 固有の点を考慮してください。

- インスタンスの作成後は、ネットワークインターフェイスの追加や削除はできません。
- マルチ NIC デプロイの場合は、NIC ごとに個別の VPC ネットワークを作成します。1 つの NIC を関連付けることができるネットワークは 1 つだけです。
- シングル NIC インスタンスの場合、GCP コンソールはデフォルトでネットワークを作成します。
- 2 つ以上のネットワークインターフェースを持つインスタンスには、最低 4 つの vCPU が必要です。
- IP 転送が必要な場合は、インスタンスの作成と NIC の設定中に IP 転送を有効にする必要があります。

シナリオ：マルチ **NIC**、マルチ **IP** スタンドアロン **VPX** インスタンスをデプロイする

このシナリオでは、NetScaler VPX スタンドアロンインスタンスを GCP にデプロイする方法を示しています。このシナリオでは、多数の NIC を持つスタンドアロン VPX インスタンスを作成します。インスタンスはバックエンドサーバー (サーバーファーム) と通信します。



次の目的に応える NIC を 3 つ作成します。

NIC	目的	VPC ネットワークに関連付けられている
NIC 0	管理トラフィック（NetScaler IP）にサービスを提供する	管理ネットワーク
NIC 1	クライアント側のトラフィック（VIP）をサービスする	クライアントネットワーク
NIC 2	バックエンド・サーバ（SNIP）との通信	バックエンドサーバーネットワーク

次の間の必要な通信ルートを設定します。

- VPX インスタンスとバックエンドサーバー。
- VPX インスタンスとパブリックインターネット上の外部ホスト。

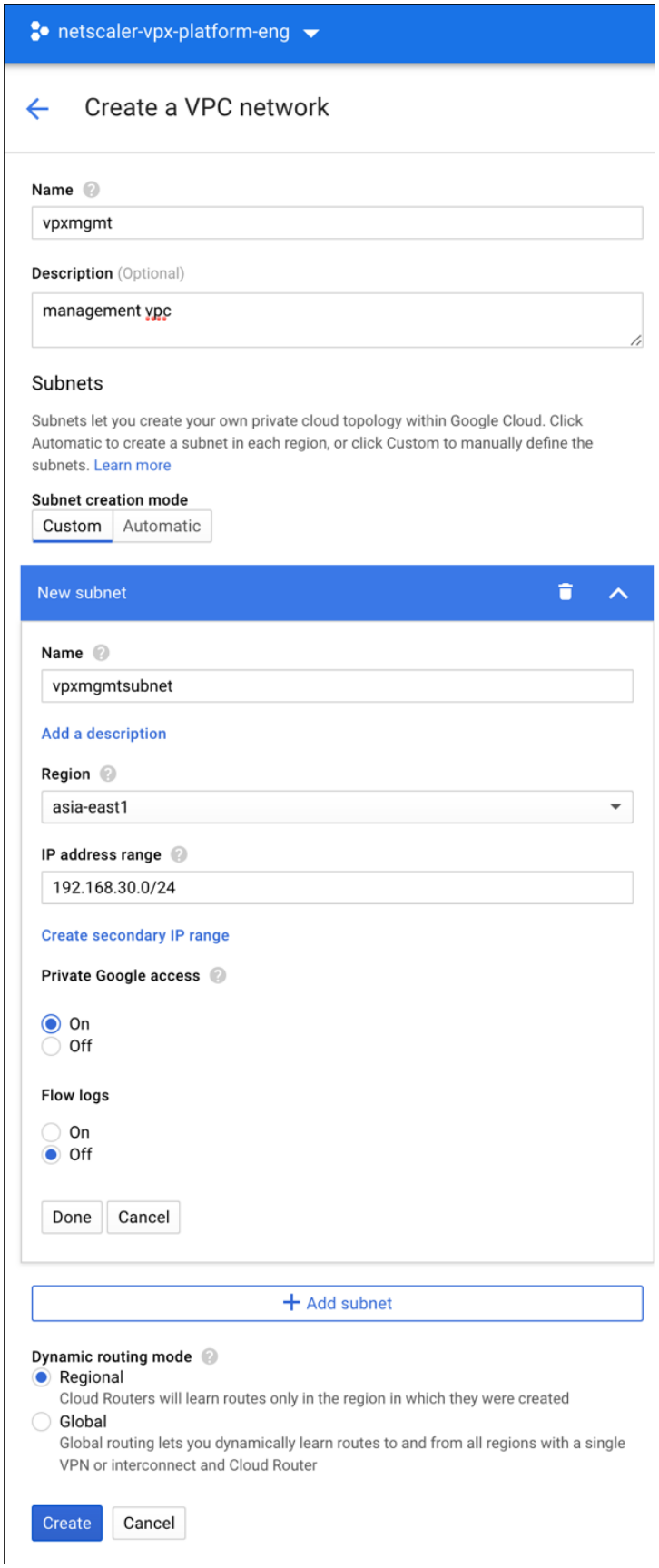
導入手順の概要

1. 3 つの異なる NIC に対して 3 つの VPC ネットワークを作成します。
2. ポート 22、80、443 のファイアウォールルールを作成します。
3. 3 つの NIC でインスタンスを作成する

> 注:
>
> VPC ネットワークを作成したのと同じリージョンにインスタンスを作成します。

手順 **1. VPC** ネットワークを作成します。

管理 NIC、クライアント NIC、およびサーバー NIC に関連付けられた 3 つの VPC ネットワークを作成します。VPC ネットワークを作成するには、**Google** コンソール **>** ネットワーク **> VPC** ネットワーク **> VPC** ネットワークの作成にログオンします。スクリーン・キャプチャに示されている必須フィールドに入力し、「作成」をクリックします。

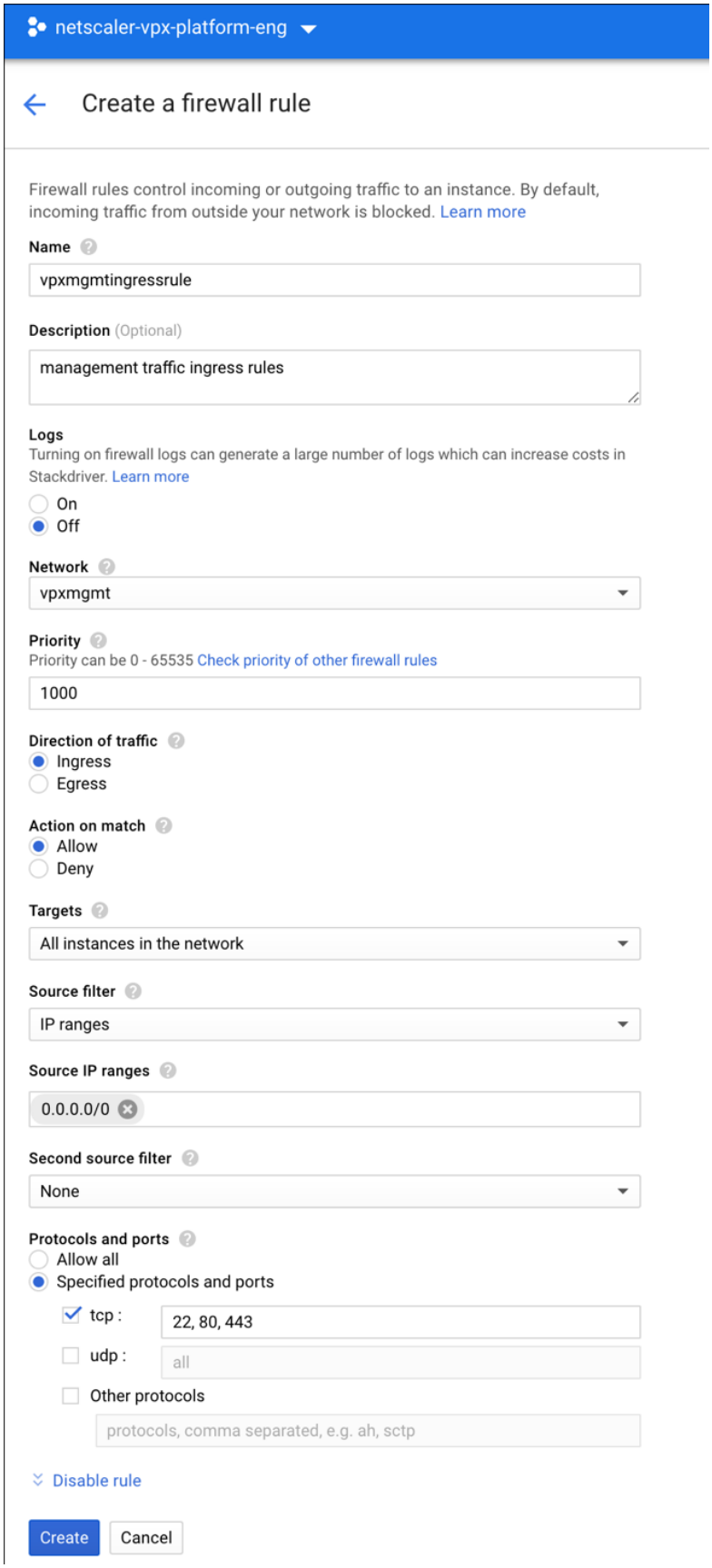


同様に、クライアント側およびサーバー側 NIC 用の VPC ネットワークを作成します。

> 注:
>
> 3 つの VPC ネットワークはすべて同じリージョン (このシナリオでは asia-east1) にある必要があります。

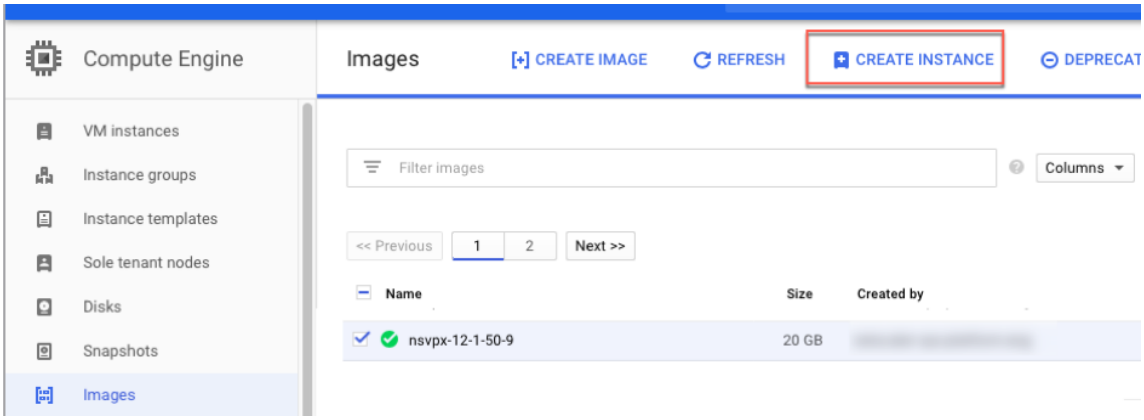
手順 **2.** ポート **22**、**80**、および **443** のファイアウォールルールを作成します。

VPC ネットワークごとに SSH（ポート 22)、HTTP（ポート 80)、HTTPS（ポート 443）のルールを作成します。ファイアウォールルールの詳細については、「ファイアウォールルールの概要」を参照してください。



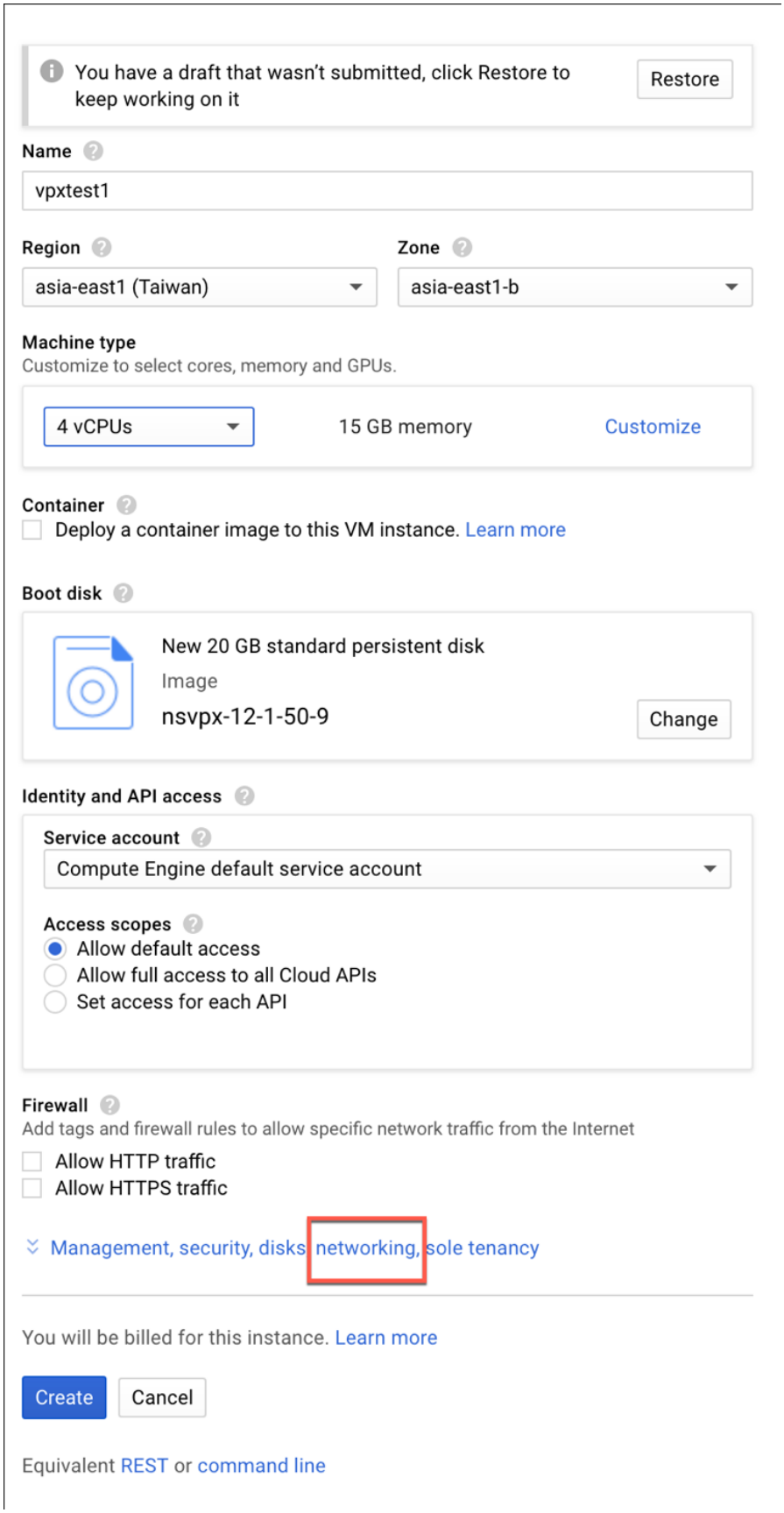
手順 **3. VPX** インスタンスを作成します。

1. GCP コンソールにログオンします。
2. [コンピュート**]** で **[**コンピュートエンジン] にカーソルを合わせ、[イメージ] を選択します。
3. イメージを選択し、[インスタンスの作成] をクリックします。

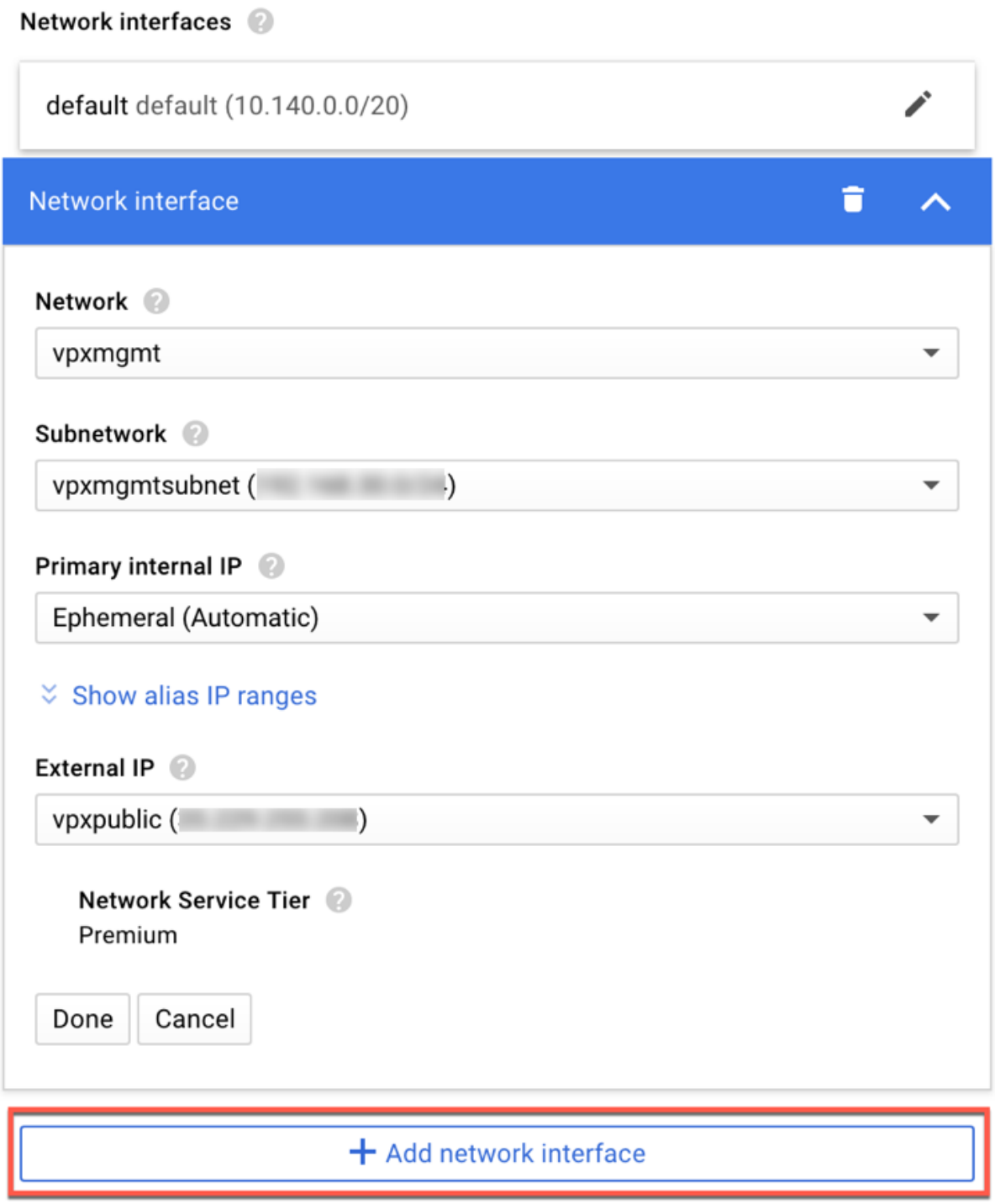


4. 複数の NIC をサポートするには、4 つの vCPU を持つインスタンスを選択します。
5. [管理、セキュリティ、ディスク、ネットワーキング、単独テナンシー] から [ネットワーク] オプションをクリックして、NIC を追加します。

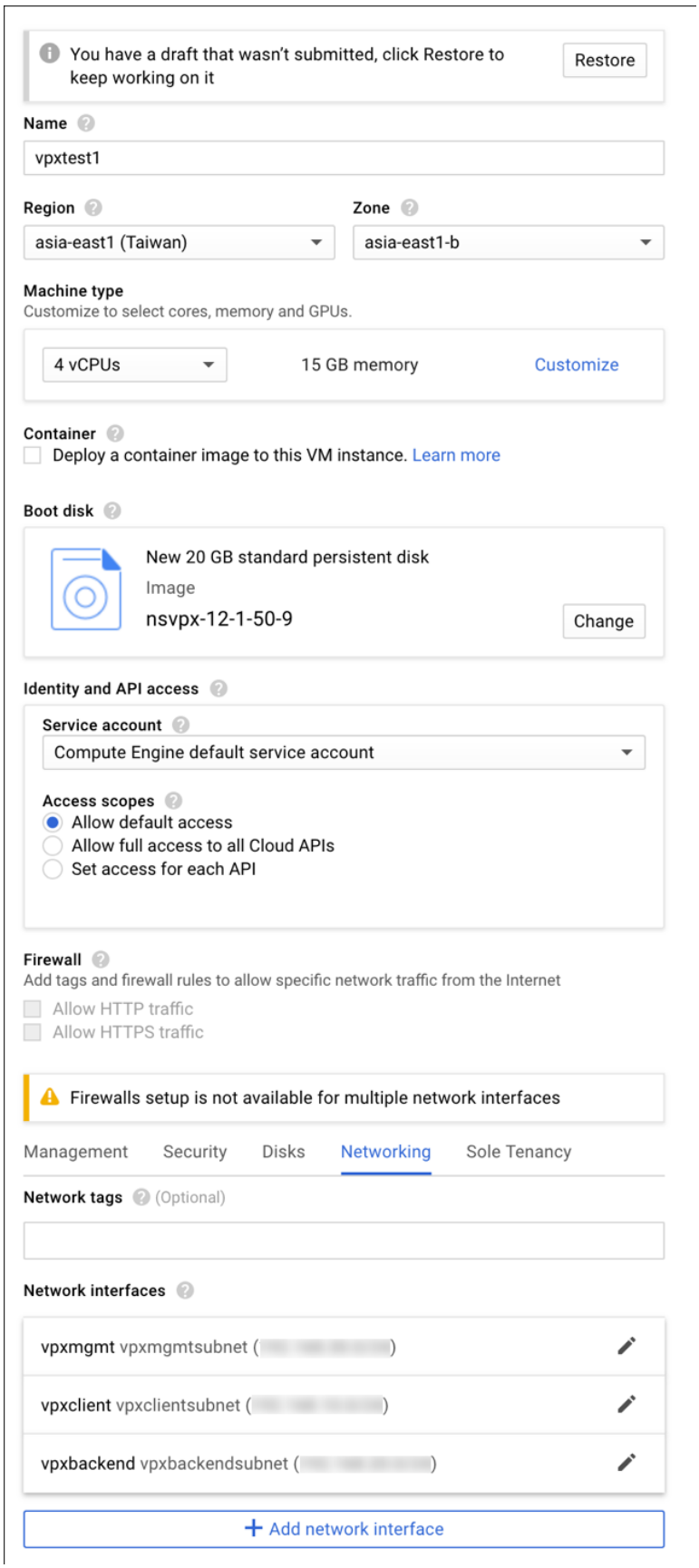
> 注:
>
> コンテナイメージは、GCP の VPX インスタンスではサポートされていません。



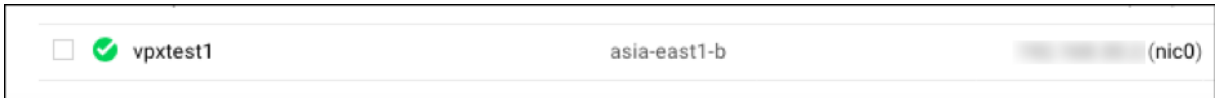
6. [ネットワークインターフェイス] で、[編集] アイコンをクリックして、デフォルトの NIC を編集します。この NIC は管理 NIC です。
7. [ネットワークインターフェイス] ウィンドウの [ネットワーク] で、管理 NIC 用に作成した VPC ネットワークを選択します。
8. 管理 NIC の場合は、静的外部 IP アドレスを作成します。[外部 IP] リストで、[**IP** アドレスの作成] をクリックします。
9. [新しい静的 **IP** アドレスを予約する] ウィンドウで、名前と説明を追加し、[予約] をクリックします。
10. [ネットワークインターフェイスの追加] をクリックして、クライアント側およびサーバー側のトラフィック用の NIC を作成します。



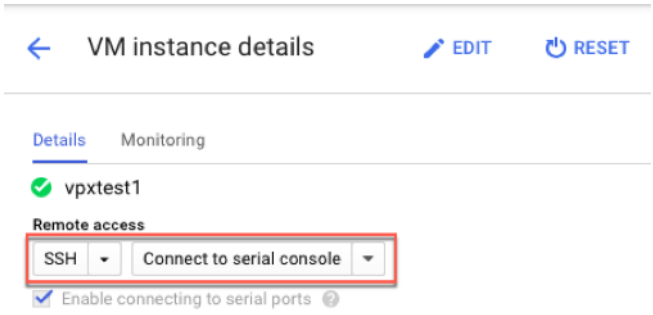
すべての NIC を作成したら、[作成] をクリックして VPX インスタンスを作成します。



インスタンスは [**VM** インスタンス] の下に表示されます。

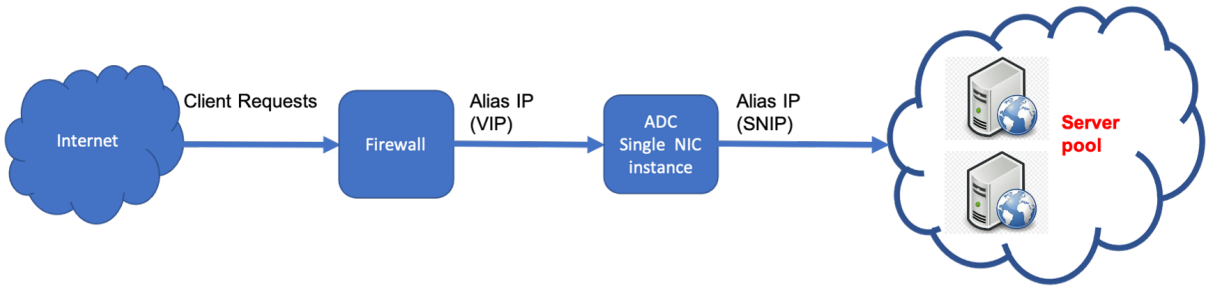


GCP SSH またはシリアルコンソールを使用して VPX インスタンスを構成および管理します。



シナリオ**:** シングル **NIC** のスタンドアロン **VPX** インスタンスをデプロイする

このシナリオでは、NetScaler VPX スタンドアロンインスタンスを単一の NIC で GCP にデプロイする方法を示しています。エイリアス IP アドレスは、この展開を実現するために使用されます。



1 つの NIC (NIC0) を作成して、次の目的を果たします。

- 管理ネットワーク内の管理トラフィック（NetScaler IP）を処理します。
- クライアントネットワーク内のクライアント側トラフィック (VIP) を処理します。
- バックエンドサーバーネットワーク内のバックエンドサーバー (SNIP) と通信します。

次の間の必要な通信ルートを設定します。

- インスタンスとバックエンドサーバー。
- パブリックインターネット上のインスタンスと外部ホスト。

導入手順の概要

1. NIC0 用の VPC ネットワークを作成します。

2. ポート 22、80、および 443 のファイアウォールルールを作成します。
3. 1 つの NIC でインスタンスを作成します。
4. VPX にエイリアス IP アドレスを追加します。
5. VPX に VIP と SNIP を追加します。
6. 負荷分散仮想サーバーを追加します。
7. インスタンスにサービスまたはサービスグループを追加します。
8. サービスまたはサービスグループをインスタンス上の負荷分散仮想サーバーにバインドします。

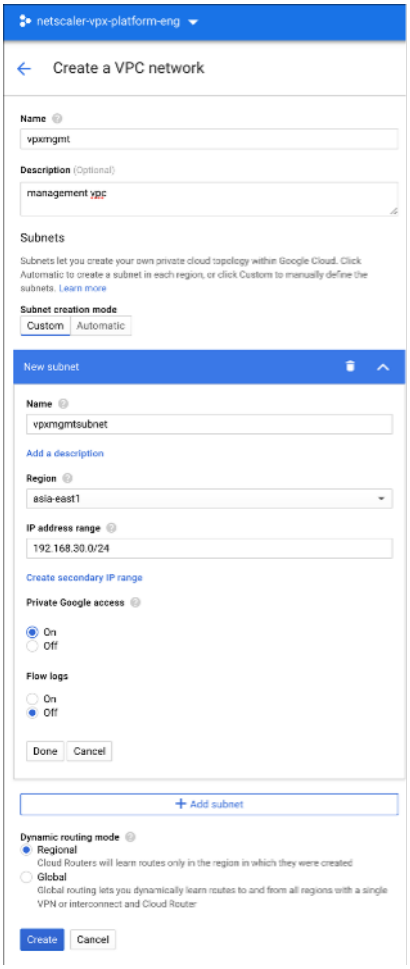
> 注記:
>
> VPC ネットワークを作成したのと同じリージョンにインスタンスを作成します。

手順 **1. 1** つの **VPC** ネットワークを作成します。

NIC0 に関連付ける VPC ネットワークを 1 つ作成します。

VPC ネットワークを作成するには、次の手順を実行します。

1. **GCP** コンソール **>** ネットワーク **> VPC** ネットワーク **> VPC** ネットワークの作成にログオンします。
2. 必須フィールドに入力し、[**Create**] をクリックします。



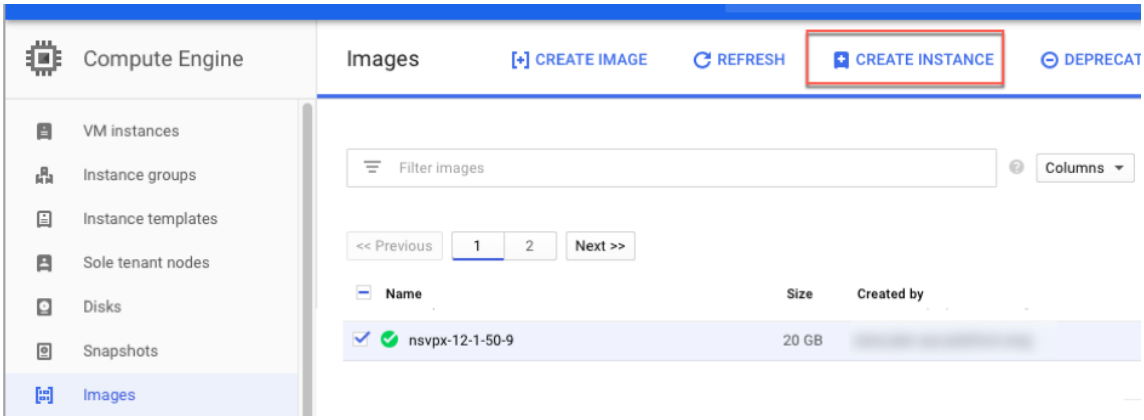
手順 **2.** ポート **22**、**80**、および **443** のファイアウォールルールを作成します。

VPC ネットワークの SSH（ポート 22）、HTTP（ポート 80）、HTTPS（ポート 443）のルールを作成します。ファイアウォールルールの詳細については、「ファイアウォールルールの概要」を参照してください。

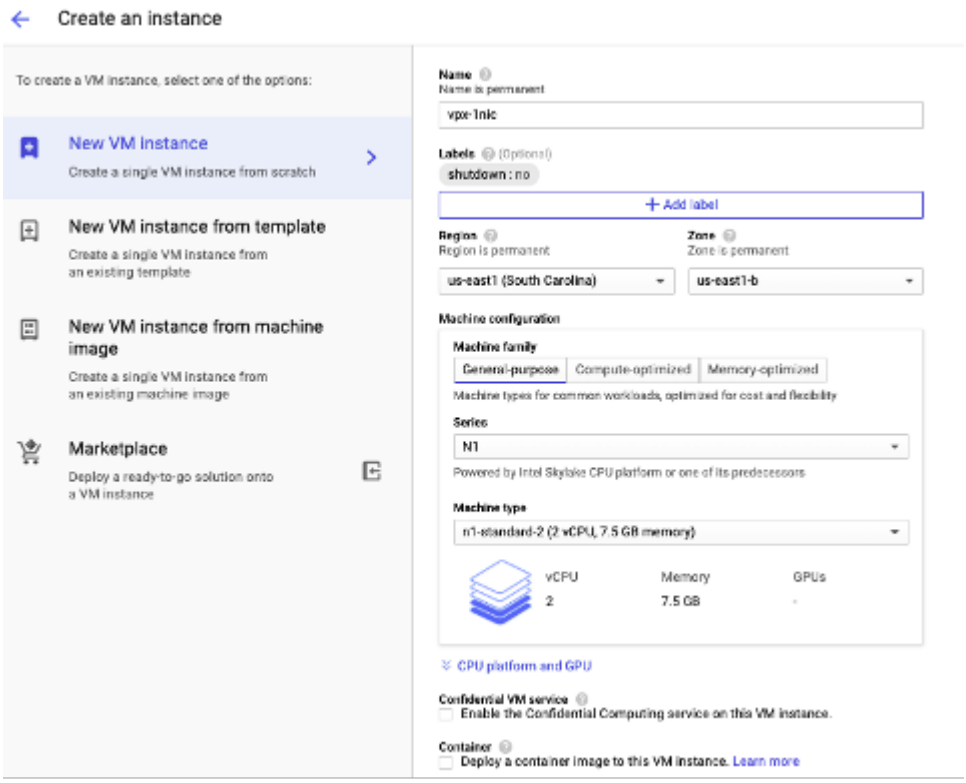
手順 **3. 1** つの **NIC** でインスタンスを作成します。

単一の NIC でインスタンスを作成するには、次の手順を実行します。

1. **GCP** コンソールにログオンします。
2. [コンピュート**]** で **[**** コンピュートエンジン**]** にカーソルを合わせ、**[**** イメージ] を選択します。
3. イメージを選択し、[インスタンスの作成] をクリックします。



4. 2 つの vCPU を持つインスタンスタイプを選択します（ADC の最小要件）。



5. [管理、セキュリティ、ディスク、** ネットワーク] ウィンドウから [ネットワーク **] タブをクリックします。
6. [ネットワークインターフェイス] で、[編集] アイコンをクリックして、デフォルトの NIC を編集します。
7. [ネットワークインターフェイス] ウィンドウの [ネットワーク] で、作成した VPC ネットワークを選択します。
8. 静的外部 IP アドレスを作成できます。[外部 **IP** アドレス**]** で、**[**IP** アドレスの作成 **] をクリックします。
9. [静的アドレスを予約] ウィンドウで、名前と説明を追加し、[予約] をクリックします。
10. [作成] をクリックして VPX インスタンスを作成します。
    新しいインスタンスが [VM インスタンス] の下に表示されます。

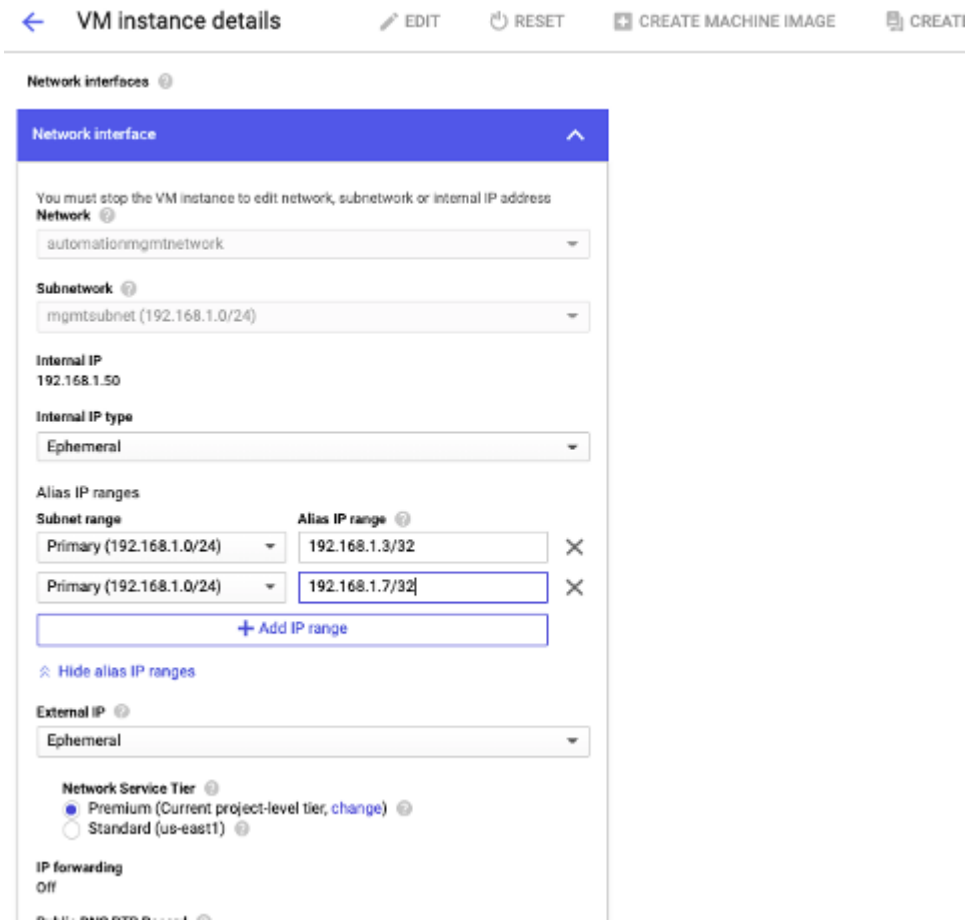
ステップ **4: VPX** インスタンスにエイリアス **IP** アドレスを追加します。

VIP アドレスと SNIP アドレスとして使用する VPX インスタンスに 2 つのエイリアス IP アドレスを割り当てます。

> 注:
>
> VPX インスタンスのプライマリ内部 IP アドレスを使用して VIP または SNIP を構成しないでください。

エイリアス IP アドレスを作成するには、次の手順を実行します。

1. VM インスタンスに移動し、[編集] をクリックします。
2. [ネットワークインターフェイス] ウィンドウで、NIC0 インターフェイスを編集します。
3. [エイリアス **IP** 範囲] フィールドに、エイリアス IP アドレスを入力します。

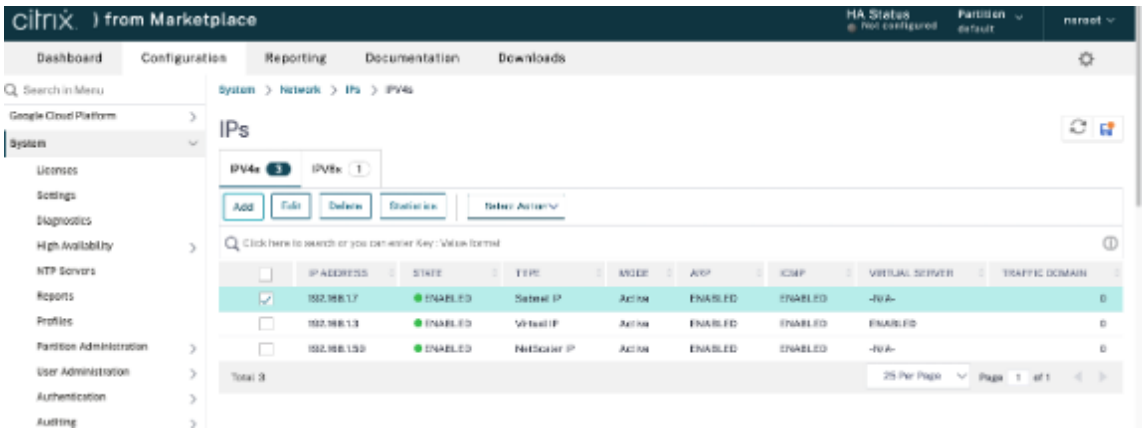


4. [完了]、[保存] の順にクリックします。
5. **VM** インスタンスの詳細ページでエイリアス IP アドレスを確認します。

手順 **5. VPX** インスタンスに **VIP** と **SNIP** を追加します。

VPX インスタンスで、クライアントエイリアス IP アドレスとサーバーエイリアス IP アドレスを追加します。

1. NetScaler GUI で、［システム］**>**［ネットワーク］**>**［**IP**］**>**［**IPv4**］に移動し、［追加］をクリックします。



2. クライアントエイリアス IP（VIP）アドレスを作成するには、次の手順を実行します。

   - VM インスタンスで VPC サブネットに設定されたクライアントエイリアス IP アドレスとネットマスクを入力します。
   - [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
   - ［作成］をクリックします。

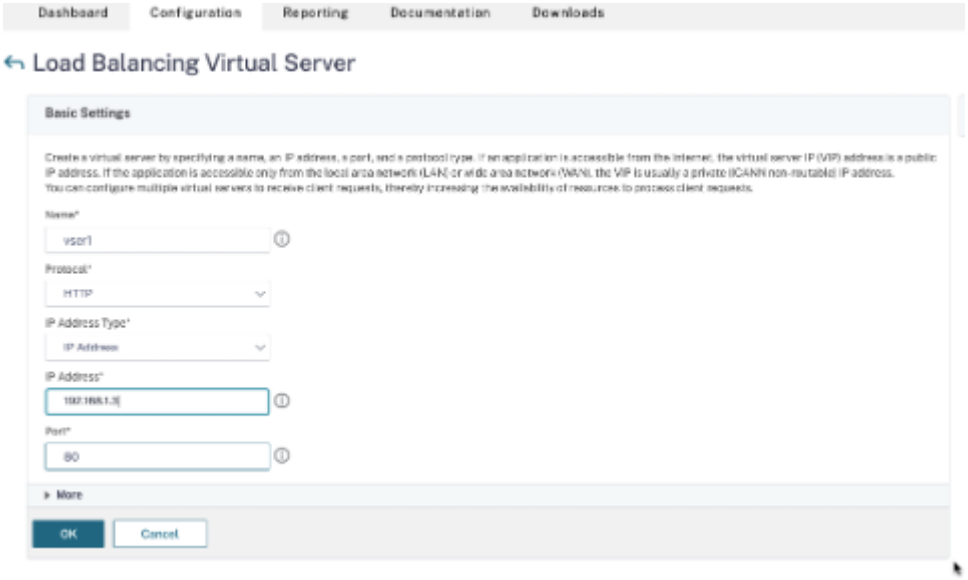
3. サーバーエイリアス IP (SNIP) アドレスを作成するには、次の手順を実行します。

   - VM インスタンスの VPC サブネットに設定されたサーバーエイリアス IP アドレスとネットマスクを入力します。
   - [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。

- ［作成］をクリックします。

ステップ **6:** 負荷分散仮想サーバーを追加します。

1. NetScaler GUI で、［構成］**>**［トラフィック管理］**>**［負荷分散］**>**［仮想サーバー］に移動し、［追加］をクリックします。
2. [名前]、[プロトコル]、[IP アドレスタイプ (IP アドレス)]、[IP アドレス] (クライアントエイリアス IP)、および [ポート] に必要な値を追加します。
3. **OK** をクリックして、負荷分散仮想サーバーを作成します。



手順 **7. VPX** インスタンスにサービスまたはサービスグループを追加します。

1. NetScaler GUI から、［構成］**>**［トラフィック管理］**>**［負荷分散］**>**［サービス］に移動し、［追加］をクリックします。
2. サービス名、IP アドレス、プロトコル、およびポートに必要な値を追加し、「**OK**」をクリックします。

手順 **8.** サービス**/**サービスグループをインスタンス上の負荷分散仮想サーバーにバインドします。

1. GUI から、[設定**] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバ] に移動します。
2. 手順 **6** で構成した負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [サービスとサービスグループ] ウィンドウで、[負荷分散仮想サーバーサービスのバインドなし] をクリックします。
4. ステップ **7** で設定したサービスを選択し、[バインド（Bind）] をクリックします。

VPX インスタンスを **GCP** にデプロイした後の注意点

- ユーザー名`nsroot`とインスタンス ID をパスワードとして VPX にログオンします。プロンプトで、パスワードを変更し、設定を保存します。

- テクニカルサポートバンドルを収集するには、慣例`show techsupport`ではなくコマンド`shell /netscaler/showtech_cloud.pl`を実行します。
- GCP コンソールから NetScaler ADC VM を削除した後、関連する NetScaler ADC 内部ターゲットインスタンスも削除します。これを行うには、gcloud CLI に移動し、次のコマンドを入力します。

```
gcloud compute -q  target-instances delete <instance-name>-
   adcinternal --zone <zone>
<!--NeedCopy-->
```

> 注:
>
> `<instance-name>-adcinternal` は、削除する必要があるターゲットインスタンスの名前です。

NetScaler VPX ライセンス

GCP 上の NetScaler ADC VPX インスタンスにはライセンスが必要です。GCP で実行されている NetScaler ADC VPX インスタンスでは、以下のライセンスオプションを使用できます。

- サブスクリプションベースのライセンス：NetScaler VPX アプライアンスは、GCP マーケットプレイスで有料インスタンスとして利用できます。サブスクリプションベースのライセンスは、従量課金制のオプションです。ユーザーは時間単位で課金されます。GCP マーケットプレイスでは、以下の VPX モデルとライセンスエディションが利用可能です。

VPX モデル	ライセンスエディション
VPX10、VPX200、VPX1000、VPX3000、VPX5000	スタンダード、アドバンス、プレミアム

- 自分のライセンスを持参 **(BYOL)**: 自分のライセンス (BYOL) を持ち込む場合は、http://support.citrix.com/article/CTX122426にある VPX ライセンスガイドを参照してください。次の操作を実行する必要があります。
  - Citrix Web サイト内のライセンスポータルを使用して、有効なライセンスを生成します。
  - ライセンスをインスタンスにアップロードします。
- **NetScaler VPX** チェックイン/チェックアウトライセンス：詳細については、「NetScalerVPX チェックイン/チェックアウトライセンス」を参照してください。

オンプレミスおよびクラウド展開用の VPX Express では、ライセンスファイルは不要です。NetScaler VPX Express の詳細については、Citrix ADC ライセンスの概要の「NetScaler VPX Express ライセンス」セクションを参照してください。

NetScaler VPX インスタンスを展開するための GDM テンプレート

NetScaler VPX Google デプロイメントマネージャー（GDM）テンプレートを使用して、GCP に VPX インスタンスを展開できます。詳細については、NetScaler GDM テンプレートを参照してください。

NetScaler マーケットプレイスのイメージ

GDM テンプレート内のイメージを使用して、NetScaler ADC アプライアンスを起動できます。

次の表は、GCP マーケットプレイスで利用可能な画像の一覧です。

解除	イメージ名	イメージの場所
13.0	citrix-adc-vpx-10-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-10-enterprise-13-0-83-29
13.0	citrix-adc-vpx-10-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-10-platinum-13-0-83-29
13.0	citrix-adc-vpx-10-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-10-standard-13-0-83-29
13.0	citrix-adc-vpx-200-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-200-enterprise-13-0-83-29
13.0	citrix-adc-vpx-200-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-200-platinum-13-0-83-29
13.0	citrix-adc-vpx-200-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-200-standard-13-0-83-29
13.0	citrix-adc-vpx-1000-advanced-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-advanced-13-0-83-29

解除	イメージ名	イメージの場所
13.0	citrix-adc-vpx-1000-premium-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-premium-13-0-83-29
13.0	citrix-adc-vpx-1000-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-standard-13-0-83-29
13.0	citrix-adc-vpx-3000-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-enterprise-13-0-83-29
13.0	citrix-adc-vpx-3000-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-platinum-13-0-83-29
13.0	citrix-adc-vpx-3000-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-standard-13-0-83-29
13.0	citrix-adc-vpx-5000-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-enterprise-13-0-83-29
13.0	citrix-adc-vpx-5000-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-platinum-13-0-83-29
13.0	citrix-adc-vpx-5000-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-standard-13-0-83-29
13.0	citrix-adc-vpx-byol-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-byol-13-0-83-29
13.0	citrix-adc-vpx-express-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-express-13-0-83-29

解除	イメージ名	イメージの場所
13.0	citrix-adc-vpx-waf-1000-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-waf-1000-13-0-83-29
13.1	citrix-adc-vpx-10-enterprise-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-10-enterprise-13-1-9-60
13.1	citrix-adc-vpx-10-platinum-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-10-platinum-13-1-9-60
13.1	citrix-adc-vpx-10-standard-13-1-9-60	projects/citrix-master-project/global/imagescitrix-adc-vpx-10-standard-13-1-9-60
13.1	citrix-adc-vpx-200-enterprise-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-200-enterprise-13-1-9-60
13.1	citrix-adc-vpx-200-platinum-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-200-platinum-13-1-9-60
13.1	citrix-adc-vpx-200-standard-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-200-standard-13-1-9-60
13.1	citrix-adc-vpx-1000-advanced-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-advanced-13-1-9-60
13.1	citrix-adc-vpx-1000-premium-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-premium-13-1-9-60
13.1	citrix-adc-vpx-1000-standard-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-standard-13-1-9-60

解除	イメージ名	イメージの場所
13.1	citrix-adc-vpx-3000-enterprise-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-enterprise-13-1-9-60
13.1	citrix-adc-vpx-3000-platinum-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-platinum-13-1-9-60
13.1	citrix-adc-vpx-3000-standard-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-standard-13-1-9-60
13.1	citrix-adc-vpx-5000-enterprise-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-enterprise-13-1-9-60
13.1	citrix-adc-vpx-5000-platinum-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-platinum-13-1-9-60
13.1	citrix-adc-vpx-5000-standard-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-standard-13-1-9-60
13.1	citrix-adc-vpx-byol-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-byol-13-1-9-60
13.1	citrix-adc-vpx-express-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-express-13-1-9-60
13.1	citrix-adc-vpx-waf-1000-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-waf-1000-13-1-9-60

リソース

- 複数のネットワークインターフェースを持つインスタンスの作成

- VM インスタンスの作成と起動

関連情報

- VPX の高可用性ペアを Google Cloud Platform に展開する

VPX の高可用性ペアを Google Cloud Platform に展開する

August 15, 2023

Google Cloud Platform（GCP）上の 2 つの NetScaler ADC VPX インスタンスを、高可用性（HA）アクティブ/パッシブペアとして構成できます。1 つのインスタンスをプライマリノードとして、もう 1 つのインスタンスをセカンダリノードとして設定すると、1 次ノードは接続を受け入れ、サーバーを管理します。セカンダリノードはプライマリを監視します。何らかの理由で 1 次ノードが接続を受け入れることができない場合、2 次ノードが引き継ぎます。

HA の詳細については、「高可用性」を参照してください。

ノードは同じリージョンにある必要がありますが、同じゾーンまたは異なるゾーンにある可能性があります。詳細については、「リージョンとゾーン」を参照してください。

各 VPX インスタンスには、少なくとも 3 つの IP サブネット（Google VPC ネットワーク）が必要です。

- 管理サブネット
- クライアント側サブネット（VIP）
- バックエンド向けサブネット (SNIP、MIP など)

Citrix では、標準の VPX インスタンスには 3 つのネットワークインターフェイスを推奨しています。

VPX 高可用性ペアは、次の方法でデプロイできます。

- 外部固定 IP アドレスの使用
- プライベート IP アドレスを使用する
- プライベート IP アドレスを持つシングル NIC 仮想マシンの使用

VPX 高可用性ペアを GCP にデプロイするための GDM テンプレート

NetScaler Google デプロイメントマネージャー（GDM）テンプレートを使用して、GCP に VPX 高可用性ペアを展開できます。詳細については、NetScaler GDM テンプレートを参照してください。

GCP での VPX 高可用性ペアの転送ルールのサポート

転送ルールを使用して、GCP に VPX 高可用性ペアをデプロイできます。

転送ルールの詳細については、「転送ルールの概要」を参照してください。

前提条件

- 転送ルールは VPX インスタンスと同じリージョンにある必要があります。
- ターゲットインスタンスは VPX インスタンスと同じゾーンにある必要があります。
- プライマリノードとセカンダリノードの両方のターゲットインスタンスの数が一致する必要があります。

例**:**

`us-east1`リージョンには、`us-east1-b`ゾーンにプライマリ VPX、`us-east1-c`ゾーンにセカンダリ VPX がある高可用性ペアがあります。プライマリ VPX には、ターゲットインスタンスが`us-east1-b`ゾーンにある転送ルールが設定されます。`us-east1-c`ゾーン内のセカンダリ VPX のターゲットインスタンスを構成して、フェイルオーバー時に転送ルールを更新します。

制限事項

VPX の高可用性展開では、バックエンドでターゲットインスタンスで構成された転送ルールのみがサポートされます。

Google Cloud Platform に外部の静的 IP アドレスを指定した VPX 高可用性ペアをデプロイする

August 15, 2023

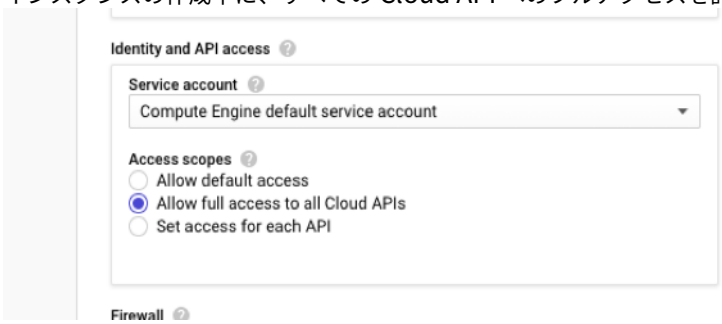
VPX 高可用性ペアは、外部の静的 IP アドレスを使用して GCP にデプロイできます。プライマリノードのクライアント IP アドレスは、外部の静的 IP アドレスにバインドする必要があります。フェールオーバー時に、外部スタティック IP アドレスはセカンダリノードに移動され、トラフィックが再開されます。

静的外部 IP アドレスは、リリースを決定するまでプロジェクト用に予約されている外部 IP アドレスです。IP アドレスを使用してサービスにアクセスする場合、その IP アドレスを予約して、プロジェクトのみが使用できるようにすることができます。詳細については、「静的外部 IP アドレスの予約」を参照してください。

HA の詳細については、「高可用性」を参照してください。

はじめに

- Google Cloud Platform への NetScaler ADC VPX インスタンスのデプロイで説明されている制限、ハードウェア要件、注意点をお読みください。この情報は、HA 配置にも適用されます。
- GCP プロジェクトで クラウドリソースマネージャー **API** を有効にします。
- インスタンスの作成中に、すべての Cloud API へのフルアクセスを許可します。



- GCP サービスアカウントに関連付けられた IAM ロールに次の IAM 権限があることを確認します。

```
REQUIRED_INSTANCE_IAM_PERMS = [

"compute.addresses.use",
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.setMetadata"
"compute.instances.addAccessConfig",
"compute.instances.deleteAccessConfig",
"compute.instances.get",
"Compute.instances.list",
"compute.networks.useExternalIp",
"compute.subnetworks.useExternalIp",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.targetInstances.create",
"compute.zones.list",
"compute.zoneOperations.get",
]
<!--NeedCopy-->
```

- 管理インターフェイス以外のインターフェイスでエイリアス IP アドレスを設定している場合は、GCP サービスアカウントに次の追加の IAM 権限があることを確認してください。

```
"compute.instances.updateNetworkInterface"
<!--NeedCopy-->
```

- プライマリノードで GCP 転送ルールを構成した場合は、「GCP での VPX 高可用性ペアの転送ルールのサポート」に記載されている制限と要件を読み、フェールオーバー時に新しいプライマリに更新します。

Google Cloud Platform に VPX HA ペアを展開する方法

HA 展開手順の概要を次に示します。

1. 同じリージョンに VPC ネットワークを作成します。たとえば、アジア東です。
2. 同じリージョンに 2 つの VPX インスタンス (プライマリノードとセカンダリノード) を作成します。同じゾーンまたは異なるゾーンに配置できます。たとえば、アジア東-1a、アジア東-Ib。
3. NetScaler GUI または ADC CLI コマンドを使用して、両方のインスタンスで高可用性設定を構成します。

手順 1. VPC ネットワークを作成する

要件に基づいて VPC ネットワークを作成します。管理 NIC、クライアント NIC、サーバー NIC に関連付けるために 3 つの VPC ネットワークを作成することをお勧めします。

VPC ネットワークを作成するには、次の手順を実行します。

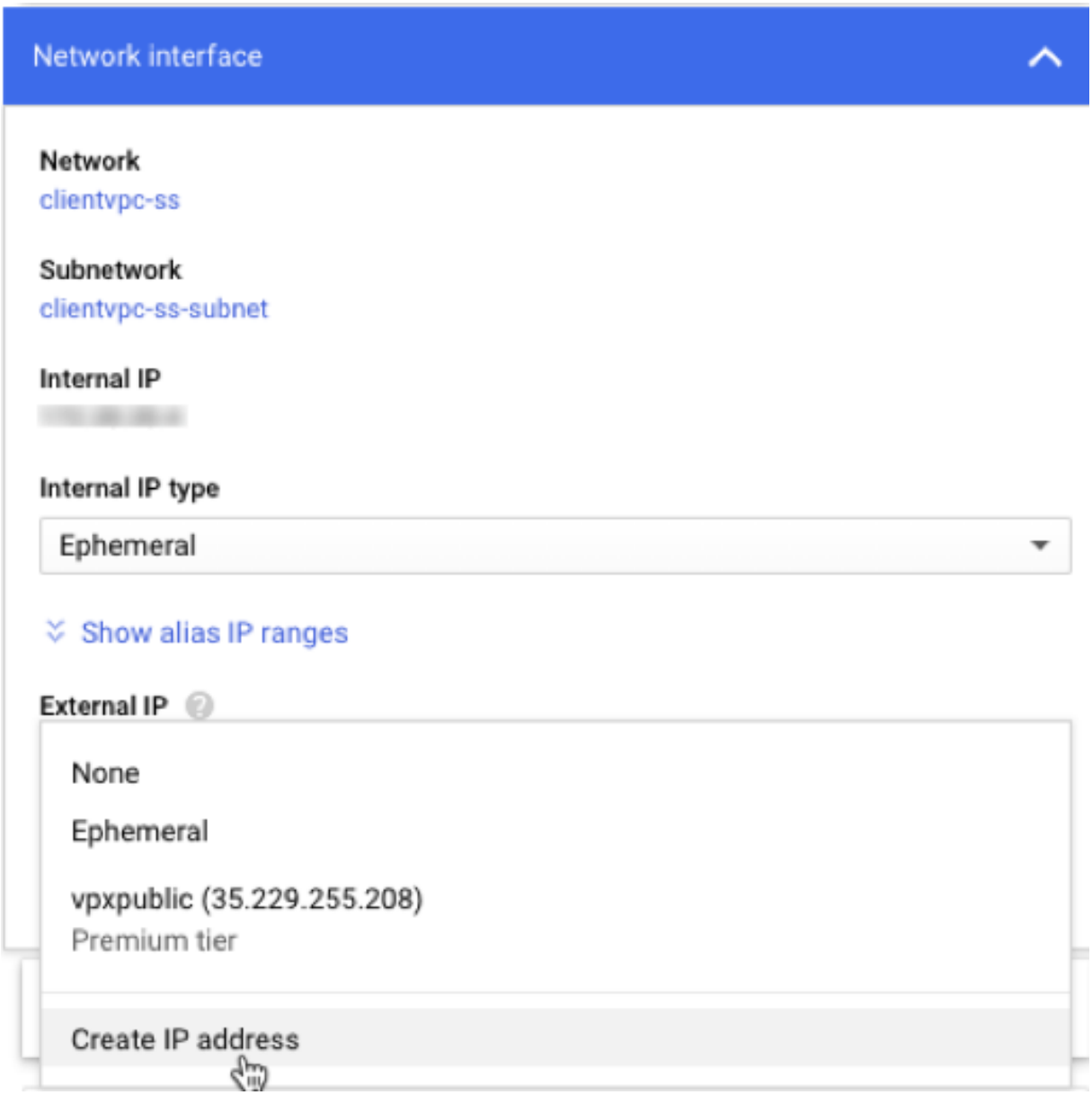
1. **Google** コンソール **>** ネットワーク **> VPC** ネットワーク **> VPC** ネットワークの作成にログオンします。
2. 必須フィールドに入力し、[**Create**] をクリックします。

詳細については、「Google Cloud Platform での NetScaler ADC VPX インスタンスのデプロイ」の「**VPC** ネットワークの作成」セクションを参照してください。

手順 2. 2 つの VPX インスタンスを作成する

シナリオ：Multi-NIC、Multi-IP スタンドアロン VPX インスタンスを展開するの手順に従って、VPX インスタンスを 2 つ作成します。

> 重要
>
> プライマリノードのクライアント IP アドレス (VIP) に静的外部 IP アドレスを割り当てます。既存の予約済み IP アドレスを使用するか、新しい予約済み IP アドレスを作成できます。静的外部 IP アドレスを作成するには、[ネットワークインターフェイス] **>** [外部 **IP**] に移動し、[**IP** アドレスの作成] をクリックします。



フェールオーバー後、古いプライマリが新しいセカンダリになると、スタティック外部 IP アドレスは古いプライマリから移動し、新しいプライマリに接続されます。詳細については、Google Cloud ドキュメント「静的外部 IP アドレスを予約する」を参照してください。

VPX インスタンスを構成したら、VIP アドレスと SNIP アドレスを構成できます。詳細については、「NetScaler ADC 所有の IP アドレスの構成」を参照してください。

手順 **3.** 高可用性を構成する

Google Cloud Platform でインスタンスを作成した後、CLI 用 NetScaler ADC GUI を使用して HA を構成できます。

GUI を使用した **HA** の設定　ステップ **1:** 両方のインスタンスで INC モードで高可用性をセットアップします。

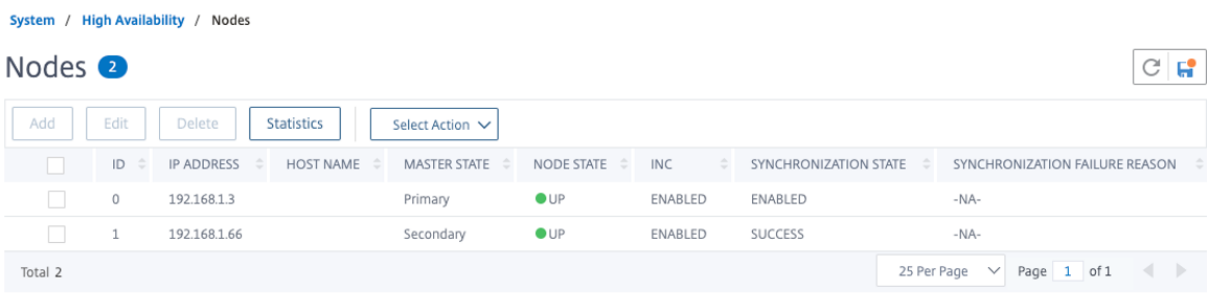
プライマリノードで、次の手順を実行します。

1. GCP Console からノードのユーザー名`nsroot`とインスタンス ID をパスワードとしてインスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノードの **IP** アドレス] フィールドに、セカンダリノードの管理 NIC のプライベート IP アドレスを入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. [作成] をクリックします。

セカンダリノードで、次の手順を実行します。

1. GCP Console からノードのユーザー名`nsroot`とインスタンス ID をパスワードとしてインスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノード **IP** アドレス] フィールドに、プライマリノードの管理 NIC のプライベート IP アドレスを入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. [作成] をクリックします。

先に進む前に、[**Nodes**] ページにセカンダリノードの同期状態が **SUCCESS** と表示されていることを確認してください。



> 注
>
> これで、セカンダリノードは、プライマリノードと同じログオン資格情報を持ちます。

ステップ **2:** 両方のノードに仮想 IP アドレスとサブネット IP アドレスを追加します。

プライマリノードで、次の手順を実行します。

1. [システム**] > [**ネットワーク**] > [IP] > [IPv4**] に移動し、[追加] をクリックします。
2. 次の手順に従って、プライマリ VIP アドレスを追加します。

    a) プライマリ・インスタンスのクライアント側インターフェイスの内部 IP アドレスと、VM インスタンスのクライアント・サブネットに対して構成されたネットマスクを入力します。

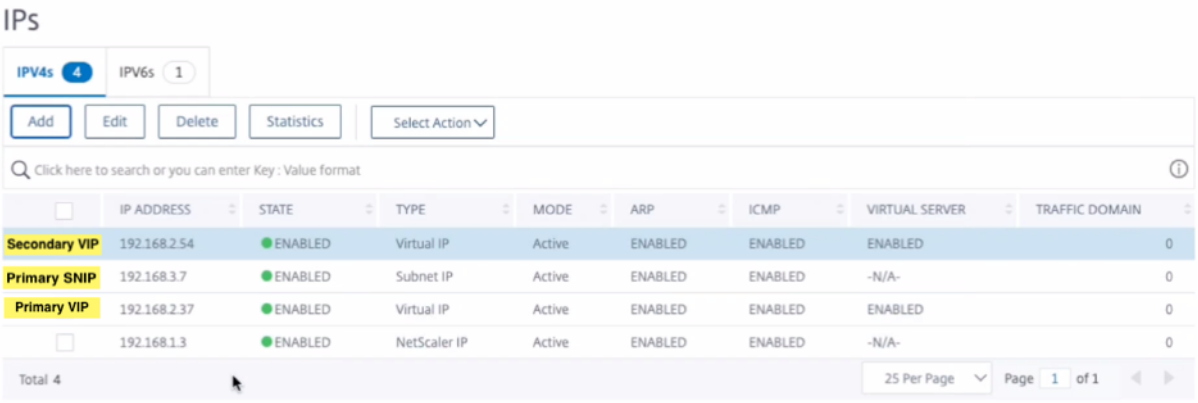
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
c) ［作成］をクリックします。

3. 次の手順に従って、プライマリ SNIP アドレスを追加します。

a) プライマリ・インスタンスのサーバ側インターフェイスの内部 IP アドレスと、プライマリ・インスタンスのサーバ・サブネットに対して構成されたネットマスクを入力します。
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
c) ［作成］をクリックします。

4. 次の手順に従って、セカンダリ VIP アドレスを追加します。

a) セカンダリインスタンスのクライアント側インターフェイスの内部 IP アドレスと、VM インスタンスのクライアントサブネットに設定されたネットマスクを入力します。
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
c) ［作成］をクリックします。



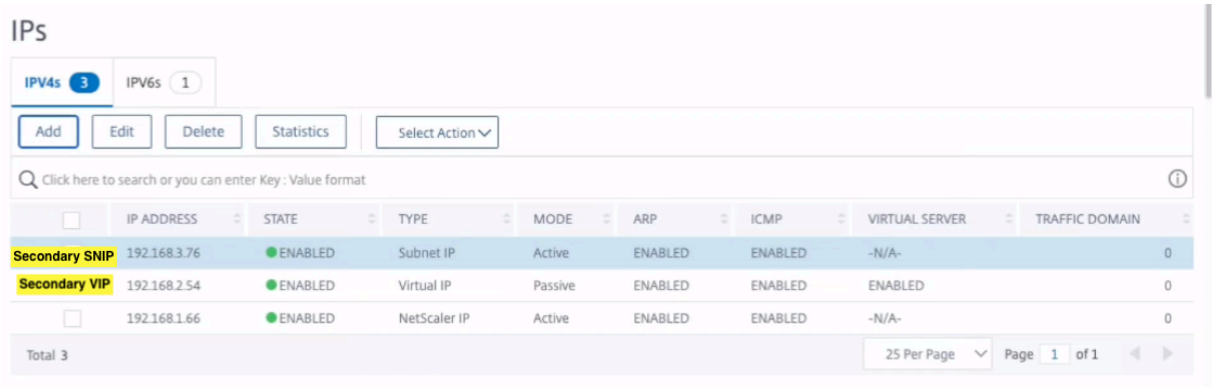
セカンダリノードで、次の手順を実行します。

1. [システム] **>** [ネットワーク] **>** **[IP] > [IPv4]** に移動し、[追加] をクリックします。
2. 次の手順に従って、セカンダリ VIP アドレスを追加します。

a) セカンダリインスタンスのクライアント側インターフェイスの内部 IP アドレスと、VM インスタンスのクライアントサブネットに設定されたネットマスクを入力します。
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。

3. 次の手順に従って、セカンダリ SNIP アドレスを追加します。

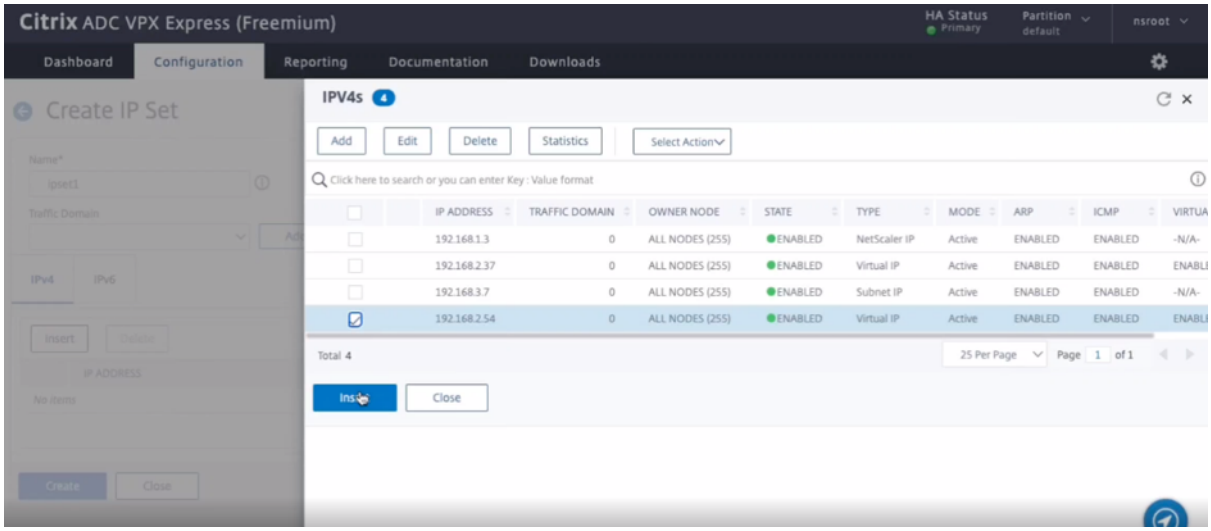
a) セカンダリインスタンスのサーバ側インターフェイスの内部 IP アドレスと、セカンダリインスタンスのサーバサブネットに設定されたネットマスクを入力します。
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
c) ［作成］をクリックします。



ステップ **3:** IP セットを追加し、両方のインスタンスで IP セットをセカンダリ VIP にバインドします。

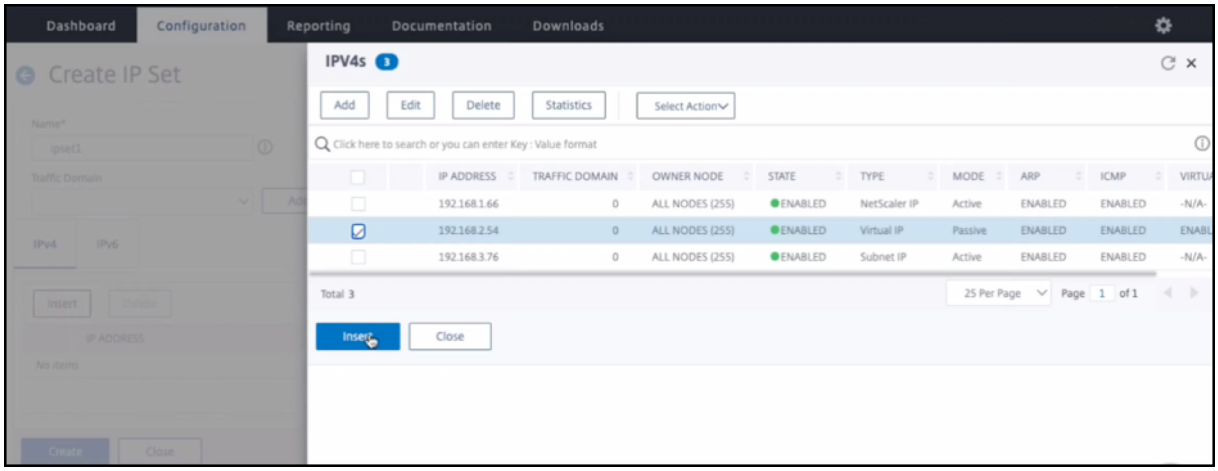
プライマリノードで、次の手順を実行します。

1. **System > Network > IP Sets > Add** に移動します。
2. IP セット名を追加し、[**Insert**] をクリックします。
3. [**IPv4**] ページで、仮想 IP（セカンダリ VIP）を選択し、[挿入] をクリックします。
4. [**Create**] をクリックして IP セットを作成します。



セカンダリノードで、次の手順を実行します。

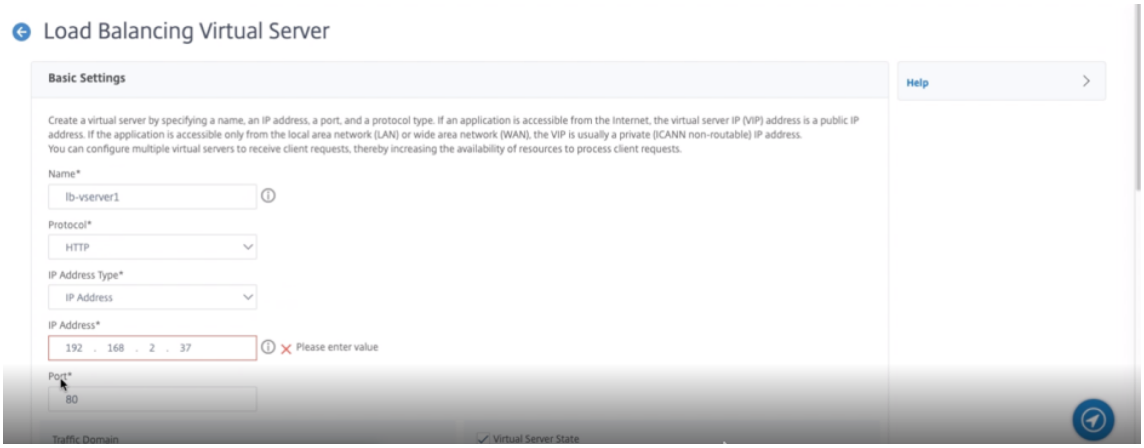
1. **System > Network > IP Sets > Add** に移動します。
2. IP セット名を追加し、[**Insert**] をクリックします。
3. [**IPv4**] ページで、仮想 IP（セカンダリ VIP）を選択し、[挿入] をクリックします。
4. [**Create**] をクリックして IP セットを作成します。



> 注
>
> IP セット名は、両方のインスタンスで同じである必要があります。

ステップ **4:** プライマリインスタンスに負荷分散仮想サーバーを追加します。

1. [設定] **>** [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] **>** [追加] に移動します。
2. [名前]、[プロトコル]、[IP アドレスタイプ (IP アドレス)]、[IP アドレス] (プライマリ VIP)、および [ポート] に必要な値を追加します。



3. [詳細] クリックします。[**IP** 範囲 **IP** セット設定] に移動し、ドロップダウンメニューから [**IPSet**] を選択し、ステップ **3** で作成した IPSet を指定します。
4. **OK** をクリックして、負荷分散仮想サーバーを作成します。

ステップ **5:** プライマリノードにサービスまたはサービスグループを追加します。

1. **[設定] > [**トラフィック管理**] > [**負荷分散**] > [**サービス**] >[**追加**]** に移動します。
2. サービス名、IP アドレス、プロトコル、およびポートに必要な値を追加し、[**OK**] をクリックします。

ステップ **6:** サービスまたはサービスグループをプライマリノードの負荷分散仮想サーバーにバインドします。

1. **[設定] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動します。
2. 手順 **4** で構成した負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [サービスとサービスグループ] タブで、[負荷分散仮想サーバーサービスバインドなし] をクリックします。
4. 手順 **5** で構成したサービスを選択し、[バインド] をクリックします。

構成を保存します。強制フェールオーバーの後、セカンダリは新しいプライマリになります。古いプライマリ VIP の外部スタティック IP は、新しいセカンダリ VIP に移動します。

CLI を使用した高可用性の設定　ステップ **1:** 両方のインスタンスで INC モードで高可用性をセットアップします。

プライマリノードで、次のコマンドを入力します。

```
add ha node 1 <sec_ip> -inc ENABLED
<!--NeedCopy-->
```

セカンダリノードで、次のコマンドを入力します。

```
add ha node 1 <prim_ip> -inc ENABLED
<!--NeedCopy-->
```

sec_ipは、セカンダリノードの管理 NIC の内部 IP アドレスを指します。

prim_ipは、プライマリノードの管理 NIC の内部 IP アドレスを指します。

ステップ **2:** 両方のノードに仮想 IP とサブネット IP を追加します。

プライマリノードで、次のコマンドを入力します。

```
add ns ip <primary_vip> <subnet> -type VIP

add ns ip <secondary_vip> <subnet> -type VIP

add ns ip <primary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

primary_vipは、プライマリ・インスタンスのクライアント側インターフェイスの内部 IP アドレスを指します。

secondary_vipは、セカンダリインスタンスのクライアント側インターフェイスの内部 IP アドレスを指します。

primary_snipは、プライマリ・インスタンスのサーバ側インターフェイスの内部 IP アドレスを指します。

セカンダリノードで、次のコマンドを入力します。

```
add ns ip <secondary_vip> <subnet> -type VIP

add ns ip <secondary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```


secondary_vipは、セカンダリインスタンスのクライアント側インターフェイスの内部 IP アドレスを指します。

secondary_snipは、セカンダリインスタンスのサーバ側インターフェイスの内部 IP アドレスを指します。

ステップ **3:** IP セットを追加し、両方のインスタンスで IP セットをセカンダリ VIP にバインドします。

プライマリノードで、次のコマンドを入力します。

```
add ipset <ipsetname>
bind ipset <ipsetname> <secondary VIP>
<!--NeedCopy-->
```

セカンダリノードで、次のコマンドを入力します。

```
add ipset <ipsetname>
bind ipset <ipsetname> <secondary VIP>
<!--NeedCopy-->
```

> 注
>
> IP セット名は、両方のインスタンスで同じである必要があります。

ステップ **4:** プライマリ・インスタンスに仮想サーバを追加します。

次のコマンドを入力します。

```
add <server_type> vserver <vserver_name> <protocol> <primary_vip> <port
    > -ipset <ipset_name>
<!--NeedCopy-->
```

ステップ **5:** プライマリインスタンスにサービスまたはサービスグループを追加します。

次のコマンドを入力します。

```
add service <service_name> <service_ip_address> <protocol> <port>
<!--NeedCopy-->
```

ステップ **6:** サービス/サービスグループをプライマリインスタンス上の負荷分散仮想サーバーにバインドします。

次のコマンドを入力します。

```
bind <server_type> vserver <vserver_name> <service_name>
<!--NeedCopy-->
```

> 注記:
>
> 設定を保存するには、コマンドsave configを入力します。そうしないと、インスタンスの再起動後に設定が失われます。

ステップ **7:** 設定を確認します。

プライマリクライアント NIC に接続されている外部 IP アドレスが、フェールオーバー時にセカンダリに移動することを確認します。

1. 外部 IP アドレスに cURL 要求を行い、それが到達可能であることを確認します。
2. プライマリインスタンスで、フェイルオーバーを実行します。

   GUI から、[設定] **>** [システム] **>** [高可用性] **>** [アクション] **>** [強制フェールオーバー] に移動します。

   CLI から、次のコマンドを入力します。

   ```
   force ha failover -f
   <!--NeedCopy-->
   ```

   GCP コンソールで、セカンダリインスタンスに移動します。外部 IP アドレスは、フェールオーバー後にセカンダリのクライアント NIC に移動されている必要があります。
3. 外部 IP に cURL 要求を発行し、再び到達可能であることを確認します。

Google Cloud Platform にプライベート IP アドレスを指定した 1 つの NIC VPX 高可用性ペアをデプロイします

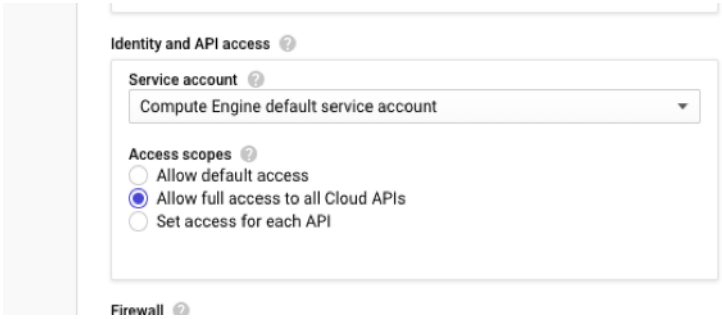
August 15, 2023

プライベート IP アドレスを使用して、単一の NIC VPX 高可用性ペアを GCP にデプロイできます。クライアント IP (VIP) アドレスは、プライマリノードのエイリアス IP アドレスとして設定する必要があります。フェールオーバー時に、クライアント IP アドレスがセカンダリノードに移動され、トラフィックが再開されます。各ノードのサブネット IP (SNiP) アドレスもエイリアス IP 範囲として設定する必要があります。

高可用性の詳細については、「高可用性」を参照してください。

はじめに

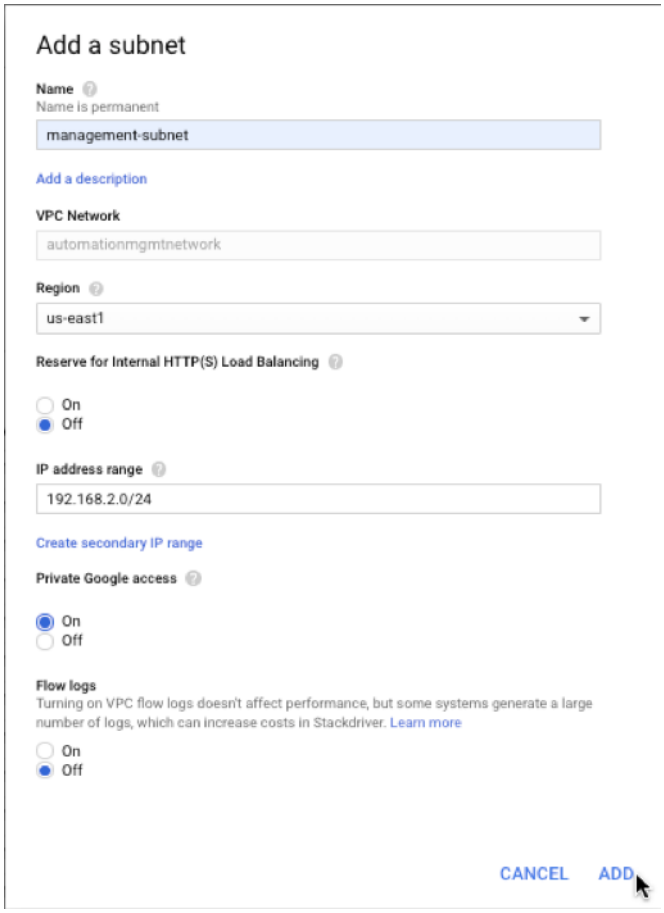
- Google Cloud Platform への NetScaler ADC VPX インスタンスのデプロイで説明されている制限、ハードウェア要件、注意点をお読みください。この情報は、高可用性展開にも適用されます。
- GCP プロジェクトで クラウドリソースマネージャー **API** を有効にします。
- インスタンスの作成中に、すべての Cloud API へのフルアクセスを許可します。



- GCP サービスアカウントに次の IAM 権限があることを確認します。

```
REQUIRED_INSTANCE_IAM_PERMS = [
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.setMetadata",
"compute.instances.get",
"compute.instances.list",
"compute.instances.updateNetworkInterface",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.targetInstances.create",
"compute.zones.list",
"compute.zoneOperations.get",
]
<!--NeedCopy-->
```

- VM がインターネットにアクセスできない場合は、VPC サブネットでプライベート **Google** アクセスを有効にする必要があります。



- プライマリノードで GCP 転送ルールを構成した場合は、「GCP での VPX 高可用性ペアの転送ルールのサポート」に記載されている制限と要件を読み、フェールオーバー時に新しいプライマリに更新します。

VPX 高可用性ペアを Google Cloud Platform にデプロイする方法

NIC が 1 つの HA ペアを導入する手順の概要は次のとおりです。

1. 1 つの VPC ネットワークを作成します。
2. 同じリージョンに 2 つの VPX インスタンス（プライマリノードとセカンダリノード）を作成します。同じゾーンまたは異なるゾーンに配置できます。たとえば、アジア東-1a、アジア東-Ib。
3. NetScaler GUI または ADC CLI コマンドを使用して、両方のインスタンスで高可用性設定を構成します。

手順 1. VPC ネットワークを 1 つ作成

VPC ネットワークを作成するには、次の手順を実行します。

1. **Google** コンソール **>** **[**ネットワーク**]** **>** **[VPC** ネットワーク**]** **>** **[VPC** ネットワークの作成**]** にログインします。

2. 必須フィールドに入力し、[**Create**] をクリックします。

詳細については、「Google Cloud Platform での NetScaler ADC VPX インスタンスのデプロイ」の「**VPC** ネットワークの作成」セクションを参照してください。

手順 **2. 2** つの **VPX** インスタンスを作成する

「シナリオ：単一 NIC のスタンドアロン VPX インスタンスを展開する」に記載されている手順 1 から手順 3 に従って、2 つの VPX インスタンスを作成します。

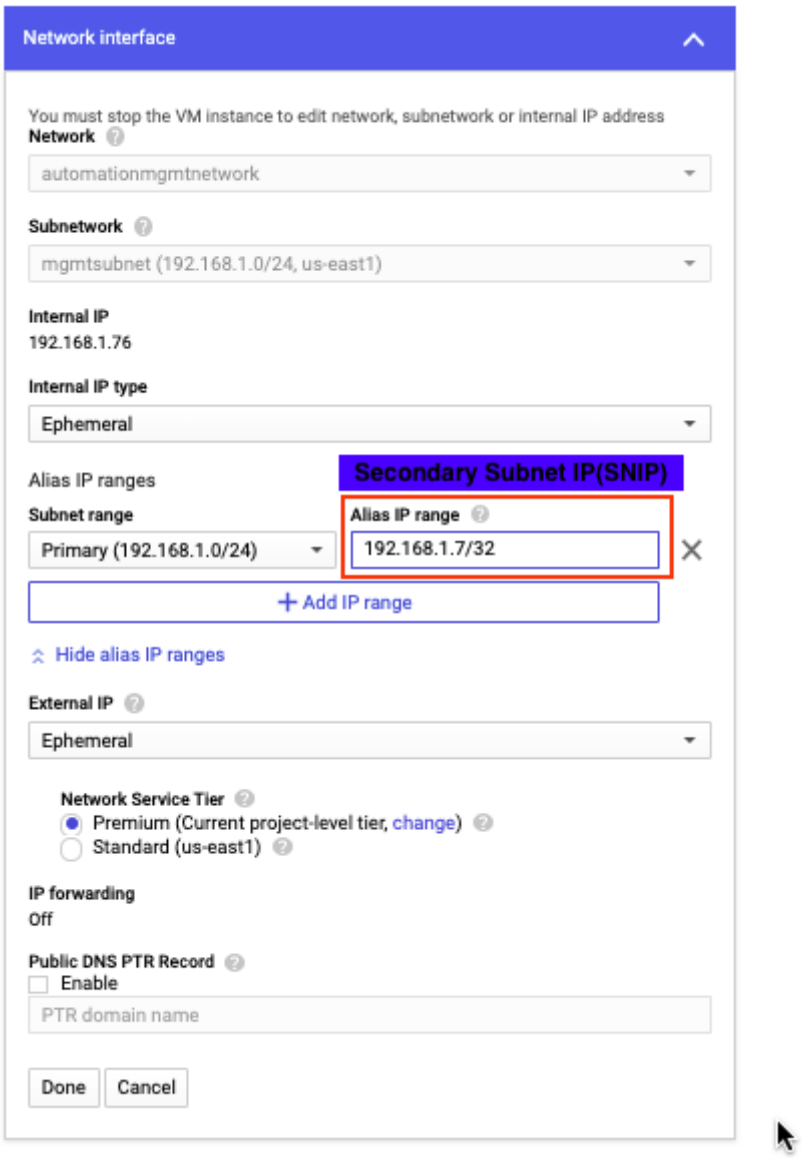
> 重要：
>
> クライアントエイリアス IP アドレスをプライマリノードにのみ割り当て、サーバエイリアス IP アドレスをプライマリノードとセカンダリノードに割り当てます。VPX インスタンスの内部 IP アドレスを使用して VIP または SNIP を構成しないでください。

クライアントとサーバーのエイリアス IP アドレスを作成するには、プライマリノードで次の手順を実行します。

1. VM インスタンスに移動し、[編集] をクリックします。
2. ネットワークインターフェースウィンドウで、クライアント (NIC0) インターフェースを編集します。
3. [エイリアス **IP** 範囲（Alias IP range）] フィールドに、クライアントエイリアス IP アドレスを入力します。
4. 「**IP** 範囲を追加」をクリックし、サーバーのエイリアス IP アドレスを入力します。

サーバエイリアス IP アドレスを作成するには、セカンダリノードで次の手順を実行します。

1. VM インスタンスに移動し、[編集] をクリックします。
2. ネットワークインターフェースウィンドウで、クライアント (NIC0) インターフェースを編集します。
3.「エイリアス **IP** 範囲」フィールドに、サーバーのエイリアス IP アドレスを入力します。



フェイルオーバー後、古いプライマリが新しいセカンダリになると、クライアントのエイリアス IP アドレスが古いプライマリから移動され、新しいプライマリに接続されます。

VPX インスタンスを構成したら、仮想 (VIP) アドレスとサブネット IP (SNIP) アドレスを構成できます。詳細については、「NetScaler ADC 所有の IP アドレスの構成」を参照してください。

手順 **3.** 高可用性を構成する

Google Cloud Platform でインスタンスを作成した後、NetScaler GUI または CLI を使用して高可用性を構成できます。

GUI を使用した高可用性の構成

ステップ **1:** 両方のノードで INC Enabled モードで高可用性を設定します。

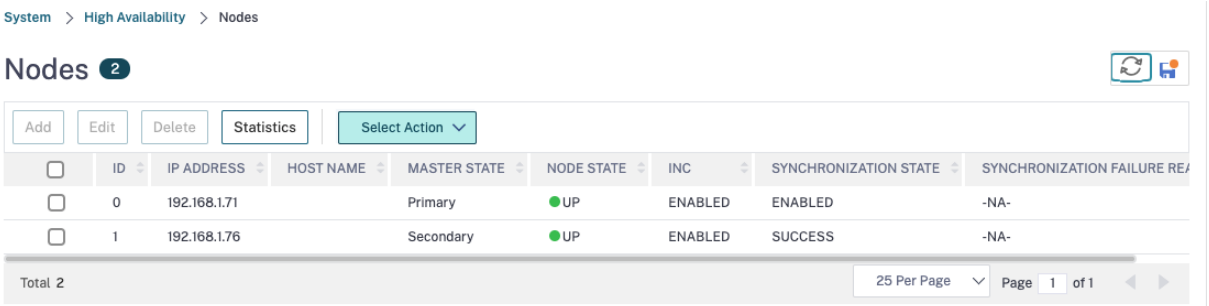
プライマリノードで、次の手順を実行します。

1. GCP Console からノードのユーザー名`nsroot`とインスタンス ID をパスワードとしてインスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノードの **IP** アドレス] フィールドに、セカンダリノードの管理 NIC のプライベート IP アドレスを入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. [作成] をクリックします。

セカンダリノードで、次の手順を実行します。

1. GCP Console からノードのユーザー名`nsroot`とインスタンス ID をパスワードとしてインスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノード **IP** アドレス] フィールドに、プライマリノードの管理 NIC のプライベート IP アドレスを入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. [作成] をクリックします。

先に進む前に、[**Nodes**] ページにセカンダリノードの同期状態が **SUCCESS** と表示されていることを確認してください。



> 注:
>
> セカンダリノードがプライマリノードと同期されると、セカンダリノードにはプライマリノードと同じログオン認証情報が割り当てられます。

ステップ **2:** 両方のノードに仮想 IP アドレスとサブネット IP アドレスを追加します。

プライマリノードで、次の手順を実行します。

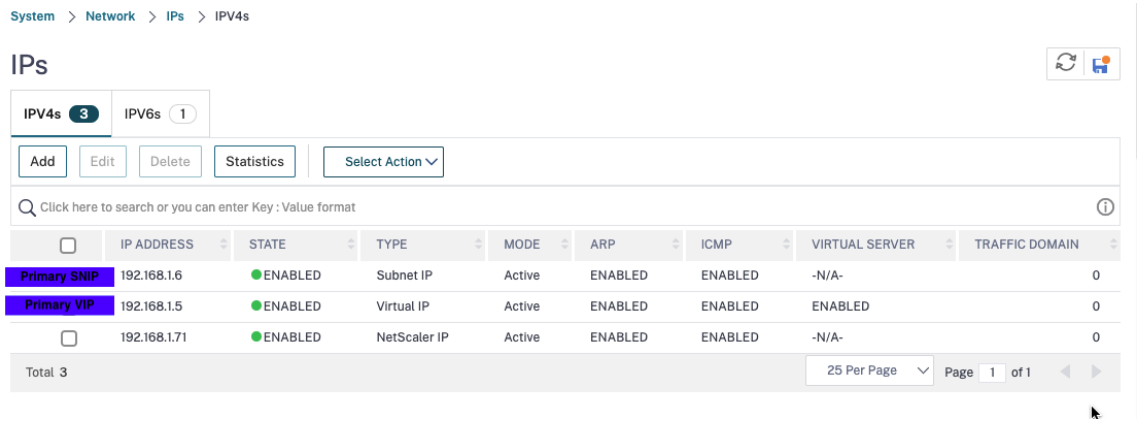
1. [システム**] > [**ネットワーク**] > [IP] > [IPv4**] に移動し、[追加] をクリックします。

2. クライアントエイリアス IP（VIP）アドレスを作成するには、次の手順を実行します。

   a) プライマリ VM インスタンスの VPC サブネットに設定されているクライアントエイリアス IP アドレスとネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
   c) ［作成］をクリックします。

3. サーバーエイリアス IP (SNIP) アドレスを作成するには、次の手順を実行します。

   a) プライマリ VM インスタンスの VPC サブネットに設定されているサーバーエイリアス IP アドレスとネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
   c) ［作成］をクリックします。



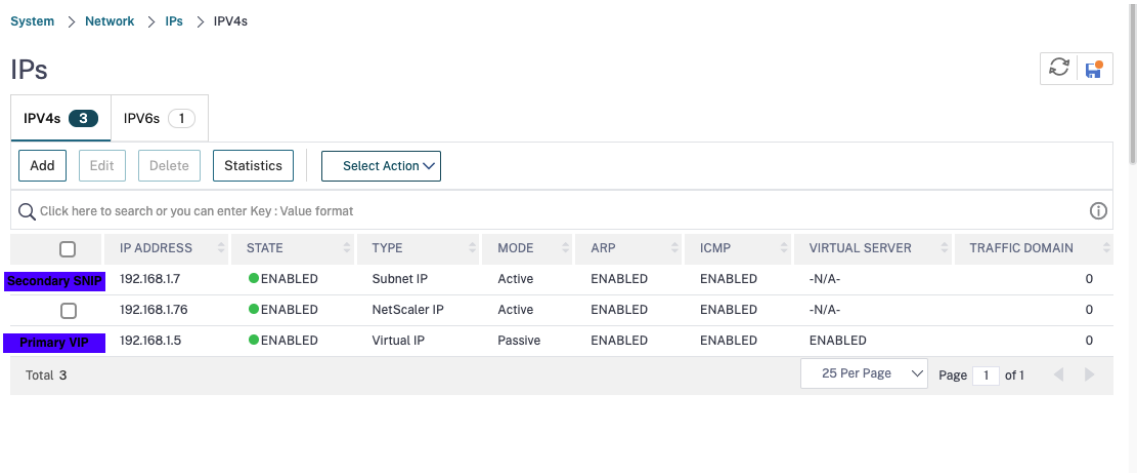
セカンダリノードで、次の手順を実行します。

1. [システム] **>** [ネットワーク] **>** **[IP] > [IPv4]** に移動し、[追加] をクリックします。
2. クライアントエイリアス IP（VIP）アドレスを作成するには、次の手順を実行します。

   a) プライマリ VM インスタンスの VPC サブネットに設定されているクライアントエイリアス IP アドレスとネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
   c) ［作成］をクリックします。

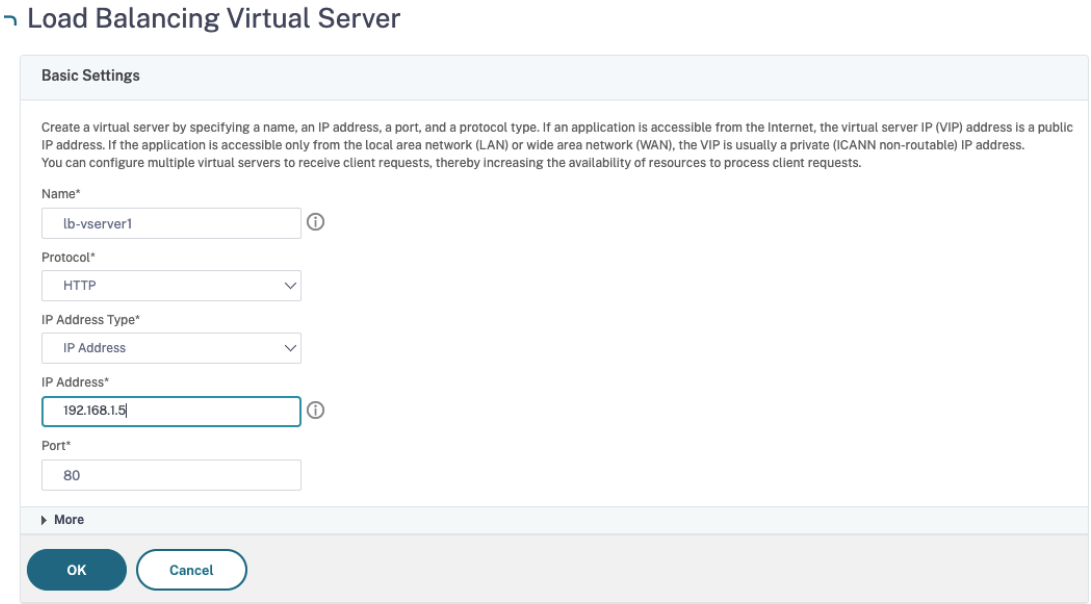
3. サーバーエイリアス IP (SNIP) アドレスを作成するには、次の手順を実行します。

   a) セカンダリ VM インスタンスの VPC サブネットに設定されているサーバエイリアス IP アドレスとネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
   c) ［作成］をクリックします。



ステップ **3:** プライマリノードに負荷分散仮想サーバーを追加します。

1. [設定] **>** [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] **>** [追加] に移動します。
2. 名前、プロトコル、IP アドレスタイプ（IP アドレス）、IP アドレス（プライマリクライアントエイリアス IP アドレス）、およびポートに必要な値を追加し、「**OK**」をクリックします。



ステップ **4:** プライマリノードにサービスまたはサービスグループを追加します。

1. **[**設定**] > [**トラフィック管理**] > [**負荷分散**] > [**サービス**] >[**追加**]** に移動します。
2. サービス名、IP アドレス、プロトコル、およびポートに必要な値を追加し、[**OK**] をクリックします。

ステップ **5:** サービスまたはサービスグループをプライマリノードの負荷分散仮想サーバーにバインドします。

1. **[**設定**] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動します。
2. 手順 **3** で構成した負荷分散仮想サーバーを選択し、[編集] をクリックします。

3. [サービスとサービスグループ] タブで、[負荷分散仮想サーバーサービスバインドなし] をクリックします。
4. 手順 **4** で構成したサービスを選択し、[バインド] をクリックします。

ステップ **6:** 構成を保存します。

強制フェールオーバーの後、セカンダリは新しいプライマリになります。古いプライマリのクライアントエイリアス IP (VIP) が新しいプライマリに移動します。

CLI を使用した高可用性の設定

ステップ **1:** NetScaler CLI を使用して、両方のインスタンスで **INC** 対応モードで高可用性を設定します。

プライマリノードで、次のコマンドを入力します。

```
add ha node 1 <sec_ip> -inc ENABLED
<!--NeedCopy-->
```

セカンダリノードで、次のコマンドを入力します。

```
add ha node 1 <prim_ip> -inc ENABLED
<!--NeedCopy-->
```

sec_ipは、セカンダリノードの管理 NIC の内部 IP アドレスを指します。

prim_ipは、プライマリノードの管理 NIC の内部 IP アドレスを指します。

ステップ **2:** プライマリノードとセカンダリノードの両方に VIP と SNIP を追加します。

プライマリノードで次のコマンドを入力します。

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP

<!--NeedCopy-->
```

> 注:
>
> VM インスタンスのクライアントサブネットに設定されているエイリアス IP アドレスとネットマスクを入力します。

```
add ns ip <primary_server_alias_ip> <subnet> -type SNIP
<!--NeedCopy-->
```

セカンダリノードで次のコマンドを入力します。

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP
<!--NeedCopy-->
```


> 注:
>
> VM インスタンスのクライアントサブネットに設定されているエイリアス IP アドレスとネットマスクを入力します。

```
add ns ip <secondary_server_alias_ip> <subnet> -type SNIP
<!--NeedCopy-->
```

> 注:
>
> VM インスタンスのサーバサブネットに設定されているエイリアス IP アドレスとネットマスクを入力します。

ステップ **3:** プライマリノードに仮想サーバを追加します。

次のコマンドを入力します。

```
add <server_type> vserver <vserver_name> <protocol> <
   primary_client_alias_ip> <port>
<!--NeedCopy-->
```

ステップ **4:** プライマリノードにサービスまたはサービスグループを追加します。

次のコマンドを入力します。

```
add service <service_name> <service_ip_address> <protocol> <port>
<!--NeedCopy-->
```

ステップ **5:** サービスまたはサービスグループをプライマリノードの負荷分散仮想サーバーにバインドします。

次のコマンドを入力します。

```
bind <server_type> vserver <vserver_name> <service_name>
<!--NeedCopy-->
```

> 注:
>
> 設定を保存するには、コマンド`save config`を入力します。そうしないと、インスタンスの再起動後に設定が失われます。

プライベート **IP** アドレスを持つ **VPX** 高可用性ペアを **Google Cloud Platform** にデプロイする

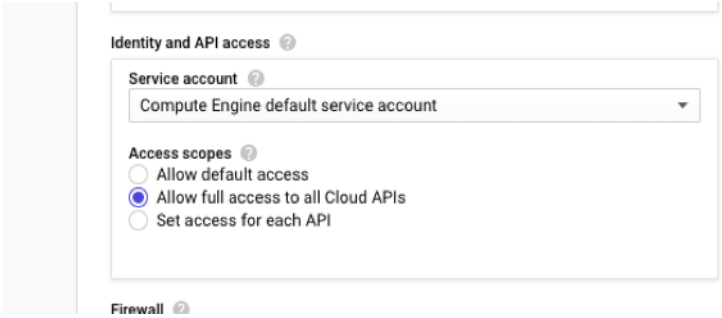
August 15, 2023

プライベート IP アドレスを使用して、VPX 高可用性ペアを GCP にデプロイできます。クライアント IP（VIP）は、プライマリノードのエイリアス IP アドレスとして設定する必要があります。フェールオーバー時に、クライアント IP アドレスがセカンダリノードに移動され、トラフィックが再開されます。

高可用性の詳細については、「高可用性」を参照してください。

はじめに

- Google Cloud Platform への NetScaler ADC VPX インスタンスのデプロイで説明されている制限、ハードウェア要件、注意点をお読みください。この情報は、高可用性展開にも適用されます。
- GCP プロジェクトで クラウドリソースマネージャー **API** を有効にします。
- インスタンスの作成中に、すべての Cloud API へのフルアクセスを許可します。



- GCP サービスアカウントに次の IAM 権限があることを確認します。

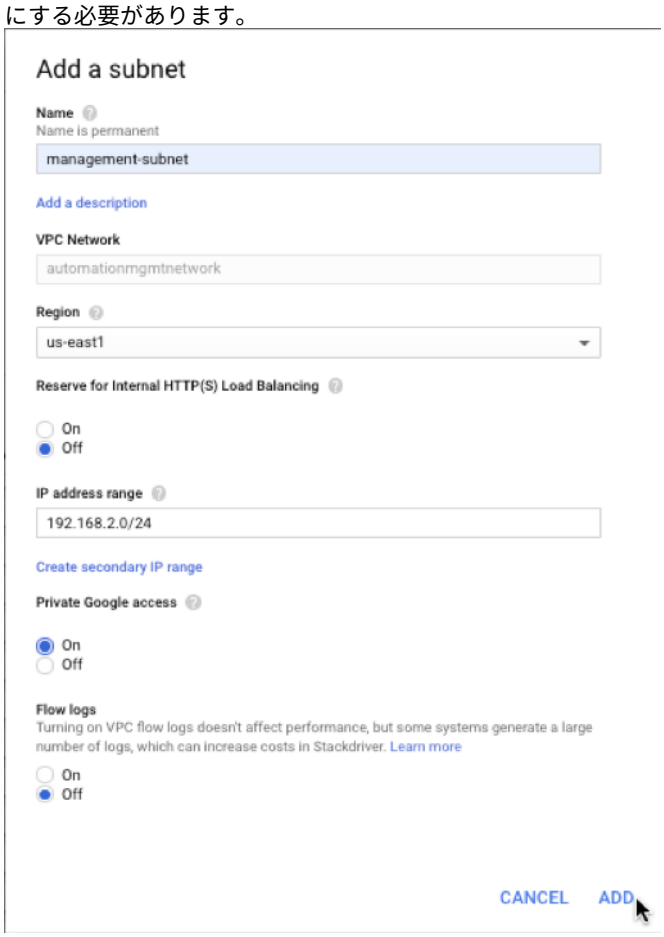
```
REQUIRED_INSTANCE_IAM_PERMS = [
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.setMetadata",
"compute.instances.get",
"compute.instances.list",
"compute.instances.updateNetworkInterface",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.targetInstances.create",
"compute.zones.list",
"compute.zoneOperations.get",
]
<!--NeedCopy-->
```

- 管理インターフェイス以外のインターフェイスに外部 IP アドレスを設定している場合は、GCP サービスアカウントに次の追加の IAM 権限があることを確認します。

```
REQUIRED_INSTANCE_IAM_PERMS = [
"compute.addresses.use"
"compute.instances.addAccessConfig",
"compute.instances.deleteAccessConfig",
"compute.networks.useExternalIp",
"compute.subnetworks.useExternalIp",
]
<!--NeedCopy-->
```

- 仮想マシンにインターネットアクセスがない場合は、管理サブネットでプライベート **Google Access** を有効

にする必要があります。



- プライマリノードで GCP 転送ルールを構成した場合は、「GCP での VPX 高可用性ペアの転送ルールのサポート」に記載されている制限と要件を読み、フェールオーバー時に新しいプライマリに更新します。

VPX 高可用性ペアを Google Cloud Platform にデプロイする方法

ここでは、高可用性展開手順の概要を示します。

1. 同じリージョンに VPC ネットワークを作成します。たとえば、アジア東です。
2. 同じリージョンに 2 つの VPX インスタンス (プライマリノードとセカンダリノード) を作成します。同じゾーンまたは異なるゾーンに配置できます。たとえば、アジア東-1a、アジア東-Ib。
3. NetScaler GUI または ADC CLI コマンドを使用して、両方のインスタンスで高可用性設定を構成します。

手順 1. VPC ネットワークを作成する

要件に基づいて VPC ネットワークを作成します。管理 NIC、クライアント NIC、サーバー NIC に関連付けるために 3 つの VPC ネットワークを作成することをお勧めします。

VPC ネットワークを作成するには、次の手順を実行します。

1. **Google** コンソール **>** ネットワーク **> VPC** ネットワーク **> VPC** ネットワークの作成にログオンします。
2. 必須フィールドに入力し、[**Create**] をクリックします。

詳細については、「Google Cloud Platform での NetScaler ADC VPX インスタンスのデプロイ」の「**VPC** ネットワークの作成」セクションを参照してください。

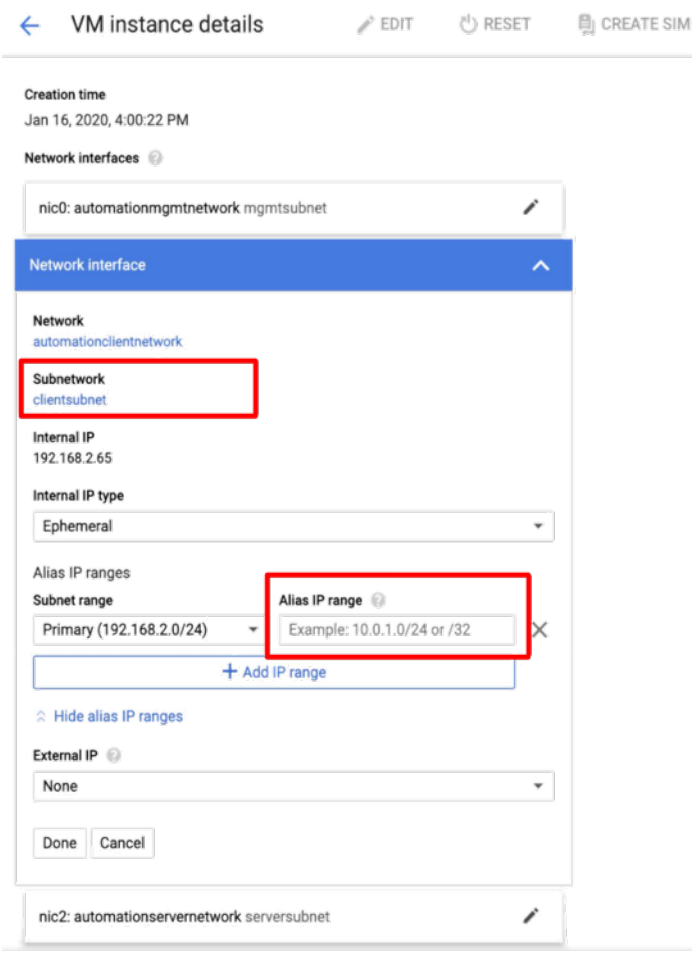
手順 **2. 2** つの **VPX** インスタンスを作成する

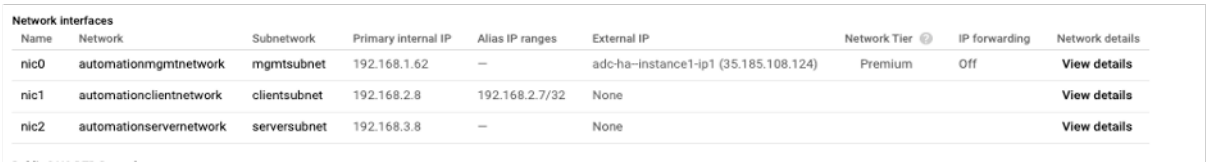
シナリオ：Multi-NIC、Multi-IP スタンドアロン VPX インスタンスを展開するの手順に従って、VPX インスタンスを 2 つ作成します。

> 重要：
>
> クライアントエイリアス IP アドレスをプライマリノードに割り当てます。VPX インスタンスの内部 IP アドレスを使用して VIP を構成しないでください。

クライアントエイリアス IP アドレスを作成するには、次の手順を実行します。

1. VM インスタンスに移動し、[編集] をクリックします。
2. [ネットワークインターフェイス（Network Interface）] ウィンドウで、クライアントインターフェイスを編集します。
3. [エイリアス **IP** 範囲（Alias IP range）] フィールドに、クライアントエイリアス IP アドレスを入力します。





Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	automationmgmtnetwork	mgmtsubnet	192.168.1.62	–	adc-ha--instance1-ip1 (35.185.108.124)	Premium	Off	View details
nic1	automationclientnetwork	clientsubnet	192.168.2.8	192.168.2.7/32	None			View details
nic2	automationservernetwork	serversubnet	192.168.3.8	–	None			View details

フェールオーバー後、古いプライマリが新しいセカンダリになると、エイリアス IP アドレスは古いプライマリから移動し、新しいプライマリに接続されます。

VPX インスタンスを構成したら、仮想 (VIP) アドレスとサブネット IP (SNIP) アドレスを構成できます。詳細については、「NetScaler ADC 所有の IP アドレスの構成」を参照してください。

手順 **3.** 高可用性を構成する

Google Cloud Platform でインスタンスを作成した後、NetScaler GUI または CLI を使用して高可用性を構成できます。

GUI を使用した高可用性の構成

ステップ **1:** 両方のノードで INC Enabled モードで高可用性を設定します。

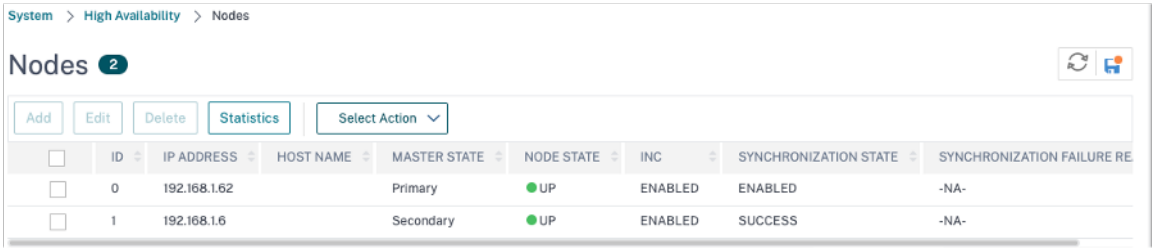
プライマリノードで、次の手順を実行します。

1. GCP Console からノードのユーザー名nsrootとインスタンス ID をパスワードとしてインスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノードの **IP** アドレス] フィールドに、セカンダリノードの管理 NIC のプライベート IP アドレスを入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. [作成] をクリックします。

セカンダリノードで、次の手順を実行します。

1. GCP Console からノードのユーザー名nsrootとインスタンス ID をパスワードとしてインスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノード **IP** アドレス] フィールドに、プライマリノードの管理 NIC のプライベート IP アドレスを入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. [作成] をクリックします。

先に進む前に、[**Nodes**] ページにセカンダリノードの同期状態が **SUCCESS** と表示されていることを確認してください。



> 注
>
> セカンダリノードがプライマリノードと同期されると、セカンダリノードにはプライマリノードと同じログオン認証情報が割り当てられます。

ステップ **2:** 両方のノードに仮想 IP アドレスとサブネット IP アドレスを追加します。

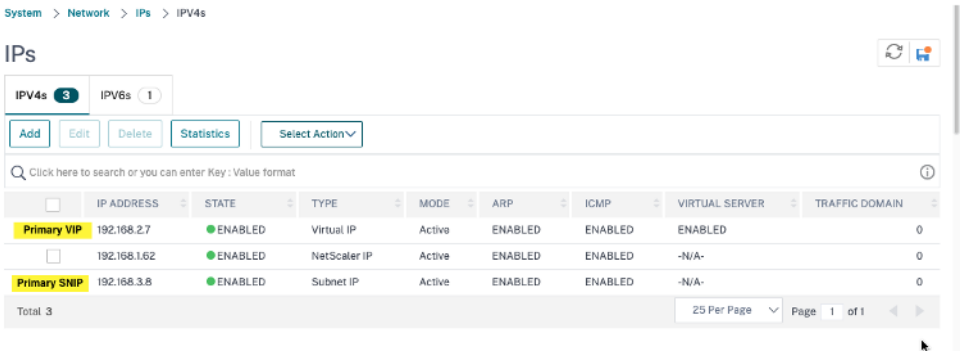
プライマリノードで、次の手順を実行します。

1. [システム**] > [**ネットワーク**] > [IP] > [IPv4**] に移動し、[追加] をクリックします。
2. クライアントエイリアス IP（VIP）アドレスを作成するには、次の手順を実行します。

a) 仮想マシンインスタンスのクライアントサブネットに設定されたエイリアス IP アドレスとネットマスクを入力します。
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
c) ［作成］をクリックします。

3. サーバ IP (SNIP) アドレスを作成するには、次の手順を実行します。

a) プライマリ・インスタンスのサーバ側インターフェイスの内部 IP アドレスと、サーバ・サブネットに設定されたネットマスクを入力します。
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
c) ［作成］をクリックします。



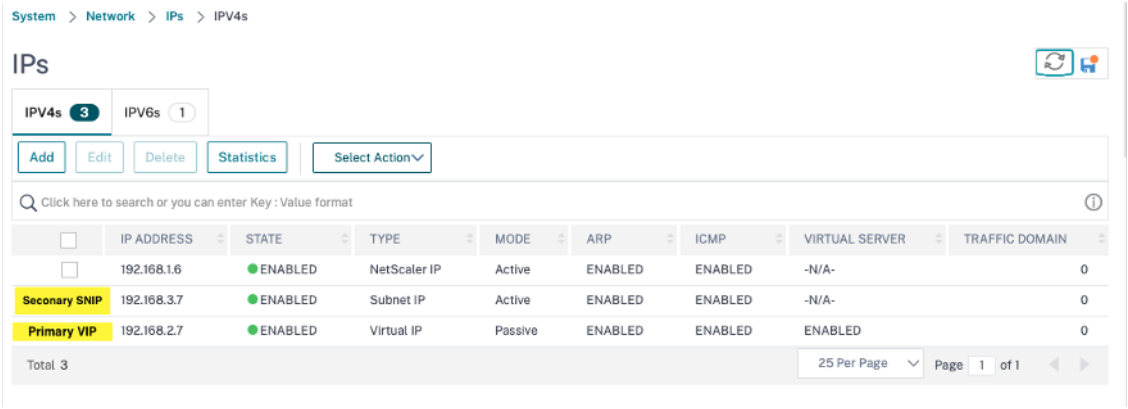
セカンダリノードで、次の手順を実行します。

1. [システム] **>** [ネットワーク] **>** **[IP] > [IPv4]** に移動し、[追加] をクリックします。
2. クライアントエイリアス IP（VIP）アドレスを作成するには、次の手順を実行します。

a) プライマリ VM インスタンスのクライアントサブネットに対して構成された Alias IP アドレスとネットマスクを入力します。
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
c) ［作成］をクリックします。

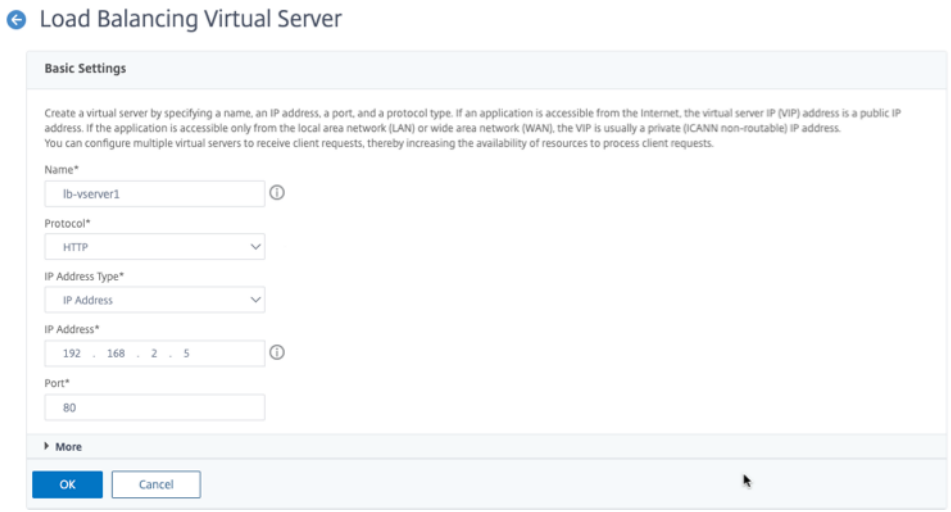
3. サーバ IP (SNIP) アドレスを作成するには、次の手順を実行します。

a) セカンダリインスタンスのサーバ側インターフェイスの内部 IP アドレスと、サーバサブネットに設定されたネットマスクを入力します。
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
c) ［作成］をクリックします。



ステップ **3:** プライマリノードに負荷分散仮想サーバーを追加します。

1. [設定] **>** [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] **>** [追加] に移動します。
2. 名前、プロトコル、IP アドレスタイプ（IP アドレス）、IP アドレス（プライマリクライアントエイリアス IP アドレス）、およびポートに必要な値を追加し、「**OK**」をクリックします。



ステップ **4:** プライマリノードにサービスまたはサービスグループを追加します。

1. **[**設定**] > [**トラフィック管理**] > [**負荷分散**] > [**サービス**] >[**追加**]** に移動します。
2. サービス名、IP アドレス、プロトコル、およびポートに必要な値を追加し、[**OK**] をクリックします。

ステップ **5:** サービスまたはサービスグループをプライマリノードの負荷分散仮想サーバーにバインドします。

1. **[**設定**] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動します。
2. 手順 **3** で構成した負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [サービスとサービスグループ] タブで、[負荷分散仮想サーバーサービスバインドなし] をクリックします。
4. 手順 **4** で構成したサービスを選択し、[バインド] をクリックします。

ステップ **5:** 構成を保存します。

強制フェールオーバーの後、セカンダリは新しいプライマリになります。古いプライマリからのクライアントエイリアス IP (VIP) とサーバエイリアス IP (SNIP) が新しいプライマリに移動します。

CLI を使用した高可用性の設定

ステップ **1:** NetScaler CLI を使用して、両方のインスタンスで **INC** 対応モードで高可用性を設定します。

プライマリノードで、次のコマンドを入力します。

```
add ha node 1 <sec_ip> -inc ENABLED
<!--NeedCopy-->
```

セカンダリノードで、次のコマンドを入力します。

```
add ha node 1 <prim_ip> -inc ENABLED
<!--NeedCopy-->
```

sec_ipは、セカンダリノードの管理 NIC の内部 IP アドレスを指します。

prim_ipは、プライマリノードの管理 NIC の内部 IP アドレスを指します。

ステップ **2:** 両方のノードに VIP と SNIP を追加します。

プライマリノードで次のコマンドを入力します。

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP

<!--NeedCopy-->
```

> 注:
>
> 仮想マシンインスタンスのクライアントサブネットに設定されたエイリアス IP アドレスとネットマスクを入力します。

```
add ns ip <primary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

primary_snipは、プライマリインスタンスのサーバ側インターフェイスの内部 IP アドレスを指します。

セカンダリノードで次のコマンドを入力します。

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP
<!--NeedCopy-->
```

> 注
>
> プライマリ VM インスタンスのクライアントサブネットに対して構成された Alias IP アドレスとネットマスクを入力します。


```
add ns ip <secondary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

secondary_snipは、セカンダリインスタンスのサーバ側インターフェイスの内部 IP アドレスを指します。

> 注:
>
> VM インスタンスのサーバサブネットに設定された IP アドレスとネットマスクを入力します。

ステップ **3:** プライマリノードに仮想サーバを追加します。

次のコマンドを入力します。

```
add <server_type> vserver <vserver_name> <protocol> <
   primary_client_alias_ip> <port>
<!--NeedCopy-->
```

ステップ **4:** プライマリノードにサービスまたはサービスグループを追加します。

次のコマンドを入力します。

```
add service <service_name> <service_ip_address> <protocol> <port>
<!--NeedCopy-->
```

ステップ **5:** サービスまたはサービスグループをプライマリノードの負荷分散仮想サーバーにバインドします。

次のコマンドを入力します。

```
bind <server_type> vserver <vserver_name> <service_name>
<!--NeedCopy-->
```

> 注:
>
> 設定を保存するには、コマンドsave configを入力します。そうしないと、インスタンスの再起動後に設定が失われます。

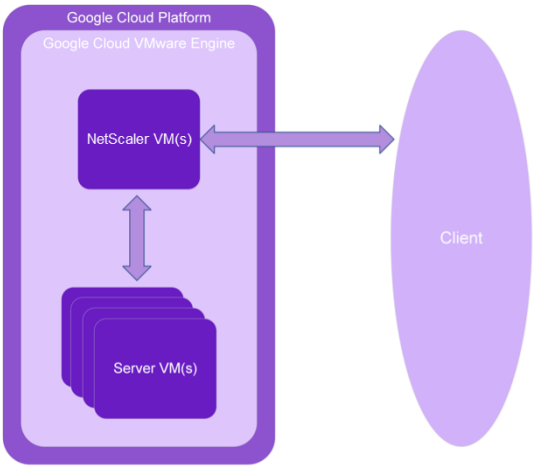
Google Cloud VMware Engine に NetScaler VPX インスタンスをインストールする

December 8, 2023

Google Cloud VMware Engine（GCVE）は、専用のベアメタルの Google Cloud Platform インフラストラクチャから構築された vSphere クラスタを含むプライベートクラウドを提供します。最小初期デプロイメントは 3 ホストですが、追加ホストは一度に 1 つずつ追加できます。プロビジョニングされたすべてのプライベートクラウドには、vCenter Server、vSAN、vSphere、NSX-T があります。

GCVE を使用すると、必要な数の ESX ホストを使用して Google Cloud Platform 上にクラウドソフトウェア定義データセンター（SDDC）を作成できます。GCVE は NetScaler VPX の導入をサポートします。GCVE はオンプレミス vCenter と同じユーザーインターフェイスを提供します。ESX ベースの NetScaler VPX デプロイメントと同じように機能します。

次の図は、管理者またはクライアントがインターネット経由でアクセスできる Google Cloud Platform 上の GCVE を示しています。管理者は GCVE を使用してワークロードまたはサーバー VM を作成、管理、構成できます。管理者は OpenVPN 接続を使用して GCVE のウェブベースの vCenter と NSX-T Manager にアクセスできます。vCenter を使用して GCVE 内に NetScaler VPX インスタンス（スタンドアロンまたは HA ペア）とサーバ仮想マシンを作成し、NSX-T Manager を使用して対応するネットワークを管理できます。GCVE 上の NetScaler VPX インスタンスは、オンプレミスの VMware ホストクラスタと同様に機能します。GCVE は、管理インフラストラクチャへの OpenVPN 接続を使用して管理できます。



前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- Google Cloud VMware エンジンとその前提条件の詳細については、Google Cloud VMware エンジンのドキュメントを参照してください。
- Google Cloud VMware Engine のデプロイに関する詳細については、「Google Cloud VMware Engine プライベートクラウドのデプロイ」を参照してください。
- ポイントツーサイト VPN ゲートウェイを使用してプライベートクラウドに接続し、Google Cloud VMware Engine にアクセスして管理する方法の詳細については、「Google Cloud VMware Engine プライベートクラウドへのアクセス」を参照してください。
- VPN クライアントマシンで、NetScaler VPX アプライアンスのセットアップファイルをダウンロードします。
- 仮想マシンが接続する VMware SDDC 上に、適切な NSX-T ネットワークセグメントを作成します。詳細については、「Google Cloud VMware Engine でのネットワークセグメントの追加」を参照してください。
- VPX ライセンスファイルを入手します。NetScaler VPX インスタンスライセンスの詳細については、「ライセンスの概要」を参照してください。

- GCVE プライベートクラウドに作成または移行された仮想マシン (VM) は、ネットワークセグメントに接続する必要があります。

VMware クラウドのハードウェア要件

次の表に、VMware SDDC が各 VPX nCore 仮想アプライアンスに対して提供する必要がある仮想コンピューティングリソースを示します。

表 1. NetScaler VPX インスタンスの実行に必要な最小限の仮想コンピューティングリソース

コンポーネント	条件
メモリ	2 GB
仮想 CPU（VCPU）	2
仮想ネットワークインターフェイス	VMware SDDC では、VPX ハードウェアをバージョン 7 以降にアップグレードした場合、最大 10 個の仮想ネットワークインターフェイスをインストールできます。
ディスク領域	20 GB

> 注
>
> ハイパーバイザーに必要なディスク領域は含まれません。

VPX 仮想アプライアンスを実稼働で使用するには、フルメモリ割り当てを予約する必要があります。

OVF ツール 1.0 のシステム要件

Open Virtualization Format Tool（OVF Tool）は、Windows および Linux システムで実行できるクライアントアプリケーションです。次の表は、OVF ツールをインストールするための最小システム要件を示しています。

表 2. OVF ツールのインストールに必要な最小システム要件

コンポーネント	条件
オペレーティングシステム	VMware からの詳細な要件については、http://kb.vmware.com/で『OVF ツールユーザーガイド』の PDF ファイルを検索してください。
CPU	最低 750MHz、1GHz 以上推奨
RAM	最小 1 GB、推奨 2 GB
NIC	100Mbps 以上の NIC。

OVF のインストールについては、http://kb.vmware.com/で『OVF ツールユーザーガイド』の PDF ファイルを検索してください。

NetScaler VPX セットアップファイルのダウンロード

VMware ESX 用の NetScaler VPX インスタンスセットアップパッケージは、オープン仮想マシン（OVF）フォーマット標準に準拠しています。これらのファイルは、Citrix Web サイトからダウンロードできます。ログオンするには、Citrix アカウントが必要です。Citrix アカウントをお持ちでない場合は、http://www.citrix.comのホームページにアクセスしてください。［新しいユーザー］リンクをクリックし、指示に従って新しい Citrix アカウントを作成します。

ログオンしたら、Citrix のホームページから次のパスをナビゲートします。

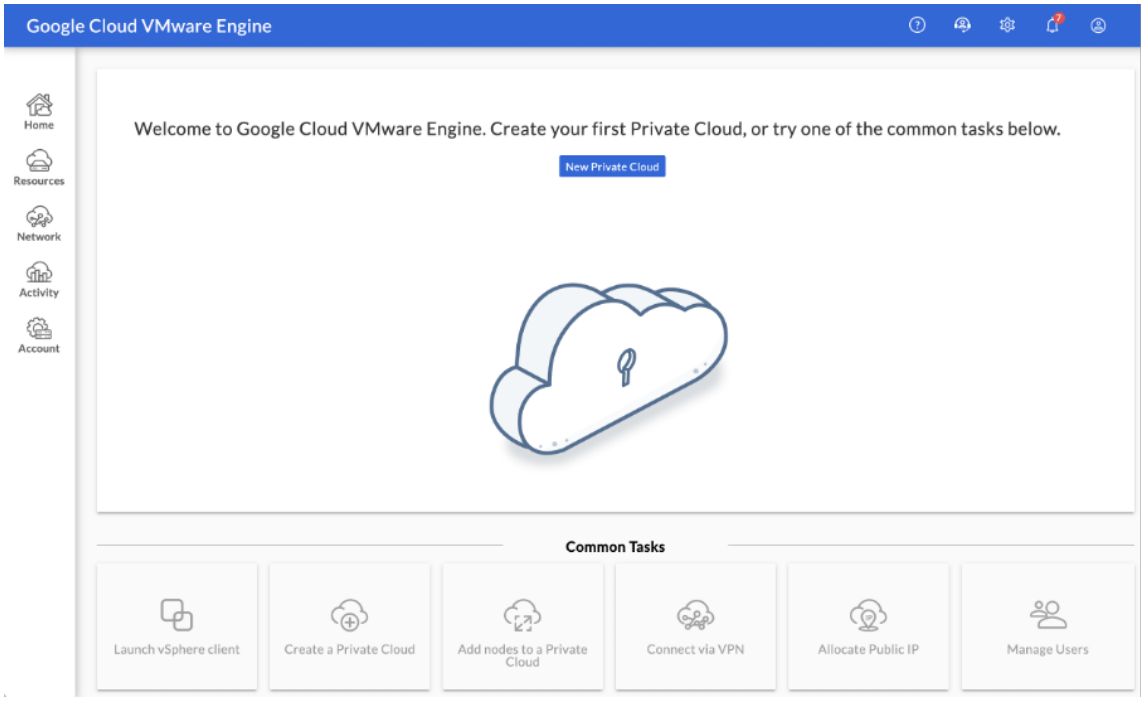
Citrix.com > ダウンロード **> NetScaler >** 仮想アプライアンス。

次のファイルを、ESX サーバーと同じネットワーク上のワークステーションにコピーします。3 つのファイルをすべて同じフォルダーにコピーします。

- NSVPX-ESX-<release number>--disk1.vmdk <build number> (たとえば、nsvpx-ESX-13.0-79.64-disk1.vmdk)
- NSVPX-ESX-<release number>-.ovf <build number> (たとえば、nsvpx-ESX-13.0-79.64.OVF)
- NSVPX-ESX-<release number>-.mf <build number> (たとえば、nsvpx-ESX-13.0-79.64.mf)

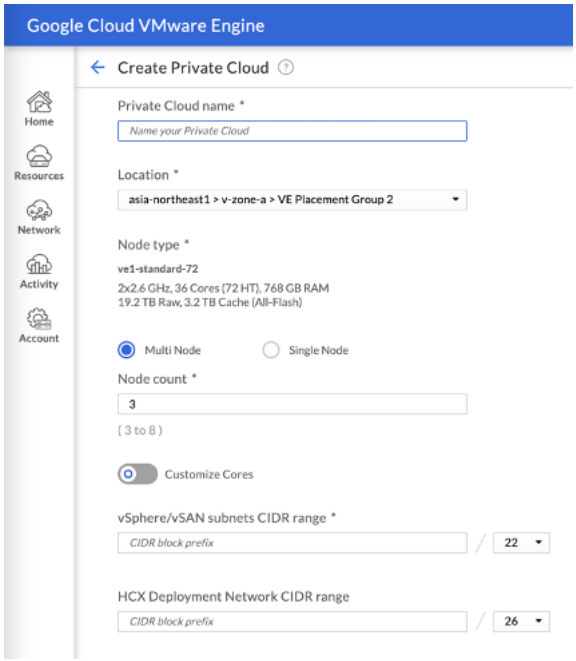
Google Cloud VMware エンジンをデプロイする

1. GCVE ポータルにログインし、ホームに移動します。



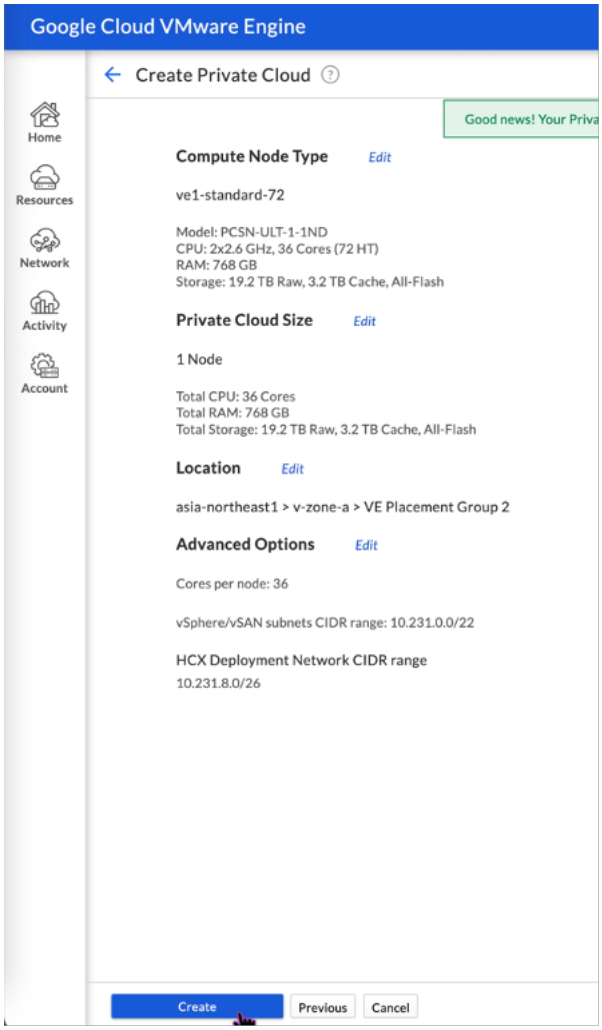
2. 「新規プライベートクラウド」ページで、次の詳細を入力します。

   - プライベートクラウドのデフォルトクラスタを作成するには、最低 3 つの ESXi ホストを選択します。
   - **vSphere/vSAN** サブネットの **CIDR** 範囲フィールドには、/22 アドレススペースを使用します。
   - **HCX** デプロイメントネットワーク **CIDR** 範囲フィールドには、/26 アドレススペースを使用します。
   - 仮想ネットワークの場合、CIDR 範囲がオンプレミスまたは他の GCP サブネット (仮想ネットワーク) と重複していないことを確認します。



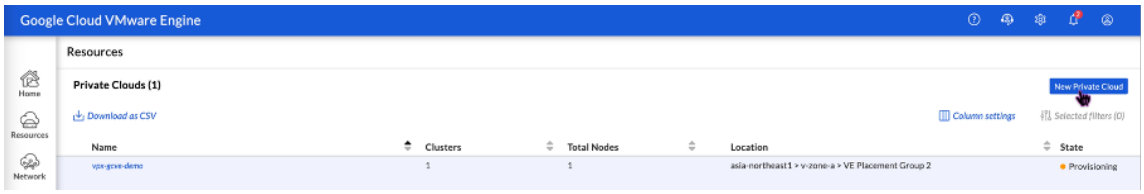
3. [確認して作成] をクリックします。

4. 設定を確認します。設定を変更する必要がある場合は、[前へ] をクリックします。



5. [作成] をクリックします。プライベートクラウドのプロビジョニングプロセスが開始されます。プライベートクラウドのプロビジョニングには最大 2 時間かかることがあります。

6. 「リソース」に移動して、作成されたプライベートクラウドを確認します。

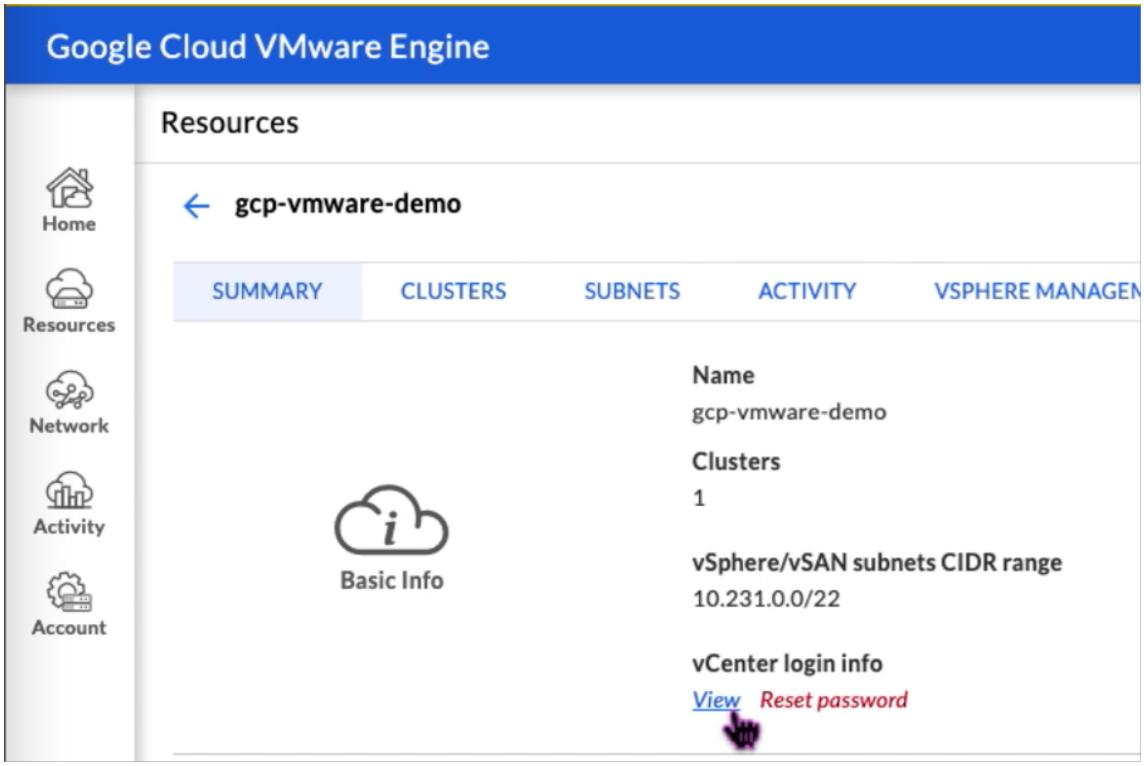


7. このリソースにアクセスするには、ポイントツーサイト VPN を使用して GCVE に接続する必要があります。詳細については、次のドキュメントを参照してください。

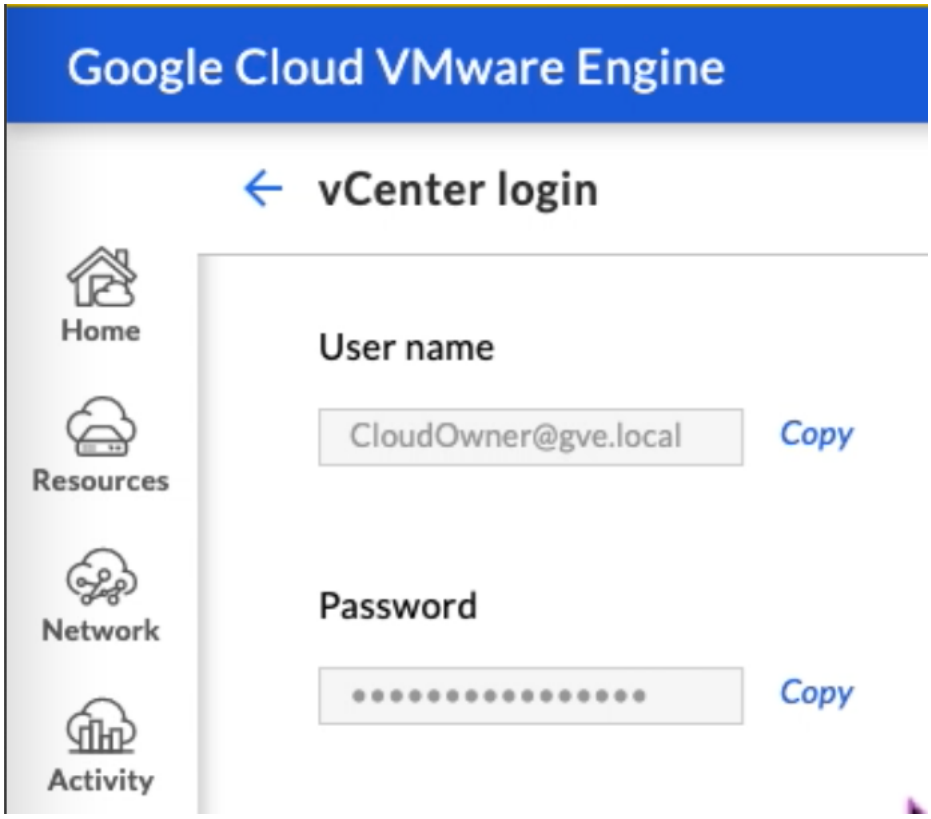
   - VPN ゲートウェイ
   - VPN を使用して接続する

プライベートクラウド **vCenter** ポータルにアクセスする

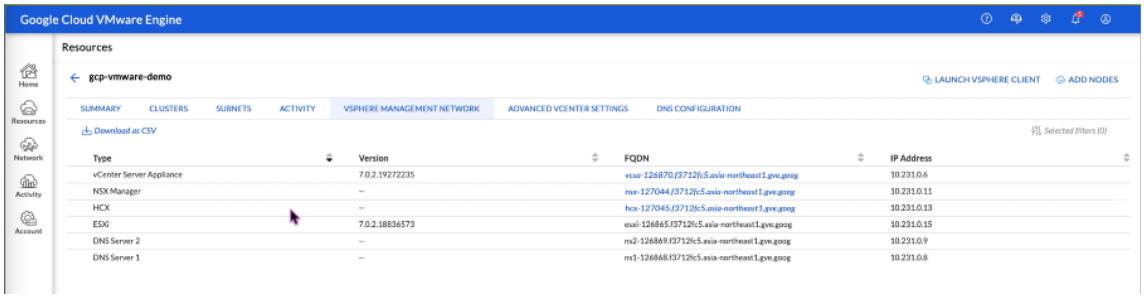
1. Google Cloud VMware Engine プライベートクラウドに移動します。[概要] タブの [**vCenter** ログイン情報] で、[表示] をクリックします。



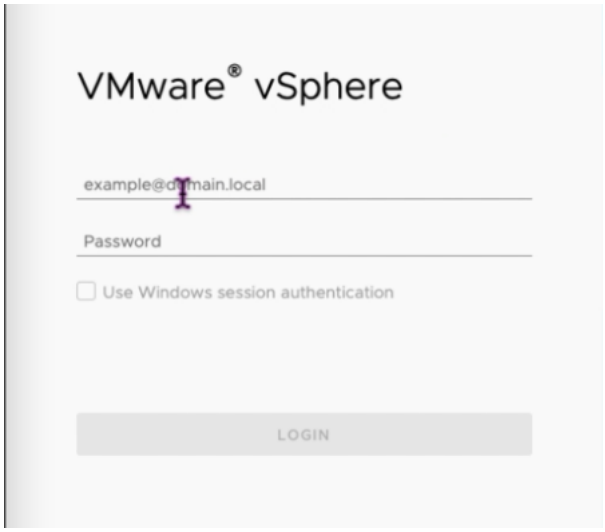
2. vCenter の認証情報を書き留めます。



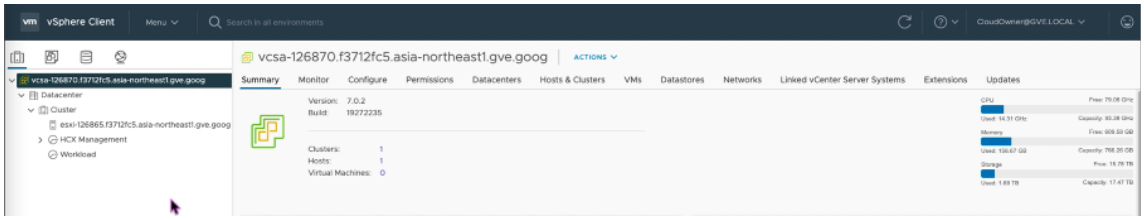
3. vSPHERE CLIENT の起動をクリックして **vSphere** クライアントを起動するか、**VSPHERE** 管理ネットワークに移動して **vCenter Server** アプライアンスの FQDN をクリックします。



4. この手順のステップ 2 でメモした vCenter 認証情報を使用して VMware vSphere にログインします。



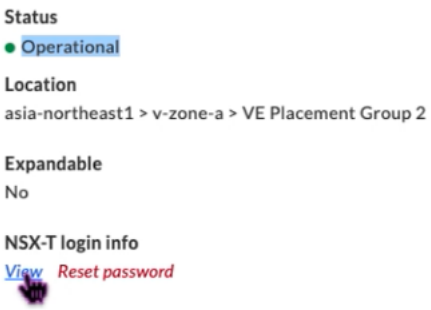
5. vSphere クライアントでは、GCVE ポータルで作成した ESXi ホストを確認できます。



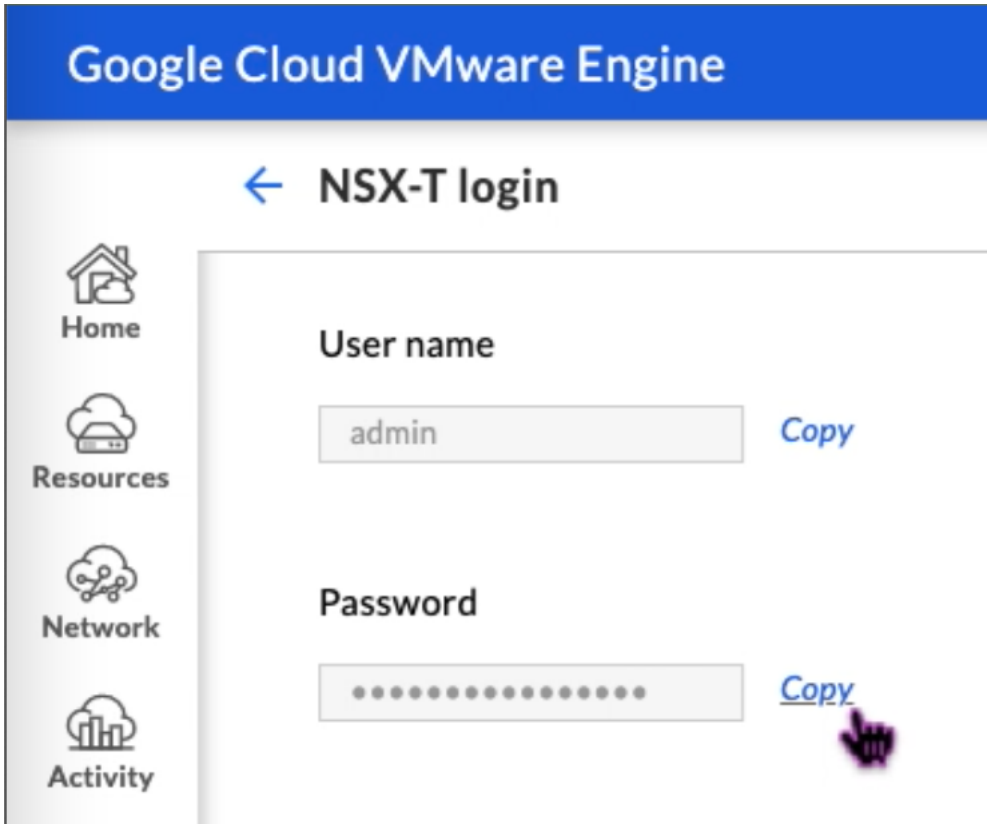
GCVE NSX-T ポータルで **NSX-T** セグメントを作成します

NSX-T セグメントは、Google Cloud VMware エンジンコンソールの NSX マネージャから作成および設定できます。これらのセグメントはデフォルトの Tier-1 ゲートウェイに接続され、これらのセグメントのワークロードは East-West および North-South 接続を取得します。セグメントを作成すると、vCenter に表示されます。

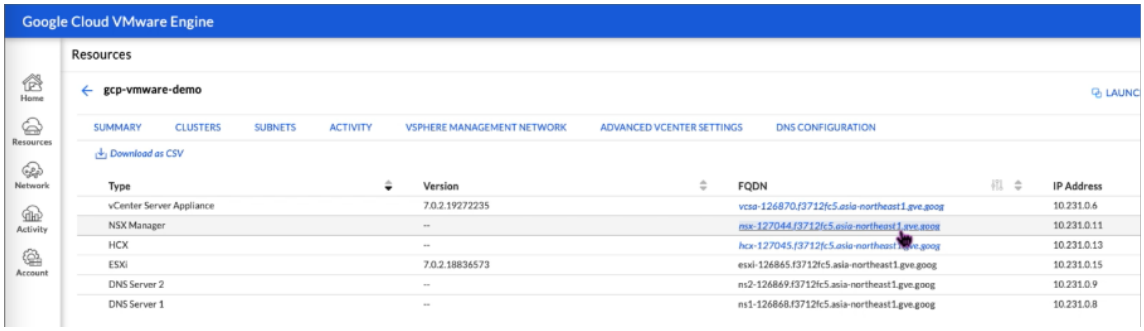
1. GCVE プライベートクラウドの **[概要]-> [NSX-T** ログイン情報] で、[表示] を選択します。



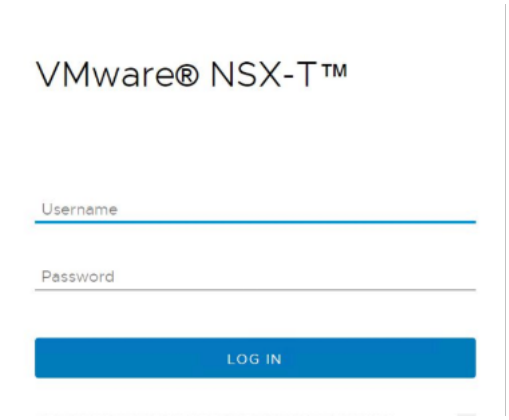
2. NSX-T の認証情報を書き留めておきます。



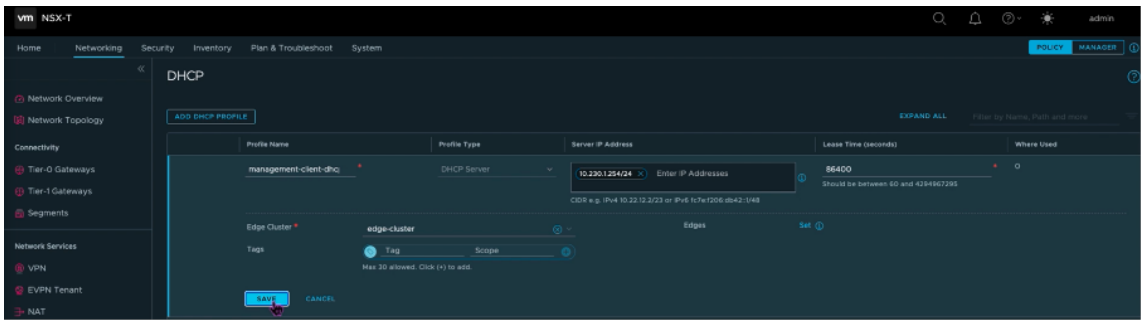
3. [**VSPHERE** 管理ネットワーク] に移動して NSX Manager を起動し、**NSX Manager** の FQDN をクリックします。



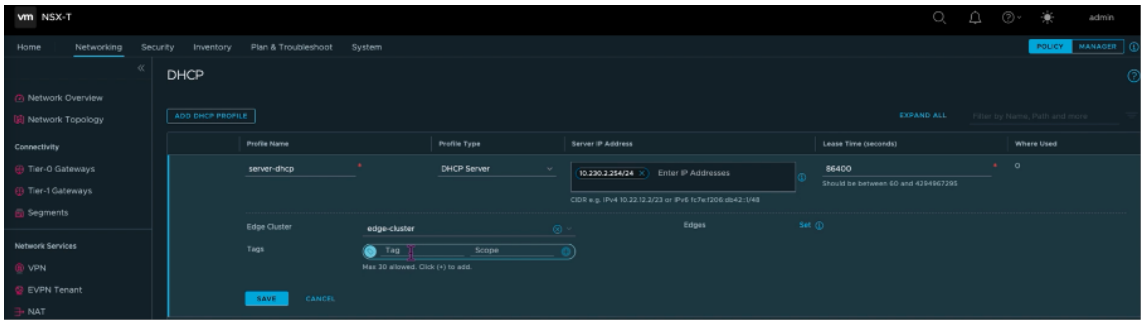
4. この手順のステップ 2 でメモした認証情報を使用して NSX Manager にログインします。



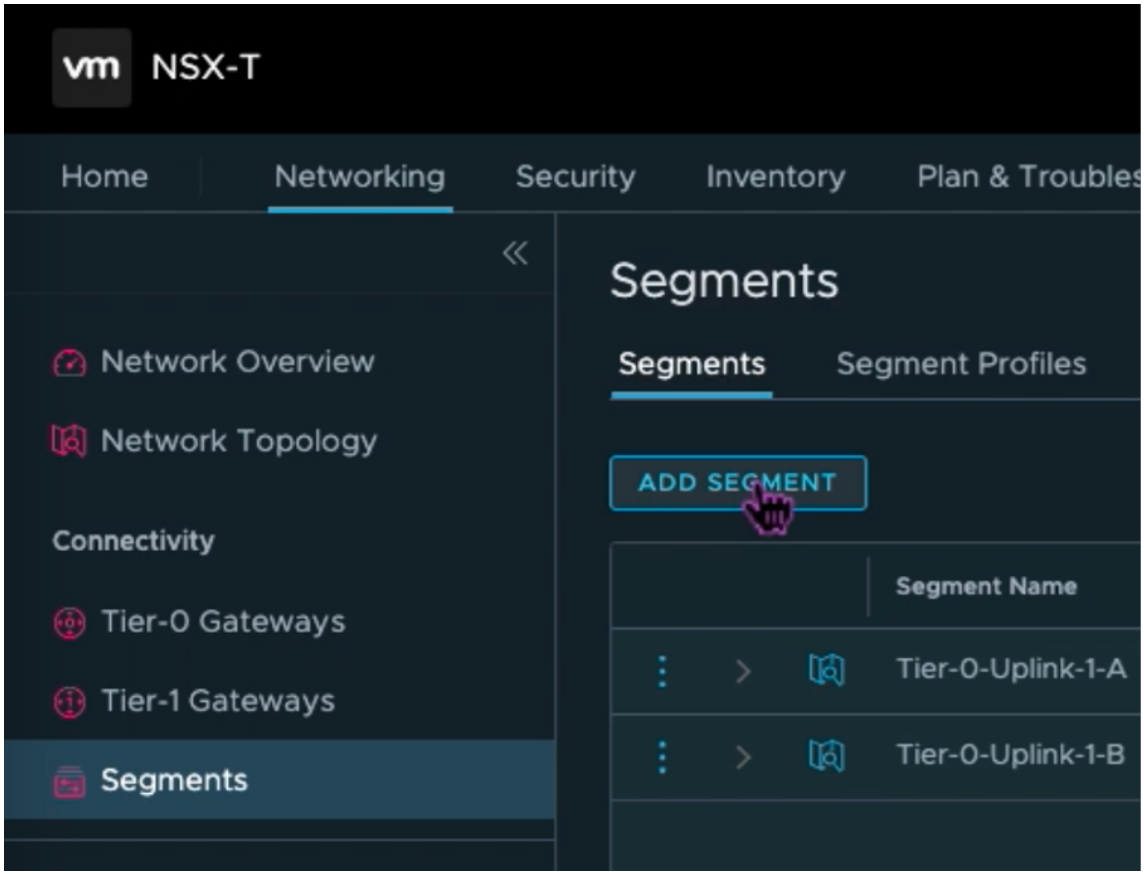
5. 新しいセグメントまたはサブネットの DHCP サービスを設定します。
6. サブネットを作成する前に、DHCP サービスを設定します。
7. NSX-T で、[ネットワーク] **> [DHCP]** に移動します。ネットワークダッシュボードには、サービスが Tier-0 ゲートウェイを 1 つと Tier-1 ゲートウェイを 1 つ作成していることがわかります。
8. DHCP サーバーのプロビジョニングを開始するには、「**DHCP** プロファイルの追加」をクリックします。
9. DHCP 名フィールドに、クライアント管理プロファイルの名前を入力します。
10. プロファイルタイプとして **DHCP** サーバーを選択します。
11. 「サーバー **IP** アドレス」列に、DHCP サービスの IP アドレス範囲を指定します。
12. **Edge** クラスタを選択します。
13. [**Save**] をクリックして、DHCP サービスを作成します。



14. サーバの DHCP 範囲について、手順 6～13 を繰り返します。



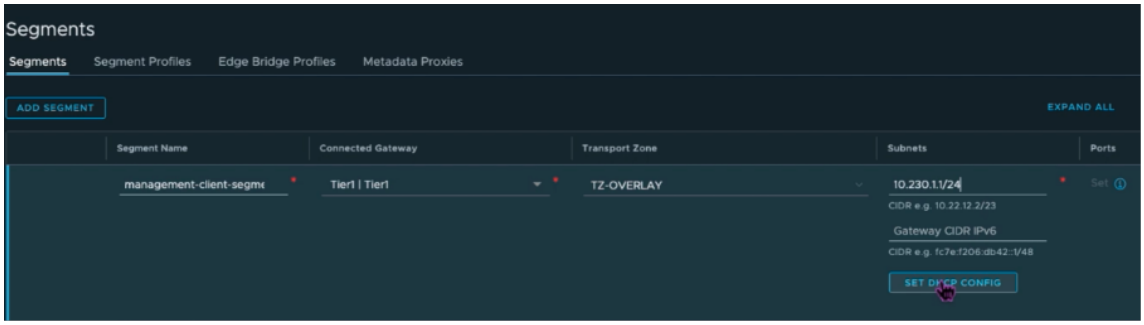
15. 2 つのセグメントを作成します。1 つはクライアントと管理インターフェイス用、もう 1 つはサーバーインターフェイス用です。
16. NSX-T で、[ネットワーク**] > [**セグメント**]** に移動します。
17. [**Add Segment**] をクリックします。



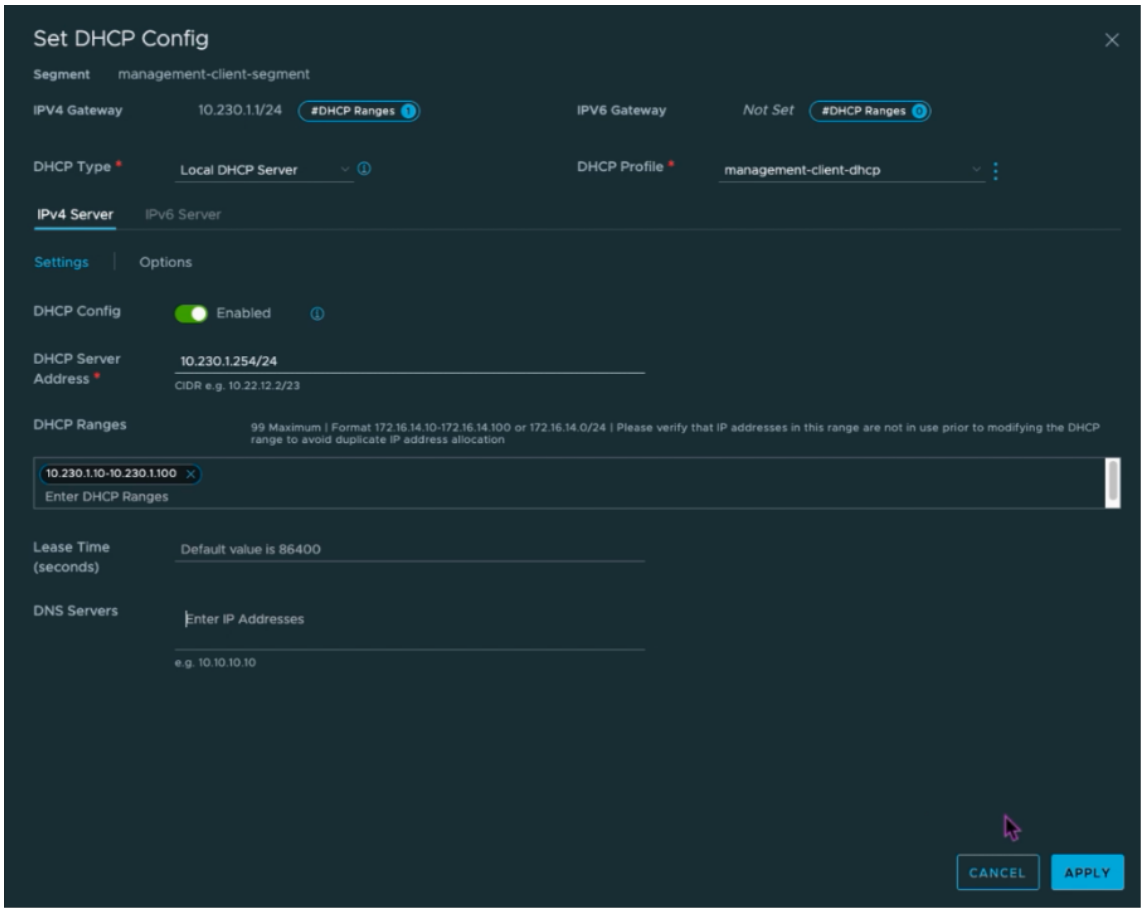
18. セグメント名フィールドに、クライアント管理セグメントの名前を入力します。
19. 接続されたゲートウェイリストで、**Tier1** を選択して Tier-1 ゲートウェイに接続します。

「トランスポートゾーン」リストで「**TZ-OVERLAY」を　オーバーレイ **。
選択します

20.
21. [サブネット] 列に、サブネット範囲を入力します。サブネット範囲で **.** 1を最後のオクテットとして指定します。例：10.12.2.1/24。

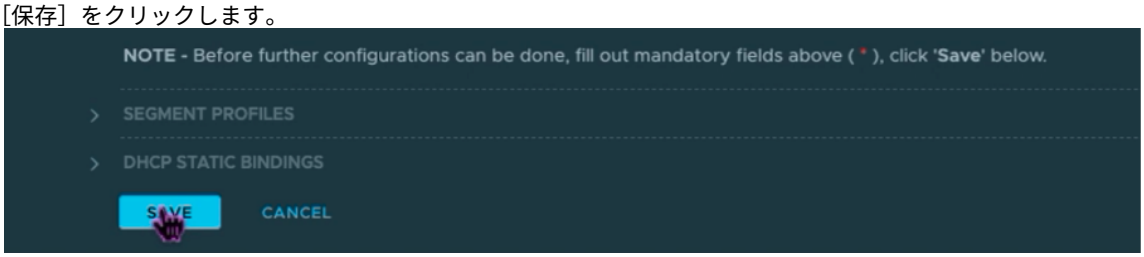


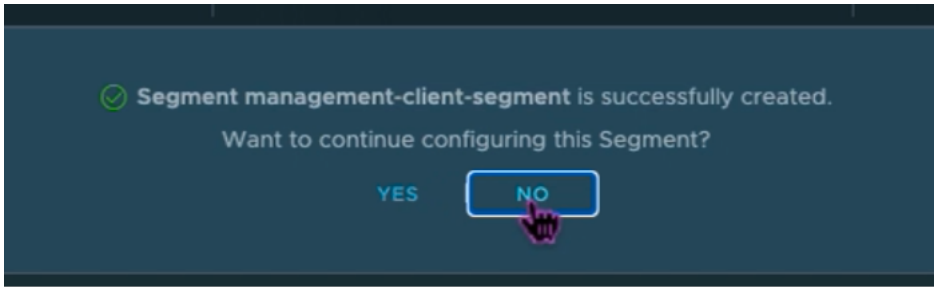
22.「**DHCP** 構成を設定」をクリックし、「**DHCP** 範囲」フィールドに値を入力します。



23. [**Apply**] をクリックして DHCP 設定を保存します。

24.［保存］をクリックします。





25. サーバーセグメントについても手順 17 ～24 を繰り返します。
26. 仮想マシンの作成時に vCenter でこれらのネットワークセグメントを選択できるようになりました。

詳細については、「最初のサブネットの作成」を参照してください。

VMware クラウドへの NetScaler VPX インスタンスのインストール

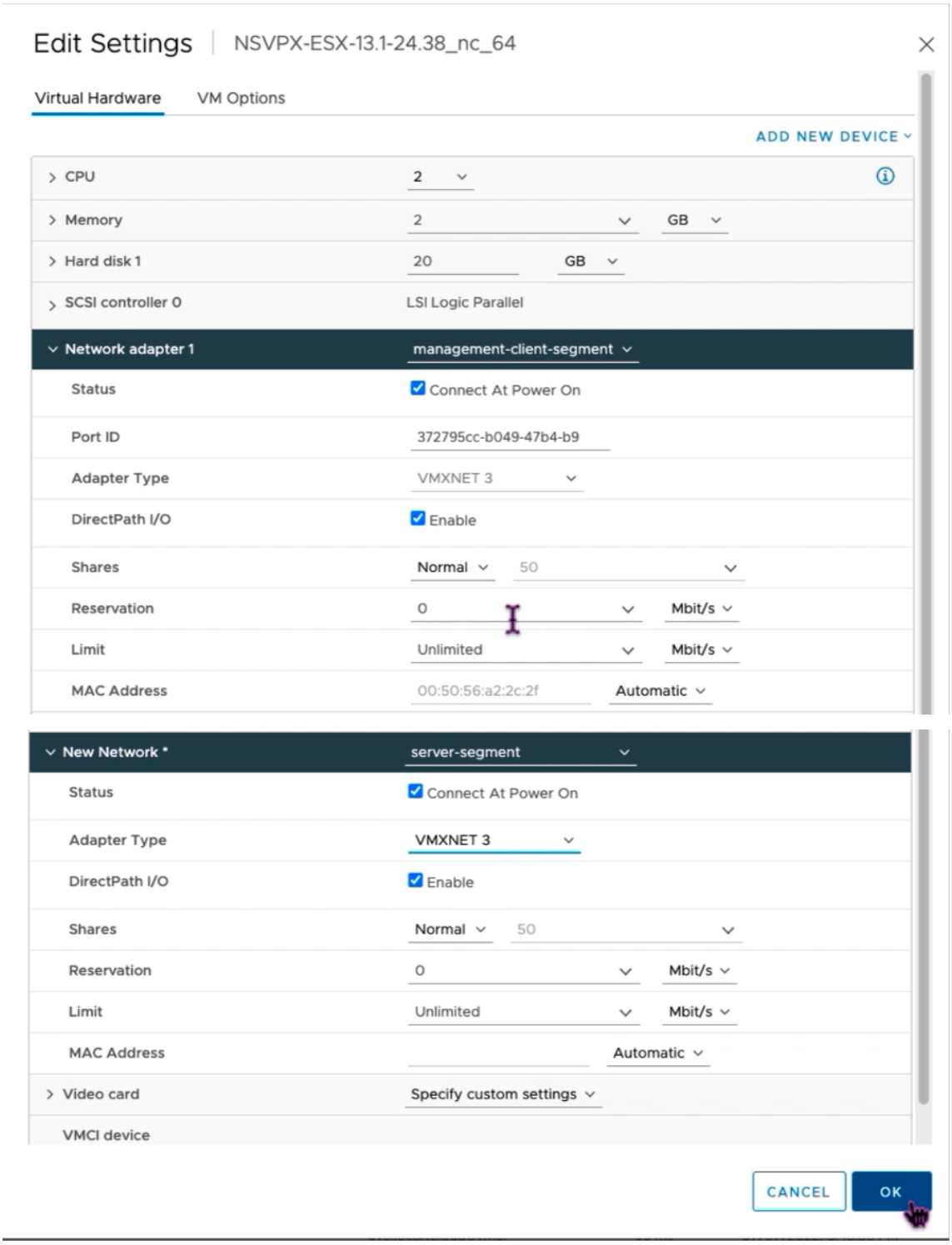
GCVE にプライベートクラウドをインストールして設定したら、vCenter を使用して VMware Engine に仮想アプライアンスをインストールできます。インストールできる仮想アプライアンスの数は、プライベートクラウドで使用可能なリソースの量によって異なります。

NetScaler VPX インスタンスをプライベートクラウドにインストールするには、プライベートクラウドのポイントツーサイト VPN に接続されたデスクトップで以下の手順を実行します。

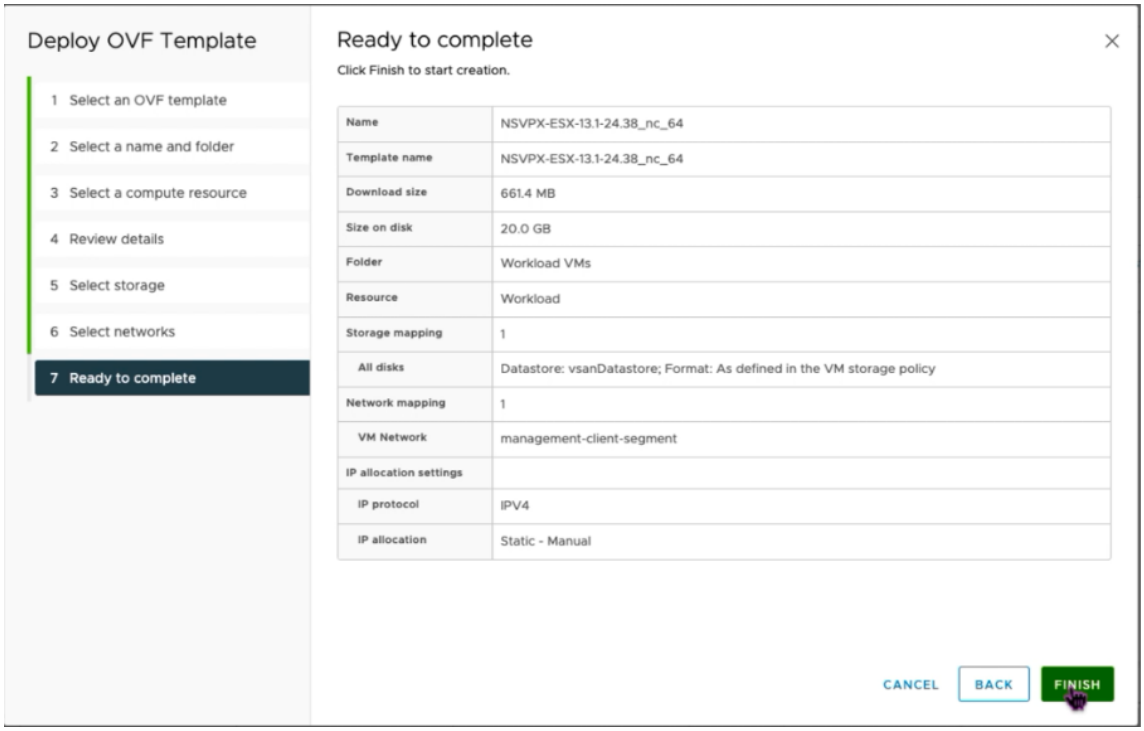
1. ESXi ホスト用の NetScaler VPX インスタンスセットアップファイルを、NetScaler ダウンロードサイトからダウンロードします。
2. プライベートクラウドのポイントツーサイト VPN に接続されたブラウザで VMware vCenter を開きます。
3. [ユーザー名] フィールドと [パスワード] フィールドに管理者の資格情報を入力し、[ログイン] をクリックします。
4. [**File**] メニューの [**Deploy OVF Template**] を選択します。
5. [**OVF** テンプレートのデプロイ] ダイアログボックスの [ファイルからの展開] フィールドで、NetScaler VPX インスタンスセットアップファイルを保存した場所を参照し、.ovf ファイルを選択し、[次へ] をクリックします。

   > 注:
   >
   > デフォルトでは、NetScaler VPX インスタンスは E1000 ネットワークインターフェイスを使用します。VMXNET3 インターフェイスで ADC を展開するには、E1000 ではなく VMXNET3 インターフェイスを使用するように OVF を変更します。VMXNET3 インターフェイスの可用性は GCP インフラストラクチャによって制限され、Google Cloud VMware Engine では利用できない場合があります。

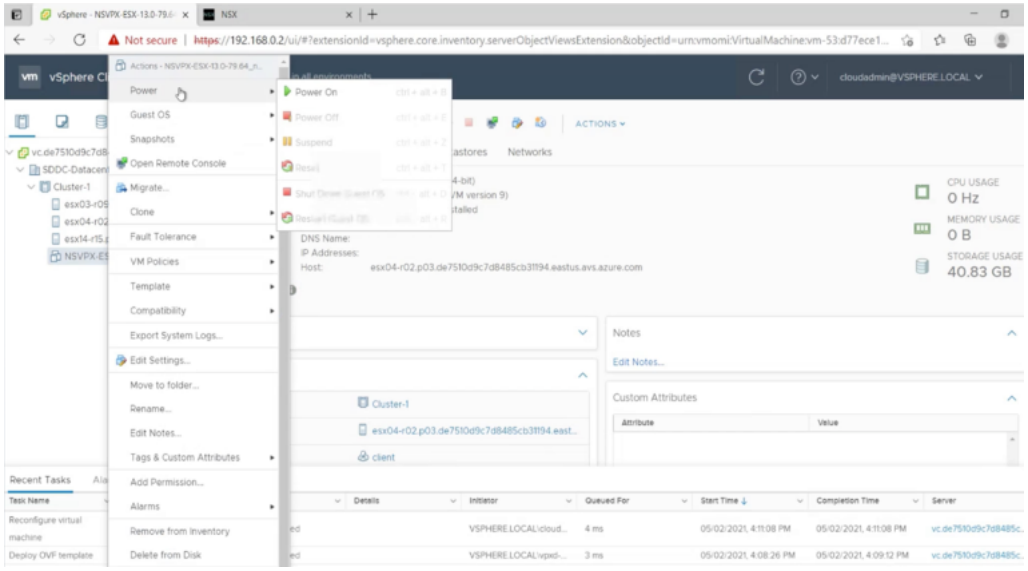
6. 仮想アプライアンスの OVF テンプレートに表示されるネットワークを、NSX-T Manager で設定したネットワークにマッピングします。[**OK**] をクリックします。



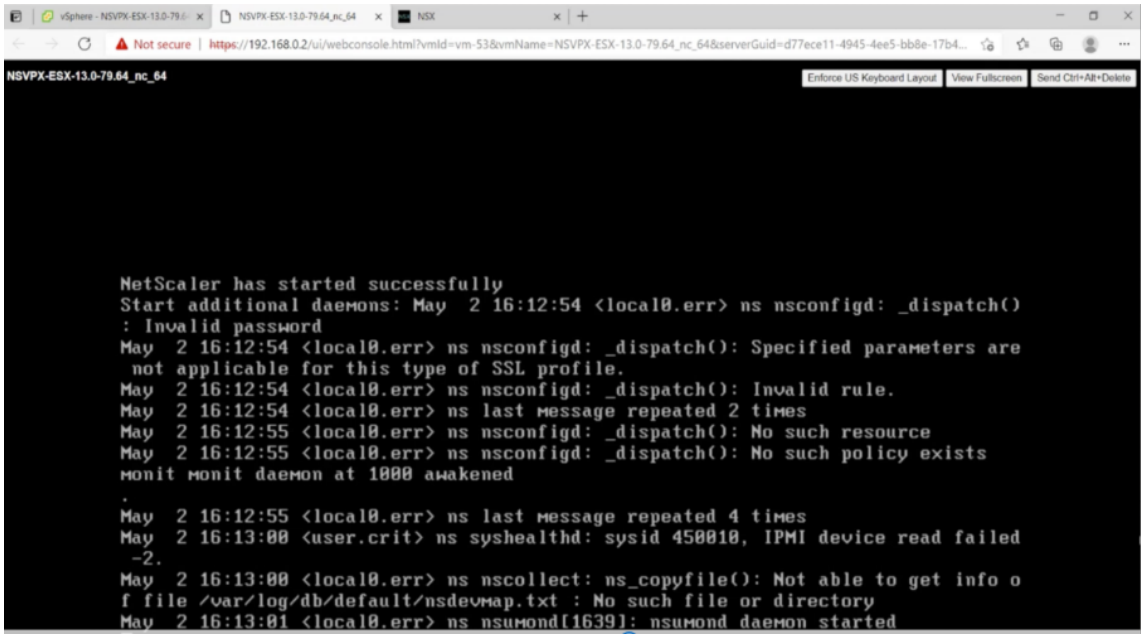
7. [完了] をクリックして VMware クラウドへの仮想アプライアンスのインストールを開始します。



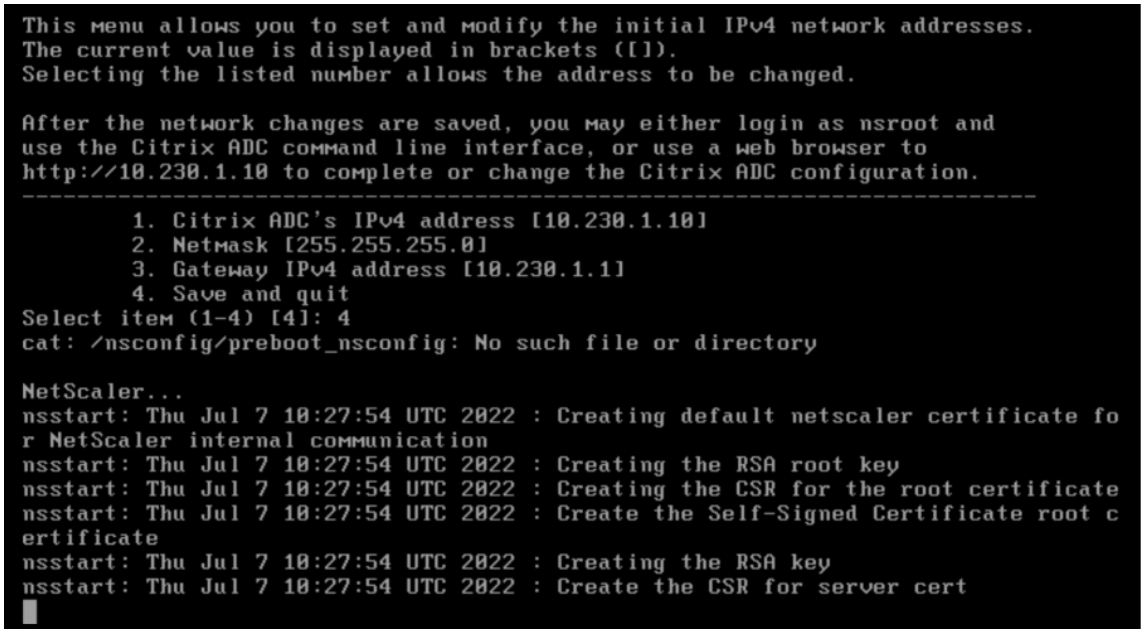
8. これで、NetScaler VPX インスタンスを起動する準備ができました。ナビゲーションペインで、インストールした NetScaler VPX インスタンスを選択し、右クリックメニューから［パワーオン］を選択します。コンソールポートをエミュレートするには、「**Web** コンソールの起動」タブをクリックします。



9. これで、vSphere クライアントから NetScaler 仮想マシンに接続されています。



10. 初回起動時に、ADC インスタンスの管理 IP とゲートウェイを設定します。



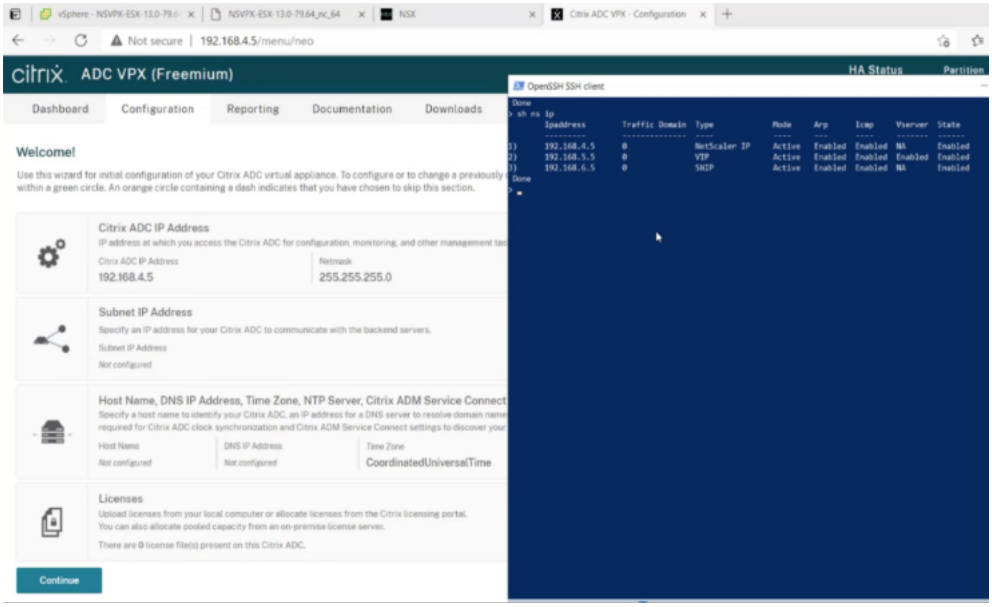
11. SSH キーを使用して NetScaler アプライアンスにアクセスするには、CLI で次のコマンドを入力します。

```
ssh nsroot@<management IP address>
<!--NeedCopy-->
```

例:

```
ssh nsroot@10.230.1.10
<!--NeedCopy-->
```

12. ADC の設定は、`show ns ip`コマンドを使用して確認できます。

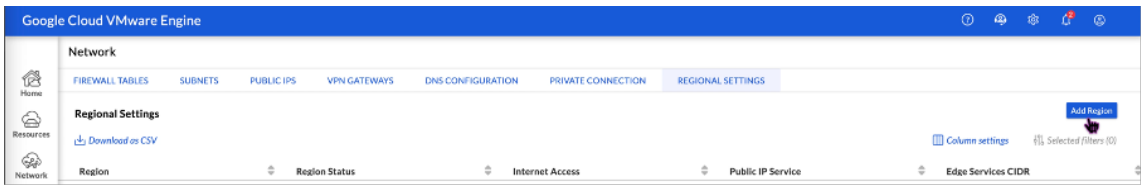


VMware クラウド上の NetScaler VPX インスタンスにパブリック IP アドレスを割り当てる

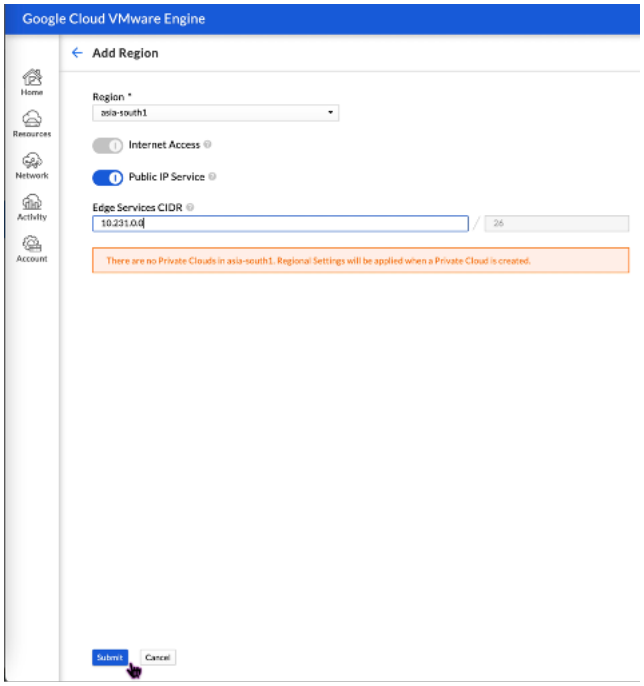
GCVE に NetScaler VPX インスタンスをインストールして構成したら、クライアントインターフェイスにパブリック IP アドレスを割り当てる必要があります。VM にパブリック IP アドレスを割り当てる前に、Google Cloud リージョンでパブリック IP サービスが有効になっていることを確認してください。

新しいリージョンのパブリック IP サービスを有効にするには、次の手順に従います。

1. GCVE コンソールで、[ネットワーク] **>** [地域設定] **>** [地域の追加] に移動します。



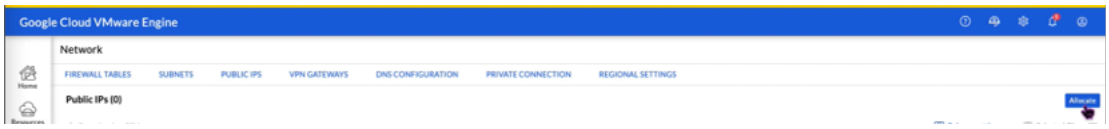
2. 地域を選択し、** インターネットアクセスとパブリック **IP** サービスを有効にします **。
3. エッジサービス CIDR を割り当てて、CIDR 範囲がオンプレミスまたは他の GCP/GCVE サブネット (仮想ネットワーク) と重複しないようにします。



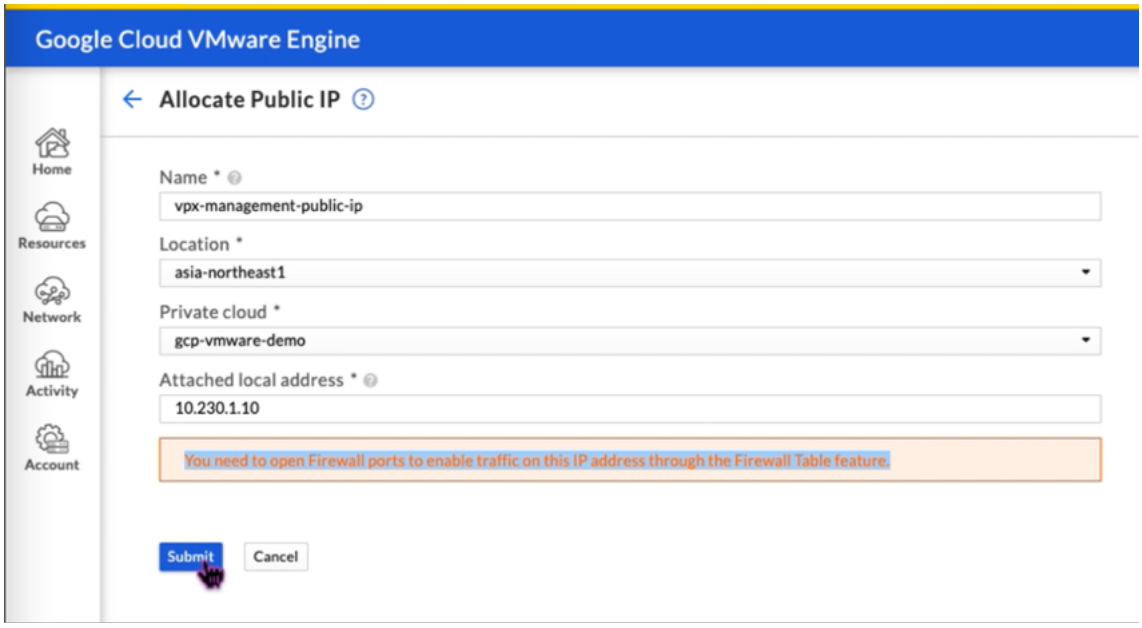
4. 数分後に、選択したリージョンのパブリック IP サービスが有効になります。

GCVE 上の NetScaler VPX インスタンスのクライアントインターフェイスにパブリック IP を割り当てるには、GCVE ポータルで以下の手順を実行します。

1. GCVE コンソールで、[ネットワーク**] > [**パブリック **IP] > [**割り当て] に移動します。



2. パブリック IP の名前を入力します。地域を選択し、IP を使用するプライベートクラウドを選択します。
3. パブリック IP をマッピングするインターフェイスのプライベート IP を指定します。** これはクライアントインターフェースのプライベート **IP** になります ** 。
4. [**Submit**] をクリックします。



5. パブリック IP は数分で使用可能になります。
6. パブリック IP を使用する前に、ファイアウォールルールを追加してパブリック IP へのアクセスを許可する必要があります。詳細については、「ファイアウォールルール」を参照してください。

バックエンドの **GCP Auto Scaling** サービスを追加する

October 25, 2023

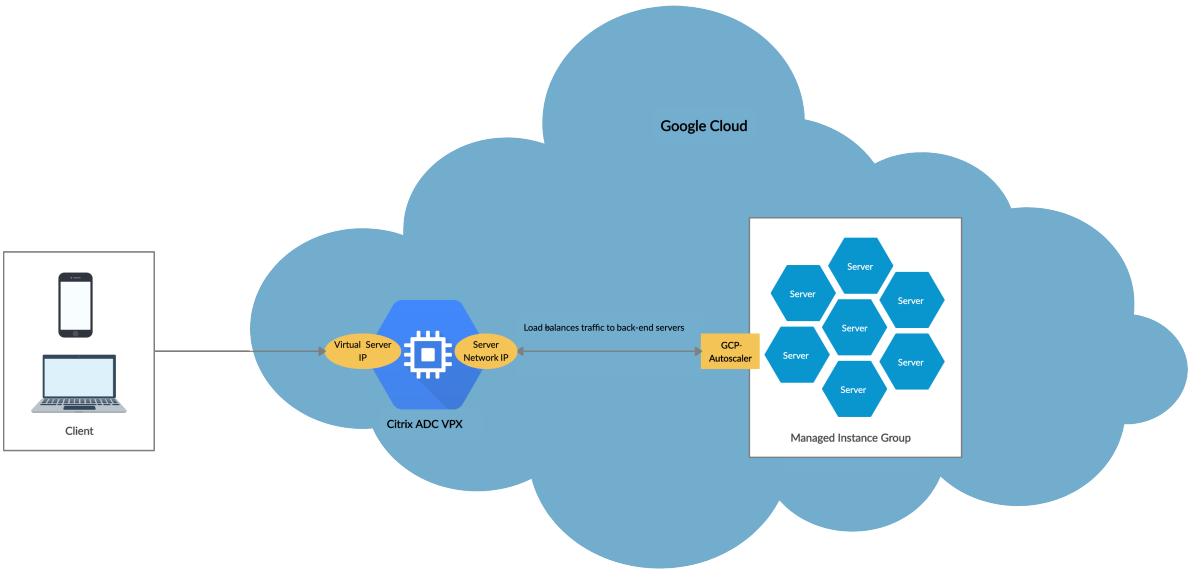
クラウドでアプリケーションを効率的にホストするには、アプリケーションの需要に応じて、簡単で費用対効果の高いリソース管理が必要です。増加する需要を満たすには、ネットワークリソースを拡大する必要があります。需要が収まったら、十分に活用されていないリソースによる不必要なコストを避けるために規模を縮小する必要があります。アプリケーションの実行コストを最小限に抑えるには、トラフィック、メモリ、CPU の使用などを常に監視する必要があります。しかし、トラフィックを手動で監視するのは煩雑です。アプリケーション環境を動的にスケールアップまたはスケールダウンするには、トラフィックの監視プロセスを自動化し、必要に応じてリソースを増減する必要があります。

GCP 自動スケーリングサービスと統合された NetScaler VPX インスタンスには、次の利点があります。

- 負荷分散と管理: 需要に応じてサーバーをスケールアップおよびスケールダウンするように自動構成します。VPX インスタンスはバックエンドサブネット内のマネージドインスタンスグループを自動検出し、負荷を分散するマネージドインスタンスグループを選択できます。仮想 IP アドレスとサブネット IP アドレスは、VPX インスタンスで自動構成されます。
- 高可用性: 複数のゾーンにまたがるマネージドインスタンスグループを検出し、サーバーの負荷を分散します。
- ネットワークの可用性の向上:VPX インスタンスは以下をサポートします。

- 同じ配置グループのバックエンドサーバー
- 異なるゾーンのバックエンドサーバー

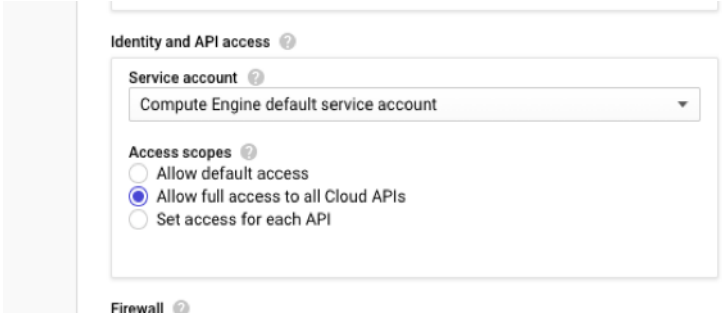
この図は、負荷分散仮想サーバーとして機能する NetScaler ADC VPX インスタンスで GCP 自動スケーリングサービスがどのように機能するかを示しています。



はじめに

NetScaler VPX インスタンスで自動スケーリングの使用を開始する前に、次のタスクを完了する必要があります。

- 要件に応じて、GCP 上に NetScaler ADC VPX インスタンスを作成します。
  - NetScaler VPX インスタンスの作成方法の詳細については、「Google Cloud Platform での NetScaler ADC VPX インスタンスの展開」を参照してください。
  - VPX インスタンスを HA モードでデプロイする方法の詳細については、「VPX 高可用性ペアを Google Cloud Platform にデプロイする」を参照してください。
- GCP プロジェクトで クラウドリソースマネージャー **API** を有効にします。
- インスタンスの作成中に、すべての Cloud API へのフルアクセスを許可します。



- GCP サービスアカウントに次の IAM 権限があることを確認します。

```
REQUIRED_INSTANCE_IAM_PERMS = [

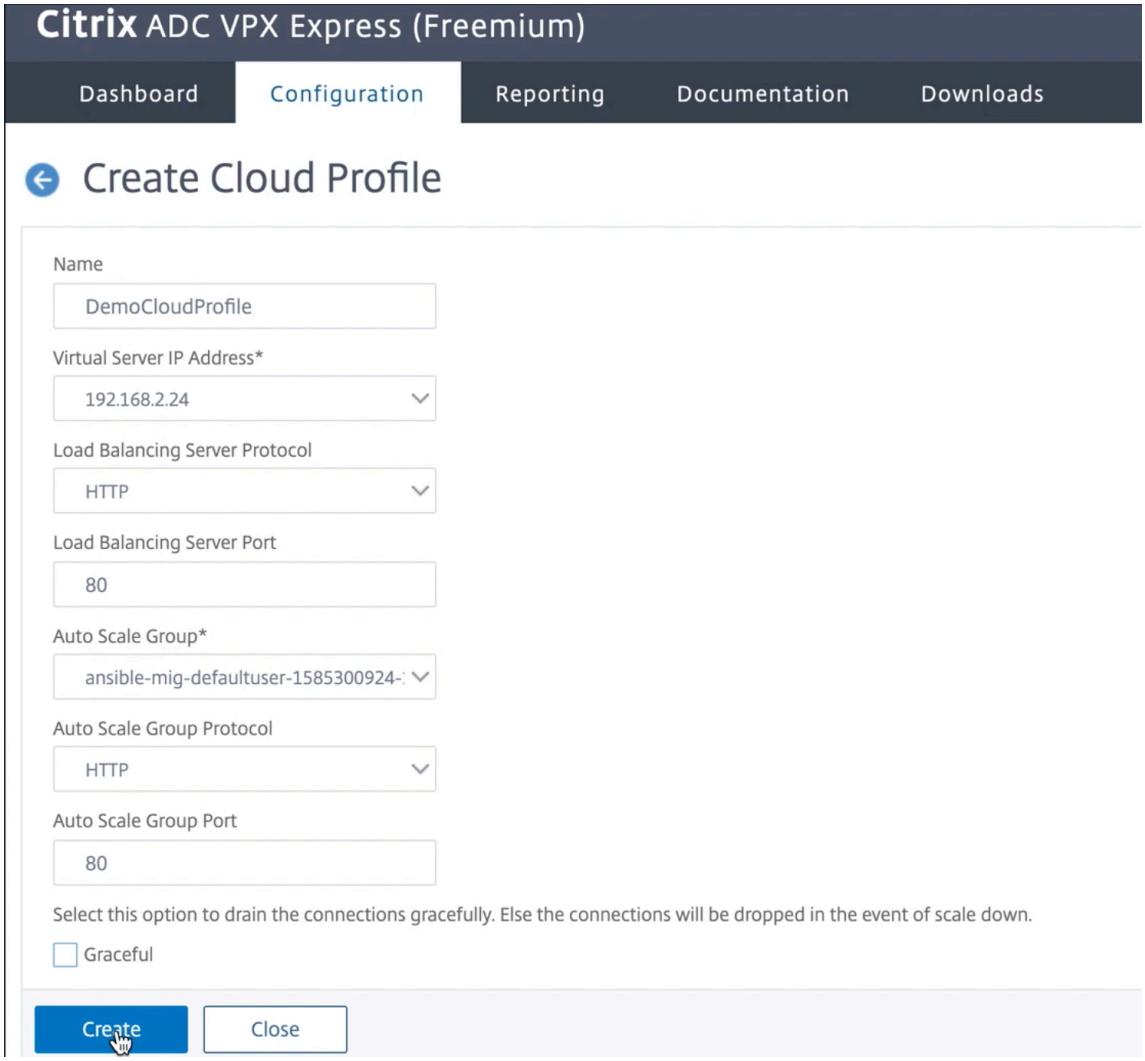
"compute.instances.get",
"compute.zones.list",
"compute.instanceGroupManagers.list",
"compute.instanceGroupManagers.get"
]
<!--NeedCopy-->
```

- 自動スケーリングを設定するには、以下が設定されていることを確認してください。
  - インスタンステンプレート
  - マネージドインスタンスグループ
  - 自動スケーリングポリシー

GCP 自動スケーリングサービスを NetScaler VPX インスタンスに追加する

GUI を使用して、ワンクリックで VPX インスタンスに自動スケーリングサービスを追加できます。次の手順を実行して、VPX インスタンスに自動スケーリングサービスを追加します。

1. nsrootの認証情報を使用して VPX インスタンスにログオンします。
2. NetScaler VPX インスタンスに初めてログオンすると、デフォルトの Cloud Profile ページが表示されます。ドロップダウンメニューから GCP マネージドインスタンスグループを選択し、[作成] をクリックしてクラウドプロファイルを作成します。

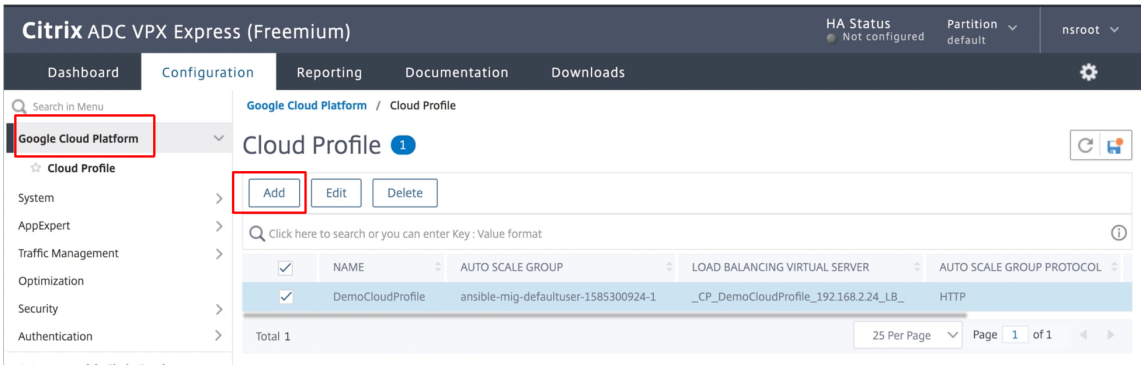


- [**Virtual Server IP Address**] フィールドは、インスタンスに関連付けられたすべての IP アドレスから自動的に入力されます。
- **Autoscale** グループは、GCP アカウントで設定されたマネージドインスタンスグループから事前設定されています。
- [自動スケールグループプロトコル] と [自動スケールグループポート] を選択するときは、サーバが構成済みのプロトコルとポートでリッスンしていることを確認します。サービスグループに適切なモニターをバインドします。デフォルトでは、TCP モニターが使用されます。
- **Graceful** チェックボックスはサポートされていないのでオフにしてください。

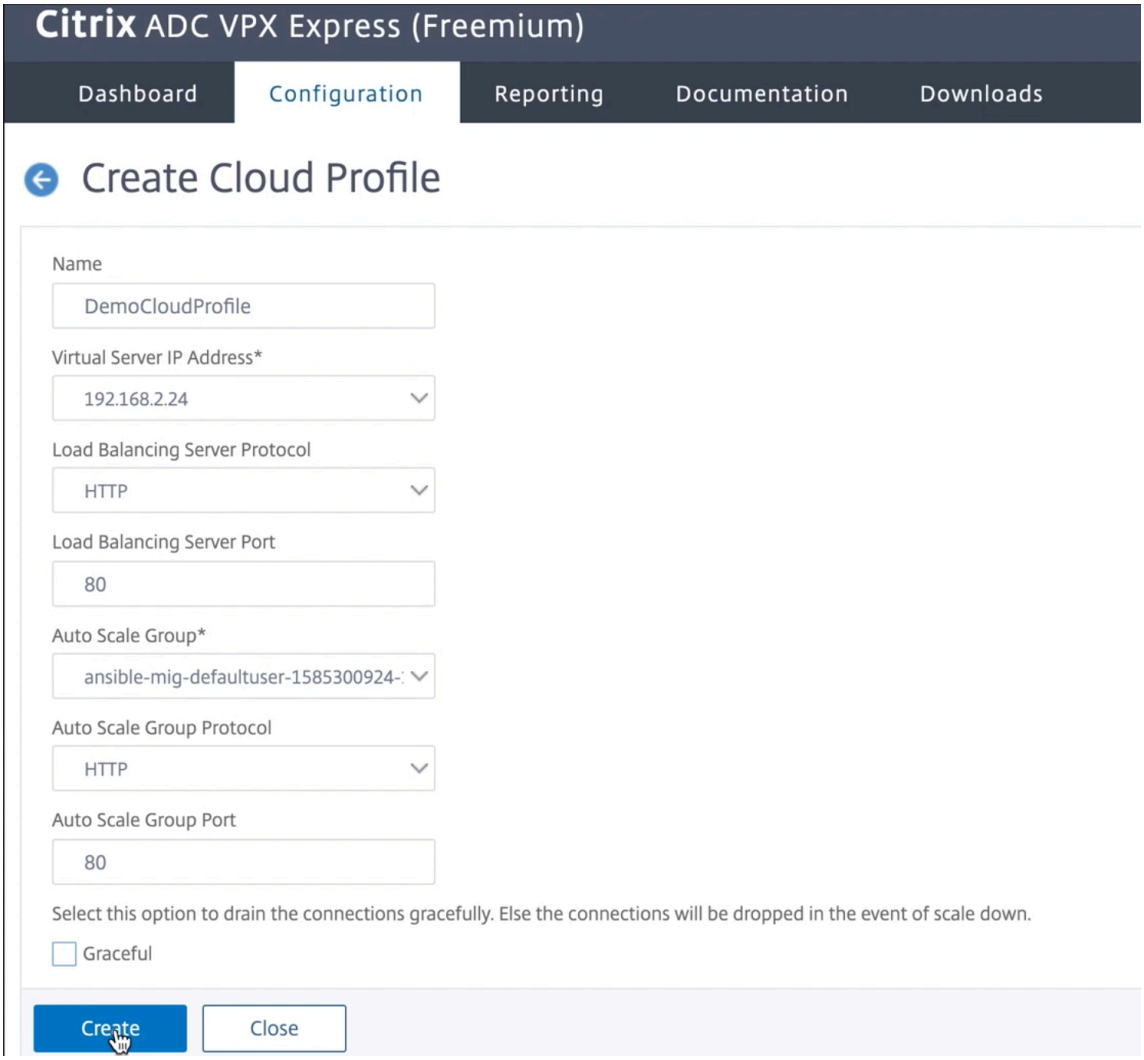
> 注記:
>
> SSL プロトコルタイプ Auto Scaling の場合、クラウドプロファイルを作成すると、証明書がないために負荷分散仮想サーバーまたはサービスグループがダウンします。証明書は、仮想サーバまたはサービスグループに手動でバインドできます。

3. 初めてログオンした後、クラウドプロファイルを作成する場合は、GUI で [システム] **> [Google Cloud**

Platform] > [クラウドプロファイル] に移動し、[追加] をクリックします。

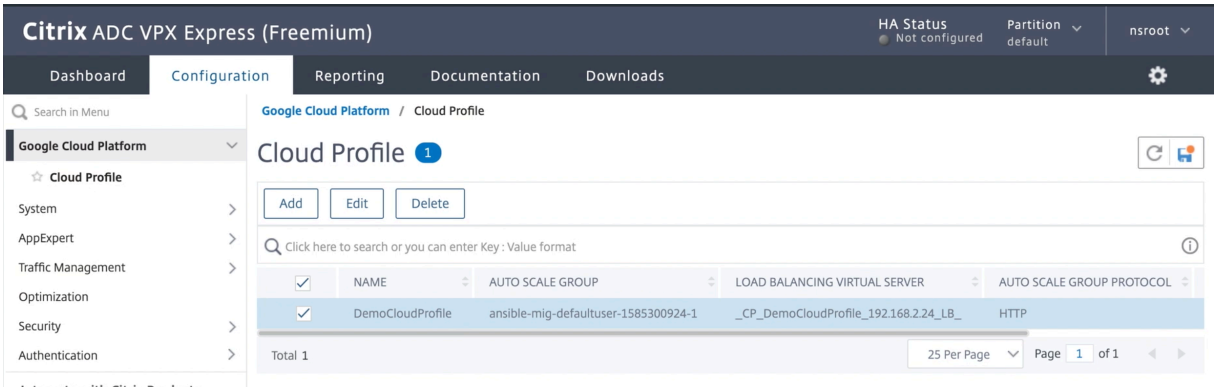


クラウドプロファイルの作成設定ページが表示されます。



Cloud Profile は、NetScaler 負荷分散仮想サーバーと、マネージドインスタンスグループのサーバーとしてメンバーを含むサービスグループを作成します。バックエンドサーバーは、VPX インスタンスで構成された SNIP を介して到達可能である必要があります。

> 注:
>
> NetScaler リリース 13.1-42.x 以降では、GCP の同じマネージドインスタンスグループを使用して、（異なるポートを使用して）サービスごとに異なるクラウドプロファイルを作成できます。したがって、NetScaler VPX インスタンスは、パブリッククラウド内の同じ自動スケーリンググループを持つ複数のサービスをサポートします。



GCP 上の NetScaler VPX インスタンスの VIP スケーリングサポート

October 25, 2023

NetScaler アプライアンスはクライアントとサーバーの間に設置され、クライアント要求とサーバー応答は NetScaler アプライアンスを経由します。一般的な設置では、アプライアンス上で構成された仮想サーバーによって接続ポイントが提供され、クライアントはこれを使用してアプライアンスの背後にあるアプリケーションにアクセスします。展開に必要なパブリック仮想 IP（VIP）アドレスの数は、ケースバイケースで異なります。

GCP アーキテクチャでは、インスタンスの各インターフェイスが異なる VPC に接続されるように制限します。GCP 上の VPC はサブネットの集合であり、各サブネットはリージョンのゾーンにまたがることができます。さらに、GCP には次の制限があります。

- パブリック IP アドレス数と NIC の数が 1:1 でマッピングされています。NIC に割り当てることができるパブリック IP アドレスは 1 つだけです。
- 大容量のインスタンスタイプには最大 8 つの NIC しか接続できません。

たとえば、n1-standard-2 インスタンスは 2 つの NIC しか持つことができず、追加できるパブリック VIP は 2 つに制限されています。詳細については、「VPC リソースクォータ」を参照してください。

NetScaler VPX インスタンスでより大規模なパブリック仮想 IP アドレスを実現するために、インスタンスのメタデータの一部として VIP アドレスを構成できます。NetScaler VPX インスタンスは、GCP が提供する転送ルールを内部で使用して VIP スケーリングを実現します。NetScaler VPX インスタンスは、構成済みの VIP にも高可用性を提供します。

メタデータの一部として VIP アドレスを設定した後、転送ルールの作成に使用するのと同じ IP を使用して LB 仮想サーバを設定できます。そのため、転送ルールを使用することで、GCP 上の NetScaler VPX インスタンスでパブリック VIP アドレスを使用する際のスケーリング上の制限を緩和できます。

転送ルールの詳細については、「転送ルールの概要」を参照してください。

HA の詳細については、「高可用性」を参照してください。

注意事項

- Google は、各仮想 IP 転送ルールに対して追加費用を請求します。実際のコストは、作成されるエントリの数によって異なります。関連するコストは、Google の価格設定ドキュメントから確認できます。
- 転送ルールは、パブリック VIP にのみ適用されます。展開でプライベート IP アドレスが VIP として必要な場合は、エイリアス IP アドレスを使用できます。
- 転送ルールは、LB 仮想サーバーを必要とするプロトコルに対してのみ作成できます。VIP は、その場で作成、更新、または削除できます。同じ VIP アドレスを持つが、プロトコルが異なる新しい負荷分散仮想サーバーを追加することもできます。

はじめに

- NetScaler VPX インスタンスは、GCP にデプロイする必要があります。
- 外部 IP アドレスは予約する必要があります。詳細については、「静的外部 IP アドレスの予約」を参照してください。
- GCP サービスアカウントに次の IAM 権限があることを確認します。

```
REQUIRED_IAM_PERMS = [
"compute.addresses.list",
"compute.addresses.get",
"compute.addresses.use",
"compute.forwardingRules.create",
"compute.forwardingRules.delete",
"compute.forwardingRules.get",
"compute.forwardingRules.list",
"compute.instances.use",
"compute.subnetworks.use",
"compute.targetInstances.create"
"compute.targetInstances.get"
"compute.targetInstances.use",
]

<!--NeedCopy-->
```

- GCP プロジェクトで クラウドリソースマネージャー **API** を有効にします。
- スタンドアロン VPX インスタンスで VIP スケーリングを使用する場合は、GCP サービスアカウントに次の IAM 権限があることを確認してください。


```
REQUIRED_IAM_PERMS = [
"compute.addresses.list",
"compute.addresses.get",
"compute.addresses.use",
"compute.forwardingRules.create",
"compute.forwardingRules.delete",
"compute.forwardingRules.get",
"compute.forwardingRules.list",
"compute.instances.use",
"compute.subnetworks.use",
"compute.targetInstances.create",
"compute.targetInstances.list",
"compute.targetInstances.use",
]
<!--NeedCopy-->
```

- 高可用性モードで VIP スケーリングを使用する場合は、GCP サービスアカウントに次の IAM 権限があることを確認してください。

```
REQUIRED_IAM_PERMS = [
"compute.addresses.get",
"compute.addresses.list",
"compute.addresses.use",
"compute.forwardingRules.create",
"compute.forwardingRules.delete",
"compute.forwardingRules.get",
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.use",
"compute.instances.get",
"compute.instances.list",
"compute.instances.setMetadata",
"compute.subnetworks.use",
"compute.targetInstances.create",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.zones.list",
]
<!--NeedCopy-->
```

> 注記:
>
> 高可用性モードでは、サービスアカウントに所有者または編集者の役割がない場合は、サービスアカウントにサービスアカウントユーザーの役割を追加する必要があります。

NetScaler VPX インスタンスでの **VIP** スケーリング用の外部 **IP** アドレスを構成する

1. Google Cloud コンソールで、[**VM** インスタンス] ページに移動します。
2. 新しい VM インスタンスを作成するか、既存のインスタンスを使用します。

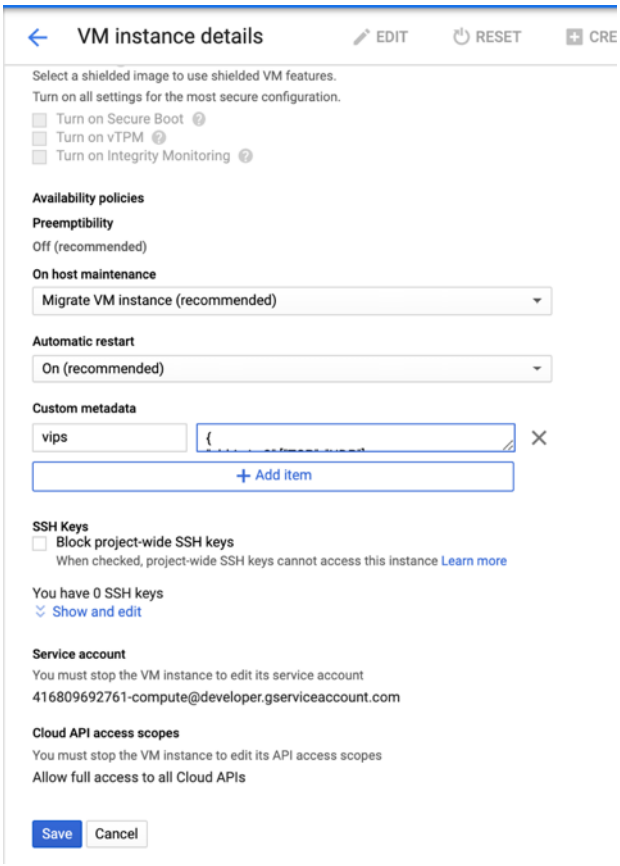
3. インスタンス名をクリックします。**VM** インスタンスの詳細ページで、[編集] をクリックします。
4. 次のように入力して、カスタムメタデータを更新します。

   - キー = VIP
   - 値 = 次の JSON 形式で値を指定します。

     {
     「外部予約 IP の名前」: [プロトコルのリスト]、
     }

   GCP は次のプロトコルをサポートしています。

   - AH
   - ESP
   - ICMP
   - SCT
   - TCP
   - UDP



詳細については、「カスタムメタデータ」を参照してください。

カスタムメタデータの例:

{
“external-ip1-name”:[“TCP”,“UDP”],
“external-ip2-name”:[“ICMP”,“AH”]
}

この例では、NetScaler VPX インスタンスは、IP、プロトコルのペアごとに 1 つの転送ルールを内部で作成します。メタデータエントリは、転送ルールにマッピングされます。この例では、メタデータエントリに対して作成される転送ルールの数を把握するのに役立ちます。

次の 4 つの転送ルールが作成されます。

a) external-ip1-name と TCP
b) external-ip1-name と UDP
c) external-ip2-name と ICMP
d) external-ip2-name と AH

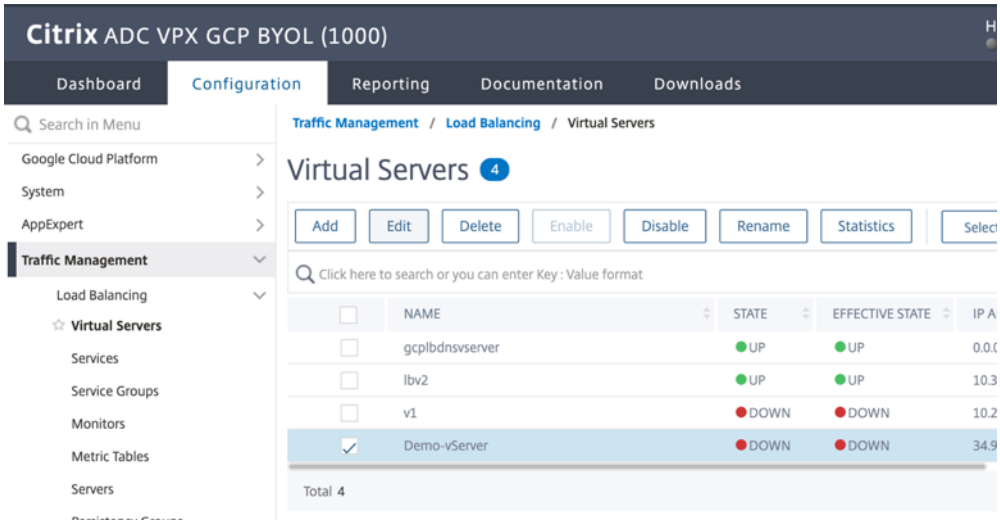
> 注記:
>
> HA モードでは、プライマリインスタンスにのみカスタムメタデータを追加する必要があります。フェールオーバー時に、カスタムメタデータが新しいプライマリに同期されます。

5. [保存] をクリックします。

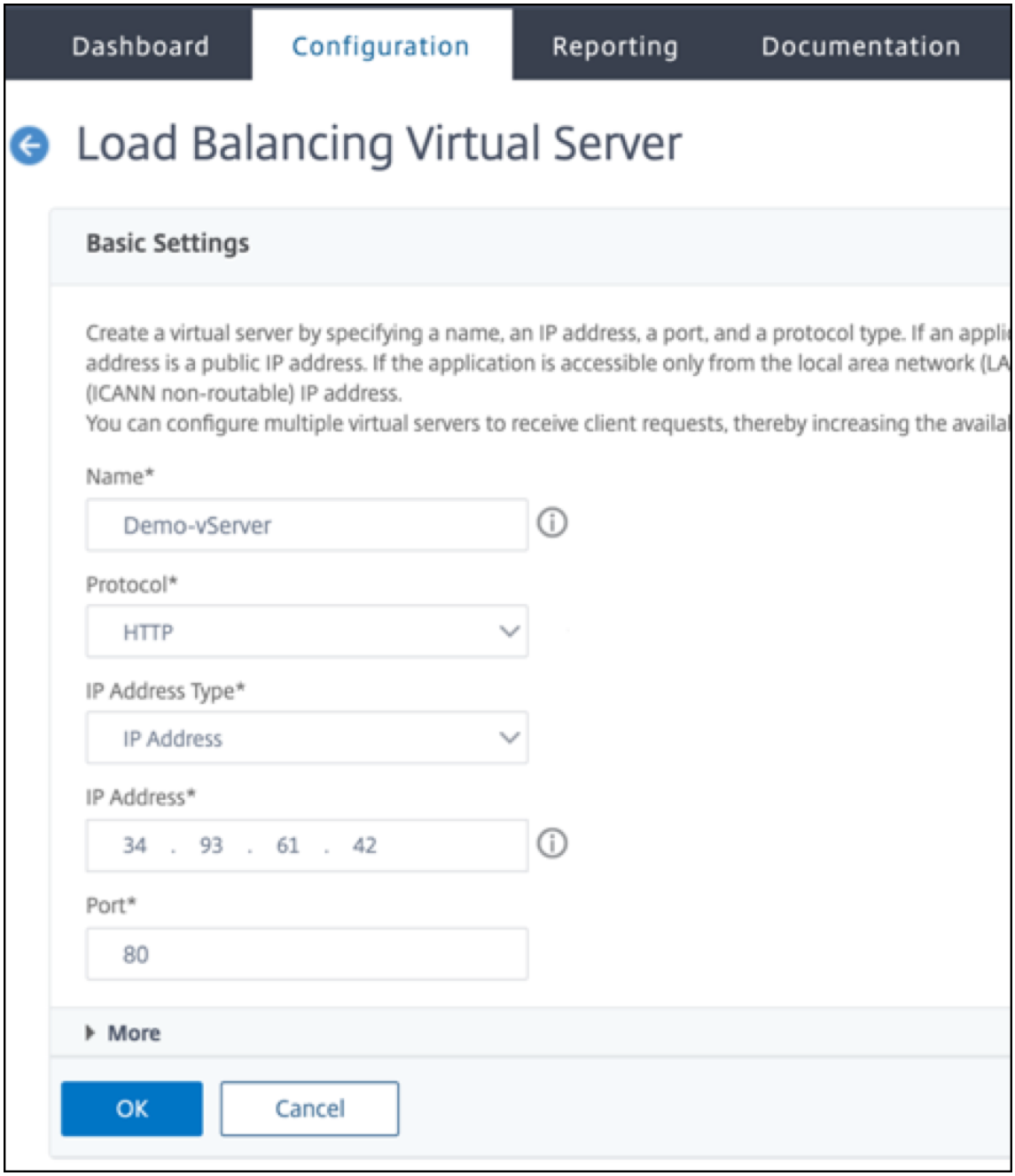
NetScaler VPX インスタンスで外部 **IP** アドレスを使用した負荷分散仮想サーバーのセットアップ

ステップ **1:** 負荷分散仮想サーバーを追加します。

1. [設定] **>** [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] **>** [追加] に移動します。

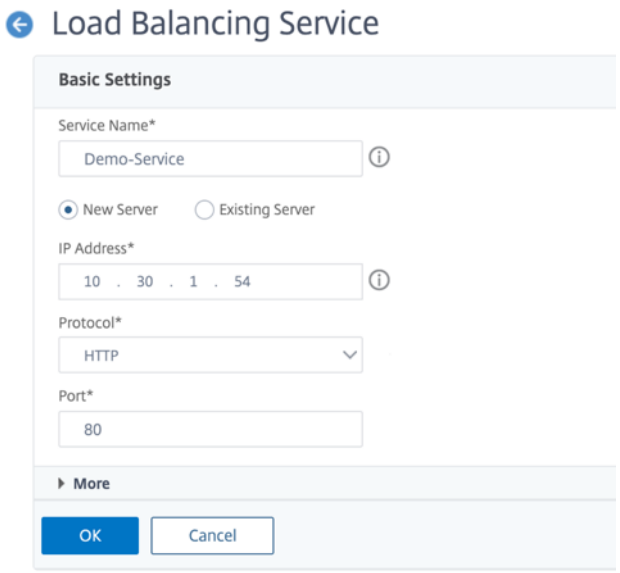


2. 名前、プロトコル、IP アドレスタイプ（IP アドレス）、IP アドレス（ADC で VIP として追加される転送ルールの外部 IP アドレス）、およびポートに必要な値を追加し、「**OK**」をクリックします。



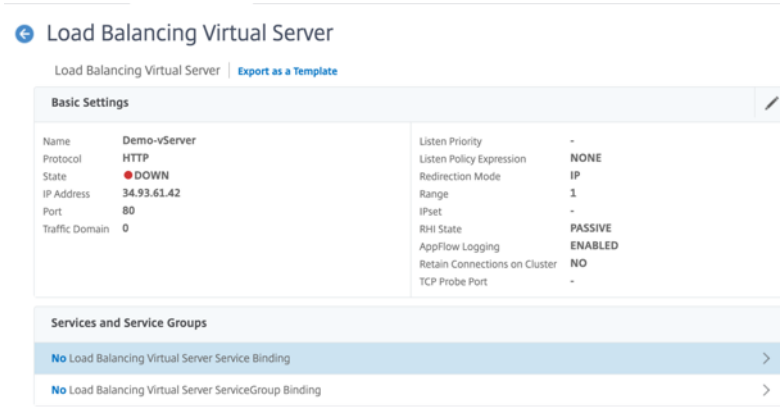
ステップ **2:** サービスまたはサービスグループを追加します。

1. **[**設定**] > [**トラフィック管理**] > [**負荷分散**] > [**サービス**] >[**追加**]** に移動します。
2. サービス名、IP アドレス、プロトコル、およびポートに必要な値を追加し、[**OK**] をクリックします。

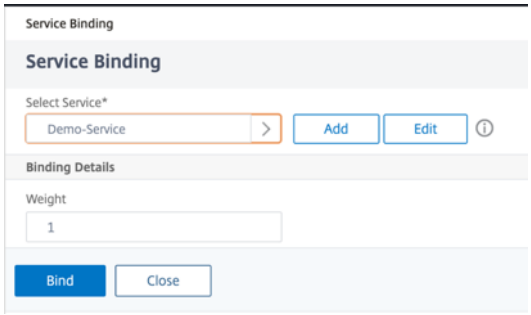


ステップ **3:** サービスまたはサービスグループを負荷分散仮想サーバーにバインドします。

1. **[**設定**] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動します。
2. 手順 **1** で構成した負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [サービスとサービスグループ] ページで、[負荷分散仮想サーバーサービスのバインドなし] をクリックします。



4. 手順 **3** で構成したサービスを選択し、[バインド] をクリックします。



5. 構成を保存します。

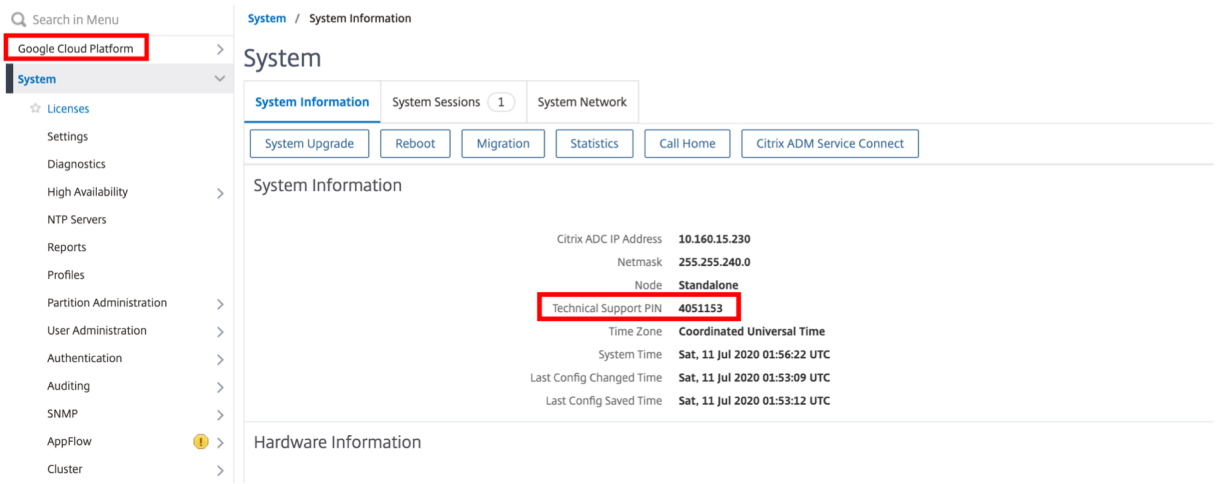
GCP での VPX インスタンスのトラブルシューティング

August 15, 2023

Google Cloud Platform （GCP）は、NetScaler VPX インスタンスへのコンソールアクセスを提供します。デバッグできるのは、ネットワークが接続されている場合だけです。インスタンスのシステムログを表示するには、コンソールにアクセスしてシステムログファイルを確認します。

NetScaler は、GCP 上で有料の NetScaler VPX インスタンス（時間単位のユーティリティライセンス）をサポートします。サポートケースを提出するには、GCP アカウント番号とサポート PIN コードを見つけて、NetScaler サポートに連絡してください。名前とメールアドレスの入力を求められます。サポート PIN を見つけるには、VPX GUI にログオンし、システムページに移動します。

サポート PIN を示すシステムページの例を次に示します。



NetScaler VPX インスタンスのジャンボフレーム

August 15, 2023

NetScaler VPX アプライアンスは、最大 9216 バイトの IP データを含むジャンボフレームの送受信をサポートしています。ジャンボフレームでは、標準の IP MTU サイズ（1500 バイト）を使用するよりも効率的に大きなファイルを送信することができます。

NetScaler アプライアンスは、以下の展開シナリオでジャンボフレームを使用することができます。

- ジャンボで受信/ジャンボで送信。アプライアンスがデータをジャンボフレームで受信し、それをジャンボフレームで送信します。
- 非ジャンボで受信/ジャンボで送信。アプライアンスがデータを通常のフレームで受信し、それをジャンボフレームで送信します。
- ジャンボで受信/非ジャンボで送信。アプライアンスがデータをジャンボフレームで受信し、それを通常のフレームで送信します。

詳細については、「NetScaler ADC アプライアンスでのジャンボフレームサポートの構成」を参照してください。

ジャンボフレームのサポートは、次の仮想化プラットフォームで実行されている NetScaler ADC VPX アプライアンスで利用できます。

- VMware ESX
- Linux-KVM プラットフォーム
- Citrix XenServer
- Amazon Web Services（AWS）

VPX アプライアンスのジャンボフレームは、MPX アプライアンスのジャンボフレームと同様に機能します。ジャンボフレームおよびそのユースケースについて詳しくは、「MPX アプライアンスでのジャンボフレームの構成」を参照してください。MPX アプライアンスのジャンボフレームの使用例は、VPX アプライアンスにも当てはまります。

VMware ESX で実行中の **VPX** インスタンスのジャンボフレームを構成する

VMware ESX サーバーで実行されている NetScaler ADC VPX アプライアンスでジャンボフレームを構成するには、次のタスクを実行します。

1. VPX アプライアンスのインターフェイスまたはチャネルの MTU を 1501～9000 の範囲の値に設定します。CLI または GUI を使用して MTU サイズを設定します。VMware ESX 上で動作する NetScaler VPX アプライアンスは、最大 9000 バイトの IP データのみを含むジャンボフレームの送受信をサポートします。
2. 管理アプリケーションを使用して、VMware ESX サーバーの対応する物理インターフェイスで同じ MTU サイズを設定します。VMware ESX の物理インターフェイスでの MTU サイズの設定の詳細については、http://vmware.com/を参照してください。

Linux-KVM サーバーで実行されている **VPX** インスタンスのジャンボフレームを構成する

Linux-KVM サーバーで実行されている NetScaler VPX アプライアンスでジャンボフレームを構成するには、次のタスクを実行します。

1. VPX アプライアンスのインターフェイスまたはチャネルの MTU を 1501～9216 の範囲の値に設定します。NetScaler VPX CLI または GUI を使用して MTU サイズを設定します。
2. 管理アプリケーションを使用して、Linux-KVM サーバーの対応する物理インターフェイスで同じ MTU サイズを設定します。Linux-KVM の物理インターフェイスの MTU サイズの設定の詳細については、http://www.linux-kvm.org/を参照してください。

Citrix XenServer 上で実行されている VPX インスタンスのジャンボフレームを構成する

Citrix XenServer で実行されている NetScaler VPX アプライアンスでジャンボフレームを構成するには、次のタスクを実行します。

1. XenCenter を使用して XenServer に接続します。
2. MTU を変更する必要があるネットワークを使用するすべての VPX インスタンスをシャットダウンします。
3. [ネットワーク] タブで、ネットワーク-[ネットワーク 0/1/2] を選択します。
4. [プロパティ] を選択し、MTU を編集します。

XenServer でジャンボフレームを構成した後、ADC アプライアンスでジャンボフレームを構成できます。詳細については、「NetScaler ADC アプライアンスでのジャンボフレームサポートの構成」を参照してください。

AWS で実行中の VPX インスタンスのジャンボフレームを設定する

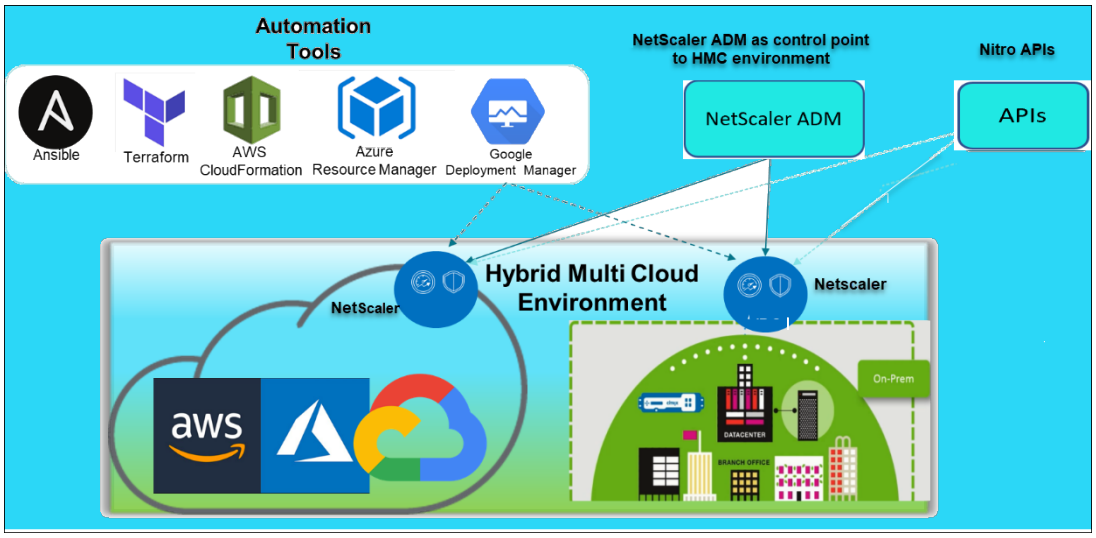
Azure 上の VPX では、ホストレベルの構成は不要です。VPX でジャンボフレームを構成するには、NetScaler ADC アプライアンスでのジャンボフレームサポートの構成に記載されている手順に従います。

NetScaler の導入と構成を自動化する

March 20, 2024

NetScaler には、ADC の展開と構成を自動化するための複数のツールが用意されています。このドキュメントでは、さまざまな自動化ツールの概要と、ADC 構成の管理に使用できるさまざまな自動化リソースの参照について説明します。

次の図は、ハイブリッドマルチクラウド（HMC）環境における NetScaler 自動化の概要を示しています。



NetScaler コンソールを使用して **NetScaler** を自動化

NetScaler Console は、分散型 ADC インフラストラクチャへの自動化コントロールポイントとして機能します。NetScaler Console は、ADC アプライアンスのプロビジョニングからアップグレードまで、一連の包括的な自動化機能を提供します。ADM の主な自動化機能は次のとおりです:

- AWS での NetScaler VPX インスタンスのプロビジョニング
- Azure での NetScaler VPX インスタンスのプロビジョニング
- StyleBooks
- 構成ジョブ
- 構成監査
- ADC アップグレード
- SSL 証明書の管理
- 統合- GitHub、ServiceNow、イベント通知の統合

自動化に関する **NetScaler** コンソールのブログとビデオ

- StyleBooks を使用したアプリケーションの移行
- ADM スタイルブックを使用して ADC 構成を CI/CD と統合する
- ADM によるパブリッククラウドの NetScaler 導入の簡素化
- NetScaler コンソールサービスが NetScaler のアップグレードを容易にする 10 の方法

NetScaler Console には、全体的な IT 自動化の一環として NetScaler コンソールと NetScaler を統合するさまざまな機能の API も用意されています。詳しくは、「NetScaler コンソールサービス API」を参照してください。

Terraform を使用して **NetScaler** を自動化する

Terraform は、クラウド、インフラストラクチャ、またはサービスのプロビジョニングと管理に、インフラストラクチャをコードアプローチとして採用するツールです。NetScaler テラフォームリソースは、GitHub で使用できます。詳細なドキュメントと使用法については、GitHub を参照してください。

- NetScaler Terraform モジュールにより、負荷分散や GSLB などのさまざまなユースケースに合わせて ADC を構成できます
- AWS に ADC をデプロイするための Terraform クラウドスクリプト
- Azure に ADC をデプロイするための Terraform クラウドスクリプト
- Terraform クラウドスクリプトで ADC を GCP にデプロイ
- NetScaler VPX および Azure パイプラインを使用したブルーグリーン展開

Terraform の **ADC** 自動化に関するブログとビデオ

- Terraform で NetScaler の展開を自動化

- Terraform を使用した AWS の HA セットアップで ADC をプロビジョニングおよび設定する

領事-**Terraform-Sync** を使用して **NetScaler** を自動化する

NetScaler Consul-Terraform-Sync（CTS）モジュールにより、アプリケーションチームはサービスの新しいインスタンスを NetScaler に自動的に追加または削除できます。必要な ADC 構成の変更を行うために、IT 管理者やネットワーキングチームに手動でチケットを提出する必要はありません。

- ネットワークインフラストラクチャ自動化のための NetScaler 領事 Terraform-Sync モジュール
- Citrix-HashiCorp 共同ウェビナー: Terraform Enterprise および NetScaler 向けの領事 Terraform-Sync を使用した動的ネットワーキング

Ansible を使用して **NetScaler** を自動化する

Ansible は、インフラストラクチャをコードとして実現する、オープンソースのソフトウェアプロビジョニング、構成管理、およびアプリケーションデプロイメントツールです。NetScaler Ansible モジュールとサンプルプレイブックは、GitHub にあります。詳細なドキュメントと使用法については、GitHub を参照してください。

- ADC を構成するための Ansible モジュール
- ADC Ansible モジュールドキュメント/リファレンスガイド
- ADM 用の Ansible モジュール

Citrix は認定された AnsibleAutomation パートナーです。Red Hat Ansible オートメーションプラットフォームのサブスクリプションをお持ちのユーザーは、Red Hat オートメーションハブから NetScalerコレクションにアクセスできます。

Terraform と **Ansible** の自動化ブログ

- Citrix、HashiCorp 統合パートナー・オブ・ザ・イヤーに選出
- Citrix は Red Hat Ansible オートメーションプラットフォーム認定パートナーになりました
- アプリケーションの配信とセキュリティのための Terraform と Ansible Automation

ADC 展開用のパブリッククラウドテンプレート

パブリッククラウドテンプレートは、パブリッククラウドでのデプロイメントのプロビジョニングを簡素化します。さまざまな環境で、さまざまな NetScaler テンプレートを使用できます。使用方法の詳細については、それぞれの GitHub リポジトリを参照してください。

AWS CFT:

- AWS で NetScaler VPX をプロビジョニングするための CFT

Azure Resource Manager (ARM) テンプレート**:**

- Azure で NetScaler VPX をプロビジョニングするための ARM テンプレート

Google Cloud デプロイメントマネージャー **(GDM)** テンプレート**:**

- Google で NetScaler VPX をプロビジョニングするための GDM テンプレート

テンプレートに関する動画

- クラウドフォーメーションテンプレートを使用して NetScaler HA を AWS にデプロイ
- AWS クイックスタートを使用してアベイラビリティーゾーン全体に NetScaler HA
- GDM テンプレートを使用した GCP での NetScaler HA の展開

AWS クイックスタート

- NetScaler Web App Firewall クイックスタート
- AWS でのウェブアプリケーション向け NetScaler VPX の AWS クイックスタート

NITRO API

NetScaler NITRO プロトコルを使用すると、表現状態転送（REST）インターフェイスを使用して、NetScaler アプライアンスをプログラムで構成および監視できます。そのため、NITRO アプリケーションはあらゆるプログラミング言語で開発することができます。Java、.NET、または Python で開発する必要があるアプリケーションの場合、NITRO API は、個別のソフトウェア開発キット (SDK) としてパッケージ化された関連ライブラリを通じて公開されます。

- NITRO API ドキュメント
- NITRO API を使用した ADC ユースケースの設定例

よくある質問

December 8, 2023

次のセクションでは、Citrix アプリケーション Delivery Controller（ADC）VPX に基づいて FAQ を分類するのに役立ちます。

- 機能と機能

- 暗号化
- 価格設定と梱包
- NetScaler VPX エクスプレス
- ハイパーバイザー
- キャパシティプランニングまたはサイジング
- システム要件
- その他の技術的なよくある質問

機能と機能

NetScaler VPX とは何ですか**?**

NetScaler VPX は、業界標準のサーバーにインストールされた Hypervisor でホストできる仮想 ADC アプライアンスです。

NetScaler VPX には、すべての **Web** アプリケーション最適化機能が **ADC** アプライアンスとして含まれていますか

はい。NetScaler VPX には、すべての負荷分散、トラフィック管理、アプリケーションアクセラレーション、アプリケーションセキュリティ（NetScaler Gateway および Citrix アプリケーションファイアウォールを含む）、およびオフロード機能が含まれています。NetScaler の機能と機能の完全な概要については、「アプリケーションの配信方法」を参照してください。

Citrix アプリケーションファイアウォールを **NetScaler VPX** で使用する場合、制限はありますか**?**

NetScaler VPX 上の Citrix アプリケーションファイアウォールは、NetScaler アプライアンスと同じセキュリティ保護を提供します。Citrix アプリケーションファイアウォールのパフォーマンスまたはスループットは、プラットフォームによって異なります。

NetScaler VPX 上の **NetScaler Gateway** と **NetScaler** アプライアンスの **NetScaler Gateway** の間に違いはありますか**?**

機能的には、それらは同じです。NetScaler VPX 上の NetScaler Gateway は、NetScaler ソフトウェアリリース 9.1 で利用可能なすべての NetScaler Gateway 機能をサポートします。ただし、NetScaler アプライアンスは専用の SSL アクセラレーションハードウェアを提供するため、NetScaler VPX インスタンスよりも優れた SSL VPN スケーラビリティを提供します。

Hypervisor 上で実行できるという明らかな違い以外に、**NetScaler VPX** は **NetScaler** 物理アプライアンスとどのように違いますか

顧客に行動の違いが見られる主な領域は 2 つあります。1 つ目は、NetScaler VPX は多くの NetScaler アプライアンスと同じパフォーマンスを提供できないことです。2 つ目は、NetScaler アプライアンスは独自の L2 ネットワーク機能を組み込んでいるが、NetScaler VPX は L2 ネットワークサービスのために Hypervisor に依存しているということです。一般的に、NetScaler VPX の展開方法は制限されません。物理 NetScaler アプライアンスに構成されている特定の L2 機能は、基盤となる Hypervisor で構成する必要があります。

NetScaler VPX は、アプリケーションデリバリー市場でどのように役割を果たしていますか**?**

NetScaler VPX は、アプリケーション配信市場のゲームを次のように変えます。

- NetScaler アプライアンスをさらに手頃な価格にすることで、NetScaler VPX は、あらゆる IT 組織が NetScaler アプライアンスを展開できるようにします。これは、最もミッションクリティカルな Web アプリケーションだけでなく、すべての Web アプリケーション用です。
- NetScaler VPX を使用すると、データセンター内でネットワーキングと仮想化をさらに統合できます。NetScaler VPX は、仮想サーバーでホストされている Web アプリケーションを最適化するためだけに使用することはできません。また、Web アプリケーションの配信自体を、どこでも簡単かつ迅速に展開できる仮想化サービスにすることができます。IT 組織は、Web アプリケーション配信インフラストラクチャのプロビジョニング、自動化、チャージバックなどのタスクに標準的なデータセンタープロセスを使用します。
- NetScaler VPX は、物理アプライアンスだけを使用する場合は実用的ではない新しい展開アーキテクチャを開きます。NetScaler VPX および NetScaler MPX アプライアンスは、圧縮やアプリケーションファイアウォール検査などのプロセッサ負荷の高いアクションを処理するために、各アプリケーションの個々のニーズに合わせてベースで使用できます。データセンターエッジでは、NetScaler MPX アプライアンスは、初期トラフィック分散、SSL 暗号化または復号化、サービス拒否（DoS）攻撃防止、グローバル負荷分散など、大量のネットワーク全体のタスクを処理します。高性能の NetScaler MPX アプライアンスと展開しやすい NetScaler VPX 仮想アプライアンスを組み合わせることで、データセンター全体のコストを削減しながら、最新の大規模データセンター環境に比類のない柔軟性とカスタマイズ機能を提供します。

NetScaler VPX はシトリック **Citrix** デリバリーセンター戦略にどのように適合していますか

NetScaler VPX を利用することで、Citrix デリバリーセンターの全サービスを仮想化されたサービスとして利用できます。Citrix XenCenter で利用可能な強力な管理、プロビジョニング、監視、およびレポート機能によって、Citrix デリバリーセンター全体がメリットを得られます。これは、ほぼすべての環境に迅速に導入でき、どこからでも一元的に管理できます。1 つの統合された仮想化アプリケーション配信インフラストラクチャにより、組織はデスクトップ、クライアント/サーバーアプリケーション、Web アプリケーションを配信できます。

暗号化

NetScaler VPX は **SSL** オフロードをサポートしていますか**?**

はい。ただし、NetScaler VPX はすべての SSL 処理をソフトウェアで行うため、NetScaler VPX は NetScaler アプライアンスと同じ SSL パフォーマンスを提供しません。NetScaler VPX は、毎秒最大 750 の新しい SSL トランザクションをサポートできます。

NetScaler VPX をホストするサーバーにインストールされているサードパーティの **SSL** カードは、**SSL** 暗号化または復号化を高速化しますか**?**

なしサードパーティの SSL カードをサポートしていると、NetScaler VPX を特定のハードウェア実装に関連付けることはできません。これにより、データセンター内の任意の場所で NetScaler VPX を柔軟にホストする組織の能力が大幅に低下します。NetScaler MPX アプライアンスは、NetScaler VPX が提供するよりも高い SSL スループットが必要な場合に使用する必要があります。

NetScaler VPX は、物理的な **NetScaler** アプライアンスと同じ暗号化暗号をサポートしていますか**?**

VPX は、ECDSA を除くすべての暗号化暗号を物理 NetScaler アプライアンスとしてサポートします。

NetScaler VPX SSL トランザクションスループットとは何ですか**?**

SSL トランザクションのスループットについては、NetScaler VPX のデータシートを参照してください 。

価格設定と梱包

NetScaler VPX はどのようにパッケージ化されていますか

NetScaler VPX の選択は、NetScaler アプライアンスの選択に似ています。まず、お客様は、機能要件に基づいて NetScaler エディションを選択します。次に、スループット要件に基づいて、特定の NetScaler VPX 帯域幅層を選択します。NetScaler VPX は、スタンダード、アドバンスエディション、およびプレミアムエディションで利用できます。NetScaler VPX は、10Mbps（VPX 10）から 100Gbps（VPX 100G）まで対応している。詳細については、NetScaler VPX のデータシートを参照してください。

NetScaler VPX の価格はすべての **Hypervisor** で同じですか**?**

はい。

すべての **Hypervisor** で **VPX** に同じ **NetScaler SKU** が使用されていますか**?**

はい。

NetScaler VPX ライセンスをある **Hypervisor** から別の **Hypervisor** に移動できますか（たとえば、**VMware** から **Hyper-V** へ）**?**

はい。NetScaler VPX ライセンスは、基盤となる Hypervisor から独立しています。NetScaler VPX 仮想マシンをある Hypervisor から別の Hypervisor に移動する場合は、新しいライセンスを取得する必要はありません。ただし、既存の NetScaler VPX ライセンスを再ホストする必要がある場合があります。

NetScaler VPX インスタンスはアップグレードできますか**?**

はい。スループット制限と NetScaler ファミリエディションの両方をアップグレードできます。両方のタイプのアップグレードのアップグレード SKU が利用可能です。

NetScaler VPX を高可用性ペアに展開する場合、必要なライセンスはいくつですか

NetScaler 物理アプライアンスと同様に、NetScaler 高可用性構成には 2 つのアクティブなインスタンスが必要です。したがって、お客様は 2 つのライセンスを購入する必要があります。

NetScaler VPX Express と 90 日間の無料トライアル

NetScaler VPX Express には、**NetScaler** 標準機能がすべて含まれていますか? **NetScaler Gateway**、および **Citrix Virtual Apps**（旧 **XenApp**）**Web** インターフェイスおよび **XML** ブローカーの負荷分散は含まれていますか

はい。NetScaler VPX Express には、NetScaler 標準のすべての機能が含まれています。NetScaler リリース 12.0～56.20 以降、Citrix は VPX Express の動作を変更しました。

NetScaler VPX Express には、**NetScaler** 標準機能がすべて含まれていますか? **NetScaler Gateway** と、**Citrix Virtual Apps Web** インターフェイスと **XML** ブローカーの負荷分散が含まれていますか

NetScaler リリース 12.0～56.20 以降、VPX Express は、ゲートウェイ機能を除く NetScaler スタンダードエディション機能セットを提供します。12.0—56.20 以前のリリースでは、VPX Express には標準エディションのすべての機能が含まれています。

NetScaler VPX Express にはライセンスが必要ですか

新しい NetScaler VPX Express リリース（12.0〜56.20 以降）では、VPX Express は無料で、インストールにライセンスファイルを必要とせず、コミットメントもありません。すでに VPX Express ライセンスをお持ちの場合、以前の VPX Express の動作は保持されます。VPX *Express* ライセンスファイルが削除され、12.0—56.20 以降のリリースが使用されている場合、新しい VPX Express の動作が有効になります。

NetScaler VPX Express ライセンスは期限切れになりますか

新しい VPX Express では、いいえ。ライセンスと有効期限はありません。VPX Express ライセンスをすでにお持ちの場合、ライセンスはダウンロード後 1 年で期限切れになります。

NetScaler VPX Express には、**5** つの無料の **NetScaler** ゲートウェイ同時実行ライセンスが含まれていますか

はい、VPX Express ライセンスを所有している場合は可能です。

顧客がダウンロードできる **NetScaler VPX Express** の数に制限はありますか

ファイブ。

NetScaler VPX Express は、**NetScaler MPX** アプライアンスと同じ暗号化暗号をサポートしていますか

一般的な可用性のために、NetScaler アプライアンスでサポートされている同じ強力な暗号化暗号はすべて、NetScaler VPX および NetScaler VPX Express で利用できます。これは、同じ輸出入規制の対象となります。

NetScaler VPX Express のテクニカルサポートケースを報告できますか

なしテクニカルサポートケースを提出するには、VPX-10、VPX-200、VPX-1000、VPX-3000 などの小売 NetScaler VPX ライセンスが必要です。ただし、NetScaler VPX Express ユーザーは、NetScaler VPX ナレッジセンターの両方を自由に使用でき、Z ディスカッションフォーラムを使用してコミュニティにヘルプをリクエストできます。

NetScaler VPX Express を製品版にアップグレードできますか**?**

はい。必要な小売 NetScaler VPX ライセンスを購入し、対応するライセンスを NetScaler VPX Express インスタンスに適用するだけです。

ハイパーバイザー

NetScaler VPX はどの **VMware** バージョンをサポートしていますか

NetScaler VPX は、バージョン 3.5 以降では、VMware ESX と ESXi の両方をサポートしています。詳細については、「サポートマトリックスと使用上のガイドライン」を参照してください。

VMware の場合、**VPX** に割り当てることができる仮想ネットワーク・インタフェースはいくつですか**?**

最大 10 個の仮想ネットワークインターフェイスを NetScaler VPX に割り当てることができます。

vSphere から、**NetScaler VPX** コマンドラインにどのようにアクセスできますか

VMware vSphere クライアントは、コンソールタブから NetScaler VPX コマンドラインへの組み込みアクセスを提供します。また、任意の SSH または Telnet クライアントを使用してコマンドラインにアクセスすることもできます。NetScaler VPX の NSIP アドレスは、SSH または Telnet クライアントで使用できます。

NetScaler VPX GUI にはどのようにアクセスできますか

NetScaler VPX GUI にアクセスするには、任意のブラウザーのアドレスフィールドに、NetScaler VPX の NSIP（たとえば、`http://NSIP address`）を入力します。

同じ **VMware ESX** にインストールされている **2** つの **NetScaler VPX** インスタンスを高可用性セットアップで構成できますか**?**

はい、でもお勧めできません。ハードウェア障害は、両方の NetScaler VPX インスタンスに影響します。

2 つの異なる **VMware ESX** システム上で実行されている **2** つの **NetScaler VPX** インスタンスを、高可用性セットアップで構成できますか**?**

はい。これは、高可用性セットアップで推奨されます。

VMware の場合、インターフェイス関連のイベントは **NetScaler VPX** でサポートされていますか

なしインターフェイス関連のイベントはサポートされていません。

VMware の場合、タグ付き **VLAN** は **NetScaler VPX** でサポートされていますか**?**

はい。NetScaler タグ付き VLAN は、リリース 11.0 以降の NetScaler VPX でサポートされています。詳しくは、NetScaler のドキュメントを参照してください。

VMware の場合、リンクアグリゲーションと **LACP** は **NetScaler VPX** でサポートされていますか**?**

なしリンクアグリゲーションと LACP は、NetScaler VPX ではサポートされていません。リンクアグリゲーションは VMware レベルで設定する必要があります。

NetScaler VPX ドキュメントにはどのようにアクセスするのですか

このドキュメントは、NetScaler VPX GUI から入手できます。ログイン後、[ドキュメント] タブを選択します。

キャパシティプランニングまたはサイジング

NetScaler VPX で期待できるパフォーマンスは何ですか

NetScaler VPX は、優れたパフォーマンスを提供します。NetScaler VPX を使用して達成可能な特定のパフォーマンスレベルについては、NetScaler VPX のデータシートを参照してください 。

サーバーの **CPU** パワーが変化することを考えると、**NetScaler** インスタンスの最大パフォーマンスをどのように見積もることができますか**?**

より高速な CPU を使用すると（ライセンスで許可されている最大値まで）パフォーマンスが向上しますが、低速の CPU を使用すると、パフォーマンスが確実に制限されます。

NetScaler VPX の帯域幅またはスループットの制限は、インバウンドのみのトラフィック、またはインバウンドとアウトバウンドの両方のトラフィックですか

NetScaler VPX 帯域幅制限は、要求トラフィックか応答トラフィックかにかかわらず、NetScaler への着信トラフィックにのみ適用されます。これは、NetScaler VPX-1000（たとえば）が 1 Gbps のインバウンドトラフィックと 1 Gbps のアウトバウンドトラフィックの両方を同時に処理できることを示します。インバウンドおよびアウトバウンドトラフィックは、要求および応答トラフィックと同じではありません。NetScaler では、エンドポイントからのトラフィック（リクエストトラフィック）とオリジンサーバーからのトラフィック（レスポンストラフィック）の両方が「インバウンド」（つまり、NetScaler に着信）です。

同じサーバー上で **NetScaler VPX** 複数のインスタンスを実行できますか**?**

はい。ただし、物理サーバーにホストで実行されている合計ワークロードをサポートするのに十分な CPU と I/O 容量があることを確認してください。そうしなければ NetScaler VPX のパフォーマンスに影響する可能性があります。

NetScaler VPX の複数のインスタンスが物理サーバーで実行されている場合、**NetScaler VPX** インスタンスごとの最小ハードウェア要件は何ですか**?**

各 NetScaler VPX インスタンスには、2 GB の物理 RAM、20 GB のハードディスク容量、および 2 つの vCPU を割り当てる必要があります。重要な展開では、システムがメモリに制約のある環境で動作するため、VPX に 2 GB の RAM を使用することはお勧めしません。これにより、スケール、パフォーマンス、または安定性に関連する問題が発生する可能性があります。推奨されるのは 4 GB の RAM または 8 GB の RAM です。

> 注:
>
> NetScaler VPX は、レイテンシーに敏感で高性能な仮想アプライアンスです。期待されるパフォーマンスを実現するには、アプライアンスに vCPU 予約、メモリ予約、ホストでの vCPU ピン接続が必要です。また、ホスト上でハイパースレッディングを無効にする必要があります。ホストがこれらの要件を満たさない場合、高可用性フェイルオーバー、VPX インスタンス内の CPU スパイク、VPX CLI へのアクセスにおける低速化、ピットボスデーモンのクラッシュ、パケットドロップ、低スループットなどの問題が発生します。
>
> すべての VPX インスタンスが事前定義された条件を満たしていることを確認してください。

NetScaler VPX と他のアプリケーションを同じサーバーでホストできますか**?**

はい。たとえば、NetScaler VPX、Citrix Virtual Apps Web インターフェイス、および Citrix Virtual Apps XML ブローカーはすべて仮想化でき、同じサーバー上で実行できます。最高のパフォーマンスを得るには、実行中のすべてのワークロードをサポートするのに十分な CPU および I/O 容量が物理ホストにあることを確認します。

単一の **NetScaler VPX** インスタンスに **CPU** コアを追加すると、そのインスタンスのパフォーマンスが向上しますか**?**

ライセンスに応じて、NetScaler VPX インスタンスは現在、最大 4 つの vCPU を使用できます。より多くの CPU を使用できる NetScaler VPX インスタンスに CPU を追加すると、パフォーマンスが向上します。

NetScaler VPX がアイドル状態のにもかかわらず、**CPU** の **90**%以上を消費しているように見えるのはなぜですか**?**

これは正常な動作であり、NetScaler アプライアンスは同じ動作を示します。NetScaler VPX CPU 使用率の実程度を確認するには、NetScaler CLI で stat CPU コマンドを使用するか、NetScaler GUI から NetScaler VPX CPU 使用率を表示します。NetScaler パケット処理エンジンは、やるべき作業がない場合でも、常に「仕事を探している」

ことです。したがって、CPU を制御し、それを解放しないためにすべてを行います。NetScaler VPX がインストールされたサーバーでは、NetScaler VPX が CPU 全体を消費しているような外観になります（Hypervisor の観点から）。CLI または GUI を使用した「NetScaler 内部」からの CPU 使用率を見ると、NetScaler VPX CPU 容量が使用されている様子が表示されます。

システム要件

NetScaler VPX 最小ハードウェア要件を教えてください

次の表では、NetScaler VPX 最小ハードウェア要件について説明しています。

種類	要件
プロセッサ	インテル Xeon または AMD EPYC を搭載したデュアルコアサーバー。
メモリ	最低 2 GB。ただし、4 GB が推奨されます。
ディスク	最低 20 GB のハードドライブ。
ハイパーバイザー	Citrix Hypervisor 5.6 以降、VMware ESX/ESXi 3.5 以降、または Hyper-V を搭載した Windows Server 2008 R2
ネットワーク接続性	最小値は 100 Mbps ですが、1 Gbps が推奨されます。
NIC	使用しているハイパーバイザーと互換性のある NIC。

> 注:
>
> 重要な導入環境では、NetScaler VPX には 4 GB のメモリが推奨されます。NetScaler VPX は 2 GB のメモリを搭載しているため、メモリの制約が非常に厳しい環境でも動作します。これにより、スケール、パフォーマンス、または安定性に関連する問題が発生する可能性があります。

システム要件の詳細については、NetScaler VPX のデータシートを参照してください。

> 注:
>
> NetScaler 13.1 リリース以降、VMware ESXi ハイパーバイザー上の NetScaler VPX インスタンスは AMD EPYC プロセッサをサポートしています。

Intel VT-X って何ですか

これらの機能は、「ハードウェアアシスト」または「仮想化アシスト」とも呼ばれ、ゲスト OS によって実行される機密または特権 CPU 命令をハイパーバイザにトラップします。これにより、Hypervisor でのゲスト OS（NetScaler VPX 用の BSD）のホスティングが簡単になります。

VT-X はどれくらい一般的ですか

事実上、過去 2 年以内に出荷されたすべてのサーバが VT-X をサポートしている可能性があります。多くのサーバには、BIOS で仮想化支援が無効になっている状態で出荷されます。NetScaler VPX を実行できないと仮定する前に、サーバーでこの設定を変更する必要があるかどうかを確認してください。

NetScaler VPX ハードウェア互換性リスト（**HCL**）はありますか**?**

サーバーが Intel VT-X をサポートしている限り、NetScaler VPX は、基盤となる Hypervisor と互換性のあるサーバーで実行する必要があります。サポートされるプラットフォームの包括的なリストについては、Hypervisor HCL を参照してください。

NetScaler VPX はどのバージョンの **NetScaler OS** をベースにしていますか

NetScaler VPX は、NetScaler 9.1 以降のリリースをベースにしています。

NetScaler VPX は **BSD** 上で動作するので、**BSD Unix** がインストールされているサーバーでネイティブに実行できますか

いいえ。NetScaler VPX を実行するには、Hypervisor が必要です。Hypervisor のサポートの詳細については、NetScaler VPX のデータシートを参照してください。

その他の技術的なよくある質問

複数の **NIC** を持つ物理サーバでのリンクアグリゲーションは機能しますか**?**

LACP はサポートされていません。Citrix Hypervisor では、静的リンクアグリゲーションがサポートされ、4 つのチャネルと 7 つの仮想インターフェイスの制限があります。VMware の場合、静的リンクアグリゲーションは NetScaler VPX 内ではサポートされていませんが、VMware レベルで構成できます。

MAC ベースの転送 **(MBF)** は **VPX** でサポートされていますか**? NetScaler** アプライアンスの実装から変更はありますか**?**

MBF はサポートされており、NetScaler アプライアンスと同じように動作します。Hypervisor は基本的に、NetScaler VPX から受信したすべてのパケットを外部に切り替え、逆に切り替えます。

NetScaler VPX のアップグレードプロセスはどのように実行されますか**?**

アップグレードは、NetScaler アプライアンスの場合と同じ方法で実行されます。カーネルファイルをダウンロードし、GUI で install ns またはアップグレードユーティリティを使用します。

フラッシュとディスク容量はどのように割り当てられますか。それを変えることはできますか

/flash = 965M
/var = 14G
各 NetScaler VPX インスタンスに最低 2 GB のメモリを割り当てる必要があります。NetScaler VPX ディスクイメージは、最大 4 GB のコアダンプ、ログファイル、トレースファイルを取得して格納するためのスペースなど、保守性を考慮して 20 GB のサイズにしました。これより小さいディスクイメージを生成することは可能ですが、現時点ではこれを実行する予定はありません。/flashおよび/varは両方とも同じディスクイメージ内にあります。互換性のために別々のファイルシステムとして保管されています。
メモリ割り当ての推奨事項の詳細については、NetScaler VPX データシートを参照してください。

新しいハードドライブを追加して、**NetScaler VPX** インスタンスのスペースを増やすことはできますか

はい。NetScaler リリース 13.1 ビルド 21.x 以降では、2 台目のディスクを追加して NetScaler VPX インスタンスのディスク容量を増やすことができます。2 番目のディスクを接続すると、「/var/crash」ディレクトリが自動的にこのディスクにマウントされます。2 つ目のディスクは、コアファイルの保存とロギングに使用されます。コアファイルとログファイルの保存に使用される既存のディレクトリは、以前と同様に機能します。

> 注:
>
> データの損失を防ぐために、NetScaler アプライアンスのダウングレード時に外部バックアップを作成します。

クラウド上の NetScaler VPX インスタンスに新しいハードディスクドライブ（HDD）を接続する方法については、以下を参照してください。

- Azure ドキュメンテーション

  > 注:
  >
  > Azure にデプロイされた VPX インスタンスにセカンダリディスクを接続するには、Azure VM のサイズにローカルの一時ディスクがあることを確認してください。詳細については、「ローカル一時ディスクなしの Azure VM サイズ」を参照してください。

- AWS ドキュメント
- GCP ドキュメント

> 警告:
>
> 新しい HDD を VPX に追加した後、新しい HDD に移動されたファイルで動作するスクリプトの一部が、次の条件で失敗することがあります。
>
> 「link」シェルコマンドを使用して、新しい HDD に移動されたファイルへのハードリンクを作成した場合。
>
> シンボリックリンクを使用するには、そのようなコマンドはすべて「ln-s」に置き換えなければなりません。また、失敗したスクリプトを適宜修正してください。

NetScaler VPX ビルドの番号付けと、他のビルドとの相互運用性に関して、どのようなことが期待できますか**?**

NetScaler VPX には、9.1 と同様のビルド番号が付けられています。Cl (クラシック) と 9.1. Nc (nCore) リリース。例えば 9.1_97.3.vpx、9.1_97.3.nc、9.1_97.3.cl。

NetScaler VPX を **NetScaler** アプライアンスを使用した高可用性セットアップの一部にすることはできますか**?**

サポートされていない構成です。

NetScaler VPX に表示されるすべてのインターフェイスは、**Hypervisor** 上のインターフェイスの数に直接関係していますか

いいえ。Hypervisor 上に物理 NIC が 1 つしかない NetScaler VPX 構成ユーティリティを使用して、最大 7 つのインターフェイス（VMware では 10）を追加できます。

Citrix Hypervisor XenMotion または **VMware VMotion** または **Hyper-V** ライブマイグレーションを使用して、**NetScaler VPX** アクティブなインスタンスを移動できますか

NetScaler VPX は、XenMotion または Hyper-V ライブマイグレーションをサポートしていません。vMotion は、NetScaler 12.1 リリース以降からサポートされています。詳細については、「リリースノート」を参照してください。

ライセンスサーバーの概要

March 20, 2024

NetScaler は、組織のニーズに合わせて、MPX および VPX アプライアンス向けの幅広い製品エディションとライセンスモデルを提供しています。

NetScaler アプライアンスを適切に動作させるには、NetScaler ファミリエディションライセンスの 1 つが必要です。ADC 製品ラインには、次の 3 つのファミリーエディションがあります:

- Standard Edition

  > 注:
  >
  > Standard エディションは販売終了 (EOS) に達しており、更新のみ可能です。

- Advanced Edition
- Premium Edition

詳細については、データシートを参照してください。データシートは www.netscaler.com で入手できます。

NetScaler のエディションを選択します。次に、次の基準に基づいて MPX または VPX ライセンスオファリングを選択します。

- 永久およびサブスクリプション (年間および時間単位のサブスクリプション)
- vCPU と帯域幅
- オンプレミスとクラウド

NetScaler VPX Express ライセンス

オンプレミスおよびクラウド展開用の VPX Express は、ライセンスファイルを必要とせず、次の機能を提供します:

- 20Mbps の帯域幅
- NetScaler Gateway および L4 および L7 防御を除く、すべての ADC 標準ライセンス機能
- 最大 250 の SSL セッション
- 20Mbps の SSL スループット

VPX Express ライセンスを次の 2 つのオプションにアップグレードできます。

1. スタンドアロンの NetScaler VPX ライセンス。
2. VPX インスタンス用の NetScaler プールキャパシティライセンス。詳しくは、「NetScaler プールキャパシティ」を参照してください。

> 重要
>
> クラスタリングは、VPX パブリッククラウドの Standard Edition、および VPX Express ライセンスで利用できます。

NetScaler プールキャパシティライセンス

NetScaler Application Delivery Management（ADM）を使用して、共通の帯域幅とインスタンスプールで構成されるライセンスフレームワークを作成します。詳しくは、「NetScaler プールキャパシティ」を参照してください。

> 注:
>
> NetScaler Console は、プール型ライセンスとセルフマネージド型プールライセンスの両方をホストできます。必要なライセンスを使用するには、NetScaler でライセンスサーバーを構成し、適切なプールから容量をチェックアウトします。プールライセンスと自己管理プールライセンスの ADC CLI と GUI の設定手順は同じです。

NetScaler 自己管理プールライセンス

NetScaler リリース 13.1 ビルド 30.x 以降、NetScaler インスタンスはセルフマネージドプールライセンスをサポートします。このライセンスを使用すると、ライセンスサーバーへのライセンスファイルのアップロードを簡略化および自動化できます。NetScaler Console を使用して、共通の帯域幅または vCPU とインスタンスプールで構成されるライセンスフレームワークを作成します。

自己管理プールライセンスを使用するには、NetScaler でライセンスサーバーを`SelfManagedPool`ライセンスモードに構成し、必要な容量を確認します。NetScaler アプライアンスを再起動した後に`show ns license`コマンドを使用して、構成されているライセンスを確認します。

> 重要
>
> システムがプールキャパシティライセンスで構成されているが、トラフィックフローに影響を与えずに自己管理プールライセンスに移行する場合は、ターゲットサーバーに必要な自己管理プールライセンスがあることを確認します。
>
> 次の互換性のあるライセンス間でのみ移行できます。
>
> - 容量を自己管理プールにプールし、逆にプールしました。
> - vCPU からセルフマネージド vCPU へ、そして逆に。
>
> ライセンスを移行するには、次のコマンドを実行します。
>
> ```
> add ns licenseserver (<licenseServerIP> | <serverName>)-forceUpdateIP
>  -licensemode [CICO | Pooled | SelfManagedPool | VCPU | SelfManagedvCPU
> ]
> ```
>
> 例:
>
> ```
> add licenseserver 192.0.2.246 -forceUpdateIP -licensemode selfManagedvcPU
> ```


CLI を使用して自己管理プールライセンスを設定する

ライセンスサーバー構成を NetScaler アプライアンスに追加するには、次のコマンドを実行します。

```
add ns licenseserver (<licenseServerIP> | <serverName>) [-port <
    positive_integer>] -licensemode [CICO | Pooled | SelfManagedPool |
    VCPU | SelfManagedvCPU]
<!--NeedCopy-->
```

例:

```
add ns licenseserver 192.0.2.246 -port 27000 -licensemode
    SelfManagedPool
<!--NeedCopy-->
```

> 注:
>
> `show ns licenseserverpool`コマンドは、指定したライセンスモードに基づくライセンスのみを表示します。したがって、ライセンスはより速くフェッチされます。すべてのライセンスのインベントリを取得するには、`show ns licenseserverpool -getallLicenses` コマンドを実行します。ライセンスモードが指定されていない場合、プールキャパシティライセンスがデフォルトで表示されます。

システム容量を変更するには、次のコマンドを実行します。

```
set ns capacity ((-bandwidth <positive_integer> -unit ( Gbps | Mbps ))
    | -platform <platform>) [-Edition <Edition>]
<!--NeedCopy-->
```

例:

```
set ns capacity -bandwidth 3 -unit gbps -edition enterprise
<!--NeedCopy-->
```

> 注:
>
> キャパシティは、ライセンスサーバーのライセンスプールからチェックアウトされます。

NetScaler アプライアンスを再起動するには、次のコマンドを実行します。

```
reboot [-warm]
<!--NeedCopy-->
```

ライセンスされているすべての機能と設定されているライセンスモードの状態を表示するには、次のコマンドを実行します。

```
show ns license
<!--NeedCopy-->
```

`show ns licenseserverpool` コマンドのサンプル出力**:**


```
> add licenseserver                    -licensemode SelfManagedPool
 Done
> sh licenseserverpool
        Instance Total                          : 200
        Instance Available                      : 199
        Standard Bandwidth Total                : 10.00 Gbps
        Standard Bandwidth Available            : 10.00 Gbps
        Enterprise Bandwidth Total              : 10.00 Gbps
        Enterprise Bandwidth Available          : 7.00 Gbps
        Platinum Bandwidth Total                : 0 Mbps
        Platinum Bandwidth Available            : 0 Mbps
```

show ns licenseserverpool -getallLicenses コマンドのサンプル出力**:**

```
> sh licenseserverpool -getallLicenses
        Instance Total                          : 40
        Instance Available                      : 33
        Standard Bandwidth Total                : 210.00 Gbps
        Standard Bandwidth Available            : 210.00 Gbps
        Enterprise Bandwidth Total              : 50.00 Gbps
        Enterprise Bandwidth Available          : 50.00 Gbps
        Platinum Bandwidth Total                : 210.00 Gbps
        Platinum Bandwidth Available            : 205.00 Gbps
        VPX8000P Total                          : 1
        VPX8000P Available                      : 1
        Standard CPU Total                      : 100
        Standard CPU Available                  : 100
        Enterprise CPU Total                    : 100
        Enterprise CPU Available                : 100
        Platinum CPU Total                      : 25
        Platinum CPU Available                  : 20
```

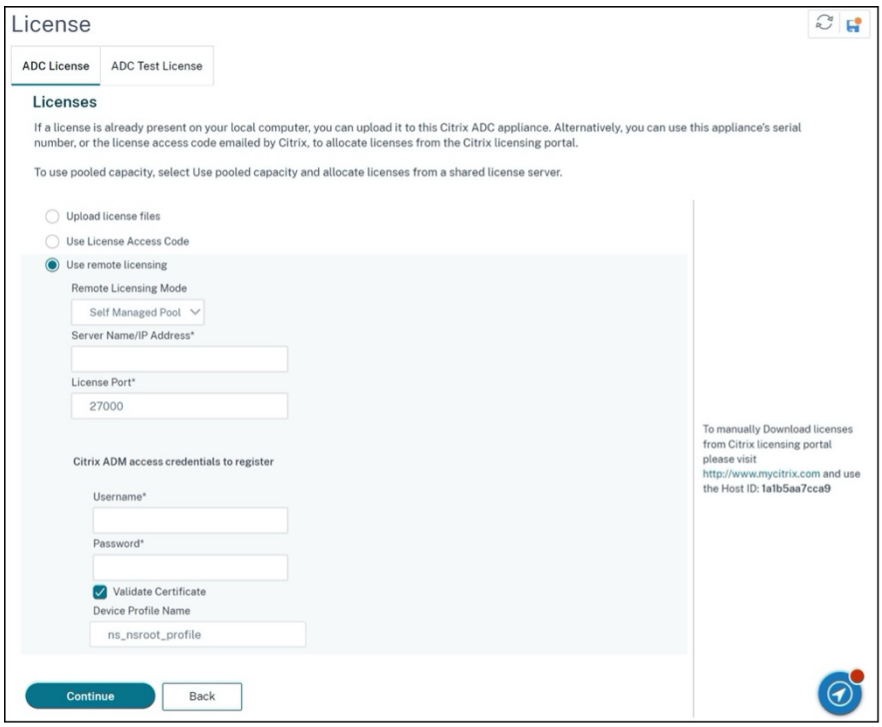
show license コマンドのサンプル出力**:**

```
> show license
        License status:
                        Web Logging: YES
                    Surge Protection: YES
                      Load Balancing: YES
                   Content Switching: YES
                   Cache Redirection: YES
                  Compression Control: YES
                     Delta Compression: NO
                       SSL Offloading: YES
          Global Server Load Balancing: YES
                        GSLB Proximity: YES
                     Dynamic Routing: YES
                   Content Filtering: YES
                 Content Accelerator: NO
                  Integrated Caching: NO
                             SSL VPN: YES  (Maximum users = 1000)  (Maximum ICA users = Unlimited)
                                 AAA: YES
                        OSPF Routing: YES
                         RIP Routing: YES
                         BGP Routing: YES
                             Rewrite: YES
           IPv6 protocol translation: YES
                Application Firewall: NO
                           Responder: YES
                      NetScaler Push: YES
                             AppFlow: YES
                          CloudBridge: NO
                        ISIS Routing: YES
                          Clustering: YES
                            CallHome: YES
                               AppQoE: YES
                     Appflow for ICA: YES
              Front End Optimization: YES
                     Large Scale NAT: YES
                           MDP Proxy: YES
                          Reputation: NO
                      URL Filtering: NO
                  Video Optimization: NO
                       Forward Proxy: NO
                  SSL Interception: NO
            Remote Content Inspection: YES
                      Adaptive TCP: NO
          Connection Quality Analytics: NO
                      Bot Management: NO
                         API Gateway: NO
                    Model Number ID: 3000
                       License Type: Enterprise License
                    Licensing mode: Self Managed Pool
Done
```


GUI を使用して自己管理プールライセンスを構成する

次の手順を実行して、自己管理プールライセンスを構成します。

1. システム **>** ライセンス **> ADC** ライセンス **>** ライセンスの管理 **>** 新しいライセンスの追加に移動します。
2. [ライセンス] ページで、[リモートライセンスの使用] ラジオボタンを選択し、[リモートライセンスモード] からライセンスモードを選択します。
3. サーバーの IP アドレスとライセンスポートの詳細を入力します。
4. NetScaler コンソールのアクセス資格情報を入力します。
5. [続行] をクリックします。



関連情報

Citrix ライセンスシステム

クラウドでの **VPX** ライセンス

VPX デプロイメントは、Azure、AWS、Google などのパブリッククラウドプロバイダーでサポートされています。詳しくは、次のドキュメントを参照してください。

- VPX-Azure ライセンス
- VPX-AWS ライセンス
- VPX-GCP ライセンス

ライセンスを割り当てて適用する

February 15, 2024

NetScaler MPX および VPX ADC GUI では、ハードウェアシリアル番号（HSN）またはライセンスアクセスコードを使用してライセンスを割り当てることができます。または、ローカルコンピューターにライセンスが既に存在する場合は、それをアプライアンスにアップロードできます。

ライセンスの返却や再割り当てなど、他のすべての機能については、ライセンスポータルを使用する必要があります。オプションで、ライセンスの割り当てにライセンスポータルを引き続き使用できます。詳細については、「ライセンスの管理」を参照してください。

Citrix ライセンスガイド

Citrix ライセンスガイドには、NetScaler アプライアンスへのライセンスのインストールや他の NetScaler 製品へのライセンスのインストールに関する情報も記載されています。詳しくは、「NetScaler ライセンスガイド」を参照してください。

前提条件

> 注
>
> 高可用性ペアのアプライアンスごとに個別のライセンスを購入します。同じタイプのライセンスが両方のアプライアンスにインストールされていることを確認してください。たとえば、あるアプライアンスのプレミアムライセンスを購入する場合、もう一方のアプライアンス用に別のプレミアムライセンスを購入する必要があります。

ハードウェアシリアル番号またはライセンスアクセスコードを使用してライセンスを割り当てるには、次の手順を実行します。

- アプライアンスを介してパブリックドメインにアクセスできる必要があります。たとえば、アプライアンスは www.citrix.com にアクセスできる必要があります。ライセンス割り当てソフトウェアは、ライセンスの Citrix ライセンスポータルに内部的にアクセスします。パブリックドメインにアクセスするには:
  - プロキシサーバーを使用するか、DNS サーバーを設定します。
  - NetScaler アプライアンスで NetScaler IP（NSIP）アドレスまたはサブネット IP（SNIP）アドレスを構成します。
- ライセンスはハードウェアにリンクされているか、有効なライセンスアクセスコードを持っている必要があります。ライセンスを購入すると、Citrix からライセンスアクセスコードが電子メールで送信されます。

GUI を使用してライセンスを割り当てる

ライセンスがすでにハードウェアにリンクされている場合は、ライセンス割り当てプロセスでハードウェアシリアル番号を使用できます。それ以外の場合は、ライセンスアクセスコードを入力する必要があります。

展開の必要に応じて、ライセンスを部分的に割り当てることができます。たとえば、ライセンスファイルに 10 個のライセンスが含まれていて、現在の要件が 6 つのライセンスだけである場合、ここで 6 つのライセンスを割り当てて、後でさらにライセンスを割り当てることができます。ライセンスファイルに存在するライセンスの合計数を超えるライセンスを割り当てることはできません。

ライセンスを割り当てるには

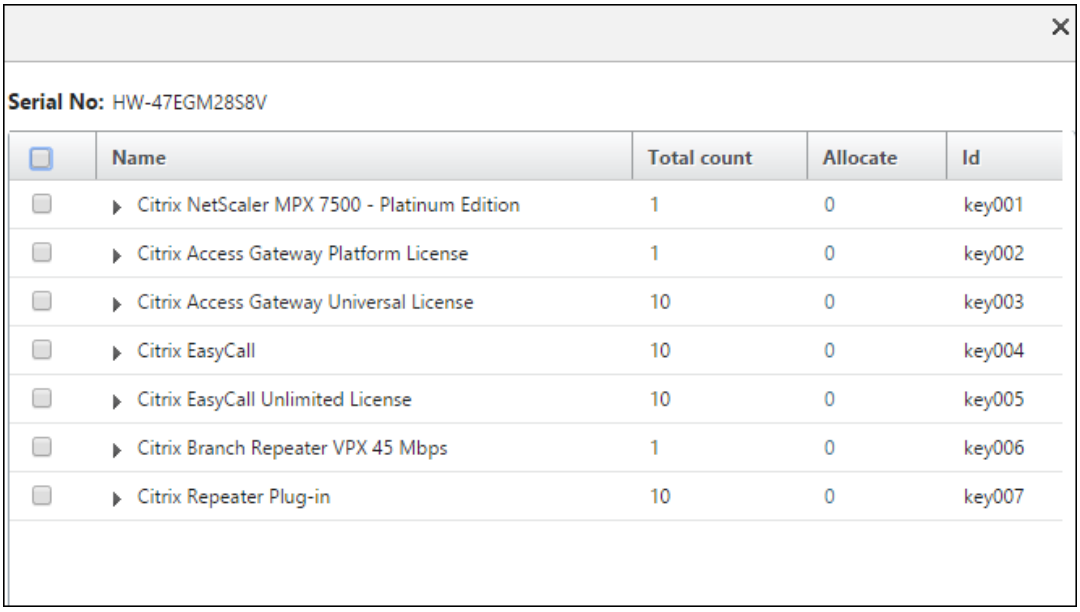
1. Web ブラウザーに NetScaler アプライアンスの IP アドレスを入力します（例: `http://192.168.100.1`）。
2. [User Name] ボックスと [Password] ボックスに管理者資格情報を入力します。
3. [**Configuration**] タブで、[**System**] **>** [**Licenses**] の順に移動します。
4. 詳細ウィンドウで、[ライセンスの管理]、[新しいライセンスの追加] の順にクリックし、次のいずれかのオプションを選択します。

    - シリアル番号を使用: ソフトウェアはアプライアンスのシリアル番号を内部的に取得し、この番号を使用してライセンスを表示します。
    - ライセンスアクセスコードを使用する: Citrix は、購入したライセンスのライセンスアクセスコードを電子メールで送信します。テキストボックスにライセンスアクセスコードを入力します。

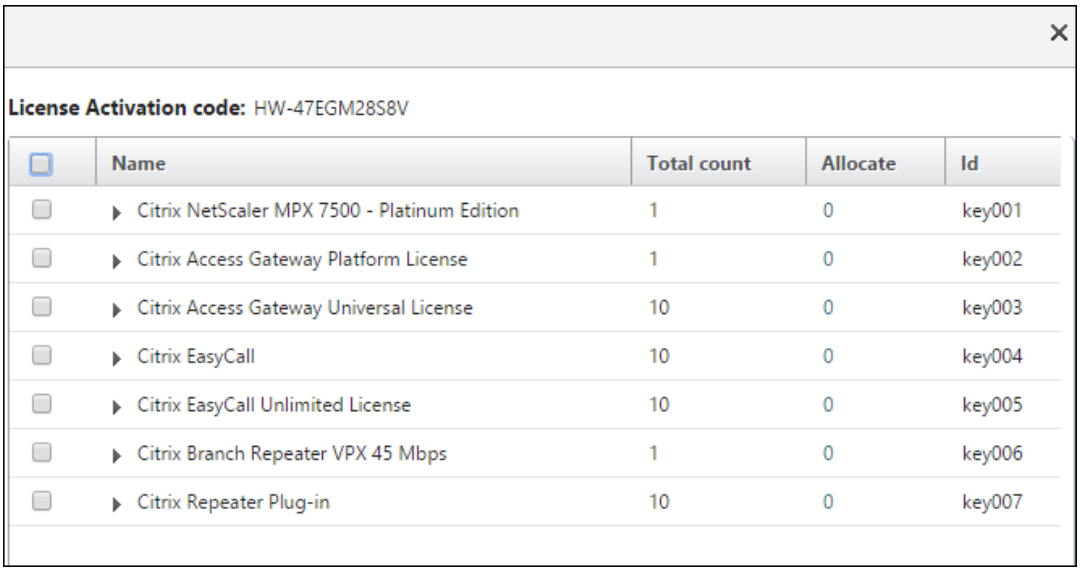
      NetScaler アプライアンスでインターネット接続を構成しない場合は、プロキシサーバーを使用できます。[**Connect through Proxy Server**] チェックボックスを選択し、プロキシサーバーの IP アドレスとポートを指定します。

5. [**Get Licenses**] をクリックします。選択したオプションに応じて、次のいずれかのダイアログボックスが表示されます。

    - [ハードウェアシリアル番号] を選択すると、次のダイアログボックスが表示されます。

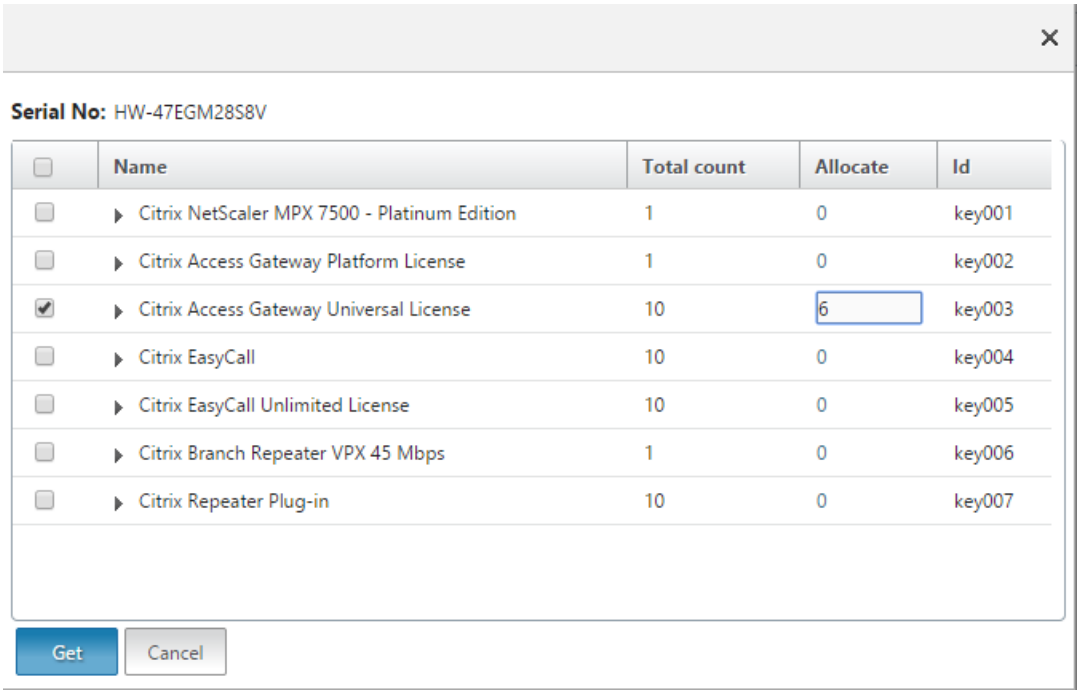


- ライセンスアクセスコードを選択した場合は、次のダイアログボックスが表示されます。

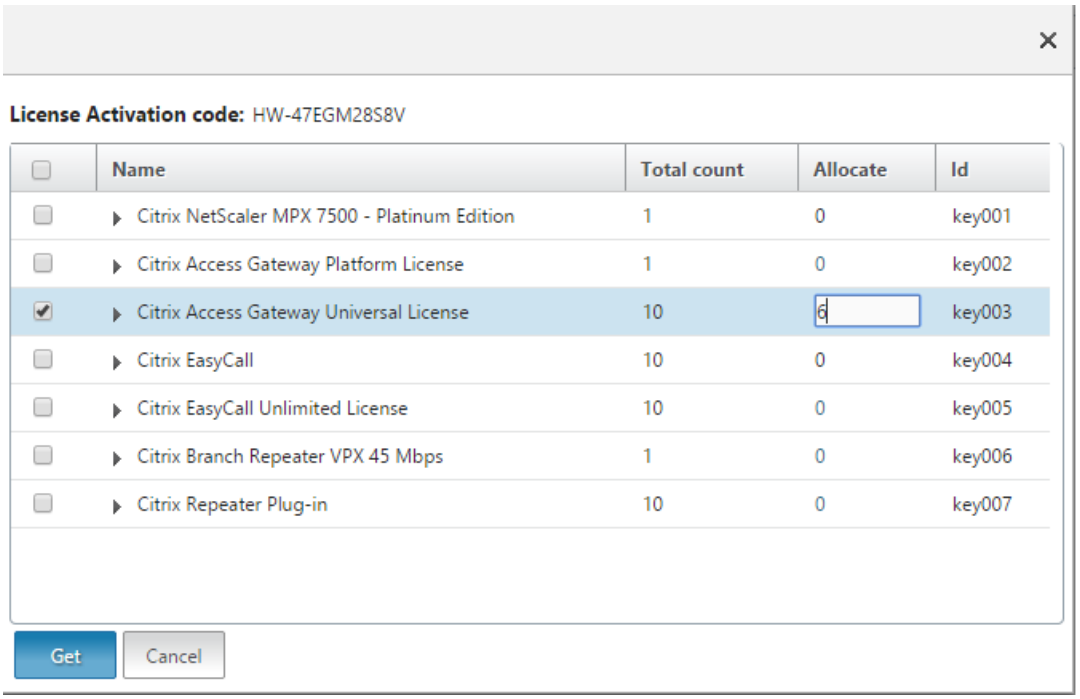


6. ライセンスの割り当てに使用するライセンスファイルを選択します。
7. [**Allocate**] 列に、割り当てるライセンスの数を入力します。次に、[取得] をクリックします。

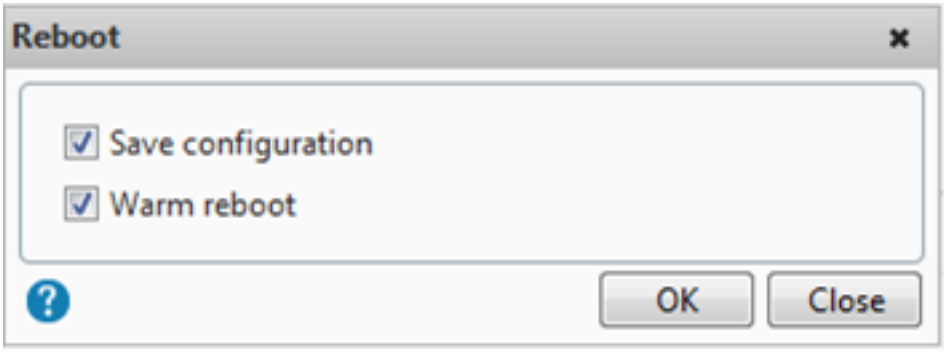
   - ハードウェアシリアル番号（Hardware Serial Number）を選択した場合は、次のスクリーンショットに示すように、ライセンス数を入力します。



- ライセンスアクセスコードを選択した場合は、次のスクリーンショットに示すように、ライセンス数を入力します。



8. ライセンスを有効にするには、[restart]をクリックします。
9. 再起動ダイアログボックスで、「**OK**」をクリックして変更を続行するか、「閉じる」をクリックして変更をキャンセルします。



ライセンスをインストールする

ライセンスポータルにアクセスしてライセンスファイルをローカルコンピュータにダウンロードした場合は、ライセンスをアプライアンスにアップロードする必要があります。

GUI を使用してライセンスファイルをインストールするには

1. Web ブラウザーに NetScaler アプライアンスの IP アドレスを入力します（例: `http://192.168.100.1`）。
2. [User Name] ボックスと [Password] ボックスに管理者資格情報を入力します。
3. [構成] タブで、[システムライセンス] に移動します。
4. 詳細ペインで、[**Manage Licenses**] をクリックします。
5. [**Add New License**] をクリックし、[**Upload license files from a local computer**] を選択します。
6. [**Browse**] をクリックします。ライセンスファイルの場所に移動し、ライセンスファイルを選択して、[**Open**] をクリックします。
7. [restart] をクリックしてライセンスを適用します。
8. 再起動ダイアログボックスで、「**OK**」をクリックして変更を続行するか、「閉じる」をクリックして変更をキャンセルします。

CLI を使用してライセンスをインストールするには

1. PuTTY などの **SSH** クライアントを使用して、**ADC** アプライアンスへの **SSH** 接続を開きます。
2. 管理者の資格情報を使用して ADC アプライアンスにログオンします。
3. シェルプロンプトに切り替え、`nsconfig`ディレクトリにライセンスサブディレクトリを作成します（存在しない場合）。このディレクトリに 1 つ以上の新しいライセンスファイルをコピーします。

   例

   ```
   1 login: nsroot
   2 Password: nsroot
   ```



```
Last login: Mon Aug  4 03:37:27 2008 from 10.102.29.9
Done
> shell
Last login: Mon Aug  4 03:51:42 from 10.103.25.64
root@ns# mkdir /nsconfig/license
root@ns# cd /nsconfig/license
<!--NeedCopy-->
```

1 つ以上の新しいライセンスファイルをこのディレクトリにコピーします。

> 注
>
> NetScaler アプライアンスは、コマンドラインインターフェイスを使用してライセンスをインストールするときに、再起動オプションの入力を求めません。reboot-w コマンドを実行してシステムをウォームリスタートするか、restart コマンドを実行してシステムを正常に再起動します。

ライセンスされた機能を確認する

機能を使用する前に、ライセンスがその機能をサポートしていることを確認してください。

CLI を使用してライセンスされた機能を確認するには

1. PuTTY などの **SSH** クライアントを使用して、**ADC** アプライアンスへの **SSH** 接続を開きます。
2. 管理者の資格情報を使用して ADC アプライアンスにログオンします。
3. コマンドプロンプトで sh ns license コマンドを入力して、ライセンスでサポートされている機能を表示します。

   例

```
sh ns license
        License status:
                            Web Logging: YES
                        Surge Protection: YES
                          .......

                          Responder: YES
 Done
<!--NeedCopy-->
```

GUI を使用してライセンスされた機能を確認するには

1. Web ブラウザで、ADC アプライアンスの IP アドレス（http://192.168.100.1など）を入力します。
2. [User Name] ボックスと [Password] ボックスに管理者資格情報を入力します。
3. [ユーザー名] と [パスワード] を入力し、[ログイン] をクリックします。

4. ナビゲーションペインで、[システム] を展開し、[ライセンス] をクリックします。ライセンスされた機能の横に緑色のチェックマークが表示されます。

機能を有効または無効にする

NetScaler アプライアンスを初めて使用するときは、その機能を使用する前に機能を有効にする必要があります。機能を有効にする前に設定すると、警告メッセージが表示されます。設定は保存されますが、機能が有効になった後にのみ適用されます。

CLI を使用して機能を有効にするには

コマンドプロンプトで次のコマンドを入力して、機能を有効にして構成を確認します。

- enable feature <FeatureName>
- show feature

例

```
enable feature lb cs
done
 >show feature

        Feature                         Acronym
           Status
        -------                         -------
           ------
 1)     Web Logging                     WL                      OFF
 2)     Surge Protection                SP                      ON
 3)     Load Balancing                  LB                      ON
 4)     Content Switching               CS                      ON
 5)     Cache Redirection               CR                      ON
 .
 .
 .
 24)    NetScaler Push                  push                    OFF
 Done
<!--NeedCopy-->
```

次に、負荷分散（lb）とコンテンツスイッチング（cs）を有効にする例を示します。

特定の機能でライセンスキーを使用できない場合、その機能について次のエラーメッセージが表示されます。

エラー: 機能がライセンスされていません

注: オプション機能を有効にするには、機能固有のライセンスが必要です。たとえば、NetScaler Advanced Edition ライセンスを購入してインストールしたとします。ただし、統合キャッシュ機能を有効にするには、AppCache ライセンスを購入してインストールする必要があります。

CLI を使用して機能を無効にするには

コマンドプロンプトで次のコマンドを入力して、機能を無効にし、構成を確認します:

- disable feature <FeatureName>
- show feature

  例

  次に、負荷分散（LB）を無効にする例を示します。

```
> disable feature lb
 Done
> show feature

        Feature                         Acronym
           Status
        -------                         -------
           ------
 1)     Web Logging                     WL                      OFF
 2)     Surge Protection                SP                      ON
 3)     Load Balancing                  LB                      OFF
 4)     Content Switching               CS                      ON
 .
 .
 .
 24)    NetScaler Push                  push                    OFF
 Done
>
<!--NeedCopy-->
```

NetScaler ライセンス有効期限アラートを構成する

デフォルトでは、ADC ライセンスの有効期限が 30 日以下になると、GUI アラートが表示されます。

NetScaler ライセンスの有効期限が切れる前の指定された日数から、次のアラート操作を実行するように NetScaler アプライアンスを構成できます。

- NetScaler GUI にライセンス有効期限の警告バナーを表示します。
- 「NS_LICENSE_EXPIRY」SNMP アラームが有効な場合は、ライセンスの有効期限情報を含む SNMP トラップを設定済みのトラップリスナーに定期的に送信します。

ライセンスの有効期限が切れると、NetScaler アプライアンスは自動的に再起動してライセンスを取り消します。NetScaler アプライアンスが Citrix サービスプロバイダー（CSP）ライセンスを使用している場合、アプライアンスは自動的に再起動してライセンスを取り消しません。ただし、ユーザーがアプライアンスを再起動すると、ライセンスなしとして再起動します。

CLI を使用して **NetScaler** ライセンスの有効期限切れアラートの日数を指定するには:

コマンドプロンプトで入力します:

- ライセンスパラメータの設定 **[-ライセンス期限切れアラート時間 XTM ポジティブ _ 整数 $ スタート $**]**
- **sh licenseparameters**

例:

```
> set licenseparameters -licenseexpiryalerttime 200
Done

> sh licenseparameters
...
    Licenseexpiryalerttime:  200
<!--NeedCopy-->
```

NetScaler GUI を使用して **NetScaler** ライセンス有効期限アラートの日数を指定するには:

1. 構成 > システム > ライセンス **>** ライセンスの管理に移動します **。
2. [通知設定] で、[編集] ボタンをクリックして、NetScaler ライセンスの有効期限アラートの日数を指定します。

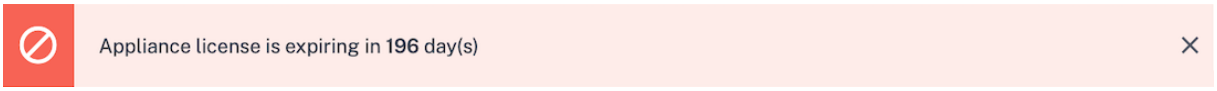
ライセンスの有効期限情報を確認する

NetScaler ライセンスの有効期限情報は、GUI または CLI で確認できます。

GUI を使用して **NetScaler** ライセンスの有効期限情報を確認するには:

設定 **>** システム **>** ライセンスに移動します。

ADC ライセンスの有効期限が、NetScaler ライセンスの有効期限アラートに指定された日数以下になると、GUI アラートが表示されます。



CLI を使用してライセンスの有効期限情報を確認するには、次の手順を実行します。

コマンド「show ns license」を入力します。

```
> sh license
    License status:

    Web Logging: YES
    Surge Protection: YES

    Web Logging: YES
    Surge Protection: YES

    ...
```



```
11
12 Days to expiry: 196
13
14 Done
15 >
16 <!--NeedCopy-->
```

NetScaler アプライアンスを再起動せずにライセンスファイルを検証する

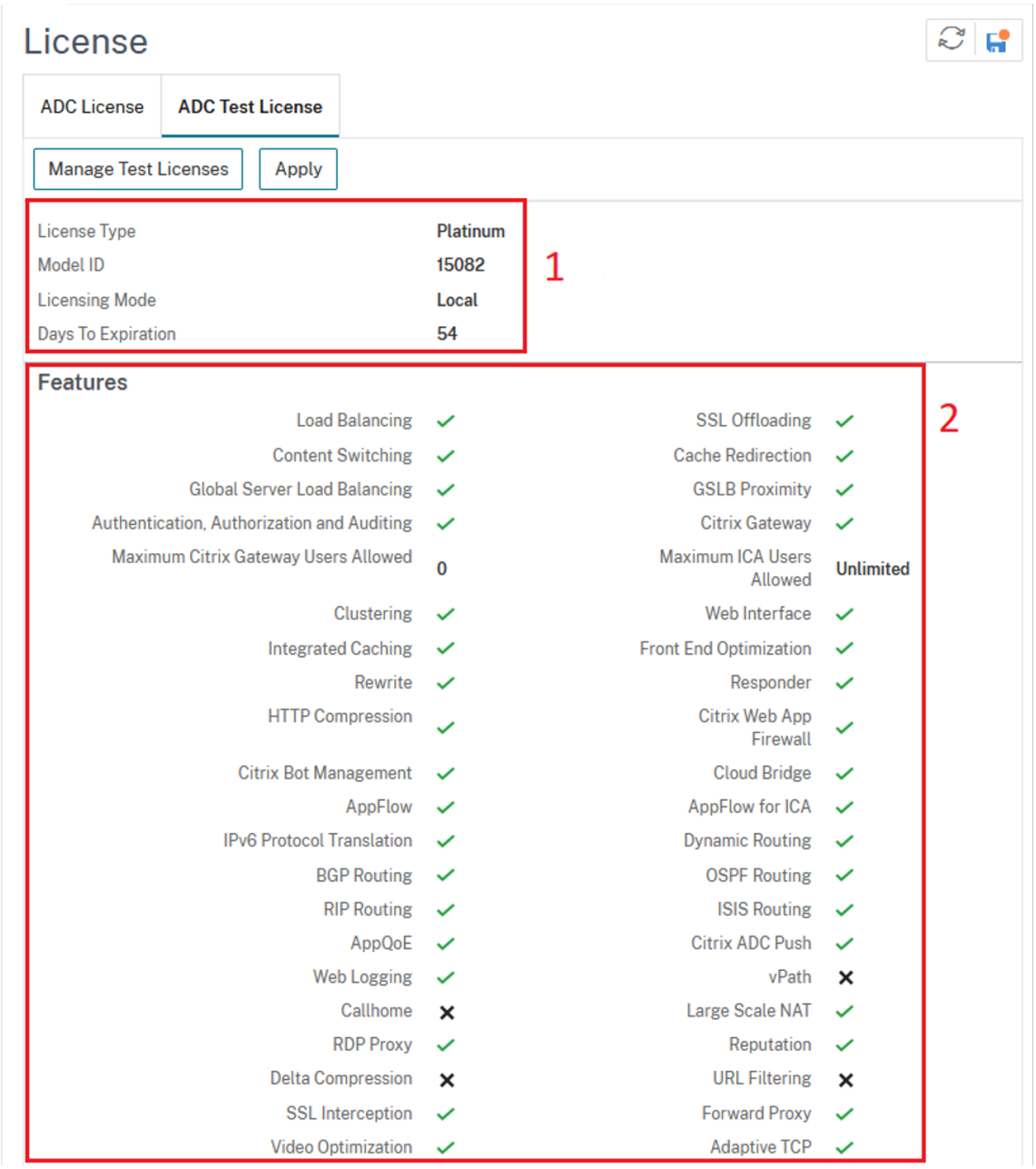
この機能を使用すると、NetScaler アプライアンスに適用しなくても、ライセンスをテストして、特定のライセンスで使用できるすべての機能を確認できます。このオプションを使用すると、NetScaler アプライアンスを再起動せずに新しいライセンスをテストできます。

この機能は GUI と CLI の両方で使用できます。

GUI を使用してライセンスファイルを検証する

1. **[**システム**]-> [**ライセンス**]** に移動します。
2. **ADC** テストライセンスタブで、「テストライセンスの管理」をクリックします。
3. [アップロード] をクリックし、1 つまたは複数のライセンスファイルをアップロードします。複数のライセンスファイルをアップロードした場合、すべてのライセンスファイルの和集合が計算されます。
4. ライセンスファイルのアップロードが完了したら、再度 **ADC Test License** をクリックすると、アップロードされたライセンスのライセンス機能が表示されます。

   セクション 1 にはライセンス情報が表示され、セクション 2 にはライセンスに含まれるすべての機能が表示されます。



5. 表示された情報を確認し、[適用] をクリックしてライセンスを使用してください。NetScaler アプライアンスを再起動（ウォーム）してライセンスを有効にします。即時再起動は必須ではなく、現在のライセンスは次回の再起動まで適用されます。

CLI を使用してライセンスファイルを検証する

1. テストライセンスファイルを ADC アプライアンスの path: /nsconfig/testlicenseにコピーします。

   例:


```
scp CNS_15082_SERVER_PLT_Retail.lic nsroot@<ns_ip>:/nsconfig/
   testlicense/
<!--NeedCopy-->
```

2. ライセンスファイルが正しい場所にコピーされているかどうかを確認してください。

   例:

```
ls /nsconfig/testlicense/  CNS_15082_SERVER_PLT_Retail.lic
<!--NeedCopy-->
```

3. `show ns testlicense` コマンドを実行してライセンス情報を確認します。

```
> sh ns testlicense
    License status:
                         Web Logging: YES
                   Surge Protection: YES
                     Load Balancing: YES
                  Content Switching: YES
                  Cache Redirection: YES
                Compression Control: YES
                  Delta Compression: NO
                     SSL Offloading: YES
      Global Server Load Balancing: YES
                                .....
                         API Gateway: YES
                   Model Number ID: 15082
                      License Type: Platinum License
                     Licensing mode: Local
                Days to expiration: 54
<!--NeedCopy-->
```

4. 表示された情報を確認し、`apply ns testlicense` コマンドを実行してライセンスを適用します。NetScaler アプライアンスを再起動（ウォーム）してライセンスを有効にします。

```
> apply ns testlicense

Warning: The configuration changes will not take effect until the
   system is rebooted
Done
> reboot -w
Are you sure you want to restart NetScaler (Y/N)? [N]:Y
Done
<!--NeedCopy-->
```

ライセンスをアップグレードする

大容量のライセンスを購入することで、NetScaler アプライアンスをあるファミリエディションから別のファミリエディションに、およびある容量範囲から別の容量範囲にアップグレードできます。

アップグレードには次の 2 つのタイプがあります。

- エディションのアップグレード：スタンダードからアドバンス、スタンダードからプレミアム、アドバンスからプレミアム。エディションのアップグレードは、同じ帯域幅内である必要があります。
- 容量のアップグレード：vCPU と帯域幅の両方について、容量を小さいものから大きいものにアップグレードできます。容量のアップグレードは、同じエディション（スタンダード、アドバンス、プレミアム）でのみ実行できます。

キャパシティとエディションの両方をアップグレードする場合は、まずキャパシティをアップグレードし、アプライアンスを再起動してから、エディションをアップグレードします。

例:VPX 10 Mbps スタンダードエディションのライセンスを VPX 200 Mbps プレミアムエディションにアップグレードするには、アップグレードを 2 つのステップで実行する必要があります。

- VPX は 10Mbps Standard Edition から 200Mbps Standard Edition にアップグレードする。
- VPX は 200Mbps Standard Edition から 200Mbps プレミアムエディションにアップグレードする。

> 注
>
> NetScaler Application Delivery Management（ADM）を使用して、共通の帯域幅とインスタンスプールで構成されるライセンスフレームワークを作成できます。詳細については、「NetScaler プール容量」を参照してください。

関連情報

- Citrix ライセンスシステム
- NetScaler VPX ライセンスを割り当てる方法

データガバナンス

March 20, 2024

NetScaler コンソールサービスコネクトとは何ですか**?**

NetScaler アプリケーション配信管理（ADM）サービス接続は、NetScaler MPX、SDX、VPX インスタンス、および NetScaler Gateway アプライアンスを NetScaler Console サービスにシームレスにオンボーディングできるようにする機能です。この機能により、NetScaler インスタンスまたは NetScaler Gateway アプライアンスは NetScaler Console サービスに自動的かつ安全に接続し、システム、使用状況、およびテレメトリデータを NetScaler Console サービスに送信できます。このデータに基づいて、NetScaler コンソールサービス上の NetScaler インフラストラクチャに関する洞察と推奨事項が得られます。

NetScaler Console サービス接続機能を使用して、NetScaler インスタンスまたは NetScaler Gateway アプライアンスを NetScaler コンソールサービスにオンボーディングします。オンプレミスでもクラウドでも、すべての NetScaler および NetScaler Gateway アセットを管理することもできます。また、パフォーマンスの問題、高いリソース使用率、重大なエラーなどをすばやく特定するのに役立つ豊富な可視性機能へのアクセスもメリットがあります。NetScaler コンソールサービスは、NetScaler インスタンスとアプリケーションに幅広い機能を提供します。NetScaler コンソールサービスの詳細については、「NetScalerアプリケーション配信管理サービス」を参照してください

> 重要
>
> - NetScaler Gateway アプライアンスは、NetScaler コンソールサービス接続機能もサポートしています。わかりやすくするために、NetScaler Gateway アプライアンスは連続するセクションでは明示的に呼び出されません。

NetScaler コンソールサービスとは何ですか**?**

NetScaler Console サービスは、NetScaler インスタンスの管理、監視、調整、自動化、トラブルシューティングに役立つクラウドベースのソリューションです。また、NetScaler インスタンス、およびアプリケーションの状態、パフォーマンス、セキュリティに関する分析的洞察と精選された機械学習ベースの推奨事項も提供します。詳しくは、「NetScaler コンソールサービスの概要」を参照してください

NetScaler コンソールサービス接続はどのように有効になっていますか**?**

NetScaler またはゲートウェイをインストールするか、リリース 13.0 ビルド 61.xx 以降にアップグレードすると、NetScaler Console サービス接続はデフォルトで有効になります。

NetScaler Console サービス接続を使用してキャプチャされるデータにはどのようなものがありますか**?**

NetScaler Console サービス接続を使用して次の詳細がキャプチャされます:

- **NetScaler** の詳細
  - シリアル ID
  - エンコードされたシリアル ID
  - ホスト ID
  - UUID
  - 管理 IP アドレス
  - ホスト名
  - バージョン
  - ビルドタイプ

  - 構築
  - ライセンスの種類
  - ハイパーバイザー
  - 導入タイプ (スタンドアロン/HA)
  - プラットフォームタイプ
  - プラットフォームの説明
  - システム ID
  - ADC で有効になっているモード
  - ADC で有効になっている機能

- ライセンス情報

  - NetScaler でライセンスされている機能
  - ライセンス番号

- 主な使用状況メトリック

  - システム日時
  - CPU の使用率
  - 管理 CPU パーセンテージ
  - スループット
  - SSL 新規セッション
  - SSL 暗号化スループット
  - SSL 復号化スループット
  - システム稼働時間

- 構成

  - ns.conf ファイル

> 注
>
> NetScaler Console サービス接続は、NetScaler アプライアンスから NetScaler Console サービスに`ns.conf`ファイルを送信する前に、暗号化されたパスワードまたはハッシュされたパスワードを匿名化します。NetScaler Console サービス接続は、`-encrypted`または`-passcrypt`パラメーターを確認し、関連する暗号化値またはハッシュ値を`XXXX`に置き換えます。次に、NetScaler Console サービス接続は`ns.conf`ファイルをエンコードして圧縮し、NetScaler Console サービスエンドポイントに送信します。

- 重大なエラーの詳細

  - ハードディスク障害
  - SSL カードの障害
  - 電源ユニット (PSU) の障害

  - フラッシュドライブの障害
  - ウォームリブート
  - 持続的なメモリ使用量が 90% を超えるか、メモリリークが発生している
  - レート制限の持続的な低下

- **NITRO** オートメーションツールの使い方
  - Ansible、Terraform、NITRO SDK などの自動化ツールの使用

- 診断の詳細

> 注:
>
> ADM 診断ツールでは、次の診断詳細が使用されます。詳しくは、NetScaler コンソールの診断ツールのトピックを参照してください 。

  - ADC CLI ステータス
  - ADC DNS ステータス
  - ADM エンドポイント「adm.cloud.com」へのネットワーク接続ステータス
  - ADM エンドポイント「agent.adm.cloud.com」へのネットワーク接続ステータス
  - ADM トラストサービス「trust.citrixnetworkapi.net」へのネットワーク接続ステータス
  - ADM ダウンロードサイト「download.citrixnetworkapi.net」へのネットワーク接続ステータス

データはどのように使用されますか

データを収集することで、NetScaler はお客様の NetScaler インストールに関する次のような詳細な情報をタイムリーに提供できます。

- 主要な指標。CPU、メモリ、スループット、SSL スループットに関する主要なメトリックスの詳細、および NetScaler インスタンスでの異常な動作を強調表示します。
- 重大なエラー。NetScaler インスタンスで発生した可能性のある重大なエラー。
- 導入アドバイザリ。スタンドアロンモードでデプロイされているが、スループットが高く、単一障害点に対して脆弱な NetScaler インスタンスを特定します。
- 診断ツール。ADC インスタンスを NetScaler Console にオンボードすると、ADC インスタンスが正常にオンボーディングされない原因となるいくつかの問題が発生することがあります。問題のトラブルシューティングを行うには、診断ツールを手動で使用するか、ADM GUI で診断情報を確認します。詳細については、「診断ツール」を参照してください。

収集されたデータはどのくらいの期間保持されますか

収集されたデータはすべて 13 か月以内に保持されます。

NetScaler から NetScaler Console サービス接続機能を無効にしてサービスの使用を終了する場合、以前に収集されたデータは 30 日後に削除されます。

データはどこに保存され、どの程度安全ですか

NetScaler Console サービスコネクトによって収集されたすべてのデータは、米国、欧州連合、オーストラリアおよびニュージーランド（ANZ）の 3 つの地域のいずれかに保存されます。詳しくは、「地理的な考慮事項」を参照してください。

データは、データベース層で厳密なテナント分離を使用して安全に保存されます。

NetScaler コンソールのサービス接続を無効にするにはどうすればよいですか?

NetScaler Console サービス接続によるデータ収集を無効にする場合は、「NetScaler Console サービス接続を有効または無効にする方法」を参照してください。

NetScaler アプライアンス用の NetScaler コンソールサービスコネクトの概要

April 15, 2024

NetScaler Console サービスは、NetScaler インスタンスの管理、監視、調整、自動化、トラブルシューティングに役立つクラウドベースのソリューションです。また、アプリケーションの正常性、パフォーマンス、およびセキュリティに関する分析情報と厳選された機械学習ベースの推奨事項も提供します。詳細については、「NetScaler Application Delivery Management サービス」を参照してください。

NetScaler アプリケーション配信管理（ADM）サービス接続は、NetScaler インスタンスの NetScaler コンソールサービスへのシームレスなオンボーディングを可能にする機能です。この機能により、NetScaler インスタンスと NetScaler Console サービスは総合的なソリューションとして機能し、お客様にさまざまなメリットをもたらします。

NetScaler Console サービス接続機能により、NetScaler インスタンスは自動的に NetScaler Console サービスに接続し、システム、使用状況、およびテレメトリデータを NetScaler Console サービスに送信できます。このデータに基づいて、NetScaler Console サービスは、次のような NetScaler とゲートウェイのインフラストラクチャに関するいくつかの洞察と推奨事項を提供します:

- 脆弱な ADC アプライアンスに焦点を当てた、セキュリティアドバイザリインサイト。
- アップグレードのアドバイザリーインサイトは、メンテナンス終了と寿命の終了に達している、または到達しようとしている ADC アプライアンスを強調します。
- パフォーマンスの問題、高いリソース使用量、および重大なエラーの迅速な識別。

NetScaler コンソールサービスの機能を活用するには、NetScaler インスタンスを NetScaler コンソールサービスにオンボーディングすることを選択できます。オンボーディングプロセスは ADM サービス接続を使用し、スムーズかつ高速なエクスペリエンスを提供します。

> 注意事項
>
> - NetScaler コンソールサービス接続が、NetScaler MPX、SDX、VPX の各インスタンスと NetScaler Gateway アプライアンスで利用できるようになりました。
> - この NetScaler Console サービス接続機能を使用する NetScaler Console サービスのイニシアチブは、ADM サービス接続ベースのロータッチオンボーディングです。詳しくは、「NetScaler Console サービス接続を使用した NetScaler インスタンスのロータッチオンボーディング」を参照してください。
> - ADC インスタンスで ADM サービス接続が有効になっている場合、特定の診断の詳細が ADM サービスに自動的に送信されます。

詳しくは、「データガバナンス」を参照してください。

> 重要
>
> NetScaler Console サービス接続はプローブデータの収集に失敗し、次の条件が満たされている場合、ADC アプライアンスを ADM サービスにオンボーディングするのに役立ちません。
>
> - `NSinternal`ユーザーアカウントが無効になっている。
> - SSH 公開キーが設定されていません。
>
> 上記のシナリオを克服するには、次のいずれかを実行することをお勧めします。
>
> - `set ns param -internaluserlogin ENABLED`を使用して、`internaluser`ユーザアカウントを有効にします。
> - 公開キー認証を構成する。詳しくは、「パスワードなしで SSH キーを使用して NetScaler アプライアンスにアクセスする
>   」を参照してください。

NetScaler コンソールサービスはどのようにして **NetScaler** コンソールサービスとサポートを接続しますか**?**

ここでは、NetScaler の NetScaler コンソールサービス接続機能が NetScaler コンソールサービスとどのように連携するかについての大まかなワークフローを示します。

1. NetScaler アプライアンスの NetScaler コンソールサービス接続機能は、定期的なプローブ要求を使用して NetScaler Console サービスに自動的に接続します。
2. このリクエストにはシステム、使用状況、およびテレメトリデータが含まれており、NetScaler Console サービスはこれらを使用して NetScaler インフラストラクチャに関するいくつかの洞察と推奨事項を提供します。例：パフォーマンスの問題、高いリソース使用量、および重大なエラーをすばやく特定します。
3. 分析情報と推奨事項を確認して、ADC インスタンスを NetScaler Console サービスにオンボーディングして NetScaler インスタンスの管理を開始することを決定できます。

4. オンボーディングを決定したら、NetScaler Console のサービス接続機能を使用してオンボーディングをシームレスに完了できます。

NetScaler コンソールのサービスコネクトはどのバージョンの NetScaler でサポートされていますか

NetScaler Console サービス接続は、すべての NetScaler プラットフォームとすべてのアプライアンスモデル（MPX、VPX、SDX）でサポートされています。NetScaler リリース 13.0 ビルド 61.xx 以降、NetScaler アプライアンスの NetScaler コンソールサービス接続はデフォルトで有効になっています。

NetScaler コンソールのサービス接続を有効にする方法を教えてください

NetScaler の既存のお客様で、NetScaler リリース 13.0 ビルド 61.xx にアップグレードすると、アップグレードプロセスの一部として NetScaler Console サービス接続がデフォルトで有効になります。

NetScaler の新規お客様で、NetScaler リリース 13.0 ビルド 61.xx をインストールすると、インストールプロセスの一部として NetScaler Console サービス接続がデフォルトで有効になります。

> 注
>
> 新しい NetScaler アプライアンスとは異なり、既存の NetScaler アプライアンスは Citrix Insight Service（CIS）または Call Home を介してルートを検索します。

NetScaler Console サービス接続を有効または無効にする方法を教えてください

NetScaler コンソールのサービス接続は、CLI、GUI、または NITRO API メソッドから有効または無効にできます。

CLI の使用

NetScaler コンソールサービスを有効にするには、CLI を使用して接続します

コマンドプロンプトで入力します：

```
set adm parameter  -admserviceconnect ENABLED
```

CLI を使用して NetScaler コンソールサービス接続を無効にするには

コマンドプロンプトで入力します：

```
set adm parameter  -admserviceconnect DISABLED
```

> 重要
>
> NetScaler がリリース 13.0 ビルド 61.xx の場合、NetScaler サービス接続を有効または無効にするパラメ

> ータ名は「自動接続」です。たとえば、サービス接続を有効にするには、`set adm parameter -autoconnect ENABLED`コマンドを使用します。

GUI の使用

NetScaler GUI を使用して NetScaler コンソールサービス接続を無効にするには

1. Web ブラウザーに、NetScaler アプライアンスの IP アドレス（http://192.0.2.10など）を入力します。
2. [**User Name**] と [**Password**] に管理者の資格情報を入力します。
3. [**System**]、[**Settings**]、[**Configure ADM Parameters**] の順に選択します。
4. **ADM** パラメータの設定ページで、**NetScaler Console** サービス接続を有効にするダイアログボックスをクリアし、「**OK**」をクリックします。

NITRO API の使用

NetScaler コンソールのサービス接続を無効にするには、**NITRO** コマンドを使用します。

- NetScaler リリース 13.0 ビルド 61.xx では、次のコマンドを使用して NetScaler コンソールサービス接続を有効または無効にできます:

  ```
  - curl -X PUT -H "Content-Type:application/json"http://192.0.2.10/
    nitro/v1/config/admparameter> -d '{ "admparameter":{ "autoconnect
    ":"enabled" } } ' -u nsroot:Test@1
  ```

- NetScaler リリース 13.0 ビルド 64.xx から、「自動接続」パラメータ名が`admserviceconnect`に変更されます。次のコマンドを使用して NetScaler Console サービス接続を無効にできます:

  ```
  - curl -X PUT -H “Content-Type:application/json” http://192.0.2.10/
    nitro/v1/config/admparameter -d '{ "admparameter":{ "admserviceconnect
    ":"disabled" } } ' -u nsroot:Test@1
  ```

診断ツール

ADC インスタンスを NetScaler Console にオンボードすると、ADC インスタンスが正常にオンボーディングされない原因となるいくつかの問題が発生することがあります。問題のトラブルシューティングを行うには、診断ツールを手動で使用するか、ADM GUI で診断情報を確認します。

- ADM Service Connect を使用してキャプチャされた詳細について詳しくは、「データガバナンス」を参照してください。
- 診断ツールについて詳しくは、「診断ツール」を参照してください。

NetScaler コンソールの組み込みエージェントの動作

NetScaler リリース 13.0 ビルド 61.xx 以降では、NetScaler インスタンスで使用できる NetScaler Console の組み込みエージェントが ADM サービスと通信するようになりました。それぞれの ADC インスタンスで、手動での初期化を必要とせずに通信できます。ADM サービスとの通信が確立された後、組み込みエージェントは、定期的に最新のソフトウェアバージョンに自動アップグレードすることにより、常緑を維持します。

以前は、ADM サービスとの通信を確立したり、定期的な自動アップグレードを行うために、`mastools`コマンドを使用して ADC インスタンスで組み込みエージェントを初期化する必要がありました。
詳細については、「インスタンスを管理するための ADC 組み込みエージェントの設定」を参照してください。

参照ドキュメント

NetScaler Console サービス接続の詳細については、以下のトピックを参照してください:

- データガバナンス: データガバナンス。
- NetScaler コンソールサービス: NetScaler アプリケーションデリバリー管理サービス。

NetScaler アプライアンスのアップグレードとダウングレード

March 20, 2024

NetScaler 13.1 では、機能が強化された新機能や更新された機能が提供されています。機能拡張についてはすべて、リリース発表に付属のリリースノートに記載されています。ソフトウェアをアップグレードする前に、リリースノートをご一読ください。

このセクションでは、**Citrix**ADC GUI または CLI を使用して NetScaler アプライアンス（MPX および VPX）ファームウェアをアップグレードおよびダウングレードする方法について説明します **。

また、**NetScaler** コンソールを使用して **NetScaler** アプライアンスをアップグレードすることもできます。詳しくは、次のトピックを参照してください:

- NetScaler コンソールサービスが NetScaler のアップグレードを容易にする 10 の方法
- NetScaler コンソールサービスを使用して NetScaler インスタンスをアップグレードする
- NetScaler コンソールソフトウェアを使用して NetScaler インスタンスをアップグレードする

NetScaler SDX アプライアンスのアップグレードについて詳しくは、「シングルバンドルアップグレード」を参照 してください。

> 注
>
> NetScaler バージョン 13.1 以降では、廃止予定の従来のポリシーベースの機能が NetScaler アプライアンスから削除されました。詳しくは、「クラシックポリシーの廃止に関するよくある質問」の表を参照してください。

はじめに

April 15, 2024

アップグレードまたはダウングレードプロセスを開始する前に、次の点を確認してください。

- 組織のサポート契約を評価します。アプライアンスのシリアル番号、サポート契約、および Citrix のテクニカルサポートまたは Citrix 認定パートナーからのサポートを受けるための連絡先の詳細を記録します。
- NetScaler アプライアンスのアップグレードに割り当てられた時間。組織の変更管理手順に従います。アップグレードの実行に 2 倍の時間を割り当てます。各 NetScaler アプライアンスをアップグレードするのに十分な時間を割り当てます。
- NetScaler ライセンスシステムは、NetScaler VPX アプライアンスの Customer Success Services（CSS）メンバーシップライセンスの検証を強制します。NetScaler VPX アプライアンスをアップグレードする前に、アプライアンスの現在の CSS メンバーシップが有効で、有効期限が切れていないことを確認してください。

  現在の CSS メンバーシップの有効期限が、アップグレードする NetScaler 製品バージョンの CSS 資格日と同じかそれ以降であることを確認してください。

  CSS メンバーシップの有効期限が CSS 資格日より前の場合、既存のライセンスはアップグレードされたバージョンの NetScaler VPX アプライアンスでは機能しません。この機能により、ライセンスの不正使用を防ぐことができます。NetScaler VPX アプライアンスをアップグレードする前に、CSS メンバーシップを更新する必要があります。

  NetScaler VPX リリースの一覧と CSS の利用資格日については、「NetScaler 製品の CustomerSuccess Services 資格日」を参照してください。

  CSS の詳細については、「Customer Success Services」を参照してください。
- Citrix では、メジャーリリースを 1 つずつアップグレードすることを推奨しています。たとえば、NetScaler アプライアンスがリリース 12.1 にあり、リリース 13.1 にアップグレードする場合は、まずアプライアンスをリリース 13.0 にアップグレードし、次にリリース 13.1 にアップグレードします。
- ライセンスフレームワークとライセンスの種類。ソフトウェアエディションのアップグレードには、次のような新しいライセンスが必要になる場合があります。
  - 標準版から上級版へのアップグレード、または
  - スタンダードエディションからプレミアムエディション、または

  - プレミアムエディションへのアドバンスエディション。

  バージョン 13.1 にアップグレードしても、既存の NetScaler ライセンスは引き続き機能します。詳細については、「ライセンス」を参照してください。

- 新規および非推奨のコマンド、パラメータ、および SNMP OIDを確認します。
- NetScaler MPX ハードウェアおよびソフトウェアの互換性マトリクスを確認します。
- NetScaler Gateway のログオンページがカスタマイズされている場合は、UI テーマがデフォルトに設定されていることを確認します。
- LOM をアップグレードする場合は、LOM ファームウェアのアップグレードページを確認します。
- Citrix ADC ダウンロードから NetScaler ファームウェアをダウンロードします。NetScaler ファームウェアをダウンロードする詳細な手順については、NetScaler リリースパッケージをダウンロードするを参照してください。
- バックアップファイル。構成ファイル、カスタマイズファイル、証明書、モニタースクリプト、ライセンスファイルなどのバックアップを手動で実行するか、NetScaler CLI または GUI- バックアップと復元を使用したバックアップについては、次のドキュメントを参照してください。

  - バックアップ用のその他の一般的なカスタマイズファイルについては、次のリストを参照してください。
    * /nsconfig/monitors/*.pl
    * /nsconfig/rc.netscaler
  - カスタマイズフォルダをバックアップして削除します。/var/customizationsこれは通常下です。カスタマイズの例として、ロゴ付きのログオンページがあります。カスタマイズフォルダーをコピーした後、アプライアンスをアップグレードする前に、NetScaler アプライアンスからフォルダーを削除する必要があります。カスタマイズを適してアップグレードすると、いくつかの問題が発生する可能性があります。

  > 重要:
  >
  > Citrix では、上記のバックアップ手順を確認することを強くお勧めします。NetScaler アプライアンスで更新が完了しない場合のアクションプランを用意します。

- アップグレードを実行する前に、NetScaler アプライアンス用の/var および/flash ディレクトリに十分な領域があることを確認します。/var には 5 GB の空き領域が必要です (アップグレードバンドルの場合は 1 GB + アップグレードプロセスに 4 GB)
  /flash には、新しいカーネル上でコピーするのに十分な領域が必要です。これは 140 MB から 160MB の間で異なります。少なくとも 250 MB の空き領域があることを確認します。
  /var のディスク領域をクリアする方法の詳細については、「NetScaler アプライアンスの問題をログに記録するために/var ディレクトリの空き領域を解放する方法」を参照してください。
  /flash のディスク領域のクリアの詳細については、https://support.citrix.com/article/CTX133587を参照してください。

- 事前構成チェックツールを使用してアップグレードを実行する前に、無効な構成がないことを確認します。また、このツールは、アップグレードプロセス中および`installns`スクリプトの一部としてデフォルトで実行されます。無効な構成によるアップグレードの失敗を避けるため、アップグレード前に事前構成チェックツールを実行することをお勧めします。詳細については、「事前構成チェックツール」を参照してください。

  無効な構成がある場合は、次の手順を実行して、`nspepi` ツールを使用して無効な構成を有効な構成に変換します。

  1. `nspepi`ツールを実行して`ns.conf`ファイルを変換します。`nspepi`ツールについて詳しくは、「NSPEPI ツールを使用したポリシー表現の変換」を参照してください。
  2. `nspepi`ツールによって生成された`warn_ns.conf`ファイルを確認し、エラーがあれば修正します。
  3. 以下のオプションのいずれかを使用して変換された構成を適用して、変換された構成をテストします。
     a) `clear config -f basic` コマンドを実行し、`source /nsconfig/new_ns.conf` コマンドを実行して変換された構成を取得します。
     b) クラシックコマンドを手動で（ポリシーのバインドを解除して削除して）削除し、`source /nsconfig/new_ns.conf` コマンドを実行して変換された構成を取得します。
  4. 構成を保存します。

  > 注:
  >
  > ビルドをアップグレードするための`installns`スクリプトでオプション`Y`を選択した場合、事前構成チェックは実行されません。

  クラシックポリシーの廃止について詳しくは、「クラシックポリシーの廃止に関する FAQ」を参照してください。

- NetScaler アプライアンスの整合性を検証します。NetScaler ハードウェアアプライアンスを使用している場合は、`fsck`を実行して、ディスクチェックを実行し、NetScaler ハードドライブの整合性を検証することを強くお勧めします。エラーが発生した場合は、ハードディスクドライブをリセットし、ディスクチェックコマンドを繰り返します。エラーメッセージが再び表示される場合は、NetScaler サポートに連絡して問題を詳しく調査してください。

  - fsck コマンドを使用して、ハードドライブのディスクの整合性を検証します。詳細については、CTX122845を参照してください。
  - 診断バンドルファイルを使用して NetScaler アプライアンスの整合性を検証し、分析のためにログを Citrix Insight Service にアップロードします。詳細については、「テクニカルサポートバンドルの収集方法」を参照してください。

- NetScaler VPX サポートマトリックスと使用ガイドラインを確認してください。
- FAQ セクションを確認してください。
- テスト環境でアップグレード手順を確認します。

NetScaler アプライアンスをアップグレードまたはダウングレードするための前提条件の詳細については、以下のサポート記事を参照してください。

- CTX220371：NetScaler アップグレード前とアップグレード後に記事を読む必要がある

クラシックポリシーを使用する構成のアップグレードに関する考慮事項

January 9, 2024

クラシックポリシーは、NetScaler リリース 12.0 ビルド 56.20 から廃止されました。リリース 13.1 以降、クラシックポリシーサポートは一部の機能から削除されました。リリース 13.1 リリースでサポートされていない機能またはコマンドの完全なリストについては、「NetScaler Classic ポリシーベースの特徴と機能のステータス変更のお知らせ」の表 **2** を参照してください。

リリース 13.1 以降にアップグレードするための前提条件として、従来のポリシーを NSPEPI ツールでサポートされている機能の高度なポリシーに変換することをお勧めします。NSPEPI ツールでサポートされている機能については、「nspepi 変換ツールで処理されるコマンドまたは機能」を参照してください。アップグレード前に高度なポリシーに変換しないと、構成が無効なため、アップグレードは失敗します。

> 重要:
>
> リリース 13.1 以降にアップグレードする前に、事前構成チェックツールを実行することをお勧めします。このツールにより、廃止または削除された機能に関連するコマンドが構成に含まれていないことが保証されます。廃止または削除された機能に関連するコマンドがある場合、ツールはエラーを返し、構成が失われる可能性があります。事前設定チェックツールの最新バージョンは、公開されている GitHub リンク-からダウンロードすることをお勧めします。https://github.com/citrix/ADC-scripts/tree/master/nspepi
>
> 詳細については、アップグレード前の事前構成チェックツールを参照してください。

事前構成チェックの一環としてエラーが発生した場合は、`nspepi` ツールを使用して無効な構成を有効な構成に変換することをお勧めします。公開されている GitHub リンク- https://github.com/citrix/ADC-scripts/tree/master/nspepiから`nspepi`ツールの最新バージョンをダウンロードしてください。`nspepi`ツールを使用したポリシー表現の変換の詳細については、「NSPEPI ツールを使用したポリシー表現の変換」を参照してください。

`nspepi`ツールは、一部のコマンドまたは機能の変換を処理しません。完全なリストについては、「nspepi 変換ツールで処理されないコマンドまたは機能 」を参照してください。

変換する必要がある各構成について、次の手順を実行します。

1. Citrix ADC リリース 12.1 または 13.0 で`nspepi`ツールを実行して`ns.conf`ファイルを変換します。NSPEPI ツールは、プレフィックスが`new_`と`warn_`の 2 つのファイルを生成します。`new_`プレフィックスの付いたファイルには変換された構成が含まれ、`warn_`プレフィックスの付いたファイルには警告とエラーが含まれます。

2. warn_プレフィックスの付いたファイルを確認し、エラーがあれば修正します。
3. 変換した構成をテストします。以下のオプションのいずれかを使用して、変換した設定を適用します。

    a) clear config -f basic コマンドを実行し、source /nsconfig/new_ns.conf コマンドを実行して変換された構成を取得します。

    b) クラシックコマンドを手動で（ポリシーのバインドを解除して削除して）削除し、source /nsconfig/new_ns.conf コマンドを実行して変換された構成を取得します。

    > 注:
    >
    > clear config -f basicコマンドを使用してクリアされない構成では、「リソースはすでに存在しています」というエラーが表示される場合が あります。これらのエラーは無視できます。

4. 構成を保存します。

無効な構成を有効な構成に正常に変換したら、他の前提条件があるかどうかを確認し、アップグレードに進みます。詳しくは、「NetScaler アプライアンスのアップグレードとダウングレード」を参照してください。

/etc ディレクトリ内のカスタマイズされた設定ファイルのアップグレードに関する考慮事項

August 15, 2023

/etc ディレクトリでは、以下の設定ファイルの変更がサポートされています。

- inetd.conf
- syslog.conf
- newsyslog.conf
- ntp.conf
- crontab
- host.conf
- hosts
- ttys
- sshd_config
- httpd.conf
- monitrc
- rc.conf
- ssh_config
- localtime
- issue

- issue.net
- ldap.conf
- motd

> 注:
>
> アプライアンスで実行されている NetScaler ADC ビルドによっては、上記のリストに新しいファイルが追加される場合があります。NetScaler ADC コマンドラインインターフェイスで次のシェルコマンドを実行すると、更新されたファイルリストを表示できます。
>
> ```
> grep NSETC= /etc/rc
> ```

/etcディレクトリ内の構成ファイルを変更して/nsconfigディレクトリにコピーした場合、永続性を維持するために、Citrix ADC アプライアンスは/nsconfig中のファイルを指すシンボリックリンクを/etcに作成します。

たとえば、次のようになります: /etc/httpd.conf -> /nsconfig /httpd.conf

リリースパッケージには、/etc ディレクトリに独自のバージョンの設定ファイルが含まれている場合があります。これらの構成ファイルには、NetScaler ADC アプライアンスが正常に機能するために必要な重要な更新が含まれています。NetScaler ADC アプライアンスをリリースにアップグレードすると、/etc ディレクトリ内の構成ファイルがリリースアップデートを含む構成ファイルに置き換えられます。

/etcディレクトリにある、カスタマイズされた設定ファイルexample.confの例を考えてみましょう。example.confファイルは、永続性を維持するために/nsconfigディレクトリにコピーされます。NetScaler ADC アプライアンスは、/nsconfigで次のファイルを指すシンボリックリンクを/etcに作成します。/etc/example.conf -> / nsconfig /example.conf

また、リリースパッケージには、重要な更新を含む独自のバージョンのexample.confが含まれています。NetScaler ADC アプライアンスをリリースにアップグレードすると、次の動作が観察されます。

シンボリックリンク/etc/example.confはすでに存在するため、NetScaler ADC アプライアンスは、アップグレードプロセス中にexample.confのリリースパッケージコピーをディレクトリ/etcに配置しません。

example.confのリリースパッケージコピーには重要な更新が含まれているため、/etcディレクトリに更新がないと、NetScaler ADC アプライアンスが失敗したり、正しく機能しなくなる可能性があります。

アップグレードの変更とカスタマイズを保持する手順

リリースの更新とカスタマイズの両方が失われないようにするには、次の手順を実行します。

- アップグレード前の手順:
  - アップグレード前にカスタマイズしたファイルをバックアップする
  - アップグレード前にカスタマイズしたファイルの永続性を削除する
- アップグレード後の手順:

- アップグレードしたファイルにカスタマイズを適用し、アップグレード後に永続性を追加

  > 重要:
  >
  > /etcフォルダ内のカスタマイズしたファイルを直接置き換えないでください。/etcファイルをバックアップ用のカスタマイズされたファイルに直接置き換えると、アップグレードプロセス中にファイルに追加されたリリースアップデートが削除されます。

アップグレード前にカスタマイズしたファイルをバックアップする

アプライアンスをアップグレードする前に、/nsconfig ディレクトリにあるカスタマイズファイルのバックアップを作成します。

/var/nsconfig_backup ディレクトリを作成し、カスタマイズしたファイルをこのディレクトリに移動します。つまり、シェルプロンプトで次のコマンドを実行して、/etcディレクトリで変更して/nsconfigにコピーしたファイルを移動します。

```
mv /nsconfig/<filename> /var/nsconfig_backup/
<!--NeedCopy-->
```

例:

```
mv /nsconfig/httpd.conf /var/nsconfig_backup/
<!--NeedCopy-->
```

アップグレード前にカスタマイズしたファイルの永続性を削除する

アプライアンスをアップグレードする前に、/nsconfigファイルを指している/etcシンボリックリンクを削除します。

1. シェルプロンプトで次のコマンドを実行して、/etc ディレクトリ内の既存のシンボリックリンクを確認します。

   ```
   ls -la /etc
   <!--NeedCopy-->
   ```

2. シェルプロンプトで次のコマンドを実行して、/nsconfigファイルを指す/etcシンボリックリンクを削除します。

   ```
   unlink /etc/<filename>
   <!--NeedCopy-->
   ```

   例:

   ```
   unlink /etc/httpd.conf
   <!--NeedCopy-->
   ```

3. シェルプロンプトで次のコマンドを実行して、シンボリックリンクが削除されたことを確認します。


```
cat /etc/<filename>
<!--NeedCopy-->
```

例:

```
cat /etc/httpd.conf
<!--NeedCopy-->
```

シンボリックリンクが削除されると、このコマンドはコンテンツを表示しません。

アップグレードしたファイルにカスタマイズを適用し、アップグレード後に永続性を追加

変更した/nsconfig設定ファイルのバックアップを/var/nsconfig_backupに取った場合は、アプライアンスのアップグレード後に次の操作を行います。

1. /var/nsconfig_backupと/etcディレクトリにあるファイルを比較します。リリースアップデートをすでに含んでいる/etcファイルに、適切な変更を手動で追加します。

   > 重要:
   >
   > /etcファイルを/var/nsconfig_backupファイルに直接置き換えると、アップグレードプロセス中にファイルに追加されたリリースアップデートが削除されます。この更新プログラムの削除により、関連する NetScaler ADC 機能が失敗するか、正しく動作しなくなる可能性があります。

2. 永続性を維持するには、シェルプロンプトで次のコマンドを実行して、/etcディレクトリにある更新されたファイルを/nsconfigディレクトリにコピーします。

   ```
   cp /etc/<filename> /nsconfig/
   <!--NeedCopy-->
   ```

   例:

   ```
   cp /etc/httpd.conf /nsconfig/
   <!--NeedCopy-->
   ```

3. /var/nsconfig_backupディレクトリにあるカスタマイズしたファイルごとに、上記の 2 つの手順を繰り返します。
4. アプライアンスを再起動して、変更を有効にします。

アップグレードに関する考慮事項 - SNMP 構成

August 15, 2023

SNMP アラームのタイムアウトパラメータは内部オプションであり、アラーム設定には影響しません。

これらの SNMP アラーム設定に何も変更を加えていなくても、実行構成（sh running）と保存済み構成（ns.conf）の SNMP アラーム設定にタイムアウトパラメータが表示されることがあります。

タイムアウト設定の問題が修正されたリリースビルドにアップグレードすると、SNMP 構成が誤ってデフォルト値にリセットされます。

アップグレード中は、次の SNMP アラーム（設定されている場合）が影響を受けます。

- APPFW-BUFFER-OVERFLOW
- APPFW-COOKIE
- APPFW-CSRF-TAG
- APPFW-DENY-URL
- APPFW-FIELD-CONSISTENCY
- APPFW-FIELD-FORMAT
- APPFW-POLICY-HIT
- APPFW-REFERER-HEADER
- APPFW-SAFE-COMMERCE
- APPFW-SAFE-OBJECT
- APPFW-SQL
- APPFW-START-URL
- APPFW-VIOLATIONS-TYPE
- APPFW-XML-ATTACHMENT
- APPFW-XML-DOS
- APPFW-XML-SCHEMA-COMPILE
- APPFW-XML-SOAP-FAULT
- APPFW-XML-SQL
- APPFW-XML-VALIDATION
- APPFW-XML-WSI
- APPFW-XML-XSS
- APPFW-XSS
- CLUSTER-BACKPLANE-HB-MISSING
- CLUSTER-NODE-HEALTH
- CLUSTER-NODE-QUORUM
- CLUSTER-VERSION-MISMATCH
- COMPACT-FLASH-ERRORS
- CONFIG-CHANGE
- CONFIG-SAVE
- HA-BAD-SECONDARY-STATE
- HA-NO-HEARTBEATS
- HA-SYNC-FAILURE

- HA-VERSION-MISMATCH
- HARD-DISK-DRIVE-ERRORS
- HA-STATE-CHANGE
- HA-STICKY-PRIMARY
- PORT-ALLOC-FAILED
- SYNFLOOD

NetScaler を次のリリースビルドにアップグレードすると、これらの SNMP アラーム構成が影響を受けます。

- リリース 11.1 ビルド 61.2 以降
- リリース 12.0 ビルド 61.0 以降
- リリース 12.1 ビルド 30.1 以降
- リリース 13.0 ビルド 51.4 以降

例

クラスタノードヘルス SNMP アラームの例を考えてみましょう。

```
CLUSTER-NODE-HEALTH SNMP alarm is set up by using the NetScaler command
     line:

> set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -
    severity Major

> save config
<!--NeedCopy-->
```

この SNMP アラーム設定は、保存された設定ファイル（ns.conf）に次のように表示されます。

```
set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -severity
    Major -timeout 86400

<!--NeedCopy-->
```

上記のリリースビルドのいずれかにアップグレードすると、ns.log ファイルに次のエラーが表示されます。

```
May 23 09:14:46 <local0.err> ns nsconfigd: __init_config_filter(): (
    null) line 0: No such argument [-timeout]>> set snmp alarm CLUSTER-
    NODE-HEALTH -time 111 -state DISABLED -severity Major -timeout
    86400.
<!--NeedCopy-->
```

アップグレード後、SNMP アラーム設定はデフォルト値にリセットされます。

回避方法

この問題を回避するには、以下のいずれかを行います。

- アップグレードの前に、保存された構成ファイル (ns.conf) の SNMP 構成からタイムアウト設定を削除します。
- アップグレード後、タイムアウトパラメータなしで SNMP アラームを再設定します。

NetScaler ADC リリースパッケージをダウンロードする

August 15, 2023

NetScaler ADC リリースパッケージをダウンロードするには、次の手順を実行します。

1. Web ブラウザで NetScaler ADC ダウンロードページを開きます 。
2. [NetScaler ADC ダウンロード] ページで、更新する **NetScaler ADC** リリース を展開します。
3. 適切なカテゴリのいずれかを展開し、NetScaler ビルドリックをクリックします。たとえば、NetScaler ADC ファームウェアのバージョンをダウンロードするには、[ファームウェア] を展開し、ダウンロードする NetScaler ADC ビルドをクリックします。
4. 選択した NetScaler ADC ビルドページで、[ビルド] セクションを展開し、[ファイルの ダウンロード] をクリックして NetScaler ADC ビルドパッケージをダウンロードします。

> 注:
> チェックサムは、ダウンロードしたビルドパッケージが Web サイトでホストされている実際のパッケージと一致することを確認するためのものです。チェックサムは、ビットが正しいことを確認するための重要なチェックです。

NetScaler ADC スタンドアロンアプライアンスのアップグレード

October 25, 2023

システムソフトウェアをアップグレードする前に、「開始する前に 」セクションを読み、必要なファイルのバックアップや NetScaler ADC ファームウェアのダウンロードなどの前提条件を完了してください。

GUI を使用して NetScaler ADC スタンドアロンアプライアンスをアップグレードする

GUI を使用してスタンドアロンの NetScaler ADC をリリース 13.1 にアップグレードするには、次の手順に従います。

1. Web ブラウザーで、NetScaler IP アドレスを入力します（例: `http://10.102.29.50`）。
2. [ユーザー名] と [パスワード] に管理者の資格情報 (nsroot/nsroot) を入力し、[ログオン] をクリックします。

3. [システム] に移動し、[システムアップグレード] をクリックします。
4. [ファイルの選択] メニューから、[ローカル] または [アプライアンス] の適切なオプションを選択します。アプライアンスオプションを使用する場合は、最初にファームウェアを NetScaler ADC にアップロードする必要があります。WinSCP などの任意のファイル転送方法を使用して、NetScaler ADC ファームウェアをアプライアンスにアップロードできます。
5. 正しいファイルを選択し、[アップグレード（**Upgrade**）] をクリックします。
6. 指示に従ってソフトウェアをアップグレードします。
7. プロンプトが表示されたら、[再起動] を選択します。

アップグレード後、アプライアンスにアクセスする前に、すべてのブラウザインスタンスを閉じ、コンピュータのキャッシュをクリアします。

CLI を使用して **NetScaler ADC** スタンドアロンアプライアンスをアップグレードする

CLI を使用してスタンドアロンの NetScaler ADC をリリース 13.1 にアップグレードするには、次の手順を実行します。

次の手順では、<release>および<releasenumber>はアップグレードするリリースバージョンを表し、<targetbuildnumber>はアップグレードするビルド番号を表します。この手順には、アップグレード中に /etc ディレクトリにプッシュされた更新が失われないようにするオプションの手順が含まれています。

1. PuTTY などの SSH クライアントを使用して、アプライアンスへの SSH 接続を開きます。
2. 管理者の資格情報を使用して、アプライアンスにログオンします。実行構成を保存します。プロンプトで、次のように入力します。

   ```
   save config
   ```

3. 次のコマンドを実行して、シェルプロンプトに切り替えます。

   ```
   shell
   ```

4. ns.conf ファイルのコピーを作成します。シェルプロンプトで、次のように入力します。

   - `cd /nsconfig`
   - `cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber>`

   設定ファイルは別のコンピュータにバックアップする必要があります。
5. 重要:

   > アップグレードの変更とカスタマイズの両方を、アップグレードされた NetScaler ADC アプライアンスに適用することが重要です。そのため、/etc ディレクトリにカスタマイズした構成ファイルがある場合は、「カスタマイズされた構成ファイルのアップグレードに関する考慮事項」のアップグレード前の

> 手順を実行します。

6. インストールパッケージの場所を作成します。シェルプロンプトで、次のように入力します。

   - `cd /var/nsinstall`
   - `cd <releasenumber>`

     > 注:
     >
     > 目的のリリース番号ディレクトリが存在しない場合は、次のコマンドを使用してディレクトリを作成します。
     >
     > `mkdir <releasenumber>`
     >
     > 例:
     >
     > `mkdir 13.1`

   - `mkdir build_<targetbuildnumber>`
   - `cd build_<targetbuildnumber>`

7. WinSCP などのファイル転送方法を使用して、ダウンロード済みの NetScaler ADC ファームウェアを上記の手順で作成したビルドディレクトリにコピーします。NetScaler ADC ファームウェアのダウンロードの詳細については、「開始する前に 」セクションを参照してください。
8. インストールパッケージの内容を展開します。例:

   ```
   tar -xvzf build-13.1-37.2_nc_64.tgz
   ```

9. installns スクリプトを実行して、新しいバージョンのシステムソフトウェアをインストールします。

   ```
   ./installns
   ```

10. プロンプトが表示されたら、NetScaler ADC を再起動します。
11. 重要:

    > アップグレードの変更とカスタマイズの両方を、アップグレードされた NetScaler ADC アプライアンスに適用することが重要です。そのため、`/etc` ディレクトリにカスタマイズした構成ファイルがある場合は、「カスタマイズされた構成ファイルのアップグレードに関する考慮事項」のアップグレード後の手順を実行します。

以下は、NetScaler ADC ファームウェアのアップグレードの例です。

```
login: nsroot

Password: nsroot

Last login: Mon Apr 17 15:05:05 2018 from 10.252.243.134

Done
```



```
> save config

> shell

Last login: Mon Apr 17 15:05:05 2018 from 10.252.243.134

root@NSnnn# cd /var/nsinstall

root@NSnnn# cd 13.1

root@NSnnn# mkdir build_43.1

root@NSnnn# cd build_43.1

root@NSnnn# ftp <FTP server IP address>

ftp> mget build-13.1-41.1_nc.tgz

ftp> bye

root@NSnnn# tar xzvf build-13.1-41.1_nc.tgz

root@NSnnn# ./installns

installns version (13.1-41.1) kernel (ns-13.1-41.1_nc.gz)

...

Copying ns-13.1-41.1_nc.gz to /flash/ns-13.1-41.1_nc.gz ...

...

Installation has completed.

Reboot NOW? [Y/N] Y
```

CLI を使用して NetScaler スタンドアロンアプライアンスをアップグレードしてアップグレードする方法については、このビデオをご覧ください 。

NITRO API を使用して **NetScaler ADC** スタンドアロンアプライアンスをアップグレードする

NITRO API を使用して NetScaler ADC をアップグレードまたはダウングレードするには、NetScaler アップグレードとダウングレードを単一の API で自動化するを参照してください。

アップグレード後、**NetScaler ADC** アプライアンスのエンティティステータスを確認します

NetScaler ADC アプライアンスをアップグレードした後、次のエンティティのステータスを確認します。

- 仮想サーバは UP 状態です
- モニタはアップ状態です
- GSLB サイトは問題なく同期します。
- すべての証明書がアプライアンスに存在します。
- すべてのライセンスがアプライアンスに存在します。

NetScaler 13.1 ソフトウェアアップデートの確認とインストール

パフォーマンスを向上させるために、アップデートが利用可能になったときに NetScaler ADC ソフトウェアをアップデートします。NetScaler アップデートには、機能の改善、パフォーマンスの修正、または機能強化が含まれる場合があります。リリースノートを読んで、アップデートで利用可能な修正と機能強化を確認してください。ソフトウェアアップデートを確認してインストールするには、次の手順を実行します。

1. NetScaler ADC ホームページで、**** 右上隅の **nsroot** メニューから［アップデートの確認 **］をクリックします。
2. [利用可能な最新のシステムソフトウェアの更新] ページで、インストール可能なソフトウェア更新プログラムを確認します。
3.「ダウンロード」をクリックして、NetScaler ダウンロード Web サイトからインストールパッケージをダウンロードします 。
4. ソフトウェアパッケージをダウンロードしたら、CLI または GUI の手順で更新プログラムをインストールします。

> 注
>
> [アップデートの確認（**Check for Update** ）] リンクにアクセスできるのは、HTTPS プロトコルではなく HTTP プロトコルを使用して GUI にログインしている場合だけです。

関連情報

次のリソースは、NetScaler ADC アプライアンスのアップグレードまたはダウングレードに関する関連情報を提供します。

- ビデオチュートリアル- CLI を使用して NetScaler ADC をアップグレードする方法

NetScaler ADC スタンドアロンアプライアンスのダウングレード

October 25, 2023

CLI または GUI を使用して、スタンドアロンの NetScaler ADC で以前のリリースにダウングレードできます。

> 注:
>
> ダウングレード時に設定が失われる可能性があります。ダウングレード前とダウングレード後の設定を比較し、不足しているエントリを手動で再入力します。

CLI を使用して NetScaler ADC アプライアンスをダウングレードする

リリース 13.1 を実行している NetScaler ADC スタンドアロンアプライアンスを以前のリリースにダウングレードするには、以下の手順に従います。

この手順では、<release>と<releasenumber>はダウングレードするリリースバージョンを表し、<targetbuildnumber>はダウングレードするビルド番号を表します。

1. PuTTY などの SSH クライアントを使用して、NetScaler ADC への SSH 接続を開きます。
2. 管理者の資格情報を使用して NetScaler ADC にログオンします。実行構成を保存します。プロンプトで、次のように入力します。

   設定を保存
3. ns.conf ファイルのコピーを作成します。シェルプロンプトで、次のように入力します。

   a) `cd /nsconfig`
   b) `cp ns.conf ns.conf.NS<currentbuildnumber>`

   設定ファイルのコピーを別のコンピュータにバックアップする必要があります。
4. <releasenumber> 設定ファイル (ns.conf.ns<releasenumber>) を ns.conf にコピーします。シェルプロンプトで、次のように入力します。

   ```
   1 cp ns.conf.NS<releasenumber> ns.conf
   2 <!--NeedCopy-->
   ```

   > 注:
   >
   > ns.conf.NS<releasenumber>は、システムソフトウェアをリリースバージョン<releasenumber>から現在のリリースバージョンにアップグレードしたときに自動的に作成されるバックアップ構成ファイルです。
   >
   > ダウングレード時に設定に多少の損失が生じる可能性があります。アプライアンスの再起動後、ステップ 3 で保存した設定を実行構成と比較し、ダウングレード前に設定された機能とエンティティを調整します。変更を行った後、実行構成を保存します。
   >
   > 重要:
   >
   > ルーティングが有効になっている場合は、ステップ 5 を実行します。それ以外の場合は、ステップ 6 に進みます。

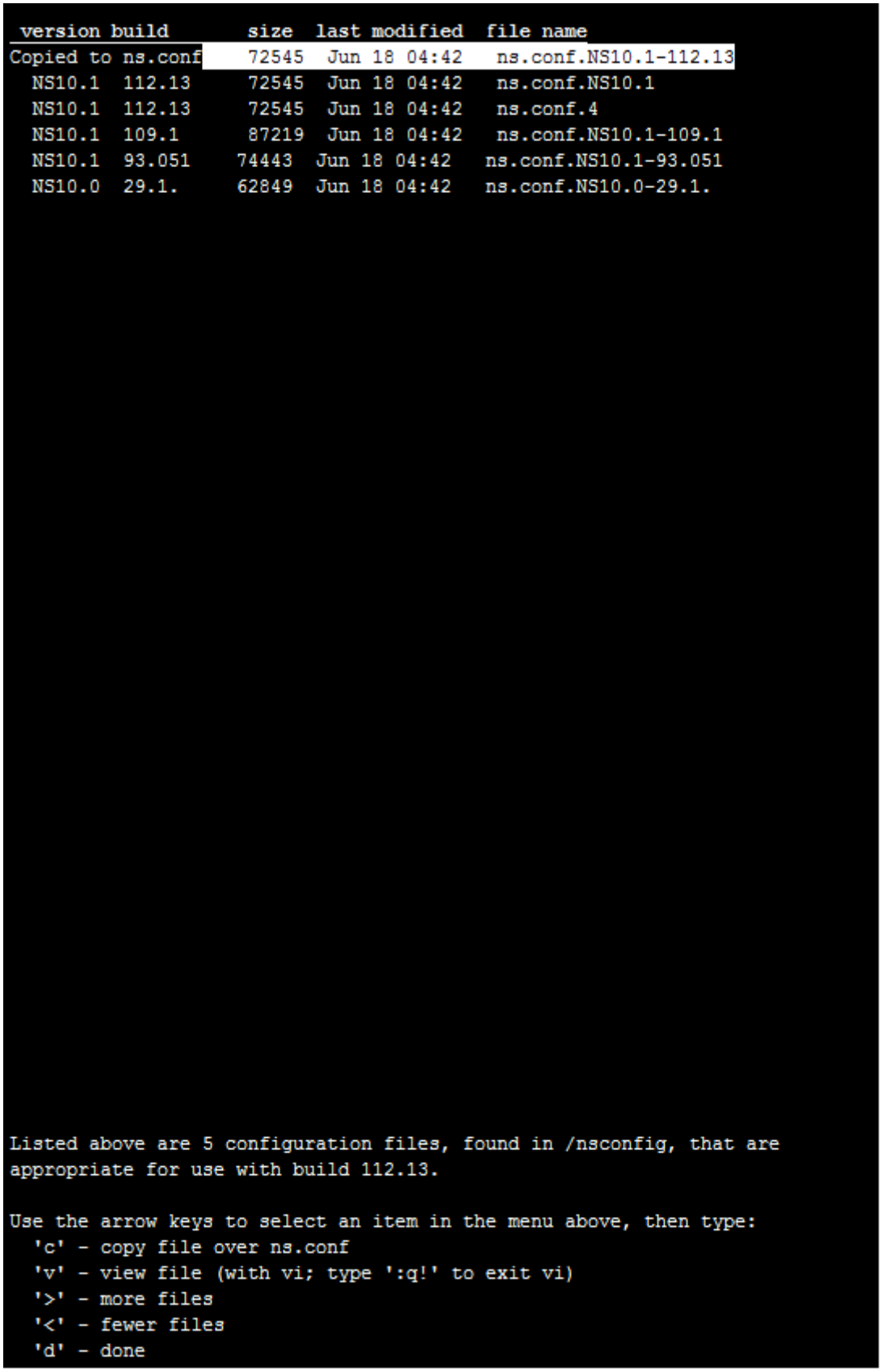
5. ルーティングが有効な場合、zebos.conf ファイルには設定が含まれます。シェルプロンプトで、次のように入力します。

```
cd /nsconfig
cp ZebOS.conf ZebOS.conf.NS
cp ZebOS.conf.NS<targetreleasenumber> ZebOS.conf
<!--NeedCopy-->
```

6. ディレクトリを/var/nsinstall/<releasenumber>nsinstallに変更するか、存在しない場合は作成します。
7. ディレクトリをbuild_<targetbuildnumber>に変更するか、存在しない場合は作成します。
8. インストールパッケージ (build-<release>-<targetbuildnumber>.tgz) をこのディレクトリにダウンロードまたはコピーし、インストールパッケージの内容を展開します。
9. installnsスクリプトを実行して、新しいバージョンのシステムソフトウェアをインストールします。スクリプトは/etcディレクトリを更新します。

   ダウングレードするビルドの設定ファイルがアプライアンス上に存在する場合は、その設定をロードするように求められます。

   図 1：設定ファイルが存在する場合はメニューをダウングレードする



フラッシュドライブの空き領域が新しいビルドをインストールするのに十分でない場合、NetScaler ADC はインストールを中止します。フラッシュドライブを手動でクリーンアップし、インストールを再開します。

例:

```
login: nsroot

Password: nsroot

Last login: Mon Apr 24 02:06:52 2017 from 10.102.29.9

Done

> save config

> shell

root@NSnnn# cp ns.conf.NS10.5 ns.conf

root@NSnnn# cd /var/nsinstall

root@NSnnn# mkdir 10.5nsinstall

root@NSnnn# cd 10.5nsinstall

root@NSnnn# mkdir build_57

root@NSnnn# cd build_57

root@NSnnn# ftp 10.102.1.1

ftp> mget build-10.5-57_nc.tgz

ftp> bye

root@NSnnn# tar -xzvf build-10.1-125_nc.tgz

root@NSnnn# ./installns

installns version (10.5-57) kernel (ns-10.5-57.gz)

...

...

...

Copying ns-10.5-57.gz to /flash/ns-10.5-57_nc.gz ...

Changing /flash/boot/loader.conf for ns-10.5-57 ...


```



```
Installation has completed.


Reboot NOW? [Y/N] Y
<!--NeedCopy-->
```

GUI を使用して NetScaler ADC アプライアンスをダウングレードする

GUI のアップグレードウィザードを使用して、リリース 13.1 を実行している NetScaler ADC アプライアンスを以前のリリースにダウングレードできます。

> 注:
>
> GUI を使用して、リリース 13.1 を実行している NetScaler ADC アプライアンスをリリース 10.5 以前に直接ダウングレードすることはできません。ダウングレードには CLI の使用をお勧めします。
>
> NetScaler ADC のリリースライフサイクルの詳細については、製品マトリックスサイトを参照してください 。
>
> 一度に 1 つのメジャーリリースにダウングレードすることをお勧めします。
>
> たとえば、NetScaler ADC アプライアンスがリリース 13.1 にあり、リリース 12.1 にダウングレードする場合は、最初にアプライアンスをリリース 13.0 にダウングレードし、次にリリース 12.1 にダウングレードする必要があります。

GUI を使用してリリース 13.1 を実行している NetScaler ADC アプライアンスを以前のリリースにダウングレードするには、以下の手順に従います。

1. Web ブラウザーで、NetScaler IP アドレスを入力します（例: `http://10.102.29.50`）。
2. [ユーザー名] と [パスワード] に管理者の資格情報を入力し、[ログオン] をクリックします。
3. [システム] に移動し、[システムアップグレード] をクリックします。
4. [ファイルの選択] メニューから、[ローカル] または [アプライアンス] の適切なオプションを選択します。アプライアンスオプションを使用する場合は、最初にファームウェアを NetScaler ADC にアップロードする必要があります。WinSCP などの任意のファイル転送方法を使用して、NetScaler ADC ファームウェアをアプライアンスにアップロードできます。
5. 正しいファイルを選択し、[アップグレード（**Upgrade**）] をクリックします。
6. 指示に従ってソフトウェアをダウングレードします。
7. プロンプトが表示されたら、[再起動] を選択します。

ダウングレード後、アプライアンスにアクセスする前に、すべてのブラウザインスタンスを閉じ、コンピュータのキャッシュをクリアします。

関連情報

次のリソースは、NetScaler ADC アプライアンスのアップグレードまたはダウングレードに関する関連情報を提供します。

- ビデオチュートリアル- CLI を使用して NetScaler ADC をアップグレードする方法

高可用性ペアをアップグレードする

December 8, 2023

高可用性セットアップにおける NetScaler の要件の 1 つは、プライマリノードとセカンダリノードの両方に同じ NetScaler ソフトウェアバージョン（リリース）をインストールすることです。したがって、ソフトウェアが両方のノードでアップグレードされていることを確認してください。

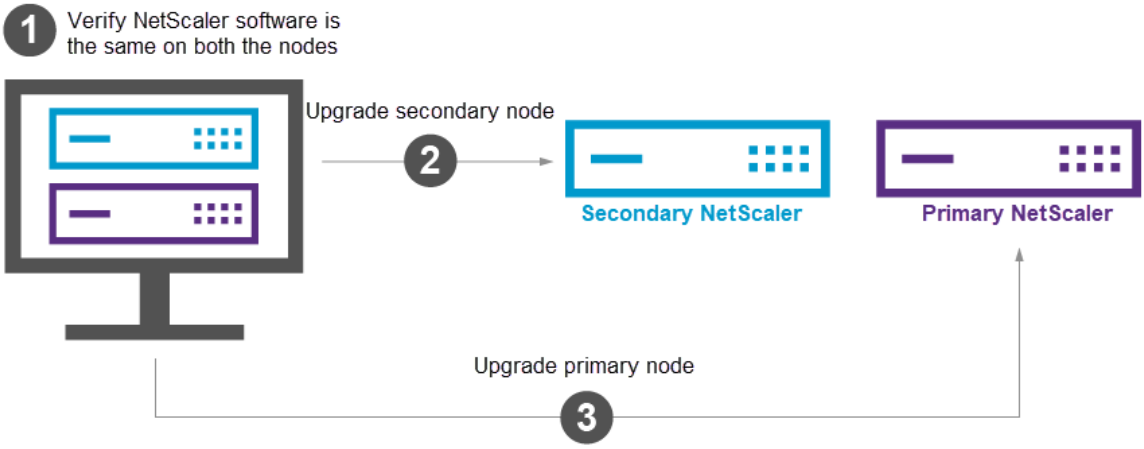
スタンドアロンインスタンスのアップグレードと同じ手順に従って、高可用性ペアの両方のノードをアップグレードできますが、高可用性ペアのアップグレードには他の考慮事項が適用されます。

高可用性ペアで NetScaler ファームウェアのアップグレードを開始する前に、「開始する前に 」セクションに記載されている前提条件をお読みください。また、いくつかの HA 固有の点を考慮する必要があります。

注意事項

> 重要:
>
> アップグレードした NetScaler に、アップグレードした変更とカスタマイズ内容の両方を適用することが重要です。したがって、`/etc` ディレクトリにカスタマイズした構成ファイルがある場合は、アップグレードを続行する前に、「カスタマイズされた構成ファイルに関するアップグレードの考慮事項 」を参照してください。

- 最初にセカンダリノードをアップグレードし、次にプライマリノードをアップグレードします。NetScaler は CLI または GUI を使用してアップグレードできます。



- 高可用性（HA）セットアップの両方のノードで異なるバージョンの NetScaler ソフトウェアが実行されている場合、次の機能は無効になります：
  - HA 設定の同期
  - HA コマンドの伝播
  - ステートサービス情報の HA 同期
  - セッションの接続ミラーリング (接続フェイルオーバー)
  - 永続性セッション情報の HA 同期

  > 注：
  >
  > 両方のノードが同じ NetScaler ソフトウェアバージョン（14.1 など）の異なるビルド（8.x と 4.x など）を実行している場合、機能の動作は次のように変わります。
  >
  > - 両方のビルドの内部 HA バージョンが異なる場合、これらの機能は無効になります。
  > - これらの機能は、両方のビルドの内部 HA バージョンが同じであれば正常に機能します。

  NetScaler ビルドの内部 HA バージョンが変更されているかどうかを確認するには、「NetScaler ビルドの新しい内部 HA バージョン」セクションを参照してください。

- HA 構成の 2 つのノードが異なる NetScaler ソフトウェアバージョンを実行している場合、または 2 つのノードが同じバージョンの異なるビルドを実行している場合、`sync HA files`コマンドの`All`モードでのファイルの同期は正常に機能します。詳細については、「高可用性セットアップでの設定ファイルの同期」を参照してください。

NetScaler ビルドの新しい内部 **HA** バージョン

次の表は、内部 HA バージョンが変更された NetScaler ビルドの一覧です：

| リリース 14.1| リリース 13.1| リリース 13.0| リリース 12.1|
|–|–|–|–|

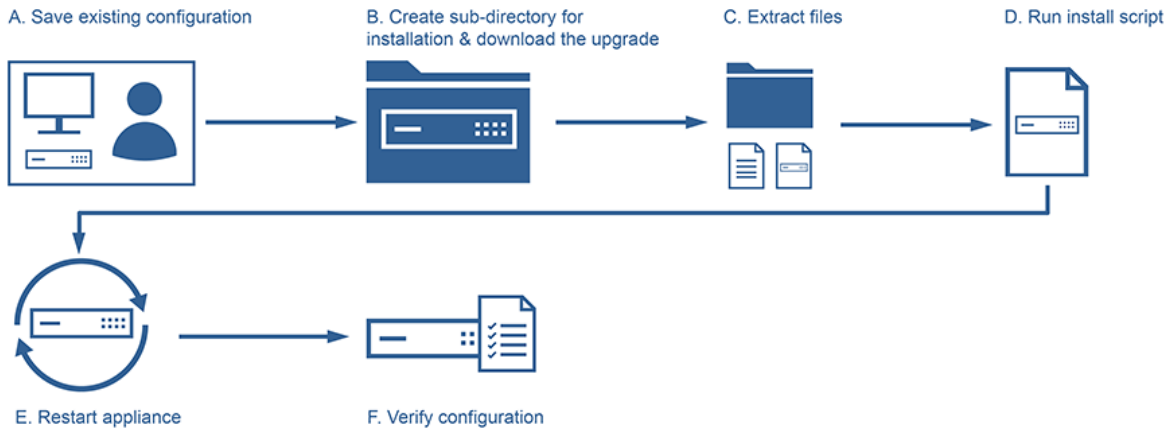
ビルド 12.30	ビルド 50.23	ビルド 92.21	ビルド 65.35
ビルド 8.50	ビルド 49.15	ビルド 90.12	ビルド 62.27
ビルド 4.42	ビルド 48.47	ビルド 87.9	ビルド 61.19
	ビルド 45.64	ビルド 86.17	ビルド 60.19
	ビルド 37.38	ビルド 85.19	ビルド 59.16
	ビルド 33.54	ビルド 84.11	ビルド 58.15
	ビルド 30.52	ビルド 82.45	ビルド 57.18
	ビルド 27.59	ビルド 79.64	ビルド 56.22
	ビルド 24.38	ビルド 76.31	ビルド 55.24
	ビルド 21.50	ビルド 71.44	ビルド 50.31
	ビルド 17.42	ビルド 67.43	ビルド 49.37
	ビルド 12.51	ビルド 64.35
	ビルド 9.60	ビルド 61.48
	ビルド 4.44	ビルド 58.32
		ビルド 52.24
		ビルド 41.28

CLI を使用して高可用性ペアをアップグレードする

アップグレードプロセスには、次の手順が含まれます:

1. 2 次ノード上のソフトウェアのアップグレード
2. 1 次ノード上のソフトウェアのアップグレード

次の図は、ノード上のソフトウェアをアップグレードする手順を示しています:



2 次ノード上のソフトウェアのアップグレード

1. 「CLI を使用した NetScaler スタンドアロンアプライアンスのアップグレード」の説明に従ってセカンダリノードをアップグレードします。
2. nsroot資格情報を使用して NetScaler CLI にログインします。
3. 次のコマンドを実行して、NetScaler の状態を表示します。

```
show ha node
<!--NeedCopy-->
```

コマンド出力には、ノードがセカンダリノードであり、同期が無効になっていることが示されている必要があります。

4. 次のコマンドを実行して、プライマリノードとして強制フェイルオーバーとテイクオーバーを実行します:

```
force failover
<!--NeedCopy-->
```

5. ノードがプライマリノードになったことを確認します。

新しいプライマリノードの設定例を次に示します。

```
login: nsroot
Password: nsroot
Last login: Monday Apr  17 08:37:26 2017 from 10.102.29.9
Done
show ha node
        2 nodes:
1)      Node ID:      0
        IP:        10.0.4.2
        Node State: UP
        Master State: Primary
        ...
        Sync State: AUTO DISABLED
        Propagation: AUTO DISABLED
        ...
Done
<!--NeedCopy-->
```

1 次ノード上のソフトウェアのアップグレード

> 注:
>
> 「セカンダリノード上のソフトウェアのアップグレード」手順が完了すると、元のプライマリノードはセカンダリノードになります。

1. 「CLI を使用した NetScaler スタンドアロンアプライアンスのアップグレード」の説明に従ってセカンダリノードをアップグレードします。

2. nsroot資格情報を使用して NetScaler CLI にログインします。
3. 次のコマンドを実行して、NetScaler の状態を表示します。

```
show ha node
<!--NeedCopy-->
```

コマンド出力には、ノードがセカンダリノードであり、ノードのステータスが UP であることが示されている必要があります。

4. 次のコマンドを実行して強制フェイルオーバーを実行し、ノードがプライマリノードであることを確認します:

```
force failover
<!--NeedCopy-->
```

5. ノードがプライマリノードであることを確認します。

新しいプライマリノードと新しいセカンダリノードの設定例を次に示します。

```
show ha node
    Node ID:      0
    IP:   10.0.4.11
    Node State: UP
    Master State: Primary
    ...
    ...
    INC State: DISABLED
    Sync State: ENABLED
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    ...
    ...
    Local node information
    Critical Interfaces: 1/1
Done

Show ha node
    Node ID:      0
    IP:   10.0.4.2
    Node State: UP
    Master State: Secondary
    ..
    ..
    INC State: DISABLED
    Sync State: SUCCESS
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    . .
    . .
    Local node information:
```



```
36      Critical Interfaces: 1/1
37  Done
38  <!--NeedCopy-->
```

GUI を使用して高可用性ペアをアップグレードする

ADC GUI を使用して高可用性セットアップで NetScaler ペアをアップグレードするには、次の手順に従います。NetScaler インスタンスの NETSCALER-A（プライマリ）と NETSCALER-B（セカンダリ）の高可用性セットアップの例を考えてみましょう。

1. セカンダリノードをアップグレードします。管理者資格情報を使用してセカンダリノード GUI にログインし、「GUI を使用した NetScaler スタンドアロンアプライアンスのアップグレード」の説明に従ってノードをアップグレードします。
2. 強制フェイルオーバー。「ノードの強制フェイルオーバー」の説明に従って、GUI を使用してセカンダリノードで強制フェールオーバーを実行します。

   フェイルオーバー操作後、セカンダリノードがプライマリとして引き継ぎ、プライマリノードが新しいセカンダリノードになります。例の HA セットアップのフェールオーバー操作の後：

   - NETSCALER-B が新しいプライマリになります
   - NETSCALER-A が新しいセカンダリになります
3. 元のプライマリノード（新しいセカンダリノード）をアップグレードします。新しいセカンダリノード GUI（NETSCALER-A）にログインし、「GUI を使用した NetScaler スタンドアロンアプライアンスのアップグレード」の説明に従ってノードをアップグレードします。
4. 強制フェイルオーバー。「ノードの強制フェールオーバー」の説明に従って、GUI を使用して新しいセカンダリノード（NETSCALER-A）で強制フェイルオーバーを実行します。

   この 2 回目のフェイルオーバー操作の後、両方のノードの状態は、HA アップグレード操作を開始する前と同じ状態に戻ります。例の HA セットアップのフェールオーバー操作の後：

   - NETSCALER-A がプライマリになる
   - NETSCALER-B がセカンダリになります
5. アップグレードプロセスを確認します。両方のノードの GUI にログインします。[システム**] > [**高可用性] に移動し、詳細ページで両方のノードの HA 状態を確認します。また、GUI の上部ペインに表示されているアップグレードバージョンの詳細も確認してください。

GUI を使用して高可用性セットアップをアップグレードする方法については、このビデオをご覧ください 。

インサービスソフトウェアアップグレードのサポートにより、ダウンタイムゼロのアップグレードを実行するための高可用性を実現

February 15, 2024

高可用性セットアップ (HA) の通常のアップグレードプロセス中に、ある時点で、両方のノードで異なるソフトウェアビルドが実行されます。これら 2 つのビルドの内部 HA バージョン番号は同じでも異なっていてもかまいません。

両方のビルドの HA バージョン番号が異なる場合、既存のデータ接続の接続フェールオーバーは (有効になっていても) サポートされません。つまり、既存のデータ接続がすべて失われ、ダウンタイムにつながります。

この問題に対処するには、サービス内ソフトウェアアップグレード（ISSU）を使用して HA セットアップを行うことができます。ISSU では、アップグレードプロセスの強制フェールオーバー操作ステップに代わる移行機能が導入されました。移行機能では、既存の接続が考慮され、強制フェイルオーバー操作も含まれます。

移行操作の実行後、新しいプライマリノードは既存の接続に関連するトラフィック (要求と応答) を常に受信しますが、古いプライマリノードに誘導します。古いプライマリノードはデータトラフィックを処理し、送信先に直接送信します。

拡張 **ISSU** の仕組み

HA セットアップの通常のアップグレードプロセスは次のステップで構成されます:

1. セカンダリノードをアップグレードします。このステップには、セカンダリノードのソフトウェアアップグレードとノードの再起動が含まれます。
2. 強制フェイルオーバー強制フェイルオーバーを実行すると、アップグレードされたセカンダリノードがプライマリに、プライマリノードがセカンダリノードになります。
3. 新しいセカンダリノードをアップグレードします。このステップには、新しいセカンダリノードのソフトウェアアップグレードとノードの再起動が含まれます。

ステップ 1 からステップ 3 までの時間枠では、両方のノードで異なるソフトウェアビルドが実行されます。これら 2 つのビルドの内部 HA バージョンは同じでも異なっていてもかまいません。

両方のビルドの HA バージョン番号が異なる場合、既存のデータ接続の接続フェールオーバーは (有効になっていても) サポートされません。つまり、既存のデータ接続がすべて失われ、ダウンタイムにつながります。

HA セットアップの ISSU アップグレードプロセスは次のステップで構成されています:

1. セカンダリノードをアップグレードします。このステップには、セカンダリノードのソフトウェアアップグレードとノードの再起動が含まれます。
2. **ISSU** 移行オペレーション。このステップには、強制フェイルオーバー操作が含まれ、既存の接続が処理されます。移行操作を実行すると、新しいプライマリノードは常に既存の接続に関連するトラフィック（要求と応

答）を受信しますが、GRE トンネルに設定された SYNC VLAN（設定されている場合）を介して古いプライマリノードに転送します。古いプライマリノードはデータトラフィックを処理し、送信先に直接送信します。ISSU 移行操作は、既存の接続がすべて閉じられると完了します。

3. 新しいセカンダリノードをアップグレードします。このステップには、新しいセカンダリノードのソフトウェアアップグレードとノードの再起動が含まれます。

はじめに

HA セットアップで ISSU プロセスを実行する前に、次の前提条件、制限事項、および注意点を確認してください：

- ピア NSIP アドレスの MAC アドレスが解決されるインターフェースの容量が、クライアントまたはサーバーインターフェースの容量と同じかそれ以上であることを確認します。たとえば、次のシナリオを考えてみましょう：
  - ピア NSIP アドレスの MAC アドレスはインターフェイス 1/x で解決され、データインターフェイスは 10/x です。このシナリオでは、MAC アドレスが解決されるインターフェイスの容量がデータインターフェイスの容量よりも少ないため、ISSU を実行しないでください。
  - ピア NSIP アドレスの MAC アドレスはインターフェイス 10/x で解決され、データインターフェイスは 10/x です。このシナリオでは、MAC アドレスが解決されるインターフェイスの容量がデータインターフェイスと同じであるため、ISSU を実行できます。
- HA セットアップの両方のノードでSYNC VLANが設定されていることを確認します。詳細については、「VLAN への高可用性同期トラフィックの制限」を参照してください。

> 注：
>
> 同期 VLAN 設定は L2 HA でのみサポートされます。HA-INC モードではサポートされていません。

- Microsoft Azure は GRE トンネリングをサポートしていないため、ISSU は Microsoft Azure クラウドではサポートされていません。
- ISSU 中は HA 設定の伝達と同期が機能しません。
- ISSU は IPv6 HA セットアップではサポートされていません。
- ISSU は管理パーティションではサポートされていません。
- ISSU は、次のセッションではサポートされていません：
  - ジャンボフレーム
  - IPv6 セッション
  - 大規模 NAT (LSN)
- INC モードの HA セットアップでは、ISSU 移行操作はクライアント側の接続のみを移行します。両方の HA ノードには独立した SNIP 構成があるため、サーバー側接続の移行は不要です。
- 同期 VLAN の設定では、同期 VLAN の MTU を 42 バイト以上増やすことを推奨します。

構成の手順

ISSU には、HA セットアップの通常のアップグレードプロセスにおける強制フェールオーバー操作に代わる移行機能があります。移行機能では、既存の接続が考慮され、強制フェイルオーバー操作も含まれます。

HA セットアップの ISSU プロセスでは、セカンダリノードをアップグレードした直後に移行操作を実行します。2 つのノードのどちらからでもマイグレーション操作を実行できます。

CLI 手順

CLI を使用して HA 移行操作を実行するには:

コマンドプロンプトで次のように入力します:

```
start ns migration
<!--NeedCopy-->
```

GUI プロシージャ

GUI を使用して HA 移行操作を実行するには:

[システム] **> [**システム情報**] > [**移行**]** タブに移動します。[移行を開始] をクリックします。

ISSU 統計情報を表示する

HA セットアップの現在の ISSU プロセスを監視するための ISSU 統計情報を表示できます。ISSU 統計情報には、次の情報が表示されます:

- ISSU 移行オペレーションの現在のステータス
- ISSU 移行操作の開始時刻
- ISSU 移行操作の終了時刻
- ISSU ロールバック操作の開始時刻
- ISSU 移行操作の一部として処理される接続の総数
- ISSU 移行操作の一部として処理されている残りの接続数

CLI または GUI を使用して、いずれかの HA ノードの ISSU 統計情報を表示できます。

CLI 手順

CLI を使用して ISSU 統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで次のように入力します:


```
show ns migration
<!--NeedCopy-->
```

GUI プロシージャ

GUI を使用して ISSU 統計情報を表示するには、次の手順を実行します:

[システム] **>** **[**システム情報**]** **>** **[**移行**]** タブに移動します。[**Click]** をクリックすると、移行の詳細が表示されます。

ISSU 統計情報を表示する**-**古いプライマリノードが処理している既存の接続のリスト

`show migration`操作の`dumpsession`（`Dump Session`）オプションを使用すると、ISSU 移行操作の一部として古いプライマリノードが提供している既存の接続のリストを表示できます。

`dumpsession`オプションを指定した show migration 操作は、ISSU 操作中に新しいプライマリノードでのみ実行する必要があります。

CLI 手順

CLI を使用して古いプライマリノードが処理している既存の接続のリストを表示するには:

コマンドプロンプトで次のように入力します:

```
show ns migration  -dumpsession YES
<!--NeedCopy-->
```

```
> sh migration -dumpsession yes

Index   remote-IP-port          local-IP-port           idle-time(x 10
    ms)

1       192.0.2.10      22      192.0.2.1       15998       703
2       198.51.100.20   7375    98.51.100.2     22          687
3       203.0.113.30    5506    203.0.113.3     22          687


<!--NeedCopy-->
```

GUI プロシージャ

GUI を使用して古いプライマリノードが処理している既存の接続のリストを表示するには:

[システム] **>** **[**システム情報**]** **>** **[**移行**]** タブに移動します。「クリック 」をクリックしてマイグレーション接続を表示します。

ISSU プロセスのロールバック

HA セットアップで、インサービスソフトウェアアップグレード（ISSU）プロセスのロールバックがサポートされるようになりました。ISSU ロールバック機能は、ISSU 移行操作中の HA セットアップが安定しない、または想定どおりに最適なレベルで実行されていないことが確認された場合に役立ちます。

ISSU ロールバックは、ISSU 移行操作が進行中の場合に適用されます。ISSU の移行操作がすでに完了している場合、ISSU ロールバックは機能しません。つまり、ISSU の移行操作が進行中の場合は、ISSU ロールバック操作を実行する必要があります。

ISSU ロールバックは、ISSU ロールバック操作がトリガーされたときの ISSU 移行操作の状態に応じて異なる動作をします：

- **ISSU** 移行操作中に強制フェールオーバーがまだ発生していない。ISSU ロールバックは ISSU 移行操作を停止し、両方のノードに保存されている ISSU 移行に関連する内部データをすべて削除します。現在のプライマリノードはプライマリノードとして残り、既存の接続と新しい接続に関連するデータトラフィックを引き続き処理します。
- **ISSU** 移行操作中に強制フェールオーバーが発生しました。ISSU 移行処理中に HA フェールオーバーが発生した場合、新しいプライマリノード（N1 など）は新しい接続に関連するトラフィックを処理します。古いプライマリノード（新しいセカンダリノード、たとえば N2）は、古い接続（ISSU 移行操作前の既存の接続）に関連するトラフィックを処理します。

  ISSU ロールバックは ISSU 移行操作を停止し、強制フェールオーバーをトリガーします。新しいプライマリノード (N2) は、新しい接続に関連するトラフィックの処理を開始します。新しいプライマリノード（N2）は、古い接続（ISSU 移行操作の前に確立された既存の接続）に関連するトラフィックも引き続き処理します。つまり、ISSU 移行操作の前に確立された既存の接続は失われません。

  新しいセカンダリノード（N1）は、既存の接続（ISSU 移行操作中に作成された新しい接続）をすべて削除し、トラフィックを処理しません。つまり、ISSU 移行操作の強制フェールオーバー後に確立された既存の接続は、永久に失われます。

構成の手順

NetScaler CLI または GUI を使用して、ISSU ロールバック操作を実行できます。

CLI 手順

CLI を使用して ISSU ロールバック操作を実行するには、次の手順を実行します。

コマンドプロンプトで次のように入力します：

```
stop ns migration
<!--NeedCopy-->
```


GUI プロシージャ

GUI を使用して ISSU ロールバック操作を実行するには、次の手順を実行します:

[システム] **> [**システム情報**] > [**移行**]** タブに移動します。[移行を停止] をクリックします。

インサービスソフトウェアアップグレードプロセスの **SNMP** トラップ

HA セットアップのインサービスソフトウェアアップグレード（ISSU）プロセスでは、ISSU 移行操作の開始時と終了時に次の SNMP トラップメッセージがサポートされます。

SNMP トラップ	説明
マイグレーション開始	この SNMP トラップは、ISSU 移行操作の開始時に生成され、設定された SNMP トラップリスナーに送信されます。
マイグレーション完了	この SNMP トラップは、ISSU 移行操作が完了すると生成され、設定された SNMP トラップリスナーに送信されます。

プライマリノード（ISSU プロセスの開始前）は常にこれら 2 つの SNMP トラップを生成し、設定された SNMP トラップリスナーに送信します。

ISSU SNMP トラップに関連付けられた SNMP アラームはありません。つまり、SNMP アラームに関係なくこれらのトラップが生成されます。必要なのは、トラップ SNMP リスナーの設定だけです。

SNMP トラップロスナの構成の詳細については、NetScaler での SNMP トラップを参照してください。

高可用性ペアをダウングレードする

August 15, 2023

コマンドラインインターフェイスを使用して、高可用性ペアの任意のリリースにダウングレードできます。GUI はダウングレードプロセスをサポートしていません。

高可用性ペアの NetScaler ADC ペアのシステムソフトウェアをダウングレードするには、まずセカンダリノードでソフトウェアをダウングレードし、次にプライマリノードでダウングレードする必要があります。各ノードを個別にダウングレードする方法については、NetScaler ADC スタンドアロンアプライアンスのダウングレードを参照してください。

> 重要
>
> ダウングレード時に設定が失われる可能性があります。ダウングレード前とダウングレード後の構成を比較し、欠落しているエントリがあれば手動で再入力する必要があります。

インストール、アップグレード、およびダウングレードプロセスに関連する問題のトラブルシューティング

April 15, 2024

インストール、アップグレード、またはダウングレードプロセスの完了後にアプライアンスが期待どおりに動作しない場合、最初に行うべきことは、問題の最も一般的な原因を確認することです。

トラブルシューティングのリソース

最良の結果を得るには、次のリソースを使用して、NetScaler インストール、アップグレード、またはダウングレードに関連する問題のトラブルシューティングを行います。

- アプライアンスの設定ファイル。高可用性ペアの場合は、両方のアプライアンスの設定ファイル。
- アプライアンスからの次のファイル:
  - 関連する newnslog ファイル。
  - ns.log ファイル。
  - メッセージファイル。
- ネットワークトポロジ図。

問題と解決策

インストール、アップグレード、ダウングレードに関する最も一般的な問題と、その解決のヒントを以下に示します。

1. 問題

   NetScaler MPX アプライアンスのアップグレードは、ハードウェアとソフトウェアの非互換性のために失敗します。

   解像度

   NetScaler MPX ハードウェアとソフトウェアの互換性マトリクスを参照し 、ソフトウェアリリースが NetScaler ADC MPX ハードウェアでサポートされているかどうかを確認します。

2. 問題

   NetScaler VPX アプライアンスとハイパーバイザーの互換性がないため、NetScaler VPX アプライアンスのアップグレードが失敗します。

   解像度

   NetScaler VPX アプライアンスおよびハイパーバイザーの互換性マトリクスを参照し 、NetScaler VPX アプライアンスモデルがハイパーバイザーでサポートされているかどうかを確認します。

3. 問題

   NetScaler ADC アプライアンスのアップグレードは、ハードウェアエラーのために失敗します。

   解像度

   NetScaler アプライアンスの整合性を検証します。NetScaler ADC ハードウェアアプライアンスをお持ちの場合は、`fsck`を実行してディスクチェックを実行し、NetScaler ADC ハードディスクの整合性を検証することをお勧めします。

   詳細については、「NetScaler ADC アプライアンスのファイルシステムの整合性を検証する方法」を参照してください。

4. 問題

   GUI ストールを使用して NetScaler ADC アプライアンスをアップグレードする。

   解像度

   ブラウザを更新して、アップグレードが進行しているかどうかを確認します。

5. 問題

   NetScaler ADC アプライアンスのアップグレードは、/var ディレクトリの空き容量が少ないために失敗する

   解像度

   /var ディレクトリ上のスペースを解放します。詳細については、「/var ディレクトリの空き領域を増やす方法」を参照してください。

6. 問題

   ソフトウェアのダウングレード後、NetScaler ADC にアクセスできない

   原因

   ソフトウェアのダウングレードプロセス中に、既存のリリースおよびビルドの設定ファイルが以前のリリースおよびビルドの設定ファイルと一致しない場合、アプライアンスは設定をロードできず、デフォルトの IP アドレスがアプライアンスに割り当てられます。

   解像度

   - コンソールからアプライアンスにアクセスできることを確認します。
   - アプライアンスの NSIP アドレスとルートを確認します。

- IP アドレスがデフォルトの IP アドレス 192.168.100.1 に変更されている場合は、必要に応じて IP アドレスを変更します。
- アプライアンスがアクセス可能であることを確認します。

7. 問題

   アップグレード中に、同期用のコマンドを実行すると、次のメッセージが表示されます。

   コマンドはセカンダリノードで失敗しましたが、プライマリノードでは成功しました。

   解像度

   高可用性（HA）同期の進行中は、依存コマンド（set /unset /bind /unbind）を実行しないでください。

8. 問題

   アップグレードプロセス中、force failover コマンドを実行すると、トラフィックは新しいプライマリノードを通過しません。

   解像度

   - ネットワークトポロジとスイッチの設定に問題がないか確認します。
   - set l2param-garpreply ENABLED コマンドを実行して、GARP 応答を有効にします。
   - 仮想 MAC をまだ使用していない場合は、使用してみてください。
   - プライマリノードから sendarp —a コマンドを実行します。

9. 問題

   NetScaler ADC アプライアンスをアップグレードまたはダウングレードした後、SSH によるアプライアンスへの接続に失敗します。

   解像度

   NetScaler ADC アプライアンスで次の操作を実行します。

   - /nsconfig/ssh/ssh_host_*で古いホストキーまたは安全でないホストキーを削除します。
   - /nsconfig/sshd_configでカスタム SSHD 設定を確認し、それがまだ関連性があり、互換性があるかどうかを確認します。それに応じて、カスタム SSHD 設定の名前を変更するか、削除します。
   - NetScaler ADC アプライアンスをコールドリブートする

10. 問題

    HA ペアでは、force HA failover コマンドを実行した後も、デバイスはリブートし続けます。アップグレード後、セカンダリデバイスが起動しない。

    解像度

    /var ディレクトリがいっぱいかどうかを確認します。その場合は、古いインストールファイルを削除します。df —h コマンドを実行して、使用可能なディスク領域を表示します。

11. 問題

    HA ペアをアップグレードすると、ノードの 1 つが state UNKNOWN としてリストされます。

    解像度

    - 両方のノードが同じビルドを実行しているかどうかを確認します。ビルドが同じではなく、HA ノードのバージョンが一致しない場合、show ha node コマンドを実行すると、一部のフィールドが UNKNOWN と表示されます。
    - セカンダリアプライアンスが到達可能かどうかを確認します。

12. 問題

    NetScaler ADC をアップグレードすると、インターフェイスには、負荷分散仮想サーバーとサービスのほとんどがダウンしていることが示されます。

    解像度

    SNIP アドレスがセカンダリアプライアンスでアクティブであることを確認します。また、show service コマンドを入力して、サービスが実行されているかどうかを確認します。

13. 問題

    アップグレードの実行後、セカンダリアプライアンスですべての仮想サーバがダウンしています。

    解像度

    次のコマンドを実行して、HA ステートと HA 同期を有効にします。

    - set node hastate enable
    - set node hasync enable

    HA を無効にすることは推奨されません。

14. 問題

    ダウングレードを実行した後、NetScaler ADC は正常に起動しません。

    解像度

    正しいライセンスがインストールされているかどうかを確認します。

15. 問題

    HA ペアでは、アップグレードの実行後に一部の機能が同期されません。

    解像度

    sync ha file misc コマンドを実行して、プライマリノードからセカンダリノードに設定ファイルを同期します。

16. 問題

    再起動中に、次のエラーメッセージが表示されます。

ns.conf の 1 つ以上のコマンドが失敗しましたどうしたらいいですか?

解像度

ns.conf ファイル内のコマンドが 255 バイトの制限を超えていないことを確認します。255 バイト制限に対して長すぎるポリシーを作成するコマンドでは、パターンセットを使用してポリシーを短縮できます。

例:

```
add cs policy p11 -rule 'HTTP.REQ.URL.ENDSWITH_ANY("
   ctx_file_extensions")'
Done
<!--NeedCopy-->
```

ctx_file_extensions は、多数の拡張機能をカバーするデフォルトのパッチセットです。デフォルトのパターンセットに加えて、ユーザ定義のパターンセットを作成できます。次のコマンドを実行して、パッチセットを追加します。

```
add patset <name>
<!--NeedCopy-->
```

注: パッチセットは、リリース 9.3 以降でのみサポートされます。

17. 問題

NetScaler VPX アプライアンスをアップグレードすると、/var のスペースを解放するように指示されます。どのファイルを削除すればよいですか?

解像度

/var/tmp/ ディレクトリから古いインストールファイルを削除します。また、/flash から不要なファイルを削除します。

18. 問題

セカンダリアプライアンスで force HA failover コマンドを実行すると、グラフィカルユーザインターフェイス（GUI）に接続できません。

解像度

コマンドラインインターフェイスを使用してセカンダリアプライアンスにログオンし、set ns ip<IP>-gui enabled コマンドを実行して GUI へのアクセスを有効にします。

19. 問題

アップグレードを実行した後、Java アプレットをロードする必要がある GUI 上のリンク（アップグレードウィザードまたはライセンスウィザード）をクリックすると、次のエラーメッセージが表示されます。**GUI** バージョンがカーネルのバージョンと一致しません。このインスタンスを閉じ、**Java** プラグインのキャッシュをクリアして、再度開いてください。

解像度

- GUI を使用して NetScaler ADC にログオンします。
- NetScaler Gateway > グローバル設定に移動します。
- [設定] の [グローバル設定の変更] をクリックします。
- 詳細ウィンドウの [クライアントエクスペリエンス] で、[UI テーマ] の一覧から [既定] を選択します。
- 「OK」をクリックします。

20. 問題

    何らかの理由で NetScaler ADC アプライアンスのアップグレードが失敗した場合、バックアップファイルを使用してアプライアンスを復元するにはどうすればよいですか?

    解像度

    アップグレードが失敗した場合は、バックアップされたファイルを使用して、アプライアンスを以前のバージョンの NetScaler ADC アプライアンスに復元します。詳細については、「NetScaler ADC アプライアンスのバックアップと復元」を参照してください。

    NetScaler ADC クラスターセットアップのバックアップと復元の詳細については、「クラスター設定のバックアップと復元」を参照してください。

21. 問題

    NetScaler ADC アプライアンスのアップグレードに失敗した後にライセンスが見つからない場合、問題を解決するにはどうすればよいですか?

    解像度

    ライセンスが不足している場合、またはライセンスを再割り当てする場合は、次のトピックの「ライセンスの概要」を参照してください。

> 注
>
> これらのトラブルシューティング手順は、複数のリリースにまたがってソフトウェアをダウングレードする際の設定が失われる問題にも適用されます。
>
> その他の問題については、リリースノート、ナレッジセンターの記事、および FAQ を参照してください。

よくある質問

August 15, 2023

NetScaler ADC ファームウェアのアップグレードに関する質問への回答については、「インストール、アップグレード、ダウングレードに関するよくある質問 」を参照してください。

新規および非推奨のコマンド、パラメータ、**SNMP OID**

August 15, 2023

このセクションでは、新規および非推奨のコマンド、パラメータ、および SNMP OID を示します。

新しいコマンド

次の表に、リリース 13.1 の新しいコマンドを示します。

コマンドグループ	コマンド
クラウド	統計クラウド

新しいパラメーター

コマンドグループ	コマンドとパラメータ
アプリケーションファイアウォール	add appfw profile [-clientIpExpression <expression>] ;set appfw profile [-clientIpExpression <expression>]; show appfw profile [-clientIpExpression <expression>]
ボット	add bot profile [-verboseLogLevel (NONE | HTTP_FULL_HEADER)]; set bot profile [-verboseLogLevel (NONE | HTTP_FULL_HEADER)]; show bot profile [verbose Log Level]; set cloud ngsparameter [-csvserverTicketingDecouple (YES | NO)]; show cloud ngsparameter [-csvserverTicketingDecouple]

コマンドグループ	コマンドとパラメータ
GSLB か	set gslb parameter [-GSLBSyncSaveConfigCommand (ENABLED | DISABLED)]、show gslb parameter [GSLBSyncSaveConfigCommand]
NS	set ns tcpParam [-delinkClientServerOnRST (ENABLED | DISABLED)]、show ns tcpParam [delinkClientServerOnRST]
RDP	add rdp clientprofile [-rdpValidateClientIP (ENABLE | DISABLE)];set rdp clientprofile [-rdpValidateClientIP (ENABLE | DISABLE)];show rdp clientprofile [-rdpValidateClientIP]

非推奨のコマンド

コマンドグループ	Commands
NS	add ns trafficDomain; rm ns trafficDomain; bind ns trafficDomain; unbind ns trafficDomain; enable ns trafficDomain; disable ns trafficDomain; show ns trafficDomain; stat ns trafficDomain

コマンドグループ	Commands
WI	add wi site; rm wi site; set wi site; bind wi site; unbind wi site; show wi site; install wi **package**; uninstall wi **package**; show wi **package**
WF	install wf **package**; uninstall wf **package**; show wf **package**; add wf site; rm wf site; set wf site; show wf site

廃止予定の機能が削除されました

次の非推奨機能は削除され、NetScaler ADC バージョン 13.1 以降では構成できなくなりました。

- フィルタ機能 (コンテンツフィルタまたは CF とも呼ばれる)-アクション、ポリシー、およびバインディング。
- SPDY、確実接続（SC)、プライオリティキューイング（PQ)、HTTP サービス拒否（DoS)、および HTML インジェクション機能。
- SSL、コンテンツスイッチング、キャッシュリダイレクト、圧縮、およびアプリケーションファイアウォールに関する従来のポリシー。
- コンテンツスイッチングポリシーの`url`パラメータと`domain`パラメータ。
- 負荷分散永続性ルールのクラシック式。
- 書き換えアクションの`pattern`パラメータ。
- 書き換えアクションの`bypassSafetyCheck`パラメータ。
- 高度な定義式の`SYS.EVAL_CLASSIC_EXPR`。
- `patclass`設定エンティティ。
- 高度な定義式で引数のない`HTTP.REQ.BODY`です。
- 高度な定義式の Q および S プレフィックス。
- 圧縮パラメータ設定の`policyType`パラメータ。(CLI コマンド`set cmp parameter`)。

変換には`nspepi`ツールを使用できます。このツールは NetScaler ADC アプライアンスバージョン 13.0 または 12.1 で実行する必要があります。

詳細については、「クラシックポリシーの廃止に関する FAQ」を参照してください。

また、最新バージョンのツールを使用してクラシック構成から詳細構成に移行するには、「GitHub の NetScaler ADC スクリプト」を参照してください。

新しい **SNMP OID**

詳細については、『SNMP OID リファレンスガイド』を参照してください。

テレコムサービスプロバイダー向けソリューション

August 15, 2023

情報通信技術（ICT）とは、インターネットユーザーをアプリやデータに近づけることです。最新のデータセンターテクノロジーにより、ユーザー、アプリ、データをどこにでも配置できるようになりました。ユーザーは、オフィスや自宅、または空港などの場所からアプリやデータにアクセスできます。アプリとデータは、企業の構内、パブリッククラウド、プライベートクラウド、またはハイブリッドホストのいずれかに配置できます。その結果、生産性が向上しただけでなく、所有コストとメンテナンスのコストも削減されました。

サービスプロバイダーは、ユーザーのアプリやデータをネットワーク経由で伝送するために必要なコアインフラストラクチャを提供します。コアインフラストラクチャは数百万のサブスクライバーと多種多様なアプリやデータに対応しているため、規模とプロトコルサポートの要件は非常に高いです。コアインフラストラクチャは、データプレーンとコントロールプレーンという 2 つの主要なタイプのトラフィックを処理します。これらの各プレーンには、それぞれ独自のスケールとプロトコルサポート要件があります。

データプレーンは、ユーザーアプリとデータをエンドツーエンド、つまりエンドユーザー機器とアプリケーションサーバー間で伝送するコアインフラストラクチャの一部です。アプリやデータにアクセスするユーザーの数は数千万人にも上るため、スループットと IP アドレスの要件は非常に高くなります。ネットワーク上のすべてのユーザーは一意に識別できる必要があります。そうして初めて、サービスプロバイダーはトラフィックを制御し、ネットワークの使用状況を監視し、ユーザー固有のサービスを提供し、情報を正しく記録できます。今日のクライアントデバイスとアプリケーションサーバーの多くは、IPv6 をネイティブにサポートしています。コアインフラストラクチャは、IPv4 と IPv6 のクライアントとサーバーの組み合わせをサポートするだけでなく、IPv4 と IPv6 間の相互通信のためのテクノロジーも提供する必要があります。最後に、サービスプロバイダーは、（エンドユーザー体験に直接関係する）サービスの質と、中断のないサービスの可用性によって評価されます。データプレーンは、品質と可用性の両方を同時に実現できる十分な耐障害性を備えている必要があります。

コントロールプレーンインフラストラクチャは、ユーザートラフィックを管理し、ビジネスおよびネットワーク運用サービスを維持します。このプレーンで実行される多くのプロトコルの中で最も重要なのは、Diameter、Radius、SMPP です。Diameter ameter は基本プロトコルであり、その上で他のいくつかの機能固有のプロトコルが開発されています。次に例を示します：

- ポリシーおよび課金適用機能 (PCEF) とポリシーおよび課金ルール機能 (PCRF) 間の Gx インターフェイス
- オンライン課金システム（OCS）と Cisco パケットデータネットワークゲートウェイ（PGW）/ポリシーおよび課金実施機能（PCEF）間の Gy インターフェイス

コントロールプレーントラフィックの量は、ユーザーのアクティビティに正比例します。コントロールプレーントラフィックを管理するために、サービスプロバイダーは負荷分散やコンテンツスイッチングなどのいくつかの ADC 機

能を使用します。制御プレーントラフィックをきめ細かく制御する必要があります。これは、データプレーントラフィックの複雑さと同じです。

サービスプロバイダーは厳しいサービスレベル契約 (SLA) を満たす必要があり、コンプライアンスについて規制当局から徹底的に精査されます。データやコントロールプレーンのトラフィックを管理しながら要件を順守するには、サービスプロバイダーはインフラストラクチャを機敏に、予算内で、簡単にアップグレードでき、柔軟性を維持する必要があります。現在の市場で最も強力で高度な ADC である NetScaler 製品は、サービスプロバイダー環境には当然のことです。

大規模 **NAT**

August 15, 2023

> 注
>
> この機能は、NetScaler Advanced エディションまたは Premium エディションのライセンスで利用できます。

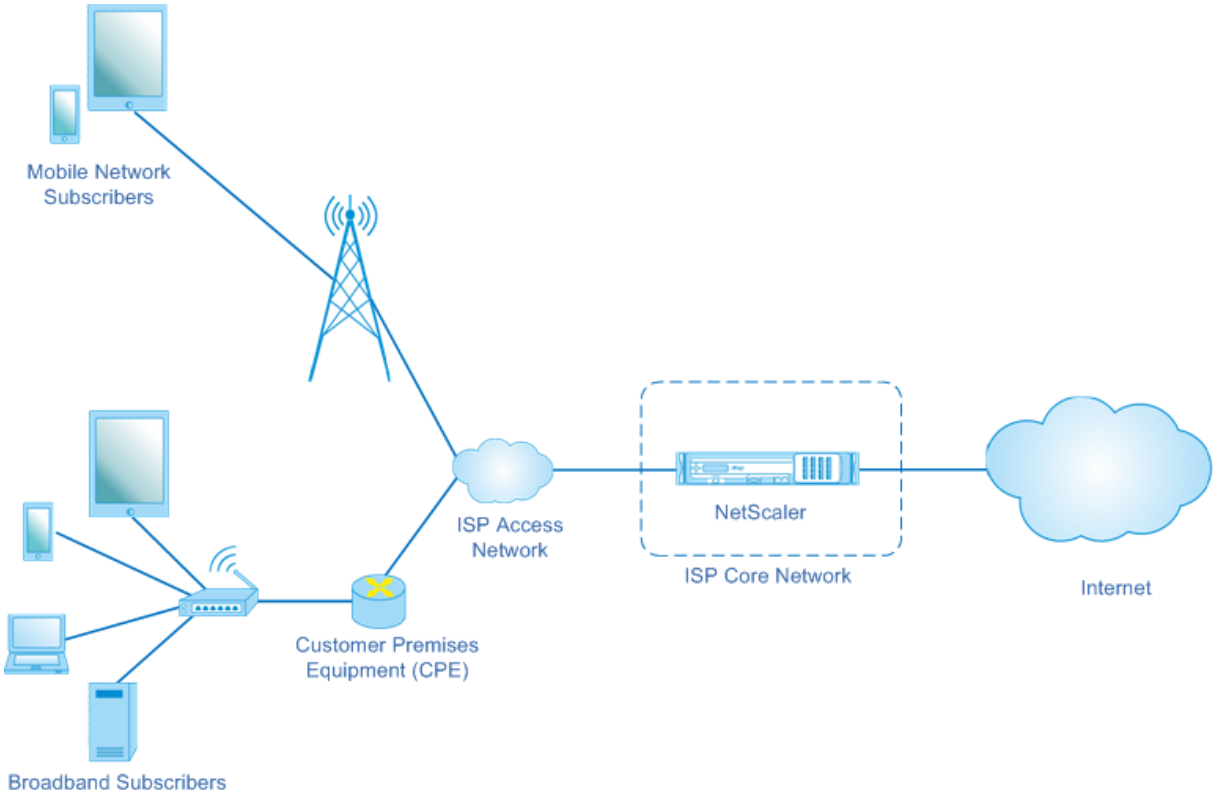
インターネットの驚異的な成長により、パブリック IPv4 アドレスが不足しています。大規模 NAT (LSN/CGNAT) はこの問題を解決し、多数のインターネットユーザー間で少数のパブリック IPv4 アドレスを共有することで、利用可能なパブリック IPv4 アドレスを最大限に活用します。

LSN はプライベート IPv4 アドレスをパブリック IPv4 アドレスに変換します。これには、多数のプライベート IP アドレスをより少ないパブリック IPv4 アドレスに集約するためのネットワークアドレスとポート変換方法が含まれています。LSN は NAT を大規模に処理するように設計されています。NetScaler LSN 機能は、インターネットサービスプロバイダー（ISP）や通信事業者が何百万もの翻訳を提供して多数のユーザー（サブスクライバー）を非常に高いスループットでサポートする場合に非常に役立ちます。

LSN アーキテクチャ

NetScaler 製品を使用する ISP の LSN アーキテクチャは、ISP のコアネットワークに展開された NetScaler アプライアンスを介してインターネットにアクセスするプライベートアドレス空間のサブスクライバー（インターネットユーザー）で構成されます。加入者は ISP のアクセスネットワークを介して ISP に接続されます。通常、インターネットの商用利用契約者は、ISP のアクセスネットワークに直接接続されます。これらのサブスクライバにサービスを提供するには、1 レベルの NAT（NAT44）のみが必要です。

ただし、非営利契約者は通常、NAT も実装しているルーターやモデムなどの顧客宅内機器（CPE）を利用することになります。この 2 つのレベルの NAT によって NAT444 モデルが作成されます。LSN 機能用の NetScaler アプライアンスを ISP のコアネットワークに展開することは、契約者には意識されず、契約者や CPE の設定を変更する必要はありません。



Large Scale NAT

NetScaler アプライアンスは、インターネット宛のすべてのサブスクライバーパケットを受信します。アプライアンスには、LSN に使用する定義済みの NAT IP アドレスのプールが設定されています。NetScaler アプライアンスは LSN 機能を使用して、パケットの送信元 IP アドレス（プライベート）とポートを NAT IP アドレス（パブリック）と NAT ポートに変換し、パケットをインターネット上の宛先に送信します。アプライアンスは、LSN 機能を使用するすべてのアクティブセッションの記録を保持します。これらのセッションは LSN セッションと呼ばれます。NetScaler アプライアンスは、セッションごとにサブスクライバーの IP アドレスとポート、および NAT IP アドレスとポートの間のマッピングも管理します。これらのマッピングは LSN マッピングと呼ばれます。NetScaler アプライアンスは、LSN セッションと LSN マッピングから、特定のセッションに属する（インターネットから受信した）応答パケットを認識します。アプライアンスは、応答パケットの宛先 IP アドレスとポートを NAT IP アドレス:port からサブスクライバの IP アドレス:port に変換し、変換されたパケットを加入者に送信します。

NetScaler アプライアンスでサポートされている **LSN** 機能

以下に、NetScaler アプライアンスでサポートされている LSN 機能の一部について説明します。

NAT リソース割り当て

NetScaler アプライアンスは、事前に定義された NAT リソースプールから NAT IP アドレスとポートを加入者に割り当て、パケットを変換して外部ホスト（インターネット）に送信します。NetScaler アプライアンスは、サブスクライバーの次のタイプの NAT IP アドレスとポート割り当てをサポートします。

- 決定論的。NetScaler アプライアンスは、NAT IP アドレスとポートブロックを各サブスクライバーに割り当てます。アプライアンスは、これらのサブスクライバに NAT リソースを順番に割り当てます。最初の NAT IP アドレスのポートの最初のブロックを最初のサブスクライバ IP アドレスに割り当てます。次の範囲のポートが次のサブスクライバに割り当てられ、NAT アドレスに次のサブスクライバ用のポートが足りなくなるまで続きます。その時点で、次の NAT アドレスの最初のポートブロックが加入者に割り当てられ、以降も同様に加入者に割り当てられます。

  NetScaler アプライアンスは、サブスクライバーに割り当てられた NAT IP アドレスとポートブロックを記録します。接続の場合、マッピングされた NAT IP アドレスとポートブロックだけで加入者を識別できます。このため、NetScaler アプライアンスは作成または削除された LSN セッションを記録しません。ポートブロック全体が使用されている場合、NetScaler アプライアンスはサブスクライバーからの新しい接続をすべて切断します。

- ダイナミック。NetScaler アプライアンスは、サブスクライバーの接続用にランダムな NAT IP アドレスと LSN NAT プールのポートを割り当てます。構成でポートブロック割り当てが有効になっている場合、アプライアンスは初めて接続を開始するときに、加入者にランダムな NAT IP アドレスとポートブロックを割り当てます。次に、NetScaler アプライアンスは、この NAT IP アドレスと、割り当てられたブロックのポートの 1 つを、このサブスクライバーからの後続の各接続に割り当てます。ポートのブロック全体が使用されている場合、アプライアンスは新しい接続を開始するときに、加入者に新しいランダムなポートブロックを割り当てます。新しいポートブロック内のポートの 1 つが新しい接続に割り当てられます。

IP プーリング

次の NAT リソース割り当てオプションは、既存のセッションにランダムな NAT IP アドレスとポートが割り当てられたサブスクライバの以降のセッションで使用できます。

- ペアリング済み。NetScaler アプライアンスは、同じサブスクライバーに関連するすべてのセッションに同じ NAT IP アドレスを割り当てます。そのアドレスで使用できるポートがなくなると、アプライアンスはサブスクライバからの新しい接続をすべて切断します。このオプションは、同じ送信元 IP アドレスで複数のセッションを作成する必要がある特定のアプリケーション (たとえば、RTP または RTCP プロトコルを使用するピアツーピアアプリケーションなど) を適切に機能させるために必要です。
- ランダム。NetScaler アプライアンスは、同じサブスクライバーに関連するさまざまなセッションに、プールからランダムな NAT IP アドレスを割り当てます。

LSN マッピングの再利用

NetScaler アプライアンスは、同じサブスクライバーの IP アドレスとポートから発信される新しい接続に既存の LSN マップを再利用できます。NetScaler LSN 機能では、次の種類の LSN マッピングの再利用がサポートされます。

1. エンドポイントに依存しません。NetScaler アプライアンスは、同じサブスクライバーの IP アドレスとポート（X: X）から任意の外部 IP アドレスとポートに送信される後続のパケットに LSN マッピングを再利用しま

す。このタイプの LSN マップの再利用は、VOIP およびピアツーピアアプリケーションを適切に機能させるのに役立ちます。

2. アドレスにより異なります。NetScaler アプライアンスは、外部ポートに関係なく、同じサブスクライバーの IP アドレスとポート（X: X）から同じ外部 IP アドレス（Y）に送信される後続のパケットに LSN マッピングを再利用します。
3. アドレスポートにより異なります。NetScaler アプライアンスは、マッピングがアクティブな間に、同じ内部 IP アドレスとポート（X: X）から同じ外部 IP アドレスとポート（Y: Y）に送信される後続のパケットに LSN マッピングを再利用します。

LSN フィルタリング

NetScaler アプライアンスは、アクティブな LSN セッションと LSN マッピングに基づいて外部ホストからのパケットをフィルタリングできます。サブスクライバ IP: ポート（X: X）、NAT IP: ポート（N: N）、および外部ホスト IP: ポート（Y: Y）のマッピングを含む LSN マッピングの例を考えてみましょう。NetScaler LSN 機能は次のタイプのフィルタリングをサポートしています。

1. エンドポイントに依存しません。NetScaler アプライアンスは、外部ホスト IP アドレスとポートソース（z: Z）に関係なく、サブスクライバー IP: ポート（X: X）を表す NAT IP: Port（N: N）宛てではないパケットのみを除外します。NetScaler アプライアンスは、x: X 宛てのパケットをすべて転送します。つまり、任意の外部ホストからサブスクライバへのパケットを許可するには、サブスクライバから任意の外部 IP アドレスにパケットを送信するだけで十分です。このタイプのフィルタリングは、VOIP およびピアツーピアアプリケーションを適切に機能させるのに役立ちます。
2. アドレスにより異なります。NetScaler アプライアンスは、サブスクライバーの IP: ポート（x: x）を表す NAT IP: ポート（N: N）宛てではないパケットを除外します。さらに、加入者が以前に y: AnyPort（外部ポートに依存しない）にパケットを送信したことがない場合、アプライアンスは外部ホスト IP アドレスと N: N 宛のポート（Y: Y）からのパケットをフィルタリングします。つまり、特定の外部ホストからパケットを受信するには、加入者が最初にその特定の外部ホストの IP アドレスにパケットを送信する必要があります。
3. アドレスポートにより異なります。NetScaler アプライアンスは、サブスクライバーの IP: ポート（x: x）を表す NAT IP: ポート（N: N）宛てではないパケットを除外します。さらに、加入者が以前に y: y にパケットを送信したことがない場合、アプライアンスは外部ホストの IP アドレスとポート (Y: Y) から N: n 宛てのパケットを除外します。つまり、特定の外部ホストからパケットを受信するには、加入者が最初にその特定の外部 IP アドレスとポートにパケットを送信する必要があります。

クォータ

NetScaler アプライアンスは、各サブスクライバーの NAT ポートとセッションの数を制限して、サブスクライバー間でリソースを公平に分配できます。NetScaler アプライアンスは、サブスクライバーグループのセッション数を制限して、さまざまなサブスクライバーグループ間でリソースを公平に分配することもできます。

- ポートクォータ。NetScaler アプライアンスは、特定のプロトコルで各サブスクライバーが一度に使用する LSN NAT ポートを制限できます。たとえば、各サブスクライバを最大 500 個の TCP NAT ポートに制限できます。サブスクライバーの LSN NAT マッピングが上限に達しても、NetScaler アプライアンスは指定されたプロトコルの追加の NAT ポートをそのサブスクライバーに割り当てません。
- サブスクライバーセッションの制限。サブスクライバの同時セッション数は、ポートクォータを超える場合があります。NetScaler アプライアンスは、特定のプロトコルで各サブスクライバーに許可される LSN セッションを制限できます。LSN セッションの数がサブスクライバーの制限に達すると、NetScaler アプライアンスはサブスクライバーが指定されたプロトコルのセッションを追加で開くことを許可しません。
- グループセッションの制限。NetScaler アプライアンスは、特定のプロトコルのサブスクライバーグループに許可される LSN セッションの総数を制限できます。LSN セッションの総数が特定のプロトコルのグループの上限に達すると、NetScaler アプライアンスはグループのサブスクライバーが指定されたプロトコルのセッションを追加で開くことを許可しません。たとえば、グループの UDP セッションを最大 10000 に制限したとします。このグループの UDP セッションの合計数が 10000 に達すると、NetScaler アプライアンスはグループのどのサブスクライバーも追加の UDP セッションを開くことを許可しません。

アプリケーション層ゲートウェイ

一部のアプリケーション層プロトコルでは、IP アドレスとプロトコルポート番号もパケットのペイロードで通信されます。プロトコルのアプリケーション層ゲートウェイは、パケットのペイロードを解析し、プロトコルが LSN 上で引き続き機能するように必要な変更を行います。

NetScaler アプライアンスは、次のプロトコルの ALG をサポートしています。

- FTP
- ICMP
- TFTP
- PPTP
- SIP
- RTSP

ヘアピンサポート

NetScaler アプライアンスは、NAT IP アドレスを使用したサブスクライバーまたは内部ホスト間の通信をサポートします。NAT IP アドレスを使用する 2 人の加入者間のこの種の通信は、ヘアピンフローと呼ばれます。ヘアピンフローはデフォルトで有効になっており、無効にすることはできません。

LSN を構成する前の考慮事項

August 15, 2023

NetScaler アプライアンスで LSN を構成する前に、次の点を考慮してください。

- RFC 6888、5382、5508、4787 で説明されている大規模な NAT のさまざまなコンポーネントについて理解していることを確認してください。
- エンドポイント独立マッピング (EIM) とエンドポイント独立フィルタリング (EIF) はデフォルトで無効になっています。VoIP およびピアツーピア (P2P) アプリケーションを正しく機能させるには、これらのオプションを有効にする必要があります。
- **LSN** のロギング:LSN 情報のロギングに関する考慮事項は次のとおりです。
  - Citrix では、LSN 情報を NetScaler アプライアンスではなく外部ログサーバーに記録することをお勧めします。外部サーバにログオンすると、アプライアンスが多数の LSN ログエントリ（数百万単位）を作成するときに、最適なパフォーマンスが促進されます。
  - Citrix では、TCP 経由の SYSLOG または NSLOG の使用を推奨しています。デフォルトでは、SYSLOG は UDP を使用し、NSLOG は TCP のみを使用してログ情報をログサーバに転送します。TCP は、完全なデータを転送するために UDP よりも信頼性が高いです。
  - TCP 経由の SYSLOG には次の制限が適用されます。
    * Syslog over TCP ソリューションでは、認証、整合性チェック、プライバシーは提供されません。
    * NetScaler アプライアンスは、TCP プロトコルを使用して外部ログサーバーへの SYSLOG メッセージの配信を確認します。
- 高可用性:LSN 向け NetScaler アプライアンスの高可用性に関する考慮事項は次のとおりです。
  - Citrix では、すべての LSN セッションが中断されずにシームレスに動作するように、2 つの NetScaler アプライアンスの高可用性環境で LSN 機能を構成することをお勧めします。
  - 高可用性環境では、Citrix は次のことを推奨します。
    * すべての HA 関連通信に VLAN を専用にするための SYNC VLAN パラメータを設定します。
    * プライマリノードの対称 RSS キーをセカンダリノードに同期して、多数の LSN マッピングとセッションをステートフル同期します。
    * フェールオーバー後にすべての VLAN で GARP ブロードキャストがフラッディングしないように、LSN IP アドレスのサブネットを VLAN にバインドします。
  - NetScaler アプライアンスの高可用性展開では、ALG 関連のセッションはセカンダリアプライアンスにミラーリングされません。
- アプリケーション層ゲートウェイ（**ALG**）: NetScaler アプライアンスの ALG に関する考慮事項は次のとおりです。
  - SIP ALG では次のものはサポートされていません。
    * マルチキャスト IP アドレス
    * 暗号化された SDP
    * TLS 経由の SIP メッセージ
    * SIP メッセージ内の FQDN 変換
    * SIP メッセージの認証

        * トラフィックドメイン、管理パーティション、NetScaler クラスター
        * マルチパート本文を含む SIP メッセージ。
    - RTSP ALG では次のものはサポートされていません。
        * マルチキャスト RTSP セッション
        * UDP 経由の RTSP セッション
        * NetScaler トラフィックドメイン、管理パーティション、および NetScaler クラスター
    - NetScaler アプライアンスは、IPsec プロトコルの ALG をサポートしていません。

- NetScaler アプライアンスにいくつかの LSN セッションが存在しているときに LSN 機能を無効にすると、これらのセッションは設定されたタイムアウト間隔の間存続します。
- LSN は RNAT よりも優先されます。指定した LSN サブスクライバからのパケットが RNAT ルールにも一致する場合、そのパケットは LSN 設定に従って変換されます。
- LSN セッションのみに関連するパケットの転送は、NetScaler アプライアンスのルーティングテーブルに基づいています。
- サブネット IP アドレスとは異なり、加入者の接続の LSN NAT IP アドレスの選択は、宛先 IP アドレスのルーティングエントリに基づいていません。
- インバウンドパケットでは、スタティック LSN マッピングがダイナミック LSN マッピングよりも優先されます。
- アウトバウンドパケットでは、LSN アプリケーションプロファイルがスタティックマッピングよりも優先されます。
- NetScaler アプライアンスに多数の（100 万を超える）LSN セッションが存在する場合、Citrix ではすべてのセッションではなく選択した LSN セッションを表示することをおすすめします。コマンドラインインターフェイスまたは設定ユーティリティで、選択パラメータを使用して LSN セッション操作を表示します。
- LSN 機能に割り当てられるアクティブメモリの量を減らすには、構成済みメモリ設定を変更した後に NetScaler アプライアンスをウォームリスタートする必要があります。ウォームリスタートを行わないと、アクティブメモリの量を増やすことしかできません。

LSN の構成手順

August 15, 2023

NetScaler アプライアンスでの LSN の構成は、次のタスクで構成されます。

1. グローバル **LSN** パラメータを設定します。グローバルパラメータには、LSN 機能用に確保されている NetScaler メモリの量と、高可用性セットアップでの LSN セッションの同期が含まれます。
2. **LSN** クライアントエンティティを作成し、サブスクライバーをそのエンティティにバインドします。LSN クライアントエンティティは、NetScaler アプライアンスに LSN を実行させたいトラフィックの対象となるサブスクライバーのセットです。クライアントエンティティには、加入者を識別するための IPv4 アドレスと拡張 ACL ルールが含まれます。LSN クライアントは 1 つの LSN グループにしかバインドできません。コマンド

ラインインターフェイスには、LSN クライアントエンティティを作成し、サブスクライバーを LSN クライアントエンティティにバインドする 2 つのコマンドがあります。設定ユーティリティは、これら 2 つの操作を 1 つの画面にまとめます。

3. **LSN** プールを作成し、**NAT IP** アドレスをそれにバインドします。LSN プールは、NetScaler アプライアンスが LSN を実行するために使用する NAT IP アドレスのプールを定義します。プールには、ポートブロック割り当てや NAT タイプ (確定的または動的) などのパラメータが割り当てられます。LSN グループにバインドされた LSN プールは、同じグループにバインドされた LSN クライアントエンティティのすべてのサブスクライバに適用されます。バインドできるのは、同じ NAT タイプ設定の LSN プールと LSN グループだけです。複数の LSN プールを 1 つの LSN グループにバインドできます。ダイナミック NAT では、LSN プールを複数の LSN グループにバインドできます。デターミニスティック NAT では、LSN グループにバインドされたプールを他の LSN グループにバインドすることはできません。コマンドラインインターフェイスには、LSN プールを作成し、NAT IP アドレスを LSN プールにバインドするための 2 つのコマンドがあります。設定ユーティリティは、これら 2 つの操作を 1 つの画面にまとめます。
4. **(**オプション**)** 指定したプロトコルの **LSN** トランスポートプロファイルを作成します。LSN トランスポートプロファイルは、加入者が特定のプロトコルに対して設定できる最大 LSN セッションや最大ポート使用量など、さまざまなタイムアウトと制限を定義します。各プロトコル (TCP、UDP、ICMP) の LSN トランスポートプロファイルを LSN グループにバインドします。プロファイルは複数の LSN グループにバインドできます。LSN グループにバインドされたプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、TCP、UDP、および ICMP プロトコルのデフォルト設定の 1 つの LSN トランスポートプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトトランスポートプロファイルと呼ばれます。LSN グループにバインドした LSN トランスポートプロファイルは、そのプロトコルのデフォルトの LSN トランスポートプロファイルよりも優先されます。
5. （オプション）指定したプロトコルの **LSN** アプリケーションプロファイルを作成し、宛先ポートのセットをそれにバインドします。LSN アプリケーションプロファイルは、特定のプロトコルと宛先ポートセットに対するグループの LSN マッピングと LSN フィルタリング制御を定義します。宛先ポートセットの場合、各プロトコル（TCP、UDP、ICMP）の LSN プロファイルを LSN グループにバインドします。プロファイルは複数の LSN グループにバインドできます。LSN グループにバインドされた LSN アプリケーションプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、すべての宛先ポートの TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN アプリケーションプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトアプリケーションプロファイルと呼ばれます。指定された宛先ポートセットの LSN アプリケーションプロファイルを LSN グループにバインドすると、バインドされたプロファイルは、その宛先ポートセットでそのプロトコルのデフォルトの LSN アプリケーションプロファイルをオーバーライドします。コマンドラインインターフェイスには、LSN アプリケーションプロファイルを作成し、宛先ポートのセットを LSN アプリケーションプロファイルにバインドするための 2 つのコマンドがあります。設定ユーティリティは、これら 2 つの操作を 1 つの画面にまとめます。
6. **LSN** グループを作成し、**LSN** プール、**(**オプション**) LSN** トランスポートプロファイル、**(**オプション**) LSN** アプリケーションプロファイルを **LSN** グループにバインドします。LSN グループは、LSN クライアント、LSN プール、LSN トランスポートプロファイル、および LSN アプリケーションプロファイルで構成されるエ

ンティティです。グループには、ポートブロックサイズや LSN セッションのロギングなどのパラメータが割り当てられます。パラメータ設定は、LSN グループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。バインドできるのは、同じ NAT タイプ設定の LSN プールと LSN グループだけです。複数の LSN プールを 1 つの LSN グループにバインドできます。ダイナミック NAT では、LSN プールを複数の LSN グループにバインドできます。デターミニスティック NAT では、LSN グループにバインドされたプールを他の LSN グループにバインドすることはできません。LSN グループにバインドできる LSN クライアントエンティティは 1 つだけで、LSN グループにバインドされた LSN クライアントエンティティは他の LSN グループにバインドできません。コマンドラインインターフェイスには、LSN グループを作成し、LSN プール、LSN トランスポートプロファイル、LSN アプリケーションプロファイルを LSN グループにバインドするための 2 つのコマンドがあります。設定ユーティリティは、これら 2 つの操作を 1 つの画面にまとめます。

次の表は、NetScaler アプライアンスで作成できるさまざまな LSN エンティティとバインディングの最大数を示しています。これらの制限は、NetScaler アプライアンスで使用可能なメモリによっても異なります。

LSN エンティティとバインディング	上限
LSN クライアント	1024
LSN プール	128
LSN グループ	1024
LSN クライアントにバインドできるサブスクライバネットワーク	64
LSN クライアントにバインドできる拡張 ACL	1024
プール内の NAT IP アドレス	4096
LSN グループにバインドできる LSN プール	8
同じ LSN プールを使用できる LSN グループ	16
LSN グループにバインドできる LSN トランスポートプロファイル	3 (TCP、UDP、および ICMP プロトコルにそれぞれ 1 つ)
同じ LSN トランスポートプロファイルを使用できる LSN グループ	8
LSN グループにバインドできる LSN アプリケーションプロファイル	64
同じ LSN アプリケーションプロファイルを使用できる LSN グループ	8
LSN アプリケーションプロファイルにバインドできるポート範囲	8

コマンドラインインターフェイスを使用した設定

コマンドラインインターフェイスを使用して **LSN** クライアントを作成するには

コマンドプロンプトで入力します。

```
add lsn client <clientname>

show lsn client
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してネットワークアドレスまたは **ACL** ルールを **LSN** クライアントにバインドするには

コマンドプロンプトで入力します。

```
bind lsn client <clientname> ((-network <ip_addr> [-netmask <netmask>]
   [-td<positive_integer>]) | -aclname <string>)

show lsn client
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** プールを作成するには

コマンドプロンプトで入力します。

```
add lsn pool <poolname> [-nattype ( DYNAMIC | DETERMINISTIC )] [-
   portblockallocation ( ENABLED | DISABLED )] [-portrealloctimeout <
   secs>] [-maxPortReallocTmq <positive_integer>]

show lsn pool
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **IP** アドレス範囲を **LSN** プールにバインドするには

コマンドプロンプトで入力します。

```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```

注:LSN プールから LSN IP アドレスを削除するには、unbind lsn pool コマンドを使用してください。

コマンドラインインターフェイスを使用して **LSN** トランスポートプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn transportprofile <transportprofilename> <transportprotocol> [-
    sessiontimeout <secs>] [-finrsttimeout <secs>] [-portquota <
    positive_integer>] [-sessionquota <positive_integer>] [-
    portpreserveparity ( ENABLED | DISABLED )] [-portpreserverange (
    ENABLED | DISABLED )] [-syncheck ( ENABLED | DISABLED )]

show lsn transportprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** アプリケーションプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn appsprofile <appsprofilename> <transportprotocol> [-ippooling (
    PAIRED | RANDOM )] [-mapping <mapping>] [-filtering <filtering>][-
    tcpproxy ( ENABLED | DISABLED )] [-td <positive_integer>]

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してアプリケーションプロトコルのポート範囲を **LSN** アプリケーションプロファイルにバインドするには

コマンドプロンプトで入力します。

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** グループを作成するには

コマンドプロンプトで入力します。

```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC |
    DETERMINISTIC )] [-portblocksize <positive_integer>] [-logging (
    ENABLED | DISABLED )] [-sessionLogging ( ENABLED | DISABLED )][-
    sessionSync (ENABLED | DISABLED )] [-snmptraplimit <positive_integer
    >] [-ftp ( ENABLED | DISABLED )]

show lsn group
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **LSN** プロファイルと **LSN** プールを **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -appsprofilename <string>)

show lsn group
<!--NeedCopy-->
```

設定ユーティリティを使った設定

LSN クライアントを設定し、構成ユーティリティを使用して IPv4 ネットワークアドレスまたは ACL ルールをバインドするには

[システム] > [大規模 **NAT**] > [クライアント] に移動し、クライアントを追加してから、IPv4 ネットワークアドレスまたは ACL ルールをクライアントにバインドします。

設定ユーティリティを使用して LSN プールを設定し、NAT IP アドレスをバインドするには

[システム**] > [**大規模 **NAT**] > [プール] に移動し、プールを追加してから、NAT IP アドレスまたは NAT IP アドレスの範囲をプールにバインドします。

構成ユーティリティを使用して LSN トランスポートプロファイルを構成するには

1. [システム**] > [**大規模 **NAT] > [**プロファイル] に移動します。
2. 詳細ウィンドウで [トランスポート] タブをクリックし、トランスポートプロファイルを追加します。

構成ユーティリティを使用して LSN アプリケーションプロファイルを構成するには

1. [システム**] > [**大規模 **NAT] > [**プロファイル] に移動します。
2. 詳細ペインで [アプリケーション] タブをクリックし、アプリケーションプロファイルを追加します。

設定ユーティリティを使用して LSN グループを設定し、LSN クライアント、プール、トランスポートプロファイル、およびアプリケーションプロファイルをバインドするには

[システム**] > [**大規模 **NAT] > [**グループ] に移動し、グループを追加してから、LSN クライアント、プール、トランスポートプロファイル、およびアプリケーションプロファイルをグループにバインドします。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- add lsn client
  - clientname

    LSN クライアントエンティティの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号

(=)、およびハイフン (-) 文字のみを含める必要があります。LSN クライアントの作成後は変更できません。次の要件は CLI にのみ適用されます。名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲んでください (たとえば、「lsn client1」や「lsn client1」)。

これは必須の議論です。最大長：127

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- bind lsn client
  - clientname

    LSN クライアントエンティティの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。LSN クライアントの作成後は変更できません。次の要件は CLI にのみ適用されます。名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲んでください (たとえば、「lsn client1」や「lsn client1」)。

    これは必須の議論です。最大長：127
  - network

    NetScaler アプライアンスに大規模 NAT を実行させたいトラフィックの対象となる LSN サブスクライバーまたはサブスクライバーネットワークの IPv4 アドレス。
  - ネットマスク

    Network パラメータで指定された IPv4 アドレスのサブネットマスク。

    デフォルト値:255.255.255.255
  - td

    このサブスクライバまたはサブスクライバネットワーク（ネットワークパラメータで指定）が属するトラフィックドメインの ID。

    ID を指定しない場合、加入者または加入者ネットワークはデフォルトのトラフィックドメインの一部になります。

    デフォルト値:0

    最小値:0

    最大値:4094
  - aclname

    アクションが ALLOW に設定されている拡張 ACL の名前。拡張 ACL ルールで指定された条件により、NetScaler アプライアンスが大規模 NAT を実行する対象となる LSN サブスクライバーからのトラフィックが識別されます。最大長：127

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- add lsn pool
  - プール名

    LSN プールの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。LSN プールの作成後は変更できません。次の要件は CLI にのみ適用されます。名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲みます (たとえば、「lsn pool1」や「lsn pool1」)。

    これは必須の議論です。最大長：127

  - ナットタイプ

    (LSN グループにバインドされた LSN クライアントエンティティの）サブスクライバの NAT IP アドレスと（LSN グループにバインドされた LSN プールからの）ポート割り当てのタイプ:

    使用可能なオプションは次のように機能します。

    * 確定的：（LSN グループにバインドされた LSN クライアントの）各サブスクライバに NAT IP アドレスとポートブロックを割り当てます。NetScaler アプライアンスは、これらのサブスクライバーに NAT リソースを順番に割り当てます。NetScaler アプライアンスは、開始 NAT IP アドレスの最初のポートブロック（LSN グループのポートブロックサイズパラメーターによって決定されるブロックサイズ）を最初のサブスクライバー IP アドレスに割り当てます。次の範囲のポートが次のサブスクライバに割り当てられ、NAT アドレスに次のサブスクライバ用のポートが足りなくなるまで続きます。この場合、次の NAT アドレスの最初のポートブロックが加入者に使用され、以降も同様です。各サブスクライバには確定的な NAT IP アドレスとポートブロックが割り当てられるため、ロギングをしなくてもサブスクライバを識別できます。接続では、NAT の IP アドレスとポート、および宛先 IP アドレスとポートのみに基づいて加入者を識別できます。
    * ダイナミック：加入者接続用に、ランダムな NAT IP アドレスと LSN NAT プールのポートを割り当てます。ポートブロック割り当てが（LSN プールで）有効で、ポートブロックサイズが（LSN グループで）指定されている場合、NetScaler アプライアンスは、加入者が初めて接続を開始するときに、加入者にランダムな NAT IP アドレスとポートブロックを割り当てます。アプライアンスは、このサブスクライバからのさまざまな接続に、この NAT IP アドレスと（割り当てられたポートブロックからの）ポートを割り当てます。加入者が割り当てたポートブロックからすべてのポートが（異なる加入者接続に）割り当てられる場合、アプライアンスは加入者に新しいランダムなポートブロックを割り当てます。バインドできるのは、同じ NAT タイプ設定の LSN プールと LSN グループだけです。複数の LSN プールを 1 つの LSN グループにバインドできます。

      設定可能な値: ダイナミック、デターミニスティック

      デフォルト値: ダイナミック

- ポートブロック割り当て

  NAT 割り当てがダイナミック NAT に設定されている場合、NAT IP アドレスの使用可能な NAT ポートプールからランダムな NAT ポートブロックをサブスクライバごとに割り当てます。サブスクライバーから開始された接続では、NetScaler アプライアンスはサブスクライバーに割り当てられた NAT ポートブロックから NAT ポートを割り当てて LSN セッションを作成します。

  バインドされた LSN グループのポートブロックサイズを設定する必要があります。サブスクライバーの場合、サブスクライバーに割り当てられたポートブロックからすべてのポートが割り当てられる場合、NetScaler アプライアンスはサブスクライバーに新しいランダムなポートブロックを割り当てます。

  Deterministic NAT では、このパラメータはデフォルトで有効になっており、無効にすることはできません。

  設定可能な値：ENABLED, DISABLED

  デフォルト値: 無効

- portrealloctimeout

  LSN NAT ポートの割り当て解除（LSN マッピングの削除時）から、新しい LSN セッションへの再割り当てまでの待機時間（秒単位）。このパラメータは、古いマッピングとセッションとセッションの衝突を防ぐために必要です。これにより、別のサブスクライバにリダイレクトされるのではなく、確立されたすべてのセッションが中断されます。これは次の場所で使用されるポートには適用されません。

  * Deterministic NAT
  * アドレス依存フィルタリングとアドレスポート依存フィルタリング
  * ポートブロック割り当てによるダイナミック NAT

  このような場合、ポートはすぐに再割り当てされます。

  デフォルト値:0

  最大値:600

- maxPortReallocTmq

  各 NAT IP アドレスにポート再割り当てタイムアウトが適用されるポートの最大数。つまり、NAT IP アドレスごとに再割り当てタイムアウトが適用される、割り当て解除されたポートキューの最大サイズです。

  キューサイズがいっぱいになると、割り当てが解除された次のポートがただちに新しい LSN セッションに再割り当てされます。

  デフォルト値:65536

  最大値:65536

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- bind lsn pool
  - プール名

    LSN プールの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。LSN プールの作成後は変更できません。次の要件は CLI にのみ適用されます。名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲みます (たとえば、「lsn pool1」や「lsn pool1」)。

    これは必須の議論です。最大長：127
  - lsnip

    LSN の NAT IP アドレスとして使用する IPv4 アドレスまたは IPv4 アドレスの範囲。

    プールが作成されると、これらの IPv4 アドレスは、LSN タイプの NetScaler 所有 IP アドレスとして NetScaler アプライアンスに追加されます。LSN プールに関連付けられた LSN IP アドレスは、他の LSN プールと共有できません。このパラメータに指定する IP アドレスは、NetScaler が所有する IP アドレスと同様に、NetScaler アプライアンスにすでに存在していてはなりません。コマンドラインインターフェイスでは、範囲をハイフンで区切ります。例:10.102.29.30-10.102.29.189。後で LSN IP アドレスの一部またはすべてをプールから削除し、LSN プールに IP アドレスを追加できます。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- lsn トランスポートプロファイルを追加する
  - transportprofilename

    LSN トランスポートプロファイルの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。LSN トランスポートプロファイルの作成後は変更できません。次の要件は CLI にのみ適用されます。名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲んでください (たとえば、「lsn transport profile1」や「lsn transport profile1」)。

    これは必須の議論です。最大長：127
  - トランスポートプロトコル

    LSN トランスポートプロファイルパラメータを設定するプロトコル。

    これは必須の議論です。

    設定可能な値:TCP、UDP、ICMP

- セッションタイムアウト

  アイドル状態の LSN セッションのタイムアウト (秒単位)。LSN セッションがこの値を超える時間アイドル状態になると、NetScaler アプライアンスはそのセッションを削除します。

  このタイムアウトは、いずれかのエンドポイントから FIN または RST メッセージを受信したときの TCP LSN セッションには適用されません。

  デフォルト値:120

  最小値:60

- 1 回目のタイムアウト

  エンドポイントの 1 つから FIN または RST メッセージを受信した後の TCP LSN セッションのタイムアウト（秒単位)。

  TCP LSN セッションが（NetScaler アプライアンスが FIN または RST メッセージを受信した後の）アイドル状態がこの値を超えると、NetScaler アプライアンスはそのセッションを削除します。

  NetScaler アプライアンスの LSN 機能は TCP LSN セッションの状態情報を保持しないため、このタイムアウトはもう一方のエンドポイントからの FIN、RST、ACK メッセージの送信に対応し、両方のエンドポイントが接続を適切に閉じることができます。

  デフォルト値:30

- portquota

  指定されたプロトコルで各サブスクライバが一度に使用する LSN NAT ポートの最大数。たとえば、各サブスクライバは最大 500 の TCP NAT ポートに制限できます。サブスクライバーの LSN NAT マッピングが上限に達しても、NetScaler アプライアンスはサブスクライバーに追加の NAT ポートを割り当てません。

  デフォルト値:0

  最小値:0

  最大値:65535

- sessionquota

  指定されたプロトコルで各サブスクライバに許可される同時 LSN セッションの最大数。LSN セッションの数がサブスクライバーの制限に達すると、NetScaler アプライアンスはサブスクライバーが追加のセッションを開くことを許可しません。

  デフォルト値:0

  最小値:0

  最大値:65535

- portpreserveparity

サブスクライバポートとマップされた LSN NAT ポート間のポートパリティを有効にします。たとえば、サブスクライバーが奇数番号のポートから接続を開始すると、NetScaler アプライアンスはこの接続に奇数番号の LSN NAT ポートを割り当てます。RTP または RTCP プロトコルを使用するピアツーピアアプリケーションなど、送信元ポートを偶数または奇数にする必要があるプロトコルが正しく機能するには、このパラメータを設定する必要があります。

設定可能な値：ENABLED, DISABLED

デフォルト値: 無効

- portpreserverange

  加入者が既知のポート（0 ～1023）から接続を開始する場合、この接続には既知のポート範囲（0 ～1023）の NAT ポートを割り当てます。たとえば、加入者がポート 80 から接続を開始した場合、NetScaler アプライアンスはポート 100 をこの接続の NAT ポートとして割り当てることができます。

  このパラメータは、ポートブロック割り当てのないダイナミック NAT に適用されます。割り当てられたポートの範囲に既知のポートが含まれる場合は、Deterministic NAT にも適用されます。

  使用可能なすべての NAT IP アドレスの既知のポートがすべて異なるサブスクライバー接続（LSN セッション）で使用され、サブスクライバーが既知のポートから接続を開始すると、NetScaler アプライアンスはこの接続を切断します。

  設定可能な値：ENABLED, DISABLED

  デフォルト値: 無効

- syncheck

  NetScaler アプライアンスに LSN-NAT セッションが存在しない接続では、非 SYN パケットをサイレントドロップします。

  このパラメーターを無効にすると、NetScaler アプライアンスは SYN 以外のパケットをすべて受け入れ、この接続の新しい LSN セッションエントリを作成します。

  NetScaler アプライアンスがこのようなパケットを受信する理由は次のとおりです。

  * 接続用の LSN セッションは存在していましたが、設定されたセッションタイムアウトを超える時間 LSN セッションがアイドル状態だったため、NetScaler アプライアンスによってこのセッションが削除されました。
  * このようなパケットは DoS 攻撃の一部になる可能性があります。

  設定可能な値：ENABLED, DISABLED

  デフォルト値: 有効

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- add lsn appsprofile

- appsprofilename

  LSN アプリケーションプロファイルの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。LSN アプリケーションプロファイルの作成後は変更できません。次の要件は CLI にのみ適用されます。名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲みます (たとえば、「lsn アプリケーションプロファイル 1」や「lsn アプリケーションプロファイル 1」)。

  これは必須の議論です。最大長：127

- トランスポートプロトコル

  この LSN アプリケーションプロファイルのパラメータが適用されるプロトコルの名前。

  これは必須の議論です。

  設定可能な値:TCP、UDP、ICMP

- ippooling

  同じサブスクライバに関連するセッションの NAT IP アドレス割り当てオプション。

  使用可能なオプションは次のように機能します。

  * ペアリング：NetScaler アプライアンスは、同じサブスクライバーに関連するすべてのセッションに同じ NAT IP アドレスを割り当てます。（同じサブスクライバーまたは複数のサブスクライバーの）LSN セッションで NAT IP アドレスのすべてのポートが使用されると、NetScaler アプライアンスはサブスクライバーからの新しい接続をすべて切断します。
  * ランダム—NetScaler アプライアンスは、同じサブスクライバーに関連するさまざまなセッションに、プールからランダムな NAT IP アドレスを割り当てます。

  このパラメータはダイナミック NAT 割り当てにのみ適用されます。

  設定可能な値: ペア、ランダム

  デフォルト値: ランダム

- マッピング

  同じサブスクライバ IP アドレスとポートから送信される後続のパケットに適用する LSN マッピングのタイプ。

  サブスクライバ IP: ポート（X: X)、NAT IP: ポート（N: N)、および外部ホスト IP: ポート（Y: Y）のマッピングを含む LSN マッピングの例を考えてみましょう。

  使用可能なオプションは次のように機能します。

  * **ENDPOINT-INDENDEPENTED**：同じサブスクライバの IP アドレスとポート（X: X）から任意の外部 IP アドレスとポートに送信される後続のパケットに LSN マッピングを再利用します。

* **ADDRESS-DEPENTED**：外部ポートに関係なく、同じサブスクライバの IP アドレスとポート（X: X）から同じ外部 IP アドレス（Y）に送信される後続のパケットに LSN マッピングを再利用します。
* **ADDRESS-PORT-DEPENDEN**：マッピングがまだアクティブな間に、同じ内部 IP アドレスとポート（X: X）から同じ外部 IP アドレスとポート（Y: Y）に送信される後続のパケットに LSN マッピングを再利用します。

指定できる値: エンドポイント独立、アドレス依存、アドレスポート依存

デフォルト値:ADDRESS-PORT-DEPENDENT

– filtering

外部ホストから送信されるパケットに適用するフィルタのタイプ。

サブスクライバ IP: ポート（X: X）、NAT IP: ポート（N: N）、および外部ホスト IP: ポート（Y: Y）のマッピングを含む LSN マッピングの例を考えてみましょう。

使用可能なオプションは次のように機能します。

* エンドポイント・インディペンデント：外部ホストの IP アドレスとポートソース（Z: Z）に関係なく、サブスクライバの IP アドレスとポート X: X 宛ではないパケットのみを除外します。NetScaler アプライアンスは、x: X 宛てのパケットをすべて転送します。つまり、任意の外部ホストからサブスクライバへのパケットを許可するには、サブスクライバから任意の外部 IP アドレスにパケットを送信するだけで十分です。
* アドレス依存：加入者の IP アドレスとポート X: x 宛ではないパケットを除外します。さらに、クライアントが以前に y: AnyPort（外部ポートに依存しない）にパケットを送信したことがない場合、アプライアンスは Y: Y からのサブスクライバ（X: X）宛てのパケットをフィルタリングします。つまり、特定の外部ホストからパケットを受信するには、加入者が最初にその特定の外部ホストの IP アドレスにパケットを送信する必要があります。
* **ADDRESS PORT** DEPENDENDEND（デフォルト）：加入者の IP アドレスとポート（X: X）宛てではないパケットを除外します。さらに、NetScaler アプライアンスは、加入者が以前に Y: y にパケットを送信したことがない場合、加入者（X: X）宛ての Y: Y からのパケットを除外します。つまり、特定の外部ホストからパケットを受信するには、加入者が最初にその外部 IP アドレスとポートにパケットを送信する必要があります。

指定できる値: エンドポイント独立、アドレス依存、アドレスポート依存

デフォルト値:ADDRESS-PORT-DEPENDENT

– tcpproxy

TCP プロキシを有効にします。これにより、NetScaler アプライアンスはレイヤー 4 機能を使用して TCP トラフィックを最適化できます。

設定可能な値：ENABLED, DISABLED

デフォルト値: 無効

- td

  LSN の実行後に NetScaler アプライアンスがアウトバウンドトラフィックを送信するトラフィックドメインの ID。

  ID を指定しない場合、アプライアンスは ID が 0 のデフォルトのトラフィックドメインを介してアウトバウンドトラフィックを送信します。

  デフォルト値:65535

  最大値:65535

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- bind lsn appsprofile
  - appsprofilename

    LSN アプリケーションプロファイルの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。LSN アプリケーションプロファイルの作成後は変更できません。次の要件は CLI にのみ適用されます。名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲みます (たとえば、「lsn アプリケーションプロファイル 1」や「lsn アプリケーションプロファイル 1」)。

    これは必須の議論です。最大長：127

  - lsnport

    加入者からの着信パケットの宛先ポートと一致するポート番号またはポート番号の範囲。宛先ポートが一致すると、LSN アプリケーションプロファイルが LSN セッションに適用されます。ポートの範囲はハイフンで区切ります。たとえば、40-90 と入力します。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- add lsn group
  - グループ名

    LSN グループの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。LSN グループの作成後は変更できません。次の要件は CLI にのみ適用されます。名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲みます (たとえば、「lsn group1」や「lsn group1」)。

    これは必須の議論です。最大長：127

- clientname

  LSN グループに関連付ける LSN クライアントエンティティの名前。1 つの LSN グループに関連付けることができる LSN クライアントエンティティは 1 つだけです。LSN グループを作成した後は、この関連付けを削除したり、別の LSN クライアントエンティティに置き換えたりすることはできません。

  これは必須の議論です。最大長：127

- ナットタイプ

  加入者の NAT IP アドレスと（バインドされた LSN プールからの）ポート割り当てのタイプ:

  使用可能なオプションは次のように機能します。

  * 確定的：（LSN グループにバインドされた LSN クライアントの）各サブスクライバに NAT IP アドレスとポートブロックを割り当てます。NetScaler アプライアンスは、これらのサブスクライバーに NAT リソースを順番に割り当てます。NetScaler アプライアンスは、開始 NAT IP アドレスの最初のポートブロック（LSN グループのポートブロックサイズパラメーターによって決定されるブロックサイズ）を最初のサブスクライバー IP アドレスに割り当てます。次の範囲のポートが次のサブスクライバに割り当てられ、NAT アドレスに次のサブスクライバ用のポートが足りなくなるまで続きます。この場合、次の NAT アドレスの最初のポートブロックが加入者に使用され、以降も同様です。各サブスクライバには確定的な NAT IP アドレスとポートブロックが割り当てられるため、ロギングをしなくてもサブスクライバを識別できます。接続では、NAT の IP アドレスとポート、および宛先 IP アドレスとポートのみに基づいて加入者を識別できます。
  * ダイナミック：加入者の接続に、ランダムな NAT IP アドレスと LSN NAT プールのポートを割り当てます。ポートブロック割り当てが（LSN プールで）有効で、ポートブロックサイズが（LSN グループで）指定されている場合、NetScaler アプライアンスは、加入者が初めて接続を開始するときに、加入者にランダムな NAT IP アドレスとポートブロックを割り当てます。アプライアンスは、このサブスクライバからのさまざまな接続に、この NAT IP アドレスと（割り当てられたポートブロックからの）ポートを割り当てます。加入者が割り当てたポートブロックからすべてのポートが（異なる加入者接続に）割り当てられる場合、アプライアンスは加入者に新しいランダムなポートブロックを割り当てます。

  設定可能な値: ダイナミック、デターミニスティック

  デフォルト値: ダイナミック

- portblocksize

  各サブスクライバに割り当てられる NAT ポートブロックのサイズ。

  ダイナミック NAT にこのパラメータを設定するには、バインドされた LSN プールでポートブロック割り当てパラメータを有効にする必要があります。Deterministic NAT では、ポートブロック割り当てパラメーターは常に有効で、無効にすることはできません。

  動的 NAT では、NetScaler アプライアンスは、NAT IP アドレスの使用可能な NAT ポートプールから、各サブスクライバーにランダムな NAT ポートブロックを割り当てます。サブスクライバの場合、サブス

クライバに割り当てられたポートブロックからすべてのポートが割り当てられる場合、アプライアンスはサブスクライバに新しいランダムポートブロックを割り当てます。

- logging

  この LSN グループで作成または削除されたログマッピングエントリとセッション。NetScaler アプライアンスは、ログとセッションログの両方のパラメーターが有効になっている場合にのみ、この LSN グループの LSN セッションをログに記録します。

  アプライアンスは、既存の Syslog および監査ログフレームワークを使用して LSN 情報を記録します。関連する NSLOG アクションと SYLOG アクションエンティティの LSN パラメータを有効にして、グローバルレベルの LSN ロギングを有効にする必要があります。Logging パラメーターを有効にすると、NetScaler アプライアンスはこの LSN グループの LSN マッピングと LSN セッションに関連するログメッセージを生成します。次に、アプライアンスはこれらのログメッセージを NSLOG アクションと SYSLOG アクションエンティティに関連付けられたサーバーに送信します。

  LSN マッピングエントリのログメッセージは、次の情報で構成されます。

  * NetScaler アプライアンスの NSIP アドレス
  * タイムスタンプ
  * エントリータイプ (マッピングまたはセッション)
  * LSN マッピングエントリが作成されたか、削除されたか
  * 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
  * NAT IP アドレスとポート
  * プロトコル名
  * 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が表示される場合があります。
    - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートは記録されません
    - アドレス依存マッピングでは、宛先 IP アドレスのみ (ポートは記録されない) が記録される
    - 宛先 IP アドレスとポートはアドレス-ポート依存マッピング用にログに記録されます

  設定可能な値: ENABLED, DISABLED

  デフォルト値: 無効

- sessionLogging

  LSN グループ用に作成または削除されたログセッション。NetScaler アプライアンスは、ログとセッションログの両方のパラメーターが有効になっている場合にのみ、この LSN グループの LSN セッションをログに記録します。

  LSN セッションのログメッセージは、次の情報で構成されます。

  * NetScaler アプライアンスの NSIP アドレス
  * タイムスタンプ
  * エントリータイプ (マッピングまたはセッション)
  * LSN セッションが作成されるか削除されるか

* 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
* NAT IP アドレスとポート
* プロトコル名
* 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

設定可能な値：ENABLED, DISABLED

デフォルト値: 無効

- sessionSync

  高可用性 (HA) 展開では、この LSN グループに関連するすべての LSN セッションの情報をセカンダリノードと同期します。フェールオーバー後、確立された TCP 接続と UDP パケットフローはアクティブなままになり、セカンダリノード（新しいプライマリ）で再開されます。

  この設定が機能するには、グローバルセッション同期パラメータを有効にする必要があります。

  設定可能な値：ENABLED, DISABLED

  デフォルト値: 有効

- snmptraplimit

  1 分間に生成できる LSN グループの SNMP トラップメッセージの最大数。

  デフォルト値:100

  最小値:0

  最大値:10000

- ftp

  FTP プロトコルのアプリケーション層ゲートウェイ (ALG) を有効にします。一部のアプリケーション層プロトコルでは、通常、IP アドレスとプロトコルポート番号がパケットペイロードで通信されます。ALG として機能する場合、アプライアンスはパケットのペイロードを変更して、プロトコルが LSN 上で引き続き機能するようにします。

  注：NetScaler アプライアンスには、ICMP および TFTP プロトコル用の ALG も含まれています。ICMP プロトコルの ALG はデフォルトで有効になっており、無効にする設定はありません。TFTP プロトコルの ALG はデフォルトで無効になっています。エンドポイントに依存しないマッピング、エンドポイントに依存しないフィルタリング、宛先ポートを 69（TFTP 用の既知のポート）に設定した UDP LSN アプリケーションプロファイルを LSN グループにバインドすると、ALG は LSN グループに対して自動的に有効になります。

  設定可能な値：ENABLED, DISABLED

  デフォルト値: 有効

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- bind lsn group
  - グループ名

    LSN グループの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。LSN グループの作成後は変更できません。次の要件は CLI にのみ適用されます。名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲みます (たとえば、「lsn group1」や「lsn group1」)。

    これは必須の議論です。最大長：127

  - プール名

    指定した LSN グループにバインドする LSN プールの名前。バインドできるのは、同じ NAT タイプ設定の LSN プールと LSN グループだけです。複数の LSN プールを 1 つの LSN グループにバインドできます。

    デターミニスティック NAT では、LSN グループにバインドされたプールを他の LSN グループにバインドすることはできません。ダイナミック NAT では、LSN グループにバインドされたプールを複数の LSN グループにバインドできます。最大長：127

  - transportprofilename

    指定した LSN グループにバインドする LSN トランスポートプロファイルの名前。設定を指定するプロトコルごとにプロファイルをバインドします。

    デフォルトでは、TCP、UDP、および ICMP プロトコルのデフォルト設定の 1 つの LSN トランスポートプロファイルは、作成時に LSN グループにバインドされます。このプロファイルはデフォルトトランスポートと呼ばれます。

    LSN グループにバインドした LSN トランスポートプロファイルは、そのプロトコルのデフォルトの LSN トランスポートプロファイルよりも優先されます。最大長：127

  - appsprofilename

    指定した LSN グループにバインドする LSN アプリケーションプロファイルの名前。宛先ポートの各セットについて、設定を指定する各プロトコルのプロファイルをバインドします。

    デフォルトでは、すべての宛先ポートの TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN アプリケーションプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトアプリケーションプロファイルと呼ばれます。

    指定された宛先ポートセットの LSN アプリケーションプロファイルを LSN グループにバインドすると、バインドされたプロファイルは、その宛先ポートセットでそのプロトコルのデフォルトの LSN アプリケーションプロファイルをオーバーライドします。最大長：127

LSN の構成例

August 15, 2023

次に、コマンドラインインターフェイスを使用して LSN を設定する例を示します。

1 つのサブスクライバネットワーク、**1** つの **LSN NAT IP** アドレス、およびデフォルト設定を使用して簡単な **LSN** 設定を作成します。

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done
<!--NeedCopy-->
```

LSN 加入者を識別するための拡張 **ACL** を含む **LSN** 設定を作成します。

```
add ns acl LSN-ACL-2 ALLOW -srcIP 192.0.2.10-192.0.2.20

Done

apply acls

Done

add lsn client LSN-CLIENT-2

Done

bind lsn client LSN-CLIENT-2  - aclname LSN-ACL-2

Done
```



```
16
17 add lsn pool LSN-POOL-2
18
19 Done
20
21 bind lsn pool LSN-POOL-2 203.0.113.5-203.0.113.10
22
23 Done
24
25 add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2
26
27 Done
28
29 bind lsn group LSN-GROUP-2 -poolname LSN-POOL-2
30
31 Done
32 <!--NeedCopy-->
```

HTTP プロトコル **(**ポート **80)** にはエンドポイントに依存しないマッピング、**SSH** プロトコル **(**ポート **22)** にはアドレスとポートに依存するマッピングを使用して **LSN** 構成を作成します。また、**TCP** プロトコルには最大 **1000** 個の **NAT** ポート、**UDP** プロトコルには **100** 個の **NAT** ポートを使用するように、各サブスクライバを制限してください。**TCP** プロトコルの同時セッション数が最大 **2000** になるように、各サブスクライバを制限します。**TCP** プロトコルの同時セッション数が最大 **30000** になるようにグループを制限します。

```
1  add lsn client LSN-CLIENT-3
2
3  Done
4
5  bind lsn client LSN-CLIENT-3 -network 192.0.3.0 -netmask 255.255.255.0
6
7  Done
8
9  add lsn pool LSN-POOL-3
10
11 Done
12
13 bind lsn pool LSN-POOL-3 203.0.113.11
14
15 Done
16
17 add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-3
18
19 Done
20
21 bind lsn group LSN-GROUP-3 -poolname LSN-POOL-3
22
23 Done
24
25 add lsn appsprofile LSN-APPS-HTTPPROFILE-3 TCP -mapping ENDPOINT-
       INDEPENDENT
26
27 Done
```



```
28
29 bind lsn appsprofile LSN-APPS-HTTPPROFILE-3 80
30
31 Done
32
33 bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-HTTPPROFILE
       -3
34
35 Done
36
37 add lsn appsprofile LSN-APPS-SSHPROFILE-3 TCP -mapping ADDRESS-PORT-
       DEPENDENT
38
39 Done
40
41 bind lsn appsprofile LSN-APPS-SSHPROFILE-3 22
42
43 Done
44
45 bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-SSHPROFILE
       -3
46
47 Done
48
49 add lsn transportprofile LSN-TRANS-PROFILE-TCP-3 TCP -portquota 1000 -
       sessionquota 2000 -groupSessionLimit 30000
50
51 Done
52
53 bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-TCP
       -3
54
55 Done
56
57 add lsn transportprofile LSN-TRANS-PROFILE-UDP-3 UDP -portquota 100
58
59 Done
60
61 bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-UDP
       -3
62
63 Done
64 <!--NeedCopy-->
```

多数のサブスクライバ用の **LSN** 設定を作成します。

```
1 add lsn client LSN-CLIENT-4
2
3 Done
4
5 bind lsn client LSN-CLIENT-4 -network 192.0.4.0 -netmask 255.255.255.0
6
7 Done
```



```
8
9  bind lsn client LSN-CLIENT-4 -network 192.0.5.0 -netmask 255.255.255.0
10
11 Done
12
13 bind lsn client LSN-CLIENT-4 -network 192.0.6.0 -netmask 255.255.255.0
14
15 Done
16
17 bind lsn client LSN-CLIENT-4 -network 192.0.7.0 -netmask 255.255.255.0
18
19 Done
20
21 bind lsn client LSN-CLIENT-4 -network 192.0.8.0 -netmask 255.255.255.0
22
23 Done
24
25 add lsn pool LSN-POOL-4
26
27 Done
28
29 bind lsn pool LSN-POOL-4 203.0.113.30-203.0.113.40
30
31 Done
32
33 bind lsn pool LSN-POOL-4 203.0.113.45-203.0.113.50
34
35 Done
36
37 bind lsn pool LSN-POOL-4 203.0.113.55-203.0.113.60
38
39 Done
40
41 add lsn group LSN-GROUP-4 -clientname LSN-CLIENT-4
42
43 Done
44
45 bind lsn group LSN-GROUP-4 -poolname LSN-POOL-4
46
47 Done
48
49 add lsn appsprofile LSN-APPS-WELLKNOWNPROFILE-4 TCP -mapping ENDPOINT-
      INDEPENDENT
50
51 Done
52
53 bind lsn appsprofile LSN-APPS-WELLKNOWN-PORTS-PROFILE-4 1- 1023
54
55 Done
56
57 bind lsn group LSN-GROUP-4 -applicationprofilename LSN-APPS-WELLKNOWN-
      PORTS-PROFILE-4
58
```



```
Done
<!--NeedCopy-->
```

複数の **LSN** グループ間で **NAT** リソースを共有する **LSN** 設定を作成します。この例では、**LSN** プール **LSN-POOL-5** は **LSN** グループ **LSN-GROUP-5** および **LSN-GROUP-6** と共有されています。

```
add lsn client LSN-CLIENT-5

Done

bind lsn client LSN-CLIENT-5 -network 192.0.15.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-5

Done

bind lsn pool LSN-POOL-5 203.0.113.12-203.0.113.14

Done

add lsn group LSN-GROUP-5 -clientname LSN-CLIENT-5

Done

bind lsn group LSN-GROUP-5 -poolname LSN-POOL-5

Done

add lsn client LSN-CLIENT-6

Done

bind lsn client LSN-CLIENT-6 -network 192.0.16.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-6

Done

bind lsn pool LSN-POOL-6 203.0.113.15-203.0.113.18

Done

add lsn group LSN-GROUP-6 -clientname LSN-CLIENT-6

Done

bind lsn group LSN-GROUP-6 -poolname LSN-POOL-6

Done
```



```
48
49 bind lsn group LSN-GROUP-6 -poolname LSN-POOL-5
50
51 Done
52 <!--NeedCopy-->
```

Deterministic NAT リソース割り当てを使用して **LSN** 設定を作成します。

```
1  add lsn client LSN-CLIENT-7
2
3  Done
4
5  bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask 255.255.255.0
6
7  Done
8
9  add lsn pool LSN-POOL-7 -nattype DETERMINISTIC
10
11 Done
12
13 bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23
14
15 Done
16
17 add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype
      DETERMINISTIC -portblocksize 1024
18
19 Done
20
21 bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7
22
23 Done
24 <!--NeedCopy-->
```

同じネットワークアドレスを持つ複数の加入者ネットワークで、各ネットワークが異なるトラフィックドメインに属する **LSN** 設定を作成します。また、**HTTP** プロトコル **(**ポート **80)** に関連するアウトバウンドトラフィックを制限して、特定のトラフィックドメイン **(td 5)** 経由で送信するようにします。

```
1  add lsn client LSN-CLIENT-8
2
3  Done
4
5  bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
       -td 1
6
7  Done
8
9  bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
       -td 2
10
11 Done
12
13 bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
```



```
      -td 3
14
15 Done
16
17 add lsn pool LSN-POOL-8
18
19 Done
20
21 bind lsn pool LSN-POOL-8 203.0.113.80-203.0.113.86
22
23 Done
24
25 add lsn group LSN-GROUP-8 -clientname LSN-CLIENT-8
26
27 Done
28
29 bind lsn group LSN-GROUP-8 -poolname LSN-POOL-8
30
31 Done
32
33 add lsn appsprofile LSN-APPS-HTTP-PROFILE-8 TCP -td 5
34
35 Done
36
37 bind lsn appsprofile LSN-APPS-HTTP-PROFILE-8 80
38
39 Done
40
41 bind lsn group LSN-GROUP-8 -applicationprofilename LSN-APPS-HTTP-
      PROFILE-8
42
43 Done
44 <!--NeedCopy-->
```

特定のプロトコル **(TCP)** のアウトバウンドトラフィックを制限し、特定のトラフィックドメイン **(td 5)** を介して送信する **LSN** 設定を作成します。エンドポイントに依存しないフィルタリングにより、このプロトコル **(TCP)** に関連するインバウンドトラフィックを任意のトラフィックドメインで受信できます。

```
1  add lsn client LSN-CLIENT-9
2
3  Done
4
5  bind lsn client LSN-CLIENT-9 -network 192.0.9.0 -netmask 255.255.255.0
      -td 1
6
7  Done
8
9  add lsn pool LSN-POOL-9
10
11 Done
12
13 bind lsn pool LSN-POOL-9 203.0.113.90
14
```



```
Done

add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

Done

bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done

add lsn appsprofile LSN-APPS-PROFILE-9 TCP -filtering ENDPOINT-
    INDEPENDENT -td 5

Done

bind lsn group LSN-GROUP-9 -approfile LSN-APPS-PROFILE-9

Done
<!--NeedCopy-->
```

アウトバウンド **HTTP (**ポート **80)** トラフィックを制限して、特定のトラフィックドメイン **(td 10)** 経由で送信する **LSN** 設定を作成します。アドレス依存フィルタリングでは、指定されたトラフィックドメイン **(td 10)** でこのプロトコル **(HTTP)** に関連するインバウンドトラフィックを受信します。

```
add lsn client LSN-CLIENT-10

Done

bind lsn client LSN-CLIENT-10 -network 192.0.10.0 -netmask
    255.255.255.0 -td 1

Done

add lsn pool LSN-POOL-10

Done

bind lsn pool LSN-POOL-10 203.0.113.100

Done

add lsn group LSN-GROUP-10 -clientname LSN-CLIENT-10

Done

bind lsn group LSN-GROUP-10 -poolname LSN-POOL-10

Done

add lsn appsprofile LSN-APPS-PROFILE-10 TCP -mapping ENDPOINT -
    INDEPENDENT -filtering ADDRESS-DEPENDENT -td 10

Done
```



```
28
29 bind lsn appsprofile LSN-APPS-PROFILE-10 80
30
31 Done
32
33 bind lsn group LSN-GROUP-10 -approfile LSN-APPS-PROFILE-10
34
35 Done
36 <!--NeedCopy-->
```

静的 **LSN** マップの構成

August 15, 2023

NetScaler アプライアンスは、サブスクライバーの IP アドレス: ポートと NAT IP アドレス: ポート間の 1 対 1 の LSN マッピングを手動で作成することをサポートしています。スタティック LSN マッピングは、NAT IP: Port に対して開始された接続がサブスクライバの IP アドレス: ポートに確実にマッピングされるようにする場合に便利です。たとえば、内部ネットワークにある Web サーバー。

コマンドラインインターフェイスを使用して静的 **LSN** マッピングを作成するには

コマンドプロンプトで入力します。

```
1 add lsn static <name> <transportprotocol> <subscrIP> <subscrPort> [-td
     <positive_integer>] [<natIP> [<natPort>]] [-destIP <ip_addr> [-dsttd
      <positive_integer>]]
2 -  show lsn static
3 <!--NeedCopy-->
```

構成ユーティリティを使用して静的 **LSN** マッピングを作成するには

[システム] > [大規模 NAT] > [スタティック] に移動し、新しいスタティックマッピングを追加します。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

add lsn static name

LSN スタティックマッピングエントリの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。LSN グループの作成後は変更できません。次の要件は CLI にのみ適用されます。

名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲んでください (たとえば、「lsn static1”や‘lsn static1’)。これは必須の議論です。最大長：127

トランスポートプロトコル

LSN マッピングエントリのプロトコル。これは必須の議論です。設定可能な値:TCP、UDP、ICMP

購読

LSN マッピングエントリの LSN サブスクライバの IPv4 アドレス。これは必須の議論です。

定期購読

LSN マッピングエントリの LSN サブスクライバのポート。これは必須の議論です。最大値:65535

td

加入者が属するトラフィックドメインの ID。ID を指定しない場合、加入者はデフォルトのトラフィックドメインの一部であると見なされます。デフォルト値:0、最小値:0、最大値:4094

NaTiP

このマッピングエントリの NAT IP アドレスとして使用される IPv4 アドレスは、NetScaler アプライアンスに LSN タイプとしてすでに存在しています。

NAT ポート

この LSN マッピングエントリの NAT ポート。

デスティップ

LSN マッピングエントリの宛先 IP アドレス。

dsttd

NetScaler アプライアンスからこの LSN マッピングエントリの宛先 IP アドレスにアクセスできるトラフィックドメインの ID。ID を指定しない場合、宛先 IP アドレスには ID が 0 のデフォルトのトラフィックドメインを介して到達可能であると見なされます。デフォルト値:0、最小値:0、最大値:4094

ワイルドカードポートスタティックマップ

スタティックマッピングエントリは通常、加入者の IP アドレス: ポートと NAT IP アドレス: ポート間の 1 対 1 の LSN マッピングです。1 対 1 の静的 LSN マッピングエントリでは、加入者の 1 つのポートだけがインターネットに公開されます。

状況によっては、加入者のすべてのポート（64K）をインターネットに公開する必要がある場合があります（たとえば、内部ネットワークでホストされ、各ポートで異なるサービスを実行しているサーバ)。これらの内部サービスにインターネット経由でアクセスできるようにするには、サーバーのすべてのポートをインターネットに公開する必要があります。

この要件を満たす方法の 1 つは、ポートごとに 1 つずつ、64K の 1 対 1 のスタティックマッピングエントリを追加することです。64K エントリの作成は非常に面倒で大きな作業です。また、このような多数の構成エントリは、NetScaler アプライアンスのパフォーマンスの問題につながる可能性があります。

もう 1 つの簡単な方法は、スタティックマッピングエントリでワイルドカードポートを使用することです。加入者のすべてのポートをインターネットに公開するには、NAT ポートとサブスクライバポートのパラメータをワイルドカード文字（*）に設定し、プロトコルパラメータを ALL に設定したスタティックマッピングエントリを 1 つ作成する必要があります。ワイルドカードスタティックマッピングエントリと一致するサブスクライバのインバウンド接続またはアウトバウンド接続の場合、NAT 操作後もサブスクライバのポートは変更されません。

加入者が開始したインターネット接続がワイルドカード静的マッピングエントリと一致する場合、NetScaler アプライアンスは接続を開始した加入者ポートと同じ番号の NAT ポートを割り当てます。同様に、インターネットホストは、加入者のポートと同じ番号の NAT ポートに接続することによって加入者のポートに接続されます。

IPv4 サブスクライバーのすべてのポートにアクセスできるようにする **NetScaler** アプライアンスの構成

IPv4 サブスクライバーのすべてのポートにアクセスできるように NetScaler アプライアンスを構成するには、次の必須パラメーター設定を使用してワイルドカードスタティックマップを作成します。

- プロトコル = すべて
- サブスクライバポート = *
- NAT ポート = *

ワイルドカードスタティックマップでは、1 対 1 のスタティックマップとは異なり、NAT IP パラメータの設定は必須です。また、ワイルドカードスタティックマップに割り当てられた NAT IP アドレスは、他のサブスクライバには使用できません。

コマンドラインインターフェイスを使用してワイルドカードスタティックマップを作成するには

コマンドプロンプトで入力します。

```
add lsn static <name> ALL <subscrIP> *  <natIP> * [-td <
    positive_integer>] [-destIP <ip_addr> [-dsttd <positive_integer>]]

show lsn static
```



```
<!--NeedCopy-->
```

構成例

次のワイルドカードスタティックマップの設定例では、IP アドレスが 192.0.2.10 のサブスクライバのすべてのポートに NAT IP 203.0.11.33 を介してアクセスできるようになっています。

設定例:

```
add lsn static NAT44-WILDCARD-STATIC-1 ALL  192.0.2.10 * 203.0.113.33 *

Done
<!--NeedCopy-->
```

アプリケーションレイヤーゲートウェイの構成

August 15, 2023

一部のアプリケーション層プロトコルでは、IP アドレスとプロトコルポート番号もパケットのペイロードで通信されます。プロトコルのアプリケーション層ゲートウェイは、パケットのペイロードを解析し、プロトコルが LSN 上で引き続き機能するように必要な変更を行います。

NetScaler アプライアンスは、次のプロトコルの ALG をサポートしています。

- FTP
- ICMP
- TFTP
- PPTP
- SIP
- RTSP

FTP、**ICMP**、および **TFTP** プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

LSN 設定の LSN グループの FTP オプションを有効または無効にすることで、LSN 設定の FTP プロトコルの ALG を有効または無効にできます。

ICMP プロトコルの ALG はデフォルトで有効になっており、無効にする設定はありません。

TFTP プロトコルの ALG はデフォルトで無効になっています。エンドポイントに依存しないマッピング、エンドポイントに依存しないフィルタリング、宛先ポートを 69（TFTP 用の既知のポート）に設定した UDP LSN アプリケーションプロファイルを LSN グループにバインドすると、TFTP ALG は LSN 設定で自動的に有効になります。

FTP ALG の **LSN** 設定の例:

次の LSN 設定の例では、IP アドレスが 192.0.2.30 ～192.0.2.100 の範囲にあるサブスクライバに対して FTP ALG が有効になっています。

```
add ns acl LSN-ACL-1 ALLOW -srcIP 192.0.2.30-192.0.2.100

Done

apply acls

Done

add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1  -aclname LSN-ACL

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.10

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -FTP ENABLED

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done
<!--NeedCopy-->
```

TFTP ALG の **LSN** 設定の例**:**

次の LSN 設定例では、TFTP プロトコル（UDP ポート 69）のエンドポイント独立マッピングとエンドポイント独立フィルタリングが有効になっています。NetScaler アプライアンスは、この LSN 構成に対して TFTP ALG を自動的に有効にします。

```
add lsn client LSN-CLIENT-2

Done

bind lsn client LSN-CLIENT-2 -network 198.51.100.0 -netmask
```



```
    255.255.255.0
6
7 Done
8
9 add lsn pool LSN-POOL-2
10
11 Done
12
13 bind lsn pool LSN-POOL-2 203.0.113.10-203.0.113.11
14
15 Done
16
17 add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2
18
19 Done
20
21 bind lsn group LSN-GROUP-2 -poolname pool1 LSN-POOL-2
22
23 Done
24
25 add lsn appsprofile LSNAPPSPROFILE-TFTP-2 UDP -mapping ENDPOINT-
    INDEPENDENT  - filtering  ENDPOINT-INDEPENDENT
26
27 Done
28
29 bind lsn appsprofile LSNAPPSPROFILE-TFTP-2 69
30
31 Done
32
33 bind lsn group LSN-GROUP-1 -applicationprofilename LSNAPPSPROFILE-TFTP
    -2
34
35 Done
36 <!--NeedCopy-->
```

PPTP プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

NetScaler アプライアンスは、ポイントツーポイントトンネリングプロトコル（PPTP）用のアプリケーション層ゲートウェイ（ALG）をサポートしています。

PPTP は、TCP/IP ベースのデータネットワーク上にトンネルを作成することにより、リモートクライアントから企業サーバーへのデータの安全な転送を可能にするネットワークプロトコルです。PPTP は PPP パケットを IP パケットにカプセル化し、インターネット経由で送信します。PPTP は、通信する PPTP ネットワークサーバ（PNS）と PPTP アクセスコンセントレータ（PAC）のペアごとにトンネルを確立します。トンネルを設定すると、拡張汎用ルーティングカプセル化（GRE）を使用して PPP パケットを交換します。GRE ヘッダーのコール ID は、特定の PPP パケットが属するセッションを示します。

NetScaler アプライアンスは、デフォルトの TCP ポート 1723 に到着した PPTP パケットを認識します。アプライアンスは PPTP 制御パケットを解析し、コール ID を変換し、NAT IP アドレスを割り当てます。クライアントとサーバー間の双方向データ通信では、NetScaler アプライアンスはサーバーの呼び出し ID に基づいて LSN セッションエントリを作成し、クライアントの呼び出し ID に基づいて LSN セッションを作成します。次に、アプライアンスは GRE データパケットを解析し、2 つの LSN セッションエントリに基づいてコール ID を変換します。

PPTP プロトコルの場合、NetScaler アプライアンスにはアイドル状態の PPTP LSN セッションのタイムアウト設定も含まれています。PPTP LSN セッションがタイムアウト設定を超える時間アイドル状態になると、NetScaler アプライアンスはそのセッションを削除します。

制限事項:

NetScaler アプライアンスでの PPTP ALG の制限は次のとおりです。

- PPTP ALG はヘアピン LSN フローではサポートされていません。
- PPTP ALG は、どの RNAT 構成でも動作するようにサポートされていません。
- PPTP ALG は NetScaler クラスターではサポートされていません。

PPTP ALG の設定

NetScaler アプライアンスでの PPTP ALG の設定は、次のタスクで構成されます。

- LSN 設定を作成し、その上で PPTP ALG を有効にします。LSN 構成では、LSN グループに PPTP ALG 設定が含まれます。LSN 構成の作成手順については、「LSN の設定手順」を参照してください。
- （任意）アイドル状態の PPTP LSN セッションのグローバルタイムアウトを設定します。

CLI を使用して LSN 構成の PPTP ALG を有効にするには

コマンドプロンプトで入力します。

```
add lsn group <groupname> -clientname <string> [-pptp ( ENABLED |
   DISABLED )]

show lsn group
<!--NeedCopy-->
```

CLI を使用してアイドル状態の PPTP LSN セッションのグローバルタイムアウトを設定するには

コマンドプロンプトで入力します。

```
set appAlgParam -pptpGreIdleTimeout <positive_integer>

show appAlgParam
<!--NeedCopy-->
```


例:

次の LSN 設定例では、192.0.2.0/24 ネットワークのサブスクライバに対して PPTP ALG が有効になっています。

また、アイドル状態の PPTP LSN セッションのタイムアウトは 200 秒に設定されています。

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -pptp ENABLED

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done

set appAlgParam -pptpGreIdleTimeout 200

Done
<!--NeedCopy-->
```

SIP プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

セッション開始プロトコル（SIP）で大規模 NAT（LSN）を使用するのは複雑です。SIP メッセージには SIP ヘッダーと SIP 本文に IP アドレスが含まれるためです。SIP で LSN を使用する場合、SIP ヘッダーには発信者と受信者に関する情報が含まれ、デバイスはこの情報を変換して外部ネットワークから隠します。SIP 本体には、メディア送信用の IP アドレスとポート番号を含むセッション記述プロトコル（SDP）情報が含まれています。

SIP ALG は次の RFC に準拠しています。

- RFC 3261
- RFC 3581

- RFC 4566
- RFC 4475

> 注
>
> SIP ALG は、NetScaler スタンドアロンアロンアプライアンス、NetScaler 高可用性セットアップ、および NetScaler クラスターセットアップでサポートされています。

SIP ALG の仕組み

IP アドレス変換の実行方法は、メッセージのタイプと方向によって異なります。メッセージには次のいずれかを使用できます。

- インバウンドリクエスト
- アウトバウンドレスポンス
- アウトバウンドリクエスト
- インバウンドレスポンス

送信メッセージの場合、SIP クライアントのプライベート IP アドレスとポート番号は、LSN 構成時に指定された NetScaler 所有のパブリック IP アドレスとポート番号（*LSN プール IP* アドレスとポート番号と呼ばれる）に置き換えられます。受信メッセージでは、LSN プール IP アドレスとポート番号がクライアントのプライベートアドレスに置き換えられます。メッセージにパブリック IP アドレスが含まれている場合、NetScaler SIP ALG はそれらを保持します。また、次の場所にピンホールが作成されます。

- LSN プールの IP アドレスとポートはプライベートクライアントに代わって行われます。これにより、パブリックネットワークからこの IP アドレスとポートに到着したメッセージは SIP メッセージとして扱われます。
- パブリッククライアントに代わってパブリック IP アドレスとポート。これにより、プライベートネットワークからこの IP アドレスとポートに到着したメッセージは SIP メッセージとして扱われます。

SIP メッセージがネットワーク経由で送信されると、SIP アプリケーション層ゲートウェイ (ALG) はメッセージから情報を収集し、次のヘッダーの IP アドレスを LSN プール IP アドレスに変換します。

- 経由
- 連絡
- Route
- レコードルート

次の SIP 要求メッセージの例では、LSN がヘッダーフィールドの IP アドレスを置き換えて外部ネットワークから隠しています。

```
1 INVITE adam@10.102.185.156 SIP/2.0 Via: SIP/2.0/UDP 192.170.1.161:62914
      From: eve@10.120.210.3 To: adam@10.102.185.156 Call-ID: a12abcde@10
     .120.210.3 Contact: adam@10.102.185.156 Route: <sip:netscreen@10
     .150.20.3:5060> Record-Route: <sip:netscreen@10.150.20.3:5060>
2 <!--NeedCopy-->
```


SDP 情報を含むメッセージが到着すると、SIP ALG はメッセージから情報を収集し、次のフィールドの IP アドレスを LSN プール IP アドレスとポート番号に変換します。

- c= (接続情報)

  このフィールドは、セッションレベルまたはメディアレベルで表示できます。次の形式で表示されます。

  c=<network-type><address-type><connection-address>

  宛先 IP アドレスがユニキャスト IP アドレスの場合、SIP ALG は m= フィールドで指定された IP アドレスとポート番号を使用してピンホールを作成します。

- m= (メディアアナウンス)

  このフィールドはメディアレベルで表示され、メディアの説明が含まれます。次の形式で表示されます。

  m=<media><port><transport><fmt list>

- a= (information about the media field)

  このフィールドは、セッションレベルまたはメディアレベルで、次の形式で表示できます。

  a=<attribute>

  a=<attribute>:<value>

次の SDP セクションのサンプルを抜粋すると、リソース割り当てのために変換されるフィールドが示されています。

o=user 2344234 55234434 IN IP4 10.150.20.3

C=IN IP4 10.150.20.3

m=audio 43249 RTP/AVP 0

次の表は、SIP ペイロードがどのように変換されるかを示しています。

インバウンドリクエスト (パブリックからプライベートへ)	変更後:	なし
		From
		Call-ID
		経由
		Request-URI
		連絡先
		レコードルート
		ルート

アウトバウンドレスポンス（プライベートからパブリックへ）	変更後：	なし From Call-ID 経由 Request-URI 連絡先 レコードルート ルート
アウトバウンドリクエスト (プライベートからパブリックへ)	変更後：	なし From Call-ID 経由 Request-URI 連絡先 レコードルート ルート
インバウンドレスポンス (公開から非公開へ)	変更後：	なし From Call-ID 経由 Request-URI 連絡先 レコードルート ルート

SIP ALG の制限事項

SIP ALG には次の制限があります。

- SDP ペイロードのみがサポートされます。
- 以下はサポートされていません:
  - マルチキャスト IP アドレス
  - 暗号化された SDP
  - SIP TLS
  - FQDN トランスレーション
  - SIP レイヤー認証
  - TD/partitioning
  - マルチパートボディ
  - IPv6 ネットワーク経由の SIP メッセージ
  - ラインフォールディング

テスト済みの **SIP** クライアントとプロキシサーバー

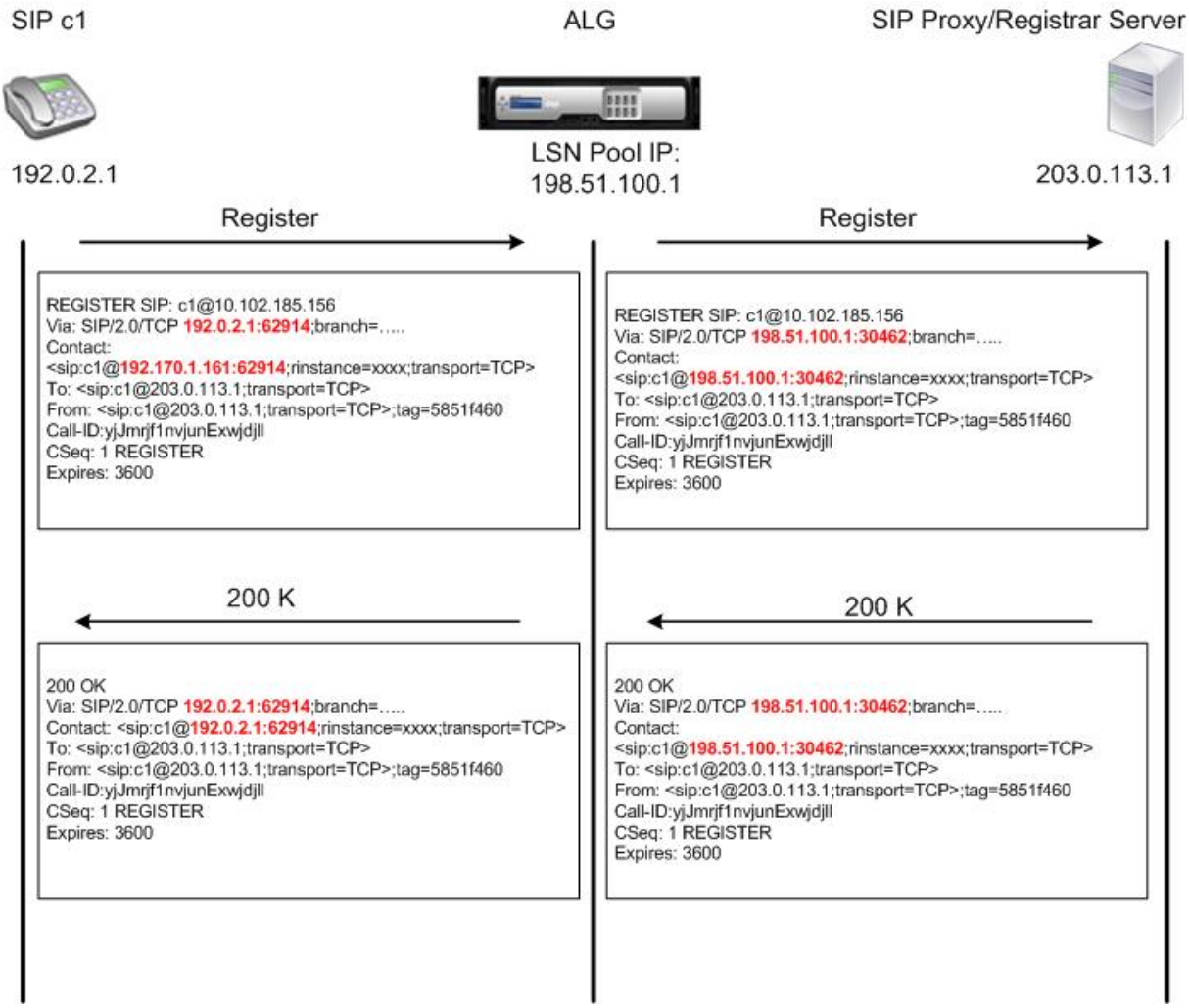
次の SIP クライアントとプロキシサーバーは SIP ALG でテストされています。

- **SIP** クライアント: エックスライト、ゾイパー、エキガ。Avaya
- プロキシサーバー**:OpenSIPS**

LSN SIP シナリオ**:** プライベートネットワーク **(**パブリックネットワーク**)** 外の **SIP** プロキシ

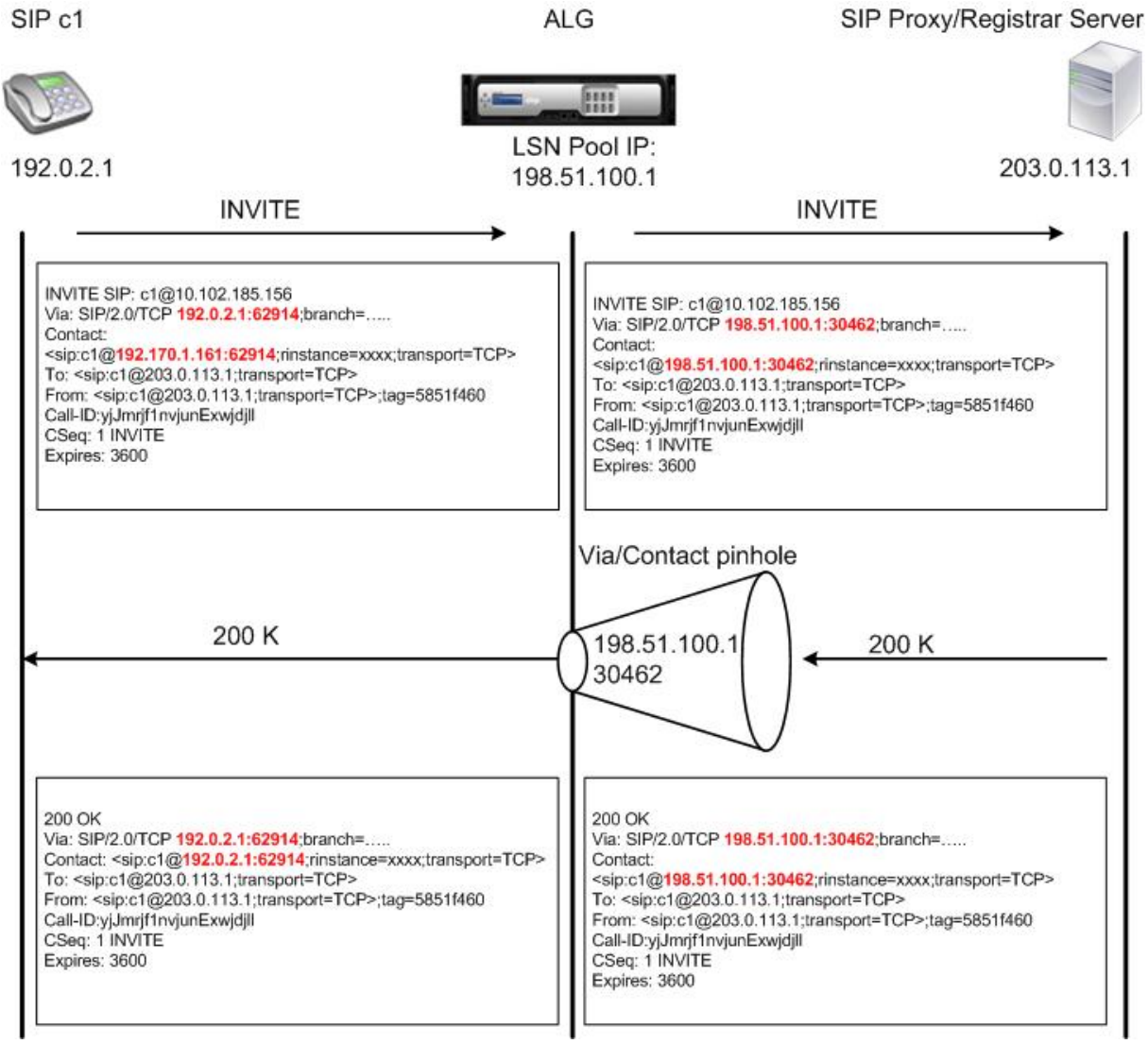
SIP クライアント登録

一般的な SIP コールでは、SIP クライアントは REGISTER 要求を作成して SIP レジストラに送信することによって SIP レジストラに登録する必要があります。NetScaler アプライアンスの SIP ALG は要求をインターセプトし、要求内の IP アドレスとポート番号を LSN 構成で提供された LSN プールの IP アドレスとポート番号に置き換え、要求を SIP レジストラに転送します。次に、SIP ALG は NetScaler 構成のピンホールを開き、SIP クライアントと SIP レジストラ間のさらなる SIP 通信を可能にします。SIP レジストラは、LSN プール IP アドレスとポート番号を使用して 200 OK 応答を SIP クライアントに送信します。NetScaler アプライアンスはこの応答をピンホールでキャプチャし、SIP ALG が SIP ヘッダーを置き換え、元の連絡先、経路、ルート、およびレコードルートの SIP フィールドをメッセージに戻します。次に、SIP ALG はメッセージを SIP クライアントに転送します。次の図は、SIP ALG が SIP コール登録フローで LSN を使用する方法を示しています。



発信コール

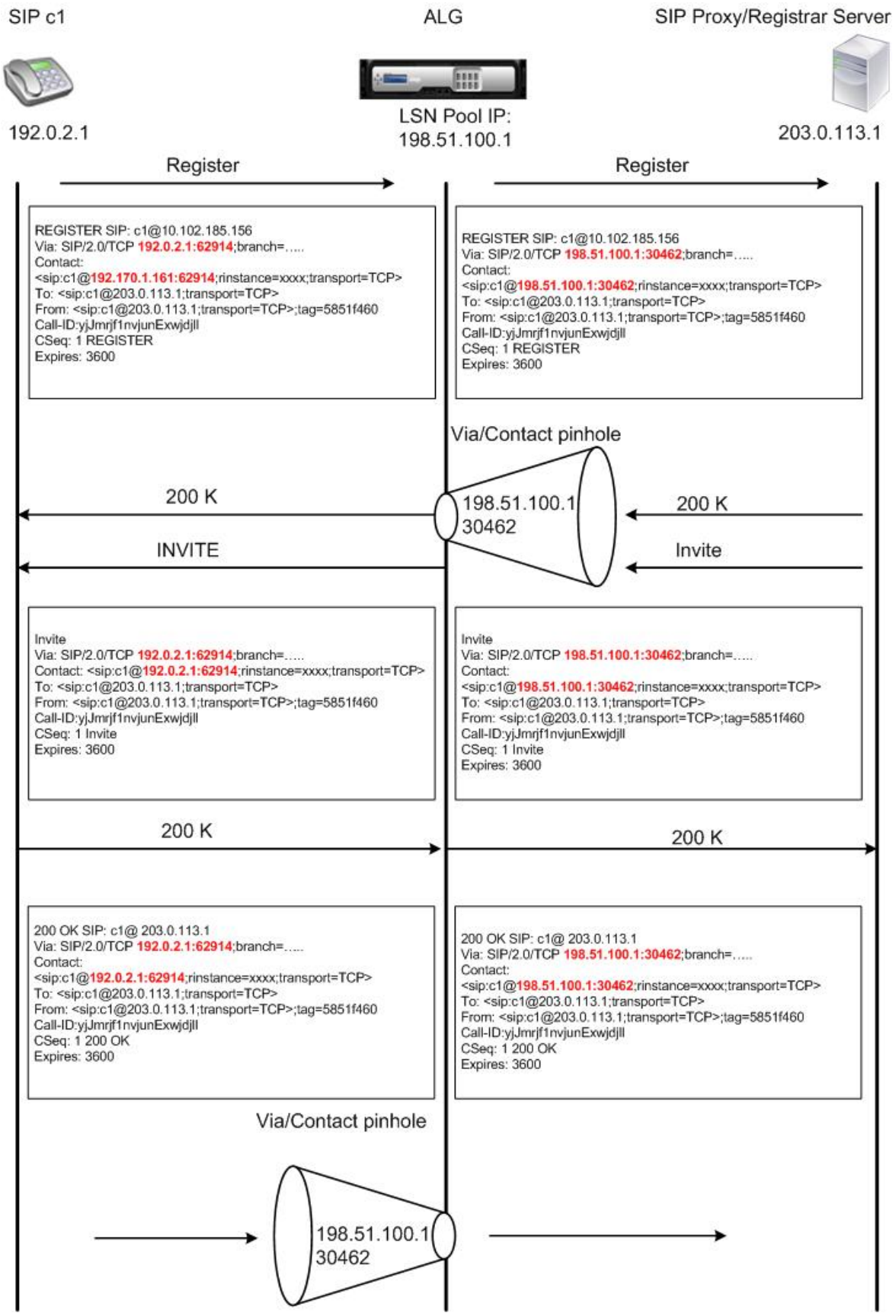
SIP コールは、内部ネットワークから外部ネットワークに送信される SIP INVITE メッセージで開始されます。SIP ALG は、Via、Contact、Route、Record-Route の SIP ヘッダーフィールドの IP アドレスとポート番号で NAT を実行し、それらを LSN プールの IP アドレスとポート番号に置き換えます。LSN は、後続の SIP メッセージ用に SIP コールでこれらのマッピングを保存します。次に、SIP ALG は NetScaler 構成に個別のピンホールを開き、SDP ヘッダーと SIP ヘッダーで指定されている動的に割り当てられたポート上の NetScaler アプライアンスを介して SIP とメディアを経由できるようにします。200 OK のメッセージが NetScaler に到着すると、作成されたピンホールの 1 つによってキャプチャされます。SIP ALG は SIP ヘッダーを置き換え、元の Contact、Via、Route、Record-Route の SIP フィールドを復元してから、メッセージを内部 SIP クライアントに転送します。



着信コール

SIP 着信コールは、外部クライアントから内部ネットワークへの SIP INVITE メッセージで開始されます。SIP レジストラは、内部 SIP クライアントが SIP レジストラに登録されたときに作成されたピンホールを使用して、INVITE メッセージを内部ネットワークの SIP クライアントに転送します。

SIP ALG は、Via、Contact、Route、Record-Route の SIP ヘッダーフィールドの LSN IP アドレスとポート番号で NAT を実行し、それらを内部 SIP クライアントの IP アドレスとポート番号に変換して、要求を SIP クライアントに転送します。内部 SIP クライアントから送信された 200 OK 応答メッセージが NetScaler アプライアンスに届くと、SIP ALG は Via、Contact、Route、および Record-Route の SIP ヘッダーフィールドの IP アドレスとポート番号に対して NAT を実行し、それらを LSN プールの IP アドレスとポート番号に変換して、応答メッセージを SIP レジストラに転送し、さらに SIP 送信方向にピンホールを開きます。P コミュニケーション。



通話終了

BYE メッセージはコールを終了します。デバイスは BYE メッセージを受信すると、他のメッセージと同様にメッセージ内のヘッダーフィールドを変換します。ただし、BYE メッセージは 200 OK で受信側が確認する必要があるため、ALG は 200 OK の送信時間を考慮して、コールティアダウンを 15 秒間遅延させます。

同じネットワーク内のクライアント間のコール

同じネットワーク内のクライアント A とクライアント B の両方がコールを開始すると、SIP メッセージは外部ネットワークの SIP プロキシを介してルーティングされます。SIP ALG は、クライアント A からの INVITE を通常の発信コールとして処理します。クライアント B は同じネットワークにあるため、SIP プロキシは INIVITE を NetScaler アプライアンスに送り返します。SIP ALG は INIVITE メッセージを調べ、そのメッセージにクライアント A の NAT IP アドレスが含まれていることを確認し、クライアント B にメッセージを送信する前に、そのアドレスをクライアント A のプライベート IP アドレスに置き換えます。クライアント間で呼び出しが確立されると、NetScaler はクライアント間のメディア転送に関与しません。

その他の **LSN SIP** シナリオ**:** プライベートネットワーク内の **SIP** プロキシ

プライベートネットワーク内で SIP プロキシサーバーをホストする場合、Citrix では次のいずれかを行うことをお勧めします。

- プライベート SIP プロキシのスタティック LSN マッピングを設定します。詳細については、スタティック LSN マップの設定を参照してください。NAT ポートが SIP ALG プロファイルで設定されているポートと同じであることを確認します。
- 非武装地帯 (DMZ) 内に SIP プロキシサーバを設定します。

図 1: SIP コール登録

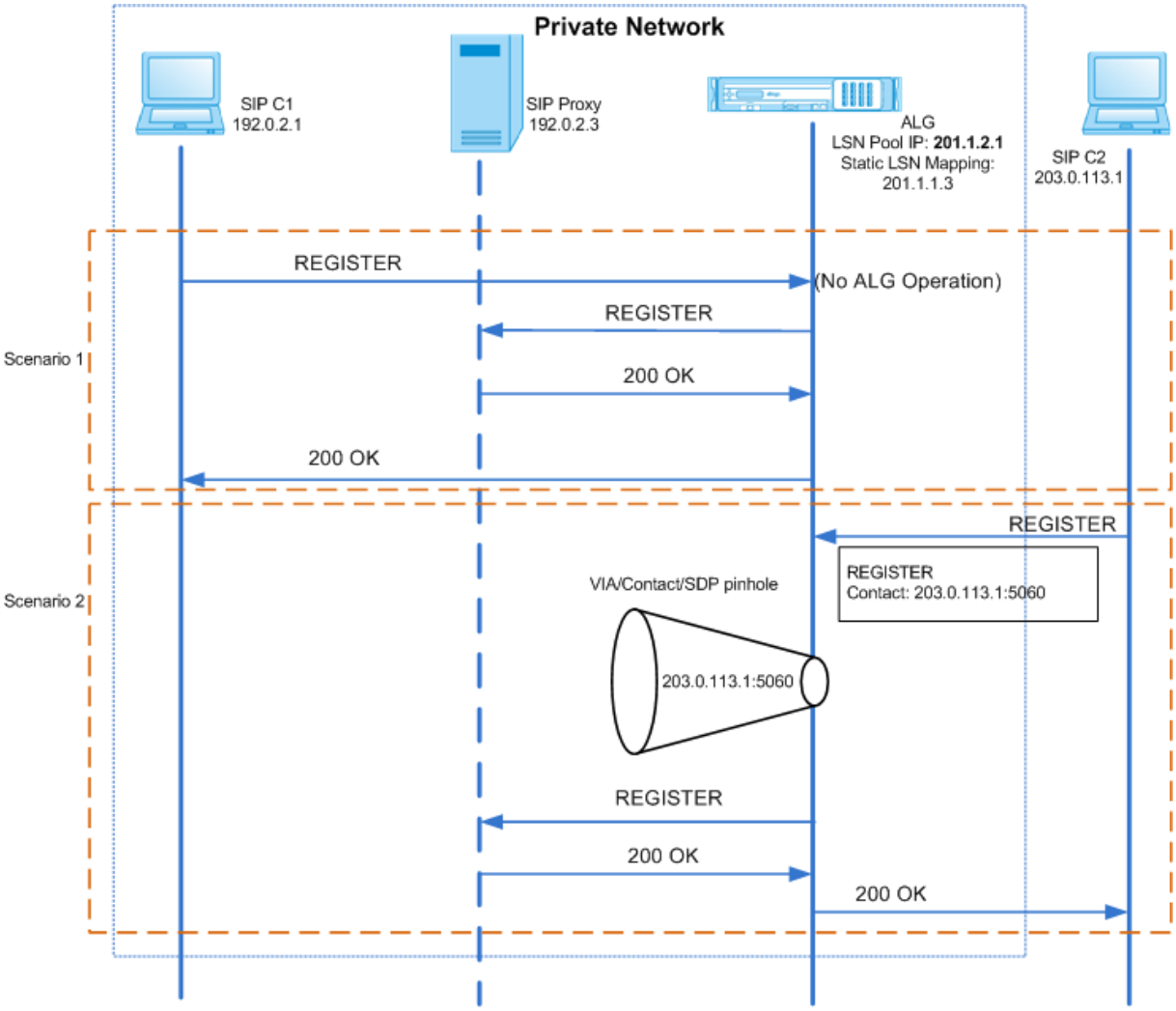


図 2: SIP 着信コールフロー

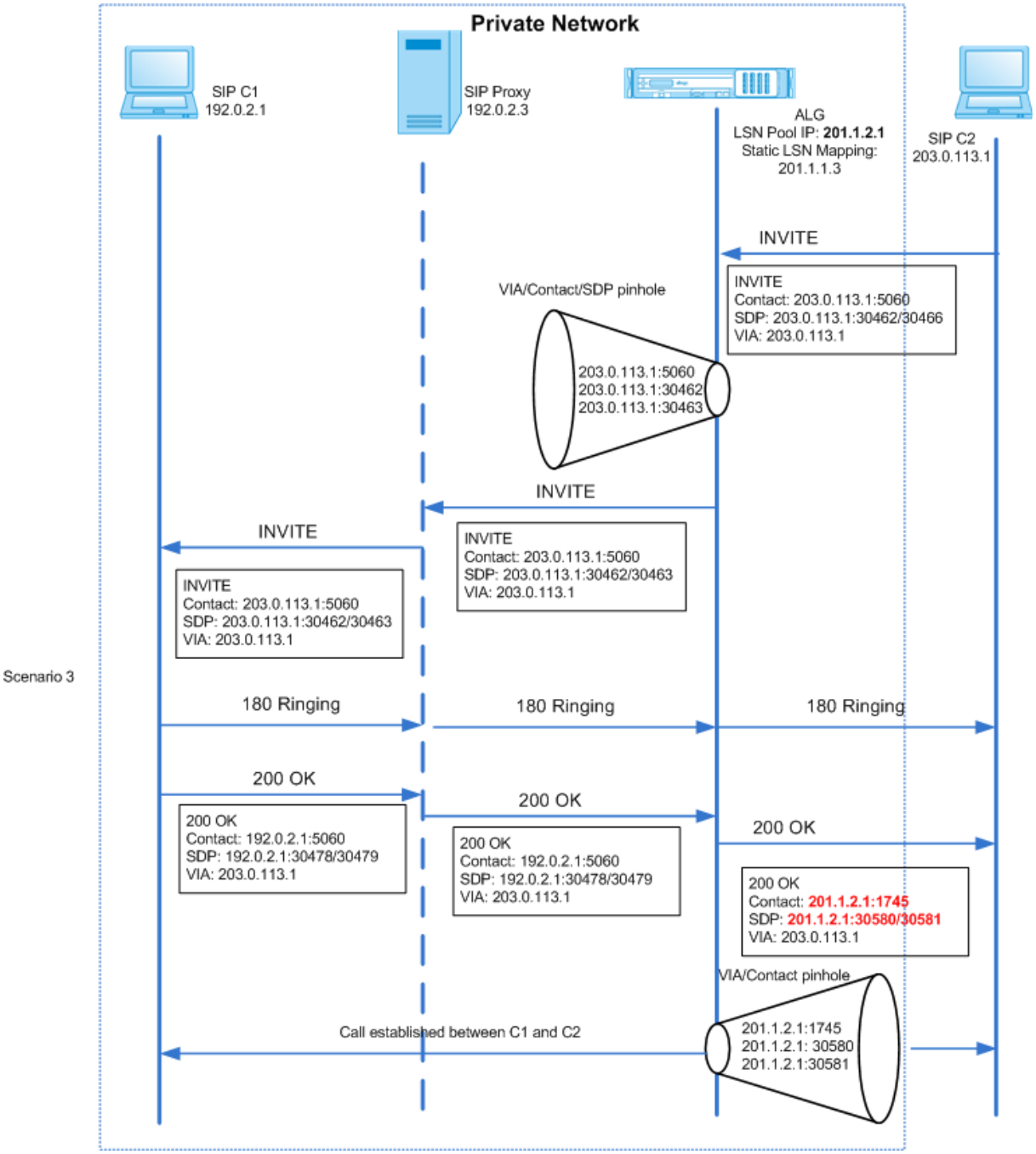


図 1 と図 2 は、次のシナリオを示しています。

- シナリオ 1: プライベートネットワークの SIP クライアントは、同じネットワーク内の SIP プロキシサーバに登録されます。SIP クライアントと SIP プロキシサーバは同じネットワークにあるため、ALG 操作は実行されません。
- シナリオ 2: パブリックネットワークの SIP クライアントは、プライベートネットワークの SIP プロキシサーバに登録されます。パブリック SIP クライアントからの REGISTER メッセージは、アプライアンス上で構成された静的 LSN マッピングを使用して NetScaler アプライアンスに送信され、アプライアンスはさらに SIP

操作を行うためのピンホールを作成します。

- シナリオ 3: SIP 着信コールフロー SIP 着信コールは、外部ネットワークから内部ネットワークへの SIP INVITE メッセージで開始されます。NetScaler アプライアンスは、NetScaler アプライアンスで構成された静的 LSN マップを介して、外部ネットワークにある SIP クライアント C2 から INVITE メッセージを受信します。
  アプライアンスはピンホールを作成し、INVITE メッセージを SIP プロキシに転送します。次に、SIP プロキシは INVITE メッセージを内部ネットワークの SIP クライアント C1 に転送します。次に、SIP クライアント C1 は 180 件と 200 件の OK メッセージを SIP プロキシに送信し、SIP プロキシはそのメッセージを NetScaler アプライアンスを介して SIP クライアント C2 に転送します。
  内部 SIP クライアント C1 から送信された 200 OK 応答メッセージが NetScaler に到着すると、SIP ALG は Via、Contact、Route、Record-Route の SIP ヘッダーフィールドと SDP フィールドの IP アドレスとポート番号に NAT を実行し、それらを LSN プールの IP アドレスとポート番号に置き換えます。次に、SIP ALG は応答メッセージを SIP クライアント C2 に転送し、送信方向にピンホールを開いてさらにの SIP 通信を行います。

監査ログのサポート

LSN 監査ログ構成で ALG を有効にすると、LSN ログの一部として ALG 情報をログに記録できます。LSN ロギングの詳細については、「LSN のロギングとモニタリング」を参照してください。LSN ログの ALG エントリのログメッセージは、次の情報で構成されます。

- タイムスタンプ
- SIP メッセージのタイプ (SIP 要求など)
- SIP クライアントの送信元 IP アドレスとポート
- SIP プロキシの宛先 IP アドレスとポート
- NAT IP アドレスとポート
- SIP メソッド
- シーケンス番号
- SIP クライアントが登録されているかどうか
- 発信者のユーザー名とドメイン
- 受信者のユーザー名とドメイン

監査ログのサンプル**:**

リクエスト**:**

```
1 07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG
    ALG_SIP_INFO_PACKET_EVENT 169 0 : Infomsg: "SIP request" - Group: g2
     - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE3Zjc3NjM. - Transport
    : TCP - Source_IP: 192.169.1.165 - Source_port: 57952 -
    Destination_IP: 10.102.185.156 - Destination_port: 5060 - Natted_IP:
     10.102.185.191 - Natted_port: 10313 - Method: REGISTER -
    Sequence_Number: 3060 - Register: YES - Content_Type: -
```



```
    Caller_user_name: 156_pvt_1 - Callee_user_name: 156_pvt_1 -
    Caller_domain_name: - Callee_domain_name: -
<!--NeedCopy-->
```

応答:

```
07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG
    ALG_SIP_INFO_PACKET_EVENT 170 0 : Infomsg: "SIP response" - Group:
    g2 - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE3Zjc3NjM. -
    Transport: TCP - Response_code 200 - Source_IP: 10.102.185.156 -
    Source_port: 5060 - Destination_IP: 192.169.1.165 - Destination_port
    : 57952 - Natted_IP: 10.102.185.191 - Natted_port: 10313 -
    Sequence_Number: 3060 - Content_Type: - Caller_user_name: 156_pvt_1
    - Callee_user_name: 156_pvt_1 - Caller_domain_name: -
    Callee_domain_name: -
<!--NeedCopy-->
```

SIP ALG の設定

LSN 設定の一部として SIP ALG を設定する必要があります。LSN の設定手順については、「LSN の設定手順」を参照してください。LSN の設定時には、次のことを確認してください。

- LSN アプリケーションプロファイルを追加する際に、次のパラメータを設定します。
  - IP Pooling = PAIRED
  - アドレスとポートのマッピング = エンドポイントに依存しない
  - フィルタリング = エンドポイントに依存しない

重要: SIP ALG が機能するためには、フルコーン NAT の設定が必須です。

例:

```
add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT
<!--NeedCopy-->
```

- SIP ALG プロファイルを作成し、送信元ポート範囲または宛先ポート範囲のいずれかを定義していることを確認します。

例:

```
add lsn sipalgprofile sipalgprofile_tcp -sipsrcportrange 1-65535 -
    sipdstportrange 5060 -openViaPinhole ENABLED -openRecordRoutePinhole
     ENABLED  - sipTransportProtocol TCP
<!--NeedCopy-->
```

- LSN グループの作成時に、SIP ALG = ENABLED に設定します。

例:


```
add lsn group g1 -clientname c1 -sipalg ENABLED
<!--NeedCopy-->
```

- SIP ALG プロファイルを LSN グループにバインドします。

SIP ALG 設定の例:

次の設定例は、単一の加入者ネットワーク、単一の LSN NAT IP アドレス、SIP ALG 固有の設定を使用して単純な LSN 設定を作成し、SIP ALG を設定する方法を示しています。

```
add lsn pool p1

Done

bind lsn pool p1 10.102.185.190

Done

add lsn client c1

Done

bind lsn client c1 -network 192.170.1.0 -netmask 255.255.255.0

Done

add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

add lsn appsprofile app_udp UDP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile app_tcp 1-65535

Done

bind lsn appsprofile app_udp 1-65535

Done

add lsn sipalgprofile sipalgprofile_tcp -sipdstportrange 5060 -
    openViaPinhole ENABLED -openRecordRoutePinhole ENABLED  -
    sipTransportProtocol TCP

Done

add lsn sipalgprofile sipalgprofile_udp -sipdstportrange 5060 -
    openViaPinhole ENABLED -openRecordRoutePinhole ENABLED -
```



```
      sipTransportProtocol UDP
38
39 Done
40
41 add lsn group g1 -clientname c1 -sipalg ENABLED
42
43 Done
44
45 bind lsn group g1 -poolname p1
46
47 Done
48
49 bind lsn group g1 -appsprofilename app_tcp
50
51 Done
52
53 bind lsn group g1 -appsprofilename app_udp
54
55 Done
56
57 bind lsn group g1 -sipalgprofilename sipalgprofile_tcp
58
59 Done
60
61 bind lsn group g1 -sipalgprofilename sipalgprofile_udp
62
63 Done
64 <!--NeedCopy-->
```

RTSP プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

リアルタイムストリーミングプロトコル (RTSP) は、リアルタイムのメディアデータを転送するためのアプリケーションレベルのプロトコルです。エンドポイント間のメディアセッションの確立と制御に使用される RTSP は、メディアクライアントとメディアサーバー間の制御チャネルプロトコルです。典型的な通信は、クライアントとストリーミングメディアサーバーの間です。

プライベートネットワークからパブリックネットワークにメディアをストリーミングするには、ネットワーク上で IP アドレスとポート番号を変換する必要があります。NetScaler の機能には、RTSP 用のアプリケーション層ゲートウェイ（ALG）が含まれています。これを大規模 NAT（LSN）と組み合わせて使用すると、メディアストリームを解析し、プロトコルがネットワーク上で引き続き機能するように必要な変更を加えることができます。

IP アドレス変換の実行方法は、メッセージのタイプと方向、およびクライアント/サーバ環境でサポートされるメディアの種類によって異なります。メッセージは次のように翻訳されます。

- アウトバウンドリクエスト—LSN プール IP アドレスと呼ばれる NetScaler 所有のパブリック IP アドレスへ

のプライベート IP アドレスです。

- インバウンド応答：プライベート IP アドレスへの LSN プール IP アドレス。
- インバウンドリクエスト-翻訳なし。
- アウトバウンド応答：LSN プール IP アドレスへのプライベート IP アドレス。

> 注
>
> RTSP ALG は、NetScaler スタンドアロンアロンアプライアンス、NetScaler 高可用性セットアップ、および NetScaler クラスターセットアップでサポートされています。

RTSP ALG の制限事項

RTSP ALG は以下をサポートしていません。

- マルチキャスト RTSP セッション
- UDP 経由の RTSP セッション
- TD/管理パーティショニング
- RSTP 認証
- HTTP トンネリング

RTSP と LSN のシナリオ

通常、RTSP SETUP 要求では、1 つのメディアストリームの転送方法を指定します。リクエストには、メディアストリーム URL とトランスポート指定子が含まれます。この指定子には通常、RTP データ（オーディオまたはビデオ）を受信するための 1 つのローカルポートと、RTCP データ（メタ情報）を受信するための別のローカルポートが含まれます。サーバーの応答は通常、選択したパラメーターを確認し、サーバーが選択したポートなど、不足している部分を埋めます。集約再生要求を送信する前に、SETUP コマンドを使用して各メディアストリームを設定する必要があります。

一般的な RTSP 通信では、パブリックネットワークのメディアクライアントがプライベートネットワークのメディアサーバーに SETUP 要求を送信します。RSTP ALG は要求をインターセプトし、メディアストリームでパブリック IP アドレスとポート番号を LSN プール IP アドレスと LSN ポート番号に置き換えます。

プライベートネットワークのメディアサーバーは、LSN プール IP アドレスと LSN ポート番号を使用して、パブリックネットワークのメディアクライアントに 200 OK 応答を送信します。NetScaler RTSP ALG は応答をインターセプトし、LSN プールの IP アドレスと LSN ポート番号をメディアクライアントのパブリック IP アドレスとポート番号に置き換えます。

RTSP ALG の設定

RTSP ALG を LSN 設定の一部として設定します。LSN の設定手順については、「LSN の設定手順」を参照してください。LSN の設定時には、次のことを確認してください。

- LSN プールを追加するときに、**NAT** タイプを **[** 決定的] または [ダイナミック] に設定します。
- LSN アプリケーションプロファイルを追加する際に、次のパラメータを設定します。
  - IP Pooling = PAIRED
  - アドレスとポートのマッピング = エンドポイントに依存しない
  - フィルタリング = エンドポイントに依存しない
- RTSP ALG プロファイルを作成し、その RTSP ALG プロファイルを LSN グループにバインドします

RTSP ALG 設定の例**:**

次の設定例は、単一のサブスクライバネットワーク、単一の LSN NAT IP アドレス、および RTSP ALG 設定を使用して単純な LSN 設定を作成する方法を示しています。

```
enable ns feature WL SP LB CS LSN

Done

add lsn pool pool1 -nattype DETERMINISTIC

Done

bind lsn pool pool1 10.102.218.246

Done

add lsn client client1

Done

bind lsn client client1 -network 200.200.200.11 -netmask 255.255.255.0

Done

add lsn appsprofile app1 TCP -ippooling PAIRED -mapping ENDPOINT-
   INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

add lsn appsprofile app2 UDP -ippooling PAIRED -mapping ENDPOINT-
   INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile app1 1-65535

Done

bind lsn appsprofile app2 1-65535

Done

```



```
add lsn rtspalgprofile rtspalgprofiledefault -rtspIdleTimeout 1000 -
    rtspportrange 554

Done

add lsn group group1 -clientname client1 -nattype DETERMINISTIC -
    portblocksize 512 -rtspalg ENABLED

Done

bind lsn group group1 -poolname pool1

Done

bind lsn group group1 -appsprofilename app1

Done

bind lsn group group1 -appsprofilename app2

Done

bind lsn group group1 -rtspalgprofilename rtspalgprofiledefault

Done
<!--NeedCopy-->
```

IPSec プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

2 つのネットワークデバイス（クライアントとサーバなど）間の通信が IPsec プロトコルを使用する場合、（UDP 経由の）IKE トラフィックはポートフィールドを使用しますが、カプセル化セキュリティペイロード（ESP）トラフィックはポートフィールドを使用しません。パス上の NAT デバイスが、同じ宛先の 2 つ以上のクライアントに同じ NAT IP アドレス（ただしポートは異なる）を割り当てると、その NAT デバイスはポート情報を含まない返される ESP トラフィックを区別できず、適切にルーティングできなくなります。そのため、IPSec ESP トラフィックは NAT デバイスで失敗します。

NAT トラバーサル（NAT-T）対応 IPsec エンドポイントは、IKE フェーズ 1 中に中間 NAT デバイスの存在を検出し、後続の IKE および ESP トラフィック（ESP を UDP にカプセル化）はすべて UDP ポート 4500 に切り替えます。ピア IPsec エンドポイントで NAT-T がサポートされていない場合、IPSec で保護された ESP トラフィックは UDP カプセル化なしで送信されます。そのため、IPSec ESP トラフィックは NAT デバイスで失敗します。

NetScaler アプライアンスは、大規模な NAT 構成用の IPSec アプリケーション層ゲートウェイ（ALG）機能をサポートしています。IPSec ALG は IPSec ESP トラフィックを処理し、セッション情報を保持します。これにより、IPsec エンドポイントが NAT-T（ESP トラフィックの UDP カプセル化）をサポートしていない場合でもトラフィッ

クに障害が発生することはありません。

IPsec ALG の仕組み

IPsec ALG は、クライアントとサーバの間の IKE トラフィックを監視し、クライアントとサーバ間の IKE フェーズ 2 メッセージ交換を常に 1 回だけ許可します。

特定のフローの双方向 ESP パケットを受信すると、IPSec ALG はその特定のフローの NAT セッションを作成して、後続の ESP トラフィックがスムーズに流れるようにします。ESP トラフィックは、フローや方向ごとに固有のセキュリティパラメータインデックス (SPI) によって識別されます。IPSec ALG は、大規模な NAT を実行するために、送信元ポートと宛先ポートの代わりに ESP SPI を使用します。

ゲートがトラフィックを受信しない場合、ゲートはタイムアウトします。両方のゲートがタイムアウトすると、別の IKE フェーズ 2 交換が許可されます。

IPsec ALG タイムアウト

NetScaler アプライアンスの IPSec ALG には、次の 3 つのタイムアウトパラメータがあります。

- **ESP** ゲートタイムアウト。クライアントとサーバー間で双方向の ESP トラフィックが交換されない場合に、NetScaler アプライアンスが特定のサーバーの特定の NAT IP アドレス上の特定のクライアントの IPsec ALG ゲートをブロックする最大時間。
- **IKE** セッションタイムアウト。そのセッションに IKE トラフィックがない場合に、NetScaler アプライアンスが IKE セッション情報を削除する前に保持する最大時間。
- **ESP** セッションタイムアウト。そのセッションに ESP トラフィックがない場合に、NetScaler アプライアンスが ESP セッション情報を削除する前に保持する最大時間。

IPsec ALG を設定する前に考慮すべきポイント

IPSec ALG の設定を開始する前に、次の点を考慮してください。

- IPsec プロトコルのさまざまなコンポーネントを理解する必要があります。
- IPsec ALG は DS-Lite 構成および大規模な NAT64 構成ではサポートされていません。
- IPsec ALG はヘアピン LSN フローではサポートされていません。
- IPsec ALG は RNAT 構成では機能しません。
- IPsec ALG は NetScaler クラスターではサポートされていません。

構成の手順

NetScaler アプライアンスで大規模な NAT44 用の IPSec ALG を構成するには、次のタスクで構成されます。

- **LSN** アプリケーションプロファイルを作成し、**LSN** 設定にバインドします。アプリケーションプロファイルを設定する際に、次のパラメータを設定します。
  - Protocol=UDP
  - IP Pooling = PAIRED
  - Port=500

アプリケーションプロファイルを LSN 設定の LSN グループにバインドします。LSN 構成の作成手順については、「LSN の設定手順」を参照してください。

- **IPSec ALG** プロファイルを作成します。IPsec プロファイルには、IKE セッションタイムアウト、ESP セッションタイムアウト、ESP ゲートタイムアウトなど、さまざまな IPsec タイムアウトが含まれます。IPSec ALG プロファイルを LSN グループにバインドします。IPSec ALG プロファイルには、次のデフォルト設定があります。
  - IKE セッションタイムアウト = 60 分
  - ESP セッションタイムアウト = 60 分
  - ESP ゲートタイムアウト = 30 秒
- **IPsec ALG** プロファイルを **LSN** 設定にバインドします。IPsec ALG プロファイルを LSN 設定にバインドすると、IPsec ALG が LSN 設定に対して有効になります。IPSec ALG プロファイルパラメータを LSN グループで作成されたプロファイルの名前に設定して、IPSec ALG プロファイルを LSN 構成にバインドします。IPsec ALG プロファイルは複数の LSN グループにバインドできますが、LSN グループには 1 つの IPsec ALG プロファイルしか割り当てることができません。

コマンドラインインターフェイスを使用して **LSN** アプリケーションプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn appsprofile <appsprofilename> UDP -ippooling PAIRED

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して宛先ポートを **LSN** アプリケーションプロファイルにバインドするには

コマンドプロンプトで入力します。

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **LSN** アプリケーションプロファイルを **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> -appsprofilename <string>

show lsn group
<!--NeedCopy-->
```

CLI を使用して **IPSec ALG** プロファイルを作成するには

コマンドプロンプトで入力します。

```
add ipsecalg profile <name> [-ikeSessionTimeout <positive_integer>] [-
    espSessionTimeout <positive_integer>] [-espGateTimeout <
    positive_integer>] [-connfailover ( ENABLED | DISABLED)

show ipsecalg profile <name>
<!--NeedCopy-->
```

CLI を使用して **IPSec ALG** プロファイルを **LSN** 設定にバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> -poolname <string> - ipsecAlgProfile <string
    >

show lsn group <name>
<!--NeedCopy-->
```

GUI を使用して **LSN** アプリケーションプロファイルを作成し、それを **LSN** 設定にバインドするには

[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[アプリケーション] タブをクリックし、LSN アプリケーションプロファイルを追加して LSN グループにバインドします。

GUI を使用して **IPSec ALG** プロファイルを作成するには **** [システム] > [大規模 **NAT**] > [プロファイル] に移動し、[**IPSEC ALG]** タブをクリックして、**IPSec ALG** プロファイルを追加します。

GUI を使用して **IPSec ALG** プロファイルを **LSN** 設定にバインドするには **

1. [システム] > [大規模 **NAT**] > [**LSN** グループ] に移動し、LSN グループを開きます。
2. 詳細設定で、**+ IPSEC ALG** プロファイルをクリックして、作成した IPsec ALG プロファイルを LSN グループにバインドします。

構成例

次の大規模な NAT44 設定の例では、192.0.2.0/24 ネットワークのサブスクライバに対して IPSec ALG が有効になっています。さまざまな IPsec タイムアウト設定を持つ IPsec ALG プロファイル IPSECALGPROFILE-1 が作成され、LSN グループ LSN グループ -1 にバインドされます。

設定例:

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.9

Done

add lsn appsprofile LSN-APPSPROFILE-1 UDP -ippooling PAIRED

Done

bind lsn appsprofile LSN-APPSPROFILE-1 500

Done

add ipsecalg profile  IPSECALGPROFILE-1 -ikeSessionTimeout 45  -
   espSessionTimeout 40  - espGateTimeout 20 -connfailover ENABLED

Done

bind lsn group LSN-GROUP-1 -appsprofilename LSN-APPSPROFILE-1

Done

bind lsn group LSN-GROUP-1 -poolname LSN-POOL-1

Done

bind lsn group LSN-GROUP-1 - ipsecAlgProfile IPSECALGPROFILE-1

Done
<!--NeedCopy-->
```


LSN のログ記録と監視

August 15, 2023

LSN 情報を記録して、問題の診断、トラブルシューティング、法的要件を満たすことができます。LSN 統計カウンタを使用して現在の LSN セッションを表示することで、LSN 機能のパフォーマンスを監視できます。

LSN のロギング

LSN 情報のロギングは、ISP が法的要件を満たし、いつでもトラフィックの送信元を特定するために必要となる重要な機能の 1 つです。

NetScaler アプライアンスは、LSN マッピングエントリと、LSN グループごとに作成または削除された LSN セッションを記録します。LSN グループのロギングパラメータとセッションロギングパラメータを使用して、LSN グループの LSN 情報のロギングを制御できます。これらはグループレベルのパラメータで、デフォルトでは無効になっています。NetScaler アプライアンスは、ログとセッションログの両方のパラメーターが有効になっている場合にのみ、LSN グループの LSN セッションをログに記録します。

次の表は、ロギングおよびセッションロギングパラメータのさまざまな設定に対する LSN グループのロギング動作を示しています。

ログ	セッションロギング	ロギング動作
有効	有効	LSN マッピングエントリと LSN セッションを記録します。
有効	無効	LSN マッピングエントリをログに記録しますが、LSN セッションは記録しません。
無効	有効	マッピングエントリも LSN セッションも記録しません。

LSN マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）。
- タイムスタンプ
- エントリータイプ (マッピング)
- LSN マッピングエントリが作成されたか削除されたか
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名

- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が表示される場合があります。
  - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートは記録されません。
  - アドレス依存マッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレス-ポート依存マッピング用にログに記録されます。

LSN セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）。
- タイムスタンプ
- エントリータイプ (セッション)
- LSN セッションが作成されるか削除されるか
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

アプライアンスは、既存の Syslog および監査ログフレームワークを使用して LSN 情報を記録します。関連する NSLOG アクションと SYLOG アクションエンティティの LSN パラメータを有効にして、グローバルレベルの LSN ロギングを有効にする必要があります。Logging パラメーターを有効にすると、NetScaler アプライアンスはこの LSN グループの LSN マッピングと LSN セッションに関連するログメッセージを生成します。次に、アプライアンスはこれらのログメッセージを NSLOG アクションと SYSLOG アクションエンティティに関連付けられたサーバーに送信します。

LSN 情報をログに記録するには、Citrix は次のことを推奨します。

- LSN 情報を NetScaler アプライアンスではなく外部ログサーバーに記録します。外部サーバにログオンすると、アプライアンスが大量（数百万単位）の LSN ログエントリを作成するときに、最適なパフォーマンスが得られます。
- TCP 経由の SYSLOG、または NSLOG を使用する。デフォルトでは、SYSLOG は UDP を使用し、NSLOG は TCP のみを使用してログ情報をログサーバに転送します。TCP は、完全なデータを転送するために UDP よりも信頼性が高いです。

  注記:

  - NetScaler アプライアンスで生成された SYSLOG は、外部のログサーバーに動的に送信されます。
  - SYSLOG over TCP を使用する場合、TCP 接続がダウンしているか、SYSLOG サーバーがビジー状態である場合、NetScaler ADC アプライアンスはログをバッファに保存し、接続がアクティブになるとデータを送信します。

ログの構成の詳細については、「監査ログ」を参照してください。

LSN ロギングの設定は、次の作業で構成されます。

- **NetScaler** アプライアンスをロギング用に構成します。このタスクには、NetScaler アプライアンスのさまざまなエンティティとパラメーターの作成と設定が含まれます。
  - **SYSLOG** または **NSLOG** 監査ロギング設定を作成します。監査ログ設定の作成には、次のタスクが含まれます。
    * NSLOG または SYSLOG 監査アクションを作成し、LSN パラメータを有効にします。監査アクションは、ログサーバーの IP アドレスを指定します。
    * SYSLOG または NSLOG 監査ポリシーを作成し、監査アクションを監査ポリシーにバインドします。監査アクションは、ログサーバーの IP アドレスを指定します。オプションで、外部ログサーバーに送信されるログメッセージの転送方法を設定できます。デフォルトでは UDP が選択されているので、転送方法を TCP に設定して信頼性の高い転送メカニズムを実現できます。監査ポリシーをシステムグローバルにバインドしてください。
    * SYSLOG または NSLOG 監査ポリシーを作成し、監査アクションを監査ポリシーにバインドします。
    * 監査ポリシーをシステムグローバルにバインドします。
      注: 既存の監査ログ構成の場合は、監査アクションで指定されたサーバーに LSN 情報を記録するための LSN パラメーターを有効にするだけです。
  - ロギングとセッションロギングのパラメーターを有効にします。LSN グループを追加する際、またはグループを作成した後に、ロギングおよびセッションロギングパラメータを有効にします。NetScaler アプライアンスは、これらの LSN グループに関連するログメッセージを生成し、LSN パラメーターが有効になっている監査アクションのサーバーに送信します。
- ログサーバーの設定。このタスクには、目的のサーバに SYSLOG または NSLOG パッケージをインストールすることが含まれます。このタスクには、SYSLOG または NSLOG の構成ファイルに NetScaler アプライアンスの NSIP アドレスを指定することも含まれます。NSIP アドレスを指定すると、サーバは NetScaler ADC アプライアンスから送信されるログ情報を識別して、ログファイルに保存できます。

ログの構成の詳細については、「監査ログ」を参照してください。

コマンドラインインターフェイスを使用した **SYSLOG** の設定

コマンドラインインターフェイスを使用して **LSN** ロギング用の **SYSLOG** サーバーアクションを作成するには　コマンドプロンプトで入力します。

```
1 add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel
      <logLevel>... [-transport (TCP)] [-lsn ( ENABLED | DISABLED )]
2 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** ロギング用の **SYSLOG** サーバーポリシーを作成するには　コマンドプロンプトで入力します。


```
add audit syslogPolicy <name> <rule> <action>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** ロギング用に **SYSLOG** サーバーポリシーをシステムグローバルにバインドするには　コマンドプロンプトで入力します。

```
bind system global [<policyName> [-priority <positive_integer>]]
<!--NeedCopy-->
```

設定ユーティリティを使用した **SYSLOG** 設定

構成ユーティリティを使用して **LSN** ロギング用の **SYSLOG** サーバーアクションを設定するには

1. [システム] > [監査] > [**Syslog**] に移動し、[サーバー] タブで新しい監査サーバーを追加するか、既存のサーバーを編集します。
2. LSN ロギングを有効にするには、大規模な **NAT** ロギングオプションを選択します。
3. **(**オプション**) TCP** 経由の **SYSLOG** を有効にするには、**TCP** ロギングオプションを選択します。

構成ユーティリティを使用して **LSN** ロギング用の **SYSLOG** サーバーポリシーを構成するには　[システム] > [監査] > [**Syslog**] に移動し、[ポリシー] タブで新しいポリシーを追加するか、既存のポリシーを編集します。

設定ユーティリティを使用して **LSN** ロギング用に **SYSLOG** サーバポリシーをシステムグローバルにバインドするには

1. [システム] > [監査] > [**Syslog**] に移動します。
2.「ポリシー」タブの「アクション」リストで、「グローバルバインディング」をクリックして監査グローバルポリシーをバインドします。

コマンドラインインターフェイスを使用した **NSLOG** 設定

コマンドラインインターフェイスを使用して **LSN** ロギング用の **NSLOG** サーバーアクションを作成するには　コマンドプロンプトで入力します。

```
add audit nslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel> ... [-lsn ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** ロギング用の **NSLOG** サーバーポリシーを作成するには　コマンドプロンプトで入力します。


```
add audit nslogPolicy <name> <rule> <action>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **NSLOG** サーバーポリシーをシステムグローバルにバインドして **LSN** ロギングを行うには　コマンドプロンプトで入力します。

```
bind system global [<policyName>]
<!--NeedCopy-->
```

構成ユーティリティを使用した **NSLOG** 設定

構成ユーティリティを使用して **LSN** ロギング用の **NSLOG** サーバーアクションを設定するには

1. [システム] > [監査] > [**Nslog**] に移動し、[サーバー] タブで新しい監査サーバーを追加するか、既存のサーバーを編集します。
2. LSN ロギングを有効にするには、大規模な **NAT** ロギングオプションを選択します。

構成ユーティリティを使用して **LSN** ロギング用の **NSLOG** サーバーポリシーを構成するには　[システム] > [監査] > [**Nslog**] に移動し、[ポリシー] タブで新しいポリシーを追加するか、既存のポリシーを編集します。

構成ユーティリティを使用して **NSLOG** サーバポリシーをシステムグローバルにバインドして **LSN** ロギングを行うには

1. [システム] > [監査] > [**Nlog**] に移動します。
2.「ポリシー」タブの「アクション」リストで、「グローバルバインディング」をクリックして監査グローバルポリシーをバインドします。

例　次の設定では、LSN ログを含むログエントリを保存する 2 台の SYSLOG サーバと 2 台の NSLOG サーバを指定します。LSN ロギングは LSN グループ LSN-GROUP-2 と LSN-GROUP-3 に設定されます。

NetScaler アプライアンスは、これらの LSN グループの LSN マッピングと LSN セッションに関連するログメッセージを生成し、指定されたログサーバーに送信します。

```
add audit syslogAction SYS-ACTION-1 198.51.101.10 -logLevel ALL -lsn
    ENABLED
Done
add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done
bind system global SYSLOG-POLICY-1
Done

add audit syslogAction SYS-ACTION-2 198.51.101.20 -logLevel ALL -lsn
    ENABLED
```



```
Done
add audit syslogPolicy SYSLOG-POLICY-2 ns_true SYS-ACTION-2
Done
bind system global SYSLOG-POLICY-2
Done

add audit nslogAction NSLOG-ACTION-1 198.51.101.30 -logLevel ALL -lsn
    ENABLED
Done
add audit nslogPolicy NSLOG-POLICY-1 ns_true NSLOG-ACTION-1
Done
bind system global NSLOG-POLICY-1
Done
add audit nslogAction NSLOG-ACTION-2 198.51.101.40 -logLevel ALL -lsn
    ENABLED
Done
add audit nslogPolicy NSLOG-POLICY-2 ns_true NSLOG-ACTION-2
Done
bind system global NSLOG-POLICY-2
Done

add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-2  - logging ENABLED  -
    sessionLogging ENABLED
Done
set lsn group LSN-GROUP-2   - logging ENABLED  - sessionLogging ENABLED
Done
<!--NeedCopy-->
```

次の設定では、TCP を使用して外部の SYSLOG サーバー 192.0.2.10
にログメッセージを送信するための SYSLOG 設定を指定します。

```
add audit syslogAction SYS-ACTION-1 192.0.2.10 -logLevel ALL -transport
    TCP
Done

add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done

bind system global SYSLOG-POLICY-1
Done
<!--NeedCopy-->
```

次の表は、設定されたログサーバに保存されている各タイプのサンプル LSN ログエントリを示しています。これらの LSN ログエントリは、NSIP アドレスが 10.102.37.115 の NetScaler アプライアンスによって生成されます。

LSN ログエントリタイプ	サンプルログエントリ
LSN セッションの作成	Local4.Informational 10.102.37.115 08/05/2014:09:59:48 GMT 0-PPE-0 : LSN LSN_SESSION 2581750 : SESSION CREATED Client IP:Port:TD 192.0.2.10: 15136:0, NatIP:NatPort 203.0.113.6: 6234, Destination IP:Port:TD 198.51.100.9: 80:0, Protocol: TCP
LSN セッションの削除	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_SESSION 3871790 : SESSION DELETED Client IP:Port:TD 192.0.2.11: 15130:0, NatIP:NatPort 203.0.113.6: 7887, Destination IP:Port:TD 198.51.101.2:80:0, Protocol: TCP
LSN マッピングの作成	Local4.Informational 10.102.37.115 08/05/2014:09:59:47 GMT 0-PPE-0 : LSN LSN_MAPPING 2581580 : EIM CREATED Client IP:Port 192.0.2.15: 14567, NatIP:NatPort 203.0.113.5: 8214, Protocol: TCP
LSN マッピングの削除	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_MAPPING 3871700 : EIM DELETED Client IP:Port 192.0.3.15: 14565, NatIP:NatPort 203.0.113.11: 8217, Protocol: TCP

最小限のロギング

確定的な LSN 構成とポートブロックを使用した動的 LSN 構成により、LSN ログの量は大幅に減少します。これら 2 種類の構成では、NetScaler アプライアンスは NAT IP アドレスとポートブロックを加入者に割り当てます。NetScaler アプライアンスは、サブスクライバーへの割り当て時にポートブロックのログメッセージを生成します。NetScaler アプライアンスは、NAT IP アドレスとポートブロックが解放されたときにもログメッセージを生成します。接続の場合、マッピングされた NAT IP アドレスとポートブロックだけで加入者を識別できます。このため、NetScaler アプライアンスは作成または削除された LSN セッションを記録しません。また、アプライアンスは、セッション用に作成されたマッピングエントリも、マッピングエントリが削除されたときもログに記録しません。

確定的な LSN 設定およびポートブロック付きの動的 LSN 設定の最小ロギング機能はデフォルトで有効になっており、無効にする設定はありません。つまり、NetScaler アプライアンスは、確定的な LSN 構成とポートブロック付きの動的 LSN 構成に対して自動的に最小限のロギングを行います。この機能を無効にするオプションはありません。アプライアンスは、設定されたすべてのログサーバにログメッセージを送信します。

各ポートブロックのログメッセージは、次の情報で構成されます。

- NetScaler アプライアンスの NSIP アドレス
- タイムスタンプ
- エントリータイプは「デターミニスティック」または「ポートブロック」
- ポートブロックが割り当てられているか解放されているか
- 加入者の IP アドレスと割り当てられた NAT IP アドレスとポートブロック
- プロトコル名

確定的な **LSN** 設定のための最小限のロギング

IP アドレスが 192.0.17.1、192.0.17.2、192.0.17.3、および 192.0.17.4 の 4 人のサブスクライバを対象とした、シンプルで確定的な LSN 設定の例を考えてみましょう。

この LSN 設定では、ポートブロックサイズは 32768 に設定され、LSN NAT IP アドレスプールの IP アドレスは 203.0.113.19-203.0.113.23 の範囲にあります。

```
1  add lsn client LSN-CLIENT-7
2  Done
3  bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask
       255.255.255.253
4  Done
5  add lsn pool LSN-POOL-7 -nattype DETERMINISTIC
6  Done
7  bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23
8  Done
9  add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype
       DETERMINISTIC -portblocksize 32768
10 Done
11 bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7
12 Done
13 <!--NeedCopy-->
```

NetScaler アプライアンスは、LSN NAT IP プールから、設定されたポートブロックサイズに基づいて、LSN NAT IP アドレスとポートブロックを各サブスクライバーに順番に事前割り当てします。最初の NAT IP アドレス（203.0.113.19）の最初のポートブロック（1024-33791）を、最初のサブスクライバの IP アドレス（192.0.17.1）に割り当てます。次の範囲のポートが次のサブスクライバに割り当てられ、NAT アドレスに次のサブスクライバ用のポートが足りなくなるまで続きます。その時点で、次の NAT IP アドレスの最初のポートブロックが加入者に割り当てられ、以降も同様に加入者に割り当てられます。アプライアンスは、NAT IP アドレスと各サブスクライバに割り当てられたポートブロックを記録します。

NetScaler アプライアンスは、これらのサブスクライバーに対して作成または削除された LSN セッションを記録しません。アプライアンスは、LSN 構成に関する次のログメッセージを生成します。

```
1  1) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
       LSN_DETERMINISTIC 79201453 0 :  Dtrstc ALLOC Client 12.0.0.241,
       NatInfo 50.0.0.2:59904 to 60415
```



```
2) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201454 0 :  Dtrstc ALLOC Client 12.0.0.242,
    NatInfo 50.0.0.2:60416 to 60927
3) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243,
    NatInfo 50.0.0.2:60928 to 61439
4) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243,
    NatInfo 50.0.0.2:60928 to 61439
<!--NeedCopy-->
```

LSN 設定を削除すると、割り当てられた NAT IP アドレスとポートブロックが各サブスクライバから解放されます。アプライアンスは、各サブスクライバから解放された NAT IP アドレスとポートブロックを記録します。LSN 設定を削除すると、アプライアンスはサブスクライバごとに次のログメッセージを生成します。

```
1) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201706 0 :  Dtrstc FREE Client 12.0.0.238,
    NatInfo 50.0.0.2:58368 to 58879
2) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201707 0 :  Dtrstc FREE Client 12.0.0.239,
    NatInfo 50.0.0.2:58880 to 59391
3) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201708 0 :  Dtrstc FREE Client 12.0.0.240,
    NatInfo 50.0.0.2:59392 to 59903
<!--NeedCopy-->
```

ポートブロックによるダイナミック **LSN** 設定の最小ロギング

ネットワーク 192.0.2.0/24 のすべてのサブスクライバに対してポートブロックを使用する単純な動的 LSN 設定の例を考えてみましょう。この LSN 設定では、ポートブロックサイズは 1024 に設定され、LSN NAT IP アドレスプールの IP アドレスは 203.0.113.3 ～203.0.113.4 の範囲にあります。

```
set lsn parameter -memLimit 4000
Done
add lsn client LSN-CLIENT-1
Done
bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done
add lsn pool LSN-POOL-1
Done
bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
Done
add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
Done
bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done
<!--NeedCopy-->
```

NetScaler アプライアンスは、サブスクライバーが初めてセッションを開始するときに、設定されたポートブロックサイズに基づいて、LSN NAT IP プールからランダムな NAT IP アドレスとポートブロックを割り当てます。

NetScaler は、このサブスクライバーに割り当てられた NAT IP アドレスとポートブロックを記録します。アプライアンスは、このサブスクライバに対して作成または削除された LSN セッションをログに記録しません。すべてのポートがサブスクライバに割り当てられたポートブロックから（異なるサブスクライバセッション用に）割り当てられる場合、アプライアンスはサブスクライバに追加セッション用の新しいランダム NAT IP アドレスとポートブロックを割り当てます。NetScaler は、サブスクライバーに割り当てられたすべての NAT IP アドレスとポートブロックを記録します。

IP アドレス 192.0.2.1 のサブスクライバがセッションを開始すると、アプライアンスは次のログメッセージを生成します。ログメッセージには、アプライアンスが NAT IP アドレス 203.0.113.3 とポートブロック 1024-2047 を加入者に割り当てたことが示されています。

```
1 03/23/2015:00:07:12 GMT  Informational 0-PPE-3 : default LSN
     LSN_PORTBLOCK 106725793 0 :  Portblock ALLOC Client 12.0.2.72,
     NatInfo 203.0.113.3:1024 to 2047, Proto:TCP
2 <!--NeedCopy-->
```

割り当てられた NAT IP アドレスと割り当てられたポートブロック内のポートの 1 つを使用するセッションがなくなると、割り当てられた NAT IP アドレスとポートブロックが加入者から解放されます。NetScaler は、NAT IP アドレスとポートブロックがサブスクライバーから解放されたことをログに記録します。アプライアンスは、割り当てられた NAT IP アドレス（203.0.113.3）と割り当てられたポートブロック（1024-2047）を使用するセッションがなくなると、IP アドレス 192.0.2.1 のサブスクライバに次のログメッセージを生成します。ログメッセージには、NAT IP アドレスとポートブロックが加入者から解放されたことが示されます。

```
1 03/23/2015:00:11:09 GMT Informational 0-PPE-3 : default LSN
     LSN_PORTBLOCK 106814342 0 :  Portblock FREE Client 12.0.3.122,
     NatInfo 203.0.113.3: 1024 to 2047, Proto:TC
2 <!--NeedCopy-->
```

SYSLOG サーバーの負荷分散

NetScaler ADC アプライアンスは、構成されたすべての外部ログサーバーに SYSLOG イベントとメッセージを送信します。その結果、冗長なメッセージが保存され、システム管理者の監視が困難になります。この問題に対処するために、NetScaler ADC アプライアンスは、メンテナンスとパフォーマンスを向上させるために外部ログサーバー間で SYSLOG メッセージを負荷分散できる負荷分散アルゴリズムを提供します。サポートされる負荷分散アルゴリズムには、ラウンドロビン、最小帯域幅、カスタムロード、最小接続、最小パケット、および AuditLogHash が含まれます。

コマンドラインインターフェイスを使用した **SYSLOG** サーバーの負荷分散

サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

```
1 add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP |
     SYSLOGUDP)> <port>
2 <!--NeedCopy-->
```


負荷分散仮想サーバーを追加し、サービスタイプを SYSLOGTCP または SYSLOGTCP、負荷分散方法を AUDITLOGHASH として指定します。

```
add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
    <AUDITLOGHASH>]
<!--NeedCopy-->
```

サービスを負荷分散仮想サーバーに接続します。

```
Bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つ負荷分散サーバー名を指定します。

```
add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
     <logLevel>]
<!--NeedCopy-->
```

ルールとアクションを指定して SYSLOG ポリシーを追加します。

```
add syslogpolicy <name> <rule> <action>
<!--NeedCopy-->
```

ポリシーを有効にするには、SYSLOG ポリシーをシステムグローバルにバインドします。

```
bind system global <policyName>
<!--NeedCopy-->
```

構成ユーティリティを使用した **SYSLOG** サーバーの負荷分散

1. サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

   [トラフィック管理] > [サービス] に移動し、[追加] をクリックして、プロトコルとして [SYLOGTCP] または [SYSLOGUDP] を選択します。

2. 負荷分散仮想サーバーを追加し、サービスタイプを SYSLOGTCP または SYSLOGTCP、負荷分散方法を AUDITLOGHASH として指定します。

   [トラフィック管理] > [仮想サーバー] に移動し、[追加] をクリックして、プロトコルとして SYLOGTCP または SYSLOGUDP を選択します。

3. サービスを負荷分散仮想サーバーに移行し、サービスに接続します。

   サービスを負荷分散仮想サーバーに接続します。

   [トラフィック管理] > [仮想サーバー] に移動し、仮想サーバーを選択してから、負荷分散方法で [Audit-LogHash] を選択します。

4. SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つ負荷分散サーバー名を指定します。

   [システム] > [監査] に移動し、[サーバー] をクリックし、[サーバー] の [LB Vserver] オプションを選択してサーバーを追加します。

5. ルールとアクションを指定して SYSLOG ポリシーを追加します。

   [システム] > [Syslog] に移動し、[ポリシー] をクリックして、SYSLOG ポリシーを追加します。

6. ポリシーを有効にするには、SYSLOG ポリシーをシステムグローバルにバインドします。

   [システム] > [Syslog] に移動し、SYSLOG ポリシーを選択して [アクション] をクリックし、[グローバルバインディング] をクリックして、ポリシーをシステムグローバルにバインドします。

例:

次の構成では、AUDITLOGHASH を負荷分散方法として使用して、外部ログサーバー間で SYSLOG メッセージの負荷分散を指定します。NetScaler ADC アプライアンスは、サービス、service1、service2、およびサービス 3 の間で負荷分散される SYSLOG イベントとメッセージを生成します。

```
add service service1 192.0.2.10 SYSLOGUDP 514
Done

add service service2 192.0.2.11 SYSLOGUDP 514
Done

add service service3 192.0.2.11 SYSLOGUDP 514
Done

add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
Done

bind lb vserver lbvserver1 service1
Done

bind lb vserver lbvserver1 service2
Done

bind lb vserver lbvserver1 service3
Done

add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
Done

add syslogpolicy syspol1 ns_true sysaction1
Done

bind system global syspol1
Done
<!--NeedCopy-->
```


HTTP ヘッダー情報のロギング

NetScaler アプライアンスは、NetScaler の LSN 機能を使用する HTTP 接続のリクエストヘッダー情報をログに記録できるようになりました。HTTP リクエストパケットの次のヘッダー情報を記録できます。

- HTTP リクエストの送信先の URL。
- HTTP リクエストで指定された HTTP メソッド。
- HTTP リクエストで使用される HTTP バージョン。
- HTTP リクエストを送信したサブスクライバーの IP アドレス。

ISP は HTTP ヘッダーログを使用して、特定のサブスクライバーの HTTP プロトコルに関連する傾向を確認できます。たとえば、ISP はこの機能を使用して、一部の加入者の中で最も人気のある Web サイトを検索できます。

HTTP ヘッダーログプロファイルは、ロギングを有効または無効にできる HTTP ヘッダー属性 (URL や HTTP メソッドなど) のコレクションです。その後、HTTP ヘッダーログプロファイルは LSN グループにバインドされます。次に、NetScaler アプライアンスは、LSN グループに関連するすべての HTTP リクエストの HTTP ヘッダー属性（バインドされた HTTP ヘッダーログプロファイルでロギング用に有効になっている）を記録します。次に、アプライアンスは設定されたログサーバーにログメッセージを送信します。

HTTP ヘッダーログプロファイルは複数の LSN グループにバインドできますが、LSN グループには 1 つの HTTP ヘッダーログプロファイルしか割り当てることができません。

コマンドラインインターフェイスを使用して **HTTP** ヘッダーログプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **HTTP** ヘッダーログプロファイルを **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```


例

次の LSN 設定の例では、HTTP ヘッダーログプロファイル HTTP-Header-Log-1 が LSN グループ LSN-GROUP-1 にバインドされています。ログプロファイルには、すべての HTTP 属性（URL、HTTP メソッド、HTTP バージョン、ホスト IP アドレス）のロギングが有効になっているので、LSN グループに関連する（ネットワーク 192.0.2.0/24 の）サブスクライバーからの HTTP 要求に対してこれらのすべての属性がログに記録されます。

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done

set lsn parameter -memLimit 4000
Done

add lsn client LSN-CLIENT-1
Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done

add lsn pool LSN-POOL-1
Done

bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done

bind lsn group  LSN-GROUP-1  -httphdrlogprofilename HTTP-HEADER-LOG-1
Done
<!--NeedCopy-->
```

NetScaler は、LSN 構成例に属するサブスクライバーの 1 人が HTTP リクエストを送信すると、次の HTTP ヘッダーログメッセージを生成します。

ログメッセージから、IP アドレス 192.0.2.33 のクライアントが HTTP メソッド GET と HTTP バージョン 1.1 を使用して URL example.com に HTTP リクエストを送信したことがわかります。

```
03/19/2015:16:24:04 GMT  Informational 0-PPE-1 : default LSN Message 59
     0 :  "LSN Client IP:TD 10.102.37.118:0 URL: example.com Host:
    192.0.2.33 Version: HTTP1.1 Method: GET"
<!--NeedCopy-->
```


MSISDN 情報のロギング

モバイルステーション統合加入者電話番号（MSISDN）は、複数のモバイルネットワークにわたって加入者を一意に識別する電話番号です。MSISDN には、加入者の事業者を識別する国コードと国内宛先コードが関連付けられています。

モバイルネットワークのサブスクライバーの LSN ログエントリに MSISDNS を含めるように NetScaler アプライアンスを構成できます。LSN ログに MSISDN があると、管理者はポリシーや法律に違反したモバイル加入者や、合法的な傍受機関から情報を要求されたモバイル加入者を迅速かつ正確にバックトレースできます。

次の LSN ログエントリの例には、LSN 設定のモバイル加入者からの接続の MSISDN 情報が含まれています。ログエントリには、MSISDN が E 164:5556543210 のモバイル加入者が、NAT IP: ポート 203.0.113. 3:45195 を介して宛先 IP: ポート 23.0.0. 1:80 に接続されたことが示されています。

ログエントリタイプ	サンプルログエントリ
LSN セッションの作成	Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN マッピングの作成	Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN セッションの削除	Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN マッピング	Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP

MSISDN 情報を **LSN** ログに含めるには、次のタスクを実行します

- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、LSN 設定の LSN ログに MSISDN 情報を含めるかどうかを指定するログサブスクライバー ID パラメータが含まれます。LSN ログプロファイルを作成するときに、ログサブスクライバー ID パラメータを有効にします。
- **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドします。作成された LSN ログプロファイル名にログプロファイル名パラメータを設定して、作成された LSN ログプロファイルを LSN 構成の LSN グループにバインドします。大規模な NAT の設定手順については、LSN の設定手順を参照してください。

CLI を使用して **LSN** ログプロファイルを作成するには　コマンドプロンプトで入力します。

```
add lsn logprofile <logprofilename -logSubscriberID ( ENABLED |
   DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには　コマンドプロンプトで入力します。

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

設定例**:**

この LSN 設定の例では、LSN ログプロファイルのログサブスクライバー ID パラメータが有効になっています。プロファイルは LSN グループ LSN-GROUP-9 にバインドされます。MSISDN 情報は、（ネットワーク 192.0.2.0/24 の）モバイル加入者からの接続の LSN セッションと LSN マッピングログに含まれます。

```
add lsn logprofile  LOG-PROFILE-MSISDN-9   -logSubscriberID ENABLED

Done
add lsn client LSN-CLIENT-9

Done
bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0

Done
add lsn pool LSN-POOL-9

Done
bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4

Done
add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

```



```
Done
bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done
bind lsn group  LSN-GROUP-9  -logprofilename LOG-PROFILE-MSISDN-9

Done
<!--NeedCopy-->
```

現在の **LSN** セッションの表示

現在の LSN セッションを表示して、NetScaler アプライアンス上の不要または非効率的な LSN セッションを検出できます。選択パラメータに基づいて、すべての LSN セッションまたは一部の LSN セッションを表示できます。

注：NetScaler アプライアンスに 100 万を超える LSN セッションが存在する場合、Citrix では選択パラメーターを使用してすべてのセッションではなく選択した LSN セッションを表示することをお勧めします。

コマンドラインインターフェイスを使用した設定

コマンドラインインターフェイスを使用してすべての **LSN** セッションを表示するには　コマンドプロンプトで入力します。

```
show lsn session
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して選択的な **LSN** セッションを表示するには　コマンドプロンプトで入力します。

```
show lsn session [-clientname <string>] [-network <ip_addr> [-netmask <
    netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <
    port>]]
<!--NeedCopy-->
```

例　NetScaler に存在するすべての LSN セッションを表示するには


```
> show lsn session
          SubscrIP              SubscrPort      SubscrTD              DstIP                       DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.10              15136           0               198.51.100.9            80              0       203.0.113.6     6234            TCP     OUT
2.      192.0.2.11              15130           0               198.51.101.2            80              0       203.0.113.6     7887            TCP     OUT
3.      192.0.2.12              16136           0               198.51.100.3            80              0       203.0.113.6     9807            TCP     OUT
4.      192.0.2.13              18148           0               198.51.101.6            80              0       203.0.113.6     4657            TCP     OUT
5.      192.0.2.14              13560           0               198.51.101.7            80              0       203.0.113.7     9341            TCP     OUT
6.      192.0.2.15              14567           0               198.51.100.8            80              0       203.0.113.5     8214            TCP     OUT
7.      192.0.2.15              16890           0               198.51.101.1            80              0       203.0.113.5     8214            TCP     OUT
8.      192.0.2.16              12345           0               198.51.102.9            80              0       203.0.113.5     1678            TCP     OUT
9.      192.0.2.19              19876           0               198.51.103.8            80              0       203.0.113.5     1567            TCP     OUT
10.     192.0.2.20              10989           0               198.51.104.19           80              0       203.0.113.11    1343            TCP     OUT
11.     192.0.3.13              18149           0               198.51.101.61           80              0       203.0.113.11    4653            TCP     OUT
12.     192.0.3.14              13510           0               198.51.101.74           80              0       203.0.113.11    9344            TCP     OUT
13.     192.0.3.15              14565           0               198.51.100.82           80              0       203.0.113.11    8217            TCP     OUT
14.     192.0.3.15              16899           0               198.51.101.12           80              0       203.0.113.11    8219            TCP     OUT
15.     192.0.3.16              12343           0               198.51.102.99           80              0       203.0.113.11    1673            TCP     OUT

Done
```

LSN クライアントエンティティ LSN-CLIENT-2 に関連するすべての LSN セッションを表示するには

```
> show lsn session -clientname LSN-CLIENT-2
        SubscrIP                SubscrPort      SubscrTD              DstIP                     DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.10              15136           0               198.51.100.9            80              0       203.0.113.6     6B234           TCP     OUT
2.      192.0.2.11              15130           0               198.51.101.2            80              0       203.0.113.6     7887            TCP     OUT
3.      192.0.2.12              16136           0               198.51.100.3            80              0       203.0.113.6     9807            TCP     OUT
4.      192.0.2.13              18148           0               198.51.101.6            80              0       203.0.113.6     4657            TCP     OUT
5.      192.0.2.14              13560           0               198.51.101.7            80              0       203.0.113.7     9341            TCP     OUT
6.      192.0.2.15              14567           0               198.51.100.8            80              0       203.0.113.5     8214            TCP     OUT
7.      192.0.2.15              16890           0               198.51.101.1            80              0       203.0.113.5     8214            TCP     OUT
8.      192.0.2.16              12345           0               198.51.102.9            80              0       203.0.113.5     1678            TCP     OUT
9.      192.0.2.19              19876           0               198.51.103.8            80              0       203.0.113.5     1567            TCP     OUT
10.     192.0.2.20              10989           0               198.51.104.19           80              0       203.0.113.11    1343            TCP     OUT

Done
```

NAT IP アドレスとして 203.0.113.5 を使用するすべての LSN セッションを表示するには

```
> show lsn session –natIP 203.0.113.5
SubscrIP                SubscrPort      SubscrTD        DstIP                   DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.15              14567           0               198.51.100.8            80              0       203.0.113.5     8214            TCP     OUT
2.      192.0.2.15              16890           0               198.51.101.1            80              0       203.0.113.5     8214            TCP     OUT
3.      192.0.2.16              12345           0               198.51.102.9            80              0       203.0.113.5     1678            TCP     OUT
4.      192.0.2.19              19876           0               198.51.103.8            80              0       203.0.113.5     1567            TCP     OUT
Done
```

設定ユーティリティを使った設定

構成ユーティリティを使用してすべての **LSN** セッションまたは選択した **LSN** セッションを表示するには

1. [システム] > [大規模 NAT] > [セッション] に移動し、[NAT44] タブをクリックします。
2. 選択パラメータに基づいて LSN セッションを表示するには、[検索（Search）] をクリックします。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- show lsn session
  - clientname

    LSN クライアントエンティティの名前。最大長：127
  - network

    加入者の IP アドレスまたはネットワークアドレス。

- ネットマスク

  ネットワークパラメータで指定された IP アドレスのサブネットマスク。

  デフォルト値:255.255.255.255

- td

  LSN クライアントエンティティのトラフィックドメイン ID。

  デフォルト値:0

  最小値:0

  最大値:4094

- NaTiP

  LSN セッションで使用されるマップされた NAT IP アドレス。

LSN 統計情報の表示

LSN 機能に関連する統計情報を表示して、LSN 機能のパフォーマンスを評価したり、問題をトラブルシューティングしたりできます。LSN 機能または特定の LSN グループの統計情報の概要を表示できます。統計カウンターには、NetScaler アプライアンスが最後に再起動されてからのイベントが反映されます。NetScaler アプライアンスを再起動すると、これらのカウンタはすべて 0 にリセットされます。

コマンドラインインターフェイスを使用してすべての **LSN** 統計を表示するには

コマンドプロンプトで入力します。

```
stat lsn
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して指定した **LSN** グループの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat lsn group [<groupname>]
<!--NeedCopy-->
```

例

```
> stat lsn

Large Scale NAT statistics
                                                        Rate(/s)
                                                           Total
```



```
5  LSN TCP Received Packets                              0
       40
6  LSN TCP Received Bytes                                0
       3026
7  LSN TCP Transmitted Packets                           0
       40
8  LSN TCP Transmitted Bytes                             0
       3026
9  LSN TCP Dropped Packets                               0
                          0
10 LSN TCP Current Sessions                              0
                          0
11 LSN UDP Received Packets                              0
                          0
12 LSN UDP Received Bytes                                0
                          0
13 LSN UDP Transmitted Packets                           0
                          0
14 LSN UDP Transmitted Bytes                             0
                          0
15 LSN UDP Dropped Packets                               0
                          0
16 LSN UDP Current Sessions                              0
                          0
17 LSN ICMP Received Packets                             0
       982
18 LSN ICMP Received Bytes                               0
       96236
19 LSN ICMP Transmitted Packets                          0
                          0
20 LSN ICMP Transmitted Bytes                            0
                          0
21 LSN ICMP Dropped Packets                              0
       982
22 LSN ICMP Current Sessions                             0
                          0
23 LSN Subscribers                                       0
                          1
24
25 Done
26
27 > stat lsn group LSN-GROUP-1
28
29 LSN Group Statistics
30                                                  Rate (/s)
                                                       Total
31 TCP Translated Pkts                                   0
       40
32 TCP Translated Bytes                                  0
       3026
33 TCP Dropped Pkts                                      0
                          0
34 TCP Current Sessions                                  0
```



```
                          0
UDP Translated Pkts                                 0
                          0
UDP Translated Bytes                                0
                          0
UDP Dropped Pkts                                    0
                          0
UDP Current Sessions                                0
                          0
ICMP Translated Pkts                                0
                          0
ICMP Translated Bytes                               0
                          0
ICMP Dropped Pkts                                   0
                          0
ICMP Current Sessions                               0
                          0
Current Subscribers                                 0
                          1

Done
<!--NeedCopy-->
```

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- stat lsn group
  - グループ名

    LSN グループの名前。最大長：127
  - 細部

    詳細な出力を指定します (詳細な統計情報を含む)。出力はかなり大きくなることがあります。この引数がない場合、出力には要約のみが表示されます。
  - fullValues

    数字と文字列を完全な形式で表示するように指定します。このオプションがないと、長い文字列は短くなり、大きい数字は省略されます。
  - ntimes

    統計情報を表示する回数を 7 秒間隔で指定します。

    デフォルト値:1
  - logFile

    入力として使用するログファイルの名前。
  - clearstats

統計/カウンタをクリア

設定可能な値: ベーシック、フル

コンパクトロギング

LSN 情報のロギングは、ISP が法的要件を満たし、いつでもトラフィックの送信元を特定できるようにするために必要となる重要な機能の 1 つです。その結果、最終的には膨大な量のログデータになり、ISP はロギングインフラストラクチャの維持に多額の投資をする必要があります。

コンパクトロギングは、イベント名とプロトコル名のショートコードを含む表記変更を行うことでログサイズを小さくする手法です。たとえば、C はクライアント、SC は作成されたセッション、T は TCP です。コンパクトロギングでは、ログサイズを平均 40% 削減できます。

次の NAT44 マッピング作成ログエントリの例は、コンパクトロギングの利点を示しています。

|-|-|
|Default logging format|02/02/2016:01:13:01 GMT Informational 0-PPE-2 : default LSN LSN_ADDRPORT_MAPPING 85 0 : A&PDM CREATED ClientIP:Port:TD1.1.1.1:6500:0,NatIP:NatPort8.8.8.8:47902, DestinationIP:Port:TD2.2.2.2:80:0, Protocol: TCP|
|Compact logging format|02/02/2016:01:14:57 GMT Info 0-PE2:default LSN 87 0:A&PDMC|C-1.1.1.1:6500:0|N-8.8.8.9:51066|D-2.2.2.2:80:0|T|

構成の手順

LSN 情報をコンパクト形式で記録するには、次のタスクを実行します。

- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、LSN 設定の情報をコンパクト形式で記録するかどうかを指定する Log Compact パラメータが含まれます。
- **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドします。Log Profile Name パラメータを作成した LSN ログプロファイル名に設定して、作成した LSN ログプロファイルを LSN 設定の LSN グループにバインドします。この LSN グループのすべてのセッションとマッピングは、コンパクト形式でログに記録されます。

CLI を使用して **LSN** ログプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn logprofile <logprofilename> -logCompact (ENABLED|DISABLED)

show lsn logprofile
<!--NeedCopy-->
```


CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

設定例:

```
add lsn logprofile  LOG-PROFILE-COMPACT-9 -logCompact ENABLED

Done
add lsn client LSN-CLIENT-9
Done
bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0
Done
add lsn pool LSN-POOL-9
Done
bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4
Done
add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9
Done
bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9
Done
bind lsn group  LSN-GROUP-9  - logProfileName LOG-PROFILE-COMPACT-9
Done
<!--NeedCopy-->
```

IPFIX ロギング

NetScaler アプライアンスは、LSN イベントに関する情報をインターネットプロトコルフロー情報エクスポート（IPFIX）形式で、構成済みの IPFIX コレクターのセットに送信することをサポートしています。アプライアンスは既存の AppFlow 機能を使用して、LSN イベントを IPFIX 形式で IPFIX コレクターに送信します。

IPFIX ベースのロギングは、次の大規模な NAT44 関連イベントに使用できます。

- LSN セッションの作成または削除。
- LSN マッピングエントリの作成または削除。
- デターミニスティック NAT におけるポートブロックの割り当てまたは割り当て解除
- ダイナミック NAT におけるポートブロックの割り当てまたは割り当て解除
- サブスクライバセッションクォータを超えると。

IPFIX ロギングを設定する前に考慮すべきポイント

IPSec ALG の設定を開始する前に、次の点を考慮してください。

- NetScaler ADC アプライアンスで AppFlow 機能および IPFIX コレクタを構成する必要があります。手順については、「AppFlow 機能の構成」トピックを参照してください。

構成の手順

LSN 情報を IPFIX 形式で記録するには、次のタスクを実行します。

- **AppFlow** 構成で **LSN** ロギングを有効にします。AppFlow 構成の一部として LSN ロギングパラメータを有効にします。
- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、IPFIX 形式のログ情報を有効または無効にする IPFIX パラメータが含まれます。
- **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドします。LSN ログプロファイルを 1 つまたは複数の LSN グループにバインドします。バインドされた LSN グループに関連するイベントは IPFIX 形式で記録されます。

CLI を使用して **AppFlow** 構成で **LSN** ロギングを有効にするには　コマンドプロンプトで入力します。

```
set appflow param -lsnLogging ( ENABLED | DISABLED )

show appflow param
<!--NeedCopy-->
```

CLIat を使用して **LSN** ログプロファイルを作成するには、コマンドプロンプトで　コマンドプロンプトで入力します。

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには　コマンドプロンプトで入力します。

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

GUI を使用して **LSN** ログプロファイルを作成するには　[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[ログ] タブをクリックして、ログプロファイルを追加します。

GUI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには

1. [システム] > [大規模 **NAT**] > [**LSN** グループ**]** に移動し、**LSN** グループを開きます。
2. 詳細設定で、**+ Log Profile** をクリックして、作成したログプロファイルを LSN グループにバインドします。

TCP SYN アイドルタイムアウト

August 15, 2023

SYN アイドルタイムアウトは、NetScaler アプライアンスで LSN を使用する TCP 接続を確立するためのタイムアウトです。設定したタイムアウト期間内に TCP セッションが確立されない場合、NetScaler はそのセッションを削除します。SYN アイドルタイムアウトは、SYN フラッド攻撃からの保護に役立ちます。LSN 設定では、LSN グループエンティティには SYN アイドルタイムアウト設定が含まれます。

例**:**

次の LSN 設定例では、192.0.2.0/24 ネットワークからの加入者に関連する TCP 接続の SYN アイドルタイムアウトが 30 秒に設定されています。

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  - synidletimeout 30

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done
<!--NeedCopy-->
```


負荷分散構成で **LSN** 構成を上書きする

August 15, 2023

デフォルトでは、LSN 設定はどのロードバランシング設定よりも優先されます。両方の設定に一致するトラフィックの負荷分散構成で大規模ネットワーク（LSN）構成をオーバーライドするには、Override LSN パラメータを有効にしたネットプロファイルを作成し、このプロファイルを負荷分散構成の仮想サーバーにバインドします。LSN 設定の LSN IP アドレスを適用する代わりに、ロードバランシング設定の USNIP または USIP 設定がトラフィックに適用されます。

このオプションは、NetScaler アプライアンスや付加価値サービス（ファイアウォールや最適化デバイスなど）を含む LSN 展開で役立ちます。このタイプの展開では、アプライアンス上の LSN 構成をトラフィックに適用する前に、NetScaler アプライアンス上の入力トラフィックがこれらの付加価値サービスを通過する必要があります。NetScaler アプライアンスが入力トラフィックを付加価値サービスに送信するには、負荷分散構成が作成され、アプライアンスで LSN のオーバーライドが有効になります。負荷分散構成には、ANY 型の仮想サーバーにバインドされた負荷分散サービスと呼ばれる付加価値サービスが含まれます。仮想サーバーには、付加価値サービスに送信されるトラフィックを識別するためのリッスンポリシーが設定されています。

CLI を使用してネットプロファイルの **lsn** のオーバーライドを有効にするには

ネットプロファイルの追加時に **override lsn** を有効にするには、コマンドプロンプトで次のように入力します

```
add netProfile <name> -overrideLsn ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

ネットプロファイルの追加時に **override lsn** を有効にするには、コマンドプロンプトで次のように入力します

```
set netProfile <name> -overrideLsn ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

GUI を使用してネットプロファイルの **lsn** をオーバーライドできるようにするには

1. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
2. ネットプロファイルを追加または変更するときに、**Override LSN** パラメータを設定します。

次の構成例では、ネットプロファイル NETPROFILE-OVERRIDELSN-1 で LSN オーバーライドオプションが有効になっており、負荷分散仮想サーバー LBVS-1 にバインドされています。

設定例:

```
add netprofile NETPROFILE-OVERRIDELSN-1 -overrideLsn ENABLED

Done

set lb vserver LBVS-1 -netprofile NETPROFILE-OVERRIDELSN-1

Done
<!--NeedCopy-->
```

LSN セッションを消去する

August 15, 2023

不要または非効率的な LSN セッションを NetScaler アプライアンスから削除できます。アプライアンスは、これらのセッションに割り当てられたリソース（NAT IP アドレス、ポート、メモリなど）をすぐに解放し、リソースを新しいセッションで使用できるようにします。アプライアンスは、削除されたセッションに関連する後続のパケットもすべてドロップします。NetScaler アプライアンスからすべてまたは選択した LSN セッションを削除できます。

コマンドラインインターフェイスを使用してすべての LSN セッションをクリアするには

コマンドプロンプトで入力します。

```
flush lsn session

show lsn session
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して選択的な LSN セッションをクリアするには

コマンドプロンプトで入力します。

```
flush lsn session [-clientname <string>] [-network <ip_addr> [-netmask
    <netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <
    port>]]

show lsn session
<!--NeedCopy-->
```

例

NetScaler に存在するすべての LSN セッションを消去します


```
flush lsn session

Done
<!--NeedCopy-->
```

LSN クライアントエンティティ LSN-CLIENT-1 に関連するすべての LSN セッションをクリアします

```
flush lsn session -clientname LSN-CLIENT-1

Done
<!--NeedCopy-->
```

トラフィックドメイン 100 に属する LSN クライアントエンティティ LSN-CLIENT-2 のサブスクライバネットワーク (192.0.2.0) に関連するすべての LSN セッションをクリアします

```
flush lsn session -clientname LSN-CLIENT-2 -network 192.0.2.0 -
    netmask 255.255.255.0 -td 100

Done
<!--NeedCopy-->
```

構成ユーティリティを使用してすべての **LSN** セッションをクリアするには

[システム] > [大規模 NAT] > [セッション] に移動し、[フラッシュセッション] をクリックします。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- フラッシュ LSN セッション
  - clientname

    LSN クライアントエンティティの名前。最大長：127
  - network

    加入者の IP アドレスまたはネットワークアドレス。
  - ネットマスク

    ネットワークパラメータで指定された IP アドレスのサブネットマスク。

    デフォルト値:255.255.255.255
  - td

    LSN クライアントエンティティのトラフィックドメイン ID。

    デフォルト値:0

    最小値:0

最大値:4094

- NaTiP

  LSN セッションで使用されるマップされた NAT IP アドレス。

- NAT ポート

  LSN セッションで使用されるマッピングされた NAT ポート。

SYSLOG サーバーの負荷分散

August 15, 2023

NetScaler ADC アプライアンスは、構成されたすべての外部ログサーバーに SYSLOG イベントとメッセージを送信します。その結果、冗長なメッセージが保存され、システム管理者の監視が困難になります。この問題に対処するために、NetScaler ADC アプライアンスは、メンテナンスとパフォーマンスを向上させるために外部ログサーバー間で SYSLOG メッセージを負荷分散できる負荷分散アルゴリズムを提供します。サポートされる負荷分散アルゴリズムには、ラウンドロビン、最小帯域幅、カスタムロード、最小接続、最小パケット、および AuditLogHash が含まれます。

コマンドラインインターフェイスを使用した SYSLOG サーバーの負荷分散

コマンドプロンプトで入力します。

サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

```
add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP |
    SYSLOGUDP)> <port>
<!--NeedCopy-->
```

負荷分散仮想サーバーを追加し、サービスタイプを SYSLOGTCP または SYSLOGTCP、負荷分散方法を AUDITLOGHASH として指定します。

```
add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
    <AUDITLOGHASH>]
<!--NeedCopy-->
```

サービスを負荷分散仮想サーバーにバインドします。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

1. SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つ負荷分散サーバー名を指定します。


```
add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
    <logLevel>]
<!--NeedCopy-->
```

ルールとアクションを指定して SYSLOG ポリシーを追加します。

```
add syslogpolicy <name> <rule> <action>
<!--NeedCopy-->
```

ポリシーを有効にするには、SYSLOG ポリシーをシステムグローバルにバインドします。

```
bind system global <policyName>
<!--NeedCopy-->
```

構成ユーティリティを使用した **SYSLOG** サーバーの負荷分散

1. サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

   [トラフィック管理] > [サービス] に移動し、[追加] をクリックして、プロトコルとして [SYLOGTCP] または [SYSLOGUDP] を選択します。

2. 負荷分散仮想サーバーを追加し、サービスタイプを SYSLOGTCP または SYSLOGTCP、負荷分散方法を AUDITLOGHASH として指定します。

   [トラフィック管理] > [仮想サーバー] に移動し、[追加] をクリックして、プロトコルとして SYLOGTCP または SYSLOGUDP を選択します。

3. サービスを負荷分散仮想サーバーに移行し、サービスに接続します。

   サービスを負荷分散仮想サーバーに接続します。

   [トラフィック管理] > [仮想サーバー] に移動し、仮想サーバーを選択してから、負荷分散方法で [Audit-LogHash] を選択します。

4. SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つ負荷分散サーバー名を指定します。

   [システム] > [監査] に移動し、[サーバー] をクリックし、[サーバー] の [LB Vserver] オプションを選択してサーバーを追加します。

5. ルールとアクションを指定して SYSLOG ポリシーを追加します。

   [システム] > [Syslog] に移動し、[ポリシー] をクリックして、SYSLOG ポリシーを追加します。

6. ポリシーを有効にするには、SYSLOG ポリシーをシステムグローバルにバインドします。

   [システム] > [Syslog] に移動し、SYSLOG ポリシーを選択して [アクション] をクリックし、[グローバルバインディング] をクリックして、ポリシーをシステムグローバルにバインドします。

例:

次の構成では、AUDITLOGHASH を負荷分散方法として使用して、外部ログサーバー間で SYSLOG メッセージの負荷分散を指定します。NetScaler ADC アプライアンスは、サービス、service1、service2、およびサービス 3 の間で負荷分散される SYSLOG イベントとメッセージを生成します。

```
add service service1 192.0.2.10 SYSLOGUDP 514

add service service2 192.0.2.11 SYSLOGUDP 514

add service service3 192.0.2.11 SYSLOGUDP 514

add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH

bind lb vserver lbvserver1 service1

bind lb vserver lbvserver1 service2

bind lb vserver lbvserver1 service3

add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All

add syslogpolicy syspol1 ns_true sysaction1

bind system global syspol1
<!--NeedCopy-->
```

制限事項:

NetScaler ADC アプライアンスは、ログサーバー間で SYSLOG メッセージを負荷分散する外部負荷分散仮想サーバーをサポートしていません。

ポート制御プロトコル

August 15, 2023

NetScaler アプライアンスは、大規模 NAT（LSN）用の Port Control Protocol（PCP）をサポートするようになりました。ISP の加入者アプリケーションの多くは、インターネットからアクセスできる必要があります (たとえば、インターネット経由で監視する IP カメラなどの Internet of Things (IOT) デバイス)。この要件を満たす 1 つの方法は、静的な大規模 NAT（LSN）マップを作成することです。しかし、加入者の数が非常に多い場合、スタティック LSN NAT マップを作成することは現実的な解決策ではありません。

Port Control Protocol（PCP）を使用すると、加入者は自分や他のサードパーティデバイス用の特定の LSN NAT マッピングを要求できます。大規模な NAT デバイスは LSN マップを作成し、サブスクライバに送信します。サブスクライバは、サブスクライバに接続できる NAT IP アドレス:NAT ポートをインターネット上のリモートデバイスに送信します。

アプリケーションは通常、LSN マッピングがタイムアウトしないように、大規模な NAT デバイスに頻繁にキープアライブメッセージを送信します。PCP は、アプリケーションが LSN マッピングのタイムアウト設定を学習できるようにすることで、このようなキープアライブメッセージの頻度を減らすのに役立ちます。これにより、ISP のアクセスネットワークの帯域幅消費とモバイルデバイスのバッテリー消費量を削減できます。

PCP はクライアント/サーバーモデルで、UDP トランスポートプロトコルで実行されます。NetScaler アプライアンスは PCP サーバーコンポーネントを実装し、RFC 6887 に準拠しています。

構成の手順

PCP を設定するには、次のタスクを実行します。

- (オプション) PCP プロファイルを作成します。PCP プロファイルには、PCP 関連パラメータ（マッピングやピアの PCP リクエストのリスニングなど）の設定が含まれます。PCP プロファイルは PCP サーバーにバインドできます。PCP サーバーにバインドされた PCP プロファイルは、そのすべての設定を PCP サーバーに適用します。PCP プロファイルは複数の PCP サーバーにバインドできます。デフォルトでは、デフォルトのパラメータ設定を持つ 1 つの PCP プロファイルがすべての PCP サーバーにバインドされます。PCP サーバーにバインドした PCP プロファイルは、そのサーバーのデフォルトの PCP プロファイル設定よりも優先されます。デフォルトの PCP プロファイルには、次のパラメータ設定があります。
  - マッピング: 有効
  - ピア: 有効
  - 最小マップ寿命:120 秒
  - 最大ライフタイム:86400 秒
  - アナウンス数:10
  - サードパーティ: 無効
- PCP サーバーを作成し、それに PCP プロファイルをバインドします。NetScaler アプライアンスに PCP サーバーを作成して、サブスクライバーからの PCP 関連のリクエストとメッセージを受信します。PCP サーバーにアクセスするには、サブネット IP (SNIP) アドレスを割り当てる必要があります。デフォルトでは、PCP サーバーはポート 5351 で受信します。
- PCP サーバーを LSN 構成の LSN グループにバインドします。作成した PCP サーバーを指定する PCP サーバーパラメーターを設定して、作成した PCP サーバーを LSN 構成の LSN グループにバインドします。作成された PCP サーバには、この LSN グループの加入者だけがアクセスできます。

> 注
>
> 大規模な NAT 設定の PCP サーバは、ACL ルールで識別されたサブスクライバからの要求には対応しません。

CLI を使用して **PCP** プロファイルを作成するには

コマンドプロンプトで入力します。


```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

CLI を使用して **PCP** サーバーを作成するには

コマンドプロンプトで入力します。

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```

NAT44 のサンプル構成

次の設定例では、PCP サーバの PCP-SERVER-9 はデフォルトの PCP 設定で LSN グループ LSN-GROUP-9 にバインドされています。PCP-SERVER-9 は、ネットワーク 192.0.2.0/24 内のサブスクライバーからの PCP 要求を処理します。

設定例:

```
add pcp server PCP-SERVER-9 192.0.3.9

Done

add lsn client LSN-CLIENT-9

Done

bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-9

Done

bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4

Done

add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

```



```
Done

bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done

bind lsn group  LSN-GROUP-9  -pcpServer PCP-SERVER-9

Done
<!--NeedCopy-->
```

クラスターセットアップの **LSN44**

August 15, 2023

NetScaler クラスターセットアップでは、大規模な NAT44 構成がサポートされます。

NetScaler クラスターは、単一のシステムとして構成および管理される NetScaler アプライアンスのグループです。NetScaler クラスターはスケーラビリティと可用性を提供します。クラスター構成の各 NetScaler アプライアンスは、独立した LSN エンティティとして機能し、単一のシステムとして管理されます。

クラスタ設定の LSN 設定は、特定の LSN IP アドレスプールが一度に 1 つのノードだけが所有する点を除いて、スタンドアロンアプライアンスと同じです。つまり、LSN IP プールエンティティは、特定のノードのスポットエンティティとして設定されます。クラスタ設定のすべてのノードには、特定の LSN IP プールエンティティを設定できます。LSN セッションに関連するパケットが NAT 操作を実行したのと同じクラスタノードで受信されるように、ポリシーベースのバックプレーン (PBS) ステアリングが設定されています。PBS は、LSN セッションの受信関連パケットを同じクラスタノードに転送します。

設定例:

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 -ownerNode 1 203.0.113.3

Done

bind lsn pool LSN-POOL-1 -ownerNode 2 203.0.113.3
```



```
18
19 Done
20
21 add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1
22
23 Done
24
25 bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
26
27 Done
28
29 add ns acl b1 ALLOW -srcIP = 192.0.2.0-192.0.2.255 -type DFD -dfdhash
      SIP
30
31
32 Done
33
34 apply ns acls -type DFD
35
36 Done
37 <!--NeedCopy-->
```

Dual-Stack Lite

August 15, 2023

IPv4 アドレスの不足と、IPv4 に対する IPv6 の優位性から、多くの ISP が IPv6 インフラストラクチャへの移行を開始しています。しかし、公衆インターネットのほとんどが依然として IPv4 のみを使用しており、多くの加入者が IPv6 をサポートしていないため、移行中も、ISP は IPv6 とともに IPv4 を引き続きサポートする必要があります。

デュアルスタックライト（DS-Lite）は、IPv6 インフラストラクチャを備えた ISP が IPv4 加入者をインターネットに接続するための IPv6 移行ソリューションです。DS-Lite は、IPv4-in-IPv6 トンネリングを使用して、IPv6 アクセスネットワーク上のトンネルを介して ISP に加入者の IPv4 パケットを送信します。IPv6 パケットはカプセル化を解除して加入者の IPv4 パケットを回復し、NAT アドレスとポート変換、その他の LSN 関連の処理を行った後にインターネットに送信されます。応答パケットは同じパスを経由してサブスクライバに到達します。

NetScaler アプライアンスは DS-Lite 環境の AFTR コンポーネントを実装しており、RFC 6333 に準拠しています。

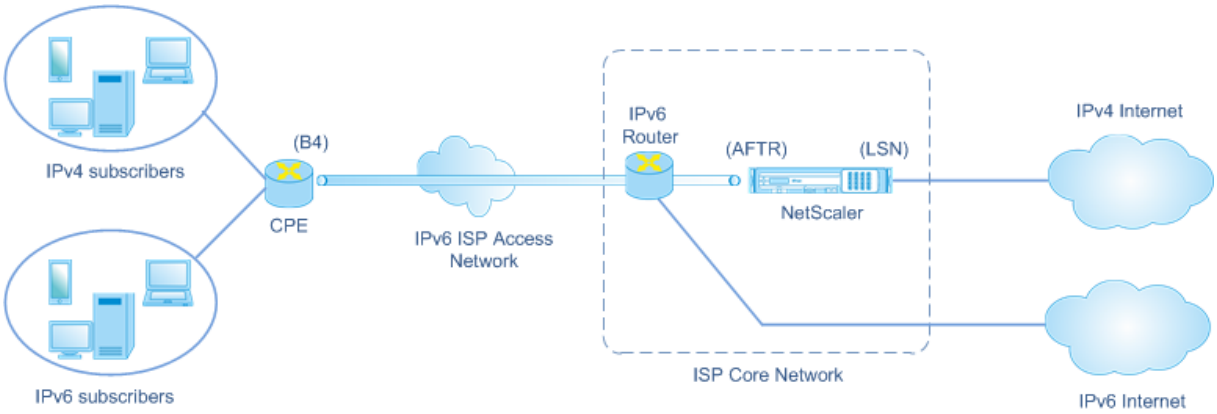
アーキテクチャ

ISP のデュアルスタック Lite アーキテクチャは、次のコンポーネントで構成されています。

- ベーシックブリッジングブロードバンド **(B4)**。ベーシックブリッジングブロードバンド（B4）は、加入者の敷地内にあるデバイスまたはコンポーネントです。通常、B4 は加入者施設内の CPE デバイスのコンポーネン

トです。IPv4 サブスクライバは、B4 コンポーネントを含む CPE デバイスを介して IPv6 のみの ISP アクセスネットワークに接続されます。B4 の主な機能は、B4 とアドレスファミリ移行ルータ（AFTR）の間で IPv6 トンネルを開始して、加入者の IPv4 要求または応答パケットをトンネル経由で送受信することです。B4 には、B4 トンネルエンドポイントアドレスと呼ばれる IPv6 アドレスが含まれます。B4 はこのアドレスを使用して IPv6 パケットを AFTR に送信し、AFTR からパケットを受信します。

- アドレスファミリ移行ルータ **(AFTR)**。AFTR は、ISP のコアネットワークに存在するデバイスまたはコンポーネントです。AFTR は B4 デバイスからの IPv6 トンネルを終了します。つまり、IPv6 トンネルは加入者構内の B4 と ISP コアネットワークの AFTR の間に形成されます。AFTR は B4 から受信した IPv6 パケットをカプセル化解除して、加入者の元の IPv4 パケットを復元します。AFTR は、IPv4 パケットを LSN デバイスまたはコンポーネントに送信します。LSN は、NAT アドレスとポート変換 (NAT 44)、その他の LSN 関連の処理を実行した後、IPv4 パケットを宛先にルーティングします。AFTR には、AFTR トンネルエンドポイントアドレスと呼ばれる IPv6 アドレスが含まれます。AFTR は、このアドレスを使用して IPv6 パケットを B4 に送信し、B4 から IPv6 パケットを受信します。NetScaler アプライアンスは AFTR コンポーネントを実装しています。
- ソフトワイヤー。B4 と AFTR の間に作成される IPv6 トンネルはソフトウェアワイヤと呼ばれます。



NetScaler アプライアンスを使用する ISP の DS-Lite アーキテクチャは、プライベートアドレス空間のサブスクライバーが、ISP のコアネットワークに展開された NetScaler アプライアンスを介してインターネットにアクセスする構成です。IPv4 サブスクライバは DS-Lite B4 機能を含む CPE デバイスに接続されます。CPE デバイスは、ISP の IPv6 専用アクセスネットワークを介して ISP コアネットワークに接続されます。NetScaler アプライアンスには、DS-Lite AFTR および LSN 機能が含まれています。

CPE デバイスに接続された IPv4 サブスクライバには、手動で、または CPE デバイス上で実行されている DHCP サーバを介してプライベート IPv4 アドレスが割り当てられます。CPE デバイスでは、AFTR トンネルのエンドポイントアドレスは手動で、または DHCPv6 を介して指定されます。CPE デバイスの構成はベンダーによって異なるため、このドキュメントの範囲外です。

IPv4 サブスクライバーからインターネット上の場所宛のリクエストパケットを受信すると、CPE デバイスの B4 コンポーネントは IPv4 パケットを IPv6 パケットにカプセル化し、ISP コアネットワーク内の NetScaler アプライアンスに送信します。NetScaler アプライアンスの AFTR 機能は、IPv6 パケットをカプセル化解除して、サブスクライバーの元の IPv4 パケットを復元します。NetScaler アプライアンスの LSN 機能は、IPv4 パケットの送信元 IP アドレスとポートを、構成済みの NAT プールから選択された NAT IP アドレスと NAT ポートに変換し、そのパケッ

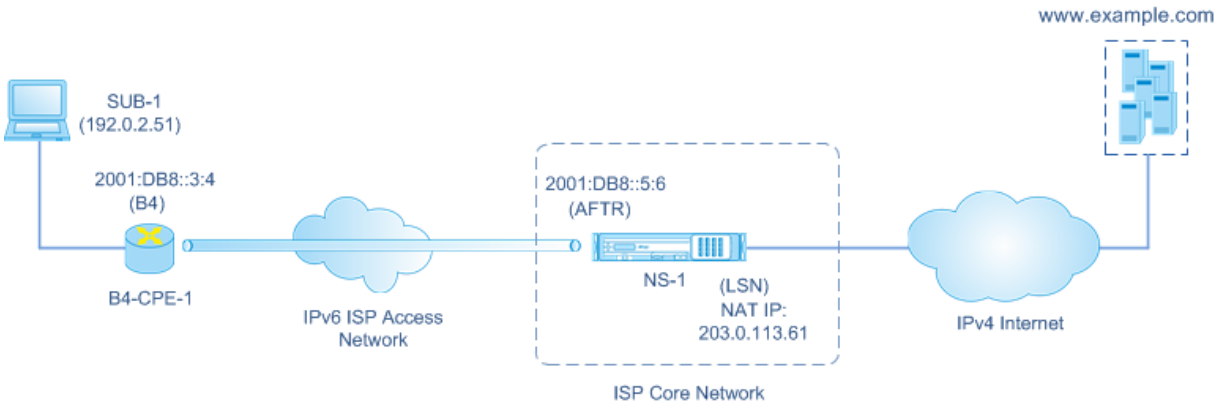
トをインターネット上の宛先に送信します。

アプライアンスは、AFTR および LSN 機能を使用するすべてのアクティブセッションの記録を保持します。これらのセッションは DS-Lite セッションと呼ばれます。NetScaler アプライアンスは、各 DS-Lite セッションの B4 IPv6 アドレス、サブスクライバーの IPv4 アドレスとポート、NAT IPv4 アドレスとポートの間のマッピングも管理します。これらのマッピングは DS-Lite LSN マッピングと呼ばれます。NetScaler アプライアンスは、DS-Lite セッションエントリと DS-LSN マッピングエントリから、（インターネットから受信した）応答パケットを特定の DS-Lite セッションに属するものとして認識します。

NetScaler アプライアンスが特定の DS-Lite セッションに属する応答パケットを受信すると、アプライアンスの LSN 機能は応答パケットの宛先 IP アドレスとポートを NAT IP アドレスとポートからサブスクライバーの IP アドレスとポートに変換し、AFTR 機能は結果のパケットを IPv6 パケットにカプセル化して CPE デバイスに送信します。CPE デバイスの B4 機能は、IPv6 パケットをカプセル化解除して IPv4 応答パケットを回復してから、IPv4 パケットを加入者に送信します。

例

ISP のコアネットワークにある NetScaler NS-1、サブスクライバープレミスの CPE デバイス B4-CPE-1、および単一の IPv4 サブスクライバー SUB-1 で構成される DS-Lite 環境の例を考えてみましょう。B4-CPE-1 は DS-Lite 機能の B4 機能をサポートしています。



次の表に、この例で使用されている設定の一覧を示します。

エンティティ	名前	詳細
サブスクライバ SUB-1 の IPv4 アドレス		192.0.2.51
B4 デバイス上のソフトウェアワイヤエンドポイントの IPv6 アドレス（B4-CPE-1）		2001:DB8::3:4
AFTR デバイス上のソフトウェア・ワイヤ・エンドポイントの IPv6 アドレス（NS-1）		2001:DB8::5:6

NetScaler アプライアンス **NS-1** の設定**:**

エンティティ	名前	詳細
LSN クライアント	LSN-DSLITE-CLIENT-1	Network6 (Identifying traffic from B4 devices) = 2001:DB8::3:0/100
LSN プール	LSN-DSLITE-POOL-1	LSN IP (NAT IP) = 203.0.113.61-203.0.113.70
IPv6 プロファイル	LSN-DSLITE-PROFILE-1	タイプ = DS-LITE; IPv6 アドレス (IPv6 アドレスの後) = NetScaler が所有する IPv6 タイプの IPv6 アドレスの 1 つ = 2001: DB8:: 5:6
LSN グループ	LSN-DSLITE-GROUP-1	LSN クライアント = LSN-DSLITE-CLIENT-1; LSN プール = LSN-DSLITE-POOL-1; IPv6 プロファイル = LSN-DSLITE-PROFILE-1

次に、この例のトラフィックフローを示します。

1. IPv4 サブスクライバ SUB-1 は (http://www.example.com/) に要求を送信します。IPv4 パケットには次のものが含まれます。
   - ソース IP アドレス = 192.0.2.51
   - 送信元ポート = 2552
   - 宛先 IP アドレス = 198.51.100.250
   - 宛先ポート = 80
2. IPv4 要求パケットを受信すると、B4-CPE-1 はそれを IPv6 パケットのペイロードにカプセル化し、IPv6 パケットを NS-1 に送信します。IPv6 パケットには次のものが含まれます。
   - ソース IP アドレス = 2001: DB8:: 3:4
   - ターゲット IP アドレス = 2001: DB8:: 5:6
3. NS-1 が IPv6 パケットを受信すると、AFTR モジュールは IPv6 ヘッダーを削除してパケットのカプセル化を解除します。結果のパケットは、SUB-1 の元の IPv4 要求パケットです。
4. NS-1 の LSN モジュールは、パケットの送信元 IP アドレスとポートを、設定された NAT プールから選択された NAT IP アドレスと NAT ポートに変換します。変換された IPv4 パケットには次の内容が含まれます。
   - ソース IP アドレス = 203.0.113.61
   - 送信元ポート = 3002
   - 宛先 IP アドレス = 198.51.100.250
   - 宛先ポート = 80

5. LSN モジュールは、この DS Lite セッションの LSN マッピングとセッションエントリも作成します。マッピングには次の情報が含まれます。

   - IPv6 パケットの送信元 IP アドレス (B4-CPE-1 の IPv6 アドレス) = 2001: DB8:: 3:4
   - IPv4 パケットの送信元 IP アドレス (サブワンの IPv4 アドレス) = 192.0.2.51
   - IPv4 パケットの送信元ポート = 2552
   - NAT IP アドレス = 203.0.113.61
   - NAT ポート = 3002

6. NS-1 は、生成された IPv4 パケットをインターネット上の宛先に送信します。
7. www.example.com のサーバーはリクエストパケットを処理し、レスポンスパケットを送信します。IPv4 応答パケットには次の内容が含まれます。

   - ソース IP アドレス = 198.51.100.250
   - 送信元ポート = 80
   - 宛先 IP アドレス = 203.0.113.61
   - デスティネーションポート = 3002

8. IPv4 パケットを受信すると、NS-1 は LSN マッピングとセッションエントリを調べ、IPv4 応答パケットが DS Lite セッションに属していることを確認します。NS-1 の LSN モジュールは、宛先 IP アドレスとポートを変換します。IPv4 パケットには次のものが含まれます。

   - ソース IP アドレス = 198.51.100.250
   - 送信元ポート = 80
   - ターゲット IP アドレス = 192.0.2.51
   - デスティネーションポート = 2552

9. NS-1 の AFTR モジュールは、IPv4 パケットを IPv6 パケットにカプセル化し、IPv6 パケットを B4-CPE-1 に送信します。IPv6 パケットには次のものが含まれます。

   - ソース IP アドレス = 2001: DB8:: 5:6
   - ターゲット IP アドレス = 2001: DB8:: 3:4

10. パケットを受信すると、B4-CPE-1 は IPv6 ヘッダーを削除して IPv6 パケットをカプセル化解除し、生成された IPv4 パケットを CL-1 に送信します。

DS-Lite を構成する前の考慮事項

August 15, 2023

NetScaler アプライアンスで DS-Lite を構成する前に、次の点を考慮してください。

1. RFC 6333 で説明されている DS-Lite のさまざまなコンポーネントを理解する必要があります。
2. NetScaler アプライアンスの DS-Lite 構成では、LSN コマンドセットを使用します。DS-Lite 構成では、LSN クライアントエンティティは、B4 デバイスからのトラフィックを識別するための IPv6 アドレス、IPv6 ネットワークアドレス、または ACL6 ルールを指定します。DS-Lite 構成には、NetScaler ADC アプライアンス上の IPv6 アドレスの AFTR コンポーネントを指定する IPv6 プロファイルも含まれています。NetScaler LSN 機能の詳細については、「大規模 NAT」を参照してください。
3. DS-Lite 構成の場合、NetScaler ADC アプライアンスは、次のいずれかのプロトコルに属する IPv4 パケットに対して LSN をサポートします。NetScaler アプライアンスは、他のプロトコルに属する IPv4 パケットをドロップします。
    - TCP
    - UDP
    - ICMP
4. NetScaler アプライアンスは以下の ALG DS-Lite をサポートしています。
    - ICMP
    - FTP
    - TFTP
    - セッション開始プロトコル (SIP)
    - リアルタイムストリーミングプロトコル (RTSP)

DS-Lite の構成

August 15, 2023

NetScaler アプライアンスの DS-Lite 構成では、LSN コマンドセットを使用します。DS-Lite 構成では、LSN クライアントエンティティは、B4 デバイスからのトラフィックを識別するための IPv6 アドレス、IPv6 ネットワークアドレス、または ACL6 ルールを指定します。NetScaler LSN 機能の詳細については、「大規模 NAT」を参照してください。DS-Lite 構成には、NetScaler ADC アプライアンス上の DS-Lite AFTR コンポーネントの IPv6 アドレス（SNIP6 タイプ）を指定する IPv6 プロファイルも含まれています。

NetScaler アプライアンスでの DS-Lite の構成は、次のタスクで構成されます。

- グローバル **LSN** パラメータを設定します。グローバルパラメータには、LSN 機能用に確保されている NetScaler メモリの量と、高可用性セットアップでの LSN セッションの同期が含まれます。
- **B4 CPE** デバイスからのトラフィックを識別するための **LSN** クライアントエンティティを作成します。LSN クライアントエンティティは DS-Lite B4 デバイスのセットを指します。クライアントエンティティには、これらの B4 デバイスからのトラフィックを識別するための IPv6 アドレス、IPv6 ネットワークアドレス、または ACL6 ルールが含まれます。LSN クライアントは 1 つの LSN グループにしかバインドできません。コマン

ドラインインターフェイスには、LSN クライアントエンティティを作成し、サブスクライバーを LSN クライアントエンティティにバインドする 2 つのコマンドがあります。設定ユーティリティは、これら 2 つの操作を 1 つの画面にまとめます。

- **LSN** プールを作成し、**NAT IP** アドレスをそのプールにバインドします。LSN プールは、NetScaler アプライアンスが LSN を実行するために使用する NAT IP アドレスのプールを定義します。コマンドラインインターフェイスには、LSN プールを作成し、NAT IP アドレスを LSN プールにバインドする 2 つのコマンドがあります。設定ユーティリティは、これら 2 つの操作を 1 つの画面にまとめます。
- **LSN IP6** プロファイルを作成します。LSN IP6 プロファイルは、NetScaler アプライアンス上の DS-Lite AFTR コンポーネントの IPv6 アドレスを定義します。IPv6 アドレスは、NetScaler が所有する SNIP6 タイプの IPv6 アドレスのいずれかである必要があります。
- **(**オプション**)** 指定したプロトコルの **LSN** トランスポートプロファイルを作成します。LSN トランスポートプロファイルは、加入者が特定のプロトコルで使用できる LSN セッションの最大数や最大ポート使用量など、さまざまなタイムアウトと制限を定義します。各プロトコル (TCP、UDP、ICMP) の LSN トランスポートプロファイルを LSN グループにバインドします。プロファイルは複数の LSN グループにバインドできます。LSN グループにバインドされたプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、TCP、UDP、および ICMP プロトコルのデフォルト設定の 1 つの LSN トランスポートプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトトランスポートプロファイルと呼ばれます。LSN グループにバインドした LSN トランスポートプロファイルは、そのプロトコルのデフォルトの LSN トランスポートプロファイルよりも優先されます。
- （オプション）指定したプロトコルの **LSN** アプリケーションプロファイルを作成し、宛先ポートのセットをそれにバインドします。LSN アプリケーションプロファイルは、特定のプロトコルと宛先ポートセットに対するグループの LSN マッピングと LSN フィルタリング制御を定義します。宛先ポートセットの場合、各プロトコル（TCP、UDP、ICMP）の LSN プロファイルを LSN グループにバインドします。プロファイルは複数の LSN グループにバインドできます。LSN グループにバインドされた LSN アプリケーションプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、すべての宛先ポートの TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN アプリケーションプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトアプリケーションプロファイルと呼ばれます。指定された宛先ポートセットの LSN アプリケーションプロファイルを LSN グループにバインドすると、バインドされたプロファイルは、その宛先ポートセットでそのプロトコルのデフォルトの LSN アプリケーションプロファイルをオーバーライドします。コマンドラインインターフェイスには、LSN アプリケーションプロファイルを作成し、宛先ポートのセットを LSN アプリケーションプロファイルにバインドするための 2 つのコマンドがあります。設定ユーティリティは、これら 2 つの操作を 1 つの画面にまとめます。
- **LSN** グループを作成し、**LSN** プール、**LSN IPv6** プロファイル、**(**オプション**) LSN** トランスポートプロファイル、**(**オプション**) LSN** アプリケーションプロファイルを **LSN** グループにバインドします。LSN グループは、LSN クライアント、LSN IPv6 プロファイル、LSN プール、LSN トランスポートプロファイル、および LSN アプリケーションプロファイルで構成されるエンティティです。グループには、ポートブロックサイズや LSN セッションのロギングなどのパラメータが割り当てられます。パラメータ設定は、LSN グループにバ

インドされた LSN クライアントのすべてのサブスクライバに適用されます。LSN グループにバインドできる LSN IPv6 プロファイルは 1 つだけで、LSN グループにバインドされた LSN IPv6 プロファイルは他の LSN グループにバインドできません。バインドできるのは、同じ NAT タイプ設定の LSN プールと LSN グループだけです。複数の LSN プールを 1 つの LSN グループにバインドできます。LSN グループにバインドできる LSN クライアントエンティティは 1 つだけで、LSN グループにバインドされた LSN クライアントエンティティは他の LSN グループにバインドできません。コマンドラインインターフェイスには、LSN グループを作成し、LSN プール、LSN トランスポートプロファイル、および LSN アプリケーションプロファイルを LSN グループにバインドするための 2 つのコマンドがあります。設定ユーティリティは、これら 2 つの操作を 1 つの画面にまとめます。

コマンドラインを使った設定

コマンドラインインターフェイスを使用して **LSN** クライアントを作成するには**:**

コマンドプロンプトで入力します。

```
add lsn client <clientname>

show lsn client
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **IPv6** ネットワークまたは **ACL6** ルールを **LSN** クライアントにバインドするには**:**

コマンドプロンプトで入力します。

```
bind lsn client <clientname> (-network6 <ipv6_addr|*>| -acl6name <
   string>)

show lsn client
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** プールを作成するには**:**

コマンドプロンプトで入力します。

```
add lsn pool <poolname> [-nattype ( DYNAMIC )] [-portblockallocation (
   ENABLED | DISABLED )] [-portrealloctimeout <secs>] [-
   maxPortReallocTmq <positive_integer>]

show lsn pool
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **IP** アドレス範囲を **LSN** プールにバインドするには**:**

コマンドプロンプトで入力します。

```
bind lsn pool <poolname> <lsnip>

```



```
show lsn pool
<!--NeedCopy-->
```

注**:** LSN プールから LSN IP アドレスを削除するには、unbind lsn pool コマンドを使用してください。

コマンドラインインターフェイスを使用して **LSN IPv6** プロファイルを設定するには:

コマンドプロンプトで入力します。

```
add lsn ip6profile <name>  -type DS-Lite  -network6 < ipv6_addr|*s >

show lsn ip6profile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** トランスポートプロファイルを作成するには**:**

コマンドプロンプトで入力します。

```
add lsn transportprofile <transportprofilename> <transportprotocol> [-
    sessiontimeout <secs>] [-finrsttimeout <secs>] [-portquota <
    positive_integer>] [-sessionquota <positive_integer>] [-
    portpreserveparity ( ENABLED | DISABLED )] [-portpreserverange (
    ENABLED | DISABLED )] [-syncheck ( ENABLED | DISABLED )]

show lsn transportprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** アプリケーションプロファイルを作成するには**:**

コマンドプロンプトで入力します。

```
add lsn appsprofile <appsprofilename> <transportprotocol> [-ippooling (
    PAIRED | RANDOM )] [-mapping <mapping>] [-filtering <filtering>][-
    tcpproxy ( ENABLED | DISABLED )] [-td <positive_integer>]

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してアプリケーションプロトコルのポート範囲を **LSN** アプリケーションプロファイルにバインドするには**:**

コマンドプロンプトで入力します。

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** グループを作成するには**:**

コマンドプロンプトで入力します。


```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC )]
    [-portblocksize <positive_integer>] [-logging (ENABLED | DISABLED )]
     [-sessionLogging ( ENABLED | DISABLED )][-sessionSync ( ENABLED |
    DISABLED )] [-snmptraplimit<positive_integer>] [-ftp ( ENABLED |
    DISABLED )] [-pptp ( ENABLED |DISABLED )] [-sipalg ( ENABLED |
    DISABLED )] [-rtspalg ( ENABLED |DISABLED )] [-ip6profile <string>]

show lsn group
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** プロトコルプロファイルと **LSN** プールを **LSN** グループにバインドするには**:**

コマンドプロンプトで入力します。

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -httphdrlogprofilename <string> | -appsprofilename <
    string> | -sipalgprofilename <string> | rtspalgprofilename <string>)

show lsn group
<!--NeedCopy-->
```

構成ユーティリティによる構成

設定ユーティリティを使用して **LSN** クライアントを設定し、**IPv6** ネットワークアドレスまたは **ACL6** ルールをバインドするには**:**

[システム] > [大規模 **NAT**] > [クライアント] に移動し、クライアントを追加してから、IPv6 ネットワークアドレスまたは ACL6 ルールをクライアントにバインドします。

設定ユーティリティを使用して **LSN** プールを設定し、**NAT IP** アドレスをバインドするには:

[システム] > [大規模 **NAT**] > [プール] に移動し、プールを追加して、NAT IP アドレスまたは NAT IP アドレスの範囲をプールにバインドします。

構成ユーティリティを使用して **LSN IPv6** プロファイルを構成するには:

[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[**IPv6**] タブをクリックして、DS-Lite AFTR に IPv6 アドレスを割り当てます。

設定ユーティリティを使用して **LSN** トランスポートプロファイルを設定するには:

1. [システム] > [大規模 **NAT**] > [プロファイル] に移動します。
2. 詳細ウィンドウで [トランスポート**]** をクリックし、トランスポートプロファイルを追加します。

設定ユーティリティを使用して **LSN** アプリケーションプロファイルを設定するには:

1. [システム] > [大規模 **NAT**] > [プロファイル] に移動します。
2. 詳細ペインで [アプリケーション] をクリックし、アプリケーションプロファイルを追加します。

設定ユーティリティを使用して **LSN** グループを設定し、**LSN** クライアント、**LSN IPv6** プロファイル、プール、トランスポートプロファイル、およびアプリケーションプロファイルをバインドするには:

[システム] > **[**大規模 **NAT]** > [グループ] に移動し、グループを追加してから、LSN クライアント、LSN IPv6 プロファイル、プール、トランスポートプロファイル、およびアプリケーションプロファイルをグループにバインドします。

```
> add lsn client LSN-DSLITE-CLIENT-1
Done
> bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
Done
> add lsn pool LSN-DSLITE-POOL-1
Done
> bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
> add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
   DB8::5:6
Done
> add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
   portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1
Done
> add lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
Done
```

DS-Lite のログ記録と監視

DS-Lite の情報を記録して、問題の診断やトラブルシューティング、法的要件を満たすことができます。NetScaler ADC アプライアンスは、DS-Lite 情報をログに記録するための LSN ロギング機能をすべてサポートしています。DS-Lite ロギングを構成するには、「LSN のロギングとモニタリング」で説明した LSN ロギングを設定する手順を使用します。

DS-Lite LSN マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイムスタンプ
- エントリータイプ (マッピング)
- DS-Lite LSN マッピングエントリが作成されたか削除されたか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が表示される場合があります。
  - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートは記録されません。

  - アドレス依存マッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレス-ポート依存マッピング用にログに記録されます。

DS-Lite セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイムスタンプ
- エントリータイプ (セッション)
- DS-Lite セッションが作成されるか削除されるか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

次の表は、設定されたログサーバに保存されている各タイプのサンプル DS-Lite ログエントリを示しています。これらのログエントリは、NSIP アドレスが 10.102.37.115 の NetScaler アプライアンスによって生成されます。DS-Lite 情報をログに記録して、問題の診断やトラブルシューティング、法的要件を満たすことができます。NetScaler ADC アプライアンスは、DS-Lite 情報をログに記録するための LSN ロギング機能をすべてサポートしています。DS-Lite ロギングを構成するには、「LSN のロギングとモニタリング」で説明した LSN ロギングを設定する手順を使用します。

DS-Lite LSN マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイムスタンプ
- エントリータイプ (マッピング)
- DS-Lite LSN マッピングエントリが作成されたか削除されたか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が表示される場合があります。
  - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートは記録されません。
  - アドレス依存マッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレス-ポート依存マッピング用にログに記録されます。

DS-Lite セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）

- タイムスタンプ
- エントリータイプ (セッション)
- DS-Lite セッションが作成されるか削除されるか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

次の表は、設定されたログサーバに保存されている各タイプのサンプル DS-Lite ログエントリを示しています。これらのログエントリは、NSIP アドレスが 10.102.37.115 の NetScaler アプライアンスによって生成されます。

LSN ログエントリタイプ	サンプルログエントリ
DS-Lite セッション作成	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP
DS-Lite セッション削除	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP
DS-Lite LSN マッピングの作成	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

DS-Lite LSN マッピングの削除	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

現在の **DS-Lite** セッションの表示

現在の DS-Lite セッションを表示して、NetScaler アプライアンス上の不要なセッションや非効率的なセッションを検出できます。選択パラメータに基づいて、すべてまたは一部の DS-Lite セッションを表示できます。

コマンドラインインターフェイスを使用した設定

コマンドラインインターフェイスを使用してすべての **DS-Lite** セッションを表示するには**:**

コマンドプロンプトで入力します。

```
show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して選択した **DS-Lite** セッションを表示するには**:**

コマンドプロンプトで入力します。

```
show lsn session  -nattype DS-Lite [-clientname <string>] [-network <
   ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
   ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

例**:**

次のサンプル出力は、NetScaler アプライアンスに存在するすべての DS-Lite セッションを示しています。

```
show lsn session  -nattype DS-Lite
  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP
     NatPort Proto Dir

1.  2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61
   3002 TCP OUT

2.  2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61
   52862 TCP OUT

```



```
3.  2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61
    48116 ICMP OUT

4.  2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69
    48305 TCP OUT

Done
<!--NeedCopy-->
```

設定ユーティリティを使った設定

構成ユーティリティを使用してすべての DS-Lite セッションまたは選択した DS-Lite セッションを表示するには

1. [システム] **>** [****** 大規模 **NAT**] > [セッション] に移動し、**[**DS-Lite**] タブをクリックします。
2. 選択パラメータに基づいて **DS-Lite** セッションを表示するには、「検索」をクリックします。

DS-Lite セッションのクリア

不要または非効率的な DS-Lite セッションを NetScaler アプライアンスから削除できます。アプライアンスは、これらのセッションに割り当てられたリソース（NAT IP アドレス、ポート、メモリなど）をすぐに解放し、リソースを新しいセッションで使用できるようにします。アプライアンスは、削除されたセッションに関連する後続のパケットもすべてドロップします。NetScaler アプライアンスからすべてまたは選択した DS-Lite セッションを削除できます。

コマンドラインインターフェイスを使用してすべての **DS-Lite** セッションをクリアするには**:**

コマンドプロンプトで入力します。

```
flush lsn session -nattype DS-Lite
show lsn session -nattype DS-Lite
```

コマンドラインインターフェイスを使用して選択した **DS-Lite** セッションをクリアするには**:**

コマンドプロンプトで入力します。

```
flush lsn session  - nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]

show lsn session  - nattype DS-Lite
<!--NeedCopy-->
```

設定ユーティリティを使用してすべての **DS-Lite** セッションまたは選択した **DS-Lite** セッションをクリアするには:

1. [システム] > [大規模 **NAT**] > [セッション] に移動し、[**DS-Lite**] タブをクリックします。
2. 「フラッシュセッション」をクリックします。

DS-Lite 静的マップの構成

August 15, 2023

NetScaler アプライアンスは、次の情報間のマッピングを含む DS-Lite LSN マッピングの手動作成をサポートしています。

- 加入者の IP アドレスとポート、および B4 デバイスまたはコンポーネントの IPv6 アドレス
- NAT IP アドレスとポート

スタティック DS-Lite LSN マッピングは、NAT IP アドレスとポートに対して開始された接続を、指定した B4 デバイス（内部ネットワークにある Web サーバなど）を介して加入者の IP アドレスとポートに確実にマップしたい場合に便利です。

注: この機能は、リリース 11.0 ビルド 64.x 以降でサポートされています。

コマンドラインを使用して DS-Lite スタティック LSN マッピングを作成するには

コマンドプロンプトで入力します。

```
add lsn static <name> <transportprotocol> <subscrIP> <subscrPort> [-td
    <positive_integer>] [-network6 <B4_ADDR>] [<natIP> [<natPort>]] [-
    destIP<ip_addr> [-dsttd <positive_integer>]]

show lsn static
<!--NeedCopy-->
```

パラメータの説明

add lsn static

- name

LSN スタティックマッピングエントリの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。LSN グループの作成後は変更できません。次の要件は CLI にのみ適用されます。名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲んでください (たとえば、「ds-lite lsn static1」や「ds-lite lsn static1」)。これは必須の議論です。最大長：127

- transportprotocol

DS-Lite LSN マッピングエントリのプロトコル。

- subscrIP

DS-Lite LSN マッピングエントリのサブスクライバの IPv4 アドレス。

- subscrPort

DS-Lite LSN マッピングエントリのサブスクライバのポート。

- Network6

B4 デバイスまたはコンポーネントの IPv6 アドレス。

- td

B4 デバイスが属するトラフィックドメインの ID。B4 デバイスの IPv6 アドレスは network6 パラメータで指定されます。ID を指定しない場合、B4 デバイスはデフォルトのトラフィックドメインの一部と見なされます。

- NaTiP

このマッピングエントリの NAT IP アドレスとして使用される IPv4 アドレスは、NetScaler アプライアンスに LSN タイプとしてすでに存在しています。

- natPort

この DS-Lite LSN マッピングエントリの NAT ポート。

- destIP

DS-Lite LSN マッピングエントリの宛先 IP アドレス。

- dsttd

NetScaler アプライアンスからこの DS-Lite LSN マッピングエントリの宛先 IP アドレスにアクセスできるトラフィックドメインの ID。ID を指定しない場合、宛先 IP アドレスには ID が 0 のデフォルトのトラフィックドメインを介して到達可能であると見なされます。

構成ユーティリティを使用して **DS-Lite** 静的 **LSN** マッピングを作成するには

[システム] > [大規模 NAT] > [スタティック] に移動し、新しい DS-Lite スタティック LSN マッピングを追加します。

DS-Lite 用の確定的 **NAT** 割り当ての構成

August 15, 2023

DS-Lite LSN 展開における確定的 NAT 割り当ては、NetScaler アプライアンスが LSN NAT IP プールから、指定されたポートブロックサイズに基づいて、LSN NAT IP アドレスとポートブロックを各サブスクライバー（B4 デバイスの背後にあるサブスクライバー）に事前に割り当てる NAT リソース割り当ての一種です。

注**:** この機能は、リリース 11.0 ビルド 64.x 以降でサポートされています。

アプライアンスは、これらのサブスクライバに NAT リソースを順番に割り当てます。最初の NAT IP アドレスのポートの最初のブロックを最初のサブスクライバ IP アドレスに割り当てます。次の範囲のポートが次のサブスクライバに割り当てられ、NAT アドレスに次のサブスクライバ用のポートが足りなくなるまで続きます。その時点で、次の NAT アドレスの最初のポートブロックが加入者に割り当てられ、以降も同様に加入者に割り当てられます。

NetScaler アプライアンスは、サブスクライバーに割り当てられた NAT IP アドレスとポートブロックを記録します。接続の場合、マッピングされた NAT IP アドレスとポートブロックだけで加入者を識別できます。このため、NetScaler アプライアンスは LSN セッションの作成または削除を記録しません。

DS-Lite サブスクライバに設定できるデターミニスティック・ポート・ブロックは 1 つだけです。ポートブロック全体が使用されている場合、NetScaler アプライアンスはサブスクライバーからの新しい接続をすべて切断します。

例**:** デターミニスティック **DS-Lite**

この例では、デターミニスティック DS-Lite 構成には、IP アドレスが 192.0.17.5、192.0.17.6、192.0.17.7、および 192.0.17.8 の 4 人のサブスクライバが含まれています。これらの IPv4 サブスクライバは、IPv6 アドレス 2001:: DB8:: 3:4 の B4 デバイスの背後にいます。この設定では、ポートのブロックサイズは 20480 に設定され、LSN NAT IP アドレスプールの IP アドレスは 203.0.113.41-203.0.113.42 の範囲にあります。

NetScaler アプライアンスは、LSN NAT IP プールから、設定されたポートブロックサイズに基づいて、LSN NAT IP アドレスとポートブロックを各サブスクライバーに順番に事前割り当てします。最初の NAT IP アドレス（203.0.113.41）の最初のポートブロック（1024-21503）を、最初のサブスクライバの IP アドレス（192.0.17.5）に割り当てます。次の範囲のポートが次のサブスクライバに割り当てられ、NAT アドレスに次のサブスクライバ用のポートが足りなくなるまで続きます。その時点で、次の NAT IP アドレスの最初のポートブロックが加入者に割り当てられ、以降も同様に加入者に割り当てられます。NetScaler は、各サブスクライバーに割り当てられた NAT IP アドレスとポートブロックを記録します。

NetScaler アプライアンスは、これらのサブスクライバーに対して作成または削除された LSN セッションを記録しません。

次の表は、この例で各サブスクライバに割り当てられた NAT IP アドレスとポートブロックの一覧です。

サブスクライバ IP アドレス	割り当てられた NAT IP アドレス	割り当てられたポートブロック	B4 の IPv6 アドレス
192.0.17.5	203.0.113.41	1024 - 21503	2001:DB8::3:4
192.0.17.6	203.0.113.41	21504 - 41983	2001:DB8::3:4

192.0.17.7	203.0.113.41	41984 - 62463	2001:DB8::3:4
192.0.17.8	203.0.113.42	1024 - 21503	2001:DB8::3:4

構成の手順

DS-Lite 設定の一部として Deterministic NAT を設定する必要があります。DS-Lite の設定手順については、DS-Lite の設定を参照してください。

DS-Lite の設定時には、次のことを確認してください。

- LSN プールと LSN グループを追加するときは、NAT タイプパラメータを Deterministic に設定します。
- デフォルト値をそのまま使用できない限り、LSN グループを追加するときに必要なポートブロックサイズパラメーターを設定します。

デターミニスティック **DS-Lite** を設定する前に考慮すべきポイント

デターミニスティック DS-Lite を設定する前に、次の点を考慮してください。

- 各サブスクライバの完全な IP アドレスは、別の add lsn client コマンドで、Network パラメータと Netmask パラメータを設定して指定する必要があります。(ネットマスクを 255.255.255.255 に設定してください。) また、Network6 パラメータで指定した B4 デバイスの IPv4 アドレスは完全でなければなりません (/128 プレフィックス)。つまり、Network パラメータと Network6 パラメータは、それぞれ/32 ビットマスクと/128 プレフィックス以外のアドレスを受け付けません。
- NetScaler アプライアンスは、確定的 DS-Lite 構成では指定されていないが、確定的 DS-Lite 構成で指定された B4 デバイスの背後にあるサブスクライバーからの接続を切断します。
- NetScaler アプライアンスは、同じ IPv4 アドレスを持つサブスクライバーが異なる B4 デバイスの背後にいる場合、異なるサブスクライバーと認識します。サブスクライバの IPv4 アドレスと B4 デバイスの組み合わせにより、DS-Lite 構成の LSN クライアントエンティティに固有のサブスクライバが定義されます。

デターミニスティック **DS-Lite** 構成の例**:**

次の構成では、「例: 決定的 DS-Lite」セクションに記載されている設定を使用しています。

```
add lsn client LSN-DSLITE-CLIENT-10

Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.5 -netmask
   255.255.255.255 -network6 2001:DB8::3:4/128

Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.6 -netmask
   255.255.255.255 -network6 2001:DB8::3:4/128
```



```
8
9  Done
10 bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.7 -netmask
       255.255.255.255 -network6 2001:DB8::3:4/128
11
12 Done
13 bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.8 -netmask
       255.255.255.255 -network6 2001:DB8::3:4/128
14
15 Done
16 add lsn pool LSN-DSLITE-POOL-10 -nattype DETERMINISTIC
17
18 Done
19 bind lsn pool LSN-DSLITE-POOL-10  203.0.113.41-203.0.113.42
20
21 Done
22 add lsn ip6profile LSN-DSLITE-PROFILE-10 -type DS-Lite -network6 2001:
       DB8::5:6
23
24 Done
25 add lsn group LSN-DSLITE-GROUP-10 -clientname LSN-DSLITE-CLIENT-10 -
       nattype DETERMINISTIC -portblocksize 20480 -ip6profile LSN-DSLITE-
       PROFILE-10
26
27 Done
28 bind lsn group LSN-DSLITE-GROUP-10 -poolname LSN-DSLITE-POOL-10
29
30 Done
31 <!--NeedCopy-->
```

DS-Lite 用のアプリケーションレイヤーゲートウェイの構成

August 15, 2023

一部のアプリケーション層プロトコルでは、IP アドレスとプロトコルポート番号もパケットのペイロードで通信されます。プロトコルのアプリケーション層ゲートウェイ (AGL) は、パケットのペイロードを解析し、プロトコルが DS-Lite 上で動作し続けるように必要な変更を行います。

NetScaler アプライアンスは、DS-Lite 用の次のプロトコルの ALG をサポートしています。

- FTP
- ICMP
- TFTP
- SIP
- RTSP

FTP、ICMP、および TFTP プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

DS-Lite 構成の FTP プロトコルの ALG を有効または無効にするには、構成の LSN グループの FTP ALG オプションを有効または無効にします。

ICMP プロトコルの ALG はデフォルトで有効になっており、無効にする設定はありません。

TFTP プロトコルの ALG はデフォルトで無効になっています。エンドポイントに依存しないマッピング、エンドポイントに依存しないフィルタリング、宛先ポートを 69（TFTP の場合は既知のポート）に設定した UDP LSN アプリケーションプロファイルを LSN グループにバインドすると、DS-Lite 設定で TFTP ALG が自動的に有効になります。

SIP プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

セッション開始プロトコル（SIP）で DS-Lite を使用するのは複雑です。SIP メッセージには SIP ヘッダーと SIP 本文に IP アドレスが含まれるためです。SIP で LSN を使用する場合、SIP ヘッダーには発信者と受信者に関する情報が含まれ、デバイスはこの情報を変換して外部ネットワークから隠します。SIP 本体には、メディア送信用の IP アドレスとポート番号を含むセッション記述プロトコル（SDP）情報が含まれています。DS-Lite 用 SIP ALG は RFC 3261、RFC 3581、RFC 4566、および RFC 4475 に準拠しています。

> 注
>
> SIP ALG は、NetScaler スタンドアロンアロンアプライアンス、NetScaler 高可用性セットアップ、および NetScaler クラスターセットアップでサポートされています。

SIP ALG の制限事項

DS-Lite 用の SIP ALG には次の制限があります。

- SDP ペイロードのみがサポートされます。
- 以下はサポートされていません:
  - マルチキャスト IP アドレス
  - 暗号化された SDP
  - SIP TLS
  - FQDN トランスレーション
  - SIP レイヤー認証
  - 管理パーティション

- マルチパートボディ
- ラインフォールディング

SIP ALG の設定

LSN 設定の一部として SIP ALG を設定する必要があります。LSN の設定手順については、DS-Lite の設定を参照してください。LSN の設定時には、次のことを確認してください。

- LSN アプリケーションプロファイルを追加する際には、次のパラメータを設定します。
  - IP Pooling = PAIRED
  - アドレスとポートのマッピング = エンドポイントに依存しない
  - フィルタリング = エンドポイントに依存しない
- SIP ALG プロファイルを作成し、送信元ポート範囲または宛先ポート範囲のいずれかを定義していることを確認します。SIP ALG プロファイルを LSN グループにバインドします
- LSN グループの SIP ALG を有効にする

CLI を使用して **LSN** 設定の **SIP ALG** を有効にするには

コマンドプロンプトで入力します。

```
add lsn group <groupname> -clientname <string>[-sipalg ( ENABLED |
    DISABLED )]

show lsn group<groupname>
<!--NeedCopy-->
```

CLI を使用して **LSN** 設定の **SIP ALG** を有効にするには

コマンドプロンプトで入力します。

```
add lsn sipalgprofile<sipalgprofilename>[-dataSessionIdleTimeout<
    positive_integer>][-sipSessionTimeout<positive_integer>][-
    registrationTimeout<positive_integer>][-sipsrcportrange<port[-port
    ]>][-sipdstportrange<port[-port]>][-openRegisterPinhole ( ENABLED |
    DISABLED )][-openContactPinhole ( ENABLED | DISABLED )][-
    openViaPinhole ( ENABLED | DISABLED )][-openRecordRoutePinhole (
    ENABLED | DISABLED )]-sipTransportProtocol ( TCP | UDP )[-
    openRoutePinhole ( ENABLED | DISABLED )][-rport ( ENABLED | DISABLED
     )]

show lsn sipalgprofile<sipalgprofilename>
<!--NeedCopy-->
```


構成例

次の DS-Lite 設定の例では、ネットワーク 2001: DB8:: 3:0 /96 の B4 デバイスからの TCP トラフィックに対して SIP ALG が有効になっています。

```
add lsn client LSN-DSLITE-CLIENT-1
Done
bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/96
Done
add lsn pool LSN-DSLITE-POOL-1
Done
bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
   DB8::5:6
Done
add lsn appsprofile LSN-DSLITE-APPS-PROFILE-1 TCP -ippooling PAIRED -
   mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn sipalgprofile SIPALGPROFILE-1  -sipdstportrange 5060 -
   sipTransportProtocol TCP
Done
add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
   portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1 -sipalg ENABLED
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -appsprofilename LSN-DSLITE-APPS-
   PROFILE-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -sipalgprofilename SIPALGPROFILE-1
Done
<!--NeedCopy-->
```

RTSP プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

リアルタイムストリーミングプロトコル (RTSP) は、リアルタイムのメディアデータを転送するためのアプリケーションレベルのプロトコルです。エンドポイント間のメディアセッションの確立と制御に使用される RTSP は、メディアクライアントとメディアサーバー間の制御チャネルプロトコルです。典型的な通信は、クライアントとストリーミングメディアサーバーの間です。

プライベートネットワークからパブリックネットワークにメディアをストリーミングするには、ネットワーク上で IP アドレスとポート番号を変換する必要があります。NetScaler の機能には、RTSP 用のアプリケーション層ゲートウェイ（ALG）が含まれています。これを大規模 NAT（LSN）と組み合わせて使用すると、メディアストリームを解析し、プロトコルがネットワーク上で引き続き機能するように必要な変更を加えることができます。

IP アドレス変換の実行方法は、メッセージのタイプと方向、およびクライアント/サーバ環境でサポートされるメディアの種類によって異なります。メッセージは次のように翻訳されます。

- アウトバウンドリクエスト—LSN IP アドレスと呼ばれる NetScaler 所有のパブリック IP アドレスへのプライベート IP アドレスです。
- インバウンド応答：プライベート IP アドレスへの LSN IP アドレス。
- インバウンドリクエスト-翻訳なし。
- アウトバウンド応答：LSN プール IP アドレスへのプライベート IP アドレス。

> 注
>
> RTSP ALG は、NetScaler スタンドアロンアロンアプライアンス、NetScaler 高可用性セットアップ、および NetScaler クラスターセットアップでサポートされています。

RTSP ALG の制限事項

RTSP ALG は以下をサポートしていません。

- マルチキャスト RTSP セッション
- UDP 経由の RTSP セッション
- 管理パーティション
- RTSP 認証
- HTTP トンネリング

RTSP ALG の設定

RTSP ALG を LSN 設定の一部として設定します。LSN の設定手順については、DS-Lite の設定を参照してください。LSN の設定時には、次のことを確認してください。

- LSN アプリケーションプロファイルを追加する際には、次のパラメータを設定します。
  - IP Pooling = PAIRED
  - アドレスとポートのマッピング = エンドポイントに依存しない
  - フィルタリング = エンドポイントに依存しない
- LSN グループの RTSP ALG を有効にする
- RTSP ALG プロファイルを作成し、その RTSP ALG プロファイルを LSN グループにバインドします

CLI を使用して **LSN** 構成の **RTSP ALG** を有効にするには

コマンドプロンプトで入力します。


```
add lsn group <groupname> -clientname <string> [-rtspalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

CLI を使用して **LSN** 構成の **RTSP ALG** を有効にするには

コマンドプロンプトで入力します。

```
add lsn rtspalgprofile <rtspalgprofilename> [-rtspIdleTimeout <
    positive_integer>] -rtspportrange <port[-port]> [-
    rtspTransportProtocol (TCP|UDP)]

show lsn rtspalgprofile <rtspalgprofilename>
<!--NeedCopy-->
```

RTSP ALG 設定の例

次の DS-Lite 設定の例では、ネットワーク 2001: DB8:: 4:0 /96 の B4 デバイスからの TCP トラフィックに対して RTSP ALG が有効になっています。

RTSP ALG 設定の例**:**

```
add lsn client LSN-DSLITE-CLIENT-5
Done
bind lsn client LSN-DSLITE-CLIENT-5 -network6 2001:DB8::4:0/96
Done
add lsn pool LSN-DSLITE-POOL-5
Done
bind lsn pool LSN-DSLITE-POOL-5 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-5 -type DS-Lite -network6 2001:
    DB8::5:6
Done
add lsn appsprofile LSN-DSLITE-APPS-PROFILE-5 TCP -ippooling PAIRED -
    mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn rtspalgprofile RTSPALGPROFILE-5 -rtspIdleTimeout 1000 -
    rtspportrange 554
Done
add lsn group LSN-DSLITE-GROUP-5 -clientname LSN-DSLITE-CLIENT-5 -
    portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-5 -rtspalg ENABLED
Done
bind lsn group LSN-DSLITE-GROUP-5 -poolname LSN-DSLITE-POOL-5
Done
bind lsn group LSN-DSLITE-GROUP-5 -appsprofilename LSN-DSLITE-APPS-
    PROFILE-5
Done
```



```
bind lsn group LSN-DSLITE-GROUP-5 -rtspalgprofilename RTSPALGPROFILE-5
Done
<!--NeedCopy-->
```

DS-Lite のログ記録と監視

August 15, 2023

DS-Lite の情報を記録して、問題の診断やトラブルシューティング、法的要件を満たすことができます。NetScaler ADC アプライアンスは、DS-Lite 情報をログに記録するための LSN ロギング機能をすべてサポートしています。DS-Lite ロギングを構成するには、「LSN のロギングとモニタリング」で説明した LSN ロギングを設定する手順を使用します。

DS-Lite LSN マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイムスタンプ
- エントリータイプ (マッピング)
- DS-Lite LSN マッピングエントリが作成されたか削除されたか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が表示される場合があります。
  - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートは記録されません。
  - アドレス依存マッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレス-ポート依存マッピング用にログに記録されます。

DS-Lite セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイムスタンプ
- エントリータイプ (セッション)
- DS-Lite セッションが作成されるか削除されるか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名

- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

次の表は、設定されたログサーバに保存されている各タイプのサンプル DS-Lite ログエントリを示しています。これらのログエントリは、NSIP アドレスが 10.102.37.115 の NetScaler アプライアンスによって生成されます。DS-Lite 情報をログに記録して、問題の診断やトラブルシューティング、法的要件を満たすことができます。NetScaler ADC アプライアンスは、DS-Lite 情報をログに記録するための LSN ロギング機能をすべてサポートしています。DS-Lite ロギングを構成するには、「LSN のロギングとモニタリング」で説明した LSN ロギングを設定する手順を使用します。

DS-Lite LSN マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイムスタンプ
- エントリータイプ (マッピング)
- DS-Lite LSN マッピングエントリが作成されたか削除されたか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が表示される場合があります。
  - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートは記録されません。
  - アドレス依存マッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレス-ポート依存マッピング用にログに記録されます。

DS-Lite セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイムスタンプ
- エントリータイプ (セッション)
- DS-Lite セッションが作成されるか削除されるか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

次の表は、設定されたログサーバに保存されている各タイプのサンプル DS-Lite ログエントリを示しています。これらのログエントリは、NSIP アドレスが 10.102.37.115 の NetScaler アプライアンスによって生成されます。

LSN ログエントリタイプ	サンプルログエントリ
DS-Lite セッション作成	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP
DS-Lite セッション削除	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP
DS-Lite LSN マッピングの作成	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP
DS-Lite LSN マッピングの削除	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

現在の **DS-Lite** セッションの表示

現在の DS-Lite セッションを表示して、NetScaler アプライアンス上の不要なセッションや非効率的なセッションを検出できます。選択パラメータに基づいて、すべてまたは一部の DS-Lite セッションを表示できます。

コマンドラインインターフェイスを使用してすべての **DS-Lite** セッションを表示するには

コマンドプロンプトで入力します。


```
show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して選択した **DS-Lite** セッションを表示するには

コマンドプロンプトで入力します。

```
show lsn session  -nattype DS-Lite [-clientname <string>] [-network <
   ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
   ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

次のサンプル出力は、NetScaler アプライアンスに存在するすべての DS-Lite セッションを示しています。

> show lsn session –nattype DS-Lite

```
  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP
     NatPort Proto Dir

1. 2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61
   3002 TCP OUT

2. 2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61
   52862 TCP OUT

3. 2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61
   48116 ICMP OUT

4. 2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69
   48305 TCP OUT
Done
<!--NeedCopy-->
```

設定ユーティリティを使った設定

構成ユーティリティを使用してすべての DS-Lite セッションまたは選択した DS-Lite セッションを表示するには

1. [システム] **> [**** 大規模 **NAT**] > [セッション] に移動し、**[**DS-Lite**] タブをクリックします。
2. 選択パラメータに基づいて **DS-Lite** セッションを表示するには、「検索」をクリックします。

DS-Lite セッションのクリア

不要または非効率的な DS-Lite セッションを NetScaler アプライアンスから削除できます。アプライアンスは、これらのセッションに割り当てられたリソース（NAT IP アドレス、ポート、メモリなど）をすぐに解放し、リソースを新しいセッションで使用できるようにします。アプライアンスは、削除されたセッションに関連する後続のパケット

もすべてドロップします。NetScaler アプライアンスからすべてまたは選択した DS-Lite セッションを削除できます。

コマンドラインインターフェイスを使用してすべての **DS-Lite** セッションをクリアするには

コマンドプロンプトで入力します。

```
flush lsn session  -nattype DS-Lite

show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して選択した **DS-Lite** セッションをクリアするには

コマンドプロンプトで入力します。

```
flush lsn session  -nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]

show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

構成ユーティリティを使用してすべての **DS-Lite** セッションまたは選択した **DS-Lite** セッションをクリアするには

1. [システム] > [大規模 **NAT**] > [セッション] に移動し、[**DS-Lite**] タブをクリックします。
2.「フラッシュセッション」をクリックします。

HTTP ヘッダー情報のロギング

NetScaler アプライアンスは、DS-Lite 機能を使用する HTTP 接続のリクエストヘッダー情報をログに記録できます。HTTP リクエストパケットの次のヘッダー情報を記録できます。

- HTTP リクエストの送信先の URL
- HTTP リクエストで指定された HTTP メソッド
- HTTP リクエストで使用される HTTP バージョン
- HTTP 要求を送信したサブスクライバの IPv4 アドレス

ISP は HTTP ヘッダーログを使用して、複数のサブスクライバーの HTTP プロトコルに関連する傾向を確認できます。たとえば、ISP はこの機能を使用して、複数のサブスクライバーの中で最も人気のある Web サイトを検索できます。

構成の手順

次のタスクを実行して、HTTP ヘッダー情報を記録するように NetScaler アプライアンスを構成します。

- **HTTP** ヘッダーログプロファイルを作成します。HTTP ヘッダーログプロファイルは、ロギングを有効または無効にできる HTTP ヘッダー属性 (URL や HTTP メソッドなど) のコレクションです。
- **HTTP** ヘッダーを **DS-Lite LSN** 構成の **LSN** グループにバインドします。HTTP ヘッダーログプロファイル名パラメーターを作成した HTTP ヘッダーログプロファイルの名前に設定して、HTTP ヘッダーログプロファイルを LSN 構成の LSN グループにバインドします。次に、NetScaler アプライアンスは、LSN グループに関連するすべての HTTP リクエストの HTTP ヘッダー情報を記録します。HTTP ヘッダーログプロファイルは複数の LSN グループにバインドできますが、LSN グループには 1 つの HTTP ヘッダーログプロファイルしか割り当てることができません。

コマンドラインインターフェイスを使用して **HTTP** ヘッダーログプロファイルを作成するには　コマンドプロンプトで入力します。

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **HTTP** ヘッダーログプロファイルを **LSN** グループにバインドするにはコマンドプロンプトで入力します。

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```

構成例

次の DS-Lite LSN 設定では、HTTP ヘッダーログプロファイル HTTP-Header-Log-1 は LSN グループ LSN-DSLITE-GROUP-1 にバインドされています。ログプロファイルには、すべての HTTP 属性 (URL、HTTP メソッド、HTTP バージョン、ホスト IP アドレス) のロギングが有効になっているので、これらの属性はすべて B4 デバイス (ネットワーク 2001: DB 8:5001:: /96 内) からの HTTP リクエストに対してログに記録されます。

設定例**:**

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1

Done

```



```
add lsn client LSN-DSLITE-CLIENT-1

Done

bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100

Done

add lsn pool LSN-DSLITE-POOL-1

Done

bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70

Done

add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
   DB8::5:6

Done

add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
   portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1

Done

bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1

Done

bind lsn group LSN-DSLITE-GROUP-1 -httphdrlogprofilename HTTP-HEADER-
   LOG-1

Done
<!--NeedCopy-->
```

IPFIX ロギング

NetScaler アプライアンスは、LSN イベントに関する情報をインターネットプロトコルフロー情報エクスポート（IPFIX）形式で、構成済みの IPFIX コレクターのセットに送信することをサポートしています。アプライアンスは既存の AppFlow 機能を使用して、LSN イベントを IPFIX 形式で IPFIX コレクターに送信します。

IPFIX ベースのロギングは、次の DS_Lite 関連イベントで使用できます。

- LSN セッションの作成または削除。
- LSN マッピングエントリの作成または削除。
- デターミニスティック NAT におけるポートブロックの割り当てまたは割り当て解除
- ダイナミック NAT におけるポートブロックの割り当てまたは割り当て解除
- サブスクライバセッションクォータを超えると。

IPFIX ロギングを設定する前に考慮すべきポイント

IPSec ALG の設定を開始する前に、次の点を考慮してください。

- NetScaler ADC アプライアンスで AppFlow 機能および IPFIX コレクタを構成する必要があります。手順については、AppFlow 機能の構成を参照してください。

構成の手順

LSN 情報を IPFIX 形式で記録するには、次のタスクを実行します。

- **AppFlow** 構成で **LSN** ロギングを有効にします。AppFlow 構成の一部として LSN ロギングパラメータを有効にします。
- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、IPFIX 形式のログ情報を有効または無効にする IPFIX パラメータが含まれます。
- **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドします。LSN ログプロファイルを 1 つまたは複数の LSN グループにバインドします。バインドされた LSN グループに関連するイベントは IPFIX 形式で記録されます。

CLI を使用して **AppFlow** 構成で **LSN** ロギングを有効にするには　コマンドプロンプトで入力します。

```
set appflow param -lsnLogging (ENABLED |DISABLED )

show appflow param
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを作成するには、コマンドプロンプトで次のように入力します　コマンドプロンプトで入力します。

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには　コマンドプロンプトで入力します。

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```


GUI を使用して **LSN** ログプロファイルを作成するには　[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[ログ] タブをクリックして、ログプロファイルを追加します。

GUI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには

1. [システム] > [大規模 **NAT**] > [**LSN** グループ**]** に移動し、**LSN** グループを開きます。
2. 詳細設定で、**+ Log Profile** をクリックして、作成したログプロファイルを LSN グループにバインドします。

DS-Lite のポート制御プロトコル

August 15, 2023

NetScaler アプライアンスは、大規模 NAT（LSN）用の Port Control Protocol（PCP）をサポートするようになりました。ISP の加入者アプリケーションの多くは、インターネットからアクセスできる必要があります (たとえば、インターネット経由で監視する IP カメラなどの Internet of Things (IOT) デバイス)。この要件を満たす 1 つの方法は、静的な大規模 NAT（LSN）マップを作成することです。しかし、加入者の数が非常に多い場合、スタティック LSN NAT マップを作成することは現実的な解決策ではありません。

Port Control Protocol（PCP）を使用すると、加入者は自分や他のサードパーティデバイス用の特定の LSN NAT マッピングを要求できます。大規模な NAT デバイスは LSN マップを作成し、サブスクライバに送信します。サブスクライバは、サブスクライバに接続できる NAT IP アドレス:NAT ポートをインターネット上のリモートデバイスに送信します。

アプリケーションは通常、LSN マッピングがタイムアウトしないように、大規模な NAT デバイスに頻繁にキープアライブメッセージを送信します。PCP は、アプリケーションが LSN マッピングのタイムアウト設定を学習できるようにすることで、このようなキープアライブメッセージの頻度を減らすのに役立ちます。これにより、ISP のアクセスネットワークの帯域幅消費とモバイルデバイスのバッテリー消費量を削減できます。

PCP はクライアント/サーバーモデルで、UDP トランスポートプロトコルで実行されます。NetScaler アプライアンスは PCP サーバーコンポーネントを実装し、RFC 6887 に準拠しています。

構成の手順

PCP を設定するには、次のタスクを実行します。

- (オプション) PCP プロファイルを作成します。PCP プロファイルには、PCP 関連パラメータ（マッピングやピアの PCP リクエストのリスニングなど）の設定が含まれます。PCP プロファイルは PCP サーバーにバインドできます。PCP サーバーにバインドされた PCP プロファイルは、そのすべての設定を PCP サーバーに適用します。PCP プロファイルは複数の PCP サーバーにバインドできます。デフォルトでは、デフォルトのパラメータ設定を持つ 1 つの PCP プロファイルがすべての PCP サーバーにバインドされます。PCP サーバ

ーにバインドした PCP プロファイルは、そのサーバーのデフォルトの PCP プロファイル設定よりも優先されます。デフォルトの PCP プロファイルには、次のパラメータ設定があります。

  - マッピング: 有効
  - ピア: 有効
  - 最小マップ寿命:120 秒
  - 最大ライフタイム:86400 秒
  - アナウンス数:10
  - サードパーティ: 無効

- PCP サーバーを作成し、それに PCP プロファイルをバインドします。NetScaler アプライアンスに PCP サーバーを作成して、サブスクライバーからの PCP 関連のリクエストとメッセージを受信します。PCP サーバーにアクセスするには、サブネット IP (SNIP) アドレスを割り当てる必要があります。デフォルトでは、PCP サーバーはポート 5351 で受信します。
- PCP サーバーを LSN 構成の LSN グループにバインドします。作成した PCP サーバーを指定する PCP サーバーパラメーターを設定して、作成した PCP サーバーを LSN 構成の LSN グループにバインドします。作成された PCP サーバには、この LSN グループの加入者だけがアクセスできます。
  注: 大規模な NAT 設定の PCP サーバは、ACL ルールで識別されたサブスクライバからの要求には対応しません。

CLI を使用して **PCP** プロファイルを作成するには

コマンドプロンプトで入力します。

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

CLI を使用して **PCP** サーバーを作成するには

コマンドプロンプトで入力します。

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```


DS-LITE のサンプル構成

次の設定例では、PCP-DSLITE-PROFILE-1 の PCP 設定を持つ PCP-SERVER-1 が LSN グループ LSN-DSLITE-GROUP-1 にバインドされています。PCP-SERVER-9 serves PCP requests from IPv4 subscribers behind B4 devices from network 2001:DB8::3:0/100.

設定例:

```
add pcp profile PCP-DSLITE-PROFILE-1 -minMapLife 300
Done
add pcp server PCP-DSLITE-SERVER-1 192.0.3.10  -pcpProfile PCP-DSLITE-PROFILE-1
Done
add lsn client LSN-DSLITE-CLIENT-1
Done
bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
Done
add lsn pool LSN-DSLITE-POOL-1
Done
bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:DB8::5:6
Done
add lsn group LSN-DSLITE-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname  PCP-NAT64-SERVER-1
Done
<!--NeedCopy-->
```

大規模 **NAT64**

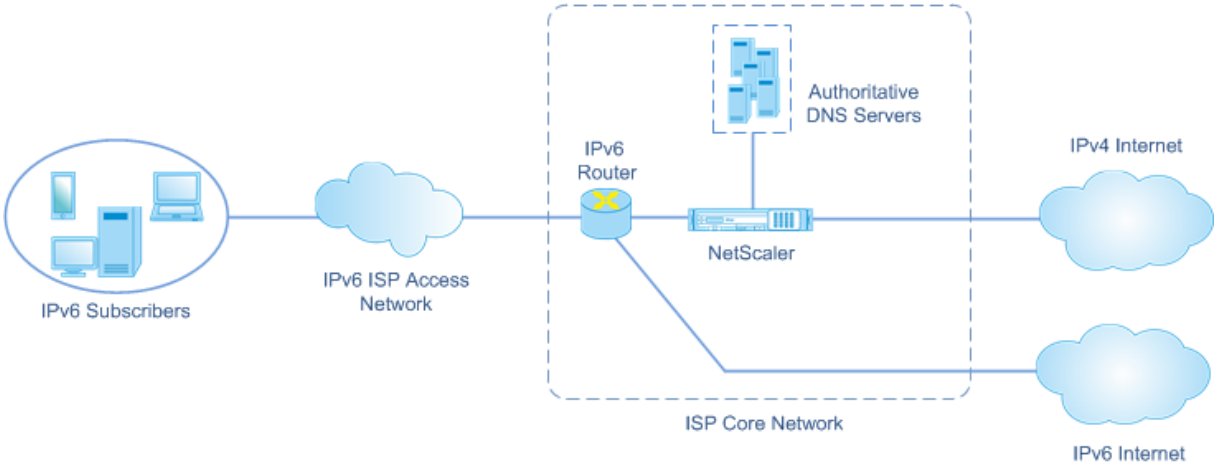
August 15, 2023

IPv4 アドレスの枯渇が差し迫っているため、ISP は IPv6 インフラストラクチャへの移行を開始しました。ただし、公衆インターネットのほとんどが依然として IPv4 を使用しているため、移行中も、ISP は IPv6 とともに IPv4 を引き続きサポートする必要があります。大規模 NAT64 は、IPv6 インフラストラクチャを備えた ISP が IPv6 のみの加入者を IPv4 インターネットに接続するための IPv6 移行ソリューションです。DNS64 は、IPv6 のみのクライアントによる IPv4 専用ドメインの検出を可能にするソリューションです。DNS64 を大規模な NAT64 と組み合わせて使用すると、IPv6 のみのクライアントと IPv4 のみのサーバー間のシームレスな通信が可能になります。

NetScaler アプライアンスは大規模な NAT64 と DNS64 を実装し、RFC 6145、6146、6147、6052、3022、2373、2765、2464 に準拠しています。

アーキテクチャ

NetScaler アプライアンスを使用する ISP の NAT64 アーキテクチャは、ISP のコアネットワークに導入された NetScaler アプライアンスを介して IPv4 インターネットにアクセスする IPv6 サブスクライバーで構成されています。IPv6 加入者は、ISP の IPv6 専用アクセスネットワークを介して ISP コアネットワークに接続されます。



NetScaler アプライアンスの大規模な NAT64 機能により、NetScaler アプライアンス上のセッション情報を維持したまま、IPv6 クライアントと IPv4 サーバー間の通信を IPv6 から IPv4 へのパケット変換によって、またはその逆が可能になります。NetScaler DNS64 機能は、IPv4 専用ドメインの DNS AAAA レコードを合成してサブスクライバーに送信することにより、IPv4 のみのドメインを IPv6 サブスクライバーに提供します。

大規模な NAT64 には、NAT64 プレフィックスと NAT IPv4 プールという 2 つの主要コンポーネントがあります。DNS64 には、NAT64 プレフィックスと同じ値を持つ DNS64 プレフィックスという 1 つの主要コンポーネントがあります。

インターネット上の IPv4 専用 Web サーバーでホストされているドメイン名に対する AAAA リクエストを IPv6 のみのサブスクライバーから受信すると、NetScaler DNS64 機能はドメイン名の AAAA レコードを合成してサブスクライバーに送信します。AAAA レコードは、DNS64 プレフィックス（NAT64 プレフィックスに設定されている）とドメイン名の実際の IPv4 アドレスを連結することによって合成されます。

これで、加入者には、目的のドメイン名に対応する IPv6 宛先アドレスが割り当てられました。サブスクライバーは、合成された IPv6 アドレスに要求を送信します。IPv6 リクエストを受信すると、大規模な NetScaler NAT64 機能が IPv6 リクエストパケットを IPv4 リクエストパケットに変換します。大規模な NAT64 は、IPv4 要求の宛先アドレスを IPv4 アドレスに設定します。IPv4 アドレスは、IPv6 アドレスから NAT64 プレフィックスを取り除くことにより、IPv6 要求の宛先アドレスから抽出されます。宛先ポートは IPv6 リクエストから保持されます。また、大規模 NAT64 は、IPv4 パケットの送信元 IP アドレス: 送信元ポートを、設定された NAT プールから選択された NAT IP アドレス:NAT ポートに設定します。

アプライアンスは、大規模な NAT64 機能を使用するすべてのアクティブセッションの記録を保持します。これらのセッションは大規模な NAT64 セッションと呼ばれます。また、アプライアンスは、大規模な NAT64 セッションごとに、サブスクライバの IPv6 アドレスとポート、NAT IPv4 アドレスとポートの間のマッピングを管理します。これらのマッピングは大規模な NAT64 マッピングと呼ばれます。NetScaler アプライアンスは、大規模な NAT64 セッシ

ョンエントリと大規模な NAT64 マッピングエントリから、（インターネットから受信した）応答パケットを特定の NAT64 セッションに属するものとして認識します。

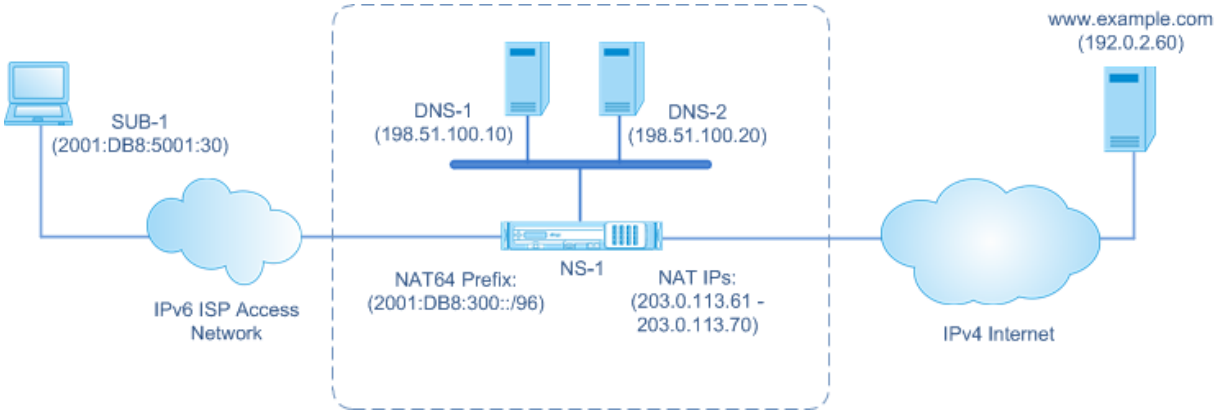
アプライアンスは、特定の NAT64 セッションに属する IPv4 応答パケットを受信すると、NAT64 セッションに保存されている情報を使用して IPv4 パケットを IPv6 パケットに変換し、IPv6 応答パケットを加入者に送信します。

例:**NAT64** および **DNS64** デプロイメントのトラフィックフロー

NetScaler アプライアンス NS-1 と、ISP のコアネットワークにある 2 つのローカル DNS サーバー（DNS-1 と DNS-2）、および IPv6 サブスクライバーの SUB-1 で構成される大規模な NAT64 および DNS64 環境の例を考えてみましょう。SUB-1 は、ISP の IPv6 アクセスネットワークを介して NS-1 に接続されています。NS-1 には、IPv6 サブスクライバ SUB-1 と IPv4 ホスト（内部および外部）間の通信を可能にする大規模な NAT64 および DNS64 構成が含まれています。

大規模な NAT64 構成には、IPv6 リクエストを IPv4 リクエストに変換したり、IPv4 レスポンスを IPv6 レスポンスに変換したりするための NAT64 プレフィックス（2001：DB 8:300:: /96）と NAT IPv4 プールが含まれています。

DNS64 構成には、DNS 負荷分散仮想サーバー LBVS-DNS64-1 (2001: DB 8:9999:: 99) と DNS64 プレフィックス (2001: DB 8:300:: /96) が含まれます。LBVS-DNS64-1 は、ISP のサブスクライバーにとってローカル DNS サーバー DNS-1 および DNS-2 となります。NAT64 プレフィックスと同じ値を持つ DNS64 プレフィックスは、DNS サーバー DNS-1 および DNS-2 から受信した DNS A レコードから DNS AAAA レコードを合成するために使用されます。NS-1 は、IPv4 ホストを解決するための DNS 要求に対して SUB-1 に合成された AAAA レコードで応答します。



DNS64 トラフィックフロー

IPv6 サブスクライバ SUB-1 と `www.example.com`、インターネット上の IPv4 専用の Web サーバにあるサイトとの間では、次のようにトラフィックが流れます。

1. IPv6 サブスクライバ SUB-1 は、指定された DNS サーバ (2001: DB 8:9999:: 99) に`www.example.com`の DNS AAAA リクエストを送信します。

2. NetScaler アプライアンス NS1 の DNS 負荷分散仮想サーバー LBVS-DNS64-1（2001: DB 8:9999:: 99）が AAAA リクエストを受信します。LBVS-DNS64-1 のロードバランシングアルゴリズムは、DNS サーバー DNS-1 を選択し、AAAA リクエストをそのサーバーに転送します。
3. DNS-1 は、www.example.comで使用できる AAAA レコードがないため、空のレコードまたはエラーメッセージを返します。
4. LBVS-DNS64-1 では DNS64 オプションが有効になっていて、CL1 からの AAAA リクエストが DNS64-Policy-1 で指定された条件と一致するため、NS1 はwww.example.comの IPv4 アドレスの DNS A リクエストを DNS-1 に送信します。
5. DNS-1 の応答には、192.0.2.60 というレコードが A レコードとして返されます。www.example.com
6. NS1 の DNS64 モジュールは、LBVS-DNS64-1 に関連する DNS64 プレフィックス (2001: DB 8:300:: /96) と www.example.com= 2001: DB 8:300:: 192.0.2.60 の IPv4 アドレス (192.0.2.60) を連結してwww.example.comの AAAA レコードを合成します
7. NS1 は、合成された AAAA レコードを IPv6 クライアント CL1 に送信します。NS1 は A レコードもメモリにキャッシュします。NS1 はキャッシュされた A レコードを使用して、後続の AAAA リクエスト用に AAAA レコードを合成します。

NAT64 トラフィックフロー

1. IPv6 サブスクライバー SUB-1 は 2001: DB 8:5001:30 www.example.comにリクエストを送信します。IPv6 パケットには次のものが含まれます。

   - ソース IP アドレス = 2001: DB 8:5001:30
   - 送信元ポート = 2552
   - 宛先 IP アドレス = 2001: DB 8:300:: 192.0.2.60
   - 宛先ポート = 80

2. IPv6 サブスクライバー SUB-1 は 2001: DB 8:5001:30 www.example.comにリクエストを送信します。IPv6 パケットには次のものが含まれます。

   - ソース IP アドレス = 2001: DB 8:5001:30
   - 送信元ポート = 2552
   - 宛先 IP アドレス = 2001: DB 8:300:: 192.0.2.60
   - 宛先ポート = 80

3. NS-1 が IPv6 パケットを受信すると、大規模な NAT64 モジュールは、次のように変換された IPv4 要求パケットを作成します。

   - 送信元 IP アドレス = 設定されている NAT プール (203.0.113.61) で使用可能な IPv4 アドレスのいずれか
   - 送信元ポート = NAT IPv4 アドレス (3002) が割り当てられているポートのいずれか
   - 宛先 IP アドレス = IPv6 アドレス (192.0.2.60) から NAT64 プレフィックス (2001: DB 8:300:: /96) を削除して IPv6 リクエストの宛先アドレスから抽出された IPv4 アドレス

- 送信先ポート = IPv6 リクエストの送信先ポート (80)

4. 大規模な NAT64 モジュールは、この大規模な NAT64 フローのマッピングとセッションエントリも作成します。セッションとマッピングのエントリには、次の情報が含まれます。

   - IPv6 パケットの送信元 IP アドレス = 2001: DB 8:5001:30
   - IPv6 パケットの送信元ポート = 2552
   - NAT IP アドレス = 203.0.113.61
   - NAT ポート = 3002
   - NS-1 は、生成された IPv4 パケットをインターネット上の宛先に送信します。

5. 要求パケットを受信すると、`www.example.com` サーバはパケットを処理し、NS-1 に応答パケットを送信する。IPv4 応答パケットには次の内容が含まれます。

   - 送信元 IP アドレス = 192.0.2.60
   - 送信元ポート = 80
   - 宛先 IP アドレス = 203.0.113.61
   - デスティネーションポート = 3002

6. IPv4 応答パケットを受信すると、NS-1 は大規模 NAT64 マッピングとセッションエントリを調べ、IPv4 応答パケットが大規模な NAT64 セッションに属していることを検出します。大規模な NAT64 モジュールは、変換された IPv6 応答パケットを作成します。

   - ソース IP アドレス = 2001: DB 8:300:: 192.0.2.60
   - 送信元ポート = 80
   - ターゲット IP アドレス = 2001: DB 8:5001:30
   - デスティネーションポート = 2552

7. NS-1 は、変換された IPv6 応答をクライアント SUB-1 に送信します。

NetScaler アプライアンスでサポートされる大規模な **NAT64** 機能

NetScaler ADC アプライアンスの大規模 NAT64 は、標準の LSN 機能セットをサポートしています。これらの LSN 機能の詳細については、「LSN アーキテクチャ」を参照してください。

NetScaler アプライアンスでサポートされる大規模な NAT64 機能の一部を次に示します。

- ALG。SIP、RTSP、FTP、ICMP、TFTP プロトコル用のアプリケーション層ゲートウェイ（ALG）のサポート。
- 確定型/固定型 NAT ロギングを最小限に抑えるため、サブスクライバへのポートブロックの事前割り当てがサポートされます。
- マッピング。エンドポイント独立マッピング (EIM)、アドレス依存マッピング (ADM)、およびアドレスポート依存マッピング (APDM) をサポートします。
- フィルタリング。エンドポイント独立フィルタリング (EIF)、アドレス依存フィルタリング (ADF)、およびアドレスポート依存フィルタリング (APDF) をサポートします。

- クォータ。ポート数、サブスクライバあたりのセッション、および LSN グループあたりのセッション数の制限を設定できます。
- スタティックマッピング。大規模な NAT64 マッピングの手動定義をサポートします。
- ヘアピンフロー。NAT IP アドレスを使用するサブスクライバまたは内部ホスト間の通信をサポートします。
- 464XLAT 接続。IPv6 サブスクライバホスト上の IPv4 専用アプリケーションと IPv6 ネットワーク経由のインターネット上の IPv4 ホスト間の通信をサポートします。
- 可変長の NAT64 プレフィックスと DNS64 プレフィックス。NetScaler アプライアンスは、32、40、48、56、64、96 の長さの NAT64 および DNS64 プレフィックスの定義をサポートしています。
- 複数の NAT64 および DNS64 プレフィックス。NetScaler アプライアンスは、複数の NAT64 および DNS64 プレフィックスをサポートします。
- LSN クライアント。IPv6 プレフィックスと拡張 ACL6 ルールを使用して、大規模な NAT64 のサブスクライバーを指定または識別できます。
- ロギング。法執行機関向け NAT64 セッションのロギングをサポートします。また、次のロギングもサポートされています。
  - 信頼性の高いシスログ。TCP 経由で外部ログサーバーへの SYSLOG メッセージの送信をサポートし、より信頼性の高い転送メカニズムを実現します。
  - ログサーバーの負荷分散。冗長なログメッセージの保存を防ぐための外部ログサーバーの負荷分散をサポートします。
  - 最小限のロギング。確定的な LSN 構成またはポートブロック付きの動的 LSN 構成により、大規模な NAT64 ログ量が大幅に減少します。
  - **MSISDN** 情報をログに記録します。加入者の MSISDN 情報を大規模な NAT64 ログに含めることで、インターネット上の加入者のアクティビティを識別および追跡できます。

大規模 **NAT64** の構成に関する考慮事項

August 15, 2023

大規模な NAT64 と DNS64 の設定を始める前に、次の点を考慮してください。

1. RFC で説明されている大規模な NAT64 のさまざまなコンポーネントを必ず理解してください。
2. NetScaler アプライアンスは、大規模 NAT64 用の以下の ALG のみをサポートします。
   - FTP
   - TFTP
   - ICMP
   - SIP
   - RTSP
3. 2 つの NetScaler アプライアンスの高可用性セットアップでは、大規模な NAT64 セッション同期（接続ミラーリング）はサポートされていません。

DNS64 の構成

August 15, 2023

NetScaler アプライアンスでステートフル NAT64 構成に必要なエンティティを作成するには、次の手順が必要です。

- DNS サービスを追加します。DNS サービスは、NetScaler ADC アプライアンスが DNS プロキシサーバーとして機能する DNS サーバーを論理的に表現したものです。サービスのオプションパラメータの設定の詳細については、「負荷分散」を参照してください。
- DNS64 アクションと DNS64 ポリシーを追加し、DNS64 アクションを DNS64 ポリシーにバインドします。DNS64 ポリシーは、関連する DNS64 アクションの設定に従って、DNS64 処理のトラフィックと照合する条件を指定します。DNS64 アクションは、必須の DNS64 プレフィックスと、オプションの除外ルールとマップルール設定を指定します。
- DNS 負荷分散仮想サーバーを作成し、DNS サービスと DNS64 ポリシーをそれにバインドします。DNS 負荷分散仮想サーバーは、バインドされた DNS サービスに代表される DNS サーバーの DNS プロキシサーバーとして機能します。仮想サーバに着信するトラフィックは、DNS64 処理用のバインドされた DNS64 ポリシーと照合されます。負荷分散仮想サーバーのオプションパラメータの設定の詳細については、「負荷分散」を参照してください。

> 注
>
> コマンドラインインターフェイスにはこれら 2 つのタスク用の個別のコマンドがありますが、GUI ではこれらを 1 つのダイアログボックスにまとめています。

- DNS レコードのキャッシュを有効にします。NetScaler ADC アプライアンスのグローバルパラメータを有効にして、DNS プロキシ操作によって取得される DNS レコードをキャッシュします。DNS レコードのキャッシュの有効化の詳細については、「DNS レコードのキャッシュの有効化」を参照してください。

コマンドラインインターフェイスを使用して DNS タイプのサービスを作成するには

コマンドプロンプトで入力します。

```
add service <name> <IP> <serviceType> <port> …
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して DNS64 アクションを作成するには

コマンドプロンプトで入力します。

```
add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule <
    expression>] [-excludeRule <expression>]
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **DNS64** ポリシーを作成するには

コマンドプロンプトで入力します。

```
add dns policy64 <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **DNS** 負荷分散仮想サーバーを作成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED
    ) [-bypassAAAA ( YES | NO)] ...
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **DNS** サービスと **DNS64** ポリシーを **DNS** 負荷分散仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName> ...

bind lb vserver <name> -policyName <string> -priority <positive_integer
    > ...
<!--NeedCopy-->
```

設定例:

```
add service SVC-DNS-1 203.0.113.50 DNS 53
Done
add service SVC-DNS-2 203.0.113.60 DNS 53
Done
add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
Done
add dns Policy64 DNS64-Policy-1 -rule "CLIENT.IPv6.SRC.IN_SUBNET(2001:
    DB8:5001::/64)" -action DNS64-Action-1
Done
add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
Done
bind lb vserver LBVS-DNS64-1 SVC-DNS-1
Done
bind lb vserver LBVS-DNS64-1 SVC-DNS-2
Done
bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
Done
<!--NeedCopy-->
```


大規模 **NAT64** の構成

August 15, 2023

NetScaler アプライアンスの大規模な NAT64 構成では、LSN コマンドセットを使用します。大規模な NAT64 構成では、LSN クライアントエンティティは IPv6 サブスクライバーを識別するための IPv6 アドレスまたは IPv6 ネットワークアドレス、または ACL6 ルールを指定します。NAT64 構成には、NAT64 プレフィックスを指定する IPv6 プロファイルも含まれています。

NetScaler アプライアンスでの NAT64 の構成は、次のタスクで構成されます。

- グローバル LSN パラメータを設定します。グローバルパラメータには、LSN 機能用に確保されている NetScaler メモリの量と、高可用性セットアップでの LSN セッションの同期が含まれます。
- IPv6 加入者からのトラフィックを識別するための LSN クライアントエンティティを作成します。LSN クライアントエンティティは IPv6 サブスクライバーのセットを指します。クライアントエンティティには、これらのサブスクライバーからのトラフィックを識別するための IPv6 アドレスまたは IPv6 ネットワークプレフィックス、または ACL6 ルールが含まれます。LSN クライアントは 1 つの LSN グループにしかバインドできません。コマンドラインインターフェイスには、LSN クライアントエンティティを作成し、サブスクライバーを LSN クライアントエンティティにバインドする 2 つのコマンドがあります。GUI は、これら 2 つの操作を 1 つの画面にまとめます。
- LSN プールを作成し、NAT IP アドレスをそのプールにバインドします。LSN プールは、NetScaler アプライアンスが大規模な NAT64 を実行するために使用する NAT IP アドレスのプールを定義します。コマンドラインインターフェイスには、LSN プールを作成し、NAT IP アドレスを LSN プールにバインドする 2 つのコマンドがあります。GUI は、これら 2 つの操作を 1 つの画面にまとめます。
- LSN IP6 プロファイルを作成します。LSN IP6 プロファイルは、大規模な NAT64 構成の NAT64 プレフィックスを定義します。
- (オプション) 指定したプロトコルの LSN トランスポートプロファイルを作成します。LSN トランスポートプロファイルでは、大規模な NAT64 セッションの最大数や、加入者が特定のプロトコルで使用できる最大ポート使用量など、さまざまなタイムアウトと制限を定義します。各プロトコル (TCP、UDP、ICMP) の LSN トランスポートプロファイルを LSN グループにバインドします。プロファイルは複数の LSN グループにバインドできます。LSN グループにバインドされたプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、TCP、UDP、および ICMP プロトコルのデフォルト設定の 1 つの LSN トランスポートプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトトランスポートプロファイルと呼ばれます。LSN グループにバインドした LSN トランスポートプロファイルは、そのプロトコルのデフォルトの LSN トランスポートプロファイルよりも優先されます。
- （オプション）指定したプロトコルの LSN アプリケーションプロファイルを作成し、宛先ポートのセットをそれにバインドします。LSN アプリケーションプロファイルは、特定のプロトコルと宛先ポートセットに対するグループの LSN マッピングと LSN フィルタリング制御を定義します。宛先ポートセットの場合、各プロトコル（TCP、UDP、ICMP）の LSN プロファイルを LSN グループにバインドします。プロファイルは複数の

LSN グループにバインドできます。LSN グループにバインドされた LSN アプリケーションプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、すべての宛先ポートの TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN アプリケーションプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトアプリケーションプロファイルと呼ばれます。指定された宛先ポートセットの LSN アプリケーションプロファイルを LSN グループにバインドすると、バインドされたプロファイルは、その宛先ポートセットでそのプロトコルのデフォルトの LSN アプリケーションプロファイルをオーバーライドします。コマンドラインインターフェイスには、LSN アプリケーションプロファイルを作成し、宛先ポートのセットを LSN アプリケーションプロファイルにバインドするための 2 つのコマンドがあります。GUI は、これら 2 つの操作を 1 つの画面にまとめます。

- LSN グループを作成し、LSN プール、LSN IPv6 プロファイル、(オプション) LSN トランスポートプロファイル、(オプション) LSN アプリケーションプロファイルを LSN グループにバインドします。LSN グループは、LSN クライアント、LSN IPv6 プロファイル、LSN プール、LSN トランスポートプロファイル、および LSN アプリケーションプロファイルで構成されるエンティティです。グループには、ポートブロックサイズや LSN セッションのロギングなどのパラメータが割り当てられます。パラメータ設定は、LSN グループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。LSN グループにバインドできる LSN IPv6 プロファイルは 1 つだけで、LSN グループにバインドされた LSN IPv6 プロファイルは他の LSN グループにバインドできません。バインドできるのは、同じ NAT タイプ設定の LSN プールと LSN グループだけです。複数の LSN プールを 1 つの LSN グループにバインドできます。LSN グループにバインドできる LSN クライアントエンティティは 1 つだけで、LSN グループにバインドされた LSN クライアントエンティティは他の LSN グループにバインドできません。コマンドラインインターフェイスには、LSN グループを作成し、LSN プール、LSN トランスポートプロファイル、および LSN アプリケーションプロファイルを LSN グループにバインドするための 2 つのコマンドがあります。GUI では、これら 2 つの操作が 1 つの画面にまとめられています。

コマンドラインを使った設定

コマンドラインインターフェイスを使用して、さまざまな構成を作成できます。以下の手順に従ってください。

コマンドラインインターフェイスを使用して **LSN** クライアントを作成するには

コマンドプロンプトで入力します。

```
add lsn client <clientname

show lsn client
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **IPv6** ネットワークまたは **ACL6** ルールを **LSN** クライアントにバインドするには

コマンドプロンプトで入力します。

```
bind lsn client <clientname> (-network6 <ipv6_addr|*>| -acl6name <
    string>)

show lsn client
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LAN** プールを作成するには

コマンドプロンプトで入力します。

```
add lsn pool <poolname>

show lsn pool <poolname>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **NAT IP** アドレスを **LSN** プールにバインドするには

コマンドプロンプトで入力します。

```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```

> 注
>
> LSN プールから NAT IP（LSN IP アドレス）アドレスを削除するには、unbind lsn pool コマンドを使用します。

コマンドラインインターフェイスを使用して **LSN IPv6** プロファイルを構成するには

コマンドプロンプトで入力します。

```
add lsn ip6profile <name>  -type NAT64 -natprefix <ipv6_addr|*>

show lsn ip6profile
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **LSN** トランスポートプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn transportprofile <transportprofilename> <transportprotocol> [-
    sessiontimeout <secs>] [-finrsttimeout <secs>] [-portquota <
    positive_integer>] [-sessionquota <positive_integer>] [-
    portpreserveparity ( ENABLED | DISABLED )] [-portpreserverange (
    ENABLED | DISABLED )] [-syncheck ( ENABLED | DISABLED )]

show lsn transportprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** アプリケーションプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn appsprofile <appsprofilename> <transportprotocol> [-ippooling (
    PAIRED | RANDOM )] [-mapping <mapping>] [-filtering <filtering>][-
    tcpproxy ( ENABLED | DISABLED )]

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してアプリケーションプロトコルのポート範囲を **LSN** アプリケーションプロファイルにバインドするには

コマンドプロンプトで入力します。

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** グループを作成するには

コマンドプロンプトで入力します。

```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC |
    DETERMINISTIC )] [-portblocksize <positive_integer>] [-logging(
    ENABLED | DISABLED )] [-sessionLogging ( ENABLED | DISABLED )][-
    sessionSync ( ENABLED | DISABLED )] [-snmptraplimit<positive_integer
    >] [-ftp ( ENABLED | DISABLED )] [-sipalg ( ENABLED | DISABLED )] [-
    rtspalg ( ENABLED |DISABLED )] [-ip6profile <string>]

show lsn group
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **LSN** プロトコルプロファイルと **LSN** プールを **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -httphdrlogprofilename <string> | -appsprofilename <
    string> | -sipalgprofilename <string> | rtspalgprofilename <string>)

show lsn group
<!--NeedCopy-->
```

大規模な **NAT64** 構成の例

大規模な NAT64 の構成例は次のとおりです。

デフォルト設定での単純で大規模な **NAT64** 構成**:**

```
add lsn client LSN-NAT64-CLIENT-1

bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96

add lsn pool LSN-NAT64-POOL-1

bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70

add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96

add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1

bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1

<!--NeedCopy-->
```

加入者を識別するための拡張 **ACL6** ルールを備えたシンプルで大規模な **NAT64** 構成:

```
add ns acl6 LSN-NAT64-ACL-2 ALLOW  -srcIPv6 = 2001:DB8:5002::20 - 2001:
    DB8:5002::200

apply acl6s

add lsn client LSN-NAT64-CLIENT-2

bind lsn client LSN-NAT64-CLIENT-2  -acl6name LSN-NAT64-ACL-2

add lsn pool LSN-NAT64-POOL-2

bind lsn pool LSN-NAT64-POOL-2 203.0.113.5-203.0.113.10

```



```
add lsn ip6profile LSN-NAT64-PROFILE-2 -type NAT64 -natprefix 2001:DB8
    :302::/96

add lsn group LSN-NAT64-GROUP-2 -clientname LSN-NAT64-CLIENT-2 -
    ip6profile LSN-NAT64-PROFILE-2

bind lsn group LSN-NAT64-GROUP-2 -poolname LSN-NAT64-POOL-2

<!--NeedCopy-->
```

確定的な **NAT** リソース割り当てによる大規模な **NAT64** 構成**:**

```
add lsn client LSN-NAT64-CLIENT-7

bind lsn client LSN-NAT64-CLIENT-7 -network6 2001:DB8:1002::7/128

add lsn pool LSN-NAT64-POOL-7 -nattype DETERMINISTIC

bind lsn pool LSN-NAT64-POOL-7 203.0.113.24-203.0.113.27

add lsn ip6profile LSN-NAT64-PROFILE-7 -type NAT64 -natprefix 2001:DB8
    :307::/96

add lsn group LSN-NAT64-GROUP-7 -clientname LSN-NAT64-CLIENT-7 -
    ip6profile LSN-NAT64-PROFILE-7 -nattype DETERMINISTIC -portblocksize
     256

bind lsn group LSN-NAT64-GROUP-7 -poolname LSN-POOL-7

<!--NeedCopy-->
```

大規模 **NAT64** 用のアプリケーションレイヤーゲートウェイの構成

August 15, 2023

一部のアプリケーション層プロトコルでは、IP アドレスとプロトコルポート番号もパケットペイロードで通信されます。プロトコルのアプリケーション層ゲートウェイは、パケットのペイロードを解析し、プロトコルが大規模な NAT64 上で動作し続けるように必要な変更を行います。

NetScaler アプライアンスは、大規模 NAT64 用の次のプロトコルの ALG をサポートしています。

- FTP
- ICMP
- TFTP
- SIP
- RTSP

FTP、ICMP、および TFTP プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

大規模な NAT64 構成の FTP プロトコルの ALG を有効または無効にするには、構成の LSN グループの FTP ALG オプションを有効または無効にします。

ICMP プロトコルの ALG はデフォルトで有効になっており、無効にする設定はありません。

TFTP プロトコルの ALG はデフォルトで無効になっています。エンドポイントに依存しないマッピング、エンドポイントに依存しないフィルタリング、宛先ポートを 69（TFTP の既知のポート）に設定した UDP LSN アプリケーションプロファイルを LSN グループにバインドすると、大規模な NAT64 構成で TFTP ALG が自動的に有効になります。

SIP プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

セッション開始プロトコル (SIP) で大規模な NAT64 を使用するのは複雑です。SIP メッセージには SIP ヘッダーと SIP 本文に IP アドレスが含まれるためです。SIP で LSN を使用する場合、SIP ヘッダーには発信者と受信者に関する情報が含まれ、デバイスはこの情報を変換して外部ネットワークから隠します。SIP 本体には、メディア送信用の IP アドレスとポート番号を含むセッション記述プロトコル（SDP）情報が含まれています。大規模な NAT64 用の SIP ALG は RFC 3261、RFC 3581、RFC 4566、および RFC 4475 に準拠しています。

> 注
>
> SIP ALG は、NetScaler スタンドアロンアロンアプライアンス、NetScaler 高可用性セットアップ、および NetScaler クラスターセットアップでサポートされています。

SIP ALG の制限事項

大規模な NAT64 用の SIP ALG には次の制限があります。

- SDP ペイロードのみがサポートされます。
- 以下はサポートされていません:
  - マルチキャスト IP アドレス
  - 暗号化された SDP
  - SIP TLS
  - FQDN トランスレーション
  - SIP レイヤー認証

- トラフィックドメイン
- 管理パーティション
- マルチパートボディ
- ラインフォールディング

SIP ALG の設定

LSN 設定の一部として SIP ALG を設定する必要があります。LSN の設定手順については、「大規模な NAT64 の設定」を参照してください。LSN の設定時には、次のことを確認してください。

- LSN アプリケーションプロファイルを追加する際には、次のパラメータを設定します。
  - IP Pooling = PAIRED
  - アドレスとポートのマッピング = エンドポイントに依存しない
  - フィルタリング = エンドポイントに依存しない
- SIP ALG プロファイルを作成し、送信元ポート範囲または宛先ポート範囲のいずれかを定義していることを確認します。SIP ALG プロファイルを LSN グループにバインドします。
- LSN グループの SIP ALG を有効にします。

CLI を使用して LSN 設定の SIP ALG を有効にするには

コマンドプロンプトで入力します。

```
add lsn group <groupname> -clientname <string> [-sipalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

CLI を使用して LSN 設定の SIP ALG を有効にするには

コマンドプロンプトで入力します。

```
add lsn sipalgprofile <sipalgprofilename>[-dataSessionIdleTimeout <
    positive_integer>][-sipSessionTimeout <positive_integer>] [-
    registrationTimeout <positive_integer>] [-sipsrcportrange <port[-
    port]>] [-sipdstportrange <port[-port]>] [-openRegisterPinhole (
    ENABLED | DISABLED )] [-openContactPinhole ( ENABLED | DISABLED )]
    [-openViaPinhole ( ENABLED | DISABLED )] [-openRecordRoutePinhole (
    ENABLED | DISABLED )]-sipTransportProtocol ( TCP | UDP ) [-
    openRoutePinhole ( ENABLED | DISABLED )] [-rport ( ENABLED |
    DISABLED )]

show lsn sipalgprofile <sipalgprofilename
<!--NeedCopy-->
```


構成例

次の大規模な NAT64 設定の例では、ネットワーク 2001: DB 8:1003:: /96 内のサブスクライバデバイスからの TCP トラフィックに対して SIP ALG が有効になっています。

```
add lsn client LSN-NAT64-CLIENT-9

Done
bind lsn client LSN-NAT64-CLIENT-9 -network6 2001:DB8:1002::/96

Done
add lsn pool LSN-NAT64-POOL-9

Done
bind lsn pool LSN-NAT64-POOL-9 203.0.113.90

Done
add lsn ip6profile LSN-NAT64-PROFILE-9 -type NAT64 -natprefix 2001:DB8
   :309::/96

Done
add lsn appsprofile LSN-NAT64-APPS-PROFILE-9 TCP -ippooling PAIRED -
   mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done
add lsn sipalgprofile SIPALGPROFILE-9  -sipdstportrange 5060  -
   sipTransportProtocol TCP

Done
add lsn group LSN-NAT64-GROUP-9 -clientnameLSN-NAT64-CLIENT-9 -
   ip6profile LSN-NAT64-PROFILE-7 -sipalg ENABLED

Done
bind lsn group LSN-NAT64-GROUP-9 -poolnameLSN-NAT64-POOL-9
Done
bind lsn group LSN-NAT64-GROUP-9 -appsprofilename LSN-NAT64-APPS-
   PROFILE-9
Done
bind lsn group LSN-NAT64-GROUP-9 -sipalgprofilename SIPALGPROFILE-9
Done
<!--NeedCopy-->
```

RTSP プロトコル用のアプリケーションレイヤーゲートウェイ

August 15, 2023

リアルタイムストリーミングプロトコル (RTSP) は、リアルタイムのメディアデータを転送するためのアプリケーションレベルのプロトコルです。エンドポイント間のメディアセッションの確立と制御に使用される RTSP は、メディ

アクライアントとメディアサーバー間の制御チャネルプロトコルです。典型的な通信は、クライアントとストリーミングメディアサーバーの間です。

プライベートネットワークからパブリックネットワークにメディアをストリーミングするには、ネットワーク上で IP アドレスとポート番号を変換する必要があります。NetScaler の機能には、RTSP 用のアプリケーション層ゲートウェイ（ALG）が含まれています。これを大規模 NAT（LSN）と組み合わせて使用すると、メディアストリームを解析し、プロトコルがネットワーク上で引き続き機能するように必要な変更を加えることができます。

IP アドレス変換の実行方法は、メッセージのタイプと方向、およびクライアント/サーバ環境でサポートされるメディアの種類によって異なります。メッセージは次のように翻訳されます。

- アウトバウンドリクエスト—LSN IP アドレスと呼ばれる NetScaler 所有のパブリック IP アドレスへのプライベート IP アドレスです。
- インバウンド応答：プライベート IP アドレスへの LSN IP アドレス。
- インバウンドリクエスト-翻訳なし。
- アウトバウンド応答：LSN プール IP アドレスへのプライベート IP アドレス。

> 注
>
> RTSP ALG は、NetScaler スタンドアロンアロンアプライアンス、NetScaler 高可用性セットアップ、および NetScaler クラスターセットアップでサポートされています。

RTSP ALG の制限事項

RTSP ALG は以下をサポートしていません。

- マルチキャスト RTSP セッション
- UDP 経由の RTSP セッション
- 管理パーティション
- RTSP 認証
- HTTP トンネリング

RTSP ALG の設定

RTSP ALG を LSN 設定の一部として設定します。LSN の設定手順については、「大規模な NAT64 の設定」を参照してください。設定時には、次のことを確認してください。

- LSN アプリケーションプロファイルを追加する際には、次のパラメータを設定します。
  - IP Pooling = PAIRED
  - アドレスとポートのマッピング = エンドポイントに依存しない
  - フィルタリング = エンドポイントに依存しない
- LSN グループの RTSP ALG を有効にする
- RTSP ALG プロファイルを作成し、その RTSP ALG プロファイルを LSN グループにバインドします

CLI を使用して **LSN** 構成の **RTSP ALG** を有効にするには

コマンドプロンプトで入力します。

```
add lsn group <groupname> -clientname <string> [-rtspalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

CLI を使用して **LSN** 構成の **RTSP ALG** を有効にするには

コマンドプロンプトで入力します。

```
add lsn rtspalgprofile <rtspalgprofilename> [-rtspIdleTimeout <
    positive_integer>] -rtspportrange <port[-port]> [-
    rtspTransportProtocol (TCP|UDP)]

show lsn rtspalgprofile <rtspalgprofilename>
<!--NeedCopy-->
```

RTSP ALG 設定の例

次の大規模な NAT64 設定の例では、ネットワーク 2001: DB 8:1002:: /96 内のサブスクライバデバイスからの TCP トラフィックに対して RTSP ALG が有効になっています。

```
add lsn client LSN-NAT64-CLIENT-9
Done
bind lsn client LSN-NAT64-CLIENT-9 -network6 2001:DB8:1002::/96
Done
add lsn pool LSN-NAT64-POOL-9
Done
bind lsn pool LSN-NAT64-POOL-9 203.0.113.90
Done
add lsn ip6profile LSN-NAT64-PROFILE-9 -type NAT64 -natprefix 2001:DB8
    :309::/96
Done
add lsn appsprofile LSN-NAT64-APPS-PROFILE-9 TCP -ippooling PAIRED -
    mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn rtspalgprofile RTSPALGPROFILE-9 -rtspIdleTimeout 1000 -
    rtspportrange 554
Done
add lsn group LSN-NAT64-GROUP-9 -clientname LSN-NAT64-CLIENT-9 -
    ip6profile LSN-NAT64-PROFILE-7 -rtspalg ENABLED
Done
bind lsn group LSN-NAT64-GROUP-9 -poolname LSN-NAT64-POOL-9
Done
```



```
bind lsn group LSN-NAT64-GROUP-9 -appsprofilename LSN-NAT64-APPS-
    PROFILE-9
Done
bind lsn group LSN-NAT64-GROUP-9  -rtspalgprofilename RTSPALGPROFILE-9
Done
<!--NeedCopy-->
```

静的大規模 **NAT64** マップの構成

August 15, 2023

NetScaler アプライアンスは、次の情報間のマッピングを含む NAT64 マッピングの手動作成をサポートしています。

- 加入者の IP アドレスとポート
- NAT IP アドレスとポート

静的大規模な NAT64 マッピングは、NAT IP アドレス: ポートに対して開始された IPv4 接続を IPv6 に変換し、サブスクライバの IP アドレス: ポート（内部ネットワークにある Web サーバなど）にマッピングする場合に便利です。

コマンドラインを使用して大規模な **NAT64** マッピングを作成するには

コマンドプロンプトで入力します。

```
add lsn static <name> <transportprotocol> <subscrIP> <subscrPort>  [<
    natIP> [<natPort>]] [-destIP <ip_addr> [-dsttd <positive_integer>]]

show lsn static
<!--NeedCopy-->
```

ワイルドカードポートスタティックラージスケール **NAT64** マップ

静的で大規模な NAT64 マッピングエントリは、通常、サブスクライバの IPv6 アドレス: ポートと NAT IPv4 アドレス: ポート間の 1 対 1 のマッピングです。1 対 1 の静的で大規模な NAT64 マッピングエントリでは、加入者 IP アドレスの 1 つのポートだけがインターネットに公開されます。

状況によっては、加入者 IP アドレスのすべてのポート（64K-NAT IPv4 アドレスの最大ポート数に制限される）をインターネットに公開しなければならない場合があります（たとえば、内部ネットワークでホストされ、各ポートで異なるサービスを実行しているサーバなど）。これらの内部サービスにインターネット経由でアクセスできるようにするには、サーバーのすべてのポートをインターネットに公開する必要があります。

この要件を満たす方法の 1 つは、ポートごとに 1 つずつ、つまり 1 対 1 のスタティックマッピングエントリを 64,000 個追加することです。これらのエントリを作成するのは非常に面倒で大変な作業です。また、このような多数の構成エントリは、NetScaler アプライアンスのパフォーマンスの問題につながる可能性があります。

より簡単な方法は、スタティックマッピングエントリでワイルドカードポートを使用することです。NAT ポートとサブスクライバポートのパラメータをワイルドカード文字（*）に設定し、プロトコルパラメータを ALL に設定したスタティックマッピングエントリを 1 つ作成するだけで、すべてのプロトコルのサブスクライバ IP アドレスのすべてのポートがインターネットに公開されます。

ワイルドカードスタティックマッピングエントリと一致するサブスクライバのインバウンド接続またはアウトバウンド接続の場合、NAT 操作後もサブスクライバのポートは変更されません。加入者が開始したインターネット接続がワイルドカード静的マッピングエントリと一致する場合、NetScaler アプライアンスは接続を開始した加入者ポートと同じ番号の NAT ポートを割り当てます。同様に、インターネットホストは、加入者のポートと同じ番号の NAT ポートに接続することによって加入者のポートに接続されます。

サブスクライバー IPv6 アドレスのすべてのポートにアクセスできるように NetScaler アプライアンスを構成するには、次の必須パラメーター設定を使用してワイルドカードスタティックマップを作成します。

- プロトコル = すべて
- サブスクライバポート = *
- NAT ポート = *

ワイルドカードスタティックマップでは、1 対 1 のスタティックマップとは異なり、NAT IP パラメータの設定は必須です。また、ワイルドカードスタティックマップに割り当てられた NAT IP アドレスは、他のサブスクライバには使用できません。

コマンドラインインターフェイスを使用してワイルドカードスタティックマップを作成するには

コマンドプロンプトで入力します。

```
add lsn static <name> ALL <subscrIP> *  <natIP> * [-td <
   positive_integer>] [-destIP <ip_addr>

show lsn static
<!--NeedCopy-->
```

次のワイルドカードスタティックマップの設定例では、IP アドレスが 2001: DB 8:5001:: 3 のサブスクライバのすべてのポートに NAT IP 203.0.11.33 を介してアクセスできるようになっています。

```
add lsn static NAT64-WILDCARD-STATIC-1 ALL 2001:DB8:5001::3 *
   203.0.113.33 *
Done
<!--NeedCopy-->
```


大規模 **NAT64** のログ記録と監視

August 15, 2023

大規模な NAT64 情報をログに記録して、問題の診断やトラブルシューティング、法的要件を満たすことができます。統計カウンタを使用し、関連する現在のセッションを表示することで、大規模な NAT64 デプロイメントのパフォーマンスを監視できます。

大規模な **NAT64** のロギング

ISP が法的要件を満たし、いつでもトラフィックの送信元を特定するには、大規模な NAT64 情報を記録する必要があります。

大規模な NAT64 マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）。
- タイムスタンプ。
- エントリタイプ (マッピング)。
- マッピングエントリが作成されたか、削除されたか。
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID。
- NAT IP アドレスとポート。
- プロトコル名。
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が表示される場合があります。
  - エンドポイントに依存しないマッピングのため、宛先 IP アドレスとポートはログに記録されません。
  - アドレス依存のマッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレスポート依存のマッピング用にログに記録されます。

大規模な NAT64 セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイムスタンプ
- エントリータイプ (セッション)
- セッションが作成されるか削除されるか
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

次の表は、設定されたログサーバーに保存されている各タイプの大規模な NAT64 ログエントリのサンプルを示しています。ログエントリには、IPv6 アドレスが 2001: db 8:5001:9 のサブスクライバが、2016 年 4 月 7 日 14:07:57 グリニッジ標準時 14:07:57 から 14:10:59 までの間に、NAT IP: ポート 203.0.113. 63:45195 を介して宛先 IP: ポート 23.0.0. 1:80 に接続されたことが示されています。

ログエントリタイプ	サンプルログエントリ
セッション作成	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_SESSION 5532 0 : SESSION CREATED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
マッピング作成	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_ADDR_MAPPING 5533 0 : ADM CREATED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:TD 23.0.0.1:80, Protocol: TCP
セッション削除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_SESSION 25012 0 : SESSION DELETED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
マッピング削除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM DELETED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

構成の手順

LSN グループのロギングパラメータとセッションロギングパラメータを設定することで、大規模な NAT64 構成の大規模な NAT64 情報のロギングを設定できます。これらはグループレベルのパラメータで、デフォルトでは無効になっています。NetScaler アプライアンスは、ログとセッションログの両方のパラメーターが有効になっている場合にのみ、LSN グループの大規模な NAT64 セッションをログに記録します。

次の表は、ロギングおよびセッションロギングパラメータのさまざまな設定に対する LSN グループのロギング動作を示しています。

ログ	セッションロギング	ロギング動作
有効	有効	LSN マッピングエントリと LSN セッションをログに記録します。
有効	無効	LSN マッピングエントリをログに記録するが、LSN セッションは記録しない
無効	有効	マッピングエントリも LSN セッションもログに記録しない

CLI を使用して大規模な **NAT64** 情報をログに記録するには

LSN グループの追加時にロギングパラメータとセッションロギングパラメータを設定するには、コマンドプロンプトで次のように入力します。

```
add lsn group <groupname> -clientname <string> [-logging (ENABLED|
    DISABLED)] [-sessionLogging (ENABLED|DISABLED)]

show lsn group
<!--NeedCopy-->
```

既存の LSN グループのロギングパラメータとセッションロギングパラメータを設定するには、コマンドプロンプトで次のように入力します。

```
set lsn group <groupname> [-logging (ENABLED|DISABLED)] [-
    sessionLogging (ENABLED|DISABLED)]

show lsn group
<!--NeedCopy-->
```

構成例

この大規模な NAT64 構成の例では、LSN グループ LSN-NAT64-GROUP-1 のロギングパラメータとセッションロギングパラメータが有効になっています。

NetScaler アプライアンスは、サブスクライバー（ネットワーク 2001：DB 8:5001:: /96）からの接続に関する大規模な NAT64 セッションとマッピング情報をログに記録します。

設定例:

```
add lsn client LSN-NAT64-CLIENT-1 Done
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
```



```
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1  -logging ENABLED -sessionLogging
    ENABLED
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
<!--NeedCopy-->
```

大規模な **NAT64** 用の **MSISDN** 情報のロギング

モバイルステーション統合加入者電話番号（MSISDN）は、複数のモバイルネットワークにわたって加入者を一意に識別する電話番号です。MSISDN には、加入者の事業者を識別する国コードと国内宛先コードが関連付けられています。

NetScaler アプライアンスは、モバイルネットワークのサブスクライバーの大規模な NAT64 LSN ログエントリに MSISDNS を含めるように構成できます。LSN ログに MSISDN が含まれていると、ポリシーや法律に違反したモバイル加入者や、合法的な傍受機関から情報を要求されたモバイル加入者のバックトレースを迅速かつ正確に行うことができます。

次の LSN ログエントリの例には、LSN 設定のモバイル加入者からの接続の MSISDN 情報が含まれています。ログエントリには、MSISDN が E 164:5556543210 で IPv6 アドレスが 2001: db 8:5001:: 9 のモバイル加入者が、2016 年 4 月 7 日 14:07:57 グリニッジ標準時 14:07:57 から 14:10:59 までの間に、NAT IP: ポート 203.0.113. 63:45195 を介して宛先 IP: ポート 23.0.0. 1:80 に接続されたことが示されています。

ログエントリタイプ	サンプルログエントリ
セッション作成	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_SESSION 5532 0 : SESSION CREATED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
マッピング作成	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_ADDR_MAPPING 5533 0 : ADM CREATED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:TD 23.0.0.1:80, Protocol: TCP

ログエントリタイプ	サンプルログエントリ
セッション削除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_SESSION 25012 0 : SESSION DELETED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
マッピング削除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM DELETED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

構成の手順

MSISDN 情報を LSN ログに含めるには、次のタスクを実行します。

- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、LSN 設定の LSN ログに MSISDN 情報を含めるかどうかを指定するログサブスクライバー ID パラメータが含まれます。
- LSN ログプロファイルを LSN 設定の LSN グループにバインドします。作成した LSN ログプロファイル名にログプロファイル名パラメータを設定して、作成した LSN ログプロファイルを LSN 設定の LSN グループにバインドします。MSISDN 情報は、この LSN グループのモバイル加入者に関連するすべての LSN ログに含まれます。

CLI を使用して **LSN** ログプロファイルを作成するには　コマンドプロンプトで入力します。

```
add lsn logprofile <logprofilename> -logSubscriberID ( ENABLED |
   DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **NAT64 LSN** 設定の **LSN** グループにバインドするには　コマンドプロンプトで入力します。

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```


構成例

この NAT64 LSN 設定の例では、LSN ログプロファイル LOG-PROFILE-MSISDN-1 のログサブスクライバー ID パラメータが有効になっています。LOG-PROFILE-MSISDN-1 は LSN グループ LSN-NAT64-GROUP-1 にバインドされています。MSISDN 情報は、(ネットワーク 2001: DB 8:5001:: /96) モバイルサブスクライバからの接続の LSN セッションと LSN マッピングログに含まれます。

```
add lsn logprofile  LOG-PROFILE-MSISDN-1  -logSubscriberID ENABLED
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
   :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
   ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1 -logprofilename  LOG-PROFILE-MSISDN-1
Done
<!--NeedCopy-->
```

大規模 **NAT** 用のコンパクトロギング

LSN 情報のロギングは、ISP が法的要件を満たし、いつでもトラフィックの送信元を特定できるようにするために必要となる重要な機能の 1 つです。その結果、最終的には膨大な量のログデータになり、ISP はロギングインフラストラクチャの維持に多額の投資をする必要があります。

コンパクトロギングは、イベント名とプロトコル名のショートコードを含む表記変更を行うことでログサイズを小さくする手法です。たとえば、C はクライアント、SC は作成されたセッション、T は TCP です。コンパクトロギングでは、ログサイズを平均 40% 削減できます。

構成の手順

LSN 情報をコンパクト形式で記録するには、次のタスクを実行します。

1. LSN ログプロファイルを作成します。LSN ログプロファイルには、LSN 設定の情報をコンパクト形式で記録するかどうかを指定する Log Compact パラメータが含まれます。

2. LSN ログプロファイルを LSN 設定の LSN グループにバインドします。Log Profile Name パラメータを作成した LSN ログプロファイル名に設定して、作成した LSN ログプロファイルを LSN 設定の LSN グループにバインドします。この LSN グループのすべてのセッションとマッピングは、コンパクト形式でログに記録されます。

CLI を使用して **LSN** ログプロファイルを作成するには　コマンドプロンプトで入力します。

```
add lsn logprofile <logprofilename> -logCompact (ENABLED|DISABLED)

show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには　コマンドプロンプトで入力します。

```
bind lsn group <groupname> -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

NAT64 のサンプル構成**:**

```
add lsn logprofile  LOG-PROFILE-COMPACT-1 -logCompact ENABLED
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
   :300::/96
Done
add lsn group LSN-NAT64-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
   ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1  - logProfileName LOG-PROFILE-COMPACT-1
Done
<!--NeedCopy-->
```


HTTP ヘッダー情報のロギング

NetScaler アプライアンスは、NetScaler の大規模な NAT64 機能を使用する HTTP 接続のリクエストヘッダー情報をログに記録できます。HTTP リクエストパケットの次のヘッダー情報を記録できます。

- HTTP リクエストの送信先の URL
- HTTP リクエストで指定された HTTP メソッド
- HTTP リクエストで使用される HTTP バージョン
- HTTP 要求を送信したサブスクライバの IPv6 アドレス

ISP は HTTP ヘッダーログを使用して、複数のサブスクライバーの HTTP プロトコルに関連する傾向を確認できます。たとえば、ISP はこの機能を使用して、複数のサブスクライバーの中で最も人気のある Web サイトを検索できます。

構成の手順

次のタスクを実行して、HTTP ヘッダー情報を記録するように NetScaler アプライアンスを構成します。

- HTTP ヘッダーログプロファイルを作成します。HTTP ヘッダーログプロファイルは、ロギングを有効または無効にできる HTTP ヘッダー属性 (URL や HTTP メソッドなど) のコレクションです。
- HTTP ヘッダーを大規模な NAT64 構成の LSN グループにバインドします。HTTP ヘッダーログプロファイル名パラメーターを作成した HTTP ヘッダーログプロファイルの名前に設定して、HTTP ヘッダーログプロファイルを LSN 構成の LSN グループにバインドします。次に、NetScaler アプライアンスは、LSN グループに関連するすべての HTTP リクエストの HTTP ヘッダー情報を記録します。HTTP ヘッダーログプロファイルは複数の LSN グループにバインドできますが、LSN グループには 1 つの HTTP ヘッダーログプロファイルしか割り当てることができません。

コマンドラインインターフェイスを使用して **HTTP** ヘッダーログプロファイルを作成するには　コマンドプロンプトで入力します。

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **HTTP** ヘッダーログプロファイルを **LSN** グループにバインドするにはコマンドプロンプトで入力します。

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
```



```
<!--NeedCopy-->
```

構成例

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done
add lsn client LSN-NAT64-CLIENT-1 Done
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
   :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
   ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1 -httphdrlogprofilename HTTP-HEADER-LOG
   -1
Done
<!--NeedCopy-->
```

現在の大規模な **NAT64** セッションの表示

NetScaler アプライアンス上の不要なセッションや非効率的なセッションを検出するために、現在の大規模な NAT64 セッションを表示できます。選択パラメータに基づいて、すべてまたは一部の大規模な NAT64 セッションを表示できます。

> 注
>
> NetScaler アプライアンス上に 100 万を超える大規模な NAT64 セッションが存在する場合、Citrix では、選択パラメーターを使用して選択した大規模な NAT64 セッションをすべて表示するのではなく、選択パラメーターを使用して表示することをお勧めします。

コマンドラインインターフェイスを使用して大規模な **NAT64** セッションをすべて表示するには

コマンドプロンプトで入力します。

```
show lsn session  -nattype NAT64
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して大規模な **NAT64** セッションを選択的に表示するには

コマンドプロンプトで入力します。

```
show lsn session  -nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname
    <string>] [-natIP <ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

大規模な **NAT64** 統計情報の表示

大規模な NAT64 モジュールに関連する統計を表示し、そのパフォーマンスを評価したり、問題をトラブルシューティングしたりできます。すべての大規模な NAT64 構成または特定の大規模な NAT64 構成の統計の概要を表示できます。統計カウンターには、NetScaler アプライアンスが最後に再起動されてからのイベントが反映されます。NetScaler アプライアンスを再起動すると、これらのカウンタはすべて 0 にリセットされます。

コマンドラインインターフェイスを使用して大規模な **NAT64** の統計情報をまとめて表示するには

コマンドプロンプトで入力します。

```
stat lsn nat64
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、指定した大規模な **NAT64** 構成の統計情報を表示するには

コマンドプロンプトで入力します。

```
stat lsn group <groupname>
<!--NeedCopy-->
```

大規模な **NAT64** セッションのクリア

不要または非効率的な大規模 NAT64 セッションを NetScaler アプライアンスから削除できます。アプライアンスは、これらのセッションに割り当てられたリソース（NAT IP アドレス、ポート、メモリなど）をすぐに解放し、リソースを新しいセッションで使用できるようにします。アプライアンスは、削除されたセッションに関連する後続のパケットもすべてドロップします。NetScaler アプライアンスからすべてまたは選択した大規模 NAT64 セッションを削除できます。

コマンドラインインターフェイスを使用して大規模な **NAT64** セッションをすべてクリアするには

コマンドプロンプトで入力します。


```
flush lsn session  -nattype NAT64

show lsn session  -nattype NAT64
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して選択的に大規模な **NAT64** セッションをクリアするには

コマンドプロンプトで入力します。

```
flush lsn session  -nattype NAT64 [-network6 <ipv6_addr|*>] [-
   clientname <string>] [-natIP <ip_addr> [-natPort <port>]]

show lsn session  -nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname
    <string>] [-natIP <ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

設定例:

NetScaler アプライアンスに存在する大規模な NAT64 セッションをすべてクリアします

```
flush lsn session   -nattype NAT64
Done
<!--NeedCopy-->
```

クライアントエンティティ LSN-NAT64-CLIENT-1 に関連する大規模な NAT64 セッションをすべてクリアします

```
flush lsn session  -nattype NAT64 -clientname LSN-NAT64-CLIENT-1
Done
<!--NeedCopy-->
```

LSN クライアントエンティティ LSN-NAT64-CLIENT-2 のサブスクライバネットワーク (2001: DB 8:5001:: /96) に関連する大規模な NAT64 セッションをすべてクリアします

```
flush lsn session  -nattype NAT64  -network6 2001:DB8:5001::/96 -
   clientname LSN-NAT64-CLIENT-2
Done
<!--NeedCopy-->
```

IPFIX ロギング

NetScaler アプライアンスは、LSN イベントに関する情報をインターネットプロトコルフロー情報エクスポート (IPFIX) 形式で、構成済みの IPFIX コレクターのセットに送信することをサポートしています。アプライアンスは既存の AppFlow 機能を使用して、LSN イベントを IPFIX 形式で IPFIX コレクターに送信します。

IPFIX ベースのロギングは、次の NAT64 関連イベントで使用できます。

- LSN セッションの作成または削除。
- LSN マッピングエントリの作成または削除。
- デターミニスティック NAT におけるポートブロックの割り当てまたは割り当て解除
- ダイナミック NAT におけるポートブロックの割り当てまたは割り当て解除
- サブスクライバセッションクォータを超えると。

IPFIX ロギングを設定する前に考慮すべきポイント

IPSec ALG の設定を開始する前に、次の点を考慮してください。

- NetScaler ADC アプライアンスで AppFlow 機能および IPFIX コレクタを構成する必要があります。手順については、AppFlow 機能の構成を参照してください。

構成の手順

LSN 情報を IPFIX 形式で記録するには、次のタスクを実行します。

- **AppFlow** 構成で **LSN** ロギングを有効にします。AppFlow 構成の一部として LSN ロギングパラメータを有効にします。
- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、IPFIX 形式のログ情報を有効または無効にする IPFIX パラメータが含まれます。
- **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドします。LSN ログプロファイルを 1 つまたは複数の LSN グループにバインドします。バインドされた LSN グループに関連するイベントは IPFIX 形式で記録されます。

CLI を使用して **AppFlow** 構成で **LSN** ロギングを有効にするには　コマンドプロンプトで入力します。

```
set appflow param -lsnLogging ( ENABLED | DISABLED )

show appflow param
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを作成するには、コマンドプロンプトで次のように入力します　コマンドプロンプトで入力します。

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```


CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには　コマンドプロンプトで入力します。

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

GUI を使用して **LSN** ログプロファイルを作成するには　[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[ログ] タブをクリックして、ログプロファイルを追加します。

GUI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには

1. [システム] > [大規模 **NAT**] > [**LSN** グループ**]** に移動し、**LSN** グループを開きます。
2. 詳細設定で、**+ Log Profile** をクリックして、作成したログプロファイルを LSN グループにバインドします。

大規模 **NAT64** のポート制御プロトコル

August 15, 2023

NetScaler アプライアンスは、大規模 NAT（LSN）用の Port Control Protocol（PCP）をサポートするようになりました。ISP の加入者アプリケーションの多くは、インターネットからアクセスできる必要があります (たとえば、インターネット経由で監視する IP カメラなどの Internet of Things (IOT) デバイス)。この要件を満たす 1 つの方法は、静的な大規模 NAT（LSN）マップを作成することです。しかし、加入者の数が非常に多い場合、スタティック LSN NAT マップを作成することは現実的な解決策ではありません。

Port Control Protocol（PCP）を使用すると、加入者は自分や他のサードパーティデバイス用の特定の LSN NAT マッピングを要求できます。大規模な NAT デバイスは LSN マップを作成し、サブスクライバに送信します。サブスクライバは、サブスクライバに接続できる NAT IP アドレス:NAT ポートをインターネット上のリモートデバイスに送信します。

アプリケーションは通常、LSN マッピングがタイムアウトしないように、大規模な NAT デバイスに頻繁にキープアライブメッセージを送信します。PCP は、アプリケーションが LSN マッピングのタイムアウト設定を学習できるようにすることで、このようなキープアライブメッセージの頻度を減らすのに役立ちます。これにより、ISP のアクセスネットワークの帯域幅消費とモバイルデバイスのバッテリー消費量を削減できます。

PCP はクライアント/サーバーモデルで、UDP トランスポートプロトコルで実行されます。NetScaler アプライアンスは PCP サーバーコンポーネントを実装し、RFC 6887 に準拠しています。

構成の手順

PCP を設定するには、次のタスクを実行します。

- **(**オプション**) PCP** プロファイルを作成します。PCP プロファイルには、PCP 関連パラメータ（マッピングやピアの PCP リクエストのリスニングなど）の設定が含まれます。PCP プロファイルは PCP サーバーにバインドできます。PCP サーバーにバインドされた PCP プロファイルは、そのすべての設定を PCP サーバーに適用します。PCP プロファイルは複数の PCP サーバーにバインドできます。デフォルトでは、デフォルトのパラメータ設定を持つ 1 つの PCP プロファイルがすべての PCP サーバーにバインドされます。PCP サーバーにバインドした PCP プロファイルは、そのサーバーのデフォルトの PCP プロファイル設定よりも優先されます。デフォルトの PCP プロファイルには、次のパラメータ設定があります。
  - マッピング: 有効
  - ピア: 有効
  - 最小マップ寿命:120 秒
  - 最大ライフタイム:86400 秒
  - アナウンス数:10
  - サードパーティ: 無効
- **PCP** サーバーを作成し、それに **PCP** プロファイルをバインドします。NetScaler アプライアンスに PCP サーバーを作成して、サブスクライバーからの PCP 関連のリクエストとメッセージを受信します。PCP サーバーにアクセスするには、サブネット IP (SNIP) または (SNIP6) アドレスを割り当てる必要があります。デフォルトでは、PCP サーバーはポート 5351 で受信します。
- **PCP** サーバーを **LSN** 構成の **LSN** グループにバインドします。作成した PCP サーバーを指定する PCP サーバーパラメーターを設定して、作成した PCP サーバーを LSN 構成の LSN グループにバインドします。作成された PCP サーバには、この LSN グループの加入者だけがアクセスできます。

> 注
>
> 大規模な NAT 設定の PCP サーバは、ACL ルールで識別されたサブスクライバからの要求には対応しません。

CLI を使用して **PCP** プロファイルを作成するには

コマンドプロンプトで入力します。

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

CLI を使用して **PCP** サーバーを作成するには

コマンドプロンプトで入力します。


```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```

NAT64 のサンプル構成

次の設定例では、PCP-PROFILE-1 の PCP 設定を持つ PCP サーバー PCP-SERVER-1 が LSN グループ LSN-NAT64-GROUP-1 にバインドされています。PCP-SERVER-1 は、ネットワーク 2001: DB 8:5001:: /96 の IPv6 サブスクライバーからの PCP リクエストを処理します。

設定例:

```
add pcp profile PCP-PROFILE-1 -minMapLife 400
Done
add pcp server PCP-SERVER-1 2001:DB8:6001::90  -pcpProfile PCP-PROFILE
    -1
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1  - pcpServer PCP-NAT64-SERVER-1
Done
<!--NeedCopy-->
```

クラスターセットアップの **LSN64**

August 15, 2023

NetScaler クラスターのセットアップでは、大規模な NAT64 構成がサポートされます。

NetScaler クラスターは、単一のシステムとして構成および管理される NetScaler アプライアンスのグループです。NetScaler クラスターはスケーラビリティと可用性を提供します。クラスター構成の各 NetScaler アプライアンス

は、独立した LSN エンティティとして機能し、単一のシステムとして管理されます。

クラスタ設定の LSN 設定は、特定の LSN IP アドレスプールが一度に 1 つのノードだけが所有する点を除いて、スタンドアロンアプライアンスと同じです。つまり、LSN IP プールエンティティは、特定のノードのスポットエンティティとして設定されます。クラスタ設定のすべてのノードには、特定の LSN IP プールエンティティを設定できます。LSN セッションに関連するパケットが NAT 操作を実行したのと同じクラスタノードで受信されるように、ポリシーベースのバックプレーン (PBS) ステアリングが設定されています。PBS は、LSN セッションの受信関連パケットを同じクラスタノードに転送します。

設定例:

```
add lsn client LSN-NAT64-CLIENT-1

Done

bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96

Done

add lsn pool LSN-NAT64-POOL-1

Done

bind lsn pool LSN-NAT64-POOL-1 -ownerNode 1  203.0.113.61 -
    203.0.113.70

Done

bind lsn pool LSN-NAT64-POOL-1 -ownerNode 2  203.0.113.101 -
    203.0.113.110

Done

add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96

Done

add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1

Done

bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1

Done

add ns acl6 NAT64-DFD ALLOW -srcIPv6 = 2001:DB8:5001:: -type DFD -
    dfdhash SIP -dfdprefix 64

Done

```



```
apply ns acls6 -type DFD

Done
<!--NeedCopy-->
```

トランスレーションを使用したアドレスとポートのマッピング

August 15, 2023

変換を使用するアドレスとポートのマッピング（MAP-T）は、_.MAP-T IPv6 インフラストラクチャを備えた ISP が IPv4 加入者を IPv4 インターネットに接続するための IPv6 移行ソリューションであり、ステートレス IPv4 および IPv6 アドレス変換テクノロジーに基づいて構築されています。MAP-T は、カスタマーエッジ（CE）デバイスとボーダールーター（ISP コアネットワーク内）で二重変換（IPv4 から IPv6、またはその逆）を実行するメカニズムです。

MAP-T 展開では、CE デバイスはステートフル NAPT44 変換とステートレス NAT46 変換の組み合わせを実装します。CE デバイスは NAT-IP とポートブロックを取得し、DHCPv6 またはその他の方法による変換に使用されます。

加入者デバイスからの IPv4 パケットが CE デバイスに到着すると、CE デバイスは NAPT44 を実行して NAPT44 バインディング情報を保存します。NAT44 変換後、パケットは NAT46 変換され、ISP のコアネットワークにあるボーダールーター (BR) デバイスに転送されます。BR デバイスは CE デバイスから IPv6 パケットを受信し、IPv6 ヘッダーに埋め込まれた NAT-IP とポートブロックを抽出して検証し、IPv4 パケットを IPv4 インターネットに転送します。BR は、インターネットから IPv4 パケットを受信すると、IPv4 パケットを IPv6 パケットに変換し、その IPv6 パケットを CE デバイスに送信します。

MAP-T は BR デバイス上ではステートレスなので、BR デバイスがトラフィックに対して NAT を実行する必要はありません。代わりに、NAT 機能は CE デバイスに委任されます。BR デバイスのこの委任機能とステートレス機能により、BR の導入をトラフィック量に比例して拡張できます。

NetScaler アプライアンスは、RFC 7599 で説明されている MAP-T ソリューションの BR 機能を実装しています。

MAP-T の設定

NetScaler アプライアンスでの MAP-T の設定は、次のタスクで構成されます。

- デフォルトのマッピングルールを追加
- 基本的なマッピングルールを追加
- CE デバイスの IPv4 NAT アドレス範囲を基本的なマッピングルールにバインドする
- マップドメインを追加し、基本マッピングルールとデフォルトマッピングルールをドメインにバインドします

CLI を使用してデフォルトのマッピングルールを追加するには

コマンドプロンプトで入力します。

```
add MapDmr <name> -BRIpv6Prefix ( <ipv6_addr> | <*> )

show MapDmr <name>
<!--NeedCopy-->
```

CLI を使用して基本的なマッピングルールを追加するには

コマンドプロンプトで入力します。

```
add MapBmr <name> -RuleIpv6Prefix <ipv6_addr> | <*>  [-psidoffset <
    positive_integer>] [-EAbitLength <positive_integer>]  [-psidlength <
    positive_integer>]

show MapBmr <name>
<!--NeedCopy-->
```

CLI を使用して **CE** デバイスの **IPv4 NAT** アドレス範囲を基本的なマッピングルールにバインドするには

コマンドプロンプトで入力します。

```
bind MapBmr <name> (-network <ip_addr>  [-netmask <netmask>])

show MapBmr <name>
<!--NeedCopy-->
```

CLI を使用してマップドメインを追加するには

コマンドプロンプトで入力します。

```
add MapDomain <name> -MapDmrName <string>

show MapDomain <name>
<!--NeedCopy-->
```

CLI を使用して基本的なマッピングルールをマップドメインにバインドするには

コマンドプロンプトで入力します。

```
bind MapDomain <name> -MapBmrName <string>

show MapDomain <name>
```



```
<!--NeedCopy-->
```

設定例

```
add mapdmr DMR-1 -BRIpv6Prefix 2002:db8::/64

Done

add mapbmr BMR-1 -ruleIpv6Prefix 2002:db8:89ab::/48 -eAbitLength 16 -
    psidlength 8 -psidoffset 6

Done

bind mapbmr BMR-1 -network 192.0.1.0 -netmask 255.255.255.0

Done

add MapDomain MAP-DOMAIN-1 -mapdmrname DMR-1

Done

bind MapDomain MAP-DOMAIN-1 -mapbmrname BMR-1
Done
<!--NeedCopy-->
```

Telco 利用者管理

August 15, 2023

通信事業者ネットワークの加入者数はかつてない速さで増加しており、その管理はサービスプロバイダーにとって課題となっています。より新しく、より高速で、よりスマートなデバイスにより、ネットワークと加入者管理システムに対する需要が高まっています。各加入者に同じサービス標準を提供することはもはや現実的ではなく、加入者ごとにトラフィックを処理することが不可欠です。

NetScaler アプライアンスは、ポリシーと課金ルール機能（PCRF）に保存されている情報に基づいて加入者をプロファイリングするインテリジェンスを提供します。モバイル加入者がインターネットに接続すると、パケットゲートウェイは加入者に IP アドレスを関連付け、データパケットをアプライアンスに転送します。アプライアンスはサブスクライバ情報を動的に受信するか、スタティックサブスクライバを設定できます。この情報により、アプライアンスは、コンテンツスイッチング、統合キャッシュ、リライト、レスポンダなどの豊富なトラフィック管理機能をサブスクライバごとに適用してトラフィックを管理できます。

サブスクライバーを管理するように NetScaler アプライアンスを構成する前に、サブスクライバーセッションを格納するモジュールにメモリを割り当てる必要があります。ダイナミックサブスクライバの場合は、アプライアンスがセッション情報を受信するインターフェイスを設定する必要があります。スタティックサブスクライバには ID を割り当てる必要があり、それらをポリシーに関連付けることができます。

次の操作もできます。

- 加入者ポリシーの適用と管理。
- 完全な IPv6 アドレスではなく IPv6 プレフィックスだけを使用して加入者を一意に識別するようにアプライアンスを設定します。
- ポリシーを使用して、動的サブスクライバとスタティックサブスクライバの両方の TCP トラフィックを最適化します。これらのポリシーは、さまざまな TCP プロファイルをさまざまなタイプのユーザーに関連付けます。
- NetScaler アプライアンス上のアイドルセッションを管理します。
- ログサーバーへのロギングを有効にします。
- 削除されたサブスクライバセッションの LSN セッションを削除します。

サブスクライバセッションストアモジュールへのメモリ割り当て

各サブスクライバセッションエントリは 1 KB のメモリを消費します。任意の時点で 500,000 のサブスクライバセッションを保存するには、500 MB のメモリが必要です。この値は、「show extendedmemoryparam」コマンドの出力の一部として表示される最小メモリ要件に追加する必要があります。次の例では、出力は 3 つのパケットエンジンと 8 GB のメモリを搭載した NetScaler VPX インスタンスのものです。

このアプライアンスに 500,000 のサブスクライバセッションを保存するには、設定されたメモリが 2058+500 MB（500,000 x 1 KB = 500 MB）である必要があります。

> 注
>
> 設定するメモリは 2 MB の倍数でなければならず、最大メモリ使用量制限を超えてはなりません。変更を有効にするには、アプライアンスを再起動する必要があります。

例

```
show extendedmemoryparam
       Extended Memory Global Configuration. This memory is utilized by
           LSN and Subscriber Session Store Modules:
        Active Memory Usage: 0 MBytes
        Configured Memory Limit: 0 MBytes
        Minimum Memory Required: 2058 MBytes
        Maximum Memory Usage Limit: 2606 MBytes
    Done
    set extendedmemoryparam -memLimit 2558
    Done
    show extendedmemoryparam
            Extended Memory Global Configuration. This memory is
               utilized by LSN and Subscriber Session Store Modules:

            Active Memory Usage: 2558 MBytes
            Configured Memory Limit: 2558 MBytes
            Minimum Memory Required: 2058 MBytes
```



```
16                Maximum Memory Usage Limit: 2606 MBytes
17      Done
18  <!--NeedCopy-->
```

ダイナミックサブスクライバー用のインターフェースの設定

NetScaler アプライアンスは、次のいずれかのタイプのインターフェイスを介して加入者情報を動的に受信します。

- Gx インターフェイス
- RADIUS インターフェイス
- RADIUS および Gx インターフェイス

> 注
>
> - NetScaler リリース 12.0 ビルド 57.19 以降、クラスタ展開では Gx インターフェイスがサポートされます。詳細については、クラスタトポロジの Gx インターフェイスを参照してください。
> - HA セットアップでは、サブスクライバセッションはセカンダリノードで継続的に同期されます。フェールオーバーが発生しても、サブスクライバ情報はセカンダリノードで引き続き使用できます。

Gx インターフェース

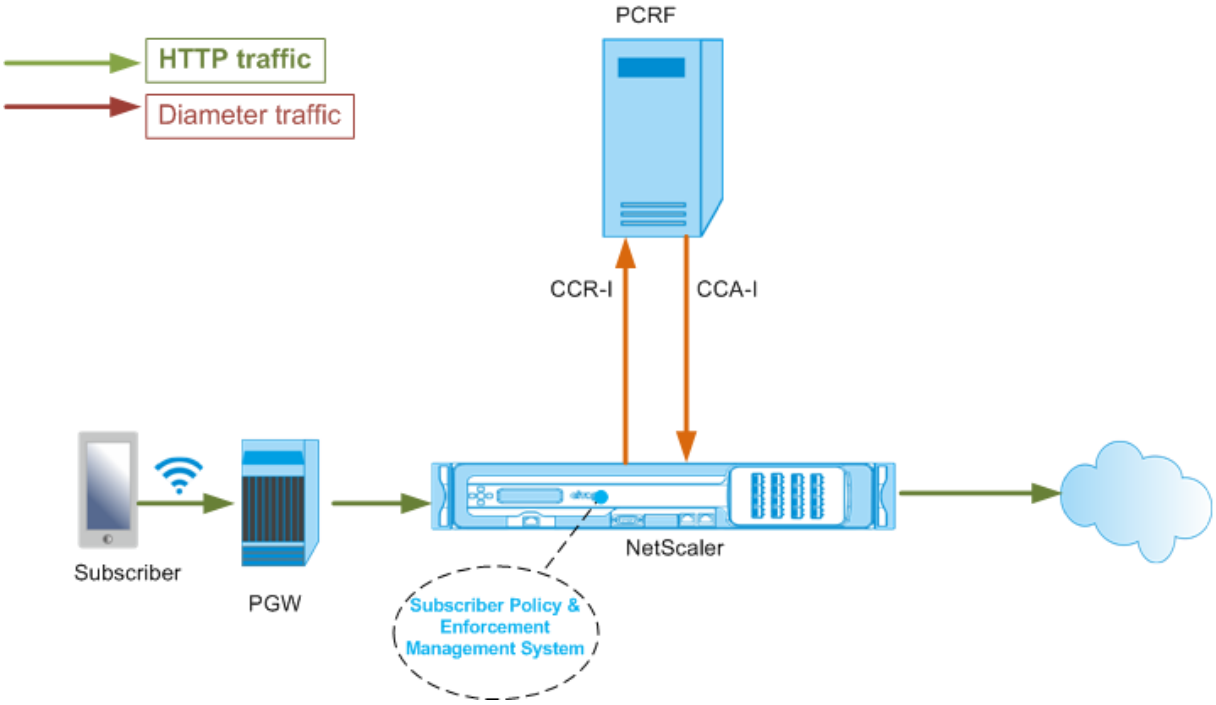
Gx インターフェイス（3GPP 29.212 で規定）は Diameter プロトコルに基づく標準インターフェイスで、PCRF と電話会社ネットワーク内のポリシーおよび課金機能（PCEF）エンティティとの間でポリシー制御と課金ルールを交換できます。

IP-CAN セッションが確立されると、パケットゲートウェイは MSISDN などの加入者 ID と、加入者に関するフレーム IP アドレス情報を Diameter メッセージとして PCRF に転送します。データパケットがパケットゲートウェイ（PGW）からアプライアンスに到着すると、アプライアンスはサブスクライバの IP アドレスを使用して PCRF にクエリを実行し、サブスクライバ情報を取得します。これはセカンダリ PCEF 機能とも呼ばれます。

アプライアンスが Gx インターフェイスを介して受信したポリシーおよび課金制御（PCC）ルールは、サブスクライバセッション中、つまり PCRF がセッションリリース原因 AVP とともに Re-Auth-Request（RAR）メッセージを送信するか、サブスクライバセッションが CLI または設定ユーティリティから終了されるまで、アプライアンスに保存されます。既存のサブスクライバに更新がある場合、PCRF はその更新を RAR メッセージで送信します。サブスクライバセッションは、サブスクライバがネットワークにログオンしたときに開始され、サブスクライバがログオフすると終了します。

> 注: PCRF サーバーがダウンしている場合、NetScaler アプライアンスは保留中または受信中の Gx サブスクライバーリクエストに対してネガティブセッションを作成します。PCRF サーバーが再び稼働すると、NetScaler アプライアンスは、特定のサブスクライバーリクエストを実行する前にネガティブセッションの有効期限が切れるのを待って、大量のリクエストを防ぎます。

次の図は、高レベルのトラフィックフローを示しています。データプレーントラフィックは HTTP であることを前提としています。アプライアンスは Gx インターフェイスを介して PCRF サーバにクレジットコントロールリクエスト（CCR）を送信し、クレジットコントロールアンサー（CCA）で PCC ルールと、オプションで特定のサブスクライバに適用されるその他の情報（無線アクセス技術（RAT）タイプなど）を受信します。PCC ルールには、1 つ以上のポリシー (ルール) 名とその他のパラメータが含まれます。アプライアンスはこの情報を使用して、アプライアンスに保存されている定義済みのルールを取得し、トラフィックの流れを指示します。また、この情報は、サブスクライバセッション中にサブスクライバポリシーとエンフォースメント管理システムに保存されます。サブスクライバセッションが終了すると、アプライアンスはサブスクライバに関するすべての情報を破棄します。



次の例は、Gx インターフェイスを設定するためのコマンドを示しています。コマンドは太字で表示されます。

Gx インターフェイスを設定するには、次のタスクを実行します　各 Gx インターフェイスに DIAMETER サービスを追加します。次に例を示します:

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868

add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

アドレス指定できない DIAMETER 負荷分散仮想サーバーを追加し、ステップ 1 で作成したサービスをこの仮想サーバーにバインドします。複数のサービスの場合は、特定のセッションが同じ PCRF サーバーによって処理されるように、PersistenceType と PersistAvPno を指定します。次に例を示します:

```
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

```



```
bind lb vserver vdiam pcrf-svc1

bind lb vserver vdiam pcrf-svc2
<!--NeedCopy-->
```

NetScaler の直径 ID とレルムを設定します。ID とレルムは、Gx クライアントから送信される直径メッセージのオリジンホストおよびオリジンレルム AVP として使用されます。次に例を示します:

```
set ns diameter  -identity netscaler.com  -realm com
<!--NeedCopy-->
```

ステップ 2 で作成した仮想サーバーを PCRF 仮想サーバーとして使用するように Gx インターフェイスを設定します。Gx クライアントから送信される直径メッセージの宛先レルム AVP として使用する PCRF レルムを指定します。次に例を示します:

```
set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf.com
<!--NeedCopy-->
```

サブスクライバインターフェイスタイプを GXOnly に設定します。次に例を示します:

```
set subscriber param -interfaceType GxOnly
<!--NeedCopy-->
```

Gx インターフェイスの設定とステータスを確認するには、次のように入力します。

```
show subscriber gxinterface
<!--NeedCopy-->
```

例

```
show subscriber gxinterface
    Gx Interface parameters:
        PCRF Vserver: vdiam (DOWN)
        Gx Client Identity...:  netscaler1.com
        Gx Client Realm ..........:  com
        PCRF Realm: epc.mnc030.mcc234.3gppnetwork.org
        Hold Packets On Subscriber Absence: YES
        CCR Request Timeout: 4 Seconds
        CCR Request Retry Attempts: 1
        Gx HealthCheck enabled: NO
        Gx HealthCheck TTL : 30 Seconds
        CER Request Timeout: 10 Seconds
        RevalidationTimeout: 30 Seconds
        NegativeTTL: 60 Seconds
        NegativeTTL Limited Success: NO
        Purge SDB on Gx Failure: YES
        ServicePath AVP code: 262099    ServicePath AVP VendorID: 3845
        PCRF Connection State: PCRF is not ready
    Done

<!--NeedCopy-->
```


引数

vServer　Gx 接続が確立される負荷分散またはコンテンツスイッチング仮想サーバーの名前。仮想サーバーのサービスタイプは DIAMETER または SSL_DIAMETER でなければなりません。このパラメータはサービスパラメータと相互に排他的です。そのため、Gx インターフェースでサービスと仮想サーバーの両方を設定することはできません。

サービス　Gx 接続が確立されている PCRF に対応する DIAMETER または SSL_DIAMETER サービスの名前。このパラメータは vserver パラメータと相互に排他的です。そのため、Gx インターフェースでサービスと仮想サーバーの両方を設定することはできません。

pcrfRealm　メッセージのルーティング先となる PCRF のレルム。これは、NetScaler Gx クライアントが（Diameter ameter ノードとして）デスティネーションレルム AVP で使用されるレルムです。

holdOnSubscriberAbsence　Yes に設定すると、加入者セッション情報が PCRF サーバから取得されるまでパケットが保留されます。No に設定すると、サブスクライバセッション情報が PCRF サーバから取得されるまで、デフォルトのサブスクライバプロファイルが適用されます。デフォルトのサブスクライバプロファイルが設定されていない場合、サブスクライバ属性を使用するエクスプレッションには UNDEF が設定されます。

requestTimeout　Gx CCR リクエストが完了するまでの時間 (秒単位)。この時間内にリクエストが完了しない場合、リクエストは RequestRetryAttempts パラメータで指定された回数だけ再送信されます。再送信しても要求が完了しない場合は、デフォルトのサブスクライバプロファイルがこのサブスクライバに適用されます。デフォルトのサブスクライバプロファイルが設定されていない場合、サブスクライバ属性を使用するエクスプレッションには UNDEF が設定されます。ゼロはタイムアウトを無効にします。デフォルト値:10

requestRetryAttempts　requestTimeout パラメータで指定された値の範囲内でリクエストが完了しなかった場合に、リクエストを再送信する必要がある回数を指定します。デフォルト値:3。

healthCheck　Gx ピアのインラインヘルスチェックを有効にするには、「Yes」に設定します。有効にすると、NetScaler は DWR パケットを PCRF サーバーに送信します。Gx セッションがアイドル状態になると、HealthCheck タイマーが期限切れになり、DWR パケットが開始され、PCRF サーバがアクティブかどうかが確認されます。デフォルト値：いいえ。
注：このパラメーターは、NetScaler 12.1 ビルド 51.xx 以降でサポートされています。

healthCheckTTL ウォッチドッグ監視用に定義された時間（秒単位）。ヘルスチェックの TTL 時間が経過すると、DWR が送信され、PCRF サーバのステータスがチェックされます。CCR、CCA、RAR、または RAA メッセージのいずれかがタイマーをリセットします。
最小値:6 秒。デフォルト値:30 秒。
注：このパラメーターは、NetScaler 12.1 ビルド 51.xx 以降でサポートされています。

cerRequestTimeout 機能交換リクエストを再送信するまでの時間（秒単位）。NetScaler は、この構成された時間内に PCRF から CEA を受信しない場合、新しい CER メッセージを開始します。
PCRF サーバから応答がない場合、アプライアンスは CER メッセージを 5 回送信しようとします。CER メッセージが 5 回送信されても応答がない場合、アプライアンスは TCP 接続を閉じて障害を報告します。タイムアウト値が 0 に設定されている場合、アプリケーションヘルスチェック機能は無効になります。
最小値:0 秒。デフォルト値:0 秒。
注：このパラメーターは、NetScaler 12.1 ビルド 51.xx 以降でサポートされています。

revalidationTimeout セッションでの PCRF アクティビティの後に Gx CCR-U 要求が送信されるまでの時間（秒単位）。RAR または CCA メッセージはすべてタイマーをリセットします。値がゼロの場合、アイドルタイムアウトは無効になります。

negativeTTL サーバがダウンしている、応答がない、または応答が受信されなかったために PCRF で解決されなかったセッションに対して Gx CCR-I 要求が再送信されるまでの時間（秒単位）。PCRF サーバを絶えずポーリングする代わりに、TTL が負の場合、アプライアンスは未解決のセッションを保持することになります。ネガティブセッションの場合、アプライアンスはデフォルトのサブスクライバプロファイル（設定されている場合）から、RADIUS アカウンティングメッセージ（受信されている場合）から属性を継承します。値がゼロの場合、ネガティブセッションは無効になります。サブスクライバセッションを取得できなくても、アプライアンスはネガティブセッションをインストールしません。デフォルト値:600

negativeTTLLimitedSuccess Yes に設定すると、部分成功応答コード (2002) に対してネガティブセッションが作成されます。「いいえ」に設定すると、通常のセッションが作成されます。デフォルト値：いいえ。

このパラメーターは、NetScaler 12.1 ビルド 49.xx 以降でサポートされています。

purgeSDBonGxFailure Yes に設定すると、Gx インターフェイスに障害が発生したときにサブスクライバデータベースがフラッシュされます。Gx インターフェイスの障害には、DWR モニタリング (有効な場合) とネットワークヘルスチェック (有効な場合) の両方が含まれます。Yes に設定すると、すべてのサブスクライバセッションがクリアされます。
デフォルト値: いいえ
注：このパラメーターは、NetScaler 12.1 ビルド 51.xx 以降でサポートされています。

servicePathAVP　PCRF がサブスクライバに適用されるサービスパスを送信する AVP コード。

servicePathVendorid　PCRF がサブスクライバに適用されるサービスパスを送信する AVP のベンダー ID。

GUI を使用して **Gx** インターフェイスを設定するには

1. [トラフィック管理] > [サブスクライバー] > [パラメータ] に移動します。
2. [サブスクライバーパラメータの構成] をクリックします。
3.「インタフェースタイプ」で「**GxOnly**」を選択します。
4. すべての必須パラメータの値を指定します。
5.［**OK**］をクリックします。

確立された **Gx** 接続での転送障害を検出

注：この機能は NetScaler 12.1 ビルド 51.xx 以降でサポートされています。

NetScaler アプライアンスは、デバイスウォッチドッグリクエスト（DWR）メッセージとデバイスウォッチドッグ応答（DWA）メッセージを使用して、確立された Gx 接続での転送障害を検出するように構成できます。

Gx セッションが確立されると、セッションがアイドル状態かどうかを検出する定義済みのタイマーがトリガーされます。アイドルタイムタイマーの期限が切れると、DWR メッセージが送信されます。アイドルタイムタイマーは、NetScaler アプライアンスが確立された Gx セッションでメッセージを受信するたびにリセットされます。ピアが使用できるかどうかは、DWR メッセージの送信後に DWA メッセージに基づいて確認されます。

- DWA を受信すると、ピアのアベイラビリティが確認され、ウォッチドッグタイマーがリセットされます。
- DWA が受信されず、ウォッチドッグタイマーが 2 回連続して期限切れになった場合、セッションはダウンしていてピアが使用できないと見なされます。アプライアンスはセッションを終了し、Gx ピアとの新しいセッションを確立しようとします。

ウォッチドッグタイマーが応答なしで 2 回期限切れになると、NetScaler アプライアンスは Gx 接続に障害があると見なし、接続終了を開始します。接続が閉じられると、他のウォッチドッグリクエストは Gx ピアに送信されません。NetScaler アプライアンスは、PCRF リクエストに対して次に使用可能な Gx セッションを使用します。

CLI を使用して確立された **Gx** 接続での転送障害を検出するには　コマンドプロンプトで入力します。

```
set subscriber gxInterface [-vServer <string>] [-service <string>] [-
    healthCheck ( YES | NO )] [-healthCheckTTL<positive_integer>][-
    cerRequestTimeout <positive_integer>] [-purgeSDBonGxFailure ( YES |
    NO )]
<!--NeedCopy-->
```

例**:**

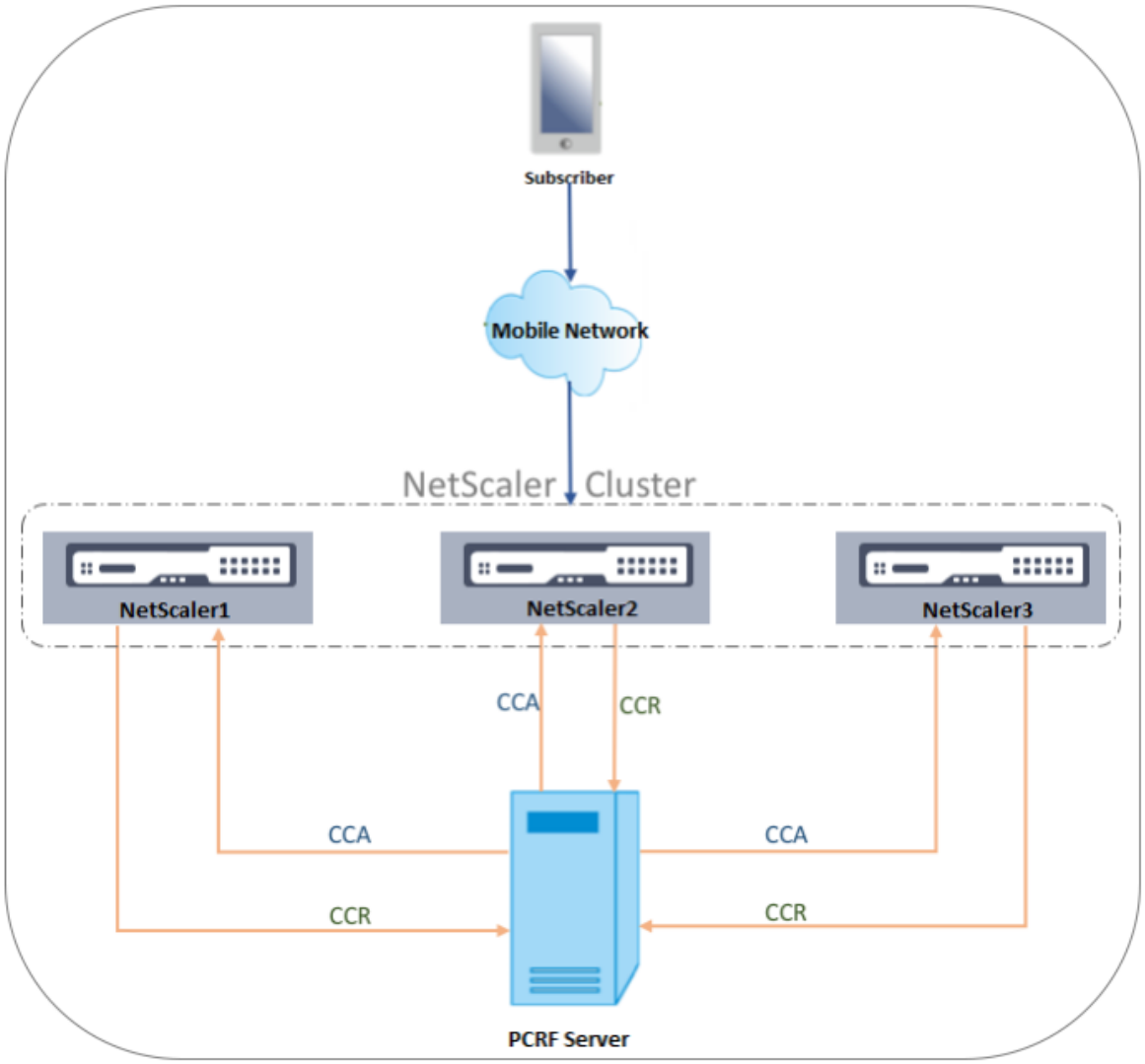

```
set subscriber gxInterface set subscriber gxInterface -vServer vdiam -
    healthCheck YES -healthCheckTTL 31 -cerRequestTimeout 15
    purgeSDBonGxFailure  YES
<!--NeedCopy-->
```

GUI を使用して確立された **Gx** 接続での転送障害を検出するには

1. [トラフィック管理] > [サブスクライバー] > [パラメータ] に移動します。
2. [サブスクライバーパラメータの構成] をクリックします。
3.「インタフェースタイプ」で「**GxOnly**」を選択します。
4. すべての必須パラメータの値を指定します。
5.「ヘルス・チェック」を選択し、「ヘルス・チェック **TTL** 」と「**CER** リクエスト・タイムアウト」の値を指定します。
6. [**OK**] をクリックします。

クラスタ・トポロジーの **Gx** インターフェース

NetScaler アプライアンスは、クラスタートポロジーの Gx インターフェイスをサポートします。



クラスター内の NetScaler ノードは、Gx インターフェイスを介して外部 PCRF サーバーと通信します。ノードがクライアントトラフィックを受信すると、アプライアンスは次の処理を実行します。

- CCR-I 要求を PCRF サーバに送信して、加入者情報を取得します。
- PCRF サーバは CCR-A で応答します。
- 次に、NetScaler ノードは受信したサブスクライバー情報をサブスクライバーストアに保存し、ルールをクライアントトラフィックに適用します。

各ノードは独立したサブスクライバストアを維持し、サブスクライバセッションは他のノードと同期されません。

Diameter Base Protocol RFC 6733 によると、直径プロトコルを介して他のピアと通信するには、各ピアに固有の直径識別情報を設定する必要があります。そのため、クラスタ展開では、直径が同一になるように設定されていることがわかります。各ノードの直径パラメータ（アイデンティティ、レルム、サーバクローズプロパゲーション）は、GUI または CLI を使用して個別に設定できます。

ノードをクラスターに追加すると、デフォルトの直径パラメーター (identity=netscaler.com、realm=com、

ServerClosePropogation=NO) が引き継がれます。ノードを追加したら、各ノードの直径パラメータを設定する必要があります。

GUI を使用して直径パラメータを設定するには

1. [システム] > [設定**]** に移動します。
2. 詳細ウィンドウで、[**Diameter** パラメータの変更] をクリックします。
3. **Diameter** パラメーターページで、直径パラメーターを構成する **NetScaler** ノードを選択し、「構成」をクリックします。
4. 「Diameter パラメータの設定」ページで、選択したノードの直径アイデンティティ、直径レルム、およびサーバークローズプロパゲーションを設定します。
5. [**OK**] をクリックします。

CLI を使用して直径パラメータを設定するには　コマンドプロンプトで入力します。

```
set ns diameter [-identity <string>] [-ownerNode <positive_integer>]
<!--NeedCopy-->
```

引数

ユーザー情報　Diameter 識別は、Diameter ノードを一意に識別するために使用されます。Diameter 構成を設定する前に、NetScaler アプライアンス（Diameter ノード）に固有の直径 ID を割り当てる必要があります。

たとえば、ns diameter-identity netscaler.com-ownerNode 1 に設定します。そのため、NetScaler システムで直径メッセージに ID を使用する必要がある場合は常に、RFC3588 で定義されているオリジンホスト AVP として「netscaler.com」を使用します。

最大長：255

オーナーノード　ownerNode は、直径 ID が設定されているクラスターノードの ID を表します。オーナーノードは CLIP を通してのみ構成できます。

最小値:0

最大値:31

例**:**

set ns diameter -identity netscaler1.com -ownerNode 1

注:

show ns diameter コマンドには ownerNode オプションも追加されています。

例**:**


```
show diameter -ownerNode <0-31>
<!--NeedCopy-->
```

show ns diameter コマンドを実行すると、特定のノードの直径パラメータが表示されます。

クラスタ展開用の **Gx** インターフェイスを設定するには　Gx インターフェイスを設定するには、次のタスクを実行します。

各 Gx インターフェイスに DIAMETER サービスを追加します。

例**:**

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868
add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

DIAMETER 負荷分散仮想サーバーを追加し、ステップ 1 で作成したサービスをこの仮想サーバーにバインドします。

例**:**

```
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam pcrf-svc1

bind lb vserver vdiam pcrf-svc2
<!--NeedCopy-->
```

すべてのクラスターノードで NetScaler の直径 ID とレルムを設定します。ID とレルムは、Gx クライアントから送信される直径メッセージのオリジンホストおよびオリジンレルム AVP として使用されます。

例**:**

```
set ns diameter -identity node0.netscaler.com -realm netscaler.com -
    ownerNode 0

set ns diameter -identity node1.netscaler.com -realm netscaler.com -
    ownerNode 1
<!--NeedCopy-->
```

手順 2 で作成した仮想サーバを PCRF 仮想サーバとして使用するように Gx インターフェイスを設定し、PCRF レルムも設定します。

例**:**

```
set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf.com

Set the subscriber interface type to GxOnly.
<!--NeedCopy-->
```


例**:**

```
set subscriber param -interfaceType GxOnly
<!--NeedCopy-->
```

Gx インターフェイスの設定とステータスを確認するには、次のように入力します。

```
show subscriber gxinterface
<!--NeedCopy-->
```

RADIUS インターフェイス

RADIUS インターフェイスを使用すると、IP-CAN セッションが確立されると、パケットゲートウェイは RADIUS アカウンティング開始メッセージのサブスクライバ情報を RADIUS インターフェイスを介してアプライアンスに転送します。RADIUS リスナータイプのサービスは、RADIUS アカウンティングメッセージを処理します。RADIUS クライアントの共有シークレットを追加します。共有シークレットが設定されていない場合、RADIUS メッセージは通知なしでドロップされます。次の例は、RADIUS インターフェイスを設定するためのコマンドを示しています。コマンドは太字で表示されます。

RADIUS インターフェイスを設定するには、次のタスクを実行します。

RADIUS メッセージを受信する SNIP アドレスに RADIUS リスナーサービスを作成します。次に例を示します:

```
add service srad1 192.0.0.206 RADIUSLISTENER 1813
<!--NeedCopy-->
```

このサービスを使用するように加入者の RADIUS インターフェイスを設定します。次に例を示します:

```
set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

サブスクライバインターフェイスタイプを RADIUSOnly に設定します。次に例を示します:

```
set subscriber param -interfaceType RadiusOnly
<!--NeedCopy-->
```

サブネットと共有シークレットを指定する RADIUS クライアントを追加します。次に例を示します:

```
add radius client 192.0.2.0/24 -radkey client123
<!--NeedCopy-->
```

0.0.0.0/0 のサブネットは、それがすべてのクライアントのデフォルトの共有シークレットであることを意味します。RADIUS インターフェイスの設定とステータスを確認するには、次のように入力します。

```
show subscriber radiusInterface
<!--NeedCopy-->
```

RADIUS インターフェイスパラメータ:

Radius Listener Service: srad1(UP)

Done

例**:**

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868

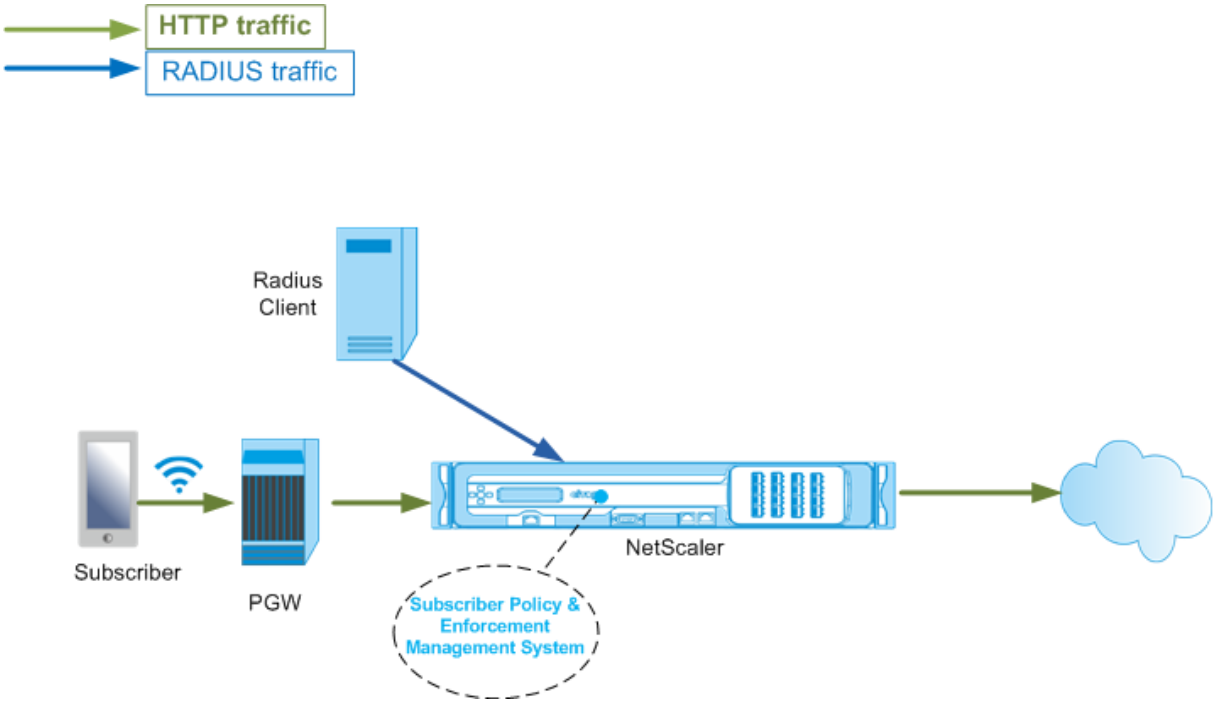
add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

引数

ListeningService　RADIUS アカウンティング要求を処理する RADIUS リスニングサービスの名前。

svrState　RADIUS リスニングサービスの状態。

次の図は、高レベルのトラフィックフローを示しています。



GUI を使用して **RADIUS** 専用インターフェイスを設定するには

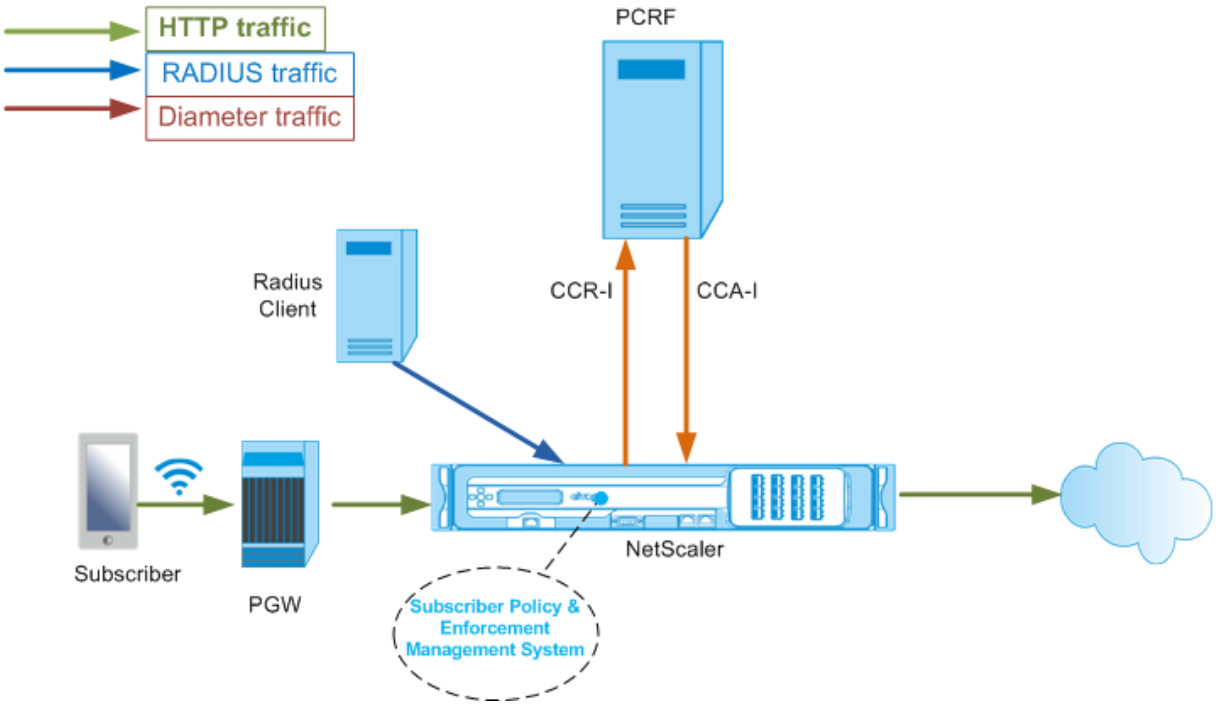
1. [トラフィック管理] > [サブスクライバー] > [パラメータ] に移動します。
2. [サブスクライバーパラメータの構成] をクリックします。
3. 「インタフェース・タイプ」で「**RADIUSOnly**」を選択します。
4. すべての必須パラメータの値を指定します。
5. [**OK**] をクリックします。

RADIUS および **Gx** インターフェイス

RADIUS および Gx インターフェイスでは、IP-CAN セッションが確立されると、パケットゲートウェイは MSISDN などのサブスクライバー ID と、加入者に関するフレーム IP アドレス情報を RADIUS インターフェイスを介してアプライアンスに転送します。アプライアンスはこのサブスクライバ ID を使用して Gx インターフェイスの PCRF にクエリを実行し、サブスクライバ情報を取得します。これはプライマリ PCEF 機能として知られています。次の例は、RADIUS および Gx インターフェイスを設定するためのコマンドを示しています。

```
set subscriber param -interfaceType RadiusandGx
add service pcrf-svc 203.0.113.1 DIAMETER 3868
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263
bind lb vserver vdiam pcrf-svc
set subscriber gxInterface -vServer vdiam -pcrfRealm testrealm1.net -
    holdOnSubscriberAbsence YES -revalidationTimeout 60 -negativeTTL 120
add service srad1 192.0.0.206 RADIUSLISTENER 1813    set subscriber
    radiusInterface -listeningService srad1
<!--NeedCopy-->
```

次の図は、高レベルのトラフィックフローを示しています。



GUI を使用して **RADIUSandGX** インターフェイスを設定するには

1. [トラフィック管理] > [サブスクライバー] > [パラメータ] に移動します。
2. [サブスクライバーパラメータの構成] をクリックします。
3.「インタフェース・タイプ」で、「**RADIUS**」と「**GX**」を選択します。
4. すべての必須パラメータの値を指定します。

5. [**OK**] をクリックします。

スタティックサブスクライバーの設定

コマンドラインまたは構成ユーティリティを使用して、NetScaler アプライアンス上でサブスクライバーを手動で構成できます。スタティックサブスクライバを作成するには、一意のサブスクライバ ID を割り当て、オプションで各サブスクライバにポリシーを関連付けます。次の例は、スタティックサブスクライバを設定するコマンドを示しています。

次の例では、**SubscriptionIdValue** は国際電話番号を指定し、**SubscriptionIDType** (この例では E164) は国際電話番号の一般的な形式を指定します。

```
1     add subscriber profile 203.0.113.6 -subscriberRules policy1 policy2
          -subscriptionIdType E164 -subscriptionIdvalue 98767543211
2     add subscriber profile 2002::a66:e8d3/64 -subscriberRules policy1
         policy3 -subscriptionIdtype E164 -subscriptionIdvalue
         98767543212
3     add subscriber profile 203.0.24.2 10 -subscriberRules policy2
         policy3 -subscriptionIdtype E164 -subscriptionIdvalue
         98767543213
4 <!--NeedCopy-->
```

設定済みの加入者プロファイルを表示するには、次のように入力します。

```
show subscriber profile
```

```
1     > show subscriber profile
2
3     1)  Subscriber IP: 203.0.24.2 VLAN:10
4         Profile Attributes:
5                 Active Rules: policy2, policy3
6                 Subscriber Id Type: E164
7                 Subscriber Id Value: 98767543213
8     2)  Subscriber IP: 2002::/64
9         Profile Attributes:
10                Active Rules: policy1, policy3
11                Subscriber Id Type: E164
12                Subscriber Id Value: 98767543212
13    3)  Subscriber IP: 203.0.113.6
14        Profile Attributes:
15                Active Rules: policy1, policy2
16                Subscriber Id Type: E164
17                Subscriber Id Value: 98767543211
18
19     Done
20 <!--NeedCopy-->
```


デフォルト加入者プロファイル

アプライアンスのサブスクライバセッションストアでサブスクライバ IP アドレスが見つからない場合は、デフォルトのサブスクライバプロファイルが使用されます。次の例では、デフォルトの加入者プロファイルに加入者ルール policy1 が追加されています。

```
    > add subscriber profile * -subscriberRules policy1
<!--NeedCopy-->
```

サブスクライバーセッションの表示とクリア

次のコマンドを使用して、すべてのスタティックサブスクライバセッションとダイナミックサブスクライバセッションを表示します。

> show subscriber sessions

```
    > show subscriber sessions
    1)  Subscriber IP: 2002::/64
        Session Attributes:
                Active Rules: policy1, policy3
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543212
    2)  Subscriber IP: *
        Session Attributes:
                Active Rules: policy1
    3)  Subscriber IP: 203.0.24.2 VLAN:10
        Session Attributes:
                Active Rules: policy2, policy3
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543213
    4)  Subscriber IP: 203.0.113.6
        Session Attributes:
                Active Rules: policy1, policy2
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543211
    5)  Subscriber IP: 192.168.0.11
        Session Attributes:
                Idle TTL remaining: 361 Seconds
                Active Rules: policy1
                Subscriber Id Type: E164
                Subscriber Id Value: 1234567811
                Service Path: policy1
                AVP(44): 34 44 32 42 42 38 41 43  2D 30 30 30 30 30 30
                    31 31
                AVP(257): 00 01 C0 A8 0A 02
                PCRF-Host: host.pcrf.com
                AVP(280): 74 65 73 74 2E 63 6F 6D

    Done
<!--NeedCopy-->
```


次のコマンドを使用して、1 つのセッションまたはセッションストア全体をクリアします。IP アドレスを指定しない場合、サブスクライバセッションストア全体がクリアされます。

```
clear subscriber sessions <ip>
<!--NeedCopy-->
```

加入者ポリシーの適用および管理システム

NetScaler アプライアンスは、サブスクライバーの IP アドレスを、サブスクライバーポリシーの適用および管理システムのキーとして使用します。

サブスクライバーエクスプレッションを追加して、サブスクライバーポリシー適用および管理システムにあるサブスクライバー情報を読み取ることができます。これらの表現は、統合キャッシュ、リライト、レスポンダー、コンテンツスイッチングなどの NetScaler 機能用に構成されたポリシールールやアクションで使用できます。

次のコマンドは、サブスクライバベースのレスポンダアクションとポリシーを追加する例です。サブスクライバール値が「pol1」の場合、ポリシーは true と評価されます。

```
    add responder action error_msg respondwith '"HTTP/1.1 403 OKrnrn" +
        " You are  not authorized to access Internet"'
    add responder policy no_internet_access "SUBSCRIBER.RULE_ACTIVE("
       pol1")" error_msg
<!--NeedCopy-->
```

次の例は、サブスクライバベースの書き換えアクションとポリシーを追加するコマンドを示しています。このアクションは、サブスクライバーセッションに AVP (45) の値を使用して HTTP ヘッダー「X-Nokia-MSISDn」を挿入します。

```
    > add rewrite action AddHDR-act insert_http_header X-Nokia-MSISDN "
       SUBSCRIBER.AVP(45).VALUE"
    > add rewrite policy AddHDR-pol "HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.
       URL).EQUALS_ANY("patset-test")" AddHDR-act
<!--NeedCopy-->
```

次の例では、アプライアンスに 2 つのポリシーが設定されています。アプライアンスがサブスクライバ情報をチェックし、サブスクライバールが cache_enable の場合、キャッシュを実行します。サブスクライバールが cache_disable の場合、アプライアンスはキャッシュを実行しません。

```
    > add cache policy nocachepol -rule "SUBSCRIBER.RULE_ACTIVE("
       cache_disable")" - action NOCACHE
    > add cache policy cachepol -rule "SUBSCRIBER.RULE_ACTIVE("
       cache_enable")" - action CACHE -storeInGroup cg1
<!--NeedCopy-->
```

「SUBSCRIBER」で始まる式の一覧については、「ポリシー設定ガイド」を参照してください。

> 重要:
>
> NetScaler ADC ソフトウェアリリース 12.1 では、加入者インターフェイスが GxOnly に設定されている場合、IPANDVLAN キー検索方式がサポートされています。詳細については、IP アドレスと VLAN ID キールックアップ方式を参照してください。

IPv6 プレフィクスベースのサブスクライバセッション

電話会社のユーザは、完全な IPv6 アドレスではなく IPv6 プレフィックスで識別されます。NetScaler アプライアンスは、データベース（サブスクライバーストア）内のサブスクライバーを識別するために、完全な IPv6 アドレス（/128）の代わりにプレフィックスを使用するようになりました。PCRF サーバとの通信（CCR-I メッセージなど）に、アプライアンスは完全な IPv6 アドレスの代わりにフレーム付き IPv6 プレフィックス AVP を使用するようになりました。デフォルトのプレフィックス長は /64 ですが、別の値を使用するようにアプライアンスを構成できます。

コマンドラインを使用して **IPv6** プレフィックスを構成するには

```
set subscriber param [-ipv6PrefixLookupList <positive_integer> ...]
```

以下の最初のコマンド例は 1 つのプレフィックスを設定し、2 番目のサンプルコマンドは複数のプレフィックスを設定します。

```
1     set subscriber param -ipv6PrefixLookupList 64
2     set subscriber param -ipv6PrefixLookupList 64 72 96
3 <!--NeedCopy-->
```

構成ユーティリティを使用して **IPv6** プレフィックスを構成するには

1. [トラフィック管理] > [サブスクライバー] > [パラメータ] に移動します。
2. 詳細ペインの [設定] で [サブスクライバパラメータの設定] をクリックし、[**IPv6** プレフィックス検索リスト] で 1 つまたは複数のプレフィックスを指定します。

IP アドレスと VLAN ID キーの検索方法

NetScaler アプライアンスは、サブスクライバーの IP アドレスを、サブスクライバーポリシーの適用および管理システムへの重要な検索方法として使用します。この方法は、IP アドレスが重複している場合は効果がありません。このような場合は、VLAN ID を追加のサブスクライバ検索タイプとして使用できます。IPANDVLAN キールックアップ方式は、サブスクライバインターフェイスが GxOnly に設定されている場合にのみサポートされます。IPANDVLAN がルックアップ方法として構成されている場合、NetScaler アプライアンスは以下を実行します。

- IPv4 サブスクライバの Gx クエリに発信元の VLAN ID を含めます。

- すべての Gx 応答に Gx VLAN AVP を含めます。ただし、VLAN ID が一致しない場合、アプライアンスは応答を無視します。

たとえば、アプライアンスが GXSessionid-A: IPv4-B: VLAN-C で CCR-I を送信し、応答に GxSessionid-A: IPV4-B: VLAN-D が含まれている場合、応答はドロップされ、デフォルトのサブスクライバエントリが作成されます。

> 注
>
> - インターフェイスタイプ RADIUSandGX と RADIUSOnly をキータイプ IPAND VLAN と一緒に設定することはできません。
> - トラフィックが IPv6 アドレスからのものである場合、NetScaler アプライアンスは IP ルックアップ方式を使用します。

CLI を使用して **IP** または **IPANDVLAN** をキールックアップ方式として設定するには

コマンドプロンプトで入力します。

```
set subscriber param [-keytype ( IP | IPANDVLAN )] [-interfaceType <
   interfaceType>]
<!--NeedCopy-->
```

例**:**

```
set subscriber param -keytype IPANDVLAN -interfaceType GxOnly

set subscriber param -keytype IP -interfaceType GxOnly
<!--NeedCopy-->
```

> 注
>
> キータイプパラメータを IP から IPANDVLAN に変更すると、逆にすべてのサブスクライバデータが消去されます。

VLAN パラメータ

VLAN パラメータは、次のコマンドにも追加されます。

```
add subscriber profile <ip>@ [-vlan]

set subscriber profile <ip>@ [-vlan] [-subscriptionIdType  <
   subscriptionIdType>]

show subscriber profile [<ip>@] [-vlan]

rm subscriber profile <ip>@ [-vlan <positive_integer>]
<!--NeedCopy-->
```


引数

IP　加入者の IP アドレスを表します。これは必須の引数であり、加入者プロファイルを追加した後は変更できません。

vlan　サブスクライバが配置されている VLAN 番号を表します。加入者プロファイルを追加した後に VLAN 番号を変更することはできません。

最小値:1

最大値:4096

```
add subscriber profile 192.0.2.23 10

set subscriber profile 192.0.2.23 10 -subscriptionIdtype E164

show subscriber profile 192.0.2.23 10

rm subscriber profile 192.0.2.23 10

<!--NeedCopy-->
```

GUI を使用してキールックアップ方式として **IP** または **IPANDVLAN** を設定するには

1. [トラフィック管理] > [サブスクライバ] > [パラメータ] に移動します。
2.「サブスクライバパラメータの設定」をクリックします。
3.「キータイプ」で、要件に応じて「**IP** 」または「**IPANDVLAN** 」を選択します。
4. 設定を完了し、「**OK**」をクリックします。

Telco ネットワーク内の加入者セッションのアイドルセッション管理

NetScaler アプライアンスでのサブスクライバーセッションクリーンアップは、RADIUS アカウンティング停止メッセージ、Diameter RAR（セッションリリース）メッセージ、または「サブスクライバーセッションのクリア」コマンドなどのコントロールプレーンイベントに基づいています。導入環境によっては、RADIUS クライアントまたは PCRF サーバからのメッセージがアプライアンスに届かない場合があります。また、トラフィックが多い場合、メッセージが失われる可能性があります。長時間アイドル状態のサブスクライバーセッションは、NetScaler アプライアンスのメモリと IP リソースを消費し続けます。アイドルセッション管理機能では、設定可能なタイマーを使用してアイドルセッションを識別し、指定されたアクションに基づいてこれらのセッションをクリーンアップします。

このサブスクライバからのトラフィックがデータプレーンまたはコントロールプレーンで受信されない場合、セッションはアイドル状態と見なされます。更新、終了 (PCRF に通知してからセッションを削除する)、削除 (PCRF に通知しない) アクションを指定できます。アクションは、idle timeout パラメータで指定された時間だけセッションがアイドル状態になった後にのみ実行されます。

コマンドラインを使用してアイドルセッションのタイムアウトと関連するアクションを設定するには

```
set subscriber param [-idleTTL <positive_integer>] [-idleAction <
    idleAction>]
<!--NeedCopy-->
```

例:

```
set subscriber param -idleTTL 3600 -idleAction ccrTerminate

set subscriber param -idleTTL 3600 -idleAction ccrUpdate

set subscriber param -idleTTL 3600 -idleAction delete
<!--NeedCopy-->
```

アイドルセッションのタイムアウトを無効にするには、アイドルタイムアウトをゼロに設定します。

set subscriber param –idleTTL 0

設定ユーティリティを使用してアイドルセッションのタイムアウトと関連するアクションを設定するには

1. [トラフィック管理] > [サブスクライバー] > [パラメータ] に移動します。
2. 詳細ペインの **[**** 設定] で、[サブスクライバーパラメータの設定] をクリックし、[アイドル時間] と **[**アイドルアクション] を指定します。**

サブスクライバーセッションイベントロギング

サブスクライバロギングを有効にすると、サブスクライバ固有の RADIUS および Gx コントロールプレーンメッセージを追跡し、履歴データを使用してサブスクライバのアクティビティを分析できます。重要な属性には、MSISDN とタイムスタンプがあります。次の属性もログに記録されます。

- セッションイベント (インストール、更新、削除、エラー)
- Gx メッセージタイプ (CCR-I、CCR-U、CCR-T、RAR)
- Radius メッセージタイプ (開始、停止)
- サブスクライバー IP
- サブスクライバ ID タイプ (MSISDN (E164)、IMSI)
- サブスクライバー ID の値

これらのログを使用すると、IP アドレスと (可能な場合は MSISDN) でユーザーを追跡できます。

ローカルまたはリモートの syslog または nslog サーバへのサブスクライバセッションロギングを有効にできます。次の例は、リモート syslog サーバへのサブスクライバロギングを有効にする方法を示しています。

```
      > add syslogAction sysact1 192.0.2.0 -loglevel EMERGENCY ALERT
        CRITICAL ERROR WARNING NOTICE INFORMATIONAL -subscriberlog
        enabled
```



```
<!--NeedCopy-->
```

これらのログから、セッションが更新、削除、作成（インストール）された時間など、ユーザーに関連するあらゆるアクティビティについて知ることができます。さらに、エラーメッセージも記録されます。

例:

1. 次のログエントリは、RadiusAndGX セッションの作成、セッション更新、およびセッション削除の例です。

   > 09/30/2015:16:29:18 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 147 0 : Session Install, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001
   >
   > 09/30/2015:16:30:18 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 148 0 : Session Update, GX MsgType: CCR-U, IP: 100.10.1.1, ID: E164 - 30000000001
   >
   > 09/30/2015:17:27:56 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 185 0 : Session Delete, GX MsgType: CCR-T, RADIUS MsgType: Stop, IP: 100.10.1.1, ID: E164 - 30000000001

2. 次のログエントリは、PCRF サーバでサブスクライバが見つからない場合や、アプライアンスが PCRF サーバに接続できない場合などのエラーメッセージの例です。

   > 09/30/2015:16:44:15 GMT Error 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 169 0 : Failure Reason: PCRF failure response, GX MsgType: CCR-I, IP: 100.10.1.1
   >
   > Sep 30 13:03:01 09/30/2015:16:49:08 GMT 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 176 0 : Failure Reason: Unable to connect to PCRF, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001#000#000#000#000#000#000#000#000#000#000#000#000#00

サブスクライバ対応 **LSN** セッション終了

以前のリリースでは、RADIUS Accounting STOP または PCRF-RAR メッセージの受信時、または TTL の有効期限切れやフラッシュなどの他のイベントの結果としてサブスクライバセッションが削除された場合、設定された LSN タイムアウト期間が経過した後にのみ、サブスクライバの対応する LSN セッションが削除されます。このタイムアウトが切れるまで開いたままになっている LSN セッションは、引き続きアプライアンスのリソースを消費します。

リリース 11.1 から、新しいパラメータ (サブスクセッション削除) が追加されました。このパラメータを有効にして、サブスクライバ情報をサブスクライバデータベースから削除すると、そのサブスクライバに対応する LSN セッションも削除されます。このパラメータを無効にすると、LSN タイムアウト設定で指定されたとおりにサブスクライバセッションがタイムアウトします。

CLI を使用してサブスクライバ対応 **LSN** セッション終了を設定するには

コマンドプロンプトで入力します。

set lsn parameter -subscrSessionRemoval (DISABLED)
ENABLED

```
    > set lsn parameter -subscrSessionRemoval ENABLED
    Done
    > sh lsn parameter
                LSN Global Configuration:

                Active Memory Usage: 0 MBytes
                Configured Memory Limit: 0 MBytes
                Maximum Memory Usage Limit: 912 MBytes
                Session synchronization: ENABLED
                Subscriber aware session removal: ENABLED
<!--NeedCopy-->
```

GUI を使用してサブスクライバ対応 **LSN** セッション終了を設定するには

1. [システム] > [大規模 **NAT**] に移動します。
2.「はじめに」で、「**LSN** パラメータの設定」をクリックします。
3. サブスクライバ対応セッション削除パラメータを設定します。

トラブルシューティング

デプロイが期待どおりに機能しない場合は、次のコマンドを使用してトラブルシューティングを行います。

- show subscriber gxinterface
  このコマンドの出力には、次のエラーメッセージが含まれる場合があります (ここでは、推奨される応答と共に示しています)。
  - Gx インターフェイスが設定されていません-set subscriber param コマンドを使用して正しいインターフェイスタイプを設定してください。
  - PCRF が設定されていません-GXInterface で Diameter 仮想サーバーまたはサービスを設定してください。set subscriber gx interface コマンドを使用して、このインターフェイスに Diameter 仮想サーバーまたはサービスを割り当ててください。
  - PCRF が準備完了ではありません。詳細については、対応する vserver/service を確認してください。show LB vserver または show service コマンドを使用して、サービスの状態を確認してください。
  - NetScaler は PCRF からの CEA を待っています。PCRF と NetScaler 間の機能ネゴシエーションが失敗している可能性があります。これは断続的な状態である可能性があります。それでも解決しない場合は、PCRF サーバーの DIAMETER 設定を確認してください。
  - メモリはサブスクライバセッションを保存するように設定されていません。’set extendedmemoryparam-memlimit <>’ を使用してください。-set extendedmemory-param コマンドを使用して拡張メモリを設定してください。

- show subscriber radiusinterface
  このコマンドの出力が「未設定」の場合は、set subscriber radiusinterface コマンドを使用して RADIUS-Listener サービスを指定してください。

サブスクライバーロギングが有効になっている場合は、ログファイルからより詳細な情報を取得できます。

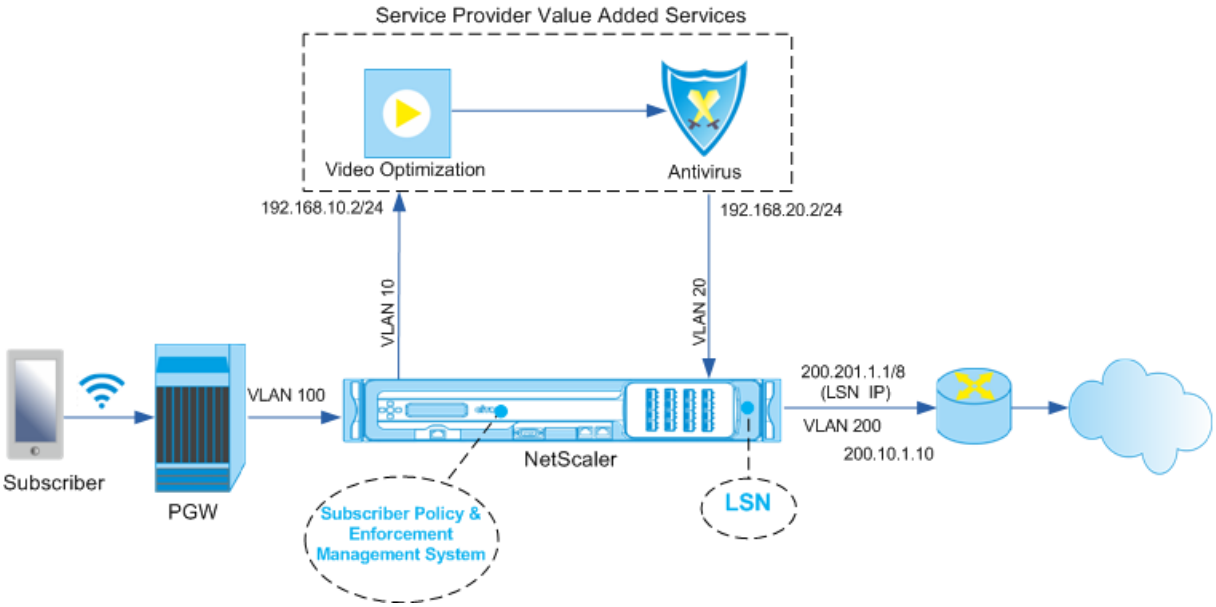
利用者に応じたトラフィックステアリング

August 15, 2023

トラフィックステアリングは、加入者のトラフィックをある地点から別の地点に誘導します。サブスクライバーがネットワークに接続すると、パケットゲートウェイは IP アドレスをサブスクライバーに関連付け、データパケットを NetScaler アプライアンスに転送します。アプライアンスは Gx インターフェイスを介して PCRF サーバと通信し、ポリシー情報を取得します。ポリシー情報に応じて、アプライアンスは次のいずれかのアクションを実行します。

- データパケットを別のサービスセットに転送します (次の図を参照)。
- パケットをドロップしてください。
- アプライアンスに LSN が設定されている場合は、大規模 NAT（LSN）のみを実行してください。

次の図に示されている値は、図に続く CLI プロシージャで設定されます。NetScaler アプライアンス上のコンテンツスイッチ仮想サーバーは、定義されたルールに応じて、付加価値サービスに要求を転送するか、スキップし、LSN を実行した後にパケットをインターネットに送信します。



CLI を使用して上記のデプロイのトラフィックステアリングを設定するには

アプライアンスのサブネット IP (SNIP) アドレスを追加します。

例:

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 100.100.100.1 255.0.0.0 -type snip

add ns ip 200.200.200.1 255.0.0.0 -type snip

add ns ip 100.1.1.1 255.0.0.0 -type snip

add ns ip 200.201.1.1 255.0.0.0 -type snip
<!--NeedCopy-->
```

VLAN を追加します。VLAN は、アプライアンスがトラフィックの送信元を識別するのに役立ちます。VLAN をインターフェイスとサブネット IP アドレスにバインドします。

例:

```
add vlan 10

add vlan 20

add vlan 100

add vlan 200

bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1 255.255.255.0

bind vlan 100 -ifnum 1/2 -tagged -IPAddress 100.1.1.1 255.0.0.0

bind vlan 200 -ifnum 1/3 -tagged -IPAddress 200.1.1.1 255.0.0.0
<!--NeedCopy-->
```

加入者トラフィックがアプライアンスに到着する VLAN を指定します。サブスクライバセッション内のサービスパス名を検索する場所をアプライアンスに指示するサービスパス AVP を指定します。プライマリ PCEF 機能については、インターフェイスタイプを RADIUS と GX と指定します。

例:

```
set ns param -servicePathIngressVLAN 100

set subscriber gxinterface -servicepathAVP 1001 1005 -
   servicepathVendorid 10415

set subscriber param -interfaceType RadiusAndGx
<!--NeedCopy-->
```

Diameter タイプのサービスと仮想サーバーを設定し、サービスを仮想サーバーにバインドします。次に、PCRF レ

ルムとサブスクライバ Gx インターフェイスパラメータを指定します。PCEF の主な機能として、RADIUS リスナーサービスと RADIUS インターフェイスを設定します。

例:

```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -
    holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 RADIUSListener 1813

set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

VAS を入力 VLAN に関連付けるサービス機能を追加します。サービスパスを追加してチェーンを定義します。つまり、パケットの送信先の VAS と、その VAS に送信する順序を指定します。サービスパス名は通常 PCRF によって送信されます。ただし、次のいずれかに当てはまる場合は、デフォルトのサブスクライバプロファイル（*）のサービスパスが適用されます。

- PCRF には加入者情報はありません。
- 加入者情報にはこの AVP は含まれていません。
- アプライアンスは PCRF にクエリを実行できません。たとえば、PCRF を表すサービスは DOWN です。

この名前を含むサービスパス AVP は、グローバル設定の一部としてすでに設定されている必要があります。サービス関数をサービスパスにバインドします。サービスインデックスは、VAS がチェーンに追加される順序を指定します。最も大きい数字 (255) はチェーンの始まりを示します。

例:

```
add ns servicefunction SF1 -ingressVLAN 20

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

add subscriber profile * -subscriberrules default_path
<!--NeedCopy-->
```

LSN 設定を追加します。つまり、NAT プールを定義し、アプライアンスが LSN を実行する必要があるクライアントを特定します。

```
add lsn pool pool1
```



```
2
3 bind lsn pool pool1 200.201.1.1
4
5 add lsn client client1
6
7 bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0
8
9 add lsn group group1 -clientname client1
10
11 bind lsn group group1 -poolname pool1
12 <!--NeedCopy-->
```

アプライアンスはデフォルトで LSN を実行します。LSN をオーバーライドするには、overridelSN パラメータを有効にしたネットプロファイルを作成し、このプロファイルを付加価値サービス（VAS）用に設定されたすべての負荷分散仮想サーバーにバインドする必要があります。

例**:**

```
1 add netprofile np1
2
3 set netprofile np1 -overrideLsn ENABLED
4
5 set lb vserver vs1 -netprofile np1
6 <!--NeedCopy-->
```

アプライアンスで VAS を設定します。これには、サービスと仮想サーバーを作成し、サービスを仮想サーバーにバインドすることが含まれます。

```
1 add service vas1 192.168.10.2 ANY 80 -usip YES
2
3 add service sint 200.10.1.10 ANY 80 -usip YES
4
5 add lb vserver vs1 ANY -m MAC -l2Conn ON
6
7 add lb vserver vint ANY -m MAC -l2Conn ON
8
9 bind lb vserver vs1 vas1
10
11 bind lb vserver vint sint
12 <!--NeedCopy-->
```

コンテンツスイッチング (CS) 構成を追加します。これには、仮想サーバー、ポリシー、およびそれらに関連するアクションが含まれます。トラフィックは CS 仮想サーバに到着し、適切な負荷分散仮想サーバにリダイレクトされます。仮想サーバーとサービス機能を関連付ける式を定義します。

例**:**

```
1 add cs vserver cs1 ANY * 80 -l2Conn ON
2
3 add cs action csact1 -targetLBVserver vs1
4
5 add cs action csactint -targetLBVserver vint
```



```
6
7  add cs policy cspol1 -rule  SUBSCRIBER.SERVICEPATH.IS_NEXT("SF1") &&
       SYS.VSERVER("vs1").STATE.EQ(UP)" -action csact1
8
9  bind cs vserver cs1 -policyName cspol1 -priority 110
10
11 bind cs vserver cs1 -lbvserver vint
12 <!--NeedCopy-->
```

GUI を使用してアプライアンスのトラフィックステアリングを設定するには

1. [システム] > [ネットワーク] > [**IP**] に移動し、サブネット IP アドレスを追加します。
2. [システム] > [ネットワーク] > [**VLAN**] に移動し、VLAN を追加し、VLAN をインターフェイスとサブネット IP アドレスにバインドします。
3. [トラフィック管理] > [サービスチェーン] > [サービスパスの入力 **VLAN** の設定**]** に移動し、入力 **VLAN** を指定します。
4. [トラフィック管理] > [サブスクライバ] > [パラメータ] > [サブスクライバパラメータの設定] に移動し、以下を指定します。

   - インターフェイスの種類: **RADIUS** と **GX** を指定します。
   - Diameter 仮想サーバ、PCRF レルム、およびサブスクライバ GX インターフェイスパラメータを設定します。
   - RADIUS インターフェイスパラメータを指定します。

5. [トラフィック管理] > [サービスチェーン] > [サービス機能] に移動し、サービス機能を追加して付加価値サービスを入力 VLAN に関連付けます。
6. [システム] > [ネットワーク] > [大規模 **NAT**] に移動します。[プール] をクリックし、プールを追加します。[クライアント] をクリックし、クライアントを追加します。「グループ」をクリックしてグループを追加し、クライアントを指定します。グループを編集し、プールをこのグループにバインドします。
7. [システム] > [ネットワーク] > [ネットプロファイル] に移動し、ネットプロファイルを追加します。[**LSN** のオーバーライド] を選択します。必要に応じて、[システム] > [ネットワーク] > [設定**] > [**レイヤ **3** パラメータの設定] に移動し、[**LSN** の上書き] が選択されていないことを確認します。
8. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、アプライアンス上の仮想サーバーと付加価値サービスを設定します。サービスとネットプロファイルを仮想サーバーにバインドします。
9. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバー、ポリシー、およびアクションを設定します。ターゲットの負荷分散仮想サーバーを指定します。

GUI を使用してアプライアンスのサービスチェーニングを設定するには

1. [システム] > [ネットワーク] > [**IP**] に移動し、サブネット IP アドレスを追加します。
2. [システム] > [ネットワーク] > [**VLAN**] に移動し、VLAN を追加し、VLAN をインターフェイスとサブネット IP アドレスにバインドします。

3. [トラフィック管理] > [サービスチェーン] > [サービスパスの入力 **VLAN** の設定**]** に移動し、入力 **VLAN** を指定します。
4. [トラフィック管理] > [サブスクライバ] > [パラメータ] > [サブスクライバパラメータの設定] に移動し、以下を指定します。

    - インターフェイスの種類: **RADIUS** と **GX** を指定します。
    - Diameter 仮想サーバ、PCRF レルム、およびサブスクライバ GX インターフェイスパラメータを設定します。
    - RADIUS インターフェイスパラメータを指定します。

5. [トラフィック管理] > [サービスチェーン] > [サービス機能] に移動し、サービス機能を追加して付加価値サービスを入力 VLAN に関連付けます。
6. [システム] > [ネットワーク] > [大規模 **NAT**] に移動します。[プール] をクリックし、プールを追加します。[クライアント] をクリックし、クライアントを追加します。「グループ」をクリックしてグループを追加し、クライアントを指定します。グループを編集し、プールをこのグループにバインドします。
7. [システム] > [ネットワーク] > [ネットプロファイル] に移動し、ネットプロファイルを追加します。[**LSN** のオーバーライド] を選択します。必要に応じて、[システム] > [ネットワーク] > [設定**] > [**レイヤ **3** パラメータの設定] に移動し、[**LSN** の上書き] が選択されていないことを確認します。
8. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、アプライアンス上の仮想サーバーと付加価値サービスを設定します。サービスとネットプロファイルを仮想サーバーにバインドします。
9. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバー、ポリシー、およびアクションを設定します。ターゲットの負荷分散仮想サーバーを指定します。

利用者に応じたサービスチェーン

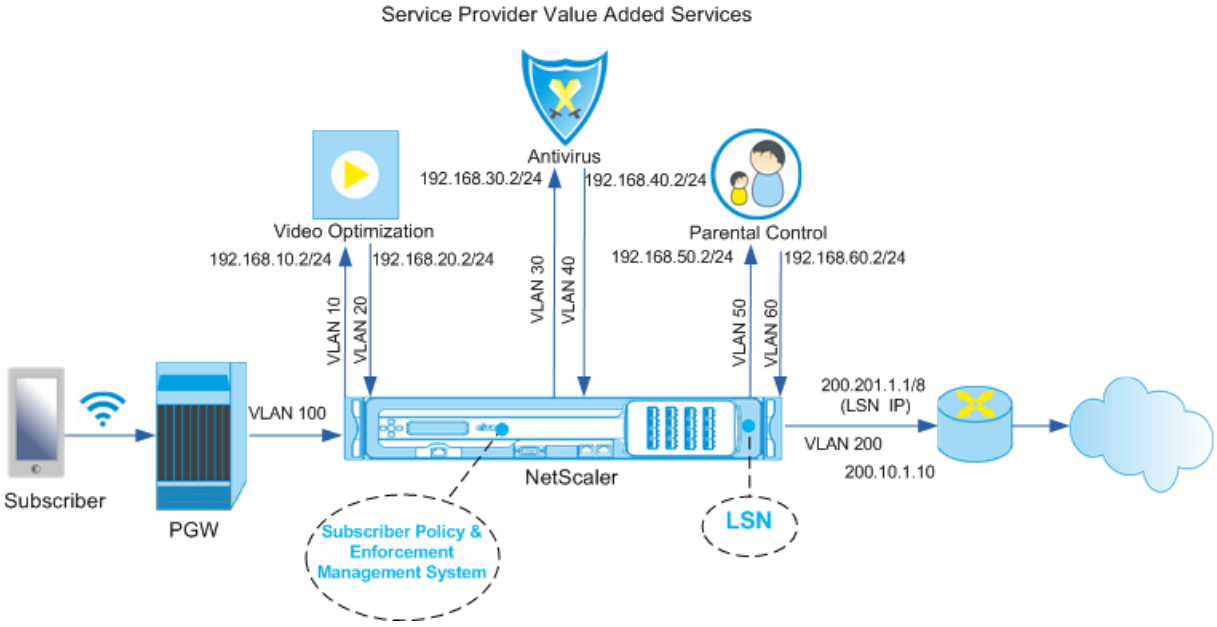
August 15, 2023

通信ネットワークを通過するデータトラフィックが大幅に増加しているため、サービスプロバイダーがすべてのトラフィックをすべての付加価値サービス（VAS）に誘導することはもはや現実的ではありません。サービスプロバイダーは、VAS の使用を最適化し、トラフィックをインテリジェントに誘導してユーザーエクスペリエンスを向上させる必要があります。たとえば、動画を含まないトラフィックには動画最適化は必要ありません。さらに、加入者が 4G ネットワークに接続している場合、コンテンツは高解像度（HD）でストリーミングできるため、ビデオの最適化は必要ない場合があります。ただし、ビデオを最適化すると、3G ネットワークのユーザーのエクスペリエンスが向上します。同様に、キャッシュはより高速で優れたユーザーエクスペリエンスを提供し、サブスクライバープランによっては有効にできます。VAS のもう一つの例は保護者による制限です。親が未成年の子供に携帯電話を提供する場合、子供がアクセスするウェブサイトを何らかの形で管理したいと思うでしょう。

上記とそれ以上のことを実現するには、サービスプロバイダーは加入者ごとに付加価値サービスを提供できなければなりません。つまり、サービスプロバイダーネットワーク内のエンティティは、加入者情報を抽出し、この情報に基づいてパケットをインテリジェントにステアリングできなければなりません。

サービスチェイニングは、加入者からのトラフィックがインターネットに到達する前に通過しなければならないサービスのセットを決定します。NetScaler は、すべてのトラフィックをすべてのサービスに送信する代わりに、サブスクライバーに対して定義されたポリシーに基づいて、サブスクライバーからのすべてのリクエストを特定のサービスセットにインテリジェントにルーティングします。

次の図は、サービスチェーンに関係するエンティティを示しています。表示されている値は、図の後の手順で設定されています。NetScaler アプライアンス上のコンテンツスイッチ仮想サーバーは、定義されたルールに応じて、付加価値サービスに要求を転送するか、スキップし、LSN を実行した後にパケットをインターネットに送信します。



CLI を使用して上記のデプロイメントのサービスチェーニングを設定するには

アプライアンスのサブネット IP (SNIP) アドレスを追加します。

例**:**

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 192.168.30.1 255.255.255.0 -type snip

add ns ip 192.168.40.1 255.255.255.0 -type snip

add ns ip 192.168.50.1 255.255.255.0 -type snip

add ns ip 192.168.60.1 255.255.255.0 -type snip

add ns ip 100.1.1.1 255.0.0.0 -type snip

add ns ip 200.201.1.1 255.0.0.0 -type snip
```



```
<!--NeedCopy-->
```

VLAN を追加します。VLAN は、アプライアンスがトラフィックの送信元を識別するのに役立ちます。VLAN をインターフェイスとサブネット IP アドレスにバインドします。VAS ごとに入力 VLAN と出力 VLAN を追加します。

例:

```
add vlan 10

add vlan 20

add vlan 30

add vlan 40

add vlan 50

add vlan 60

add vlan 100

add vlan 200

bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.30.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.40.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.50.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.60.1 255.255.255.0

bind vlan 100 -ifnum 1/2 -tagged -IPAddress 100.1.1.1 255.0.0.0

bind vlan 200 -ifnum 1/3 -tagged -IPAddress 200.201.1.1 255.0.0.0
<!--NeedCopy-->
```

加入者トラフィックがアプライアンスに到着する VLAN を指定します。サブスクライバセッション内のサービスパス名を検索する場所をアプライアンスに指示するサービスパス AVP を指定します。プライマリ PCEF 機能については、インターフェイスタイプを RADIUS と GX と指定します。

例:

```
set ns param -servicePathIngressVLAN 100

set subscriber gxinterface -servicepathAVP 1001 1005 -
    servicepathVendorid 10415

set subscriber param -interfaceType RadiusAndGx
<!--NeedCopy-->
```


Diameter タイプのサービスと仮想サーバーを設定し、サービスを仮想サーバーにバインドします。次に、PCRF レルムとサブスクライバ Gx インターフェイスパラメータを指定します。PCEF の主な機能として、RADIUS リスナーサービスと RADIUS インターフェイスを設定します。

例:

```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -
    holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 RADIUSListener 1813

set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

VAS を入力 VLAN に関連付けるサービス機能を追加します。サービスパスを追加してチェーンを定義します。つまり、パケットの送信先の VAS と、その VAS に送信する順序を指定します。サービスパス名は通常 PCRF によって送信されます。ただし、次のいずれかに当てはまる場合は、デフォルトのサブスクライバプロファイル（*）のサービスパスが適用されます。

- PCRF には加入者情報はありません。
- 加入者情報にはこの AVP は含まれていません。
- アプライアンスは PCRF にクエリを実行できません。たとえば、PCRF を表すサービスは DOWN です。

この名前を含むサービスパス AVP は、事前にグローバル設定の一部として設定しておく必要があります。サービス関数をサービスパスにバインドします。サービスインデックスは、VAS がチェーンに追加される順序を指定します。最も大きい数字 (255) はチェーンの始まりを示します。

例:

```
add ns servicefunction SF1 -ingressVLAN 20

add ns servicefunction SF2 -ingressVLAN 40

add ns servicefunction SF3 -ingressVLAN 60

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

bind ns servicepath pol1 -servicefunction SF2 -index 254
```



```
12
13 bind ns servicepath pol1 -servicefunction SF3 -index 253
14
15 add ns servicepath pol2
16
17 bind ns servicepath pol2 -servicefunction SF2 -index 255
18
19 add ns servicepath pol3
20
21 bind ns servicepath pol3 -servicefunction SF1 -index 255
22
23 add subscriber profile * -subscriberrules default_path
24 <!--NeedCopy-->
```

LSN 設定を追加します。つまり、NAT プールを定義し、アプライアンスが LSN を実行する必要があるクライアントを特定します。

例:

```
1  add lsn pool pool1
2
3  bind lsn pool pool1 200.201.1.1
4
5  add lsn client client1
6
7  bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0
8
9  add lsn group group1 -clientname client1
10
11 bind lsn group group1 -poolname pool1
12 <!--NeedCopy-->
```

アプライアンスはデフォルトで LSN を実行します。LSN をオーバーライドするには、overrideLSN パラメータを有効にしたネットプロファイルを作成し、このプロファイルを付加価値サービス（VAS）用に構成されたすべての負荷分散仮想サーバーにバインドする必要があります。

例:

```
1  add netprofile np1
2
3  set netprofile np1 -overrideLsn ENABLED
4
5  set lb vserver vs1 -netprofile np1
6  <!--NeedCopy-->
```

アプライアンスで VAS を設定します。これには、サービスと仮想サーバーを作成し、サービスを仮想サーバーにバインドすることが含まれます。

例:

```
1  add service vas1 192.168.10.2 ANY 80 -usip YES
2
```



```
add service vas2 192.168.30.2 ANY 80 -usip YES

add service vas3 192.168.50.2 ANY 80 -usip YES

add service sint 200.10.1.10 ANY 80 -usip YES

add lb vserver vs1 ANY -m MAC -l2Conn ON

add lb vserver vs2 ANY -m MAC -l2Conn ON

add lb vserver vs3 ANY -m MAC -l2Conn ON

add lb vserver vint ANY -m MAC -l2Conn ON

bind lb vserver vs1 vas1

bind lb vserver vs2 vas2

bind lb vserver vs3 vas3

bind lb vserver vint sint
<!--NeedCopy-->
```

コンテンツスイッチング (CS) 構成を追加します。これには、仮想サーバー、ポリシー、およびそれらに関連するアクションが含まれます。トラフィックは CS 仮想サーバに到着し、適切な負荷分散仮想サーバにリダイレクトされます。仮想サーバーとサービス機能を関連付ける式を定義します。

例**:**

```
add cs vserver cs1 ANY * 80 -l2Conn ON

add cs action csact1 -targetLBVserver vs1

add cs action csact2 -targetLBVserver vs2

add cs action csact3 -targetLBVserver vs3

add cs action csactint -targetLBVserver vint

add cs policy cspol1 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF1") &&
   SYS.VSERVER("vs1").STATE.EQ(UP)" -action csact1

add cs policy cspol2 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF2") &&
   SYS.VSERVER("vs2").STATE.EQ(UP)" -action csact2

add cs policy cspol3 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF3") &&
   SYS.VSERVER("vs3").STATE.EQ(UP)" -action csact3

bind cs vserver cs1 -policyName cspol1 -priority 110

bind cs vserver cs1 -policyName cspol2 -priority 120

bind cs vserver cs1 -policyName cspol3 -priority 130
```



```
22
23 bind cs vserver cs1 -lbvserver vint
24 <!--NeedCopy-->
```

GUI を使用してアプライアンスのサービスチェーニングを設定するには

1. [システム] > [ネットワーク] > [**IP**] に移動し、サブネット IP アドレスを追加します。
2. [システム] > [ネットワーク] > [**VLAN**] に移動し、VLAN を追加し、VLAN をインターフェイスとサブネット IP アドレスにバインドします。
3. [トラフィック管理] > [サービスチェーン] > [サービスパスの入力 **VLAN** の設定**]** に移動し、入力 **VLAN** を指定します。
4. [トラフィック管理] > [サブスクライバ] > [パラメータ] > [サブスクライバパラメータの設定] に移動し、以下を指定します。
   - インターフェイスの種類: **RADIUS** と **GX** を指定します。
   - Diameter 仮想サーバ、PCRF レルム、およびサブスクライバ GX インターフェイスパラメータを設定します。
   - RADIUS インターフェイスパラメータを指定します。
5. [トラフィック管理] > [サービスチェーン] > [サービス機能] に移動し、サービス機能を追加して付加価値サービスを入力 VLAN に関連付けます。
6. [システム] > [ネットワーク] > [大規模 **NAT**] に移動します。[プール] をクリックし、プールを追加します。[クライアント] をクリックし、クライアントを追加します。「グループ」をクリックしてグループを追加し、クライアントを指定します。グループを編集し、プールをこのグループにバインドします。
7. [システム] > [ネットワーク] > [ネットプロファイル] に移動し、ネットプロファイルを追加します。[**LSN** のオーバーライド] を選択します。必要に応じて、[システム] > [ネットワーク] > [設定**] > [**レイヤ **3** パラメータの設定] に移動し、[**LSN** の上書き] が選択されていないことを確認します。
8. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、アプライアンス上の仮想サーバーと付加価値サービスを設定します。サービスとネットプロファイルを仮想サーバーにバインドします。
9. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバー、ポリシー、およびアクションを設定します。ターゲットの負荷分散仮想サーバーを指定します。

TCP 最適化による利用者に応じたトラフィックステアリング

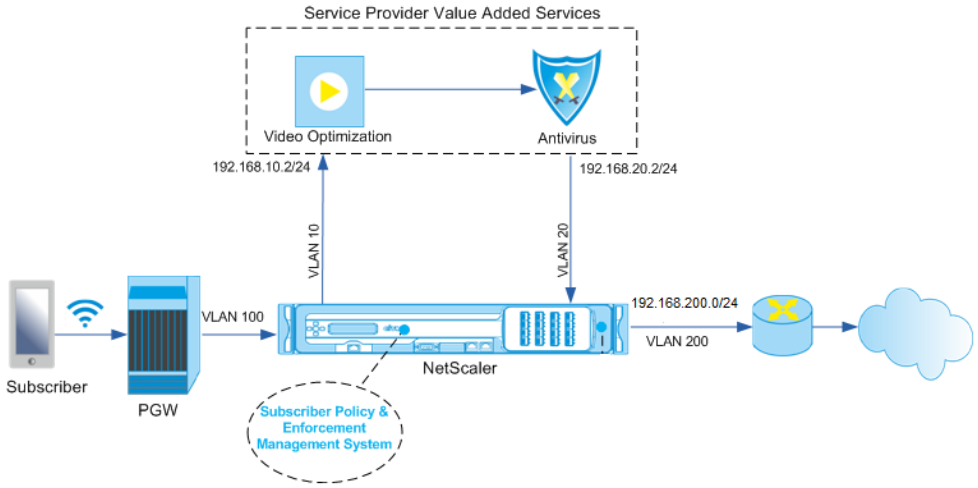
August 15, 2023

トラフィックステアリングは、加入者のトラフィックをある地点から別の地点に誘導します。サブスクライバーがネットワークに接続すると、パケットゲートウェイは IP アドレスをサブスクライバーに関連付け、データパケットを NetScaler アプライアンスに転送します。アプライアンスは Gx インターフェイスを介して PCRF サーバと通信し、

サブスクライバポリシー情報を取得します。ポリシー情報に応じて、アプライアンスは次のいずれかのアクションを実行します。

- データパケットを別のサービスセットに転送します (次の図を参照)。
- TCP 最適化のみを実行してください。

次の図に示されている値は、図に続く CLI プロシージャで設定されます。NetScaler アプライアンス上のコンテンツスイッチング仮想サーバーは、定義されたルールに応じて、付加価値サービスに要求を送信するか、要求をスキップして TCP 最適化を実行し、パケットをインターネットに送信します。



> 注
>
> 以下に示す設定のサポートは、リリース 11.1 ビルド 50.10 で導入されました。

CLI を使用して上記のデプロイメントのトラフィックステアリングを設定するには:

1. アプライアンスのサブネット IP (SNIP) アドレスを追加します。

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 192.168.100.1 255.255.255.0 -type snip

add ns ip 192.168.200.1 255.255.255.0 -type snip

add ns ip 10.102.232.236 255.255.255.0  - type snip
<!--NeedCopy-->
```

2. VLAN を追加します。VLAN は、アプライアンスがトラフィックの送信元を識別するのに役立ちます。VLAN をインターフェイスとサブネット IP アドレスにバインドします。

```
add vlan 10
```



```
2
3  add vlan 20
4
5  add vlan 100
6
7  add vlan 200
8
9  add vlan 102
10
11 bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1
       255.255.255.0
12
13 bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1
       255.255.255.0
14
15 bind vlan 100 -ifnum 1/2 -tagged -IPAddress 192.168.100.1
       255.255.255.0
16
17 bind vlan 200 -ifnum 1/2 -tagged -IPAddress 192.168.200.1
       255.255.255.0
18
19 bind vlan 102  - ifnum 1/1  - tagged  - IPAddress 10.102.232.236
       255.255.255.0
20 <!--NeedCopy-->
```

3. Diameter タイプのサービスと仮想サーバーを設定し、サービスを仮想サーバーにバインドします。PCRF レルムとサブスクライバ Gx インターフェイスパラメータの値を指定します。また、加入者セッション内でアプライアンスがサービスパス名を検索できる場所を示すサービスパス AVP も指定してください。PCEF の主な機能については、RADIUS リスナーサービスと RADIUS インターフェイスを設定し、インターフェイスタイプを「RADIUSandGX」と指定します。

```
1  add service sd1 10.102.232.200 DIAMETER 3868
2
3  add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER
       -persistAVPno 263
4
5  bind lb vserver vdiam sd1
6
7  set ns diameter -identity netscaler.sc1.net -realm pcrf1.net
8
9  set extendedmemoryparam -memLimit 2558
10
11 set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net
12
13 set subscriber gxinterface -servicepathAVP 1001 1005 -
       servicepathVendorid 10415
14
15 add service srad1 10.102.232.236 RADIUSListener 1813
16
17 set subscriber radiusInterface -listeningService srad1
18
19 set subscriber param -interfaceType RadiusAndGx
```



```
20 <!--NeedCopy-->
```

4. 次のいずれかに当てはまる場合に適用されるデフォルトのサブスクライバプロファイル (*) を指定します。

   - PCRF には加入者情報はありません。
   - 加入者情報には、サービスパス AVP は含まれません。
   - アプライアンスは PCRF にクエリを実行できません。たとえば、PCRF を表すサービスは DOWN です。

```
1 add subscriber profile * -subscriberrules default_path
2 <!--NeedCopy-->
```

5. VAS と TCP 最適化パスの TCP プロファイルをそれぞれ作成します。VAS に送られるトラフィックは、VAS を出る前も出た後も、TCP 最適化は行われません。そのため、VAS プロファイルの TCP モードは TRANSPARENT に設定し、TCP プロファイルの TCP モードは ENDPOT に設定する必要があります。

   TCP プロファイル VAS の追加—TCP モードトランスペアレント

   add ns tcpProfile TCPOpt -WS ENABLED -SACK ENABLED -WSVal 8 -mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT

6. VAS サーバのロードバランシングを設定します。TCP タイプのアドレス指定不可の仮想サーバーを作成します。VAS サーバの IP アドレスを使用して TCP サービスを作成し、そのサービスを仮想サーバにバインドします。仮想サーバとサービスは、VAS パス用に作成されたトランスペアレントな TCP プロファイルを使用します。

```
1  add service vas1 192.168.10.2 TCP * -usip YES -useproxyport NO -
       TCPB NO -tcpProfileName VAS
2
3  add service vas2 192.168.10.3 TCP * -usip YES -useproxyport NO -
       TCPB NO -tcpProfileName VAS
4
5  add lb vserver vs1 TCP -m MAC -l2Conn ON  - tcpProfileName VAS
6
7  bind lb vserver vs1 vas1
8
9  bind lb vserver vs1 vas2
10 <!--NeedCopy-->
```

7. VAS 出力トラフィックをキャプチャするロードバランシング仮想サーバを追加します。この仮想サーバは VAS 出力 VLAN を監視し、トランスペアレントな TCP プロファイルを使用します。

```
1 add lb vserver vsint TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ(20)"
       - Listenpriority 30  - l2Conn ON  - tcpProfileName VAS
2 <!--NeedCopy-->
```


8. ワイヤレス側の VLAN 内のすべてのトラフィックを受信し、TCP 最適化パス用に作成されたエンドポイント TCP プロファイルを使用する TCP 最適化仮想サーバーを追加します。

```
add lb vserver vs-TcpOpt TCP * * -Listenpolicy "client.vlan.id.eq(100)" -Listenpriority 20 -l2Conn ON -tcpProfileName TCPOpt
<!--NeedCopy-->
```

9. コンテンツスイッチング (CS) 構成を追加します。これには、仮想サーバー、ポリシー、およびそれらに関連するアクションが含まれます。CS 仮想サーバはトラフィックを受信し、定義された CS ポリシーに従って適切な負荷分散仮想サーバにリダイレクトします。ワイヤレス側 VLAN 内のトラフィックを最も優先して受信し、エンドポイントの TCP プロファイルを使用する CS TCP 仮想サーバを作成します。「vas」がサブスクライバルールの場合に TRUE と評価される CS ポリシーを作成し、トラフィックを VAS に誘導する CS アクションを指定します。TCP 最適化仮想サーバーをデフォルトの LB 仮想サーバーにします。ルールが「vas」以外のサブスクライバートラフィックは、デフォルトの LB 仮想サーバを経由します。

```
add cs vserver cs1 TCP * * -Listenpolicy "client.vlan.id.eq(100)" -Listenpriority 10 -l2Conn ON  -tcpProfileName TCPOpt

add cs action csact1 -targetLBVserver vs1

add cs policy cspol1 -rule  SUBSCRIBER.RULE_ACTIVE("vas") && SYS.VSERVER("vs1").STATE.EQ(UP)" -action csact1

bind cs vserver cs1 -policyName cspol1

bind cs vserver cs1 -lbvserver vs-TcpOpt
<!--NeedCopy-->
```

10. 静的ルートまたはポリシーベースのルートをインターネットに追加します。この構成では動的ルーティングもサポートされています。次の例では、ポリシーベースのルートを使用しています。

```
add ns pbr pbr-vlan100-to-vlan200 ALLOW -nextHop 192.168.200.10 -vlan 100 -priority 10

add ns pbr pbr-vlan20-to-vlan200 ALLOW -nextHop 192.168.200.10 -vlan 20 -priority 11

apply ns pbrs
<!--NeedCopy-->
```

> 注
>
> CS ポリシーには、サブスクライバ表現に加えて IP アドレスとポート番号を含めることができます。たとえば、SUBSCRIBER.RULE_ACTIVE (「vas」) && && (CLIENT.TCP.DSTPORT.EQ (80) など)

> -
> - IPv6 サブスクライバをサポートする IPv6 構成（アドレス、ルート、PBR）を追加します。Happy Eyeballs クライアントアプリケーションは、VAS と TCP の最適化パスの両方でスムーズに動作します。
> - VLAN、IP アドレス、PBR、LB 仮想サーバを VAS（vs1、vs2 など）の前に追加すると、複数のサブスクライバフローをサポートできます。CS 仮想サーバ「cs1」と LB 仮想サーバ「vsint」のリッスンポリシーを変更して、追加の VLAN を含めてください。

ポリシーベースの **TCP** プロファイルの選択

August 15, 2023

NetScaler アプライアンスは、サブスクライバー属性に基づいて TCP 最適化を実行するように構成できます。たとえば、アプライアンスは、ユーザー機器 (UE) が接続されているネットワークに基づいて、実行時に異なる TCP プロファイルを選択できます。その結果、TCP プロファイルにいくつかのパラメーターを設定し、ポリシーを使用して適切なプロファイルを選択することで、モバイルユーザーの操作性を向上させることができます。

4G ネットワーク経由で接続する加入者用と、他のネットワーク経由で接続するユーザ用に、個別の TCP プロファイルを作成します。無線アクセステクノロジータイプ（RAT タイプ）などの加入者パラメータに基づいて選択されるポリシールールを定義します。次の例では、RAT タイプが EUTRAN の場合、より高速な接続をサポートする TCP プロファイルが選択されます (例 1)。その他すべての RAT タイプの値では、異なる TCP プロファイルが選択されます (例 2)。

無線アクセステクノロジーとそのポリシー設定の詳細については、RFC 29.212を参照してください。

> 注
>
> RAT タイプ AVP（AVP コード 1032）は「列挙」タイプで、UE にサービスを提供する無線アクセステクノロジーを識別するために使用されます。
>
> 値「1004」は、ラットがユートランであることを示します。

例 **1**:

```
add ns tcpProfile tcp2 -WS ENABLED -SACK ENABLED -WSVal 8 -initialCwnd
    16 - oooQSize 15000 -slowStartIncr 1 -bufferSize 1000000 -flavor BIC
     - dynamicReceiveBuffering DISABLED -sendBuffsize 1000000 -dsack
    DISABLED -maxcwnd 4000000 -fack ENABLED -minRTO 500 -maxburst 15

add appqoe action appact2 -priority HIGH -tcpprofile tcp2

add appqoe policy apppol2 -rule "SUBSCRIBER.AVP(1032).VALUE.
    GET_UNSIGNED32(0, BIG_ENDIAN).EQ(1004)" -action appact2

bind cs vserver <name>  -policyname apppol2 -priority 20 -type request
<!--NeedCopy-->
```


例 **2**:

```
add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -initialCwnd
    16 - oooQSize 15000 -slowStartIncr 1 -bufferSize 150000 -flavor BIC
    - dynamicReceiveBuffering DISABLED -sendBuffsize 150000 -dsack
    DISABLED -maxcwnd 4000000 -fack ENABLED -minRTO 200 -maxburst 15

add appqoe action appact1 -priority HIGH -tcpprofile tcp1

add appqoe policy apppol1 -rule "SUBSCRIBER.AVP(1032).VALUE.
    GET_UNSIGNED32(0, BIG_ENDIAN).NE(1004)" -action appact1

bind cs vserver <name>  -policyname apppol1 -priority 10 -type request
<!--NeedCopy-->
```

Diameter、**SIP**、および **SMPP** プロトコルに基づくコントロールプレーントラフィックの負荷分散

August 15, 2023

コントロールプレーントラフィックが増加すると、トラフィックがサーバ間で最適に分散されないため、サーバがボトルネックになる可能性があります。そのため、メッセージは負荷分散する必要があります。NetScaler アプライアンスは、Diameter、SIP、SMPP の負荷分散をサポートしています。

SIP

NetScaler を使用すると、UDP または TCP（TLS を含む）を介して SIP メッセージをプロキシサーバーのグループに負荷分散できます。NetScaler には、特定の SIP セッションのパケットを同じ負荷分散 SIP サーバーに転送する Call-ID ベースのパーシスタンスと Call-ID ハッシュ負荷分散方式もあります。
NetScaler のデフォルト表現言語には、セッション開始プロトコル（SIP）接続で動作する多数の式が含まれています。これらの表現は、要求/応答ベースで動作する SIP プロトコルのポリシーで使用することを目的としています。これらの式は、コンテンツスイッチング、レート制限、レスポンダ、および書き換えポリシーで使用できます。

詳細については、「SIP サーバグループのロードバランシング」を参照してください。

SMPP

Short Message Peer to Peer（SMPP）プロトコルを使用して、個人と銀行、広告主、ディレクトリサービスなどの付加価値サービスプロバイダーとの間で毎日何百万ものショートメッセージが交換されています。多くの場合、サーバーが過負荷になり、トラフィックがサーバー間で最適に分散されないため、メッセージの配信が遅れます。
NetScaler アプライアンスは、サーバー全体にメッセージを最適に分散し、パフォーマンスの低下やシステム停止を防ぎます。NetScaler アプライアンス:

- サーバーから送信されるメッセージとクライアントから送信されるメッセージの負荷を分散します
- メッセージセンターの状態を監視します
- メッセージセンターのコンテンツスイッチングサポートを提供
- 連結メッセージを処理します

制限: メッセージセンターからの 59 バイトを超えるメッセージ ID はサポートされていません。メッセージセンターから返されたメッセージ ID の長さが 59 バイトを超える場合、補助操作は失敗し、NetScaler ADC アプライアンスはエラーメッセージで応答します。

詳細については、SMPP ロードバランシングを参照してください。

Diameter

Diameter ameter は、50 を超えるプロトコル（アプリケーションとも呼ばれる）が構築された基本プロトコルです。そのため、電話会社のネットワークで生成されるトラフィックの直径は大きくなります。この直径のトラフィックを最適に維持するために、NetScaler アプライアンスは負荷分散とコンテンツスイッチングを実行し、リレーエージェントとして機能します。さらに、アプライアンスにはリライト機能とレスポンダー機能があります。アプライアンスは Diameter メッセージのレート制限をサポートしています。

詳細については、「DiametR ロードバランシングの設定」を参照してください。

通信サービスプロバイダーに負荷分散、キャッシュ、ログ記録などの **DNS** インフラストラクチャ/トラフィックサービスを提供する

August 15, 2023

通信サービスプロバイダーは、NetScaler アプライアンスを DNS プロキシとして機能するように構成できます。DNS プロキシの重要な機能である DNS レコードのキャッシュは、NetScaler ADC アプライアンスではデフォルトで有効になっています。これにより、NetScaler アプライアンスは翻訳の繰り返しに迅速に対応できるため、カスタマーエクスペリエンスが向上し、帯域幅も節約できます。は DNS ネームサーバからの応答をキャッシュします。アプライアンスは DNS クエリを受信すると、クエリされたドメインをキャッシュで確認します。クエリされたドメインのアドレスがキャッシュに存在する場合、NetScaler アプライアンスは対応するアドレスをクライアントに返します。それ以外の場合は、クエリを DNS ネームサーバーに転送し、DNS ネームサーバーがアドレスの可用性を確認し、NetScaler アプライアンスに返します。その後、NetScaler アプライアンスはアドレスをクライアントに返します。

以前にキャッシュされたドメインへのリクエストの場合、NetScaler アプライアンスは、構成済みの DNS サーバーにクエリを実行せずにキャッシュからドメインのアドレスレコードを処理するため、帯域幅を節約できます。

11.0 リリース以降、NetScaler は受信した DNS リクエストとクライアントに送信した応答もログに記録します。通信サービスプロバイダーは、このログを次の目的で使用できます。

- クライアントに対する DNS 応答を監査します。
- DNS クライアントを監査する
- DNS 攻撃の検出と防止
- トラブルシューティング

詳細については、「ドメインネームシステム」を参照してください。

通信サービスプロバイダーのコアネットワーク全体で **GSLB** を使用して利用者に負荷分散を提供する

August 15, 2023

スケーラビリティ、高可用性、およびパフォーマンスは、サービスプロバイダーの導入にとって不可欠です。多くのサービスプロバイダーはインフラストラクチャを 1 か所または複数の場所に導入していますが、これらの導入には次のような固有の制限があります。

- サイトがパブリックインターネットの全部または一部に接続できなくなると、ユーザーや顧客はそのサイトにアクセスできなくなり、ビジネスに大きな影響を与える可能性があります。
- 地理的に離れた場所からサイトにアクセスするユーザーは、大きく変動する遅延が発生する可能性があり、HTTP がコンテンツを転送するために何度も往復する必要があるため、さらに悪化します。

NetScaler アプライアンスのグローバルサーバー負荷分散（GSLB）は、複数の地理的場所に配置されたサイト間でトラフィックを分散することで、これらの問題を克服します。GSLB は、インターネット上のさまざまな場所からコンテンツを提供することで、ネットワーク帯域幅のボトルネックの影響を軽減し、特定のサイトでネットワーク障害が発生した場合の堅牢性を実現します。ユーザーは、リクエスト時に最も近いサイトまたは最も負荷の少ないサイトに自動的に誘導されるため、長時間のダウンロード遅延やサービスの中断の可能性を最小限に抑えることができます。

NetScaler アプライアンスのグローバルサーバー負荷分散は次の用途に使用できます。

- アクティブデータセンターとスタンバイデータセンターで構成されるアクティブ/スタンバイデータセンター設定を構成することによるディザスタリカバリまたは高可用性。災害によってフェイルオーバーが発生すると、スタンバイデータセンターが稼働状態になります。
- 複数のアクティブなデータセンターで構成されるアクティブ-アクティブデータセンターのセットアップを構成することにより、高い可用性と速度を実現します。クライアント要求は、アクティブなデータセンター間で負荷分散されます。
- 近接設定を設定して、地理的距離またはネットワーク距離が最も近いデータセンターにクライアントの要求を転送します。
- フル DNS 解像度では、GSLB は A、AAAA、CNAME タイプの DNS クエリを処理し、DNS 機能オプションは MX や PTR など、他のすべてのタイプの DNS クエリを処理できます。また、再帰的な解決が有効になって

いる場合、アプライアンスは NetScaler ADC アプライアンスで構成されていないドメイン名に対する DNS クエリを転送します。

詳しくは、「グローバルサーバー負荷分散」を参照してください。

キャッシュリダイレクト機能を使用した帯域幅使用率

August 15, 2023

インターネット上のウェブトラフィックの量は膨大で、そのトラフィックの大部分は冗長です。複数のクライアントが Web サーバーに同じコンテンツを繰り返し要求するため、帯域幅の使用が非効率になります。インターネットサービスプロバイダー（ISP）は NetScaler アプライアンスのキャッシュリダイレクト機能を使用して、オリジンサーバーではなくキャッシュサーバーからコンテンツを提供することで、オリジン Web サーバーから各リクエストを処理する必要がなくなります。NetScaler アプライアンスは、受信したリクエストを分析し、キャッシュ可能なデータのリクエストをキャッシュサーバーに送信し、キャッシュ不可能なリクエストと動的 HTTP リクエストをオリジンサーバーに送信します。NetScaler のキャッシュリダイレクト機能はポリシーベースで、デフォルトでは、ポリシーに一致するリクエストはオリジンサーバーに送信され、他のすべてのリクエストはキャッシュサーバーに送信されます。コンテンツスイッチングとキャッシュリダイレクトを組み合わせて、選択的なコンテンツをキャッシュし、特定の種類の要求コンテンツに対して特定のキャッシュサーバーからコンテンツを提供できます。

詳細については、「キャッシュリダイレクト」を参照してください。

NetScaler ADC の TCP 最適化

August 15, 2023

NetScaler アプライアンスは、最新の 3.5G および 4G ネットワークに適した高度な TCP チューニングと最適化の手法と機能を備えているため、ユーザーエクスペリエンスと認識されるダウンロード速度が大幅に向上します。

このセクションでは、以下に関連する詳細な説明を中心に説明します。

- TCP 最適化のための適切な NetScaler T1000 シリーズモデルの選択とモバイルネットワークへの挿入
- TCP の最適化だけでなく、T1 デバイスの適切なレイヤ 2 およびレイヤ 3 設定に関する詳細な設定手順

このセクションには、以下のトピックが含まれています。

- Getting Started
- 管理ネットワーク
- Licensing
- 高可用性

Getting Started

August 15, 2023

ハードウェア

NetScaler にはさまざまな NetScaler モデルが用意されていますが、大まかに次の 2 つの要因に基づいて判断できます。

- 容量は、現在、ローエンドの VPX アプライアンスの数百 Mbps から、ハイエンドの 25000 MPX シリーズアプライアンスの 160Gbps までさまざまです
- 通信事業者向けグレード。通信事業者のデータセンター向けに T1000 シリーズも用意されています。

NetScaler の営業担当者またはサポート担当者が、デモ、トライアル、または本番環境のニーズに適したハードウェアを選択するお手伝いをします。

このセクションの残りの部分では、NetScaler T1200 をリファレンスハードウェアとして使用します。使用可能なインターフェイスの数と表記（注の*を参照）、または NetScaler VPX 制限事項が十分に文書化されていること（注の*を参照）に関連する表面的な違いはさておき、選択した NetScaler モデルに関係なく、説明はほとんどそのまま適用されるはずです。

> 注
>
> * たとえば、T1010 モデルには、このドキュメントで使用されている 10/x 表記ではなく、通常 1/1-1/12 と表示されている 12x1GbE しかありません。
>
> ** NetScaler VPX インスタンスは通常、LACP アグリゲーションをサポートしていません。VLAN タグをサポートしていない場合もあります。

初期セットアップ

シリアルコンソール経由

シリアルケーブルを接続すると、次の認証情報を使用して NetScaler アプライアンスにログオンできます。

- ユーザー名：nsroot
- パスワード：nsroot

ログインしたら、以下のスクリーンキャプチャに示すように、NetScaler アプライアンスの基本的な詳細を設定します。

例**:**

```
set ns config  - IPAddress <ip_addr> -netmask <netmask>

saveconfig

reboot -warm
<!--NeedCopy-->
```

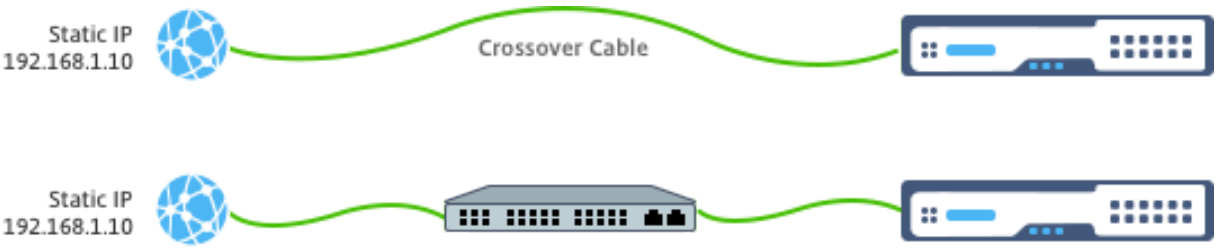
アプライアンスを再起動したら、SSH を使用して T1100 ノードをさらに構成できます。

LOM を通じて

NetScaler アプライアンスのフロントパネルにあるライトアウト管理（LOM）ポートにより、オペレーターはオペレーティングシステムとは独立してアプライアンスをリモートで監視および管理できます。オペレーターは、LOM ポートを介して NetScaler アプライアンスに接続することで、IP アドレスの変更、電源の再投入、およびコードダンプを実行できます。

LOM ポートのデフォルト IP アドレスは 192.168.1.3 です

フィギュア。LOM モジュールの初期設定



ラップトップにスタティック IP を設定し、クロスオーバーケーブルで LOM インターフェイスに直接接続するか、LOM インターフェイスと同じブロードキャストドメイン内のスイッチに接続します。

http://192.168.1.3 初期設定では、Web ブラウザにポートのデフォルトアドレスを入力し、LOM ポートのデフォルト IP アドレスを変更します。

詳細については、『設定ガイド』を参照してください。

ソフトウェア

モバイルネットワーク向けの NetScaler TCP 最適化は常に進化しています。このドキュメントで説明されている機能とチューニングには、NetScaler Telco のビルドが必要です。NetScaler Telco のビルドの例を次に示します。

例**:**

```
show ver

NetScaler NS11.0: Build 64.957.nc, Date: Aug 26 2016, 02:00:23
<!--NeedCopy-->
```

T1000 に適切なビルドリビジョンが付属していない場合は、NetScaler カスタマーサポートに連絡してください。

> 重要
>
> 両方のアプライアンスに同じソフトウェアイメージが必要です。

SSH クライアント

NetScaler アプライアンスは、CLI または HTML5 GUI のいずれかを使用して構成できます。ただし、このセクションでは CLI ベースの説明のみを提供します。

CLI には NetScaler ADC シリアルコンソールからアクセスできますが、通常は SSH クライアントを使用してリモート NetScaler ADC 構成を行うことをお勧めします。

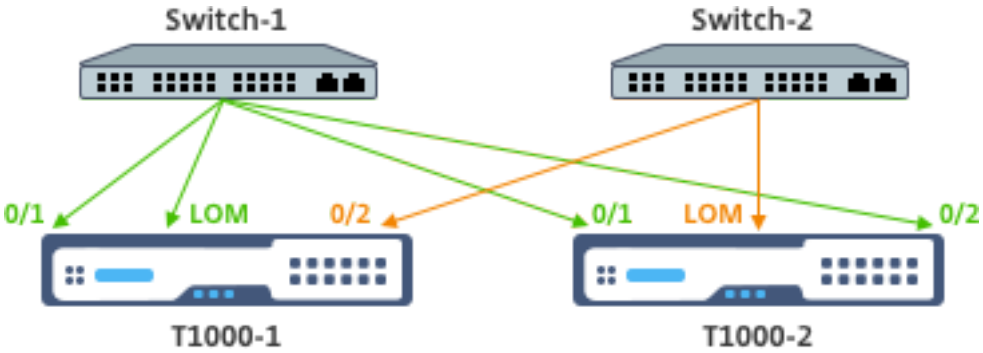
管理ネットワーク

August 15, 2023

接続

ほとんどの NetScaler デバイスには、0/1 および 0/2 と表記された冗長 1GbE OAM ポートがあります。スイッチに障害が発生した場合に冗長性を確保するには、関連するポートを異なるアップストリームスイッチに接続する必要があります。

推奨接続の大まかな概要を次の図に示します。



NetScaler アプライアンスを管理ネットワークに接続した後は、CLI と GUI への SSH または Web 接続をそれぞれ使用して、以降の構成手順をリモートで実行できます。

ルーティング

add route コマンドを使用して、管理ネットワークに適した任意のルートを設定できます。次に示すように、関連するゲートウェイには NSIP サブネット上でアクセスできる必要があります。

例:

```
add route <network>  <netmask>  <gateway>
<!--NeedCopy-->
```

ライセンス

August 15, 2023

NetScaler ADC アプライアンスに有効なライセンスファイルをインストールする必要があります。ライセンスは、予想される最大 Gi-LAN スループットと同じ数の Gbps を少なくともサポートする必要があります。

ライセンスファイルは、以下のスクリーンキャプチャに示すように、SCP クライアントを介してアプライアンスの/nsconfig/license にコピーする必要があります。

例:

```
shell ls /nsconfig/license/

CNS_V3000_SERVER_PLT_Retail.lic ssl
<!--NeedCopy-->
```

次のスクリーンキャプチャに示すように、ウォームリスタートを実行して新しいライセンスを適用します。

例:


```
reboot -warm

Are you sure you want to restart NetScaler (Y/N)? [N]:y

Done
<!--NeedCopy-->
```

再起動が完了したら、show license CLI を使用して、ライセンスが正しく適用されていることを確認します。

以下の例では、3Gbps Premium ライセンスが正常にインストールされています。

例:

```
> show license

              License status:

                                      Web Logging: YES

                                                ...

                                  Model Number ID: 3000

                                     License Type: Premium License

Done

<!--NeedCopy-->
```

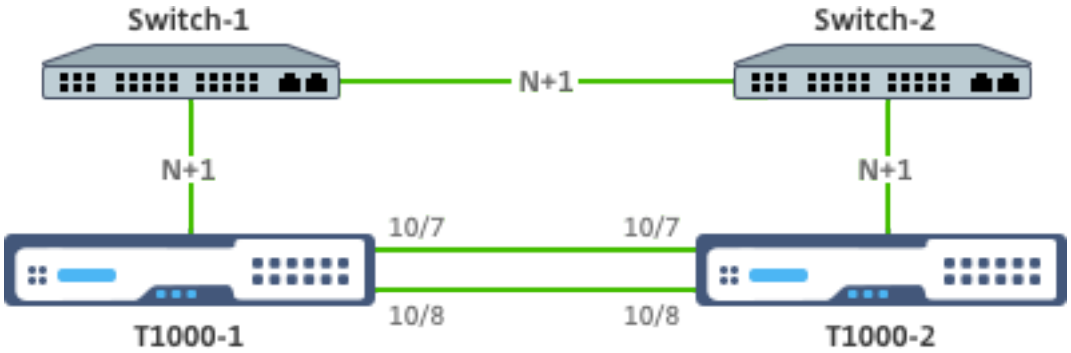
高可用性

August 15, 2023

高可用性（HA）とは、NetScaler デバイスペアのアクティブ/スタンバイ操作モードを指します。各デバイスには専用の管理 IP アドレスがあります。その他の IP アドレスはすべて、ペアのアクティブデバイスが所有します。

接続

NetScaler HA ペアには複数の接続オプションがありますが、最も推奨される接続オプションを次の図に示します。



上の図では、接続で説明されているように、各 T1000 とそれぞれのスイッチ間の N+1 赤色のリンクは N+1 冗長性を意味します。たとえば、45 Gbps の GiLAN N=5 が適切な値であると考えると、各スイッチとそれぞれの T1000 間および 2 つのスイッチ間で 6 x 10 GbE LACP チャネルが使用されます。

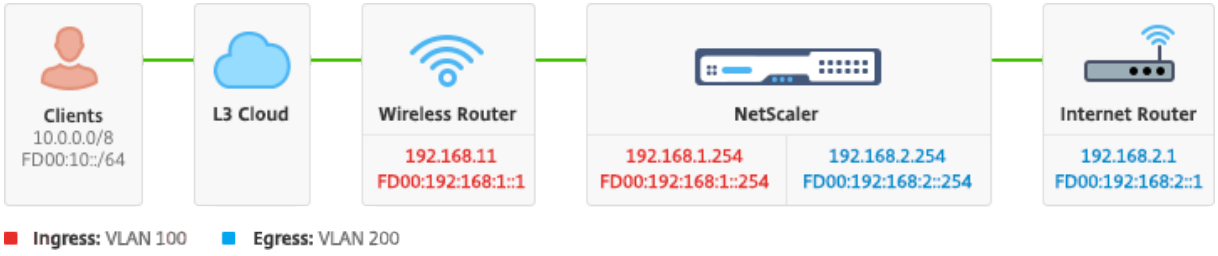
OAM ネットワークから HA 通信を分離するために、NetScaler ペア間には追加のリンクのペアを推奨します。

Gi-LAN 統合

August 15, 2023

通常、NetScaler ADC アプライアンスは、L3 ルーターと同様に、Gi-LAN に個別の L3 インラインノードとして挿入されます。

図:Gi-lan の単純な描写



接続

十分な冗長性を確保するために、アップストリームスイッチへの物理的な NetScaler 接続をお勧めします。たとえば、合計（アップリンク + ダウンリンク）24Gbps を処理する Gi-LAN に NetScaler ADC アプライアンスが挿入されていると仮定すると、4x10GbE 以上のインターフェイスとの接続を推奨します。これにより、リンク障害が発生した場合に N+1 の冗長性が効果的に提供されます。

アップストリームスイッチの関連ポートは、LACP ポート集約用に設定する必要があります。NetScaler 関連する構成の概要を以下に示します。

接続構成**:**


```
set interface 10/1 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/2 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/3 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/4 -tagall ON -lacpMode ACTIVE -lacpKey 1
<!--NeedCopy-->
```

「show interface」コマンドを使用すると、LACP の適切な機能を確認できます。

インターフェイスの表示**:**

```
sh interface LA/1

    1)      Interface LA/1 (802.3ad Link Aggregate) #39

             flags=0x4100c020 <ENABLED, UP, AGGREGATE, UP, HAMON, 802.1
                q>

             MTU=1500, native vlan=1, MAC=02:e0:ed:33:88:b0, uptime 340
                h11m56s

             Requested: media NONE, speed AUTO, duplex NONE, fctl NONE,

             throughput 0

             Actual: throughput 4000

             LLDP Mode: NONE,

             RX: Pkts(918446) Bytes(110087414) Errs(0) Drops(795989)
                Stalls(0)

             TX: Pkts(124113) Bytes(15255532) Errs(0) Drops(0) Stalls
                (0)

             NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0)
                Muted(0)

             Bandwidth thresholds are not set.

Disable the remaining unused interfaces and turn off the monitor.

set interface 10/5 -haMonitor OFF
<!--NeedCopy-->
```

コマンド:

```
set interface 10/24 -haMonitor OFF

disable interface 10/5

```



```
disable interface 10/24
<!--NeedCopy-->
```

物理インターフェイスの構成は、2 つの NetScaler ユニット間で共有されません。したがって、HA ペアを展開する場合は、上記のコマンドを両方の NetScaler ADC ノードで実行する必要があります。

HA 設定

他のすべての構成パラメータは、HA ペアの NetScaler ADC ノード間で共有されます。したがって、HA 同期は、他の構成コマンドを実行する前に有効にしておく必要があります。基本的な HA 設定には、次の手順が含まれます。

1. まったく同じ NetScaler ADC ハードウェア、ソフトウェア、ライセンスの使用：異なるモデル（T1100 と MPX21550 など）またはファームウェアレベルが異なる同じモデル間では、HA ペアはサポートされません。既存の HA ペアのアップグレード（リリース 11.1 へのアップグレード）に関する適切な手順を参照してください。

2. HA ペアの確立。

例:

```
netscaler-1> add HA node 1 <netscaler-2-NSIP>

netscaler-2> add HA node 1 <netscaler-1-NSIP>
<!--NeedCopy-->
```

3. いずれかのノードで次のコマンドを実行して HA ペアの確立を確認します。両方のノードが認識され、一方はプライマリ（アクティブ）で、もう一方はセカンダリ（スタンバイ）になります。

例:

```
show HA node
<!--NeedCopy-->
```

4. フェイルセーフモードと maxFlips を有効にします。これにより、両方のノードでルートモニタに障害が発生しても、アクティブ/スタンバイステータスが常に切り替わることなく、少なくとも 1 つのノードがアクティブのままになります。

例:

```
set HA node  - failsafe ON

set HA node -maxFlips 3 -maxFlipTime 1200
<!--NeedCopy-->
```

5. 最後に、OAM ネットワークではなく、専用のイントラ NetScaler ADC ポートで高可用性同期を実行できるようにします。

例:

```
add vlan 4080 -aliasName syncVlan
```



```
2
3 set HA node -syncvlan 4080
4 <!--NeedCopy-->
```

> 注
>
> 上の例のコマンドの VLAN 4080 は、文字どおりに解釈すべきではありません。未使用の VLAN-ID は予約されている可能性があります。

VLAN 設定

物理インターフェイスを適切に設定したら、適切な Gi-LAN VLAN を設定することができます。たとえば、100/101 の VLAN 識別子を持つ入力/出力 VLAN ペアを持つ、かなり単純な Gi-LAN 環境を考えてみましょう。

次のコマンドは、前の手順で作成した LACP チャネルの上に関連する VLAN を設定します。

```
1 add vlan 100
2 add vlan 101
3 bind vlan 100 -ifnum LA/1 -tagged
4 bind vlan 101 -ifnum LA/1 -tagged
5 <!--NeedCopy-->
```

IPv4 設定

通常、NetScaler ADC アプライアンスでは、VLAN ごとに 1 つの SNIP が必要です。次の例では、このページの冒頭にある Gi-LAN 統合図で概説されているネットワークに /24 サブネットマスクがあることを前提としています。

```
1 add ns ip 192.168.1.254 255.255.255.0 -vserver DISABLED -mgmtAccess
     DISABLED
2 add ns ip 192.168.2.254 255.255.255.0 -vserver DISABLED -mgmtAccess
     DISABLED
3 <!--NeedCopy-->
```

SNIP を設定したら、適切な VLAN に関連付ける必要があります。

```
1 bind vlan 100 -IPAddress 192.168.1.254 255.255.255.0
2 bind vlan 101 -IPAddress 192.168.2.254 255.255.255.0
3 <!--NeedCopy-->
```

IPv4 スタティックルーティング

「管理ネットワーク」セクションで説明した例では、2 つのスタティックルーティングルールしか求めません。

- 入力ルータを経由してクライアントへの 10.0.0.0/8 スタティックルート
- 出力ルータを経由したインターネットへのデフォルトルート

例:

```
add route 0.0.0.0 0.0.0.0 192.168.2.1
add route 10.0.0.0 255.0.0.0 192.168.1.1
<!--NeedCopy-->
```

IPv4 ポリシーベース (VLAN-VLAN) ルーティング

NetScaler ADC アプライアンスでは、静的ルーティングの代わりにポリシーベースのルーティングが可能で、ルーティングの決定は通常、宛先 IP ではなく着信インターフェイスや VLAN に対してキーイングされます。ポリシーベースのルーティングは、クライアントの送信元 IP アドレス範囲が定期的に変更される場合や、パケットの宛先 IP アドレスだけではルーティングの決定に到達できない場合の (つまり、クライアント IP アドレスが重複している場合)、必須の考慮事項です。複数の VLAN にまたがる)

例:

```
add ns pbr fromWirelessToInternet ALLOW -nextHop 192.168.2.1 -vlan
    100 -priority 10

Done

 add ns pbr fromInternetToWireless ALLOW -nextHop 192.168.1.1 -vlan
     200 -priority 20

Done

apply ns pbrs
<!--NeedCopy-->
```

IPv6 設定

次のコマンドは、VLAN ごとに IPv6 SNIP を割り当てます。次の例では、図: このページの Gi-LAN の簡単な描写に概説されているネットワークに /64 サブネットマスクがあることを前提としています。

コマンド:

```
add ns ip6 fd00:192:168:1::254/64 -vServer DISABLED -mgmtAccess
    DISABLED
add ns ip6 fd00:192:168:2::254/64 -vServer DISABLED -mgmtAccess
    DISABLED
bind vlan 100 -IPAddress fd00:192:168:1::254/64
bind vlan 200 -IPAddress fd00:192:168:2::254/64
<!--NeedCopy-->
```


IPv6 ルーティング

IPv6 アドレッシングが完了すると、IPv6 スタティックルーティングが構成されることがあります。

- fd 00:10:: /64 入力ルーター経由のクライアントへのスタティックルート
- 出力ルータを経由したインターネットへのデフォルトルート

例:

```
add route6 fd00:10::/64 fd00:192:168:1::1
add route6 ::/0 fd00:192:168:2::1
<!--NeedCopy-->
```

または、ポリシーベースルーティングを使用します。

例:

```
add ns pbr6 fromWirelessToInternetv6 ALLOW -vlan 100 -priority 10 -
   nextHop fd00:192:168:2::1

add ns pbr6 fromInternetToWirelessv6 ALLOW -vlan 200 -priority 20 -
   nextHop fd00:192:168:1::1

apply ns pbr6
<!--NeedCopy-->
```

LACP の冗長性とフェールオーバー

HA 構成の場合は、スループットオプションを利用して LACP チャネルの下限しきい値を設定することをお勧めします。たとえば、HA ペアの各 NetScaler アプライアンスとアップストリームスイッチの間に 25 Gbps Gi-LAN と 4x10GbE チャネルを設定して、N+1 リンクの冗長性を提供するとします。

例:

```
set interface LA/1 -haMonitor ON -throughput 29000
<!--NeedCopy-->
```

プライマリアプライアンスとアップストリームスイッチの間に二重リンク障害が発生した場合、サポートできる最大 Gi-LAN スループットは 20 Gbps に低下します。上記の例で 29 Gbps の下限しきい値を設定すると、セカンダリアプライアンス（同様のリンク障害が発生していない）に冗長スイッチオーバーイベントが発生し、Gi-LAN トラフィックは影響を受けません。

ルートモニター

LACP の冗長性に加えて、ルートモニタチェックを設定し、HA ペアの設定に関連付けることもできます。ルートモニターチェックは、NetScaler ADC アプライアンスとネクストホップルーターの間の障害を検出するのに役立ちます。特に、ルーターが直接接続されておらず、アップストリームスイッチを介して接続されている場合に役立ちます。

セクション 2.5.1 のサンプル GiLAN ごとの一般的な HA ルートモニタの設定を次に示します。

```
add route 192.168.1.0 255.255.255.0 192.168.1.1 -msr ENABLED -monitor
    arp
add route 192.168.2.0 255.255.255.0 192.168.2.1 -msr ENABLED -monitor
    arp
bind HA node -routeMonitor 192.168.1.0 255.255.255.0
bind HA node -routeMonitor 192.168.2.0 255.255.255.0
<!--NeedCopy-->
```

TCP 最適化設定

August 15, 2023

TCP 最適化を構成する前に、NetScaler ADC アプライアンスに次の基本構成設定を適用します。

初期設定**:**

```
enable ns feature LB IPv6PT
enable ns mode FR L3 USIP MBF Edge USNIP PMTUD
disable ns feature SP
disable ns mode TCPB
set lb parameter -preferDirectRoute NO
set lb parameter -vServerSpecificMac ENABLED
set l4param -l2ConnMethod Vlan
set rsskeytype -rsstype SYMMETRIC
set ns param -useproxyport DISABLED
<!--NeedCopy-->
```

> 注
>
> rsskeytype システムパラメータを変更した場合は、NetScaler アプライアンスを再起動します。

TCP ターミネーション

NetScaler T1 が TCP 最適化を適用するには、まず着信 TCP トラフィックを終了する必要があります。そのためには、ワイルドカード TCP vserver を作成し、入力トラフィックをインターセプトしてインターネットルーターに転送するように設定する必要があります。

スタティックまたはダイナミックルーティング環境

静的ルーティングまたは動的ルーティングが導入されている環境では、vserver はルーティングテーブル情報を使用してパケットをインターネットルーターに転送できます。デフォルトルートはインターネットルーターを指している

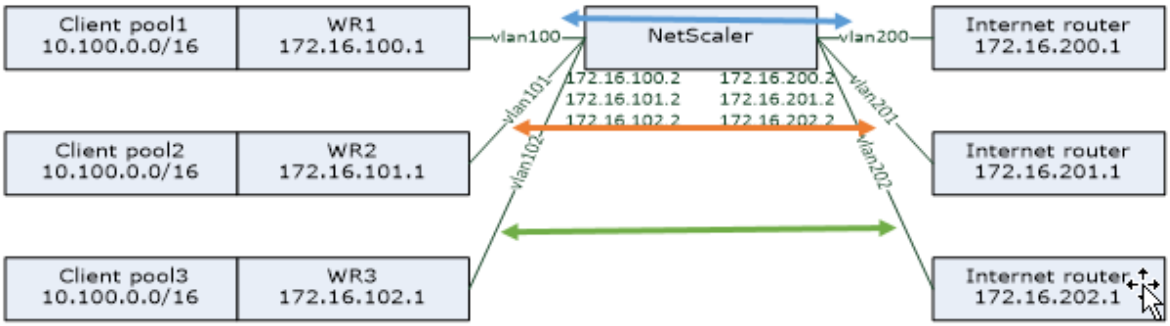
必要があり、ワイヤレスルーターへのクライアントサブネットのルーティングエントリも設定されている必要があります。

例:

```
1 add lb vserver vsrv-wireless TCP * * -persistenceType NONE -
     Listenpolicy "CLIENT.VLAN.ID.EQ(100) && SYS.VSERVER("vsrv-wireless")
     .STATE.EQ(UP)" -m IP -cltTimeout 9000
2 add route 0.0.0.0 0.0.0.0 192.168.2.1
3 add route 10.0.0.0 255.0.0.0 192.168.1.1
4 <!--NeedCopy-->
```

VLAN 間 (PBR) 環境

加入者トラフィックが複数のフローに分割され、受信トラフィックパラメータに基づいて異なるルータに転送する必要があるお客様の環境があります。ポリシーベースルーティング（PBR）を使用すると、VLAN、MAC アドレス、インターフェイス、送信元 IP、送信元ポート、宛先 IP アドレス、宛先ポートなどの着信パケットパラメータに基づいてパケットをルーティングできます。



例:

```
1 add lb vserver vsrv-wireless TCP * * -m IP -l2Conn ON -listenpolicy "
     CLIENT.VLAN.ID.EQ(100) || CLIENT.VLAN.ID.EQ(101) || CLIENT.VLAN.ID.
     EQ(102)"
2
3 add ns pbr pbr-vlan100-to-vlan200 ALLOW -vlan 100 -nexthop 172.16.200.1
4
5 add ns pbr pbr-vlan101-to-vlan201 ALLOW -vlan 101 -nexthop 172.16.201.1
6
7 add ns pbr pbr-vlan102-to-vlan202 ALLOW -vlan 102 -nexthop 172.16.202.1
8 <!--NeedCopy-->
```

ポリシーベースルーティングを使用して TCP 最適化トラフィックをルーティングすることは、リリース 11.1 50.10 で追加された新機能です。以前のリリースでは、VLAN ごとに複数の「モード MAC」仮想サーバーエンティティを持つことが、複数の VLAN 環境の代替ソリューションでした。各仮想サーバーには、特定のフローのインターネットルーターを表すバインドされたサービスがあります。

例:

```
add server internet_router_1 172.16.200.1

add server internet_router_2 172.16.201.1

add server internet_router_3 172.16.202.1

add service svc-internet-1 internet_router_1 TCP * -usip YES -
   useproxyport NO

add service svc-internet-2 internet_router_2 TCP * -usip YES -
   useproxyport NO

add service svc-internet-3 internet_router_3 TCP * -usip YES -
   useproxyport NO

bind service svc-internet-1 -monitorName arp

bind service svc-internet-2 -monitorName arp

bind service svc-internet-3 -monitorName arp

add lb vserver vsrv-wireless-1 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
   (100) && SYS.VSERVER("vsrv-wireless-1").STATE.EQ(UP)" -m MAC -l2Conn
    ON

add lb vserver vsrv-wireless-2 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
   (101) && SYS.VSERVER("vsrv-wireless-2").STATE.EQ(UP)" -m MAC -l2Conn
    ON

add lb vserver vsrv-wireless-3 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
   (102) && SYS.VSERVER("vsrv-wireless-3").STATE.EQ(UP)" -m MAC -l2Conn
    ON

bind lb vserver vsrv-wireless-1 svc-internet-1

bind lb vserver vsrv-wireless-2 svc-internet-2

bind lb vserver vsrv-wireless-3 svc-internet-3
<!--NeedCopy-->
```

> 注:
>
> vserver モードは MAC ですが、これまでの例ではモード IP になっています。これは、サービスが vserver にバインドされている場合に、宛先 IP 情報を保持するために必要です。また、追加の PBR 設定では、最適化されていないトラフィックをルーティングする必要があります。

TCP の最適化

すぐに使用できる NetScaler ADC TCP ターミネーションは、TCP パススルー機能用に構成されています。TCP パススルーとは、基本的に、NetScaler T1 がクライアント/サーバーの TCP ストリームを透過的に傍受できるが、クライアント/サーバーバッファを個別に保持したり、最適化技術を適用したりしないことを意味します。

TCP 最適化を有効にするには、nstcpprofile という名前の TCP プロファイルを使用して TCP 構成を指定します。この構成は、サービスレベルまたは仮想サーバレベルで TCP 構成が提供されていない場合に使用されます。次のように変更する必要があります。

コマンド:

```
add ns tcpProfile nstcpprofile -WS ENABLED -SACK ENABLED -WSVal 8 -mss
    1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA
    ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -
    spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -
    fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT
<!--NeedCopy-->
```

> 注:
>
> プロファイルが明示的に作成されず、vserver とサービスにバインドされていない場合、プロファイル nstcp_default_profile がデフォルトでバインドされます。
>
> 複数の TCP プロファイルが必要な場合は、追加の TCP プロファイルを作成して適切な仮想サーバーに関連付けることができます。

コマンド:

```
add ns tcpProfile custom_profile -WS ENABLED -SACK ENABLED -WSVal 8 -
    mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA
    ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -
    spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -
    fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT

set lb vserver vsrv-wireless -tcpProfileName custom_profile
<!--NeedCopy-->
```

> 注:
>
> vserver-m MAC と service を使用するデプロイでは、同じプロファイルをサービスに関連付ける必要があります。

```
set service svc-internet -tcpProfileName custom_profile
<!--NeedCopy-->
```


TCP 最適化機能

NetScaler ADC アプライアンスの関連する TCP 最適化機能のほとんどは、対応する TCP プロファイルを通じて公開されます。TCP プロファイルを作成する際に考慮すべき一般的な CLI パラメータは次のとおりです。

1. ウィンドウスケーリング **(WS)**: TCP ウィンドウスケーリングにより、TCP 受信ウィンドウサイズを 65535 バイト以上に増やすことができます。これにより、TCP のパフォーマンスが全体的に向上し、特に高帯域幅で遅延が長いネットワークで改善されます。レイテンシを削減し、TCP での応答時間を改善するのに役立ちます。
2. 選択的確認応答 **(SACK)**: TCP SACK は、全体的なスループット容量を低下させる複数のパケット損失の問題に対処します。選択的確認により、受信者は正常に受信されたすべてのセグメントについて送信者に通知できるため、送信者は失われたセグメントのみを再送信できます。この手法は、T1 が全体的なスループットを向上させ、接続遅延を減らすのに役立ちます。
3. ウィンドウスケーリング係数 **(WSVal)**: 新しいウィンドウサイズの計算に使用される係数。NS がアドバタイズするウィンドウが少なくともバッファサイズと等しくなるように、高い値に設定する必要があります。
4. 最大セグメントサイズ **(MSS)**: 単一の TCP セグメントの MSS。この値は、中間ルータとエンドクライアントの MTU 設定によって異なります。1460 の値は MTU が 1500 に相当します。
5. **MaxBurst**: バーストで許可される TCP セグメントの最大数。
6. 初期輻輳ウィンドウサイズ **(InitialCWnd)**: TCP の初期輻輳ウィンドウサイズによって、トランザクションの開始時に未処理のまま残せるバイト数が決まります。これにより、T1 は回線の輻輳を気にすることなく、これだけ多くのバイトを送信できます。
7. **OOO** パケットキューの最大サイズ **(OOOQSize)**: TCP はアウトオブオーダーキューを維持して、OOO パケットを TCP 通信に保持します。この設定は、パケットをランタイムメモリに保持する必要があるためにキューサイズが大きい場合、システムメモリに影響します。したがって、ネットワークの種類とアプリケーションの特性に基づいて、これを最適なレベルに保つ必要があります。
8. 最小 **RTO (minRTO)**: TCP 再送信タイムアウトは、内部実装ロジックに基づいて受信した ACK ごとに計算されます。デフォルトの再送信タイムアウトは最初は 1 秒ですが、この設定で微調整できます。これらのパケットの 2 回目の再送信では、RTO は N*2 で計算され、N*4…N*8…は最後の再送信試行まで続きます。
9. **BufferSize/SendBuffSize**: これらは、T1 がサーバーから受信し、クライアントに送信せずに内部的にバッファリングできる最大データ量を指します。これらは、基盤となる伝送チャネルの帯域幅遅延積よりも大きい（少なくとも 2 倍の）値に設定する必要があります。
10. フレーバー: これは TCP 輻輳制御アルゴリズムを指します。有効な値は、デフォルト、BIC、CUBIC、ウェストウッド、ナイルです。
11. 動的受信バッファリング: メモリとネットワークの状態に基づいて受信バッファを動的に調整できます。固定サイズのバッファーをサーバーから先に読み込んで、クライアントのダウンロードパイプをいっぱいにするのに必要なだけバッファをいっぱいにします。後者は TCP プロファイルで指定され、通常は 2*BDP などの条件に基づいて接続します。NetScaler T1 は、クライアントのネットワーク状態を監視し、サーバーから先に読み取る必要がある量を推定します。
12. キープアライブ（**KA**）: 定期的に TCP キープアライブ（KA）プローブを送信して、ピアがまだ稼働しているかどうかを確認します。
13. RST ウィンドウ減衰: スプーフィング攻撃から TCP を防御します。シーケンス番号が無効な場合は、訂正 ACK

を返信します。

14. **RstMaxAck:** ウィンドウ外であっても最も大きい **ACK** シーケンス番号をエコーする RST の受け入れを有効または無効にします。
15. **SpoofSynDrop**: スプーフィングを防ぐために無効な SYN パケットをドロップします。
16. 明示的な混雑通知 **(ecn)**: ネットワークの混雑状況の通知をデータの送信者に送信し、データの混雑やデータ破損の是正措置を講じます。
17. フォワード **RTO** リカバリ: スプリアス再送信の場合、輻輳制御設定は元の状態に戻ります。
18. **TCP** 最大輻輳ウィンドウ **(maxcwnd)**: ユーザが設定可能な TCP 最大輻輳ウィンドウサイズ。
19. 転送確認 **(FACK)**: ネットワーク内の未処理のデータバイトの総数を明示的に測定し、再送信タイムアウト時に送信者 (T1 またはクライアント) がネットワークに注入されるデータ量を制御できるようにすることで、TCP の輻輳を回避します。
20. **tcpmode**: 特定のプロファイルの TCP 最適化モード。TCP 最適化モードには、トランスペアレントモードとエンドポイントモードの 2 つがあります。

    - エンドポイント。このモードでは、アプライアンスはクライアント接続とサーバー接続を別々に管理します。
    - トランスペアレント。トランスペアレントモードでは、クライアントは仮想サーバを介さずにサーバに直接アクセスする必要があります。クライアントはサーバーにアクセスできる必要があるので、サーバー IP アドレスはパブリックにする必要があります。

アイドル状態の接続をサイレントドロップする

通信ネットワークでは、NetScaler ADC アプライアンスの TCP 接続のほぼ 50% がアイドル状態になり、アプライアンスは RST パケットを送信して接続を閉じます。無線チャネルを介して送信されるパケットは、それらのチャネルを不必要にアクティブ化し、メッセージが殺到し、その結果、アプライアンスから大量のサービス拒否メッセージが生成されます。デフォルトの TCP プロファイルに dropHalfClosedConnonTimeout パラメータと dropEstConnonTimeout パラメータが含まれるようになりました。これらはデフォルトでは無効になっています。両方を有効にすると、ハーフクローズ接続でも確立された接続でも、接続がタイムアウトしたときに RST パケットがクライアントに送信されることはありません。アプライアンスは接続を切断するだけです。

```
set ns tcpProfile nstcpprofile  -DropHalfClosedConnOnTimeout ENABLED
set ns tcpProfile nstcpprofile  -DropEstConnOnTimeout ENABLED
<!--NeedCopy-->
```

分析とレポート

February 15, 2024

TCP 速度レポートは、TCP のダウンロードおよびアップロードのパフォーマンスの尺度として、TCP 接続統計を抽出する NetScaler ADC 機能であり、NetScaler Application Delivery Management（ADM）の TCP Insight レ

ポートで使用されます 。これを実現するために、NetScaler ADC は各 TCP 接続を監視し、アイドルタイムアウトベースでパケットバーストを特定し、識別された最大バーストに関する主要なメトリック（バイト数、再送信されたバイト数、継続時間など）を報告します。TCP 速度レポート機能はデフォルトで有効になっており、TCP と HTTP の両方の vserver をサポートし、AppFlow/ULFD レポートインフラストラクチャに依存します。

リアルタイム統計

August 15, 2023

stat コマンドを使用して、TCP 最適化が適切に適用されていることを確認できます。

コマンド:

```
> stat lb vserver vsrv-wireless
Virtual Server Summary
                      vsvrIP  port     Protocol        State   Health
                         actSvcs
vsrv...eless               *     0          TCP           UP      100
           1

            inactSvcs
vsrv...eless       0
Virtual Server Statistics
                                          Rate (/s)
                                             Total
Vserver hits                                      0
    10
Requests                                          0
                      0
Responses                                         0
                      0
Request bytes                                     0
    1580
Response bytes                                    0
    532594360
Total Packets rcvd                                0
    216463
Total Packets sent                                0
    369898
Current client connections                       --
                      0
Current Client Est connections                   --
                      0
Current server connections                       --
                      0
Requests in surge queue                          --
                      0
Requests in vserver's surgeQ                     --
                      0
```



```
22 Requests in service's surgeQs                          --
                         0
23 Spill Over Threshold                                   --
                         0
24 Spill Over Hits                                        --
                         0
25 Labeled Connection                                     --
                         0
26 Push Labeled Connection                                --
                         0
27 Deferred Request                                        0
                         0
28 Invalid Request/Response                               --
                         0
29 Invalid Request/Response Dropped                       --
                         0
30 Bound Service(s) Summary
31                           IP  port          Type         State     Hits
                                   Hits/s
32 svc-internet     192.168.2.2     0           TCP            UP        10
          0/s
33
34                  Req    Req/s      Rsp    Rsp/s Throughp ClntConn
                       SurgeQ
35 svc-internet       0      0/s        0      0/s         0         0
             0
36                   SvrConn   ReuseP  MaxConn ActvTran  SvrTTFB      Load
37 svc-internet         0         0        0         0         0         0
```

Total カウンタは、運用システムの場合は常に増加します。また、Rate カウンタは 0 以外でなければなりません。

> 注
>
> 上の出力は、動作しているのにアイドル状態のラボシステムからのもので、ゼロレートを説明しています。

SNMP

August 15, 2023

SNMP エージェントは、リモートデバイス（SNMP マネージャ）からシステム固有の情報を問い合わせることができます。クエリに基づいて、エージェントは管理情報ベース（MIB）で要求されたデータの等しいオブジェクト識別子（OID）を検索し、その情報を SNMP マネージャーに送信します。電話会社の導入に最も役立つ SNMP OID は次のとおりです。

メモリ

- **resMemUsage (1.3.6.1.4.1.5951.4.1.1.41.2)**

NetScaler でのメモリ使用率のパーセンテージ。

パケットエンジン **CPU**

- **resCpuUsage (1.3.6.1.4.1.5951.4.1.1.41.1)**

CPU 使用率のパーセンテージ。

- **nsCPUTable (1.3.6.1.4.1.5951.4.1.1.41.6)**

この表には、NetScaler の各 CPU に関する情報が含まれています。

インデックス作成日:NSCPUNAME

- **nsCPUname (1.3.6.1.4.1.5951.4.1.1.41.6.1.1)**

CPU の名前。

- **NSCPU** の使用状況 **(1.3.6.1.4.1.5951.4.1.41.6.1.2)**

CPU 使用率のパーセンテージ。

スループット

- **allNicTotRxMbits (1.3.6.1.4.1.5951.4.1.1.71.1)**

NetScaler アプライアンスが受信したメガビットの数。

- **allNicTotTxMbits (1.3.6.1.4.1.5951.4.1.1.71.2)**

NetScaler アプライアンスによって送信されたメガビットの数。

- **ipTotRxPkts (1.3.6.1.4.1.5951.4.1.1.43.25)**

IP パケットを受信しました。

- **ipTotRxMbits (1.3.6.1.4.1.5951.4.1.1.43.27)**

メガビットの IP データが受信されました。

- **ipTotTxPkts (1.3.6.1.4.1.5951.4.1.1.43.28)**

IP パケットが送信されました。

- **ipTotTxMbits (1.3.6.1.4.1.5951.4.1.1.43.30)**

メガビットの IP データが送信されました。

接続

アクティブな接続**:**

- **tcpActiveServerConn (1.3.6.1.4.1.5951.4.1.1.46.8)**

現在リクエストに応答しているサーバーへの接続。

合計接続数**:**

- **tcpCurServerConn (1.3.6.1.4.1.5951.4.1.1.46.1)**

サーバー接続 (「開始」、「確立」、および「終了」状態の接続を含む)。

- **tcpCurClientConn (1.3.6.1.4.1.5951.4.1.1.46.2)**

クライアント接続 ([開始]、[確立]、[終了] 状態の接続を含む)。

注**:** SYN-Cookie のため、オープン状態のクライアントは含まれません

- **tcpTotZombieCltConnFlushed (1.3.6.1.4.1.5951.4.1.1.46.26)**

クライアントがしばらくアイドル状態だったためにフラッシュされたクライアント接続。

- **tcpTotZombieSvrConnFlushed (1.3.6.1.4.1.5951.4.1.1.46.27)**

キューにクライアントリクエストがしばらくなかったためにフラッシュされたサーバー接続。

エラー

- **tcpErrSynGiveUp (1.3.6.1.4.1.5951.4.1.1.46.37)**

NetScaler で接続を確立しようとしましたが、タイムアウトしました。

- **tcpErrRetransmitGiveUp (1.3.6.1.4.1.5951.4.1.1.46.60)**

その接続でパケットを 7 回再送信した後、NetScaler が接続を終了する回数。再送信は、受信側がパケットを確認しない場合に発生します。

- **ifInDiscards (1.3.6.1.2.1.2.2.1.13)**

上位層プロトコルへの配信を妨げるためにエラーが検出されなかったにもかかわらず破棄される受信パケットの数。このようなパケットを廃棄する理由の 1 つとして、バッファスペースを解放することが考えられます。

- **ifOutDiscards (1.3.6.1.2.1.2.2.1.19)**

送信を妨げるためにエラーが検出されなかったにもかかわらず破棄するように選択された送信パケットの数。このようなパケットを廃棄する理由の 1 つとして、バッファスペースを解放することが考えられます。

- **ifErrTxOverflow (1.3.6.1.4.1.5951.4.1.1.54.1.36)**

NetScaler アプライアンスが起動またはインターフェイス統計がクリアされてから、指定されたインターフェイスでの送信中にオーバーフローキューを通過したパケットの数。この値が増加するのは、混雑しているポートだけです。

最適化/バイパス接続

- **TCP** 最適化が有効 **(1.3.6.1.4.1.5951.4.1.46.131)**

TCP 最適化で有効になっている接続の総数。

- **tcpOptimizationBypassed (1.3.6.1.4.1.5951.4.1.1.46.132)**

TCP 最適化をバイパスした接続の総数。

技術レシピ

August 15, 2023

NetScaler T1 モデルには、複雑な決定を実行時に評価できる高度な機能と強力なポリシー構成言語が用意されています。

T1000 の機能とポリシー設定ガイドで実現できる可能性のあるすべての機能を評価することはできませんが、技術資料では、通信事業者が提示するさまざまな要件の実装を検討します。「レシピ」をそのまま使用したり、環境に合わせて自由に再利用したりできます。

ユーザーごとの接続制限

NetScaler T1 モデルは、一意のサブスクライバー IP あたりの接続数を制限するように構成できます。以下の構成では、IP（CLIENT.IP.SRC）ごとに N 個の同時 TCP 接続が許可されます。設定されたしきい値を超える接続を試みるたびに、T1 は RST を送信します。1 ユーザーあたり最大 2 つの同時接続の場合:

コマンド:

```
1 add stream selector streamSel_usrlimit CLIENT.IP.SRC
2 add ns limitIdentifier limitId_usrlimit -threshold 2 -mode CONNECTION -
     selectorName streamSel_usrlimit
3 add responder policy respPol_usrlimit "SYS.CHECK_LIMIT("
     limitId_usrlimit")" RESET
4 bind lb vserver vsrv-wireless -policyName respPol_usrlimit -priority 1
     -gotoPriorityExpression END
5 <!--NeedCopy-->
```


仮想サーバーのスムーズな挿入/削除

多くのオペレーターは、NetScaler T1 モデルが TCP 最適化のためにインラインでアクティブ化されている場合や、メンテナンスの目的で無効になっている場合に、TCP 接続の中断を懸念しています。vserver の導入時に既存の接続が切断されないようにするには、TCP 最適化用に vserver を設定またはアクティブ化する前に、次の設定を適用する必要があります。

コマンド:

```
add ns acl acl-ingress ALLOW  -vlan 100
add forwardingSession fwd-ingress  -aclname acl-ingress
apply ns acls
<!--NeedCopy-->
```

転送セッションはルーティング（スタティック、ダイナミック、または PBR）に加えて有効であり、ルーティングされるトラフィックのセッションエントリを作成します（L3 モード）。既存の接続はすべて、対応するセッションにより転送セッションによって処理され、vserver が導入されると新しい TCP 接続のみのキャプチャを開始します。

ACL は、メモリを消費する不要なトラフィックのセッションを作成しないように、vserver などの特定のポートのみをキャプチャするように設定できます。もう 1 つのオプションは、vserver のアクティベーション後に特定の設定を削除することです。

メンテナンスのために、仮想サーバーを無効にして、その状態が OUT OF SERVICE と表示されている必要があります。この場合、仮想サーバーはデフォルトですべての接続を直ちに終了します。vserver が既存の接続を引き続き提供し、新しい接続を受け付けないようにするには、次の設定を適用する必要があります。

コマンド:

```
set lb vserver vsrv-wireless  -downStateFlush DISABLED
<!--NeedCopy-->
```

新しい接続はルーティングテーブルを通過し、転送セッションにより対応するセッションエントリが作成されます。

ポリシーベースの **TCP** プロファイリング

ポリシーベースの TCP プロファイル選択により、オペレーターは異なるトラフィックドメイン（3G または 4G など）から来るクライアントに対して TCP プロファイルを動的に設定できます。一部の QoS メトリックはこれらのトラフィックドメインによって異なるため、パフォーマンスを向上させるには、TCP パラメータの一部を動的に変更する必要があります。3G と 4G のクライアントが同じ仮想サーバーにアクセスして同じ TCP プロファイルを使用する場合を考えてみましょう。これにより、一部のクライアントのパフォーマンスに悪影響を及ぼします。AppQoE 機能はこれらのクライアントを分類し、仮想サーバー上の TCP プロファイルを動的に変更できます。

例:

```
enable feature AppQoE

```



```
add ns tcpProfile nstcpprofile1 -WS ENABLED -SACK ENABLED -WSVal 8 -mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -slowStartIncr 1 -bufferSize 4000000 -flavor BIC -KA ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -tcpmode ENDPOINT

add ns tcpProfile nstcpprofile2 -WS ENABLED -SACK ENABLED -WSVal 8 -mss 1460 -maxBurst 15 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -slowStartIncr 1 -bufferSize 128000 -flavor BIC -KA ENABLED -sendBuffsize 6000000 -rstWindowAttenuate ENABLED -spoofSynDrop ENABLED -frto ENABLED -maxcwnd 64000 -fack ENABLED -tcpmode ENDPOINT

add appqoe action action_1 -priority HIGH -tcpprofile nstcpprofile1

add appqoe action action_2 -priority HIGH -tcpprofile nstcpprofile2

add appqoe policy appqoe_4G -rule "CLIENT.VLAN.ID.EQ(100)" -action action_1

add appqoe policy appqoe_3G -rule "CLIENT.VLAN.ID.EQ(200)" -action action_2

bind lb vserver vsrv-wireless -policyName appqoe_4G -priority 100

bind lb vserver vsrv-wireless -policyName appqoe_3G -priority 110
<!--NeedCopy-->
```

NetScaler T1 モデルでは、Gx、Radius、または Radius and Gx インターフェイスを介してサブスクライバー情報を動的に受信し、サブスクライバーごとに異なる TCP プロファイルを適用できます。

コマンド:

```
add appqoe action action_1 -priority HIGH -tcpprofile nstcpprofile1

add appqoe action action_2 -priority HIGH -tcpprofile nstcpprofile2

add appqoe policy appqoe_4G -rule "SUBSCRIBER.RULE_ACTIVE("3G")" -action action_1

add appqoe policy appqoe_3G -rule "SUBSCRIBER.RULE_ACTIVE("4G")" -action action_2
<!--NeedCopy-->
```

NetScaler T1 モデルとオペレータコントロールプレーンネットワークとの統合については、Telco サブスクライバ管理を参照してください。

スケーラビリティ

August 15, 2023

TCP の最適化はリソースを大量に消費するため、たとえハイエンドのアプライアンスであっても、単一の NetScaler アプライアンスでは高い Gi-LAN スループットを維持できない場合があります。ネットワークの容量を拡張するために、NetScaler アプライアンスを N+1 クラスター構成で展開できます。クラスター展開では、NetScaler アプライアンスは単一のシステムイメージとして連携して機能します。クライアントトラフィックは、外部スイッチデバイスを使用してクラスタノード全体に分散されます。

トポロジ

図 1 は、4 つの T1300-40G ノードで構成されるクラスタの例です。

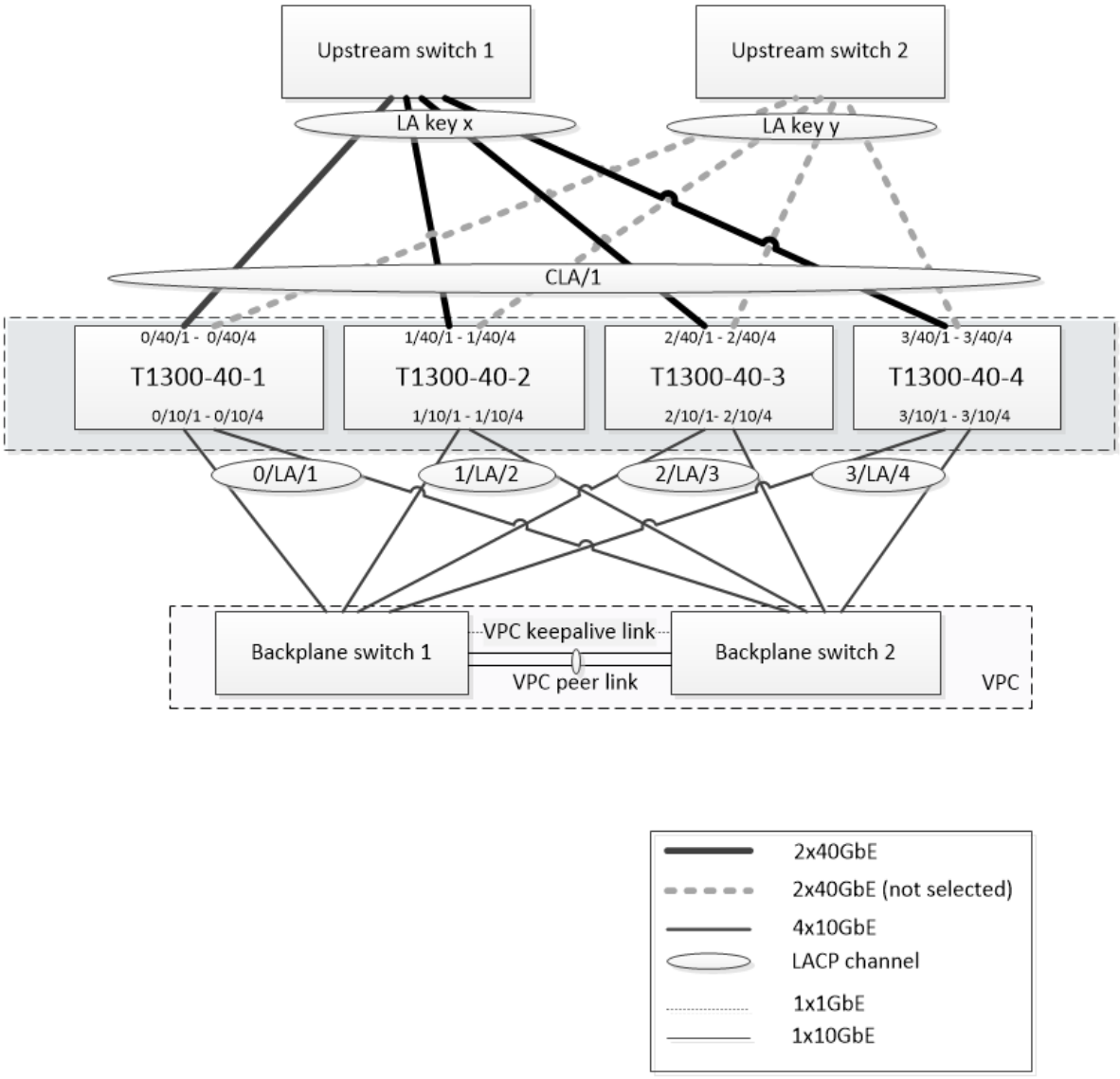


図 1 に示すセットアップには、次のプロパティがあります。

1. すべてのクラスターノードは同じネットワーク (L2 クラスターとも呼ばれます) に属します。
2. データプレーンとバックプレーンのトラフィックは異なるスイッチによって処理されます。
3. Gi-LAN のスループットが 200 Gbps で、T1300-40G アプライアンスが 80 Gbps のスループットを維持できると仮定すると、3 台の T1300-40G アプライアンスが必要です。1 つのクラスターノードに障害が発生した場合に冗長性を確保するために、合計 4 つのアプライアンスを導入しています。
4. 各ノードは最大 67 Gbps のトラフィック（通常の動作状態では 50 Gbps、単一クラスタノード障害の場合は 67 Gbps）を受信するため、アップストリームスイッチへの 2 x 40 Gbps 接続が必要です。スイッチに障害が発生した場合に冗長性を確保するために、アップストリームスイッチを 2、3 台配置し、接続数を 2 倍にしています。
5. クラスタリンクアグリゲーション (CLAG) は、クラスタノード全体にトラフィックを分散するために使用されます。1 つの CLAG がクライアントとサーバの両方のトラフィックを処理します。CLAG ではリンク冗長性が

有効になっているため、一度に 1 つの「サブチャネル」だけが選択され、トラフィックを処理します。一部のリンクに障害が発生したり、スループットが指定されたしきい値を下回ったりすると、他のサブチャネルが選択されます。

6. アップストリームスイッチは対称的なポートチャネルのロードバランシング（たとえば、Cisco IOS 7.0 (8) N1 (1) の source-dest-IP 専用アルゴリズム）を実行して、転送トラフィックフローと逆方向トラフィックフローが同じクラスタノードで処理されるようにします。このプロパティは、TCP のパフォーマンスを低下させるパケットの順序変更が不要になるので望ましいものです。
7. データトラフィックの 50% はバックプレーンに送られると予想されます。つまり、各ノードは最大 34 Gbps (通常の動作状態では 25 Gbps、単一クラスタノードに障害が発生した場合は 34 Gbps)、他のクラスタノードに伝送されます。したがって、各ノードにはバックプレーンスイッチへの少なくとも 4x10G 接続が必要です。スイッチに障害が発生した場合に冗長性を確保するために、バックプレーンスイッチを 2、3 台配置し、接続数を 2 倍にしています。リンク冗長性は現在バックプレーンではサポートされていないため、スイッチレベルの冗長性を実現するには Cisco VPC または同等のテクノロジーが必要です。
8. ステアリングパケットの MTU サイズは 1578 バイトなので、バックプレーンスイッチは 1500 バイトを超える MTU をサポートする必要があります。

注**:** 図 1 に示す設計は、T1120 および T1310 アプライアンスにも適用できます。T1310 には 10GbE ポートがないため、バックプレーン接続には 40GbE インターフェイスを使用します。

注**:** このドキュメントでは例として Cisco VPC を使用していますが、Cisco 以外のスイッチを使用する場合は、Juniper の MLAG などの同等の代替ソリューションを使用できます。

注**:** CLAG の代わりに ECMP などの他のトポロジも可能ですが、この特定のユースケースでは現在サポートされていません。

NetScaler T1000 クラスターでの **TCP** 最適化の構成

物理インストール、物理接続、ソフトウェアのインストール、およびライセンスが完了したら、実際のクラスタ構成に進むことができます。以下に説明する構成は、図 1 に示すクラスタに適用されます。

注：クラスター構成の詳細については、「NetScaler ADC クラスターの設定」を参照してください。

図 1 の 4 つの T1300 ノードには、次の NSIP アドレスがあるとします。

NSIP アドレスを持つ **4** つの **T1300** ノード：

```
T1300-40-1: 10.102.29.60
T1300-40-2: 10.102.29.70
T1300-40-3: 10.102.29.80
T1300-40-4: 10.102.29.90
```

クラスタは、クラスタ IP (CLIP) アドレスを使用して管理されます。このアドレスは 10.78.16.61 であると想定されます。

クラスターのセットアップ

図 1 に示すクラスタの構成を開始するには、クラスタに追加する最初のアプライアンス (たとえば、T1300-40-1) にログオンし、次の操作を行います。

1. コマンドプロンプトで、次のコマンドを入力します。

   コマンド:

```
> add cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE
> add ns ip 10.102.29.61 255.255.255.255 -type clip
> enable cluster instance 1
> save ns config
> reboot  - warm
```

2. アプライアンスが再起動したら、クラスタ IP（CLIP）アドレスに接続し、残りのノードをクラスタに追加します。

   コマンド:

```
> add cluster node 1 10.102.29.70 -state ACTIVE
> add cluster node 2 10.102.29.80 -state ACTIVE
> add cluster node 3 10.102.29.90  - state ACTIVE
> save ns config
```

3. 新しく追加された各ノードの NSIP アドレスに接続し、クラスタに参加します。

   コマンド:

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot  - warm
```

4. ノードが再起動したら、バックプレーン構成に進みます。クラスタ IP アドレスで次のコマンドを入力して、各クラスタノードのバックプレーンリンクの LACP チャネルを作成します。

   コマンド:

```
> set interface 0/10/[1-8]  - lacpkey 1  - lacpmode ACTIVE
> set interface 1/10/[1-8]  - lacpkey 2  - lacpmode ACTIVE
> set interface 2/10/[1-8]  - lacpkey 3  - lacpmode ACTIVE
> set interface 3/10/[1-8]  - lacpkey 4  - lacpmode ACTIVE
```

5. 同様に、バックプレーンスイッチでダイナミック LA と VPC を設定します。バックプレーンスイッチインターフェイスの MTU が 1578 バイト以上であることを確認します。

6. チャネルが動作していることを確認します。

   コマンド:

```
> show channel 0/LA/1
```



```
> show channel 1/LA/2
> show channel 2/LA/3
> show channel 3/LA/4
```

7. クラスタノードバックプレーンインターフェイスを設定します。

   コマンド：

```
> set cluster node 0 -backplane 0/LA/1
> set cluster node 1 -backplane 1/LA/2
> set cluster node 2 -backplane 2/LA/3
> set cluster node 3  - backplane 3/LA/4
```

8. クラスタのステータスを確認し、クラスタが動作していることを確認します。

```
> show cluster instance
> show cluster node
```

クラスターの設定の詳細については、「NetScaler ADC クラスターの設定」を参照してください。

クラスタノード間でのトラフィックの分散

NetScaler クラスターを形成したら、クラスターリンクアグリゲーション（CLAG）を展開してトラフィックをクラスターノードに分散します。1 つの CLAG リンクでクライアントトラフィックとサーバトラフィックの両方を処理できます。

クラスタ IP アドレスで、次のコマンドを実行して、図 1 に示すクラスタリンクアグリゲーション (CLAG) グループを作成します。

コマンド：

```
> set interface 0/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 1/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 2/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 3/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
```

外部スイッチでダイナミックリンク集約を設定します。

次に、次のようにリンク冗長性を有効にします。

コード**:**

```
> set channel CLA/1 -linkRedundancy ON -lrMinThroughput 240000
```

最後に、次のように入力してチャネルのステータスを確認します。

コマンド：

```
> show channel CLA/1
```


チャネルは UP で、実際のスループットは 320000 である必要があります。

クラスタリンクアグリゲーションの詳細については、以下のトピックを参照してください。

- 動的クラスタリンク集約
- LACP を使用したクラスタ内のリンク冗長性。

MAC ベースフォワーディング（MBF）を使用するため、次のようにリンクセットを設定し、CLAG グループにバインドします。

コマンド：

```
> add linkset LS/1
> bind linkset LS/1 -ifnum CLA/1
```

リンクセットの詳細については、以下のトピックを参照してください。

- リンクセットの設定
- リンクセットでのクラスタ LA チャネルの使用

VLAN と IP アドレスの設定

ストライプ IP 構成を使用します。これは、IP アドレスがすべてのノードでアクティブであることを意味します（デフォルト設定）。このトピックの詳細については、「ストライプ、部分的にストライプ、およびスポッティングされた構成」を参照してください。

1. 入力および出力 SNIP を追加します。

   コマンド：

   ```
   > add ns ip 172.16.30.254 255.255.255.0  -type SNIP
   > add ns ip 172.16.31.254 255.255.255.0  -type SNIP
   > add ns ip6 fd00:172:16:30::254/112  -type SNIP
   > add ns ip6 fd00:172:16:31::254/112  -type SNIP
   ```

2. 対応する入力 VLAN と出力 VLAN を追加します。

   コマンド：

   ```
   > add vlan 30 -aliasName wireless
   > add vlan 31 -aliasName internet
   ```

3. IP およびリンクセットを使用して VLAN をバインドします。

   コマンド：

   ```
   > bind vlan 31 -ifnum LS/1 -tagged
   > bind vlan 30 -ifnum LS/1 -tagged
   > bind vlan 30 -IPAddress 172.16.30.254 255.255.255.0
   > bind vlan 31 -IPAddress 172.16.31.254 255.255.255.0
   ```



```
> bind vlan 30 -IPAddress fd00:172:16:30::254/112
> bind vlan 31 -IPAddress fd00:172:16:31::254/112
```

必要に応じて、さらに多くの入力 VLAN と出力 VLAN を追加できます。

TCP 最適化の設定

この時点で、クラスタ固有のコマンドをすべて適用しました。設定を完了するには、TCP 最適化の設定で説明されている手順に従います。

ダイナミックルーティングの設定

NetScaler クラスターは、お客様のネットワークの動的ルーティング環境に統合できます。以下は、BGP ルーティングプロトコルを使用した動的ルーティング設定の例です（OSPF もサポートされています）。

1. CLIP アドレスから、入力 IP アドレスと出力 IP アドレスで BGP と動的ルーティングを有効にします。

   コマンド：

   ```
   > enable ns feature bgp
   > set ns ip 172.16.30.254  -dynamicRouting ENABLED
   > set ns ip 172.16.31.254  -dynamicRouting ENABLED
   ```

2. vtysh を開き、出力側に BGP を設定します。

   コード**:**

   ```
   > shell
   root@ns# vtysh
   ns# configure terminal
   ns(config)# router bgp 65531
   ns(config-router)# network 10.0.0.0/24
   ns(config-router)# neighbor 172.16.31.100 remote-as 65530
   ns(config-router)# neighbor 172.16.31.100 update-source
       172.16.31.254
   ns(config-router)# exit
   ns(config)# ns route-install propagate
   ns(config)# ns route-install default
   ns(config)# ns route-install bgp
   ns(config)# exit
   ```

3. デフォルトルートを NetScaler ADC クラスタにアドバタイズするように、出力側 BGP ピアを設定します。次に例を示します：

   コマンド：

   ```
   router bgp 65530
    bgp router-id 172.16.31.100
    network 0.0.0.0/0
   ```



```
4   neighbor 172.16.31.254 remote-as 65531
```

4. 入力側を設定するには、同様の手順に従います。
5. vtysh から、次のように入力して、構成がすべてのクラスタノードに伝播されることを確認します。

   コマンド:

```
1 ns# show running-config
```

6. 最後に、各クラスタノードの NSIP アドレスにログオンし、BGP ピアからアドバタイズされたルートを確認します。

   コマンド:

```
1 > show route | grep BGP
```

TCP Nile を使用した TCP パフォーマンスの最適化

August 15, 2023

TCP は、データ伝送におけるネットワークの輻輳を回避するために、以下の最適化手法と輻輳制御戦略（またはアルゴリズム）を使用します。

渋滞制御戦略

Transmission Control Protocol（TCP）は、インターネット接続の確立と管理、転送エラーの処理、Web アプリケーションとクライアントデバイスの円滑な接続に長い間使用されてきました。しかし、パケットロスはネットワークの輻輳だけに依存するわけではなく、輻輳が必ずしもパケットロスを引き起こすわけではないため、ネットワークトラフィックの制御はより困難になっています。したがって、輻輳を測定するには、TCP アルゴリズムはパケット損失と帯域幅の両方に焦点を当てる必要があります。

ナイルアルゴリズム

Citrix システムズは、LTE、LTE Advanced、3G などの高速ネットワーク向けに設計された TCP 最適化アルゴリズムである NILE という新しい輻輳制御アルゴリズムを開発しました。Nile は、フェーディング、ランダムまたは輻輳による損失、リンク層再送信、キャリアアグリゲーションに起因する固有の課題に対応します。

NILE アルゴリズム:

- ラウンドトリップ時間の測定値に基づいてキュー遅延を推定します。

- 測定されたキュー遅延に反比例する輻輳ウィンドウ増加関数を使用します。この方法では、標準の TCP 方式よりもネットワークの輻輳ポイントに近づくのが遅くなり、輻輳時のパケット損失が減少します。
- 推定キュー遅延を使用して、ネットワーク上のランダム損失と輻輳に基づく損失を区別できます。

通信サービスプロバイダーは、自社の TCP インフラストラクチャで NILE アルゴリズムを使用して次のことを行えます。

- モバイルネットワークと長距離ネットワークの最適化—NILE アルゴリズムは、標準の TCP と比較して高いスループットを実現します。この機能は、モバイルネットワークや長距離ネットワークにとって特に重要です。
- アプリケーションで認識される遅延を低減し、サブスクライバーエクスペリエンスを向上させる：ナイルアルゴリズムは、パケット損失情報を使用して送信ウィンドウサイズを大きくすべきか小さくすべきかを判断し、キューイング遅延情報を使用して増減のサイズを決定します。この転送ウィンドウサイズの動的設定により、ネットワーク上のアプリケーション遅延が減少します。

コマンドラインインターフェイスを使用して **NILE** サポートを設定するには

コマンドプロンプトで、次のように入力します。

```
set ns tcpProfile <name> [-flavor NILE]
<!--NeedCopy-->
```

設定ユーティリティーを使った **NILE** サポートの設定

1. [システム] > [プロファイル] > [**TCP** プロファイル**]** に移動し、**[**TCP**** プロファイル] をクリックします。
2. 「**TCP** フレーバー」ドロップダウンリストから「**NILE**」を選択します。

例**:**

```
set ns tcpProfile tcpprofile1 -flavor NILE
<!--NeedCopy-->
```

比例レート回復 **(PRR)** アルゴリズム

TCP 高速回復メカニズムは、パケット損失によるウェブの遅延を低減します。新しい比例レート回復 (PRR) アルゴリズムは、損失回復中に TCP データを評価する高速回復アルゴリズムです。輻輳制御アルゴリズムで選択したターゲットウィンドウに適した割合を使用して、Rate-Halving を模したパターンになっています。これにより、ウィンドウの調整が最小限に抑えられ、リカバリ終了時の実際のウィンドウサイズは Slow-Start のしきい値 (ssthresh) に近い値になります。

TCP ファストオープン **(TFO)**

TCP Fast Open（TFO）は、TCP の最初のハンドシェイク中にクライアントとサーバー間で迅速かつ安全なデータ交換を可能にする TCP メカニズムです。この機能は、NetScaler アプライアンスの仮想サーバーにバインドされた TCP プロファイルの TCP オプションとして使用できます。TFO は、NetScaler アプライアンスが生成する TCP Fast Open Cookie（セキュリティクッキー）を使用して、仮想サーバーへの TFO 接続を開始するクライアントを検証および認証します。TFO メカニズムを使用すると、1 回のフルラウンドトリップに必要な時間だけアプリケーションのネットワーク遅延を減らすことができます。これにより、短い TCP 転送で発生する遅延を大幅に減らすことができます。

TFO の仕組み

クライアントが TFO 接続を確立しようとすると、最初の SYN セグメントに TCP Fast Open Cookie が含まれ、それ自体が認証されます。認証が成功すると、NetScaler アプライアンス上の仮想サーバーは、スリーウェイハンドシェイクの最後の ACK セグメントを受信していなくても、SYN-ACK セグメントにデータを含めることができます。これにより、データを交換する前に三者間のハンドシェイクを必要とする通常の TCP 接続と比較して、最大 1 回の往復を節約できます。

クライアントとバックエンドサーバーは、次の手順を実行して TFO 接続を確立し、最初の TCP ハンドシェイク中にデータを安全に交換します。

1. クライアント自体を認証するための TCP ファストオープンクッキーがない場合、クライアントは SYN パケットでファストオープンクッキーリクエストを NetScaler アプライアンス上の仮想サーバーに送信します。
2. 仮想サーバにバインドされた TCP プロファイルで TFO オプションが有効になっている場合、アプライアンスは（クライアントの IP アドレスを秘密鍵で暗号化することにより）Cookie を生成し、生成された Fast Open Cookie を TCP オプションフィールドに含む SYN-ACK でクライアントに応答します。
3. クライアントは、アプライアンス上の同じ仮想サーバーへの今後の TFO 接続に備えて Cookie をキャッシュします。
4. クライアントが同じ仮想サーバーへの TFO 接続を確立しようとすると、キャッシュされた Fast Open Cookie（TCP オプションとして）を含む SYN を HTTP データとともに送信します。
5. NetScaler アプライアンスは Cookie を検証し、認証が成功すると、サーバーは SYN パケット内のデータを受け入れ、SYN-ACK、TFO クッキー、HTTP レスポンスでイベントを確認します。

注**:** クライアント認証が失敗した場合、サーバーはデータをドロップし、セッションタイムアウトを示す SYN のみでイベントを確認します。

1. サーバー側では、サービスにバインドされた TCP プロファイルで TFO オプションが有効になっている場合、NetScaler アプライアンスは、接続しようとしているサービスに TCP Fast Open Cookie が存在するかどうかを判断します。
2. TCP Fast Open Cookie が存在しない場合、アプライアンスは SYN パケットで Cookie リクエストを送信します。

3. バックエンドサーバーが Cookie を送信すると、アプライアンスはその Cookie をサーバー情報キャッシュに保存します。
4. アプライアンスに特定の宛先 IP ペアの Cookie が既にある場合は、古い Cookie が新しい Cookie に置き換えられます。
5. 仮想サーバーが同じ SNIP アドレスを使用して同じバックエンドサーバーに再接続しようとしたときに Cookie がサーバー情報キャッシュに存在する場合、アプライアンスは SYN パケット内のデータを Cookie と結合し、バックエンドサーバーに送信します。
6. バックエンドサーバーは、データと SYN の両方でイベントを確認します。

注：サーバーが SYN セグメントのみでイベントを確認した場合、NetScaler アプライアンスは元のパケットから SYN セグメントと TCP オプションを削除した直後にデータパケットを再送信します。

TCP ファストオープンの設定

TCP Fast Open（TFO）機能を使用するには、関連する TCP プロファイルで TCP Fast Open オプションを有効にし、TFO Cookie Timeout パラメータをそのプロファイルのセキュリティ要件に適した値に設定します。

コマンドラインを使用して **TFO** を有効または無効にするには　コマンドプロンプトで、次のコマンドのいずれかを入力して、新規または既存のプロファイルの TFO を有効または無効にします。

注**:** デフォルト値は DISABLED です。

```
add tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
set tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
unset tcpprofile <TCP Profile Name> - tcpFastOpen
<!--NeedCopy-->
```

例:

add tcpprofile Profile1 -tcpFastOpen
Set tcpprofile Profile1 -tcpFastOpen Enabled
unset tcpprofile Profile1 -tcpFastOpen

コマンドラインインターフェイスを使用して **TCP Fast Open Cookie** のタイムアウト値を設定するには

コマンドプロンプトで入力します。

```
set tcpparam  - tcpfastOpenCookieTimeout <Timeout Value>
<!--NeedCopy-->
```

例**:**

```
set tcpprofile  - tcpfastOpenCookieTimeout 30secs
<!--NeedCopy-->
```


GUI を使用して **TCP** ファストオープンを設定するには

1. [構成] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。
2.「**TCP** プロファイルの設定」ページで、「**TCPFast Open** 」チェックボックスを選択します。
3. [**OK]**、[完了**]** の順にクリックします。

GUI を使用して **TCP** ファストクッキーのタイムアウト値を設定するには

[構成] > [システム] > [設定**] > [TCP** パラメータの変更] に移動し、次に [**TCP** パラメータの設定] ページに移動して TCP Fast Open Cookie のタイムアウト値を設定します。

TCP ハイスタート

新しい TCP プロファイルパラメーター hystart により、Hystart アルゴリズムが有効になります。Hystart アルゴリズムは、終了する安全なポイント (ssthresh) を動的に決定するスロースタートアルゴリズムです。これにより、大量のパケットロスを発生させることなく、輻輳回避への移行が可能になります。この新しいパラメータはデフォルトでは無効になっています。

輻輳が検出されると、Hystart は輻輳回避フェーズに入ります。これを有効にすると、パケット損失の多い高速ネットワークでのスループットが向上します。このアルゴリズムは、トランザクションの処理中に最大帯域幅に近い状態を維持するのに役立ちます。そのため、スループットを向上させることができます。

TCP ハイスタートの設定

Hystart 機能を使用するには、関連する TCP プロファイルで Cubic Hystart オプションを有効にします。

コマンドラインインターフェイス **(CLI)** を使用して **Hystart** を設定するには

コマンドプロンプトで、次のコマンドのいずれかを入力して、新規または既存の TCP プロファイルで Hystart を有効または無効にします。

```
add tcpprofile <profileName> -hystart ENABLED
set tcpprofile <profileName> -hystart ENABLED
unset tcprofile <profileName> -hystart
<!--NeedCopy-->
```

例:

```
add tcpprofile Profile1 - tcpFastOpen
Set tcpprofile Profile1 - tcpFastOpen Enabled
unset tcpprofile Profile1 - tcpFastOpen
<!--NeedCopy-->
```


GUI を使用して **Hystart** サポートを設定するには

1. [構成] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。
2.「**TCP** プロファイルの設定」ページで、「**Cubic Hystart** 」チェックボックスを選択します。
3. [**OK]**、[完了**]** の順にクリックします。

最適化手法

TCP では、以下の最適化手法と方法を使用してフロー制御を最適化します。

ポリシーベースの **TCP** プロファイル選択

今日のネットワークトラフィックは、かつてないほど多様で帯域幅を大量に消費しています。トラフィックの増加に伴い、サービス品質 (QoS) が TCP のパフォーマンスに与える影響は大きくなります。QoS を強化するために、ネットワークトラフィックのクラスごとに異なる TCP プロファイルを使用して AppQoE ポリシーを設定できるようになりました。AppQoE ポリシーは、仮想サーバーのトラフィックを分類して、3G、4G、LAN、WAN などの特定のタイプのトラフィックに最適化された TCP プロファイルを関連付けます。

この機能を使用するには、TCP プロファイルごとにポリシーアクションを作成し、AppQoE ポリシーにアクションを関連付け、負荷分散仮想サーバーにポリシーをバインドします。

ポリシーベースの **TCP** プロファイル選択の設定

ポリシーベースの TCP プロファイル選択の設定は、次のタスクで構成されます。

- AppQoE を有効にする。TCP プロファイル機能を設定する前に、AppQoE 機能を有効にする必要があります。
- AppQoE アクションを追加します。AppQoE 機能を有効にしたら、TCP プロファイルを使用して AppQoE アクションを設定します。
- AppQoE ベースの TCP プロファイル選択の設定さまざまなクラスのトラフィックに TCP プロファイル選択を実装するには、NetScaler アプライアンスが接続を識別し、正しい AppQoE アクションを各ポリシーにバインドできる AppQoE ポリシーを構成する必要があります。
- AppQoE ポリシーを仮想サーバにバインドします。AppQoE ポリシーを構成したら、それらを 1 つ以上の負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーにバインドする必要があります。

コマンドラインインターフェイスを使用した構成

コマンドラインインターフェイスを使用して AppQoE を有効にするには:

コマンドプロンプトで次のコマンドを入力して機能を有効にし、有効になっていることを確認します。


```
enable ns feature appqoe

show ns feature
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **AppQoE** アクションを作成する際に **TCP** プロファイルをバインドするには　コマンドプロンプトで、tcpprofiletobind オプションを指定して次の AppQoE アクションコマンドを入力します。

TCP プロファイルのバインディング**:**

```
add appqoe action <name> [-priority <priority>] [-respondWith ( ACS |
    NS ) [<CustomFile>] [-altContentSvcName <string>] [-altContentPath <
    string>] [-maxConn <positive_integer>] [-delay <usecs>]] [-polqDepth
     <positive_integer>] [-priqDepth <positive_integer>] [-
    dosTrigExpression <expression>] [-dosAction ( SimpleResponse |
    HICResponse )] [-tcpprofiletobind <string>]

show appqoe action
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **AppQoE** ポリシーを構成するには

コマンドプロンプトで入力します。

```
add appqoe policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **AppQoE** ポリシーを負荷分散、キャッシュリダイレクト、またはコンテンツスイッチング仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind cs vserver cs1 -policyName <appqoe_policy_name> -priority <
    priority>
bind lb vserver <name> - policyName <appqoe_policy_name> -priority <
    priority>
bind cr vserver <name> -policyName <appqoe_policy_name> -priority <
    priority>
<!--NeedCopy-->
```

例**:**

```
add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -nagle
    ENABLED -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 500 -
    slowStartIncr 1 -bufferSize 4194304 -flavor BIC -KA ENABLED -
```



```
    sendBuffsize 4194304 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -dsack enabled -frto ENABLED -maxcwnd 4000000 -fack ENABLED
    -tcpmode ENDPOINT

add appqoe action appact1 -priority HIGH -tcpprofile tcp1

add appqoe policy apppol1 -rule "client.ip.src.eq(10.102.71.31)" -
    action appact1

bind lb vserver lb2 -policyName apppol1 -priority 1 -
    gotoPriorityExpression END -type REQUEST

bind cs vserver cs1 -policyName apppol1 -priority 1 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

GUI を使用したポリシーベースの **TCP** プロファイリングの設定

GUI を使用して AppQoE を有効にするには

1. [システム] > [設定] に移動します。
2. 詳細ウィンドウで、「拡張機能の設定」をクリックします。
3. 「拡張機能の設定」ダイアログで、「**AppQoE** 」チェックボックスを選択します。
4. [**OK**] をクリックします。

GUI を使用して **AppQoE** ポリシーを設定するには

1. **[**** アプリエキスパート**] > [**AppQoE****] > [アクション] に移動します。****
2. 詳細ウィンドウで、次のいずれかの操作を行います。
3. 新しいアクションを作成するには、[追加] をクリックします。
4. 既存のアクションを変更するには、アクションを選択し、[編集] をクリックします。
5. 「**AppQoE** アクションの作成」または「**AppQoE** アクションの設定」画面で、パラメータの値を入力または選択します。ダイアログボックスの内容は、「AppQoE アクションを構成するためのパラメーター」で説明されているパラメーターに次のように対応します (アスタリスクは必須パラメーターを示します)。

   a) 名前—名前
   b) アクションタイプ—次の式で応答
   c) 優先度—優先度
   d) ポリシーキューの深さ: POLQ の深さ
   e) キューの深さ—PRIQ の深さ
   f) DOS アクション—ディスコクション

6. [作成] をクリックします。

GUI を使用して **AppQoE** ポリシーをバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、サーバーを選択して [編集] をクリックします。
2.「ポリシー」セクションで、(+) をクリックして AppQoE ポリシーをバインドします。
3.「ポリシー」スライダーで、次の操作を行います。

    a) ドロップダウンリストから AppQoE としてポリシータイプを選択します。
    b) ドロップダウンリストからトラフィックタイプを選択します。

4.「ポリシーバインディング」セクションで、次の操作を行います。

    a)「新規」をクリックして、新しい AppQoE ポリシーを作成します。
    b) 既存のポリシーをクリックして、ドロップダウンリストから AppQoE ポリシーを選択します。

5. バインディングの優先順位を設定し、ポリシーを仮想サーバにバインドをクリックします。
6.［完了］をクリックします。

SACK ブロック生成

1 つのデータウィンドウで複数のパケットが失われると、TCP のパフォーマンスが低下します。このようなシナリオでは、選択的受信確認（SACK）メカニズムと選択的繰り返し再送信ポリシーを組み合わせることで、この制限を克服できます。順不同のパケットが受信されるたびに、SACK ブロックを生成する必要があります。

順序が狂ったパケットが再構成キューブロックに収まる場合は、そのブロックにパケット情報を挿入し、完全なブロック情報を SACK-0 に設定します。順序が狂ったパケットが再構成ブロックに収まらない場合は、そのパケットを SACK-0 として送信し、前の SACK ブロックを繰り返します。順序が合っていないパケットが重複していて、パケット情報が SACK-0 に設定されている場合は、ブロックを D-SACK します。

注**:** 確認済みのパケット、または既に受信された順序の悪いパケットの場合、そのパケットは D-SACK と見なされます。

クライアントリネギング

NetScaler アプライアンスは、SACK ベースのリカバリ中のクライアント更新を処理できます。

PCB 上の **end_point** をマーキングするためのメモリチェックでは、使用可能なメモリの合計が考慮されない

NetScaler アプライアンスでは、使用可能な合計メモリを使用せずに、メモリ使用量のしきい値を 75% に設定すると、新しい TCP 接続が TCP 最適化をバイパスします。

SACK ブロックの欠落による不必要な再送信

非エンドポイントモードで DUPACKS を送信するときに、順序が狂ったパケットのいくつかで SACK ブロックが欠落していると、サーバからの再送信がさらにトリガーされます。

接続数の **SNMP** が過負荷のため最適化をバイパスしました

過負荷が原因で TCP 最適化がバイパスされた接続数を追跡するために、次の SNMP ID が NetScaler アプライアンスに追加されました。

1. 1.3.6.1.4.1.5951.4.1.46.13 (TCP 最適化が有効)。TCP 最適化で有効になっている接続の総数を追跡します。
2. 1.3.6.1.4.1.5951.4.1.46.132 (TCP 最適化バイパス)。接続の総数を追跡するには、TCP 最適化をバイパスしました。

ダイナミック受信バッファ

TCP パフォーマンスを最大化するために、NetScaler ADC アプライアンスは TCP 受信バッファサイズを動的に調整できるようになりました。

トラブルシューティングガイドライン

August 15, 2023

テクニカルサポート

すべてのトラブルシューティングとエスカレーションクエリには、現在の構成、インストールされているファームウェアのバージョン、ログファイル、未処理のコアなどをキャプチャする最新の NetScaler テクニカルサポートバンドルが必要です。

例**:**

```
show techsupport

showtechsupport data collector tool - $Revision: #5 $!
...
<!--NeedCopy-->
```

すべてのデータは以下で収集されます


```
...
Archiving all the data into "/var/tmp/support/collector_P_192
    .168.121.117_18Jun2015_09_53.tar.gz" ....
Created a symbolic link for the archive with /var/tmp/support/support.
    tgz
/var/tmp/support/support.tgz  ---- points to ---> /var/tmp/support/
    collector_P_192.168.121.117_18Jun2015_09_53.tar.gz
<!--NeedCopy-->
```

テクニカルサポートバンドルが生成されたら、SCP を使用してコピーできます。

トレース

NetScaler TCP 最適化の問題は通常、適切にトラブルシューティングするために NetScaler トレースを必要とします。ただし、同じセルで、同じ時間帯に、同じユーザー機器やアプリケーションを使用して、同様の条件でトレースをキャプチャするようにしてください。

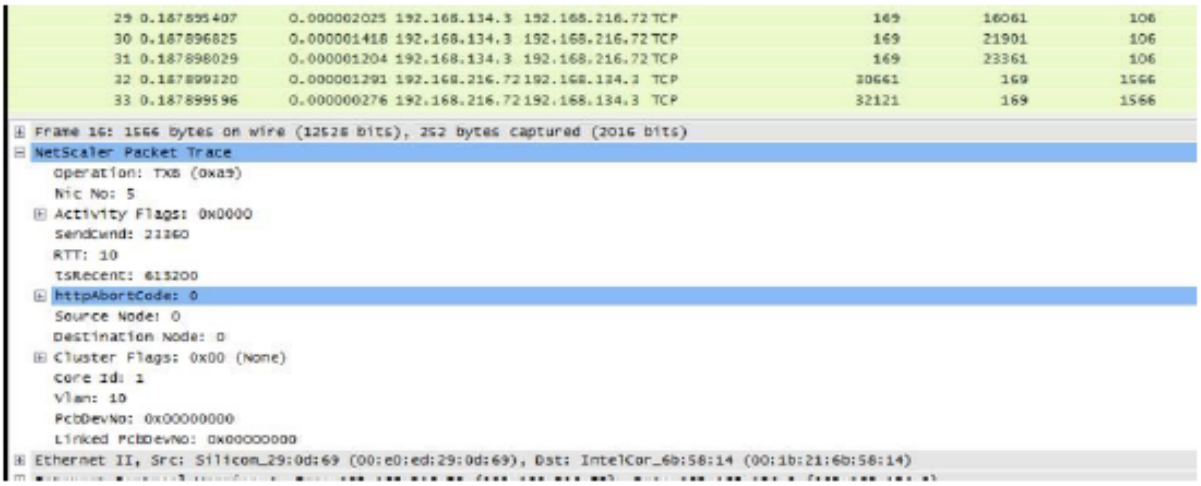
nstrace の起動コマンドと nstrace 停止コマンドを使用してトレースをキャプチャできます。

- トレース上の無関係で不要なパケットがキャプチャされないように、適切なフィルターを使用することを強くお勧めします。たとえば、start nstrace-filter ‘IP == 10.20.30.40’ を使用すると、ユーザー機器の IP アドレスである IP アドレス 10.20.30.40 との間で送受信されるパケットのみをキャプチャできます。
- -tcpdump オプションを使用すると、デバッグに必要な nstrace ヘッダーが削除されるので、使用しないでください。

トレース分析

NetScaler トレースがキャプチャされると、Wireshark 1.12 以降で表示される可能性があります。以下のスクリーンキャプチャに示すように、キャプチャされたトレースに適切な NetScaler Packet Trace ヘッダーが含まれていることを確認します。

以下の図のように、追加のデバッグヘッダーも表示されています。



接続テーブル

問題が TCP 最適化に関連していて、再現できる場合や進行中の場合は、プライマリ T1 ノードから問題が発生したときに接続テーブルも取得するのが最適です。

テーブルを取得するには、BSD シェルに切り替えて次のコマンドを実行する必要があります。

```
shell
...

nscli -U 127.0.0.1:nsroot:nsroot show connectiontable -detail full link
    > /var/tmp/contable.log
<!--NeedCopy-->
```

> 注
>
> コマンドはより長い時間実行され、その時点で管理 CPU に負荷がかかることがありますが（接続テーブルのエントリ数によって異なります）、サービスには影響しません。

よくある質問

August 15, 2023

タイムアウト

> 重要
>
> *nsapimgr* ノブを使用する前に、Citrix カスタマーサポートに相談してください。

NetScaler T1 仮想サーバーとサービスに設定できるさまざまなアイドル接続タイムアウトの一覧を以下に示します。仮想サーバーまたはサービスレベルでクライアントまたはサーバー接続に設定されたアイドルタイムアウトは、TCP ESTABISHLED 状態でアイドル状態の接続にのみ適用されます。

- 負荷分散仮想サーバーの CLTTimeout パラメータは、アプライアンスが接続を閉じる前にクライアントから負荷分散仮想サーバーへの接続がアイドル状態でなければならない時間を秒単位で指定します。
- Service SvrTimeout パラメータは、アプライアンスからサービスまたはサーバーへの接続がアイドル状態でなければならない時間を秒単位で指定します。この時間を超えると、アプライアンスは接続を閉じます。
- Service cltTimeout パラメータは、クライアントからサービスへの接続がアイドル状態でなければならない時間を秒単位で指定します。この時間を超えると、アプライアンスは接続を閉じます。

サービスが負荷分散仮想サーバーにバインドされている場合、負荷分散仮想サーバーの cltTimeout が優先され、サービスのサービス cltTimeout は無視されます。

負荷分散仮想サーバーにサービスがバインドされていない場合、サーバー側の接続にはグローバルアイドルタイムアウト、つまり TCPServer が使用されます。次のように構成できます。

コマンド:

```
set ns timeout  -tcpServer 9000
<!--NeedCopy-->
```

他の状態の接続では、タイムアウト値が異なります。

- ハーフオープン接続のアイドルタイムアウト:120 秒 (ハードコード値)
- TIME_WAIT 接続のアイドルタイムアウト:40 秒 (ハードコードされた値)
- ハーフクローズ接続のアイドルタイムアウト。デフォルトは 10 秒で、スニペットを使用して 1 秒から 600 秒の間で設定できます

コマンド:

```
 set ns timeout  -halfclose 10
<!--NeedCopy-->
```

ハーフクローズタイムアウトがトリガーされると、接続はゾンビ状態に移行します。ゾンビタイムアウトの期限が切れると、ゾンビクリーンアップが開始され、T1 はデフォルトで特定の接続に対してクライアント側とサーバ側の両方で RST を送信します。

- ゾンビタイムアウト: 非アクティブな TCP 接続をクリーンアップするためにゾンビクリーンアッププロセスを実行する必要がある間隔。デフォルトのタイムアウト値は 120 秒で、1 秒から 600 秒の間で設定できます。

コマンド:


```
set ns timeout  - zombie 120
<!--NeedCopy-->
```

最大セグメントサイズ表

NetScaler T1 アプライアンスは、システムメモリスタックでハーフオープン接続を維持する代わりに、SYN クッキーを使用して SYN フラッド攻撃を防御します。アプライアンスは TCP 接続を要求する各クライアントに Cookie を送信しますが、ハーフオープン接続の状態は維持されません。代わりに、アプライアンスは、最後の ACK パケットを受信したときだけ、または HTTP トラフィックの場合は HTTP 要求を受信したときにのみ接続にシステムメモリを割り当てます。これにより、SYN 攻撃が防止され、正規のクライアントとの通常の TCP 通信が中断されることなく継続できます。特定の機能はデフォルトで有効になっており、無効にするオプションはありません。

ただし、標準の SYN Cookie では接続の最大セグメントサイズ (MSS) 値を 8 つしか使用できないため、注意が必要です。接続 MMS があらかじめ定義されている値と一致しない場合、クライアント側とサーバー側の両方で、次に利用可能な低い値を選択します。

あらかじめ定義されている TCP 最大セグメントサイズ (MSS) 値は次のとおりです。これらは新しい nsapimgr ノブで設定できます。

1460	1440	1330	1220	956	536	384	128

新しい MSS テーブル:

- ジャンボフレームサポートを含める必要はありません。デフォルトでは、MSS テーブルにはジャンボフレーム用に 8 つの値が予約されていますが、テーブル設定を変更して標準のイーサネットサイズのフレームのみを含めることができます。
- 16 個の値が必要です
- 値は降順でなければなりません
- 最後の値として 128 を含める必要があります

新しい MSS テーブルが有効な場合、テーブルが保存され、SYN-Cookie のローテーション時に古い値が切り替わります。そうしないと、新しいテーブルはエラーを返します。変更は新しい接続に適用されますが、既存の接続では、接続が期限切れになるか終了するまで古い MSS テーブルが保持されます。

NetScaler アプライアンスの現在の MSS テーブルを表示するには、次のコマンドを入力します。

コマンド:

```
>shell

#nsapimgr -d mss_table
```


例:

```
#nsapimgr -d mss_table

MSS table

{
 9176,9156,8192,7168,6144,4196,3072,2048,1460,1440,1330,1212,956,536,384,128
      }


Done.
```

mss テーブルを変更するには、次のコマンドを入力します。

コマンド:

```
>shell

#nsapimgr -s mss_table=<16 comma seperated values>
```

例:

```
#nsapimgr -ys mss_table
   =9176,9156,8192,7168,6144,4196,3072,2048,1460,1400,1330,1212,956,536,384,128


# nsapimgr -d mss_table

MSS table

{
 9176,9156,8192,7168,6144,4196,3072,2048,1460,1400,1330,1212,956,536,384,128
      }


Done.
```

標準イーサネットサイズの値を使用する例を 次に示します。

例:

```
#nsapimgr -ys mss_table
   =1460,1440,1420,1400,1380,1360,1340,1320,1300,1280,1260,1212,956,536,384,128


# nsapimgr -d mss_table

MSS table

{
 1460,1440,1420,1400,1380,1360,1340,1320,1300,1280,1260,1212,956,536,384,128
      }

```



```
10
11  Done.
```

NetScaler ADC アプライアンスの再起動後もこの変更を永続的にするには、「/nsconfig/rc.netscaler」ファイルにコマンド#`nsapimgr -ys mss_table=<16 comma seperated values>` を含めます。「rc.netscaler」ファイルが存在しない場合は、「/nsconfig」フォルダーの下に作成し、コマンドを追加します。

メモリ過負荷保護

NetScaler Packet Processing Engine（PPE）は、その PPE が使用しているメモリが指定された上限値を超えると、TCP 最適化から接続をバイパスし始めます。*PPE* のメモリ使用率が *2.6 GB* を超えると、最適化による新しい接続のバイパスが開始されます。既存の接続 (以前に最適化が認められた接続) は引き続き最適化されます。このウォーターマーク値は意図的に選択されたものであり、チューニングにはお勧めしません。

> 注
>
> ウォーターマーク値を変更する正当な理由があると思われる場合は、カスタマーサポートに連絡してください。

Happy Eyeballs クライアントへのサポート

NetScaler アプライアンスが状態が不明な宛先の SYN を受信した場合、アプライアンスはまずサーバーの到達可能性をチェックし、次にクライアントを認識します。このプロービングメカニズムにより、デュアル IP スタックを持つクライアントは、デュアルスタックのインターネットサーバーの到達可能性を検出できます。クライアントは、IPv6 と IPv4 の両方のアクセスが可能であることを検出すると、より迅速に応答するサーバーへの接続を確立し、もう一方をリセットします。NetScaler アプライアンスの接続がリセットされると、対応するサーバー側の接続がリセットされます。

注：この機能には、NetScaler アプライアンスで無効化/有効化できるユーザー構成可能な TCP 設定はありません。

Happy Eyeballs のサポートについて詳しくは、RFC 6555 を参照してください。

NetScaler ビデオ最適化

August 15, 2023

> 警告**:**
>
> ビデオの最適化は、フォワードプロキシ電話会社ソリューションでのみサポートされます。他のタイプのユースケースでは動画最適化を有効にしないでください。ビデオ最適化は、管理パーティションとクラスタトポロ

> ジではサポートされていません。

NetScaler アプライアンスは、モバイルネットワーク上のビデオトラフィックの ABR ビデオトラフィックを最適化する最適化技術と機能を提供します。これにより、ユーザーエクスペリエンスが向上し、ネットワーク帯域幅全体の消費量が減少します。

このセクションには、以下のトピックが含まれています。

- Getting Started
- Licensing
- TCP 経由のビデオ最適化の構成
- UDP 経由のビデオ最適化の設定

Getting Started

August 15, 2023

メディアファイルがモバイルネットワーク上のトラフィック量を増加させており、より高速なネットワーク技術への移行により、暗号化されたビデオトラフィックの量が劇的に増加しています。従来のメディア配信技術（プログレッシブダウンロード）では、高い伝送速度で許容できる体感品質（QoE）を提供することができません。これにより、アダプティブビットレート (ABR) プロトコルが導入されました。利用可能なネットワーク帯域幅に合わせてストリーミングビットレートを調整し、ビデオを受信する端末の能力に合わせてストリーミング品質を制限できます。ただし、ABR プロトコルは、モバイルネットワークではインターネット経由のように機能しません。そのため、モバイル事業者は ABR トラフィックを最適化する必要があります。

NetScaler アプライアンスには、受信ビデオトラフィックを検出し、ABR ビデオを選択的に最適化する独自の機能があります。

NetScaler のビデオ最適化の仕組み

NetScaler アプライアンスは、暗号化された ABR トラフィック（Facebook のビデオトラフィックを含む）を TCP 経由で識別し、YouTube ABR トラフィックを QUIC 経由で識別して最適化できます。アプライアンスには次の機能があります。

1. HTTP 経由でプログレッシブダウンロード (PD) ビデオを検出します。
2. HTTP 経由で ABR ビデオを検出し、最適化します。
3. HTTPS 経由で ABR ビデオを検出し、最適化します。
4. QUIC 経由で YouTube ABR 動画を検出し、最適化します。

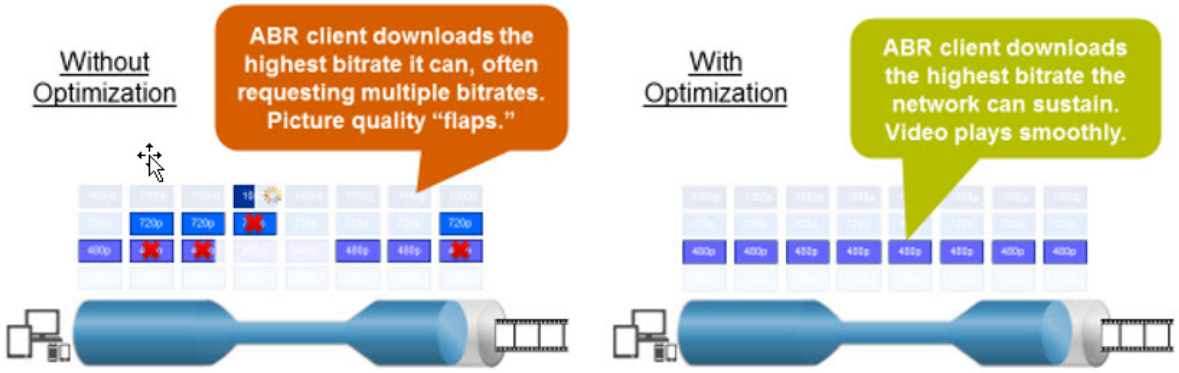
また、アプライアンスは以下のサポートドメインを使用して、TCP および QUIC プロトコル上のビデオトラフィックを検出します。

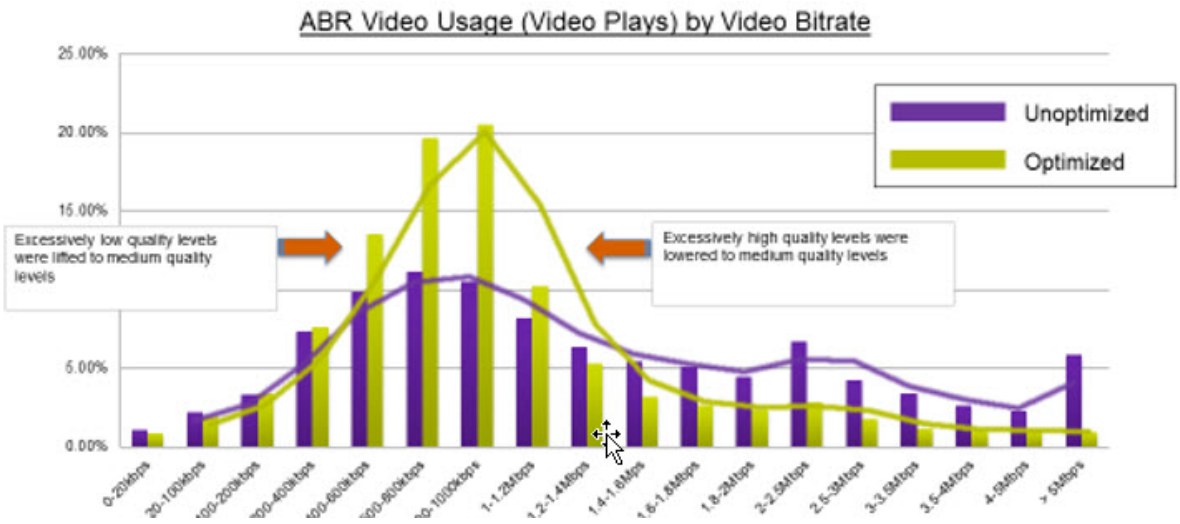
- TCP 経由の暗号化されていない ABR ビデオ。アプライアンスは、標準準拠のビデオストリーミング Web サイトをすべて検出します。アプライアンスは、応答ビデオペイロードヘッダー、URL、および HTTP ヘッダーを検査して ABR セッションを検出します。
- TCP 経由で暗号化された ABR ビデオ。アプライアンスは、ドメイン、SSL ヘッダー、トラフィックパターンに基づく汎用的でヒューリスティックなアルゴリズムを使用して ABR セッションを検出します。これにより、このアプライアンスには上位のビデオ Web サイトを 95% の精度で検出するサポートが組み込まれています。今後も新しいタイプのビデオもサポートしていきます。NetScaler には、特定の地域または国の暗号化率の高い ABR サイトを追加検証して、ネットワークのカバレッジを確保するプログラムもあります。
- QUIC 経由で暗号化された ABR ビデオ。アプライアンスは、YouTube などの QUIC ベースのビデオプロバイダーの ABR セッションを検出します。検出アルゴリズムは、QUIC ヘッダーとドメインを活用したヒューリスティックに基づいています。NetScaler は引き続き、QUIC を使用する新しいビデオサイトのサポートを追加します。

長所

ABR ビデオトラフィックを最適化すると、次のようなメリットがあります。

- ピーク時間における混雑時にネットワークを管理する。
- 動画再生の一貫性を向上させ動画の再生速度低下を抑える。
- 新しい動画サービスオファリング（Binge-on 動画サービスなど）を有効にする。
- 顧客が持続可能で最適な動画品質を選択できるようにする。
- サブスクライバーに一貫性のあるユーザーエクスペリエンスを提供する。





TCP 経由のビデオ最適化

NetScaler による TCP 経由の ABR トラフィックの最適化は次のように機能します。

1. アプライアンスが TCP 経由で受信する HTTP または HTTPS トラフィックは、対応する負荷分散仮想サーバーに送信されます。
2. 仮想サーバーにバインドされた組み込みの検出ポリシーと、他の独自の検出アルゴリズムを組み合わせてトラフィックを評価します。
3. ポリシーでは、組み込みのビデオ検出シグネチャのセットを使用してビデオタイプを検出します。トラフィックを照合するポリシーは、ビデオタイプを次のいずれかに分類するアクションを適用します。

    a) クリアテキスト PD
    b) クリアテキスト ABR
    c) 暗号化された ABR
    d) その他

4. 同じ仮想サーバーにバインドされた最適化ポリシーがトラフィックを評価し、トラフィックに適用する最適化ビットレートを決定します。
5. 最適化ビットレートは、トラフィックがクリアテキスト ABR または暗号化 ABR のどちらかの場合に適用されます。

モバイルサービスプロバイダーは、2G、3G、4G のモバイルトラフィックのダウンロード速度を設定することで、エクスペリエンス品質（QoE）を向上させることができます。これにより、動画の開始時間やバッファリングイベントが減少します。最適化により、ビデオセッションで消費されるネットワーク帯域幅の量も削減できます。

最適化技術には、動的バースト制御とランダムサンプリングが含まれます。

ダイナミック・バースト・コントロール

NetScaler ABR 最適化は、変化するネットワーク条件に動的に適応します。設定したペーシングレートの 1.3 倍の初期バーストレートを 15 秒間使用できます。初期バーストレートは、複数のセッションが同じ TCP 接続または TCP 接続グループを使用している場合でも、最適化された ABR ビデオセッションの開始時に適用されます。

アプライアンスは、ネットワークがサポートするビットレートが設定されたペーシングレートを下回った場合のリカバリーバーストもサポートします。たとえば、有効ビットレートが最初のバーストの 7 秒で低下し、15 秒で回復した場合、アプライアンスは次のバーストサイクルで損失を回復します。これにより、アプライアンスはすべての加入者のネットワーク帯域幅を動的に最適化し、画質がピクセルごとに一定に保たれるようにします。

注: 初期バースト中に回復バーストが発生した場合、ペーシングビットレートが最大回復バーストレートと初期バーストレートを超えないようにしてください（初期バースト係数の上に回復バースト係数を追加しないでください）。そうしないと、処理速度が速すぎて、メディアプレーヤーが高画質モードに移行する可能性があります。ただし、必要に応じて、初期バーストの期間を延長して未使用の帯域幅を補うことができます。

ランダムサンプリング

NetScaler アプライアンスはビデオ最適化による節約額を見積もるために、ランダムサンプリングを実装しています。この手法では、アプライアンスは検出されたビデオトラフィックのうち、設定可能なパーセンテージをランダムに選択します（ランダムサンプリングパラメータは 0 ～100 の範囲の整数なので、1% 未満は不可能です）。これらのランダムに選択され、最適化されていないトランザクション（およびセッション）は参照グループになり、（バイトサイズやタイマーフィールドなどの他の特性とともに）トランザクションログで識別されます。最適化されたセッションの特性も記録され、レポートエンジンは最適化されたグループと参照グループの統計を比較して、最適化による節約額（ABR 最適化による節約を含む）を見積もります。

UDP 経由のビデオ最適化

Google は QUIC と呼ばれる新しいトランスポートプロトコルを導入しました。グーグルの QUIC プロトコルは TCP+TLS+HTTP/2 とよく似ていて、UDP の上に実装されています。NetScaler ADC は、QUIC プロトコルでストリーミングされた YouTube の ABR 動画を検出し、ABR 動画の最適化を TCP 経由の ABR と同様の方法で適用できます。

ライセンス

August 15, 2023

ビデオ最適化機能は、基本的な CBM ライセンスと CBM Premium ライセンスを購入すれば Telco プラットフォームで動作し、他の NetScaler プラットフォームでは CNS Premium ライセンスを購入すれば機能します。ビデオ最適化機能を設定する前に、アプライアンスに適切なライセンスが必要です。

Telco プラットフォームのライセンスサポート:

- **CBM_TXXX_SERVER_Retail.lic**
- **CBM_TPRE_SERVER_Retail.lic**
- **CNS_WEBF_SSERVER_Retail.lic**

ここで、XXX はスループットです。たとえば、NetScaler T1000 のようになります。

他の NetScaler プラットフォームのライセンスサポート:

- **CNS_XXX_SERVER_PLT_Retail.lic**

ここで、XXX はスループットです。

プレミアムライセンスファイルをアップロードするには、以下の手順に従います。

1. NetScaler ADC アプライアンスに有効なライセンスファイルをインストールする必要があります。ライセンスは、予想される最大 Gi-LAN スループットと同じ数の Gbps を少なくともサポートする必要があります。

   ライセンスファイルは、以下のスクリーンキャプチャに示すように、SCP クライアントを介してアプライアンスの/nsconfig/license にコピーする必要があります。

   ```
   > shell ls /nsconfig/license/
   CNS_V3000_SERVER_PLT_Retail.lic ssl
   <!--NeedCopy-->
   ```

2. 次のスクリーンキャプチャに示すように、ウォームリスタートを実行して新しいライセンスを申請します。

   ```
   > reboot -warm
   Are you sure you want to restart NetScaler (Y/N)? [N]:y
    Done
   <!--NeedCopy-->
   ```

3. 再起動が完了したら、show license CLI を使用して、ライセンスが正しく適用されていることを確認します。

以下の例では、Premium エディションの Premium ライセンスが正常にインストールされています。

```
    > show license

                License status:

                                        Video Optimization: YES

                                                  ...

                                    Model Number ID: 110050

                                       License Type: Premium License
<!--NeedCopy-->
```


TCP 経由のビデオ最適化の構成

February 15, 2024

> 警告:
>
> ビデオの最適化の一環として、ビデオペーシング機能は廃止され、今後のリリースでは NetScaler ADC アプライアンスから削除されます。

TCP 経由でビデオトラフィックを最適化するには、まずビデオ最適化機能を有効にします。次に、アプライアンスは組み込みの検出ポリシーをアクティブにして、着信ビデオトラフィックを検出し、ビデオのタイプを識別します。各ビデオタイプのユーザ設定可能な最適化ポリシーは、トラフィックの最適化に必要な最適化ビットレートを指定します。

CLI を使用した TCP 経由のビデオ最適化の設定

NetScaler ADC アプライアンスでビデオの最適化を構成するには、次のタスクを実行します。

1. ビデオ最適化機能を有効にします。
2. HTTP および HTTPS トラフィック用の仮想サーバーを追加します。
3. すべての組み込み検出ポリシーを HTTP トラフィック用の負荷分散仮想サーバーにバインドします。
4. すべての組み込み検出ポリシーを、HTTPS トラフィック用の SSL ブリッジ負荷分散仮想サーバーにバインドします。
5. HTTP および HTTPS トラフィックに必要な最適化ポリシーを追加します。
6. HTTP トラフィックの負荷分散仮想サーバーに最適化ポリシーをバインドします。
7. HTTPS トラフィック用に最適化ポリシーを SSL ブリッジ負荷分散仮想サーバーにバインドします。

ビデオ最適化の有効化

NetScaler ADC アプライアンスでビデオトラフィックを検出、最適化、およびレポートする場合は、ビデオの最適化機能を有効にし、最適化をオンに設定する必要があります。この機能を有効にすると、組み込みの検出ポリシーを使用して着信ビデオトラフィックを識別できます。また、最適化ポリシーを設定して、暗号化された ABR トラフィックを最適化できます。ABR ビデオトラフィックを最適化するには、ダウンロードビットレート（ペーシングレートとも呼ばれる）を設定する必要があります。

また、負荷分散機能を有効にする必要があります。HTTPS トラフィックにビデオ最適化を使用する場合は、SSL 機能を有効にする必要があります。

ビデオ最適化機能を有効にするには　コマンドプロンプトで、次のコマンドを入力します。


```
enable ns feature VideoOptimization
<!--NeedCopy-->
```

> 注
>
> ビデオ最適化のパフォーマンスとビデオインサイトレポートを監視する場合は、AppFlow 機能を有効にしてから、NetScaler Application Delivery Management（ADM）のビデオ分析機能にアクセスする必要があります。詳細については、Video Insight ドキュメントを参照してください。

HTTP および **HTTPS** ビデオトラフィック用の仮想サーバの作成

NetScaler ADC アプライアンスは、さまざまな種類の着信ビデオトラフィックを検出および最適化するために、異なる仮想サーバーを使用します。アプライアンスは、TCP トラフィック用に次のタイプの仮想サーバーをサポートします。

- **HTTP** 負荷分散仮想サーバー。HTTP ビデオトラフィックを検出するために、アプライアンスは HTTP 負荷分散仮想サーバを使用します。アプライアンスがクライアントから受信する HTTP ビデオ要求を管理します。
- **SSL** ブリッジ負荷分散仮想サーバー。暗号化されたビデオトラフィックを検出するには、アプライアンスで SSL ブリッジ仮想サーバを設定する必要があります。

HTTP ビデオトラフィックを検出するための **HTTP** 負荷分散仮想サーバーを追加するには

コマンドプロンプトで、次のように入力します。

```
add lb vserver <name> HTTP * 80 -persistenceType NONE
<!--NeedCopy-->
```

例:

```
add lb vserver ProxyVserver-HTTP HTTP * 80 -persistenceType NONE -
    cltTimeout 120
<!--NeedCopy-->
```

HTTPS ビデオトラフィックを検出するための **SSL** ブリッジ仮想サーバーを追加するには

コマンドプロンプトで、次のように入力します。

```
add lb vserver <name> SSL_BRIDGE * 443 -persistenceType NONE
<!--NeedCopy-->
```

例:

```
add lb vserver ProxyVserver-SSL SSL_BRIDGE * 443 -persistenceType NONE
    -cltTimeout 180
<!--NeedCopy-->
```


HTTP 負荷分散仮想サーバーへの組み込み検出ポリシーのバインド

HTTP 接続を介してビデオトラフィックを検出するには、すべての組み込み検出ポリシーを負荷分散仮想サーバーにバインドする必要があります。ポリシーは、ポリシータイプに応じて、要求時間または応答時間処理のいずれかにバインドする必要があります。

> 注:
>
> ns_videoopt_http_body_detectionビデオ最適化ポリシーは CONNECTHTTP リクエストメソッドをサポートしていません。

さまざまなビデオタイプの検出ポリシーを **HTTP** 負荷分散仮想サーバにバインドするには

コマンドプロンプトで、タイプごとに適切なコマンドを入力します。使用可能なコマンドは次のとおりです。

```
bind lb vserver <name> -policyName ns_videoopt_http_abr_netflix -
    priority <integer> -type (REQUEST | RESPONSE)

bind lb vserver <name>   -policyName ns_videoopt_http_abr_netflix2 -
    priority <integer> -type (REQUEST | RESPONSE)

bind lb vserver <name>   -policyName ns_videoopt_http_abr_youtube -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver <name>    -policyName ns_videoopt_http_pd_youtube -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver <name>    -policyName ns_videoopt_http_pd_youtube2  -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver  <name>   -policyName ns_videoopt_http_pd_youtube3 -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver  <name>  -policyName ns_videoopt_http_abr_generic -
    priority <integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```

例**:**

```
bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_abr_netflix -priority 400  type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_abr_netflix2 -priority 500 -type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_abr_youtube -priority 600  -type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_pd_youtube -priority 800  -type RESPONSE

```



```
bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_pd_youtube2 -priority 900 -type RESPONSE

bind lb vserver  ProxyVserver-HTTP   -policyName
    ns_videoopt_http_pd_youtube3 -priority 1000 -type REQUEST

bind lb vserver  ProxyVserver-HTTP   -policyName
    ns_videoopt_http_abr_generic -priority 1100 -type RESPONSE
<!--NeedCopy-->
```

HTTP 本文コンテンツ検出ポリシーの負荷分散仮想サーバーへのバインド

HTTP 経由でビデオトラフィックを検出するには、本文コンテンツ検出ポリシーを負荷分散仮想サーバーにバインドする必要があります。次のコマンドを使用できます。

```
bind lb vserver <name> -policyName ns_videoopt_http_body_detection -
    priority <integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```

例:

```
bind lb vserver ProxyVserver-HTTP -policyName
    ns_videoopt_http_body_detection -priority 1500 -type REQUEST
<!--NeedCopy-->
```

SSL ブリッジ負荷分散仮想サーバへの組み込み検出ポリシーのバインド

HTTPS 接続を介してビデオトラフィックを検出するには、組み込みの検出ポリシーを SSL ブリッジ負荷分散仮想サーバーにバインドする必要があります。

検出ポリシーを **SSL** ブリッジ負荷分散仮想サーバにバインドするには

コマンドプロンプトで、タイプごとに適切なコマンドを入力します。使用可能なコマンドは次のとおりです。

```
bind lb vserver <name> -policyName ns_videoopt_https_abr_netflix -
    priority <positive_integer> -type (REQUEST | RESPONSE)

bind lb vserver <name> -policyName ns_videoopt_https_abr_youtube -
    priority <positive_integer> -type (REQUEST | RESPONSE)

bind lb vserver <name> -policyName ns_videoopt_https_abr_generic -
    priority <positive_integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```

例:


```
bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_netflix -priority 120 -type REQUEST

bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_youtube -priority 140 -type REQUEST

bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_generic -priority 150 -type REQUEST
<!--NeedCopy-->
```

ABR トラフィックのペーシングのための最適化ポリシーの追加

ABR トラフィックを最適化するには、最適化ポリシーと関連するアクションを設定する必要があります。次に、検出ポリシーをバインドしたのと同じ負荷分散仮想サーバーにポリシーをバインドします。ポリシーごとにアクションを最初に作成し、ポリシーの作成時にアクションを含めることができます。

最適化アクションを追加するには

コマンドプロンプトで入力します。

```
add videooptimization pacingaction <action Name> -rate <integer> [-
    comment <string>]
<!--NeedCopy-->
```

rate パラメータは、トラフィックを送信するレート（ペーシングレート）を Kbps 単位で指定します。

例**:**

```
add videooptimization pacingaction MyOptAct2000 -rate 2000
<!--NeedCopy-->
```

最適化ポリシーを追加するには

コマンドプロンプトで入力します。

```
add videooptimization pacingpolicy <name> -rule <expression> -action <
    string>
<!--NeedCopy-->
```

例**:**

```
add videooptimization pacingpolicy myOptPolicy2000 -rule TRUE -action
    MyOptAct2000
<!--NeedCopy-->
```


HTTP 負荷分散仮想サーバーへの最適化ポリシーのバインド

HTTP 接続を介して ABR ビデオトラフィックを最適化するには、検出ポリシーがバインドされている負荷分散仮想サーバに最適化ポリシーをバインドする必要があります。

最適化ポリシーを負荷分散仮想サーバーにバインドするには

コマンドプロンプトで、次のコマンドを入力します。

```
bind lb vserver <name> -policyName <policy_name> -priority <
   positive_integer> -type  (REQUEST | RESPONSE)
<!--NeedCopy-->
```

例**:**

```
bind lb vserver ProxyVserver-HTTP -policyName myOptPolicy2000 -priority
    3400 -type  REQUEST
<!--NeedCopy-->
```

SSL ブリッジ仮想サーバへの最適化ポリシーのバインド

HTTPS 接続を介して ABR ビデオトラフィックを最適化するには、組み込み検出ポリシーがバインドされている SSL Bridge 仮想サーバに最適化ポリシーをバインドする必要があります。

暗号化トラフィックのペーシング用に最適化ポリシーを **SSL Bridge** 仮想サーバにバインドするには

コマンドプロンプトで、次のコマンドを入力します。

```
bind lb vserver <name> -policyName <policy_name> -priority <
   positive_integer> -type  (REQUEST |RESPONSE)
<!--NeedCopy-->
```

例**:**

```
bind lb vserver ProxyVserver-SSL -policyName myOptPolicy2000 -priority
   3400 -type  REQUEST
<!--NeedCopy-->
```

ビデオ最適化のペーシングパラメータの設定

CLI では、ランダムサンプリングの割合などのビデオ最適化ペーシングパラメータを設定できます。

ランダムサンプリングのパーセンテージを設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
set videooptimization parameter  - RandomSamplingPercentage <realNumber>
<!--NeedCopy-->
```

ここで、実数は 0.0 から 100.0 までの値です。

例**:**

```
set videooptimization parameter -RandomSamplingPercentage 50
<!--NeedCopy-->
```

GUI を使用した **TCP** 経由のビデオ最適化の設定

GUI を使用すると、次のことが可能になります。

- ビデオ最適化機能を有効にします。
- HTTP 負荷分散仮想サーバーを作成します。
- SSL ブリッジ負荷分散仮想サーバーを作成します。
- 組み込みの検出ポリシーを HTTP 負荷分散仮想サーバーにバインドします。
- 組み込みの検出ポリシーを SSL ブリッジ負荷分散仮想サーバーにバインドします。
- 最適化ポリシーを作成します。
- 最適化アクションを作成します。
- 最適化ペーシングパラメータを設定します。
- 最適化ポリシーを HTTP トラフィック用の仮想サーバーの負荷分散にバインドします。
- HTTPS トラフィック用に最適化ポリシーを SSL ブリッジ負荷分散仮想サーバーにバインドします。

ビデオ最適化機能を有効にするには

1. ナビゲーションペインで、[システム] を展開し、[設定**]** をクリックします。
2. [設定] ページで、[拡張機能の構成] リンクをクリックします。
3. [拡張機能の構成] ページで、[ビデオの最適化] チェックボックスをオンにします。
4. 「**OK**」をクリックし、「閉じる」をクリックします。

HTTP トラフィック用の負荷分散仮想サーバーを作成するには

1. NetScaler ADC アプライアンスにサインインし、[トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [負荷分散仮想サーバー] 画面で、次のパラメータを設定します。

a) **Name**：負荷分散仮想サーバーの名前。
b) プロトコル。プロトコルタイプを HTTP として選択
c) **IP** アドレスタイプ。IP アドレスタイプ:IPv4 または IPv6。
d) **IP** アドレス。仮想サーバに割り当てられた IPv4 または IPv6 アドレス。
e) ポート。仮想サーバーのポート番号。

4.「**OK**」をクリックして、その他のオプションのパラメータの設定を続行します。詳細は、「仮想サーバーの作成」を参照してください。
5. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTPS トラフィック用の負荷分散仮想サーバーを作成するには

1. NetScaler ADC アプライアンスにサインインし、[トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [負荷分散仮想サーバー] 画面で、次のパラメータを設定します。

a) **Name**：負荷分散仮想サーバーの名前。
b) プロトコル。プロトコルタイプとして [SSL ブリッジ] を選択します。
c) **IP** アドレスタイプ。IP アドレスタイプ:IPv4 または IPv6。
d) **IP** アドレス。仮想サーバに割り当てられた IPv4 または IPv6 アドレス。
e) ポート。仮想サーバーのポート番号。

4.「**OK**」をクリックして、その他のオプションのパラメータの設定を続行します。詳細については、仮想サーバーの作成を参照してください。
5. [作成] をクリックし、[閉じる] をクリックします。

組み込みの検出ポリシーを負荷分散仮想サーバーにバインドするには

1. NetScaler ADC アプライアンスにサインインし、[トラフィック管理] > [負荷分散] > [仮想サーバー] 画面に移動します。
2. 詳細ペインで、負荷分散仮想サーバーを選択し、[編集] をクリックします。

a) [詳細設定] セクションで、[ポリシー] をクリックします。
b) [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
c) [**Policies**] セクションで、次のパラメータを設定します。
d)「ポリシー」を選択します。ドロップダウンリストからビデオ最適化検出ポリシーを選択します。
e)「タイプ」を選択します。ポリシータイプとして [要求] を選択します。
f)［続行］をクリックします。

3. リストからビデオ検出ポリシーを選択し、[閉じる] をクリックします。

組み込みの検出ポリシーを **SSL** ブリッジ負荷分散仮想サーバーにバインドするには

1. NetScaler ADC アプライアンスにログオンし、[トラフィック管理] > [負荷分散] > [仮想サーバー] 画面に移動します。
2. 詳細ウィンドウで、SSL ブリッジ負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [詳細設定] セクションで、[ポリシー] をクリックします。
4. [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
5. [**Policies**] セクションで、次のパラメータを設定します。

   a)「ポリシー」を選択します。ドロップダウンリストからビデオ最適化検出ポリシーを選択します。
   b)「タイプ」を選択します。ポリシータイプとして [要求] を選択します。

6. [続行] をクリックします。
7. リストからビデオ検出ポリシーを選択し、[閉じる] をクリックします。

ビデオ最適化アクションを作成するには

1. NetScaler ADC アプライアンスにログオンし、［構成］>［最適化］>［ビデオの最適化］>［ペーシング］>［アクション］に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **[**ビデオ最適化ペーシングアクションの作成] ページで、次のパラメータを設定します。

   a) **Name**：最適化アクションの名前。
   b) **ABR** 最適化レート（**Kbps**）。ABR ビデオトラフィックを送信するペーシングレート。ABR 最適化のデフォルトレートは 1000 Kbps です。最小値は 1 で、最大値は 2147483647 です。
   c) [コメント]。アクションの簡単な説明。

4. **[**作成**]** して **[**閉じる**]** をクリックします。

ビデオ最適化ポリシーを作成するには

1. NetScaler ADC アプライアンスにログオンし、［構成］>［最適化］>［ビデオの最適化］>［ペーシング］>［ポリシー］に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **[**ビデオ最適化ペーシングポリシーの作成] ページで、次のパラメータを設定します。

   a) **Name**：最適化ポリシーの名前
   b) **Expression**：ポリシーを実装するカスタム正規表現式。
   c) 操作。着信ビデオトラフィックを処理するポリシーに関連付けられた最適化アクション。
   d) **UNDEF** アクション。着信要求が最適化ポリシーと一致しない場合の未定義のイベント。
   e) [コメント]。ポリシーの簡単な説明。
   f) ログアクション。目的のログメッセージを作成する監査ログアクションを選択します。

4. [作成] をクリックし、[閉じる**]** をクリックします。

ビデオ最適化のペーシングパラメータを設定するには

1. NetScaler アプライアンスにログオンし、[構成] > [最適化] > [****** ビデオの最適化 ******] に移動します。
2. [ビデオの最適化] ページで、[ビデオの最適化設定の変更] リンクをクリックします。
3. [ビデオの最適化設定**]** ページで、次のパラメーターを設定します。

    a) ランダムサンプリングパーセンテージ **(%)**。ランダムサンプリング用に選択されたパケットの割合。

4.「**OK**」をクリックして「閉じる」をクリックします。

ビデオ最適化ポリシーを **HTTP** 負荷分散仮想サーバーにバインドするには

1. NetScaler アプライアンスにログオンし、[構成] > [最適化] > [****** ビデオの最適化 ******] に移動します。
2. [ビデオの最適化] ページで、[ビデオ最適化ペーシングポリシーマネージャ] リンクをクリックします。
3. 次のパラメータを設定します。

    a) バインドポイント。リクエストまたはレスポンスの処理中に最適化ポリシーを適用するポイント。
    b) 接続タイプ。要求または応答としての接続タイプ。
    c) 仮想サーバー。ポリシーをバインドする負荷分散仮想サーバー。
    d) ［続行］をクリックします。

4. [バインドポイント] セクションで、次のいずれかの操作を行います。

    a) リストからポリシーを選択します。
    b) [バインドの追加] をクリックして、[ポリシーのバインド] スライダにアクセスします。
        i. 既存のポリシーを選択するか、新しいポリシーを追加します。
        ii. バインドの詳細を入力し、[バインド] をクリックします。

5. ［閉じる］をクリックします。

ビデオ最適化ポリシーを **SSL** ブリッジ負荷分散仮想サーバーにバインドするには

1. **NetScaler** アプライアンスにログオンし、**[**** 構成] > [最適化] > [****** ビデオ最適化 ******] に移動します。******
2. [ビデオの最適化] ページで、[ビデオ最適化ペーシングポリシーマネージャ] リンクをクリックします。
3. [ビデオ最適化ポリシーマネージャ] ページで、次のパラメータを設定します。

    a) バインドポイント。要求/応答の処理中に最適化ポリシーを適用するポイント。
    b) 接続タイプ。接続タイプはリクエストまたはレスポンスです。
    c) 仮想サーバー。ポリシーをバインドする SSL ブリッジ負荷分散仮想サーバー。

4. ［続行］をクリックします。

5. [バインドポイント] セクションで、次のいずれかの操作を行います。

    a) リストからポリシーバインディングを選択します。
    b) [バインドの追加] をクリックして、[ポリシーのバインド] スライダにアクセスします。
        i. 既存のポリシーを選択するか、新しいポリシーを追加します。
        ii. バインドの詳細を入力し、[バインド] をクリックします。

6. [閉じる] をクリックします。

UDP によるビデオ最適化の設定

August 15, 2023

UDP 経由の QUIC ABR ビデオトラフィックを最適化するには、まずビデオ最適化機能を有効にします。設定が完了すると、アプライアンスは QUIC ベースの ABR ビデオトラフィックを検出し、アプライアンスに設定されている最適化ビットレートを適用します。

CLI を使用した QUIC のビデオ最適化の設定

UDP 経由の QUIC ビデオトラフィックのビデオ最適化を設定するには、次のタスクを実行する必要があります。

1. ビデオ最適化を有効にします。
2. QUIC サービスを作成します。
3. QUIC 負荷分散仮想サーバーを作成します。
4. QUIC Web サービスを負荷分散仮想サーバーにバインドします。
5. QUIC ベースの UDP トラフィックをペーシングするためのビデオ最適化ポリシーを作成します。
6. 最適化ポリシーを QUIC ベースの負荷分散仮想サーバーにバインドします。

QUIC トラフィックのビデオ最適化を有効にする

NetScaler アプライアンスにビデオトラフィックを検出、最適化、レポートさせるには、ビデオ最適化機能を有効にし、最適化をオンに設定する必要があります。

> 注
>
> QUIC トラフィックにビデオ最適化を使用する場合は、負荷分散と AppFlow 機能を有効にする必要があります。

動画最適化を有効にするには　コマンドプロンプトで、次のコマンドを入力します。

```
enable ns feature VideoOptimization
<!--NeedCopy-->
```

QUIC トラフィック用サービスの作成

NetScaler アプライアンスは、負荷分散仮想サーバーの QUIC サービスを使用して、静的ルーティングモードで出力ルーターに接続します。

> 注
>
> 現在、動的ルーティングはサポートされていません。

QUIC ビデオトラフィックの負荷分散 **Web** サービスを作成するには　コマンドプロンプトで入力します。

```
add service <name> <router-IP> <serviceType> <port> -usip yes -
   useproxyport  [yes | no]
<!--NeedCopy-->
```

例:

```
add service svc-quic 10.102.29.200 QUIC 443  -usip yes  -useproxyport
   no

where IP address is the internet router address.
<!--NeedCopy-->
```

QUIC トラフィック用の負荷分散仮想サーバーの作成

NetScaler アプライアンスは、負荷分散仮想サーバーを使用して、UDP 経由の QUIC ビデオトラフィックを検出して最適化します。

QUIC ビデオトラフィック用の負荷分散仮想サーバーを作成するには　コマンドプロンプトで入力します。

```
add lb vserver <name> <serviceType> <ip> <port> -m MAC
<!--NeedCopy-->
```

例:

```
add lb vserver vs-quic QUIC * 443 -persistenceType NONE -m MAC -
   cltTimeout 120
<!--NeedCopy-->
```


QUIC Web サービスを負荷分散仮想サーバーにバインドする

QUIC トラフィック用の Web サービスと負荷分散仮想サーバーを作成したら、サービスを仮想サーバーにバインドする必要があります。

Web サービスを **QUIC** ビデオトラフィックの負荷分散仮想サーバーにバインドするには　コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

```
bind lb vserver vs-quic svc-quic
<!--NeedCopy-->
```

QUIC ベースの **UDP** トラフィックのビデオ最適化ポリシーの作成

QUIC ベースの UDP トラフィックを最適化するには、最適化ペーシングポリシーとそのアクションを設定する必要があります。次に、ポリシーを QUIC ベースの負荷分散仮想サーバーにバインドする必要があります。ポリシーごとに、まずアクションを作成して、ポリシーに関連付けることができます。

最適化アクションを追加するには　コマンドプロンプトで入力します。

```
add videooptimization pacingaction <action Name> -rate <integer> [-
    comment <string>]
<!--NeedCopy-->
```

ここで、**rate** パラメータはトラフィックを送信するレート (ペーシングレート) を Kbps 単位で指定します。

例**:**

```
set videooptimization parameter -QUICPacingRate 1000
<!--NeedCopy-->
```

ここで、1000 は必要なペーシングレート (K ビット/秒) を表します。

最適化ポリシーを追加するには　コマンドプロンプトで入力します。

```
add videooptimization pacingpolicy <name> -rule <expression> -action <
    string>
<!--NeedCopy-->
```

例**:**


```
add videooptimization pacingpolicy myOptPolicy2000 -rule TRUE -action
    MyOptAct2000
<!--NeedCopy-->
```

最適化ポリシーを **QUIC** 負荷分散仮想サーバーにバインドする　UDP 接続で QUIC ビデオトラフィックを最適化するには、最適化ポリシーを QUIC 負荷分散仮想サーバーにバインドする必要があります。

最適化ポリシーを **QUIC** 負荷分散仮想サーバーにバインドするには　コマンドプロンプトで、次のコマンドを入力します。

```
bind lb vserver <name> -policyName <policy_name> -priority <
    positive_integer> -type (REQUEST)
<!--NeedCopy-->
```

> 注
>
> ペーシングポリシーは、リクエスト時にのみ QUIC 負荷分散仮想サーバーにバインドする必要があります。

例**:**

```
bind lb vserver vs-quic -policyName myOptPolicy2000 -priority 3400 -
    type  REQUEST
<!--NeedCopy-->
```

GUI を使用した **QUIC** のビデオ最適化の設定

GUI を使用してアプライアンスの機能を設定するには、次のタスクを実行する必要があります。

1. 動画最適化を有効にする
2. QUIC サーバーの設定
3. QUIC サービスの設定
4. QUIC 負荷分散仮想サーバーの設定
5. QUIC Web サービスを負荷分散仮想サーバーにバインドします
6. 最適化ポリシーを作成します。
7. 最適化アクションを作成します。
8. 最適化ペーシングパラメータを設定します。
9. 最適化ポリシーを QUIC トラフィックの負荷分散仮想サーバーにバインドします。

動画最適化を有効にするには

1. **NetScaler** アプライアンスにログオンし、**[**** システム] > [設定] に移動します。******

2. 詳細ページで、「拡張機能の設定」リンクを選択します。
3. [拡張機能の構成] ページで、[ビデオの最適化] チェックボックスをオンにします。

QUIC サーバーを作成するには

1. NetScaler アプライアンスにログオンし、[トラフィック管理] > [負荷分散] > [サーバー] 画面に移動します。
2. 詳細ペインで、[追加] をクリックします。
3.「サーバーの作成」ページで、次のパラメーターを設定します。

   a) [名前]。QUIC サーバーの名前。
   b) IP アドレス。QUIC サーバの IP アドレス
   c) トラフィックドメイン。サーバーのドメイン名。
   d) 作成後に有効化します。サーバーの初期状態。
   e) [コメント]。サーバーに関する簡単な情報。

4. ［作成］をクリックします。

QUIC サービスを作成するには

1. NetScaler アプライアンスにログオンし、[トラフィック管理] > [負荷分散] > [サービス] 画面に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. 負荷分散サービスページで、次のパラメータを設定します。

   a) サービス名。QUIC サービスの名前。
   b) **IP** アドレス。QUIC サービスに割り当てられた IP アドレス。
   c) プロトコル。プロトコルを QUIC として選択します。
   d) ポート。Web サービスのポート番号。

4.「**OK**」をクリックして続行します。その後、その他のオプションのパラメータを設定できます。詳細については、「サービスの設定」を参照してください。
5. オプションのパラメータを設定したら、[**OK**] をクリックして [閉じる] をクリックします。

負荷分散仮想サーバーを作成するには

1. NetScaler ADC アプライアンスにログオンし、[トラフィック管理] > [負荷分散] > [仮想サーバー] 画面に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. 負荷分散仮想サーバーページで、次のパラメーターを設定します。

   a) **Name**：負荷分散仮想サーバーの名前。
   b) プロトコル。サービスが QUIC リクエストを送信するために使用するプロトコル。
   c) IP アドレスタイプ。IP アドレスタイプ:IPv4 または IPv6。
   d) **IP** アドレス。仮想サーバーに割り当てられた IP 4 または IP6 IP アドレス。

e) ポート。仮想サーバーのポート番号。

4. 「OK」をクリックして、その他のオプションのパラメータの設定を続行します。詳細については、仮想サーバーの作成を参照してください。

負荷分散仮想サーバーを **QUIC** サービスにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. サービスとサービスグループをクリックして、負荷分散仮想サーバーサービスバインディング画面にアクセスします。
3. QUIC ベースの Web サービスを選択し、「バインド」をクリックします。
4. [完了] をクリックします。

負荷分散仮想サーバーを **QUIC** サービスにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. サービスとサービスグループをクリックして、負荷分散仮想サーバーサービスバインディング画面にアクセスします。
3. QUIC ベースの Web サービスを選択し、「バインド」をクリックします。
4. [完了] をクリックします。

QUIC トラフィックのビデオ最適化アクションを作成するには

1. NetScaler ADC アプライアンスにログオンし、[構成] > [最適化] > [ビデオの最適化] > [ペーシング] > [アクション] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **[**ビデオ最適化ペーシングアクションの作成] ページで、次のパラメータを設定します。

   a) **Name**: 最適化アクションの名前。
   b) **ABR** 最適化レート（**Kbps**）。ABR ビデオトラフィックを送信するペーシングレート。ABR 最適化のデフォルトレートは 1000 Kbps です。最小値は 1 で、最大値は 2147483647 です。
   c) [コメント]。アクションの簡単な説明。

4. **[**作成**]** して **[**閉じる**]** をクリックします。

QUIC トラフィックのビデオ最適化ポリシーを作成するには

1. NetScaler ADC アプライアンスにログオンし、[構成] > [最適化] > [ビデオの最適化] > [ペーシング] > [ポリシー] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **[**ビデオ最適化ペーシングポリシーの作成] ページで、次のパラメータを設定します。

   a) [名前]。最適化ポリシーの名前

b) 式。ポリシーを実装するカスタムの regrex 表現。
c) アクション。着信ビデオトラフィックを処理するポリシーに関連付けられた最適化アクション。
d) UNDEF アクション。着信要求が最適化ポリシーと一致しない場合の未定義のイベント。
e) [コメント]。ポリシーの簡単な説明。
f) ログアクション。目的のログメッセージを作成する監査ログアクションを選択します。

4. [作成] をクリックし、[閉じる] をクリックします。

ビデオ最適化ポリシーを **QUIC** 負荷分散仮想サーバーにバインドするには

1. **NetScaler** アプライアンスにログオンし、**[**** 構成] > [最適化] > [****** ビデオ最適化 **] に移動します。**
2. [ビデオの最適化] ページで、[ビデオ最適化ペーシングポリシーマネージャ] リンクをクリックします。
3. [ビデオ最適化ポリシーマネージャ] ページで、次のパラメータを設定します。

a) バインドポイント。リクエスト処理中に最適化ポリシーを適用するポイント。注: ペーシングポリシーは、リクエスト時にのみ QUIC 負荷分散仮想サーバーにバインドする必要があります。
b) 接続タイプ。要求または応答としての接続タイプ。
c) 仮想サーバー。ポリシーをバインドする負荷分散仮想サーバー。

4. [続行] をクリックします。
5. [バインドポイント] セクションで、次のいずれかの操作を行います。

a) リストからポリシーを選択します。
b) [バインドの追加] をクリックして、[ポリシーのバインド] スライダにアクセスします。
   i. 既存のポリシーを選択するか、新しいポリシーを追加します。
   ii. バインドの詳細を入力し、[バインド] をクリックします。

6. [閉じる] をクリックします。

NetScaler URL フィルタリング

August 15, 2023

URL フィルタリングは、URL に含まれる情報を使用して Web サイトをポリシーベースで制御します。この機能は、ネットワーク管理者がモバイルネットワーク上の悪意のある Web サイトへのユーザーアクセスを監視および制御するのに役立ちます。

管理者は、URL 分類機能または URL リスト機能を使用して URL フィルタリングポリシーを構成できます。

URL リスト。アプライアンスにインポートされた URL セット内の URL へのアクセスをブロックすることにより、ブラックリストに登録された Web サイトおよび Web ページへのアクセスを制御します。

URL 分類。事前に定義されたカテゴリのリストに基づいてトラフィックをフィルタリングすることにより、Web サイトや Web ページへのアクセスを制御します。

URL リスト

August 15, 2023

URL リスト機能を使用すると、カスタマイズした URL リスト (最大 100 万エントリ) へのアクセスを制御できます。この機能は、仮想サーバーにバインドされた URL フィルタリングポリシーを適用して Web サイトをフィルタリングします。

管理者は、URL リストを NetScaler アプライアンスにインポートする必要があります。このインポートされたリストは、*URL* セットと呼ばれるポリシーデータセットとして内部的に保存されます。次に、アプライアンスは受信 URL リクエストに独自の高速 URL マッチングアルゴリズムを適用します。受信 URL リクエストがセット内のエントリと一致する場合、アプライアンスは関連するポリシーアクションを適用してアクセスを制御します。

URL リストタイプ

URL セットの各エントリには、URL と、オプションでそのメタデータ (URL カテゴリ、カテゴリグループ、またはその他の関連データ) を含めることができます。メタデータを含む URL の場合、アプライアンスはメタデータを評価するポリシー式を使用します。詳細については、「URL セット」を参照してください。

カスタム **URL** リスト。最大 1,000,000 個の URL エントリからなるカスタマイズされた URL セットを作成し、それをテキストファイルとしてアプライアンスにインポートできます。リストには、メタデータ (URL カテゴリのように) メタデータの有無にかかわらずの URL を含めることができます。NetScaler プラットフォームは、メタデータの存在を自動的に検出します。また、インポートされたリストを安全に保存することもできます。詳細については、「URL セット」を参照してください。

URL リストをホストし、NetScaler ADC アプライアンスを構成して、手動で操作しなくてもリストを定期的に更新できます。URL リストが更新されると、アプライアンスはポリシー表現を使用して各受信 URL を評価し、許可、ブロック、リダイレクト、ユーザーへの通知などのアクションを適用することで、メタデータとカテゴリを自動的に検出し、ユーザーへの通知を行うことができます。

URL リストポリシー表現

次の表に、着信トラフィックの評価に使用できる基本的な式を示します。URL リストをアプライアンスにインポートすると、*URL* セットと呼ばれます。

式	操作
`<URL expression>.URLSET_MATCHES_ANY(<URLSET>)`	URL が URL セット内のいずれかのエントリと完全に一致する場合、TRUE と評価されます。

式	操作
<URL expression>.GET_URLSET_METADATA(<URLSET>)	GET_URLSET_METADATA () 式は、URL が URL セット内のいずれかのパターンと完全に一致する場合、関連するメタデータを返します。一致しない場合は、空の文字列が返されます。
<URL expression>.GET_URLSET_METADATA(<URLSET>).EQ(<METADATA>)	一致したメタデータが<METADATA>と等しい場合は TRUE と評価されます。
<URLexpression>.GET_URLSET_METADATA(<URLSET>).TYPECAST_LIST_T(‘,’).GET(0).EQ(<CATEGORY>)	一致したメタデータがカテゴリの先頭にある場合は TRUE と評価されます。このパターンを使用すると、メタデータ内の個別のフィールドをエンコードできますが、1st 一致するのはフィールドだけです。
HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL)	ホストパラメータと URL パラメータを結合し、<URL expression> マッチングに使用します。

URL リストポリシーアクション

URL リストに一致する URL に対する最も一般的な強制措置は、アクセスを制限することです。目的の URL リストマッチング表現と実施アクションを含む URL リストポリシーを作成します。ポリシーグループの使用状況は、受信トラフィックのタイプ（HTTP または HTTPS）とアプライアンスに設定されている仮想サーバによって異なります。HTTP トラフィックにはレスポンダーポリシーを、HTTPS トラフィックにはビデオ最適化ポリシーを使用できます。ポリシー内の表現と一致する URL に適用するアクションを指定します。次の表は、実行可能なアクションの一覧です。

アクションタイプ	ポリシー	説明
ALLOW	レスポンダー	リクエストがターゲット URL にアクセスすることを許可します。
リダイレクト	レスポンダー	ターゲットとして指定された URL にリクエストをリダイレクトします。
DENY	レスポンダー	リクエストを拒否します。
RESET	レスポンダー、ビデオ最適化	接続をリセットします。
DROP	レスポンダー、ビデオ最適化	接続を切断します。

前提条件

URL リスト機能を設定するには、次のサーバーが設定されていることを確認してください。

DNS リクエスト用の DNS サーバー

ホスト名 URL から URL セットをインポートする場合は、DNS サーバーを設定する必要があります。

コマンドプロンプトで入力します。

```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>) [-state (
    ENABLED | DISABLED )] [-type <type>] [-dnsProfileName <string>]
<!--NeedCopy-->
```

例:

```
add dns nameServer 10.140.50.5
<!--NeedCopy-->
```

カスタム URL リストのインポート

URL セットをインポートするには、URL セットのトピックを参照してください 。

HTTP トラフィックの URL リストの設定

NetScaler アプライアンスは、HTTP および HTTPS トラフィックをサポートします。HTTP トラフィック用の負荷分散仮想サーバーを構成し、URL リストポリシーをサーバーにバインドするには、次の手順を実行します。

- URL リストアクションを追加します。
- URL リストポリシーを追加します。
- HTTP トラフィック用の HTTP 負荷分散仮想サーバーの追加
- URL リストポリシーを HTTP トラフィック用の HTTP 負荷分散仮想サーバーにバインドします

URL リストアクションを追加するには

コマンドプロンプトで、次のように入力します。

```
add responder action <name> <type> (<target> | <htmlpage>) [-comment <
    string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <
    string>]
<!--NeedCopy-->
```


HTTP トラフィック用の **HTTP** 負荷分散仮想サーバーを追加するには

コマンドプロンプトで、次のように入力します。

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT
    imeout <secs>]
<!--NeedCopy-->
```

例:

```
add lb vserver vsrv-HTTP HTTP * 80 -persistenceType NONE -cltTimeout
    120
<!--NeedCopy-->
```

URL リストポリシーを **HTTP** 負荷分散仮想サーバーにバインドするには

コマンドプロンプトで、次のように入力します。

```
bind lb vserver <vServerName> -policyName <string> [-priority <
    positive_integer>]
<!--NeedCopy-->
```

HTTPS トラフィックの **URL** リストの設定

NetScaler アプライアンスは、HTTP および HTTPS トラフィックをサポートします。HTTPS トラフィック用の SSL ブリッジ負荷分散仮想サーバーを構成し、URL リストポリシーをサーバーにバインドするには、次の手順を実行します。

- URL リストアクションを追加します。
- URL リストポリシーを追加します。
- HTTP トラフィック用の SSL ブリッジ負荷分散仮想サーバーの追加
- URL リストポリシーを HTTP トラフィック用の SSL ブリッジ負荷分散仮想サーバーにバインドします

HTTPS トラフィックの **URL** リストポリシーを追加するには

コマンドプロンプトで入力します。

```
add videooptimization detectionpolicy <name> -rule <expression> -action
     <string> [-undefAction <string>] [-comment <string>] [-logAction <
    string>]
<!--NeedCopy-->
```


SSL ブリッジ負荷分散仮想サーバーを追加するには

コマンドプロンプトで次のように入力します。

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT
    imeout <secs>]
<!--NeedCopy-->
```

例**:**

```
add lb vserver vsrv-HTTPS SSL_BRIDGE * 443 -persistenceType NONE -
    cltTimeout 180
<!--NeedCopy-->
```

CLI を使用して **URL** リストポリシーを **SSL** ブリッジ負荷分散にバインドするには

コマンドプロンプトで次のように入力します。

```
bind lb vserver <vServerName> -policyName <string> [-priority <
    positive_integer>]
<!--NeedCopy-->
```

GUI を使用した URL リストの設定

GUI を使用すると、次のことが可能になります。

- URL リストをインポートします。
- URL リストを追加します。
- URL リストアクションを設定します。
- HTTP トラフィックの URL リストポリシーを設定します。
- HTTP トラフィック用の HTTP 負荷分散仮想サーバーを追加します。
- HTTPS トラフィック用の SSL ブリッジ負荷分散仮想サーバーを追加します。
- URL リストポリシーを HTTP 負荷分散仮想サーバーにバインドします。
- URL リストポリシーを SSL ブリッジ負荷分散仮想サーバーにバインドします。

URL リストをインポートするには

1. ナビゲーションペインで、**AppExpert** > **URL** セットを展開します。
2. 詳細ペインで、[インポート] をクリックします。
3.「**URL** セットの設定」ページで、次のパラメータを設定します。

    a) **Name**: URL セットの名前。
    b) **URL**。URL セットにアクセスする場所の Web アドレス。

c) 上書き。以前にインポートした URL セットを上書きします。
d) 区切り記号。CSV ファイルレコードを区切る文字シーケンス。
e) 行区切り記号。CSV ファイルで使用される行区切り文字。「/n」のように 1 文字の値を使用できます。
f) 間隔。URL セットが更新される間隔 (秒単位)。15 分未満に四捨五入されます。
g) プライベートセット。URL セットのエクスポートを禁止するオプション
h) カナリアの **URL**。URL セットのコンテンツを秘密にしておくべきかどうかをテストするための内部 URL。URL の最大長は 2047 文字です。

4.「作成」をクリックし、「閉じる」をクリックします。

URL リストを追加するには

1. ナビゲーションペインで、**AppExpert** > **URL** セットを展開します。
2. 詳細ペインで、[追加] をクリックします。
3.「**URL** セットの作成」ページで、次のパラメータを設定します。

a) **Name**: インポート時に指定された URL セットの名前。
b) [コメント**]**。URL セットに関する簡単な説明。

4.［作成］をクリックします。

URL リストアクションを設定するには

1. NetScaler アプライアンスにログオンし、「構成」タブページに移動します。
2. メニューペインで、AppExpert**>** レスポンダー > アクションに移動します。
3. 詳細ペインで、[追加] をクリックします。
4.「レスポンダーアクションの作成」ページで、次のパラメーターを設定します。

a) **Name**: URL リストポリシーアクションの名前。
b) 種類。アクションタイプを選択します。
c) **Expression**: エクスプレッションエディタを使用してポリシーエクスプレッションを作成します。
d) [コメント**]**。ポリシーアクションに関する簡単な説明。

5. **[**作成**]** して **[**閉じる**]** をクリックします。

URL リストポリシーを設定するには

1. ナビゲーションペインで、AppExpert**>** レスポンダー > ポリシーを展開します。
2. 詳細ペインで、[追加] をクリックします。
3. **[**レスポンダーポリシーの作成] ページで、次のパラメータを設定します。
   1.**Name**: URL リストポリシーアクションの名前。
   2. アクション。ポリシーに関連付けたい URL リストアクションを選択します。

3. ログアクション。ログアクションを選択します。
4. **AppFlow**。AppFlow アクションを選択します。
5. **Expression**：エクスプレッションエディタを使用してポリシーエクスプレッションを作成します。

a) [コメント**]**。ポリシーに関する簡単な説明。

4. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTP 負荷分散仮想サーバーを追加するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [負荷分散仮想サーバー] 画面で、次のパラメータを設定します。

a) **Name**：負荷分散仮想サーバーの名前。
b) プロトコル。プロトコルタイプを HTTP として選択します。
c) **IP** アドレスタイプ。IP アドレス指定可能なタイプ。
d) **IP** アドレス。仮想サーバーに割り当てられた IP 4 または IP6 IP アドレス。
e) ポート。仮想サーバーのポート番号。

4.「**OK**」をクリックして、その他のオプションのパラメータの設定を続行します。詳細は、「仮想サーバーの作成」を参照してください。

URL リストポリシーを **HTTP** 負荷分散仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] 画面に移動します。
2. 詳細ペインで、負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [詳細設定] セクションで、[ポリシー] をクリックします。
4. [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
5. [**Policies**] セクションで、次のパラメータを設定します。

a)「ポリシー」を選択します。ドロップダウンリストから URL 分類ポリシーを選択します。
b)「タイプ」を選択します。ポリシータイプとして [要求] を選択します。

6. [続行] をクリックします。
7.「ポリシー」ページで、リストから URL リストポリシーを選択し、「選択」をクリックします。
8.「ポリシー」スライダーで、「****** バインドして閉じる ******」をクリックします。

HTTPS トラフィックの **URL** リストポリシーを追加するには

1. **NetScaler** アプライアンスにログオンし、**[**** 構成] > [最適化] > [****** ビデオ最適化 ******] > [検出] に移動します。******

2. 検出ページで、「ビデオ最適化検出ポリシー」リンクをクリックします。
3. 「ビデオ最適化検出ポリシー」ページで、「追加」をクリックします。
4. ビデオ最適化検出ポリシーの作成ページで、次のパラメータを設定します。

    a) **Name**：最適化ポリシーの名前
    b) **Expression**：カスタム表現を使用してポリシーを設定します。
    c) アクション。着信ビデオトラフィックを処理するポリシーに関連付けられた最適化アクション。
    d) **UNDEF** アクション。着信要求が最適化ポリシーと一致しない場合の未定義のイベント。
    e) [コメント]。ポリシーの簡単な説明。
    f) ログアクション。ログメッセージに対して実行するアクションを指定する監査ログアクションを選択します。

5. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTPS トラフィック用の **SSL** ブリッジ負荷分散仮想サーバーを追加するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [負荷分散仮想サーバー] 画面で、次のパラメータを設定します。

    a) **Name**：負荷分散仮想サーバーの名前。
    b) プロトコル。プロトコルタイプとして [SSL ブリッジ] を選択します。
    c) **IP** アドレスタイプ。IP アドレスタイプ:IPv4 または IPv6。
    d) **IP** アドレス。仮想サーバーに割り当てられた IPv4 または IP6VIP アドレス。
    e) ポート。仮想サーバーのポート番号。

4. 「**OK**」をクリックして、その他のオプションのパラメータの設定を続行します。詳細については、「仮想サーバーの作成」トピックを参照してください。

URL リストポリシーを **SSL** ブリッジ負荷分散仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] 画面に移動します。
2. 詳細ウィンドウで、SSL ブリッジ負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [詳細設定] セクションで、[ポリシー] をクリックします。
4. [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
5. 次のパラメータを設定します。

    a) ポリシーを選択します。ドロップダウンリストからビデオ検出ポリシーを選択します。
    b) 「タイプ」を選択します。ポリシータイプとして [要求] を選択します。

6. [続行] をクリックします。
7. リストからビデオ検出ポリシーを選択し、[閉じる] をクリックします。

監査ログメッセージの設定

監査ログを使用すると、URL リストプロセスのどの段階でも条件や状況を確認できます。NetScaler アプライアンスが受信 URL を受信すると、レスポンダーポリシーに URL セットの高度なポリシー表現が含まれている場合、監査ログ機能は URL 内の URL セット情報を収集し、監査ログで許可されているターゲットの詳細をログメッセージとして保存します。

ログメッセージには次の情報が含まれます。

1. タイムスタンプ。
2. ログメッセージタイプ。
3. 定義済みのログレベル (重大、エラー、通知、警告、情報、デバッグ、アラート、緊急)。
4. URL セット名、ポリシーアクション、URL などのメッセージ情報をログに記録します。

URL リスト機能の監査ログを設定するには、次のタスクを完了する必要があります。

1. 監査ログを有効化。
2.「監査ログの作成」メッセージアクション。
3. [監査ログメッセージ] アクションで URL リストレスポンダポリシーを設定します。

詳細については、「監査ログ」を参照してください。

URL リストのセマンティクス

次の表は、URL の一致パターンを一覧表示し、URL リスト内の URL が受信リクエスト URL とどのように照合されるかを説明しています。たとえば、www.example.com/bar というパターンは www.example.com/bar の 1 ページにのみマッチします。URL が‘www.example.com/bar’で始まるすべてのページと一致させるには、URL の末尾にアスタリスク (*) を追加します。

説明	URL パターン	一致しました	一致しません
サブドメインマッチング	domain.com	domain.com; www.domain.com; sub.one.domain.com	yourdomain.com; wwwdomain.com
URL マッチング、正確なパス	domain.com/example/bar	domain.com/example/bar; www.domain.com/example/bar; sub.one.domain.com/example/bar	www.domain.com/example/bar/index.html; sub.one.domain.com/example/bar/index.html
URL マッチング、正確なパス	domain.com/example/bar	domain.com/example/bar; www.domain.com/example/bar; sub.one.domain.com/example/bar	www.domain.com/example/bar/index.html?key=value; www.domain.com/example/bar/index.html?; sub.one.domain.com/example/bar/index.html

説明	URL パターン	一致しました	一致しません
URL マッチング、サブパスマッチング	ドメイン.com/example/bar/	domain.com/example/bar/; www.domain.com/example/bar/; www.domain.com/example/bar/index.html; domain.com/example/bar/index.html/one.jpg	

URL の分類

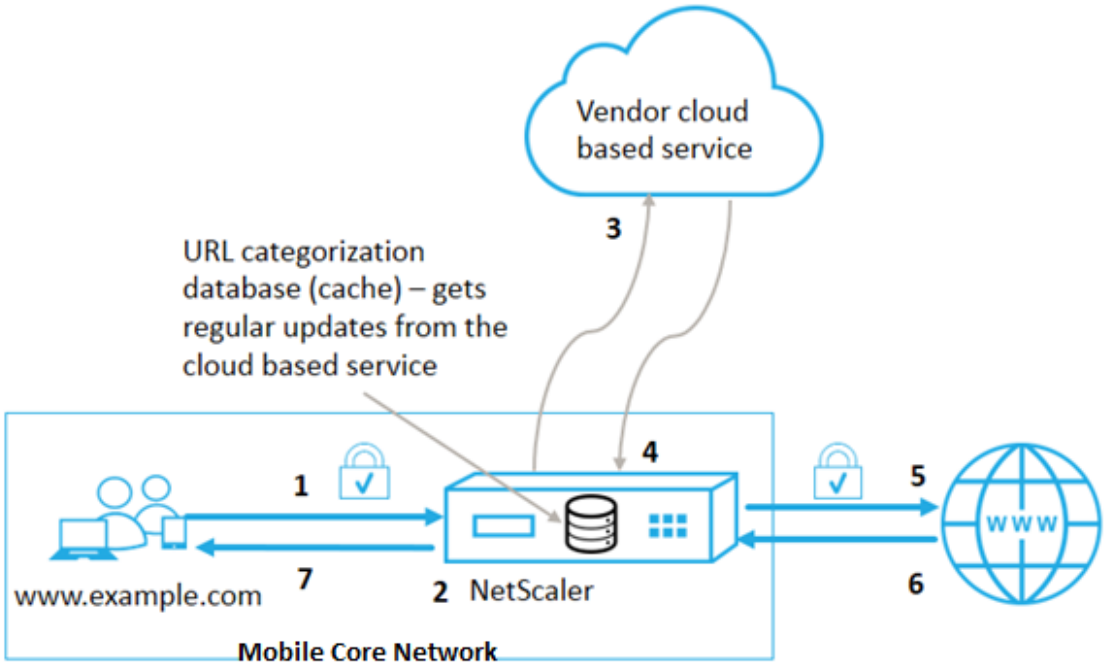
August 15, 2023

URL の分類は、特定の Web サイトおよび Web サイトのカテゴリへのユーザーアクセスを制限します。NetSTARと連携したサブスクライブサービスであるこの機能により、企業のお客様は、市販の分類データベースを使用して Web トラフィックをフィルタリングできます。NetSTARデータベースには、ソーシャルネットワーキング、ギャンブル、アダルトコンテンツ、ニューメディア、ショッピングなど、さまざまなカテゴリに分類された膨大な数（数十億）の URL があります。分類に加えて、各 URL には、サイトの履歴リスクプロファイルに基づいて最新のレピュテーションスコアが保持されます。カテゴリ、カテゴリグループ（テロ、違法薬物など）、またはサイトレピュテーションスコアに基づいて高度なポリシーを設定することで、NetSTARデータを使用してトラフィックをフィルタリングできます。

たとえば、マルウェアに感染したサイトなどの危険なサイトへのアクセスをブロックしたり、アダルトコンテンツやエンターテイメントストリーミングメディアへのアクセスを選択的に制限したりできます。

URL 分類の仕組み

次の図は、NetScaler URL Filtering サービスを商用 URL 分類データベースおよびクラウドサービスと統合して頻繁に更新する方法を示しています。



コンポーネントは次のように相互作用します。

1. クライアントはインターネットにバインドされた URL リクエストを送信します。
2. NetScaler ポリシーは、URL 分類データベースから取得した分類の詳細（カテゴリ、カテゴリグループ、サイトレピュテーションスコアなど）に基づいてリクエストを評価しようとします。データベースがカテゴリの詳細を返す場合、プロセスは手順 5 にジャンプします。
3. データベースが分類の詳細を返さない場合、リクエストは URL 分類ベンダーが管理するクラウドベースの検索サービスに送信されます。ただし、アプライアンスは応答を待たない。代わりに、URL を未分類としてマークし、手順 5 に進みます。ただし、クラウドクエリのフィードバックを引き続き監視し、それを使用してキャッシュを更新し、将来のリクエストでクラウドルックアップのメリットを得られるようにします。
4. NetScaler アプライアンスは、クラウドベースのサービスから URL カテゴリの詳細（カテゴリ、カテゴリグループ、およびレピュテーションスコア）を受け取り、クラウドキャッシュに保存します。
5. ポリシーで URL が許可されている場合、リクエストはオリジンサーバーに送信されます。そうしないと、アプライアンスは要求をドロップまたはリダイレクトするか、カスタム HTML ページで応答します。
6. オリジンサーバーは、要求されたデータで NetScaler アプライアンスに応答します。
7. アプライアンスは応答をクライアントに送信します。

URL フィルタリング機能を使用すると、政府が発行した安全なインターネット使用規則に違反するサイトを検出し、それらのサイトをブロックするポリシーを実装できます。成人向けコンテンツ、ストリーミングメディア、ソーシャルネットワーキングをホストするサイト。子供にとって安全でないと特定された、または違法として禁止されているサイト。

前提条件

この機能は、基本的な CBM ライセンスと CBM Premium ライセンスを購入すれば Telco プラットフォームで動作し、他の NetScaler プラットフォームでは CNS Premium ライセンスを購入すれば機能します。

注: ベーシック CBM ライセンスと CBM Premium ライセンスに加えて、アプライアンスには URL Threat Intelligence ライセンスと 1 年または 3 年間のサブスクリプションサービスが必要です。この機能を有効にして設定する前に、次のライセンスをインストールする必要があります。

Telco プラットフォームのライセンスサポート:

- **CBM_TXXX_SERVER_Retail.lic**
- **CBM_TPRE_SERVER_Retail.lic**
- **CNS_WEBF_SSERVER_Retail.lic**

ここで、XXX はスループットです。たとえば、NetScaler T1000 のようになります。

他の NetScaler プラットフォームのライセンスサポート:

- **CNS_XXX_SERVER_PLT_Retail.lic**

ここで、XXX はスループットです。

URL 分類ポリシー表現

次の表は、受信 URL を識別し、設定されたアクションを適用するためのさまざまな URL 分類ポリシー表現を示しています。

式	操作
<text>. URL_CATEGORIZE (<min_reputation>, <max_reputation>)	URL_CATEGORY オブジェクトを返します。レピュテーションスコアは 1 から 4 までの数字です。オブジェクトを取得するには、すべてのレピュテーションスコアは<min reputation>に 0.0 を使用します。. もしあれば が 0 より大きい場合、返されるオブジェクトにはレピュテーションが以下のカテゴリが含まれていません. もしあれば が 0 より大きい場合、返されるオブジェクトにはレピュテーションが次のカテゴリが含まれていません。. カテゴリがタイムリーに解決に失敗した場合は、undef 値が返されます。
。カテゴリー	このオブジェクトのカテゴリ文字列を返します。URL にカテゴリがない場合、または URL の形式が正しくない場合、戻り値は「未分類」です。

式	操作
。グループ	オブジェクトのカテゴリグループを識別する文字列を返します。これは、カテゴリのより高いレベルのグループです。これは、URL カテゴリに関する詳細情報を必要としない操作に役立ちます。URL にカテゴリがない場合、または URL の形式が正しくない場合、戻り値は「未分類」です。
。評判	レピュテーションスコアを 1 ～4 の数値で返します。4 は最もリスクの高いレピュテーションを示します。カテゴリが「未分類」の場合、レピュテーション値は 2 です。

ポリシー表現の例

ポリシー	ポリシー表現
検索エンジンカテゴリに含まれる URL のリクエストを選択するポリシー	レスポンダーポリシー p1 ‘HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL) .URL_CATEGORIZE (0,0) を追加します。カテゴリー.EQ (「検索エンジン」)
アダルトカテゴリグループに属する URL のリクエストを選択するポリシー	レスポンダーポリシー p1 ‘HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL) .URL_CATEGORIZE (0,0) を追加します。GROUP.EQ(“Adult”)’
レピュテーションスコアが 4 の検索エンジン URL のリクエストを選択するポリシー。	レスポンダーポリシー p2 ‘HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL) .URL_CATEGORIZE (4,0) を追加します。 CATEGORY.EQ(“Search Engine”)’
検索エンジンとショッピング URL のリクエストを選択するポリシー	add policy patset good_categories; bind policy good_categories “Search Engine”; bind policy good_categories “Shopping”; add responder policy p3 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO CATEGORY .EQUALS_ANY (「Good_categories」)
レピュテーションスコアが 4 の検索エンジン URL のリクエストを選択するポリシー。	レスポンダーポリシー p5 ‘CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE (4,0) を追加します。カテゴリー.EQ (「検索エンジン」)

URL 分類ポリシーアクション

URL フィルタリングポリシーは、トラフィックを評価して、特定のカテゴリに属するリクエストを識別します。次の表は、URL フィルタリングポリシーに割り当てることができるアクションの一覧です。

ポリシーアクション	ポリシーグループ	説明
ALLOW	レスポンダー	受信リクエストにターゲット URL へのアクセスを許可する
リダイレクト	レスポンダー	受信リクエストをターゲットとして指定された URL にリダイレクトします。
DENY	レスポンダー	受信リクエストを拒否します。
RESET	レスポンダー、ビデオ最適化	接続をリセットします。
DROP	レスポンダー、ビデオ最適化	接続をドロップします。

> 注
>
> 暗号化されたトラフィックの場合、動画最適化ポリシーには URL フィルタリングアクションを実装するアクションが含まれます。

URL 分類の設定

URL 分類を設定するには、まず URL フィルタリング機能を有効にします。次に、HTTP および HTTPS トラフィックのキャッシュメモリ制限、分類ポリシー、および仮想サーバーを設定する必要があります。CLI を使用して URL 分類を設定します。

CLI を使用して NetScaler アプライアンスで URL 分類を構成するには、次の手順を実行します。

- URL 分類を設定します。
  - URL フィルタリング機能を有効にします。
  - 共有メモリを設定してキャッシュメモリを制限します。
  - URL 分類パラメータを設定します。
- HTTP トラフィックの URL 分類を設定します。
  - URL 分類アクションを追加します。
  - URL 分類ポリシーを追加します。
  - HTTP トラフィック用の負荷分散仮想サーバーを追加します。
  - URL 分類ポリシーを負荷分散仮想サーバーにバインドします。
- HTTPS トラフィックの URL 分類を設定します。

- URL 分類ポリシーを追加します。
- SSL ブリッジ負荷分散仮想サーバーを追加します。
- URL 分類ポリシーを負荷分散仮想サーバーにバインドします。

URL 分類の設定

この機能を設定するには、URL 分類機能を有効にし、フィルタリングパラメータを設定し、共有メモリ制限を設定する必要があります。

URL フィルタリング機能を有効にするには

コマンドプロンプトで入力します。

```
enable ns feature URLFiltering VideoOptimization Responder IC SSL
AppFlow
```

共有メモリ制限を設定するには

コマンドプロンプトで入力します。

```
1 set cache parameter [-memLimit <megaBytes>]
2 <!--NeedCopy-->
```

ここで、MemLimit はキャッシュのメモリ制限です。

例:

```
set cache parameter -memLimit 10
```

URL 分類パラメータを設定するには

コマンドプロンプトで入力します。

```
1 set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer>]
     [-TimeOfDayToUpdateDB <HH:MM>]
2 <!--NeedCopy-->
```

*** 例:**

```
set urlfiltering parameter -HoursBetweenDBUpdates 3 -TimeOfDayToUpdateDB
 03:00
```


HTTP トラフィックの URL 分類の設定

HTTP トラフィックの URL 分類機能を設定するには、負荷分散仮想サーバーを構成し、URL 分類ポリシーを追加して、ポリシーを仮想サーバーにバインドする必要があります。これにより、仮想サーバーは HTTP トラフィックを受信し、ポリシー評価に基づいてフィルタリングアクションを割り当てます。

HTTP トラフィックに URL 分類アクションを追加するには

コマンドプロンプトで入力します。

```
add responder action <name> <type> (<target> | <htmlpage>)[-comment
<string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <
string>]
```

例:

```
add responder action act_url_categorize respondwith "\"HTTP/1.1 200
OK\r\n\r\n\" + HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE
(0,0).CATEGORY + \"\n\""
```

HTTP トラフィックの URL 分類ポリシーを追加するには

コマンドプロンプトで入力します。

```
add responder policy <name> <rule> <action> [<undefAction>] [-comment
 <string>] [-logAction <string>] [-appflowAction <string>]
```

例:

```
add responder policy pol_url_categorize_http "HTTP.REQ.HOSTNAME.
APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).GROUP.EQ(\"Adult\")|| HTTP
.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).GROUP.EQ(\"
Gambling\")"RESET
```

HTTP 負荷分散仮想サーバーを追加するには

HTTP トラフィック用の仮想サーバーがまだ構成されていない場合は、コマンドプロンプトで次のように入力します。

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-clt
Timeout <secs>]
```

例:

```
add lb vserver vsrv-HTTP HTTP * 80 -persistenceType NONE -cltTimeout
120
```


URL 分類ポリシーを負荷分散仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> -policyName <string> [-priority <positive_integer
>]
```

例:

```
bind lb vserver vsrv-HTTP -policyName pol_url_categorize_http -
priority 10 -gotoPriorityExpression END -type REQUEST
```

HTTPS トラフィックの URL 分類の設定

HTTPS トラフィックの URL 分類機能を設定するには、SSL ブリッジ負荷分散仮想サーバーを設定し、URL 分類ポリシーを追加して、ポリシーを SSL ブリッジ仮想サーバーにバインドする必要があります。これにより、サーバーは HTTPS トラフィックを受信し、ポリシー評価に基づいてフィルタリングアクションを割り当てます。

HTTPS トラフィックの **URL** 分類ポリシーを追加するには

コマンドプロンプトで入力します。

```
add videooptimization detectionpolicy <name> -rule <expression>
-action <string> [-undefAction <string>] [-comment <string>] [-
logAction <string>]
```

例:

```
add videooptimization detectionpolicy pol_url_categorize_https_block_adult
 -rule "CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE(0,0).CATEGORY.EQ("
Adult")' -action RESET
```

SSL ブリッジ負荷分散仮想サーバーを追加するには

コマンドプロンプトで入力します。

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT
imeout <secs>]
```

例:

```
add lb vserver vsrv-HTTPS SSL_BRIDGE * 443 -persistenceType NONE -
cltTimeout 180
```


分類ポリシーを **SSL** ブリッジ仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> -policyName <string> [-priority <positive_integer
>]
```

例**:**

```
bind lb vserver vsrv-HTTPS -policyName pol_url_categorize_https_block_adult
 -priority 20 -type REQUEST
```

GUI を使用した **URL** 分類の設定

GUI を使用すると、次のことが可能になります。

- URL 分類機能を有効にします。
- HTTP トラフィックの URL 分類アクションを追加します。
- HTTP トラフィックの URL 分類ポリシーを追加します。
- HTTPS トラフィックの URL 分類ポリシーを追加します。
- HTTP トラフィック用の負荷分散仮想サーバーを追加します。
- HTTPS トラフィック用の SSL ブリッジ負荷分散仮想サーバーを追加します。
- URL 分類ポリシーを負荷分散仮想サーバーにバインドします。
- URL 分類ポリシーを SSL ブリッジ負荷分散仮想サーバーにバインドします。
- 共有メモリ制限を設定します。
- URL 分類パラメータを設定します。

URL 分類を有効にするには

1. ナビゲーションペインで [システム] を展開し、[設定] をクリックします。
2. 「設定」ページで、「拡張機能の設定」リンクをクリックします。
3. 「拡張機能の設定」ページで、「**URL** フィルタリング」チェックボックスを選択します。
4. 「**OK**」をクリックして「閉じる」をクリックします。

URL 分類アクションを追加するには

1. ナビゲーションペインで、AppExpert**>** レスポンダー > アクションを展開します。
2. 詳細ペインで、[追加] をクリックします。
3. 「レスポンダーアクションの作成」ページで、次のパラメーターを設定します。

    a) **Name**: URL 分類ポリシーアクションの名前。
    b) 種類。アクションタイプを選択します。

c) **Expression**：エクスプレッションエディタを使用してポリシーエクスプレッションを作成します。
d) [コメント**]**。ポリシーアクションの簡単な説明。

4. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTP トラフィックの **URL** 分類ポリシーを追加するには

1. ナビゲーションペインで、AppExpert**>** レスポンダー > ポリシーを展開します。
2. 詳細ウィンドウで、[追加] をクリックします。
3. **[**レスポンダーポリシーの作成] ページで、次のパラメータを設定します。
   1.**Name**：URL 分類ポリシーアクションの名前。
   2. 操作。ポリシーに関連付けたい URL 分類アクションを選択します。
   3. ログアクション。ログアクションを選択します。
   4.**AppFlow**。AppFlow アクションを選択します。
   5.**Expression**：エクスプレッションエディタを使用してポリシーエクスプレッションを作成します。
      a) [コメント**]**。ポリシーアクションに関する簡単な説明。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTPS トラフィックの分類ポリシーを追加するには

1. **NetScaler** アプライアンスにログオンし、**[**** 構成] > [最適化] > [****** ビデオ最適化 ******] > [検出] に移動します。**
2. 検出ページで、「ビデオ最適化検出ポリシー」リンクをクリックします。
3.「ビデオ最適化検出ポリシー」ページで、「追加」をクリックします。
4. ビデオ最適化検出ポリシーの作成ページで、次のパラメータを設定します。
   a) **Name**：最適化ポリシーの名前
   b) **Expression**：カスタム表現を使用してポリシーを設定します。
   c) アクション。着信ビデオトラフィックを処理するポリシーに関連付けられた最適化アクション。
   d) **UNDEF** アクション。着信要求が最適化ポリシーと一致しない場合の未定義のイベント。
   e) [コメント]。ポリシーに関する簡単な説明。
   f) ログアクション。ログメッセージに対して実行するアクションを指定する監査ログアクションを選択します。
5. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTP トラフィック用の負荷分散仮想サーバーを追加するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動します。
2. 詳細ペインで、[追加] をクリックします。

3.「仮想サーバーの負荷分散」ページで、次のパラメーターを設定します。

   a) **Name**：負荷分散仮想サーバーの名前。
   b) プロトコル。プロトコルタイプを HTTP として選択します。
   c) **IP** アドレスタイプ。IPv4 または IPv6。
   d) **IP** アドレス。IPv4 または IPv6、仮想サーバーに割り当てられた VIP アドレス。
   e) ポート。仮想サーバーのポート番号。

4.「**OK**」をクリックして、その他のオプションのパラメータの設定を続行します。
5. **[**作成**]** して **[**閉じる**]** をクリックします。

SSL ブリッジ負荷分散仮想サーバーを追加するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動します。
2. 詳細ウィンドウで、[追加] をクリックします。
3. 負荷分散仮想サーバーページで、次のパラメーターを設定します。

   a) **Name**：負荷分散仮想サーバーの名前。
   b) プロトコル。プロトコルタイプとして [SSL ブリッジ] を選択します。
   c) **IP** アドレスタイプ。IP アドレス指定可能なタイプ。
   d) **IP** アドレス。仮想サーバーに割り当てられた IP 4 または IP6 IP アドレス。
   e) ポート。仮想サーバーのポート番号。

4. **OK** を選択して他のオプションパラメータの設定を続行します。
5. [作成] をクリックし、[閉じる] をクリックします。

URL 分類ポリシーを **HTTP** 負荷分散仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動します。
2. 詳細ペインで、負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [詳細設定] セクションで、[ポリシー] をクリックします。
4. [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
5. 次のパラメータを設定します。

   a)「ポリシー」を選択します。ドロップダウンリストから URL 分類ポリシーを選択します。
   b)「タイプ」を選択します。ポリシータイプとして [要求] を選択します。

6.［続行］をクリックします。
7. リストから URL 分類ポリシーを選択し、「閉じる」をクリックします。

分類ポリシーを **SSL** ブリッジ負荷分散仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] 画面に移動します。

2. 詳細ウィンドウで、SSL ブリッジ負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [詳細設定] セクションで、[ポリシー] をクリックします。
4.「ポリシー」セクションで、「**+** 」アイコンをクリックして「ポリシー」スライダーにアクセスします。
5. [**Policies**] セクションで、次のパラメータを設定します。

    a)「ポリシー」を選択します。ドロップダウンリストからビデオ検出ポリシーを選択します。
    b)「タイプ」を選択します。ポリシータイプとして [要求] を選択します。

6. [続行] をクリックします。
7. リストからビデオ検出ポリシーを選択し、[閉じる] をクリックします。

共有メモリ制限を設定するには

1. アプライアンスにサインオンし、[最適化] > [統合キャッシュ] に移動します。
2. 詳細ペインで、「キャッシュ設定の変更」リンクをクリックします。
3. キャッシュグローバル設定ページで、次のパラメータを設定します。

    a) メモリ使用量の上限 **(MB)**。
    b) アクティブメモリ使用量の上限。
    c) ヘッダー経由。
    d) キャッシュされる投稿本文の最大長
    e) グローバル未定義結果アクション
    f) **HA** オブジェクト永続化を有効にする
    g) キャッシュされたオブジェクトが存続することを確認
    h) プリフェッチ

4.「**OK**」をクリックして「閉じる」をクリックします。

URL 分類パラメータを設定するには

1. アプライアンスにサインインし、セキュリティに移動します。
2. 詳細ウィンドウで、[**URL** フィルタリング設定の変更] リンクをクリックします。
3. **URL** フィルタリングパラメータの設定ページで、次のパラメータを設定します。

    a) DB 更新間隔の時間。URL データベース更新の間隔をフィルタリングする時間。最小値:0、最大値:720。
    b) DB を更新する時刻。URL フィルタリングによるデータベース更新の時間帯。

4.「**OK**」をクリックし、「閉じる」をクリックします。

監査ログメッセージの設定

NetScaler アプライアンスが受信 URL を受信すると、レスポンダーポリシーに URL フィルタリング式がある場合、監査ログ機能は分類情報を収集し、構成されている任意のターゲット監査ログサーバーにログメッセージとして表示

します。情報がログに記録されます。

- 送信元 IP アドレス (要求を行ったクライアントの IP アドレス)。
- 宛先 IP アドレス（要求されたサーバの IP アドレス)。
- スキーマ、ホスト、およびドメイン名 (`http://www.example.com`) を含むリクエストされた URL。
- URL フィルタリングフレームワークが返す URL カテゴリ。
- URL フィルタリングフレームワークが返した URL カテゴリグループ。
- URL フィルタリングフレームワークが返した URL レピュテーション番号。
- URL 分類ポリシーによって実行された監査ログアクション。

URL リスト機能の監査ログを設定するには、次の作業を完了する必要があります。

1. 監査ログを有効化。
2.「監査ログの作成」メッセージアクション。
3. [監査ログメッセージ] アクションで URL リストレスポンダポリシーを設定します。

詳細については、「監査ログ 」トピックを参照してください。

SYSLOG メッセージを使用した障害エラーの保存

URL フィルタリングプロセスのどの段階でも、システムレベルの障害が発生した場合、NetScaler アプライアンスは監査ログメカニズムを使用して ns.log ファイルにログを保存します。エラーは SYSLOG 形式でテキストメッセージとして保存されるため、管理者は後でイベント発生の時系列順に表示できます。これらのログは、アーカイブのために外部 SYSLOG サーバにも送信されます。詳細については、CTX229399 を参照してください。

たとえば、URL フィルタリング SDK を初期化するときにエラーが発生した場合、エラーメッセージは次のメッセージ形式で格納されます。

Oct 3 15:43:40 <`local0.err`> ns URLFiltering[1349]: Error initializing NetStar SDK (SDK error=-1). (status=1).

NetScaler ADC アプライアンスは、4 つの異なる障害カテゴリにエラーメッセージを格納します。

- ダウンロードに失敗しました。分類データベースをダウンロードしようとしたときにエラーが発生した場合。
- 統合に失敗しました。更新を既存の分類データベースに統合するときにエラーが発生した場合。
- 初期化に失敗しました。URL 分類機能の初期化時にエラーが発生した場合は、分類パラメータを設定するか、分類サービスを終了してください。
- 検索に失敗しました。アプライアンスがリクエストの分類の詳細を取得するときにエラーが発生した場合。

URL レピュテーションスコア

URL 分類機能は、ブラックリストに登録された URL を制限するポリシーベースの制御を提供します。URL カテゴリ、レピュテーションスコア、または URL カテゴリとレピュテーションスコアに基づいて、Web サイトへのアクセ

スを制御できます。ネットワーク管理者がリスクの高いウェブサイトにアクセスするユーザーを監視する場合、URL レピュテーションスコアにバインドされたレスポンダーポリシーを使用して、そのようなリスクのある Web サイトをブロックできます。

着信 URL 要求を受信すると、アプライアンスは URL 分類データベースからカテゴリおよびレピュテーションスコアを取得します。データベースから返されたレピュテーションスコアに基づいて、アプライアンスは Web サイトにレピュテーションレーティングを割り当てます。値の範囲は 1 ～4 です。次の表に示すように、4 は最もリスクのある Web サイトのタイプです。

URL レピュテーション評価	レピュテーションコメント
1	クリーンサイト。
2	未知のサイト。
3	潜在的に危険または危険なサイトに関連している。
4	悪質なサイト。

よくある質問

August 15, 2023

このセクションでは、次の **NetScaler ADC** 機能に関する **FAQ** について説明します

- 管理パーティション
- AppFlow
- Call Home
- Clustering
- 接続管理
- コンテンツスイッチ
- Debugging
- Hardware
- 高可用性
- 統合キャッシング
- インストール、アップグレード、およびダウングレード
- 負荷分散
- NetScaler GUI
- SSL

管理パーティション

August 15, 2023

パーティションの **NetScaler** 構成ファイルはどこで入手できますか**?**

デフォルトパーティションの構成ファイル *(*ns.conf*)* は */nsconfig* ディレクトリにあります。 <partitionName> 管理パーティションの場合、ファイルは */nsconfig/partitions/* ディレクトリにあります。

パーティション化された **NetScaler** アプライアンスで統合キャッシュを構成する方法を教えてください

> 注
>
> 管理パーティションの統合キャッシュは、NetScaler 11.0 以降でサポートされています。

パーティション化された NetScaler に統合キャッシュ（IC）を構成するには、デフォルトパーティションに IC メモリを定義した後、スーパーユーザーは、すべての管理パーティションに割り当てられる IC メモリの合計がデフォルトパーティションで定義されている IC メモリを超えないように、各管理パーティションの IC メモリを構成できます。管理パーティション用に設定されていないメモリは、デフォルトパーティションで引き続き使用できます。

たとえば、2 つの管理パーティションを持つ NetScaler アプライアンスのデフォルトパーティションに 10 GB の IC メモリが割り当てられていて、2 つの管理パーティションの IC メモリ割り当てが次のようになっているとします。

- パーティション 1:4 GB
- パーティション 2:3 GB

そして、デフォルトのパーティションには 10-(4+3) = 3 GB の IC メモリが使用可能になっています。

> 注
>
> すべての IC メモリが管理パーティションで使用されている場合、デフォルトパーティションで使用できる IC メモリはありません。

管理パーティションの **L2** および **L3** パラメータの範囲はどのようになっていますか**?**

> 注
>
> - NetScaler 11.0 以降に適用されます。
> - ARP をデフォルト以外のパーティションで動作させるには、「set l2param」コマンドで「ProxyArp」パラメータを有効にする必要があります。

パーティション化された NetScaler アプライアンスでは、L2 および L3 パラメーターの更新範囲は次のとおりです。

- 「set L2Param」コマンドを使用して設定された L2 パラメータの場合、次のパラメータはデフォルトパーティションからのみ更新でき、その値はすべての管理パーティションに適用されます。

  Max Bridge Collision、BDG 設定、GARPonVRIDINTF、GARP 応答、プロキシ ARP、HA フェイルオーバー時のインターフェイスリセット、skip_proxying_bsd_traffic。

  その他の L2 パラメータは特定の管理パーティションで更新でき、その値はそれらのパーティションに固有です。

- 「set L3Param」コマンドを使用して設定された L3 パラメータでは、すべてのパラメータを特定の管理パーティションで更新でき、その値はそれらのパーティションに固有です。同様に、デフォルトパーティションで更新された値は、デフォルトパーティションにのみ適用されます。

管理パーティションで動的ルーティングを有効にする方法は?

> 注
>
> 管理パーティションの動的ルーティングは、NetScaler 11.0 以降でサポートされています。

動的ルーティング（OSPF、RIP、BGP、ISIS、BGP+）はデフォルトパーティションでデフォルトで有効になっていますが、管理パーティションでは、次のコマンドを使用して有効にする必要があります。

```
> set L3Param -dynamicRouting ENABLED
```

> 注
>
> 最大 63 のパーティションで動的ルーティングを実行できます（62 の管理パーティションと 1 つのデフォルトパーティション）。

管理パーティションで動的ルーティングを有効にすると、仮想ルーター (VR) が作成されます。

- <partition-name> 各 VR には独自の vlan0 が保持され、vlan0_ と表示されます。
- ZeBoS に公開されているバインドされていない IP アドレスはすべて vlan0 にバインドされます。
- （デフォルトパーティションの）デフォルト VR には、設定されているすべての VR が表示されます。
- デフォルト VR には、これらの VR にバインドされている VLAN が表示されます（デフォルト VLAN は除く）。

パーティションのログはどこで確認できますか?

NetScaler のログはパーティション固有ではありません。すべてのパーティションのログエントリは */var/log/* ディレクトリに保存する必要があります。

管理パーティションの監査ログを取得するにはどうすればよいですか?

パーティション化された NetScaler では、特定のパーティションに特定のログサーバーを設定することはできません。デフォルトパーティションで定義されたサーバは、すべての管理パーティションに適用されます。したがって、特定のパーティションの監査ログを表示するには、「show audit messages」コマンドを使用する必要があります。

> 注
>
> 管理パーティションのユーザーはシェルにアクセスできないため、ログファイルにアクセスできません。

管理パーティションのウェブログを取得するにはどうすればいいですか?

管理パーティションのウェブログは、次のように取得できます。

- **NetScaler 11.0** 以降のバージョンの場合

  Web ロギング機能を Web ロギングを必要とする各パーティションで有効にする必要があります。NetScaler Web ロギング（NSWL）クライアントを使用して、NetScaler はユーザーが関連付けられているすべてのパーティションの Web ログを取得します。

- **NetScaler 11.0** より前のバージョンの場合

  Web ログは `nsroot` 、他のスーパーユーザーのみが取得できます。また、デフォルトパーティションで Web ログが有効になっていても、NetScaler Web Logging（NSWL）クライアントはすべてのパーティションの Web ログを取得します。

各ログエントリのパーティションを表示するには、%P オプションを含むようにログ形式をカスタマイズします。その後、ログをフィルタリングして、特定のパーティションのログを表示できます。

管理パーティションのトレースを取得するにはどうしたらいいですか?

管理パーティションのトレースは次のように取得できます。

- **NetScaler 11.0** 以降のバージョンの場合

  パーティション化された NetScaler アプライアンスでは、`nstrace` 個々の管理パーティションで操作を実行できます。<partitionName> トレースファイルは*/var/partitions/ /nstrace/*ディレクトリに保存されます。

  注**:** GUI を使用して管理パーティションのトレースを取得することはできません。CLI を使用する必要があります。

- **NetScaler 11.0** より前のバージョンの場合

  `nstrace` この操作はデフォルトパーティションでのみ実行できます。そのため、パケットキャプチャは NetScaler システム全体で利用できます。パーティション固有のパケットキャプチャを取得するには、VLAN-ID ベースのフィルターを使用してください。

管理パーティション固有のテクニカルサポートバンドルはどうすれば入手できますか?

特定のパーティションのテクニカルサポートバンドルを取得するには、デフォルトパーティションから次のコマンドを実行します。


```
> show techsupport -scope partition <partitionName>
```

注**:** このコマンドは、システム固有の情報も提供します。

AppFlow

August 15, 2023

- **AppFlow** をサポートしている **NetScaler** のビルドはどれですか**?**

  AppFlow は、nCore ビルドのバージョン 9.3 以降を実行している NetScaler アプライアンスでサポートされています。

- **AppFlow** がデータを送信するために使用する形式を教えてください。

  AppFlow は、RFC 5101 で定義されているオープンインターネットエンジニアリングタスクフォース（IETF）標準であるインターネットプロトコルフロー情報エクスポート（IPFIX）形式で情報を送信します。IPFIX（Cisco 社製 NetFlow の標準化バージョン）は、ネットワークフロー情報を監視するために幅広く使用されています。

- **AppFlow** レコードには何が含まれていますか**?**

  AppFlow レコードには、フローの開始と終了のタイムスタンプ、パケットカウント、バイトカウントなどの標準的な NetFlow または IPFIX 情報が含まれます。AppFlow レコードには、アプリケーションレベルの情報（HTTP URL、HTTP リクエストメソッドと応答ステータスコード、サーバーの応答時間、待ち時間など）も含まれます。IPFIX フローレコードはテンプレートに基づいており、フローレコードを送信する前に送信する必要があります。

- **NetScaler** バージョン **9.3** ビルド **48.6 CL** にアップグレードした後、**GUI** から仮想サーバーを開こうとすると、「**AppFlow** 機能は **NetScaler Ncore** でのみ使用できます」というエラーメッセージが表示されるのはなぜですか?

  AppFlow は、nCore アプライアンスでのみサポートされています。仮想サーバー構成タブを開いたら、[**AppFlow**] チェックボックスをオフにします。

- **AppFlow** レコードのトランザクション **ID** には何が含まれていますか**?**

  トランザクション ID は、アプリケーションレベルのトランザクションを識別する符号なし 32 ビットの数値です。HTTP の場合、トランザクションは要求と応答のペアに対応します。この要求と応答のペアに対応するすべてのフローレコードは、同じトランザクション ID を持ちます。一般的なトランザクションには 4 つのフローレコードがあります。NetScaler ADC が単独で応答を生成する場合（統合キャッシュまたはセキュリティポリシーによって提供される）、トランザクションには 2 つのフローレコードしか存在しない可能性があります。

- **AppFlow** アクションとは何ですか**?**

AppFlow アクションは、関連付けられた AppFlow ポリシーが一致した場合にフローレコードが送信されるコレクタのセットです。

- **AppFlow** アクションがヒットしたことを確認するために、**NetScaler ADC** アプライアンスでどのようなコマンドを実行できますか**?**

  AppFlow を表示アクション。次に例を示します:

```
> show appflow action
1) Name: aFL-act-collector-1
   Collectors: collector-1
   Hits: 0
   Action Reference Count: 2
 2) Name: apfl-act-collector-2-and-3
   Collectors: collector-2, collecter-3
   Hits: 0
   Action Reference Count: 1
 3) Name: apfl-act-collector-1-and-3
   Collectors: collector-1, collecter-3
   Hits: 0
   Action Reference Count: 1
<!--NeedCopy-->
```

- **AppFlow** コレクターとは何ですか**?**

コレクターは、NetScaler アプライアンスによって生成されたフローレコードを受信します。フローレコードを送信するには、少なくとも 1 つのコレクターを指定する必要があります。最大 4 つまで指定できます。未使用のコレクターは削除できます。

- **AppFlow** を使用するにはどの **NetScaler** バージョンが必要ですか?

  NetScaler バージョン 9.3.49.5 以降を使用してください。AppFlow は nCore ビルドでのみ利用できることに注意してください。

- **AppFlow** はどのトランスポートプロトコルを使用していますか**?**

  AppFlow はトランスポートプロトコルとして UDP を使用します。

- ネットワークにファイアウォールがある場合、どのポートを開く必要がありますか**?**

  ポート 4739。これは、AppFlow コレクタが IPFIX メッセージをリッスンするために使用するデフォルトの UDP ポートです。ユーザーがデフォルトポートを変更した場合、そのポートをファイアウォールで開く必要があります。

- **AppFlow** が使用するデフォルトのポートを変更するにはどうすればよいですか**?**

  add AppFlowCollector コマンドを使用して AppFlow コレクターを追加する場合、使用するポートを指定できます。

```
> add appflowCollector coll1 -IPAddress 10.102.29.251 -port 8000
 Done
<!--NeedCopy-->
```


- **ClientTrafficOnly** を設定すると何が起こりますか**?**

  NetScaler は、クライアント側のトラフィックについてのみ AppFlow レコードを生成します。

- 一度に設定できるコレクタの数はいくつですか。

  NetScaler ADC アプライアンスでは、一度に最大 4 つの AppFlow コレクタを構成できます。NetScaler ADC アプライアンスで構成できるコレクタの最大数は 4 台であることに注意してください。

Call Home

August 15, 2023

- **NetScaler** アプライアンスの **Call Home** とは何ですか**?**

  Call Home は、NetScaler アプライアンス上の重要なイベントを監視して通知します。Call Home を有効にすると、エラー通知プロセスを自動化できます。NetScaler サポートが問題をトラブルシューティングする前に、NetScaler サポートへの電話を避け、サービスリクエストを送信し、システムデータをアップロードできるだけでなく、問題が発生する前に問題を特定して解決できます。

- **NetScaler** アプライアンスでは **Call Home** がデフォルトで有効になっていますか?

  はい。アプライアンスでは Call Home がデフォルトで有効になっています。Call Home がデフォルトで無効になっていた古いバージョンから最新のソフトウェアにアップグレードすると、アップグレードプロセスによってその機能が自動的に有効になります。後で無効にすると、更新後のすべてのアップグレードで更新された設定が記憶されます。詳細については、「Call Home」を参照してください。

- **Call Home** が機能するための前提条件を教えてください。

  インターネット接続へのアクセス。

  注：NetScaler アプライアンスにインターネット接続がない場合は、NetScaler がシステムログを生成して Citrix テクニカルサポートサーバー（CIS）にアップロードできるプロキシサーバーを構成できます。

- **Call Home** を使用するメリットは何ですか。

  - ハードウェアおよびソフトウェアのエラー状態を監視します。
  - ネットワークに影響を与える重要なイベントの発生について通知します。
  - パフォーマンスデータとシステムログを Citrix に送信して、以下を行います。
    * 製品の品質を分析し、改善します。
    * リアルタイムのトラブルシューティング情報を提供して、問題をプロアクティブに特定し、問題を迅速に解決します。

- **Call Home** をサポートしている **NetScaler** ソフトウェアのリリースはどれですか?

  NetScaler リリース 10.0 以降。

- どの **NetScaler** プラットフォームモデルが **Call Home** をサポートしていますか?

  Call Home 機能は、すべての NetScaler プラットフォームとすべてのアプライアンスモデル（MPX、VPX、SDX）でデフォルトで有効になっています。

  - NetScaler MPX: すべての MPX モデル。
  - NetScaler VPX: すべての VPX モデル。また、外部または中央のライセンスプールからライセンスを取得する VPX アプライアンスでもサポートされます。ただし、この機能は標準 VPX アプライアンスの場合と同じままです。
  - NetScaler SDX: ディスクドライブと割り当てられた SSL チップを監視し、エラーや障害がないか確認します。ただし、VPX インスタンスは電源ユニット (PSU) にアクセスできないため、ステータスは監視されません。SDX プラットフォームでは、Call Home を個々のインスタンスに直接設定することも、SVM を介して設定することもできます。

- **Call Home** にエラー状態を通知するように **SNMP** アラームを設定すべきですか。

  いいえ。SNMP と Call Home のアップロードは互いに独立しているため、エラー状態を監視するために Call Home に SNMP を設定する必要はありません。エラー状態が発生するたびに通知を受け取りたい場合は、CALLHOME-UPLOAD-EVENT SNMP アラームを設定して、Call Home のアップロードが発生するたびに SNMP アラートを生成できます。SNMP アラートは、重大なイベントの発生をローカル管理者に通知します。

- テクニカルサポートに連絡するにはどうすればいいですか?

  すべての重要なハードウェア関連イベントについて、Call Home は NetScaler へのサービスリクエストを自動的に作成します。その他のエラーについては、システムログを確認した後、NetScaler テクニカルサポートチームに連絡して、詳細な調査を依頼できます。サポートに連絡するには、https://www.netscaler.com/resources/supportをご覧ください。

- **Call Home** は **NetScaler ADC** アプライアンスでどのようなエラー状態を監視しますか**?**

  Call Home は、NetScaler アプライアンス内の次のイベントの監視をサポートしています。

  - コンパクトフラッシュドライブのエラー
  - ハードディスクドライブのエラー
  - 電源装置に障害が発生しました
  - SSL カード障害
  - ウォームリスタート
  - メモリアノマリー
  - レート制限ドロップ

- **Call Home** には別途ライセンスが必要ですか**?**

  いいえ、Call Home には個別のライセンスは必要ありません。すべての NetScaler プラットフォームライセンスで有効にできます。

- **Call Home** はどのようなデータを **NetScaler** サポートサーバーに送信し、どのくらいの頻度で送信されますか?

Call Home は、2 種類のデータを収集し、CIS に送信します。これには、次の種類のアカウントがあります。

- 基本システム情報（実行中の NetScaler バージョン、展開モード（スタンドアロン、HA、クラスタ）、ハードウェアの詳細など）。Call Home 登録時および定期的なハートビートの一部として送信されます。ハートビートは 30 日に 1 回送信されますが、この間隔は 1 ～30 日の間で設定できます。ただし、頻繁なアップロードはあまり役に立たないため、5 日未満の値は推奨されません。
- エラー状態がある場合の `show tech support bundle` の短縮版。アプライアンスが最後に起動されてから、特定のエラー状態が最初に発生したときに送信されます。つまり、同じエラー状態が再発生しても、前回の発生後にアプライアンスが再起動されない限り、別のアップロードはトリガーされません。

- **Call Home** は、プロキシサーバを介してシステムログを生成およびアップロードできますか。

  はい。NetScaler アプライアンスに直接インターネット接続がない場合は、プロキシサーバーを構成し、システムログを Citrix テクニカルサポートサーバー（CIS）にアップロードできます。

- **Call Home** データを **CIS** に送信する前に確認できますか。

  申し訳ございませんが、CIS に送信される前に Call Home データを確認することはできません。Call Home は、NetScaler サポートチームに連絡する際に提供するデータ以外のデータを収集しません。

- **Call Home** のアップロードの安全性とプライベート性はどれくらいですか。

  Call Home は、次の方法でデータセキュリティとプライバシーを提供します。

  - 安全な SSL/TLS チャネルを使用して Citrix サーバーにデータを転送します。
  - アップロードされたデータは、権限のある担当者のみがレビューし、第三者と共有することはありません。

クラスタリング

August 15, 2023

クラスタリングに関する FAQ については、ここをクリックしてください 。

接続管理

August 15, 2023

- 管理者接続とは**?**

  管理者接続は NSIP アドレスへの接続を確立し、管理者が NetScaler アプライアンスを構成および監視できるようにします。

- 管理者接続にはどのような種類がありますか**?**

  管理接続には、次の 2 種類があります。

  - SSH 接続—管理者ユーザーは SSH クライアントを使用して NSIP アドレスを介してログオンします。
  - NITRO API 接続—管理者ユーザーは、NITRO API を使用して、NetScaler ADC アプライアンスへのログオンプロセスを自動化します。

> 注
>
> 管理者ユーザーは、ブラウザを使用して NSIP アドレスに接続することで、GUI からログオンしてログオンすることもできます。GUI は内部的に NITRO API 接続を開きます。したがって、GUI セッションは NITRO API 接続と同等であり、NITRO API に関連する FAQ は GUI に適用されます。

- **NetScaler ADC** アプライアンスで許可される管理接続の数はいくつですか**?**

  アプライアンスでは、最大 20 の同時管理接続が可能です。

- 管理者ログオンにはどのログイン認証情報が必要ですか**?**

  管理者ログオンには、ユーザー名とパスワードが必要です。

  注**:** パスワードの代わりに認証キーを使用できます。

- **NetScaler** アプライアンスはどの外部認証方法をサポートしていますか?

  アプライアンスは、次の外部認証方法をサポートしています。

  - RADIUS
  - LDAP
  - TACACS

- クライアントとは**?**

  クライアントとは、管理者ユーザーが管理者接続を開くために使用するデバイス (ラップトップまたはデスクトップ) です。

- セッショントークンとは何ですか**?**

  セッショントークンは、NetScaler アプライアンスが NITRO API ログオンリクエストを送信するクライアントに発行する固有の識別子です。

  - API クライアントは、セッショントークンが期限切れでない場合、新しい TCP 接続での後続の API 要求に再利用できます。
  - GUI クライアントは内部的に NITRO API 接続を開き、GUI セッション中はセッショントークンをアクティブに保ちます。

- **NetScaler ADC** アプライアンスのアクティブセッションとは何ですか**?**

  セッションの有効期限が切れておらず、NetScaler アプライアンスとの SSH 接続が開いている場合、CLI セッションはアクティブと見なされます。

NetScaler アプライアンスのセッショントークンのタイムアウト期限が切れていない場合、NITRO API セッションはアクティブと見なされます。

- **NetScaler** は同時接続制限をどのように適用しますか**?**

  NetScaler アプライアンスは管理者接続要求（SSH または NITRO API）を受信するたびに、開いている管理者接続の数を確認します。数値が 20 未満の場合、新しい接続が開かれます。

- **NetScaler** アプライアンスの管理接続数を反映しているカウンターはどれですか**?**

  接続カウンター (nsconfigd_cur_clients) には、アクティブな接続の数が反映されます。このカウンタは、クライアントがアプライアンスへの新しい接続を開くと増加し、接続が閉じられると減少します。

- **NetScaler ADC** アプライアンス上のアクティブなトークンの数を反映するカウンタはどれですか**?**

  config_cur_tokens カウンタは、NetScaler ADC アプライアンス上のアクティブなトークンの数を反映します。

- **NetScaler ADC** アプライアンスは接続時のエラーをどのように処理しますか**?**

  NetScaler アプライアンスは、接続でエラーが発生した場合、クライアント（CLI、API、GUI）接続を直ちに閉じます。

- 管理アドレスへの接続での **CLI** または **GUI** セッションは、管理接続制限にカウントされますか。

  はい。CLI 接続と GUI 接続はすべて TCP ベースの接続であり、管理アドレスへのすべての TCP 接続は管理接続制限にカウントされます。

- **NITRO** セッションは管理接続制限にカウントされますか**?**

  NetScaler アプライアンスが発行したセッショントークンを使用して TCP 接続が開いている場合、NITRO セッションは管理接続制限にカウントされます。

- **NetScaler ADC** アプライアンスでの **API**、**GUI**、**CLI** セッションのデフォルトのタイムアウト期間はどれくらいですか**?**

  次の表に、NetScaler ADC アプライアンスでの API、GUI、および CLI セッションのデフォルトのタイムアウト期間を示します。

NetScaler のリリース	CLI のデフォルトタイムアウト期間 (分)	API のデフォルトタイムアウト期間 (分)	GUI のデフォルトタイムアウト期間 (分)
NetScaler 9.3	なし	30 分	30 分
NetScaler 10.1	なし	30 分	30 分
NetScaler 10.5 以降	15 分	30 分	15 分

- **NetScaler ADC** アプライアンスで **CLI** セッションのタイムアウトを設定するにはどうすればよいですか**?**

  CLI セッションタイムアウトを設定するには、CLI プロンプトで次のコマンドを実行します。


```
set cli mode -timeout \<xx seconds>
```

- **NITRO API** を使用する場合、デフォルトのタイムアウト期間をどのように上書きしますか**?**

  ログインオブジェクトの「タイムアウト」フィールドにタイムアウト時間を設定することで、NITRO API のデフォルトのタイムアウト期間をオーバーライドできます。セッションタイムアウトがゼロに設定されている場合、セッショントークンのタイムアウトは無限になります。

  注: タイムアウトしないセッションは引き続き管理接続数にカウントされるため、無限タイムアウトはお勧めできません。

- 管理セッションの作成後に **NetScaler ADC** アプライアンスからユーザーアカウントを削除するとどうなりますか**?**

  内部システムユーザーの場合、NetScaler ADC アプライアンスは既存の CLI または NITRO API セッションを閉じます。

  外部システムユーザの場合、セッションは有効期限が切れるまでアクティブなままです。

- **NITRO API** クライアントは、単一のセッショントークンを使用して、**NetScaler ADC** アプライアンスで複数の管理接続を開くことができますか**?**

  はい。このような接続はそれぞれ、管理接続制限にカウントされます。

- **SNIP** アドレスに対して管理アクセスが有 効になっている場合、そのアドレスへの管理者接続は、管理者接続数の制限に対してカウントされますか**?**

  はい、管理アドレス（SNIP）への管理者接続は、NetScaler 管理者接続制限に対してカウントされます。

- 最大接続数に達した後、**NetScaler ADC** 管理者が **NetScaler ADC** アプライアンスにログオンすることはできますか**?**

  はい。最大接続制限に達すると、もう 1 つの管理者接続が許可されます。

- **NITRO API** エンドポイントは、アプライアンスの **NetScaler ADC** 上で複数の管理接続を開くことができますか**?**

  はい。NITRO API エンドポイントは、複数の管理接続を開いて、NetScaler ADC アプライアンスの同時管理接続制限を使い果たすことができます。このような状況では、追加の SSH/CLI 接続が許可され、管理者は古い API セッションの強制終了や、既存の API セッションのセッションタイムアウト時間を短縮できます。

- 同じクライアントで **NetScaler ADC** アプライアンスで複数の **API** セッションを開くことはできますか**?**

  はい。クライアントは繰り返しログオンすることで、複数の API セッションを開くことができます。たとえば、クライアントは再起動後に再びログオンすることがあります。
  注：クライアントに繰り返しログオンすると、NetScaler アプライアンスの管理接続制限にカウントされます。

- **API** クライアントは **API** セッショントークンの制限全体を使用できますか**?**

  はい。API クライアントは、以前に発行されたトークンを使用せずに繰り返しログオンすることで提供される API セッショントークンの制限をすべて使用できます。

注: クライアントのセッションタイムアウトがゼロの場合、トークンは永久に有効です。新しいセッショントークンを使用して繰り返しログオンすると、API セッショントークンの制限にカウントされる可能性があります。

- **CLI** セッションは **API** セッショントークンの制限にカウントされますか**?**

  いいえ。CLI セッションは API セッショントークンの制限にはカウントされません。

- 管理者ユーザーは **telnet** を使用して **CLI** セッションを開くことができますか**?**

  いいえ。CLI セッションを開くことができるのは SSH クライアントだけです。

- さまざまな **NetScaler** リリースに適用される接続制限と **API** セッション制限はどのようになっていますか?

  次の表は、さまざまな NetScaler リリースに適用される最大同時管理接続数とアクティブな API セッション制限を示しています。

NetScaler のリリース	9.3	10.1 (130.x より前)	10.1 (130.10 より前)	10.1 (130.10 から)
同時管理接続の最大数	20	20	20	20
アクティブな API セッションの最大数 *	1000	20	1000	1000

注:

  - API セッションは、タイムアウトしていない場合はアクティブと見なされます。たとえば、500 の API セッションが作成され、100 が期限切れになった場合、400 の API セッションがアクティブになります。
  - API セッションでは、NetScaler ADC アプライアンスへの TCP 接続を開く必要はありません。

コンテンツスイッチ

August 15, 2023

- **NetScaler ADC** 以外の負荷分散アプライアンスをネットワークにインストールしました。ただし、**NetScaler ADC** アプライアンスのコンテンツスイッチング機能を使用して、クライアント要求を負荷分散アプライアンスに転送したいと思います。**NetScaler ADC** アプライアンスのコンテンツスイッチング機能を、**NetScaler ADC** 以外の負荷分散アプライアンスで使用することは可能ですか**?**

  はい。NetScaler ADC アプライアンスのコンテンツスイッチング機能は、NetScaler ADC アプライアンスまたは NetScaler ADC 以外の負荷分散アプライアンスの負荷分散機能で使用できます。ただし、NetScaler ADC 以外の負荷分散アプライアンスを使用する場合は、NetScaler ADC アプライアンスに負荷分散仮想サーバーを作成し、NetScaler ADC 以外の負荷分散アプライアンスにサービスとしてバインドしてください。

- コンテンツスイッチング仮想サーバーと負荷分散仮想サーバーはどのように異なりますか。

  コンテンツスイッチング仮想サーバーは、クライアント要求を他の仮想サーバーに送信することしかできません。サーバーとは通信しません。

  負荷分散仮想サーバーは、サーバー間でクライアントの負荷を分散し、サーバーと通信します。サーバーの可用性を監視し、さまざまな負荷分散アルゴリズムを適用してトラフィックの負荷を分散するために使用できます。

  コンテンツスイッチングは、仮想サーバーの負荷分散によって、特定の種類のコンテンツに対するクライアント要求をターゲットサーバーに送信するために使用される方法です。クライアント要求を処理するのに最適なサーバーにクライアント要求を送信できます。これにより、サーバー上のクライアント要求を処理するためのオーバーヘッドが削減されます。

- **NetScaler ADC** アプライアンスのコンテンツスイッチング機能を実装して、クライアント要求を指示したい。コンテンツスイッチング機能を使用して、どのような種類のクライアント要求を送信できますか。

  コンテンツスイッチング機能を使用すると、HTTP、HTTPS、FTP、TCP、セキュア TCP、および RTSP クライアント要求のみを転送できます。HTTPS クライアント要求を送信するには、アプライアンスで SSL オフロード機能を設定する必要があります。

- **NetScaler ADC** アプライアンスでコンテンツスイッチングルールを作成したい。コンテンツスイッチングルールを作成できるクライアント要求のさまざまな要素は何ですか。

  コンテンツスイッチングルールは、クライアント要求の次の要素とその値に基づいて作成できます。

  - URL
  - URL トークン
  - HTTP バージョン
  - HTTP ヘッダー
  - クライアントの送信元 IP アドレス
  - クライアントバージョン
  - 送信先 TCP ポート

- **NetScaler ADC** アプライアンスのコンテンツスイッチング機能がネットワークのパフォーマンスを向上させるのに役立つことを理解しています。これは正しい?

  はい。クライアントは、その処理に最も適したサーバーをクライアントから要求するように指示できます。その結果、サーバー上のクライアント要求を処理するためのオーバーヘッドが減少します。

- サイト管理性とクライアント要求に対する応答時間を向上させるために、**NetScaler ADC** アプライアンスで **NetScaler ADC** アプライアンスのどの機能を構成する必要がありますか?

  NetScaler ADC アプライアンスのコンテンツスイッチング機能を構成して、サイト管理性とクライアント要求に対する応答時間を向上させることができます。この機能を使用すると、同じドメイン名と IP アドレス内にコンテンツグループを作成できます。このアプローチは、ユーザーが参照できる異なるドメイン名や IP アドレスにコンテンツを明示的に分割する一般的なアプローチとは異なり、柔軟です。

ウェブサイトをさまざまなドメイン名と IP アドレスに分割する複数のパーティションにより、ブラウザはウェブページのコンテンツをレンダリングおよび取得するときに検出されたドメインごとに個別の接続を強制的に作成します。これらの追加の WAN 接続により、Web ページの応答時間が低下します。

- 私は **Web** サーバーファームでウェブサイトをホストしました。このタイプのセットアップでは、**NetScaler ADC** コンテンツスイッチング機能にはどのような利点がありますか**?**

  コンテンツスイッチング機能は、Web サーバーファームに基づくサイト内の NetScaler ADC アプライアンスに次の利点を提供します。

  - 同じドメインと IP アドレス内にコンテンツグループを作成して、サイトのコンテンツを管理します。
  - 同じドメインと IP アドレス内のコンテンツグループを使用して、クライアント要求に対する応答時間を短縮します。
  - ドメイン間での完全なコンテンツレプリケーションの必要性を回避します。
  - アプリケーション固有のコンテンツパーティション化を有効にします。たとえば、要求に応じて、動的コンテンツのみまたは静的コンテンツのみを処理するサーバーにクライアント要求を送信できます。
  - 同じサーバー上の複数のドメインのマルチホーミングをサポートし、同じ IP アドレスを使用します。
  - サーバーへの接続を再利用します。

- **NetScaler ADC** アプライアンスにコンテンツスイッチング機能を実装したい。各要求のさまざまなパラメータを評価した後、クライアント要求をさまざまなサーバーに送信します。コンテンツスイッチング機能を設定する場合、この設定を実装するにはどのようなアプローチが必要ですか。

  ポリシー式を使用して、コンテンツスイッチング機能のポリシーを作成できます。式は、演算子を使用してクライアント要求の修飾子をオペランドと比較することによって評価される条件です。クライアントリクエストの次のパラメータを使用して、式を作成できます。

  - メソッド-HTTP リクエストメソッド。
  - **URL**-HTTP ヘッダー内の URL。
  - **URL** トークン-URL 内の特別なトークン。
  - バージョン-HTTP リクエストのバージョン。
  - **URL** クエリ-URL クエリ LEN、URL LEN、および HTTP ヘッダーが含まれます。
  - **SOURCEIP**: クライアントの IP アドレス。

  次に、式の作成に使用できる演算子の完全なリストを示します。

  - == (equals)
  - != (not equals)
  - EXISTS
  - NOT EXISTS
  - CONTAINS
  - NOT CONTAINS
  - GT (greater than)
  - LT (less than)

また、一連の式を論理的に集約したさまざまなルールを作成することもできます。複数の式を組み合わせてルールを作成できます。式を結合するには、&& (AND) と

- 同じコンテンツスイッチング仮想サーバーに対して、ルールベースのポリシーと **URL** ベースのポリシーを構成します。同じコンテンツスイッチング仮想サーバーに対して、両方のタイプのポリシーを作成できますか。

  はい。同じコンテンツスイッチ仮想サーバーに対して、両方のタイプのポリシーを作成できます。ただし、優先順位を割り当てて、ポリシーの適切な優先順位を設定するようにしてください。

- **URL** のプレフィックスとサフィックスと共にドメイン名を評価し、それに応じてクライアント要求を指示するコンテンツスイッチングポリシーを作成します。どのタイプのコンテンツスイッチングポリシーを作成すべきですか。

  ドメインポリシーと正確な URL ポリシーを作成できます。このタイプのポリシーが評価されると、クライアント要求の完全なドメイン名と URL が構成されたものと一致する場合、NetScaler ADC アプライアンスはコンテンツグループを選択します。クライアント要求は、設定されたドメイン名と一致し、URL のプレフィックスとサフィックスが設定されている場合は、完全に一致する必要があります。

- **URL** の部分的なプレフィックスとサフィックスとともにドメイン名を評価し、それに応じてクライアント要求を指示するコンテンツスイッチングポリシーを作成します。どのタイプのコンテンツスイッチングポリシーを作成すべきですか。

  コンテンツスイッチング仮想サーバーのドメインおよびワイルドカード URL ポリシーを作成できます。このタイプのポリシーが評価されると、NetScaler ADC アプライアンスは、要求が完全なドメイン名と一致し、URL プレフィックスと部分的に一致する場合、コンテンツグループを選択します。

- ワイルドカード **URL** ポリシーとは何ですか**?**

  ワイルドカードを使用して、NetScaler ADC アプライアンスで構成した URL へのクライアント要求の部分的な URL を評価できます。ワイルドカードは、次のタイプの URL ベースのポリシーで使用できます。

  - プレフィックスのみ。たとえば、/sports/* 式は、/sports URL の下で使用可能なすべての URL に一致します。同様に、/sports * 式は、接頭辞が /sports であるすべての URL に一致します。
  - 接尾辞のみ。たとえば、/*.jsp 式は、ファイル名拡張子が jsp のすべての URL に一致します。
  - 接頭辞と接尾辞。たとえば、/sports/ *.jsp 式は、/sports/ URL の下にある jsp ファイル名拡張子を持つすべての URL に一致します。同様に、/sports *.jsp 式は、接頭辞 /sports * とファイル名拡張子が jsp のすべての URL に一致します。

- ドメインと規則のポリシーとは**?**

  ドメインとルールのポリシーを作成する場合、クライアント要求は、NetScaler ADC アプライアンスで構成されたドメインとルールと完全に一致する必要があります。

- ポリシーを評価するためのデフォルトの優先順位はどれくらいですか。

  デフォルトでは、ルールベースのポリシーが最初に評価されます。

- 一部のコンテンツがすべてのクライアント要求で同じ場合、ポリシーの評価にはどのような優先順位を使用すればよいですか**?**

  一部のコンテンツがすべてのユーザーで同じで、クライアント属性に基づいて異なるコンテンツを提供する必要がある場合は、ポリシー評価に URL ベースの優先順位を使用できます。

- コンテンツスイッチングでは、どのようなポリシー式構文がサポートされていますか**?**

  コンテンツスイッチングでは、次の 2 種類のポリシー式がサポートされています。

  - クラシック構文**-** コンテンツスイッチングの従来の構文は、キーワード REQ で始まり、詳細ポリシーよりも高度な構文です。クラシックポリシーはアクションにバインドできません。したがって、ターゲット負荷分散仮想サーバーは、コンテンツスイッチング仮想サーバーをバインドした後にのみ追加できます。
  - 高度なポリシー**:** 高度なポリシーは、通常 HTTP というキーワードで始まり、設定が簡単です。ターゲットの負荷分散仮想サーバーアクションを詳細ポリシーにバインドし、そのポリシーを複数のコンテンツスイッチ仮想サーバーで使用できます。

- 単一のコンテンツスイッチングポリシーを複数の仮想サーバーにバインドできますか**?**

  はい。定義されたアクションを含むポリシーを使用して、単一のコンテンツスイッチングポリシーを複数の仮想サーバーにバインドできます。ターゲット負荷分散仮想サーバーがコンテンツスイッチングポリシーで指定されなくなったため、アクションを使用するコンテンツスイッチングポリシーを複数のコンテンツスイッチング仮想サーバーにバインドできます。1 つのポリシーを複数のコンテンツスイッチング仮想サーバーにバインドする機能により、コンテンツスイッチング構成のサイズをさらに縮小できます。

  詳細については、以下の Knowledge Center 記事と NetScaler のドキュメントトピックを参照してください。

  - CTX122918- NetScaler ADC アプライアンス上の 2 つのコンテンツスイッチング仮想サーバーに同じコンテンツスイッチングポリシーをバインドする方法を参照してください。
  - CTX122736- ポリシーラベルを使用して同じ詳細ポリシーを複数のコンテンツスイッチング仮想サーバにバインドする方法を参照してください。
  - 基本的なコンテンツスイッチングの設定。

- 従来の式を使用してアクションベースのポリシーを作成できますか**?**

  いいえ。現在のところ、NetScaler ADC では、アクションで古典的な構文式を使用するポリシーはサポートされていません。ポリシーをバインドするときに、アクションで定義するのではなく、ターゲットの負荷分散仮想サーバーを追加する必要があります。

デバッグ

August 15, 2023

- 操作を実行したインターフェイス **(CLI**、**GUI**、または **API)** を確認する方法を教えてください。

  NetScaler は、操作が実行されるインターフェイスを追跡します。この情報は、syslog（GUI で [設定] > [システム] > [監査] > [監査メッセージ] > [Syslog メッセージ] に移動します）または ns.log ファイル（/var/log/ ディレクトリにあります）で表示できます。

  たとえば、API を介して実行される操作には、「API CMD_EXECUTED」というフラグが付けられます。

ハードウェア

August 15, 2023

MPX ハードウェアに関する FAQ については、ここをクリックしてください 。

高可用性

December 8, 2023

高可用性に関する FAQ については、ここをクリックしてください 。

統合キャッシング

August 15, 2023

コンテンツグループ

- **DEFAULT** コンテンツグループは他のコンテンツグループとどう違うのですか**?**

  DEFAULT コンテンツグループの動作は、他のグループと同じです。DEFAULT コンテンツグループを特別なものにしている唯一の属性は、オブジェクトがキャッシュされていてコンテンツグループが作成されていない場合です。オブジェクトは DEFAULT グループにキャッシュされます。

- コンテンツグループレベルの「キャッシュコントロール」オプションとは何ですか**?**

  任意のキャッシュ制御ヘッダーをブラウザに送信できます。コンテンツグループレベルのオプション-CacheControl を使用すると、ブラウザへの応答に挿入するキャッシュコントロールヘッダーを指定できます。

- コンテンツグループレベルの「**Minhit**」オプションとは何ですか**?**

  Minhitは、オブジェクトがキャッシュされる前のキャッシュポリシーの選択の最小数を指定する整数値です。この値は、コンテンツグループレベルで設定できます。CLI からこの値を設定する構文は次のとおりです。

  ```
  add/set cache contentGroup \<Content_Group_Name> [-minHits \<
  Integer>]
  ```

- **ExpireAtLastByte** オプションにはどのような用途がありますか**?**

  expireAtLastByte オプションを使用すると、統合キャッシュはダウンロード時にオブジェクトを期限切れにすることができます。その場合、未処理のリクエストのみがキャッシュから処理されます。新しいリクエストはすべてサーバーに送信されます。この設定は、株価情報の場合のようにオブジェクトが頻繁に変更される場合に便利です。この有効期限メカニズムは、フラッシュキャッシュ機能と連動します。ExpireAtLastByte オプションを設定するには、CLI から次のコマンドを実行します。

  ```
  add cache contentGroup \<Group_Name> -expireAtLastByte YES
  ```

キャッシュポリシー

- キャッシュポリシーとは

  ポリシーにより、キャッシュ可能なトランザクションとキャッシュできないトランザクションが決まります。また、ポリシーによって標準の HTTP キャッシュ動作が追加またはオーバーライドされます。ポリシーは、リクエストやレスポンスの特定の特性に応じて、CACHE や NOCACHE などのアクションを決定します。レスポンスがポリシールールと一致する場合、レスポンス内のオブジェクトは、ポリシーで設定されたコンテンツグループに追加されます。コンテンツグループを設定していない場合、オブジェクトは DEFAULT コンテンツグループに追加されます。

- ポリシーヒットとは

  選択は、リクエストまたはレスポンスがキャッシュポリシーと一致する場合に行われます。

- ミスとは?

  ミスは、リクエストまたはレスポンスがどのキャッシュポリシーとも一致しない場合に発生します。リクエストやレスポンスがキャッシュポリシーと一致していても、RFC の動作がオーバーライドされてオブジェクトがキャッシュに保存されない場合も、ミスが起こる可能性があります。

- **NetScaler** アプライアンスの統合キャッシュ機能を構成しました。次のポリシーを追加すると、エラーメッセージが表示されます。コマンドにエラーはありますか**?**

  ```
  add cache policy image_caching -rule exp1 | ns_ext_not_jpeg -
  action cache

  \> ERROR: No such command
  ```

  前述のコマンドでは、式は引用符で囲む必要があります。引用符がない場合、オペレータはパイプオペレータと見なされます。

メモリ要件

- **NetScaler** アプライアンスで実行してキャッシュに割り当てられているメモリを確認できるコマンドにはどのようなものがありますか?

  NetScaler アプライアンスのキャッシュに割り当てられたメモリを表示するには、CLI から次のコマンドのいずれかを実行します。

  - `show cache parameter`

    出力で、メモリ使用量制限パラメータの値を確認します。これはキャッシュに割り当てられる最大メモリです。

  - `show cache \<Content_Group_Name>`

    出力で、個々のコンテンツグループに使用および割り当てられているメモリを示す [メモリ使用量] パラメータと [メモリ使用量制限] パラメータの値を確認します。

- 私の **NetScaler** アプライアンスには **2 GB** のメモリが搭載されています。キャッシュの推奨メモリ制限はありますか**?**

  NetScaler アプライアンスのどのモデルでも、メモリの半分をキャッシュに割り当てることができます。ただし、Citrix では、内部メモリに依存するため、割り当てるメモリの半分弱を推奨しています。次のコマンドを実行すると、1 GB のメモリをキャッシュに割り当てることができます。

  ```
  set cache parameter -memLimit 1024
  ```

- 個々のコンテンツグループにメモリを割り当てることはできますか**?**

  はい。<Content_Group_Name><Integer>set cache パラメータ—memlimit を実行して統合キャッシュにメモリをグローバルに割り当てる場合でも <Integer>、set cache —memLimit コマンドを実行することで個々のコンテンツグループにメモリを割り当てることができます。コンテンツグループ（合計）に割り当てることができる最大メモリは、統合キャッシュに割り当てたメモリを超えることはできません。

- 統合キャッシュと **TCP** バッファのメモリの依存性について教えてください。

  NetScaler アプライアンスに 2 GB のメモリが搭載されている場合、アプライアンスは約 800 MB から 900 MB のメモリを予約し、残りは FreeBSD オペレーティングシステムに割り当てられます。そのため、統合キャッシュには最大 512 MB のメモリを割り当てることができ、残りは TCP バッファに割り当てられます。

- 統合キャッシュにグローバルメモリを割り当てないと、キャッシュプロセスに影響しますか**?**

  統合キャッシュにメモリを割り当てない場合、すべての要求がサーバーに送信されます。統合キャッシュにメモリが割り当てられていることを確認するには、show cache parameter コマンドを実行します。実際には、グローバルメモリが 0 の場合、オブジェクトはキャッシュされないので、最初に設定する必要があります。

検証コマンド

- キャッシュ統計を表示するにはどのようなオプションがありますか**?**

次のいずれかのオプションを使用して、キャッシュの統計情報を表示できます。

- `stat cache`

  キャッシュ統計の概要を表示します。

- `stat cache -detail`

  キャッシュ統計の詳細をすべて表示します。

- キャッシュされたコンテンツを表示するにはどのようなオプションがありますか**?**

  キャッシュされたコンテンツを表示するには、`show cache object` コマンドを実行します。

- キャッシュに保存されているオブジェクトの特性を表示するために実行できるコマンドは何ですか**?**

  キャッシュに保存されているオブジェクトが、たとえば GET //10.102.12.16:80/index.html の場合、アプライアンスの CLI から次のコマンドを実行すると、オブジェクトの詳細を表示できます。

  ```
  show cache object -url '/index.html'-host 10.102.3.96 -port 80
  ```

- パラメータ化されたオブジェクトをキャッシュに表示するパラメータとしてグループ名を指定することは必須ですか**?**

  はい。パラメータ化されたオブジェクトをキャッシュに表示するには、パラメータとしてグループ名を指定する必要があります。たとえば、同じルールで次のポリシーを追加したとします。

  ```
  1  add cache policy p2 -rule ns_url_path_cgibin -action CACHE  -
     storeInGroup g1
  2  add cache policy p1 -rule ns_url_path_cgibin -action CACHE -
     storeInGroup g2
  3  <!--NeedCopy-->
  ```

  この場合、複数のリクエストについて、ポリシー p1 が評価されると、選択カウンタが増え、ポリシーは select パラメータを持つ g1 グループにオブジェクトを格納します。そのため、キャッシュのオブジェクトを表示するには、次のコマンドを実行する必要があります。

  ```
  show cache object -url "/cgi-bin/setCookie.pl"-host 10.102.18.152
   groupName g1
  ```

  同様に、複数のリクエストが混在する別のセットでは、ポリシー p2 が評価されると、選択カウンタが増え、ポリシーは select パラメータのない g2 グループにオブジェクトを格納します。そのため、キャッシュのオブジェクトを表示するには、次のコマンドを実行する必要があります。

  ```
  show cache object -url "/cgi-bin/setCookie2.pl"-host 10.102.18.152
  ```

- **nscachemgr** コマンドの出力に空白のエントリがいくつかあることに気付きました。これらのエントリは何ですか**?**

  `nscachemgr`コマンドの次のサンプル出力について考えてみましょう。この出力の空白のエントリは、参考のために太字で強調表示されます。


```
root@ns# /netscaler/nscachemgr -a
//10.102.3.89:80/image8.png
//10.102.3.97:80/staticdynamic.html
//10.102.3.97:80/
//10.102.3.89:80/image1.png
//10.102.3.89:80/file5.html
//10.102.3.96:80/
//10.102.3.97:80/bg_logo_segue.png
//10.102.3.89:80/file500.html
//10.102.3.92:80/
//10.102.3.96:80/cgi-bin/rfc/ccProxyReval.pl
Total URLs in IC = 10
<!--NeedCopy-->
```

出力の空白のエントリは、GET /HTTP/1.1 のデフォルトのキャッシュプロパティによるものです。

オブジェクトをフラッシングする

- 選択したオブジェクトをキャッシュからフラッシュするにはどうしたらいいですか**?**

  オブジェクトは完全な URL で一意に識別できます。このようなオブジェクトをフラッシュするには、次のいずれかのタスクを実行できます。

  - フラッシュキャッシュ
  - フラッシュコンテンツグループ
  - 特定のオブジェクトをフラッシュします

  特定のオブジェクトをフラッシュするには、クエリパラメータを指定する必要があります。InvalParam パラメータを指定してオブジェクトをフラッシュします。このパラメータはクエリにのみ適用されます。

- キャッシュ構成の変更によってキャッシュがフラッシュされることはありますか**?**

  はい。キャッシュ構成に変更すると、すべての SET cache コマンドは本質的に適切なコンテンツグループをフラッシュします。

- サーバー上のオブジェクトを更新しました。キャッシュされたオブジェクトをフラッシュする必要がありますか**?**

  はい。サーバー上のオブジェクトを更新するときは、キャッシュされたオブジェクト、または少なくとも関連するオブジェクトとコンテンツグループをフラッシュする必要があります。統合キャッシュは、サーバーの更新による影響を受けません。キャッシュされたオブジェクトは、有効期限が切れるまで提供され続けます。

フラッシュキャッシュ

- **NetScaler** アプライアンスのフラッシュキャッシュ機能とは何ですか**?**

フラッシュクラウドという現象は、多数のクライアントが同じコンテンツにアクセスしたときに発生します。その結果、サーバーへのトラフィックが突然急増します。フラッシュキャッシュ機能により、NetScaler アプライアンスはこのような状況でサーバーにリクエストを 1 つだけ送信することでパフォーマンスを向上させることができます。他のすべての要求はアプライアンスのキューに入れられ、要求に対して単一の応答が処理されます。次のいずれかのコマンドを使用して Fast Cache 機能を有効にできます。

```
- add cache contentGroup \<Group_Name> -flashCache YES
- set cache contentGroup \<Group_Name> -flashCache YES
```

- **Flash Cache** クライアントの制限はどれくらいですか**?**

  Flash Cache クライアントの数は、NetScaler アプライアンス上のリソースの可用性によって異なります。

デフォルト動作

- **NetScaler** アプライアンスは有効期限が切れるとオブジェクトをプロアクティブに受信しますか?

  NetScaler アプライアンスは、有効期限が切れたオブジェクトをプロアクティブに受け取ることはありません。これはネガティブオブジェクトにも当てはまります。有効期限が切れた後の最初のアクセスは、サーバーへのリクエストをトリガーします。

- 統合キャッシュは、応答の受信を開始する前であっても、クライアントをキューに追加してサービスを提供しますか**?**

  はい。統合キャッシュは、応答の受信を開始する前であっても、クライアントをキューに追加して処理します。

- キャッシュ設定の **Verify cache object using** パラメーターのデフォルト値は何ですか**?**

  ホスト名と IP がデフォルト値です。

- **NetScaler** アプライアンスはログファイルにログエントリを作成しますか?

  はい。NetScaler アプライアンスはログファイルにログエントリを作成します。

- 圧縮されたオブジェクトはキャッシュに保存されていますか**?**

  はい。圧縮されたオブジェクトはキャッシュに保存されます。

他の機能との相互運用性

- 現在キャッシュに保存されており、**SSL VPN** 経由でアクセスされているオブジェクトはどうなりますか?

  キャッシュに格納され、定期的にアクセスされるオブジェクトは、キャッシュとして提供されます。SSL VPN 経由でアクセスしたときに選択します。

- **SSL VPN** を介してアクセスし、後で通常の接続を介してアクセスすると、キャッシュに格納されたオブジェクトはどうなりますか**?**

SSL VPN アクセスを介して格納されたオブジェクトは、通常の接続を介してアクセスされるときに SELECT として提供されます。

- **Web** ロギングを使用する場合、キャッシュから提供される応答を示すエントリとサーバーによって提供されるエントリをどのように区別しますか?

  統合キャッシュから返される応答の場合、サーバーログフィールドには値 IC が含まれます。サーバーから返された応答の場合、サーバーログフィールドにはサーバーから送信された値が含まれます。以下は、統合キャッシュトランザクションのログエントリのサンプルです。

  ```
  "10.102.1.52 - "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 4.0; .NET CLR 1.0.3705)""GET /"200 0 "IC"10.102.1.45"
  ```

  クライアントの要求とともに、ログに記録される応答はクライアントに送信されたものであり、必ずしもサーバーから送信されたものではありません。

> 注
>
> Web ロギングを使用する場合、統合キャッシュからの応答には、サーバーログフィールドに値 IC が含まれます。NSWL クライアントには、「%o1」形式指定子が付いたサーバログフィールドがあります。

その他

- 再失効と失効を設定するとどういう意味ですか**?**

  relexpiryとabsexpiryを設定すると、ヘッダーに表示される内容に関係なく、ヘッダーがオーバーライドされます。別の有効期限設定とコンテンツグループレベルを構成できます。relexpiryでは、ヘッダーの有効期限は、NetScaler ADC がオブジェクトを受信した時刻に基づきます。absexpiryでは、有効期限は NetScaler ADC で構成された時間に基づいています。Relexpiryは秒単位で設定されます。Absexpiry は時刻です。

- **weakpos** とヒューリスティックを設定するとどういう意味ですか**?**

  weakposとヒューリスティックはフォールバック値のようなものです。有効期限ヘッダーがある場合、最後に変更されたヘッダーが存在する場合にのみ考慮されます。NetScaler アプライアンスは、最後に変更されたヘッダーとヒューリスティックパラメーターに基づいて有効期限を設定します。ヒューリスティックな有効期限計算では、最後に変更されたヘッダーをチェックして有効期限までの時間を決定します。オブジェクトが最後に変更されてからの期間の一部が、有効期限までの時間として使用されます。長期間変更されず、有効期限が長くなる可能性が高いオブジェクトのヒューリスティック。-heurExpiryParam は、この計算で使用するパーセンテージ値を指定します。それ以外の場合、アプライアンスはweakpos値を使用します。

- 動的キャッシュを設定する前に考慮すべきことは何ですか**?**

  名前と値の形式で完全な URL クエリを含まないパラメータがある場合、またはアプライアンスがパラメータを Cookie ヘッダーまたは POST 本文で受信する場合は、動的キャッシュの設定を検討してください。動的キャッシュを設定するには、HitParams パラメーターを設定する必要があります。

- パラメータ名の **16** 進エンコーディングはどのようにサポートされていますか**?**

  NetScaler アプライアンスでは、パラメーター名で%HEXHEX エンコーディングがサポートされています。HitParams または InvalParams に指定する名前には、名前に%HEXHEX エンコーディングを含む名前を指定できます。たとえば、名前、名前%65、および n %61m%65 は同等です。

- **HitParam** パラメータを選択するプロセスはどのようなものですか**?**

  POST リクエストの HTTP ヘッダーの次の抜粋を考えてみましょう。

```
POST /data2html.asp?param1=value1&param2=&param3&param4=value4
HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-powerpoint, application/vnd.ms-excel,
application/msword, application/x-shockwave-flash, \*/\*
Referer: http://10.102.3.97/forms.html
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: 10.102.3.97
Content-Length: 153
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDQGQGGRNY=NNLLKDADEENOAFLCCDGFGDMO
S1=This+text+is+only+text%2C+not+more+and+not+less%2C+%0D%0Ajust+
   text+to+be+itself%2C+namely+%22Text%22+to+be+posted+as+text
   +%28what+else...%29&B1=Submit
<!--NeedCopy-->
```

  前述のリクエストでは、参照用に太字で強調表示されている S1 と B1 を、要件に応じて HitParams として使用できます。また、ASPSESSIONIDQGQGGRNY コンテンツグループで-MatchCookies YES を使用している場合は、これらのパラメーターを HitParams として使用することもできます。

- 応答をキャッシュできない場合、キューに入っているクライアントはどうなりますか**?**

  レスポンスがキャッシュできない場合、キュー内のすべてのクライアントは、最初のクライアントが受け取ったのと同じレスポンスを受け取ります。

- 同じコンテンツグループで毎回ポーリング（**PET**）機能と **Flash Cache** 機能を有効にすることはできますか?

  いいえ。同じコンテンツグループで PET と Flash Cache を有効にすることはできません。統合キャッシュは、Flash Cache コンテンツグループに対して AutoPET 機能を実行しません。PET 機能により、統合キャッシュはサーバに問い合わせることなく格納されたオブジェクトを処理することがなくなります。コンテンツグループに PET を明示的に設定できます。

- キューに入っているクライアントのログエントリはいつ作成されますか**?**

  アプライアンスが応答ヘッダーを受信した直後に、キューに入っているクライアントのログエントリが作成されます。ログエントリは、応答ヘッダーによってオブジェクトがキャッシュ不可にならない場合にのみ作成さ

れます。

- キャッシュ構成の **Verify Cache** オブジェクトパラメータの **DNS**、**HOSTNAME**、**HOSTNAME_AND_IP** の値はどういう意味ですか**?**

  意味は次のとおりです。

  - `set cache parameter -verifyUsing HOSTNAME`

    このコマンドは宛先 IP アドレスを無視します。

  - `set cache parameter -verifyUsing HOSTNAME_AND_IP`

    コマンドは宛先 IP アドレスと照合します。

  - `set cache parameter -verifyUsing DNS`

    コマンドは DNS サーバーを使用します。

- **weakNegRelExpiry** を **600**、つまり **10** 分に設定しました。**404** レスポンスがキャッシュされていないことに気付きました。その理由は何ですか?

  これは設定に完全に依存します。デフォルトでは、404 件の応答が 10 分間キャッシュされます。404 件のレスポンスをすべてサーバーから取得したい場合は、—WeaknegreExpiry 0 と指定します。—WeakNegRelExpiry を高くしたり低くしたりするなど、希望する値に微調整して 404 個の応答が適切にキャッシュされるようにすることができます。—ABSExpiry を肯定応答に設定した場合、期待した結果が得られない可能性があります。

- ユーザーが **Mozilla Firefox** ブラウザを使用してサイトにアクセスすると、更新されたコンテンツが提供されます。ただし、ユーザーが **Microsoft Internet Explorer** ブラウザーを使用してサイトにアクセスすると、古いコンテンツが表示されます。その理由は何でしょうか?

  Microsoft Internet Explorer ブラウザーは、NetScaler 統合キャッシュではなくローカルキャッシュからコンテンツを取得している可能性があります。その理由は、Microsoft Internet Explorer ブラウザーが応答の有効期限に関連するヘッダーを尊重していないことが原因である可能性があります。

  この問題を解決するには、Internet Explorer のローカルキャッシュを無効にして、オフラインコンテンツをクリアします。オフラインコンテンツをクリアした後、ブラウザは更新されたコンテンツを表示する必要があります。

- ヒット数がゼロの場合はどうなりますか?

  サーバー時刻と NS 時刻が同期しているかどうかを確認してください。また、WeakPosLexpiry 制限セットには、以下のように NS とサーバーの時間差が反映されている必要があります。

  ```
  root@ns180# date
  Tue May 15 18:53:52 IST 2012
  <!--NeedCopy-->
  ```

- ポリシーがヒットしたのに何もキャッシュされないのはなぜですか**?**

  メモリが統合キャッシュに割り当てられていること、および割り当てがゼロより大きいことを確認します。

- キャッシュカウンタをゼロにすることは可能ですか**?**

  キャッシュカウンタをゼロに設定するコマンドラインや GUI オプションはありません。また、キャッシュをフラッシュしても設定は行われません。ボックスを再起動すると、これらのカウンタは自動的にゼロに設定されます。

インストール、アップグレード、ダウングレード

December 8, 2023

インストールとアップグレード

特定の **NetScaler** リリースビルドパッケージをダウンロードするには**?**

特定の NetScaler リリースビルドパッケージをダウンロードする方法については、「NetScaler リリースパッケージのダウンロード」を参照してください。

NetScaler アプライアンスのシステムソフトウェアをアップグレードするには**?**

Citrix ADC アプライアンスのシステムソフトウェアのアップグレードについては、NetScaler スタンドアロンアプライアンスのアップグレードを参照してください。

NetScaler リリースビルドのリリースノートはどこにありますか**?**

NetScaler リリースビルドのリリースノートドキュメントには、リリースビルドについて以下が記載されています。

- 強化された機能
- 解決された問題
- 既知の問題

NetScaler リリースビルドのリリースノートドキュメントは、次の場所にあります。

- NetScaler ファームウェアまたは仮想アプライアンスは、特定のリリースビルドのダウンロードページをダウンロードします 。
- NetScaler ドキュメントサイトの ADC リリースノートページ

NetScaler アプライアンスのセキュリティ更新プログラムはどこにありますか**?**

NetScaler セキュリティチームは、関連するすべての NetScaler 製品の一般的な脆弱性とリスク（CVE）に関するセキュリティ速報を定期的にリリースしています。この情報はセキュリティ速報にあります。または、NetScaler サポートサイトで特定の CVE を検索することもできます。

NetScaler リリースで利用可能な **zebos.conf** ファイルの使用方法について教えてください**?**

NetScaler アプライアンスは、ルーティングスイートとして ZeBOS を使用します。NetScaler リリースで使用できる zebos.conf ファイルは、ZeboS 用の構成ファイルです。

NetScaler アプライアンスの **SSH** ポート（**22**）を他のポートに変更したいのですが。アプライアンスの **SSH** ポートを変更することはできますか**?**

はい。NetScaler アプライアンスの SSH ポートを変更するには、/nsconfig ディレクトリにある sshd_config ファイルを編集します。ファイルが /nsconfig ディレクトリに存在しない場合は、/etc ディレクトリからコピーします。

sshd_config ファイルで、ポート 22 のエントリを Port に編集します。<Number> ここで、<Number> はターゲットのポート番号です。アプライアンスを再起動して変更を有効にしたくない場合は、kill sshd コマンドを使用してプロセスを終了し、プロセスを再起動します。

フラッシュディレクトリが **NetScaler** アプライアンスにありません。フラッシュディレクトリをマウントするには、どのような手順に従わなければなりませんか

フラッシュディレクトリをマウントするには、次の手順を実行します。

1. NetScaler アプライアンスをシングルユーザーモードで起動します。

   アプライアンスが起動すると、次のメッセージが表示されます。

   すぐに起動するには [Enter] を、コマンドプロンプトには他のキーを押します。Booting [kernel] in 10 seconds…”スペースを選択すると、次のプロンプトが表示される必要があります。

   「?」と入力します。コマンドのリストについては、’help’ より詳細なヘルプを参照してください。

2. 次のコマンドを入力して FreeBSD をシングルユーザーモードで起動します。

   boot -s

   アプライアンスが起動すると、次のメッセージが表示されます。

   シェルのフルパス名を入力するか、/bin/sh のリターンを入力してください:

3. Enter キーを押して # プロンプトを表示します。

4. 次のコマンドを実行して、フラッシュディレクトリをマウントします。

```
mount /dev/ad0s1a /flash

Note: If the preceding command displays an error message about
   permissions, run the following command to check the disk for
   consistency:

fsck /dev/ad0s1a

Run the mount command again to mount the flash directory.
```

5. アプライアンスを再起動します。
6. シェルプロンプトから次のコマンドを実行して、flash ディレクトリがマウントされていることを確認します。

```
df -kh
```

パスワードを入力せずに **NetScaler** アプライアンスにログオンします。それを許可するようにアプライアンスに **SSH** を設定することは可能ですか

はい。NetScaler アプライアンスの SSH は、パスワードなしでログオンするように構成できます。ただし、ユーザー名を入力する必要があります。パスワードなしでログインできるように SSH を設定するには、次の操作を行います。

1. 次のコマンドを実行して、公開キーと秘密キーを生成します。

```
\# ssh-keygen -t rsa
```

2. 次のコマンドを実行して、ログオン先のリモートホストの.ssh ディレクトリに id_rsa.pub ファイルをコピーします。

```
\# scp id_dsa.pub \<user>@\<remote_host>/.ssh/id_dsa.pub
```

3. リモートホストにログオンします。
4. .ssh ディレクトリに移動します。
5. 次のコマンドを実行して、クライアントの公開キーを既知の公開キーに追加します。

```
\# cat id_dsa.pub >> authorized_keys2

\# chmod 640 authorized_keys2

\# rm id_dsa.pub
```


NetScaler アプライアンスの **BIOS** をリセットする手順は何ですか**?** どのような状況で **BIOS** をリセットする必要がありますか

NetScaler アプライアンスの BIOS をリセットするには、次の手順に従います。

1. シリアル・ポート経由でアプライアンスに接続します。
2. アプライアンスを起動し、起動プロセスが開始されたら Delete キーを押します。

   POST プロセス中に Delete キーを押すと、アプライアンスの BIOS 設定が表示されます。
3. BIOS 設定の終了ページをアクティブにします。
4.「最適なデフォルトをロード」オプションを選択します。「最適設定の読み込み」メッセージボックスが表示されます。
5.［OK］を選択します。
6. さまざまなタブの BIOS 設定を次のように変更します。

   タブ
7. BIOS 設定の終了ページをアクティブにします。
8. [変更を保存して終了] を選択します。
9.［OK］を選択して確定します。
10. アプライアンスが正常に起動し、アプライアンスの起動後にシリアルコンソールに出力が表示されることを確認します。

    シリアルコンソールが応答しない場合は、BIOS をリセットする必要があります。これは通常、アプライアンスをアップグレードしてシリアルコンソールが無効になった後に発生します。ただし、Telnet または SSH ユーティリティを使用してアプライアンスにアクセスすることはできます。

NetScaler アプライアンスを工場出荷時のデフォルトにリセットする必要があります。どのような手順に従わなければなりませんか

NetScaler アプライアンスを工場出荷時のデフォルトにリセットするには、NetScaler アプリケーション環境と FreeBSD 環境の 2 つの環境をリセットする必要があります。
アプライアンスの NetScaler アプリケーション環境を工場出荷時のデフォルトにリセットするには、次の手順を実行します。

1. アプライアンスの /nsconfig/ns.conf のバックアップを作成します。
2. /nsconfig/ns.conf ファイルを削除します。
3. アプライアンスを再起動します。アプライアンスの FreeBSD 環境を工場出荷時のデフォルトにリセットするには、以下を実行してください。

a) アプライアンスに新しい NetScaler コードイメージをインストールします。これにより、いくつかの FreeBSD レベルの設定ファイルがデフォルト値で上書きされます。
b) アプライアンスに追加されたすべてのユーザーとグループ、つまりデフォルトユーザーを除くすべてのユーザーとグループを削除します。
c) /etc/resolv.conf ファイルを削除します。
d) /etc/hosts ファイルに追加したエントリを削除します。
e) /etc/rc.netscaler ファイルが存在する場合は、それを削除します。
f) /etc/nsperm_group_suser ファイルを開き、すべての IOCTL エントリがコメントエントリであることを確認します。
g) /etc/rc.conf ファイルを開き、syslogd_Enable=NO エントリが syslogd_Enable=YES に変更されていないことを確認します。
h) /etc/syslog.conf ファイルを開き、ファイルに他のエントリがないことを確認します。
i) /var/nslog、/var/nstrace、および /var/crash ファイルの内容を削除します。
j) アプライアンスで syslog プロセスが有効になっていて、アプライアンスがログファイルをローカルに作成する場合は、/etc/syslog.conf ファイルにリストされているログファイルの内容を削除します。ファイルは /var/log ディレクトリに作成されます。たとえば、syslog プロセスが /var/log/events ファイルにシステムイベントを書き込み、/var/log/sslvpnevents ファイルに`sslvpn`イベントにアクセスする場合は、これらのファイルを削除します。

アプライアンスは、「**6** 月 **21** 日 **12:20:18 ns /flash/ns-10.0-47.15: [1/2] dc0: NIC** がハングアップ状態 **#663: TX 10000/10000**、**RX 0**、**HF 0**」というメッセージのようなメッセージをコンソールに表示します。このメッセージの意味は何ですか**?**

メッセージは次のコンポーネントで構成されています (ここでは例として示します)。

- #663: アプライアンスでこの状態が発生した回数。
- TX 10000/10000: アプライアンスが送信しようとしたパケットの数、および送信されたパケットの数。この例のように、両方の数値が同じ場合、NIC はアプライアンスが送信しようとしたすべてのパケットを送信しました。
- RX 0: 受信したパケットの数。この例では、パケットは受信されませんでした。
- HF0: NIC によって報告されたハードウェア問題の数。この例では、NIC からハードウェアの問題は報告されませんでした。

アプライアンスがパケットを受信しない場合、ネットワーク上ではパケットを受信しない可能性が高いため、ハング状態が報告されます。ただし、アプライアンスがインターフェイスに接続されている場合は、このエラーメッセージを無視できます。

アプライアンスで **NetScaler** リリースをアップグレードした後も、アプライアンスには以前のリリース/ビルドが表示されます。理由は何でしょうか**?**

アプライアンスには、/flash/boot/loader.conf ファイルのソフトウェアバージョン番号が表示されます。そのファイルに現在の NetScaler リリースのカーネルエントリがない場合、アプライアンスはエントリが使用できた最後の NetScaler リリースバージョンを表示します。
この問題を解決するには、次の手順を実行します:

1. カーネルファイルが /nsconfig ディレクトリに存在することを確認します。
2. /flash/boot/loader.conf ファイルでカーネルのエントリを確認してください。

   (インストールしたリリース/ビルドのカーネルのエントリがファイルにないことが予想されます)。
3. loader.conf ファイルを vi エディタなどのテキストエディタで開き、新しいリリース/ビルドのカーネルエントリを更新します。
4. ファイルを保存して閉じます。
5. /flash/boot/loader.conf.local ファイルに対してステップ 2 からステップ 4 を繰り返します。
6. ns.conf ファイルのリリース/ビルドエントリを更新します。
7. アプライアンスを再起動します。

アプライアンスの **NetScaler** リリースをアップグレードしたので、アプライアンスのフロントパネルの **LCD** ディスプレイにアウトオブサービスメッセージが表示されるか、何も表示されません。この問題はどうすれば解決できますか**?**

アプライアンスのシェルプロンプトから次のコマンドを実行します。

```
1 /netscaler/nslcd  - k
```

NetScaler のリリース/ビルドをアップグレードしました。ただし、アップグレードプロセスの後、アプライアンスは起動しません。アプライアンスのソフトウェアを以前のリリース/ビルドにダウングレードできますか**?**

はい。kernel.old カーネルファイルを使用してアプライアンスを起動できます。アプライアンスを再起動するとき、アプライアンスコンソールに PressF1 メッセージが表示されたら F1 キーを押します。**kernel.old** と入力して **Enter** キーを押します。

アプライアンスの **NetScaler** リリースをアップグレードした後、**/flash** ディレクトリからカーネルファイルを誤って削除してしまいました。その結果、アプライアンスを起動できません。このような状況でアプライアンスを起動する方法はありますか

はい。次のように、`kernel.GENERIC` カーネルファイルを使用してアプライアンスを起動できます。

1. アプライアンスを再起動するとき、アプライアンスコンソールに PressF1 メッセージが表示されたら F1 キーを押します。
2.「カーネル」と入力します。一般的な [Enter] を押します。
3. root ユーザーとしてログインします。
4. NetScaler リリースを再インストールします。
5. アプライアンスを再起動します。

アプライアンスソフトウェアをアップグレードした後、アプライアンスにログオンできず、次のメッセージが表示されます。パスワード回復手順を使用してこの問題を解決しようとしましたが、成功しませんでした。私は間違って何かをしましたか**?**

```
1 ```
2 login: nsroot
3 Password:
4 connect: No such file or directory
5 nsnet_connect: No such file or directory
6 Login incorrect
7 <!--NeedCopy--> ```
```

パスワード回復手順を使用してこの問題を解決することはできません。NetScaler リリース 12.1 以降では、起動時に実行されるImgrdデーモンに基づく新しいライセンスシステムを使用します。このデーモンが正常に動作するには、/nsconfig/rc.conf ファイルに設定されている NetScaler アプライアンスのホスト名が、ネームサーバーによって NSIP アドレスに解決される必要があります。<Host_Name> または、/nsconfig ディレクトリにホストファイルを作成し、ファイルに 127.0.0.1 エントリを追加することもできます。

また、ライセンスファイルが /nsconfig/license/ ディレクトリにコピーされていることを確認してください。

高可用性ペアのアップグレード中に、次のメッセージが繰り返し表示されます。理由は何でしょうか**?**

ns sshd [5035]: エラー: ユーザー名またはパスワードが無効です
このエラーメッセージは、高可用性ペアリングに関係するアプライアンスに、異なる NetScaler リリースまたは同じリリースの異なるビルドがインストールされている場合に表示されます。一方のアプライアンスをアップグレードまたはダウングレードし、もう一方のアプライアンスをアップグレードまたはダウングレードしなかった場合、アプライアンスには異なるバージョンがインストールされている可能性があります。
NetScaler アプライアンスの NSIP アドレスのネットマスクを変更したいのですが。システム停止を発生させずに実行できますか?
NetScaler IP のネットマスクを変更すると、短時間停止する可能性があります。必ずセカンダリアプライアンスのネットマスクを変更してから、高可用性ペアリングを解除してください。アプライアンスの機能を確認してください。すべてが期待どおりに動作したら、高可用性ペアリングを再構築します。
アプライアンスのネットマスクを変更するには、CLI プロンプトから ‘config ns‘コマンドを実行し、メニューの 2 番目のオプションを選択します。
NetScaler アプライアンスの高可用性ペアを構成しました。ソフトウェアリリースをプレビューリリースから最

終リリースにアップグレードしたところ、一部のアプライアンス構成が欠落していることに気付きました。失われた設定を取り戻すことはできますか
次の手順を使用して構成を復元できます。
1. プライマリアプライアンスの NetScaler コマンドラインにログオンします。
1. 次のコマンドを実行します:
save config
shell
\#cp /nsconfig/ns.conf /nsconfig/ns.conf.bkup
The ns.conf.bkup file is a backup for the running configuration.
1. 両方のアプライアンスのソフトウェアを最終リリースにアップグレードします。
1. プライマリアプライアンスの NetScaler コマンドラインにログオンします。
プライマリアプライアンスとセカンダリアプライアンスを別々のビルドにすることはできますか?
推奨される方法は、プライマリアプライアンスとセカンダリアプライアンスの両方で同じバージョンとビルド番号を使用することです。
高可用性 (HA) ペアの両方のアプライアンスを同時にアップグレードできますか?
いいえ。HA ペアでは、最初にセカンダリノードをアップグレードしてから、プライマリノードをアップグレードします。
詳細については、「[ハイアベイラビリティペアのアップグレード](/ja-jp/citrix-adc/13-1/upgrade-downgrade-citrix-adc-appliance/upgrade-downgrade-HA-pair.html)」を参照してください。
NetScaler は Amazon Web Services スクラウドでのファームウェアのアップグレードをサポートしていますか
はい。
NetScaler インスタンスを SDX バージョンとは別にアップグレードできますか?
NetScaler アプライアンスをアップグレードする場合、SDX バージョンをアップグレードする必要はありません。ただし、一部の機能は機能しない場合があります。
FTP サーバーを使用して NetScaler アプライアンスをアップグレードすることはできますか
いいえ。最初に NetScaler サイトからファームウェアをダウンロードし、ローカルコンピューターに保存してから、アプライアンスをアップグレードする必要があります。
GSLB 構成の NetScaler アプライアンスをアップグレードする手順は、GSLB に関係しないアプライアンスのアップグレードとは異なりますか
いいえ。アップグレード手順は基本的なアップグレード手順と似ています。唯一の違いは、さまざまなサイトのスタンドアロンまたは HA アプライアンスを段階的にアップグレードできることです。
構成が無効なため、アップグレードが失敗します。この問題を解決するにはどうすればいいですか
NetScaler 13.1 リリース以降、一部の機能では従来の式がサポートされていません。同様に、廃止された機能の一部も削除されました。廃止された機能やコマンドについて詳しくは、「[NetScaler Classic ポリシーベースの特徴と機能のステータス変更のお知らせ](https://support.citrix.com/article/CTX296948/notice-of-status-change-announcement-for-citrix-adc-formerly-netscaler-adc-classic-policy-based-features-and-functionalities)」を参照してください。
廃止または削除された機能に関連するコマンドが構成に含まれている場合、アップグレードは失敗します。リ

リース 13.1 以降では、廃止されたコマンドによりエラーが発生するため、構成が失われる可能性があります。‘nspepi‘ツールを使用して、無効または従来の構成を有効または高度な構成に変換します。‘nspepi‘ツールの詳細については、「[NSPEPI ツールを使用したポリシー表現の変換](/ja-jp/citrix-adc/13-1/appexpert/policies-and-expressions/introduction-to-policies-and-exp/converting-policy-expressions-nspepi-tool.html)」 を参照してください。
>** 注:**
>
> 無効な構成エラーが表示される場合は、‘installns‘ スクリプトの実行中に **Y** オプションを使用しないことをお勧めします。**Y** オプションを使用すると、構成チェックは行われず、無効な構成が失われる可能性があります。
>
> ![アップグレード中の構成エラーが無効です](/en-us/citrix-adc/media/installns-script-configuration-error.png)
ダウングレード
最新の NetScaler リリースがインストールされた NetScaler アプライアンスを受け取りました。しかし、ソフトウェアリリースをダウングレードしたいのですが。そうすることはできますか
いいえ。ソフトウェアリリースをダウングレードしようとすると、新しいリリースの ns.conf ファイルが以前のリリースと互換性がなく、アプライアンスが工場出荷時の設定に復元される可能性があるため、アプライアンスが期待どおりに動作しない可能性があります。
NetScaler リリースをダウングレードするときは、指示に従いました。ただし、アプライアンスには次のメッセージが表示されます。NetScaler アプライアンスではロールバック手順はどのように実行されますか
root@LBCOL03B# ./installns
installns version (10.0-47.7) kernel (ns-10.0-47.7.gz)
Note:
Installation may pause for up to 3 minutes while data is written to the flash.
Caution:
Do not interrupt the installation process.
Doing so may cause the system to become unusable.
Installation will proceed in 5 seconds, CTRL-C to abort
No Valid NetScaler Version Detected
root@LBCOL03B#
ロールバック手順は、基本的なアップグレード手順に似ています。ロールバックするターゲットビルドを選択し、ダウングレードを実行します。別のリリースにロールバックする前に、現在の設定ファイルのコピーを作成することをお勧めします。リリースからダウングレードするには、[NetScaler スタンドアロンアプライアンスのダウングレードを参照してください](/ja-jp/citrix-adc/13-1/upgrade-downgrade-citrix-adc-appliance/downgrade-standalone-appliance.html)。

負荷分散

August 15, 2023

- **NetScaler** アプライアンスで作成できるさまざまな負荷分散ポリシーにはどのようなものがありますか?

  NetScaler アプライアンスでは、次の種類の負荷分散ポリシーを作成できます。

  - 最小接続数
  - ラウンドロビン
  - 最短の応答時間
  - 最小帯域幅
  - 最小パケット数
  - URL ハッシュ
  - ドメイン名ハッシュ
  - 送信元 IP アドレスハッシュ
  - 宛先 IP アドレスハッシュ
  - 送信元 IP-宛先 IP ハッシュ
  - トークン
  - LRTM

- **NetScaler** アプライアンスを使用して負荷分散を実装することで、**Web** ファームのセキュリティを実現できますか?

  はい。NetScaler アプライアンスを使用して負荷分散を実装することで、Web ファームのセキュリティを実現できます。NetScaler アプライアンスでは、負荷分散機能の次のオプションを実装できます。

  - IP アドレスの非表示: セキュリティ上の理由と IP アドレスの保護のため、実際のサーバーをプライベート IP アドレス空間にインストールできます。NetScaler アプライアンスはサーバーに代わってリクエストを受け入れるため、このプロセスはエンドユーザーには意識されません。アドレス隠蔽モードでは、アプライアンスは 2 つのネットワークを完全に分離します。したがって、クライアントは、FTP や Telnet サーバーなどのプライベートサブネットで実行されているサービスに、そのサービスのアプライアンス上の別の VIP を介してアクセスできます。
  - ポートマッピング: セキュリティ上の理由から、実際の TCP サービスを非標準ポートでホストできるようにします。このプロセスは、NetScaler ADC アプライアンスが標準のアドバタイズ IP アドレスとポート番号でサーバーに代わって要求を受け入れるため、エンドユーザーには透過的です。

- **NetScaler ADC** アプライアンスの負荷分散に使用できるさまざまなデバイスは何ですか**?**

  NetScaler ADC アプライアンスを使用して、次のデバイスを負荷分散できます。

  - サーバーファーム
  - キャッシュまたはリバースプロキシ

  - ファイアウォールデバイス
  - 侵入検知システム
  - SSL オフロードデバイス
  - 圧縮デバイス
  - コンテンツ検査サーバー

- **Web** サイトの負荷分散機能を実装する必要があるのはなぜですか**?**

  Web サイトの負荷分散機能を実装すると、次の利点があります。

  - 応答時間の短縮**:** ウェブサイトに負荷分散機能を実装する場合の大きな利点の 1 つは、ロード時間が大幅に短縮されることです。2 つ以上のサーバーが Web トラフィックの負荷を分散しているため、各サーバーのトラフィック負荷は 1 台のサーバーだけの場合よりも少なくなります。つまり、クライアントの要求を満たすために利用できるリソースが増えるということです。これにより、ウェブサイトが速くなります。
  - 冗長性**:** 負荷分散機能を実装すると、多少の冗長性が生じます。たとえば、ウェブサイトが 3 つのサーバー間でバランスが取れていて、そのうちの 1 つがまったく応答しない場合、他の 2 つは実行し続けることができ、ウェブサイトの訪問者もダウンタイムに気付かない。負荷分散ソリューションは、利用できないバックエンドサーバーへのトラフィックの送信を直ちに停止します。

- リンクロードバランシング **(LLB)** の **Mac** ベースフォワーディング **(MBF)** オプションを無効にする必要があるのはなぜですか**?**

  - MBF オプションを有効にすると、NetScaler ADC アプライアンスは、クライアントからの着信トラフィックと同じクライアントへの発信トラフィックが同じアップストリームルーターを通過すると見なします。ただし、LLB 機能では、リターントラフィックに最適なパスを選択する必要があります。
  - MBF オプションを有効にすると、着信クライアントトラフィックを転送したルータを介して発信トラフィックを送信することによって、このトポロジ設計が中断されます。

- **NetScaler** アプライアンスで使用できるさまざまなパーシステンスタイプにはどのようなものがありますか?

  NetScaler アプライアンスは、次のパーシステンスタイプをサポートしています。

  - 接続元 IP
  - クッキーインサート
  - SSL セッション ID
  - URL パッシブ
  - Custom Server ID
  - 規則
  - DESTIP

GUI

August 15, 2023

- **Firefox** を使用して **2** つの **NetScaler ADC** 構成を比較すると、ブラウザがフリーズするようです。

  Firefox は最終的に設定の違いを表示しますが、1000 を超える違いがあると、処理にかなりの時間がかかります。Chrome を使用すると、応答が速くなります。

- **MAC Safari** ブラウザを使用して **NetScaler ADC** をアップグレードしています。アップグレードウィザードで、**[Browse]** ボタンをクリックしてアプライアンスからビルドファイルを選択すると、ダイアログボックスにファイルまたはフォルダが表示されません。また、ルートフォルダに戻ると、ダイアログボックスに最上位フォルダが表示されますが、参照できません。どうしたらいいですか**?**

  Safari ブラウザで「設定」アイコンをクリックし、「環境設定」**>**「セキュリティ」**>**「**Web** サイト設定の管理」**>**「**Java**」に移動します。[他の **Web** サイトにアクセスしたとき] 設定の値を [安全でないモードで実行する] に変更します。

- **GUI** にアクセスする前に何をすべきですか**?**

  新しいバージョンの NetScaler ADC ソフトウェアにアクセスする前に:

  - Cookie を含むブラウザのキャッシュをクリアします。
  - ブラウザのシークレットモードで GUI にアクセスします。
  - 他のブラウザで GUI にアクセスします。
  - 設定の [ソフトウェアアクセラレーションを使用する] オプションをオフにして、ブラウザを再起動します。
  - **chrome:** 拡張機能にアクセスし、[有効] チェックボックスをオフにして、Chrome ブラウザを再起動します。

- **HTTP** または **HTTPS** を使用して **GUI** にアクセスするにはどのポートを開くべきですか**?**

  NetScaler MPX、VPX、および CPX アプライアンスの HTTP および HTTPS 管理サービス（GUI）のデフォルトのポート番号を以下に示します。

  - NetScaler MPX および VPX アプライアンス：80（HTTP）および 443（HTTPS）アプライアンス
  - NetScaler CPX アプライアンス：9080（HTTP）および 9443（HTTPS）

  また、ポート 80 および 443 以外の HTTP および HTTPS 管理サービス (GUI) 用のポートを構成することもできます。詳細については、「HTTP および HTTPS 管理ポートの設定」を参照してください。

- どのブラウザで、異なるオペレーティングシステムと互換性のある **GUI** がありますか**?**

  次の表は、NetScaler GUI バージョン 12.1、13.0、および 13.1 と互換性のあるブラウザーの一覧です。

オペレーティングシステム	**Web** ブラウザー	バージョン
Windows 10 以降	Edge	110.1587.63 およびそれ以降
Windows 10 以降	Mozilla Firefox	102 およびそれ以降
Windows 10 以降	Chrome	108 およびそれ以降
MAC	Mozilla Firefox	110.0.1 およびそれ以降
MAC	Safari	15.5 およびそれ以降

SSL

August 15, 2023

SSL に関するよくある質問については、ここをクリックしてください 。

アプリケーショントラフィックの認証、承認、監査

February 15, 2024

多くの企業は、Web サイトへのアクセスを有効なユーザーのみに制限し、各ユーザーに許可されるアクセスのレベルを制御しています。認証、承認、および監査機能により、サイト管理者はアプリケーションごとにアクセス制御を個別に管理する代わりに、NetScaler アプライアンスを使用してアクセス制御を管理できます。アプライアンスで認証を行うと、アプライアンスによって保護されている同じドメイン内のすべての Web サイト間で、この情報を共有することもできます。

認証、認可、および監査を使用するには、認証プロセスを処理するように認証仮想サーバを設定し、認証を必要とする Web アプリケーションへのトラフィックを処理するトラフィック管理仮想サーバを設定する必要があります。また、各仮想サーバーに FQDN を割り当てるように DNS を構成します。仮想サーバーを構成したら、NetScaler アプライアンスを介して認証する各ユーザーのユーザーアカウントを構成します。オプションでグループを作成し、ユーザーアカウントをグループに割り当てます。ユーザーアカウントとグループを作成した後、ユーザーを認証する方法、ユーザーにアクセスを許可するリソース、およびユーザーセッションをログに記録する方法をアプライアンスに指示するポリシーを構成します。ポリシーを有効にするには、各ポリシーをグローバル、特定の仮想サーバー、または適切なユーザーアカウントまたはグループにバインドします。ポリシーを構成したら、セッション設定を構成し、セッションポリシーをトラフィック管理仮想サーバーにバインドして、ユーザーセッションをカスタマイズします。最後に、イントラネットでクライアント証明書を使用する場合は、クライアント証明書の設定をセットアップします。

認証、承認、および監査が分散環境でどのように機能するかを理解するために、従業員がオフィス、自宅、および旅行中にアクセスするイントラネットを持つ組織を考えてみましょう。イントラネット上のコンテンツは機密情報であり、安全なアクセスが必要です。イントラネットにアクセスするすべてのユーザーは、有効なユーザー名とパスワードを持っている必要があります。これらの要件を満たすために、ADC は次のことを行います。

- ユーザーがログインせずにイントラネットにアクセスした場合、ユーザーをログインページにリダイレクトします。
- ユーザーの資格情報を収集して認証サーバーに配信し、ライトウェイトディレクトリアクセスプロトコル（LDAP）を介してアクセスできるディレクトリにキャッシュします。詳細については、LDAP ディレクトリ内の属性の決定を参照してください。
- ユーザーの要求をアプリケーションサーバーに配信する前に、ユーザーが特定のイントラネットコンテンツにアクセスする権限を持っていることを確認します。
- セッションタイムアウトを維持します。このタイムアウトを過ぎた後、イントラネットへのアクセスを回復するには、ユーザーが再度認証を受ける必要があります。(タイムアウトは設定できます)。
- 無効なログイン試行を含め、ユーザーのアクセスを監査ログに記録します。

サポートされる認証の種類

- ローカル
- LDAP
- RADIUS
- SAML
- TACACS+
- クライアント証明書認証（スマートカード認証を含む）
- Web
- 高度な認証
- フォームベースの認証
- 401 ベースの認証
- ネイティブ OTP
- プッシュ通知
- メール OTP
- reCaptcha

NetScaler Gateway は、RSA SecurID、ジェムアルト・プロティバ、SafeWord もサポートしています。これらの認証の種類を構成するには、RADIUS サーバーを使用します。

認証、承認、および監査を構成する前に、NetScaler アプライアンスでの負荷分散、コンテンツスイッチング、および SSL の設定方法に精通し、理解しておく必要があります。

承認なしの認証

承認は、ユーザーがアプライアンスにログオンしたときにアクセスできるネットワークリソースを指定します。承認のデフォルト設定では、すべてのネットワークリソースへのアクセスを拒否します。デフォルトのグローバル設定を使用し、承認ポリシーを作成して、ユーザーがアクセスできるネットワークリソースを定義することをお勧めします。

アプライアンスで認可を設定するには、認可ポリシーと式を使用します。承認ポリシーを作成したら、アプライアンスで構成したユーザーまたはグループに承認ポリシーをバインドできます。

アプライアンスは、許可なしで認証のみを使用するように設定できます。許可なしで認証を設定すると、アプライアンスはグループ認可チェックを実行しません。ユーザーまたはグループに設定したポリシーは、ユーザーに割り当てられます。

認証、承認、監査の有効化

認証、認可、および監査機能を使用するには、この機能を有効にする必要があります。認証、認可、および監査機能を有効にする前に、認証、認可、および監査エンティティ（認証およびトラフィック管理仮想サーバなど）を設定できますが、エンティティは機能が有効になるまで機能しません。

CLI を使用して認証、承認、および監査を有効にするには

コマンドプロンプトで次のコマンドを入力して、認証、承認、および監査を有効にし、構成を確認します。

```
enable ns feature AAA
<!--NeedCopy-->
```

GUI を使用して認証、承認、および監査を有効にするには

1. [**System**] > [**Settings**] に移動します。
2. 詳細ウィンドウの [モードと機能] で、[基本機能の変更] をクリックします。
3. [基本機能の構成] ダイアログボックスで、[認証、承認、監査] チェックボックスをオンにします。
4. [**OK**] をクリックします。

認証の無効化

展開で認証が不要な場合は、認証を無効にできます。認証を必要としない仮想サーバーごとに、認証を無効にできます。

> 重要:
>
> 重要：認証は慎重に無効にすることをお勧めします。外部認証サーバを使用していない場合は、アプライアンスがユーザを認証できるようにローカルユーザおよびグループを作成します。認証を無効にすると、アプライアンスへの接続を制御およびモニタする認証、認可、およびアカウンティング機能の使用が停止します。ユーザーが Web アドレスを入力してアプライアンスに接続すると、ログオンページは表示されません。

認証を無効にするには

1. 構成 **> NetScaler Gateway >** 仮想サーバーに移動します。
2. 詳細ペインで仮想サーバーを選択して、[**Open**] をクリックします。
3. [基本設定**]** ページで、[認証を有効にする] チェックボックスをオフにします。

認証、承認、監査のしくみ

February 15, 2024

認証、承認、および監査は、適切な資格情報を持つすべてのクライアントが、インターネット上のどこからでも保護されたアプリケーションサーバーに安全に接続できるようにすることで、分散インターネット環境のセキュリティを提供します。この機能には、認証、承認、および監査の 3 つのセキュリティ機能が組み込まれています。認証により、NetScaler はクライアントの資格情報をローカルまたはサードパーティの認証サーバーで検証し、承認されたユーザーのみが保護対象サーバーにアクセスできるようにします。承認により、ADC は、保護されたサーバー上のどのコンテンツが各ユーザーにアクセスできるかを検証できます。監査により、ADC は、保護されたサーバー上の各ユーザーのアクティビティの記録を保持できます。

認証、承認、および監査が分散環境でどのように機能するかを理解するために、従業員がオフィス、自宅、および旅行中にアクセスするイントラネットを持つ組織を考えてみましょう。イントラネット上のコンテンツは機密情報であり、安全なアクセスが必要です。イントラネットにアクセスするすべてのユーザーは、有効なユーザー名とパスワードを持っている必要があります。これらの要件を満たすために、ADC は次のことを行います。

- ユーザーがログインせずにイントラネットにアクセスした場合、ユーザーをログインページにリダイレクトします。
- ユーザーの資格情報を収集し、認証サーバーに配信し、LDAP 経由でアクセス可能なディレクトリにキャッシュします。詳細については、LDAP ディレクトリ内の属性の決定を参照してください。
- ユーザーの要求をアプリケーションサーバーに配信する前に、ユーザーが特定のイントラネットコンテンツにアクセスする権限を持っていることを確認します。
- セッションタイムアウトを維持します。このタイムアウトを過ぎた後、イントラネットへのアクセスを回復するには、ユーザーが再度認証を受ける必要があります。(タイムアウトは設定できます)。
- 無効なログイン試行を含め、ユーザーのアクセスを監査ログに記録します。

認証、承認、監査ポリシーを構成する

ユーザーとグループを設定したら、次に認証ポリシー、承認ポリシー、および監査ポリシーを構成して、イントラネットへのアクセスを許可するユーザー、各ユーザーまたはグループがアクセスできるリソース、認証、承認、および監査の詳細レベルを定義します。は監査ログに保持されます。認証ポリシーは、ユーザーがログオンを試みたときに適用する認証の種類を定義します。外部認証を使用する場合、ポリシーは外部認証サーバも指定します。承認ポリシーは、ユーザーとグループがログオンした後にアクセスできるネットワークリソースを指定します。監査ポリシーは、監査ログのタイプと場所を定義します。

各ポリシーを有効にするには、各ポリシーをバインドする必要があります。認証ポリシーを認証仮想サーバーに、認可ポリシーを 1 つ以上のユーザーアカウントまたはグループにバインドし、監査ポリシーをグローバルに、および 1 つ以上のユーザーアカウントまたはグループにバインドします。

ポリシーをバインドするときは、そのポリシーにプライオリティを割り当てます。プライオリティによって、定義したポリシーが評価される順序が決まります。優先度は、任意の正の整数に設定できます。NetScaler オペレーティングシステムでは、ポリシーの優先順位は逆の順序で機能します。数値が大きいほど、優先度は低くなります。たとえば、プライオリティが 10、100、1000 の 3 つのポリシーがある場合、プライオリティ 10 が割り当てられたポリシーが最初に実行され、次にポリシーにプライオリティ 100 が割り当てられ、最後にポリシーにオーダー 1000 が割り当てられます。認証、認可、および監査機能は、要求が一致する各タイプのポリシーのうち最初のもののみを実装し、要求が一致する可能性のあるそのタイプの追加のポリシーは実装しません。したがって、ポリシーの優先順位は、意図した結果を得るために重要です。

他のポリシーを任意の順序で追加するための十分な余裕を残し、ポリシーをバインドするときに各ポリシー間に 50 または 100 の間隔で優先順位を設定することで、希望の順序で評価されるように設定できます。その後、既存のポリシーの優先順位を再割り当てすることなく、いつでもポリシーを追加できます。

NetScaler アプライアンスのバインディングポリシーの詳細については、NetScaler 製品ドキュメントを参照してください。

を構成します **No_Auth** 特定のトラフィックをバイパスするポリシー

トラフィック管理仮想サーバーで 401 ベース認証が有効になっている場合、認証から特定のトラフィックをバイパスするように No_Auth ポリシーを構成できるようになりました。このようなトラフィックの場合は、「No_Auth」ポリシーをバインドする必要があります。

CLI を使用して特定のトラフィックをバイパスするように **No_Auth** ポリシーを設定するには

コマンドプロンプトで入力します。

```
add authentication policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

例**:**


```
add authentication policy ldap -rule ldapAct1 -action No_Auth
<!--NeedCopy-->
```

認証、承認、監査構成の基本コンポーネント

August 15, 2023

認証、承認、および監査設定の基本的なコンポーネントは次のとおりです。

- 認証仮想サーバー -すべての認証要求は、トラフィック管理仮想サーバー（負荷分散またはコンテンツスイッチング）によって認証仮想サーバーにリダイレクトされます。この仮想サーバは、関連付けられた認証ポリシーを処理し、それに応じてアプリケーションへのアクセスを提供します。詳細については、「認証仮想サーバー」を参照してください。
- 認証プロファイル -認証プロファイルは、認証仮想サーバー、認証ホスト、認証ドメイン、および認証レベルを指定します。

1 つ以上の認証プロファイルを作成して、さまざまな認証設定を指定し、要件に基づいてこれらの認証プロファイルを関連するトラフィック管理サーバーにバインドできます。詳細については、認証プロファイルを参照してください。

- 認証ポリシー -ユーザーが NetScaler または NetScaler Gateway アプライアンスにログオンすると、作成したポリシーに従って認証されます。認証ポリシーは、表現とアクションで構成されます。認証ポリシーでは NetScaler の式を使用します。詳細については、「認証ポリシー」を参照してください。
- 承認ポリシー -承認ポリシーを構成するときに、内部ネットワーク内のネットワークリソースへのアクセスを許可または拒否するように設定できます。詳細については、承認ポリシーを参照してください。
- ユーザーとグループ： -認証、承認、および監査の基本設定を構成した後、ユーザーとグループを作成します。まず、NetScaler アプライアンスを介して認証する各人のユーザーアカウントを作成します。NetScaler アプライアンス自体によって制御されるローカル認証を使用している場合は、ローカルユーザーアカウントを作成し、それらの各アカウントにパスワードを割り当てます。詳細については、「ユーザーとグループ」を参照してください。

認証仮想サーバー

August 15, 2023

トラフィック管理仮想サーバ（負荷分散またはコンテンツスイッチング）は、すべての認証要求を認証仮想サーバにリダイレクトします。この仮想サーバは、関連付けられた認証ポリシーを処理し、それに応じてアプリケーションへのアクセスを提供します。

注：トラフィック管理ポリシーは、認証、承認、および監査仮想サーバにバインドできません。

認証仮想サーバーをセットアップする

認証仮想サーバーの設定に関連する手順は次のとおりです。

1. 認証、認可、および監査機能を有効にします。

```
enable ns feature AAA
<!--NeedCopy-->
```

2. 認証仮想サーバを設定します。SSL タイプで、SSL 証明書とキーのペアを仮想サーバーにバインドする必要があります。

```
add authentication vserver <name> SSL <ipaddress> <port>

bind ssl certkey <auth-vserver-name> <certkey>
<!--NeedCopy-->
```

3. 認証仮想サーバーのドメインの FQDN を指定します。

```
set authentication vserver <name> -authenticationDomain <FQDN>
<!--NeedCopy-->
```

4. 認証仮想サーバを関連するトラフィック管理仮想サーバに関連付けます。

   注意事項:

   - ドメインセッション Cookie が正しく機能するには、トラフィック管理仮想サーバの FQDN が認証仮想サーバの FQDN と同じドメインに存在する必要があります。トラフィック管理仮想サーバで、次の操作を行います。
     - 認証を有効にします。
     - トラフィック管理仮想サーバの認証ホストとして、認証仮想サーバの FQDN を指定します。
     - [オプション] トラフィック管理仮想サーバーの認証ドメインを指定します。
     - 認証ドメインを構成しない場合、アプライアンスは、ホスト名部分のない認証仮想サーバーの FQDN で構成される FQDN を割り当てます。たとえば、認証仮想サーバーのドメイン名が **tm.xyz.bar.com** の場合、アプライアンスは認証ドメインとして **xyz.bar.com** を割り当てます。
       * 負荷分散の場合:

```
set lb vserver <name> -authentication ON -
    authenticationhost <FQDN> [-authenticationdomain <
    authdomain>]
```



```
<!--NeedCopy-->
```

* コンテンツスイッチングの場合:

```
set cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

- 認証ドメインにドメイン全体の Cookie を設定する必要がある場合は、負荷分散仮想サーバーで認証プロファイルを有効にする必要があります。

5. 両方の仮想サーバが稼働しており、正しく設定されていることを確認します。

```
show authentication vserver <name>
<!--NeedCopy-->
```

GUI を使用して認証仮想サーバーをセットアップするには

1. 認証、認可、および監査機能を有効にします。

   [システム] **>** [設定] に移動し、[基本機能の構成] をクリックして、[認証、承認、監査] を有効にします。

2. 認証仮想サーバを設定します。

   [セキュリティ] **>** [**AAA-**アプリケーショントラフィック] **>** [仮想サーバ] に移動し、必要に応じて設定します。

3. 認証用にトラフィック管理仮想サーバを設定します。

   - 負荷分散の場合**:**

     [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、必要に応じて仮想サーバーを設定します。

   - コンテンツスイッチングの場合**:**

     [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、必要に応じて仮想サーバーを設定します。

4. - 認証の設定を確認します。

     [セキュリティ] **>** [**AAA-**アプリケーショントラフィック] **>** [仮想サーバ] に移動し、関連する認証仮想サーバの詳細を確認します。

認証仮想サーバーを構成する

認証、承認、および監査を構成するには、まず認証トラフィックを処理するように認証仮想サーバーを構成します。次に、SSL 証明書とキーのペアを仮想サーバーにバインドして、SSL 接続を処理できるようにします。
SSL の構成および証明書とキーのペアの作成の詳細については、「SSL 証明書」を参照してください。

CLI を使用して認証仮想サーバーを構成します

認証仮想サーバーを構成して構成を確認するには、コマンドプロンプトで次のコマンドを同じ順序で入力します。

```
add authentication vserver <name> ssl <ipaddress>

show authentication vserver <name>

bind ssl certkey <certkeyName>

show authentication vserver <name>

set authentication vserver <name>

show authentication vserver <name>
<!--NeedCopy-->
```

例**:**

```
add authentication vserver Auth-Vserver-2 SSL 10.102.29.77 443 Done

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: DOWN[Certkey not bound
    ] Client Idle Timeout: 180 sec Down state flush: DISABLED Disable
    Primary Vserver On Down : DISABLED Authentication : ON Current AAA
    Users: 0 Done

bind ssl certkey Auth-Vserver-2 Auth-Cert-1 Done

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: UP Client Idle Timeout
    : 180 sec Down state flush: DISABLED Disable Primary Vserver On Down
     : DISABLED Authentication : ON Current AAA Users: 0 Done

set authentication vserver Auth-Vserver-2

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: DOWN[Certkey not bound
    ] Client Idle Timeout: 180 sec Down state flush: DISABLED Disable
    Primary Vserver On Down : DISABLED Authentication : ON Current AAA
    Users: 0 Done
<!--NeedCopy-->
```

> 注**:**
>
> [認証ドメイン] パラメーターは廃止されました。認証プロファイルを使用して、ドメイン全体の Cookie を設定します。

GUI を使用して認証仮想サーバーを構成する

1. [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] に移動します。

2. 詳細ウィンドウで、次のいずれかの操作を行います。

   - 新しい認証仮想サーバーを作成するには、[追加] をクリックします。
   - 既存の認証仮想サーバーを変更するには、仮想サーバーを選択し、[編集] をクリックします。[設定] ダイアログが開き、[基本設定] 領域が展開されます。

3. パラメータの値を次のように指定します（アスタリスクは必須パラメータを示します）。

   - name*—name（以前に作成した仮想サーバでは変更できません）
   - IP アドレスタイプ *：認証仮想サーバの IP アドレスタイプ
   - IP アドレス *：認証仮想サーバの IP アドレス
   - port*：仮想サーバが接続を受け付ける TCP ポート。
   - ログイン失敗タイムアウト：failedLoginTimeout（ログインが失敗するまでの秒数。ユーザはログインプロセスを再度開始する必要があります）。
   - 最大ログイン試行回数-maxLoginAttempts（ユーザがロックアウトされる前に許可されるログイン試行回数）

   > 注**:**
   >
   > 認証仮想サーバーは SSL プロトコルとポート 443 のみを使用するため、これらのオプションは灰色表示されます。言及されていないオプションは無視できます。

4. [続行] をクリックして、[証明書] 領域を表示します。
5. [証明書**]** 領域で、この仮想サーバーで使用するすべての SSL 証明書を構成します。

   - CA 証明書を構成するには、[CA 証明書] の右側にある矢印をクリックして [CA Cert Key] ダイアログボックスを表示し、この仮想サーバーにバインドする証明書を選択して、[保存] をクリックします。
   - サーバ証明書を設定するには、[サーバ証明書（Server Certificate）] の右側にある矢印をクリックし、CA 証明書の場合と同じプロセスに従います。

6. [続行] をクリックして、[高度な認証ポリシー] 領域を表示します。
7. 高度な認証ポリシーを仮想サーバーにバインドする場合は、行の右側にある矢印をクリックして［認証ポリシー］ダイアログボックスを表示します。サーバーにバインドするポリシーを選択し、優先順位を設定して、［**OK**］をクリックします。
8. [続行] をクリックして、[基本認証ポリシー] 領域を表示します。
9. 基本認証ポリシーを作成して仮想サーバにバインドする場合は、プラス記号をクリックして [ポリシー] ダイアログボックスを表示し、プロンプトに従ってポリシーを構成し、この仮想サーバにバインドします。
10. [続行] をクリックして、[401 ベースの仮想サーバー] 領域を表示します。
11. [401 ベースの仮想サーバー] 領域で、この仮想サーバーにバインドする負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーを構成します。

- 負荷分散仮想サーバーをバインドするには、負荷分散仮想サーバーの右側にある矢印をクリックして [負荷分散仮想サーバー] ダイアログボックスを表示し、プロンプトに従います。
- コンテンツスイッチ仮想サーバーをバインドするには、コンテンツスイッチ仮想サーバーの右側にある矢印をクリックして [コンテンツスイッチ仮想サーバー] ダイアログボックスを表示し、LB 仮想サーバーをバインドする場合と同じ手順に従います。

12. グループを作成または構成する場合は、[グループ] 領域で矢印をクリックして [グループ] ダイアログボックスを表示し、プロンプトに従います。
13. 設定を確認し、完了したら [完了] をクリックします。ダイアログボックスが閉じます。新しい認証仮想サーバを作成した場合は、[**Configuration**] ウィンドウのリストに表示されます。

トラフィック管理仮想サーバー

認証仮想サーバを作成して設定したら、次にトラフィック管理仮想サーバを作成または設定し、認証仮想サーバをそれに関連付けます。トラフィック管理仮想サーバーには、負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーを使用できます。
いずれかのタイプの仮想サーバーの作成と構成の詳細については、「*Citrix* トラフィック管理ガイド」の「トラフィック管理」を参照してください。

> 注:
>
> ドメインセッション Cookie が正しく機能するには、トラフィック管理仮想サーバーの FQDN が認証仮想サーバーの FQDN と同じドメインにある必要があります。

認証を有効にし、認証サーバの FQDN をトラフィック管理仮想サーバに割り当てることで、認証、認可、および監査用のトラフィック管理仮想サーバを設定します。現在、トラフィック管理仮想サーバで認証ドメインを設定することもできます。このオプションを構成しない場合、NetScaler アプライアンスはトラフィック管理仮想サーバーに、ホスト名部分のない認証仮想サーバーの FQDN で構成される FQDN を割り当てます。たとえば、認証仮想サーバーのドメイン名が tm.xyz.bar.com の場合、アプライアンスは認証ドメインとして xyz.bar.com. を割り当てます。

CLI を使用してトラフィック管理仮想サーバを設定するには

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

```
set lb vserver <name>  -authentication ON -authenticationhost <FQDN> [-
    authenticationdomain <authdomain>]
show lb vserver <name>
set cs vserver <name>  -authentication ON -authenticationhost <FQDN> [-
    authenticationdomain <authdomain>]
show cs vserver <name>
<!--NeedCopy-->
```

例**:**


```
set lb vserver vs-cont-sw -Authentication ON -AuthenticationHost mywiki
    .index.com Done

show lb vserver vs-cont-sw vs-cont-sw (0.0.0.0:0) - TCP Type: ADDRESS
    State: DOWN Last state change was at Wed Aug 19 10:03:15 2009 (+410
    ms) Time since last state change: 5 days, 20:00:40.290 Effective
    State: DOWN Client Idle Timeout: 9000 sec Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED No. of Bound Services : 0
     (Total) 0 (Active) Configured Method: LEASTCONNECTION Mode: IP
    Persistence: NONE Connection Failover: DISABLED Authentication: ON
    Host: mywiki.index.com
Done
<!--NeedCopy-->
```

GUI を使用してトラフィック管理仮想サーバを設定するには

1. ナビゲーションペインで、次のいずれかの操作を行います。
   - **Traffic Management > Load Balancing > Virtual Servers** に移動します。
   - [トラフィック管理**] > [**コンテンツスイッチング**] > [**仮想サーバー] に移動します。
   - 詳細ウィンドウで、認証を有効にする仮想サーバーを選択し、[編集] をクリックします。
   - [ドメイン] テキストボックスに、認証ドメインを入力します。
   - 右側の [詳細設定] メニューで、[認証] を選択します。
   - [フォームベース認証] または [**401** ベース認証] のいずれかを選択し、認証情報を入力します。
     - [フォームベース認証] で、認証 FQDN（認証サーバの完全修飾ドメイン名）、認証仮想サーバ（認証仮想サーバの IP アドレス）、および認証プロファイル（認証に使用するプロファイル）を入力します。
     - 401 ベースの認証の場合は、認証仮想サーバと認証プロファイルのみを入力します。
   - [**OK**] をクリックします。ステータスバーに、仮想サーバーが正常に構成されたことを示すメッセージが表示されます。

認証、承認、監査の簡素化されたログインプロトコルのサポート

認証、認可、および監査トラフィック管理仮想サーバと認証、認可、および監査仮想サーバ間のログインプロトコルは、クエリパラメータを介して暗号化されたデータを送信するのではなく、内部メカニズムを使用するように簡素化されています。この機能を使用すると、リクエストのリプレイが防止されます。

DNS を構成する

認証プロセスで使用されるドメインセッション Cookie が正しく機能するには、認証とトラフィック管理の両方の仮想サーバーを同じドメイン内の FQDN に割り当てるように DNS を構成する必要があります。DNS アドレスレコードを構成する方法については、「ドメインネームシステム」を参照してください。

認証仮想サーバーを確認する

認証およびトラフィック管理仮想サーバーを構成した後、ユーザーアカウントを作成する前に、両方の仮想サーバーが正しく構成され、UP 状態であることを確認する必要があります。

CLI を使用した **NoAuth** 認証の設定

コマンドプロンプトで、次のコマンドを入力します。

```
show authentication vserver <name>
<!--NeedCopy-->
```

例**:**

```
show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
State: UP
Client Idle Timeout: 180 sec
Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com
Done
<!--NeedCopy-->
```

GUI を使用して **NoAuth** 認証を構成する

1. [セキュリティ**] > [NetScaler AAA-**アプリケーショントラフィック**] > [**仮想サーバー] に移動します。
   注：NetScaler Gateway から、**NetScaler Gateway**> 仮想サーバーに移動します。
2. [**AAA Virtual Servers**] ペインの情報を確認して、設定が正しいこと、および認証仮想サーバがトラフィックを受け入れていることを確認します。特定の仮想サーバーを選択して、詳細ペインに詳細情報を表示できます。

承認ポリシー

August 15, 2023

認可ポリシーを設定するときに、内部ネットワークのネットワークリソースへのアクセスを許可または拒否するように設定できます。たとえば、ユーザが 10.3.3.0 ネットワークにアクセスできるようにするには、次の式を使用します。

CLIENT.IP.DST.IN_SUBNET (10.3.0.0/16)

承認ポリシーは、ユーザーおよびグループに適用されます。ユーザーが認証されると、NetScaler Gateway は、RADIUS、LDAP、または TACACS+ サーバーからユーザーのグループ情報を取得することにより、グループ認証チェックを実行します。ユーザーがグループ情報にアクセスできる場合、NetScaler Gateway はそのグループに許可されているネットワークリソースを確認します。

ユーザーがアクセスできるリソースを制御するには、承認ポリシーを作成する必要があります。認可ポリシーを作成する必要がない場合は、デフォルトのグローバル認可を設定できます。

ファイルパスへのアクセスを拒否する式を認可ポリシー内に作成する場合は、サブディレクトリパスだけを使用でき、ルートディレクトリは使用できません。たとえば、fs.path には「\\ rootdir\\ dir1\\ dir2」が含まれているのではなく、fs.path には「\\ dir1\\ dir2」が含まれています。この例で 2 番目のバージョンを使用すると、ポリシーは失敗します。

承認ポリシーを設定したら、それをユーザーまたはグループにバインドします。

デフォルトでは、認可ポリシーは、仮想サーバにバインドされたポリシーに対して最初に検証され、次にグローバルにバインドされたポリシーに対して検証されます。ポリシーをグローバルにバインドし、ユーザー、グループ、または仮想サーバにバインドするポリシーよりもグローバルポリシーを優先する場合は、ポリシーのプライオリティ番号を変更できます。プライオリティ番号は 0 から始まります。プライオリティ番号が小さいほど、ポリシーの優先順位が高くなります。

たとえば、グローバルポリシーのプライオリティ番号が 1 で、ユーザのプライオリティが 2 の場合、グローバル認証ポリシーが最初に適用されます。

> 重要:
>
> - 従来の認可ポリシーは、TCP トラフィックにのみ適用されます。
> - 高度な承認ポリシーは、すべてのタイプのトラフィックに適用できます (TCP/UDP/ICMP/DNS).
>   - To apply policy on UDP/ICMP/DNS traffic, policies must be bound at type UDP_REQUEST, ICMP_REQUEST, and DNS_REQUEST respectively.
>   - While binding, if “type” is not explicitly mentioned or “type” is set to REQUEST, the behavior does not change from earlier builds, that is these policies are applied only to TCP traffic.

> – The policies bound at UDP_REQUEST do not apply for DNS traffic. For DNS, policies must be explicitly bound to DNS_REQUEST TCP_DNS is similar to other TCP requests.

高度な承認ポリシーの詳細については、記事https://support.citrix.com/article/CTX232237を参照してください。

承認ポリシーの設定とバインド

GUI を使用して承認ポリシーを設定する

1. **NetScaler Gateway >** ポリシー **>** 承認に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **[**名前] に、ポリシーの名前を入力します。
4. 「アクション」で、「許可」または「拒否」を選択します。
5. 「式」で、「式エディタ」をクリックします。
6. 式の設定を開始するには、[選択] をクリックし、必要な要素を選択します。
7. 式が完成したら、[完了] をクリックします。
8. [作成] をクリックします。

GUI を使用して承認ポリシーをユーザーにバインドする

1. [**NetScaler** ゲートウェイ**] > [**ユーザー管理] に移動します。
2. [**AAA** ユーザ] をクリックします。
3. 詳細ペインで、ユーザーを選択し、[編集] をクリックします。
4. [詳細設定] で、[承認ポリシー] をクリックします。
5. [ポリシーのバインド] ページで、ポリシーを選択するか、ポリシーを作成します。
6. [優先度] で、優先度番号を設定します。
7. 「タイプ」でリクエストのタイプを選択し、「**OK**」をクリックします。

GUI を使用して承認ポリシーをグループにバインドする

1. [**NetScaler** ゲートウェイ**] > [**ユーザー管理] に移動します。
2. [**AAA** グループ] をクリックします。
3. 詳細ペインでグループを選択し、[編集] をクリックします。
4. [詳細設定] で、[承認ポリシー] をクリックします。
5. [ポリシーのバインド] ページで、ポリシーを選択するか、ポリシーを作成します。
6. [優先度] で、優先度番号を設定します。
7. 「タイプ」でリクエストのタイプを選択し、「**OK**」をクリックします。

承認は、ユーザーが NetScaler Gateway にログオンしたときにアクセスできるネットワークリソースを指定します。承認のデフォルト設定では、すべてのネットワークリソースへのアクセスを拒否します。デフォルトのグローバル設定を使用し、承認ポリシーを作成して、ユーザーがアクセスできるネットワークリソースを定義することをお勧めします。

NetScaler Gateway での承認は、承認ポリシーと式を使用して構成します。承認ポリシーを作成したら、アプライアンスで構成したユーザーまたはグループに承認ポリシーをバインドできます。

デフォルトのグローバル認証

ユーザが内部ネットワークでアクセスできるリソースを定義するには、デフォルトのグローバル認可を設定します。グローバル認可を設定するには、内部ネットワーク上のネットワークリソースへのアクセスをグローバルに許可または拒否します。

作成したグローバル認可アクションは、直接またはグループを介して、関連付けられた認可ポリシーをまだ持っていないすべてのユーザに適用されます。ユーザまたはグループの認可ポリシーは、常にグローバル認可アクションを上書きします。デフォルトの承認アクションが [拒否] に設定されている場合、すべてのユーザーまたはグループに承認ポリシーを適用して、それらのユーザーまたはグループがネットワークリソースにアクセスできるようにする必要があります。この要件は、セキュリティの向上に役立ちます。

デフォルトのグローバル認証を設定するには、次の手順を実行します。

1. 構成ユーティリティの［構成］タブのナビゲーションペインで、NetScaler Gateway を展開し、［グローバル設定］をクリックします。
2. 詳細ペインの [設定] で、[グローバル設定の変更] をクリックします。
3. [セキュリティ] タブの [デフォルトの承認アクション] の横にある [許可] または [拒否] を選択し、[OK] をクリックします。

認証プロファイル

February 15, 2024

複数のトラフィック管理仮想サーバーで同じ認証設定を使用する場合は、認証仮想サーバー、認証ホスト、認証ドメイン、および認証レベルを指定する認証プロファイルを作成できます。

この認証プロファイルは、関連するトラフィック管理仮想サーバーに関連付けることができます。

認証プロファイルを設定する

CLI を使用して認証プロファイルを設定する

- 認証プロファイルを作成し、必要なパラメータを設定します。

たとえば、「AuthVS」という名前の認証仮想サーバーを使用してプロファイルを作成する場合などです。

```
add authentication authnProfile authProfile1 -authnVsName authVS
   -authenticationHost authnVS.example.com -authenticationDomain
   example.com -authenticationLevel
<!--NeedCopy-->
```

> 注:
>
> 認証の重みまたはレベルは、トラフィックがバインドされている仮想サーバーによって異なります。特定のレベルのトラフィック管理仮想サーバーに対する認証によって作成されたセッションは、上位レベルのトラフィック管理仮想サーバーへのアクセスには使用できません。

- 認証プロファイルを関連するトラフィック管理仮想サーバーにバインドします。

  たとえば、AuthProfile1 を「vserver1」という名前の負荷分散仮想サーバーにバインドする場合などです。

```
set lb vserver vserver1 -authnProfile authProfile1
<!--NeedCopy-->
```

GUI を使用して認証プロファイルを設定する

[設定] タブで、[セキュリティ] **> [AAA-**アプリケーショントラフィック] **> [**認証プロファイル] に移動し、必要に応じて認証プロファイルを設定します。

> 注:
>
> - NetScaler Gateway ウィザードを使用して認証プロファイルを作成することもできます。プロファイルには、認証ポリシーのすべての設定が含まれています。プロファイルは、認証ポリシーの作成時に設定します。
> - NetScaler Gateway ウィザードでは、選択した認証タイプを使用して認証を構成できます。ウィザードの実行後に他の認証ポリシーを構成する場合は、構成ユーティリティを使用できます。NetScaler Gateway ウィザードの詳細については、「NetScaler Gateway ウィザードを使用した設定の構成」を参照してください。

認証ポリシー

February 15, 2024

ユーザーが NetScaler または NetScaler Gateway にログオンすると、作成したポリシーに従って認証されます。認証ポリシーは式とアクションで構成されます。認証ポリシーでは NetScaler の式を使用します。

認証アクションと認証ポリシーを作成したら、それを認証仮想サーバーにバインドし、優先度を割り当てます。バインドするときは、プライマリポリシーまたはセカンダリポリシーとして指定します。プライマリポリシーは、セカンダリポリシーよりも先に評価されます。両方のタイプのポリシーを使用する設定では、通常、プライマリポリシーはより限定的なポリシーですが、セカンダリポリシーは通常、より一般的なポリシーです。これは、より具体的な基準を満たさないユーザーアカウントの認証を処理することを目的としています。ポリシーは、認証タイプを定義します。単一の認証ポリシーは、単純な認証のニーズに使用でき、通常はグローバルレベルでバインドされます。デフォルトの認証タイプ (ローカル) を使用することもできます。ローカル認証を構成する場合は、NetScaler でユーザーとグループも構成する必要があります。

複数の認証ポリシーを設定し、それらをバインドして、詳細な認証手順と仮想サーバーを作成できます。たとえば、複数のポリシーを設定して、カスケード認証と 2 要素認証を設定できます。認証ポリシーの優先順位を設定して、NetScaler がユーザー資格情報をチェックするサーバーと順序を決定することもできます。認証ポリシーには、式とアクションが含まれます。たとえば、式を True 値に設定した場合、ユーザーのログオン時にアクションによってユーザーログオンが true と評価され、ユーザーはネットワークリソースにアクセスできます。

認証ポリシーを作成したら、ポリシーをグローバルレベルまたは仮想サーバにバインドします。少なくとも 1 つの認証ポリシーを仮想サーバーにバインドすると、グローバル認証の種類が仮想サーバーにバインドされたポリシーよりも優先されない限り、ユーザーが仮想サーバーにログオンするときに、グローバルレベルにバインドした認証ポリシーは使用されません。

ユーザーが NetScaler にログオンすると、認証は次の順序で評価されます。

- 仮想サーバは、バインドされた認証ポリシーがあるかどうかがチェックされます。
- 認証ポリシーが仮想サーバーにバインドされていない場合、NetScaler はグローバル認証ポリシーを確認します。
- 認証ポリシーが仮想サーバまたはグローバルにバインドされていない場合、ユーザはデフォルトの認証タイプで認証されます。

LDAP および RADIUS 認証ポリシーを構成し、2 要素認証用にポリシーをグローバルにバインドする場合は、設定ユーティリティでポリシーを選択し、ポリシーがプライマリ認証タイプかセカンダリ認証タイプかを選択できます。グループ抽出ポリシーを構成することもできます。

> 注:
>
> NetScaler または NetScaler Gateway は認証用に UTF-8 文字のみをエンコードし、ISO-8859-1 文字を使用するサーバーとは互換性がありません。

GUI について

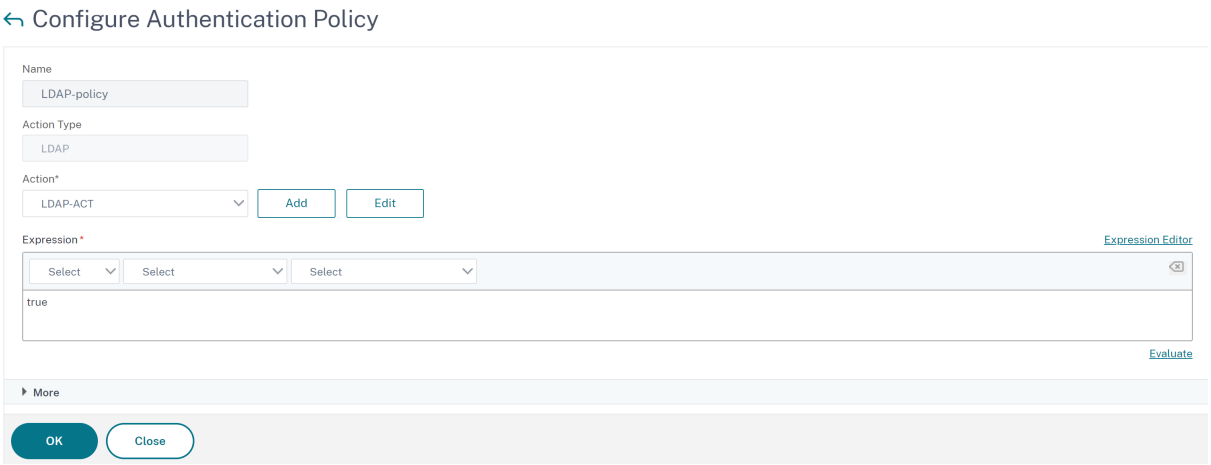
高度な認証ポリシーを作成または変更する

1. [セキュリティ] **>** [**AAA-**アプリケーショントラフィック] **>** [ポリシー] **>** [認証] **>** [詳細ポリシー] **>** [ポリシー] に移動します。

2. 認証ポリシーページで、次のタスクのいずれかを実行します。

   - 認証ポリシーを作成するには、「追加」をクリックします。「認証ポリシーの作成」ページが表示されます。
   - 必須フィールドを更新し、「作成」をクリックします。
   - 認証ポリシーを変更するには、アクションを選択し、[編集] をクリックします。「認証ポリシーの設定」ページが表示されます。必須フィールドを変更し、「**OK**」をクリックします。

     - 名前: 高度な認証ポリシーの名前。
     - アクションタイプ: ポリシーが作成される認証アクションのタイプ。
     - アクション: ポリシーが一致した場合に実行される認証アクション (LDAP、RADIUS、SAML) の名前。ドロップダウンリストに認証アクションがない場合は、[追加] をクリックします。
     - 式: 認証仮想サーバーでユーザーを認証するかどうかをポリシーが決定するために使用する NetScaler の名前付きルールまたは式の名前。高度なポリシー表現の詳細については、「高度なポリシー式」を参照してください。
     - ログアクション: リクエストがこのポリシーに一致する場合に使用するメッセージログアクションの名前。



認証ポリシーを削除する

ネットワークから認証サーバーを変更または削除した場合は、対応する認証ポリシーを NetScaler から削除してください。

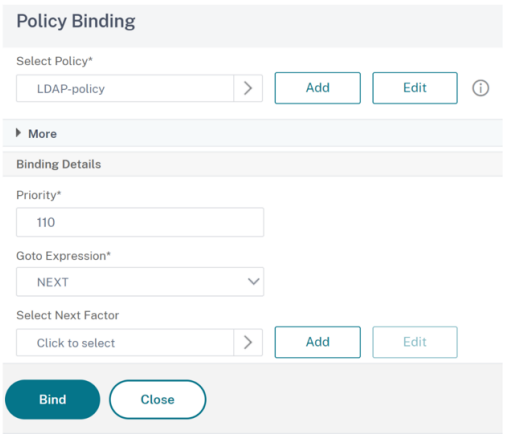
1. [セキュリティ] **>** **[AAA-**アプリケーショントラフィック] **>** **[**ポリシー] **>** **[**認証] **>** **[**詳細ポリシー] **>** **[**ポリシー] に移動します。
2. 「認証ポリシー」ページで、削除するポリシーを選択し、「削除」をクリックします。
3. [はい] をクリックして操作を確定します。

認証ポリシーをバインドする

高度な認証ポリシーを認証仮想サーバーにバインドし、認証プロファイルを使用して認証仮想サーバーを VPN 仮想サーバーにリンクする必要があります。

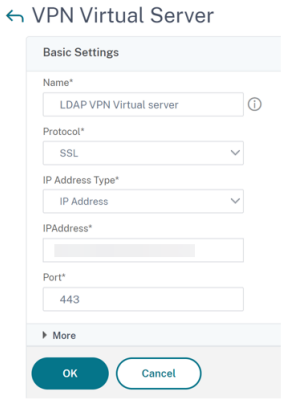
1. 認証仮想サーバーを作成します。

    - ［セキュリティ］ **>** ［**AAA-**アプリケーショントラフィック］ **>** ［仮想サーバ］に移動します。
    - 「仮想サーバーの認証」ページで、「追加」をクリックします。必須フィールドを更新し、「**OK**」をクリックします。
    - 認証仮想サーバーが存在する場合は、認証仮想サーバーページで関連するサーバーを選択します。
    - 「高度な認証ポリシー」セクションで、認証ポリシーを選択します。
    - 「ポリシーバインディング」ページでポリシーを選択し、「追加」をクリックします。優先度、GoTo 式、次の要素などのバインドの詳細を入力し、**[**バインド] をクリックします。



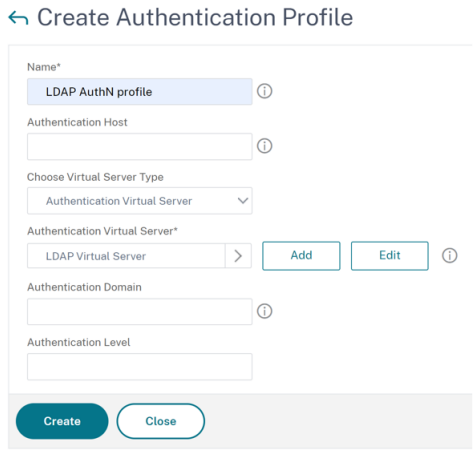
1. VPN 仮想サーバーを作成します。

    - [**NetScaler Gateway] > [**仮想サーバー] に移動します。
    - ［**NetScaler Gateway** 仮想サーバー］ページで、［追加］をクリックします。
    - 「**VPN** 仮想サーバー」ページで必須フィールドを更新し、「**OK**」をクリックします。



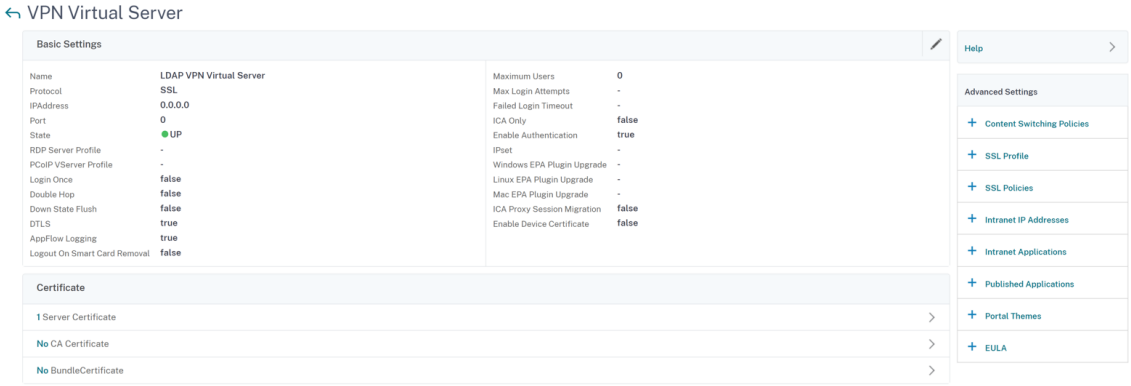
2. 認証プロファイルを作成します。

   - [セキュリティ] **> [AAA-**アプリケーショントラフィック] **> [**認証プロファイル] に移動します。
   - 「認証プロファイル」ページで、「追加」をクリックします。
   - 必須フィールドを更新し、「作成」をクリックします。



3. 認証プロファイルを使用して、認証仮想サーバーを VPN 仮想サーバーにリンクします。

   - [**NetScaler Gateway] > [NetScaler Gateway** 仮想サーバー] に移動し、VPN 仮想サーバーを選択します。「**VPN** 仮想サーバー」ページが表示されます。
   - 「認証プロファイル」セクションで、ドロップダウンリストから認証プロファイルを選択し、「**OK**」をクリックします。
   - または、**VPN Virtual server** ページの「詳細設定」セクションに移動し、「**+** 認証プロファイル」をクリックし、ドロップダウンリストから認証プロファイルを選択して、「**OK**」をクリックすることもできます。
   - [完了] をクリックします。



CLI で

コマンドプロンプトで、次のコマンドを入力します:

```
add authentication policy <name> -rule <expression> -action <string>

show authentication policy <name>

bind authentication vserver <name> -policy <policyname> [-priority <
    priority>][-secondary]

show authentication vserver <name>
<!--NeedCopy-->
```

例:

```
add authentication policy Authn-Pol-1 true


show authentication policy Authn-Pol-1
Name: Authn-Pol-1       Rule: true          Request action: LOCAL

bind authentication vserver Auth-Vserver-2 -policy Authn-Pol-1

show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT State: UP Client
    Idle
Timeout: 180 sec Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com
Primary authentication policy name:  Authn-Pol-1 Priority: 0
<!--NeedCopy-->
```


認証ポリシーの変更　コマンドプロンプトで次のコマンドを入力して、既存の認証ポリシーを変更します。

```
 set authentication policy <name> [-rule <expression>] [-action <string
    >] [-undefAction <string>] [-comment <string>] [-logAction <string
    >]
<!--NeedCopy-->
```

例

```
set authentication policy Authn-Pol-1 -rule true
<!--NeedCopy-->
```

認証ポリシーを削除する　コマンドプロンプトで次のコマンドを入力して、認証ポリシーを削除します。

```
rm authentication policy <name>
<!--NeedCopy-->
```

例

```
rm authentication localPolicy Authn-Pol-1
<!--NeedCopy-->
```

認証ポリシーをバインドする　高度なポリシーの場合は、認証プロファイルを作成し、そのプロファイルを認証仮想サーバーに関連付けます。認証プロファイルが作成されたら、認証プロファイルを VPN 仮想サーバーに関連付けます。

認証プロファイルを作成し、そのプロファイルに認証仮想サーバーを設定します。

```
 add authentication authnProfile <name> {
 -authnVsName <string> }
  {
 -AuthenticationHost <string> }
 {
 -AuthenticationDomain <string> }
 [-AuthenticationLevel <positive_integer>]

<!--NeedCopy-->
```

例

```
 add authentication authnProfile Authn-Prof-1 -authnVsName Auth-Vserver
    -2 -AuthenticationDomain "myCompany.employee.com"

<!--NeedCopy-->
```

VPN 仮想サーバーを作成し、対応する認証プロファイルを追加します。

例


```
add vpn vserver VPN-Vserver-2 ssl -authentication ON -authnprofile
    Authn-Prof-1
<!--NeedCopy-->
```

認証アクションを追加する

LOCAL 認証を使用しない場合は、明示的な認証アクションを追加する必要があります。コマンドプロンプトで、次のコマンドを入力します:

```
add authentication tacacsAction <name> -serverip <IP> [-serverPort <
    port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

例

```
add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
    serverport 1812 -authtimeout 15 -tacacsSecret "minotaur" -
    authorization OFF -accounting ON -auditFailedCmds OFF -
    defaultAuthenticationGroup "users"
<!--NeedCopy-->
```

認証アクションの設定

既存の認証アクションを構成するには、コマンドプロンプトで次のコマンドを入力します。

```
set authentication tacacsAction <name> -serverip <IP> [-serverPort <
    port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

例

```
set authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
    serverport 1812 -authtimeout 15 -tacacsSecret "minotaur" -
    authorization OFF -accounting ON -auditFailedCmds OFF -
    defaultAuthenticationGroup "users"
<!--NeedCopy-->
```

認証アクションを削除する

既存の RADIUS アクションを削除するには、コマンドプロンプトで次のコマンドを入力します。

```
rm authentication radiusAction <name>
<!--NeedCopy-->
```

例


```
rm authentication tacacsaction Authn-Act-1
<!--NeedCopy-->
```

NoAuth 認証

NetScaler は NoAuth 認証機能をサポートしているため、ユーザーがこのポリシーを実行するときにnoAuthAction、ユーザーがコマンドで DefaultAuthenticationGroup パラメーターを構成できます。管理者は、ユーザーのグループ内にこのグループが存在するかどうかをチェックして、NoAuth ポリシーによるユーザーのナビゲーションを判断できます。

NoAuth 認証を設定するには

コマンドプロンプトで次を入力します:

```
add authentication noAuthAction <name> [-defaultAuthenticationGroup <
    string>]
<!--NeedCopy-->
```

例

```
add authentication noAuthAction noauthact  -defaultAuthenticationGroup
    mynoauthgroup
<!--NeedCopy-->
```

デフォルトのグローバル認証タイプ

NetScaler Gateway をインストールして NetScaler Gateway ウィザードを実行すると、ウィザード内で認証を構成しました。この認証ポリシーは、NetScaler Gateway グローバルレベルに自動的にバインドされます。NetScaler Gateway ウィザード内で構成する認証タイプは、デフォルトの認証タイプです。NetScaler Gateway ウィザードを再度実行してデフォルトの認証タイプを変更するか、構成ユーティリティでグローバル認証設定を変更できます。

他の認証タイプを追加する必要がある場合は、構成ユーティリティを使用して NetScaler Gateway で認証ポリシーを構成し、そのポリシーを NetScaler Gateway にバインドできます。認証をグローバルに設定する場合は、認証のタイプを定義し、設定を構成し、認証できる最大ユーザ数を設定します。

ポリシーを設定してバインドしたら、優先度を設定して、優先する認証タイプを定義できます。たとえば、LDAP および RADIUS 認証ポリシーを設定します。LDAP ポリシーのプライオリティ番号が 10 で、RADIUS ポリシーのプライオリティ番号が 15 の場合、各ポリシーをバインドする場所に関係なく、LDAP ポリシーが優先されます。これをカスケード認証と呼びます。

ログオンページを NetScaler Gateway のインメモリキャッシュから配信するか、NetScaler Gateway で実行されている HTTP サーバーから配信するかを選択できます。ログオンページをインメモリキャッシュから配信するこ

とを選択した場合、NetScaler Gateway からのログオンページの配信は、HTTP サーバーからの配信よりも高速です。インメモリキャッシュからログオンページを配信するように選択すると、多数のユーザーが同時にログオンする場合の待ち時間が短縮されます。キャッシュからのログオンページの配信は、グローバル認証ポリシーの一部としてのみ構成できます。

また、認証用の特定の IP アドレスであるネットワークアドレス変換（NAT）IP アドレスを設定することもできます。この IP アドレスは認証用に固有のもので、NetScaler Gateway のサブネット、マップ、または仮想 IP アドレスではありません。この設定はオプションです。

> 注:
>
> - NetScaler Gateway ウィザードを使用して SAML 認証を構成することはできません。
> - クイック構成ウィザードを使用して、LDAP、RADIUS、およびクライアント証明書認証を構成できます。ウィザードを実行すると、NetScaler Gateway で構成されている既存の LDAP または RADIUS サーバーから選択できます。LDAP または RADIUS の設定を構成することもできます。2 要素認証を使用する場合は、主要な認証タイプとして LDAP を使用することをお勧めします。

デフォルトのグローバル認証タイプを構成する

1. GUI の［構成］タブのナビゲーションペインで［**NetScaler Gateway**］を展開し、［グローバル設定］をクリックします。
2. 詳細ウィンドウの [設定] で、[認証設定の変更] をクリックします。
3. [最大ユーザー数] に、この認証の種類を使用して認証できるユーザーの数を入力します。
4. [**NAT IP** アドレス] に、認証用の一意の IP アドレスを入力します。
5. [静的キャッシュを有効にする**]** を選択すると、ログオンページをより速く配信できます。
6. 認証が失敗した場合にユーザーにメッセージを表示するには、**[**拡張認証フィードバックを有効にする] を選択します。ユーザーが受け取るメッセージには、パスワードエラー、アカウントが無効またはロックされている、ユーザーが見つからない、などが含まれます。
7. [既定の認証タイプ] で、認証タイプを選択します。
8. 使用する認証タイプの設定を構成し、「**OK**」をクリックします。

ユーザーの現在のログイン試行回数の取得をサポート

NetScaler には、ユーザーの現在の`aaa.user.login_attempts`ログイン試行の値を式で取得するオプションがあります。この式は、引数を 1 つ (ユーザー名) とするか、引数なしのいずれかを取ります。引数がない場合、エクスプレッションは`aaa_session`または`aaa_info`からユーザ名を取得します。

`aaa.user.login_attempts`式を認証ポリシーとともに使用して、さらに処理することができます。

ユーザーごとのログイン試行回数を設定するには

コマンドプロンプトで入力します:

```
add expression er aaa.user.login_attempts
```

> 注:
>
> この`aaa.user.login_attempts`式は、「持続的ログイン試行」パラメーターが有効になっている場合は機能しません。永続的なログイン試行回数パラメーターの詳細については、「システムユーザーアカウントのロックアウト」を参照してください。

次の CLI の例では、さまざまなシナリオでのログイン試行設定について説明しています。

- 認証仮想サーバー

  ```
  set authentication vserver av_vs -maxLoginAttempts 5 -failedLoginTimeout
   100
  ```

  認証仮想サーバーとサポートされているパラメーターの詳細については、「認証仮想サーバー」を参照してください。

- NetScaler Gateway 仮想サーバー

  ```
  set vpn vserver vpn_vs -maxLoginAttempts 5 -failedLoginTimeout
  100
  ```

  NetScaler Gateway 仮想サーバーについて詳しくは、「仮想サーバー」セクションを参照してください。

- システムユーザーアカウントをロックアウトする

  ```
  set aaa parameter -maxloginAttempts 3 -failedLoginTimeout 10
  ```

  システムユーザーアカウントのロックアウトについて詳しくは、「システムユーザーアカウントを管理アクセス用にロックする」を参照してください。

  パラメータの説明**:**

  - `maxLoginAttempts`: ユーザがロックアウトされるまでに許可されるログイン試行の最大回数。
  - `failedLoginTimeout`: ログインが失敗するまでの許容秒数。ユーザーはログインプロセスを再開する必要があります。

ユーザーおよびグループ

August 15, 2023

認証、承認、および監査の基本設定を構成したら、ユーザーとグループを作成します。まず、NetScaler アプライアンスを介して認証する各ユーザーのユーザーアカウントを作成します。NetScaler アプライアンス自体によって制御されるローカル認証を使用している場合は、ローカルユーザーアカウントを作成し、それらの各アカウントにパスワードを割り当てます。

外部認証サーバーを使用している場合は、NetScaler アプライアンスでユーザーアカウントを作成することもできます。ただし、この場合、各ユーザーアカウントは外部認証サーバー上のそのユーザーのアカウントと完全に一致する必要があり、NetScaler で作成したユーザーアカウントにはパスワードを割り当てません。外部認証サーバーは、外部認証サーバーで認証するユーザーのパスワードを管理します。

外部認証サーバーを使用している場合でも、たとえば一時的なユーザー（訪問者など）のログインを許可したいが、認証サーバーでそれらのユーザー用のエントリを作成したくない場合などに、NetScaler アプライアンスでローカルユーザーアカウントを作成できます。すべてのユーザーアカウントにローカル認証を使用する場合と同様に、各ローカルユーザーアカウントにパスワードを割り当てます。

各ユーザーアカウントは、認証と承認のポリシーにバインドされている必要があります。このタスクを簡単にするために、1 つ以上のグループを作成し、それらにユーザーアカウントを割り当てることができます。その後、ポリシーを個々のユーザーアカウントではなくグループにバインドできます。

グループによるポリシーの設定

グループを構成したら、グループダイアログボックスを使用して、ユーザーアクセスを指定するポリシーと設定を適用できます。ローカル認証を使用している場合は、ユーザーを作成し、NetScaler Gateway で構成されているグループに追加します。その後、ユーザはそのグループの設定を継承します。

「グループ」ダイアログ・ボックスでは、ユーザー・グループに対して次のポリシーまたは設定を構成できます。

- ユーザー
- 承認ポリシー
- 監査ポリシー
- セッションポリシー
- トラフィックポリシー
- ブックマーク
- イントラネットアプリケーション
- イントラネット IP アドレス

構成では、複数のグループに属するユーザーがいる場合があります。さらに、各グループには、異なるパラメータが設定された 1 つ以上のバインドされたセッションポリシーがある場合があります。複数のグループに属するユーザーは、そのユーザーが属するすべてのグループに割り当てられたセッションポリシーを継承します。どのセッションポリシー評価が他方よりも優先されるかを確認するには、セッションポリシーの優先順位を設定する必要があります。

たとえば、group1 がホームページ www.homepage1.com で設定されたセッションポリシーにバインドされているとします。Group2 は、ホームページ www.homepage2.com で設定されたセッションポリシーにバインドされ

ています。これらのポリシーが、優先度番号のない、または同じ優先度番号を持つ各グループにバインドされている場合、両方のグループに属するユーザーに表示されるホームページは、最初に処理されるポリシーによって異なります。ホームページ www.homepage1.com のセッションポリシーに低い優先度の番号を設定すると、両方のグループに属するユーザーがホームページ www.homepage1.com を受信できるようになります。

セッションポリシーにプライオリティ番号が割り当てられていない場合、またはプライオリティ番号が同じ場合、precedence は次の順序で評価されます。

- ユーザー
- グループ
- 仮想サーバー
- グローバル

ポリシーがプライオリティ番号なしで同じレベルにバインドされている場合、またはポリシーのプライオリティ番号が同じ場合、評価の順序はポリシーのバインド順序に従います。最初にレベルにバインドされたポリシーは、後でバインドされたポリシーよりも優先されます。

ユーザーが複数のグループにバインドされ、各グループに IIP がバインドされている場合、そのユーザーはバインドされたグループから自由な IP を取得できます。

ユーザーとグループの作成

GUI を使用してローカルユーザーの認証、承認、監査を設定します

1. **[**** セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [NetScaler Gateway** からのユーザー] に移動し、[NetScaler Gateway] **> [**ユーザー管理] を展開して、[AAA ユーザー] をクリックします。**
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいユーザーアカウントを作成するには、[追加] をクリックします。
   - 既存のユーザーアカウントを変更するには、ユーザーアカウントを選択し、「開く」をクリックします。
3. 「**AAA** ユーザーの作成」ダイアログ・ボックスの「ユーザー名」テキスト・ボックスに、ユーザーの名前を入力します。
4. ローカル認証ユーザーアカウントを作成する場合は、「外部認証」チェックボックスをオフにして、ユーザーがログオンに使用するローカルパスワードを指定します。
5. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。ステータスバーに、ユーザーが正常に構成されたことを示すメッセージが表示されます。

構成ユーティリティを使用して、認証、承認、および監査のローカルグループを設定し、それらにユーザーを追加します

1. **[**** セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [NetScaler Gateway** からのグループ] に移動し、[NetScaler Gateway] **> [**ユーザー管理] を展開して、[AAA グループ] をクリックします。**
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいグループを作成するには、[追加] をクリックします。
   - 既存のグループを変更するには、グループを選択し、[編集] をクリックします。
3. 新しいグループを作成する場合は、**Create AAA Group** ダイアログボックスの「**GroupName** 」テキストボックスに、グループの名前を入力します。
4. 右側の「詳細設定」領域で、「**AAA** ユーザ」をクリックします。
   - グループにユーザーを追加するには、ユーザーを選択し、[追加] をクリックします。
   - グループからユーザーを削除するには、ユーザーを選択し、[削除] をクリックします。
   - 新しいユーザーアカウントを作成してグループに追加するには、プラスアイコンをクリックし、「構成ユーティリティを使用してローカルユーザーの認証、承認、および監査を設定するには」の指示に従ってください。
5. [作成] または [**OK**] をクリックします。作成したグループが **AAA Groups** ページに表示されます。

GUI を使用してグループを削除する

NetScaler Gateway からユーザーグループを削除することもできます。

1. **[**** セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [NetScaler Gateway** からのグループ] に移動し、[NetScaler Gateway] **> [**ユーザー管理**]** を展開して、 [AAA グループ] をクリックします。** 詳細ペインでグループを選択し、[削除] をクリックします。

CLI を使用してローカルユーザーの認証、承認、および監査を設定します

コマンドプロンプトで、次のコマンドを入力します。

```
add aaa group <groupname>

bind aaa group <groupname> -username <username>
<!--NeedCopy-->
```

例**:**

```
add aaa group group-2

bind aaa group group-2 -username user-2
```



```
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、認証、承認、および監査グループからユーザーを削除する

コマンドプロンプトで、グループにバインドされているユーザーアカウントごとに次のコマンドを 1 回入力して、ユーザーをグループからバインド解除します。

```
unbind aaa group <groupname> -username <username><!--NeedCopy-->
```

```
**例:**

<!--NeedCopy-->
```

unbind aaa group group-hr -username user-hr-1

```
### コマンドラインインターフェイスを使用して認証、承認、および監査グループを削除する

まず、グループからすべてのユーザーを削除します。次に、コマンドプロンプトで次のコマンドを入力してNetScaler AAAグループを削除し、構成を確認します。

<!--NeedCopy-->
```

rm aaa group

```
**例:**

<!--NeedCopy-->
```

rm aaa group group-hr

```
> **注:**
>
>ドメインなしでユーザー名がすでに追加されている場合、ドメインを使用してユーザー名を追加することはできません。ドメイン付きのユーザー名が最初に追加され、その後にドメインなしの同じユーザー名が追加された場合、NetScalerアプライアンスはそのユーザー名をユーザーリストに追加します。

次の例は、ドメインなしで同じユーザー名を追加した場合、ドメインを使用したユーザー名の追加が許可されないことを示しています。

<!--NeedCopy-->
```

> add aaa user u47985
> Done
> show aaa users


```
1) UserName: u47985
Done
add aaa user u47985@domain.com
ERROR: User already exists
‘‘‘
```

次の例は、ドメイン付きのユーザー名が最初に追加され、その後にドメインなしの同じユーザー名が追加された場合、NetScaler アプライアンスがそのユーザー名をユーザーリストに追加することを示しています。

```
> add aaa user u47985@domain.com
Done
> add aaa user u47985
Done
> sh aaa user
1)   UserName: u47985@domain.com
2)   UserName: u47985
```

‘‘‘

認証方法

December 8, 2023

NetScaler アプライアンスは、ローカルユーザーアカウントまたは外部認証サーバーを使用してユーザーを認証できます。アプライアンスでは、次の認証タイプがサポートされています。

- ローカル**:** 外部認証サーバーを参照せずに、パスワードを使用して NetScaler アプライアンスを認証します。ユーザーデータは NetScaler アプライアンスにローカルに保存されます。
- **RADIUS:** 外部の RADIUS サーバに対して認証を行います。
- **LDAP:** 外部の LDAP 認証サーバーに対して認証を行います。
- **TACACS:** 外部のターミナルアクセスコントローラアクセス制御システム (TACACS) 認証サーバに対して認証を行います。
- **CERT:** 外部認証サーバーを参照せずに、クライアント証明書を使用して NetScaler アプライアンスを認証します。
- ネゴシエーション**:** Kerberos 認証サーバーに対して認証を行います。Kerberos 認証にエラーがある場合、NetScaler は NTLM 認証を使用します。
- **SAML:** セキュリティアサーションマークアップ言語 (SAML) をサポートするサーバーに対して認証を行います。
- **SAML IDP:** NetScaler をセキュリティアサーションマークアップ言語 (SAML) ID プロバイダー (IdP) として機能するように構成します。

- **WEB: Web** サーバーに対して認証を行い、Web サーバーが HTTP リクエストで必要とする認証情報を提供し、Web サーバーの応答を分析してユーザー認証が成功したかどうかを判断します。
- ネイティブ **OTP:** NetScaler アプライアンスは、サードパーティのサーバーを使用しなくてもワンタイムパスワード (OTP) をサポートします。
- プッシュ通知**:** NetScaler Gateway は OTP のプッシュ通知をサポートしています。ユーザーは、登録したデバイスで受信した OTP を手動で入力して NetScaler Gateway にログインする必要はありません。管理者は、プッシュ通知サービスを使用してログイン通知がユーザーの登録デバイスに送信されるように NetScaler Gateway を構成できます。
- 電子メール **OTP:** 電子メール OTP 方式では、登録された電子メールアドレスに送信されるワンタイムパスワード (OTP) を使用して認証できます。いずれかのサービスで認証を試みると、サーバーは登録されているユーザーのメールアドレスに OTP を送信します。
- **reCAPTCHA** 認証**-NetScaler Gateway** は、**reCAPTCHA** の構成を簡素化する新しいファーストクラスアクション「**CaptchaAction** 」をサポートしています。reCAPTCHA はファーストクラスのアクションなので、それ自体が要因になり得ます。reCAPTCHA は nFactor フローのどこにでも注入できます。
- **nFactor** 認証**:** 多要素認証は、ユーザーがアクセスする際に複数の ID 証明を提供することを要求することで、アプリケーションのセキュリティを強化します。NetScaler アプライアンスは、多要素認証を構成するための拡張性と柔軟なアプローチを提供します。このアプローチは nFactor 認証と呼ばれます。
- **OAuth** 認証**:** OAuth 認証は、Google、Facebook、Twitter などのアプリケーションでホストされているサービスのユーザーを許可および認証します。

nFactor 認証

February 15, 2024

> 重要
>
> - nFactor 認証は、NetScaler 11.0 ビルド 62.x 以降でサポートされています。
> - nFactor 認証を NetScaler で機能させるには、アドバンストライセンスまたはプレミアムライセンスが必要です。
> - リリース 13.0 ビルド 67.x 以降、nFactor 認証は標準ライセンスでのみサポートされます。Gateway/VPN 仮想サーバー。NetScaler Gateway での nFactor 認証の詳細については、「ゲートウェイ認証の nFactor」を参照してください。
> - nFactor 認証は Linux クライアントではサポートされていません。

多要素認証では、ユーザーがアクセスするために複数の ID 証明を提供する必要があるため、アプリケーションのセキュリティが強化されます。NetScaler アプライアンスは、多要素認証を構成するための拡張性と柔軟なアプローチを提供します。このアプローチは *nFactor* 認証と呼ばれます。

nFactor 認証のしくみ

各認証ファクタは、次のタスクを実行します。

- ユーザーから認証情報を収集します。NetScaler がサポートする認証メカニズムには、LDAP、RADIUS、SAML アサーション、クライアント証明書、OAuth OpenID 接続、Kerberos などがあります。
- 指定されたクレデンシャルを評価して、認証が成功したか、失敗したか、またはグループ抽出、属性抽出などのアクションを実行するかを決定します。
- 評価結果に基づいて、アクセスが許可されるか、拒否されるか、次の要素が選択されます。
- 評価する次の要因がなくなるまで、これらの手順を繰り返します。

nFactor 認証を使用すると、次のことができます。

- 任意の数の認証要素を設定します。
- 前のファクタの実行結果に基づいて、次のファクタの選択を行います。
- ログインインターフェイスをカスタマイズします。たとえば、ラベル名、エラーメッセージ、およびヘルプテキストをカスタマイズできます。
- 認証を行わずにユーザーグループ情報を抽出します。
- 認証ファクタのパススルーを設定します。つまり、その要素には明示的なログイン操作は必要ありません。
- 異なるタイプの認証が適用される順序を設定します。NetScaler アプライアンスでサポートされている認証メカニズムはすべて、nFactor 認証設定の任意の要素として構成できます。これらの要因は、設定されている順序で実行されます。
- 認証が失敗したときに実行する必要がある認証要素に進むように NetScaler アプライアンスを構成します。そのためには、同じ条件で次に優先順位が高く、アクションが「NO_AUTH」に設定された別の認証ポリシーを設定します。次の要素を設定する必要があります。次の要素では、適用する代替認証メカニズムを指定する必要があります。

nFactor 認証のための **NetScaler Gateway** ログイン情報の暗号化

nFactor 認証を備えた NetScaler Gateway は、認証プロセス中にクライアント（ブラウザまたは SSO アプリ）から送信されたログイン要求フィールドを暗号化できます。暗号化されたログイン要求フィールドは、ユーザーの機密データが開示されないように保護するための追加のセキュリティレイヤーを提供します。

互換性のあるブラウザー

次の表に、ログイン暗号化をサポートするバージョンの詳細とブラウザの一覧を示します。

Web ブラウザー	バージョン
Chrome	78 以降
Firefox	69 以降
Edge	42 以降
Safari	11.0 以降
オペラ	66

互換性のあるクライアント

次のセクションでは、NetScaler Gateway のログイン情報の暗号化をサポートするクライアントとバージョンの詳細を一覧表示します。

- Mac の Citrix Workspace アプリは、OS バージョンが 10.14.x 以降の場合にのみ暗号化をサポートします。
- Mac の Citrix SSO アプリは、OS バージョンが 10.14.x 以降の場合にのみ暗号化をサポートします。
- Windows SSO アプリには、互換性に関する制限はありません。

CLI を使用してログイン暗号化を有効にするには

コマンドプロンプトで入力します:

```
set aaa parameter \[-loginEncryption \(ENABLED | DISABLED)]
```

> 注
>
> LoginEncryption パラメータは、デフォルトで無効になっています。有効にする必要があります。

GUI を使用してログイン暗号化を有効にするには

1. [セキュリティ**] > [AAA** ―アプリケーショントラフィック] に移動し、[認証設定**]** セクションで **[**** 認証 **AAA** 設定の変更 **] をクリックします。
2. **[AAA** パラメータの設定**]** ページで、[ログイン暗号化] オプションまでスクロールダウンし、有効にします。

> ログイン暗号化に関する重要な注意事項**:**
>
> NetScaler Gateway にログオンしようとすると、次のシナリオで、パスワード暗号化が失敗したため、ログオンを続行するにはパスワード暗号化を無効にする必要があるというエラーメッセージが表示されます。
>
> - ログイン暗号化が有効になっています。
> - サポートされていないブラウザが使用されています。

> ログイン暗号化を無効にする提案は無視してかまいません。ただし、ログオンを成功させるには、サポートされているブラウザを使用してください。

多要素（**nFactor**）の概念、エンティティ、用語

August 15, 2023

このトピックでは、nFactor 認証に関係する主要なエンティティのいくつかとその重要性について説明します。

ログインスキーマ

nFactor は、ユーザーインターフェイスである‘view’を、ランタイム処理である‘model’と切り離します。nFactor のビューはログインスキーマによって定義されます。ログインスキーマは、ユーザーに表示される内容を定義し、ユーザーからデータを抽出する方法を指定するエンティティです。

ビューを定義するために、ログインスキーマはログオンフォームを定義するディスク上のファイルを指します。このファイルは、「Citrix 共通形式プロトコル」の仕様に準拠している必要があります。このファイルは基本的に、ログオンフォームの XML 定義です。

ログインスキーマには、XML ファイルに加えて、ユーザーのログイン要求からユーザー名とパスワードを収集するための高度なポリシー式が含まれています。これらの式はオプションであり、user からのユーザー名とパスワードが予想されるフォーム変数名で届いた場合は省略できます。

ログインスキーマは、現在の資格情報セットをデフォルトの SingleSignon 資格情報として使用する必要があるかどうかも定義します。

ログインスキーマを作成するには、次の CLI コマンドを実行します。

```
1    add authentication loginSchema <name> -authenticationSchema <string>
         [-userExpression <string>] [-passwdExpression <string>] [-
        userCredentialIndex <positive_integer>] [-passwordCredentialIndex
         <positive_integer>] [-authenticationStrength <positive_integer>]
         [-SSOCredentials ( YES | NO )]
2 <!--NeedCopy-->
```

> 注**:**
>
> **ssoCredentials** は、現在の要素認証情報がデフォルトの SSO 認証情報であるかどうかを示します。デフォルト値は NO です。
>
> nFactor 認証設定では、デフォルトで SSO に最終要素の認証情報が使用されます。**ssoCredentials** 構成を使用すると、現在の要素の認証情報を使用できます。この構成が異なるファクタで設定される場合、この構成が設定された最後のファクタが優先されます。

各パラメータの詳細については、「https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-loginSchema/#add-authentication-loginschema」を参照してください。

ポリシーラベル

ポリシーラベルはポリシーの集まりです。これは NetScaler のポリシーインフラストラクチャと異質な構成ではありません。ポリシーラベルは認証要素を定義します。つまり、ユーザーからの認証情報が満たされているかどうかを判断するのに必要なすべてのポリシーが含まれています。ポリシーラベル内のポリシーはすべて同種であると見なすことができます。認証用のポリシーラベルは、書き換えなど、異なるタイプのポリシーを使用できません。言い換えると、ポリシーラベル内のすべてのポリシーは、ほとんどの場合、ユーザーからの同じパスワード/クレデンシャルを検証します。PolicyLabel のポリシーの結果は、論理 OR 条件に従います。したがって、最初のポリシーで指定された認証が成功すると、そのポリシーに続く他のポリシーはスキップされます。

ポリシーラベルは、次の CLI コマンドを実行して作成できます。

```
1 add authentication policy label mylabel  - loginSchema <>
2 <!--NeedCopy-->
```

ポリシーラベルは、ログインスキーマをプロパティとして使用します。ログインスキーマは、そのポリシーラベルのビューを定義します。ログインスキーマが指定されていない場合、暗黙的なログインスキーマ LSCHEMA_INT がそのポリシーラベルに関連付けられます。ログインスキーマは、ポリシーラベルがパススルーになるかどうかを決定します。

仮想サーバラベル

NetScaler の高度なポリシーインフラストラクチャでは、仮想サーバーは暗黙のポリシーラベルでもあります。これは、仮想サーバを複数のポリシーにバインドできるためです。ただし、仮想サーバはクライアントトラフィックのエントリポイントであり、異なるタイプのポリシーを取ることができるため、仮想サーバは特殊です。各ポリシーは、仮想サーバ内で独自のラベルの下に配置されます。したがって、仮想サーバはラベルの集合体です。

次の因子

ポリシーが仮想サーバまたはポリシーラベルにバインドされる場合は常に、次の要素で指定できます。次の要素は、特定の認証が成功した場合に何をする必要があるかを決定します。次の要素がない場合、そのユーザーの認証プロセスは終了です。

仮想サーバまたはポリシーラベルにバインドされたポリシーごとに、次の要素が異なる場合があります。これにより、すべてのポリシーが成功したときにユーザー認証の新しいパスを定義できる、究極の柔軟性が得られます。管理者はこの事実を利用して、特定のポリシーを満たしていないユーザーに対して巧妙なフォールバックファクターを作成できます。

認証なしポリシー

nFactor は NO_AUTHN と呼ばれる特別なビルトインポリシーを導入しています。NO_AUTHN ポリシーは常に認証結果として成功を返す。No-authポリシーは、次の CLI コマンドを実行して作成できます。

```
add authentication policy noauthpolicy  -rule <> -action NO_AUTHN
<!--NeedCopy-->
```

コマンドに従って、no-authentication ポリシーは任意の高度なポリシー式であることができるルールを取ります。認証結果は常に NO_AUTHN から成功する。

no-authポリシー自体は価値を付加するものではないようです。ただし、パススルーポリシーラベルとともに使用すると、ユーザ認証フローを促進する論理的な決定を柔軟に行うことができます。NO_AUTHN ポリシーとパススルーファクターは、nFactor の柔軟性に新たな次元を提供します。

注**:** 以降のセクションで、no-auth およびパススルーの使用方法を示す例を確認してください。

パススルー・ファクタ/ラベル

ユーザが仮想サーバで（第 1 要素の）認証に合格すると、以降の認証はポリシーラベルまたはユーザ定義（2 次）要素で行われます。
すべてのポリシーラベル/ファクタはログインスキーマエンティティに関連付けられ、そのファクタのビューが表示されます。これにより、ユーザーが特定の要素に到達するまでの経路に基づいてビューをカスタマイズできます。

ログインスキーマを明示的に指していない特殊な種類のポリシーラベルがあります。特殊なポリシーラベルは、実際にはビューの XML ファイルを指していないログインスキーマを指しています。これらのポリシーラベル/ファクターは「パススルー」ファクターと呼ばれます。

パススルー係数は、次の CLI コマンドを実行して作成できます。

例 1:

```
add authentication policylabel example1
<!--NeedCopy-->
```

例 2:

```
add loginschema passthrough_schema  -authenticationSchema noschema

add authentication policylabel example2  -loginschema
   passthrough_schema
<!--NeedCopy-->
```

パススルーファクターは、認証、承認、および監査サブシステムがユーザーに戻ってそのファクターのクレデンシャルセットを取得してはならないことを意味します。その代わり、認証、承認、および監査は、すでに取得した認証情報を使用して続行するためのヒントになります。これは、ユーザーの介入が望ましくない場合に便利です。たとえば、次のようにします。

- ユーザに 2 つのパスワードフィールドが表示された場合、1 つ目のファクタの後に 2 番目のファクタにユーザの介入は必要ありません。
- あるタイプ (証明書など) の認証が行われ、管理者がそのユーザーのグループを抽出する必要がある場合。

パススルー係数をNO_AUTHポリシーとともに使用して、条件付きジャンプを行うことができます。

nFactor 認証フロー

認証は常に nFactor の仮想サーバで開始されます。仮想サーバーは、ユーザーの第 1 の要素を定義します。ユーザーに最初に表示されるフォームは、仮想サーバーによって提供されます。ユーザーに表示されるログオンフォームは、ログインスキーマポリシーを使用して仮想サーバーでカスタマイズできます。ログインスキーマポリシーがない場合、ユーザー名とパスワードのフィールドが 1 つだけユーザーに表示されます。

カスタマイズされたフォームで複数のパスワードフィールドをユーザーに表示する必要がある場合は、ログインスキーマポリシーを使用する必要があります。構成された規則 (イントラネットユーザーと外部ユーザー、サービスプロバイダー A とサービスプロバイダー B など) に基づいて異なるフォームを表示できます。

ユーザー資格情報が投稿されると、最初の要素である認証仮想サーバーで認証が開始されます。認証仮想サーバには複数のポリシーを設定できるため、各ポリシーが順番に評価されます。任意の時点で、認証ポリシーが成功すると、そのポリシーに対して指定された次の要素が使用されます。次の要素がない場合、認証プロセスは終了します。次の因子が存在する場合、その因子が通過因子か標準因子かがチェックされます。パススルーの場合、その要素に関する認証ポリシーはユーザーの介入なしに評価されます。それ以外の場合は、そのファクタに関連付けられたログインスキーマがユーザに表示されます。

パススルーファクターと非認証ポリシーを使用して論理的な決定を下す例

管理者はグループに基づいて NextFactor を決定したいと考えています。

```
add authentication policylabel group check

add authentication policy admin group -rule http.req.user.is_member_of("Administrators") -action NO_AUTHN

add authentication policy nonadmins -rule true -action NO_AUTHN

bind authentication policy label group check -policy admingroup -pri 1 -nextFactor factor-for-admin

bind authentication policy label groupcheck -policy nonadmins -pri 10 -nextfactor factor-for-others

add authentication policy first_factor_policy -rule <> -action <>

bind authentication vserver <> -policy first_factor_policy -priority 10 -nextFactor groupcheck
<!--NeedCopy-->
```


多要素（**nFactor**）認証の構成

December 8, 2023

nFactor 設定を使用して、複数の認証要素を設定できます。nFactor 構成は、NetScaler Advanced エディションとプレミアムエディションでのみサポートされています。

nFactor を設定するメソッド

nFactor 認証は、次のいずれかの方法で設定できます。

- **nFactor** ビジュアライザー**:** nFactor ビジュアライザーを使用すると、単一のペインでファクターまたはポリシーラベルを簡単にリンクでき、同じペインでファクターのリンクを変更することもできます。ビジュアライザーを使用して nFactor フローを作成し、そのフローを認証、承認、および監査仮想サーバーにバインドできます。nFactor Visualizer の詳細と、ビジュアライザーを使用した nFactor の設定例については、nFactor ビジュアライザーの簡単な設定を参照してください。
- **NetScaler GUI:** 詳細については、「**nFactor** 構成に関連する構成要素」セクションを参照してください。
- **NetScaler CLI**： NetScaler CLI を使用した nFactor 構成のサンプルスニペットについては、「NetScaler CLI を使用した nFactor 構成のサンプルスニペット」を参照してください。

> 重要：このトピックには、NetScaler GUI を使用して nFactor を構成する方法の詳細が含まれています。

nFactor の設定に関係する設定要素

nFactor の設定には、次の要素が含まれます。詳細な手順については、このトピックの該当するセクションを参照してください。

構成要素	実行するタスク
AAA 仮想サーバ	AAA 仮想サーバを作成する
ログインスキーマ	ログインスキーマプロファイルの設定
高度な認証ポリシー	高度な認証ポリシーの作成

構成要素	実行するタスク
認証ポリシーラベル	認証ポリシーラベルの作成
NetScaler Gateway 用 **n** ファクター	NetScaler AAA 仮想サーバーを NetScaler Gateway 仮想サーバーとリンクするための認証プロファイルの作成

nFactor のしくみ

ユーザーが認証、承認、監査または NetScaler Gateway 仮想サーバーに接続すると、発生するイベントの順序は次のとおりです。

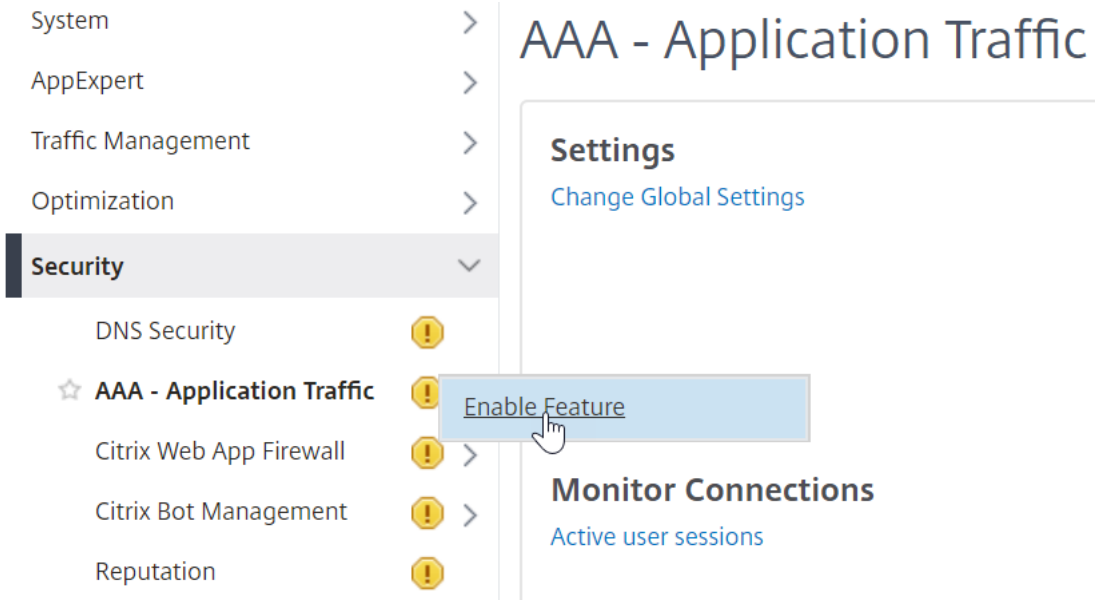
1. フォームベース認証を使用すると、認証、承認、および監査仮想サーバーにバインドされたログインスキーマが表示されます。
2. 認証、承認、および監査仮想サーバーにバインドされた高度な認証ポリシーが評価されます。
    - 高度な認証ポリシーが成功し、次の要素 (認証ポリシーラベル) が構成されると、次の要素が評価されます。Next Factor が設定されていない場合、認証は完了し、成功します。
    - 高度な認証ポリシーが失敗し、[式に移動] が [次へ] に設定されている場合、次にバインドされた高度な認証ポリシーが評価されます。いずれの高度な認証ポリシーも成功しない場合、認証は失敗します。
3. ネクストファクタ認証ポリシーラベルにログインスキーマがバインドされている場合は、ユーザーに表示されます。
4. 次の要素認証ポリシーラベルにバインドされた高度な認証ポリシーが評価されます。
    - 詳細認証ポリシーが成功し、次の要素 (認証ポリシーラベル) が構成されると、次の要素が評価されます。
    - Next Factor が設定されていない場合、認証は完了し、成功します。
5. 詳細認証ポリシーが失敗し、[式に移行] が [次へ] の場合、次にバインドされた高度な認証ポリシーが評価されます。
6. ポリシーが成功すると、認証は失敗します。

仮想サーバの認証、承認、監査

NetScaler Gateway で nFactor を使用するには、まず認証、承認、および監査仮想サーバーで nFactor を構成します。その後、認証、承認、監査の仮想サーバーを NetScaler Gateway 仮想サーバーにリンクします。

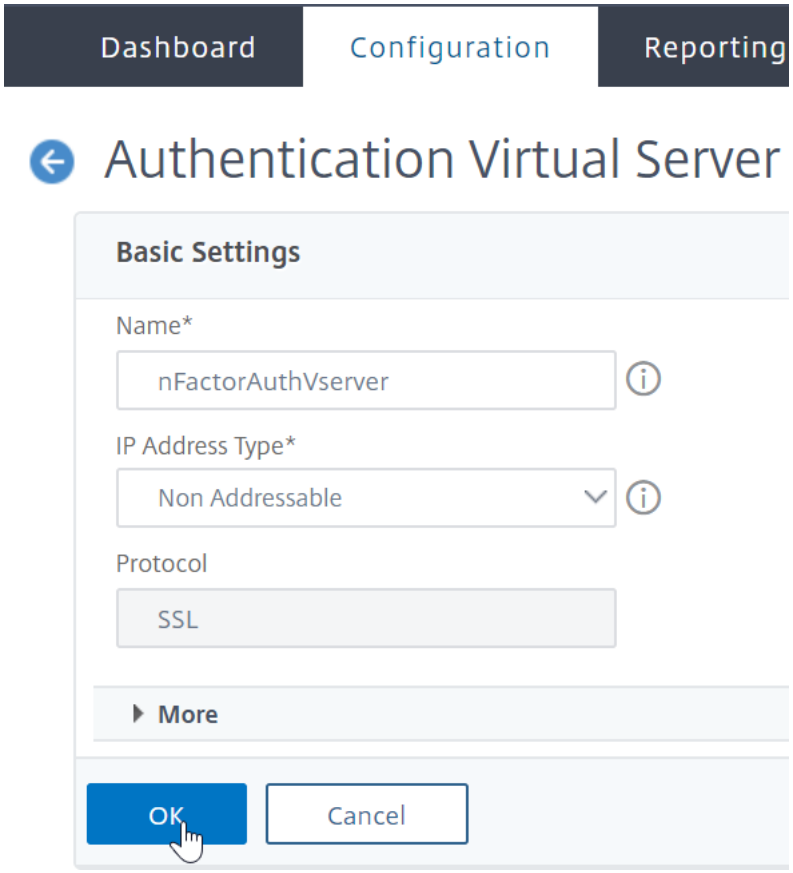
認証、承認、監査用仮想サーバーの作成

1. 認証、認可、および監査機能がまだ有効になっていない場合は、[セキュリティ] **> [AAA** ―アプリケーショントラフィック] に移動し、右クリックしてこの機能を有効にします。

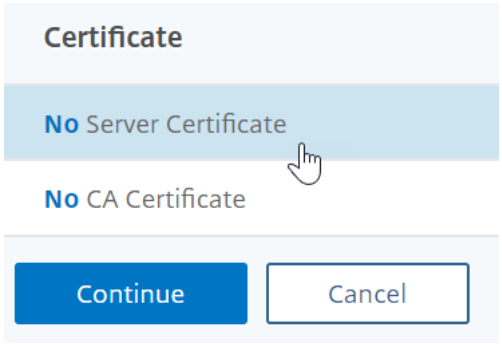


2. 設定 **>** セキュリティ **> AAA-**アプリケーショントラフィック **>** 仮想サーバにナビゲートして下さい。
3. [追加] をクリックして、認証仮想サーバーを作成します。
4. 次の情報を入力し、「**OK**」をクリックします。

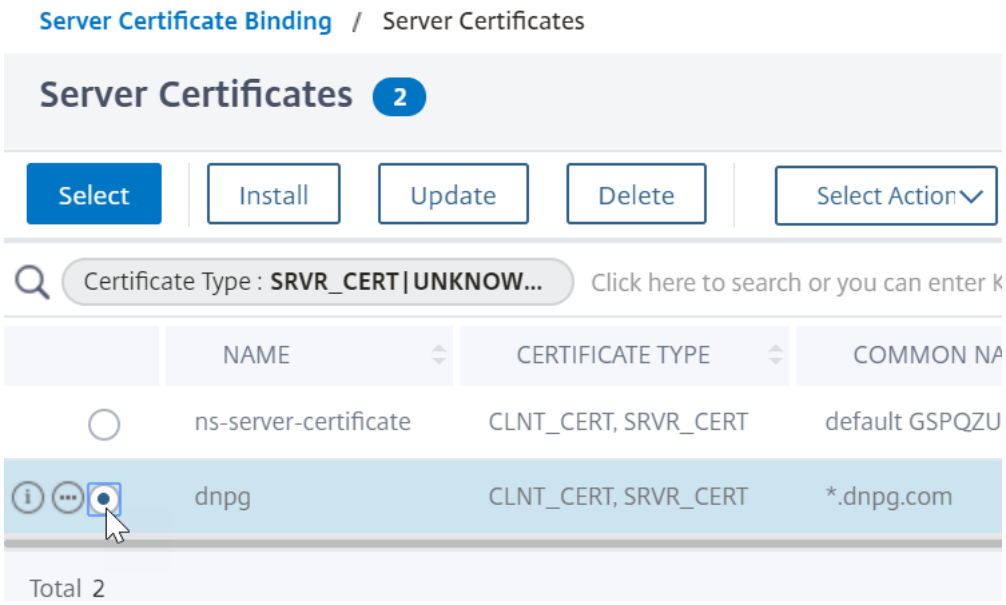
パラメーター名	パラメータの説明
Name	認証、承認、および監査仮想サーバーの名前。
IP アドレスタイプ	この仮想サーバーを NetScaler Gateway でのみ使用する場合は、[IP アドレスの種類] を［アドレス指定不可］に変更します。



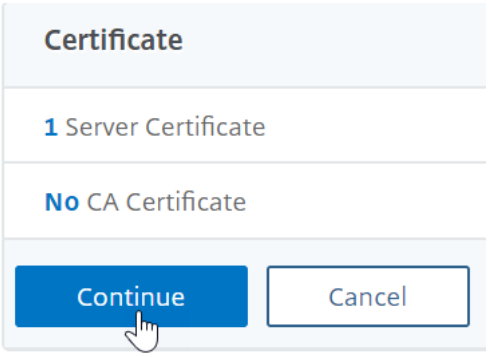
5. [証明書] で、[サーバー証明書なし] を選択します。



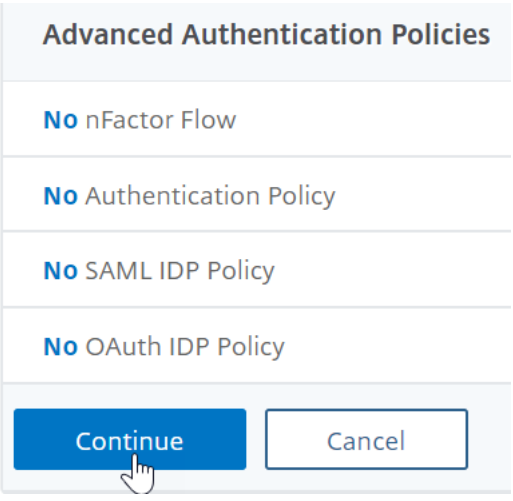
6. [クリックして選択] というテキストをクリックし、サーバー証明書を選択します。
7. 認証、承認、および監査用の仮想サーバーの証明書の横にあるラジオボタンをクリックし、[選択] をクリックします。このサーバーには直接アクセスできないため、選択した証明書は関係ありません。



8. [**Bind**] をクリックします。
9. [続行] をクリックして、[証明書] セクションを閉じます。

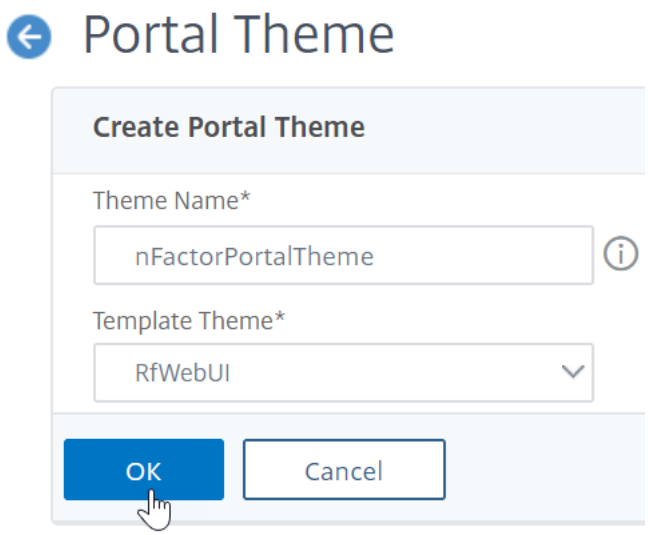


10. [続行] をクリックします。



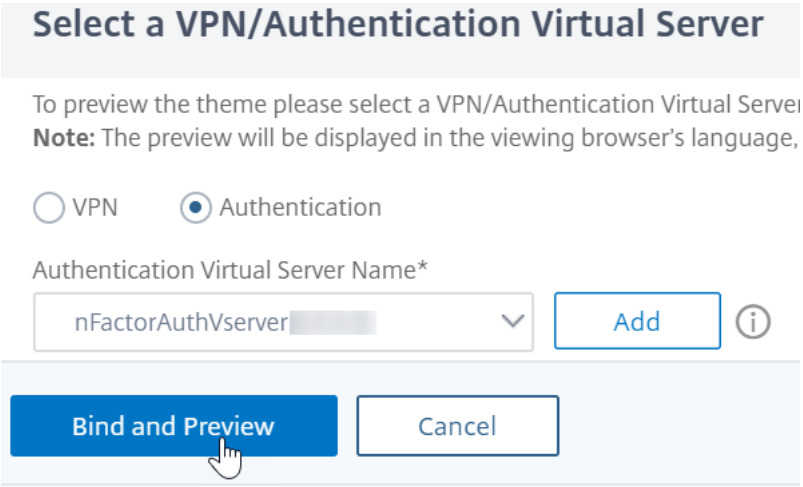
ポータル・テーマを認証、承認、監査用仮想サーバーにバインドする

1. **NetScaler Gateway >** ポータルテーマに移動し、テーマを追加します。NetScaler Gateway でテーマを作成し、後で認証、承認、および監査仮想サーバーにバインドします。
2. rfWebUI テンプレートテーマに基づいてテーマを作成します。



3. 必要に応じてテーマを調整した後、ポータル・テーマ編集ページの上部で、[クリックしてバインドして構成済みテーマを表示] をクリックします。

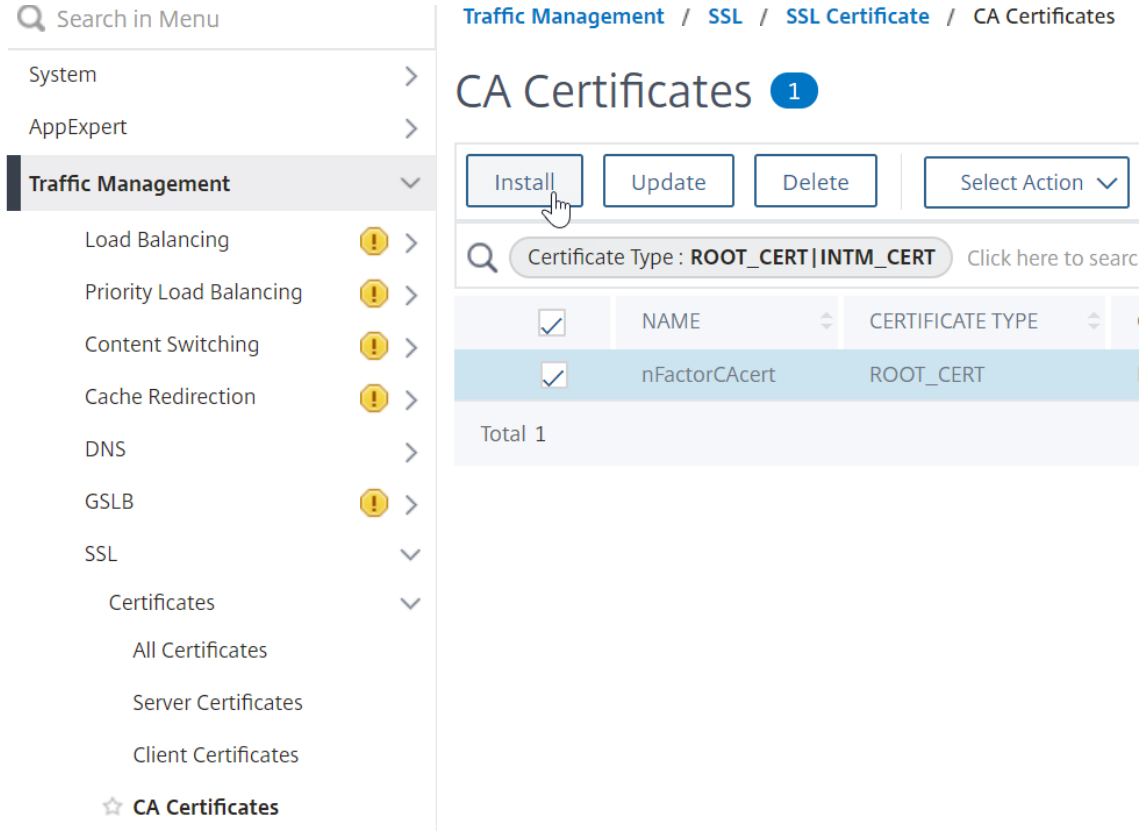
4. 選択を [認証] に変更します。[認証仮想サーバ名] ドロップダウンメニューから、認証、承認、および監査仮想サーバを選択し、[バインドとプレビュー] をクリックしてプレビューウィンドウを閉じます。

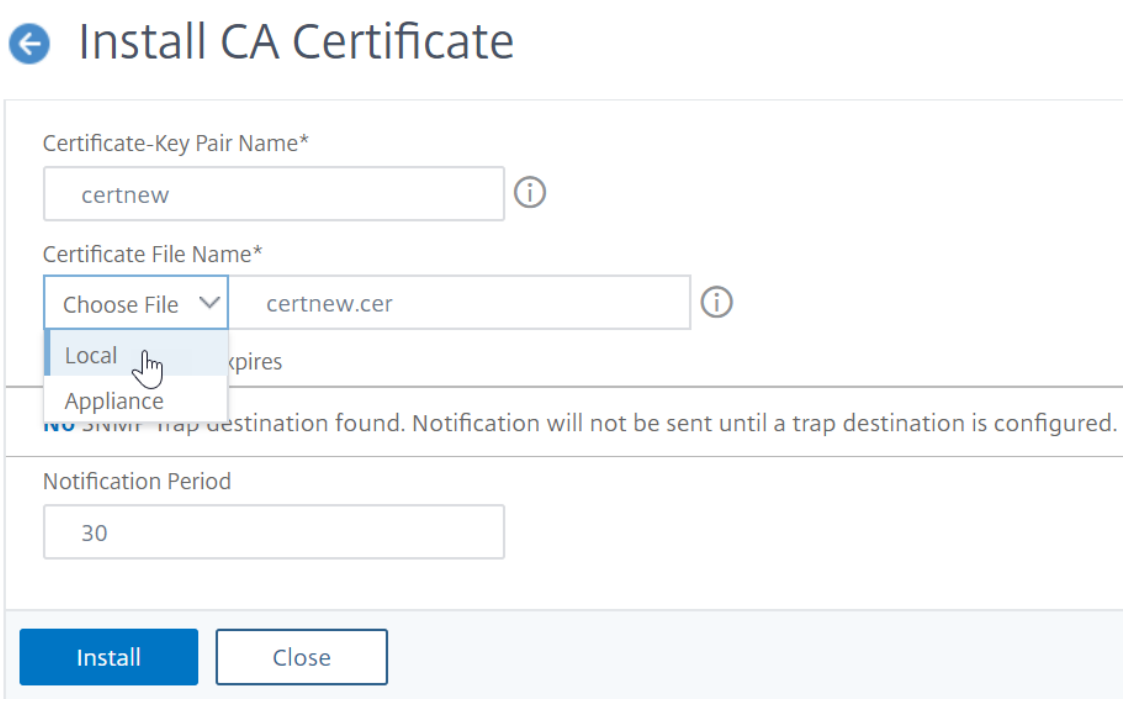


クライアント証明書認証を有効にする

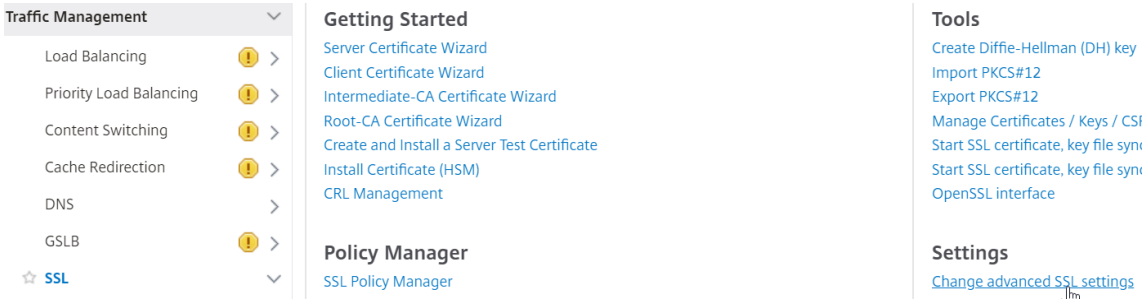
認証要素の 1 つがクライアント証明書である場合は、認証、承認、および監査仮想サーバーで SSL 構成を実行する必要があります。

1. [トラフィック管理] **> [SSL] > [**証明書] **> [CA** 証明書] に移動し、クライアント証明書の発行者のルート証明書をインストールします。ルート証明書にはキーファイルがありません。





2. [トラフィック管理] **> [SSL] > [**高度な **SSL** 設定の変更] に移動します。

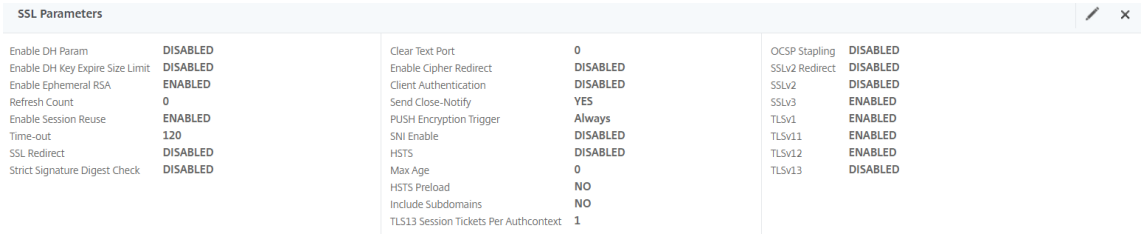


   a) 下にスクロールして、[デフォルトプロファイル] が [有効**]** になっているかどうかを確認します。「はい」の場合は、SSL プロファイルを使用してクライアント証明書認証を有効にする必要があります。それ以外の場合は、[SSL パラメーター] セクションの認証、承認、および監査仮想サーバーでクライアント証明書認証を直接有効にできます。

3. デフォルトの SSL プロファイルが有効になっていない場合は、次の手順を実行します。

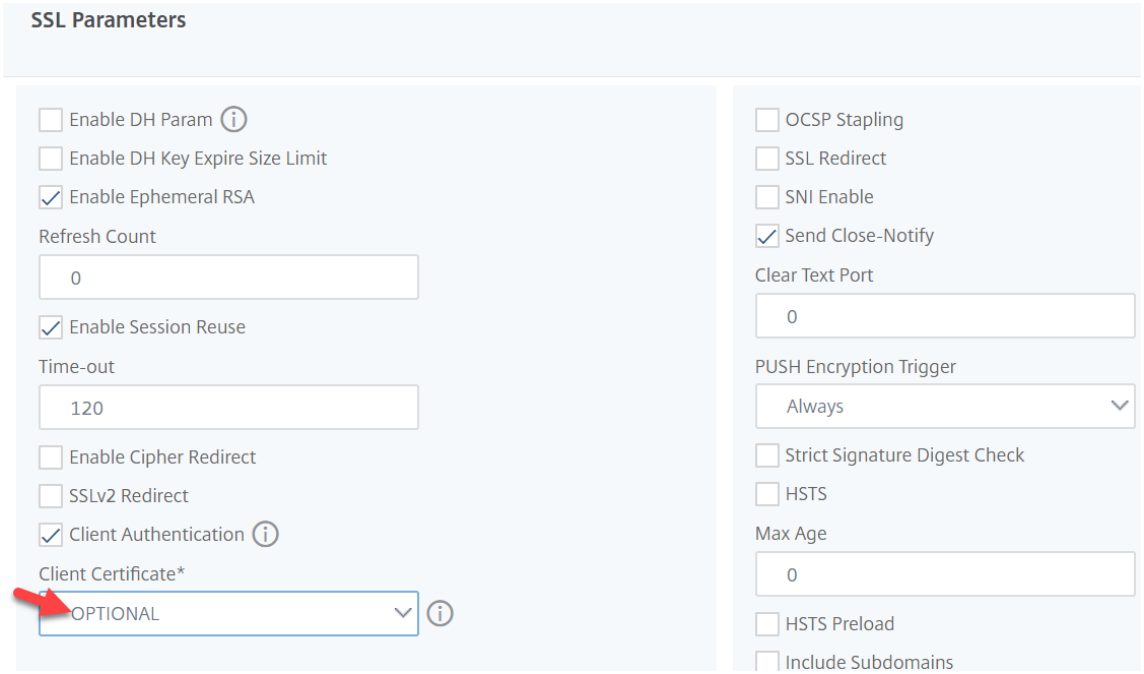
   a) [セキュリティ**] > [AAA-**アプリケーション**] > [**仮想サーバ] に移動し、既存の認証、認可、および監査仮想サーバを編集します。

   b)「**SSL** パラメータ」セクションで、鉛筆アイコンをクリックします。



a) クライアント認証オプションを有効にします。

b)「クライアント証明書」を「オプション」に設定し、「**OK**」をクリックします。



4. デフォルト SSL プロファイルが有効な場合は、クライアント認証を有効にして SSL プロファイルを作成します。

   a) 左側のメニューで、[システム] を展開し、[プロファイル] をクリックします。

   b) 右上の [SSL プロファイル] タブに切り替えます。

   c) ns_default_ssl_profile_frontend プロファイルを右クリックし、[追加] をクリックします。これにより、デフォルトプロファイルから設定がコピーされます。

   d) プロファイルに名前を付けます。このプロファイルの目的は、クライアント証明書を有効にすることです。

   e) 下にスクロールして、[クライアント認証] チェックボックスを見つけます。ボックスにチェックを入れます。

   f) [クライアント証明書] ドロップダウンメニューを [オプション] に変更します。

   g) デフォルトの SSL プロファイルをコピーしても、SSL 暗号はコピーされません。やり直す必要があります。

h) SSL プロファイルの作成が完了したら、[完了] をクリックします。

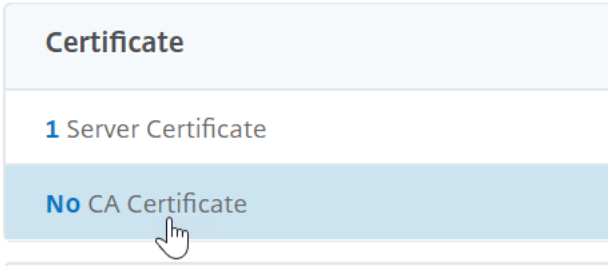
i) [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] に移動し、認証、認可、および監査仮想サーバを編集します。

j) [SSL プロファイル] セクションまでスクロールし、鉛筆をクリックします。

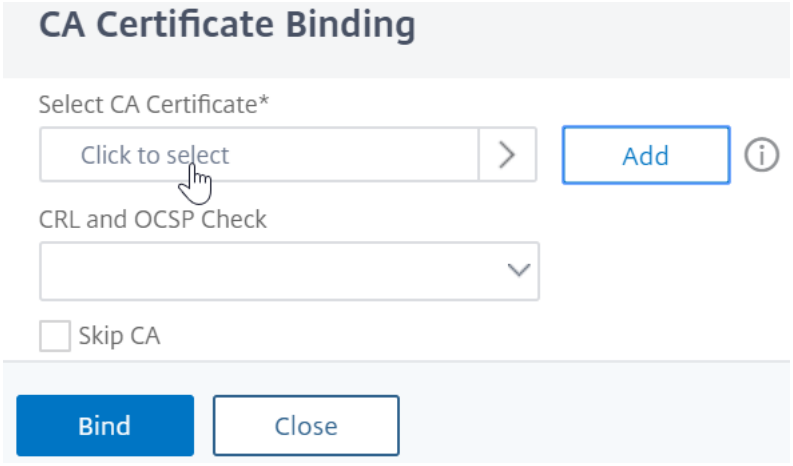
k) [SSL プロファイル] ドロップダウンメニューを、クライアント証明書が有効になっているプロファイルに変更します。[OK] をクリックします。

l) CA 証明書をバインドする手順が表示されるまで、この記事を下にスクロールします。

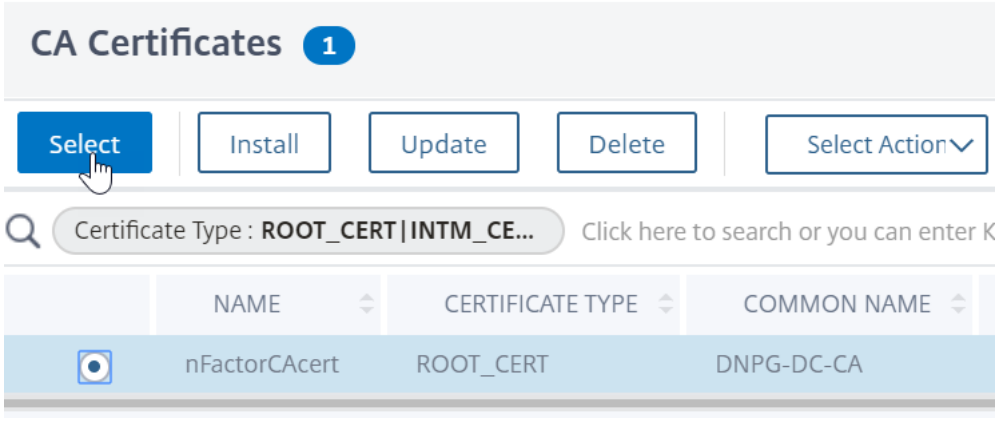
5. 左側の [証明書] セクションで、[**CA** 証明書なし] をクリックします。



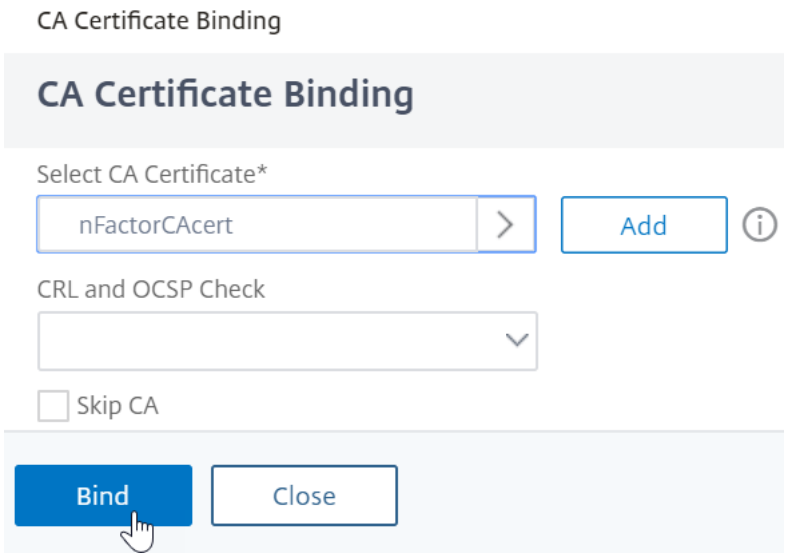
6. テキストをクリックし、クリックして選択します。



7. クライアント証明書の発行者のルート証明書の隣にあるオプションボタンをクリックし、[選択（**Select**）] をクリックします。



8. [**Bind**] をクリックします。



ログインスキーマ **XML** ファイル

ログインスキーマは、フォームベース認証ログオンページの構造を提供する XML ファイルです。

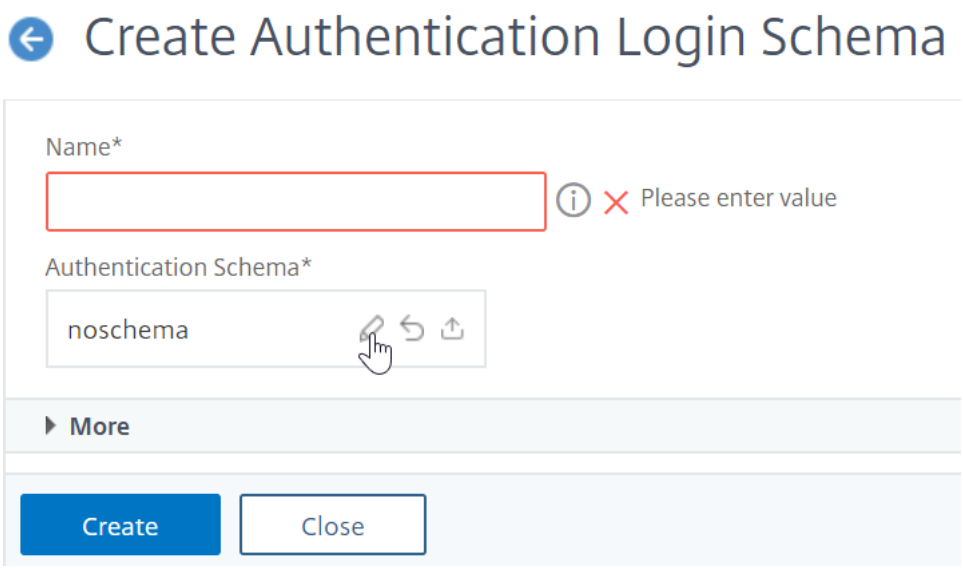
nFactor は、複数の認証要素が連鎖していることを意味します。各ファクタは、異なるログインスキーマページ/ファイルを持つことができます。一部の認証シナリオでは、ユーザーに複数のログオン画面を表示することができます。

ログインスキーマプロファイルの設定

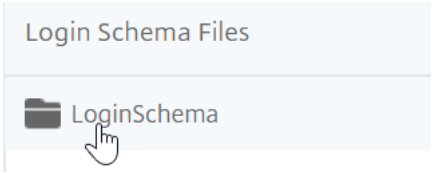
ログインスキーマプロファイルを設定するには、次の手順を実行します。

1. nFactor の設計に基づいて、ログインスキーマの.XML ファイルを作成または編集します。
2. [セキュリティ] **>** [**AAA-**アプリケーショントラフィック] **>** [ログインスキーマ] に移動します。
3. 右側の [プロファイル] タブに切り替えて、[追加] をクリックします。

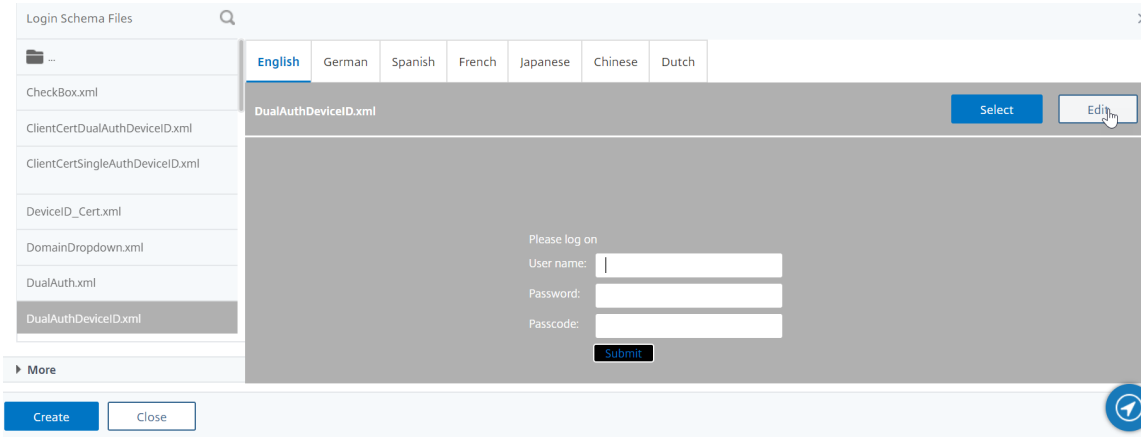
4. [認証スキーマ] フィールドで、鉛筆アイコンをクリックします。



5. **LoginSchema** フォルダをクリックすると、その中のファイルが表示されます。



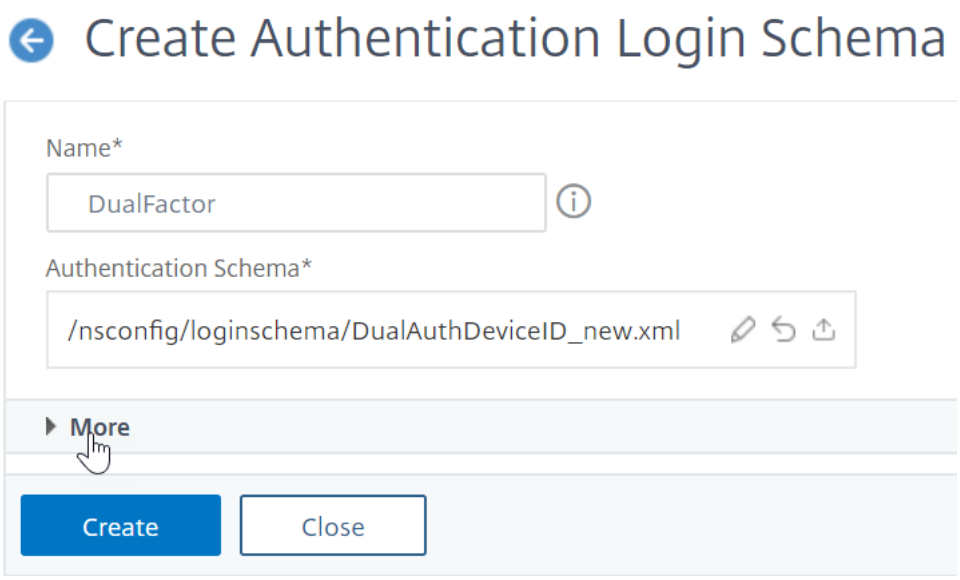
6. ファイルの 1 つを選択します。右側にプレビューが表示されます。ラベルは、右上の [編集] ボタンをクリックして変更できます。



7. 変更を保存すると、/nsConfig/loginSchema の下に新しいファイルが作成されます。

8. 右上の [選択] をクリックします。

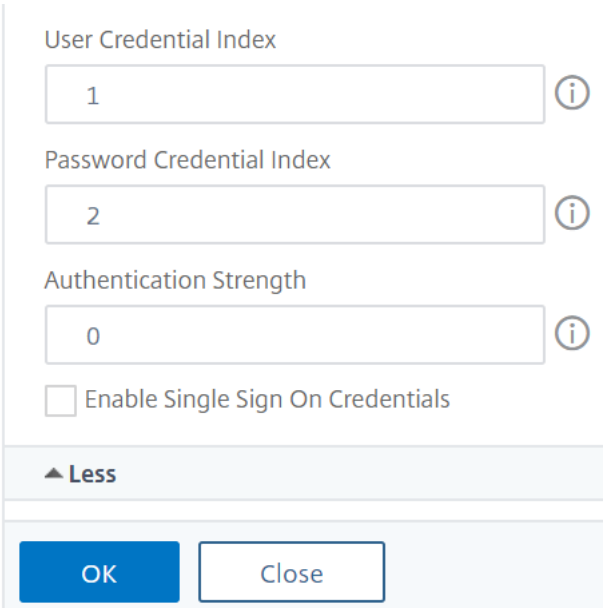
9. ログインスキーマに名前を付けて、[**More**] をクリックします。



10. StoreFront などのバックエンドサービスへのシングルサインオン（SSO）のログインスキーマに入力したユーザー名とパスワードを使用します。

    ログインスキーマに入力された資格情報をシングルサインオンの認証情報として使用するには、次のいずれかの方法を使用します。

    - [認証ログインスキーマの作成] ページの下部にある [詳細] をクリックし、[シングルサインオン資格情報の有効化] を選択します。
    - [認証ログインスキーマの作成] ページの下部にある [詳細] をクリックし、ユーザクレデンシャルインデックスとパスワードクレデンシャルインデックスに一意の値を入力します。これらの値は 1 ～16 の範囲です。後で AAA.USER.ATTRIBUTE (#) 式を使用して、トラフィックポリシー/プロファイルでこれらのインデックス値を参照します。



11. [**Create**] をクリックして、ログインスキーマプロファイルを作成します。

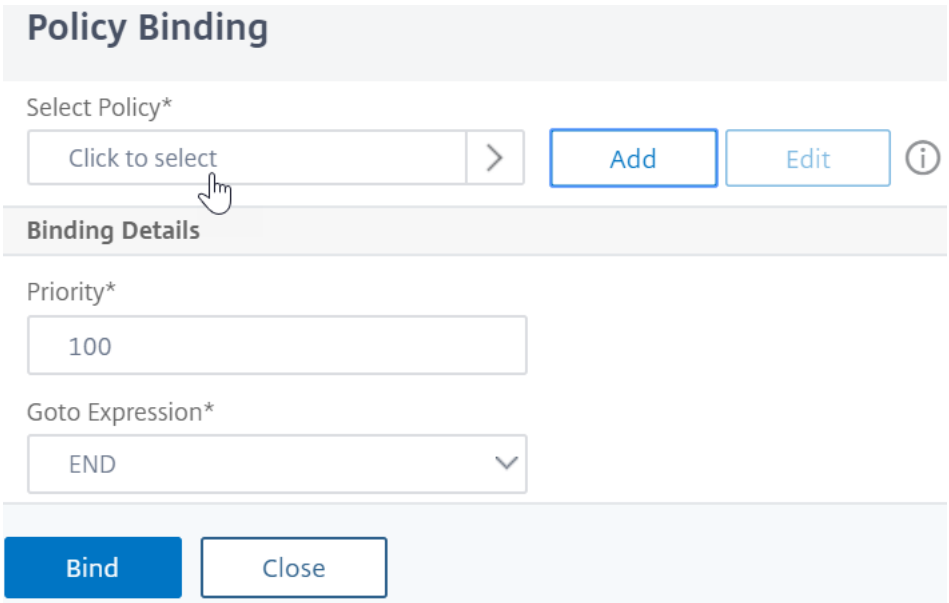
> 注**:** ログインスキーマファイル (.xml) を後で編集する場合、変更を反映させるには、ログインスキーマプロファイルを編集して、ログインスキーマ (.xml) ファイルを再度選択する必要があります。

ログインスキーマポリシーの作成とバインド

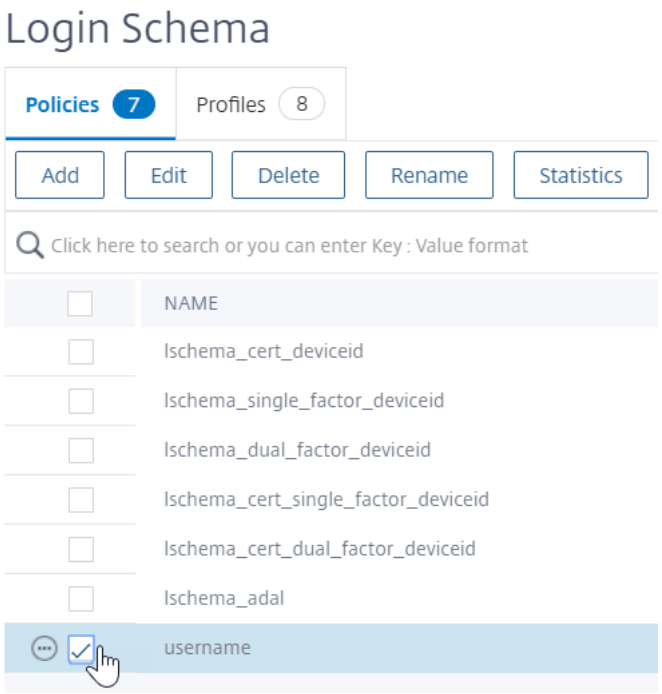
ログインスキーマプロファイルを認証、承認、および監査仮想サーバーにバインドするには、最初にログインスキーマポリシーを作成する必要があります。後で詳述するように、ログインスキーマプロファイルを認証ポリシーラベルにバインドする場合は、ログインスキーマポリシーは必要ありません。

ログインスキーマポリシーを作成してバインドするには、次の手順を実行します。

1. [セキュリティ] **>** [**AAA-**アプリケーショントラフィック] **>** [ログインスキーマ] に移動します。
2. [**Policies**] タブで [**Add**] をクリックします。
3. [**Profile**] ドロップダウンメニューを使用して、すでに作成したログインスキーマプロファイルを選択します。
4. [**Rule**] ボックスに高度なポリシー式を入力し、[**Create**] をクリックします。
5. [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] に移動し、[編集] をクリックして既存の認証、承認、および監査仮想サーバを編集します。
6. [詳細設定] 列で、[ログインスキーマ] をクリックします。
7. 「ログインスキーマ」セクションで、「ログインスキーマなし」をクリックします。
8. 「ポリシーの選択」ドロップダウンリストから認証ポリシーを選択します。



9. 「認証ポリシー」ページで、認証ポリシーの横にあるラジオボタンをクリックし、「選択」をクリックします。このリストには、ログインスキーマポリシーのみが表示されます。ログインスキーマプロファイル (ポリシーなし) は表示されません。



10. [**Bind**] をクリックします。

高度な認証ポリシー

認証ポリシーは、ポリシー式とポリシーアクションの組み合わせです。式が true の場合は、認証アクションを評価します。

高度な認証ポリシーの作成

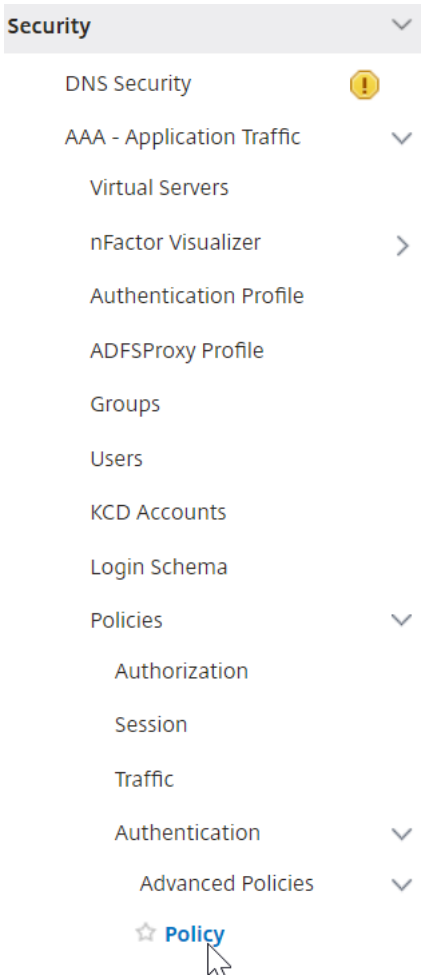
認証ポリシーは、ポリシー式とポリシーアクションの組み合わせです。式が true の場合は、認証アクションを評価します。

認証アクション/サーバー (LDAP、RADIUS、CERT、SAML など) が必要です。
高度な認証ポリシーを作成する場合、認証アクション/サーバーを作成するためのプラス (Add) アイコンが表示されます。

または、高度な認証ポリシーを作成する前に認証アクション (サーバー) を作成することもできます。認証サーバーは、[認証] **>** [ダッシュボード] の下にあります。右側の [追加] をクリックし、[サーバーの種類] を選択します。これらの認証サーバの作成手順については、ここでは詳しく説明しません。認証—NetScaler 12/NetScaler 12.1 の手順を参照してください。

高度な認証ポリシーを作成するには、次の手順を実行します。

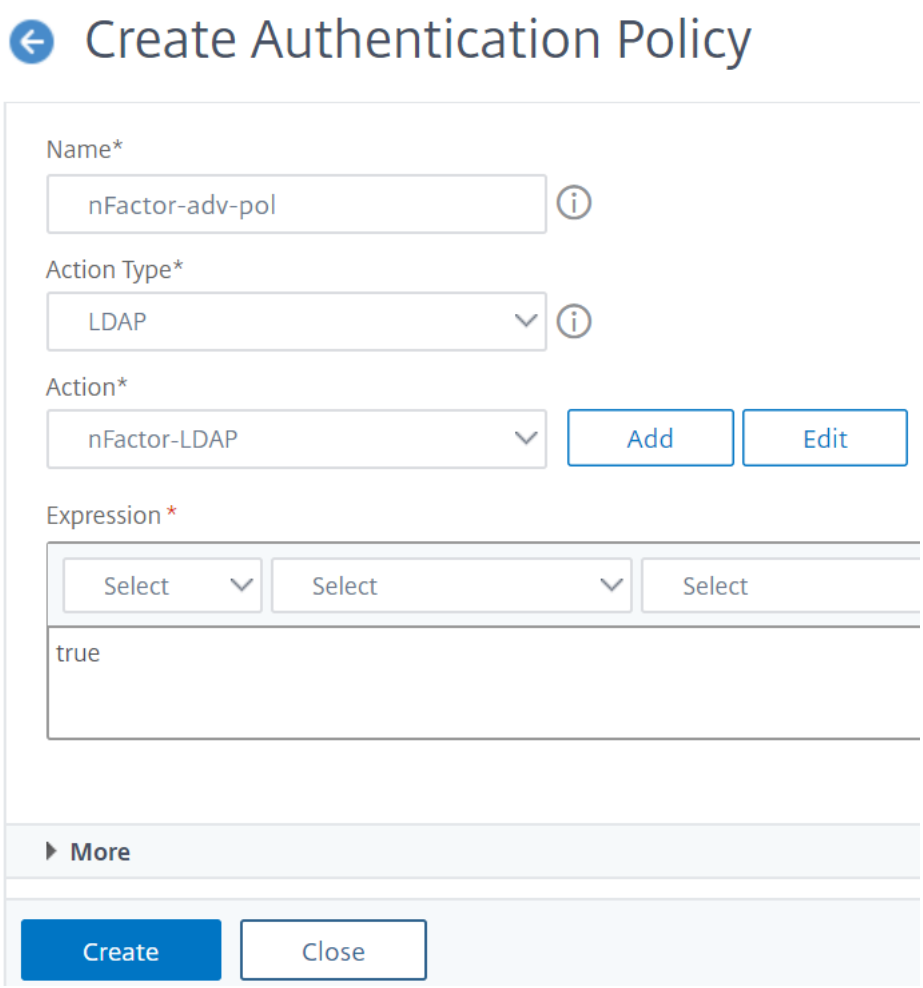
1. セキュリティ **>** **AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 高度なポリシー **>** ポリシーへのナビゲート



2. 詳細ウィンドウで、次のいずれかの操作を行います。

   - ポリシーを作成するには、[**Add**] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[編集] をクリックします。

3. [認証ポリシーの作成] または [認証ポリシーの構成] ダイアログボックスで、パラメータの値を入力または選択します。



- 名前 -ポリシー名。以前に設定したポリシーでは変更できません。
- アクションタイプ -ポリシータイプ：証明書、ネゴシエート、LDAP、RADIUS、SAML、SAMLIDP、TACACS、または WEBAUTH。
- アクション -ポリシーに関連付ける認証アクション (プロファイル)。既存の認証アクションを選択するか、プラス記号をクリックして適切なタイプのアクションを作成できます。
- ログアクション -ポリシーに関連付ける監査アクション。既存の監査アクションを選択するか、プラス記号をクリックしてアクションを作成できます。
  アクションが設定されていません。アクションを作成するには、[**Add**] をクリックして手順を完了します。
- 式 -指定したアクションを適用する接続を選択するルール。ルールは、シンプル (「true」はすべてのトラフィックを選択) または複雑にすることができます。式を入力するには、まず [式] ウィンドウの下にある左端のドロップダウンリストで式の種類を選択し、次に式テキスト領域に式を直接入力するか、[追加] をクリックして [式の追加] ダイアログボックスを開き、その中のドロップダウンリストを使用して式。)
- [**Comment**]：この認証ポリシーが適用されるトラフィックのタイプを説明するコメントを入力できます。オプションです。

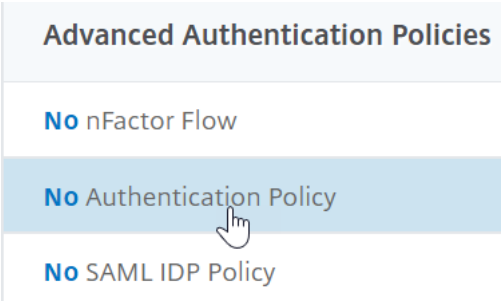
4. [**Create**] をクリックしてから、[**Close**] をクリックします。ポリシーを作成した場合、そのポリシーは [認証ポリシーおよびサーバ] ページに表示されます。

nFactor の設計に基づいて、必要に応じて追加の高度な認証ポリシーを作成します。

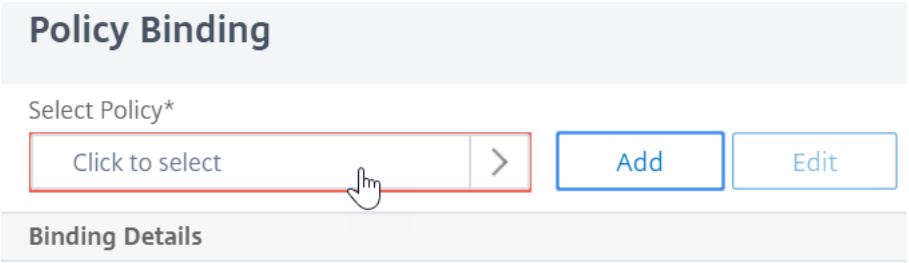
第 **1** 要素の高度な認証ポリシーを認証、承認、および監査にバインドする

最初の要素である認証、承認、および監査仮想サーバーに対して、高度な認証ポリシーを直接バインドできます。次の要素については、高度な認証ポリシーを認証ポリシーラベルにバインドする必要があります。

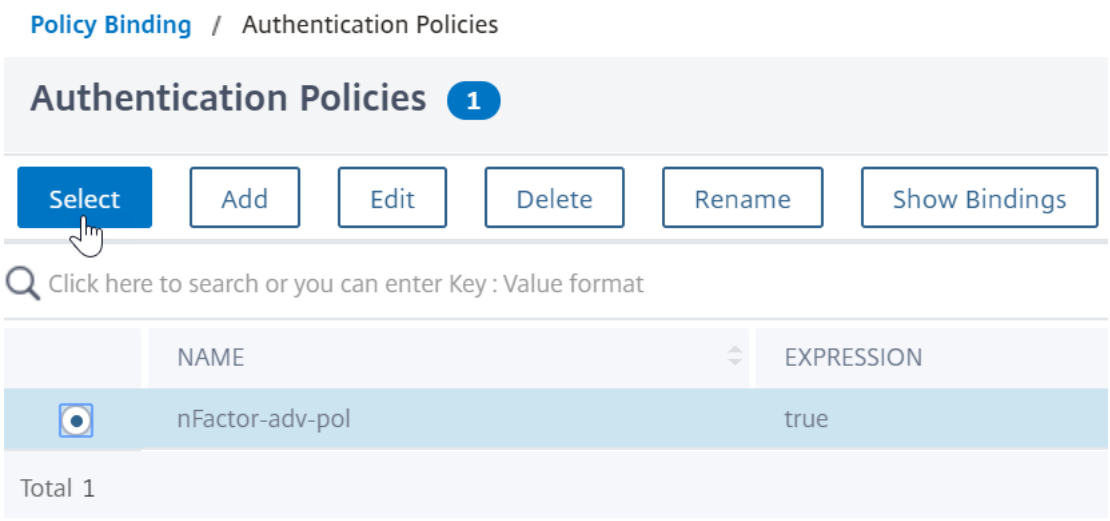
1. [セキュリティ] **>** [**AAA-**アプリケーショントラフィック] **>** [仮想サーバ] に移動します。既存の仮想サーバーを編集します。
2. 左側の [高度な認証ポリシー] セクションで、[認証ポリシーなし] をクリックします。



3. **[**ポリシーの選択] で、[クリックして選択**]** というテキストをクリックします。

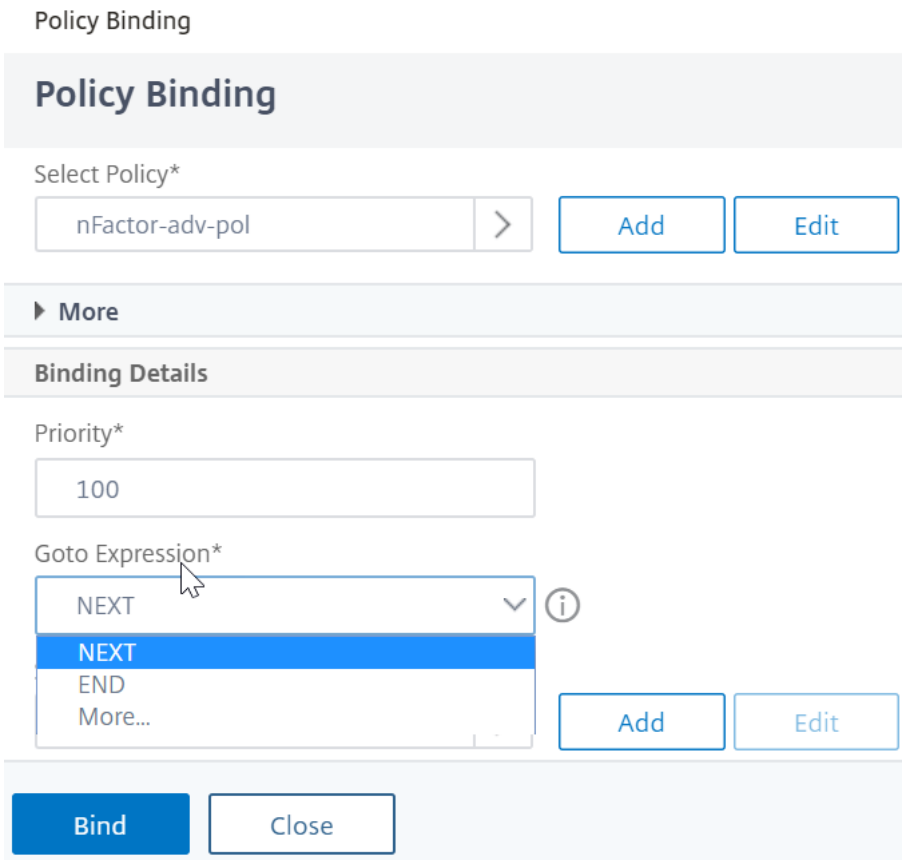


4. [高度な認証ポリシー**]** の横にあるオプションボタンをクリックし、[選択] をクリックします。

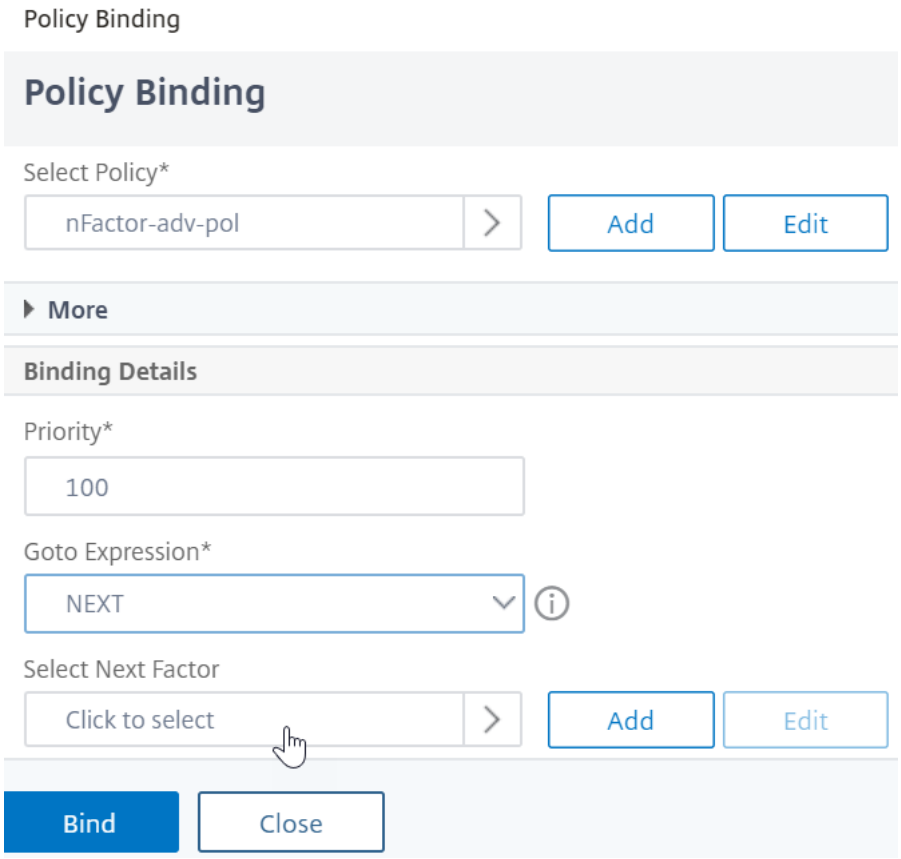


5. [バインドの詳細] セクションの [**Goto** 式] は、この高度な認証ポリシーが失敗した場合に次に何が起こるかを決定します。

   - **Goto Expression** が **NEXT** に設定されている場合、この認証、承認、および監査仮想サーバーにバインドされた次の高度な認証ポリシーが評価されます。
   - **Goto Expression** が **END** に設定されている場合、またはこの認証、承認、および監査仮想サーバーにバインドされた高度な認証ポリシーがこれ以上存在しない場合、認証は完了し、失敗としてマークされます。



6. **[次の要素を選択]** で、認証ポリシーラベルをポイントできるかを選択できます。次の要素は、高度な認証ポリシーが成功した場合にのみ評価されます。最後に、[バインド] をクリックします。



抽出された **LDAP** グループを使用して、次の認証要素を選択する

抽出された LDAP グループを使用して、実際に LDAP による認証を行わずに、次の認証要素を選択できます。

1. LDAP サーバーまたは LDAP アクションを作成または編集するときは、[認証] チェックボックスをオフにします。
2. [その他の設定] で、[グループ属性] と **[**** サブ属性名 **] で適切な値を選択します。

ポリシーラベルを認証する

高度な認証ポリシーを認証、承認、および監査仮想サーバーにバインドし、次の要素を選択すると、高度な認証ポリシーが成功した場合にのみ次の要素が評価されます。次に評価される要素は、認証ポリシーラベルです。

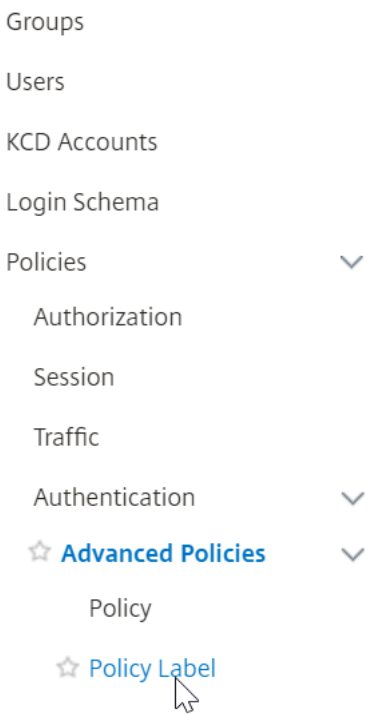
認証ポリシーラベルは、特定の要素に対する認証ポリシーのコレクションを指定します。各ポリシーラベルは 1 つの要素に対応します。また、ユーザーに提示する必要があるログインフォームも指定します。認証ポリシーラベルは、認証ポリシーまたは別の認証ポリシーラベルの次の要素としてバインドする必要があります。

注: すべての要素にログインスキーマは必要ありません。ログインスキーマプロファイルは、ログインスキーマを認証ポリシーラベルにバインドする場合にのみ必要です。

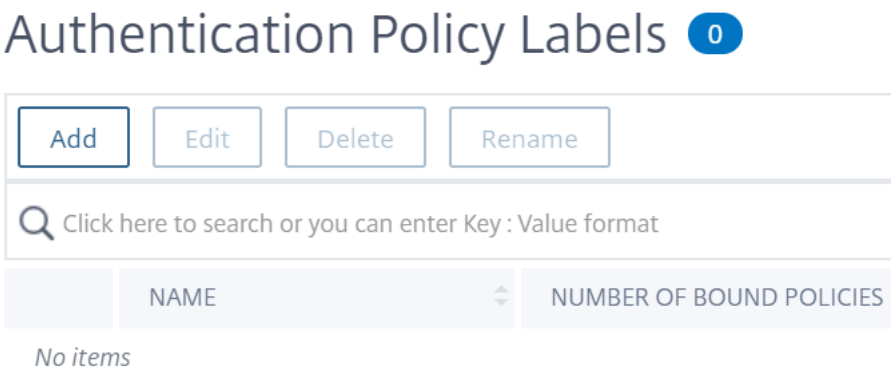
認証ポリシーラベルを作成する

ポリシーラベルは、特定のファクタの認証ポリシーを指定します。各ポリシーラベルは 1 つの要素に対応します。ポリシーラベルは、ユーザーに提示する必要があるログインフォームを指定します。ポリシーラベルは、認証ポリシーまたは別の認証ポリシーラベルの次の要素としてバインドする必要があります。通常、ポリシーラベルには、特定の認証メカニズムの認証ポリシーが含まれます。ただし、異なる認証メカニズムの認証ポリシーを持つポリシーラベルを使用することもできます。

1. [セキュリティ] **>** **[AAA-**アプリケーショントラフィック] **>** **[**ポリシー] **>** **[**認証] **>** **[**詳細ポリシー] **>** **[**ポリシーラベル] に移動します。



2. [追加] をクリックします。

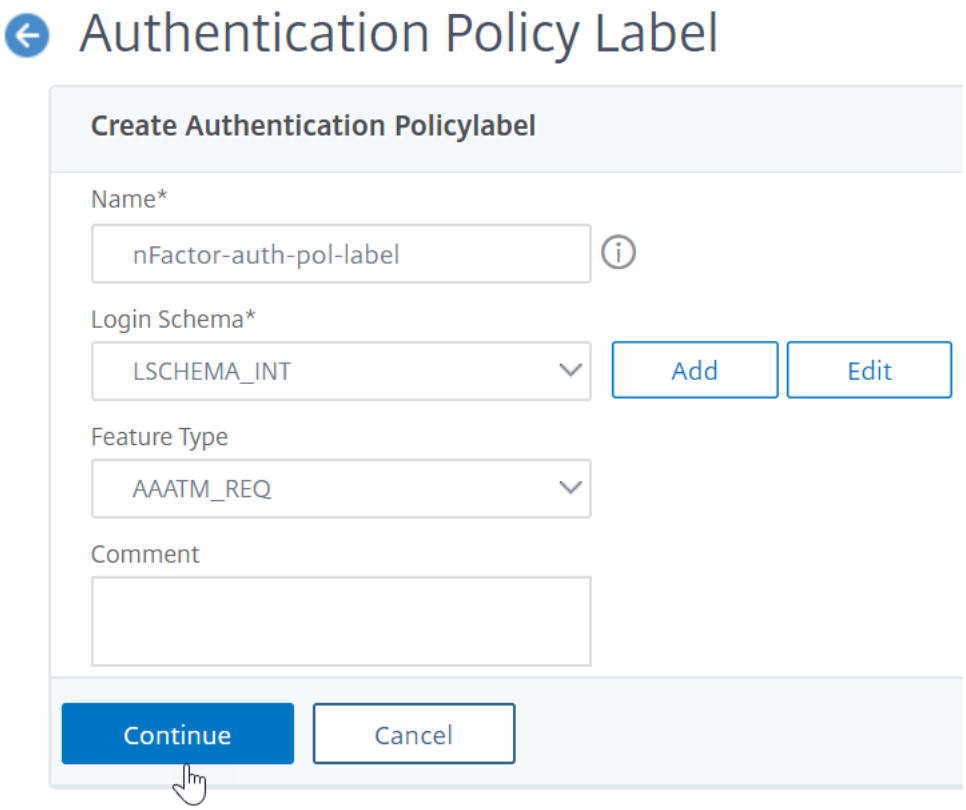


3. 次のフィールドに入力して、認証ポリシーラベルを作成します。

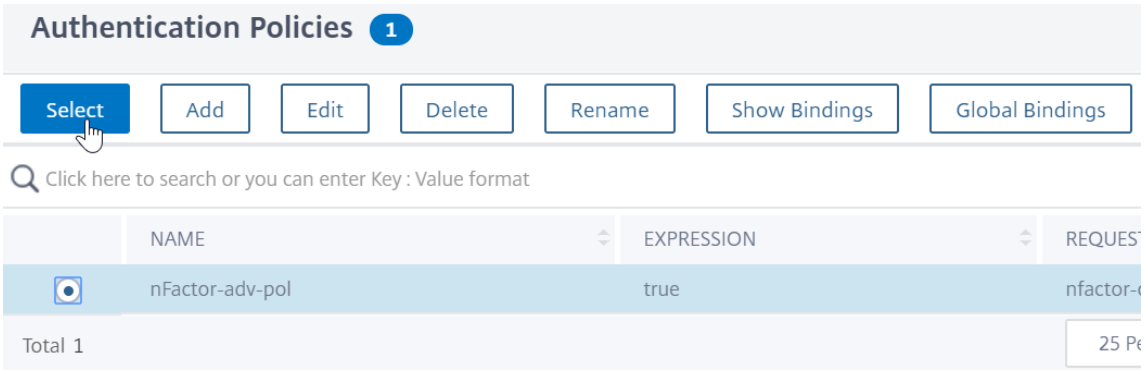
   a) 新しい認証ポリシーラベルの名前を入力します。

b) 認証ポリシーラベルに関連付けられたログインスキーマを選択します。ユーザーに対して何も表示しない場合は、スキーマなし (LSCHEMA_INT) に設定されたログインスキーマプロファイルを選択できます。

c) [続行] をクリックします。



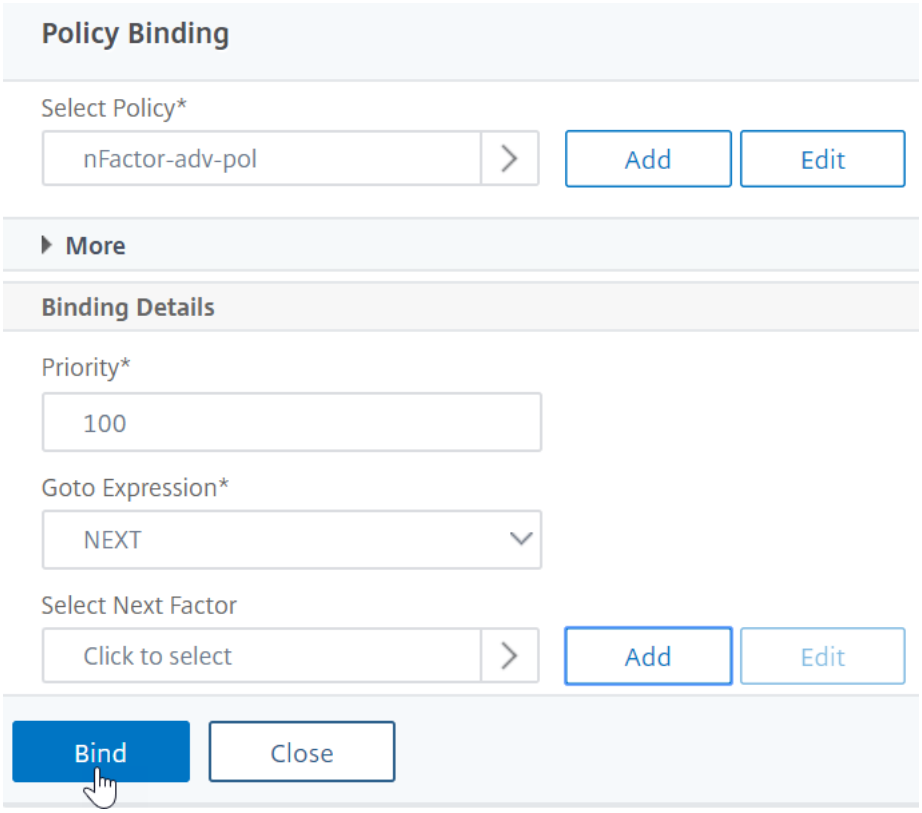
4. [ポリシーバインド] セクションで、[クリックして選択] と表示されている場所をクリックします。
5. この要素を評価する認証ポリシーを選択します。



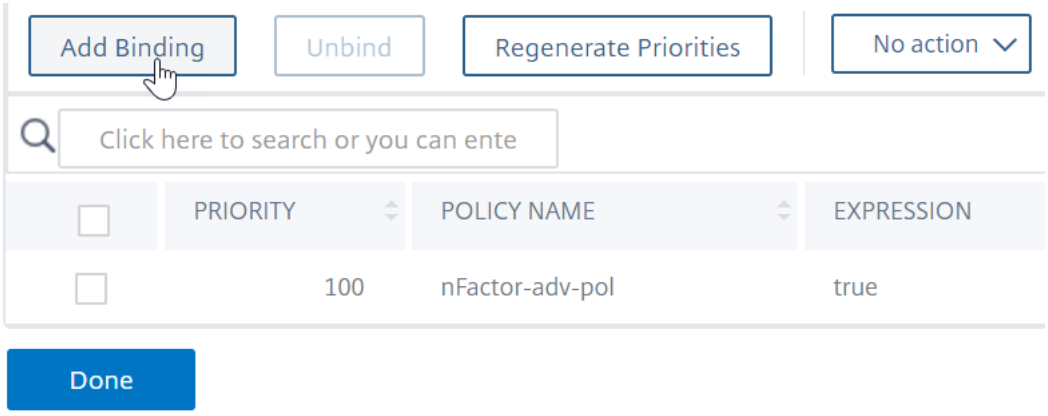
6. 次のフィールドに入力します。

a) ポリシーバインディングの優先度を入力します。

b) [式に移行] で、より高度な認証ポリシーをこの要素にバインドする場合は [次へ] を選択するか、[終了] を選択します。



7. 別の要素を追加する場合は、**[**次の要素の選択] で、次の認証ポリシーラベル (次の要素) をクリックして選択し、バインドします。
   次の要素を選択せず、この高度な認証ポリシーが成功すると、認証は成功し、完了します。
8. [**Bind**] をクリックします。
9. [**Add Binding**] をクリックすると、このポリシーラベル (ファクタ) に高度な認証ポリシーを追加できます。完了したら **[**完了] をクリックします。



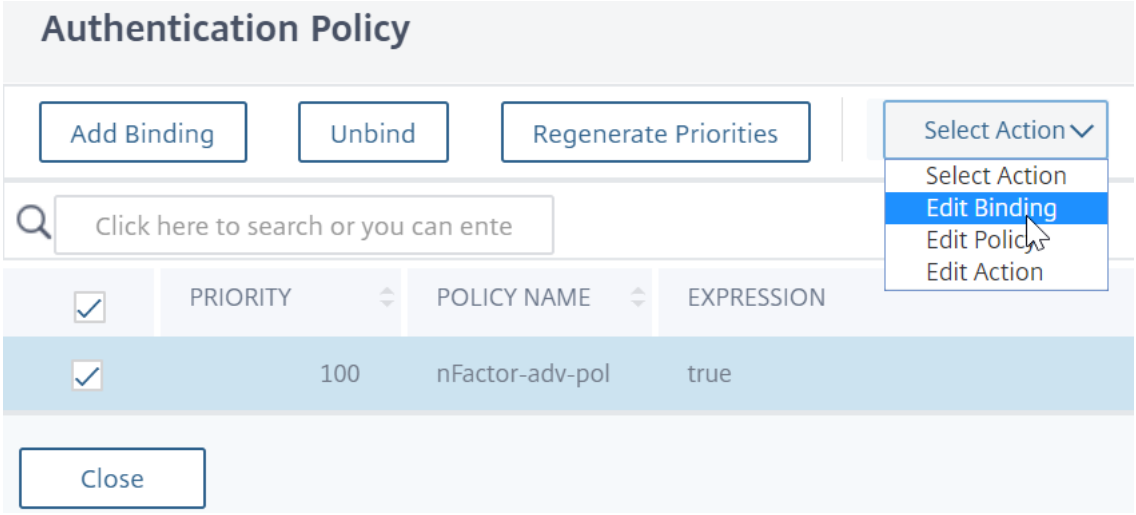
認証ポリシーラベルのバインド

ポリシーラベルを作成したら、既存の高度な認証ポリシーバインディングにポリシーラベルをバインドして、要素を連鎖させます。

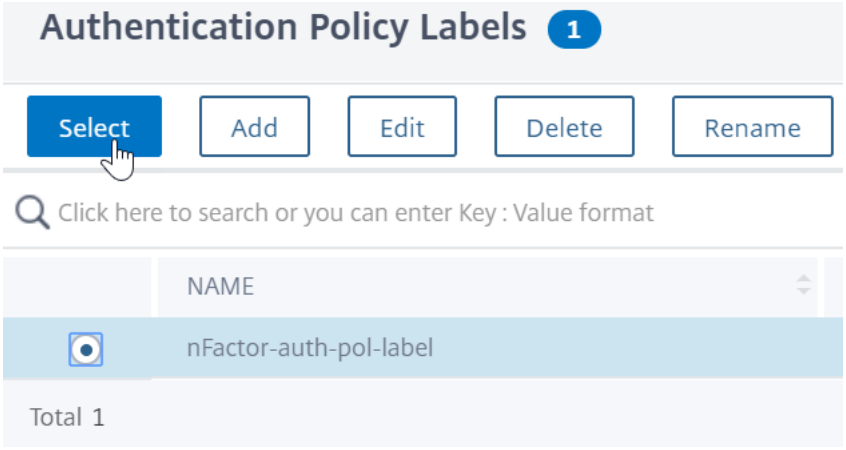
高度な認証ポリシーがバインドされている既存の認証、承認、および監査仮想サーバーを編集する場合、または別のポリシーラベルを編集して次の要素を含める場合は、次の要素を選択できます。

高度な認証ポリシーが既にバインドされている既存の認証、承認、および監査仮想サーバーを編集するには

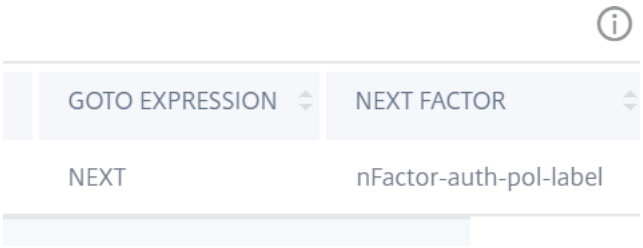
1. [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] に移動します仮想サーバを選択し、[編集（Edit）] をクリックします。
2. 左側の [高度な認証ポリシー] セクションで、既存の認証ポリシーバインドをクリックします。
3. 「アクションを選択」で、「バインドの編集」をクリックします。



4. 「次の要素を選択」をクリックし、既存の認証ポリシーラベル (次の要素) を選択します。

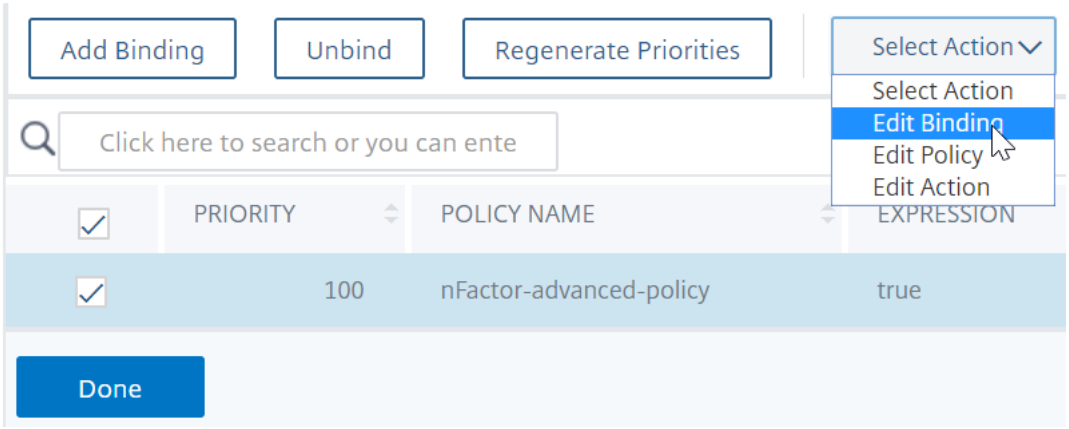


5. ポリシーバインディングページの「バインド」をクリックします。右端に「**NEXT** FACTOR」列が表示され、次のファクターの詳細が表示されます。

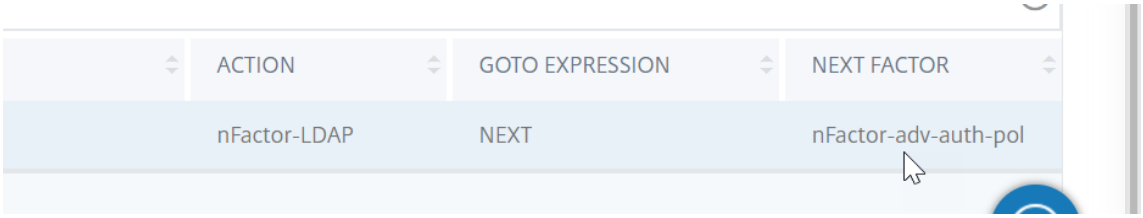


ポリシーラベルネクストファクタを別のポリシーラベルに追加するには

1. セキュリティ **>** **AAA** —アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 高度ポリシー **>** ポリシーラベルにナビゲートして下さい。
2. 「認証ポリシーラベル」ページで、認証ポリシーラベルを選択し、「編集」をクリックします。
3. 「認証ポリシー」ページで認証ポリシーラベルを選択し、「アクションの選択」で「バインドの編集」をクリックします。



4. 「バインドの詳細」**>** 「次のファクタを選択」で、次のファクタをクリックして選択します。
5. 次のファクタのポリシーラベルを選択し、[**Select**] ボタンをクリックします。
6. ポリシーバインディングページの「バインド」をクリックします。「認証ポリシーラベル」ページには、「次の要素」列があり、右端に次の要素の詳細が表示されます。

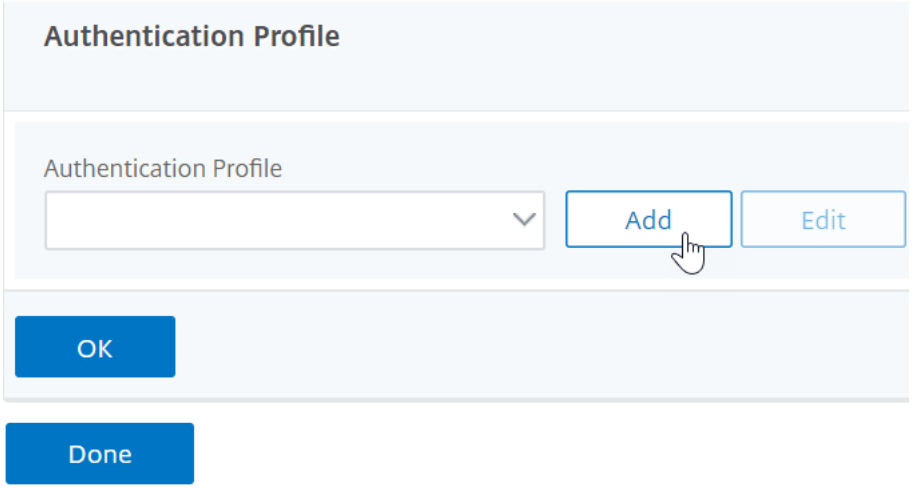


NetScaler Gateway 用 n ファクター

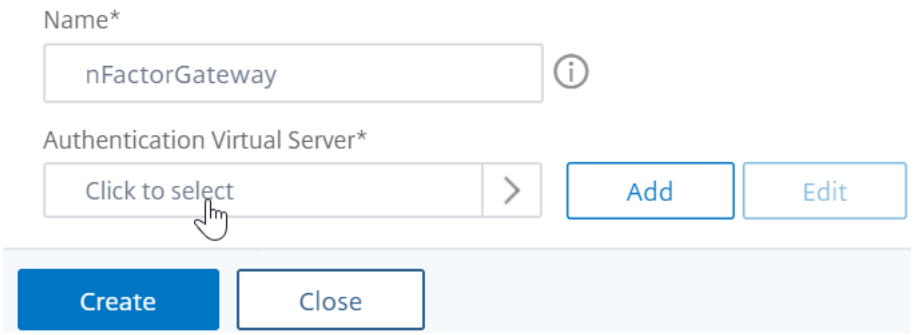
NetScaler Gateway で nFactor を有効にするには、認証プロファイルを認証、承認、および監査仮想サーバーにリンクする必要があります。

認証、承認、監査の仮想サーバーを **NetScaler Gateway** 仮想サーバーにリンクする認証プロファイルの作成

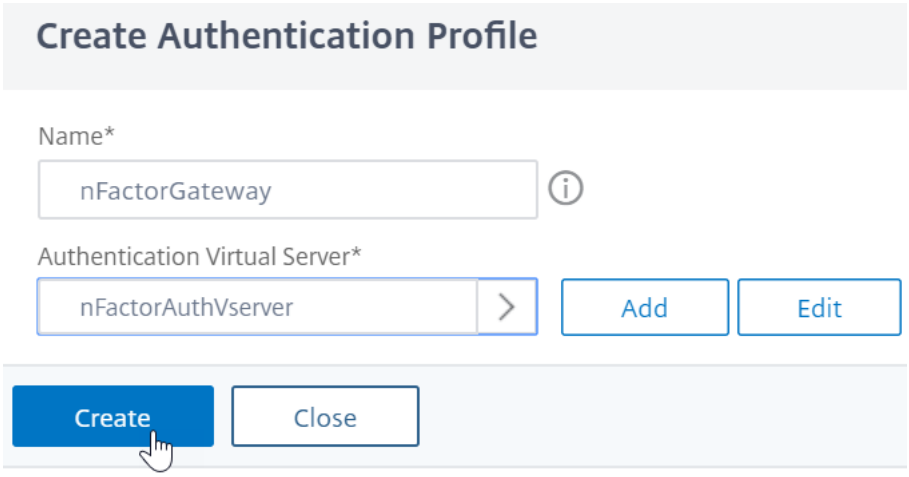
1. [**NetScaler Gateway] > [**仮想サーバー] に移動し、編集する既存のゲートウェイ仮想サーバーを選択します。
2. [詳細設定] で、[認証プロファイル] をクリックします。
3. [認証プロファイル] の [追加] をクリックします



4. 認証プロファイルの名前を入力し、[クリックして選択**]** と表示されている場所をクリックします。



5. [認証仮想サーバー] で、ログインスキーマ、高度な認証ポリシー、および認証ポリシーのラベルが構成されている既存のサーバーを選択します。認証仮想サーバーを作成することもできます。認証、承認、および監査仮想サーバーには IP アドレスは必要ありません。[**Select**] をクリックします。
6. [作成] をクリックします。



7. [**OK**] をクリックして、[認証プロファイル] セクションを閉じます。

注**:** ファクタの 1 つをクライアント証明書として設定した場合は、SSL パラメータと CA 証明書を設定する必要があります。

認証プロファイルの認証、承認、および監査仮想サーバーへのリンクが完了し、NetScaler Gateway を参照すると、nFactor 認証画面が表示されます。

SSL パラメータと **CA** 証明書の設定

認証要素の 1 つが証明書である場合は、NetScaler Gateway 仮想サーバーで何らかの SSL 構成を実行する必要があります。

1. [トラフィック管理**] > [SSL] > [**証明書**] > [CA** 証明書] に移動し、クライアント証明書の発行者のルート証明書をインストールします。認証局の証明書にはキーファイルは必要ありません。

   デフォルトの SSL プロファイルが有効な場合は、クライアント認証が有効になっている SSL プロファイルが既に作成されています。

2. **NetScaler Gateway >** 仮想サーバーに移動し、nFactor が有効になっている既存の NetScaler Gateway 仮想サーバーを編集します。

   - デフォルトの SSL プロファイルが有効になっている場合は、[編集（Edit）] アイコンをクリックします。
   - [SSL プロファイル] リストで、[クライアント認証] が有効で [オプション] に設定されている SSL プロファイルを選択します。
   - デフォルトの SSL プロファイルが有効でない場合は、[編集（Edit）] アイコンをクリックします。
   - [クライアント認証] チェックボックスをオンにします。
   - [クライアント証明書] が [任意] に設定されていることを確認します。

3. [OK] をクリックします。

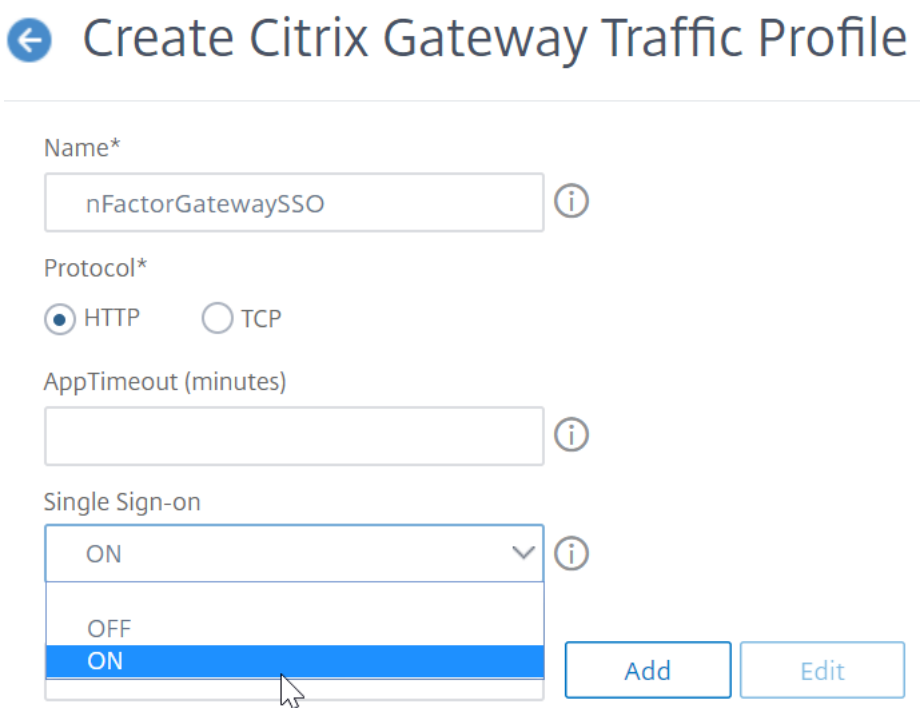
4. [証明書] セクションで、[**CA** 証明書なし] をクリックします。
5. [CA 証明書の選択] で、クライアント証明書の発行元のルート証明書をクリックして選択します。
6. [Bind] をクリックします。

注: クライアント証明書を発行した中間 CA 証明書もバインドする必要がある場合があります。

StoreFront への **nFactor** シングルサインオン用の **NetScaler Gateway** トラフィックポリシーを構成する

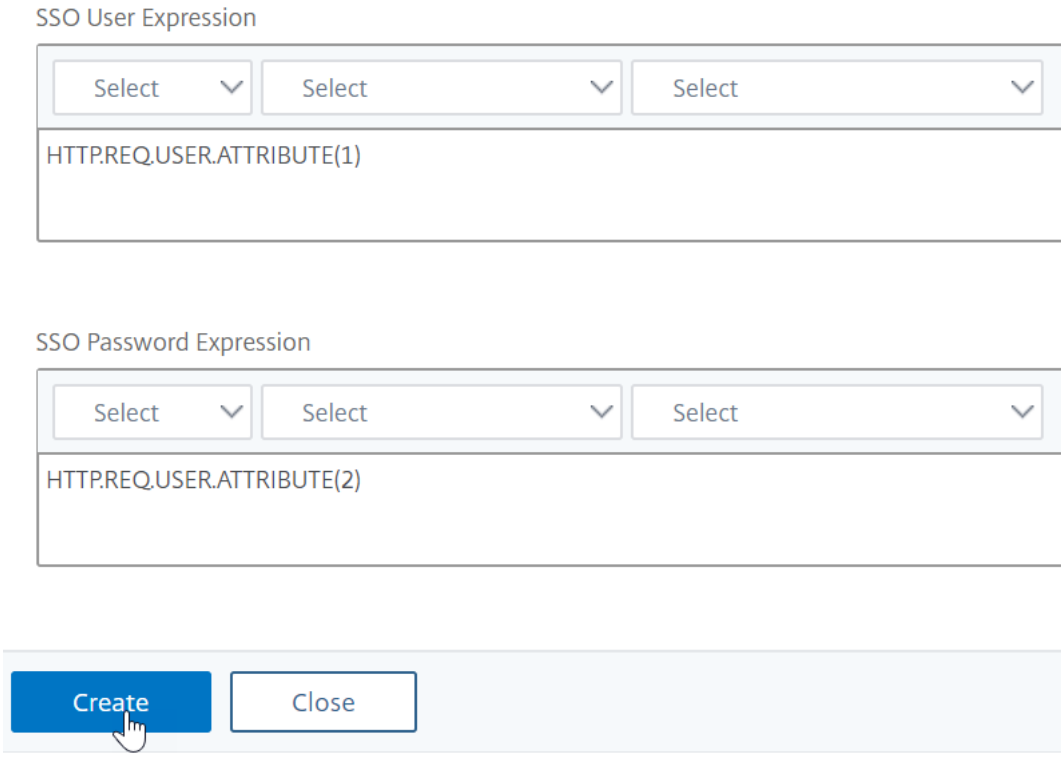
StoreFront へのシングルサインオンでは、nFactor はデフォルトで最後に入力したパスワードを使用します。LDAP が最後に入力されたパスワードでない場合は、トラフィックポリシー/プロファイルを作成して、デフォルトの nFactor 動作を上書きする必要があります。

1. **NetScaler Gateway >** ポリシー **>** トラフィックに移動します。
2. [トラフィックプロファイル] タブで、[追加] をクリックします。
3. トラフィックプロファイルの名前を入力します。**HTTP** プロトコルを選択します。
   [シングルサインオン] で、[オン] を選択します。

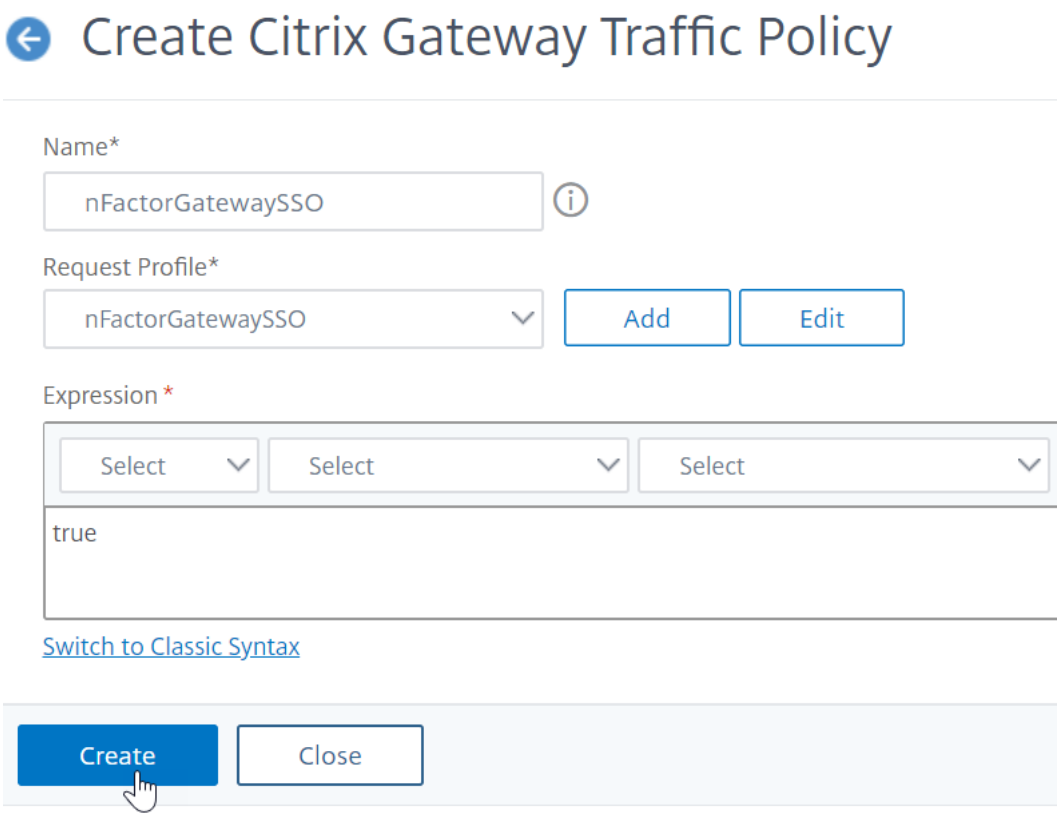


4. [**SSO** 式] で、ログインスキーマで指定されたインデックスと一致する AAA.USER.ATTRIBUTE (#) 式を入力し、[作成] をクリックします。

> 注
> AAA.USER 式は、非推奨の HTTP.REQ.USER 式を置き換えるように実装されるようになりました。



5. [トラフィックポリシー] タブをクリックし、[追加] をクリックします。
6. ポリシーの名前を入力します。前の手順で作成したトラフィックプロファイルを選択します。「エクスプレッション」に詳細なエクスプレッションを入力し、「作成」をクリックします。



7. **NetScaler Gateway> NetScaler Gateway** 仮想サーバーに移動します。

   - 既存の仮想サーバを選択し、[編集] をクリックします。
   - [ポリシー] セクションで、[**+**] 記号をクリックします。
   - **[**ポリシーの選択] で [トラフィック] を選択します
   - 「タイプの選択」で、「要求」を選択します。
   - 作成したトラフィックポリシーを選択し、[**Bind**] をクリックします。

CLI を使用した **nFactor** 設定のサンプルスニペット

nFactor 認証の段階的な設定を理解するために、最初の要素が LDAP 認証で、2 番目の要素が RADIUS 認証である 2 要素認証の展開について考えてみましょう。

このサンプル展開では、ユーザーは単一のログインフォームを使用して両方の要素にログインする必要があります。したがって、2 つのパスワードを受け入れる単一のログインフォームを定義します。最初のパスワードは LDAP 認証に使用され、もう 1 つは RADIUS 認証に使用されます。
実行される設定は次のとおりです。

1. 認証用の負荷分散仮想サーバーの構成

   add lb vserver lbvs89 HTTP 1.136.19.55 80 -AuthenticationHost auth56.aaatm.com -Authentication ON

2. 認証仮想サーバを設定します。

   add authentication vserver auth56 SSL 10.106.30.223 443 -AuthenticationDomain aaatm.com

3. ログインフォームのログインスキーマを設定し、ログインスキーマポリシーにバインドします。

   add authentication loginSchema login1 -authenticationSchema login-2passwd.xml -userCredentialIndex 1 -passwordCredentialIndex 2

   注:

   StoreFront などのバックエンドサービスへのシングルサインオン（SSO）には、ログインスキーマに入力したユーザー名とパスワードのいずれかを使用します。AAA.USER.ATTRIBUTE (#) 式を使用して、トラフィックアクションでこれらのインデックス値を参照できます。値は 1 から 16 の範囲で指定できます。

   または、次のコマンドを使用して、ログインスキーマに入力された資格情報をシングルサインオンの認証情報として使用できます。

```
add authentication loginSchema login1 -authenticationSchema login
    -2passwd.xml -SSOCredentials YES

 add authentication loginSchemaPolicy login1 -rule true -action
     login1
<!--NeedCopy-->
```

4. パススルーのログインスキーマを構成し、ポリシーラベルにバインドします。

```
add authentication loginSchema login2 -authenticationSchema
    noschema

add authentication policylabel label1 -loginSchema login2
<!--NeedCopy-->
```

5. LDAP ポリシーと RADIUS ポリシーを設定します。

```
add authentication ldapAction ldapAct1 -serverIP 10.17.103.28 -
    ldapBase "dc=aaatm, dc=com" -ldapBindDn administrator@aaatm.com
     -ldapBindDnPassword 81
    qw1b99ui971mn1289op1abc12542389b1f6c111n0d98e1d78ae90c8545901 -
    encrypted -encryptmethod ENCMTHD_3 -ldapLoginName
    samAccountName -groupAttrName memberOf -subAttributeName CN

add authentication Policy ldap -rule true -action ldapAct1

add authentication radiusAction radius -serverIP 10.101.14.3 -
    radKey
    n231d9a8cao8671or4a9ace940d8623babca0f092gfv4n5598ngc40b18876hj32
     -encrypted -encryptmethod ENCMTHD_3 -radNASip ENABLED -
    radNASid NS28.50 -radAttributeType 11 -ipAttributeType 8

add authentication Policy radius -rule true -action radius
<!--NeedCopy-->
```


6. ログインスキーマポリシーを認証仮想サーバーにバインドします。

```
bind authentication vserver auth56 -policy login1 -priority 1 -
    gotoPriorityExpression END
<!--NeedCopy-->
```

7. LDAP ポリシー (第 1 要素) を認証仮想サーバーにバインドします。

```
bind authentication vserver auth56 -policy ldap -priority 1 -
    nextFactor label1 -gotoPriorityExpression next
<!--NeedCopy-->
```

8. RADIUS ポリシー（第 2 要素）を認証ポリシーラベルにバインドします。

```
bind authentication policylabel label1 -policyName radius -
    priority 2 -gotoPriorityExpression end
<!--NeedCopy-->
```

構成を簡素化するための多要素（**nFactor**）ビジュアライザー

December 8, 2023

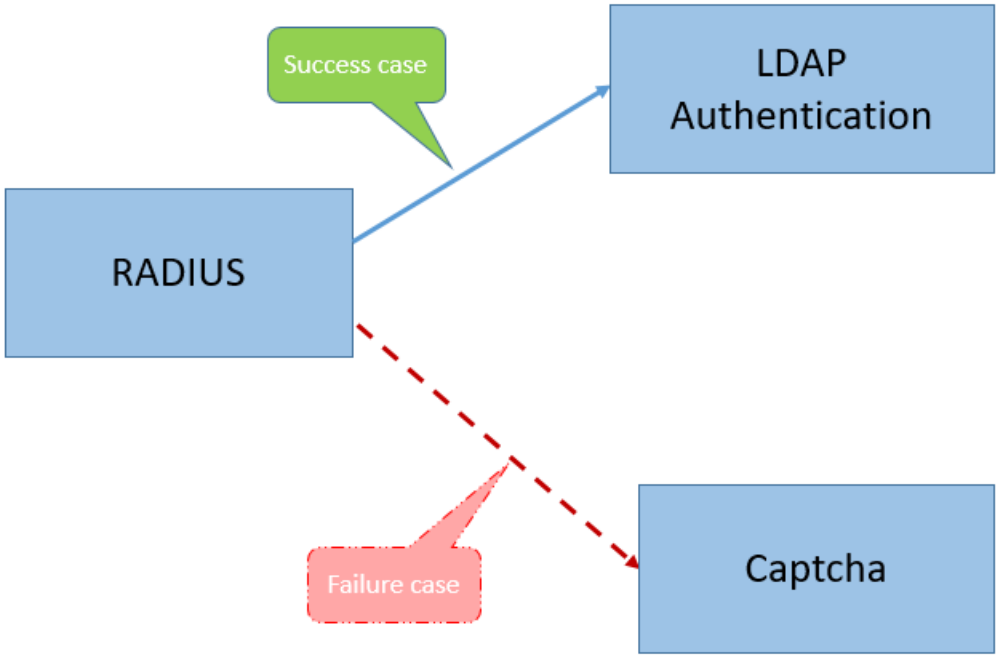
NetScaler リリース 13.0 ビルド 36.27 以降、GUI による nFactor 構成は nFactor ビジュアライザーを使用することで簡略化されています。nFactor ビジュアライザーを使用すると、管理者は各要素を見失うことなく複数の要素を追加できます。フローに組み込まれた要素のグループが 1 か所に表示されます。管理者は認証成功パスと失敗パスを別々に追加できます。フローを作成したら、管理者は nFactor フローを認証仮想サーバーにバインドする必要があります。

> 注
>
> - nFactor フローで管理者が作成したすべてのファクターは、今後の使用に備えて保持されます。
> - NetScaler 機能リリース 13.0 ビルド 64.35 以降では、nFactor ビジュアライザーを使用すると、デシジョンブロックを使用して nFactor フローを開始できます。

以前は、nFactor の設定は面倒で、管理者が多くのページにアクセスして設定する必要がありました。変更が必要な場合、管理者は毎回設定したセクションを再確認する必要がありました。また、構成全体を 1 か所に表示するオプションもありませんでした。

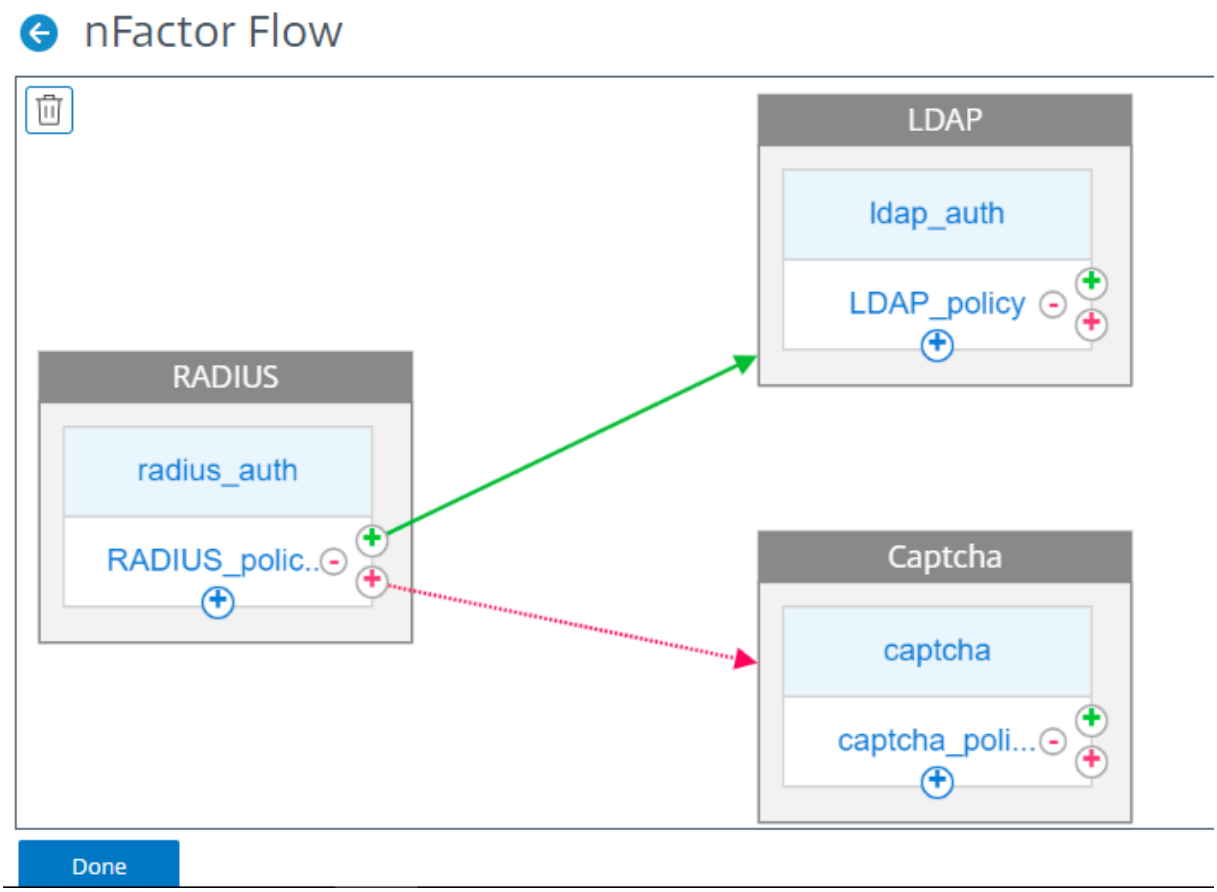
ユースケース **1: RADIUS** に続いて **LDAP** 認証、それ以外の場合は **nFactor** ビジュアライザーを使用してキャプチャにフォールバックする

第 1 レベルの認証として RADIUS 認証を行い、次に LDAP 認証を行います。RADIUS が失敗した場合、認証は Captcha にフォールバックする必要があります。



このユースケースを実現するには、nFactor ビジュアライザーを使用できます。ビジュアライザーには、このフローと関連項目を追加するために使用できるさまざまなコントロールが用意されています。

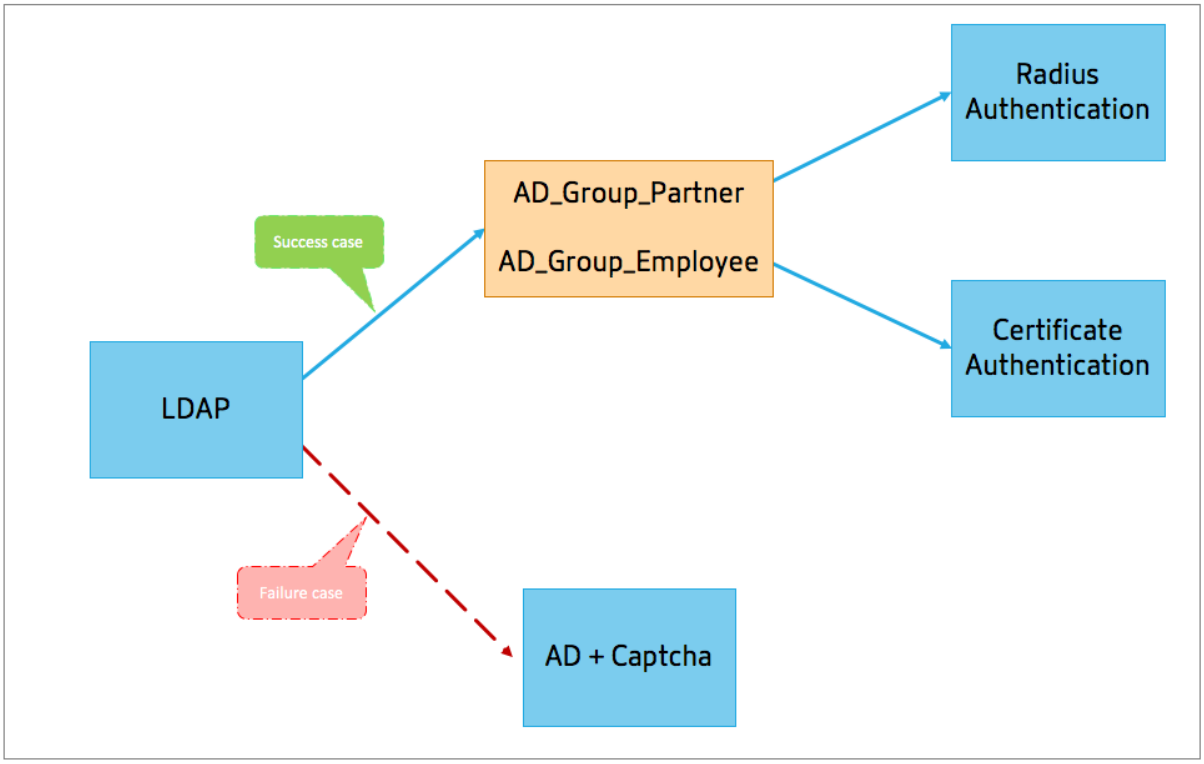
次の図は、前述のユースケースでビジュアライザーを使用して作成された nFactor フローを示しています。



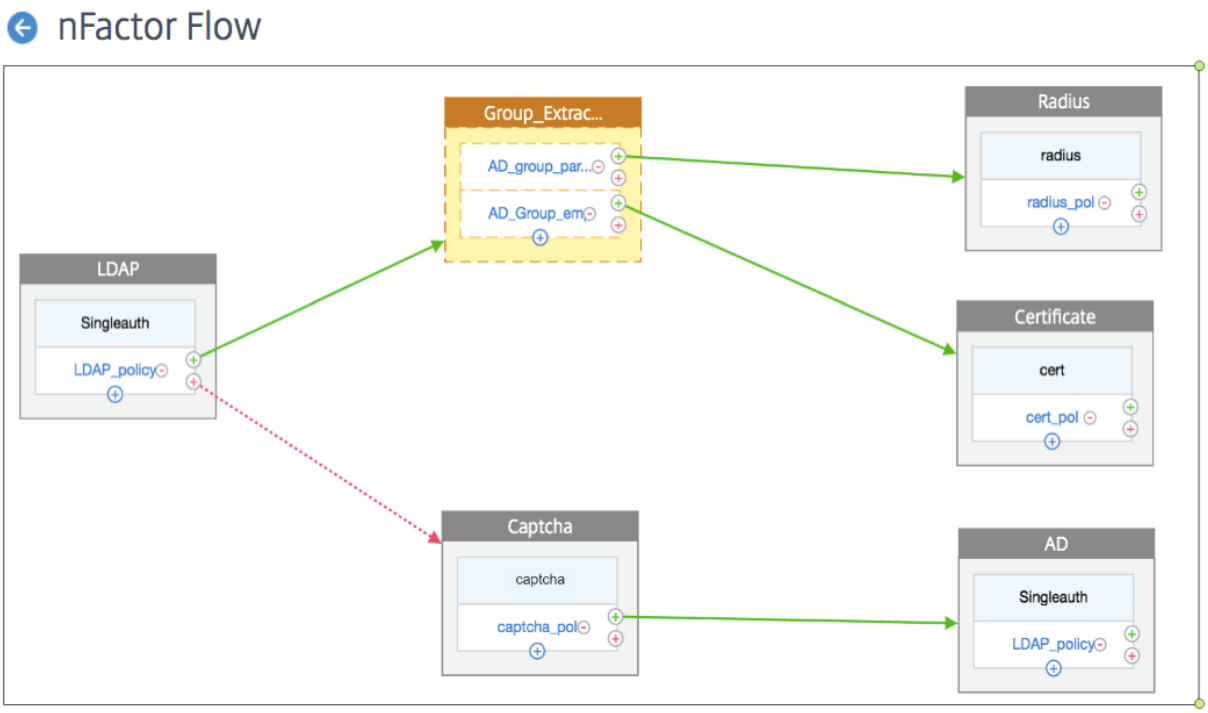
- **RADIUS**。RADIUS を最初の要素として設定します。ログインスキーマとポリシーを追加します。この例では、radius_auth と radius_Policy が追加されるログインスキーマとポリシーです。RADIUS_Policy については、成功事例の要因をもう 1 つ追加できます。この例では、成功事例として LDAP ファクターブロックが追加されています。失敗した場合は、Captcha ファクターを追加できます。
- **LDAP**。2 番目の要素として LDAP 認証を設定します。ログインスキーマとポリシーを追加します。この例では、追加されるログインスキーマとポリシーは ldap_auth と LDAP_Policy です。
- キャプチャ。RADIUS ポリシーが失敗した場合に備えて、キャプチャファクターを作成します。この例では、追加されるログインスキーマとポリシーは captcha と captcha_policy です。

使用事例 **2: LDAP** に続いて **nFactor** ビジュアライザーによる **LDAP** グループメンバーシップに基づくキャプチャによる **RADIUS/**証明書認証

第 1 レベルの認証として RADIUS 認証を行い、次に LDAP 認証を行います。RADIUS が失敗した場合、認証は Captcha にフォールバックする必要があります。



次の図は、前述のユースケースでビジュアライザーを使用して作成された nFactor フローを示しています。



- **LDAP**。LDAP を最初の要素として設定します。ログインスキーマとポリシーを追加します。この例では、SingleAuth と LDAP_Policy が追加されているログインスキーマとポリシーです。LDAP_Policy については、成功事例の要因をもう 1 つ追加できます。この例では、成功事例のデシジョンブロックが追加されていま

す。失敗した場合は、Captcha の後に AD ファクターを追加することができます。

- グループ抽出 **LDAP**。LDAP の成功事例に追加されたデシジョンブロックですか。デシジョンブロックは、ポリシー・ルールに基づいてユーザーを分岐させるための分岐要因として使用されます。ビジュアライザーでは、デシジョンブロックに NO_AUTHN ポリシーのみを設定できます。

  この例では、グループ _ 抽出 _LDAP がデシジョンブロックです。このデシジョンブロックには 2 つのポリシー (`AD_Group_Partner and AD_Group_Employee`) を追加します。ユースケースで説明したように、AD_Group_Partner ポリシーを介してルーティングされるすべてのリクエストは RADIUS 認証を使用します。したがって、このポリシーの成功事例を次の要因である RADIUS ファクターに結び付けます。同様に、ad_Group_Employee ポリシーを介してルーティングされるすべてのリクエストは、証明書認証を使用します。したがって、このポリシーの成功事例を次の要素、つまり証明書認証要素に結び付けることになります。

  - **RADIUS**。AD_Group_Partner ポリシーの成功事例では、RADIUS 認証ファクターを作成します。
  - 証明書。AD_Group_Employee ポリシーの成功事例では、証明書認証ファクターを作成します。

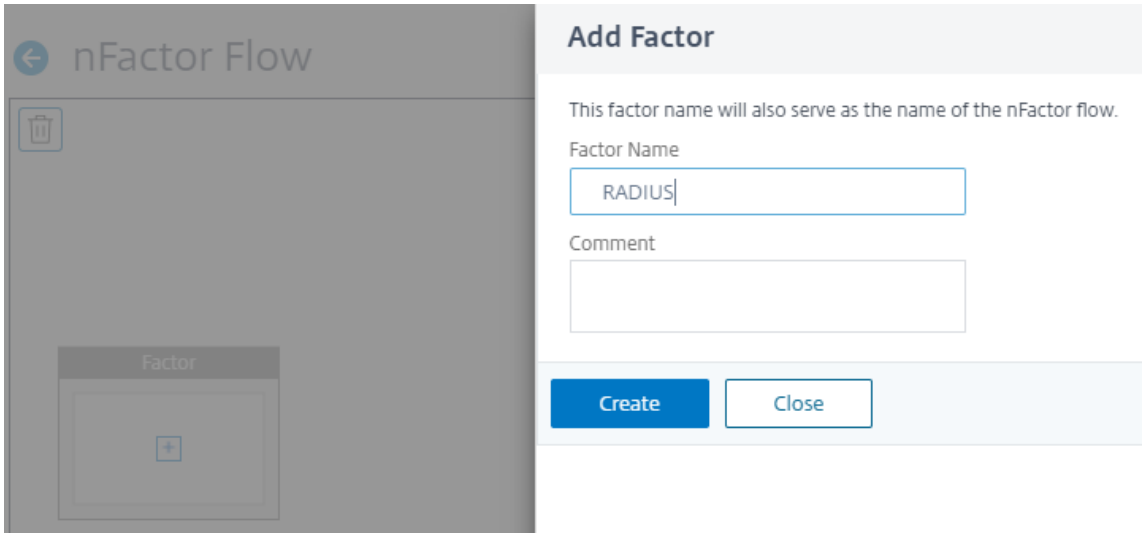
- キャプチャ。LDAP ポリシーが失敗した場合、次の 2 つのファクター、つまり Captcha と AD ファクターを作成します。

> 注
>
> - 最初にブランチアウトするユースケースがある場合は、2 つのフローを作成して別々にバインドすることも、1 つ目のフローをブランチアウトとして 1 つのフローを作成して仮想サーバーにバインドすることもできます。
> - ブロックが複数あり、nFactor Flow 画面にフロー全体を表示するには、ビジュアライザーをクリックしてフローを左端にドラッグします。
> - nFactor フローは、nFactor フローページのみを使用して変更することをお勧めします。

nFactor ビジュアライザーを使用して **nFactor** を設定するには

> 注**:**
>
> 以下の nFactor 設定は、ユースケース 1 のシナリオ構成を実行するのに役立つ簡単な例です。

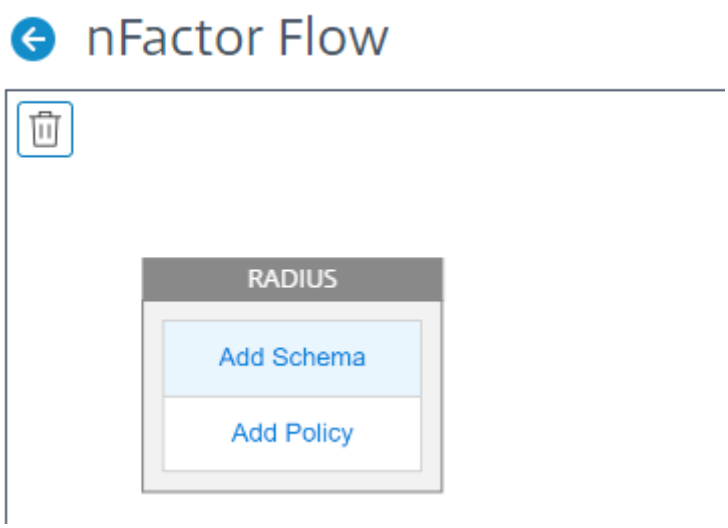
1. [セキュリティ**] > [AAA** ―アプリケーショントラフィック**] > [nFactor** ビジュアライザー**] > [nFactor** フロー] に移動します。
2. [追加] をクリックします。
3. **nFactor** フローページで、**+** をクリックしてフローの最初のファクターを追加します。最初のファクターは、この nFactor フローの識別子としても機能します。
4. ファクター名を入力して、「作成」をクリックします。



ファクター名は nFactor Flow ページのファクターブロックに表示されます。

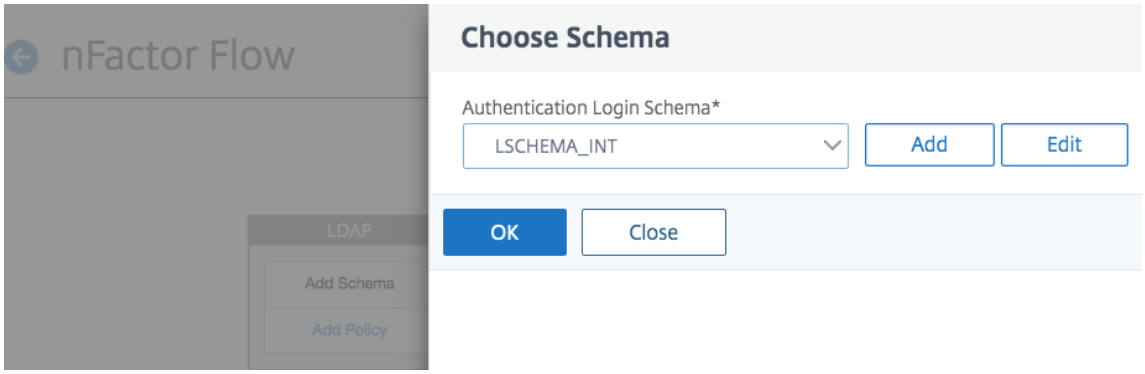
> 注:
>
> `__root`、`__<flow_name>` などのポリシーラベル名をサフィックスや`_db_`をプレフィックスとして使用しないことをお勧めします。nFactor フローで作成されるファクター名として使用されます。

5. RADIUS ファクターを作成したら、「スキーマの追加」と「ポリシーの追加」を作成する必要があります。

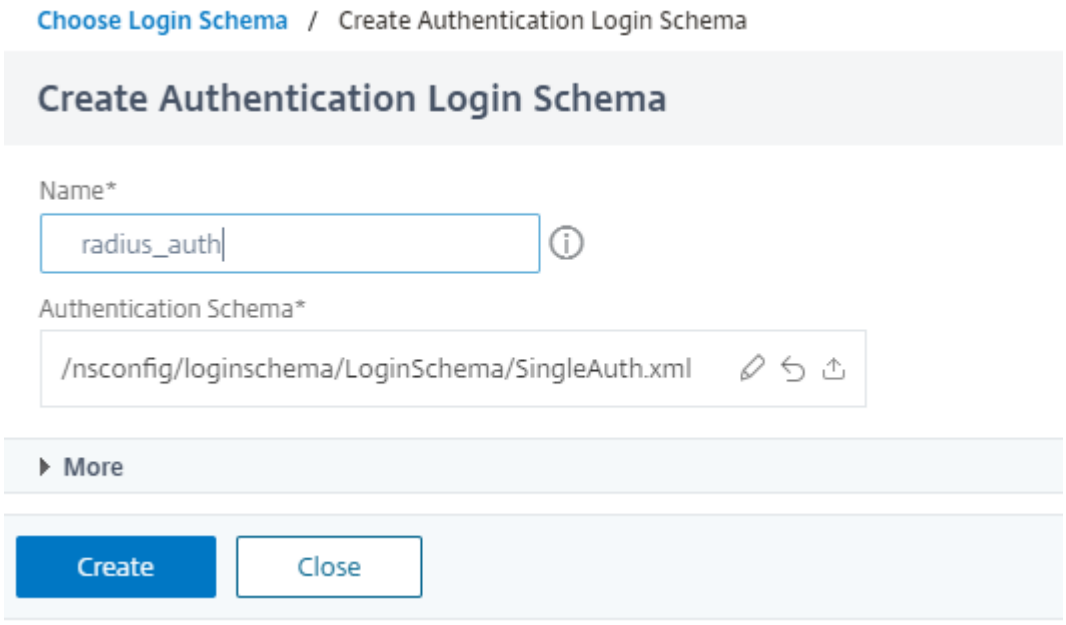


> 注
>
> 詳細については、nFactor の概念、エンティティ、および用語を参照してください。

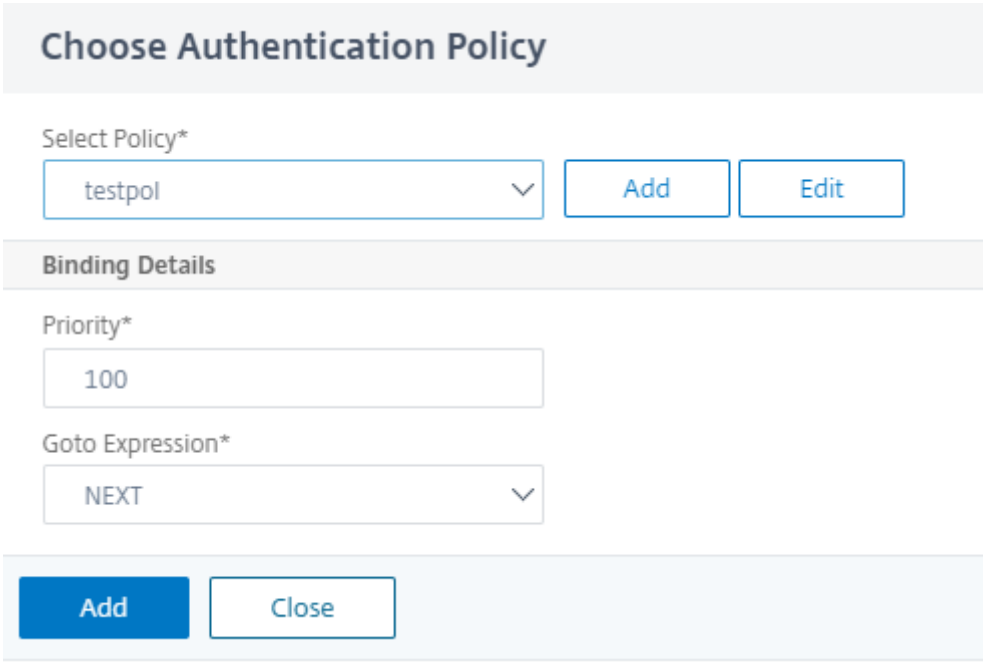
6. [スキーマの追加] をクリックします。新しいログインスキーマを追加するか、認証ログインスキーマリストから既存のログインスキーマを選択できます。



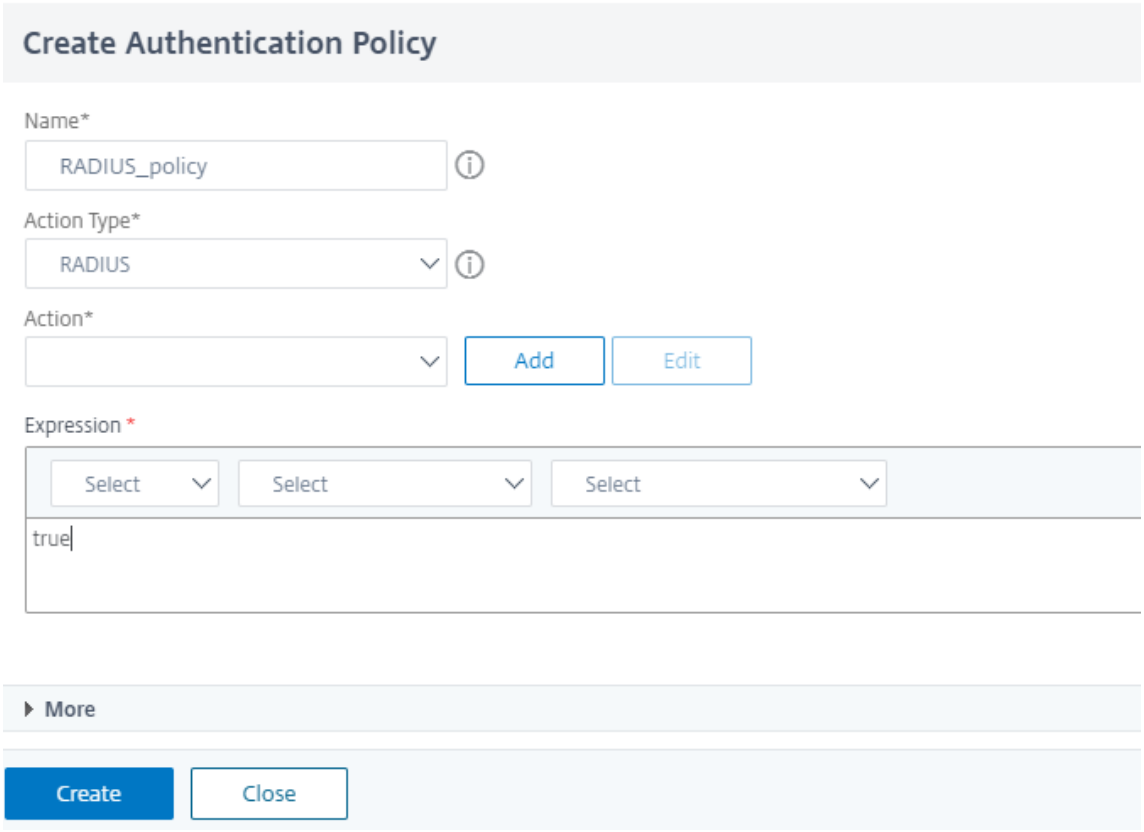
7. ログインスキーマを作成するには、「追加」をクリックし、「認証ログインスキーマの作成」ページにスキーマの名前を入力します。編集 (鉛筆アイコン) をクリックして、一覧からログインスキーマファイルを選択します。



8. 「ポリシーを追加」をクリックします。新しいポリシーを作成するか、既存の認証ポリシーを選択できます。

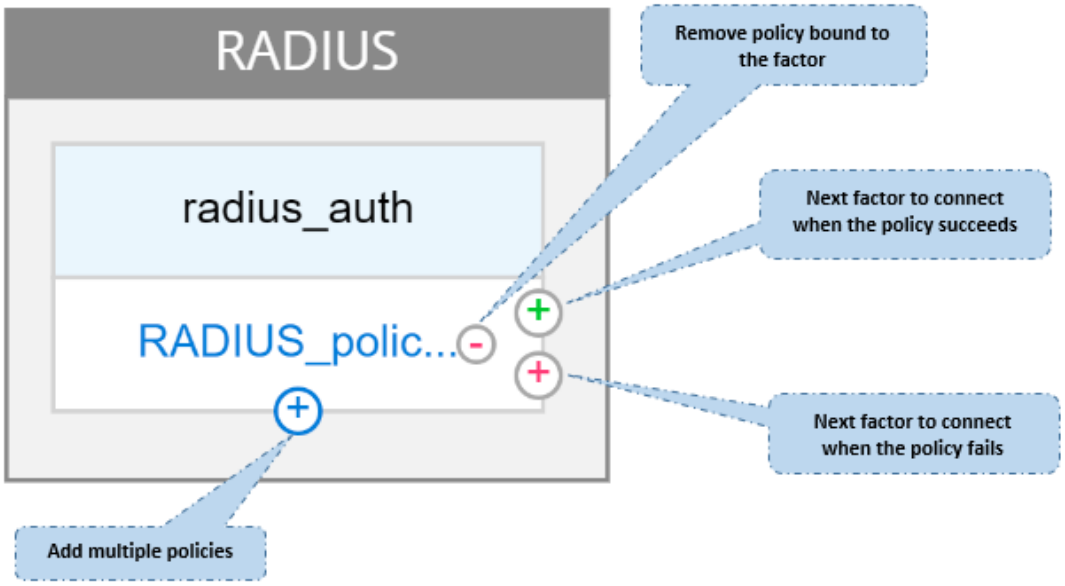


9. 新しいポリシーを作成するには、「追加」をクリックし、「認証ポリシーの作成」ページでポリシーの名前を入力して「作成」をクリックします。



10. ログインスキーマとポリシーをファクターに追加すると、次の図に示すように、ログインスキーマとポリシーがビジュアライザーのファクターに表示されます。どの要素についても、複数のポリシーを追加し、各ポリシ

ーの成否を判断する次の要素を定義できます。ファクターの一部であるポリシーを削除することもできます。



11. フローを作成したら、nFactor フローを認証仮想サーバーにバインドできます。

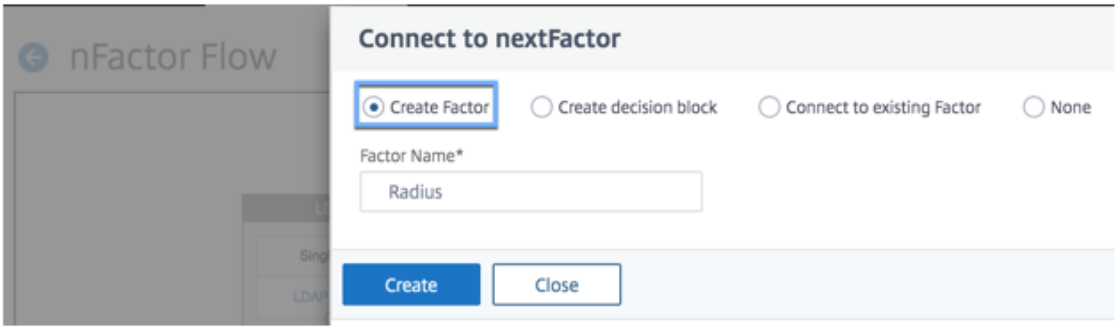
次のファクターの追加

次の要素を追加するには、要件に応じて次のオプションのいずれかを選択できます。

- ファクターの作成。ファクターを作成します。フローで作成される各ファクターは、そのフロー専用です。
- デシジョンブロックを作成します。分岐要因となるデシジョンブロックを作成します。ログインスキーマをデシジョンブロックに追加することはできません。ビジュアライザーでは、デシジョンブロックに NO_AUTHN ポリシーのみを設定できます。

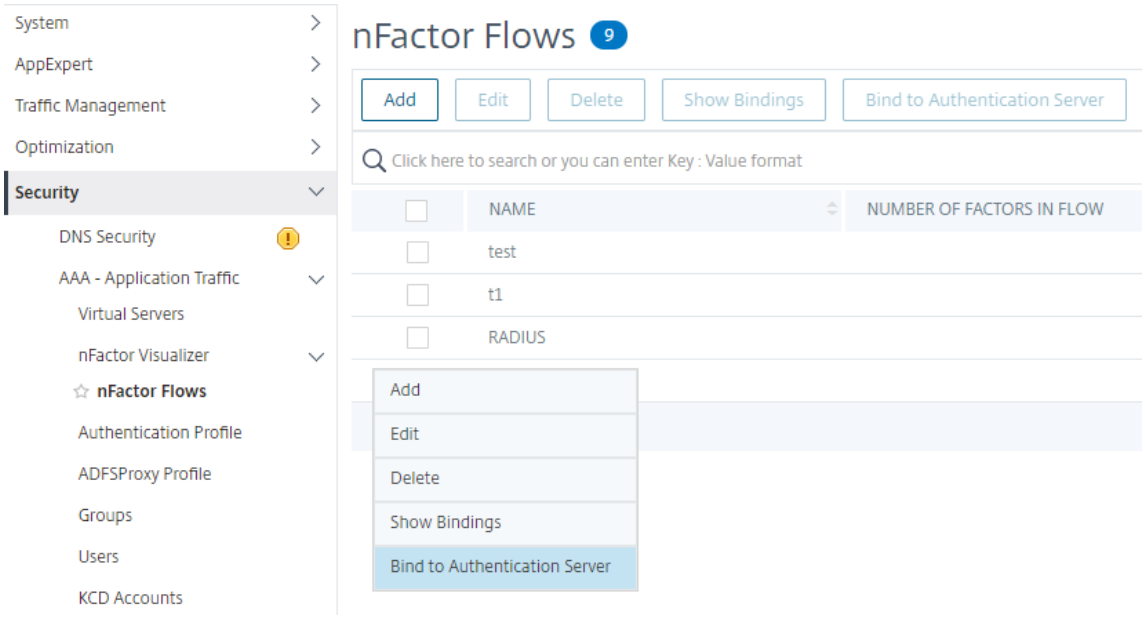
  > 注
  >
  > デシジョンブロックを追加または編集できるのは、NetScaler GUI からのみです。CLI コマンドからデシジョンブロックを設定するオプションはありません。

- 既存のファクターに接続します。既存のファクターを次のファクターとして選択します。既存のリストに表示されるすべてのファクターは、そのフロー専用に作成されます。
- なし。既存の接続を削除します。



nFactor フローを認証サーバーにバインドするには

1. **nFactor** フローページで、認証仮想サーバーにバインドしたい nFactor フローを選択します。
2. 横の省略記号をクリックして [認証サーバーにバインド**]** を選択するか、**[**nFactor** フロー] ページで [認証サーバーにバインド **] をクリックします。



3.「認証サーバーへのバインド」ページでは、次のアクションを実行できます。

   - 認証仮想サーバーを追加するには、「追加」をクリックします。
   - リストから既存の認証サーバーを選択するには、「認証サーバー」フィールドをクリックします。

4. ハンバーガーアイコンから「バインディングを表示」をクリックすると、バインディングが表示されます。
5. 認証サーバーを特定の nFactor フローからバインド解除するには、次の手順を実行します。

   - **nFactor** フローページで、ハンバーガーアイコンから「バインディングを表示」をクリックします。
   -「認証サーバーのバインディング」ページで、バインド解除する認証サーバーを選択し、「バインド解除」をクリックします。[閉じる] をクリックします。

nFactor 認証の詳細については、次のトピックを参照してください:

- 概念: 多要素 (nFactor) 認証。
- ワークフロー: nFactor 認証の仕組み。
- 構成:nFactor 認証の設定。

nFactor ビジュアライザーの機能強化

NetScaler リリース 13.0 ビルド 41.20 以降、nFactor ビジュアライザーでは以下の機能強化が行われています。

- 管理者は作成したファクターをごみ箱アイコンに移動できます。
- 認証仮想サーバーページで nFactor フローを表示します。

ゴミ箱アイコン。管理者は接続されていないノードのみを削除できます。ただし、ファクターをゴミ箱に移動しても、ファクターの基礎となるポリシーや作成されたスキーマは削除されません。

ゴミ箱アイコンを表示するには、

1. [セキュリティ**] > [AAA** ―アプリケーショントラフィック**] > [nFactor** ビジュアライザー**] > [nFactor** フロー] に移動します。左上隅にゴミ箱アイコンが表示されます。
2. ファクターを削除するには、ファクターブロックをクリックしてゴミ箱にドラッグします。

認証仮想サーバーからの **nFactor** フローを表示します。管理者は、作成した nFactor フローを認証仮想サーバーページから確認することもできます。

認証仮想サーバーページから nFactor フローを表示するには、

1. [セキュリティ**] > [AAA** ―アプリケーショントラフィック**] > [**仮想サーバ] に移動します。認証仮想サーバーページでは、次の手順を実行できます。
   - 認証仮想サーバーを追加するには、「追加」をクリックします。
   - 既存の認証仮想サーバーを編集するには、詳細ペインの [編集] オプションをクリックします。
2. 認証仮想サーバーページの「高度な認証ポリシー」に **nFactor Flow** オプションが表示されます。
3. 仮想サーバーにバインドされている nFactor フローがない場合は、[詳細認証ポリシー] セクションの [**No nFactor Flow**] オプションをクリックして、新しい nFactor フローを追加するか、リストから既存の nFactor フローを選択できます。

nFactor 拡張性

December 8, 2023

nFactor 認証フレームワークは、カスタマイズを追加できる柔軟性を提供し、ログオンインターフェイスをより直感的にし、リッチなユーザーエクスペリエンスを実現します。カスタムログインラベル、カスタムログイン認証情報、UI 表示のカスタマイズなどを追加できます。

nFactor を使用すると、各ファクターに独自のログオン画面を持つことができます。各ログオン画面には、以前の要素からの情報や、他の要素では見えない情報を表示できます。たとえば、最後の要素は、ユーザーが指示を読み、[続行] をクリックする情報ページです。

nFactor 以前は、カスタムログインページには制限があり、カスタマイズとサポートが必要でした。tmindex.html を置き換えたり、書き換えルールを適用したりして、その動作の一部を変更することができました。しかし、基盤となる機能を実現することは不可能でした。

このトピックでは、次の nFactor 関連のカスタマイズについて詳しく説明します。

- ログインラベルをカスタマイズする
- UI をカスタマイズして画像を表示する
- NetScaler nFactor ログオンフォームのカスタマイズ

仮定

nFactor、シェルコマンド、XML、およびテキストエディタに精通している。

前提条件

- このトピックで説明するカスタマイズは、RFWeb UI テーマ（またはテーマベース）が NetScaler で構成されている場合にのみ可能です。
- 認証ポリシーは、認証、承認、および監査仮想サーバーにバインドする必要があります。バインドしないと、フローが意図したとおりに機能しません。
- nFactor に関連する次のアイテムがあります
  - XML スキーマ
  - JavaScript
  - 認証アクション
  - 認証仮想サーバー
  - NetScaler バージョン 11.1 以降

ログオンラベルをカスタマイズする

ログオンラベルをカスタマイズするには、次のものが必要です。

- ログオンページの外観を記述する XML スキーマ。
- レンダリングプロセスの変更に使用される JavaScript を含む script.js ファイル。

> 注:
>
> script.js ファイルはディレクトリ/var/netscaler/logon/themes/<custom_theme>/にあります。

機能

JavaScript は XML ファイルを解析し、<Requirements> タグ内の各項目をレンダリングします。各要素は HTML フォームの 1 行に対応します。たとえば、ログインフィールドは行、パスワードフィールドは別の行、ログオンボタンも行です。新しい行を導入するには、StoreFront SDK を使用して XML スキーマファイルで新しい行を指定する必要があります。StoreFront SDK では、XML スキーマのあるログオンページで<Requirement>タグを使用し、そのタグ上の要素を定義できます。これらの要素により、JavaScript を使用して、必要な HTML 要素が何であれ、その空間に導入することができます。この場合、HTML 形式のテキストを含む行が作成されます。

使用できる XML は、次のとおりです。

```
<Requirement>
<Credential>
<Type>nsg-custom-cred</Type>
<ID>passwd</ID>
</Credential>
<Label>
<Type>nsg-custom-label</Type>
</Label>
</Requirement>
<!--NeedCopy-->
```

<Requirement>: ログオンページで提供される領域。クレデンシャルはスペースを埋め、他の部分はエンジンを正しい情報にルーティングします。この場合は、nsg-custom-credと入力します。これはプレーンテキストとして定義され、ラベルはその本文に対して定義されます。
要件 XML は JavaScript コードとペアになっており、必要な結果が得られます。

```
// Custom Label Handler for Self Service Links
CTXS.ExtensionAPI.addCustomAuthLabelHandler({

getLabelTypeName: function () {
  return "nsg-custom-label";  }
 ,
getLabelTypeMarkup: function (requirements) {

return $("< Enter your HTML codes here>");
 }
 ,
// Instruction to parse the label as if it was a standard type
parseAsType: function () {

return "plain";
 }

```



```
 }
 );
//Custom Credential Handler for Self Service Links
CTXS.ExtensionAPI.addCustomCredentialHandler({

getCredentialTypeName: function () {
  return "nsg-custom-cred";  }
 ,
getCredentialTypeMarkup: function (requirements) {

return $("<div/>");
 }
 ,
 }
 );
<!--NeedCopy-->
```

> 重要:
>
> HTML コードを追加するときは、戻り値が HTML タグで始まることを確認してください。

XML の部分はログオンページに何を表示するかを示し、JavaScript コードは実際のテキストを提供します。認証情報ハンドラがスペースを開き、ラベルがスペースを埋めます。これで、すべての認証トラフィックが書き換えや応答側から見えなくなるため、ページのルックアンドフィールを変更できます。
ログインラベルをカスタマイズする設定

1. rfWeb に基づいてテーマを作成してバインドします。

   ```
   add vpn portaltheme RfWebUI_MOD -basetheme RfWebUI

   bind vpn vserver TESTAAA  - portaltheme RfWebUI_MOD
   <!--NeedCopy-->
   ```

   テーマに基づいたファイルのパスは、/var/NetScaler/logon/themes/rfwebui_MOD というディレクトリにあります。

2. script.js ファイルの最後に次のスニペットを追加します。

   > **注:**
   >
   > 前の行を正しいファイル内に含めなかったり、JavaScript 関数を含めなかったりすると、XML がロードされません。このエラーは、ブラウザの開発者コンソールで「Undefined Type nsg-custom-cred」というテキストでのみ表示されます。

   ```
   // Custom Label Handler for Self Service Links
   CTXS.ExtensionAPI.addCustomAuthLabelHandler({

   getLabelTypeName: function () {
     return "nsg-custom-label";  }
    ,
   getLabelTypeMarkup: function (requirements) {
   ```



```
8
9  return $("<a href="https://identity.test.com/identity/faces/
       register" style="font-size: 16px;" style="text-align: center;">
       Self Registration</a><br><a href="https://identity.test.com/
       identity/faces/forgotpassword" style="font-size: 16px;" style="
       text-align: center;">Forgot Password</a><br><a href="https://
       identity.test.com/identity/faces/forgotuserlogin" style="font-
       size: 16px;" style="text-align: center;">Forgot User Login</a
       >");
10  }
11  ,
12 // Instruction to parse the label as if it was a standard type
13 parseAsType: function () {
14
15 return "plain";
16  }
17
18  }
19  );
20 //Custom Credential Handler for Self Service Links
21 CTXS.ExtensionAPI.addCustomCredentialHandler({
22
23 getCredentialTypeName: function () {
24   return "nsg-custom-cred";  }
25  ,
26 getCredentialTypeMarkup: function (requirements) {
27
28 return $("<div/>");
29  }
30  ,
31  }
32  );
33 <!--NeedCopy-->
```

> 重要:
>
> HTML コードを追加するときは、戻り値が HTML タグで始まることを確認してください。

この例で使用されるログインスキーマ

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <AuthenticateResponse xmlns="http://citrix.com/authentication/response
       /1">
3  <Status>success</Status>
4  <Result>more-info</Result>
5  <StateContext/>
6  <AuthenticationRequirements>
7  <PostBack>/nf/auth/doAuthentication.do</PostBack>
8  <CancelPostBack>/Citrix/Authentication/ExplicitForms/CancelAuthenticate
       </CancelPostBack>
9  <CancelButtonText>Cancel</CancelButtonText>
```



```
<Requirements>
<Requirement>
<Credential>
<ID>login</ID>
<SaveID>Username</SaveID>
<Type>username</Type>
</Credential>
<Label>
<Text>User name</Text>
<Type>plain</Type>
</Label>
<Input>
<AssistiveText>Please supply either domain\username or user@fully.
   qualified.domain</AssistiveText>
<Text>
<Secret>false</Secret>
<ReadOnly>false</ReadOnly>
<InitialValue></InitialValue>
<Constraint>.+</Constraint>
</Text>
</Input>
</Requirement>
<Requirement>
<Credential>
<ID>passwd</ID>
<SaveID>Password</SaveID>
<Type>password</Type>
</Credential>
<Label>
<Text>Password:</Text>
<Type>plain</Type>
</Label>
<Input>
<Text>
<Secret>true</Secret>
<ReadOnly>false</ReadOnly>
<InitialValue/>
<Constraint>.+</Constraint>
</Text>
</Input>
</Requirement>
<Requirement>
<Credential>
<Type>nsg-custom-cred</Type>
<ID>passwd</ID>
</Credential>
<Label>
<Type>nsg-custom-label</Type>
</Label>
</Requirement>
<Requirement>
<Credential>
<ID>loginBtn</ID>
```



```
<Type>none</Type>
</Credential>
<Label>
<Type>none</Type>
</Label>
<Input>
<Button>Please Log On</Button>
</Input>
</Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

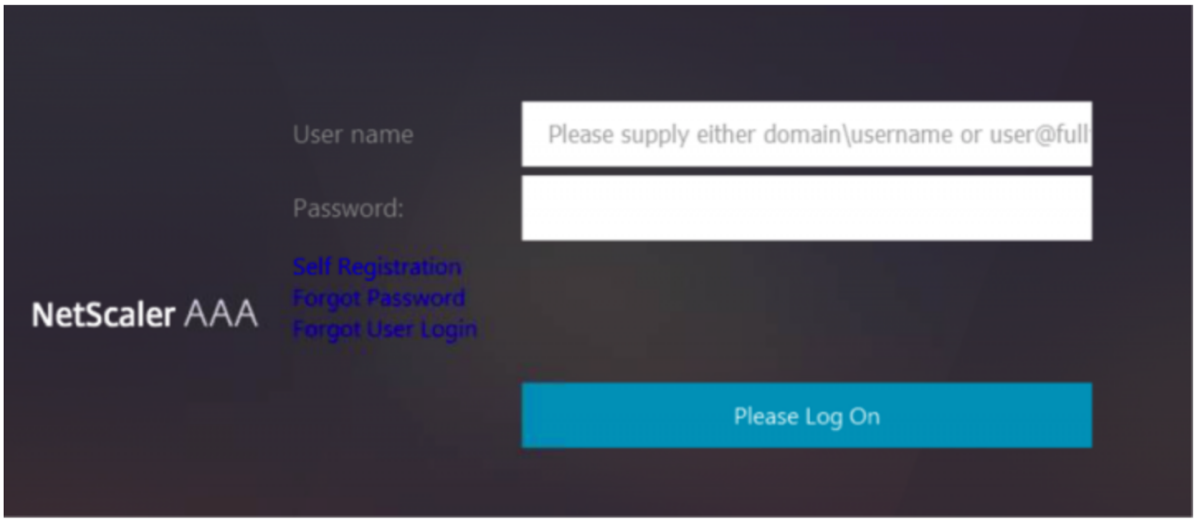
次のコマンドを実行して、カスタムスキーマを config に読み込みます。

```
add authentication loginSchema custom -authenticationSchema custom.xml

add authentication loginSchemaPolicy custom -rule true -action custom

bind authentication vserver AAATEST -policy custom -priority 100 -
   gotoPriorityExpression END
<!--NeedCopy-->
```

次の図は、この構成でレンダリングされるログインページを示しています。



UI をカスタマイズして画像を表示する

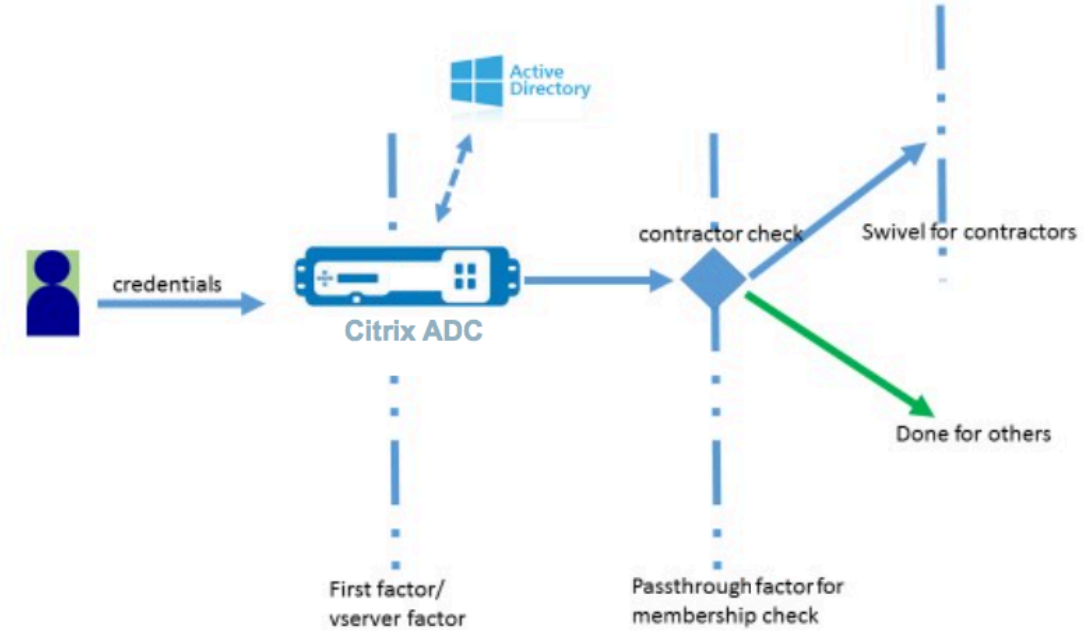
nFactor では、ログインスキーマファイルを使用して表示をカスタマイズできます。組み込みのログインスキーマファイルで提供されるもの以外に、さらにカスタマイズが必要な場合があります。たとえば、UI にハイパーリンクを表示したり、カスタムロジックを記述したりします。これは、ログインスキーマ拡張と対応する JavaScript ファイルで構成される「カスタム認証情報」を使用して実現できます。

ログインスキーマファイルは/nsconfig/loginschema/LoginSchemaディレクトリにあります。

画像を表示するための UI のカスタマイズについては、「NetScaler-Swivel」統合のデプロイフローを例として使用します。

このフローには 2 つの要因があります。

- 第 1 要素: ユーザーの AD 資格情報を確認します。
- 第 2 要因: グループメンバーシップに基づいてユーザーログオンを要求します。



このフローでは、すべてのユーザーが第 1 の要素を通過します。2 番目の要素の前に、一部のユーザーを「旋回」係数から除外できるかどうかを確認する疑似要素があります。ユーザーが「回転」係数を必要とする場合、コードを入力するための画像とテキストボックスが表示されます。

解決策

UI をカスタマイズして画像を表示するソリューションには、次の 2 つの部分があります。

- ログインスキーマ拡張。
- ログインスキーマ拡張を処理するカスタムスクリプト。

ログインスキーマ拡張　フォームのレンダリングを制御するために、カスタム‘id’/’credential’がログインスキーマに注入されます。これは、既存のスキーマを再利用し、要件に従って変更することで実現できます。

この例では、テキストフィールド (/nsconfig/loginschema/LoginSchema/OnlyPassword.xml など) が 1 つしかないログインスキーマが考慮されます。
次のスニペットがログインスキーマに追加されます。


```
<Requirement><Credential><ID>swivel_cred</ID><Type>swivel_cred</Type><
   Input><Text><Hidden>true</Hidden><InitialValue>${
 http.req.user.name }
 </InitialValue></Text></Input></Credential></Requirement>
<!--NeedCopy-->
```

このスニペットでは、認証情報の「タイプ」として「swivel_cred」が指定されています。これは組み込みの「認証情報」として認識されないため、UI はこのタイプのハンドラを探し、存在する場合はそれを呼び出します。
この認証情報には、NetScaler が動的に入力する式である初期値が送信されます。この例では、スイベルサーバーにユーザー名を通知するために使用されるユーザー名です。常に必要というわけではないかもしれませんし、他のデータで補強することもできます。これらの詳細は、必要に応じて追加する必要があります。

カスタム認証情報を処理する **Javascript**　UI はカスタム認証情報を検出すると、ハンドラーを探します。すべてのカスタム・ハンドラーは、デフォルトのポータル・テーマ用に /var/netscaler/logon/LogonPoint/custom/script.js に記述されます。

カスタムポータルテーマの場合、script.js はディレクトリ/var/netscaler/logon/themes/<custom_theme>/にあります。

カスタム認証情報のマークアップをレンダリングするために、次のスクリプトが追加されました。

```
CTXS.ExtensionAPI.addCustomCredentialHandler({

    // The name of the credential, must match the type returned by the
       server
    getCredentialTypeName: function () {
  return "swivel_cred";  }
 ,
    // Generate HTML for the custom credential
    getCredentialTypeMarkup: function (requirements) {

        var div = $("<div></div>");
        var image = $("<img/>");
        var username = requirements.input.text.initialValue; //Get the
           secret from the response
        image.attr({

            "style" : "width:200px;height:200px;",
            "id" : "qrcodeimg",
            "<Enter your server URL here>"
         }
 );
        div.append(image);
        return div;
     }

 }
 );
<!--NeedCopy-->
```


このスニペットは、「wivel_cred」のマークアップを処理するためのものです。強調表示された認証情報名は、ログインスキーマ拡張で以前に指定された「タイプ」と一致する必要があります。
マークアップを生成するには、ソースがスイベルサーバーを指すイメージを追加する必要があります。これが完了すると、UI は指定した場所からイメージをロードします。このログインスキーマにもテキストボックスがあるため、UI はそのテキストボックスをレンダリングします。

> 注:
>
> 管理者は、イメージ要素の「スタイル」を変更して、イメージのサイズを変更できます。現在、200x200 ピクセルに設定されています。

UI をカスタマイズして画像を表示するための設定

nFactor の構成は、ボトムアップで構築する方が適切です。これは、前の因子に‘nextFactor’を指定しようとすると、後続の因子の名前が必要になるため、最後の因子が先になります。

旋回係数構成:

```
add loginschema swivel_image  - authenticationSchema /nsconfig/
    loginschema/SwivelImage.xml

add authentication policylabel SwivelFactor  - loginSchema swivel_image

bind authentication policylabel SwivelFactor  - policy <policy-to-check-
    swivel-image> -priority 10
<!--NeedCopy-->
```

> 注:
>
> この例で使用されているログインスキーマから SwivelImage.xml をダウンロードします。

グループチェック設定の疑似係数:

```
add authentication policylabel GroupCheckFactor

add authentication policy contractors_auth_policy  - rule  'http.req.
    user.is_member_of( "contractors" )'  - action NO_AUTHN

add authentication policy not_contractors _auth_policy - rule true  -
    action NO_AUTHN

bind authentication policylabel GroupCheckFactor  - policy
    contractors_auth_policy  - pri 10  - nextFactor SwivelFactor

bind authentication policylabel GroupCheckFactor  - policy
    not_contractors_auth_policy  - pri 20
<!--NeedCopy-->
```

Active Directory ログインの第 **1** 要因:


```
add ldapAction <>

add authentication policy user_login_auth_policy  -rule true  -action
    <>

bind authentication vserver <> -policy user_login_auth_policy  -pri 10
     -nextFactor GroupCheckFactor
<!--NeedCopy-->
```

この構成では、暗黙的/疑似的要素の 3 つの要素が指定されています。

この例で使用されるログインスキーマ

次に、スイベル認証情報とテキストボックスを持つスキーマの例を示します。

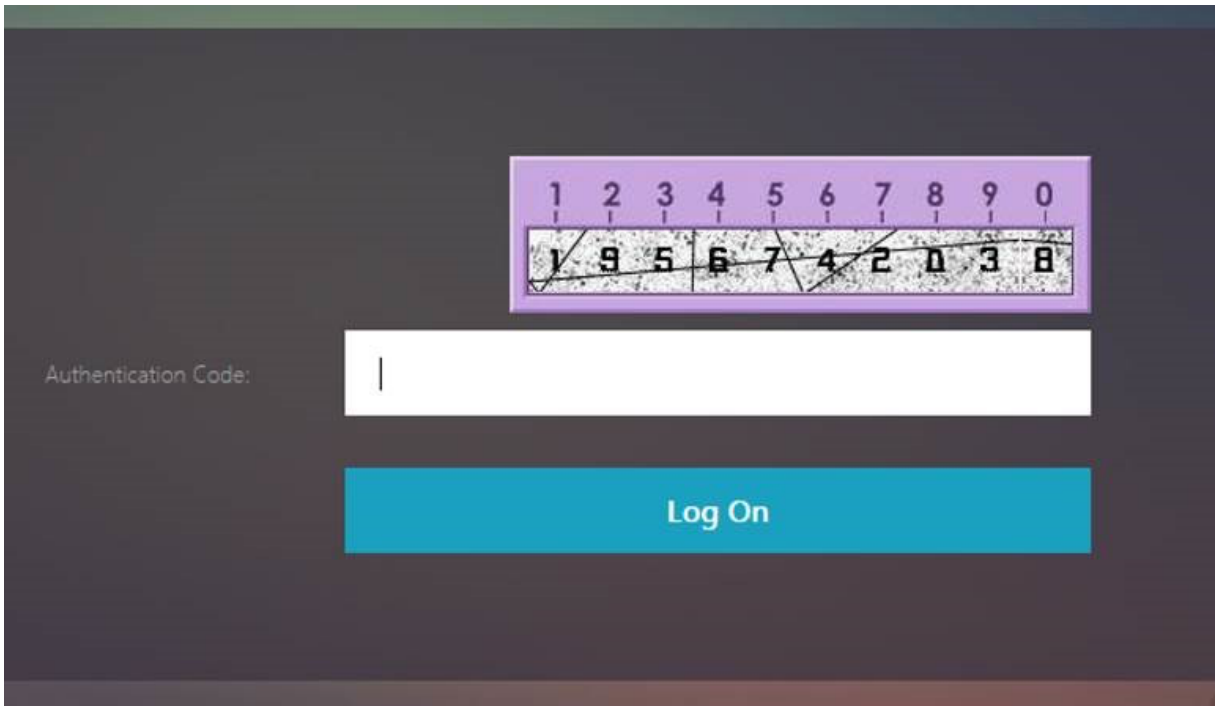
> 注:
>
> Web ブラウザ用にデータをコピーする場合、引用符の表示が異なる場合があります。ファイルに保存する前に、メモ帳などのエディターでデータをコピーします。

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response
    /1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext></StateContext>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement><Credential><ID>swivel_cred</ID><Type>swivel_cred</Type><
    Input><Text><Hidden>true</Hidden><InitialValue>${
 http.req.user.name }
 </InitialValue></Text></Input></Credential></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password
    </SaveID><Type>password</Type></Credential><Label><Text>Password:</
    Text><Type>plain</Type></Label><Input><Text><Secret>true</Secret><
    ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint
    >.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>
    Hello ${
 http.req.user.name }
 , Please enter passcode from above image.</Text><Type>confirmation</
     Type></Label><Input /></Requirement>
<Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials
    </Type></Credential><Label><Text>Remember my password</Text><Type>
    plain</Type></Label><Input><CheckBox><InitialValue>false</
    InitialValue></CheckBox></Input></Requirement>
```



```
<Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
    ><Label><Type>none</Type></Label><Input><Button>Log On</Button></
    Input></Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

出力　設定が完了すると、次のイメージが表示されます。



> 注:
>
> 画像の高さと配置は JavaScript で変更できます。

NetScaler nFactor ログオンフォームをカスタマイズしてフィールドを表示または非表示にする

NetScaler Gateway の RFWeb UI では、さまざまなカスタマイズが可能です。この機能を nFactor 認証フレームワークと組み合わせると、既存のワークフローを損なうことなく複雑なフローを構成できます。

この例では、[Logon Type] リストから OAuth と LDAP の 2 つの認証オプションを使用できます。フォームを初めて読み込むと、ユーザー名とパスワードのフィールド (LDAP が先に表示されます) が表示されます。OAuth を選択した場合、OAuth は認証をサードパーティサーバーにオフロードすることを暗示するため、すべてのフィールドが非表示になります。これにより、管理者はユーザーの利便性に応じて直感的なワークフローを構成できます。

> 注:
>
> - [Logon Type] リストの値は、スクリプトファイルに簡単に変更を加えるだけで変更できます。
> - このセクションでは、フローの UI 部分のみについて説明します。認証の実行時の処理については、この記事では説明しません。認証設定については、nFactor のドキュメントを参照することをお勧めします。

nFactor ログオンフォームをカスタマイズする方法

nFactor ログオンフォームのカスタマイズは 2 つの部分に分類できます

- 適切なログインスキーマを UI に送信する
- ログインスキーマとユーザー選択を解釈するハンドラーの作成

正しいログインスキーマを **UI** に送信する　この例では、単純なクレーム/要件がログインスキーマで送信されます。

このため、SingleAuth.xml ファイルは変更されます。SingleAuth.xml には NetScaler ファームウェアが付属しており、ディレクトリにあります。/nsconfig/loginschema/LoginSchema

ログインスキーマを送信する手順:

1. SSH 経由でログインし、シェルにドロップします (「shell」と入力します)。
2. SingleAuth.xml を別のファイルにコピーして変更します。

   > 注:
   >
   > 保存先フォルダーは、デフォルトの NetScaler ログインスキーマフォルダーとは異なります。
   >
   > cp　/nsconfig/loginschema/LoginSchema/SingleAuth.xml　/nsconfig/loginschema/SingleAuthDynamic.xml

3. SingleAuthDynamic.xml に次のクレームを追加します。

```
<Requirement><Credential><ID>nsg_dropdown</ID><Type>nsg_dropdown</Type></Credential><Label><Text>Logon Type:</Text><Type>plain</Type></Label></Requirement>
<!--NeedCopy-->
```

4. このログインスキーマを送信して最初のフォームをロードするように NetScaler を構成します。

```
add loginschema single_auth_dynamic -authenticationSchema SingleAuthDynamic.xml

add loginschemaPolicy single_auth_dynamic -rule true -action single_auth_dynamic

```



```
bind authentication vserver aaa_nfactor -policy
    single_auth_dynamic -pri 10
<!--NeedCopy-->
```

フォームを読み込み、ユーザーイベントを処理するためのスクリプト変更　管理者がログオンフォームの表示をカスタマイズできるように JavaScript を変更できます。この例では、LDAP が選択されている場合はユーザー名とパスワードのフィールドが表示され、OAuth を選択すると非表示になります。また、管理者はパスワードのみを非表示にすることもできます。

管理者は、「/var/NetScaler/ログオン/ログオンポイント/カスタム」ディレクトリにある「script.js」に次のスニペットを追加する必要があります。

> 注:
>
> このディレクトリはグローバルディレクトリなので、ポータル・テーマを作成し、"/var/netscaler/logon/themes/<THEME_NAME>"でそのフォルダー内の「script.js」ファイルを編集します。

```
CTXS.ExtensionAPI.addCustomCredentialHandler({

    // The name of the credential, must match the type returned by the
       server
    getCredentialTypeName: function () {
  return "nsg_dropdown";  }
 ,
    // Generate HTML for the custom credential
    getCredentialTypeMarkup: function (requirements) {

        var div = $("<div></div>");
        var select = $("<select name='nsg_dropdown'></select>").attr("
           id", "nsg_dropdown");

        var rsa = $("<option></option>").attr("selected", "selected").
           text("LDAP").val("LDAP");
        var OAuthID = $("<option></option>").text("OAuth").val("OAuth")
           ;
        select.append(rsa, OAuthID);

        select.change(function(e) {

            var value = $(this).val();
            var ldapPwd = $($(".credentialform").find(".
               CredentialTypepassword")[0]);
            var ldapUname = $($(".credentialform").find(".
               CredentialTypeusername"));
            if(value == "OAuth") {

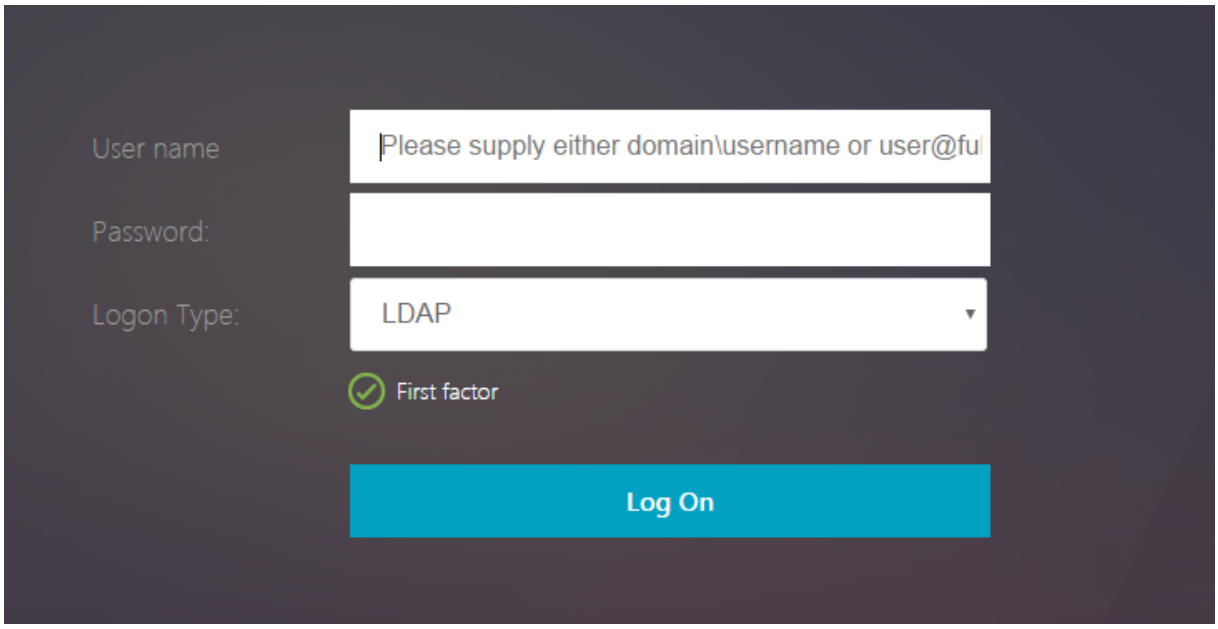
                if (ldapPwd.length)
                        ldapPwd.hide();
                if (ldapUname.length)
                        ldapUname.hide();
```



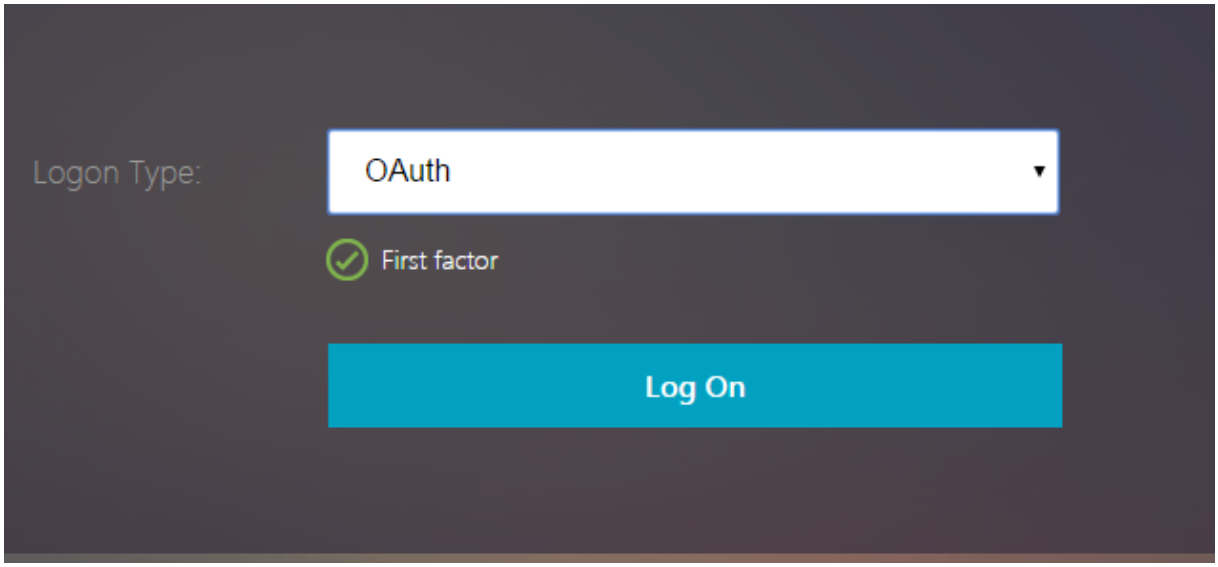
```
28                }
29    else if(value == "LDAP") {
30
31                   if (ldapPwd.length)
32                           ldapPwd.show();
33                   if (ldapUname.length)
34                           ldapUname.show();
35                }
36
37             }
38  );
39         div.append(select);
40          return div;
41       }
42
43  }
44  );
45  <!--NeedCopy-->
```

エンドユーザーエクスペリエンス

エンドユーザーが初めてログオンページを読み込むと、次の画面が表示されます。



［ログオンの種類］で［**OAuth**］が選択されている場合、［ユーザー名］および［パスワード］フィールドは非表示になります。



LDAP を選択すると、ユーザー名とパスワードが表示されます。これにより、ユーザーの選択に基づいてログオンページを動的に読み込むことができます。

この例で使用されるログインスキーマ

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response/1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext></StateContext>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement><Credential><ID>login</ID><SaveID>ExplicitForms-Username</SaveID><Type>username</Type></Credential><Label><Text>User name</Text><Type>plain</Type></Label><Input><AssistiveText>Please supply either domain\username or user@fully.qualified.domain</AssistiveText><Text><Secret>false</Secret><ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password</SaveID><Type>password</Type></Credential><Label><Text>Password:</Text><Type>plain</Type></Label><Input><Text><Secret>true</Secret><ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><ID>nsg_dropdown</ID><Type>nsg_dropdown</Type></Credential><Label><Text>Logon Type:</Text><Type>plain</Type></Label></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>First factor</Text><Type>confirmation</Type></Label><Input /></
```



```
      Requirement>
15 <Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials
      </Type></Credential><Label><Text>Remember my password</Text><Type>
      plain</Type></Label><Input><CheckBox><InitialValue>false</
      InitialValue></CheckBox></Input></Requirement>
16 <Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
      ><Label><Type>none</Type></Label><Input><Button>Log On</Button></
      Input></Requirement>
17 </Requirements>
18 </AuthenticationRequirements>
19 </AuthenticateResponse>
20 <!--NeedCopy-->
```

> 注:
>
> nFactor 関連のさまざまなトピックの詳細については、「nFactor 認証」を参照してください。

多要素（**nFactor**）を使用して **Cookie** を設定する

August 15, 2023

nFactor カスタムラベルを適用し、認証フローの要素として Cookie を設定できます。カスタムラベルを使用すると、JavaScript を使用してログインスキーマを操作できます。

Cookie を要素として設定するために、スキーマなしのログインで実行される情報をユーザーに表示する必要はありません。代わりに、ユーザーのブラウザと対話して、必要なデータを格納するようにログインスキーマに指示する必要があります。ログインスキーマは、ページが読み込まれたときに Cookie を設定するために必要です。クッキーはカスタムラベルと JavaScript コードで設定されます。

Cookie を設定する要素を実装するには、cookie.xml という名前の XML ファイルを作成し、スキーマを /nsconfig/loginschema/ ディレクトリに以下の内容で保存します。

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <AuthenticateResponse xmlns="http://citrix.com/authentication/response
      /1">
3  <Status>success</Status>
4  <Result>more-info</Result>
5  <StateContext></StateContext>
6  <AuthenticationRequirements>
7  <PostBack>/nf/auth/doAuthentication.do</PostBack>
8  <CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
9  <CancelButtonText>Cancel</CancelButtonText>
10 <Requirements>
11
12 <Requirement>
13 <Credential><ID>nsg_cookie</ID><Type>nsg_cookie</Type></Credential>
14 <Label><Text>Logon Type:</Text><Type>Plain</Type></Label>
15 </Requirement>
```



```
16
17 <Requirement>
18 <Credential><ID>loginBtn</ID><Type>none</Type></Credential>
19 <Label><Type>none</Type></Label><Input><Button>Log On</Button></Input>
20 </Requirement>
21
22 </Requirements>
23 </AuthenticationRequirements>
24 </AuthenticateResponse>
25 <!--NeedCopy-->
```

この XML では

- カスタムラベル nsg_cookie は、Cookie を作成してフォームを送信し、フォームボタンを送信するために使用されます。
- RFWebUI_Custom は、RFWebUI テーマに基づいた新しいポータルテーマです。

nFactor を使用してクッキーを設定する手順

1. RFWebUI テーマに基づいてポータルテーマを作成します。

   ```
   1 add vpn portaltheme RfWebUI_custom -basetheme RfWebUI
   2 <!--NeedCopy-->
   ```

   このコマンドは、/var/netscaler/logon/themes/rfwebUI_CUSTOM にこのテーマのフォルダーを作成します。

2. /var/netscaler/logon/themes/RfWebUI_custom/script.js ファイルを編集し、次のスクリプトを追加します。

   ```
   1  CTXS.ExtensionAPI.addCustomCredentialHandler({
   2
   3      // The name of the credential, must match the type returned by
              the server
   4      getCredentialTypeName: function () {
   5    return "nsg_cookie";  }
   6   ,
   7      // Generate HTML for the custom credential
   8      getCredentialTypeMarkup: function (requirements) {
   9
   10         var div = $("<div></div>");
   11         $(document).ready(function() {
   12
   13         //Set cookie valid for 1000 days
   14         var exdays = 1000;
   15         var d = new Date();
   16         d.setTime(d.getTime() + (exdays\*24\*60\*60\*1000));
   17         var expires = "expires="+ d.toUTCString();
   18         document.cookie = "NSC_COOKIE_NAME=CookieValue;" + expires
                  + ";path=/";
   ```



```
19
20          //Submit form
21          document.getElementById('loginBtn').click();
22           }
23   );
24          return div;
25       }
26
27   }
28   );
29  <!--NeedCopy-->
```

このコードは次の処理を実行します。

- ブラウザがページの読み込みを完了するのを待ちます。
- NSC_COOKIE_NAME という名前のクッキーを cookieValue という値で設定します。1000 日間有効です。
- フォームを自動送信します。

クッキーが作成され、ユーザーはページを操作する必要がありません。

3. ログインスキーマを作成して、set cookie ファクターを表すポリシーラベルにバインドします。

```
1  add authentication loginSchema Cookie_LS -authenticationSchema "/
      nsconfig/loginschema/cookie.xml"
2  <!--NeedCopy-->
```

4. NO_AUTHN 認証ポリシーを作成して、設定された Cookie 係数を表すポリシーラベルにバインドします。

```
1  add authentication Policy NO_AUTHN_POL -rule TRUE -action NO_AUTHN
2  <!--NeedCopy-->
```

このポリシーは常に true と評価され、ユーザは次の要素に移動するか、認証フローを完了します。

5. ポータルテーマ RFWebUI_Custom を NetScaler Gateway 仮想サーバーまたは NetScaler AAA 仮想サーバーにバインドします。

多要素（**nFactor**）認証を使用したサンプル展開

August 15, 2023

以下は、nFactor 認証を使用するサンプルデプロイメントです。

- 2 つのパスワードを前もって取得し、次の要素でパススルーします。Read
- グループの抽出に続いて、グループメンバーシップに基づく証明書または LDAP 認証。読み取り
- SAML の後に、SAML 中に抽出された属性に基づく LDAP または証明書認証が続きます。Read

- SAML を第 1 要素とし、次にグループを抽出し、抽出されたグループに基づいて LDAP または証明書認証を行います。読み取り
- 証明書からユーザー名を事前に入力しています。Read
- 401 対応トラフィック管理仮想サーバーの証明書認証とそれに続くグループ抽出。読み取り
- ユーザ名と 2 つのパスワード、グループ抽出の 3 番目の要因。Read
- 証明書は同じカスケードで LDAP にフォールバックします。証明書と LDAP 認証の両方に 1 つの仮想サーバーを使用します。読み取り
- 第 1 ファクタは LDAP で、第 2 ファクタは WebAuth です。読み取り
- ドメインは最初の要素でドロップダウンし、次にグループに基づいて異なるポリシー評価を行います。読み取り
- メール ID（またはユーザー名）の入力に基づくグループ抽出を最初の要素で設定して、次の要素認証フローを決定します。Read

手順の記事

August 15, 2023

認証、承認、監査「How to articles」は、シンプルで関連性があり、実装が簡単な記事です。これらの記事には、LDAP 認証や多要素認証など、一般的な認証、承認、監査機能の一部に関する情報が含まれています。NetScaler ADC による認証の構成とトラブルシューティングに関する一般的な記事の一部については、NetScaler ADC 認証: どうすればよいですか? を参照してください。

エンドポイント分析

事前認証エンドポイント分析スキャンを nFactor 認証の要素として構成する

認証後のエンドポイント分析スキャンを NetScaler nFactor 認証の要素として構成する

事前認証と認証後 EPA スキャンを nFactor 認証の要素として構成する

定期的なエンドポイント分析スキャンを nFactor 認証の要素として構成する

NetScaler Gateway によるドメインチェックの事前認証 EPA スキャンの設定

第 **1** ファクターと第 **2** ファクターの構成の組み合わせ

NetScaler Gateway の nFactor を第 1 要素に WebAuth を設定し、第 2 要素にパスワード変更を含む LDAP を設定します

nFactor 認証での SAML 属性抽出に基づいて、SAML の後に LDAP 認証または証明書認証を構成する

NetScaler nFactor 認証の第 1 要素として証明書認証を、第 2 要素として LDAP を構成する

NetScaler nFactor 認証では、1 つのログインスキーマと 1 つのパススルースキーマによる二要素認証を構成します。

nFactor 認証による 3 番目の要素のグループ抽出によるユーザー名と 2 つのパスワードの構成

最初のファクタのドメインドロップダウン、ユーザー名、およびパスワードフィールドを設定し、次のファクタのグループに基づくポリシー評価を設定します。

最初の要素で電子メール ID（またはユーザー名）入力ベースのグループ抽出を構成して、次の要素認証フローを決定します。

最初の要素でユーザー入力用のドメインドロップダウンリストを設定し、次の要素認証フローを決定します。

認証要素としての **EULA**

EULA を NetScaler nFactor システムの認証ファクターとして構成します

証明書からユーザー名を事前入力

NetScaler nFactor 認証の証明書からユーザー名を事前入力するように設定する

ステップアップ認証

ステップアップ認証など、ログインサイト要件が異なるアプリケーション用に nFactor を設定する

SAML 認証

August 15, 2023

セキュリティアサーションマークアップ言語 (SAML) は、シングルサインオン機能を提供する XML ベースの認証メカニズムで、OASIS セキュリティサービス技術委員会によって定義されています。

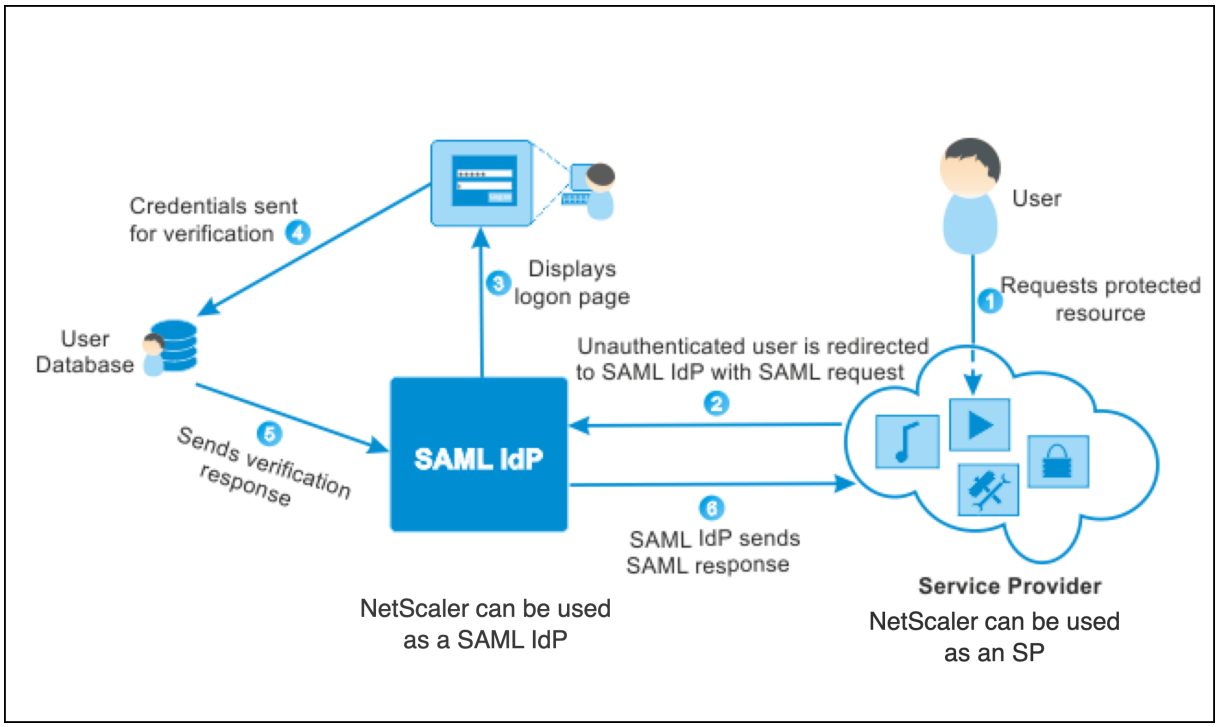
> 注
>
> NetScaler 12.0 Build 51.x 以降、多要素（nFactor）認証を備えた SAML サービスプロバイダー（SP）として使用される NetScaler アプライアンスは、ログインページのユーザー名フィールドに事前入力されるようになりました。アプライアンスは、SAML 認証リクエストの一部として NameID 属性を送信し、NetScaler SAML ID プロバイダー（IdP）から NameID 属性値を取得して、ユーザー名フィールドに事前入力します。

SAML 認証を使用する理由

サービスプロバイダー (LargeProvider) が顧客 (BigCompany) のために多数のアプリケーションをホストするシナリオを考えてみましょう。BigCompany には、これらのアプリケーションにシームレスにアクセスする必要があるユーザーがいます。従来の設定では、LargeProvider は BigCompany のユーザーのデータベースを管理する必要がありました。これにより、以下の各利害関係者にいくつかの懸念が生じます。

- LargeProvider はユーザーデータのセキュリティを確保する必要があります。
- BigCompany は、自社のデータベースだけでなく、LargeProvider が管理するユーザーデータベースでも、ユーザーを検証し、ユーザーデータを最新の状態に保つ必要があります。たとえば、BigCompany データベースから削除されたユーザーは、LargeProvider データベースからも削除する必要があります。
- ユーザーは、ホストされている各アプリケーションに個別にログオンする必要があります。

SAML 認証メカニズムは別の方法を提供します。次の導入図は、SAML の仕組み (SP 開始フロー) を示しています。



従来の認証メカニズムが抱えていた懸念は、次のように解決されます。

- LargeProvider は、BigCompany ユーザー向けにデータベースを管理する必要はありません。ID 管理から解放された LargeProvider は、より良いサービスの提供に集中できます。
- BigCompany は、LargeProvider ユーザーデータベースが自社のユーザーデータベースと同期していることを確認する責任を負いません。
- ユーザーは、LargeProvider でホストされている 1 つのアプリケーションに一度ログオンすると、そこでホストされている他のアプリケーションに自動的にログオンできます。

NetScaler アプライアンスは、SAML サービスプロバイダー（SP）および SAML アイデンティティプロバイダー

(IdP）として展開できます。関連トピックを読んで、NetScaler アプライアンスで実行する必要がある構成を理解してください。

SAML サービスプロバイダーとして構成された NetScaler アプライアンスは、対象者制限チェックを実施できるようになりました。オーディエンス制限条件は、SAML 返信者が指定されたオーディエンスの少なくとも 1 つのメンバーである場合にのみ有効と評価されます。

SAML アサーションの属性をグループ属性として解析するように NetScaler アプライアンスを構成できます。これらをグループ属性として解析すると、アプライアンスはグループにポリシーをバインドできます。

SAML サービスプロバイダーとしての NetScaler

March 20, 2024

SAML サービスプロバイダー (SP) は、サービスプロバイダーによってデプロイされる SAML エンティティです。ユーザーが保護されたアプリケーションにアクセスしようとすると、SP はクライアント要求を評価します。クライアントが認証されていない (有効な NSC_TMAA または NSC_TMAS Cookie がない) 場合、SP は要求を SAML ID プロバイダー (IdP) にリダイレクトします。

また、SP は IdP から受信した SAML アサーションも検証します。

NetScaler アプライアンスを SP として構成すると、トラフィック管理仮想サーバー（負荷分散またはコンテンツスイッチング）は、関連する SAML アクションに関連するすべてのユーザー要求を受信します。

NetScaler アプライアンスは、ログアウト時の POST バインディングとリダイレクトバインディングもサポートしています。

> 注
>
> NetScaler アプライアンスは、アプライアンスまたは任意の外部 SAML IdP 上で SAML IdP が構成されている環境では、SAML SP として使用できます。

SAML SP、NetScaler アプライアンスとして使用する場合:

- SAML トークンからユーザー情報 (属性) を抽出できます。その後、この情報を NetScaler アプライアンスで構成されたポリシーで使用できます。たとえば、GroupMember 属性と **emailaddress** 属性を抽出する場合、SAMLAction で **Attribute2** パラメーターを GroupMember として、**Attribute3** パラメーターを **emailaddress** として指定します。

  > 注
  >
  > ユーザー名、パスワード、ログアウト URL などのデフォルト属性は、属性 1 ～16 で抽出しないでください。これらは暗黙的に解析され、セッションに保存されるためです。

- 受信 SAML アサーションから最大 127 バイトの属性名を抽出できます。以前の制限は 63 バイトでした。

- ポスト、リダイレクト、アーティファクトバインディングをサポートします。

  > 注
  >
  > インフレートまたはデコード後のアサーションが 10K を超える場合は、大量のデータにリダイレクトバインディングを使用しないでください。

- アサーションを復号化できます。
- SAML アサーションから複数値の属性を抽出できます。これらの属性は、次のようなネストされた XML タグとして送信されます。

  <AttributeValue> <AttributeValue>Value1</AttributeValue>
  <AttributeValue>Value2</AttributeValue>
  \</AttributeValue\>

  > 注
  >
  > NetScaler 13.0 ビルド 63.x 以降では、SAML 属性の個々の最大長が最大 4 万バイトまで許可されるようになっています。すべての属性のサイズは 40K バイトを超えてはなりません。

  以前の XML で表示される場合、NetScaler アプライアンスは、Value1 のみを抽出する古いファームウェアとは対照的に、Value1 と Value2 の両方を特定の属性の値として抽出できます。

- SAML アサーションの有効性を指定できます。

  NetScaler SAML IdP とピア SAML SP のシステム時刻が同期していない場合、メッセージはどちらかの当事者によって無効にされる可能性があります。このような状況を回避するために、アサーションが有効な期間を設定できるようになりました。

  この期間は「スキュー時間」と呼ばれ、メッセージを受け付けることができる分数を指定します。スキュータイムは SAML SP と SAML IdP で設定できます。

- 認証リクエストで「ForceAuth」という追加属性を外部 IdP (ID プロバイダー) に送信できます。デフォルトでは、ForceAuthn は「False」に設定されています。「True」に設定すると、既存の認証コンテキストに関係なく IdP に認証を強制するよう提案できます。また、アーティファクトバインディングが設定されている場合、NetScaler SP はクエリパラメータで認証リクエストを行います。

CLI を使用して NetScaler アプライアンスを SAML SP として構成する

1. SAML SP アクションを設定します。

   例

   次のコマンドは、認証されていないユーザー要求をリダイレクトする SAML アクションを追加します。

   ```
   add authentication samlAction SamlSPAct1 -metadataUrl "https://
   ksidp1.ksaaa.local/metadata/samlidp/SAML_IDP_profile"-samlIdPCertName
   ```



```
 nssp -samlSigningCertName nssp -samlRedirectUrl https://auth1.
example.com -relaystateRule "AAA.LOGIN.RELAYSTATE.EQ(\"https://lb
.example1.com/\")"
```

注意事項

- SamLAction コマンドので-samlIdPCertName提供される証明書は、シグニチャの検証を成功させるには、IdP の対応する証明書と一致する必要があります。
- SAML は RSA 証明書のみをサポートします。HSM や FIPS などの他の証明書はサポートされていません。
- 式の末尾に「/」を付けた完全なドメイン名を使用することをお勧めします。
- 認証仮想サーバーが SAML IdP として構成されている場合、SAML SP アクションで使用する必要があるメタデータ URL はhttps://<netscaler-saml-idp-fqdn>/metadata/samlidp/SAML_IDP_profileです。
- 複数の SAML ポリシーが IdP チェーンの一部である場合は、最初の SAML ポリシーにのみリレーステートルールを設定すれば十分です。

このコマンドの詳細については、「https://developer-docs.netscaler.com/en-us/adc-command-reference-int/13-1/authentication/authentication-samlaction」と「https://support.citrix.com/article/CTX316577」を参照してください。

2. SAML ポリシーを設定します。

   例

   次のコマンドは、以前に定義した SAML アクションをすべてのトラフィックに適用する SAML ポリシーを定義します。

   ```
   add authentication policy SamlSPPol1 -rule true -action SamlSPAct1
   ```

3. SAML ポリシーを認証仮想サーバーにバインドします。

   例

   次のコマンドは、SAML ポリシーを「av_saml」という名前の認証仮想サーバーにバインドします。

   ```
   bind authentication vserver av_saml -policy SamlSPPol1
   ```

4. 認証仮想サーバーを適切なトラフィック管理仮想サーバーにバインドします。

   例:

   次のコマンドは、「lb1_ssl」という名前の負荷分散仮想サーバーを追加し、「av_saml」という名前の認証仮想サーバーを負荷分散仮想サーバーに関連付けます。

   ```
   add lb vserver lb1_ssl SSL 10.217.28.224 443 -persistenceType
    NONE -cltTimeout 180 -AuthenticationHost auth1.example.com -
   Authentication ON -authnVsName av_saml
   ```


このコマンドの詳細については、https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlActionを参照してください。

GUI を使用して NetScaler アプライアンスを SAML SP として構成する

1. [セキュリティ] > [AAA ポリシー] > [認証] > [基本ポリシー] > [SAML] に移動します。
2. [サーバ] タブを選択し、[追加] をクリックし、次のパラメータの値を入力して、[作成] をクリックします。

   パラメータの説明:

   - 名前-サーバーの名前。
   - リダイレクト URL-ユーザーが認証する URL。一部の IdP には、SAML 設定でないとアクセスできない特別な URL があります。
   - シングルログアウト URL-クライアントを IdP に送り返してサインアウト処理を完了するタイミングを NetScaler が認識できるように指定された URL。この単純な展開では使用しません。
   - SAML バインディング-SP と IdP の間で SAML リクエスタとレスポンダのメッセージを転送するために使用されるメカニズム。NetScaler が SP として機能する場合、ポスト、リダイレクト、アーティファクトのバインディングがサポートされます。デフォルトのバインド方法は POST です。

     > 注:
     >
     > Artifact バインディングでは、SP と IdP の転送メカニズムが同じである必要があります。

   - ログアウトバインディング-SAML ログアウトメッセージの転送メカニズムを指定します。デフォルトのバインドメカニズムは Post です。
   - IdP 証明書名-SAML 署名証明書の下にある IdP 証明書証明書 (Base64)
   - ユーザーフィールド- 必要に応じて SP が抽出するユーザー名を含む IdP の SAML 認証フォームのセクション。
   - 署名証明書名-NetScaler が IdP への認証リクエストに署名するために使用する SAML SP 証明書（秘密鍵付き）を選択します。IdP が認証要求署名を検証できるように、同じ証明書（秘密キーなし）を IdP にインポートする必要があります。ほとんどの IdP は署名証明書名を必要としません。
   - issuerName -識別子。SP と IdP の両方で指定される固有の ID で、サービスプロバイダーを相互に識別しやすくなります。
   - 署名されていないアサーションを拒否-IdP からのアサーションに署名が必要な場合に指定できるオプション。デフォルトのオプションは［オン］です。
     - ON: 署名のないアサーションを拒否します
     - STRICT: レスポンスとアサーションの両方が署名されていることを確認します
     - OFF: 署名なしのアサーションを許可します

- Audience-IdP によって送信されたアサーションが適用可能なオーディエンス。これは通常、サービスプロバイダーを表すエンティティ名または URL です。
- 署名アルゴリズム-SAML トランザクションの署名/検証に使用されるアルゴリズム。デフォルト値は RSA-SHA256 です。
- ダイジェスト方式-SAML トランザクションのダイジェストの計算/検証に使用されるアルゴリズム。デフォルト値は SHA256 です。
- リレーステートルール-このパラメータの詳細については、「relaystateRule パラメータに関する注意点」を参照してください。
- 状態チェックルール-SAML エンドポイントの HTTP リクエストを検証するために評価される式を設定します。
- デフォルト認証グループ-抽出されたグループに加えて、認証が成功した場合に選択されるデフォルトグループ。
- Group Name Field：ユーザグループを含むアサーション内のタグの名前。
- スキュー時間（分）-このオプションでは、NetScaler サービスプロバイダーが受信アサーションで許容するクロックスキューを分単位で指定します。たとえば、16:00 にスキュー時間を 10 分に設定した場合、SAML アサーションは 15:50 から 16:10（合計 20 分）まで有効です。デフォルトのスキュー時間は 5 分です。
- 2 要素-SAML 後の第 2 要素認証を有効にします。
- アサーションコンシューマサービスインデックス-この設定に対応するメタデータエントリのインデックス/ID。
- 属性消費サービスインデックス-IdP での属性仕様のインデックス/ID。IdP は、このインデックスを使用して SP によって要求された属性を検索し、それらの属性を SAML アサーションで送信します。
- リクエストされた認証コンテキスト- 応答で返される認証ステートメントのコンテキスト要件を指定します。
- 認証クラスタイプ-IdP から要求される認証クラスタイプを指定します。
- カスタム認証クラスタイプ-SP から SAML IdP に送信される認証リクエストの一部として送信されるカスタム認証クラス参照を指定します。
- サムプリントを送信-SAML リクエストの証明書の代わりにサムプリントを送信します。
- ユーザー名を強制-2 要素認証の実行中に、SAML アサーションから抽出されたユーザー名をログインページで編集できるかどうかを選択します。
- 強制認証-NetScaler からリクエストを受信した IdP で認証を強制します。
- SAML レスポンスを保存-ユーザーセッションがアクティブな限り、SAML レスポンス全体を保存します。

3. 対応する SAML ポリシーを作成します。

   [セキュリティ] **>** **[AAA** アプリケーショントラフィック] **>** **[**ポリシー] **>** **[**認証] **>** **[**詳細ポリシー] **>** **[**ポリシー] に移動し、[追加] をクリックします。

   「認証 **SAML** ポリシーの作成」ページで、次の詳細を入力します。

   - 名前-SAML ポリシーの名前を指定します。
   - アクションタイプ-認証アクションタイプとして **SAML** を選択します。
   - アクション-SAML ポリシーをバインドする SAML サーバープロファイルを選択します。
   - 表現-ユーザーが SAML サーバーで認証する必要があるかどうかを判断するために SAML ポリシーが使用するルールまたは式の名前を表示します。テキストボックスに「rule = true」という値を設定すると、SAML ポリシーが有効になり、対応する SAML アクションが実行されます。

4. SAML ポリシーを認証仮想サーバーにバインドします。

   [セキュリティ] **>** **[AAA-**アプリケーショントラフィック] **>** **[**仮想サーバー] に移動し、SAML ポリシーを認証仮想サーバーに関連付けます。

5. 認証サーバーを適切なトラフィック管理仮想サーバーに関連付けます。

   [トラフィック管理] **>** **[**負荷分散]（または [コンテンツスイッチング]）> [仮想サーバー] に移動し、仮想サーバーを選択し、認証仮想サーバーをそれに関連付けます。

`relaystateRule` パラメータに関する注意点

SAML アサーション後の認証リダイレクトをスムーズに行うには、`samlAction` コマンドの **RelaysStateRule** パラメーターの式を設定します。式には、認証リダイレクトの前に、単一の公開ドメイン、またはユーザーが接続したい公開ドメインのリストを含める必要があります。たとえば、式には、認証に SAML アクションを使用するフロントエンド仮想サーバー (VPN、負荷分散、またはコンテンツスイッチング) のドメインが含まれている必要があります。

末尾に (/) が付いた完全なドメイン名を使用することをお勧めします。例: `https://example.com/`。

次の例では、単一ドメインと複数ドメインのシナリオでの RelayStateRule パラメーターの設定例について説明します:

単一ドメインの場合:

```
set samlAction <samlActionName> -relaystateRule "AAA.LOGIN.RELAYSTATE
.EQ(\"https://example1.com/\")"
```

複数のドメインの場合:

パターンセットを使用して複数のドメインを追加します:

```
1 add patset test1
2
```



```
bind patset test1 "https://example1.com/"

bind patset test1 "https://test1.com/"

bind patset test1 "https://10.11.11.112/"

set samlAction <samlActionName> -relaystateRule AAA.LOGIN.RELAYSTATE.
   CONTAINS_ANY("test1")
<!--NeedCopy-->
```

FQDN の RegEx パターンマッチングを設定するには:

```
set samlaction <samlActionName> -relaystateRule "AAA.LOGIN.RELAYSTATE.
   REGEX_MATCH(re#^https://[a-zA-Z0-9]*.example1.com/#)"
<!--NeedCopy-->
```

ドメインの先頭に「^」記号 (^https) を指定し、式の最後にフォワードスラッシュ「/」を指定します (例: example1\.com/)。

RelayStateRule エクスプレッションの定義の一部を次に示します:

```
EQ: Exact match.
CONTAINS: Domain inclusion.
CONTAINS_ANY: Match any domain from a PATSET.
REGEX_MATCH: Regex-based matching.
<!--NeedCopy-->
```

SAML IdP としての NetScaler

October 25, 2023

SAML IdP (ID プロバイダー) は、顧客ネットワークにデプロイされる SAML エンティティです。IdP は SAML SP からリクエストを受信し、認証情報を入力する必要があるログオンページにユーザーをリダイレクトします。IdP は、Active Directory（LDAP などの外部認証サーバ）でこれらの認証情報を認証し、SP に送信される SAML アサーションを生成します。

SP はトークンを検証し、要求された保護されたアプリケーションへのアクセス権をユーザーに付与します。

NetScaler アプライアンスが IdP として構成されている場合、すべての要求は、関連する SAML IdP プロファイルに関連付けられている認証仮想サーバーによって受信されます。

> 注
>
> NetScaler アプライアンスは、SAML SP がアプライアンスまたは任意の外部 SAML SP で構成されている展開環境で、IdP として使用できます。

SAML IdP、NetScaler アプライアンスとして使用する場合:

- 従来のログオンでサポートされるすべての認証方法をサポートします。
- アサーションにデジタル署名します。
- 単一要素認証と 2 要素認証をサポートします。SAML をセカンダリ認証メカニズムとして設定しないでください。
- SAML SP のパブリックキーを使用してアサーションを暗号化できます。これは、アサーションに機密情報が含まれている場合に推奨されます。
- SAML SP からのデジタル署名された要求のみを受け入れるように設定できます。
- ネゴシエート、NTLM、証明書の 401 ベースの認証メカニズムを使用して SAML IdP にログオンできます。
- nameID 属性のほかに 16 個の属性を送信するように設定できます。属性は適切な認証サーバから抽出する必要があります。それぞれについて、SAML IdP プロファイルで名前、式、形式、およびフレンドリ名を指定できます。
- NetScaler アプライアンスが複数の SAML SP 用の SAML IdP として構成されている場合、ユーザーは毎回明示的に認証しなくても、さまざまな SP 上のアプリケーションにアクセスできます。NetScaler アプライアンスは最初の認証用にセッション Cookie を作成し、それ以降のリクエストではすべてこの Cookie を認証に使用します。
- SAML アサーションで複数値属性を送信できます。
- ポストバインディングとリダイレクトバインディングをサポートします。アーティファクトバインディングのサポートは、NetScaler リリース 13.0 ビルド 36.27 で導入されました。
- SAML アサーションの有効性を指定できます。

  NetScaler SAML IdP とピア SAML SP のシステム時刻が同期していない場合、メッセージはどちらかの当事者によって無効にされる可能性があります。このような状況を回避するために、アサーションが有効な期間を設定できるようになりました。

  この時間を「スキュータイム」と呼び、メッセージを受け入れる必要がある分数を指定します。スキュータイムは SAML SP と SAML IdP で設定できます。
- IdP で事前設定されている、または IdP によって信頼されている SAML SP に対してのみアサーションを提供するように設定できます。この設定では、SAML IdP には、関連する SAML SP のサービスプロバイダー ID（または発行者名）が必要です。

  > 注
  >
  > - 先に進む前に、LDAP 認証仮想サーバーにバインドされた認証ポリシーがあることを確認してください。
  > - 必要な属性を取得するように LDAP アクションを設定する方法の詳細については、「LDAP 認証での名前と値の属性のサポート」を参照してください。

CLI を使用して NetScaler アプライアンスを SAML IdP として構成する

1. SAML IdP プロファイルを作成します。

   例

   SiteMinder を SP として、NetScaler アプライアンスを IdP として追加します。

   ```
   add authentication samlIdPProfile samlIDPProf1 -samlSPCertName
   siteminder-cert -encryptAssertion ON -metadataUrl https://samlidp
   .example.com/metadata -samlIdPCertName ns-cert -assertionConsumerServiceUR
    https://example.com/cgi/samlauth -rejectUnsignedRequests ON
   -signatureAlg RSA-SHA256 -digestMethod SHA256 -acsUrlRule AAA.
   LOGIN.SAML_REQ_ACS_URL.REGEX_MATCH(re#^https://example\.com/cgi/
   samlauth$#)
   ```

2. SAML IdP プロファイルを設定します。次の例では、IdP セッションには「UserPrincipalName」属性が含まれています。

   ```
   set samlidPProfile SAML-IDP-Profile -Attribute1 "userPrincipalName
   "-Attribute1Expr "AAA.USER.ATTRIBUTE(\"userPrincipalName\")"
   ```

   注意事項

   - SAML IdP プロファイルで、この IdP に適用可能なサービスプロバイダー URL のリストの式をとる **acsurlRule** を設定します。この式は、使用している SP によって異なります。NetScaler が SP として構成されている場合、ACS URL は`https://<SP-domain_name>/cgi/samlauth`です。一致させるには、式に完全な URL を含めることをお勧めします。
   - SAML IdP で ACS URL を 1 つだけ許可する場合は、次のコマンドを使用します。

     次の CLI 例では`https://testlb.aaa.local`を ACS URL として使用しています。

     ```
     set samlidpprofile SAML_IDP_profile -acsurlrule  "AAA.LOGIN.
        SAML_REQ_ACS_URL.eq("https://testlb.aaa.local")"
     <!--NeedCopy-->
     ```

   - SAML IdP に ACS URL を正規表現と一致させたい場合は、次の表現を使用してください。

     ```
     -acsUrlRule AAA.LOGIN.SAML_REQ_ACS_URL.REGEX_MATCH(re#^https
     ://example.com/cgi/samlauth$#)
     ```

     上記の式により、ACS URL が`https://example.com/cgi/samlauth`と一致することが保証されます。正規表現の先頭に「^」記号が付いていると、NetScaler では「https」より前は何も許可されません。正規表現の末尾に「$」記号が付いていると、NetScaler では「samlauth」の後に何も許可されないようになります。

     式　　が`-acsUrlRule AAA.LOGIN.SAML_REQ_ACS_URL.REGEX_MATCH(re#https://example.com/cgi/#)`の場合、SAML IdP は次の例に示すように任意の ACS URL

を許可します。

- https://example.com/cgi/samlauth
- abcdhttps://example.com/cgi/xyz
- https://example.com/cgi/abcde

- SAML は RSA 証明書のみをサポートします。HSM、FIPS などの他の証明書はサポートされていません。

コマンドの詳細については、「https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction」および「https://support.citrix.com/article/CTX316577」を参照してください。

- IdP ログアウト URL がリダイレクト URL と異なり、ユーザーが NetScaler ログインページに 2 分以上いる場合、ユーザーが認証を試みるとサーバーエラーHTTP/1.1 Internal Server Error 43549が表示されます。NetScaler のログには、受信したログアウト後のリダイレクト URL がユーザーの許可リストに登録されているログアウトリダイレクト URL に含まれていないことを示すメッセージが表示されます。

  この問題を解決するには、以下の例のようにパターンセットをバインドします。

  ```
  bind patset ns_aaa_oauthidp_logout_redirect_uris "https://
  FQDN and path to the logout url"
  ```

- 認証仮想サーバーが SAML SP として構成されている場合、SAML IdP プロファイルで使用する必要があるメタデータ URL は https://<netscaler-saml-sp-fqdn>/metadata/samlsp/saml_sp_actです。例:

  ```
  add authentication samlIdPProfile SAML_IDP_profile -samlIdPCertName
   aaa_local -assertionConsumerServiceURL "https://ksav.ksaaa.
  local/cgi/samlauth"-samlIssuerName "https://ksidp1.aaa.local
  /saml/login"-rejectUnsignedRequests OFF -serviceProviderID
   kslb.ksaaa.local -signAssertion NONE -SPLogoutUrl "https
  ://ksav.ksaaa.local/cgi/tmlogout"-logoutBinding REDIRECT
   -metadataUrl "https://ksav.ksaaa.local/metadata/samlsp/
  saml_sp_act"-metadataRefreshInterval 1
  ```

3. SAML 認証ポリシーを設定し、SAML IdP プロファイルをポリシーのアクションとして関連付けます。

   ```
   add authentication samlIdPPolicy samlIDPPol1 -rule true -action
   samlIDPProf1
   ```

   > 注:

> ポリシー名に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます (たとえば、「my policy」や「my policy」)。

4. 認証仮想サーバにポリシーをバインドします。

```
bind authentication vserver saml-auth-vserver -policy samlIDPPol1
 -priority 100
```

コマンドの詳細については、https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlIdPProfileを参照してください。

GUI を使用して NetScaler アプライアンスを SAML IdP として構成する

1. SAML IdP プロファイルを設定します。このプロファイルは、SP からの受信認証リクエストを検証し、SP に送信する前にアサーションを作成して署名するために使用されます。

   [セキュリティ] **> [AAA-**アプリケーショントラフィック] **> [**ポリシー] **> [**認証詳細ポリシー] **> [SAML IDP** ポリシー] に移動します。

   [サーバー] を選択し、[追加] をクリックし、次のパラメーターの値を入力して、[作成] をクリックします。

   パラメータの説明**:**

   - 名前-新しい SAML シングルサインオンプロファイルの名前。
   - SAML IDP メタデータのエクスポート-SAML IdP プロファイルのメタデータを NetScaler Gateway VPN 仮想サーバーにエクスポートする場合は、このリンクをクリックします。
   - メタデータのインポート-このオプションは SAML IdP メタデータをインポートします。このオプションは、デフォルトで有効になっています。
   - アサーションコンシューマーサービス URL-アサーションの送信先の URL。
   - サービスプロバイダログアウト URL-ログアウトメッセージの送信先となる SP エンドポイント。
   - ログアウトバインディング—SAML ログアウトメッセージの転送メカニズムを指定します。使用可能なオプションは POST とリダイレクトです。
   - SAML SP メタデータ URL —SAML IdP メタデータを取得するために使用される URL。

     > 注**:**
     >
     > SAML SP メタデータ URL を設定すると、次のパラメータが SAML IdP プロファイルから取得され、SAML SP 設定に自動的に入力されます。
     >
     > - アサーションコンシューマーサービス URL
     > - サービスプロバイダログアウト URL
     > - SP 証明書名
     > - ログアウトバインディング

  - SAML バインディング
  - 署名アサーション

- メタデータの更新間隔 (分)-指定されたメタデータ URL からメタデータを取得する間隔 (分単位)。デフォルトの時間間隔は 3600 分です。
- アサーションコンシューマーサービス URL ルール-SAML SP からの許可される ACS URL を定義する式。つまり、SAML 要求に不正な ACS URL を挿入する攻撃を防ぐために、ACS URL を許可リストに登録します。
- アサーションコンシューマーサービス URL-認証されたユーザーがリダイレクトされる URL。
- IdP 証明書名-認証ページに使用される証明書とキーのペア。
- SP 証明書名-サービスプロバイダの証明書このシナリオでは、キーは必要ありません。
- Sign Assertion- クライアントをサービスプロバイダにリダイレクトするときに、アサーションと応答に署名するオプション。
- 発行者名-IdP によって発行された SAML アサーションに含まれる文字列値。
- サービスプロバイダー ID：サービスプロバイダーの識別に役立つように SP 上で指定される一意の ID。ID は何でもかまいませんが、必ずしも URL である必要はありません。ただし、ID は SP プロファイルと IdP プロファイルの両方で同じでなければなりません。
- デフォルト認証グループ-抽出されたグループに加えて、認証が成功したときに選択されるデフォルトグループ。このグループは、管理者が nFactor フローを使用して中継側に適した構成を決定する場合に便利です。たとえば、認証ポリシーを設定する場合、次の式の一部としてデフォルトのグループ名を指定できます。

  ```
  AAA.USER.IS_MEMBER_OF("Default Authentication Group name").
  ```

- 署名されていないリクエストを拒否-SP 証明書で署名されたアサーションのみが受け付けられるように指定できるオプション。
- オーディエンス-IdP によってアサーションが送信されるオーディエンス。これは通常、SP を表すエンティティ名または URL です。
- スキュー時間（分）-スキュー時間（分）-このオプションは、NetScaler サービスプロバイダーが受信アサーションで許容するクロックスキューを分単位で指定します。たとえば、16:00 にスキュー時間を 10 分に設定した場合、SAML アサーションは 15:50 から 16:10（合計 20 分）まで有効です。デフォルトのスキュー時間は 5 分です。
- NAME ID 形式-アサーションで送信される名前識別子の形式。
- 名前 ID 式-アサーションで送信される名前識別子を取得するために評価される式。
- アサーションへの署名-IdP から送信されたアサーションの一部に署名するオプション。使用可能なオプションは、[なし]、[アサーション]、[応答]、または [両方] です。

- Signature Algorithm：IdP と SP の間のアサーションの署名と検証に使用されるアルゴリズム。IdP プロファイルと SP プロファイルで同じである必要があります。
- Digest Method：IdP と SP の間のアサーションの整合性を検証するために使用されるアルゴリズム。IdP プロファイルと SP プロファイルで同じである必要があります。
- SAML バインディング-SP と IdP の間で SAML リクエスタとレスポンダのメッセージを転送するために使用されるメカニズム。NetScaler が SP として機能する場合、ポスト、リダイレクト、アーティファクトのバインディングがサポートされます。デフォルトのバインド方法は POST です。SAML IdP ポリシーを認証仮想サーバーに関連付けます。Artifact バインディングでは、SP と IdP のトランスポートメカニズムが同じである必要があります。
- 属性 1-値を抽出して属性 1 として保存する必要がある SAML アサーション内の属性の名前。同様のパターンが残りの属性にも当てはまります。
- 属性 1Expr-評価されて属性 1 の値を取得する式。
- Attribute1FriendlyName-SAML アサーションで送信する必要がある属性 1 の名前。
- 属性 1 形式-SAML アサーションで送信される属性 1 のフォーマット。

2. SAML 認証ポリシーを設定し、SAML IdP プロファイルをポリシーのアクションとして関連付けます。

   [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**認証詳細ポリシー**] > [SAML IDP** ポリシー] に移動します。

   [ポリシー] を選択し、[追加] をクリックし、次のパラメータの値を入力して、[作成] をクリックします。

   パラメータの説明**:**

   - 名前-SAML IdP 認証ポリシーの名前。
   - アクション-このポリシーに一致するリクエストまたは接続に適用する SAML IdP プロファイルの名前。
   - ログアクション -リクエストがこのポリシーに一致する場合に使用するメッセージログアクションの名前。ドロップダウンリストからログアクションを選択するか、[追加] をクリックしてログアクションを作成します。
   - 未定義結果アクション-ポリシー評価の結果が未定義の場合に実行するアクション。未定義のイベントは内部エラー状態を示します。ビルトインアクションのみを使用できます。
   - コメント -このポリシーに関する情報を保存するための任意のコメント。

3. SAML IdP ポリシーを認証仮想サーバーに関連付けます。

   [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバー] に移動し、SAML IdP ポリシーを認証仮想サーバーにバインドします。

SAML シングルサインオンの設定

December 8, 2023

サービスプロバイダーでホストされているアプリケーション間でシングルサインオン機能を提供するには、SAML SP で SAML シングルサインオンを構成します。

コマンドラインインターフェイスを使用した SAML シングルサインオンの設定

1. SAML SSO プロファイルを設定します。

   例

   次のコマンドでは、SharePoint ポータルからの Web リンクを持つ負荷分散仮想サーバーの例を示します 。Nssp.example.com は、SharePoint サーバーの負荷分散を行うトラフィック管理の仮想サーバーです。

   ```
   add tm samlSSOProfile tm-saml-sso -samlSigningCertName nssp -
       assertionConsumerServiceURL "https://nssp2.example.com/cgi/
       samlauth" -relaystateRule "\\"https://nssp2.example.com/
       samlsso.html\\"" -sendPassword ON -samlIssuerName nssp.example
       .com
   <!--NeedCopy-->
   ```

2. SAML SSO プロファイルをトラフィックアクションに関連付けます。

   例

   次のコマンドは、SSO を有効にし、上記で作成した SAML SSO プロファイルをトラフィックアクションにバインドします。

   ```
   add tm trafficAction html_act -SSO ON -samlSSOProfile tm-saml-sso
   <!--NeedCopy-->
   ```

3. アクションをいつ実行する必要があるかを指定するトラフィックポリシーを構成します。

   例

   次のコマンドは、トラフィックアクションをトラフィックポリシーに関連付けます。

   ```
   add tm trafficPolicy html_pol "HTTP.REQ.URL.CONTAINS(\\"abc.html\\
       ")" html_act
   <!--NeedCopy-->
   ```

4. 以前に作成したトラフィックポリシーをトラフィック管理仮想サーバ（負荷分散またはコンテンツスイッチング）にバインドします。また、トラフィックポリシーをグローバルに関連付けることもできます。

> 注
>
> このトラフィック管理仮想サーバーは、SAML アクションに関連付けられた関連する認証仮想サーバーに関連付ける必要があります。

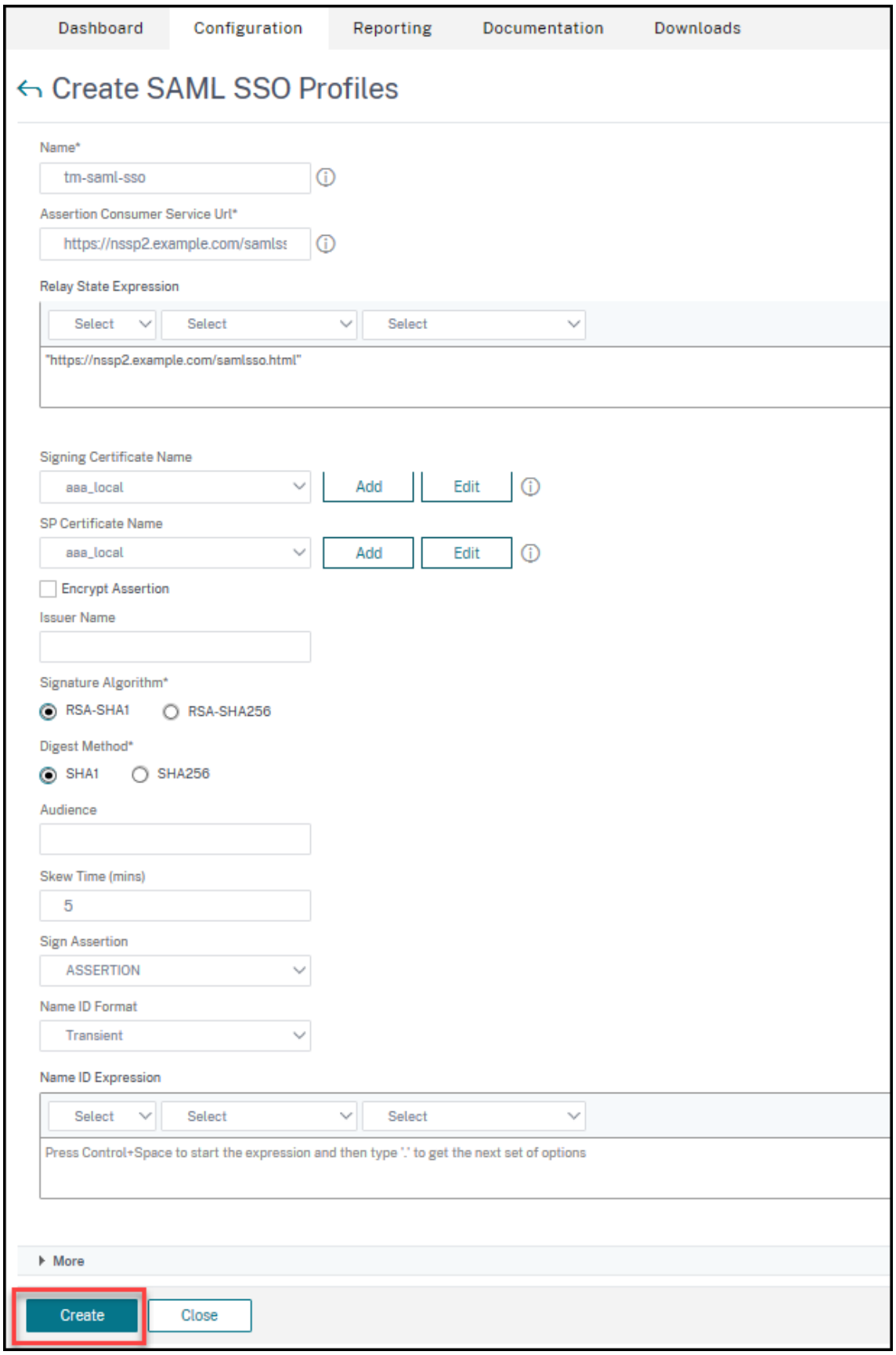
```
bind lb vserver lb1_ssl -policyName html_pol -priority 100 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

GUI を使用した SAML シングルサインオンの設定

SAML シングルサインオンを構成するには、SAML SSO プロファイル、トラフィックプロファイル、およびトラフィックポリシーを定義し、トラフィックポリシーをトラフィック管理仮想サーバーに、または NetScaler アプライアンスにグローバルにバインドする必要があります。

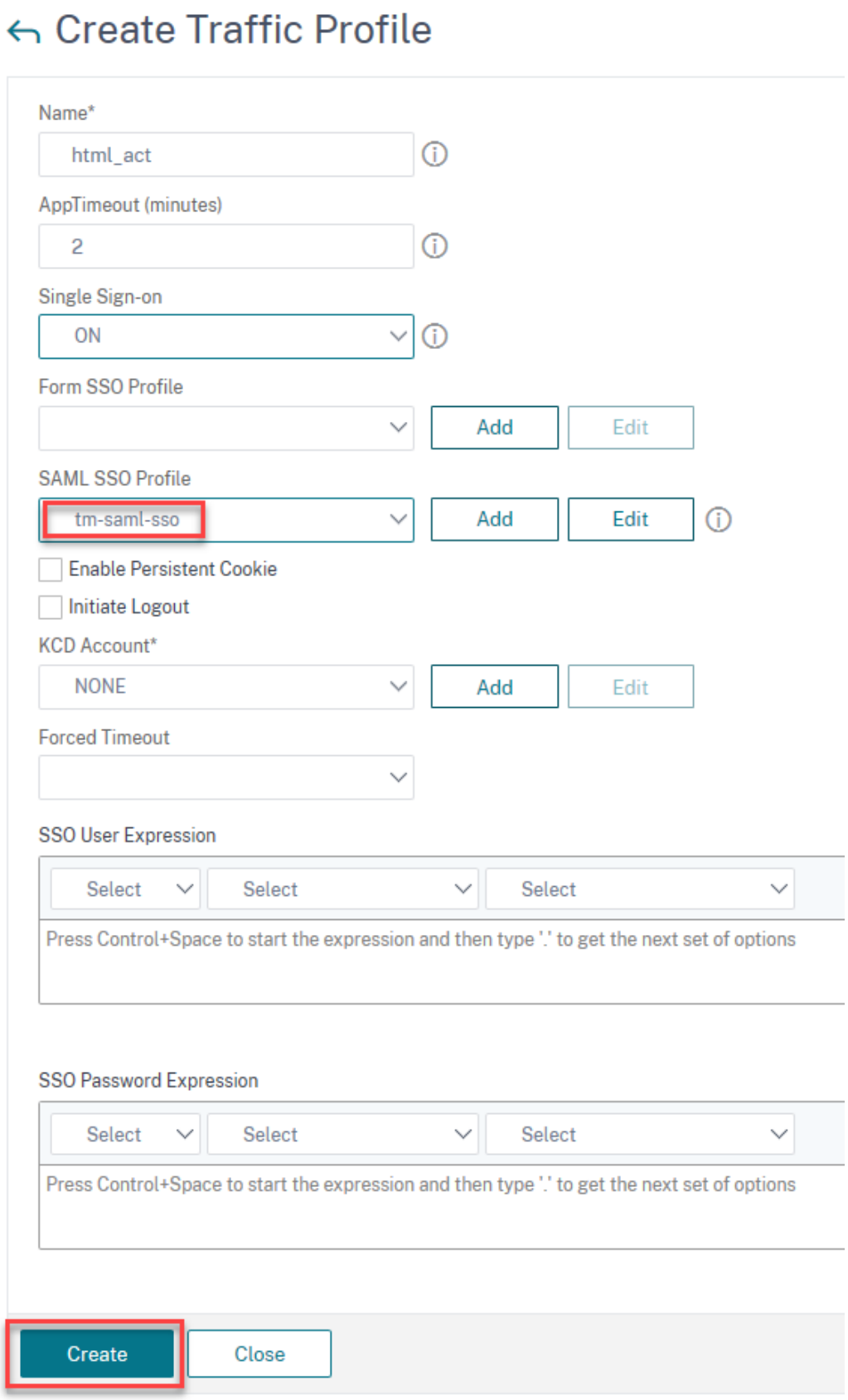
1. セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** トラフィック **> SAML SSO** プロファイルに移動し、[追加] をクリックします。
2. [**SAML SSO** プロファイルの作成] ページで、次のフィールドに値を入力し、[作成] をクリックします。

   - Name-SAML SSO プロファイルの名前
   - アサーションコンシューマサービス URL-アサーションが送信される URL
   - 署名証明書名-アサーションの署名に使用される SSL 証明書の名前
   - SP 証明書名：アサーションが暗号化されるピア/受信側の SSL 証明書の名前
   - 発行者名-NetScaler から IdP に送信されるリクエストで NetScaler を一意に識別するために使用される名前
   - 署名アルゴリズム-SAML トランザクションの署名/検証に使用するアルゴリズム
   - Digest Method-SAML トランザクションのダイジェストを計算/検証するために使用されるアルゴリズム
   - Audience-IdP によって送信されたアサーションが適用可能なオーディエンス。これは通常、サービスプロバイダを表すエンティティ名または URL です。
   - Audience-IdP によって送信されたアサーションが適用可能なオーディエンス。これは通常、サービスプロバイダを表すエンティティ名または URL です。
   - スキュー時間 (分)-アサーションが有効になる現在の時刻の両側の分数
   - アサーションへの署名-NetScaler IdP がアサーションを送信するときにアサーションの一部に署名するオプション。ユーザーの選択に基づいて、[アサーション] または [応答]、または [両方] または [なし] のいずれかに署名できます。
   - 名前 ID 形式-アサーションで送信される名前識別子の形式
   - 名前 ID 式-アサーションで送信される NameIdentifier を取得するために評価される式

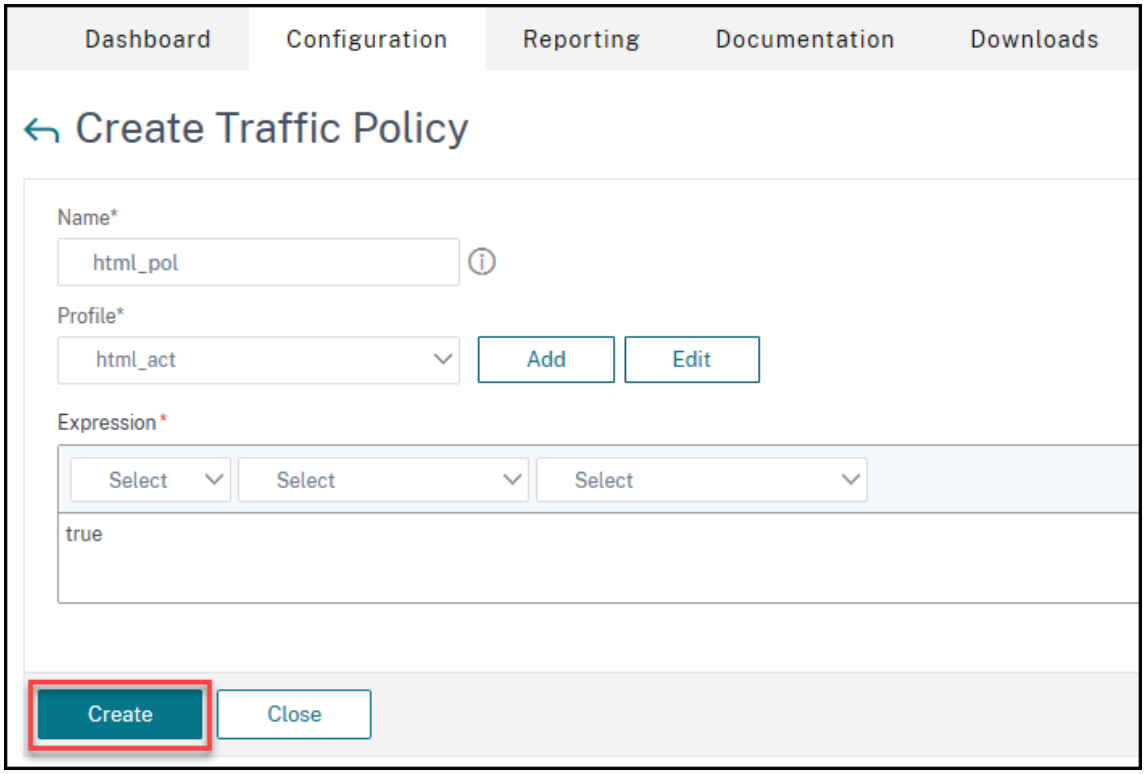


3. セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** トラフィック **>** トラフィックプロファイルに移動し、[追加] をクリックします。
4. [トラフィックプロファイルの作成] ページで、次のフィールドに値を入力し、[作成] をクリックします。

    - Name-トラフィックアクションの名前。
    - AppTimeout (分)-接続が閉じられるまでのユーザーの非アクティブ時間の間隔（分単位）。
    - シングルサインオン-[オン] を選択します。
    - SAML SSO プロファイル-作成した SAML SSO プロファイルを選択します。
    - KCD アカウント-Kerberos 制約付き委任アカウント名
    - SSO ユーザー式-SingleSignOn のユーザー名を取得するために評価される式
    - SSO パスワード式-SingleSignOn のパスワードを取得するために評価される式



5. セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** トラフィック **>** トラフィックポリシーに移動し、[追加] をクリックします。
6. [トラフィックポリシーの作成] ページで、次の値を入力し、[作成] をクリックします。
   - Name —作成するトラフィックポリシーの名前
   - Profile —作成したトラフィックプロファイルを選択します。
   - 式—ポリシーが特定の要求に応答するために使用する高度なポリシー表現。例：true。



7. トラフィックポリシーをトラフィック管理仮想サーバーにバインドするには、[構成] **> [**トラフィック管理] **> [**負荷分散] **> [**仮想サーバー] に移動し、仮想サーバーを選択します。
8. 「詳細設定」で、「ポリシー」をクリックします。
9. [ポリシーの選択] フィールドで [トラフィック] を選択し、[タイプの選択] フィールドで **[**** 要求] を選択し ****** 、[続行] をクリックします。
10. 「**Select Policy** 」フィールドで、作成したトラフィックをクリックして選択します。
11. [**Select**] をクリックします。
12. [バインド] をクリックして、トラフィックポリシーを仮想サーバーにバインドします。

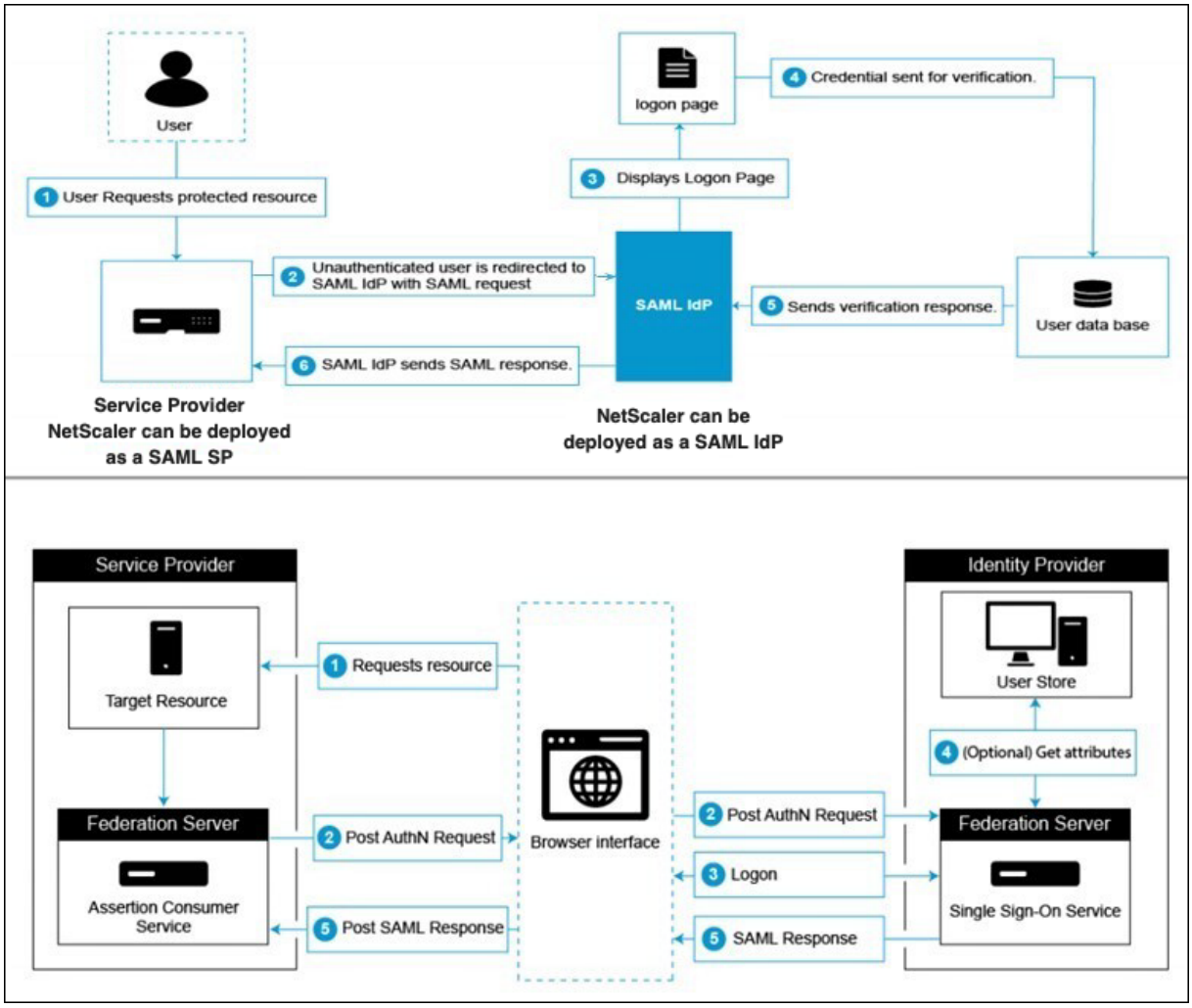
Azure AD を **SAML ID** プロバイダーとして、**NetScaler** を **SAML** サービスプロバイダーとして構成する

December 8, 2023

SAML サービスプロバイダー (SAML SP) は、サービスプロバイダーによってデプロイされる SAML エンティティです。ユーザーが保護されたアプリケーションにアクセスしようとすると、SP はクライアント要求を評価します。クライアントが認証されていない (有効な NSC_TMAA または NSC_TMAS Cookie がない) 場合、SP は要求を SAML ID プロバイダー (IdP) にリダイレクトします。また、SP は IdP から受信した SAML アサーションも検証します。

SAML アイデンティティプロバイダー (SAML IdP) は、顧客ネットワークにデプロイされる SAML エンティティです。IdP は SAML SP からリクエストを受信し、認証情報を入力する必要があるログオンページにユーザーをリダイレクトします。IdP は、これらの認証情報をユーザディレクトリ（LDAP などの外部認証サーバ）で認証し、SP に送信される SAML アサーションを生成します。SP はトークンを検証し、要求された保護されたアプリケーションへのアクセス権をユーザーに付与します。

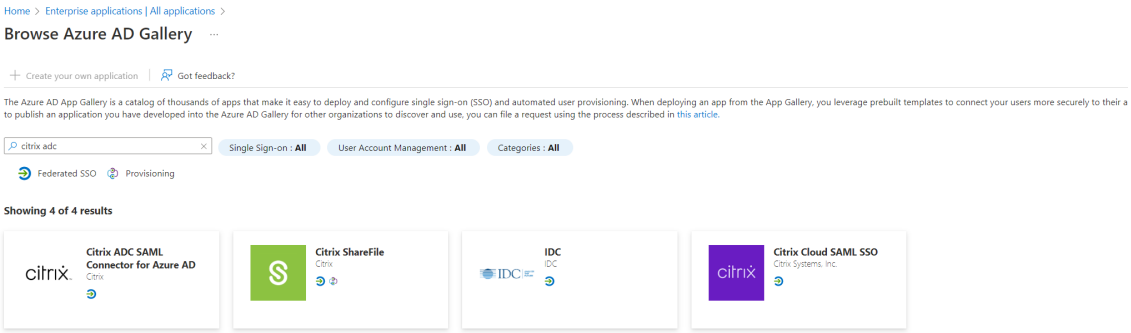
次の図は、SAML 認証メカニズムを示しています。



Azure AD サイド構成

シングルサインオン設定を構成します。

1. Azure ポータルで、[**Azure Active Directory**] をクリックします。
2. ナビゲーションペインの「管理」セクションで、「エンタープライズアプリケーション」をクリックします。Azure AD テナント内のアプリケーションのランダムなサンプルが表示されます。
3. 検索バーに「**Azure AD** 用 **NetScaler SAML** コネクタ」と入力します。



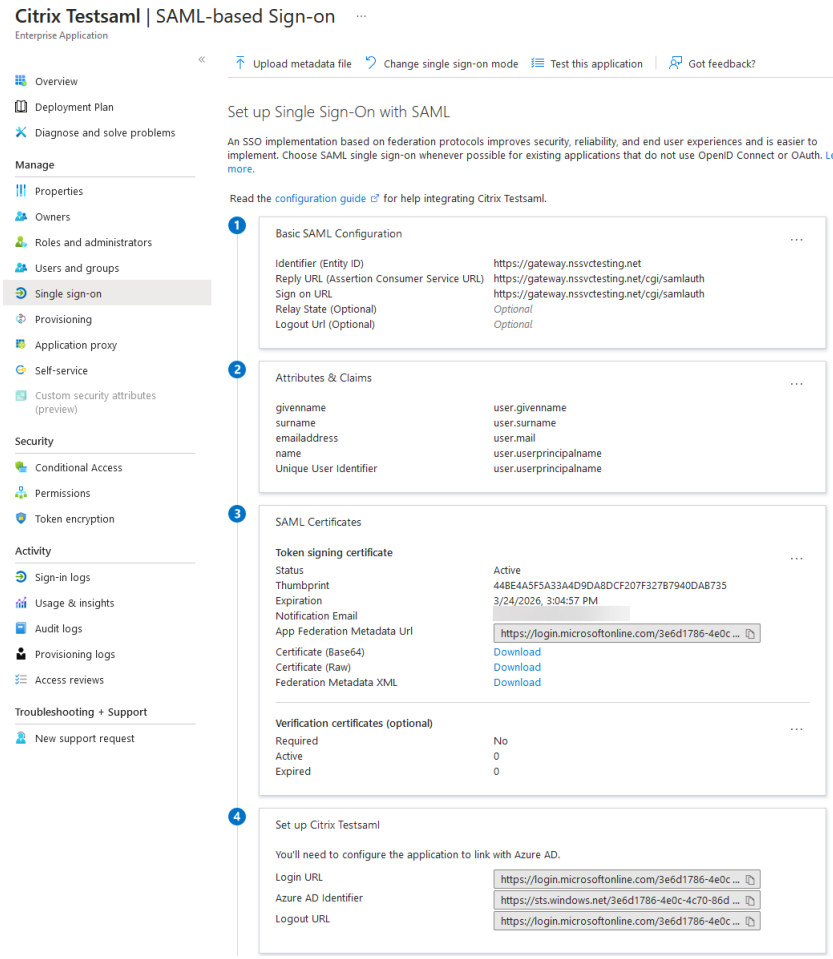
4. [管理] セクションで、[シングルサインオン] を選択します。
5. [**SAML**] を選択して、シングルサインオンを構成します。[**SAML** でシングルサインオンを設定する**-**プレビュー] ページが表示されます。ここで、Azure は SAML IdP として機能しています。
6. 基本的な **SAML** オプションを設定します。

   識別子 **(**エンティティ **ID)** -一部のアプリで必須です。シングルサインオンが設定されているアプリケーションを一意に識別します。Azure AD は、SAML トークンのオーディエンスパラメータとして識別子をアプリケーションに送信します。アプリケーションは、それを検証することが期待されます。この値は、アプリケーションによって提供されるすべての SAML メタデータにもエンティティ ID として表示されます。

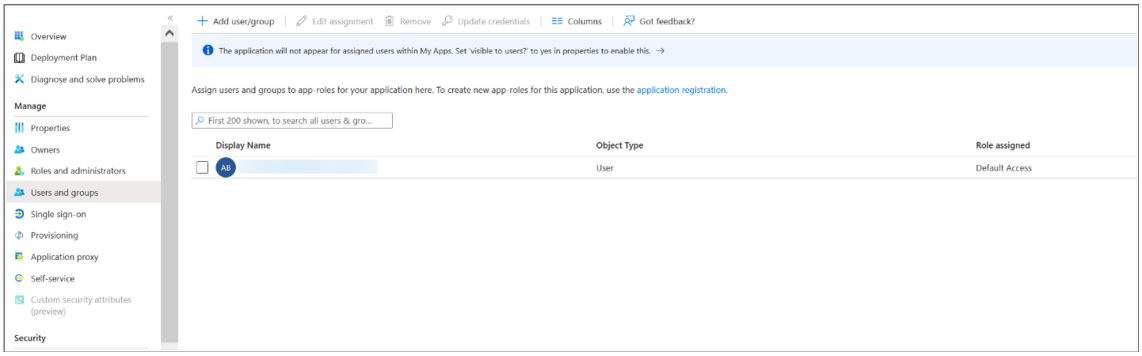
   返信 **URL** -必須。アプリケーションが SAML トークンの受信を期待する場所を指定します。応答 URL は、アサーションコンシューマサービス（ACS）URL とも呼ばれます。返信 URL をフォーマット`http(s)://<SP_URL>/cgi/samlauth`で指定します。

   サインオン **URL** -ユーザーがこの URL を開くと、サービスプロバイダーは Azure AD にリダイレクトしてユーザーを認証し、サインインします。

   Relay State： 認証の完了後にユーザーをリダイレクトする場所をアプリケーションに指定します。
7. 「**SAML** 証明書」セクションから証明書 (Base64) をダウンロードします。この証明書は、NetScaler を SAML SP として構成するときに SAMLIDPCertName として使用されます。



8. Azure AD 側の構成が完了したら、アプリケーションへのアクセスを許可されているユーザーとユーザーグループを追加します。「ユーザーとグループ」タブに移動し、「**+** ユーザー/グループを追加」をクリックします。



NetScaler 側の構成

1. SAML アクションを作成します。

   - [セキュリティ] **> [AAA** アプリケーショントラフィックポリシー] **> [**認証] **> [**詳細ポリシー] **> [**アクション] **> [SAML**] に移動します。

- 「追加」をクリックし、次のパラメータの値を入力して、「作成」をクリックします。

  パラメータの説明:

  太字のパラメーターの値は、Azure 側の構成から取得する必要があります。

  - Name: サーバーの名前
  - リダイレクト **URL** -以前に Azure AD の「NetScaler のセットアップ」セクションで使用したログイン URL を入力します。https://login.microsoftonline.com/3e6d1786-4e0c-4c70-86d2-ae7811f97f79/saml2
  - シングルログアウト URL- https://login.microsoftonline.com/3e6d1786-4e0c-4c70-86d2-ae7811f97f79/saml2
  - SAML バインディング-SP と IdP の間で SAML リクエスタとレスポンダのメッセージを転送するために使用されるメカニズム。NetScaler が SP として機能する場合、ポスト、リダイレクト、アーティファクトのバインディングがサポートされます。デフォルトのバインド方法は Post です。
  - ログアウトバインディング-SAML ログアウトメッセージの転送メカニズムを指定します。デフォルトのバインドメカニズムは Post です。
  - **IDP** 証明書名**-SAML** 証明書セクションにある **IDPCert** 証明書 **(Base64)**。

    ```
    1  add ssl certkey <IDP-CERT-NAME> -cert <Name of the IdP
       certificate downloaded above>
    2  <!--NeedCopy-->
    ```

  - ユーザーフィールド -UserPrincipalName。Azure IdP の「ユーザー属性とクレーム」セクションから取得されます。
  - 署名証明書名 -Azure AD では必要ありません。NetScaler が IdP への認証リクエストに署名するために使用する SAML SP 証明書（プライベートキー付き）を選択します。IdP が認証要求署名を検証できるように、同じ証明書（秘密キーなし）を IdP にインポートする必要があります。このフィールドは、ほとんどの IdP では必要ありません。
  - 発行者名-エンティティ ID または識別子。この場合はhttps://gateway.nssvctesting.net。負荷分散展開シナリオでは、負荷分散仮想サーバーの FQDN を使用する必要があります。
  - 署名されていないアサーションを拒否-IdP からのアサーションに署名が必要な場合に指定できるオプション。デフォルトのオプションは［オン］です。
  - Audience-IdP によって送信されたアサーションが適用されるオーディエンス。これは通常、サービスプロバイダーを表すエンティティ名または URL です。
  - 署名アルゴリズム-SAML トランザクションの署名/検証に使用されるアルゴリズム。デフォルト値は RSA-SHA256 です。
  - ダイジェスト方式-SAML トランザクションのダイジェストの計算/検証に使用されるアルゴリズム。デフォルト値は SHA256 です。

- デフォルト認証グループ-抽出されたグループに加えて、認証が成功した場合に選択されるデフォルトグループ。
- Group Name Field：ユーザグループを含むアサーション内のタグの名前。
- スキュー時間（分）-このオプションでは、NetScaler サービスプロバイダーが受信アサーションで許容するクロックスキューを分単位で指定します。たとえば、16:00 にスキュー時間を 10 分に設定した場合、SAML アサーションは 15:50 から 16:10（合計 20 分）まで有効です。デフォルトのスキュー時間は 5 分です。
- 2 つの要素- オフ
- 要求された認証コンテキスト-正確
- 認証クラスタイプ-なし
- サムプリントを送信-オフ
- ユーザー名を強制する-オン
- 認証を強制する-オフ
- SAML レスポンスを保存-オフ

2. SAML アクションに対応する SAML ポリシーを作成し、そのポリシーを認証仮想サーバーにバインドします。
   - [セキュリティ] **> [AAA** アプリケーショントラフィック] **> [**ポリシー] **> [**認証] **> [**詳細ポリシー] **> [**ポリシー] に移動し、[追加] をクリックします。
   - 「認証 **SAML** ポリシーの作成」ページで、次の詳細を入力します。
     - 名前-SAML ポリシーの名前を指定します。
     - Action Type - SAML を認証アクションの種類として選択します。
     - アクション-SAML ポリシーをバインドする SAML サーバープロファイルを選択します。
     - 表現-ユーザーが SAML サーバーで認証する必要があるかどうかを判断するために SAML ポリシーが使用するルールまたは式の名前を表示します。テキストボックスに「rule = true」という値を設定すると、SAML ポリシーが有効になり、対応する SAML アクションが実行されます。
3. SAML ポリシーを VPN 仮想サーバーにバインドし、認証プロファイルを通じて VPN 仮想サーバーを認証仮想サーバーにリンクします。バインディング手順の詳細については、「認証ポリシーをバインドする」を参照してください。

注:

- Azure AD では、SAML リクエストのサブジェクト ID フィールドは想定していません。
- NetScaler がサブジェクト ID フィールドを送信しないようにするには、NetScaler CLI で次のコマンドを入力します。
  ```
  nsapimgr_wr.sh -ys call="ns_saml_dont_send_subject"
  ```
  このコマンドは nFactor 認証ワークフローにのみ適用されます。

SAML でサポートされる機能の追加

August 15, 2023

SAML では次の機能がサポートされています。

SAML SP および IdP 構成のメタデータの読み取りと生成のサポート

NetScaler アプライアンスは、SAML サービスプロバイダー（SP）と ID プロバイダー（IdP）の両方の構成エンティティの手段としてメタデータファイルをサポートするようになりました。メタデータファイルは、エンティティの設定を記述する構造化された XML ファイルです。SP と IdP のメタデータファイルは別々です。デプロイメントに基づき、1 つの SP または IdP エンティティに複数のメタデータファイルがある場合もあります。
管理者は、NetScaler 上のメタデータファイル（SAML SP および IdP）をエクスポートおよびインポートできます。SAML SP および IdP のメタデータのエクスポートとインポートの機能については、次のセクションで説明します。

SAML SP のメタデータエクスポート

NetScaler が SAML SP として構成されていて、SAML IdP が NetScaler SP 構成を含むメタデータをインポートしようとしている例を考えてみましょう。NetScaler アプライアンスには、SAML SP 構成を指定する「samlAction」属性がすでに構成されていると仮定します。

ユーザーまたは管理者からメタデータをエクスポートするには、次に示すように NetScaler Gateway または認証仮想サーバーにクエリを実行します。

```
1 https://vserver.company.com/metadata/samlsp/<action-name>
```

SAML SP のメタデータのインポート

現在、NetScaler アプライアンスの SAML アクション構成にはさまざまなパラメーターが必要です。管理者はこれらのパラメータを手動で指定します。ただし、異なる SAML システムとの相互運用に関しては、管理者は命名法に気付かないことがよくあります。IdP のメタデータが利用可能であれば、「samlAction」エンティティの設定の大部分を避けることができます。実際、IdP メタデータファイルを指定すると、IdP 固有の設定全体が省略されることがあります。’samlAction ‘エンティティは、メタデータファイルから設定を読み取るための追加パラメータを取るようになりました。

NetScaler アプライアンスにメタデータをインポートする場合、メタデータには使用する署名アルゴリズムは含まれず、エンドポイントの詳細が含まれます。メタデータは、メタデータそのものを検証するために使用できる特定のアルゴリズムで署名できます。アルゴリズムは‘samlAction’エンティティには保存されません。

したがって、’samlAction’エンティティで指定する内容は、データを送信するときに使用されます。受信データには、NetScaler アプライアンスが処理する別のアルゴリズムが含まれる場合があります。

最大サイズは 64 K バイトのメタデータをインポートできます。

コマンドラインインターフェイスを使用してメタデータファイルをフェッチする。

```
set samlAction <name> [-metadataUrl <url> [-metadataRefreshInterval <int>] https://idp.citrix.com/samlidp/metadata.xml
```

> 注
>
> metadataRefreshInterval パラメータは、指定したメタデータ URL からメタデータ情報をフェッチする間隔（分）です。デフォルト値は 36000 です。

Metadata import for SAML IdP

「samlidpProfile」パラメータは、SP に固有の設定全体を読み取るために、新しい引数を取ります。SAML IdP の設定は、SP 固有のプロパティを SP メタデータファイルに置き換えることで簡略化できます。このファイルは HTTP 経由で照会されます。

コマンドラインインターフェイスを使用してメタデータファイルから読み込むには:

```
set samlIdPProfile <name> [-metadataUrl <url>] [-metadataRefreshInterval <int>]
```

Name-value attribute support for SAML authentication

SAML 認証属性に一意の名前と値を設定できるようになりました。名前は SAML アクションパラメータで設定され、値は名前のクエリによって取得されます。name 属性値を指定すると、管理者は属性名に関連付けられた属性値を簡単に検索できます。また、管理者は属性を値だけで覚える必要がなくなりました。

> 重要
>
> - samlAction コマンドでは、合計サイズが 2048 バイト未満で、カンマで区切って最大 64 個の属性を設定できます。
> - 属性リストを使用することをお勧めします。「属性 1 から属性 16」を使用すると、抽出された属性のサイズが大きい場合にセッションが失敗します。

CLI を使用して名前と値の属性を設定するには

コマンドプロンプトで入力します。

```
add authentication samlAction <name> [-Attributes <string>]
```

例:

```
add authentication samlAction samlAct1 -attributes  “mail,sn,userprincipalName”
```


SAML IdP に対するアサーションコンシューマーサービス **URL** のサポート

SAML ID プロバイダー（IdP）として構成された NetScaler アプライアンスは、SAML サービスプロバイダー（SP）要求を処理するためのアサーションコンシューマーサービス（ACS）インデックス作成をサポートするようになりました。SAML IdP は、ACS インデックス設定を SP メタデータからインポートするか、ACS インデックス情報を手動で入力できるようにします。
次の表は、NetScaler アプライアンスを SAML SP または SAML IdP として使用する展開環境に固有の記事の一覧です。

他の特定のデプロイメントに関するいくつかの情報:

- FIPS デバイス上の SAML SP としての NetScaler
- NetScaler を SAML IdP としてシングルサインオン用に Office365 を構成する

認証メカニズムに対する **WebView** クレデンシャルタイプのサポート

NetScaler アプライアンスの認証が AuthV3 プロトコルをサポートできるようになりました。AuthV3 プロトコルの WebView 資格情報タイプは、すべてのタイプの認証メカニズム (SAML や OAuth を含む) をサポートします。WebView 資格情報の種類は AuthV3 の一部で、Citrix Receiver およびブラウザによって Web アプリケーションに実装されます。

次の例は、NetScaler Gateway と Citrix Receiver を経由する WebView イベントのフローを説明しています。

1. Citrix Receiver は、AuthV3 プロトコルのサポートについて NetScaler Gateway とネゴシエートします。
2. NetScaler アプライアンスは肯定的に反応し、特定の開始 URL を提案します。
3. その後、Citrix Receiver は特定のエンドポイント（URL）に接続します。
4. NetScaler Gateway は、WebView を起動するための応答をクライアントに送信します。
5. Citrix Receiver は WebView を起動し、最初のリクエストを NetScaler アプライアンスに送信します。
6. NetScaler アプライアンスは URI をブラウザログインエンドポイントにリダイレクトします。
7. 認証が完了すると、NetScaler アプライアンスは WebView に完了応答を送信します。
8. WebView が終了し、セッション確立のために AuthV3 プロトコルを続行するように Citrix Receiver に制御が戻されます。

SAML SP でのセッションインデックスサイズの増加

SAML サービスプロバイダ (SP) の SessionIndex サイズが 96 バイトに増加しました。以前は、sessionIndex のデフォルトの最大サイズは 63 バイトでした。

> 注
>
> NetScaler 13.0 ビルド 36.x で導入されたサポート

SAML SP のカスタム認証クラスリファレンスのサポート

SAML action コマンドで、カスタム認証クラス参照属性を設定できます。カスタム認証クラス参照属性を使用すると、適切な SAML タグ内のクラス名をカスタマイズできます。カスタム認証クラス参照属性と名前空間は、SAML SP 認証リクエストの一部として SAML IdP に送信されます。

以前は、SAML action コマンドを使用すると、authnctxClassRef 属性に定義されている定義済みクラスのセットのみを設定できました。

> 重要
>
> customAuthnctxClassRef 属性を設定する際には、次の点を確認してください。

- クラス名には、英数字、または適切な XML タグを持つ有効な URL を含める必要があります。
- 複数のカスタムクラスを設定する必要がある場合は、各クラスをカンマで区切る必要があります。

CLI を使用して **customAuthnCtxClassRef** 属性を設定するには

コマンドプロンプトで入力します。

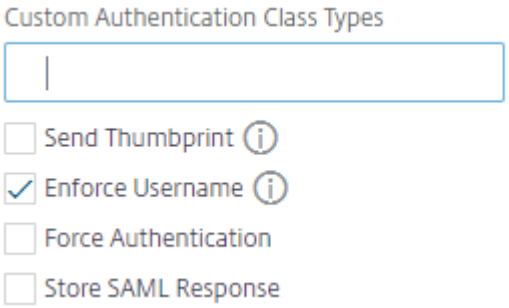
- add authentication samlAction <name> [-customAuthnCtxClassRef <string>]
- set authentication samlAction <name> [-customAuthnCtxClassRef <string>]

例:

- `add authentication samlAction samlact1 -customAuthnCtxClassRef http://www.class1.com/LoA1,http://www.class2.com/LoA2`
- `set authentication samlAction samlact2 -customAuthnCtxClassRef http://www.class3.com/LoA1,http://www.class4.com/LoA2`

GUI を使用して **customAuthnCtxClassRef** 属性を設定するには

1. セキュリティ **>** **AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** アクション **>** **SAML** に移動します。
2. [SAML] ページで [サーバー] タブを選択し、[追加] をクリックします。
3. **[**認証 **SAML** サーバーの作成] ページで、SAML アクションの名前を入力します。
4. 下にスクロールして、[カスタム認証クラスタイプ**]** セクションのクラスタイプを設定します。



SAML IdP でのアーティファクトバインディングのサポート

SAML ID プロバイダー（IdP）として構成された NetScaler アプライアンスは、アーティファクトバインディングをサポートします。Artifact バインディングは SAML IdP のセキュリティを強化し、悪意のあるユーザによるアサーションの検査を制限します。

SAML IdP に対するアサーションコンシューマーサービス **URL** のサポート

SAML ID プロバイダー（IdP）として構成された NetScaler アプライアンスは、SAML サービスプロバイダー（SP）要求を処理するためのアサーションコンシューマーサービス（ACS）インデックス作成をサポートするようになりました。SAML IdP は、ACS インデックス設定を SP メタデータからインポートするか、ACS インデックス情報を手動で入力できるようにします。

FIPS オフロードサポート

SAML サービスプロバイダーとして使用される NetScaler MPX FIPS アプライアンスは、暗号化されたアサーションをサポートするようになりました。また、SAML サービスプロバイダーまたは SAML ID プロバイダーとして機能する NetScaler MPX FIPS アプライアンスを、FIPS ハードウェアで SHA2 アルゴリズムを使用するように構成できるようになりました。

> 注
>
> FIPS モードでは、RSA-V1_5 アルゴリズムのみがキートランスポートアルゴリズムとしてサポートされます。

コマンドラインインターフェイスを使用して **FIPS** オフロードサポートを構成します。

1. SSL FIPS を追加する

   add ssl fipsKey fips-key

2. CSR を作成し、CA サーバで使用して証明書を生成します。その後、**/nsconfig/ssl** に証明書をコピーできます。このファイルが *fips3cert.cer* であると仮定します。


```
add ssl certKey fips-cert -cert fips3cert.cer -fipsKey fips-key
<!--NeedCopy-->
```

3. SAML SP モジュールの SAML アクションでこの証明書を指定する

```
set samlAction <name> -samlSigningCertName fips-cert<!--NeedCopy
-->
```

4. SAML IdP モジュールの SAMLidpProfile の証明書を使用する

```
set samlidpprofile fipstest -samlIdpCertName fips-cert<!--NeedCopy
-->
```

一般的な **SAML** 用語

SAML の一般的な用語をいくつか挙げます。

- アサーション： SAML アサーションは、ユーザの認証後に ID プロバイダからサービスプロバイダに返される XML ドキュメントです。アサーションには、SAML 標準で定義されている特定の構造があります。
- アサーションのタイプ**:** アサーションのタイプは次のとおりです。
  - 認証-ユーザーは特定の時間に、特定の手段で認証されます。
  - Authorization-指定されたリソースへのアクセスを許可または拒否されたユーザー
  - Attributes-ユーザーは指定された属性に関連付けられます。
- アサーションコンシューマサービス **(ACS):** SAML アサーションの受信と解析を担当するサービスプロバイダのエンドポイント (URL)
- オーディエンス制限： SAML アサーション内の値で、アサーションの対象となるユーザー（および対象者のみ）を指定します。「オーディエンス」はサービスプロバイダーで、通常は URL ですが、技術的には任意のデータ文字列としてフォーマットできます。
- **ID** プロバイダ **(IdP):** SAML の観点では、ID プロバイダは、サービスプロバイダからの要求に応じて、ユーザの ID を検証するエンティティです。

  ID プロバイダーは、ユーザーの ID を維持および認証する責任を負います。
- サービスプロバイダ **(SP):** SAML に関しては、サービスプロバイダ (SP) はユーザーにサービスを提供し、ユーザーが SAML を使用してサインインできるようにします。ユーザーがサインインしようとすると、SP は ID プロバイダ (IdP) に SAML 認証要求を送信します。
- **SAML** バインディング**:** SAML リクエスタとレスポンダは、メッセージを交換して通信します。これらのメッセージを転送するメカニズムを SAML バインディングと呼びます。
- **HTTP Artifact:** SAML プロトコルでサポートされるバインディングオプションの 1 つ。HTTP Artifact は、SAML リクエスタとレスポンダが HTTP User-Agent を使用していて、技術的またはセキュリティ上の理由からメッセージ全体を転送したくないシナリオで役立ちます。代わりに、SAML Artifact が送信されま

す。SAML Artifact は、完全な情報の一意の ID です。IdP は Artifact を使用して完全な情報を取得できます。Artifact issuer は、Artifact が保留されている間、状態を維持する必要があります。Artifact 解決サービス (ARS) を設定する必要があります。

HTTP Artifact はアーティファクトをクエリパラメータとして送信します。

- **HTTP POST:** SAML プロトコルでサポートされているバインディングオプションの 1 つ。

  HTTP POST は、メッセージコンテンツを POST パラメータとしてペイロードで送信します。

- **HTTP** リダイレクト**:** SAML プロトコルでサポートされるバインディングオプションの 1 つ。

  HTTP リダイレクトを使用すると、サービスプロバイダはログインが発生する ID プロバイダにユーザをリダイレクトし、ID プロバイダはユーザをサービスプロバイダにリダイレクトして戻します。HTTP リダイレクトには、User-Agent（ブラウザ）による介入が必要です。

  HTTP リダイレクトはメッセージコンテンツを URL で送信します。このため、レスポンスのサイズは通常、ほとんどのブラウザで許可されている URL の長さを超えるため、SAML レスポンスには使用できません。

  注： NetScaler アプライアンスは、ログアウト中の POST バインディングとリダイレクトバインディングをサポートします。

- メタデータ: メタデータは、SP と IdP の相互通信方法を知るための設定データで、XML 標準になります。

SAML 認証に関連する **Citrix** その他の有用な記事

SAML 認証に関する次の記事が役に立つかもしれません。

- https://support.citrix.com/article/CTX277558
- https://support.citrix.com/article/CTX259127
- https://support.citrix.com/article/CTX228135
- https://support.citrix.com/article/CTX221631
- https://support.citrix.com/article/CTX138988

OAuth 認証

August 15, 2023

認証、承認、および監査トラフィック管理機能は、OAuth および OpenID Connect (OIDC) 認証をサポートします。Google、Facebook、Twitter などのアプリケーションでホストされているサービスに対するユーザーの承認と認証を行います。

注意事項

- このソリューションが機能するには、NetScaler Advanced Edition 以上が必要です。
- アプライアンスが OIDC を使用して OAuth IdP として機能するには、NetScaler アプライアンスがバージョン 12.1 以降である必要があります。
- NetScaler ADC アプライアンス上の OAuth は、「OpenID Connect 2.0」に準拠しているすべての SAML IdP に対して認定されます。

NetScaler アプライアンスは、SAML と OIDC を使用して、サービスプロバイダー（SP）またはアイデンティティプロバイダー（IdP）として動作するように構成できます。以前は、IdP として構成された NetScaler アプライアンスは SAML プロトコルのみをサポートしていました。NetScaler 12.1 バージョン以降、NetScaler は OIDC もサポートしています。

OIDC は OAuth の承認/委任を拡張したものです。NetScaler アプライアンスは、同じクラスの他の認証メカニズムで OAuth プロトコルと OIDC プロトコルをサポートします。OIDC は OAuth のアドオンであり、ユーザー情報を収集できないトークンのみを取得する OAuth とは対照的に、認可サーバーからユーザー情報を取得する方法を提供します。

認証メカニズムは、OpenID トークンのインライン検証を容易にします。NetScaler アプライアンスは、証明書を取得してトークンの署名を検証するように構成できます。

OAuth および OIDC メカニズムを使用する主な利点は、ユーザー情報がホストされるアプリケーションに送信されないことです。したがって、個人情報の盗難のリスクは大幅に減少します。

認証、承認、監査用に構成された NetScaler アプライアンスは、HMAC HS256 アルゴリズムを使用して署名された受信トークンを受け入れるようになりました。さらに、SAML アイデンティティプロバイダ (IdP) の公開キーは、URL エンドポイントから学習するのではなく、ファイルから読み取られます。

NetScaler の実装では、アプリケーションには認証、承認、監査のトラフィック管理仮想サーバーからアクセスされます。したがって、OAuth を設定するには、認証、承認、および監査トラフィック管理仮想サーバに関連付ける OAuth ポリシーを設定する必要があります。

OpenID 接続プロトコルの設定

NetScaler アプライアンスは、OIDC プロトコルを使用して ID プロバイダーとして構成できるようになりました。OIDC プロトコルは、NetScaler ADC アプライアンスのアイデンティティ提供機能を強化します。シングルサインオンで、エンタープライズ全体でホストされているアプリケーションにアクセスできるようになりました。OIDC は、ユーザーのパスワードを転送しないで、特定の有効期間を持つトークンで動作することにより、セキュリティを強化します。また、OIDC は、アプリやサービスなどのブラウザ以外のクライアントと統合するように設計されています。そのため、多くの実装では OIDC を広く採用している。

OpenID コネクトをサポートすることの利点

- OIDC は、ユーザーが組織全体で単一の ID を持つため、複数の認証パスワードを維持するオーバーヘッドを排除します。
- OIDC は、パスワードがユーザーの ID プロバイダーとのみ共有され、アクセスするアプリケーションとは共有されないため、パスワードに堅牢なセキュリティを提供します。
- OIDC はさまざまなシステムとの相互運用性が非常に高いため、ホストされるアプリケーションが OpenID を受け入れやすくなります。
- OIDC は、ネイティブクライアントがサーバーと簡単に統合できるようにするシンプルなプロトコルです。

GUI を使用して **OpenID Connect** プロトコルを使用して **NetScaler** アプライアンスを **IdP** として構成するには

1. [構成**] > [**セキュリティ**] > [AAA** アプリケーショントラフィック**] > [**ポリシー**] > [**認証**] > [**高度なポリシー**] > [OAuth IdP]** に移動します。
2. [プロファイル] をクリックし、[追加**]** をクリックします。

   [認証 **OAuth IDP** プロファイルの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。

   - **Name**： 認証プロファイルの名前。
   - クライアント **ID**： SP を識別する一意の文字列。
   - クライアントシークレット：SP を識別する一意のシークレット。
   - リダイレクト **URL** —コード/トークンのポスト先となる SP 上のエンドポイント。
   - 発行者名—IdP を識別する文字列。
   - オーディエンス—IdP によって送信されるトークンのターゲット受信者。これは受信者がチェックするかもしれません。
   - スキュー時間—トークンが有効である時間。
   - **[Default Authentication** Group]：ポリシーの評価を簡素化し、ポリシーのカスタマイズを支援するために、このプロファイルのセッションに追加されるグループ。
3. **[**ポリシー**]** をクリックし、**[**追加**]** をクリックします。
4. [認証 **OAuth IDP** ポリシーの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。

   - 名前 —認証ポリシーの名前。
   - **Action** —以前に作成されたプロファイルの名前。
   - [**Log Action**]：要求がこのポリシーに一致したときに使用するメッセージログアクションの名前。必須の提出ではありません。
   - **Undefined-Result** アクション：ポリシー評価の結果が未定義（UNDEF）である場合に実行するアクション。必須フィールドではありません。
   - 式—ポリシーが特定の要求に応答するために使用する高度なポリシー表現。例：true。
   - **Comments** - ポリシーに関するコメント。

OAuthIdP ポリシーと **LDAP** ポリシーを認証仮想サーバーにバインドする

1. 設定 **>** セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** アクション **> LDAP** に移動します。
2. [**LDAP** アクション] 画面で、[追加] をクリックします。
3. [認証 **LDAP** サーバの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。
   - **Name** — LDAP アクションの名前
   - サーバ名**/**サーバ **IP** — LDAP サーバの FQDN または IP を指定します。
   - [セキュリティタイプ**]**、**[**ポート**]**、**[**サーバータイプ**]**、**[**タイムアウト] の適切な値を選択する
   - [認証**]** がオンになっていることを確認します
   - ベース **DN**： LDAP 検索を開始する基本。例えば、dc=aaa、dc=local。
   - 管理者バインド **DN:** LDAP サーバーへのバインドのユーザー名。たとえば、admin@aaa.local。
   - 管理者パスワード**/**パスワードの確認**:LDAP** をバインドするためのパスワード
   - [接続のテスト] をクリックして、設定をテストします。
   - サーバーログオン名属性**:**「**samAccountName**」を選択します。
   - その他のフィールドは必須ではないため、必要に応じて設定できます。
4. 設定 **>** セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** ポリシーにナビゲートして下さい。
5. [認証ポリシー] 画面で、[追加] をクリックします。
6. [認証ポリシーの作成] ページで、次のパラメータの値を設定し、[作成**]** をクリックします。
   - **Name** — LDAP 認証ポリシーの名前。
   - [アクションタイプ] —[**LDAP**] を選択します。
   - [アクション**]** — LDAP アクションを選択します。
   - 式—ポリシーが特定の要求に応答するために使用する高度なポリシー表現。例：true**。

CLI を使用して **OpenID Connect** プロトコルを使用して **NetScaler** アプライアンスを **IdP** として構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `add authentication OAuthIDPProfile <name> [-clientID <string>][-clientSecret ][-redirectURL <URL>][-issuer <string>][-audience <string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]<!--NeedCopy-->`
- `add authentication OAuthIdPPolicy <name> -rule <expression> [-action <string> [-undefAction <string>] [-comment <string>][-logAction <string>]<!--NeedCopy-->`

- add authentication ldapAction aaa-ldap-act -serverIP 10.0.0.10 -ldapBase "dc=aaa,dc=local"<!--NeedCopy-->
- ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -ldapLoginName sAMAccountName<!--NeedCopy-->
- add authentication policy aaa-ldap-adv-pol -rule **true** -action aaa -ldap-act<!--NeedCopy-->
- bind authentication vserver auth_vs -policy <ldap_policy_name> -priority 100 -gotoPriorityExpression NEXT<!--NeedCopy-->
- bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -priority 5 -gotoPriorityExpression END<!--NeedCopy-->
- bind vpn global -certkey <><!--NeedCopy-->

> 注
>
> 複数のキーをバインドできます。バインドされた証明書のパブリック部分は、jwks_uri query (https://gw/oauth/idp/certs)への応答として送信されます。

OAuth サービスプロバイダーとしての NetScaler ADC

February 15, 2024

認証、承認、および監査トラフィック管理機能は、Google、Facebook、Twitter などのアプリケーションでホストされているアプリケーションに対してユーザーを認証するための OAuth 認証をサポートします。

注意事項

- このソリューションが機能するには、NetScaler Advanced Edition 以上が必要です。
- NetScaler 上の OAuth は、「OpenID 接続 2.0」に準拠しているすべての SAML IDP に対応しています。

> 重要:
>
> コンテンツが多い Web サイトがセッションの期限切れ時に複数の認証要求を送信すると、NetScaler が CSRF エラーで応答することがあります。回避策として、OAuth ポリシーを設定するときに、メインエントリポイントであるホスト名とパスの両方に対してポリシーが設定されていることを確認することをお勧めします。

GUI を使用して OAuth を設定する

1. OAuth アクションとポリシーを設定します。

[セキュリティ] > [AAA-アプリケーショントラフィック] > [ポリシー] > [認証] > [高度なポリシー] > [ポリシー] に移動し、アクションタイプとして OAuth を使用してポリシーを作成し、必要な OAuth アクションをポリシーに関連付けます。

2. OAuth ポリシーを認証仮想サーバーに関連付けます。

   [セキュリティ] > [AAA-アプリケーショントラフィック] > [仮想サーバ] に移動し、OAuth ポリシーを認証仮想サーバに関連付けます。

> 注:
>
> 属性 (1 ～16) は OAuth レスポンスで抽出できます。現在、これらの属性は評価されません。これらは将来の参照のために追加されます。

CLI を使用して **OAuth** を設定する

1. OAuth アクションを定義します。

```
add authentication OAuthAction <name> [-OAuthType <OAuthType>] [-
    authorizationEndpoint <URL>] [-tokenEndpoint <URL>] [-
    idtokenDecryptEndpoint <URL>] [-clientID <string>][-
    clientSecret ]  [-defaultAuthenticationGroup <string>] [-
    Attribute1 <string>] [-Attribute2 <string>] [-Attribute3 <
    string>] [-Attributes <string>] [-tenantID <string>] [-
    GraphEndpoint <string>] [-refreshInterval <positive_integer>]
    [-CertEndpoint <string>][-audience <string>] [-userNameField <
    string>] [-skewTime <mins>] [-issuer <string>] [-UserInfoURL <
    URL>] [-CertFilePath <string>] [-grantType ( CODE | PASSWORD )]
     [-authentication ( ENABLED | DISABLED )] [-introspectURL <URL
    >][-allowedAlgorithms <allowedAlgorithms> ...] [-PKCE ( ENABLED
     | DISABLED )] [-tokenEndpointAuthMethod <
    tokenEndpointAuthMethod>] [-metadataUrl <URL>] [-resourceUri <
    URL>]
<!--NeedCopy-->
```

2. アクションを高度な認証ポリシーに関連付けます。

```
add authentication Policy <name> -rule <expression> -action <
    string>
<!--NeedCopy-->
```

例:

```
add authentication oauthAction a -authorizationEndpoint https://
    example.com/ -tokenEndpoint https://example.com/ -clientiD sadf
     -clientsecret df
<!--NeedCopy-->
```

認証 OAuthAction パラメータの詳細については、「認証 OAuthAction」を参照してください。

> 注:
>
> certEndpoint が指定されると、NetScaler は設定された頻度でそのエンドポイントをポーリングしてキーを学習します。

NetScaler がローカルファイルを読み取り、そのファイルからキーを解析するように構成するために、次のように新しい構成オプションが導入されました:

```
set authentication OAuthAction <name> -CertFilePath <path to local file
    with jwks>
<!--NeedCopy-->
```

OAuth 機能は、NetScaler Gateway と NetScaler の依存パーティ（RP）側と IdP 側のトークン API で次の機能をサポートするようになりました。

- PKCE（コード交換のための証明キー）のサポート
- client_assertion のサポート

OAuth 認証に対する名前と値の属性のサポート

OAuth 認証属性に一意の名前と値を設定できるようになりました。名前は OAuth アクションパラメーターで「Attributes」として設定され、名前はクエリーによって取得されます。抽出された属性は、認証、認可、および監査セッションに保存されます。管理者は、選択した属性名の指定方法に基づいて、`http.req.user.attribute("attribute name")`または`http.req.user.attribute(1)`を使用して、これらの属性をクエリできます。

属性の名前を指定することで、管理者はその属性名に関連付けられている属性値を簡単に検索できます。また、管理者は「attribute1 to attribute16」を番号だけで覚えておく必要がなくなりました。

> 重要
>
> OAuth コマンドでは、合計サイズが 1024 バイト未満の最大 64 個の属性をカンマで区切って設定できます。
>
> 注
>
> 「属性 1 から属性 16」の合計値サイズと「属性」で指定された属性の値が 10 KB 以下であれば、セッションの失敗を回避できます。

CLI を使用して名前と値の属性を設定するには

コマンドプロンプトで入力します:

```
add authentication OAuthAction <name> [-Attributes <string>]

set authentication OAuthAction <name> [-Attributes <string>]
<!--NeedCopy-->
```


例:

```
add authentication OAuthAction a1  -attributes "email,company"  -
    attribute1 email

set authentication OAuthAction oAuthAct1 -attributes "mail,sn,
    userprincipalName"
<!--NeedCopy-->
```

OAuth IdP としての NetScaler ADC

April 15, 2024

NetScaler を OpenID-Connect（OIDC）プロトコルを使用してアイデンティティプロバイダーとして構成できるようになりました。OIDC プロトコルは、NetScaler のアイデンティティ提供機能を強化します。OIDC は、ユーザーパスワードを転送せず、特定の有効期間を持つトークンを使用することで、セキュリティを強化するため、シングルサインオンでエンタープライズワイドホストアプリケーションにアクセスできるようになりました。OpenID は、アプリやサービスなどの非ブラウザクライアントと統合するようにも設計されています。したがって、OIDC プロトコルは多くの実装で広く採用されています。

> 注
>
> OIDC プロトコルを使用して OAuth IdP として動作するには、NetScaler がバージョン 12.1 以降である必要があります。

NetScaler ADC を OAuth IdP として持つ利点

- ユーザーが組織全体で 1 つの ID を持つため、複数の認証パスワードを維持するオーバーヘッドを排除します。
- パスワードは ID プロバイダでのみ共有され、アクセスするアプリケーションとは共有されないため、パスワードの堅牢なセキュリティを提供します。
- さまざまなシステムとの広範な相互運用性を提供し、ホストされているアプリケーションが OpenID を受け入れることを容易にします。

> 注
>
> このソリューションが機能するには、NetScaler Advanced Edition 以上が必要です。

GUI を使用して NetScaler を OAuth IdP として構成するには

1. 認証 OAuth IdP ポリシーを作成します。
    - [** セキュリティ] > [AAA-アプリケーショントラフィック] > [ポリシー] > [認証] > [高度なポリシー] > [OAuth IDP] **[ポリシー] に移動します。

- 「ポリシー」で、「追加」をクリックします。
- 「認証 OAuth IDP ポリシーの作成」ページで、次のパラメータの値を設定し、「作成」をクリックします。
  - 名前: 認証ポリシーの名前。
  - アクション: このポリシーに一致するリクエストまたは接続に適用する認証 OAuth IdP プロファイルの名前。詳細な手順については、ステップ 2 を参照してください。
  - ログアクション: リクエストがこのポリシーに一致する場合に使用するメッセージログアクションの名前。詳細な手順については、ステップ 3 を参照してください。この情報は入力しなくても構いません。
  - 未定義の結果アクション: ポリシー評価の結果が未定義の場合に実行するアクション。未定義のイベントは内部エラー状態を示します。使用可能なアクションは DROP と RESET です。この情報は入力しなくても構いません。
  - 式: ポリシーが特定の要求に応答するために使用する高度な表現。ポリシーと式の詳細については、「ポリシーと式」を参照してください。

2. 認証 OAuth IdP アクションを作成します。
   - 「認証 OAuth IDP ポリシーの作成」ページで、「アクション」フィールドの「追加」をクリックします。
   - 表示される [認証 OAuth IDP プロファイルの作成 **] ページで、次のパラメータの値を設定し、[作成] をクリックします。**
     - 名前: 新しい OAuth IdP シングルサインオンプロファイルの名前。
     - クライアント ID: 認証を要求している依存パーティの固有の ID。最大許容長は 127 文字です。
     - クライアントシークレット: 認証サーバーで依存パーティを承認するためのユニークなシークレット文字列。最大許容長は 239 文字です。
     - リダイレクト URL: OAuth トークンの送信先となる依存パーティの URL エンドポイント。最大許容長は 255 文字です。
     - 発行者名:NetScaler から IdP に送信されるリクエストで NetScaler を一意に識別するために使用される名前。最大許容長は 127 文字です。
     - 対象者:IdP が送信するトークンのターゲット受信者。これは受信者を表すエンティティ名または URL です。最大許容長は 127 文字です。
     - スキュー時間: このオプションは、NetScaler IdP によって送信されたトークンの有効期間を指定します。たとえば、スキュー時間が 10 分に設定されている場合、トークンは (現在の時間から 10 を引いたもの) 分から (現在の時間に 10 を加えたもの) 分まで、合計で 20 分の間有効です。デフォルトの所要時間は 5 分です。
     - デフォルト認証グループ: セッションの内部グループリストに追加されたグループ。これは、管理者が nFactor フロー内の依存パーティの設定を決定するのに役立ちます。認証ポリシーの式`AAA.USER.IS_MEMBER_OF("group name")`に使用して、依存パーティ関連の nFactor フローを識別できます。最大許容長は 63 文字です。
     - 依存パーティメタデータ URL: NetScaler IdP が構成中の依存パーティの詳細を取得できるエン

ドポイント。メタデータレスポンスには、パーティの公開鍵に依存するjwks_uriのエンドポイントが含まれている必要があります。最大許容長は 255 文字です。

- 更新間隔: 依存パーティのメタデータが更新される間隔。
- 署名サービス:NetScaler IdP がトークンを送信するときにトークンを暗号化するには、このオプションを選択します。
- 属性:ID トークンに挿入される属性の名前と値のペア。設定形式はname=value_expr@@@name2=value2_expr@@@です。この@@@形式は、名前と値のペアの間の区切り文字として使用されます。
- パスワードを送信:ID トークンで暗号化されたパスワードを送信するには、このオプションを選択します。

3. 監査メッセージアクションを作成します。

- 「認証 **OAuth IDP** ポリシーの作成」ページで、「ログアクション」フィールドの「追加」をクリックします。
- 「監査メッセージアクションの作成」ページで、次のパラメータの値を設定し、「作成」をクリックします。
  - 名前: 監査メッセージアクションの名前。
  - ログレベル: 生成されるログメッセージの重要度レベルを指定する監査ログレベル。
  - 式: ログメッセージの形式と内容を定義するデフォルトの構文式。
  - newnslog にログイン: メッセージを新しい NSLOG サーバーに送信します。

4. 認証 OAuth サーバーを作成します。

- **[**** セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**認証**] > [**高度なポリシー**] > [**アクション**] > [OAUTH** アクション] に移動し、[追加] をクリックします。**
- 「認証 OAuth サーバーの作成」ページで、必要なパラメーターの値を設定し、「作成」をクリックします。

5. OAuth IdP ポリシーを認証 OAuth サーバーにバインドします。

OAuth 機能は、NetScaler Gateway と NetScaler の依存パーティ（RP）側と IdP 側のトークン API で次の機能をサポートするようになりました。

- PKCE（コード交換のための証明キー）のサポート
- client_assertion のサポート

CLI を使用して **OIDC** プロトコルを使用して **NetScaler** を **IdP** として構成するには

コマンドプロンプトで、次のコマンドを入力します:

```
add authentication OAuthIDPProfile <name> [-clientID <string>][-
    clientSecret ][-redirectURL <URL>][-issuer <string>][-audience <
    string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]

```



```
add authentication OAuthIdPPolicy <name> -rule <expression> [-action <string>] [-undefAction <string>] [-comment <string>][-logAction <string>]

add authentication ldapAction aaa-ldap-act -serverIP 10.0.0.10 -ldapBase "dc=aaa,dc=local"

ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -ldapLoginName sAMAccountName

add authentication policy aaa-ldap-adv-pol -rule true -action aaa-ldap-act

bind authentication vserver auth_vs -policy <ldap_policy_name> -priority 100 -gotoPriorityExpression NEXT

bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -priority 5 -gotoPriorityExpression END

bind vpn global  - certkey <>
<!--NeedCopy-->
```

メモ:

- 複数のキーをバインドできます。バインドされた証明書のパブリック部分は、`jwks\\_uri query (https://gw/oauth/idp/certs)`への応答として送信されます。
- OAuth IdP イントロスペクティブエンドポイントは、プロパティ`active: true`をサポートします。
- 認証仮想サーバーが OAuth IdP として構成されている場合、既知の OAuth IdP 検出エンドポイント URL は`https://<netscaler-oauth-idp-fqdn>/oauth/idp/.well-known/openid-configuration`である必要があります。
- OAuth IdP として構成された NetScaler は、OAuth SP に送信される`.well-known-Endpoint`応答に、`client_secret_post`、`client_secret_jwt`、`private_key_jwt`および`client_secret_basic`トークンエンドポイントの認証方法を表示しません。

OIDC プロトコルでの暗号化されたトークンのサポート

OIDC メカニズムを備えた NetScaler は、暗号化されたトークンと署名されたトークンの送信をサポートするようになりました。NetScaler は JSON Web 暗号化仕様を使用して暗号化されたトークンを計算し、暗号化されたトークンのコンパクトなシリアル化のみをサポートします。OpenID トークンを暗号化するには、NetScaler は依存パーティ（RP）の公開鍵を必要とします。公開鍵は、証明書利用者の既知の設定エンドポイントをポーリングすることで動的に取得されます。

新しい「relyingPartyMetadataURL」オプションが「認証 OAuthidpProfile」プロフィールに導入されました。

CLI を使用して証明書利用者のエンドポイントを設定するには

コマンドプロンプトで入力します:

```
set authentication OAuthIDPProfile <name> [-relyingPartyMetadataURL <
    URL>] [-refreshInterval <mins>] [-status < >]
<!--NeedCopy-->
```

- **RelyingPartyMetadataURL** -NetScaler IdP が構成中の依存パーティに関する詳細を取得できるエンドポイント。メタデータ応答には、RP 公開キーの jwks_uri のエンドポイントを含める必要があります。
- **refreshInterval** -証明書を分単位で更新するために、このエンドポイントをポーリングする必要があるレートを定義します。
- **status** -ポーリング操作のステータスを反映します。NetScaler が公開鍵を正常に取得すると、ステータスは「完了」になります。

例:

```
set authentication OAuthIDPProfile sample_profile -
    relyingPartyMetadataURL https://rp.customer.com/metadata -
    refreshInterval 50 -status < >
<!--NeedCopy-->
```

エンドポイントが構成されると、NetScaler はまず依存パーティの既知のエンドポイントをポーリングして構成を読み取ります。現在、NetScaler「jwks_uri」エンドポイントのみを処理します。

- 応答に「jwks_uri」がない場合、プロファイルのステータスは完了していません。
- 応答に「jwks_uri」が存在する場合、NetScaler はそのエンドポイントもポーリングして依存パーティの公開鍵を読み取ります。

> 注:
>
> トークン暗号化では、RSAES-OAEP および AES256 GCM 暗号化タイプのアルゴリズムのみがサポートされています。

OpenID コネクトでのカスタム属性のサポート

OpenID 証明書利用者は、ユーザープロファイルの作成や承認の決定を行うために、トークンに 1 つ以上のユーザー名またはユーザープリンシパル名 (UPN) を要求する場合があります。ほとんどの場合、ユーザーグループはユーザーに認可ポリシーを適用する必要があります。場合によっては、ユーザーアカウントのプロビジョニングに、姓や姓などの詳細が必要になることがあります。

IdP として構成された NetScaler を使用すると、式を使用して OIDCid_Token に追加の属性を送信できます。高度なポリシー式は、要件に従ってカスタム属性を送信するために使用されます。NetScaler IdP は、属性に対応する式を評価し、最終的なトークンを計算します。

NetScaler は出力データに自動的にJSONifyを適用します。たとえば、数値 (SSN など) やブール値 (true または false) は引用符で囲まれません。グループなどの複数値を持つ属性は、配列マーカー (「[」と「]」) 内に配置されます。複合型属性は自動的に計算されず、要件に従ってこれらの複合値の PI 式を設定できます。

CLI を使用して証明書利用者のエンドポイントを設定するには

コマンドプロンプトで入力します:

```
set oauthidpprofile <name> -attributes <AAA-custom-attribute-pattern>
<!--NeedCopy-->
```

<AAA-custom-attribute-pattern>は、次のように記述できます。

Attribute1=PI-Expression@@@attribute2=PI-Expression@@@

‘attribute1’、’attribute2’ は ID トークンに挿入される属性の名前を表すリテラル文字列です。

> 注:
>
> 次の例では、q"{ myname=http.req.user.name@@@ssn="123456789"@@@jit="**false**"@@@groups=http.req.user.groups } " 式の最大値は 2000 バイトです。これには 4 つのカスタム属性 (myname、ssn、jit、グループ) が含まれています。各カスタム属性の最大値は 10000 バイトです。たとえば、グループ属性 (PI 式http.req.user.groupsに基づいて評価される) の OIDCid_Token には、最大 10000 バイトのデータを含めることができます。
>
> 例: set oauthidpprofile sample_1 -attributes q"{ myname=http.req.user.name@@@ssn="123456789"@@@jit="**false**"@@@groups=http.req.user.groups } "
>
> - 先行する PI 式は、属性に対して使用される値を表す高度なポリシー式です。PI 式は、「’ハードコードされた文字列’」などの文字列リテラルを送信するために使用できます。文字列リテラルは、一重引用符を二重引用符で囲み、開始パターンを二重引用符で囲みます (開始パターンは "q()")。属性の値が文字列リテラルでない場合、式は実行時に評価され、その値はトークンで送信されます。実行時の値が空の場合、対応する属性は ID トークンに追加されません。
> - 例で定義されているように、「false」は属性「jit」のリテラル文字列です。また、”ssn”には参照用にハードコードされた値があります。グループと“myname”は文字列を生成する PI 式です。

NetScaler Gateway でのアクティブ-アクティブ **GSLB** 展開のサポート

OIDC プロトコルを使用して ID プロバイダー（IdP）として構成された NetScaler Gateway は、アクティブ-アクティブ GSLB 展開をサポートできます。NetScaler Gateway IdP のアクティブ-アクティブ GSLB 展開では、受信したユーザーログインリクエストを複数の地理的場所に負荷分散できます。

> 重要
>
> セキュリティを強化するために、CA 証明書を SSL サービスにバインドし、SSL サービスで証明書検証を有効にすることをお勧めします。
> GSLB セットアップの設定の詳細については、「GSLB セットアップと設定の例」を参照してください。

NetScaler アプライアンスによる API 認証

September 12, 2023

現代のアプリケーションがクライアントとやり取りする方法にはパラダイムシフトがあります。従来、サービスへのアクセスにはブラウザクライアントが使用されていました。アプリケーションはセッションクッキーを設定してユーザーコンテキストを追跡します。最新の分散アプリケーションでは、マイクロサービス間でユーザーセッションを維持することが難しくなっています。このため、ほとんどのアプリケーションアクセスは API ベースになっています。これらの分散サービスと通信するクライアントも進化してきました。ほとんどのクライアントは、認証サーバーと呼ばれる信頼できるエンティティからトークンを取得して、ユーザーの身元とアクセスを証明します。次に、これらのクライアントは、アクセス要求のたびにトークンをアプリケーションに提示します。そのため、NetScaler のような従来のプロキシデバイスは、これらのクライアントをサポートするように進化させる必要があります。NetScaler アプライアンスは、管理者がこのようなトラフィックを処理する方法を提供します。NetScaler は、公開サービスを宛先とするすべてのトラフィックをフロントエンドする API ゲートウェイとして展開できます。API Gateway は、従来の (ハイブリッドマルチクラウドまたは HMC) 環境またはクラウドネイティブ環境にデプロイできます。API Gateway は、認証、承認、レート制限、ルーティング、キャッシュ、SSL オフロード、アプリケーションファイアウォールなどの複数のサービスを提供するために、すべてのインバウンドトラフィックを終了します。したがって、これはインフラストラクチャの重要なコンポーネントになります。

トークンの種類

API アクセス中に交換されるトークンは、主に OAuth/OpenID コネクト (OIDC) プロトコルに準拠しています。「委任アクセス」にのみ使用されるアクセストークンは OAuth プロトコルに準拠していますが、OIDC に準拠する ID トークンはユーザー情報も保持します。
アクセストークンは通常、不透明またはランダムなデータの塊です。ただし、JWT (JSON Web Token) 標準に準拠した署名付きトークンの場合もあります。ID トークンは常に署名された JWT です。

OAuth による API アクセス

NetScaler アプライアンスの OAuth 認証タイプは、OAuth プロトコルと OIDC プロトコルの両方を処理するために使用できます。OIDC は OAuth プロトコルの拡張です。

NetScaler アプライアンスの OAuthAction を使用して、ブラウザなどのインタラクティブクライアントやクライアントアプリなどのネイティブクライアントを処理できます。インタラクティブクライアントは、OIDC プロトコルを使用してログインするための ID プロバイダーにリダイレクトされます。ネイティブクライアントは帯域外でトークンを取得し、それらのトークンを NetScaler アプライアンスに提示してアクセスすることができます。

> 注:
>
> エンドポイントから取得したアクセストークンは、後続のリクエスト用にキャッシュできるため、API のパフォーマンスが向上します。

コマンドラインインターフェイスを使用してトークンキャッシュサポートを設定するには、コマンドプロンプトで次のコマンドを入力します。

```
set aaaparameter -APITokenCache <ENABLED>
<!--NeedCopy-->
```

以下のセクションでは、ネイティブクライアントが実行する API アクセス方法について説明します。

API アクセス用仮想サーバー

API アクセス用の NetScaler アプライアンスをデプロイするには、401 認証を使用してトラフィック管理（TM）仮想サーバーをデプロイします。認証 (認証、承認、監査) 仮想サーバーに関連付けられ、認証とセッションのポリシーを保持します。以下の構成スニペットは、このような仮想サーバーを 1 つ作成します。

```
Add lb vserver lb-api-access SSL <IP> 443 -authn401 On -AuthnVsName
    auth-api-access

Bind ssl vserver lb-api-access -certkeyName <ssl-cert-entity>

Add authentication vserver auth-api-access SSL
<!--NeedCopy-->
```

> 注:
>
> 構成を完了するには、サービスをトラフィック管理仮想サーバーにバインドし、認証ポリシー (後述の OAuthAction を含む) を認証仮想サーバーにバインドする必要があります。

仮想サーバーを作成したら、対応するポリシーとともに OAuthAction を追加する必要があります。OAuth アクションには、トークンの種類やその他のセキュリティメカニズムに応じて、他にもいくつかのオプションがあります。

ID トークンの OAuth 設定

ID トークンは常に署名された JWT です。つまり、ヘッダー、ペイロード、署名が含まれています。これらは自己完結型のトークンであるため、NetScaler アプライアンスはこれらのトークンをローカルで検証できます。これらのト

ークンを検証するには、アプライアンスはこれらのトークンの署名に使用された対応する秘密鍵の公開鍵を知っている必要があります。

以下は、特定の必須引数と「CerTendPoint」を使用した OAuthAction の例です。

```
Add authentication OAuthAction oauth-api-access -clientid <your-
    client-id> -clientsecret <your-client-secret> -
    authorizationEndpoint <URL to which users would be redirected for
    login> -tokenEndpoint <endpoint at which tokens could be obtained>
     -certEndpoint <URL at which public keys of IdP are published>
<!--NeedCopy-->
```

各項目の意味は次のとおりです。

- クライアント **ID**： SP を識別する一意の文字列。承認サーバーは、この ID を使用してクライアントの構成を推測します。最大文字数：127。
- **Client Secret** - ユーザーと承認サーバーによって確立されたシークレット文字列。最大長:239
- **AuthorizationEndpoint** -ユーザーが通常ログインする URL (インタラクティブクライアントを使用している場合)。
- **TokenEndpoint** -トークン/コードが取得/交換される承認サーバー上の URL
- **CerTendPoint** -認証サーバーがトークンの署名に使用される公開鍵を公開する URL。承認サーバーは複数のキーを公開し、そのうちの 1 つを選択してトークンに署名できます。

  > 注:
  >
  > クライアント ID/クライアントシークレット/承認エンドポイント/トークンエンドポイントは、API Access のオプションパラメータです。ただし、アクションエンティティはさまざまな目的で再利用できるため、これらのパラメータに値を指定することをお勧めします。

前述の構成では、ID トークンの検証には「CertEndpointPoint」が不可欠です。このエンドポイントには、トークンの署名に使用される証明書の公開鍵が含まれています。これらのパブリックキーは JWK (JSON Web キー) 仕様に対応している必要があります。

NetScaler アプライアンスで CertendPoint を構成すると、公開鍵を最新の状態に保つために、エンドポイントを定期的に（構成でカスタマイズできるデフォルトの間隔は 1 日）ポーリングします。公開鍵が利用可能になると、ADC は受信した ID トークンのローカル検証を実行できます。

不透明アクセストークンの **OAuth** 設定

不透明なトークンは、NetScaler アプライアンス上でローカルに検証できません。これらは認証サーバーで検証する必要があります。NetScaler アプライアンスは、OAuth 仕様に記載されている「イントロスペクションプロトコル」を使用してこれらのトークンを検証します。不透明トークンを検証するための新しいオプション IntrospectURL が OAuth 設定で提供されました。


```
set oauthAction oauth-api-acccess -introspectURL <URL of the
   Authorization Server for introspection>
<!--NeedCopy-->
```

イントロスペクション API の形式は、以下のようにhttps://tools.ietf.org/html/rfc7662#section-2.1の仕様に準拠しています。

```
POST /introspect HTTP/1.1
Host: server.example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
token=mF_9.B5f-4.1JqM&token_type_hint=access_token
<!--NeedCopy-->
```

認証仮想サーバーへのバインディングポリシー

OAuthAction を作成したら、それを呼び出すための対応するポリシーを作成する必要があります。

```
add authentication policy oauth-api-access -rule <> -action <oauth-
   api-access>

bind authentication vserver auth-api-access -policy oauth-api-access
   -pri 100
<!--NeedCopy-->
```

NetScaler アプライアンスのその他のセキュリティ設定

トークンの検証には、トークンの有効期間チェックが含まれます。許容時間外のトークンは拒否されます。セキュリティを強化するための追加設定は次のとおりです。これらの中には、常に設定することが推奨されるものもあります。

対象者:OAuth アクションは、トークンの受信者を指定して設定できます。すべてのトークンは、この設定された URL と照合されます。NetScaler アプライアンスには、オーディエンスフィールドが実際にアプライアンスに設定されているパターンを指すという追加機能があります。このパターンセットを使用すると、管理者はオーディエンスに複数の URL を設定できます。

```
add policy patset oauth_audiences

bind patset oauth_audiences https://app1.company.com

bind patset oauth_audiences https://app2.company.com

bind patset oauth_audiences httpsL//app1.company.com/path1

set oAuthAction oauth-api-access -audience oauth_audiences
<!--NeedCopy-->
```


前の例では、パターンセットに複数のオーディエンスが指定されています。したがって、受信トークンは、パターンセットに設定された URL のいずれかが含まれている場合にのみ許可されます。

発行者: トークンを承認するサーバーの ID。最大文字数：127。OAuth アクションでトークンの発行者を設定することをお勧めします。これにより、間違った認証サーバーによって発行されたトークンが許可されないことが保証されます。

SkewTime: NetScaler アプライアンスが受信トークンで許容する許容クロックスキューを分単位で指定します。たとえば、SkewTime が 10 の場合、トークンは (現在の時刻-10) 分から (現在時刻 + 10) 分、つまり 20 分まで有効です。デフォルト値：5

AllowedAlgorithms: このオプションにより、管理者は受信トークンの特定のアルゴリズムを制限できます。デフォルトでは、サポートされているすべてのメソッドが許可されています。ただし、これらはこのオプションを使用して制御できます。

次の構成では、RS256 と RS512 を使用するトークンのみが許可されます。

```
1   set oAuthAction oauth-api-access -allowedAlgorithms RS256 RS512
2   <!--NeedCopy-->
```

上記の構成を実行すると、RS256 と RS512 を使用するトークンのみが許可されます。

特定のトラフィックを認証からバイパスする

多くの場合、クライアントが公開しているディスカバリ API がいくつかあります。これらの API は通常、サービス自体の構成と機能を明らかにします。管理者は、次のように説明されている「認証なし」ポリシーを使用して、これらのメタデータ URL からの認証をバイパスするように NetScaler アプライアンスを構成できます。

```
1   add authentication policy auth-bypass-policy -rule <> -action
       NO_AUTHN
2
3   bind authentication vserver auth-api-access -policy auth-bypass-
       policy -pri 110
4   <!--NeedCopy-->
```

NO_AUTHN は暗黙のアクションで、ルールが一致すると認証が完了します。API アクセスの範囲を超えた NO_AUTHN アクションの用途は他にもあります。

LDAP 認証

December 8, 2023

他の種類の認証ポリシーと同様に、ライトウェイトディレクトリアクセスプロトコル (LDAP) 認証ポリシーは式とアクションで構成されます。認証ポリシーを作成したら、それを認証仮想サーバーにバインドし、プライオリティを割

り当てます。バインドするときは、プライマリポリシーまたはセカンダリポリシーとして指定します。LDAP では、標準の認証機能に加えて、他の Active Directory (AD) サーバーで、ローカルに存在しないユーザーのユーザーアカウントを検索できます。この機能は、紹介サポートまたは紹介追跡と呼ばれます。

通常、認証時に認証サーバーの IP アドレスを使用するように NetScaler を構成します。LDAP 認証サーバーでは、IP アドレスの代わりに LDAP サーバーの FQDN を使用してユーザーを認証するように ADC を設定することもできます。FQDN を使用すると、認証サーバが複数の IP アドレスのいずれかにあっても、常に 1 つの FQDN を使用する環境で、より複雑な認証、認可、および監査の設定を簡素化できます。IP アドレスではなくサーバーの FQDN を使用して認証を構成するには、認証アクションの作成時を除き、通常の構成プロセスに従います。アクションを作成するときは、ServerIP パラメータの代わりに **ServerName** パラメータを使用し、**IP** アドレスの代わりにサーバの FQDN を使用します。

LDAP サーバの IP または FQDN を使用してユーザを認証するように ADC を設定する前に、IP アドレスではなく FQDN に対して認証するように認証、承認、および監査を設定すると、認証プロセスに追加の手順が追加されることを考慮してください。ADC はユーザーを認証するたびに、FQDN を解決する必要があります。非常に多くのユーザーが同時に認証を試みると、DNS ルックアップの結果として認証プロセスが遅くなる可能性があります。

LDAP リフェラルサポートはデフォルトで無効になっており、グローバルに有効にすることはできません。LDAP アクションごとに明示的に有効にする必要があります。AD サーバーが、参照 (GC) サーバーで使用されているものと同じ`binddn credentials`を受け入れることを確認します。紹介サポートを有効にするには、紹介をフォローするように LDAP アクションを設定し、フォローする紹介の最大数を指定します。

紹介サポートが有効になっていて、NetScaler が要求に対する LDAP_REFERALR 応答を受け取ると、認証、承認、および監査は、紹介に含まれる Active Directory（AD）サーバーへの参照に続き、そのサーバー上で更新を実行します。まず、認証、承認、および監査は DNS 内の参照サーバーを検索し、そのサーバーに接続します。紹介ポリシーで SSL/TLS が必要な場合は、SSL/TLS 経由で接続します。次に、前のサーバで使用していた`binddn credentials`を使用して新しいサーバにバインドし、参照を生成したオペレーションを実行します。この機能はユーザーには透過的です。

LDAP 接続のポート番号は次のとおりです。

- 389 (セキュリティで保護されていない LDAP 接続の場合) (プレーンテキスト LDAP の場合)
- 636 セキュアな LDAP 接続 (SSL LDAP の場合)
- 3268 (Microsoft のセキュリティで保護されていない LDAP 接続の場合) (プレーンテキストのグローバルカタログサーバー用)
- 3269 (Microsoft セキュリティで保護された LDAP 接続の場合) (SSL グローバルカタログサーバー用)

次の表に、LDAP サーバのユーザ属性フィールドの例を示します。

LDAP サーバ	ユーザー属性	大文字と小文字を区別
Microsoft Active Directory サーバー	sAMAccountName	番号
Novell eDirectory	ou	はい

LDAP サーバ	ユーザー属性	大文字と小文字を区別
IBM Directory Server	uid	はい
Lotus Domino	CN	はい
Sun ONE ディレクトリ (旧 iPlanet)	uid か cn	はい

次の表に、ベース DN の例を示します。

LDAP サーバ	ベース DN
Microsoft Active Directory サーバー	DC=citrix、DC= ローカル
Novell eDirectory	ou=users、ou=dev
IBM Directory Server	cn=users
Lotus Domino	OU=City、O=Citrix、C=US
Sun ONE ディレクトリ (旧 iPlanet)	OU=People、dc=citrix、dc=com

次の表に、バインド DN の例を示します。

LDAP サーバ	バインド DN
Microsoft Active Directory サーバー	cn=Administrator、cn=Users、DC=citrix、DC=local
Novell eDirectory	cn=admin、o=citrix
IBM Directory Server	LDAP_dn
Lotus Domino	cn=Notes Administrator、O=Citrix、C=US
Sun ONE ディレクトリ (旧 iPlanet)	uid=admin,ou=Administrators, ou=TopologyManagement,o=NetscapeRoot

一般的な認証ポリシーの設定の詳細については、「認証ポリシー」を参照してください。ポリシールールで使用される NetScaler 式について詳しくは、「ポリシーと表現」を参照してください。

CLI を使用して **LDAP** 認証サーバーを作成するには

コマンドプロンプトで、次のコマンドを入力します:


```
add authentication ldapAction <name> {
 -serverIP }
  <ip\_addr|ipv6\_addr|> | {
 -serverName <string> }
  }
```

例

```
add authentication ldapAction ldap_server -serverip 1.1.1.1 -serverName
    ldap_test
```

GUI を使用して LDAP 認証サーバーを作成するには

1. システム > 認証 > 基本ポリシー > **LDAP** > サーバ > 追加にナビゲートして下さい。
2. **[**認証 **LDAP** サーバの作成] ページで、LDAP サーバのパラメータを設定します。
3. [作成] をクリックします。

CLI を使用して認証ポリシーを有効にするには

```
add authentication ldappolicy <name> <rule> [<reqAction>]
```

例:

```
add authentication ldappolicy ldap-service-policy ns_true ldap_Server
```

GUI を使用して LDAP 認証ポリシーを作成するには

1. [システム**] > [**認証**] > [**基本ポリシー**] > [LDAP] > [**ポリシー**] > [**追加**]** に移動します。
2. **[**認証 **LDAP** ポリシーの作成] ページで、LDAP ポリシーのパラメータを設定します。
3. [作成] をクリックします。

> 注
>
> LDAP サーバ/ポリシーは、[セキュリティ] タブで設定できます。セキュリティ **> AAA-**アプリケーショントラフィック > ポリシー > 認証 > 基本ポリシー **> LDAP** > サーバ/ポリシーに移動します。

CLI を使用して LDAP 参照サポートを有効にするには

コマンドプロンプトで、次のコマンドを入力します:


```
set authentication ldapAction <name> -followReferrals ON
set authentication ldapAction <name> -maxLDAPReferrals <integer>
<!--NeedCopy-->
```

例

```
set authentication ldapAction ldapAction-1 -followReferrals ON
set authentication ldapAction ldapAction-1 -maxLDAPReferrals 2
<!--NeedCopy-->
```

LDAP ユーザに対するキーベース認証のサポート

キーベース認証では、SSH を使用して LDAP サーバー内のユーザーオブジェクトに格納されている公開キーのリストを取得できるようになりました。NetScaler アプライアンスは、ロールベース認証（RBA）プロセス中に LDAP サーバーからパブリック SSH キーを抽出する必要があります。取得した公開キーは SSH と互換性があり、RBA メソッドを使用してログインできる必要があります。

「認証の追加 ldapAction」コマンドと「認証の設定 ldapAction」コマンドに新しい属性「sshPublicKey」が導入されました。この属性を使用することで、次のようなメリットが得られます。

- 取得した公開キーを格納でき、LDAP アクションはこの属性を使用して LDAP サーバーから SSH キー情報を取得します。
- 最大 24 KB の属性名を抽出できます。

> 注
>
> LDAP などの外部認証サーバは、SSH キー情報の取得にのみ使用されます。認証目的では使用されません。

次に、SSH を介したイベントのフローの例を示します。

- SSH デーモンは、パスワードフィールドを空にして AAA_AUTHENTICATE 要求を認証、承認、および監査デーモンポートに送信します。
- LDAP が SSH 公開キーを格納するように設定されている場合、認証、承認、および監査は、他の属性とともに「sshPublicKey」属性で応答します。
- SSH デーモンは、これらのキーをクライアントキーで検証します。
- SSH デーモンはリクエストペイロードでユーザー名を渡し、認証、承認、監査は汎用キーとともにこのユーザーに固有のキーを返します。

sshPublicKey 属性を構成するには、コマンドプロンプトで次のコマンドを入力します。

- add オペレーションでは、`ldapAction` コマンドの設定中に「sshPublicKey」属性を追加できます。

```
add authentication ldapAction <name> {
-serverIP <ip_addr|ipv6_addr|*> | {
-serverName <string> }
 }
```



```
5    [-serverPort <port>] … [-Attribute1 <string>] … [-Attribute16
        <string>][-sshPublicKey <string>][-authentication off]
6   <!--NeedCopy-->
```

- set オペレーションでは、既に追加された LDAPaction コマンドに「sshPublicKey」属性を設定できます。

```
1   set authentication ldapAction <name> [-sshPublicKey <string>][-
       authentication off]
2   <!--NeedCopy-->
```

LDAP 認証に対する名前と値の属性のサポート

LDAP 認証の属性に、一意の名前と値を設定できるようになりました。名前は LDAP アクションパラメータで設定され、名前はクエリーによって取得されます。この機能を使用することで、NetScaler アプライアンス管理者は以下のメリットを得ることができます。

- 属性を (値だけでなく) 名前で記憶することで、管理者の労力を最小化
- 名前に関連付けられた属性値をクエリーするように検索機能を拡張します。
- 複数の属性を抽出するオプションを提供します。

NetScaler アプライアンスのコマンドプロンプトでこの機能を構成するには、次のように入力します。

```
1  add authentication ldapAction <name> [-Attributes <string>]
2  <!--NeedCopy-->
```

例

```
1  add authentication ldapAction ldapAct1 -attributes "company, mail"
2  <!--NeedCopy-->
```

エンドツーエンドの **LDAP** 認証の検証のサポート

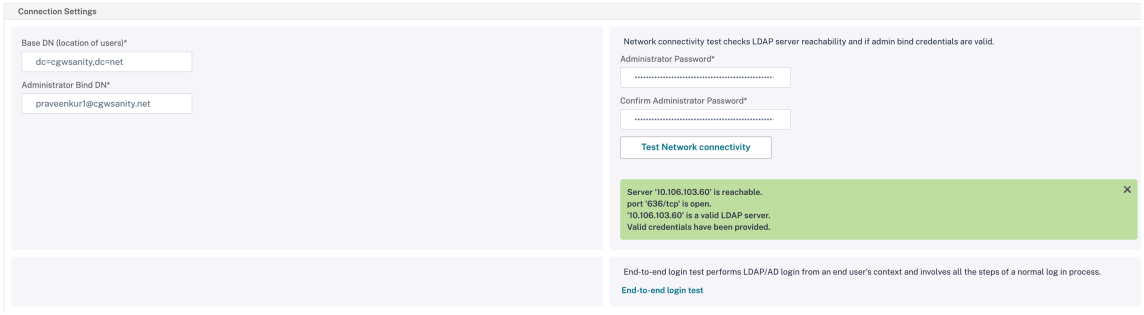
NetScaler アプライアンスは、GUI を使用してエンドツーエンドの LDAP 認証を検証できるようになりました。この機能を検証するために、GUI に新しい「テスト」ボタンが導入されました。NetScaler アプライアンスの管理者は、この機能を使用して次のメリットを得ることができます。

- フロー全体（パケットエンジン、NetScaler AAA デーモン、外部サーバー）を統合して、より優れた分析を実現
- 個々のシナリオに関連する問題の検証とトラブルシューティングにかかる時間を短縮

GUI を使用して LDAP エンドツーエンド認証のテスト結果を設定および表示するには、2 つのオプションがあります。

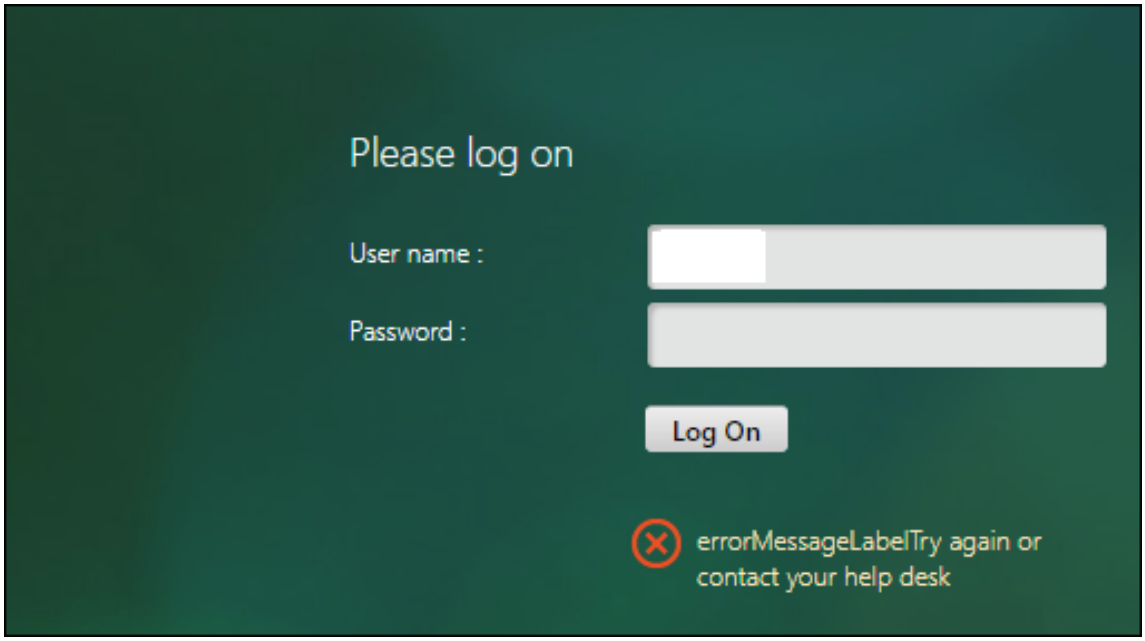
[システムから**]** オプション

1. [システム**] > [**認証**] > [**基本ポリシー**] > [LDAP]** に移動し、[サーバー]
2. 使用可能な **LDAP** アクションをリストから選択します。
3. **[**認証 **LDAP** サーバーの構成] ページで、**[**接続設定**]** セクションまで下にスクロールします。
4. [ネットワーク接続のテスト] をクリックして、LDAP サーバ接続を確認します。LDAP サーバへの接続が成功したことを示すポップアップメッセージを、TCP ポートの詳細と有効なクレデンシャルの認証情報とともに表示できます。



5. エンドツーエンドの LDAP 認証を表示するには、[エンドツーエンドログインテスト] リンクをクリックします。
6. [エンドツーエンドログインテスト] ページで、[テスト] をクリックします。
   - 認証ページで、有効な認証情報を入力してログインします。成功画面が表示されます。

   - 認証に失敗すると、エラー画面が表示されます。

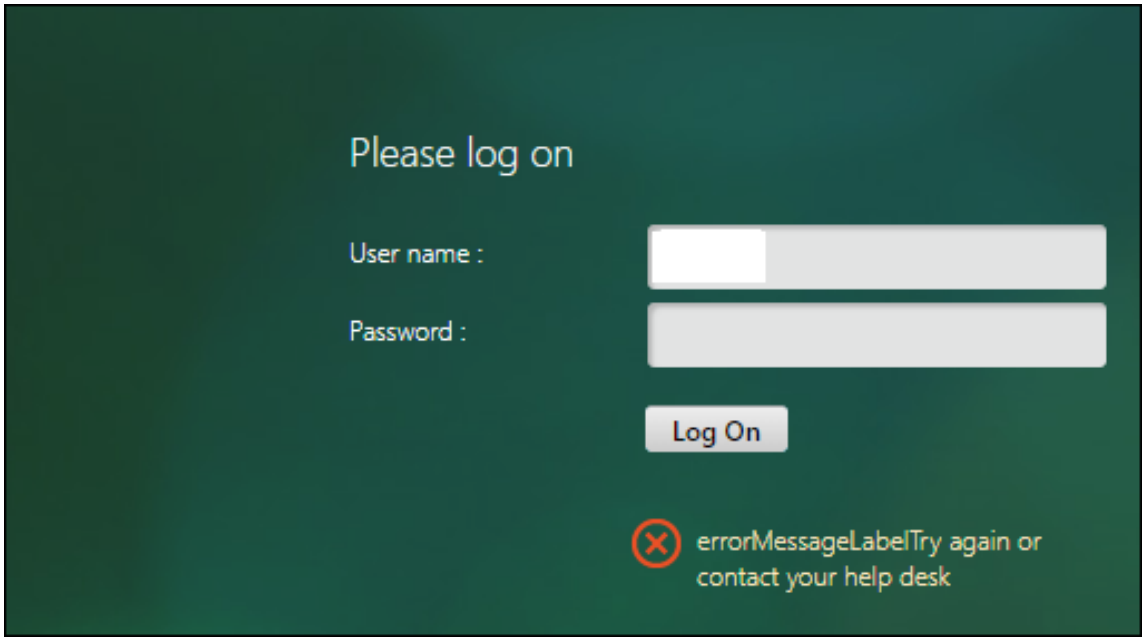


[認証**]** オプションから

1. [認証**] > [**ダッシュボード] に移動し、使用可能な LDAP アクションをリストから選択します。
2. **[**認証 **LDAP** サーバーの構成] ページの [接続設定**]** セクションには、2 つのオプションがあります。
3. LDAP サーバ接続を確認するには、[**LDAP** 到達可能性のテスト] タブをクリックします。LDAP サーバへの接続が成功したことを示すポップアップメッセージを、TCP ポートの詳細と有効なクレデンシャルの認証情報とともに表示できます。
4. エンドツーエンドの LDAP 認証ステータスを表示するには、[エンドユーザー接続のテスト] リンクをクリックします。
5. [エンドユーザー接続のテスト] ページで、[テスト] をクリックします。

    - 認証ページで、有効な認証情報を入力してログインします。成功画面が表示されます。

    - 認証に失敗すると、エラー画面が表示されます。



LDAP 認証のための **14** 日間のパスワード有効期限通知

NetScaler アプライアンスは、LDAP ベースの認証で 14 日間のパスワード有効期限通知をサポートするようになりました。この機能を使用すると、管理者はパスワードの有効期限のしきい値を日単位でエンドユーザに通知できます。14 日間のパスワード期限切れ通知は、セルフサービスパスワードリセット (SSPR) の前兆です。

> （注）
>
> パスワード期限切れ通知の最大値またはしきい値（日数）は 255 日です。

パスワード期限切れ通知のメリット

- ユーザーは自分でパスワードをリセットでき、管理者はパスワードの有効期限を日単位でエンドユーザーに柔軟に通知できます。
- エンドユーザーがパスワードの有効期限の追跡に依存する必要がなくなります。
- 有効期限が切れる前にパスワードを変更するようユーザに（日数に基づいて）VPN ポータルページへの通知を送信します。

> 注
>
> この機能は LDAP ベースの認証方式にのみ適用され、RADIUS または TACACS には適用されません。

14 日間のパスワード通知について

NetScaler アプライアンスは、LDAP 認証サーバーから 2 つの属性 (`Max-Pwd-Age and Pwd-Last-Set`) を取得します。

- **Max-Pwd-Age**。この属性は、パスワードが有効になるまでの最大時間を 100 ナノ秒間隔で示します。この値は、パスワードが設定されてからパスワードの有効期限が切れるまでに 100 ナノ秒の間隔を表す大きな整数として格納されます。
- **Pwd-Last-Set.** この属性は、アカウントのパスワードが最後に変更された日時を決定します。

NetScaler アプライアンスは、LDAP 認証サーバーから 2 つの属性を取得することで、特定のユーザーのパスワードの有効期限が切れるまでの残り時間を決定します。この情報は、認証サーバでユーザクレデンシャルが検証され、通知がユーザに返送されるときに収集されます。

新しいパラメータ「pwdExpiryNotification」が`set aaa parameter`コマンドに導入されました。このパラメーターを使用することで、管理者はパスワードの有効期限の残り日数を追跡できます。これで、NetScaler アプライアンスはエンドユーザーにパスワードの有効期限の通知を開始できます。

> 注
>
> 現在、この機能は、LDAP 実装の Microsoft AD サーバーを持つ認証サーバーでのみ動作します。OpenLDAP ベースのサーバーのサポートは後から対象となります。

14 日間のパスワード期限切れ通知を設定するイベントのフローの例を次に示します。

1. 管理者は、NetScaler アプライアンスを使用して、パスワードの有効期限（14 日）を設定します。
2. ユーザーは HTTP または HTTPS リクエストを送信して、バックエンドサーバー上のリソースにアクセスします。
3. NetScaler アプライアンスは、アクセスを提供する前に、LDAP 認証サーバーで構成されている内容を使用してユーザー資格情報を検証します。
4. NetScaler アプライアンスは、このクエリとともに、2 つの属性の詳細を取得するための要求を転送します（`Max-Pwd-Age and Pwd-Last-Set`）。
5. パスワードの有効期限が切れるまでの残り時間に基づいて、有効期限の通知が表示されます。
6. その後、ユーザは適切なアクションを実行してパスワードを更新します。

コマンドラインインターフェイスを使用して **14** 日間の有効期限通知を構成するには

> 注
>
> 14 日間の有効期限の通知は、ICA プロキシではなく、クライアントレス VPN とフル VPN のユースケースに対して構成できます。

コマンドプロンプトで、次のコマンドを入力します：

```
set aaa parameter  -pwdExpiryNotificationDays <positive_integer>

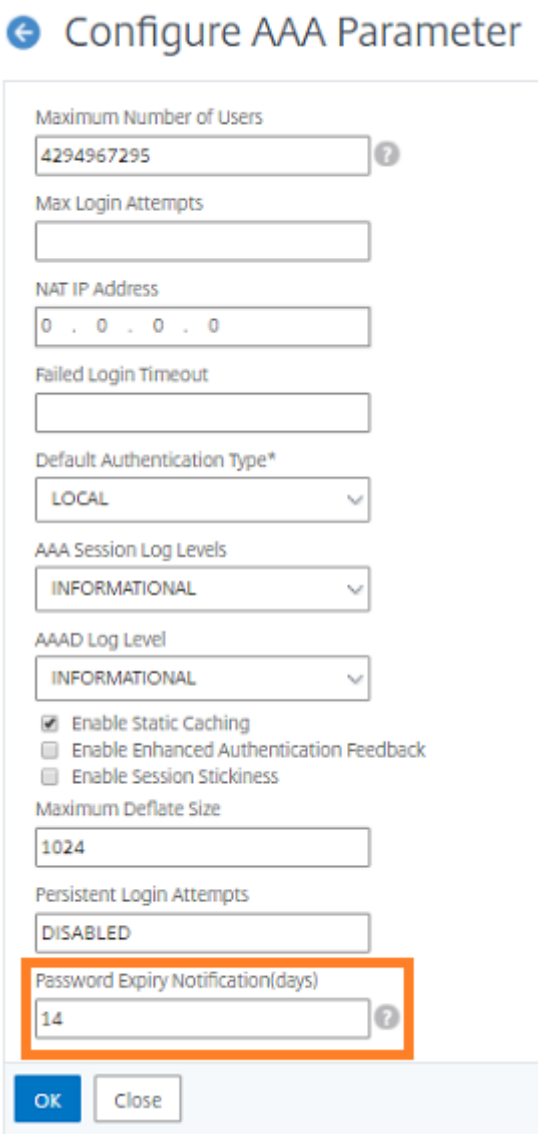
show aaa parameter
<!--NeedCopy-->
```

例

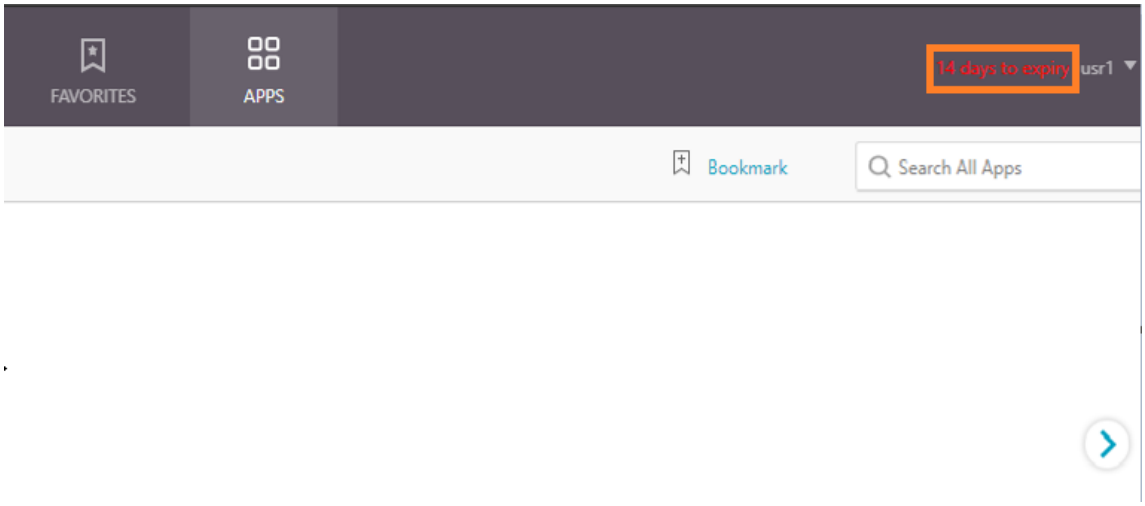

```
1      > set aaa parameter -pwdExpiryNotificationDays 14
2      Done
3      > show aaa parameter                              Configured AAA
          parameters  EnableStaticPageCaching: YES
          EnableEnhancedAuthFeedback: NO  DefaultAuthType: LOCAL
          MaxAAAUsers:           Unlimited
                                                  AAAD nat ip: None
                     EnableSessionStickiness : NO  aaaSessionLoglevel :
          INFORMATIONAL               AAAD Log Level : INFORMATIONAL
                           Dynamic address: OFF
4         GUI mode: ON
5         Max Saml Deflate Size: 1024              Password Expiry
             Notification Days: 14
6  <!--NeedCopy-->
```

GUI を使用して **14** 日間の有効期限の通知を構成するには

1. セキュリティ **> AAA-**アプリケーショントラフィック **>** 認証設定に移動します。
2. [認証 **AAA** 設定の変更] をクリックします。
3. **[AAA** パラメータの設定**]** ページで、[パスワード有効期限通知 **(**日**)]** フィールドに日数を指定します。



4.「**OK**」をクリックします。VPN ポータルページの右上隅に通知が表示されます。



管理目的で **NetScaler** アプライアンスに **LDAP** 認証を設定します

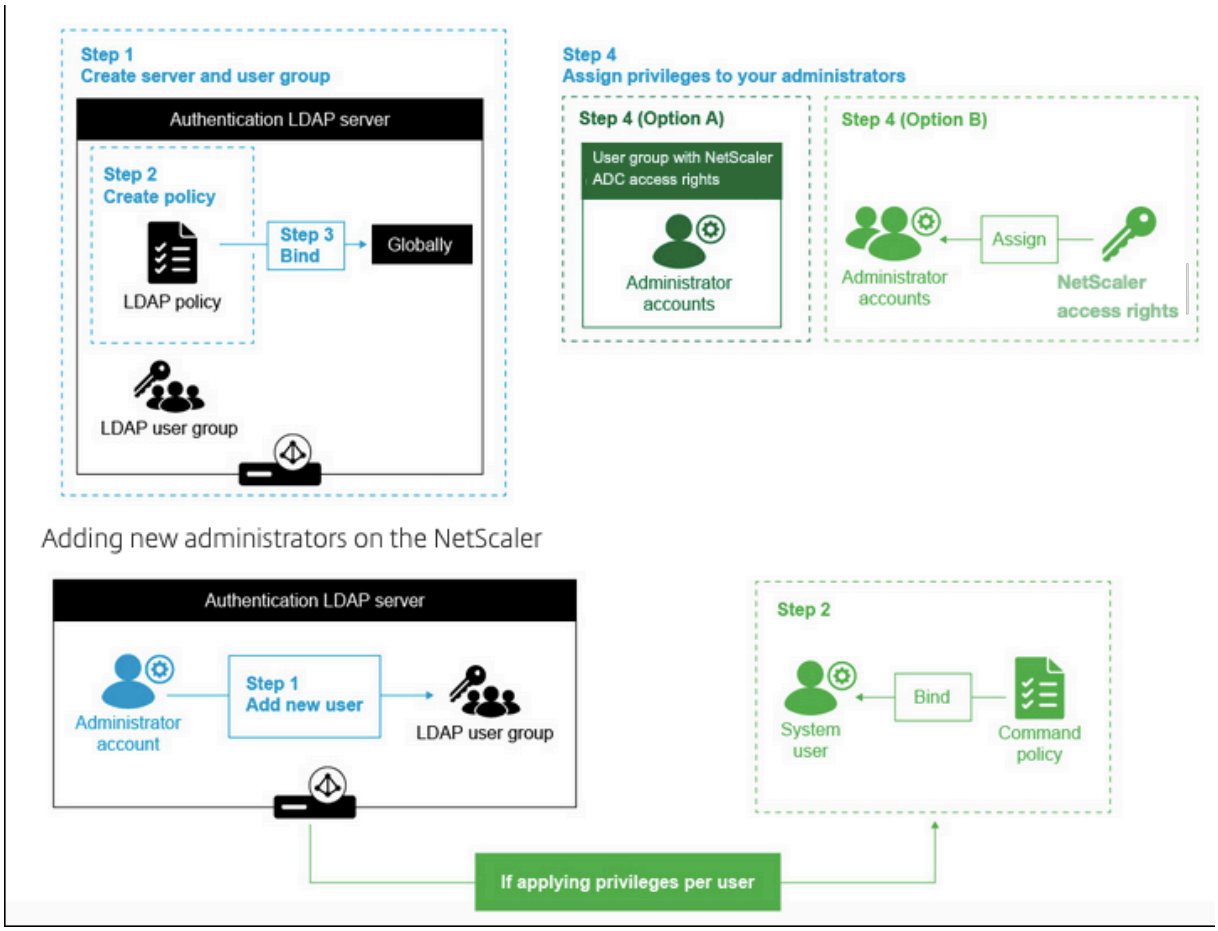
December 8, 2023

管理目的（スーパーユーザー、読み取り専用、ネットワーク権限など）を目的として、Active Directory 資格情報（ユーザー名とパスワード）を使用して NetScaler アプライアンスへのユーザーログオンを構成できます。

前提条件

- Windows Active Directory ドメインコントローラーサーバー
- NetScaler 管理者専用のドメイングループ
- NetScaler Gateway 10.1 以降のバージョン

次の図は、NetScaler アプライアンスでの LDAP 認証を示しています。



高レベルの設定手順

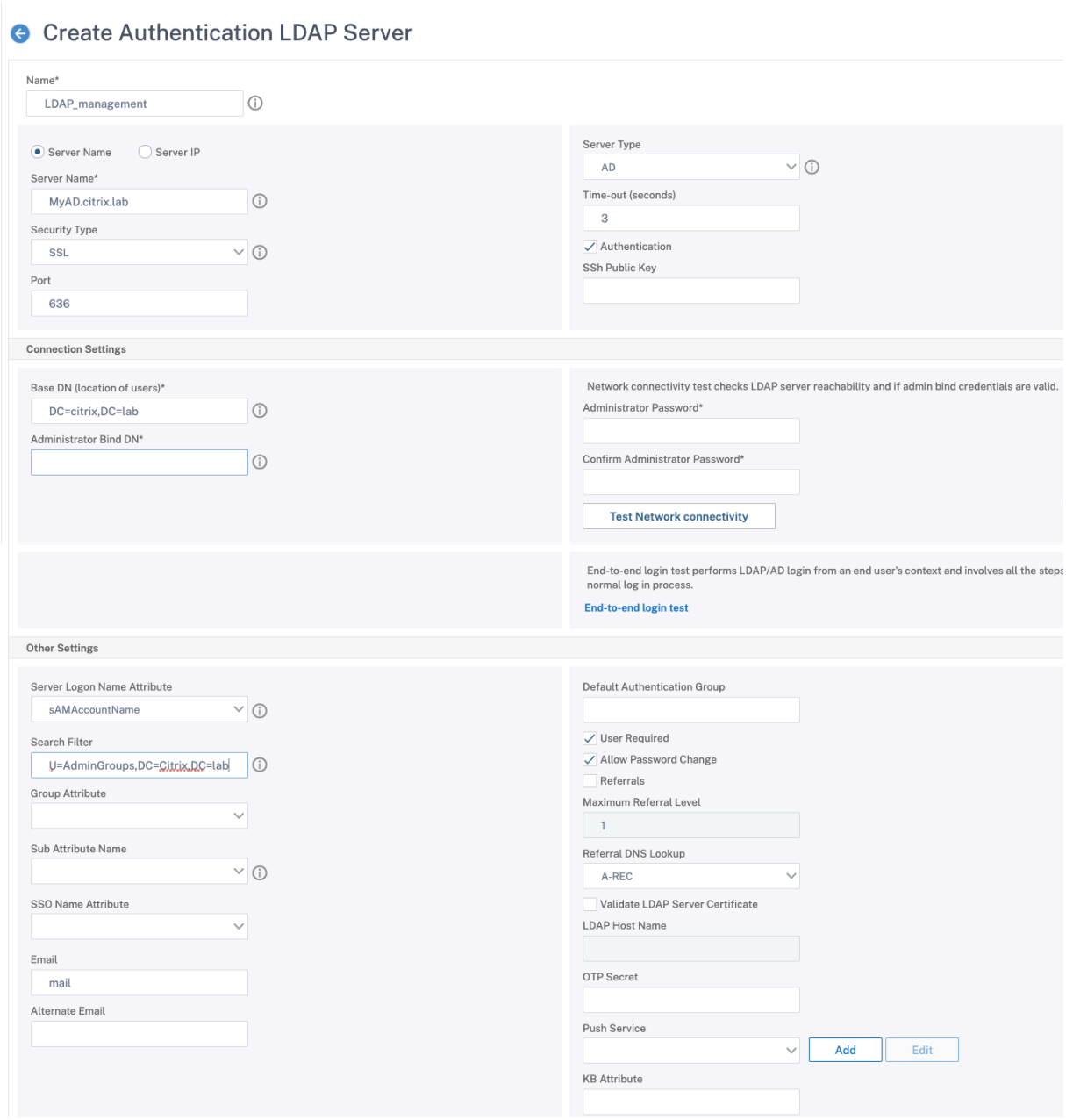
1. LDAP サーバーを作成する

2. LDAP ポリシーを作成する
3. LDAP ポリシーをバインドする
4. 次のいずれかの方法で、管理者に権限を割り当てます。

    - グループに対する権限の適用
    - ユーザーごとに権限を個別に適用する

認証 **LDAP** サーバーを作成する

1. [**System**] > [**Authentication**] > [**LDAP**] の順に選択します。
2. [サーバー] タブをクリックし、[追加] をクリックします。
3. 設定を完了し、[**Create**] をクリックします。



> 注:
>
> この例では、検索フィルターを設定してユーザーグループメンバーシップの認証をフィルタリングすることにより、アクセスは NetScaler アプライアンスに限定されます。この例で使用される値は、-&(memberOf=CN=NSG_admin、OU=AdminGroups、DC=Citrix、DC=lab) です。

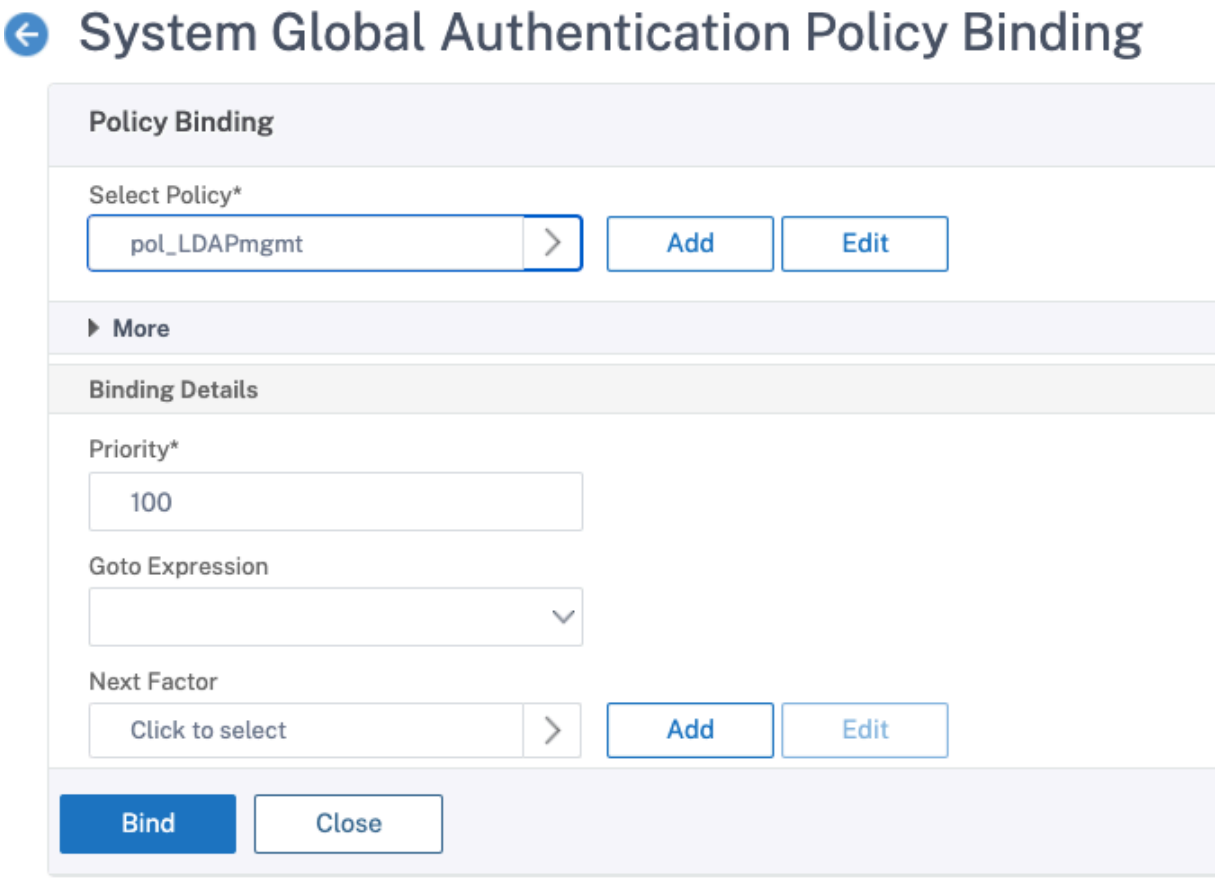
LDAP ポリシーを作成する

1. [システム] > [認証] > [詳細ポリシー] > [ポリシー] に移動します。
2. [追加] をクリックします。

3. ポリシーの名前を入力し、前の手順で作成したサーバを選択します。
4. [式] テキストフィールドに適切な式を入力し、[作成] をクリックします。

LDAP ポリシーをグローバルにバインドする

1. [システム] > [認証] > [詳細ポリシー] > [ポリシー] に移動します。
2. [認証ポリシー] ページで、[グローバルバインディング] をクリックします。
3. 作成したポリシー (この例では pol_ldapMgmt) を選択します。
4. 優先順位を適宜選択してください（数字が小さいほど優先度が高くなります）。
5. [バインド]、[完了] の順にクリックします。[グローバル境界] 列に緑色のチェックマークが表示されます。



管理者に権限を割り当てる

次の 2 つのオプションのいずれかを選択できます。

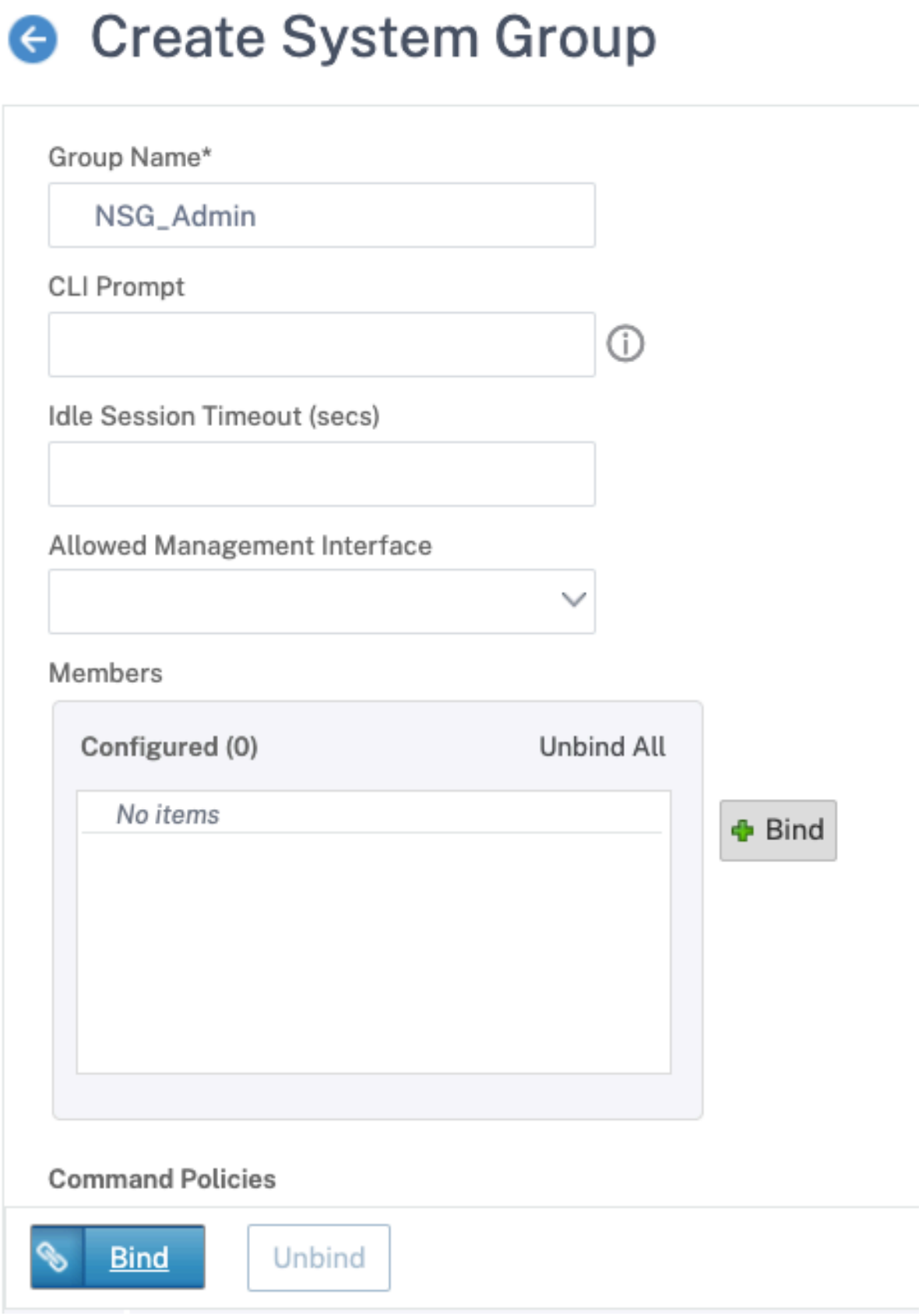
- グループに権限を適用： NetScaler アプライアンスにグループを追加し、このグループのメンバーである各ユーザーに同じアクセス権を割り当てます。

- ユーザーごとに権限を個別に適用する**:** 各ユーザー管理者アカウントを作成し、それぞれに権限を割り当てます。

グループに対する権限の適用

グループに権限を適用すると、検索フィルターで構成された Active Directory グループ（この例では NSG_Admin）のメンバーであるユーザーは、NetScaler Management インターフェイスに接続してスーパーユーザーコマンドポリシーを持つことができます。

1. [**System**] **>** [**User Administration**] **>** [**Groups**] の順に選択します。
2. 要件に従って詳細を入力し、[**Create**] をクリックします。



ユーザーが属する Active Directory グループと、ログイン時にアカウントに関連付ける必要があるコマンドポリシーレベルを定義しました。検索フィルタで設定した LDAP グループに、新しい管理者ユーザを追加できます。

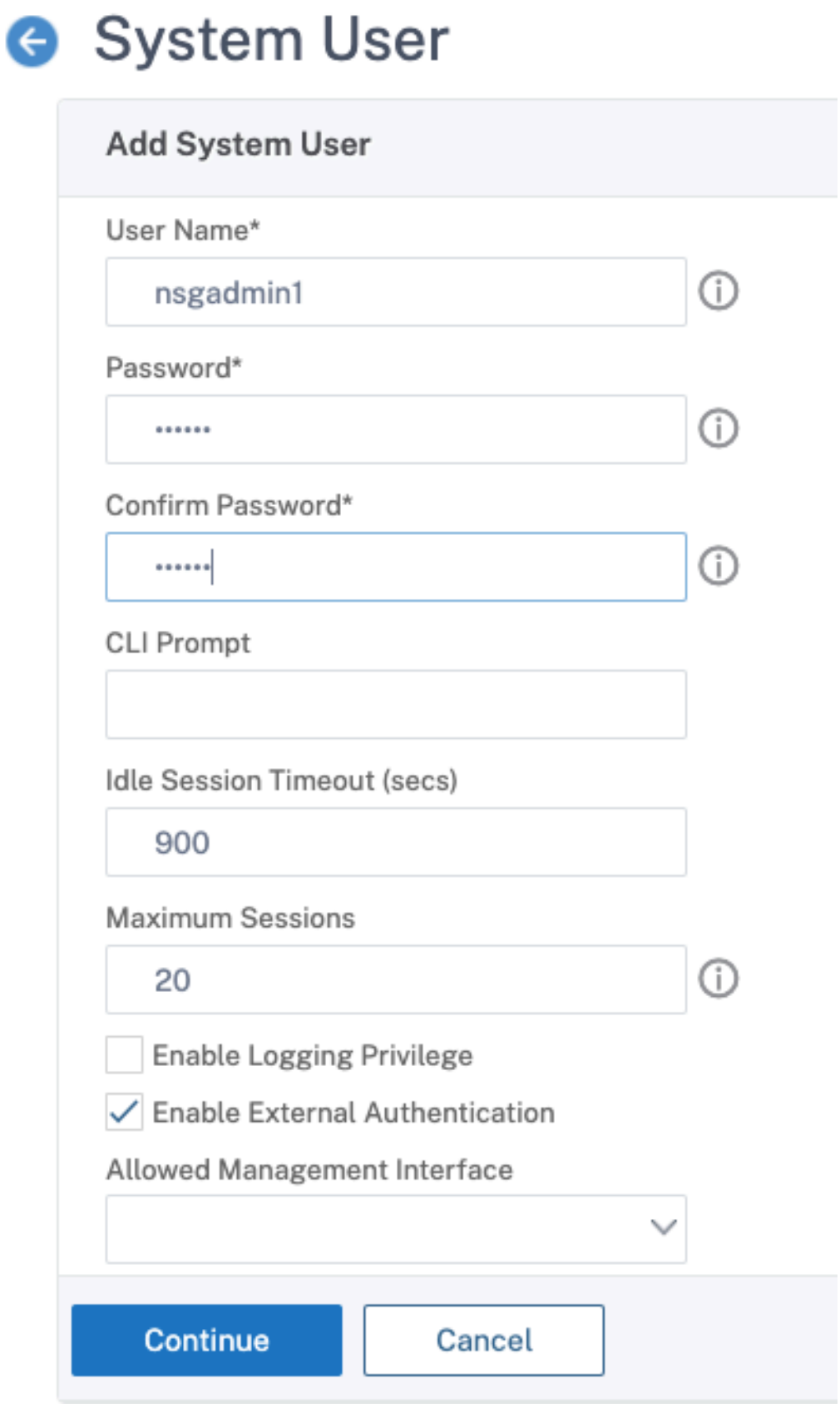
> 注**:**
>
> グループ名は Active Directory レコードと一致する必要があります。

ユーザーごとに権限を個別に適用する

このシナリオでは、検索フィルターで構成された Active Directory グループ（この例では NSG_Admin）のメンバーであるユーザーは、NetScaler 管理インターフェイスに接続できますが、NetScaler アプライアンスで特定のユーザーを作成してコマンドポリシーをそのユーザーにバインドするまでは権限がありません。

1. ［**System**］**>** ［**User Administration**］**>** ［**Users**］の順に選択します。
2. ［追加］をクリックします。
3. 要件に従って詳細を入力します。

   注**:** [外部認証を有効にする] を必ず選択してください。



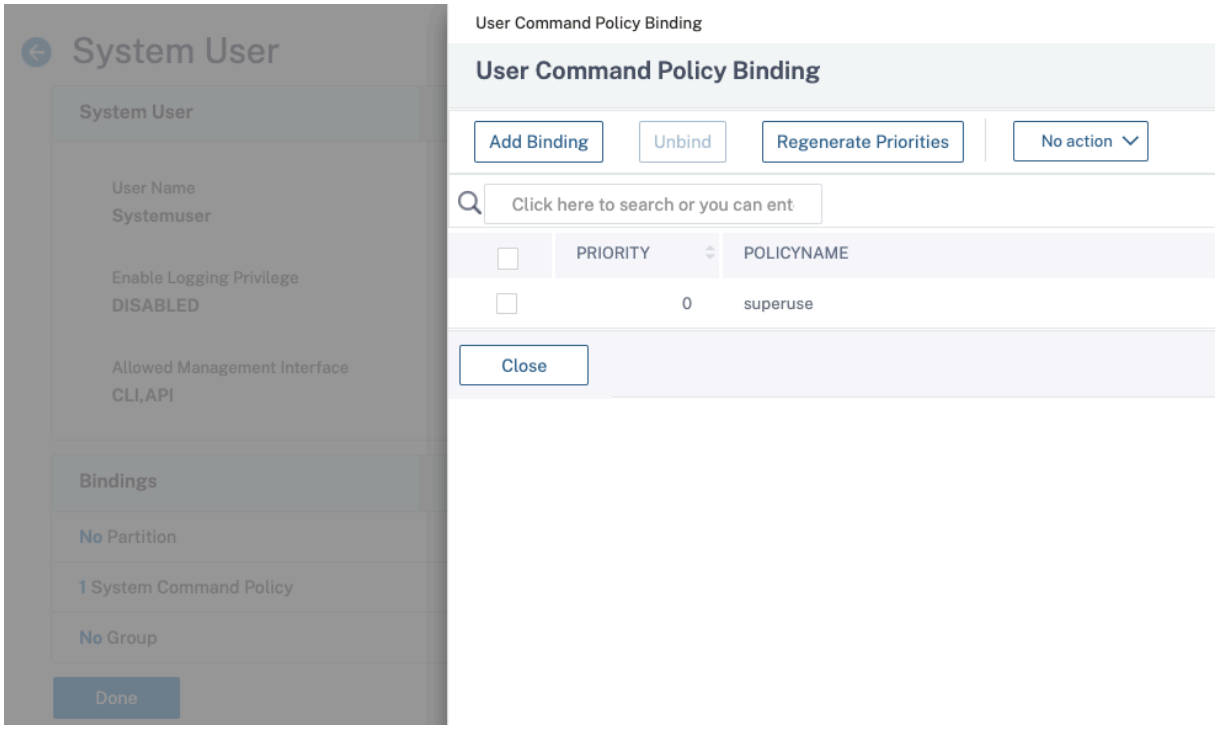
1. ［続行］をクリックします。

ログイン時にアカウントに関連付ける必要がある Active Directory ユーザーとコマンドポリシーレベルを定義しました。

> 注:
>
> - ユーザー名は、既存のユーザーの Active Directory レコードと一致する必要があります。
> - 外部認証のためにユーザーを NetScaler に追加する場合、外部認証が利用できない場合はパスワードを入力する必要があります。外部認証を正しく機能させるには、内部パスワードをユーザーアカウントの LDAP パスワードと同じにしないでください。

コマンドポリシーをユーザーに追加する

1. ［**System**］**>** ［**User Administration**］**>** ［**Users**］の順に選択します。
2. 作成したユーザーを選択し、[**Edit]** をクリックします。
3. [バインディング] で、[システムコマンドポリシー] をクリックします。
4. ユーザーに適用する正しいコマンドポリシーを選択します。
5. [バインド] をクリックして [閉じる] をクリックします。



管理者を追加するには

- 検索フィルタに設定した LDAP グループに管理者ユーザを追加します。
- NetScaler でシステムユーザーを作成し、正しいコマンドポリシーを割り当てます。

CLI を使用して管理目的で NetScaler アプライアンスの LDAP 認証を構成するには

次のコマンドを参考にして、NetScaler アプライアンス CLI でスーパーユーザー権限を持つグループのログオンを構成します。

1. LDAP サーバーを作成する

```
add authentication ldapAction LDAP_mgmt -serverIP myAD.citrix.lab
    -serverPort 636 -ldapBase "DC=citrix,DC=lab" -ldapBindDn
    readonly@citrix.lab -ldapBindDnPassword -ldapLoginName
    sAMAccountName -searchFilter "&(memberof=CN=NSG_Admin,OU=
    AdminGroups,DC=citrix,DC=lab)" -groupAttrName memberOf
<!--NeedCopy-->
```

2. 作成および LDAP ポリシー

```
add authentication ldapPolicy pol_LDAPmgmt ns_true LDAP_mgmt
<!--NeedCopy-->
```

3. LDAP ポリシーのバインド

```
bind system global pol_LDAPmgmt -priority 110
<!--NeedCopy-->
```

4. 管理者に権限を割り当てる

   - グループに権限を適用するには

```
add system group NSG_Admin
bind system group NSG_Admin -policyName superuser 100
<!--NeedCopy-->
```

   - ユーザーごとに個別に特権を適用するには

```
add system user admyoa
bind system user admyoa superuser 100
<!--NeedCopy-->
```

SSL を負荷分散仮想サーバーにオフロードした後の LDAP の設定

December 8, 2023

NetScaler アプライアンスでは、AAAD プロセスを使用して、管理アクセスまたは認証承認およびゲートウェイアクセス用の LDAP、RADIUS、TACACS などの基本認証を実行します。AAAD は管理 CPU で実行されるため、認証が断続的に失敗するという問題が発生する可能性があります。これらの障害を回避するには、負荷分散仮想サーバーを使用して SSL 機能を AAAD からオフロードできます。

SSL を負荷分散仮想サーバーにオフロードすることの利点

- AAAD のパフォーマンスが強化されました。AAAD では、SSL タイプの LDAP サーバへの認証要求ごとに、新しい SSL セッションが確立されます。AAAD プロセスは管理 CPU で実行されるため、SSL セッションを確立すると、AAAD への要求が多いときのパフォーマンスに影響します。SSL 機能を負荷分散仮想サーバーにオフロードすると、AAAD プロセスのパフォーマンスが向上します。
- クライアント証明書をサーバにレンダリングします。AAAD のクライアント LDAP ライブラリはサーバー証明書の検証のみを行い、クライアント証明書のサーバーへのレンダリングはサポートされていません。SSL 相互認証では、SSL 接続を確立するためにクライアント証明書をレンダリングする必要があるため、SSL 機能を負荷分散仮想サーバーにオフロードすることで、クライアント証明書をサーバーにレンダリングできます。

SSL を負荷分散仮想サーバーにオフロードした後に **LDAP** を設定します

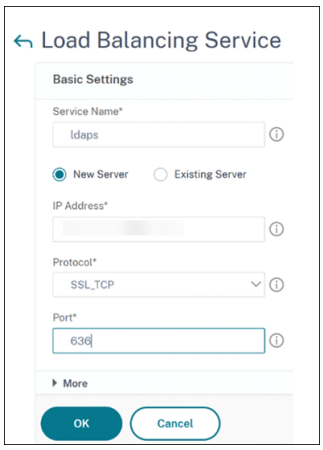
注**:** LDAP の負荷分散仮想サーバー IP アドレスを作成し、LDAP リクエストサーバーがその仮想サーバー IP アドレスを指定すると、トラフィックは SNIP から送信されます。

前提条件

- NetScaler アプライアンスが認証に使用するドメインコントローラーでセキュア LDAP が有効になっていることを確認します。デフォルトでは、エンタープライズ CA では、すべてのドメインコントローラーがドメインコントローラー証明書テンプレートを使用して証明書を登録します。
- ldp.exe を使用し、ポート 636 と SSL を介してドメインコントローラに接続して、安全な LDAP が機能していることを確認します。

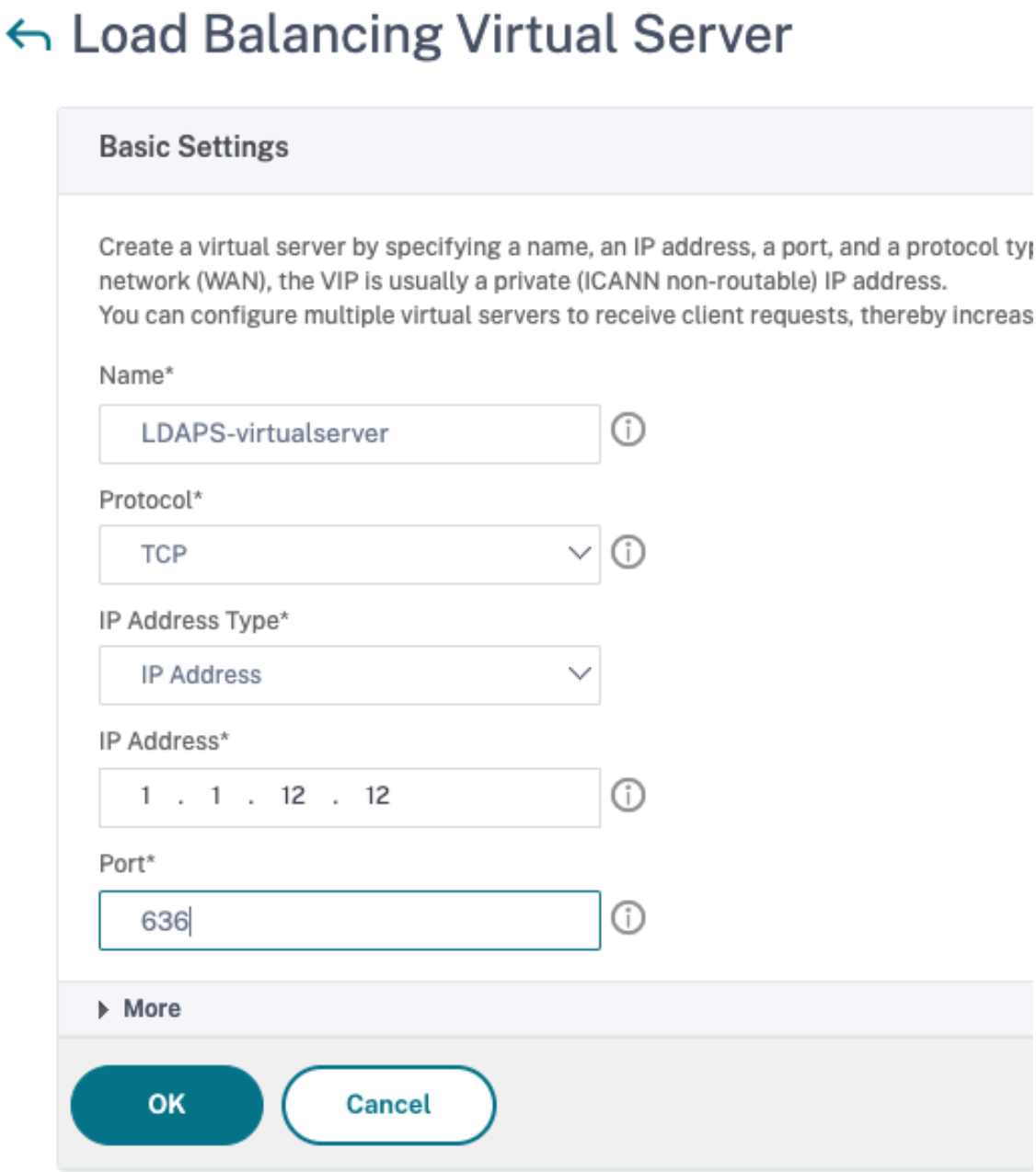
GUI を使用して **SSL** を負荷分散仮想サーバーにオフロードした後に **LDAP** を設定します

1. プロトコルを SSL_TCP に設定して負荷分散サービスを作成します。
    - [トラフィック管理**] > [**負荷分散**] > [**サービス] に移動し、[追加] をクリックします。
    - ドメインコントローラーの IP アドレスを指定し、ポート番号を 636 に設定します。
    - [**OK**] をクリックします。



2. LDAPS 負荷分散サービス用の負荷分散仮想サーバーを作成します。

   a) [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動します。
   b) プロトコルを TCP に設定し、IP アドレスを入力し、ポートを 636 に設定して、**OK** をクリックします。



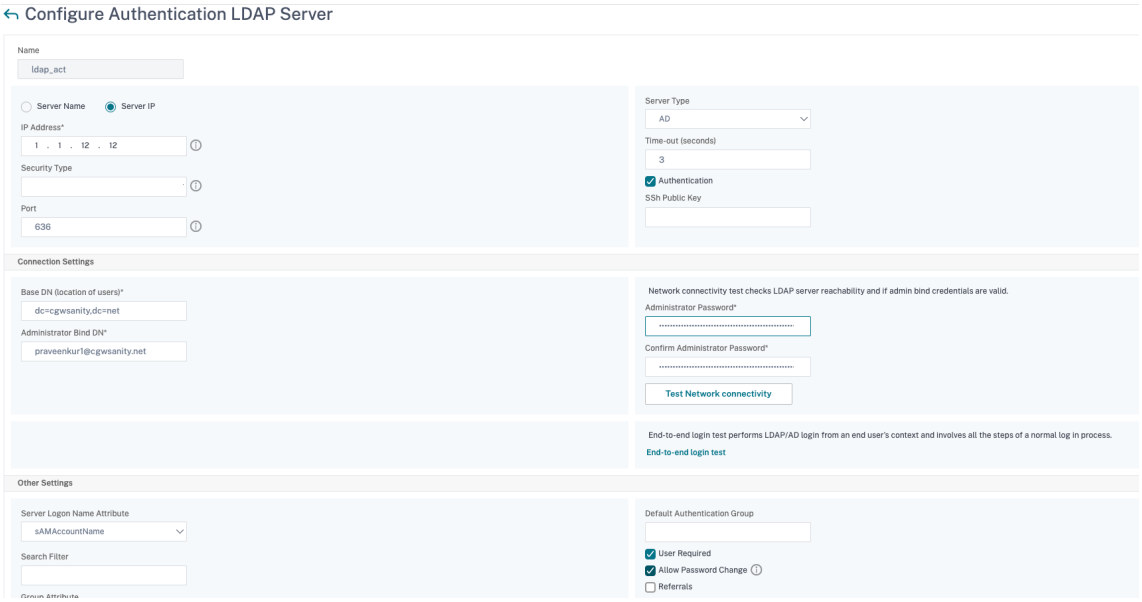
3. LDAPS サービスを負荷分散仮想サーバーにバインドします。

   - **Traffic Management > Load Balancing > Virtual Servers** に移動します。
   - LDAP 仮想サーバーを選択します。「負荷分散仮想サーバー」ページが表示されます。
   - 「サービスとサービスグループ」セクションで、「負荷分散仮想サーバーサービスバインディングなし」をクリックします。「サービスバインディング」ページが表示されます。
   - 負荷分散サービスを選択します。他の必須フィールドを更新し、「バインド」をクリックします。
   - ［完了］をクリックします。

4. 次に、LDAP 認証ポリシーサーバーを、セキュア LDAP の負荷分散仮想サーバーを指すように変更します。セ

キュリティタイプは PLAINTEXT でなければなりません。

a) **NetScaler Gateway >** ポリシー **> 認証 > LDAP** に移動します。
b) LDAP サーバーを選択し、[編集] をクリックします。
c) IP アドレスを、以前に作成した NetScaler アプライアンスでホストされている LDAPS VIP に変更します。
d) セキュリティタイプを **PLAINTEXT** に変更し、ポートを 636 に変更し、必要に応じて [パスワードの変更を許可する] チェックボックスを選択します (SLDAP ではパスワードを変更できます)。
e) [ネットワーク接続のテスト] をクリックして接続を確認します。
f) [**OK**] をクリックします。



LDAP サーバーのステータスが UP になっていることを確認できます。また、認証ログをチェックして、認証が意図したとおりに機能していることを確認します。

CLI を使用して SSL を負荷分散仮想サーバーにオフロードした後、LDAP を構成します

1. AAAD プロセス用の LDAP サーバを設定します。次の構成例では、SSL 相互認証なしで負荷分散仮想サーバーと SSL 接続を確立します。

```
add authentication ldapAction ldap_act -serverIP 1.1.12.12 -
    serverPort 636 -secTYPE PLAINTEXT -ldapBase "dc=aaatm-test,dc=
    com" -ldapBindDn administrator@aaatm-test.com -
    ldapBindDnPassword <password> -ldapLoginName samAccountName
<!--NeedCopy-->
```

2. LDAP 仮想サーバーの負荷分散仮想サーバーを構成します。負荷分散仮想サーバーのタイプは TCP です。

```
add lb vserver ldaps TCP 1.1.1.12 636 -persistenceType NONE -
    cltTimeout 9000
```



```
<!--NeedCopy-->
```

3. 負荷分散仮想サーバーのサービスを構成します。サービスタイプは SSL-TCP です。

```
add service ldaps 1.1.10.1 SSL_TCP 636
<!--NeedCopy-->
```

4. サービスの CA 証明書を設定し、サーバー証明書の検証用に「ServerAuth」パラメーターを設定します。

```
bind ssl service ldaps -certkeyName ca-cert -CA
set ssl service ldaps -serverAuth enabled
<!--NeedCopy-->
```

5. LDAP サーバにレンダリングされるサービスに証明書を添付します。

```
bind ssl service ldaps -certkeyName usr_cert [client-certificate
   for client-authentication]
<!--NeedCopy-->
```

6. サービスを負荷分散仮想サーバーにバインドします。

```
bind lb vserver ldaps ldaps
<!--NeedCopy-->
```

RADIUS 認証

August 15, 2023

他の種類の認証ポリシーと同様に、リモート認証ダイヤルインユーザーサービス (RADIUS) 認証ポリシーは、式とアクションで構成されます。認証ポリシーを作成したら、それを認証仮想サーバーにバインドし、プライオリティを割り当てます。バインドするときは、プライマリポリシーまたはセカンダリポリシーとして指定します。ただし、RADIUS 認証ポリシーの設定には、以下に説明する特定の特別な要件があります。

通常、認証時に認証サーバーの IP アドレスを使用するように NetScaler を構成します。RADIUS 認証サーバでは、ユーザ認証に IP アドレスの代わりに RADIUS サーバの FQDN を使用するように ADC を設定できるようになりました。FQDN を使用すると、認証サーバが複数の IP アドレスのいずれかにあっても、常に 1 つの FQDN を使用する環境で、より複雑な認証、認可、および監査の設定を簡素化できます。IP アドレスではなくサーバーの FQDN を使用して認証を構成するには、認証アクションの作成時を除き、通常の構成プロセスに従います。アクションを作成するときは、**ServerIp** パラメーターの代わりに ServerName パラメーターを使用します。

NetScaler が RADIUS サーバーの IP または FQDN を使用してユーザーを認証するように構成するかを決定する前に、認証、承認、および監査を IP アドレスではなく FQDN で認証するように構成すると、認証プロセスに余分な手順が追加されることを検討してください。ADC はユーザーを認証するたびに、FQDN を解決する必要があります。非常に多くのユーザーが同時に認証を試みると、DNS ルックアップの結果として認証プロセスが遅くなる可能性があります。

> 注
>
> これらの手順は、読者が既に RADIUS プロトコルに精通していて、選択した RADIUS 認証サーバをすでに設定していることを前提としています。

コマンドラインインターフェイスを使用して **RADIUS** サーバーの認証アクションを追加するには

RADIUS サーバへの認証を行う場合は、明示的な認証アクションを追加する必要があります。これを行うには、コマンドプロンプトで次のコマンドを入力します。

```
add authentication radiusAction <name> [-serverip <IP> | -serverName] <
    FQDN>][-serverPort <port>] [-authTimeout <positive_integer>] {
 -radKey  }
  [-radNASip ( ENABLED | DISABLED )][-radNASid <string>] [-radVendorID
     <positive_integer>][-radAttributeType <positive_integer>][-
     radGroupsPrefix <string>] [-radGroupSeparator <string>][-
     passEncoding <passEncoding>][-ipVendorID <positive_integer>] [-
     ipAttributeType <positive_integer>][-accounting ( ON | OFF )][-
     pwdVendorID <positive_integer> [-pwdAttributeType <
     positive_integer>]] [-defaultAuthenticationGroup <string>] [-
     callingstationid ( ENABLED | DISABLED )]

<!--NeedCopy-->
```

次の例では、Authn-Act-1 という名前の RADIUS 認証アクションを追加しています。このアクションには、サーバー IP が **10.218.24.65**、サーバーポート **1812**、認証タイムアウト **15** 分、RADIUS キー **WareThelorAx** 、NAS IP は無効になっており、NAS ID は NAS1 です。

```
add authentication radiusaction Authn-Act-1 -serverip 10.218.24.65 -
    serverport 1812 -authtimeout 15 -radkey WareTheLorax -radNASip
    DISABLED -radNASid NAS1
Done

<!--NeedCopy-->
```

次の例では同じ RADIUS 認証アクションを追加していますが、IP の代わりにサーバー FQDN **rad01.example.com** を使用しています。

```
add authentication radiusaction Authn-Act-1 -serverName rad01.example.
    com -serverport 1812 -authtimeout 15 -radkey WareTheLorax -radNASip
    DISABLED -radNASid NAS1
Done

<!--NeedCopy-->
```

コマンドラインを使用して外部 **RADIUS** サーバーの認証アクションを設定するには

既存の RADIUS アクションを設定するには、コマンドプロンプトで次のコマンドを入力します。


```
set authentication radiusAction <name> [-serverip <IP> | -serverName] <
    FQDN>][-serverPort <port>] [-authTimeout <positive_integer>] {
 -radKey  }
  [-radNASip ( ENABLED | DISABLED )][-radNASid <string>] [-radVendorID
     <positive_integer>][-radAttributeType <positive_integer>][-
     radGroupsPrefix <string>] [-radGroupSeparator <string>][-
     passEncoding <passEncoding>][-ipVendorID <positive_integer>] [-
     ipAttributeType <positive_integer>][-accounting ( ON | OFF )][-
     pwdVendorID <positive_integer> [-pwdAttributeType <
     positive_integer>]] [-defaultAuthenticationGroup <string>] [-
     callingstationid ( ENABLED | DISABLED )]

<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して外部 **RADIUS** サーバーの認証アクションを削除するには

既存の RADIUS アクションを削除するには、コマンドプロンプトで次のコマンドを入力します。

```
rm authentication radiusAction <name>

<!--NeedCopy-->
```

例

```
rm authentication radiusaction Authn-Act-1
Done

<!--NeedCopy-->
```

構成ユーティリティを使用して **RADIUS** サーバーを構成するには

> 注
>
> 構成ユーティリティでは、アクションの代わりにサーバーという用語が使用されますが、同じタスクを指します。

1. [セキュリティ] **> [AAA-**アプリケーショントラフィック] **> [**ポリシー] **> [**認証] **> [Radius**] に移動します。
2. 詳細ウィンドウの [サーバー] タブで、次のいずれかの操作を行います。
   - 新しい RADIUS サーバーを作成するには、「追加」をクリックします。
   - 既存の RADIUS サーバを変更するには、サーバを選択し、[編集] をクリックします。
3. 「認証 **RADIUS** サーバーの作成」または「認証 **RADIUS** サーバーの設定」ダイアログで、パラメーターの値を入力または選択します。[発信ステーション **ID** を送信] の下に表示されるパラメータを入力するには、[詳細] を展開します。
   - name* —RADIUSActionName (以前に設定されたアクションでは変更できません)

- 認証タイプ *—認証タイプ（RADIUS に設定、変更不可）
- サーバー名/IP アドレス *-サーバー名またはサーバー IP を選択してください
  - Server Name*—serverName <FQDN>
  - IP アドレス *—サーバー IP<IP> サーバーに IPv6 IP アドレスが割り当てられている場合は、IPv6 チェックボックスを選択します。
- Port* —サーバーポート
- タイムアウト (秒) * —認証タイムアウト
- シークレットキー *：RAD キー（RADIUS 共有シークレット）
- シークレットキーの確認 *-RADIUS 共有シークレットをもう一度入力します。(対応するコマンドラインはありません。)
- 発信側ステーション ID の送信—発信側ステーション ID
- グループベンダー識別子—ラベンダー ID
- グループ属性タイプ-RAD 属性タイプ
- IP アドレスベンダー識別子—IP ベンダー ID
- PWD ベンダー ID —PWD ベンダー ID
- パスワードエンコーディング—パッセンジャーコーディング
- デフォルト認証グループ—デフォルト認証グループ
- NAS ID —ランダサイド
- NAS IP アドレス抽出を有効にする—RADNAS IP
- グループプレフィックス—RAD グループプレフィックス
- グループセパレーター—RAD グループセパレーター
- IP アドレス属性タイプ—IP 属性タイプ
- パスワード属性タイプ—PWD 属性タイプ
- アカウンティング-アカウンティング

4. [作成] または [**OK**] をクリックします。作成したポリシーは、「サーバー」ページに表示されます。

RADIUS 属性 **66**（トンネル**-**クライアント**-**エンドポイント）のパススルーをサポート

NetScaler アプライアンスは、RADIUS 認証中に RADIUS 属性 66（トンネル-クライアント-エンドポイント）のパススルーを許可するようになりました。この機能を適用すると、クライアントの IP アドレスは委託先から二要素認証で受け取られ、リスクベースの認証決定が行われます。

「認証 RADIUSAction を追加」コマンドと「RadiusParams を設定」コマンドの両方に新しい属性「TunnelEndpointClientIp」が導入されました。

この機能を使用するには、**NetScaler** アプライアンスのコマンドプロンプトで次のように入力します。

```
add authentication radiusAction <name> {
 -serverIP <ip_addr|ipv6_addr|*> | {
 -serverName <string> }
  }
  [-serverPort <port>] … [-tunnelEndpointClientIP (ENABLED|DISABLED)]

set radiusParams {
 -serverIP <ip_addr|ipv6_addr|*> |{
 -serverName <string> }
  }
  [-serverPort<port>] … [-tunnelEndpointClientIP(ENABLED|DISABLED)]

<!--NeedCopy-->
```

例

```
add authentication radiusAction radius -severIP 1.217.22.20 -serverName
     FQDN -serverPort 1812 -tunnelEndpointClientIp ENABLED

set radiusParams -serverIp 1.217.22.20 -serverName FQDN1 -serverPort
    1812 -tunnelEndpointClientIP ENABLED

<!--NeedCopy-->
```

エンドツーエンドの **RADIUS** 認証の検証のサポート

NetScaler アプライアンスは、GUI を使用してエンドツーエンドの RADIUS 認証を検証できるようになりました。この機能を検証するために、GUI に新しい「テスト」ボタンが導入されました。NetScaler アプライアンスの管理者は、この機能を活用して次のメリットを得ることができます。

- 完全なフロー（パケットエンジン-aaa デーモン-外部サーバ）を統合し、より優れた分析を提供します。
- 個々のシナリオに関連する問題の検証とトラブルシューティングにかかる時間を短縮

GUI を使用して RADIUS エンドツーエンド認証のテスト結果を設定および表示するには、2 つのオプションがあります。

[システムから**]** オプション

1. [システム**] > [**認証**] > [**基本ポリシー**] > [RADIUS]** に移動し、 [サーバ] タブをクリックします。
2. リストから、使用可能な **RADIUS** アクションを選択します。
3.「認証 **RADIUS** サーバーの構成」ページの「接続設定」セクションには、2 つのオプションがあります。
4. RADIUS サーバ接続を確認するには、「**RADIUS** 到達可能性のテスト」タブをクリックします。

5. エンドツーエンドの RADIUS 認証を表示するには、「エンドユーザー接続のテスト」リンクをクリックします。

認証オプションから

1. [認証] **>** [ダッシュボード] に移動し、リストから使用可能な RADIUS アクションを選択します。
2. 「認証 **RADIUS** サーバーの構成」ページの「接続設定」セクションには、2 つのオプションがあります。
3. RADIUS サーバ接続を確認するには、「**RADIUS** 到達可能性のテスト」タブをクリックします。
4. エンドツーエンドの RADIUS 認証ステータスを表示するには、「エンドユーザー接続のテスト」リンクをクリックします。

TCP または TLS を使用した RADIUS 認証

December 8, 2023

リリース 13.1 ～27.59 以降、RADIUS 認証は TCP および TLS プロトコルでもサポートされています。

> 注:
>
> - **RADIUS** 到達可能性テストオプションは、TCP および TLS 転送タイプの RADIUS ではサポートされていません。
> - UDP を使用した RADIUS 認証は FIPS アプライアンスではサポートされていません。

CLI を使用して RADIUS を TCP 経由で構成する

コマンドプロンプトで次のように入力します:

```
add authentication radiusAction <name> [-serverIP] [-serverPort ] [-
   transport <transport>]
<!--NeedCopy-->
```

例:

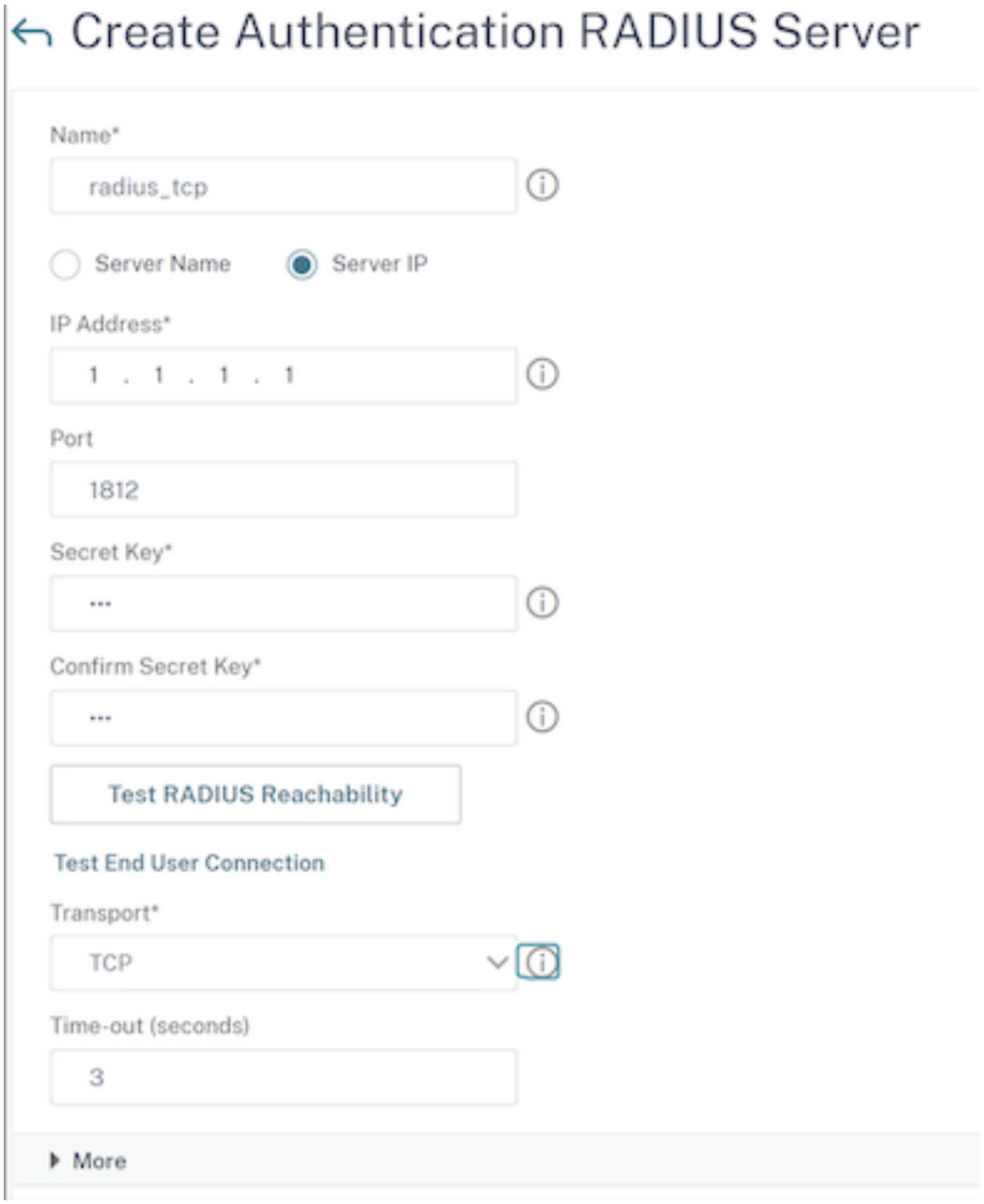
```
add authentication radiusAction RadAction -serverIP 1.1.1.1 -radkey 123
    -transport TCP
<!--NeedCopy-->
```

GUI を使用して TCP 経由の RADIUS を構成する

1. セキュリティ **> AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** アクション **> RADIUS** に移動します。

2. 既存のサーバーを選択するか、サーバーを作成します。

   サーバーの作成の詳細については、「GUI を使用して RADIUS サーバーを構成するには」を参照してください。



3. [トランスポート] で [**TCP**] を選択します。
4. ［作成］をクリックします。

CLI を使用して **TLS** 経由の **RADIUS** を構成する

コマンドプロンプトで次のように入力します:


```
add authentication radiusAction <name> [-serverIP] [-serverPort ] [-
    transport <transport>] [-targetLBVserver <string>]
<!--NeedCopy-->
```

例

```
add authentication radiusAction RadAction -serverIP 1.1.1.1 -radkey 123
    -transport TLS -targetLBVserver rad-lb
<!--NeedCopy-->
```

> 注:
>
> - サーバー名は TLS トランスポートタイプではサポートされていません。
> - TLS トランスポートタイプの場合は、TCP タイプのターゲット負荷分散仮想サーバーを構成し、タイプ SSL_TCP のサービスをこの仮想サーバーにバインドします。
> - RADIUS アクション用に構成された IP アドレスとポート番号は、構成されたターゲット負荷分散仮想サーバーの IP アドレスとポート番号と一致する必要があります。

GUI を使用して TLS 経由の RADIUS を構成する

1. セキュリティ **>** **AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** アクション **>** サーバに移動します。
2. 既存のサーバーを選択するか、サーバーを作成します。

   サーバーの作成の詳細については、「GUI を使用して RADIUS サーバーを構成するには」を参照してください。
3. [トランスポート] で [**TLS**] を選択します。
4. [ターゲット負荷分散仮想サーバー] で、仮想サーバーを選択します。負荷分散仮想サーバーの作成の詳細については、「仮想サーバーの作成」を参照してください。

   > 注:
   >
   > - サーバー名は TLS トランスポートタイプではサポートされていません。
   > - TLS トランスポートタイプの場合は、TCP タイプのターゲット負荷分散仮想サーバーを構成し、タイプ SSL_TCP のサービスをこの仮想サーバーにバインドします。
   > - RADIUS アクション用に構成された IP アドレスとポート番号は、構成されたターゲット負荷分散仮想サーバーの IP アドレスとポート番号と一致する必要があります。
5. 「作成」をクリックします。

TACACS 認証

August 15, 2023

TACACS 認証ポリシーは、外部のターミナルアクセスコントローラアクセス制御システム（TACACS）認証サーバに対して認証します。
ユーザーが TACACS サーバーで認証されると、NetScaler は同じ TACACS サーバーに接続してその後のすべての認証を行います。この機能により、プライマリ TACACS サーバが使用できなくなったときに、ADC が最初の TACACS サーバがタイムアウトするまで待機する間の遅延を防ぐことができます。これは、認可要求を 2 番目の TACACS サーバに再送信する前に発生します。

> 注:
>
> TACACS 認可サーバは、文字列長が 255 文字を超えるコマンドをサポートしていません。
>
> 回避策： TACACS 認証サーバの代わりにローカル認証を使用してください。

TACACS サーバを介して認証する場合、認証、承認、およびトラフィック管理ログの監査では、TACACS コマンドだけが正常に実行されます。これにより、実行権限のないユーザによって入力された TACACS コマンドがログに表示されなくなります。

NetScaler 12.0 Build 57.x 以降、ターミナルアクセスコントローラーのアクセス制御システム（TACACS）は、TACACS リクエストの送信中に認証、承認、および監査デーモンをブロックしません。LDAP 認証と RADIUS 認証を許可してリクエストを続行します。TACACS サーバが TACACS 要求を確認すると、TACACS 認証要求が再開されます。

> 重要:
>
> - 「clear ns config」コマンドを実行するときは、TACACS 関連の設定を変更しないことをお勧めします。
> - 詳細ポリシーの「clear ns config」コマンドで「RBAConfig」パラメータが NO に設定されている場合、詳細ポリシーに関連する TACACS 関連の設定はクリアされ、再適用されます。

TACACS 認証用の名前/値属性のサポート

TACACS 認証属性に一意の名前と値を設定できるようになりました。名前は TACACS アクションパラメータで設定され、値は名前を問い合わせることによって取得されます。name 属性値を指定すると、管理者は属性名に関連付けられた属性値を簡単に検索できます。また、管理者は属性を値だけで覚える必要がなくなりました。

> 重要
>
> - TACACSaction コマンドでは、最大 64 個の属性をカンマで区切って設定でき、合計サイズは 2048 バイト未満です。

CLI を使用して名前と値の属性を設定するには

コマンドプロンプトで入力します。

```
add authentication tacacsAction <name> [-Attributes <string>]
<!--NeedCopy-->
```

例**:**

```
add authentication tacacsAction tacacsAct1 -attributes  “mail,sn,
   userprincipalName”
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して認証アクションを追加するには

LOCAL 認証を使用しない場合は、明示的な認証アクションを追加する必要があります。コマンドプロンプトで、次のコマンドを入力します。

```
add authentication tacacsAction <name> -serverip <IP> [-serverPort <
   port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

例

```
add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
   serverport 1812                    -authtimeout 15 -tacacsSecret "
   minotaur" -authorization OFF -accounting ON -auditFailedCmds OFF -
   defaultAuthenticationGroup "users"
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して認証アクションを設定するには

既存の認証アクションを構成するには、コマンドプロンプトで次のコマンドを入力します。

```
set authentication tacacsAction <name> -serverip <IP> [-serverPort <
   port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

例

```
     > set authentication tacacsaction Authn-Act-1 -serverip
        10.218.24.65 -serverport 1812                    -authtimeout 15
        -tacacsSecret "minotaur" -authorization OFF -accounting ON -
        auditFailedCmds OFF -defaultAuthenticationGroup "users"    Done
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して認証アクションを削除するには

既存の RADIUS アクションを削除するには、コマンドプロンプトで次のコマンドを入力します。

```
rm authentication radiusAction <name>
<!--NeedCopy-->
```

例

```
rm authentication tacacsaction Authn-Act-1
<!--NeedCopy-->
```

クライアント証明書認証

December 8, 2023

オンラインバンキングの Web サイトや従業員の個人情報を含む Web サイトなど、機密性の高いコンテンツを含む Web サイトでは、認証のためにクライアント証明書が必要になる場合があります。クライアント側の証明書属性に基づいてユーザを認証するように認証、承認、および監査を構成するには、まずトラフィック管理仮想サーバでクライアント認証を有効にし、ルート証明書を認証仮想サーバにバインドします。次に、2 つのオプションのいずれかを実装します。認証仮想サーバーのデフォルトの認証タイプを CERT として構成することも、NetScaler がクライアント証明書に基づいてユーザーを認証するために何をする必要があるかを定義する証明書アクションを作成することもできます。いずれの場合も、認証サーバーは CRL をサポートしている必要があります。クライアント証明書の **subjectCN** フィールドまたは別の指定されたフィールドからユーザー名を抽出するように ADC を構成します。

認証ポリシーが構成されておらず、グローバルカスケードが構成されていない認証仮想サーバーにユーザーがログオンしようとすると、証明書の指定されたフィールドからユーザー名情報が抽出されます。必須フィールドが抽出されると、認証は成功します。SSL ハンドシェイク中にユーザーが有効な証明書を提供しなかった場合、またはユーザー名の抽出が失敗した場合、認証は失敗します。クライアント証明書を検証した後、ADC はユーザーにログオンページを表示します。

次の手順は、機能する認証、承認、および監査構成がすでに作成されていることを前提としており、クライアント証明書を使用して認証を有効にする方法のみを説明しています。また、これらの手順では、ルート証明書とクライアント証明書を取得し、ADC の /nsconfig/ssl ディレクトリに配置したことを前提としています。

クライアント証明書認証の構成

GUI を使用してクライアント証明書パラメータを設定

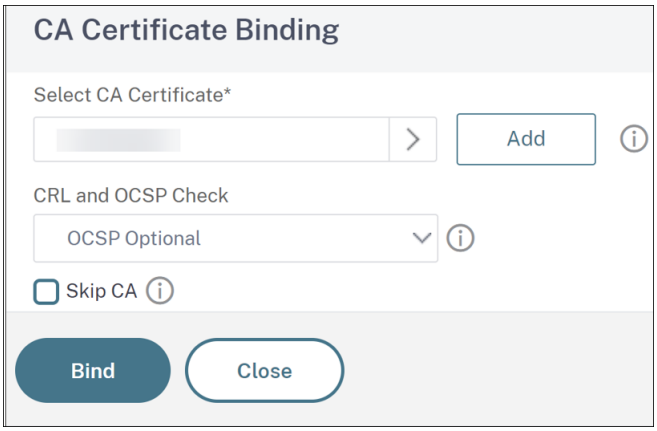
1. CA 証明書をインストールし、認証仮想サーバーにバインドします。

    a) ［セキュリティ］ **>** ［**AAA-**アプリケーショントラフィック］ **>** ［仮想サーバ］に移動します。

b) 表示される「認証仮想サーバー」ページで、クライアント証明書認証を処理するように構成する仮想サーバーを選択し、「編集」をクリックします。

c)「認証仮想サーバー」ページで、「証明書」セクションに移動し、右矢印「>」をクリックします。

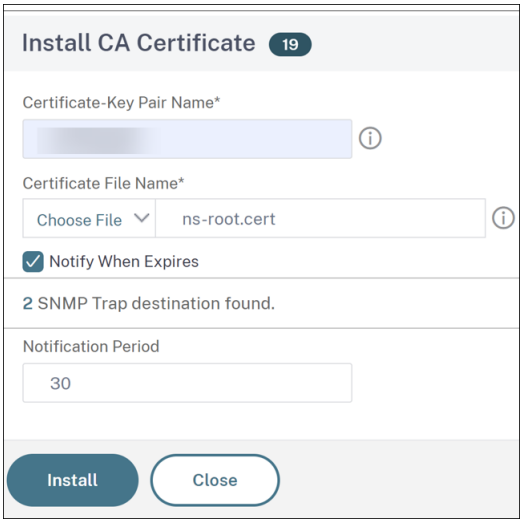
d)「**CA** 証明書バインディング」ページで、CA 証明書を選択し、他の必須フィールドを更新して、「バインド」をクリックします。



e) CA 証明書がない場合は、[追加] を選択します。

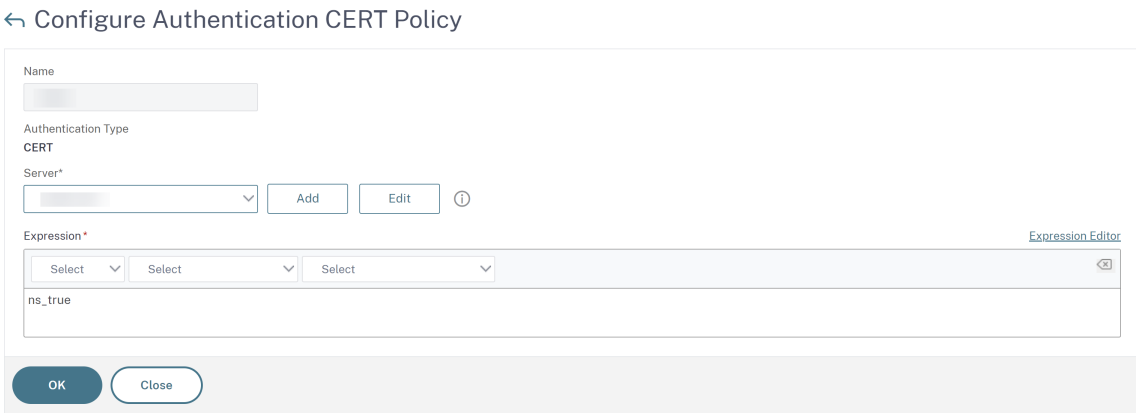
f)「証明書のインストール」ページで、次のフィールドを更新して「インストール」をクリックし、「閉じる」をクリックします。

- 証明書とキーのペア名: 証明書と秘密鍵のペアの名前
- 証明書ファイル名: 証明書とキーのペアを形成するために使用される証明書ファイルの名前。証明書ファイルは、NetScaler のハードディスクドライブまたはソリッドステートドライブにある必要があります。証明書をデフォルト以外の場所に保存すると、高可用性設定に不整合が生じる可能性があります。デフォルトのパスは /nsconfig/ssl/です。
- 通知期間: 証明書の有効期限が切れる前に、NetScaler が証明書の有効期限が近づいていることを管理者に通知する日数。
- 期限切れ時に通知: このオプションを有効にすると、証明書の有効期限が近づいたときにアラートを受け取ることができます。



g) CA 証明書がインストールされたら、**CA** 証明書バインディングページに移動し、認証仮想サーバーにバインドします。

2. **[**セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] ページに戻ります。
3. [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**認証**] > [**基本ポリシー**] > [CERT]** に移動します。
4. クライアント証明書認証を処理するように設定するポリシーを選択し、[編集] をクリックします。
5.「認証証明書ポリシーの設定」ページで、「サーバー」ドロップダウンリストに移動し、クライアント証明書認証を処理するように構成されている仮想サーバーを選択します。
6. [**OK**] をクリックします。



CLI を使用してクライアント証明書のパラメータを設定します

コマンドプロンプトで、次のコマンドを表示されている順序で入力して、証明書を構成し、構成を確認します。


```
add ssl certKey <certkeyName> -cert <certFile> -key <keyFile> -password
    -inform <inform> -expiryMonitor <expiryMonitor> -notificationPeriod
    <notificationPeriod>

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
   positive_integer>]

show ssl certKey [<certkeyName>]

set aaa parameter -defaultAuthType CERT

show aaa parameter

set aaa certParams -userNameField "Subject:CN"

show aaa certParams
<!--NeedCopy-->
```

GUI を使用してクライアント証明書の詳細認証ポリシーを設定

1. CA 証明書をインストールし、証明書とキーのペアにバインドします。

   a) ［セキュリティ］ **>** ［**AAA-**アプリケーショントラフィック］ **>** ［仮想サーバ］に移動します。
   b) 表示される「認証仮想サーバー」ページで、クライアント証明書認証を処理するように構成する仮想サーバーを選択し、「編集」をクリックします。
   c)「認証仮想サーバー」ページで、「証明書」セクションに移動し、右矢印「>」をクリックします。
   d)「**CA** 証明書バインディング」ページで、CA 証明書を選択し、他の必須フィールドを更新して、「バインド」をクリックします。
   e) CA 証明書がない場合は、[追加] を選択します。
   f)「証明書のインストール」ページで、次のフィールドを更新して「インストール」をクリックし、「閉じる」をクリックします。

      - 証明書とキーのペア名: 証明書と秘密鍵のペアの名前
      - 証明書ファイル名: 証明書とキーのペアを形成するために使用される証明書ファイルの名前。証明書ファイルは、NetScaler のハードディスクドライブまたはソリッドステートドライブにある必要があります。証明書をデフォルト以外の場所に保存すると、高可用性設定に不整合が生じる可能性があります。デフォルトのパスは /nsconfig/ssl/です。
      - 通知期間: 証明書の有効期限が切れる前に、NetScaler が証明書の有効期限が近づいていることを管理者に通知する日数。
      - 期限切れ時に通知: このオプションを有効にすると、証明書の有効期限が近づいたときにアラートを受け取ることができます。

   g) CA 証明書がインストールされたら、**CA** 証明書バインディングページに移動し、手順 4 を繰り返します。

2. **[**セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] ページに戻ります。

> 注:
>
> 仮想サーバーの有効な CA 証明書とサーバー証明書をインポートした場合は、手順 **1** と **2** をスキップできます。

3. セキュリティ **> AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシーに移動し、ポリシーを選択します。
4. 認証ポリシーページで、次のいずれかを実行します。
   - ポリシーを作成するには、[**Add**] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[編集] をクリックします。
5. 「認証ポリシーの作成」ページまたは「認証ポリシーの設定」ページで、パラメータの値を入力または選択します。
   - 名前: ポリシー名。以前に設定したポリシーの名前は変更できません。
   - アクションタイプ: 認証アクションのタイプ。
   - アクション: ポリシーが一致した場合に実行される認証アクションの名前。既存の認証アクションを選択するか、[追加] をクリックしてアクションを作成できます。
   - 表現: 指定したアクションを適用する接続を選択するルール。ルールは、シンプル (「true」はすべてのトラフィックを選択) または複雑にすることができます。式を入力するには、まず式ウィンドウの下の左端のドロップダウンリストで式の種類を選択し、次に式テキスト領域に式を直接入力するか、[追加] をクリックして [式の追加] ダイアログボックスを開き、その中のドロップダウンリストを使用して式を定義します。
   - ログアクション: 認証リクエストがこのポリシーに一致する場合に使用する監査アクションの名前。既存の監査アクションを選択するか、[追加] をクリックしてアクションを作成できます。
   - コメント: この認証ポリシーが適用されるトラフィックの種類を説明するコメントを入力できます。この情報は入力しなくても構いません。
6. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。

クライアント証明書パススルー

NetScaler は、ユーザー認証にクライアント証明書を必要とする保護対象アプリケーションにクライアント証明書を渡すように構成できるようになりました。ADC は最初にユーザーを認証し、次にクライアント証明書を要求に挿入してアプリケーションに送信します。この機能は、適切な SSL ポリシーを追加することによって構成されます。

ユーザーがクライアント証明書を提示したときのこの機能の正確な動作は、VPN 仮想サーバーの設定によって異なります。

- VPN 仮想サーバーがクライアント証明書を受け入れるように構成されているが、それを必要としない場合、ADC は証明書を要求に挿入し、保護されたアプリケーションに要求を転送します。

- VPN 仮想サーバーでクライアント証明書認証が無効になっている場合、ADC は認証プロトコルを再ネゴシエーションし、ユーザーを再認証してから、クライアント証明書をヘッダーに挿入し、要求を保護されたアプリケーションに転送します。
- VPN 仮想サーバーがクライアント証明書認証を要求するように構成されている場合、ADC はクライアント証明書を使用してユーザーを認証し、ヘッダーに証明書を挿入し、要求を保護されたアプリケーションに転送します。

いずれの場合も、クライアント証明書のパススルーを次のように構成します。

CLI を使用してクライアント証明書パススルーを作成および設定する

コマンドプロンプトで、次のコマンドを入力します:

```
add vpn vserver <name> SSL <IP> 443
<!--NeedCopy-->
```

name には、仮想サーバーの名前を置き換えます。名前には、1 ～ 127 の ASCII 文字が文字またはアンダースコア (_) で始まり、英字、数字、アンダースコア (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、ハイフン (-) のみを含む必要があります。<IP>の場合は、仮想サーバに割り当てられた IP アドレスを置き換えます。

```
set ssl vserver <name> -clientAuth ENABLED -clientCert <clientcert>
<!--NeedCopy-->
```

<name>では、作成した仮想サーバーの名前を置き換えます。<clientCert>には、次の値のいずれかを代入します。

- [disabled]: VPN 仮想サーバでのクライアント証明書認証を無効にします。
- [必須（mandatory）]: 認証にクライアント証明書を要求するように VPN 仮想サーバを設定します。
- optional: クライアント証明書認証を許可するが、要求しないように VPN 仮想サーバを設定します。

```
bind vpn vserver <name> -policy local
<!--NeedCopy-->
```

<name>で、作成した VPN 仮想サーバーの名前を置き換えます。

```
bind vpn vserver <name> -policy cert
<!--NeedCopy-->
```

<name>の場合は、作成した VPN 仮想サーバーの名前を置き換えます。

```
bind ssl vserver <name> -certkeyName <certkeyname>
<!--NeedCopy-->
```

<name>では、作成した仮想サーバーの名前を置き換えます。<certkeyName>では、クライアント証明書キーを代用します。


```
bind ssl vserver <name> -certkeyName <cacertkeyname> -CA -ocspCheck
    Optional
<!--NeedCopy-->
```

<name>では、作成した仮想サーバーの名前を置き換えます。<cacertkeyName>の場合は、CA 証明書キーを代用します。

```
add ssl action <actname> -clientCert ENABLED -certHeader CLIENT-CERT
<!--NeedCopy-->
```

<actname>では、SSL アクションの名前を置き換えます。

```
add ssl policy <polname> -rule true -action <actname>
<!--NeedCopy-->
```

<polname>では、新しい SSL ポリシーの名前を置き換えます。<actname>では、作成した SSL アクションの名前を置き換えます。

```
bind ssl vserver <name> -policyName <polname> -priority 10
<!--NeedCopy-->
```

<name>で、VPN 仮想サーバーの名前を置き換えます。

例

```
add vpn vserver vs-certpassthru SSL 10.121.250.75 443
set ssl vserver vs-certpassthru -clientAuth ENABLED -clientCert
    optional
bind vpn vserver vs-certpassthru -policy local
bind vpn vserver vs-certpassthru -policy cert
bind ssl vserver vs-certpassthru -certkeyName mycertKey
bind ssl vserver vs-certpassthru -certkeyName mycertKey -CA -ocspCheck
    Optional
add ssl action act-certpassthru -clientCert ENABLED -certHeader CLIENT-
    CERT
add ssl policy pol-certpassthru -rule true -action act-certpassthru
bind ssl vserver vs-certpassthru -policyName pol-certpassthru -priority
     10
<!--NeedCopy-->
```

認証のネゴシエート

August 15, 2023

他の種類の認証ポリシーと同様に、ネゴシエート認証ポリシーも表現とアクションで構成されます。認証ポリシーを作成したら、それを認証仮想サーバーにバインドし、プライオリティを割り当てます。バインドするときは、プライマリポリシーまたはセカンダリポリシーとして指定します。

標準の認証機能に加えて、Negotiate Action コマンドでは、情報を手動で入力する代わりに、キータブ・ファイルからユーザー情報を抽出できるようになりました。キータブに複数の SPN がある場合、認証、承認、および監査によって正しい SPN が選択されます。この機能は、コマンドラインまたは構成ユーティリティを使用して構成できます。

> 注
>
> 以下の手順は、LDAP プロトコルに精通していて、選択した LDAP 認証サーバーをすでに設定していることを前提としています。

コマンドラインインターフェイスを使用してキータブファイルからユーザー情報を抽出するように認証、承認、および監査を設定するには

コマンドプロンプトで、適切なコマンドを入力します。

```
add authentication negotiateAction <name> {
 -domain <string> }
  {
 -domainUser <string> }
  {
 -domainUserPasswd  }
  [-defaultAuthenticationGroup <string>] [-keytab <string>] [-NTLMPath
     <string>]

set authentication negotiateAction <name> {
 -domain <string> }
  {
 -domainUser <string> }
  {
 -domainUserPasswd }
  [-defaultAuthenticationGroup <string>] [-keytab <string>] [-NTLMPath
     <string>]
<!--NeedCopy-->
```

Parameter description

- 名前 -使用するネゴシエートアクションの名前。
- ドメイン -NetScaler を表すサービスプリンシパルのドメイン名。
- **DomainUser** -NetScaler プリンシパルにマップされているアカウントのユーザー名。キータブファイルが利用できない場合は、ドメインとパスワードと一緒に指定できます。キータブファイルと一緒にユーザー名を指定すると、そのキータブファイルでこのユーザーの認証情報が検索されます。最大長：127
- **DomainUserPasswd** -NetScaler プリンシパルにマップされているアカウントのパスワード。
- **DefaultAuthenticationGroup** -抽出されたグループに加えて認証が成功したときに選択されるデフォルトグループです。最大文字数：63。
- キータブ -NetScaler に送信されたケルベロスチケットの復号化に使用されるキータブファイルへのパス。キータブが使用できない場合は、ネゴシエートアクション設定でドメイン/ユーザー名/パスワードを指定できま

す。最大長：127

- **NTLMPath** -サーバーの FQDN を含む、NTLM 認証が有効になっているサイトへのパス。これは、クライアントが NTLM にフォールバックするときに使用されます。最大長：127

構成ユーティリティを使用してキータブファイルからユーザー情報を抽出するための認証、承認、および監査を設定するには

> 注
>
> 構成ユーティリティでは、アクションの代わりにサーバーという用語が使用されますが、同じタスクを指します。

1. [セキュリティ] **> [AAA-**アプリケーショントラフィック**] > [**認証**] > [**詳細ポリシー**] > [**アクション**] > [**ネゴシエートアクション**]** に移動します。
2. 詳細ウィンドウの [サーバー] タブで、次のいずれかの操作を行います。
   - 新しいネゴシエートアクションを作成する場合は、「追加」をクリックします。
   - 既存のネゴシエートアクションを変更する場合は、データペインでアクションを選択し、[編集] をクリックします。
3. 新しいネゴシエートアクションを作成する場合は、「名前」テキストボックスに新しいアクションの名前を入力します。名前の長さは 1 ～127 文字で、大文字と小文字、数字、ハイフン (-) と下線 (_) で構成できます。既存のネゴシエートアクションを変更する場合は、このステップをスキップしてください。名前は読み取り専用で、変更できません。
4. 「ネゴシエート」の「キータブ・ファイルを使用」チェック・ボックスがまだオンになっていない場合は、チェックを入れてください。
5. キータブファイルパステキストボックスに、使用するキータブファイルのフルパスとファイル名を入力します。
6. 「デフォルト認証グループ」テキストボックスに、このユーザーのデフォルトとして設定する認証グループを入力します。
7. 「作成」または「**OK**」をクリックして変更を保存します。

Kerberos 認証に高度な暗号化を使用する場合の注意点

> - キータブを使用する場合の設定例: add authentication negotiateAction neg_act_aes256 -keytab “/nsconfig/krb/lbvs_aes256.keytab”
> - **keytab** に複数の暗号化タイプがある場合は、次のコマンドを使用します。このコマンドは、ドメインユーザーパラメータを追加でキャプチャします: add authentication negotiateAction neg_act_keytab_all -keytab “/nsconfig/krb/lbvs_all.keytab” -domainUser “HTTP/lbvs.aaa.local”
> - ユーザー認証情報を使用する場合は、次のコマンドを使用してください。認証を追加 **Ne-**

> **gotiateAction neg_act_user-domain AAA.LOCAL** -DomainUser「http/lbvs.AAA.local」-DomainUserPasswd <password>
> - 正しい **domainUser** 情報が提供されていることを確認します。AD でユーザーログオン名を検索できます。

Web 認証

August 15, 2023

認証、承認、および監査により、Web サーバに対してユーザを認証できるようになりました。Web サーバが HTTP リクエストで必要とするクレデンシャルを提供し、Web サーバの応答を分析して、ユーザ認証が成功したかどうかを判断します。他のタイプの認証ポリシーと同様に、Web 認証ポリシーは式とアクションで構成されます。認証ポリシーを作成したら、それを認証仮想サーバーにバインドし、プライオリティを割り当てます。バインドするときは、プライマリポリシーまたはセカンダリポリシーとして指定します。

特定の Web サーバで Web ベース認証を設定するには、まず Web 認証アクションを作成します。Web サーバーへの認証は厳密な形式を使用しないため、アクションの作成時に Web サーバーが必要とする情報と形式を正確に指定する必要があります。そのためには、NetScaler アプライアンスの詳細ポリシーで次の項目を含む式を作成します。

- [サーバ **IP**]：認証 Web サーバの IP アドレス。
- [サーバポート（Server Port）]：認証 Web サーバのポート。
- 認証ルール—NetScaler アプライアンスの詳細ポリシーの式で、Web サーバーが想定する形式でユーザーの資格情報が含まれます。
- スキーム：HTTP（暗号化されていない Web 認証の場合）または HTTPS（暗号化された Web 認証の場合）。
- 成功ルール—Web サーバーの応答文字列と一致する NetScaler アプライアンスの詳細ポリシーの式で、ユーザーが正常に認証されたことを示します。

他のすべてのパラメータについては、add authentication action コマンドの通常の規則に従います。

次に、そのアクションに関連付けられたポリシーを作成します。このポリシーは LDAP ポリシーに似ており、LDAP ポリシーと同様に NetScaler アプライアンスの構文を使用します。

> 注
>
> これらの手順は、認証する Web サーバの認証要件をすでに理解しており、Web 認証サーバがすでに設定されていることを前提としています。

コマンドラインインターフェイスを使用して **Web** 認証アクションを構成するには

コマンドラインで Web 認証アクションを作成するには、コマンドラインで次のコマンドを入力します。


```
add authentication webAuthAction <name> -serverIP <ip_addr|ipv6_addr
    |\*> -serverPort <port|\*> [-fullReqExpr <string>] -scheme ( http |
    https ) -successRule <expression> [-defaultAuthenticationGroup <
    string>][-Attribute1 <string>][-Attribute2 <string>] [-Attribute3 <
    string>][-Attribute4 <string>] [-Attribute5 <string>][-Attribute6 <
    string>] [-Attribute7 <string>][-Attribute8 <string>] [-Attribute9 <
    string>][-Attribute10 <string>] [-Attribute11 <string>][-Attribute12
     <string>] [-Attribute13 <string>][-Attribute14 <string>] [-
    Attribute15 <string>][-Attribute16 <string>]
<!--NeedCopy-->
```

例

```
add policy expression post_data ""username=" + http.REQ.BODY(1000).
    SET_TEXT_MODE(IGNORECASE).AFTER_STR("login=").BEFORE_STR("&") + "&
    passwort=" + http.REQ.BODY(1000).SET_TEXT_MODE(IGNORECASE).AFTER_STR
    ("passwd=")"

add policy expression length_post_data "("username= " + http.REQ.BODY
    (1000).SET_TEXT_MODE(IGNORECASE).AFTER_STR("login=").BEFORE_STR("&")
     + "passwort=" + http.REQ.BODY(1000).SET_TEXT_MODE(IGNORECASE).
    AFTER_STR("passwd=")).length"

add authentication webAuthAction webAuth_POST -serverIP 10.106.187.54 -
    serverPort 80 -fullReqExpr q{
 "POST /MyPHP/auth.php HTTP/" + http.req.version.major + "." + http.req
     .version.major + "\r\nAccept:\*/\*\r\nHost: 10.106.187.54\r\
     nReferer: http://10.106.187.54/MyPHP/auth.php\r\nAccept-Language:
     en-US\r\nUser-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT
     6.1; Trident/5.0)\r\nContent-Type: application/x-www-form-
     urlencoded\r\n" + "Content-Length: " + length_post_data + "\r\
     nConnection: Keep-Alive\r\n\r\n" + post_data }
  -scheme http -successRule "http.res.status.eq(200)"
<!--NeedCopy-->
```

構成ユーティリティを使用して **Web** 認証アクションを構成するには

> 注
>
> 構成ユーティリティでは、アクションの代わりにサーバーという用語が使用されますが、同じタスクを指します。

1. セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **> LDAP** にナビゲートして下さい。
2. 詳細ウィンドウの [サーバー] タブで、次のいずれかの操作を行います。
   - 新しい Web 認証アクションを作成する場合は、[追加（**Add**）] をクリックします。
   - 既存の Web 認証アクションを変更する場合は、データペインでアクションを選択し、[編集] をクリックします。

3. 新しい Web 認証アクションを作成する場合は、[認証 **Web** サーバーの作成] ダイアログボックスの [名前] テキストボックスに新しい Web 認証アクションの名前を入力します。名前は 1 ～127 文字で、大文字と小文字、数字、ハイフン（-）とアンダースコア（_）で構成できます。既存の Web 認証アクションを変更する場合は、この手順をスキップします。名前は読み取り専用で、変更できません。
4. [**Web** サーバ **IP** アドレス] テキストボックスに、認証 Web サーバの IPv4 または IPv6 IP アドレスを入力します。アドレスが IPv6 IP アドレスの場合は、最初に [IPv6] チェックボックスをオンにします。
5. [ポート] テキストボックスに、Web サーバーが接続を受け入れるポート番号を入力します。
6. [プロトコル] ドロップダウンリストで [**HTTP**] または [**HTTPS**] を選択します。
7. [HTTP Request Expression] テキスト領域に、認証 Web サーバで想定される正確な形式でユーザのクレデンシャルを含む Web サーバリクエストを作成する PCRE 形式の正規表現を入力します。
8. 「認証を検証する式」テキスト領域に、ユーザー認証が成功したことを示す Web サーバー応答の情報を説明する NetScaler アプライアンスの高度なポリシー式を入力します。
9. 一般的な認証アクションのドキュメントの説明に従って、残りのフィールドに入力します。
10. [**OK**] をクリックします。

Web 認証用の **SMS OTP** の設定

February 15, 2024

NetScaler をサードパーティの SMS プロバイダーと統合して、認証をさらに強化できるようになりました。

NetScaler アプライアンスは、認証の第 2 要素としてユーザーのモバイルに OTP を送信するように構成できます。アプライアンスは、AD ログインの成功後に OTP に入るためのログオンフォームをユーザーに提示します。SMS OTP 認証が正常に検証されて初めて、要求されたリソースがユーザーに提示されます。

SMS OTP 認証を実現するために、NetScaler アプライアンスはバックエンドの以下の要素を利用します。

1. LDAP 認証を使用してユーザーを認証し、ユーザーの携帯電話番号を抽出します。
2. OTP を作成し、NS 変数に格納します。変数の設定と使用。
3. LDAP から抽出した携帯電話番号に WebAuth 認証方式で OTP を送信します。
4. OTP を検証します。

前提条件

機能を有効にして **SNIP** を追加

以下のコマンドを実行して、この構成に必要な機能を有効にします：


```
enable ns feature LB SSL SSLVPN AAA

add ns ip <SNIP-IP> <subnet mask> -type SNIP
<!--NeedCopy-->
```

OTP ストアを構成する

管理者は、SMS 認証に使用する OTP を保存するデータベース/ストアを設定する必要があります。

`expires` パラメータ（OTP 有効期限）の推奨設定範囲は 30 ～180 秒です。この範囲外の値があると、SMS OTP 設定と nFactor フローが失敗する可能性があります。

```
add ns variable otp_store -type "map(text(65),text(6),100000)" -
    ifValueTooBig undef -ifNoValue undef -expires 60
<!--NeedCopy-->
```

ユーザーセッションごとにランダム OTP を生成

次のコマンドを使用して、ユーザーセッションごとに 6 桁のランダム OTP を生成し、OTP ストアに保存します。

```
add ns assignment generate_otp -variable "$otp_store[AAA.USER.SESSIONID
    ]" -set ("000000" + SYS.RANDOM.MUL(1000000).
    TYPECAST_UNSIGNED_LONG_AT.TYPECAST_TEXT_T).SUFFIX(6)
<!--NeedCopy-->
```

NetScaler による SMS OTP 認証の設定

- SMS 2 要素認証機能を構成する前に、NetScaler アプライアンスの第 1 要素として LDAP 認証を構成し、認証を有効にする必要があります。LDAP 認証を設定する手順については、「構成ユーティリティを使用して LDAP 認証を構成するには」を参照してください。
- LDAP を設定し、SMS OTP 認証に使用する携帯電話番号を抽出します。

第 1 要素構成のサンプル

認証の最初の要素として LDAP を設定します。

```
 add authentication ldapAction ldap_extraction_action -serverIP <
    IP_Adress> -ldapBase OU=Sanity,dc=aaa,dc=local  -ldapBindDn
    administrator@aaa.local -ldapBindDnPassword <password> -
    ldapLoginName samaccountname -groupAttrName memberof -
    subAttributeName CN -authentication disabled

```



```
3   add authentication Policy ldap_extraction_policy -rule true -action
       ldap_extraction_action
4  <!--NeedCopy-->
```

注

携帯電話番号はAAA.USER.ATTRIBUTE(1)を使用して抽出でき、バックエンドサーバーに送信する際に含めることができます。

第 2 要素構成のサンプル

次のサンプル設定を使用して、エンドユーザーに送信される OTP が生成されます。

```
1  add authentication Policy set_otp -rule true -action generate_otp
2
3  add authentication policylabel set_otp_label -loginSchema LSCHEMA_INT
4
5   bind authentication policylabel set_otp_label  -policyName set_otp -
       priority 1
6
7  add authentication Policy cascade_noauth -rule true -action NO_AUTHN
8  <!--NeedCopy-->
```

Web 認証ポリシーとアクション

次に、LDAP ポリシーから取得した携帯電話番号とともに OTP を任意のサードパーティの SMS 送信者に送信するように Web 認証アクションを設定します。

OTP を確認し、生成された OTP をサードパーティの SMS プロバイダーまたは以下に設定されている任意の Web サーバーに送信します:

```
1  add policy expression otp_exp_post ""Message: OTP is " + $otp_store[AAA
      .USER.SESSIONID] + " for login into secure access  gateway. Valid
      till EXPIRE_TIME. Do not share the OTP with anyone for security
      reasons&Mobile:" + AAA.USER.ATTRIBUTE(1)"
2
3  add authentication webAuthAction sms_post -serverIP <web_application_ip
      > -serverPort 80 -fullReqExpr q{
4   "POST /MyPHP/auth.php HTTP/" + http.req.version.major + "." + http.req
       .version.major + "\r\nAccept:\*/\*\r\nHost: <web_application_ip> \r
       \nContent-Length:100\r\n\r\n" + otp_exp_post }
5    -scheme http -successRule true
6
7  add authentication Policy post_wpp -rule true -action sms_post
8  <!--NeedCopy-->
```


ワンタイムパスワードを確認する

次のポリシーとポリシーラベルを追加して、ユーザーセッションに OTP が存在するかどうかを確認します:

```
add authentication policylabel check_otp_label -loginSchema LSCHEMA_INT

add authentication Policy check_otp -rule "$otp_store.valueExists(AAA.USER.SESSIONID)" -action NO_AUTHN
<!--NeedCopy-->
```

ユーザーが入力した値と **OTP** ストアの値を比較することによる **OTP** 検証

次のコマンドを実行して、ユーザーが入力した値と OTP ストア値を比較して OTP を確認します:

```
add authentication loginSchema onlypassword -authenticationSchema "/nsconfig/loginschema/LoginSchema/OnlyPassword.xml"

add authentication policylabel otp_verify_label -loginSchema onlypassword

add authentication Policy otp_verify -rule "AAA.LOGIN.PASSWORD.EQ($otp_store[AAA.USER.SESSIONID])" -action NO_AUTHN
<!--NeedCopy-->
```

ポリシーを認証ポリシーラベルにバインドする

```
bind authentication policylabel set_otp_label -policyName cascade_noauth -priority 2 -gotoPriorityExpression NEXT -nextFactor check_otp_label

bind authentication policylabel check_otp_label -policyName post_wpp -priority 1 -gotoPriorityExpression NEXT -nextFactor otp_verify_label

bind authentication policylabel otp_verify_label -policyName otp_verify -priority 1 -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

認証仮想サーバーと負荷分散仮想サーバーの作成

```
 add authentication vserver avs SSL <ipadresss> 443

add ssl certKey aaa_local -cert aaatm_wild.cer -key aaatm_wild.key

bind ssl vserver avs -certkeyName aaa_local

bind authentication vserver avs -policy ldap_extraction_policy -priority 1 -nextFactor set_otp_label -gotoPriorityExpression NEXT
```



```
8
9  add lb vserver lb HTTP <ip_adresss> 80 -persistenceType NONE -
       cltTimeout 180 -AuthenticationHost <auth_server> -Authentication ON
10
11 add service svc <Ip_adresss> HTTP 80
12
13 bind lb vserver lb svc
14 <!--NeedCopy-->
```

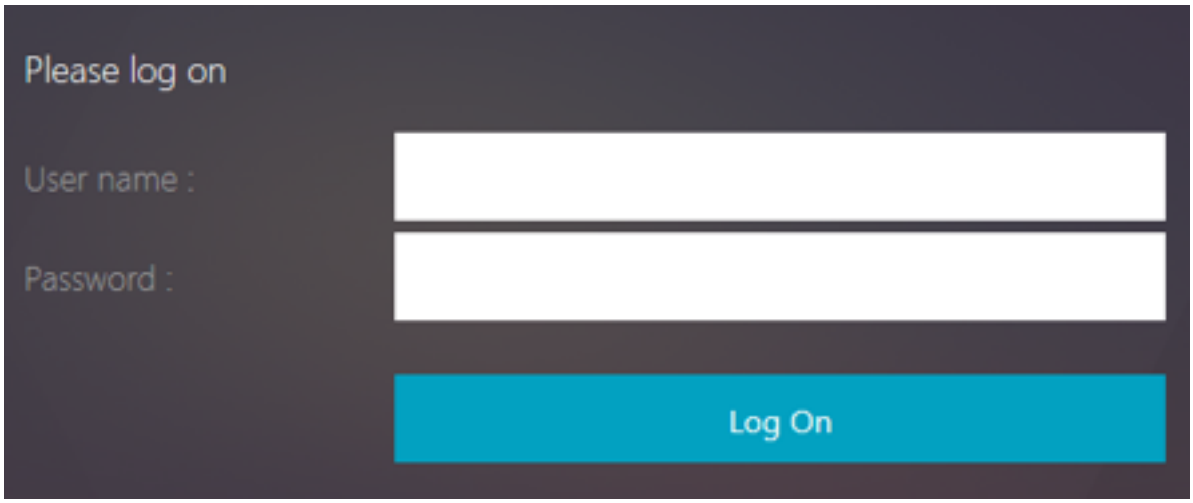
注:

カスケード認証のポリシーが追加され、エンドユーザーの信頼性が高く継続的な認証が可能になります。現在の要因が失敗した場合、ユーザーエクスペリエンスに影響がないように次の要素が評価されます。

フォームベースの認証

August 15, 2023

フォームベースの認証では、ログオンフォームがエンドユーザーに表示されます。このタイプの認証フォームは、多要素 (nFactor) 認証とクラシック認証の両方をサポートしています。



フォームベースの認証が機能するには、以下を確認してください。

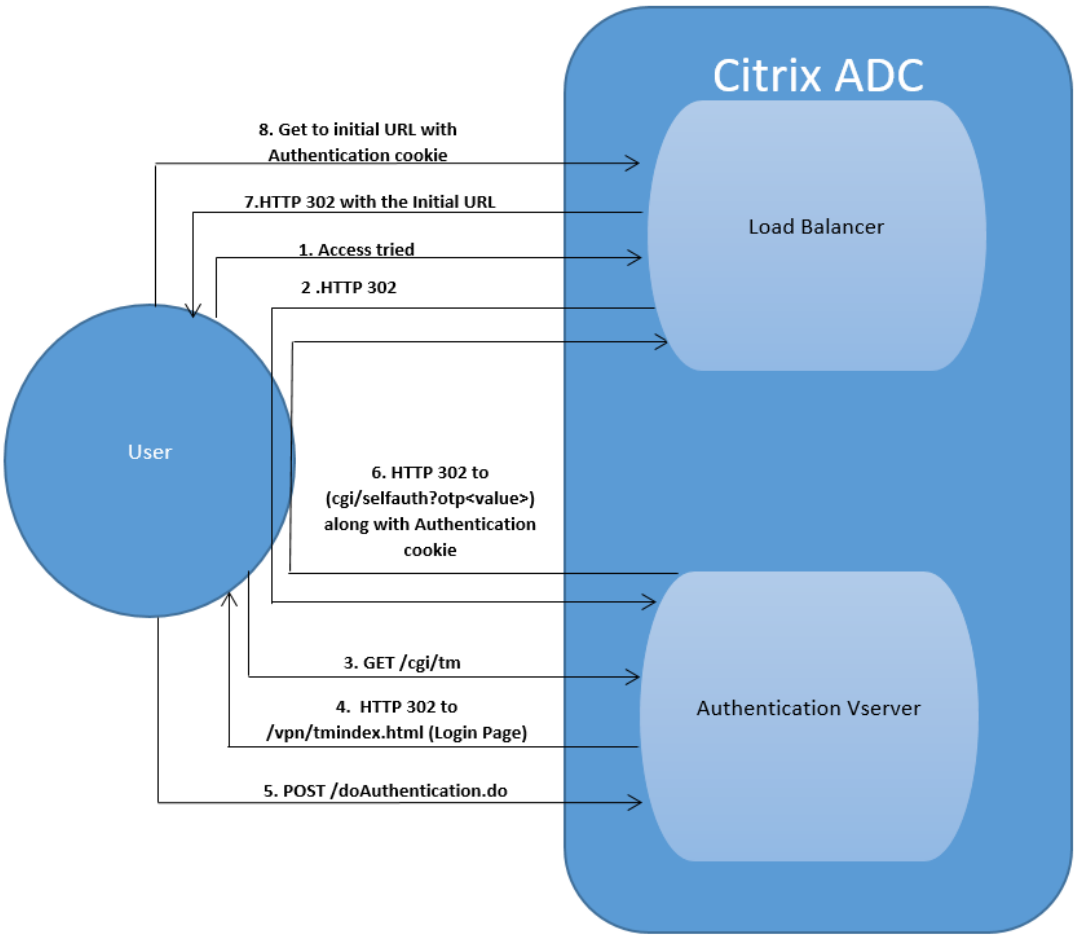
- 負荷分散仮想サーバーでは、認証がオンになっている必要があります。
- 認証のためにユーザーをリダイレクトする必要がある「AuthenticationHost」パラメーターを指定する必要があります。これを設定するためのコマンドは次のとおりです:

  ```
  1  set lb vs lb1 -authentication on  - authenticationhost aaavs-ip/
         fqdn
  ```

- フォームベース認証は、HTML をサポートするブラウザと互換性があります

次の手順では、フォームベースの認証がどのように機能するかを説明します。

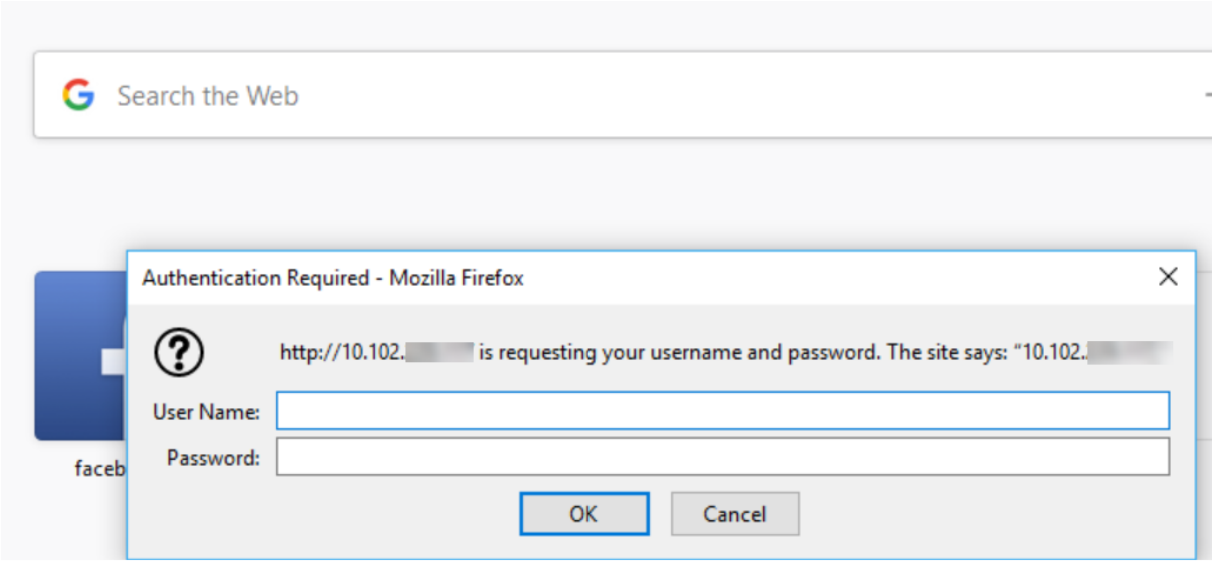
1. クライアント (ブラウザ) は、TM (ロードバランシング/CS) 仮想サーバー上の URL に対する GET リクエストを送信します。
2. TM 仮想サーバーは、クライアントが認証されていないと判断し、HTTP 302 応答をクライアントに送信します。応答には、クライアントが認証仮想サーバーに /cgi/tm の GET 要求を発行するようにする隠しスクリプトが含まれています。
3. クライアントは、ターゲット URL を含む GET /cgi/tm を認証仮想サーバに送信します。
4. 認証仮想サーバーは、ログインページにリダイレクトを送信します。
5. ユーザーは、POST /doAuthentication.do を使用して認証仮想サーバーに認証情報を送信します。認証は認証仮想サーバーによって行われます。
6. 資格情報が正しい場合、認証仮想サーバーは、負荷分散サーバーの cgi/selfauth URL に HTTP 302 応答を 1 回限りのトークン (OTP) とともに送信します。
7. 負荷分散サーバーは HTTP 302 をクライアントに送信します。
8. クライアントは、32 バイトの cookie とともに、最初の URL のターゲット URL の GET 要求を送信します。



401 ベースの認証

August 15, 2023

401 ベースの認証では、NetScaler アプライアンスはエンドユーザーにポップアップダイアログボックスを表示します。



フォームベースの AAA-TM はリダイレクトメッセージで動作します。一部のアプリケーションはリダイレクトをサポートしていません。そのような場合は、401 認証が有効になっている AAA-TM が使用されます。

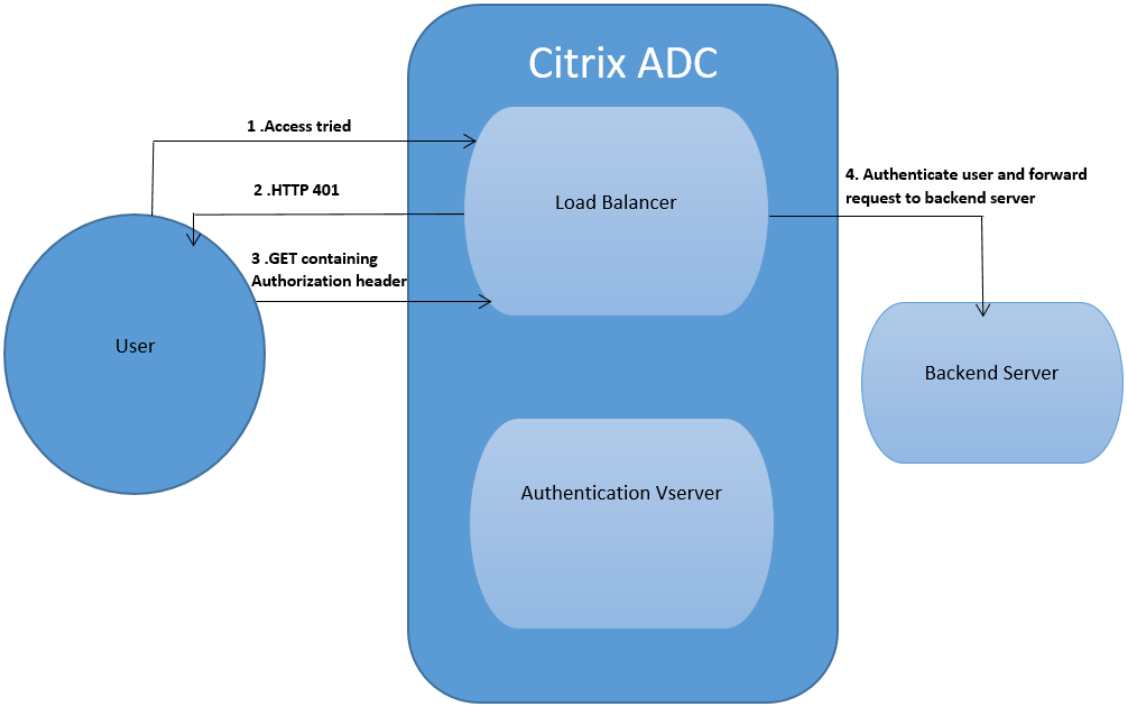
401 認証 AAA-TM が機能するには、次のパラメータを有効にします。

- 負荷分散仮想サーバーの「AuthNvsName」パラメーター値は、ユーザーの認証に使用する認証仮想サーバーの名前である必要があります。
- 'authn401' パラメータを有効にする必要があります。これを設定するためのコマンドは次のとおりです:

```
set lb vs lb1 -authn401 on -authnvsName <aaavs-name>
```

次の手順では、401 認証の動作について説明します。

1. ユーザーが負荷分散仮想サーバーを使用して特定の URL にアクセスしようとしました。
2. 負荷分散仮想サーバーは、アクセスに認証が必要であることを示す 401 HTTP 応答をユーザーに送信します。
3. ユーザーは、認証ヘッダーで資格情報を負荷分散仮想サーバーに送信します。
4. 負荷分散仮想サーバーはユーザーを認証し、そのユーザーをバックエンドサーバーに接続します。



> 重要:
>
> 401 認証がオンの負荷分散仮想サーバーでは、同じユーザーに対して短時間で複数の認証および承認セッションが作成される可能性があります。この構成により、メモリが急増する可能性があります。NetScaler アプライアンスに次の構成を適用して、エンドクライアントアプリケーションをデバッグおよび識別できます。

```
set syslogparams -userDefinedAuditlog yes

add audit messageaction 401_log_act InFORMATIONAL '"LB-401 accessed:
    User: <" + AAA.USER.NAME + "> SessionID <"+ AAA.USER.SESSIONID + ">
    Client :<" + CLIENT.IP.SRC + "> accessed URL: <" + HTTP.REQ.URL +
    ">"'

add rewritepolicy rewrite_401_log true NOREWRITE -logAction 401_log_act

bind lb vserver <lb_name> -policyName rewrite_401_log -priority 100 -
    type reqUEST
<!--NeedCopy-->
```

nFactor 認証の設定を再キャプチャする

August 15, 2023

NetScaler Gateway は、`captchaAction` 再キャプチャ構成を簡素化する新しいファーストクラスアクションをサポートしています。再キャプチャはファーストクラスのアクションなので、それ自体が要因になる可能性があります。再キャプチャは nFactor フローのどこにでも注入できます。

以前は、rfWebUi に変更を加えたカスタム WebAuth ポリシーも作成する必要がありました。`captchaAction`の導入により、JavaScript を変更する必要がなくなりました。

> 重要:
>
> スキーマ内のユーザー名またはパスワードフィールドとともに Re-CAPTCHA が使用されている場合、再キャプチャが満たされるまで [送信] ボタンは無効になります。

設定を再キャプチャ

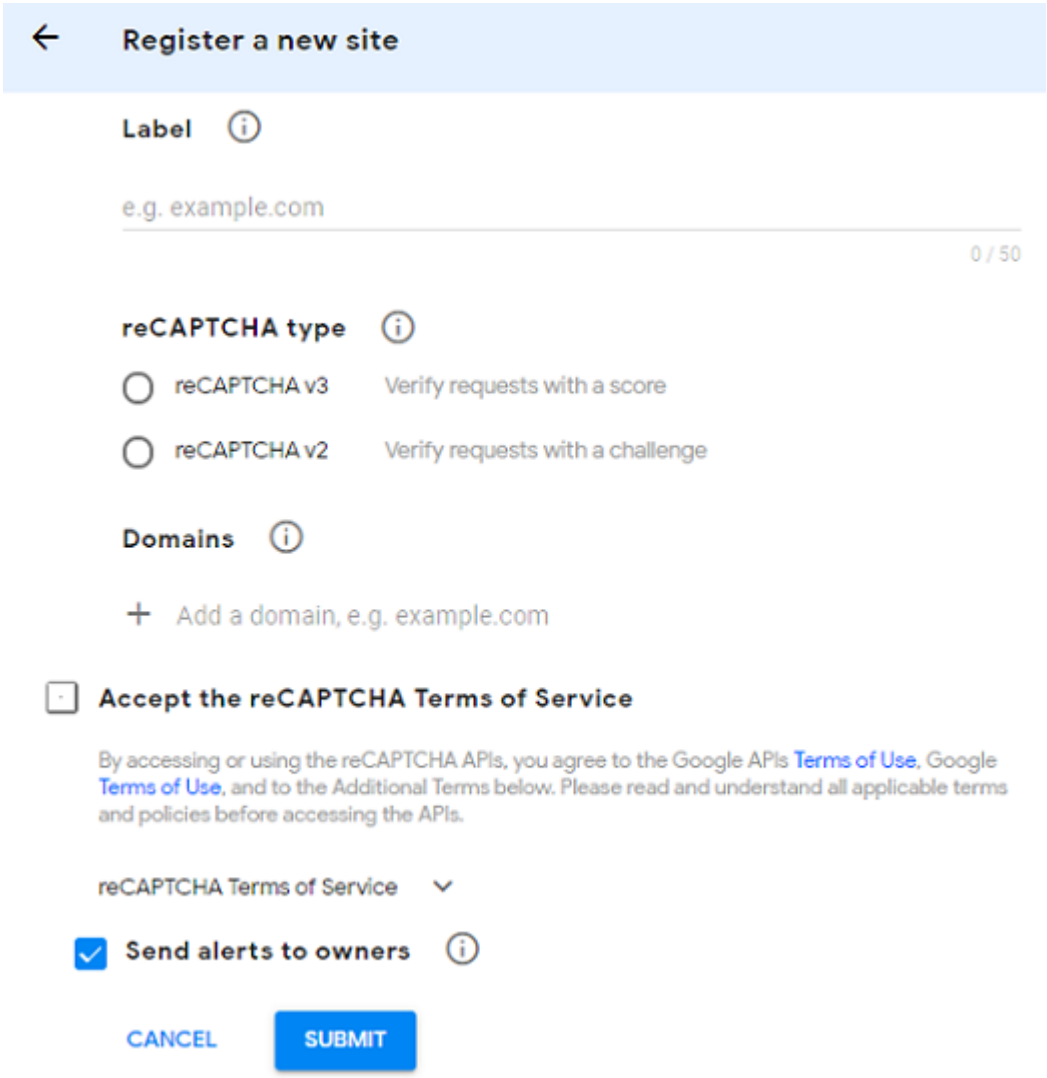
再キャプチャの設定には 2 つの部分が含まれます。

1. 再キャプチャを登録するための Google での設定。
2. ログインフローの一部として再キャプチャを使用するように NetScaler アプライアンスを構成します。

Google の設定を再キャプチャ

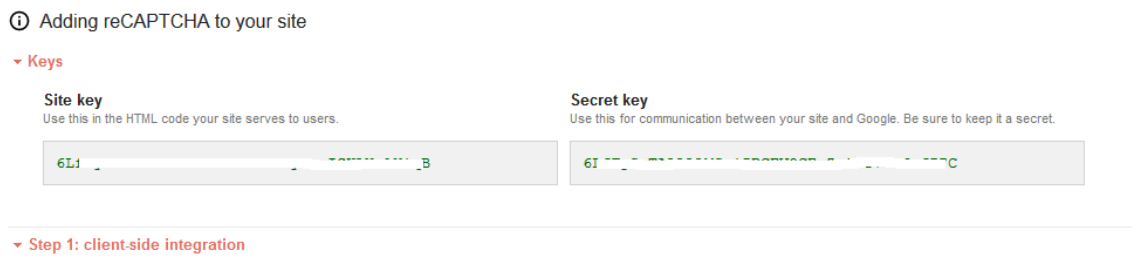
https://www.google.com/recaptcha/admin#llistで再キャプチャするドメインを登録します。

1. このページに移動すると、次の画面が表示されます。



> 注
>
> reCAPTCHA v2 のみを使用してください。見えない再キャプチャはまだプレビュー中です。

2. ドメインが登録されると、「SiteKey」と「SecretKey」が表示されます。



> 注
>
> セキュリティ上の理由から、「siteKey」と「SecretKey」はグレー表示されています。「SecretKey」は

> 安全に保管する必要があります。

NetScaler アプライアンスでの構成の再キャプチャ

NetScaler アプライアンスの Re-CAPTCHA 構成は、次の 3 つの部分に分けることができます。

- 再キャプチャ画面を表示する
- 再キャプチャレスポンスを Google サーバに投稿する
- LDAP 構成はユーザーログオンの 2 番目の要素です (オプション)

再キャプチャ画面を表示する　ログインフォームのカスタマイズは、SingleAuthCaptcha.xml ログインスキーマを介して行われます。このカスタマイズは認証仮想サーバーで指定され、ログインフォームをレンダリングするために UI に送信されます。組み込みのログインスキーマである SingleAuthCaptcha.xml は、/nsconfig/loginSchema/LoginSchema NetScaler アプライアンスのディレクトリにあります。

> 重要
>
> - SingleAuthCaptcha.xml ログインスキーマは、LDAP が第 1 要素として構成されている場合に使用できます。
> - ユースケースと異なるスキーマに基づいて、既存のスキーマを変更できます。たとえば、再キャプチャ要素（ユーザー名またはパスワードなし）または再キャプチャによる二重認証のみが必要な場合などです。
> - カスタム変更が行われた場合、またはファイルの名前が変更された場合は、すべての LoginSchemas を /nsconfig/loginschema/LoginSchemaディレクトリから親ディレクトリ /nsconfig/loginschemaにコピーすることをお勧めします。

CLI を使用して **Re-CAPTCHA** の表示を設定するには

```
add authentication loginSchema singleauthcaptcha -authenticationSchema
    /nsconfig/loginschema/SingleAuthCaptcha.xml

add authentication loginSchemaPolicy singleauthcaptcha -rule true -
    action singleauthcaptcha

add authentication vserver auth SSL <IP> <Port>

add ssl certkey vserver-cert -cert <path-to-cert-file> -key <path-to-
    key-file>

bind ssl vserver auth -certkey vserver-cert

bind authentication vserver auth -policy singleauthcaptcha -priority 5
    -gotoPriorityExpression END
<!--NeedCopy-->
```


再キャプチャレスポンスを **Google** サーバに投稿する　ユーザーに表示する必要のある再キャプチャを設定したら、管理者はその構成を Google サーバーに追加して、ブラウザからの再キャプチャ応答を確認します。

ブラウザからの **Re-CAPTCHA** 応答を確認するには

```
add authentication captchaAction myrecaptcha -sitekey <sitekey-copied-from-google> -secretkey <secretkey-from-google>

add authentication policy myrecaptcha -rule true -action myrecaptcha

bind authentication vserver auth -policy myrecaptcha -priority 1
<!--NeedCopy-->
```

AD 認証が必要かどうかを設定するには、次のコマンドが必要です。それ以外の場合は、この手順は無視してかまいません。

```
add authentication ldapAction ldap-new -serverIP x.x.x.x -serverPort 636 -ldapBase "cn=users,dc=aaatm,dc=com" -ldapBindDn adminuser@aaatm.com -ldapBindDnPassword <password> -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName memberof -subAttributeName CN -secType SSL -passwdChange ENABLED -defaultAuthenticationGroup ldapGroup

add authenticationpolicy ldap-new -rule true -action ldap-new
<!--NeedCopy-->
```

LDAP 構成はユーザーログオンの **2** 番目の要素です **(**オプション**)**　LDAP 認証は再キャプチャ後に行われ、2 番目の要素に追加します。

```
add authentication policylabel second-factor

bind authentication policylabel second-factor -policy ldap-new -priority 10

bind authentication vserver auth -policy myrecaptcha -priority 1 -nextFactor second-factor
<!--NeedCopy-->
```

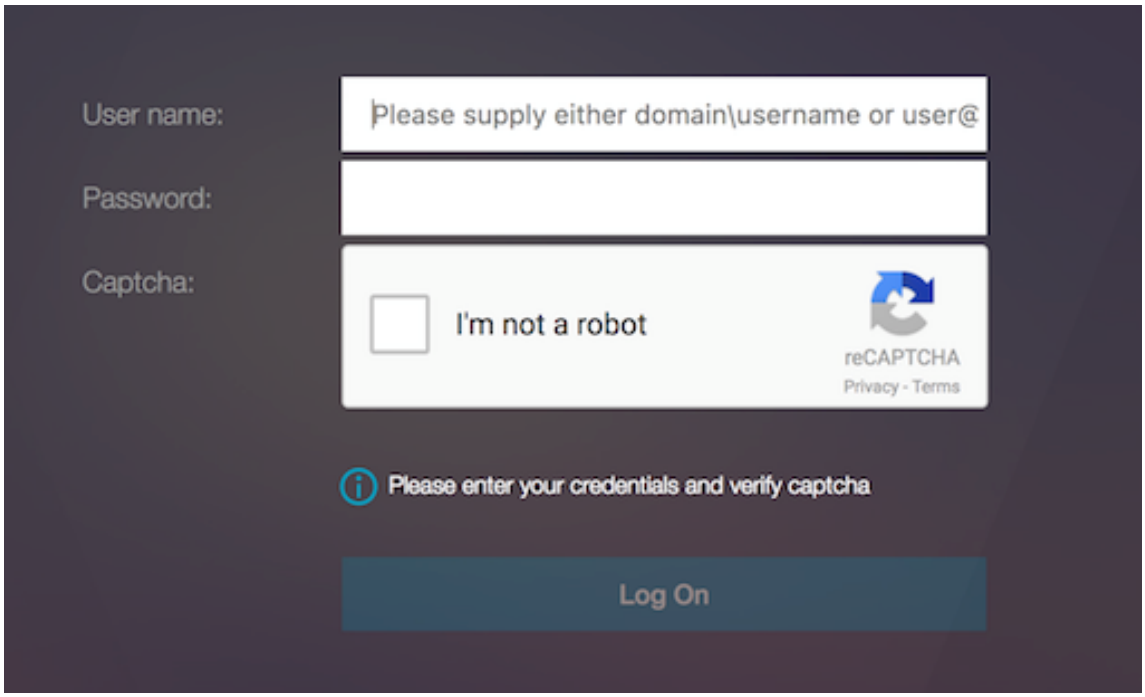
管理者は、アクセスに負荷分散仮想サーバーと NetScaler Gateway アプライアンスのどちらを使用するかに応じて、適切な仮想サーバーを追加する必要があります。負荷分散仮想サーバーが必要な場合は、管理者が次のコマンドを構成する必要があります。

```
add lb vserver lbtest HTTP <IP> <Port> -authentication ON -authenticationHost nssp.aaatm.com
<!--NeedCopy-->
```

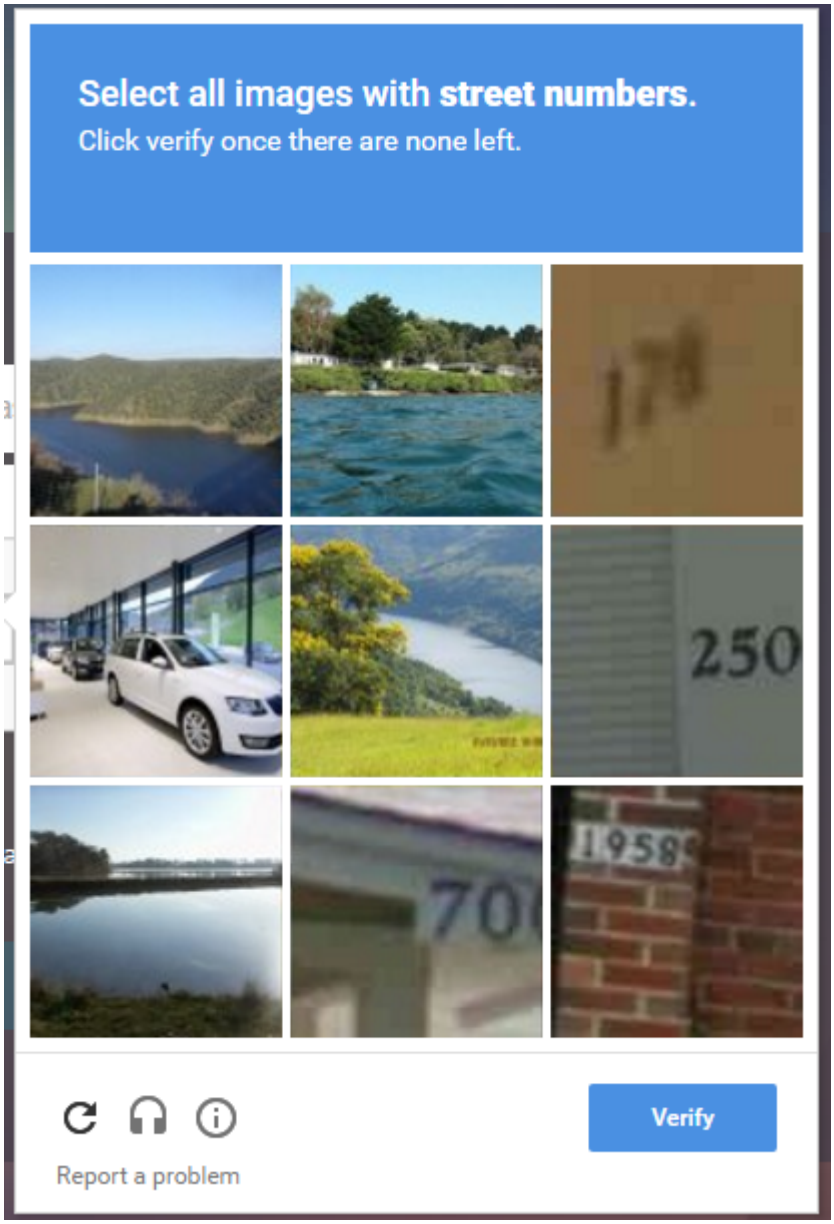
nssp.aaatm.com —認証仮想サーバに解決します。

再キャプチャのユーザー検証　前のセクションで説明したすべての手順を構成したら、次の UI が表示されるはずです。

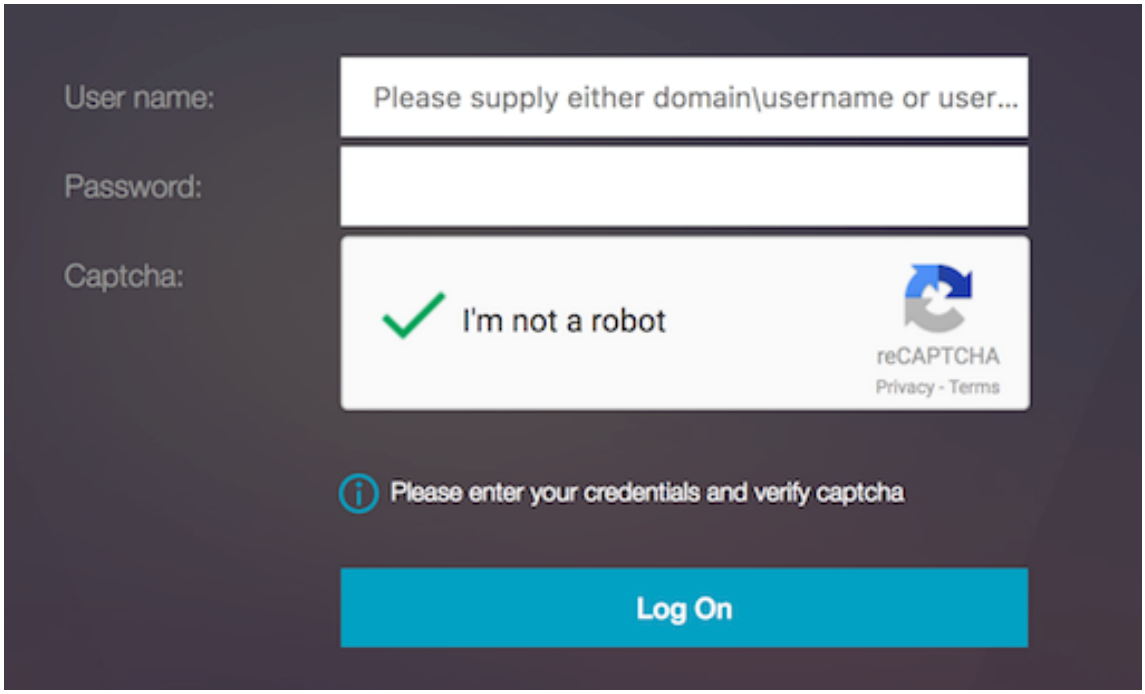
1. 認証仮想サーバーがログインページをロードすると、ログオン画面が表示されます。再キャプチャが完了するまで、ログオンは無効になります。



2. [ロボットではありません] オプションを選択します。Re-Captcha ウィジェットが表示されます。



3. 完了ページが表示される前に、一連の再キャプチャ画像をナビゲートします。
4. AD 資格情報を入力し、[ロボットではありません] チェックボックスをオンにして、[ログオン] をクリックします。認証が成功すると、目的のリソースにリダイレクトされます。



注:

- 再キャプチャが AD 認証で使用されている場合、再キャプチャが完了するまで、資格情報の [送信] ボタンは無効になります。
- 再キャプチャは独自の要因で発生します。したがって、AD のような後続の検証は、re-captcha の`nextfactor`で行う必要があります。

認証のためのネイティブ **OTP** サポート

December 8, 2023

NetScaler は、サードパーティのサーバーを使用しなくてもワンタイムパスワード（OTP）をサポートします。ワンタイムパスワードは、生成される番号またはパスコードがランダムであるため、セキュリティで保護されたサーバーを認証するための非常に安全なオプションです。以前は、乱数を生成する特定のデバイスを持つ RSA などの専門企業が、OTP を提供していました。

この機能は、資本コストと運用経費の削減に加えて、構成全体を NetScaler アプライアンスに保持することで管理者の制御を強化します。

注:

サードパーティサーバーが不要になったため、NetScaler 管理者はユーザーデバイスを管理および検証するためのインターフェイスを構成する必要があります。

OTP ソリューションを使用するには、ユーザーを NetScaler 仮想サーバーに登録する必要があります。登録は一意のデバイスごとに一度だけ必要で、特定の環境に制限することができます。登録ユーザーの設定と検証は、追加の認証ポリシーの設定と似ています。

ネイティブ **OTP** サポートの利点

- Active Directory に加えて、認証サーバー上に追加のインフラストラクチャを用意する必要がなくなるため、運用コストが削減されます。
- 構成を NetScaler アプライアンスのみに統合するため、管理者はきめ細かく制御できます。
- クライアントが期待する数値を生成するために、クライアントが追加の認証サーバーに依存する必要がなくなります。

ネイティブ **OTP** ワークフロー

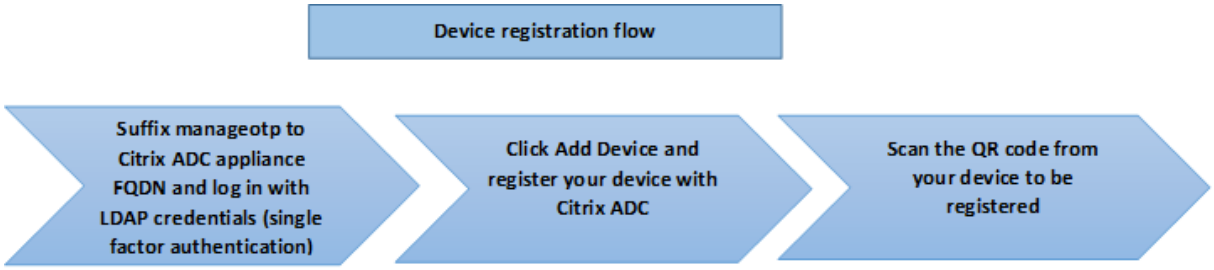
ネイティブ OTP ソリューションは 2 つのプロセスであり、ワークフローは次のように分類されます。

- デバイス登録
- エンドユーザログイン

重要:

> サードパーティのソリューションを使用している場合や、NetScaler アプライアンス以外のデバイスを管理している場合は、登録プロセスをスキップできます。追加する最後の文字列は、NetScaler が指定した形式である必要があります。

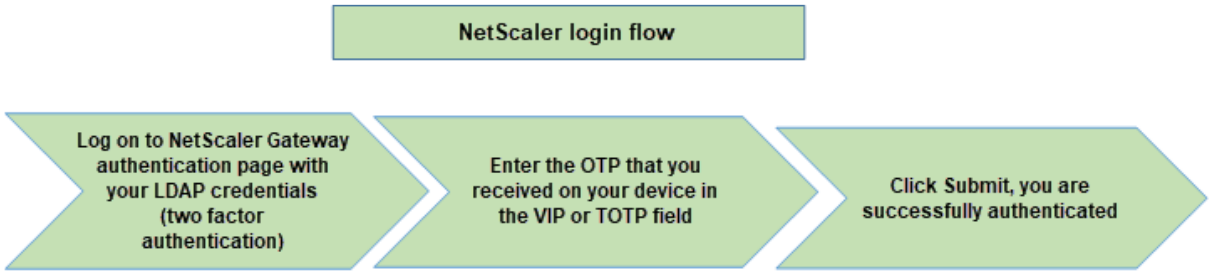
次の図は、OTP を受信する新しいデバイスを登録するためのデバイス登録フローを示しています。



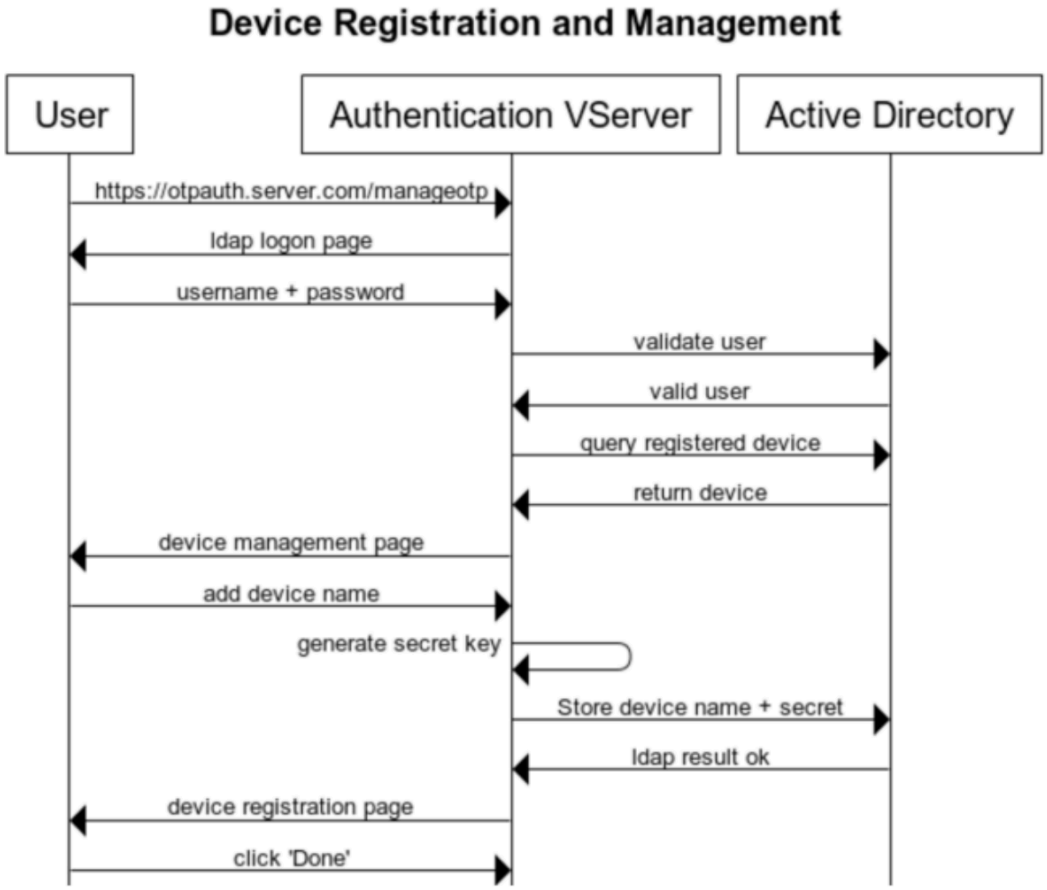
注:

> デバイスの登録は、さまざまな要素を使用して行うことができます。デバイス登録プロセスを説明するために、1 つの要素（前の図で指定）を例として使用します。

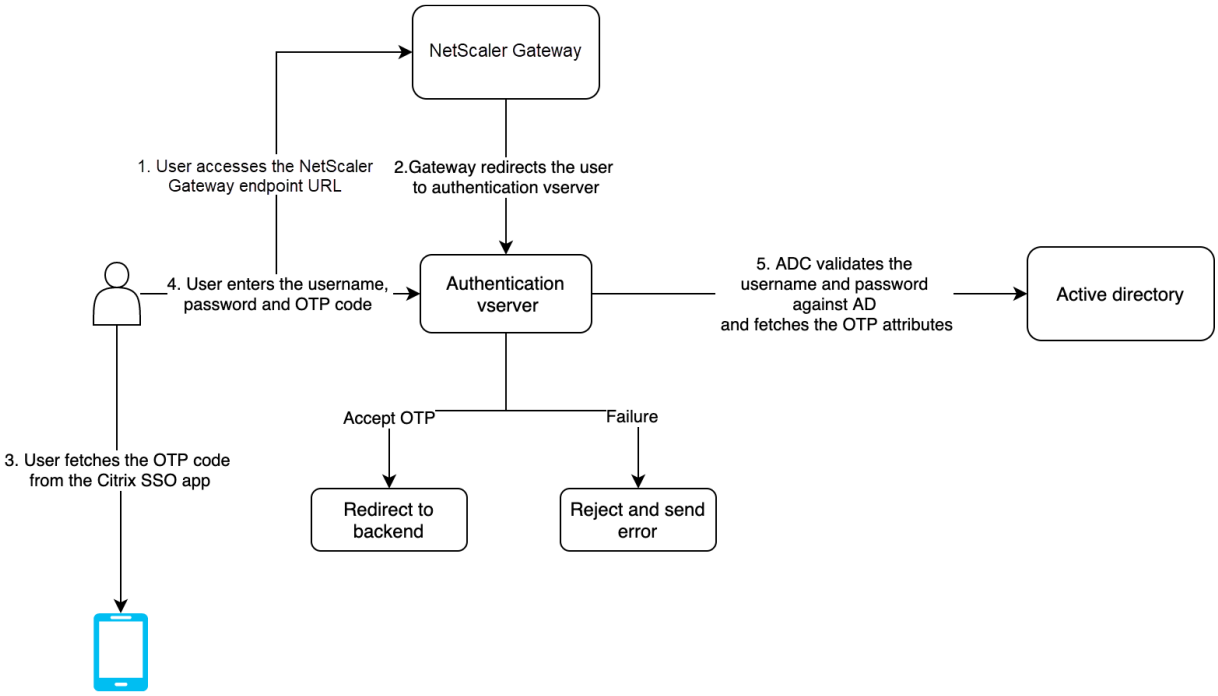
次の図は、登録済みデバイスを介した OTP の検証を示しています。



次の図は、デバイスの登録と管理フローを示しています。



次の図は、ネイティブ OTP 機能のエンドユーザフローを示しています。



前提条件

ネイティブ OTP 機能を使用するには、次の前提条件が満たされていることを確認してください。

- NetScaler 機能リリースバージョンは 12.0 ビルド 51.24 以降です。
- Advanced エディションまたは Premium エディションのライセンスが NetScaler Gateway にインストールされています。
- NetScaler は管理 IP で構成されており、管理コンソールにはブラウザとコマンドラインの両方を使用してアクセスできます。
- NetScaler は、ユーザーを認証するための認証、承認、監査仮想サーバーで構成されています。詳細については、「認証仮想サーバー」を参照してください。
- NetScaler アプライアンスは Unified Gateway で構成され、認証、承認、監査プロファイルが Gateway 仮想サーバーに割り当てられます。
- ネイティブ OTP ソリューションは、nFactor 認証フローに制限されています。ソリューションを構成するには、高度なポリシーが必要です。詳細については、「nFactor 認証の設定」を参照してください。

また、Active Directory については、次の点を確認してください:

- 属性の最小長は 256 文字です。
- 属性タイプは、ユーザパラメータなどの ‘DirectoryString’ である必要があります。これらの属性は文字列値を保持できます。
- デバイス名が英語以外の文字である場合、属性文字列タイプは Unicode である必要があります。
- NetScaler LDAP 管理者は、選択した AD 属性への書き込みアクセス権を持っている必要があります。

- NetScaler アプライアンスとクライアントマシンは、共通のネットワークタイムサーバーと同期する必要があります。

GUI を使用したネイティブ OTP の設定

ネイティブ OTP 登録は、単要素認証だけではありません。次のセクションでは、シングルファクタ認証とセカンドファクタ認証の設定について説明します。

第 1 ファクタのログインスキーマの作成

1. [セキュリティ] > [AAA アプリケーショントラフィック] > [ログインスキーマ] に移動します。
2. [プロファイル] に移動し、[追加] をクリックします。
3. [認証ログインスキーマの作成] ページで、[名前] フィールドに *lschema_single_auth_manage_otp* と入力し、**noschema** の横にある [編集] をクリックします。
4. [ログインスキーマ] フォルダをクリックします。
5. 下にスクロールして **SingleAuthManageOTP.xml** を選択し、「選択」をクリックします。
6. [作成] をクリックします。
7. [ポリシー] をクリックし、[追加] をクリックします。
8. [認証ログインスキーマポリシーの作成] 画面で、次の値を入力します。

   名前**:** lpol_single_auth_manage_otp_by_url

   プロファイル**:** リストから lschema_single_auth_manage_otp を選択します。

   規則**:** `HTTP.REQ.COOKIE.VALUE("NSC_TASS").EQ("manageotp")`

認証、承認、および監査仮想サーバーの構成

1. [セキュリティ] > [AAA-アプリケーショントラフィック] > [認証仮想サーバ] に移動します。既存の仮想サーバーを編集する場合にクリックします。詳細については、「認証仮想サーバー」を参照してください。
2. 右側のペインの [詳細設定] の [ログインスキーマ] の横にある [+] アイコンをクリックします。
3. 「ログインスキーマなし」を選択します。
4. 矢印をクリックし、**lpol_single_auth_manage_otp_by_url** ポリシーを選択し、[選択] をクリックして、[バインド] をクリックします。
5. 上にスクロールし、[高度な認証ポリシー] の下の [認証ポリシー] を **1** つ選択します。
6. **nFactor** ポリシーを 右クリックし、[バインディングの編集] を選択します。すでに設定されている nFactor ポリシーを右クリックするか、nFactor を参照して作成して [バインドの編集] を選択します。

7. [次の因子を選択] の下の矢印をクリックして既存の構成を選択するか、[追加] をクリックして因子を作成します。
8. [認証 **PolicyLabel** の作成] 画面で次のように入力し、[続行] をクリックします。

   名前: manage_otp_flow_label

   ログインスキーマ: Lschema_Int
9. [認証 **PolicyLabel**] 画面で、[追加] をクリックしてポリシーを作成します。

   ```
   Create a policy for a normal LDAP server.
   ```
10. [認証ポリシーの作成] 画面で、次のように入力します。

    名前:auth_pol_ldap_native_otp
11. [アクションタイプ] リストを使用して、[アクションタイプ] を [**LDAP**] として選択します。
12. [アクション] フィールドで、[追加] をクリックしてアクションを作成します。

    ```
    Create the first LDAP action with authentication enabled to be
    used for single factor.
    ```
13. 「認証 **LDAP** サーバーの作成」ページで、「サーバー **IP** 」ラジオボタンを選択し、「認証」の横にあるチェックボックスをオフにし、次の値を入力して「テスト接続」を選択します。次に、設定例を示します。

    名前:ldap_native_otp

    IP アドレス:192.8.xx.xx

    ベース **DN**: DC = トレーニング、DC = ラボ

    管理者:Administrator@training.lab

    パスワード: xxxxx

    ```
    Create a policy for OTP .
    ```
14. [認証ポリシーの作成] 画面で、次のように入力します。

    名前: auth_pol_ldap_otp_action
15. [アクションタイプ] リストを使用して、[アクションタイプ] を [**LDAP**] として選択します。
16. [アクション] フィールドで、[追加] をクリックしてアクションを作成します。

    ```
    Create the second LDAP action to set OTP authenticator with OTP
    secret configuration and authentication unchecked.
    ```
17. 「認証 **LDAP** サーバーの作成」ページで、「サーバー **IP** 」ラジオボタンを選択し、「認証」の横にあるチェックボックスをオフにし、次の値を入力して「テスト接続」を選択します。次に、設定例を示します。

    名前: ldap_otp_action

    IP アドレス:192.8.xx.xx

ベース **DN**: DC = トレーニング、DC = ラボ

管理者:Administrator@training.lab

パスワード: xxxxx

18. [その他の設定**]** セクションまでスクロールします。ドロップダウンメニューを使用して、次のオプションを選択します。
サーバーログオン名属性として「新規」と入力し、**userprincipalname** と入力します。
19. ドロップダウンメニューを使用して [**SSO** 名属性] を [新規] として選択し、**userprincipalname** と入力します。
20. [**OTP** シークレット] フィールドに「UserParameters」と入力し、[詳細] をクリックします。
21. 次の属性を入力します。

    属性 **1** =
    メール属性 **2** = objectGUID
    属性 **3** = immutableID
22. [**OK**] をクリックします。
23. [認証ポリシーの作成] ページで、[式] を **true** に設定し、[作成] をクリックします。
24. [認証ポリシーラベルの作成] ページで、[バインド] をクリックし、[完了**]** をクリックします。
25. [ポリシーのバインド] ページで、[バインド] をクリックします。
26. [認証ポリシー] ページで、[閉じる] をクリックし、[完了] をクリックします。

    ```
    Create OTP for OTP verification.
    ```
27. [認証ポリシーの作成] 画面で、次のように入力します。

    名前:auth_pol_ldap_otp_verify
28. [アクションタイプ] リストを使用して、[アクションタイプ] を [**LDAP**] として選択します。
29. [アクション] フィールドで、[追加] をクリックしてアクションを作成します。

    ```
    Create the third LDAP action to verify OTP.
    ```
30. 「認証 **LDAP** サーバーの作成」ページで、「サーバー **IP** 」ラジオボタンを選択し、「認証」の横にあるチェックボックスをオフにし、次の値を入力して「テスト接続」を選択します。次に、設定例を示します。

    名前:ldap_verify_otp

    IP アドレス: 192.168.xx.xx

    ベース **DN**: DC = トレーニング、DC = ラボ

    管理者:Administrator@training.lab

    パスワード: xxxxx

31. [その他の設定**]** セクションまでスクロールします。ドロップダウンメニューを使用して、次のオプションを選択します。
    サーバーログオン名属性として「新規」と入力し、**userprincipalname** と入力します。
32. ドロップダウンメニューを使用して [**SSO** 名属性] を [新規] として選択し、**userprincipalname** と入力します。
33. [**OTP** シークレット] フィールドに「UserParameters」と入力し、[詳細] をクリックします。
34. 次の属性を入力します。

    属性 **1** =
    メール属性 **2** = objectGUID
    属性 **3** = immutableID
35. [**OK**] をクリックします。
36. [認証ポリシーの作成] ページで、[式] を **true** に設定し、[作成] をクリックします。
37. [認証ポリシーラベルの作成] ページで、[バインド] をクリックし、[完了**]** をクリックします。
38. [ポリシーのバインド] ページで、[バインド] をクリックします。
39. [認証ポリシー] ページで、[閉じる] をクリックし、[完了] をクリックします。

通常の LDAP サーバー用の高度な認証ポリシーがまだ設定されていない可能性があります。
[アクションタイプ] を LDAP に変更します。
通常の LDAP サーバーを選択します。これは、認証が有効になっているサーバーです。
式に true を入力します。これは、クラシック構文の代わりに高度なポリシーを使用します。
「作成」をクリックします。

> 注:
>
> 認証仮想サーバーは、RFWebUI ポータルテーマにバインドする必要があります。サーバー証明書をサーバーにバインドします。サーバー IP ‘1.2.3.5’ には、後で使用できるように otpauth.server.com という対応する FQDN が必要です。

第 **2** 要素 **OTP** のログインスキーマの作成

1. [セキュリティ**] > [AAA** アプリケーショントラフィック**] > [**仮想サーバ] に移動します。編集する仮想サーバーを選択します。
2. 下にスクロールして、[**1** つのログインスキーマ] を選択します。
3. [**Add Binding**] をクリックします。
4. [ポリシーバインド] セクションで、[追加] をクリックしてポリシーを追加します。
5. 「認証ログインスキーマポリシーの作成」ページで、ポリシーの名前を入力し、「追加」をクリックします。
6. 「認証ログインスキーマの作成」ページで、ログインスキーマの名前を入力し、**noschema** の横にある鉛筆アイコンをクリックします。

7. **LoginSchema** フォルダ をクリックし 、**DualAuthManageOTP.xml** を選択し 、**[**選択] をクリックします。
8. [詳細] をクリックし、下にスクロールします。
9. [パスワードクレデンシャルインデックス] フィールドに、1 と入力します。これにより、nFactor はユーザーのパスワードを認証、承認、監査用の属性 #1 に保存し、後でトラフィックポリシーで StoreFront へのシングルサインオンに使用できます。これを行わないと、NetScaler Gateway はパスコードを使用して StoreFront への認証を試みますが、機能しません。
10. [作成] をクリックします。
11. [ルール] セクションで、**True** と入力します。[作成] をクリックします。
12. [**Bind**] をクリックします。
13. 認証の 2 つの要素に注目してください。[閉じる] をクリックし、[完了] をクリックします。

シングルサインオンのトラフィックポリシー

1. **NetScaler Gateway >** ポリシー **>** トラフィックに移動します。
2. [トラフィックプロファイル] タブで、[追加] をクリックします。
3. トラフィックプロファイルの名前を入力します。
4. 下にスクロールして、[SSO パスワード式] ボックスに次のように入力し、[作成] をクリックします。ここでは、第 2 要素 OTP に指定されたログインスキーマのパスワード属性を使用します。

   ```
   AAA.USER.ATTRIBUTE(1)
   ```

5. [トラフィックポリシー] タブで、[追加] をクリックします。
6. [**Name**] フィールドに、トラフィックポリシーの名前を入力します。
7. [リクエストプロファイル（Request Profile）] フィールドで、作成したトラフィックプロファイルを選択します。
8. [式] ボックスに **True** と入力します。ご使用の NetScaler Gateway 仮想サーバーがフル VPN を許可している場合は、式を次のように変更してください。

   ```
   http.req.method.eq(post)||http.req.method.eq(get)&& false
   ```

9. [作成] をクリックします。
10. トラフィックポリシーを VPN 仮想サーバーにバインドします。

    - [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**認証プロファイル] に移動します。
    - **NetScaler Gateway** 仮想サーバーを選択して認証プロファイルを構成し、[**OK**] をクリックします。
    - [**NetScaler Gateway] > [NetScaler Gateway** 仮想サーバー] に移動し、NetScaler Gateway 仮想サーバーを選択します。「**VPN** 仮想サーバー」ページが表示されます。
    - 「ポリシー」セクションで、「+」アイコンをクリックします。
    - ポリシータイプを「トラフィック」として選択し、「続行」をクリックします。

- トラフィックポリシーを選択し、[バインド] をクリックします。
- ［完了］ をクリックします。

OTP を管理するためのコンテンツスイッチングポリシーを構成する

Unified Gateway を使用している場合は、次の構成が必要です。

1. **Traffic Management > Content Switching > Policies** に移動します。コンテンツスイッチングポリシーを選択し、右クリックして [編集] を選択します。
2. 式を編集して次の OR ステートメントを評価し、「**OK**」をクリックします。

```
is_vpn_url || HTTP.REQ.URL.CONTAINS("manageotp"
```

CLI を使用したネイティブ OTP の設定

OTP デバイス管理ページを設定するには、次の情報が必要です。

- 認証仮想サーバーに割り当てられた IP
- 割り当てられた IP に対応する FQDN
- 認証仮想サーバーのサーバー証明書

> 注:
>
> ネイティブ OTP は Web ベースのソリューションのみです。

OTP デバイスの登録および管理ページを設定するには

認証仮想サーバーの作成

```
1 ```
2 add authentication vserver authvs SSL 1.2.3.5 443
3 bind authentication vserver authvs -portaltheme RFWebUI
4 bind ssl vserver authvs -certkeyname otpauthcert
5 <!--NeedCopy--> ```
```

> 注:
>
> 認証仮想サーバーは RFWebUI ポータルテーマにバインドする必要があります。サーバー証明書をサーバーにバインドします。サーバー IP ‘1.2.3.5’ には、後で使用できるように otpauth.server.com という対応する FQDN が必要です。

LDAP ログオンアクションを作成するには

```
add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP>
    -  serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
    ldapBindDnPassword <PASSWO> -ldapLoginName <USER FORMAT>
```

例:

```
add authentication ldapAction ldap_logon_action -serverIP 1.2.3.4 -
    serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
    administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
    ldapLoginName userprincipalname
```

LDAP ログオンの認証ポリシーを追加するには

```
add authentication Policy auth_pol_ldap_logon -rule true -action
    ldap_logon_action
```

Loginschema を使用して **UI** を表示するには

ログオン時にユーザー名フィールドとパスワードフィールドをユーザーに表示する

```
add authentication loginSchema lschema_single_auth_manage_otp -
    authenticationSchema "/nsconfig/loginschema/LoginSchema/
    SingleAuthManageOTP.xml"
```

デバイスの登録と管理ページを表示する

デバイスの登録と管理画面を表示するには、URL とホスト名の 2 つの方法をお勧めします。

> 注:
>
> 現在、デバイス登録とデバイス管理はブラウザを使用してのみ実行できます。

- **URL** を使う

  URL に‘/manageotp’が含まれている場合

```
- add authentication loginSchemaPolicy lpol_single_auth_manage_otp_by_ur
   -rule "http.req.cookie.value("NSC_TASS").contains("manageotp
  ")"-action lschema_single_auth_manage_otp
- bind authentication vserver authvs -policy lpol_single_auth_manage_otp
   -priority 10 -gotoPriorityExpression END
```


- ホスト名を使う

  ホスト名が「alt.server.com」の場合

```
- add authentication loginSchemaPolicy lpol_single_auth_manage_otp_by_ho
   -rule "http.req.header("host").eq("alt.server.com")"-action
  lschema_single_auth_manage_otp
- bind authentication vserver authvs -policy lpol_single_auth_manage_otp
   -priority 20 -gotoPriorityExpression END
```

CLI を使用してユーザログインページを設定するには

[ユーザーログオン（User Logon）] ページを設定するには、次の情報が必要です。

- 負荷分散仮想サーバーの IP
- 負荷分散仮想サーバーの対応する FQDN
- 負荷分散仮想サーバーのサーバー証明書

```
1  bind ssl virtual server lbvs_https -certkeyname lbvs_server_cert
2  <!--NeedCopy-->
```

負荷分散におけるバックエンドサービスは、次のように表されます。

```
1  ```
2  add service iis_backendsso_server_com 1.2.3.210 HTTP 80
3  bind lb vserver lbvs_https iis_backendsso_server_com
4  <!--NeedCopy--> ```
```

OTP パスコード検証アクションを作成するには

```
1  add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP>
       -serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
       ldapBindDnPassword <PASSWORD> -ldapLoginName <USER FORMAT> -
       authentication DISABLED -OTPSecret <LDAP ATTRIBUTE>`
```

例:

```
1  add authentication ldapAction ldap_otp_action -serverIP 1.2.3.4 -
       serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
       administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
       ldapLoginName userprincipalname -authentication DISABLED -OTPSecret
       userParameters
```

> 重要:
>
> LDAP ログオンと OTP アクションの違いは、認証を無効にして新しいパラメータOTPSecretを導入する必要があることです。AD 属性値は使用しないでください。

OTP パスコード検証の認証ポリシーを追加するには

```
add authentication Policy auth_pol_otp_validation -rule true -action
    ldap_otp_action
```

LoginSchema を介して **2** 要素認証を提示するには　2 要素認証用の UI を追加します。

```
add authentication loginSchema lscheme_dual_factor -
    authenticationSchema "/nsconfig/loginschema/LoginSchema/DualAuth.xml
    "
add authentication loginSchemaPolicy lpol_dual_factor -rule true -
    action lscheme_dual_factor
```

ポリシーラベルを介してパスコード検証係数を作成するには　次の要素の管理 OTP フローポリシーラベルを作成します（最初の要素は LDAP ログオンです）

```
add authentication loginSchema lschema_noschema -authenticationSchema
    noschema
add authentication policylabel manage_otp_flow_label -loginSchema
    lschema_noschema
```

OTP ポリシーをポリシー・ラベルにバインドするには

```
bind authentication policylabel manage_otp_flow_label -policyName
    auth_pol_otp_validation -priority 10 -gotoPriorityExpression NEXT
```

UI フローをバインドするには　LDAP ログオンに続いて、認証仮想サーバーを使用した OTP 検証をバインドします。

```
bind authentication vserver authvs -policy auth_pol_ldap_logon -
    priority 10 -nextFactor  manage_otp_flow_label -
    gotoPriorityExpression NEXT
bind authentication vserver authvs -policy lpol_dual_factor -priority
    30 -gotoPriorityExpression END
```

シングルサインオン用のトラフィックポリシーを作成して **VPN** 仮想サーバーにバインドするには

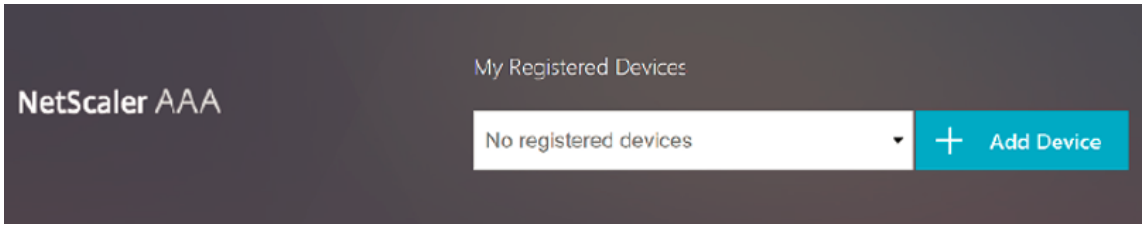
```
add vpn trafficAction vpn_html_pol http -userExpression aaa.user.
    attribute(1) -passwdExpression aaa.user.attribute(2)

add vpn trafficpolicy tf1 'http.req.method.eq(post)||http.req.method.eq
    (get) && false' vpn_html_pol

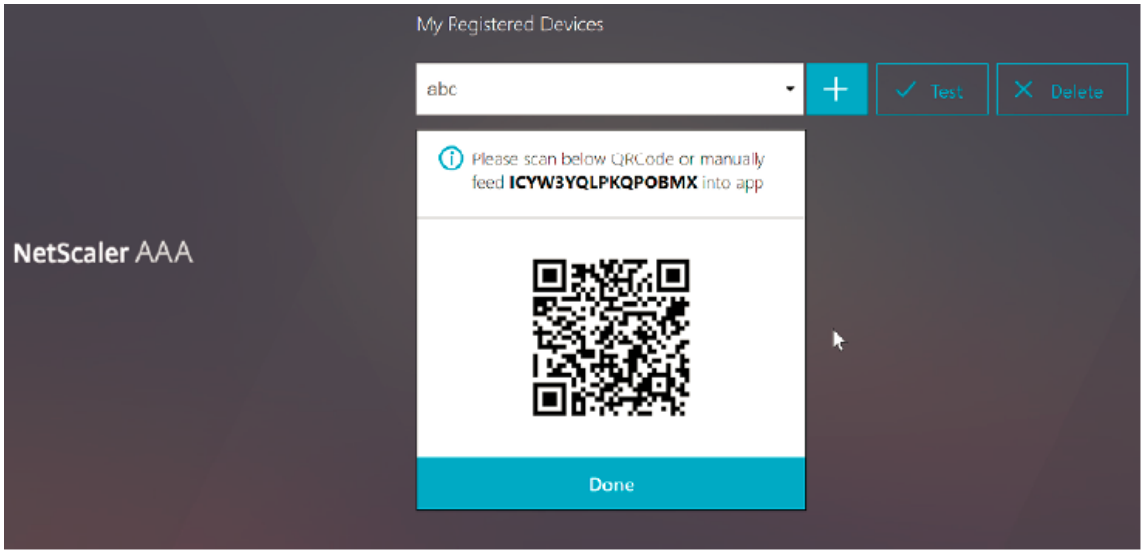
bind vpn vserver vpn1 -policy tf1 -priority 10
<!--NeedCopy-->
```


デバイスを **NetScaler** に登録する

1. ブラウザで、/manageotp というサフィックスが付いた NetScaler FQDN（最初に公開されている IP）に移動します。たとえば、https://otpauth.server.com/manageotp。ユーザーの資格情報を使用してログインします。
2. [**+**] アイコンをクリックしてデバイスを追加します。

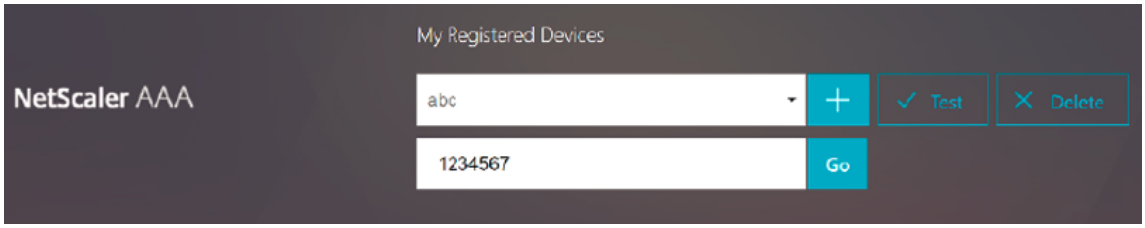


3. デバイス名を入力して **Go** を押します。画面にバーコードが表示されます。
4. [セットアップの開始] をクリックし、[バーコードのスキャン] をクリックします。
5. デバイスのカメラを QR コードの上に置きます。オプションでコードを入力できます。



> 注:
>
> 表示された QR コードは 3 分間有効です。

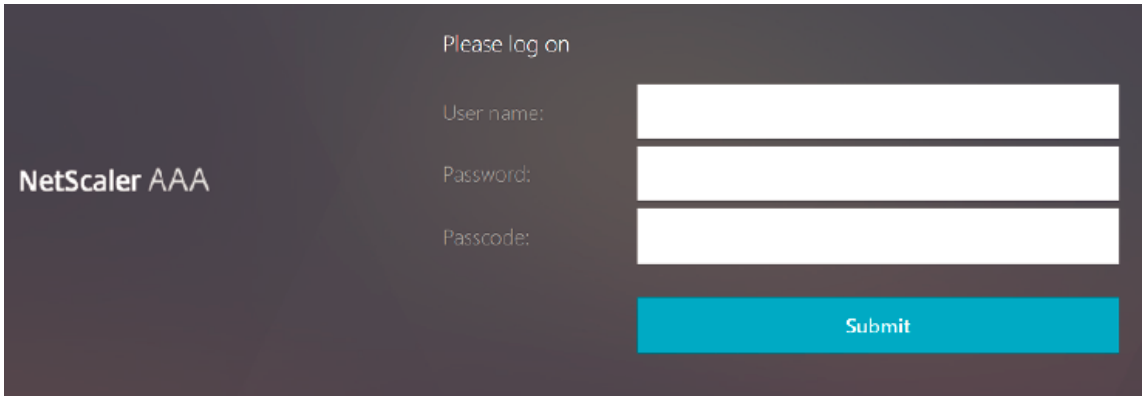
6. スキャンが成功すると、ログインに使用できる 6 桁の時刻依存コードが表示されます。



7. テストするには、QR 画面で [完了] をクリックし、右側の緑色のチェックマークをクリックします。
8. ドロップダウンメニューからデバイスを選択し、Google Authenticator のコード（赤ではなく青である必要があります）を入力し、[**Go**] をクリックします。
9. ページの右上隅にあるドロップダウンメニューを使用して、必ずログアウトしてください。

OTP を使用して **NetScaler** にログインします

1. 最初に公開する URL に移動し、Google Authenticator の OTP を入力してログオンします。
2. NetScaler のスプラッシュページを認証します。



OTP シークレットデータを暗号化された形式で保存する

December 8, 2023

NetScaler リリース 13.0 ビルド 41.20 以降、OTP シークレットデータをプレーンテキストではなく暗号化された形式で保存できるようになりました。

以前は、NetScaler アプライアンスは OTP シークレットをプレーンテキストとして AD に保存していました。OTP シークレットをプレーンテキストで保存すると、悪意のある攻撃者や管理者が他のユーザーの共有シークレットを閲覧してデータを悪用する可能性があるため、セキュリティ上の脅威となります。

暗号化パラメータは AD の OTP シークレットの暗号化を可能にします。NetScaler バージョン 13.0 ビルド 41.20 に新しいデバイスを登録し、暗号化パラメーターを有効にすると、OTP シークレットはデフォルトで暗号化された形式で保存されます。ただし、暗号化パラメータが無効になっている場合、OTP シークレットはプレーンテキスト形式で保存されます。

13.0 ビルド 41.20 より前に登録されたデバイスの場合は、ベストプラクティスとして以下を実行する必要があります。

1. 13.0 NetScaler アプライアンスを 13.0 ビルド 41.20 にアップグレードします。

2. アプライアンスの暗号化パラメータを有効にします。
3. OTP シークレット移行ツールを使用して、OTP シークレットデータをプレーンテキスト形式から暗号化形式に移行します。

OTP シークレット移行ツールの詳細については、「OTP 暗号化ツール」を参照してください。

> 重要
>
> : Citrix では、管理者に次の基準が満たされていることを確認することをお勧めします。
>
> - セルフサービスパスワードリセット機能の一部として KBA を使用していない場合は、OTP シークレットを暗号化するように新しい証明書を構成する必要があります。
>
> – To bind the certificate to VPN global, you can use the following command:
>
>  ```
>  bind vpn global -userDataEncryptionKey <certificate name>
>  ```
>
> - すでに証明書を使用して KBA を暗号化している場合は、同じ証明書を使用して OTP シークレットを暗号化できます。
> - 新しい OTP 登録は、常に最後にバインドされた証明書が最も優先されるため、その証明書で行われます。以下の例では、証明書 (cert1) をバインドしてから別の証明書 (cert2) をバインドすると、cert2 がデバイス登録の対象とみなされます。デバイス登録に必要な証明書がない場合、エンドユーザーのログインは失敗します。
>
>  ```
>  1    bind vpn global -userDataEncryptionKey otp-cert1
>  2    bind vpn global -userDataEncryptionKey otp-cert2
>  3    <!--NeedCopy-->
>  ```
>
> 次の例では、`cert2`証明書は`show vpn global`コマンド出力の最初のエントリとして表示されます。
>
> “‘
>
> show vpn global
>
> ポータルテーマ:RFWebUI
> ユーザデータ暗号化証明書:cert2
> ユーザデータ暗号化証明書:cert1
> 1) VPN クライアントレスアクセスポリシー名: ns_cvpn_owa_policy 優先度:95000
> バインドポイント:REQ_DEFAULT
> 2) VPN クライアントレスアクセスポリシー名:ns_cvpn_sp_policy 優先度:96000
> バインドポイント:REQ_DEFAULT
> 3) VPN クライアントレスアクセスポリシー名: ns_cvpn_sp2013_policy 優先度:97000
> バインドポイント:REQ_DEFAULT
> 4) VPN クライアントレスアクセスポリシー名:ns_cvpn_default_policy 優先度:100000

> バインドポイント:REQ_DEFAULT
> “

CLI を使用して **OTP** 暗号化データを有効にするには

コマンドプロンプトで入力します:

```
set aaa otpparameter [-encryption ( ON | OFF )]
```

例

```
set aaa otpparameter -encryption ON
```

GUI を使用して **OTP** 暗号化を設定するには

1. [セキュリティ**] > [AAA-**アプリケーショントラフィック] に移動し、[****** 認証設定**]** セクションで **[**認証 AAA OTP パラメータの変更 **]
2. 「**AAA OTP** パラメータの設定」ページで、「**OTP** シークレット暗号化」を選択します。
3. [OK] をクリックします。

OTP 通知を受信するエンドユーザーデバイス数の設定

管理者は、エンドユーザーが OTP 通知または認証を受け取るために登録できるデバイスの数を設定できるようになりました。

CLI を使用して **OTP** のデバイス数を設定するには

コマンドプロンプトで入力します:

```
set aaa otpparameter [-maxOTPDevices <positive_integer>]
```

例

```
set aaa otpparameter -maxOTPDevices 4
```

GUI を使用してデバイス数を設定するには

1. [セキュリティ**] > [AAA** ―アプリケーショントラフィック] に移動します。
2. [認証設定] セクションで、[認証 **AAA OTP** パラメータの変更] をクリックします。
3. [**AAA OTP** パラメータの設定**]** ページで、[設定済みの **OTP** デバイスの最大数] の値を入力します。
4. [**OK**] をクリックします。

OTP 暗号化ツール

December 8, 2023

NetScaler リリース 13.0 ビルド 41.20 以降、セキュリティを強化するため、OTP シークレットデータはプレーンテキストではなく暗号化された形式で保存されます。OTP シークレットの暗号化形式での保存は自動的に行われ、手動による介入は不要です。

以前は、NetScaler アプライアンスは OTP シークレットをプレーンテキストとして Active Directory に保存していました。OTP シークレットをプレーンテキスト形式で保存すると、悪意のある攻撃者や管理者が他のユーザーの共有シークレットを表示してデータを悪用できるため、セキュリティ上の脅威となります。

OTP 暗号化ツールには次の利点があります。

- 古い形式 (プレーンテキスト) を使用している古いデバイスがあっても、データが失われることはありません。
- 古い NetScaler Gateway バージョンとの下位互換性サポートにより、既存のデバイスと新しいデバイスの統合とサポートに役立ちます。
- OTP 暗号化ツールは、管理者が一度にすべてのユーザーのすべての OTP 秘密データを移行するのに役立ちます。

> 注:
>
> OTP 暗号化ツールは、KBA 登録データまたは電子メール登録データを暗号化または復号化しません。

OTP 暗号化ツールの使用

OTP 暗号化ツールは、次の目的で使用できます。

- 暗号化。OTP シークレットを暗号化された形式で保存します。このツールは、NetScaler に登録されているデバイスの OTP データを抽出し、プレーンテキスト形式の OTP データを暗号化形式に変換します。
- 復号化。OTP シークレットをプレーンテキスト形式に戻します。
- 証明書を更新します。管理者は、いつでも証明書を新しい証明書に更新できます。管理者は、ツールを使用して新しい証明書を入力し、すべてのエントリを新しい証明書データで更新できます。証明書パスは、絶対パスまたは相対パスのいずれかである必要があります。

> 重要
>
> - OTP 暗号化ツールを使用するには、NetScaler アプライアンスの暗号化パラメーターを有効にする必要があります。
> - ビルド 41.20 より前に NetScaler に登録されたデバイスの場合は、以下を実行する必要があります。
>   - Upgrade the 13.0 NetScaler appliance to 13.0 build 41.20.
>   - Enable the encryption parameter on the appliance.
>   - Use the OTP Secret migration tool to migrate OTP secret data from plain text format to

> encrypted format.
>
> - OTP 暗号化ツールは単一値のユーザー属性のみをサポートします。複数値のユーザー属性はサポートされていません。

プレーンテキスト形式の **OTP** シークレットデータ

例:

```
#@devicename=<16 or more bytes>&tag=<64bytes>&,
```

ご覧のとおり、古いフォーマットの開始パターンは常に「#@」で、終了パターンは常に「&」です。「devicename=」と終了パターンの間にあるすべてのデータは、ユーザ OTP データを構成します。

暗号化された形式の **OTP** シークレットデータ

OTP データの新しい暗号化形式は、次の形式になります。

例:

```
    {

         "otpdata” : {

          “devices” : {

                              “device1” :  “value1” ,
                              “device2” :  “value2” , …
                        }

             }

      }

<!--NeedCopy-->
```

ここで、値 1 は KID + IV + 暗号データの base64 でエンコードされた値です

暗号データは次のように構成されています。

```
    {

      secret:<16-byte secret>,
      tag : <64-byte tag value>
      alg: <algorithm used> (not mandatory, default is sha1, specify
          the algorithm only if it is not default)
      }

<!--NeedCopy-->
```


- 「デバイス」には、それぞれの名前に対する価値があります。値は base64 エンコード (KID) .base64 エンコード (IV) .base64 エンコード (暗号データ) です。
- KID は、OTP シークレットデータ暗号化に使用される証明書を識別するために使用されるキー ID 値です。キー ID は、OTP シークレットデータの暗号化に複数の証明書を使用する場合に特に役立ちます。
- 標準の AES アルゴリズムでは、IV は常に最初の 16 バイトまたは 32 バイトの暗号データとして送信されます。同じモデルに従うことができます。
- IV はデバイスごとに異なりますが、キーは変わりません。

> 注:
>
> 暗号化された形式の OTP データは、ユーザー属性 AD に保存されます。

OTP 暗号化ツールのセットアップ

> 注:
>
> OTP 暗号化ツールを実行するには、NetScaler アプライアンスではなく、Python 環境の代替プラットフォームを使用することをお勧めします。

OTP 暗号化ツールはディレクトリ`\var\netscaler\otptool`にあります。NetScaler ソースからコードをダウンロードし、必要な AD 認証情報を使用してツールを実行する必要があります。

- OTP 暗号化ツールを使用するための前提条件は次のとおりです。
  - このツールを実行している環境に python 3.5 以降のバージョンをインストールします。
  - pip3 以降のバージョンをインストールします。
- 次のコマンドを実行します:
  - **pip install-r requirements.txt**。要件を自動的にインストールします。
  - **python main.py**。OTP 暗号化ツールを起動します。OTP シークレットデータの移行の必要性に応じて、必要な引数を指定する必要があります。
- このツールは、シェルプロンプトから`\var\netscaler\otptool`に配置できます。
- 必要な AD 認証情報を使用してツールを実行します。

OTP 暗号化ツールインターフェイス

次の図に、OTP 暗号化ツールのインターフェイスの例を示します。インターフェイスには、暗号化/復号化/証明書のアップグレード用に定義する必要のあるすべての引数が含まれています。また、各引数の簡単な説明も取り込まれます。

オペレーション引数

暗号化、復号化、または証明書のアップグレードに OTP 暗号化ツールを使用するには、OPERATION 引数を定義する必要があります。

次の表は、OTP 暗号化ツールおよび対応する OPERATION 引数の値を使用できるシナリオの一部をまとめたものです。

シナリオ	演算引数の値とその他の引数
プレーンテキスト OTP シークレットを同じ属性で暗号化された形式に変換する	OPERATION 引数の値に 0 を入力し、ソース属性とターゲット属性に同じ値を指定します。例: `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute unixhomedirectory -operation 0 -cert_path aaatm_wild_all.cert`
プレーンテキスト OTP シークレットを別の属性で暗号化された形式に変換する	OPERATION 引数の値に 0 を入力し、ソース属性とターゲット属性に対応する値を指定します。例: `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute userparameters -operation 0 -cert_path aaatm_wild_all.cert`
暗号化されたエントリをプレーンテキストに変換し直す	OPERATION 引数の値に 1 を入力し、ソース属性とターゲット属性に対応する値を指定します。例: `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute userparameters -operation 1 -cert_path aaatm_wild_all.cert`

シナリオ	演算引数の値とその他の引数
証明書を新しい証明書に更新する	OPERATION 引数の値に 2 を入力し、対応する引数に以前の証明書と新しい証明書の詳細をすべて入力します。 例: `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -operation 2 -cert_path aaatm_wild_all.cert -new_cert_path aaatm_wild_all_new.cert`

CERT_PATH 引数

CERT_PATH 引数は、NetScaler でデータを暗号化するために使用される証明書を含むファイルです。ユーザーは、暗号化、復号化、および更新証明書の **3** つの操作すべてに対してこの引数を指定する必要があります。

CERT_PATH 引数ファイルには、証明書と関連する秘密キーの両方を PEM または CERT 形式で含める必要があります (pfx はサポートされていません)。

たとえば、certificate.cert ファイルと certificate.key ファイルが証明書ファイルとその秘密鍵に対応している場合、UNIX ライクなシステムでは、次のコマンドで cert_path フラグの値として使用できるファイル`certkey.merged`を作成します。

```
$ cat certificate.cert certificate.key > certkey.merged
$
<!--NeedCopy-->
```

> 証明書に関する注意点
>
> - ユーザーは、ユーザーデータを暗号化するために NetScaler アプライアンスでグローバルにバインドされているのと同じ証明書を提供する必要があります。
> - 証明書には、Base64 でエンコードされたパブリック証明書とそれに対応する RSA 秘密鍵を同じファイルに含める必要があります。
> - 証明書の形式は PEM または CERT のいずれかである必要があります。証明書は X509 形式に準拠している必要があります。
> - パスワードで保護された証明書形式および*.pfx* ファイルは、このツールでは受け入れられません。ユーザーは、ツールに証明書を提供する前に、PFX 証明書を*.cert* に変換する必要があります。

検索フィルタ引数

SEARCH_FILTER 引数は、Active Directory ドメインまたはユーザーをフィルタリングするために使用されます。

例:

- `-search_filter "(sAMAccountName=OTP*)"`: SAMAccountNames (ユーザーログオン名) が「OTP」で始まるユーザーをフィルタリングします。
- `-search_filter "(objectCategory=person)"`: person タイプのオブジェクトカテゴリをフィルタリングします。
- `-search_file "(objectclass=*)"`: すべてのオブジェクトをフィルタリングします。

NetScaler アプライアンスの暗号化オプションを有効にする

プレーンテキスト形式を暗号化するには、NetScaler アプライアンスの暗号化オプションを有効にする必要があります。

CLI を使用して OTP 暗号化データを有効にするには、コマンドプロンプトで次のように入力します。

```
set aaa otpparameter [-encryption ( ON | OFF )]
```

例:

```
set aaa otpparameter -encryption ON
```

OTP 暗号化ツールの使用例

OTP 暗号化ツールは、次のユースケースに使用できます。

NetScaler アプライアンスバージョン 13.0 ビルド 41.20 に新しいデバイスを登録する

新しいデバイスを NetScaler アプライアンスバージョン 13.0 ビルド 41.x に登録し、暗号化オプションが有効になっている場合、OTP データは暗号化された形式で保存されます。手動による介入を避けることができます。

暗号化オプションが有効になっていない場合、OTP データはプレーンテキスト形式で保存されます。

13.0 build 41.20 より前に登録されたデバイスの OTP データを移行する

13.0 ビルド 41.20 より前の NetScaler アプライアンスに登録されているデバイスの OTP シークレットデータを暗号化するには、次の手順を実行する必要があります。

- 変換ツールを使用して、OTP データをプレーンテキスト形式から暗号化形式に移行します。
- NetScaler アプライアンスの「暗号化」パラメーターを有効にします。

- CLI を使用して暗号化オプションを有効にするには、次の手順を実行します。
  - `set aaa otpparameter -encryption ON`
- GUI を使用して暗号化オプションを有効にするには、次の手順を実行します。
  - [セキュリティ**] > [AAA-**アプリケーショントラフィック] に移動し、[****** 認証設定**]** セクションで **[**認証 AAA OTP パラメータの変更 **]
  - [**AAA OTP** パラメータの設定] ページで、[**OTP** シークレット暗号化] を選択し、[**OK**] をクリックします。
- 有効な AD 資格情報を使用してログインします。
- 必要な場合は、さらにデバイスを登録します (オプション)。

暗号化されたデータを古い証明書から新しい証明書に移行する

管理者が証明書を新しい証明書に更新する場合、ツールには新しい証明書データエントリを更新するオプションが用意されています。

CLI を使用して証明書を新しい証明書に更新するには

コマンドプロンプトで入力します:

例:

```
python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa
.local -search_base cn=users,dc=aaa,dc=local -source_attribute
unixhomedirectory -target_attribute userparameters -operation 2 -
cert_path aaatm_wild_all.cert -new_cert_path aaatm_wild_all_new.cert
```

> 注
>
> - 証明書には秘密鍵と公開鍵の両方が必要です。
> - 現在、この機能は OTP に対してのみ提供されています。

アプライアンスを暗号化して **13.0 build 41.20** にアップグレードした後に登録されたデバイスの再暗号化または新しい証明書への移行

管理者は、証明書ですでに暗号化されているデバイスでこのツールを使用し、その証明書を新しい証明書で更新できます。

暗号化されたデータをプレーンテキスト形式に変換し直す

管理者は OTP シークレットを復号化して、元のプレーンテキスト形式に戻すことができます。OTP 暗号化ツールは、すべてのユーザーをスキャンして暗号化形式の OTP シークレットを探し、復号化された形式に変換します。

CLI を使用して証明書を新しい証明書に更新するには

コマンドプロンプトで入力します:

例:

```
python3 main.py -Host 192.0.2.1 - Port 636 -username ldapbind_user@aaa
    .local -search_base cn=users,dc=aaa,dc=local -source_attribute
    unixhomedirectory -target_attribute userparameters -operation 1
<!--NeedCopy-->
```

トラブルシューティング

このツールは、次のログファイルを生成します。

- **app.log**。すべての主要な実行ステップと、エラー、警告、および失敗に関する情報を記録します。
- **unmodified_users.txt**。プレーンテキスト形式から暗号化形式にアップグレードされなかったユーザ DN の一覧が含まれます。これらのログは、形式のエラーとして生成されるか、他の理由が原因である可能性があります。

OTP のプッシュ通知

February 15, 2024

NetScaler Gateway は OTP のプッシュ通知をサポートしています。ユーザーは、登録したデバイスで受信した OTP を手動で入力して NetScaler Gateway にログインする必要はありません。管理者は、プッシュ通知サービスを使用してログイン通知がユーザーの登録デバイスに送信されるように NetScaler Gateway を構成できます。通知を受け取ったユーザーは、通知の [許可] をタップするだけで NetScaler Gateway にログインできます。ゲートウェイは、ユーザーからの確認応答を受信すると、リクエストのソースを特定し、そのブラウザ接続に応答を送信します。

タイムアウト期間（30 秒）内に通知応答が受信されない場合、ユーザーは NetScaler Gateway ログインページにリダイレクトされます。その後、ユーザは OTP を手動で入力するか、または [通知を再送する（**Resend Notification**）] をクリックして、登録されたデバイスで通知を再度受信できます。

管理者は、プッシュ通知用に作成されたログインスキーマを使用して、プッシュ通知認証をデフォルトの認証として設定できます。

> 重要:
>
> プッシュ通知機能は NetScaler Premium エディションのライセンスで利用できます。

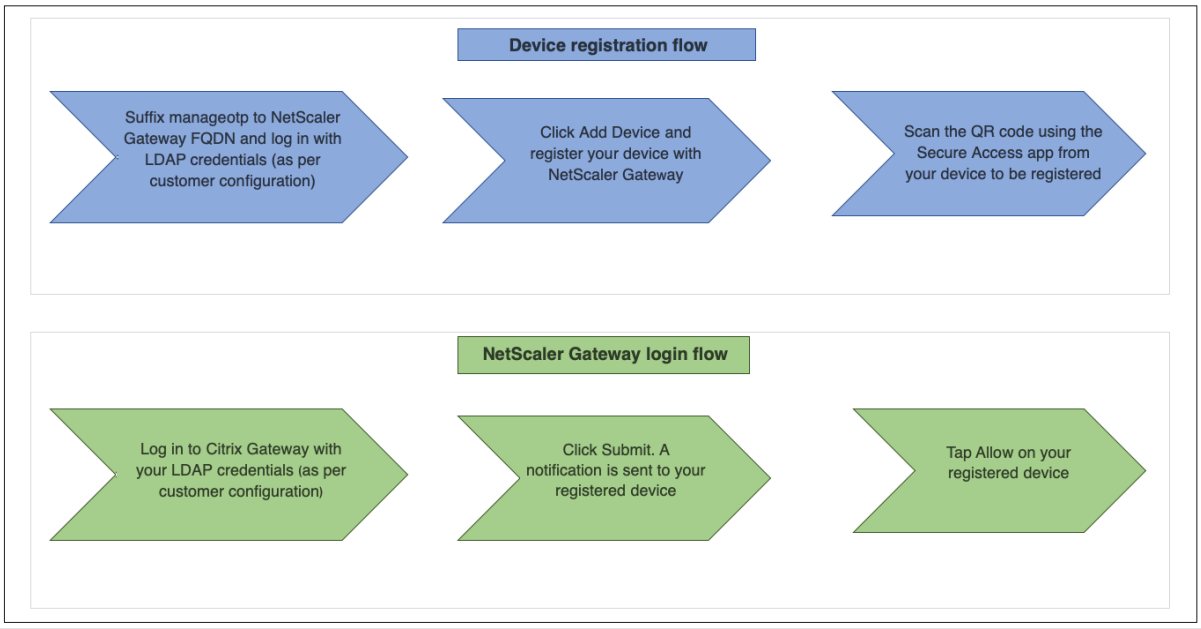
プッシュ通知の利点

- プッシュ通知は、より安全な多要素認証メカニズムを提供します。NetScaler Gateway への認証は、ユーザーがログインの試行を承認するまで成功しません。
- プッシュ通知は管理と使用が簡単です。ユーザーは、管理者の支援を必要としない Citrix SSO モバイルアプリをダウンロードしてインストールする必要があります。
- ユーザーはコードをコピーしたり覚えたりする必要はありません。認証を受けるには、デバイスをタップするだけで済みます。
- ユーザーは複数のデバイスを登録できます。

プッシュ通知の動作

プッシュ通知ワークフローは、次の 2 つのカテゴリに分類できます。

- デバイス登録
- エンドユーザログイン



プッシュ通知を使用するための前提条件

- Citrix Cloud のオンボーディングプロセスを完了します。
  1. Citrix Cloud 企業アカウントを作成するか、既存のアカウントに参加します。詳細なプロセスと手順については、「Citrix Cloud へのサインアップ」を参照してください。
  2. https://citrix.cloud.comにログインし、顧客を選択します。

3. メニューから [**ID** とアクセス管理] を選択し、[**API** アクセス] タブに移動して、アカウントのクライアントを作成します。
4. ID、シークレット、および顧客 ID をコピーします。NetScaler のプッシュサービスをそれぞれ「ClientId」と「ClientSecret」として構成するには、ID とシークレットが必要です。

> 重要:
>
> - 同じ API 認証情報を複数のデータセンターで使用できます。
> - オンプレミスの NetScaler アプライアンスは、サーバーアドレス mfa.cloud.com と trust.citrixworkspacesapi.net を解決でき、アプライアンスからアクセスできる必要があります。これは、ポート 443 を介してこれらのサーバーのファイアウォールまたは IP アドレスブロックがないことを保証するためです。

- iOS デバイス用と Android デバイス用に、それぞれ App Store と Play Store から Citrix SSO モバイルアプリをダウンロードします。プッシュ通知は、iOS 2.3.5 から Android のビルド 1.1.13 からサポートされています。
- Active Directory について、次のことを確認します。
  - 属性の最小長は 256 文字以上にする必要があります。
  - 属性タイプは、ユーザパラメータなどの‘DirectoryString’でなければなりません。これらの属性は文字列値を保持できます。
  - デバイス名が英語以外の文字である場合、属性文字列タイプは Unicode である必要があります。
  - NetScaler LDAP 管理者は、選択した AD 属性への書き込みアクセス権を持っている必要があります。
  - NetScaler とクライアントマシンは、共通のネットワークタイムサーバーと同期している必要があります。

プッシュ通知の設定

プッシュ通知機能を使用するために完了する必要がある大まかな手順は次のとおりです。

- NetScaler Gateway 管理者は、ユーザーを管理および検証するためのインターフェイスを構成する必要があります。
  1. プッシュサービスを設定します。
  2. OTP 管理とエンドユーザーログイン用に NetScaler Gateway を構成します。

     ユーザーが NetScaler Gateway にログインするには、デバイスをゲートウェイに登録する必要があります。
  3. デバイスを NetScaler Gateway に登録します。
  4. NetScaler Gateway にログインします。

プッシュサービスを作成する

1. セキュリティ > **AAA** アプリケーショントラフィック > ポリシー > 認証 > 高度なポリシー > アクション > プッシュサービスに移動し、追加をクリックします。
2. [名前] に、プッシュサービスの名前を入力します。
3. [クライアント **ID**] に、クラウド内の NetScaler Push サーバーと通信する依存パーティの固有の ID を入力します。
4. [クライアントシークレット] に、クラウド内の NetScaler Push サーバーと通信するための依存パーティの固有のシークレットを入力します。
5. [顧客 **ID**] に、クライアント ID とクライアントシークレットのペアの作成に使用される、クラウド内の顧客 ID またはアカウントの名前を入力します。

> 重要
>
> プッシュサービスには、TLS 1.2 バージョンが必要です。詳細については、TLS 1.2 の設定の詳細を参照してください。

OTP 管理とエンドユーザーログイン用の **NetScaler Gateway** の設定

OTP 管理とエンドユーザログインについては、次の手順を実行します。

- OTP 管理用のログインスキーマの作成
- 認証、承認、および監査仮想サーバーの構成
- VPN または負荷分散仮想サーバーを構成する
- ポリシーラベルの設定
- エンド・ユーザー・ログイン用のログイン・スキーマの作成

設定の詳細については、ネイティブ OTP サポートを参照してください。

重要: プッシュ通知の場合、管理者は以下を明示的に設定する必要があります。

- プッシュサービスを作成します。
- OTP 管理用のログインスキーマを作成する際には、必要に応じて SingleAuthManageOTP.xml ログインスキーマまたは同等のものを選択します。
- エンドユーザーログイン用のログインスキーマを作成する際には、必要に応じて DualAuthOrPush.xml ログインスキーマまたは同等のものを選択します。

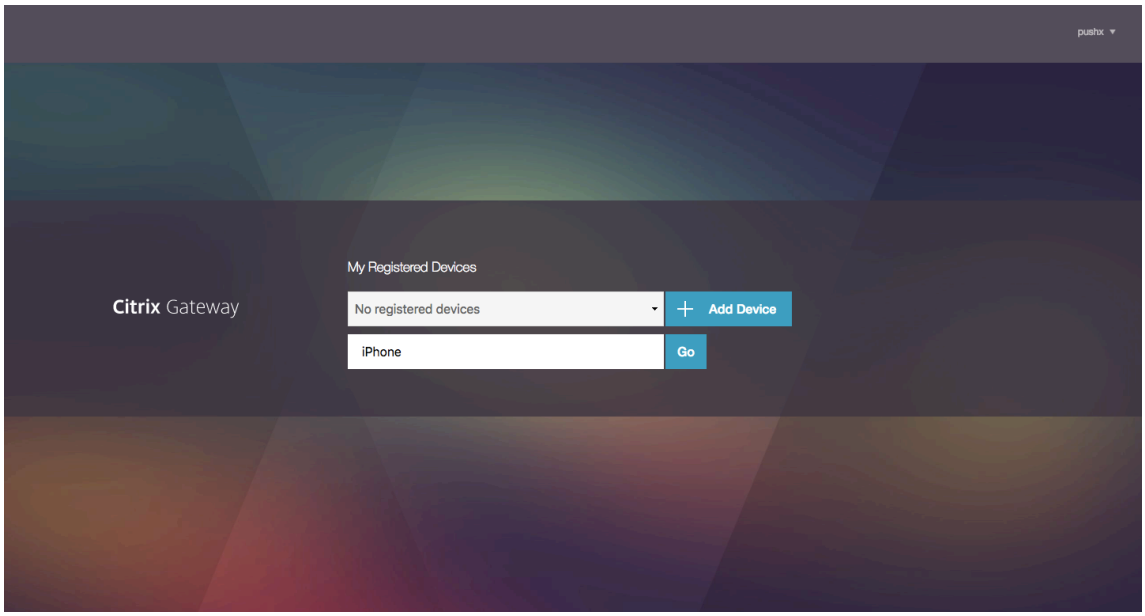
デバイスを **NetScaler Gateway** に登録する

プッシュ通知機能を使用するには、ユーザーがデバイスを NetScaler Gateway に登録する必要があります。

1. **Web** ブラウザで **NetScaler Gateway** の **FQDN** を参照し、その **FQDN** の末尾に**/manageotp** を付けます。

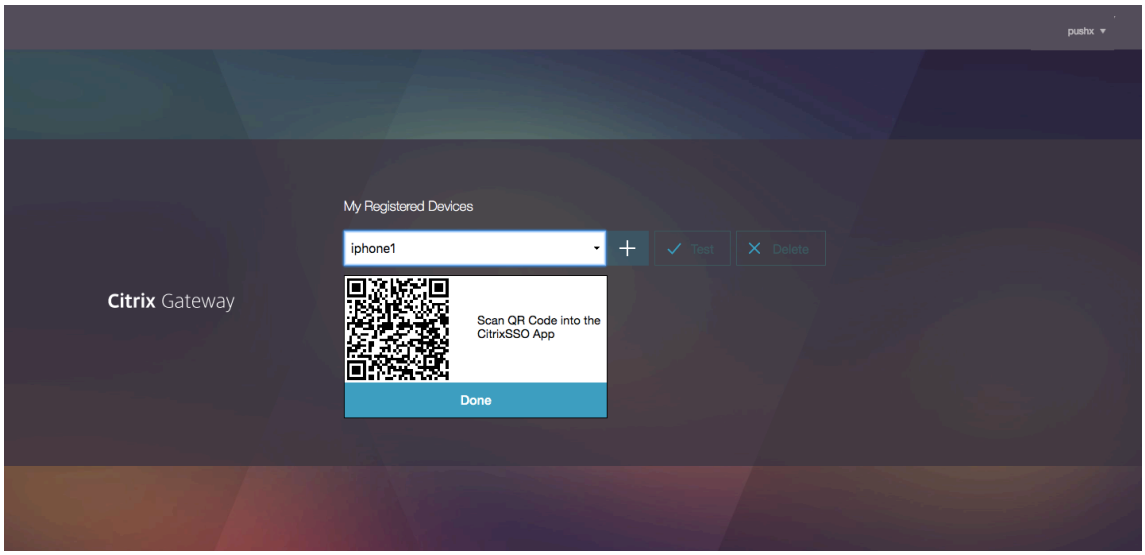
   これにより、認証ページが読み込まれます。
   例：https://gateway.company.com/manageotp

2. 必要に応じて、LDAP 認証情報または適切な 2 要素認証メカニズムを使用してログインします。



3. ［デバイスを追加］をクリックします。
4. デバイスの名前を入力し、［実行］をクリックします。

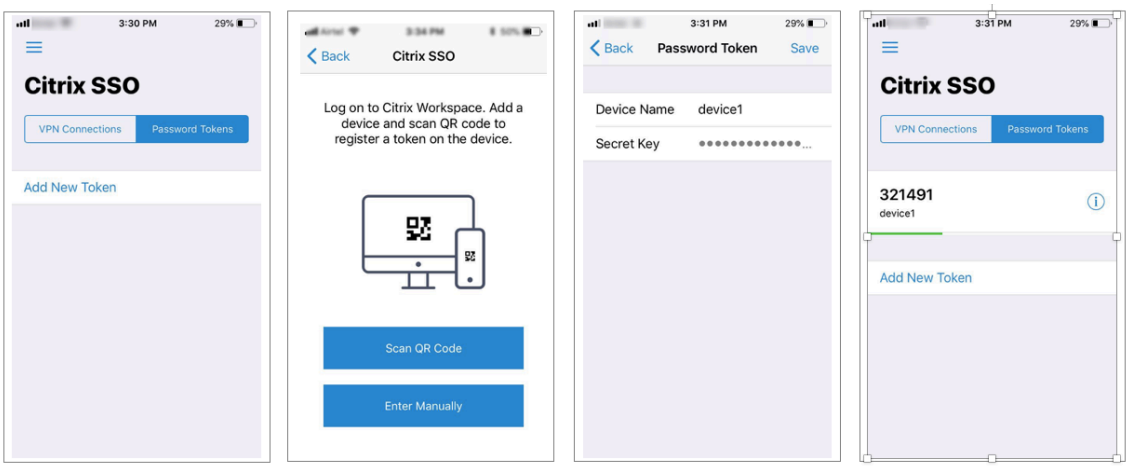
   QR コードは NetScaler Gateway のブラウザページに表示されます。



5. 登録するデバイスから Citrix SSO アプリを使用してこの QR コードをスキャンします。

Citrix SSO は QR コードを検証し、プッシュ通知のためにゲートウェイに登録します。登録プロセスでエラーがなければ、トークンはパスワードトークンページに正常に追加されます。

> 重要:
>
> QR コードに記載されているシークレットキーを手動で入力すると、ログインに失敗します。



6. 追加/管理するデバイスがない場合は、ページの右上隅にあるリストを使用してログアウトします。

ワンタイムパスワード認証をテストする

1. OTP をテストするには、リストからデバイスをクリックし、[テスト] をクリックします。
2. デバイスで受信した OTP を入力し、[**Go**] をクリックします。

   OTP 検証に成功したことを示すメッセージが表示されます。
3. ページの右上隅にあるリストを使用してログアウトします。

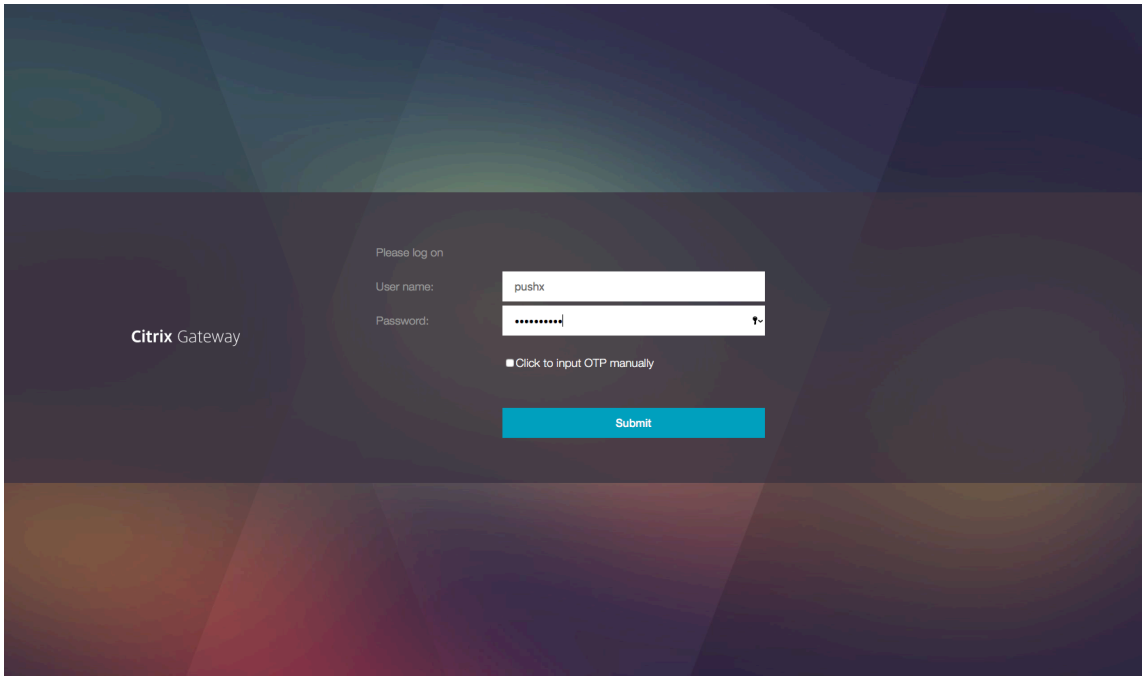
注: OTP 管理ポータルは、認証のテスト、登録済みデバイスの削除、または追加デバイスの登録にいつでも使用できます。

NetScaler Gateway へのログイン

デバイスを NetScaler Gateway に登録すると、ユーザーはプッシュ通知機能を使用して認証を行うことができます。

1. NetScaler Gateway の認証ページに移動します（例:）https://gateway.company.com

   ログインスキーマの設定に応じて、LDAP 認証情報のみを入力するように求められます。

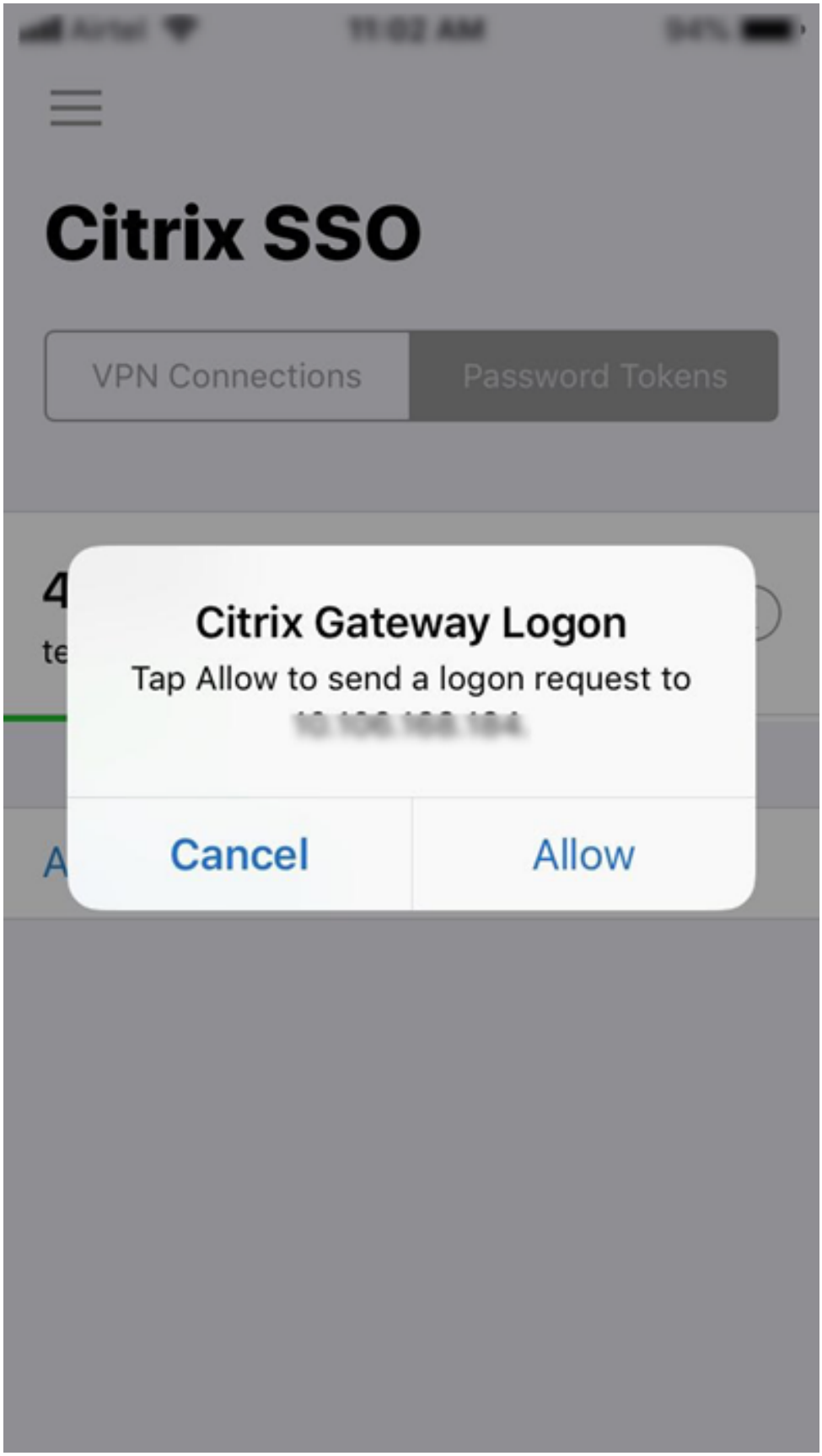


2. LDAP ユーザ名とパスワードを入力し、[送信（**Submit**）] を選択します。

   登録済みのデバイスに通知が送信されます。

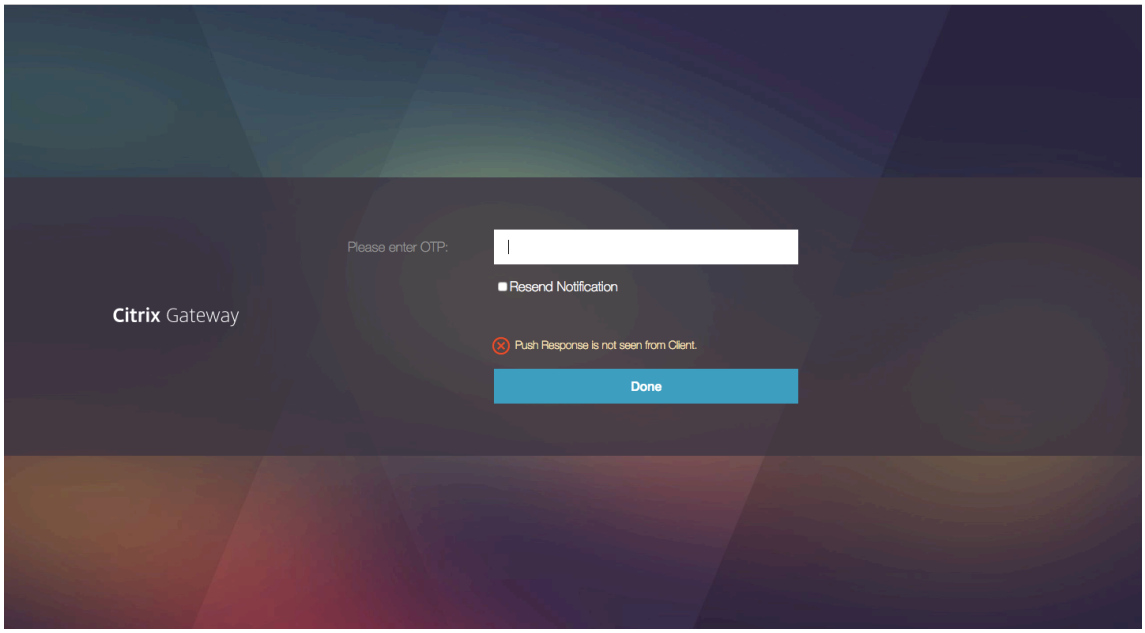
   注意**:** OTP を手動で入力する場合は、「クリックして **OTP** を手動で入力する」を選択し、「**TOTP** 」フィールドに OTP を入力する必要があります。

3. 登録済みのデバイスで Citrix SSO アプリを開き、［許可］をタップします。



注:

- iOS デバイスでは、認証の追加要素として Touch-ID/Face-ID/パスコードの入力を求められます。
- 認証サーバは、設定されたタイムアウト期間が終了するまで、プッシュサーバ通知応答を待ちます。タイムアウト後、NetScaler Gateway はログインページを表示します。その後、ユーザは OTP を手動で入力するか、または [通知を再送する（**Resend Notification**）] をクリックして、登録されたデバイスで通知を再度受信できます。選択したオプションに基づいて、ゲートウェイは入力した OTP を検証するか、登録済みデバイスで通知を再送信します。



- ログイン失敗に関する通知は登録済みデバイスに送信されません。

障害状態

- デバイスの登録は、次の場合に失敗することがあります。
  - サーバー証明書は、エンドユーザーデバイスによって信頼されていない可能性があります。
  - OTP の登録に使用した NetScaler Gateway にクライアントからアクセスできません。
- 通知は、次の場合に失敗することがあります。
  - ユーザーデバイスがインターネットに接続されていない
  - ユーザーデバイス上の通知はブロックされます
  - ユーザーがデバイス上の通知を承認しない

このような場合、認証サーバは、設定されたタイムアウト期間が経過するまで待機します。タイムアウト後、NetScaler Gateway はログインページを表示し、OTP を手動で入力するか、登録したデバイスに通知を再送信するかを選択できます。選択したオプションに基づいて、さらに検証が行われます。

障害ログ

OTP プッシュサービスに到達できない場合の予想されるログを次に示します。

- ユーザーデバイスがインターネットに接続されていないときのプッシュ通知の失敗-プッシュ: プッシュサービスの“client name”へのプッシュリクエストの準備に失敗しました。
- デバイス登録失敗ログ -プッシュ:「client name」のクラウドにプッシュリクエストを送信するためのデバイスが登録されていません。
- ユーザーがプッシュを受け入れない場合-プッシュ:「user name」でクライアントからの応答は表示されません。再試行オプションをチェックします。

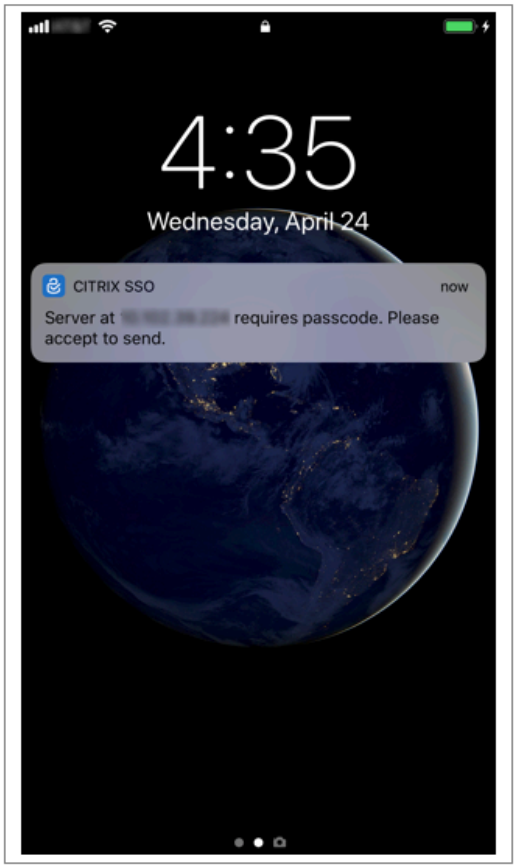
iOS での Citrix SSO アプリケーションの動作—注意すべきポイント

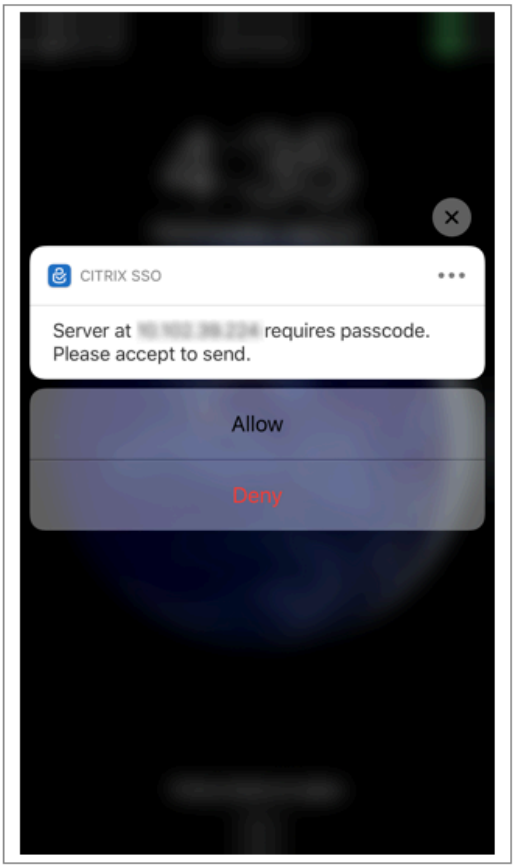
通知ショートカット

Citrix SSO iOS アプリには、ユーザーエクスペリエンスを向上させるための実用的な通知のサポートが含まれています。iOS デバイスで通知を受信し、デバイスがロックされているか、Citrix SSO アプリがフォアグラウンドになっていない場合、ユーザーは通知に組み込まれているショートカットを使用して、ログイン要求を承認または拒否できます。

通知ショートカットにアクセスするには、デバイスのハードウェアに応じて、ユーザーは通知を強制的にタッチ（3D タッチ）するか、長押しする必要があります。「ショートカットを許可」アクションを選択すると、NetScaler にログイン要求が送信されます。認証、承認、および監査仮想サーバーで認証ポリシーがどのように構成されているかに応じて、

- ログイン要求は、アプリをフォアグラウンドで起動したり、デバイスのロックを解除したりすることなく、バックグラウンドで送信される可能性があります。
- アプリは追加の要素として Touch-ID/Face-ID/Passcode の入力を求める場合があります。その場合、アプリはフォアグラウンドで起動されます。



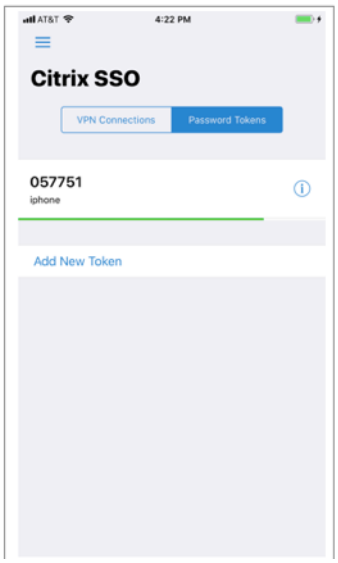


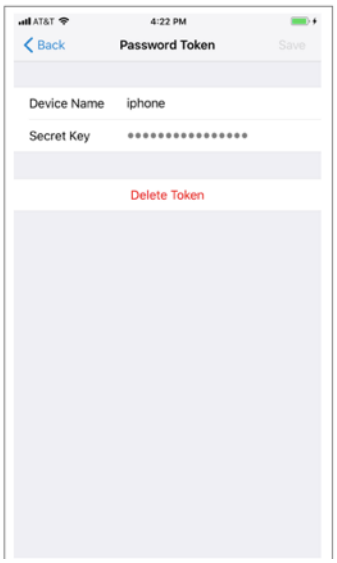
Citrix SSO からパスワードトークンを削除する

1. Citrix SSO アプリでプッシュ用に登録されたパスワードトークンを削除するには、ユーザーは次の手順を実行する必要があります。
2. ゲートウェイの iOS/Android デバイスを登録解除 (削除) します。デバイスから登録を削除するための QR コードが表示されます。
3. Citrix SSO アプリを開き、削除するパスワードトークンの情報ボタンをタップします。
4. [トークンを削除] をタップし、QR コードをスキャンします。

注記:

- QR コードが有効な場合、トークンは Citrix SSO アプリから正常に削除されます。
- デバイスがすでにゲートウェイから削除されている場合、ユーザーは QR コードをスキャンしなくても、[強制削除] をタップしてパスワードトークンを削除できます。強制的に削除すると、デバイスが NetScaler Gateway から削除されていない場合、デバイスが引き続き通知を受信する可能性があります。





電子メール **OTP** 認証

August 15, 2023

電子メール OTP は NetScaler 12.1 ビルド 51.x で導入されました。電子メール OTP 方式では、登録された電子メールアドレスに送信されるワンタイムパスワード (OTP) を使用して認証できます。いずれかのサービスで認証を試みると、サーバーは登録されているユーザーのメールアドレスに OTP を送信します。

電子メール OTP 機能を使用するには、まず代替電子メール ID を登録する必要があります。アカウントがロックアウトされた場合や AD パスワードを忘れた場合は、プライマリ電子メール ID にアクセスできないため、OTP をそのメール ID に送信できるように、代わりの電子メール ID の登録が必要です。

AD 属性の一部として代替電子メール ID をすでに指定している場合は、電子メール ID を登録せずに電子メール OTP 検証を使用できます。電子メールアドレスセクションで代替電子メール ID を指定する代わりに、電子メールアクションで同じ属性を参照できます。

前提条件

電子メール OTP 機能を設定する前に、次の前提条件を確認してください。

- NetScaler 機能リリース 12.1 ビルド 51.28 以降
- 電子メール OTP 機能は nFactor 認証フローでのみ使用できます
  - 詳細については、「https://support.citrix.com/pages/citrix-adc-authentication-how#nfactor」を参照してください。

- AAA-TM、NetScaler Gateway（ブラウザ、ネイティブプラグイン、Receiver）でサポートされています。

アクティブディレクトリ設定

- サポートされているバージョンは 2016/2012 および 2008 年の Active Directory ドメイン機能レベルです
- NetScaler LDAPBind ユーザー名には、ユーザーの AD パスへの書き込みアクセス権が必要です

メールサーバ

- 電子メール OTP ソリューションを機能させるには、SMTP サーバーでログインベースの認証が有効になっていることを確認します。NetScaler では、電子メール OTP が機能するための認証は AUTH LOGIN ベースの認証のみをサポートしています。
- AUTH LOGIN ベースの認証が有効になっていることを確認するには、SMTP サーバーで次のコマンドを入力します。ログインベースの認証が有効になっている場合は、出力に AUTH LOGIN というテキストが太字で表示されます。

```
root@ns# telnet <IP address of the SMTP server><Port number of the server>
ehlo
root@ns# telnet 10.106.3.
Trying 10.106.
Connected to 10.106.
Escape character is '^]'.
220 E2K13.NSGSanity.com Microsoft ESMTP MAIL Service ready at Fri, 22 Nov
2019 16:24:17 +0530
ehlo
250-E2K13.NSGSanity.com Hello [10.221.
250-SIZE 37748736
250-PIPELINING
250-DSN
250-ENHANCEDSTATUSCODES
250-STARTTLS
250-X-ANONYMOUSTLS
250-AUTH LOGIN
250-X-EXPS GSSAPI NTLM
250-8BITMIME
250-BINARYMIME
250-CHUNKING
250 XRDST
For information on how to enable login based authentication, see
https://support.microfocus.com/kb/doc.php?id=7020367
```

制限事項

- この機能は、認証バックエンドが LDAP の場合にのみサポートされます。
- すでに登録されている代替電子メール ID は表示されません。
- KBA 登録ページの代替電子メール ID のみ更新できません。

- 電子メール OTP 認証を認証フローの最初の要素にすることはできません。これは、堅牢な認証を実現するための仕様です。
- 代替電子メール ID と KBA の両方が同じ認証アクションを使用して設定されている場合、属性は両方で同じである必要があります。
- ネイティブプラグインと Receiver では、登録はブラウザ経由でのみサポートされます。

Active Directory 構成

- 電子メール OTP は、Active Directory 属性をユーザーデータストレージとして使用します。
- 代替メール ID を登録すると、そのメール ID が NetScaler アプライアンスに送信され、アプライアンスはそれを AD ユーザーオブジェクトに設定された KB 属性に保存します。
- 代替電子メール ID は暗号化され、設定された AD 属性に保存されます。

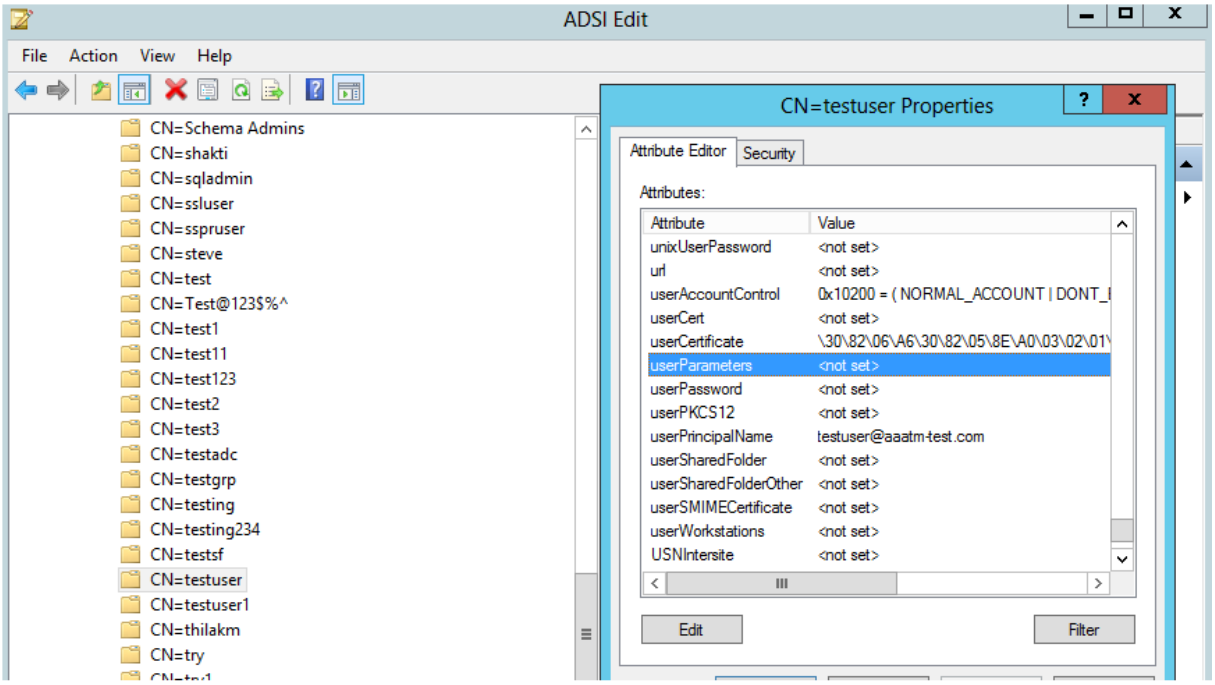
AD 属性を設定する場合は、次の点を考慮してください。

- サポートされる属性名の長さは 128 文字以上でなければなりません。
- 属性タイプは‘directoryString’でなければなりません。
- ネイティブ OTP と電子メール OTP 登録データに同じ AD 属性を使用できます。
- LDAP 管理者には、選択した AD 属性に対する書き込みアクセス権が必要です。

既存の属性の使用

この例で使用されている属性は`Userparameters`です。これは AD ユーザ内の既存の属性なので、AD 自体に変更を加える必要はありません。ただし、その属性が使用されていないことを確認する必要があります。

属性が使用されないようにするには、[**ADSI**] に移動して [user] を選択し、そのユーザーを右クリックし、属性リストまでスクロールダウンします。**UserParameters** の属性値が設定されていないことを確認する必要があります。これは、その属性が現在使用されていないことを示します。



電子メール **OTP** の設定

電子メール OTP ソリューションは、次の 2 つの部分で構成されています。

- メール登録
- メール検証

メール **ID** 登録

KBA 登録スキーマが正常に作成されたら、CLI を使用して次の設定を行います。

1. ポータルテーマと証明書を VPN Global にバインドします。

```
bind authentication vserver authvs -portaltheme RfWebUI
bind vpn global -userDataEncryptionKey c1
<!--NeedCopy-->
```

> 注:
>
> AD 属性に格納されているユーザーデータ (KB Q&A および登録済みの代替メール ID) を暗号化するには、証明書のバインドを先行する必要があります。

2. LDAP 認証ポリシーを作成します。


```
add authentication ldapAction ldap -serverIP 10.102.2.2 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samAccountName -secType SSL
add authentication Policy ldap -rule true -action ldap
<!--NeedCopy-->
```

3. 電子メール登録用の LDAP 認証ポリシーを作成します。

```
add authentication ldapAction ldap_email_registration -serverIP
    10.102.2.2 -serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -
    ldapBindDn administrator@aaatm-test.com -ldapBindDnPassword
    freebsd -ldapLoginName samAccountName -secType SSL -KBAttribute
     userParameters -alternateEmailAttr userParameters
add authentication Policy ldap_email_registration -rule true -
    action ldap_email_registration
<!--NeedCopy-->
```

4. E メール登録ログインスキーマとポリシーラベルを作成します。

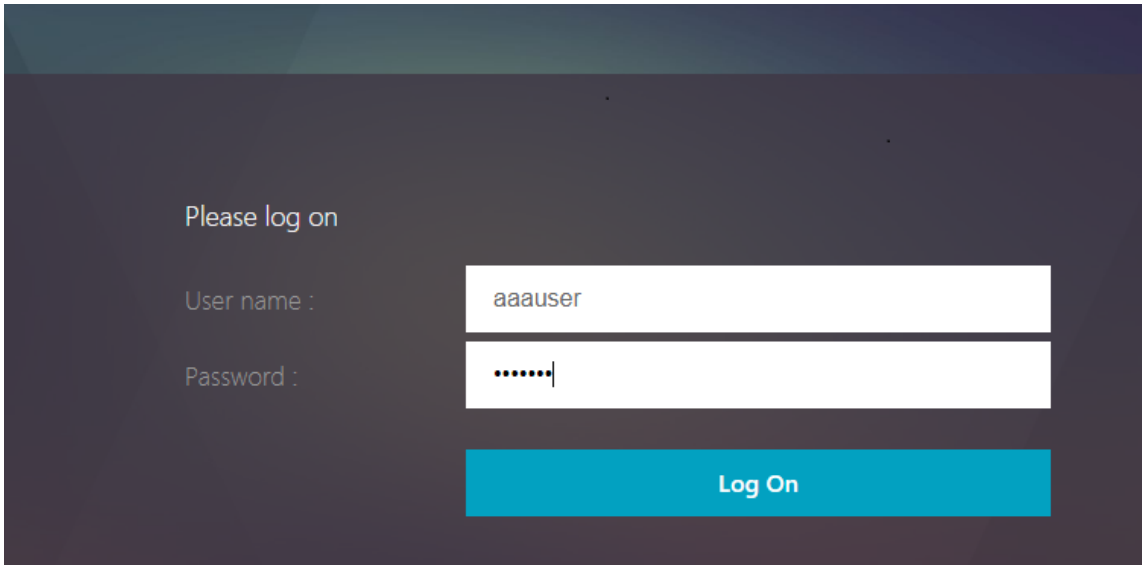
```
add authentication loginSchema onlyEmailRegistration -
    authenticationSchema /nsconfig/loginschema/LoginSchema/
    AltEmailRegister.xml
add authentication policylabel email_Registration_factor -
    loginSchema onlyEmailRegistration
bind authentication policylabel email_Registration_factor -
    policyName ldap_email_registration -priority 1 -
    gotoPriorityExpression NEXT
<!--NeedCopy-->
```

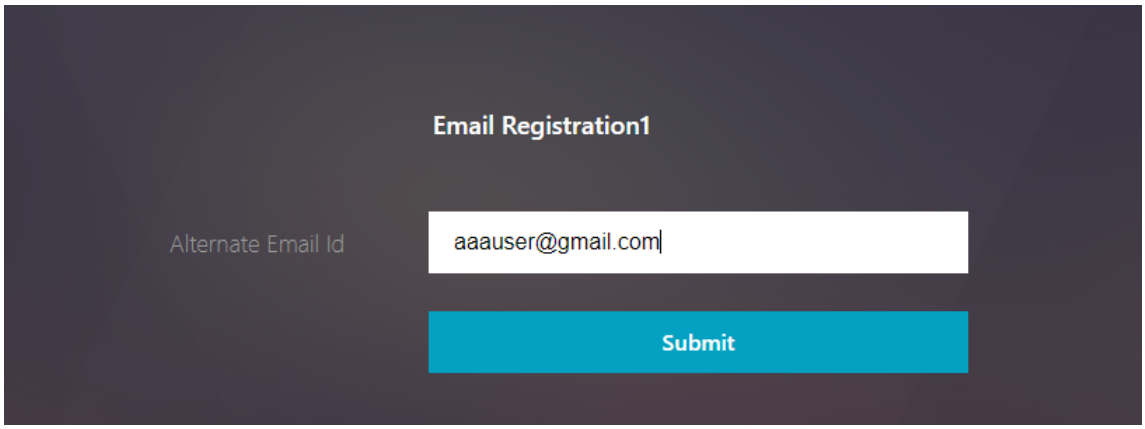
5. 認証ポリシーを認証仮想サーバーにバインドします。

```
bind authentication vserver authvs  - policy ldap -priority 1 -
    nextFactor email_Registration_factor -gotoPriorityExpression
    NEXT
<!--NeedCopy-->
```

6. 前のセクションで説明した手順をすべて設定したら、次の GUI 画面が表示されます。URL（たとえば、https://lb1.server.com/）を使用してアクセスすると、LDAP ログオン資格情報だけを必要とする初期ログインページが表示され、その後に代替の電子メール登録ページが表示されます。

   注**:** ドメインhttps://lb1.server.com/は、ゲートウェイまたは認証仮想サーバのいずれかに属することができます。





> 注:
>
> - KBA 登録と電子メール ID 登録の両方に同じ認証スキーマを使用できます。
> - KBA 登録の設定時に、[電子メール登録] セクションの [代替電子メールの登録] を選択して、代替電子メール ID を登録できます。

メール検証

E メール検証を行うには、次の手順を実行します。

1. ポータルテーマと証明書を VPN Global にバインドする

```
bind authentication vserver authvs -portaltheme RfWebUI
bind vpn global -userDataEncryptionKey c1
<!--NeedCopy-->
```


> 注:
>
> AD 属性に保存されているユーザーデータ (KB Q&A および登録済みの代替電子メール ID) を復号化するには、証明書のバインドを先行する必要があります。

2. LDAP 認証ポリシーを作成します。電子メール OTP 検証にはユーザーの電子メール ID または代替電子メール ID が必要なため、LDAP は電子メール検証係数の前の要素である必要があります。

```
add authentication ldapAction ldap1 -serverIP 10.102.2.2 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -   ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samAccountName -secType SSL -KBAttribute
    userParameters -alternateEmailAttr userParameters
add authentication Policy ldap1 -rule true -action ldap1
<!--NeedCopy-->
```

3. E メール認証ポリシーを作成します。

```
add authentication emailAction email -userName sqladmin@aaa.com -
    password freebsd-encrypted -encryptmethod ENCMTHD_3 -serverURL
    "smtps://10.2.3.3:25" -content "OTP is $code" -
    defaultAuthenticationGroup emailgrp -emailAddress "aaa.user.
    attribute("alternate_mail")"
add authentication Policy email -rule true  - action email
<!--NeedCopy-->
```

前述のコマンドでは、メールアドレスは KBA 登録時に指定された代替電子メール ID ユーザーです。

4. 電子メール OTP 検証ポリシーラベルを作成します。

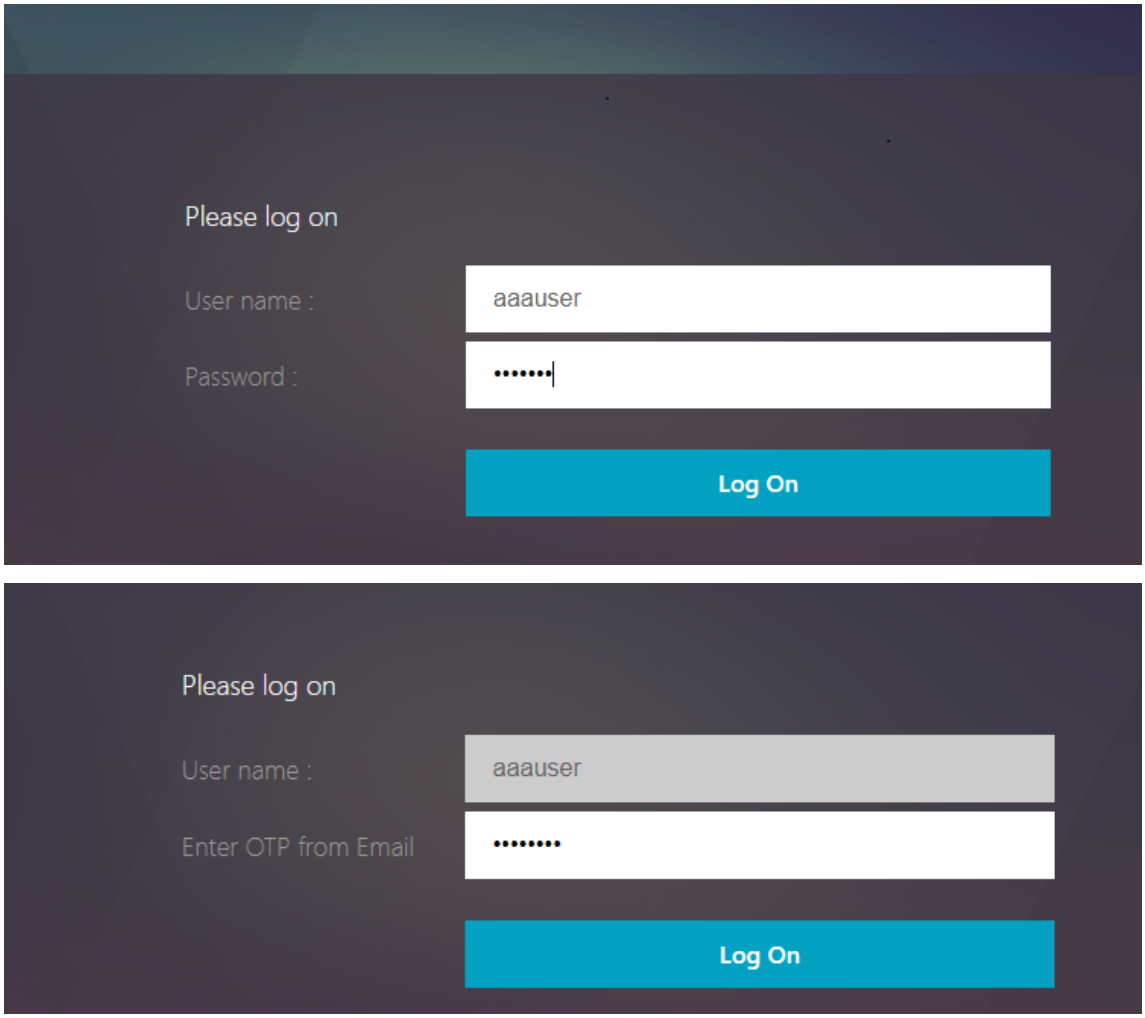
```
add authentication policylabel email_Validation_factor
bind authentication policylabel email_Validation_factor -
    policyName email -priority 1 -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

5. 認証ポリシーを認証仮想サーバーにバインドします。

```
bind authentication vserver authvs  -  policy ldap1 -priority 1 -
    nextFactor email_Validation_factor -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

6. 前のセクションで説明した手順をすべて設定したら、電子メール OTP 検証用の次の GUI 画面が表示されます。URL（たとえば、https://lb1.server.com/）を使用してにアクセスすると、LDAP ログオン資格情報だけを必要とする初期ログインページが表示され、その後に [電子メール OTP 検証] ページが表示されます。

> 注:
>
> LDAP ポリシーでは、AD 属性からユーザの電子メール ID を照会できるように`alternateEmailAttr`を設定することが重要です。



トラブルシューティング

ログを分析する前に、以下のようにログレベルを debug に設定することをお勧めします。

```
set syslogparams -loglevel DEBUG
<!--NeedCopy-->
```

登録-成功シナリオ

次のエントリは、ユーザ登録が成功したことを示します。

```
"ns_aaa_insert_hash_keyValue_entry key:kba_registered value:1"
Nov 14 23:35:51 <local0.debug> 10.102.229.76 11/14/2018:18:05:51 GMT
    0-PPE-1 : default SSLVPN Message 1588 0 :  "
    ns_aaa_insert_hash_keyValue_entry key:alternate_mail value:
    eyJ2ZXJzaW9uIjoiMSIsICJraWQiOiIxbk1oWjN0T2NjLVVvZUx6NDRwZFhxdS01dTA9IiwgImtleS
    ==.oKmvOalaOJ3a9z7BcGCSegNPMw=="
```



```
3
4 <!--NeedCopy-->
```

登録-失敗シナリオ

ユーザーのログインページに、「リクエストを完了できません」というエラーメッセージが表示されます。これは、ユーザデータを暗号化するために VPN グローバルにバインドされる証明書キーがないことを示します。

```
1 Jul 31 08:51:46 <local0.info> 10.102.229.79 07/31/2020:03:21:4 6 GMT
     0-PPE-1 : default SSLVPN Message 696 0 :  "Encrypt UserData: No
     Encryption cert is bound to vpn global"
2 Jul 31 08:51:46 <local0.info> 10.102.229.79 07/31/2020:03:21:46 GMT  0-
     PPE-1 : default SSLVPN Message 697 0 :  "KBA Register: Alternate
     email id Encrypted blob length is ZERO aaauser"
3 <!--NeedCopy-->
```

メール検証—成功シナリオ

次のエントリは、電子メール OTP 検証が成功したことを示します。

```
1 "NFactor: Successfully completed email auth, nextfactor is pwd_reset"
2 <!--NeedCopy-->
```

E メール検証—失敗シナリオ

ユーザーのログインページに、「リクエストを完了できません」というエラーメッセージが表示されます。これは、電子メールサーバーでログインベースの認証が有効になっていないため、同じ認証を有効にする必要があることを示します。

```
1 " /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp
     [100]: void ThreadWorker_SendMailJob(SMTPJob*) 0-215: [POCO][JobID:
     8]SMTP Configuration is Secure..
2 /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp[108]:
     void ThreadWorker_SendMailJob(SMTPJob*) 0-215: [POCO][JobID: 8]
     First login succeeded
3 Wed Mar  4 17:16:28 2020
4 /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/naaad.c[697]: main
     0-0: timer 2 firing...
5 /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp[127]:
     void ThreadWorker_SendMailJob(SMTPJob*) 0-0: [POCO-ERROR][JobID: 8]
     Poco SMTP Mail Dispatch  Failed. SMTP TYPE:1, SMTPException:
     Exception occurs. SMTP Exception: The mail service does not support
     LOGIN authentication: 250-smtprelay.citrix.com Hello [10.9.154.239]
6 250-SIZE 62914560
7 250-PIPELINING
8 250-DSN
```



```
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-BINARYMIME
250 CHUNKING
<!--NeedCopy-->
```

nFactor 認証の設定を再キャプチャする

August 15, 2023

NetScaler Gateway は、`captchaAction` 再キャプチャ構成を簡素化する新しいファーストクラスアクションをサポートしています。再キャプチャはファーストクラスのアクションなので、それ自体が要因になる可能性があります。再キャプチャは nFactor フローのどこにでも注入できます。

以前は、rfWebUi に変更を加えたカスタム WebAuth ポリシーも作成する必要がありました。`captchaAction`の導入により、JavaScript を変更する必要がなくなりました。

> 重要:
>
> スキーマ内のユーザー名またはパスワードフィールドとともに Re-CAPTCHA が使用されている場合、再キャプチャが満たされるまで [送信] ボタンは無効になります。

設定を再キャプチャ

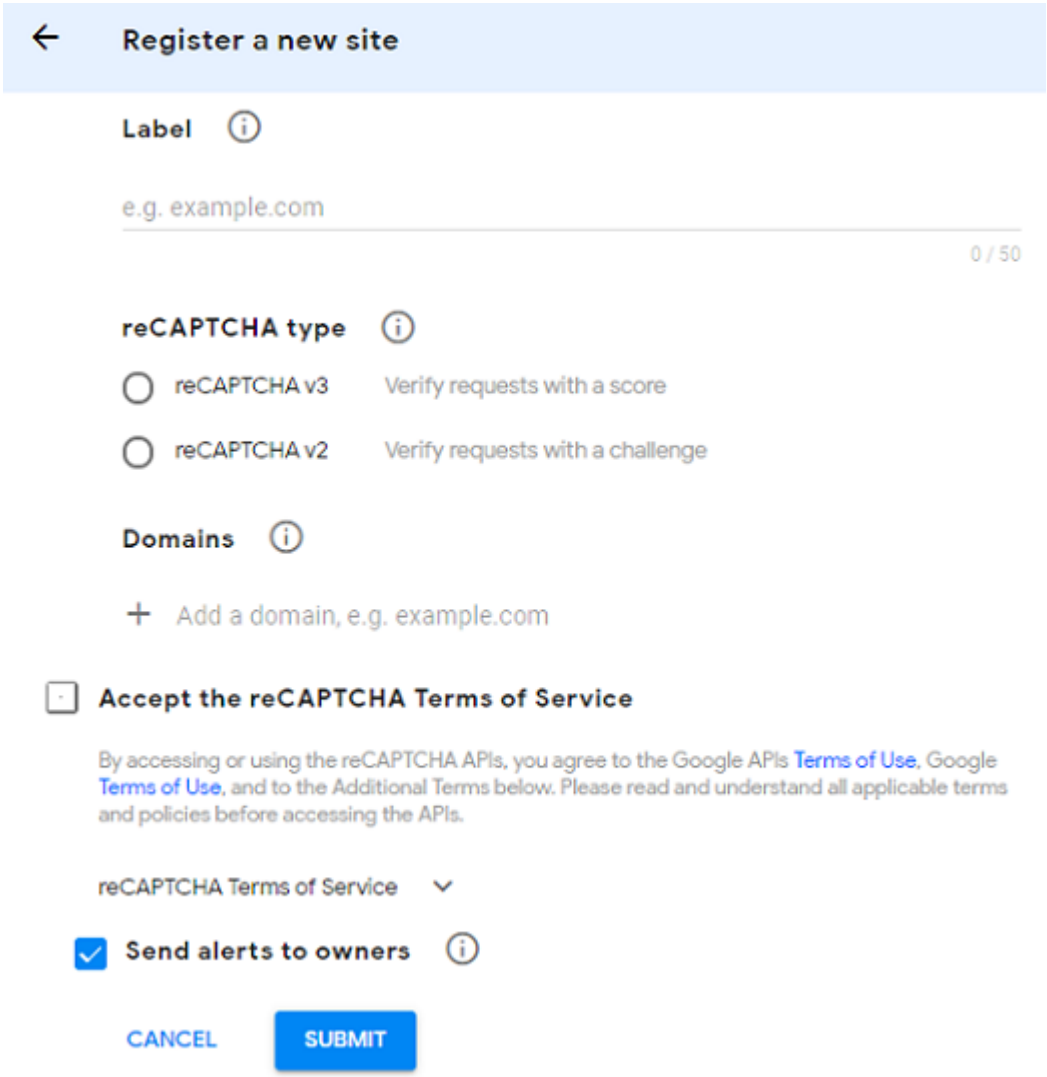
再キャプチャの設定には 2 つの部分が含まれます。

1. 再キャプチャを登録するための Google での設定。
2. ログインフローの一部として再キャプチャを使用するように NetScaler アプライアンスを構成します。

Google の設定を再キャプチャ

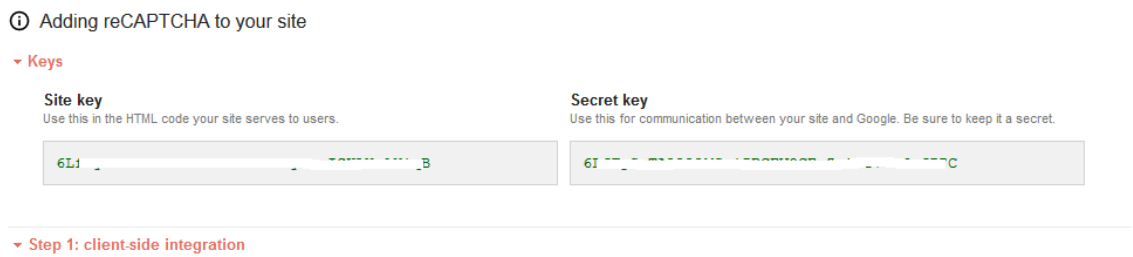
https://www.google.com/recaptcha/admin#llistで再キャプチャするドメインを登録します。

1. このページに移動すると、次の画面が表示されます。



> 注
>
> reCAPTCHA v2 のみを使用してください。見えない再キャプチャはまだプレビュー中です。

2. ドメインが登録されると、「SiteKey」と「SecretKey」が表示されます。



> 注
>
> セキュリティ上の理由から、「siteKey」と「SecretKey」はグレー表示されています。「SecretKey」は

> 安全に保管する必要があります。

NetScaler アプライアンスでの構成の再キャプチャ

NetScaler アプライアンスの Re-CAPTCHA 構成は、次の 3 つの部分に分けることができます。

- 再キャプチャ画面を表示する
- 再キャプチャレスポンスを Google サーバに投稿する
- LDAP 構成はユーザーログオンの 2 番目の要素です (オプション)

再キャプチャ画面を表示する　ログインフォームのカスタマイズは、SingleAuthCaptcha.xml ログインスキーマを介して行われます。このカスタマイズは認証仮想サーバーで指定され、ログインフォームをレンダリングするために UI に送信されます。組み込みのログインスキーマである SingleAuthCaptcha.xml は、/nsconfig/loginSchema/LoginSchema NetScaler アプライアンスのディレクトリにあります。

> 重要
>
> - SingleAuthCaptcha.xml ログインスキーマは、LDAP が第 1 要素として構成されている場合に使用できます。
> - ユースケースと異なるスキーマに基づいて、既存のスキーマを変更できます。たとえば、再キャプチャ要素（ユーザー名またはパスワードなし）または再キャプチャによる二重認証のみが必要な場合などです。
> - カスタム変更が行われた場合、またはファイルの名前が変更された場合は、すべての LoginSchemas を /nsconfig/loginschema/LoginSchemaディレクトリから親ディレクトリ /nsconfig/loginschemaにコピーすることをお勧めします。

CLI を使用して **Re-CAPTCHA** の表示を設定するには

```
add authentication loginSchema singleauthcaptcha -authenticationSchema
    /nsconfig/loginschema/SingleAuthCaptcha.xml

add authentication loginSchemaPolicy singleauthcaptcha -rule true -
    action singleauthcaptcha

add authentication vserver auth SSL <IP> <Port>

add ssl certkey vserver-cert -cert <path-to-cert-file> -key <path-to-
    key-file>

bind ssl vserver auth -certkey vserver-cert

bind authentication vserver auth -policy singleauthcaptcha -priority 5
    -gotoPriorityExpression END
<!--NeedCopy-->
```


再キャプチャレスポンスを **Google** サーバに投稿する　ユーザーに表示する必要のある再キャプチャを設定したら、管理者はその構成を Google サーバーに追加して、ブラウザからの再キャプチャ応答を確認します。

ブラウザからの **Re-CAPTCHA** 応答を確認するには

```
add authentication captchaAction myrecaptcha -sitekey <sitekey-copied-from-google> -secretkey <secretkey-from-google>

add authentication policy myrecaptcha -rule true -action myrecaptcha

bind authentication vserver auth -policy myrecaptcha -priority 1
<!--NeedCopy-->
```

AD 認証が必要かどうかを設定するには、次のコマンドが必要です。それ以外の場合は、この手順は無視してかまいません。

```
add authentication ldapAction ldap-new -serverIP x.x.x.x -serverPort 636 -ldapBase "cn=users,dc=aaatm,dc=com" -ldapBindDn adminuser@aaatm.com -ldapBindDnPassword <password> -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName memberof -subAttributeName CN -secType SSL -passwdChange ENABLED -defaultAuthenticationGroup ldapGroup

add authenticationpolicy ldap-new -rule true -action ldap-new
<!--NeedCopy-->
```

LDAP 構成はユーザーログオンの **2** 番目の要素です **(**オプション**)**　LDAP 認証は再キャプチャ後に行われ、2 番目の要素に追加します。

```
add authentication policylabel second-factor

bind authentication policylabel second-factor -policy ldap-new -priority 10

bind authentication vserver auth -policy myrecaptcha -priority 1 -nextFactor second-factor
<!--NeedCopy-->
```

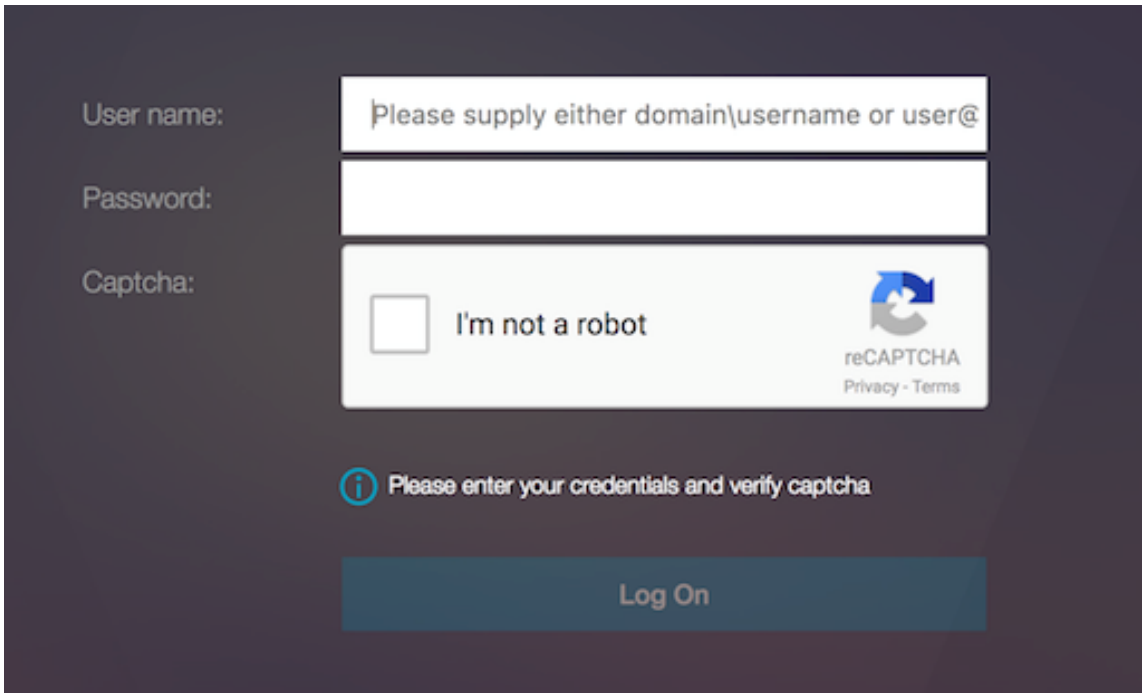
管理者は、アクセスに負荷分散仮想サーバーと NetScaler Gateway アプライアンスのどちらを使用するかに応じて、適切な仮想サーバーを追加する必要があります。負荷分散仮想サーバーが必要な場合は、管理者が次のコマンドを構成する必要があります。

```
add lb vserver lbtest HTTP <IP> <Port> -authentication ON -authenticationHost nssp.aaatm.com
<!--NeedCopy-->
```

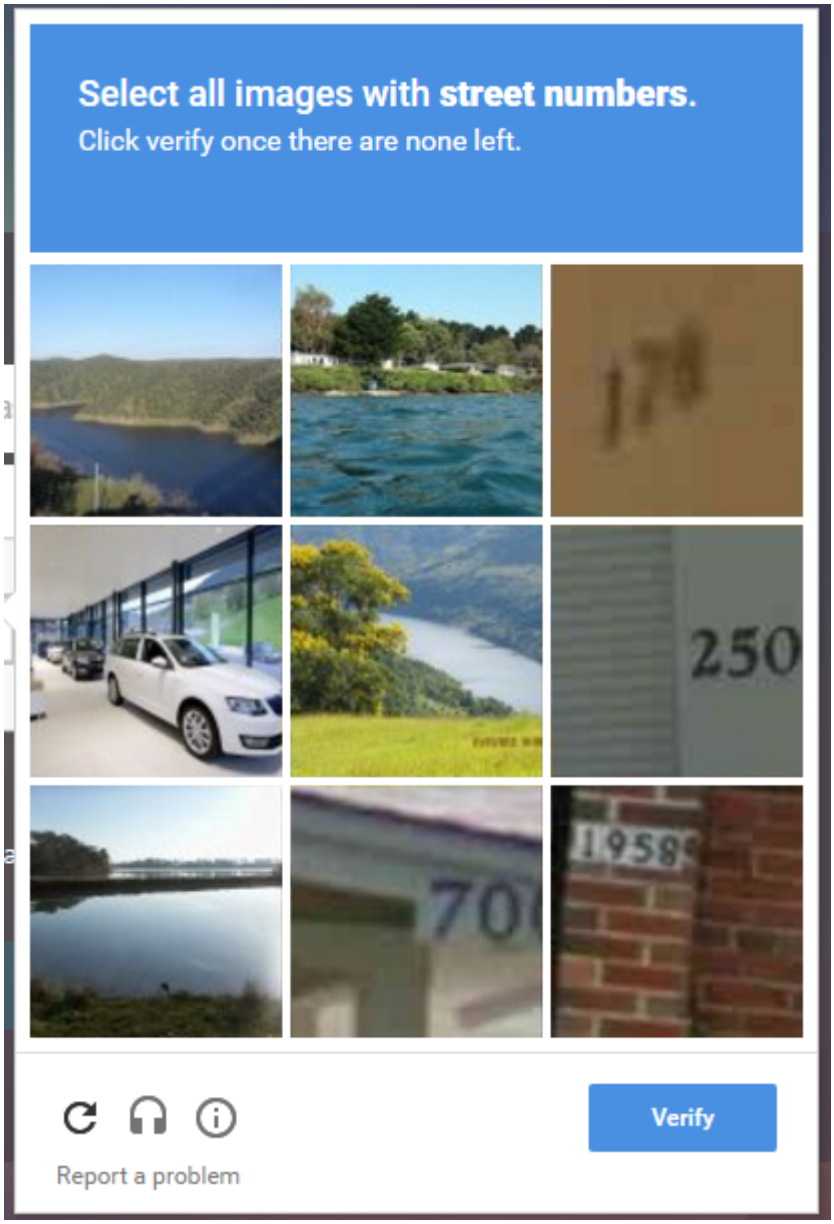
nssp.aaatm.com —認証仮想サーバに解決します。

再キャプチャのユーザー検証　前のセクションで説明したすべての手順を構成したら、次の UI が表示されるはずです。

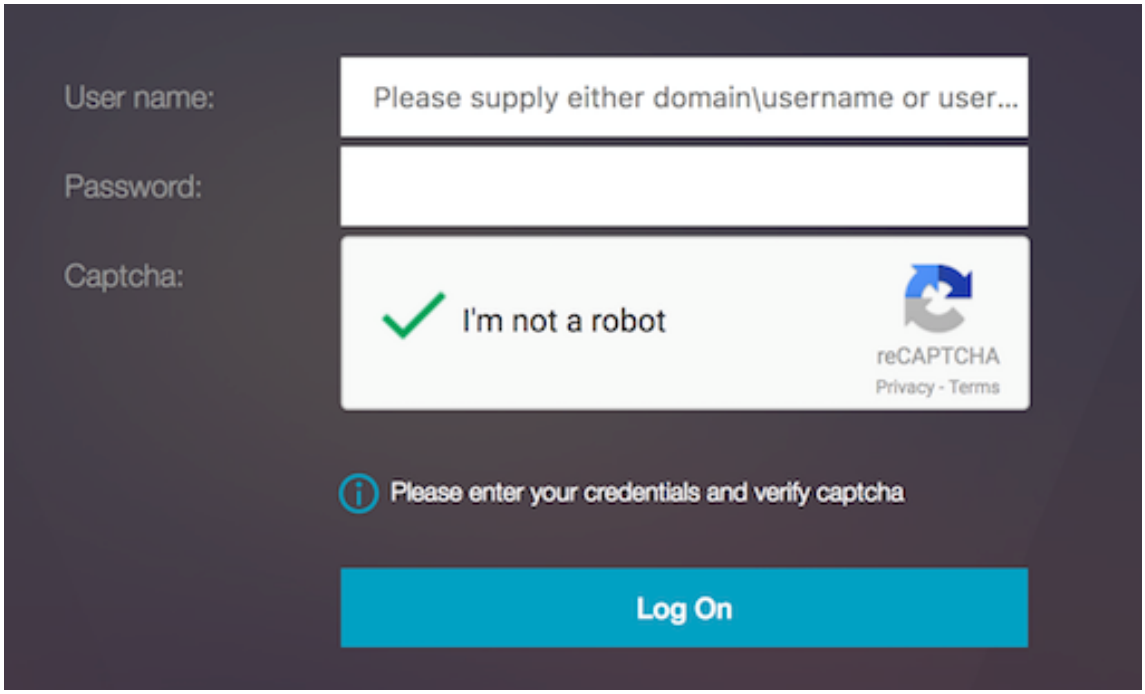
1. 認証仮想サーバーがログインページをロードすると、ログオン画面が表示されます。再キャプチャが完了するまで、ログオンは無効になります。



2. [ロボットではありません] オプションを選択します。Re-Captcha ウィジェットが表示されます。



3. 完了ページが表示される前に、一連の再キャプチャ画像をナビゲートします。
4. AD 資格情報を入力し、[ロボットではありません] チェックボックスをオンにして、[ログオン] をクリックします。認証が成功すると、目的のリソースにリダイレクトされます。



注:

- 再キャプチャが AD 認証で使用されている場合、再キャプチャが完了するまで、資格情報の [送信] ボタンは無効になります。
- 再キャプチャは独自の要因で発生します。したがって、AD のような後続の検証は、re-captcha の`nextfactor`で行う必要があります。

一般的に使用されるプロトコルの認証、承認、および監査の構成

August 15, 2023

NetScaler アプライアンスを認証、承認、および監査用に構成するには、NetScaler アプライアンスとクライアントのブラウザーで特定の設定を行う必要があります。設定は、認証、認可、および監査に使用されるプロトコルによって異なります。

NetScaler アプライアンスを Kerberos 認証用に構成する方法の詳細については、「Kerberos/NTLM による認証、承認、および監査の処理」を参照してください。

Kerberos/NTLM による認証、承認、監査の処理

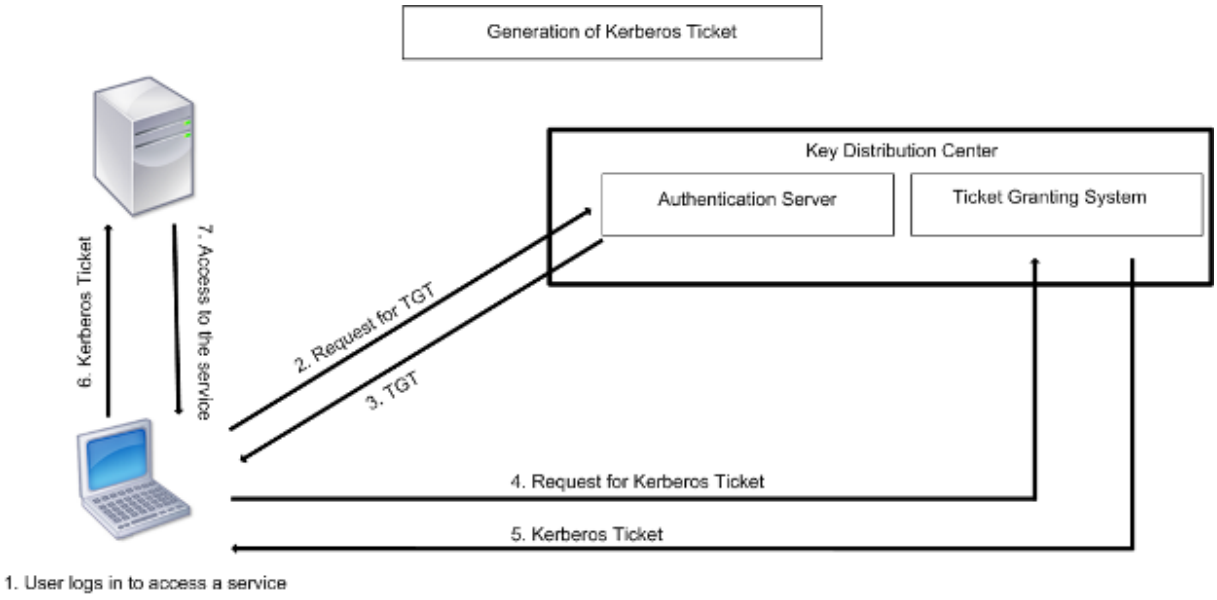
August 15, 2023

コンピュータネットワーク認証プロトコルである Kerberos は、インターネットを介した安全な通信を提供します。主にクライアント/サーバーアプリケーション向けに設計されており、クライアントとサーバーが相互の信頼性を確保できる相互認証を提供します。Kerberos は、キー配布センター (KDC) と呼ばれる信頼できるサードパーティを使用しています。KDC は、ユーザーを認証する認証サーバー (AS) とチケット交付サーバー (TGS) で構成されています。

ネットワーク上の各エンティティ (クライアントまたはサーバー) には、そのエンティティと KDC だけが知っている秘密鍵があります。このキーを知っているということは、エンティティが本物であることを意味します。ネットワーク上の 2 つのエンティティ間の通信では、KDC は Kerberos チケットまたはサービスチケットと呼ばれるセッションキーを生成します。クライアントは AS に特定のサーバーの認証情報を要求します。その後、クライアントはチケット交付チケット (TGT) と呼ばれるチケットを受け取ります。次に、クライアントは AS から受け取った TGT を使用してその身元を証明して TGS に連絡し、サービスを依頼します。クライアントがサービスを受ける資格がある場合、TGS はクライアントに Kerberos チケットを発行します。次に、クライアントは、Kerberos チケットを使用してサービスをホストしているサーバー (サービスサーバーと呼ばれる) に接続し、サービスの受信が許可されていることを証明します。Kerberos チケットの有効期限は設定可能です。クライアントは AS で自身を認証するのは 1 回だけです。物理サーバーに複数回接続する場合、AS チケットを再利用します。

次の図は、Kerberos プロトコルの基本的な機能を示しています。

図 1：ケルベロスの機能



Kerberos 認証には次の利点があります。

- より高速な認証。物理サーバーがクライアントから Kerberos チケットを取得すると、サーバーにはクライアントを直接認証するのに十分な情報があります。クライアント認証のためにドメインコントローラーに連絡する必要がないため、認証プロセスが高速になります。
- 相互認証。KDC がクライアントに Kerberos チケットを発行し、クライアントがそのチケットを使用してサービスにアクセスする場合、認証されたサーバーのみが Kerberos チケットを復号化できます。NetScaler アプライアンス上の仮想サーバーが Kerberos チケットを復号化できる場合は、仮想サーバーとクライアント

の両方が認証されていると判断できます。したがって、サーバーの認証はクライアントの認証とともに行われます。

- Windows と Kerberos をサポートする他のオペレーティングシステム間のシングルサインオン。

Kerberos 認証には次のような欠点があります。

- Kerberos には厳しい時間要件があります。認証が失敗しないように、関係するホストの時計を Kerberos サーバーの時計と同期させる必要があります。この欠点は、ネットワークタイムプロトコルデーモンを使用してホストクロックの同期を維持することで軽減できます。Kerberos チケットには利用可能期間があり、設定することができます。
- Kerberos では、中央サーバーが継続的に利用可能である必要があります。Kerberos サーバーがダウンすると、誰もログオンできなくなります。複数の Kerberos サーバーとフォールバック認証メカニズムを使用することで、このリスクを軽減できます。
- 認証はすべて一元化された KDC によって制御されるため、ローカルワークステーションのユーザーのパスワードが盗まれるなど、このインフラストラクチャが侵害されると、攻撃者は任意のユーザーになりすますことができます。信頼できるデスクトップマシンまたはラップトップのみを使用するか、ハードウェアトークンによる事前認証を強制することで、このリスクをある程度軽減できます。

Kerberos 認証を使用するには、NetScaler アプライアンスと各クライアントで認証を構成する必要があります。

認証、承認、監査における **Kerberos** 認証の最適化

NetScaler アプライアンスは、Kerberos 認証中にシステムパフォーマンスを最適化および改善するようになりました。認証、承認、および監査デーモンは、同じユーザーに対する未処理の Kerberos 要求を記憶し、キー配布センター (KDC) への負荷を回避します。これにより、要求の重複を回避できます。

NetScaler がクライアント認証用に **Kerberos** を実装する方法

December 8, 2023

> 重要
>
> Kerberos/NTLM 認証は、NetScaler 9.3 nCore リリース以降でのみサポートされており、トラフィック管理仮想サーバーの認証、承認、監査にのみ使用できます。

NetScaler は、Kerberos 認証に関連するコンポーネントを以下の方法で処理します。

キー・ディストリビューション・センター **(KDC)**

Windows 2000 サーバ以降のバージョンでは、ドメインコントローラと KDC は Windows サーバの一部です。Windows Server が稼働している場合は、ドメインコントローラと KDC が設定されていることを示します。KDC

は Active Directory サーバでもあります。

> 注
>
> すべての Kerberos インタラクションは Windows Kerberos ドメインコントローラーで検証されます。

認証サービスとプロトコルネゴシエーション

NetScaler アプライアンスは、認証、承認、および監査トラフィック管理認証仮想サーバーでの Kerberos 認証をサポートします。Kerberos 認証が失敗した場合、NetScaler は NTLM 認証を使用します。

デフォルトでは、Windows 2000 Server 以降の Windows Server バージョンでは、認証、承認、および監査に Kerberos を使用します。認証タイプとして NEGITEATE を使用して認証ポリシーを作成すると、NetScaler は認証、承認、および監査に Kerberos プロトコルを使用しようとします。クライアントのブラウザが Kerberos チケットを受信できない場合、NetScaler は NTLM 認証を使用します。このプロセスはネゴシエーションと呼ばれます。

次のいずれかの場合、クライアントは Kerberos チケットを受信できない可能性があります。

- Kerberos はクライアントではサポートされていません。
- Kerberos はクライアントで有効になっていません。
- クライアントは KDC のドメインとは別のドメインにあります。
- クライアントは KDC のアクセスディレクトリにアクセスできません。

Kerberos/NTLM 認証の場合、NetScaler は NetScaler アプライアンス上にローカルに存在するデータを使用しません。

承認

トラフィック管理仮想サーバーは、負荷分散仮想サーバーでもコンテンツスイッチング仮想サーバーでもかまいません。

監査

NetScaler アプライアンスは、以下の監査ログによる Kerberos 認証の監査をサポートしています。

- トラフィック管理のエンドユーザーアクティビティの完全な監査証跡
- SYSLOG およびハイパフォーマンス TCP ロギング
- システム管理者の完全な監査証跡
- すべてのシステムイベント
- スクリプト可能なログ形式

サポート環境

Kerberos 認証には、NetScaler 上の特定の環境は必要ありません。クライアント (ブラウザ) は Kerberos 認証をサポートする必要があります。

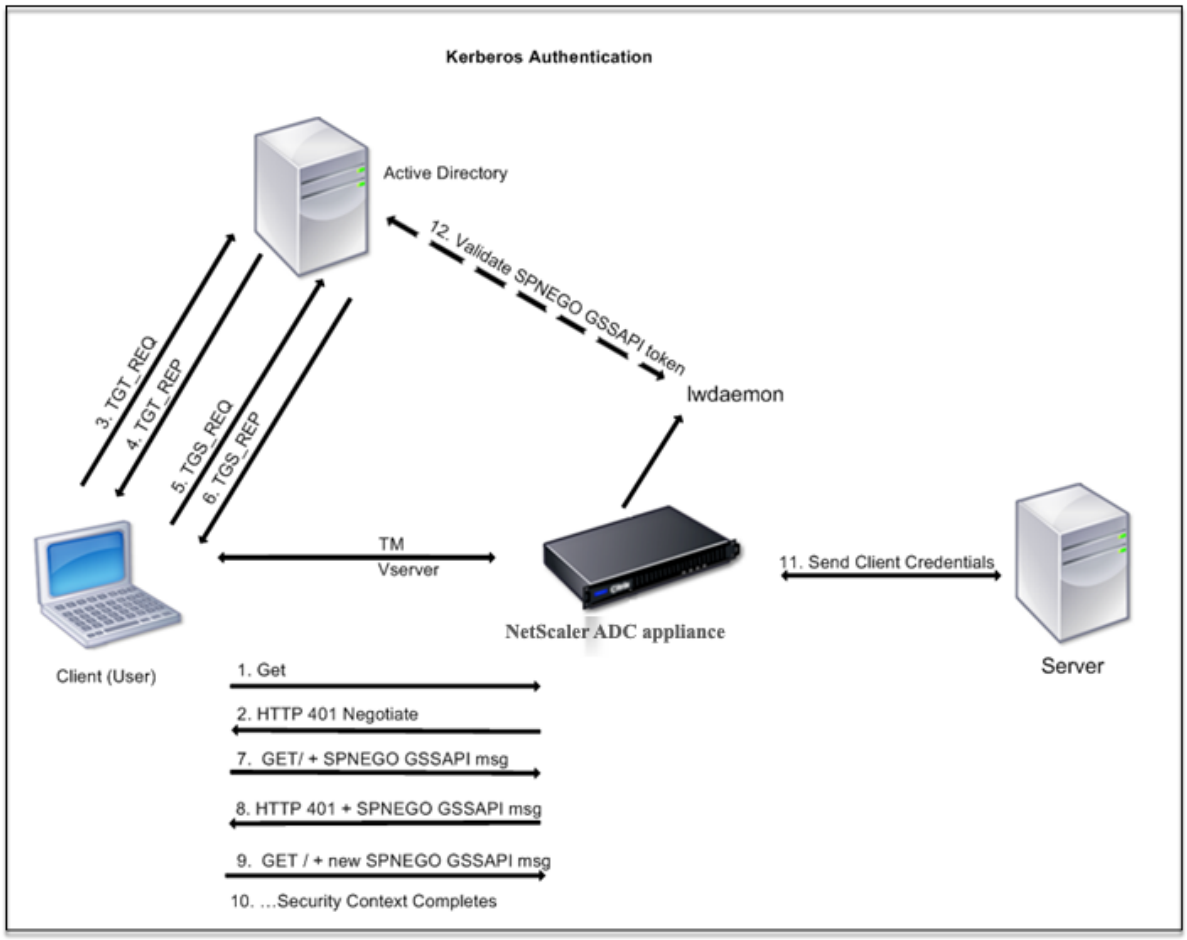
高可用性

高可用性セットアップでは、アクティブな NetScaler のみがドメインに参加します。フェイルオーバーの場合、NetScaler lwagent デーモンはセカンダリ NetScaler アプライアンスをドメインに参加させます。この機能には特別な設定は必要ありません。

Kerberos 認証プロセス

次の図は、NetScaler 環境での Kerberos 認証の一般的なプロセスを示しています。

図 1. NetScaler でのケルベロス認証プロセス



Kerberos 認証は次の段階で行われます。

クライアントは **KDC** に対して自分自身を認証します

1. NetScaler アプライアンスはクライアントから要求を受け取ります。
2. NetScaler アプライアンス上のトラフィック管理（負荷分散またはコンテンツスイッチング）仮想サーバーは、クライアントにチャレンジを送信します。
3. このチャレンジに応えるために、クライアントは Kerberos チケットを受け取ります。
    - クライアントは KDC の認証サーバーにチケット交付チケット (TGT) のリクエストを送信し、TGT を受信します。(図「Kerberos 認証プロセス」の 3、4 を参照してください。)
    - クライアントは TGT を KDC のチケット交付サーバーに送信し、Kerberos チケットを受け取ります。(図「Kerberos 認証プロセス」の 5、6 を参照してください。)

> 注
>
> クライアントに有効期限が切れていない Kerberos チケットが既にある場合は、上記の認証プロセスは必要ありません。さらに、SPNEGO をサポートする Web サービス、.NET、J2EE などのクライアントは、ターゲットサーバーの Kerberos チケットを取得し、SPNEGO トークンを作成し、HTTP リクエストを送信するときにそのトークンを HTTP ヘッダーに挿入します。クライアント認証プロセスは実行されません。

クライアントはサービスをリクエストします。

1. クライアントは、SPNEGO トークンと HTTP リクエストを含む Kerberos チケットを NetScaler 上のトラフィック管理仮想サーバーに送信します。SPNEGO トークンには必要な GSSAPI データが含まれています。
2. NetScaler アプライアンスは、クライアントと NetScaler 間のセキュリティコンテキストを確立します。NetScaler が Kerberos チケットで提供されたデータを受け入れられない場合、クライアントは別のチケットを取得するよう求められます。このサイクルは、GSSAPI データが許容範囲内になり、セキュリティコンテキストが確立されるまで繰り返されます。NetScaler 上のトラフィック管理仮想サーバーは、クライアントと物理サーバー間の HTTP プロキシとして機能します。

NetScaler アプライアンスが認証を完了します。

1. セキュリティコンテキストが完了すると、トラフィック管理仮想サーバーは SPNEGO トークンを検証します。
2. 有効な SPNEGO トークンから、仮想サーバーはユーザー ID と GSS 認証情報を抽出し、認証デーモンに渡します。
3. 認証が成功すると、Kerberos 認証が完了します。

NetScaler アプライアンスでのケルベロス認証の設定

August 16, 2023

このトピックでは、CLI と GUI を使用して NetScaler アプライアンスで Kerberos 認証を構成する詳細な手順を説明します。

CLI での **Kerberos** 認証の設定

1. 認証、承認、および監査機能を有効にして、アプライアンス上のトラフィックの認証を確実に行います。

   ns-cli-promp **ns** 機能 **AAA** を有効にする

2. キータブファイルを NetScaler アプライアンスに追加します。Kerberos 認証中にクライアントから受け取ったシークレットを復号するには、キータブファイルが必要です。1 つのキータブファイルには、NetScaler アプライアンス上のトラフィック管理仮想サーバーにバインドされているすべてのサービスの認証詳細が含まれています。

   まず、Active Directory サーバーでキータブファイルを生成してから、NetScaler アプライアンスに転送します。

   - Active Directory サーバーにログオンし、次のコマンドを使用して Kerberos 認証用のユーザーを追加します。

     ```
     net user <username> <password> /add
     ```

     > 注
     >
     > [ユーザーのプロパティ] セクションで、[次回のログオン時にパスワードを変更] オプションが選択されておらず、[パスワードが期限切れにならない] オプションが選択されていることを確認します。

   - HTTP サービスを上記のユーザーにマップし、keytab ファイルをエクスポートします。たとえば、Active Directory サーバーで次のコマンドを実行します。

     ```
     ktpass /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM
        /pass <user password> /mapuser newacp\\dummy /ptype KRB5\
        _NT\_PRINCIPAL
     ```

     > 注
     >
     > 複数のサービスに認証が必要な場合は、複数のサービスをマッピングできます。さらに多くのサービスをマップする場合は、サービスごとに上記のコマンドを繰り返します。出力ファイルには、同じ名前でも別の名前でもかまいません。

   - unix **ftp** コマンドまたはその他の任意のファイル転送ユーティリティを使用して、キータブファイルを NetScaler アプライアンスに転送します。キータブファイルを NetScaler アプライアンスの/nsconfig/krb/ディレクトリにアップロードします。

3. NetScaler アプライアンスは、完全修飾ドメイン名（FQDN）からドメインコントローラーの IP アドレスを取得する必要があります。したがって、NetScaler ADC を DNS サーバーで構成することをお勧めします。

   ns-cli-promp **DNS** ネームサーバーを追加 <ip-address>

> 注
>
> または、静的ホストエントリを追加するか、他の方法を使用して、NetScaler アプライアンスがドメインコントローラーの FQDN 名を IP アドレスに変換できるようにすることもできます。

4. 認証アクションを設定し、認証ポリシーに関連付けます。

   - ネゴシエートアクションを設定します。この構成により、Kerberos キー配布センター (KDC) として使用される Active Directory サーバーのアクション (プロファイル) が作成されます。プロファイルには、その AD KDC サーバーとの通信に必要なすべての構成データが含まれます。

     ns-cli-prompt> **add authentication negotiateAction** <name> -domain <domain name> -domainUser <domain user name> -domainUserPasswd <domain user password> -defaultAuthenticationGroup <default authentication group> -keytab <string> -NTLMPath <string>

     注: ドメインユーザおよびドメイン名の設定については、クライアントに移動し、次の例に示すように klist コマンドを使用します。

     クライアント: ユーザー名 @ AAA.LOCAL

     サーバー:http/onprem_idp.AAA.local @ AAA.LOCAL

     add authentication negotiateAction <name> -domain -domainUser <HTTP/onprem_idp.aaa.local>

   - ネゴシエートポリシーを設定し、ネゴシエートアクションをこのポリシーに関連付けます。

     ns-cli-prompt> **add authentication negotiatePolicy** <name> <rule> <reqAction>

5. 認証仮想サーバを作成し、ネゴシエートポリシーを関連付けます。

   - 認証仮想サーバーを作成します。

     ns-cli-prompt> **add authentication vserver** <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>

   - ネゴシエートポリシーを認証仮想サーバーにバインドします。

     ns-cli-prompt> **bind authentication vserver** <name> -policy <negotiatePolicyName>

6. 認証仮想サーバーをトラフィック管理 (負荷分散またはコンテンツスイッチング) 仮想サーバーに関連付けます。

   ns-cli-prompt> **set lb vserver** <name> -authn401 ON -authnVsName <string>

   > 注
   >
   > コンテンツスイッチング仮想サーバーでも同様の構成を行うことができます。

7. 次の操作を行って、設定を確認します。

- FQDN を使用して、トラフィック管理仮想サーバにアクセスします。たとえば、サンプル
- CLI でセッションの詳細を表示します。

  ns-cli-promp **AAA** セッションを表示

GUI での **Kerberos** 認証の設定

1. 認証、認可、および監査機能を有効にします。

   [システム] > [設定] に移動し、[基本機能の構成] をクリックして、認証、承認、および監査機能を有効にします。

2. 前述の CLI 手順のステップ 2 で説明したように、keytab ファイルを追加します。
3. DNS サーバーを追加します。

   [トラフィック管理**] > [DNS] > [**ネームサーバー] に移動し、DNS サーバーの IP アドレスを指定します。

4. ネゴシエートアクションとポリシーを設定します。

   [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**認証**] > [**高度なポリシー] > [ポリシー] に移動し、アクションタイプとして [ネゴシエート [**ADD**] をクリックして新しい認証ネゴシエートサーバーを作成するか、[**Edit**] をクリックして既存の詳細を設定します。

5. ネゴシエートポリシーを認証仮想サーバーにバインドします。

   [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] に移動し、ネゴシエートポリシーを認証仮想サーバに関連付けます。

6. 認証仮想サーバーをトラフィック管理 (負荷分散またはコンテンツスイッチング) 仮想サーバーに関連付けます。

   [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、関連する認証設定を指定します。

   > 注
   >
   > コンテンツスイッチング仮想サーバーでも同様の構成を行うことができます。

7. 上記の CLI 手順のステップ 7 で詳述した設定を確認します。

クライアントでの **Kerberos** 認証の設定

August 15, 2023

認証に Kerberos を使用するには、ブラウザで Kerberos サポートを設定する必要があります。Kerberos 準拠のブラウザならどれでも使用できます。Internet Explorer と Mozilla Firefox で Kerberos サポートを設定する手順は次のとおりです。他のブラウザについては、ブラウザのマニュアルを参照してください。

Internet Explorer で **Kerberos** 認証を構成するには

1. [ツール] メニューの [インターネットオプション] を選択します。
2. **[**** セキュリティ] タブの [ローカルイントラネット] をクリックし、[サイト] をクリックします。******
3. ローカルイントラネットダイアログボックスで、「イントラネットネットワークを自動的に検出する」オプションが選択されていることを確認し、「詳細設定」をクリックします。
4. ローカルイントラネットダイアログボックスで、NetScaler アプライアンス上のトラフィック管理仮想サーバーのドメインの Web サイトを追加します。指定されたサイトはローカルイントラネットサイトになります。
5. 「閉じる」または「**OK**」をクリックしてダイアログ・ボックスを閉じます。

Mozilla Firefox をケルベロス認証用に設定するには

1. お使いのコンピューターで Kerberos が適切に設定されていることを確認してください。
2. URL バーに about: config と入力します。
3. フィルターテキストボックスに「network.negotiate」と入力します。
4. network.negotiate-auth.delegation-uris を追加したいドメインに変更してください。
5. network.negotiate-auth.trusted-uris を追加したいドメインに変更してください。

> 注**:** Windows を実行している場合は、フィルターテキストボックスに sspi と入力し、network.auth.use-sspi オプションを False に変更する必要もあります。

Kerberos 認証を物理サーバーからオフロードする

August 15, 2023

NetScaler アプライアンスは、認証タスクをサーバーからオフロードできます。物理サーバーがクライアントからの要求を認証する代わりに、NetScaler はすべてのクライアント要求を認証してから、バインドされた物理サーバーに転送します。ユーザー認証は、Active Directory トークンに基づいています。

NetScaler と物理サーバー間の認証は行われず、認証のオフロードはエンドユーザーには意識されません。Windows コンピュータに最初にログオンした後、エンドユーザーはポップアップまたはログオンページに追加の認証情報を入力する必要はありません。

現在の NetScaler アプライアンスリリースでは、Kerberos 認証はトラフィック管理仮想サーバーの認証、承認、監査にのみ使用できます。Kerberos 認証は、NetScaler Gateway アドバンスドエディションアプライアンスの SSL VPN または NetScaler ADC アプライアンス管理ではサポートされていません。

Kerberos 認証には、NetScaler アプライアンスとクライアントブラウザーでの構成が必要です。

NetScaler アプライアンスで **Kerberos** 認証を構成するには

> 注
>
> 次の設定例で使用されるパスワードは単なる例であり、実際の設定パスワードではありません。

1. Active Directory にユーザーアカウントを作成します。ユーザーアカウントを作成するときは、[ユーザープロパティ] セクションで次のオプションを確認します。
   - [次回ログオン時にパスワードを変更する] オプションを選択していないことを確認します。
   - 必ず [パスワードは期限切れではない] オプションを選択してください。
2. AD サーバーの CLI コマンドプロンプトで、次のように入力します。
   - ktpass -princ HTTP/kerberos.crete.lab.net@crete.lab.net -ptype KRB5_NT_PRINCIPAL -mapuser kerbuser@crete.lab.net -mapop set -pass Citrix1 -out C:\kerbtabfile.txt

   > 注
   >
   > 上記のコマンドは必ず 1 行で入力してください。上記のコマンドの出力は C:\kerbtabfile.txt ファイルに書き込まれます。

3. セキュアコピー（SCP）クライアントを使用して、kerbtabfile.txt ファイルを NetScaler アプライアンスの/etc ディレクトリにアップロードします。
4. 次のコマンドを実行して、DNS サーバーを NetScaler アプライアンスに追加します。
   - add dns nameserver 1.2.3.4

   NetScaler アプライアンスは、DNS サーバーがないと Kerberos リクエストを処理できません。必ず、Microsoft Windows ドメインで使用されているものと同じ DNS サーバーを使用してください。
5. NetScaler のコマンドラインインターフェイスに切り替えます。
6. 次のコマンドを実行して、Kerberos 認証サーバーを作成します。
   - add authentication negotiateAction KerberosServer -domain “crete.lab.net” -domainUser kerbuser -domainUserPasswd Citrix1 -keytab /var/mykcd.keytab

   > 注
   >
   : キータブが使用できない場合は、ドメイン、domainUser、および-domainUserPasswd のパラメータを指定できます。

7. 次のコマンドを実行して、ネゴシエーションポリシーを作成します。
   - `add authentication negotiatePolicy Kerberos-Policy "REQ.IP. DESTIP == 192.168.17.200"KerberosServer<!--NeedCopy-->`
8. 次のコマンドを実行して、認証仮想サーバーを作成します。

- `add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.lab.net<!--NeedCopy-->`

9. 次のコマンドを実行して、Kerberos ポリシーを認証仮想サーバーにバインドします。

   - `bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100<!--NeedCopy-->`

10. 次のコマンドを実行して、SSL 証明書を認証仮想サーバーにバインドします。GUI NetScaler アプライアンスからインストールできるテスト証明書のいずれかを使用できます。次のコマンドを実行して、ServerTestCert サンプル証明書を使用します。

    - `bind ssl vserver Kerb-Auth -certkeyName ServerTestCert<!--NeedCopy-->`

11. IP アドレス 192.168.17.200 を使用して HTTP 負荷分散仮想サーバーを作成します。

    NetScaler 9.3 リリースが 9.3.47.8 より古い場合は、コマンドラインインターフェイスから仮想サーバーを作成するようにしてください。

12. 次のコマンドを実行して、認証仮想サーバーを構成します。

    - `set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth<!--NeedCopy-->`

13. Web ブラウザのアドレスバーにホスト名 Example を入力します。

    Kerberos 認証がブラウザで設定されていないため、Web ブラウザに認証ダイアログボックスが表示されます。

    > 注
    >
    > Kerberos 認証には、クライアント上で特定の設定が必要です。クライアントがホスト名を解決できることを確認します。これにより、Web ブラウザが HTTP 仮想サーバーに接続します。

14. クライアントコンピュータの Web ブラウザで Kerberos を構成します。

    - Internet Explorer での構成については、「Kerberos 認証用の Internet Explorer の構成」を参照してください。
    - Mozilla Firefox での設定については、Internet Explorer の Kerberos 認証の設定を参照してください。

15. 認証なしでバックエンド物理サーバにアクセスできるかどうかを確認します。

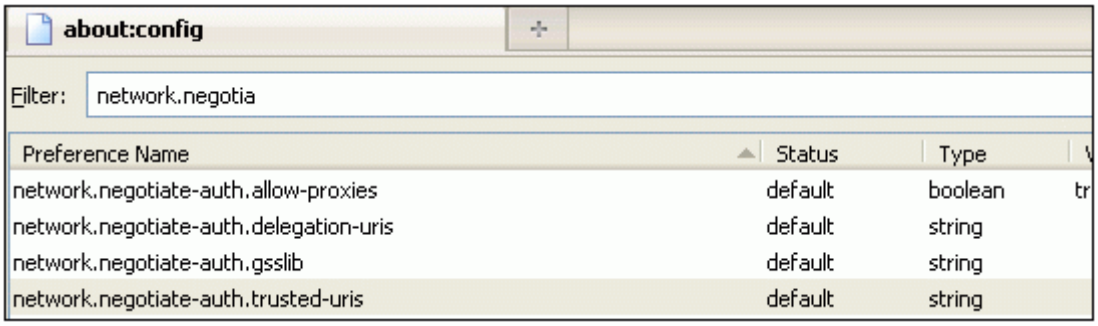
Internet Explorer で **Kerberos** 認証を構成するには

1. [ツール] メニューの [インターネットオプション] を選択します。

2. [セキュリティ] タブを有効にします。
3. [セキュリティ設定を変更するゾーンを選択してください] セクションから [ローカルイントラネット] を選択します。
4. [サイト] をクリックします。
5. [詳細設定] をクリックします。
6. URL、例] を指定し、[追加] をクリックします。
7. **Internet Explorer** を再起動します。

Mozilla Firefox をケルベロス認証用に設定するには

1. ブラウザのアドレスバーに about: config と入力します。
2. 警告免責事項をクリックします。
3. [フィルタ] ボックスに「**network.negotiate-auth.trusted-URI** 」と入力します。
4. [**Network.negotiate-auth.trusted-URI]** をダブルクリックします。サンプル画面を以下に示します。



5. [文字列値の入力] ダイアログボックスで、www.crete.lab.net を指定します。
6. Firefox を再起動します。

シングルサインオンのタイプ

August 15, 2023

NetScaler の認証、承認、および監査機能は、次のシングルサインオンタイプをサポートしています。

- **NetScaler kerberos** シングルサインオン： **NetScaler** アプライアンスは、**Kerberos** 5 プロトコルを使用したシングルサインオン（SSO）をサポートするようになりました。ユーザーは、プロキシであるアプリケーション Delivery Controller（ADC）にログオンし、保護されたリソースへのアクセスを提供します。詳細については、「NetScaler Kerberos シングルサインオン」を参照してください。

- 基本認証、ダイジェスト認証、および **NTLM** 認証用の **SSO**： NetScaler および NetScaler Gateway のシングルサインオン（SSO）構成は、グローバルレベルでもトラフィックレベルでも有効にできます。デフォルトでは SSO 設定は OFF で、管理者はトラフィックごとに SSO を有効にすることも、グローバルに有効にすることもできます。セキュリティの観点から、Citrix では管理者に SSO をグローバルにオフにし、トラフィックごとに有効にすることを推奨しています。この機能拡張は、特定のタイプの SSO メソッドをグローバルに無効にすることにより、SSO 構成をより安全にすることです。詳細については、ベーシック、ダイジェスト、NTLM 認証の SSOを参照してください。

NetScaler kerberos のシングルサインオン

August 15, 2023

NetScaler アプライアンスは、Kerberos 5 プロトコルを使用したシングルサインオン（SSO）をサポートするようになりました。ユーザーは、プロキシであるアプリケーション Delivery Controller（ADC）にログオンし、保護されたリソースへのアクセスを提供します。

NetScaler Kerberos SSO の実装では、基本認証、NTLM 認証、またはフォームベース認証に依存する SSO メソッドにユーザーのパスワードが必要です。Kerberos SSO にはユーザーのパスワードは必要ありません。ただし、Kerberos SSO に障害が発生し、NetScaler アプライアンスがユーザーのパスワードを取得した場合、NetScaler アプライアンスはそのパスワードを使用して NTLM SSO を試みます。

ユーザーのパスワードが利用可能で、KCD アカウントがレルムで構成されていて、委任されたユーザー情報が存在しない場合、Citrix AD Kerberos SSO エンジンはユーザーになりすまして承認されたリソースへのアクセスを取得します。なりすましは、制約のない委任とも呼ばれます。

NetScaler Kerberos SSO エンジンは、委任されたアカウントを使用してユーザーに代わって保護されたリソースへのアクセスを取得するように構成することもできます。この構成には、委任されたユーザーの認証情報、キータブ、または委任されたユーザー証明書とそれに一致する CA 証明書が必要です。委任されたアカウントを使用する構成は、制約付き委任と呼ばれます。

NetScaler kerberos SSO の概要

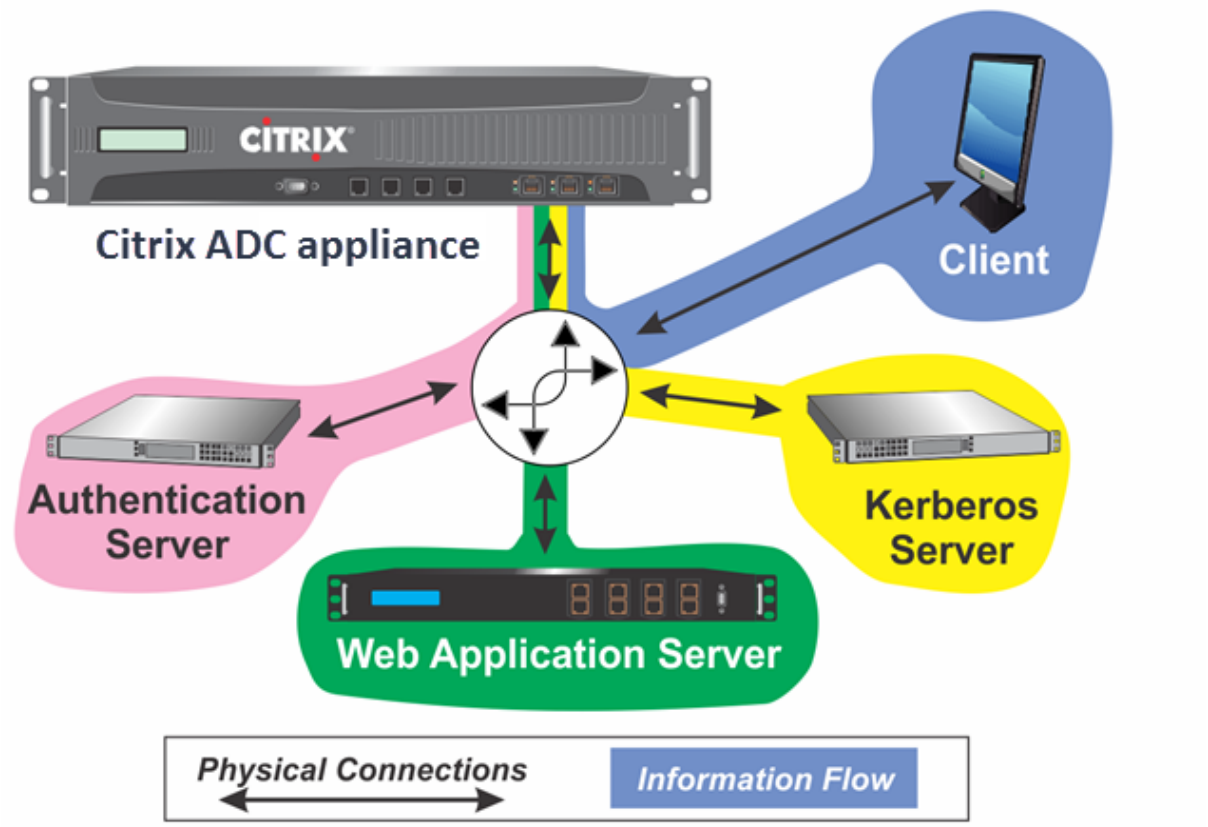
August 15, 2023

NetScaler Kerberos SSO 機能を使用するには、ユーザーはまず Kerberos またはサポートされているサードパーティ認証サーバーで認証します。認証されると、ユーザーは保護された Web アプリケーションへのアクセスをリクエストします。Web サーバーは、ユーザーがその Web アプリケーションにアクセスする権限を持っていることの証明を要求して応答します。ユーザーのブラウザは Kerberos サーバーに接続し、Kerberos サーバーはユーザーがそのリソースにアクセスする権限を持っていることを確認し、その証拠となるサービスチケットをユーザーのブラウザ

ーに提供します。ブラウザは、サービスチケットを添付した状態で、ユーザーのリクエストを Web アプリケーションサーバーに再送信します。Web アプリケーションサーバーはサービスチケットを検証し、ユーザーがアプリケーションにアクセスできるようにします。

認証、承認、および監査のトラフィック管理では、次の図に示すようにこのプロセスを実装しています。この図は、LDAP 認証と Kerberos 認証を備えた安全なネットワーク上で、NetScaler アプライアンスと認証、承認、および監査トラフィック管理を通る情報の流れを示しています。他の種類の認証を使用する認証、承認、および監査トラフィック管理環境の情報フローは基本的に同じですが、詳細が異なる場合があります。

図 1: LDAP と Kerberos を使用するセキュアなネットワーク



Kerberos 環境で認証と認可によるトラフィック管理の認証、承認、監査を行うには、次のアクションを実行する必要があります。

1. クライアントは、NetScaler アプライアンス上のトラフィック管理仮想サーバーにリソースの要求を送信します。
2. トラフィック管理仮想サーバーは要求を認証仮想サーバーに渡し、認証仮想サーバーはクライアントを認証してから要求をトラフィック管理仮想サーバーに戻します。
3. トラフィック管理仮想サーバーは、クライアントの要求をウェブアプリケーションサーバーに送信します。
4. Web アプリケーションサーバーは、Kerberos 認証を要求する 401 Unauthorized メッセージでトラフィック管理仮想サーバーに応答し、クライアントが Kerberos をサポートしていない場合は NTLM 認証にフォールバックします。

5. トラフィック管理仮想サーバーは Kerberos SSO デーモンと通信します。
6. Kerberos SSO デーモンは Kerberos サーバに接続し、保護されたアプリケーションへのアクセスを許可するサービスチケットをリクエストするためのチケット交付チケット (TGT) を取得します。
7. Kerberos SSO デーモンは、ユーザーのサービスチケットを取得し、そのチケットをトラフィック管理仮想サーバーに送信します。
8. トラフィック管理仮想サーバーは、チケットをユーザーの最初のリクエストに添付し、変更されたリクエストをウェブアプリケーションサーバーに送り返します。
9. Web アプリケーションサーバーは 200 OK メッセージで応答します。

これらのステップはクライアントには意識されず、クライアントはリクエストを送信してリクエストされたリソースを受け取るだけです。

NetScaler Kerberos SSO と認証方法の統合

認証、承認、監査のトラフィック管理認証メカニズムはすべて、NetScaler Kerberos SSO をサポートしています。認証、承認、および監査のトラフィック管理は、Kerberos SSO メカニズム、CAC（スマートカード）、SAML 認証メカニズムによる Kerberos SSO メカニズム、および NetScaler アプライアンスに対するあらゆる形式のクライアント認証をサポートします。また、クライアントが HTTP 基本認証またはフォームベース認証のいずれかを使用して NetScaler アプライアンスにログオンする場合、HTTP ベーシック、HTTP ダイジェスト、フォームベース、および NTLM（バージョン 1 および 2）の SSO メカニズムもサポートします。

次の表は、サポートされている各クライアント側の認証方法と、そのクライアント側の方法でサポートされているサーバー側の認証方法を示しています。

表 1. サポートされている認証方法

	ベーシック/ダイジェスト/NTLM	Kerberos 制約付き委任	ユーザーなりすまし
CAC (スマートカード): SSL/T LS レイヤで		X	X
フォームベース (LDAP/RADIUS/TACACS)	X	X	X
HTTP ベーシック (LDAP/RADIUS/TACACS)	X	X	X
Kerberos		X	
NT LM v1/v2		X	X
SAML		X	
SAML ツーファクター	X	X	X

	ベーシック/ダイジェスト/NTLM	Kerberos 制約付き委任	ユーザーなりすまし
2 段階認証書	X	X	X

NetScaler ADC の SSO を設定する

August 15, 2023

NetScaler SSO は、偽装または委任の 2 つの方法のいずれかで機能するように構成できます。偽装による SSO は、委任による SSO よりも簡単な設定であるため、設定で許可されている場合に適しています。なりすましによって NetScaler SSO を構成するには、ユーザーのユーザー名とパスワードが必要です。

委任によって NetScaler SSO を構成するには、ユーザーのユーザー名とパスワード、ユーザー名と暗号化されたパスワードを含むキータブ構成、または委任されたユーザー証明書と対応する CA 証明書のいずれかの形式の委任ユーザーの資格情報が必要です。

NetScaler SSO を構成するための前提条件

NetScaler SSO を構成する前に、Web アプリケーションサーバーへのトラフィックと認証を管理するように NetScaler アプライアンスを完全に構成する必要があります。したがって、これらの Web アプリケーションサーバーに対して負荷分散またはコンテンツスイッチを構成し、認証、承認、および監査を構成する必要があります。また、アプライアンス、LDAP サーバー、Kerberos サーバー間のルーティングを確認する必要があります。

ネットワークがこの方法でまだ設定されていない場合は、次の設定作業を実行します。

- Web アプリケーションサーバーごとにサーバーとサービスを構成します。
- Web アプリケーションサーバーとの間で送受信されるトラフィックを処理するようにトラフィック管理仮想サーバーを構成します。

NetScaler コマンドラインからこれらの各タスクを実行するための簡単な手順と例を以下に示します。詳細については、「認証仮想サーバーのセットアップ」を参照してください。

> 注
>
> NetScaler 13.1 以降のリリースでは、NetScaler アプライアンスからのバックエンドサーバーの Kerberos SSO 認証中に、ルートドメインとツリードメイン間のトラバーサルがサポートされます。

CLI を使用してサーバーとサービスを作成するには

NetScaler SSO がサービスの TGS（サービスチケット）を取得するには、NetScaler アプライアンス上のサーバーエンティティに割り当てられた FQDN が Web アプリケーションサーバーの FQDN と一致するか、サーバーエンティティ名が Web アプリケーションサーバーの NetBIOS 名と一致する必要があります。次のいずれかの方法を使用できます。

- Web アプリケーションサーバーの FQDN を指定して、NetScaler サーバーエンティティを構成します。
- Web アプリケーションサーバーの IP アドレスを指定して NetScaler サーバーエンティティを構成し、サーバーエンティティに Web アプリケーションサーバーの NetBIOS 名と同じ名前を割り当てます。

コマンドプロンプトで、次のコマンドを入力します。

```
1 -  add server name <serverFQDN>
2
3 -  add service name serverName serviceType port
4 <!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- サーバー名。このサーバーを参照するために使用する NetScaler アプライアンスの名前。
- **serverFQDN**。サーバーの FQDN。サーバにドメインが割り当てられていない場合は、サーバの IP アドレスを使用し、サーバエンティティ名が Web アプリケーションサーバの NetBIOS 名と一致することを確認します。
- サービス名。このサービスを指すために使用する NetScaler アプライアンスの名前。
- **type**。サービスによって使用されるプロトコル（HTTP または MSSQLSVC）。
- ポート。サービスがリッスンするポート。HTTP サービスは通常、ポート 80 でリッスンします。セキュア HTTPS サービスは通常、ポート 443 でリッスンします。

例**:**

次の例では、Web アプリケーションサーバー was1.example.com の NetScaler アプライアンスにサーバーとサービスのエントリを追加します。最初の例では Web アプリケーションサーバーの FQDN を使用し、2 番目の例では IP アドレスを使用します。

Web アプリケーションサーバーの FQDN である was1.example.com を使用してサーバーとサービスを追加するには、次のコマンドを入力します。

```
1 add server was1 was1.example.com
2 add service was1service was1 HTTP 80
3 <!--NeedCopy-->
```

Web アプリケーションサーバーの IP アドレスと NetBIOS 名 (Web アプリケーションサーバーの IP アドレスは 10.237.64.87、NetBIOS 名は WAS1) を使用してサーバーとサービスを追加するには、次のコマンドを入力します。

```
1 add server WAS1 10.237.64.87
```



```
add service was1service WAS1 HTTP 80
<!--NeedCopy-->
```

CLI を使用してトラフィック管理仮想サーバーを作成するには

トラフィック管理仮想サーバは、クライアントと Web アプリケーションサーバ間のトラフィックを管理します。トラフィック管理サーバーとして、負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーを使用できます。SSO 設定はどちらのタイプでも同じです。

負荷分散仮想サーバーを作成するには、コマンドプロンプトで次のコマンドを入力します。

```
add lb vserver <vserverName> <type> <IP> <port>
<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **vServerName**—この仮想サーバーを参照するために使用する NetScaler アプライアンスの名前。
- **type**：サービスによって使用されるプロトコル（HTTP または MSSQLSVC）。
- [**IP**]：仮想サーバに割り当てられた IP アドレス。これは通常、LAN 上の IANA 予約済みの非パブリック IP アドレスです。
- **port**：サービスがリッスンするポート。HTTP サービスは通常、ポート 80 でリッスンします。セキュア HTTPS サービスは通常、ポート 443 でリッスンします。

例**:**

tmvserver1 という名前の負荷分散仮想サーバーを、ポート 80 の HTTP トラフィックを管理する構成に追加し、LAN IP アドレス 10.217.28.20 を割り当てて、負荷分散仮想サーバーを wasservice1 サービスにバインドするには、次のコマンドを入力します。

```
add lb vserver tmvserver1 HTTP 10.217.28.20 80
bind lb vserver tmvserv1 wasservice1
<!--NeedCopy-->
```

CLI を使用して認証仮想サーバーを作成するには

認証仮想サーバは、クライアントと認証（LDAP）サーバ間の認証トラフィックを管理します。認証仮想サーバーを作成するには、コマンドプロンプトで次のコマンドを入力します。

```
add authentication vserver <authvserverName> SSL <IP> 443
<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **AuthVServerName** —この認証仮想サーバーを参照するために使用する NetScaler アプライアンスの名前。文字、数字、またはアンダースコア文字（_）で始まり、文字、数字、ハイフン（-）、ピリオド（.）ポンド

（#）、スペース（）、アットマーク（@）、等号（=）、コロン（:）、およびアンダースコア文字のみを含める必要があります。認証仮想サーバーを追加した後で rename authentication vserver コマンドを使用して変更できます。

- [**IP**]：認証仮想サーバに割り当てられた IP アドレス。トラフィック管理仮想サーバと同様に、このアドレスは通常、LAN 上の IANA 予約済みの非パブリック IP になります。
- **domain**：仮想サーバに割り当てられたドメイン。これは通常、ネットワークのドメインになります。認証仮想サーバーを構成するときは、必須ではありませんが、すべての大文字でドメインを入力するのが通例です。

例**:**

authverver1 という名前の認証仮想サーバーを構成に追加し、LAN IP 10.217.28.21 とドメイン EXAMPLE.COM を割り当てるには、次のコマンドを入力します。

```
add authentication vserver authvserver1 SSL 10.217.28.21 443
<!--NeedCopy-->
```

認証プロファイルを使用するようにトラフィック管理仮想サーバを構成するには

認証仮想サーバーは、単一ドメインまたは複数のドメインの認証を処理するように構成できます。複数のドメインの認証をサポートするように構成されている場合は、認証プロファイルを作成し、その認証プロファイルを使用するようにトラフィック管理仮想サーバーを構成して、NetScaler SSO のドメインも指定する必要があります。

> 注
>
> トラフィック管理仮想サーバは、負荷分散（lb）仮想サーバまたはコンテンツスイッチング（cs）仮想サーバのいずれかになります。次の手順は、負荷分散仮想サーバーを使用していることを前提としています。コンテンツスイッチ仮想サーバーを構成するには、set lb vserver
> を set cs vserver に置き換えるだけです。それ以外の手順は同じです。

認証プロファイルを作成し、トラフィック管理仮想サーバーで認証プロファイルを構成するには、次のコマンドを入力します。

```
-  add authentication authnProfile <authnProfileName> {
 -authvserverName <string> }
  {
 -authenticationHost <string> }
  {
 -authenticationDomain <string> }

-  set lb vserver \<vserverName\> -authnProfile <authnprofileName>
<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **authnProfileName**：認証プロファイルの名前。英字、数字、またはアンダースコア文字 (_) で始まり、1～31 の英数字またはハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア文字で構成する必要があります。

- **authvServerName**：このプロファイルが認証に使用する認証仮想サーバの名前。
- **AuthenticationHost**：認証仮想サーバのホスト名。
- 認証ドメイン—**NetScaler**SSO が認証を処理するドメイン。NetScaler アプライアンスがトラフィック管理仮想サーバー Cookie を設定するときに正しいドメインが含まれるように、認証仮想サーバーが複数のドメインの認証を実行する場合に必要です。

例**:**

example.com ドメインの認証用に authnProfile1 という名前の認証プロファイルを作成し、認証プロファイル authnProfile1 を使用するように負荷分散仮想サーバー vserver1 を構成するには、次のコマンドを入力します。

```
add authentication authnProfile authnProfile1 -authnvsName
    authvsesrver1
      -authenticationHost authvsesrver1 -authenticationDomain example.
         com
set lb vserver vserver1 -authnProfile authnProfile1
<!--NeedCopy-->
```

シングルサインオンを構成する

August 15, 2023

NetScaler シングルサインオン（SSO）を偽装認証するように構成することは、SSO よりも委任による認証を構成するよりも簡単なので、構成で許可されている場合に適しています。KCD アカウントを作成するとします。ユーザーのパスワードを使用できます。

ユーザーのパスワードがわからない場合は、委任による認証を行うように NetScaler SSO を構成できます。委任方法では、偽装による認証を行うように SSO を構成するよりも複雑ですが、状況によってはユーザーの資格情報が NetScaler アプライアンスで利用できるとは限らないという柔軟性があります。

偽装または委任の場合は、Web アプリケーションサーバーでも統合認証を有効にする必要があります。

Web アプリケーションサーバーで統合認証を有効にする

Kerberos SSO が管理する各 Web アプリケーションサーバーに NetScaler Kerberos SSO を設定するには、そのサーバーの構成インターフェイスを使用して、認証を要求するようにサーバーを構成します。Kerberos をサポートしていないクライアントの場合は NTLM にフォールバックして、Kerberos (ネゴシエート) 認証を優先的に選択します。

次に、認証を要求するように Microsoft インターネットインフォメーションサーバー (IIS) を構成する手順を示します。Web アプリケーションサーバで IIS 以外のソフトウェアを使用している場合は、その Web サーバソフトウェアのマニュアルを参照して手順を確認してください。

統合認証を使用するように **Microsoft IIS** を構成するには

1. IIS サーバーにログオンし、インターネットインフォメーションサービスマネージャーを開きます。
2. 統合認証を有効にする Web サイトを選択します。IISM が管理するすべての IIS Web サーバーに対して統合認証を有効にするには、既定の Web サイトの認証設定を構成します。個々のサービス (Exchange、Exadmin、ExchWeb、Public など) に対して統合認証を有効にするには、これらの認証設定をサービスごとに個別に構成します。
3. 既定の Web サイトまたは個々のサービスの [プロパティ] ダイアログボックスを開き、[ディレクトリセキュリティ] タブをクリックします。
4. [****** 認証とアクセス制御**]** の横にある **[**** 編集**]** を選択します。
5. 匿名アクセスを無効にしてください。
6. 統合 Windows 認証を有効にします (のみ)。統合 Windows 認証を有効にすると、Web サーバのプロトコルネゴシエーションが Negotiate、NTLM に自動的に設定される必要があります。これは、Kerberos 非対応デバイスの NTLM にフォールバックする Kerberos 認証を指定します。このオプションが自動的に選択されない場合は、プロトコルネゴシエーションを [Negotiate, NTLM] に手動で設定します。

偽装による **SSO** の設定

NetScaler SSO の KCD アカウントは、偽装によって構成できます。この構成では、NetScaler アプライアンスは、ユーザーが認証サーバーで認証されるときにユーザーのユーザー名とパスワードを取得し、それらの資格情報を使用してユーザーになりすましてチケット交付チケット（TGT）を取得します。ユーザー名が UPN 形式の場合、アプライアンスはユーザーのレルムを UPN から取得します。それ以外の場合は、初期認証時に使用された SSO ドメインまたはセッションプロファイルからユーザーの名前とレルムを抽出して取得します。

> 注
>
> domain なしでユーザー名が既に追加されている場合、domain でユーザー名を追加することはできません。ドメイン付きのユーザー名が最初に追加され、その後にドメインなしの同じユーザー名が追加された場合、NetScaler アプライアンスはそのユーザー名をユーザーリストに追加します。

KCD アカウントを設定するときは、ユーザーがアクセスしているサービスのレルムに realm パラメーターを設定する必要があります。NetScaler アプライアンスまたはセッションプロファイルによる認証でユーザーのレルムを取得できない場合は、同じレルムがユーザーのレルムとしても使用されます。

パスワードを使用して偽装して **SSO** の **KCD** アカウントを作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add aaa kcdaccount <accountname> -realmStr <realm>

<!--NeedCopy-->
```


変数の場合は、次の値を置き換えます。

- アカウント名。KCD アカウント名。
- レルム。NetScaler SSO に割り当てられたドメイン。

例

kcdccount1 という名前の KCD アカウントを追加し、kcdvserver.keytab という名前のキータブを使用するには、次のコマンドを入力します。

```
add aaa kcdAccount kcdaccount1 -keytab kcdvserver.keytab

<!--NeedCopy-->
```

NetScaler GUI を使用して Kerberos 偽装を構成する方法については、「NetScaler サポート」を参照してください。

委任による **SSO** の構成

委任による SSO を設定するには、次のタスクを実行する必要があります。

- 委任されたユーザー証明書による委任を構成する場合は、一致する CA 証明書を NetScaler アプライアンスにインストールし、NetScaler 構成に追加します。
- アプライアンスで KCD アカウントを作成します。アプライアンスはこのアカウントを使用して、保護対象アプリケーションのサービスチケットを取得します。
- Active Directory サーバを設定します。

> 注:
>
> NetScaler アプライアンスでの KCD アカウントの作成と構成について詳しくは、次のトピックを参照してください。
>
> - Kerberos/NTLM による認証、承認、監査の処理
> - NetScaler ADC がクライアント認証用に Kerberos を実装する方法
> - NetScaler アプライアンスでのケルベロス認証の設定

NetScaler アプライアンスへのクライアント **CA** 証明書のインストール

クライアント証明書を使用して NetScaler SSO を構成する場合は、クライアント証明書ドメインと一致する CA 証明書（クライアント CA 証明書）を NetScaler アプライアンスにコピーしてから、CA 証明書をインストールする必要があります。クライアント CA 証明書をコピーするには、任意のファイル転送プログラムを使用して証明書と秘密鍵ファイルを NetScaler アプライアンスに転送し、ファイルをそこに保存します。/nsconfig/ssl

クライアント **CA** 証明書を **NetScaler** アプライアンスにインストールするには　コマンドプロンプトで、次のコマンドを入力します。

```
add ssl certKey <certkeyName> -cert <cert> [(-key <key> [-password]) |
    -fipsKey <fipsKey>][-inform ( DER | PEM )][-expiryMonitor ( ENABLED
    | DISABLED | UNSET ) [-notificationPeriod <positive_integer>]] [-
    bundle ( YES | NO )]

<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **certKeyName**。クライアント CA 証明書の名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、1～31 文字で構成されている必要があります。使用できる文字は、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、ハイフン (-) です。証明書とキーのペアの作成後は変更できません。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます (「my cert」や「my cert」など)。
- 証明書。証明書とキーのペアを形成するために使用される X509 証明書ファイルのフルパス名とファイル名。証明書ファイルは、NetScaler アプライアンスの/nsconfig/ssl/ディレクトリに保存する必要があります。
- キー。X509 証明書ファイルへの秘密キーを含むファイルのフルパス名とファイル名。キーファイルは、NetScaler アプライアンスの/nsconfig/ssl/ディレクトリに保存する必要があります。
- **password**。秘密鍵を指定した場合、秘密鍵の暗号化に使用されるパスフレーズ。暗号化された秘密キーを PEM 形式でロードするには、このオプションを使用します。
- **fipsKey**。FIPS アプライアンスのハードウェアセキュリティモジュール (HSM) 内で作成された FIPS キー、または HSM にインポートされたキーの名前。

  > 注
  >
  > キーと FipsKey のどちらかを指定できますが、両方を指定することはできません。

- **inform**. 証明書と秘密キーファイルの形式 (PEM または DER)。
- **passplain**. 秘密鍵の暗号化に使用されるパスフレーズ。暗号化された秘密キーを PEM 形式で追加する場合に必要です。
- **expiryMonitor**. 証明書の有効期限が近づいたときにアラートを発行するように NetScaler アプライアンスを構成します。有効な値: 有効、無効、未設定。
- **notificationPeriod**.`expiryMonitor`が ENABLED の場合、証明書が期限切れになる前にアラートを発行するまでの日数。
- **bundle**. サーバー証明書をファイル内の発行者の証明書にリンクした後、証明書チェーンを 1 つのファイルとして解析します。可能な値: はい、いいえ。

例

次の例では、指定された委任ユーザー証明書 customer-cert.pem を customer-key.pem キーとともに NetScaler 構成に追加し、パスワード、証明書形式、有効期限モニター、および通知期間を設定します。

委任されたユーザー証明書を追加するには、次のコマンドを入力します。

```
add ssl certKey customer -cert "/nsconfig/ssl/customer-cert.pem"
-key "/nsconfig/ssl/customer-key.pem" -password "dontUseDefaultPWs!"
-inform PEM -expiryMonitor ENABLED [-notificationPeriod 14]

<!--NeedCopy-->
```

KCD アカウントの作成

委任によって NetScaler SSO を構成する場合、KCD アカウントは、ユーザーのログオン名とパスワードを使用するか、ユーザーのログオン名とキータブを使用するか、またはユーザーのクライアント証明書を使用するように構成できます。ユーザー名とパスワードを使用して SSO を構成すると、NetScaler アプライアンスは委任されたユーザーアカウントを使用してチケット交付チケット（TGT）を取得し、TGT を使用して各ユーザーが要求する特定のサービスのサービスチケットを取得します。キータブファイルを使用して SSO を構成すると、NetScaler アプライアンスは委任されたユーザーアカウントとキータブ情報を使用します。委任ユーザー証明書を使用して SSO を構成すると、NetScaler ADC アプライアンスは委任ユーザー証明書を使用します。

> 注:
>
> クロスレルムの場合、委任されたユーザーの servicePrincipalName はhost/<name>の形式である必要があります。この形式でない場合は、委任されたユーザーの servicePrincipalName <servicePrincipalName>をhost/<service-account-samaccountname>に変更します。委任されたユーザーアカウントの属性は、ドメインコントローラーで確認できます。変更する 1 つの方法は、委任されたユーザーのlogonName属性を変更することです。

パスワードを使用して委任して **SSO** の **KCD** アカウントを作成するには　コマンドプロンプトで、次のコマンドを入力します。

```
add aaa kcdAccount <kcdAccount> {
 -realmStr <string> }
  {
 -delegatedUser <string> }
  {
 -kcdPassword  }
  [-userRealm <string>]
[-enterpriseRealm <string>] [-serviceSPN <string>]
<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **kcdAccount** -KCD アカウントの名前。これは必須の議論です。最大長：31
- **realmStr** -ケルベロスの領域。最大長：255

- **delegatedUser** -Kerberos 制約付き委任を実行できるユーザー名。委任されたユーザー名は、ドメインコントローラーの ServicePrincipalName から派生します。クロスレルムの場合、委任されたユーザーの ServicePrincipalName は`host`/<`name`>の形式である必要があります。最大長:255
- **kcdPassword** -委任されたユーザーのパスワード。最大長：31
- **userRealm** -ユーザーのレルム。最大長:255
- **EnterpriseRealm** -ユーザーのエンタープライズレルム。これは、KDC がプリンシパル名ではなくエンタープライズユーザー名を要求する特定の KDC 展開でのみ指定されます。最大長：255
- **serviceSPN** -サービス SPN。指定すると、Kerberos チケットの取得に使用されます。指定しない場合、NetScaler はサービス FQDN を使用して SPN を構築します。最大長：255

例 **(UPN** 形式**):**

委任されたユーザーアカウントを UPN 形式(root)で指定して、パスワードが password1、レルムが EXAMPLE.COM の kcdaccount1 という名前の KCD アカウントを NetScaler アプライアンス構成に追加するには、次のコマンドを入力します。

```
1 add aaa kcdaccount kcdaccount1 -delegatedUser root
2 -kcdPassword password1 -realmStr EXAMPLE.COM
3
4 <!--NeedCopy-->
```

例 **(SPN** 形式**):**

委任されたユーザーアカウントを SPN 形式で指定して、パスワードが password1、レルムが EXAMPLE.COM の kcdaccount1 という名前の KCD アカウントを NetScaler アプライアンス構成に追加するには、次のコマンドを入力します。

```
1 add aaa kcdAccount kcdaccount1 -realmStr EXAMPLE.COM
2 -delegatedUser "host/kcdvserver.example.com" -kcdPassword password1
3
4 <!--NeedCopy-->
```

キータブを使用した委任による **SSO** の **KCD** アカウントの作成　認証に keytab ファイルを使用する場合は、まず keytab を作成します。キータブファイルは、`ktpass` AD サーバーにログオンしてユーティリティを使用して手動で作成することも、NetScaler 構成ユーティリティを使用してバッチスクリプトを作成し、そのスクリプトを AD サーバーで実行してキータブファイルを生成することもできます。次に、FTP または別のファイル転送プログラムを使用してキータブファイルを NetScaler アプライアンスに転送し、/nsconfig/krb ディレクトリに配置します。最後に、委任によって NetScaler ADC SSO の KCD アカウントを構成し、キータブファイルのパスとファイル名を NetScaler ADC アプライアンスに指定します。

> 注**:**
>
> クロスレルムで Keytab ファイルを KCD アカウントの一部として取得する場合は、更新された委任ユーザー名に対して次のコマンドを使用します。

> ドメインコントローラーで、更新された Keytab ファイルを作成します。
>
> ```
> ktpass /princ <servicePrincipalName-with-prefix<host/>Of-delegateUser
> >@<DC REALM in uppercase> /ptype KRB5_NT_PRINCIPAL /mapuser <DC
> REALM in uppercase>\<sAMAccountName> /pass <delegatedUserPassword
> > -out filepathfor.keytab
> ```
>
> `filepathfor.keytab` このファイルは NetScaler アプライアンスに保存でき、ADC KCD アカウントの Keytab 構成の一部として使用できます。

keytab ファイルを手動で作成するには　AD Server コマンドラインにログオンし、コマンドプロンプトで次のコマンドを入力します。

```
1 ktpass princ <SPN> ptype KRB5_NT_PRINCIPAL mapuser <DOMAIN><username>
     pass <password> -out <File_Path>
2 <!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **SPN**。KCD サービスアカウントのサービスプリンシパル名。
- ドメイン。Active Directory サーバのドメイン。
- **username**）を使用します。KSA アカウントのユーザー名。
- **password**。KSA アカウントのパスワード。
- 道。keytab ファイルが生成された後に保存されるディレクトリのフルパス名。

NetScaler 構成ユーティリティを使用してキータブファイルを生成するスクリプトを作成するには

1. セキュリティ **> AAA-**アプリケーショントラフィックに移動します。
2. データウィンドウの [**Kerberos** 制約付き委任] で、[バッチファイル] をクリックして Keytab を生成します。
3. 「**KCD (Kerberos** 制約付き委任**) Keytab** スクリプトの生成」ダイアログ・ボックスで、次のパラメータを設定します。
   - ドメインユーザー名。KSA アカウントのユーザー名。
   - ドメインパスワード。KSA アカウントのパスワード。
   - サービスプリンシパル KSA のサービスプリンシパル名。
   - 出力ファイル名。キータブファイルを AD サーバーに保存するフルパスとファイル名。
4. **[**ドメインユーザーアカウントの作成] チェックボックスをオフにします。
5. [スクリプトを生成] をクリックします。
6. Active Directory サーバにログオンし、コマンドラインウィンドウを開きます。
7. [生成されたスクリプト**]** ウィンドウからスクリプトをコピーし、Active Directory サーバのコマンドラインウィンドウに直接貼り付けます。キータブが生成され、「出力ファイル名」**(Output File Name)** として指定したファイル名のディレクトリに格納されます。

8. 任意のファイル転送ユーティリティを使用して、キータブファイルを Active Directory サーバーから NetScaler アプライアンスにコピーし、/nsconfig/krb ディレクトリに配置します。

KCD アカウントを作成するには　コマンドプロンプトで、次のコマンドを入力します。

```
add aaa kcdaccount <accountname>  -keytab <keytab>
<!--NeedCopy-->
```

例

kcdccount1 という名前の KCD アカウントを追加し、kcdvserver.keytab という名前のキータブを使用するには、次のコマンドを入力します。

```
add aaa kcdaccount kcdaccount1  -keytab kcdvserver.keytab
<!--NeedCopy-->
```

委任されたユーザー証明書を使用して委任によって **SSO** の **KCD** アカウントを作成するには　コマンドプロンプトで、次のコマンドを入力します。

```
add aaa kcdaccount <accountname> -realmStr <realm> -delegatedUser <
    user_nameSPN> -usercert <cert> -cacert <cacert>
<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- アカウント名。KCD アカウントの名前。
- **realmStr**。KCD アカウントのレルム。通常は SSO が設定されているドメインです。
- 委任されたユーザー。委任されたユーザー名 (SPN 形式)。
- **usercert**。NetScaler アプライアンス上の委任されたユーザー証明書ファイルのフルパスと名前。委任されたユーザー証明書には、クライアント証明書と秘密キーの両方が含まれていて、PEM 形式である必要があります。スマートカード認証を使用する場合は、秘密キーとともに証明書をインポートできるように、スマートカード証明書テンプレートを作成する必要があります。
- **cacert**。NetScaler アプライアンス上の CA 証明書ファイルのフルパスと名前。

例

kcdccount1 という名前の KCD アカウントを追加し、kcdvserver.keytab という名前のキータブを使用するには、次のコマンドを入力します。

```
add aaa kcdaccount kcdaccount1 -realmStr EXAMPLE.COM
     -delegatedUser "host/kcdvserver.example.com" -usercert /certs/
        usercert
     -cacert /cacerts/cacert
<!--NeedCopy-->
```


NetScaler SSO 用の **Active Directory** セットアップ

委任により SSO を構成する場合、NetScaler アプライアンスで KCD アカウントを作成するほかに、LDAP Active Directory サーバーで対応する Kerberos サービスアカウント（KSA）を作成し、サーバーを SSO 用に構成する必要があります。KSA を作成するには、Active Directory サーバでアカウント作成プロセスを使用します。Active Directory サーバで SSO を設定するには、KSA のプロパティウィンドウを開きます。[委任] タブで、[指定したサービスへの委任に対してのみこのユーザーを信頼する] と [任意の認証プロトコルを使用する] オプションを有効にします。(Kerberos only オプションは、プロトコルの移行や制約付き委任を有効にしないため、機能しません)。最後に、NetScaler SSO が管理するサービスを追加します。

> 注:
>
> [KSA アカウントのプロパティ] ダイアログボックスに [委任] タブが表示されない場合は、説明に従って KSA を構成する前に、Microsoft
> setspn コマンドラインツールを使用して、タブが表示されるように Active Directory サーバーを構成する必要があります。

Kerberos サービスアカウントの委任を構成するには

1. 作成した Kerberos サービスアカウントの [LDAP アカウント構成] ダイアログボックスで、[委任] タブをクリックします。
2. [指定したサービスへの委任に対してのみ、このユーザーを信頼する] を選択します。
3. [指定したサービスへの委任に対してのみこのユーザーを信頼する] で、[任意の認証プロトコルを使用する] を選択します。
4. [このアカウントが委任された認証情報を提示できるサービス] で、[追加**]** をクリックします。
5. ［サービスの追加］ダイアログボックスで、［ユーザー］または［コンピューター］をクリックし、サービスアカウントに割り当てるリソースをホストするサーバーを選択して、［**OK**］をクリックします。

   > 注:
   >
   > - 制約付き委任では、Kerberos が他のドメインと信頼関係を持っている場合でも、アカウントに割り当てられたドメイン以外のドメインでホストされているサービスはサポートされません。
   > - 新しいユーザが Active Directory に作成された場合は、次のコマンドを使用して`setspn`を作成します。`setspn -A host/kcdvserver.example.com example\kcdtest`

6. [サービスの追加] ダイアログボックスの [利用可能なサービス] リストに戻り、サービスアカウントに割り当てられているサービスを選択します。NetScaler SSO は、HTTP サービスと MSSQLSVC サービスをサポートします。
7. 「**OK**」をクリックします。

KCD が子ドメインをサポートできるようにするための設定変更

KCD アカウントが-delegatedUserでsamAccountNameに設定されている場合、子ドメインからサービスにアクセスするユーザーに対して KCD は機能しません。この場合、NetScaler アプライアンスと Active Directory の構成を変更できます。

- AD のサービスアカウント <service-account-samaccountname> (KCD アカウントで delegateUser として設定されている) のログオン名をhost/<service-account-samaccountname>.<completeUSERDNSDOMAIN>形式 (例: host/svc_act.child.parent.com) で変更します。

  サービスアカウントは手動で、またはktpassコマンドを使用して変更できます。ktpassは、サービスアカウントを自動的に更新します。

  ```
  ktpass /princ host/svc_act.child.parent.com@CHILD.PARENT.COM /
  ptype KRB5_NT_PRINCIPAL /mapuser CHILD\sv_act /pass serviceaccountpassword
   -out filepathfor.keytab
  ```

- NetScaler アプライアンスの KCD アカウントの委任ユーザーを変更します。
- KCD アカウントの-delegatedUser パラメータを次のように変更します。host/svc_act.child.parent.com

KCD アカウントの設定に高度な暗号化を使用する場合の注意点

> - **keytab** を使用する場合の設定例: add kcdaccount lbvs_keytab_aes256 -keytab "/nsconfig/krb/kcd2_aes256.keytab"
> - **keytab** に複数の暗号化タイプがある場合のサンプルコマンド。このコマンドは、ドメインユーザーパラメーターもキャプチャします。add kcdaccount lbvs_keytab_aes256 -keytab "/nsconfig/krb/kcd2_aes256.keytab"-domainUser "HTTP/lbvs.aaa.local”
> - ユーザー認証情報を使用する場合のサンプルコマンド: add kcdaccount kslb2_user -realmStr AAA.LOCAL -delegatedUser lbvs -kcdPassword <password>

ドメインユーザー情報

Kerberos SSO に高度な暗号化タイプを使用する場合は、正しいドメインユーザー情報が提供されていることを確認してください。ユーザーのログイン名に関する情報は、Active Directory から取得できます。

ケルベロス委任を使用して **NetScaler SSO** を構成する場合　委任を使用する Kerberos SSO で高度な暗号化タイプを使用する場合、add aaa kcdaccountコマンドのdelagatedUserパラメータはユーザーのサービスプリンシパル名 (SPN) でなければなりません。サービスプリンシパル名は大文字と小文字が区別されます。

ユーザーのサービスプリンシパル名を知るには、Active Directory ドメインコントローラーの`setspn -L <domain\user>`コマンドを使用します。例: `setspn -L EXAMPLE\username`

サービスプリンシパル名を設定するには、Active Directory ドメインコントローラーの`setspn`コマンドを使用します。キータブファイルを作成するには、Active Directory `ktpass` ドメインコントローラーのコマンドを使用します。その方法の例を以下に示します。

- セットソン:`setspn -S host/username.example.com EXAMPLE\username`
- キータブ: `ktpass /princ host/username.example.com@EXAMPLE.COM /ptype KRB5_NT_PRINCIPAL /mapuser EXAMPLE.COM\username /pass XXXX /crypto AES256-SHA1 -out <pathto.keytab.file>`

Active Directory で上記のアクションを実行したら、NetScaler `add kcdaccount`または`set kcdaccount` CLI コマンドを使用して、構成済みの SPN で KCD アカウントを更新します。

特定のユーザーアカウントの SPN を表示するには、**Active Directory** の [ユーザープロパティ] セクションに移動します。

ケルベロスインパーソネーションを使用して **NetScaler SSO** を構成する場合　偽装を使用する Kerberos SSO に高度な暗号化タイプを使用する場合は、SSO 認証情報がエンドユーザーの正しいサービスプリンシパル名とともに使用されていることを確認してください。エンドユーザーのログイン認証情報が Kerberos SSO で機能しない場合は、適切なユーザー表現を設定して SSO ユーザー名を設定します。

ユーザープリンシパル名が Active Directory 上のエンドユーザーの正しいサービスプリンシパル名である場合、次のようになります。

- エンドユーザーのログインに LDAP 認証を使用する場合は、`LDAPAction`コマンドの`ssoNameAttribute`パラメーターを使用して SSO ユーザー名を設定します。

  例: `set authentication ldapAction ldap_act -ssoNameAttribute userPrincipalName`

- ユーザーログインに他の認証方法を使用する場合は、`trafficAction`コマンドの`userExpression`パラメーターを使用してください。たとえば、ユーザー属性 1 にユーザープリンシパル名が保存されている場合、トラフィックアクションで`AAA.USER.ATTRIBUTE(1)`を使用できます。

  例:`add tm traffic action traf_act -userExpression AAA.USER.ATTRIBUTE(1)`

KCD keytab スクリプトを生成する

August 15, 2023

KCD Keytab Script ダイアログボックスはキータブスクリプトを生成し、そのスクリプトによって NetScaler での KCD の構成に必要なキータブファイルが生成されます。

設定ユーティリティを使用して **KCD** キータブスクリプトを生成するには

1. セキュリティ **> AAA-**アプリケーショントラフィックに移動します。
2. 詳細ペインの「**Kerberos** 制約付き委任」で、「バッチファイル」をクリックしてキータブを生成します。
3. 「KCD (Kerberos 制約付き委任) キータブ・スクリプトを生成」ダイアログ・ボックスで、以下の説明に従ってフィールドに入力します。

    - ドメインユーザー名**:** ドメインユーザーの名前。
    - ドメインパスワード**:** ドメインユーザーのパスワード。
    - サービスプリンシパル**:** サービスプリンシパル。
    - 出力ファイル名**:** KCD スクリプトファイルのファイル名。
    - ドメインユーザーアカウントの作成**:** このチェックボックスを選択すると、指定したドメインユーザーアカウントが作成されます。

4. [スクリプトを生成] をクリックしてスクリプトを生成します。スクリプトが生成され、「スクリプトを生成」ボタンの下の「****** 生成スクリプト ****** 」テキストボックスに表示されます。
5. スクリプトをコピーして、AD ドメインコントローラーにファイルとして保存します。次に、このスクリプトをドメインコントローラーで実行してキータブファイルを生成し、キータブファイルを NetScaler アプライアンスの/nsconfig/krb/ディレクトリにコピーする必要があります。
6. 「**OK**」をクリックします。

基本認証、ダイジェスト認証、**NTLM** 認証用の **SSO**

August 16, 2023

NetScaler と NetScaler Gateway のシングルサインオン（SSO）構成は、グローバルレベルでもトラフィックレベルでも有効にできます。デフォルトでは SSO 設定は **OFF** で、管理者はトラフィックごとに SSO を有効にすることも、グローバルに有効にすることもできます。セキュリティの観点から、**Citrix** では管理者に **SSO** をグローバルにオフにし、トラフィックごとに有効にすることを推奨しています。この機能強化は、特定の種類の SSO メソッドをグローバルに適用しないことで、SSO 構成をより安全にするためです。

> 注:
>
> NetScaler 機能リリース 13.0 ビルド 64.35 以降では、次の SSO タイプは世界中で不名誉になっています。
>
> - ベーシック認証

> - ダイジェストアクセス認証
> - ネゴシエート NTLM2 キーまたはネゴシエートサインなしの NTLM

影響を受けない **SSO** タイプ

次の SSO タイプは、この機能強化の影響を受けません。

- Kerberos 認証
- SAML 認証
- フォームベース認証
- OAuth ベアラ認証
- ネゴシエート NTLM2 キーまたはネゴシエートサイン付き NTLM

影響を受ける **SSO** 構成

影響を受ける（不適切な）SSO 構成は次のとおりです。

グローバル構成

```
set tmsessionparam -SSO ON
set vpnparameter -SSO ON
add tmsessionaction tm_act -SSO ON
add vpn sessionaction tm_act -SSO ON
<!--NeedCopy-->
```

トラフィックごとの設定

```
add vpn trafficaction tf_act http -SSO ON
add tm trafficaction tf_act -SSO ON
<!--NeedCopy-->
```

> あなたはできる enable/disable SSO 全体であり、個々の SSO タイプを変更することはできません。

適用すべきセキュリティ対策

セキュリティ対策の一環として、セキュリティに敏感な SSO タイプはグローバル設定では無視されますが、トラフィックアクション設定でのみ許可されます。
そのため、バックエンドサーバーがネゴシエート NTLM2 キーやネゴシエートサインなしでベーシック、ダイジェスト、または NTLM を想定している場合、管理者は次の構成でのみ SSO を許可できます。

トラフィックアクション

```
add vpn trafficaction tf_act  http -SSO ON
add tm trafficaction tf_act -SSO ON
<!--NeedCopy-->
```

交通政策

```
add tm trafficpolicy <name> <rule> tf_act
add vpn trafficpolicy <name> <rule> tf-act
<!--NeedCopy-->
```

> 信頼できるバックエンドサーバーでのみ SSO が有効になるように、管理者はトラフィックポリシーに適切なルールを設定する必要があります。

AAA-TM

グローバル構成に基づくシナリオ**:**

```
set tmsessionparam -SSO ON
<!--NeedCopy-->
```

回避策:

```
add tm trafficaction tf_act -SSO ON
add tm trafficpolicy tf_pol true tf_act
<!--NeedCopy-->
```

SSO が必要なすべての **LB** 仮想サーバーに次のトラフィックポリシーをバインドします。

```
bind lb vserver <LB VS Name> -policy tf_pol -priority 65345
<!--NeedCopy-->
```

セッションポリシー構成に基づくシナリオ:

```
add tmsessionaction tm_act -SSO ON
add tmsession policy <name> <rule> tm_act
add tm trafficaction tf_act -SSO ON
add tm trafficpolicy tf_pol <same rule as session Policy> tf_act
<!--NeedCopy-->
```

注意点**:**

- 前述のセッションポリシーの NetScaler AAA ユーザー/グループは、トラフィックポリシーに置き換える必要があります。
- 次のポリシーを、前のセッションポリシーの負荷分散仮想サーバーにバインドします。


```
bind lb vserver [LB VS Name] -policy tf_pol -priority 65345
<!--NeedCopy-->
```

- 他の優先順位のトラフィックポリシーが設定されている場合、前述のコマンドは正しく機能しません。

次のセクションでは、トラフィックに関連する複数のトラフィックポリシーとの競合に基づくシナリオについて説明します。

特定の TM トラフィックには、1 つの TM トラフィックポリシーだけが適用されます。SSO 機能変更のグローバル設定のため、優先度が高い（SSO 設定が必要ない）TM トラフィックポリシーがすでに適用されている場合は、優先度の低い TM トラフィックポリシーを追加で適用できない場合があります。次のセクションでは、このようなケースを確実に処理する方法について説明します。

負荷分散（**LB**）仮想サーバーには、優先順位の高い次の **3** つのトラフィックポリシーが適用されているとします。

```
add tm trafficaction tf_act1 <Addition config>
add tm trafficaction tf_act2 <Addition config>
add tm trafficaction tf_act3 <Addition config>

add tm trafficpolicy tf_pol1 <rule1> tf_act1
add tm trafficpolicy tf_pol2 <rule2> tf_act2
add tm trafficpolicy tf_pol3 <rule3> tf_act3

bind lb vserver <LB VS Name> -policy tf_pol1 -priority 100
bind lb vserver <LB VS Name> -policy tf_pol2 -priority 200
bind lb vserver <LB VS Name> -policy tf_pol3 -priority 300
<!--NeedCopy-->
```

エラーが発生しやすい方法**-**グローバル **SSO** 構成を解決するには、次の構成を追加します。

```
add tm trafficaction tf_act_default -SSO ON
add tm trafficpolicy tf_pol_default true tf_act_default

bind lb vserver <LB VS Name> -policy tf_pol_default -priority 65345
<!--NeedCopy-->
```

注**:** 上記の変更により、<tf_pol1/tf_pol2/tf_pol3> トラフィックポリシーとしてヒットするトラフィックの SSO が壊れる可能性があります。< tf_pol_default> は適用されません。

正しい方法**-**これを軽減するには、対応するトラフィックアクションごとに **SSO** プロパティを個別に適用する必要があります。

たとえば、前述のシナリオでは、tf_pol1/tf_pol3 に到達するトラフィックに対して SSO が発生するには、次の設定を < tf_pol_default>.

```
add tm trafficaction tf_act1 <Addition config> -SSO ON
add tm trafficaction tf_act3 <Addition config> -SSO ON
<!--NeedCopy-->
```


NetScaler Gateway ケース

グローバル構成に基づくシナリオ**:**

```
set vpnparameter -SSO ON
<!--NeedCopy-->
```

回避策:

```
add vpn trafficaction vpn_tf_act http  -SSO ON
add vpn trafficpolicy vpn_tf_pol  true vpn_tf_act
bind the following traffic policy to all VPN virtual server where SSO
    is expected:
bind vpn vserver vpn_vs -policy vpn_tf_pol -priority 65345
<!--NeedCopy-->
```

セッションポリシー構成に基づくシナリオ:

```
add vpn sessionaction vpn_sess_act -SSO ON
add vpnsession policy <name> <rule> vpn_sess_act
<!--NeedCopy-->
```

注意事項:

- 前述のセッションポリシーの NetScaler AAA ユーザー/グループは、トラフィックポリシーに置き換える必要があります。
- 次のポリシーを、前のセッションポリシー`bind lb virtual server [LB VS Name] -policy tf_pol -priority 65345`の LB 仮想サーバーにバインドします。
- 他の優先順位のトラフィックポリシーが設定されている場合、前述のコマンドは正しく機能しません。次のセクションでは、トラフィックに関連する複数のトラフィックポリシーとの競合に基づくシナリオについて説明します。

トラフィックに関連する複数のトラフィックポリシーとの競合に基づく機能シナリオ:

特定の NetScaler Gateway トラフィックには、1 つの VPN トラフィックポリシーのみが適用されます。SSO 機能変更のグローバル設定のため、必要な SSO 設定がない優先度の高い VPN トラフィックポリシーが他にある場合は、優先度の低い VPN トラフィックポリシーを追加して適用できない場合があります。

次のセクションでは、このようなケースを確実に処理する方法について説明します。

VPN 仮想サーバには、優先順位の高いトラフィックポリシーが 3 つ適用されているとします。

```
add vpn trafficaction tf_act1 <Addition config>
add vpn trafficaction tf_act2 <Addition config>
add vpn trafficaction tf_act3 <Addition config>

add vpn trafficpolicy tf_pol1 <rule1> tf_act1
add vpn trafficpolicy tf_pol2 <rule2> tf_act2
add vpn trafficpolicy tf_pol3 <rule3> tf_act3
```



```
8
9  bind vpn vserver <VPN VS Name> -policy tf_pol1 -priority 100
10 bind vpn vserver <VPN VS Name> -policy tf_pol2 -priority 200
11 bind vpn vserver <VPN VS Name> -policy tf_pol3 -priority 300
12 <!--NeedCopy-->
```

エラーが発生しやすい方法：グローバル SSO 構成を解決するには、次の構成を追加します。

```
1 add vpn trafficaction tf_act_default -SSO ON
2 add vpn trafficpolicy tf_pol_default true tf_act_default
3
4 bind vpn vserver <VPN VS Name> -policy tf_pol_default -priority 65345
5 <!--NeedCopy-->
```

注**:** 上記の変更により、<tf_pol1/tf_pol2/tf_pol3> トラフィックポリシーとしてヒットするトラフィックの SSO が壊れる可能性があります。< tf_pol_default> は適用されません。

正しい方法：これを軽減するには、対応するトラフィックアクションごとに SSO プロパティを個別に適用する必要があります。

たとえば、前述のシナリオでは、tf_pol1/tf_pol3 に到達するトラフィックに対して SSO が発生するには、次の設定を < tf_pol_default>.

```
1 add vpn trafficaction tf_act1 [Additional config] -SSO ON
2
3 add vpn trafficaction tf_act3 [Additional config] -SSO ON
4 <!--NeedCopy-->
```

NetScaler Gateway と認証サーバーが生成した応答の書き換え

December 8, 2023

書き換えとは、NetScaler アプライアンスが処理する要求または応答の情報を書き換えることです。書き換えは、ウェブサイトの実際の設定に関する不必要な詳細を公開することなく、要求されたコンテンツへのアクセスを提供するのに役立ちます。書き換えの概念の詳細については、「書き換え」を参照してください。

NetScaler リリースビルド 13.0-76.29 以降、書き換えポリシーのサポートは NetScaler Gateway 仮想サーバーと認証仮想サーバーが生成する応答にまで拡張されました。

> 注
>
> NetScaler Gateway 仮想サーバーと認証仮想サーバーが生成した応答の書き換えポリシーをサポートするために、AAA_RESPONSE バインドタイプが導入されました。認証仮想サーバーまたは NetScaler Gateway 仮想サーバーによって生成された応答がパケットエンジンからのものである場合は、AAA_RESPONSE バインドタイプを構成します。

書き換えを使用する例

リライトを使用すると、オンプレミスの NetScaler で使用可能なリソースを Citrix Cloud 環境と共有できます。これは、CORS オリジンリソース共有を実装することで安全に実現できます。書き換えは、CORS ヘッダーを実装するために以下のように使用することができる。

設定例

```
add rewrite action cors_header_action insert_http_header access-control-allow-credentials \"true\"

add rewrite policy cors_header_pol true cors_header_action

add rewrite action non_cors_header_action insert_http_header X-Frame-Options \'\"DENY\"\'

add rewrite policy non_cors_header_pol true non_cors_header_action

bind authentication vserver av_cors -policy cors_header_pol -priority 100 -type AAA_RESPONSE

bind vpn vserver av_cors -policy cors_header_pol -priority 100 -type AAA_RESPONSE
```

> 注:
>
> GUI を使用して書き換えアクションとポリシーを設定する方法については、「書き換え」を参照してください。

NetScaler Gateway および認証仮想サーバーが生成する応答に対するコンテンツセキュリティポリシー応答ヘッダーのサポート

April 15, 2024

NetScaler リリースビルド 13.0～76.29 以降、NetScaler Gateway および認証仮想サーバーが生成する応答では、コンテンツセキュリティポリシー（CSP）応答ヘッダーがサポートされます。

Content-Security-Policy（CSP）レスポンスヘッダーは、クロスサイトスクリプティング（CSS）攻撃を回避するためにブラウザが使用するポリシーの組み合わせです。
HTTP CSP レスポンスヘッダーを使用すると、Web サイト管理者は、ユーザーエージェントが特定のページに対してロードできるリソースを制御できます。いくつかの例外を除いて、ポリシーには主にサーバーオリジンとスクリプトエンドポイントの指定が含まれます。これにより、クロスサイトスクリプティング攻撃から保護できます。
CSP ヘッダーは、ブラウザーがページをレンダリングする方法を変更し、CSS を含むさまざまなクロスサイトインジェクションから保護するように設計されています。ウェブサイトの適切な操作を妨げないように、ヘッダー値を正し

く設定することが重要です。たとえば、ヘッダーがインライン JavaScript の実行を妨げるように設定されている場合、Web サイトはそのページでインライン JavaScript を使用してはいけません。

CSP レスポンスヘッダーの利点は次のとおりです。

- CSP 応答ヘッダーの主な機能は、CSS 攻撃を防ぐことです。
- サーバーでは、コンテンツのロード元となるドメインを制限することに加えて、どのプロトコルを使用できるかを指定できます。たとえば（理想的にはセキュリティの観点から）、サーバーはすべてのコンテンツを HTTPS を使用してロードする必要があることを指定できます。
- CSP は、「tmindex.html」や「homepage.html」などのファイルを保護することで、NetScaler をクロスサイトスクリプティング攻撃から保護するのに役立ちます。ファイル“tmindex.html”は認証に関連しており、ファイル“homepage.html”は公開されたアプリ/リンクに関連しています。

NetScaler Gateway および認証仮想サーバーが生成する応答用のコンテンツセキュリティポリシーヘッダーの構成

CSP ヘッダーを有効にするには、CSP HTTP ヘッダーを返すように Web サーバーを設定する必要があります。

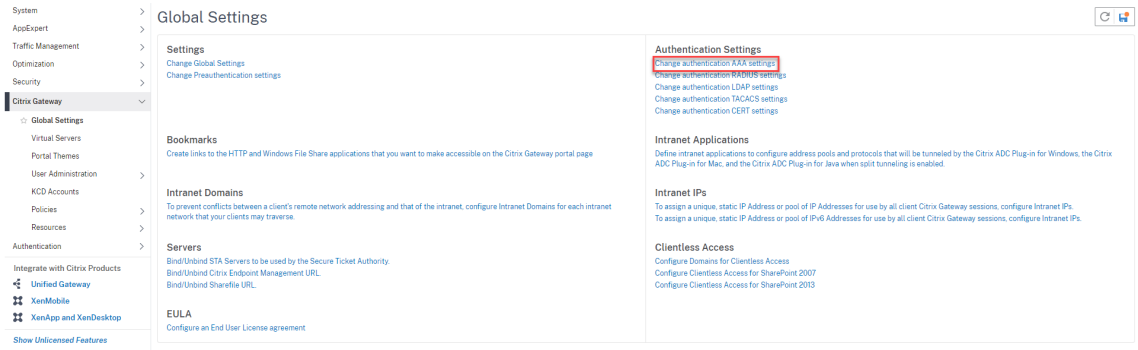
> 注意事項
>
> - デフォルトでは、CSP ヘッダーは無効になっています。
> - デフォルトの CSP ポリシーを有効または無効にする場合は、次のコマンドを実行することをお勧めします。`Flush cache contentgroup loginstaticobjects`
> - /logon/LogonPoint/index.html の CSP を変更するには、ディレクトリ`/var/netscaler/logon`の下にあるログオンディレクトリに対応するセクションで、必要に応じて「ヘッダーセット Content-Security-Policy」値を変更します。
> - GUI を使用して書き換えアクションとポリシーを設定する方法については、「書き換え」を参照してください。

CLI を使用して仮想サーバーと NetScaler Gateway が生成する応答を認証するように CSP を構成するには、コマンドプロンプトで次のコマンドを入力します。

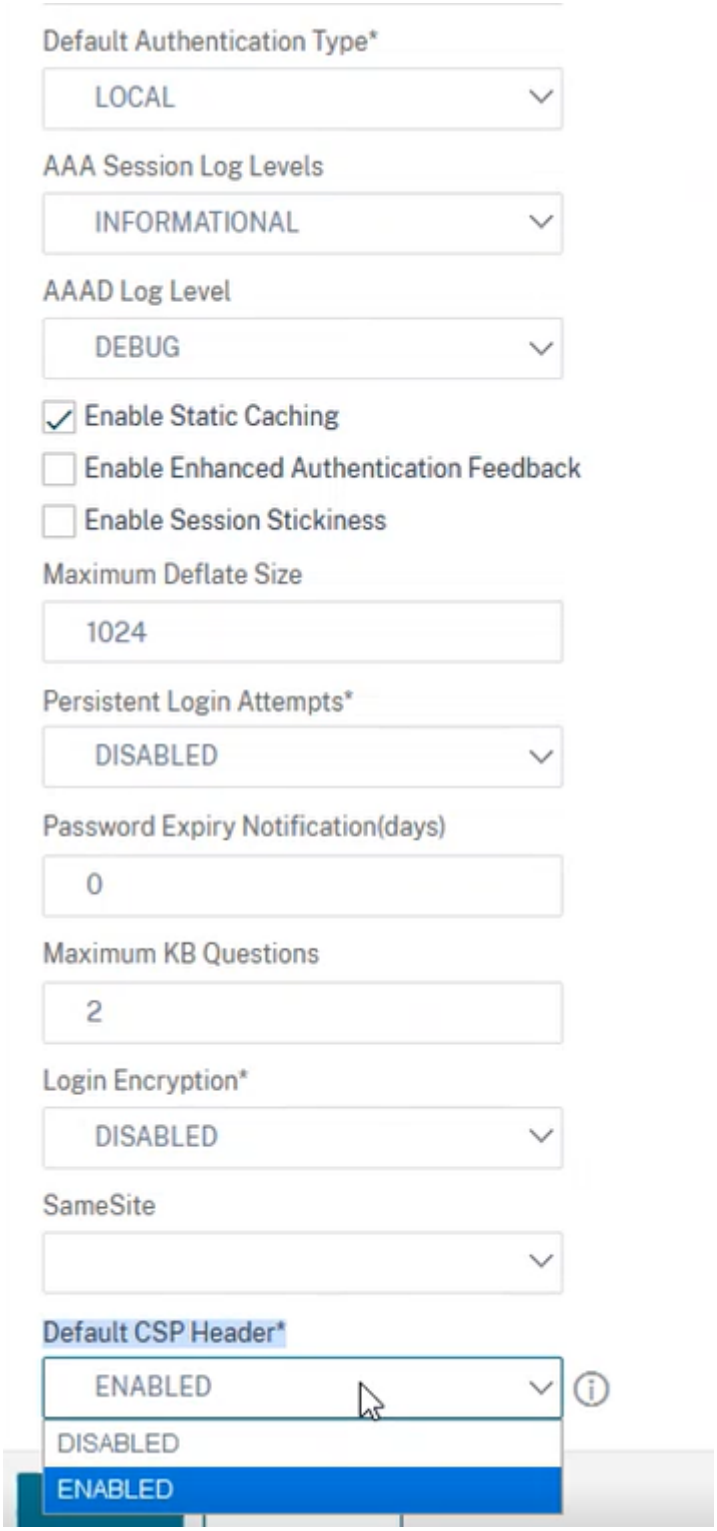
```
1 set aaa parameter -defaultCSPHeader <ENABLE/DISABLE>
```

NetScaler Gateway 用の CSP を構成し、GUI を使用して仮想サーバーが生成した応答を認証すること。

1. **NetScaler Gateway >** グローバル設定に移動し、［認証設定］の下の［認証 **AAA** 設定の変更］をクリックします。



2. [**AAA** パラメータの設定] ページで、[デフォルト **CSP** ヘッダーで有効] フィールドを選択します。



カスタムコンテンツ-セキュリティ-ポリシーヘッダー

AAA エンドポイントが生成する応答の VPN 仮想サーバと認証仮想サーバのリライトポリシーを使用してカスタム CSP ヘッダーを設定できます。

以下は、次の 2 つの指定されたソースからのイメージとスクリプトのみをそれぞれ含めるための CSP ヘッダーのカスタマイズの例です。https://company.fqdn.com、https://example.com。

```
add rewrite action modify_csp insert_http_header Content-Security-
    Policy ""default-src 'self'; script-src 'self' https://company.fqdn.
    com 'unsafe-inline' 'unsafe-eval'; connect-src 'self'; img-src http
    ://localhost:* https://example.com 'self' data: http: https:; style-
    src 'self' 'unsafe-inline'; font-src 'self'; frame-src 'self'; child
    -src 'self' com.citrix.agmacepa://* citrixng://* com.citrix.
    nsgclient://*; form-action 'self'; object-src 'self'; report-uri /
    nscsp_violation/report_uri""

add rewrite policy csp_pol true modify_csp
<!--NeedCopy-->
```

ポリシーを認証仮想サーバーにバインドするには、

```
bind authentication vserver auth_vs -policy csp_pol -priority 1 -type
    AAA_RESPONSE

bind authentication vserver auth_vs -policy csp_pol -priority 2 -type
    RESPONSE
<!--NeedCopy-->
```

ポリシーを VPN 仮想サーバーにバインドするには、

```
bind vpn vserver vpn_vs -policy csp_pol -priority 1 -type AAA_RESPONSE

bind vpn vserver vpn_vs -policy csp_pol -priority 2 -type RESPONSE
<!--NeedCopy-->
```

> 注:
>
> 特定の AAA 生成応答では、コンテンツスイッチングおよび負荷分散仮想サーバでカスタム CSP ヘッダーを設定できません。

セルフサービスパスワードリセット

December 8, 2023

セルフサービスパスワードリセットは、Web ベースのパスワード管理ソリューションです。NetScaler アプライアンスと NetScaler Gateway の認証、承認、および監査機能の両方で使用できます。これにより、ユーザーはパスワ

ードの変更に対する管理者の支援に依存する必要がなくなります。

セルフサービスパスワードリセットにより、エンドユーザは次のシナリオでパスワードを安全にリセットまたは作成できます。

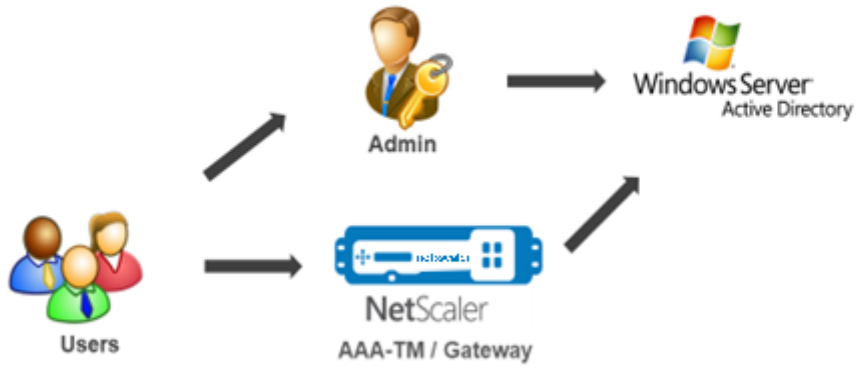
- ユーザーがパスワードを忘れた。
- ユーザーはログオンできません。

これまでは、エンドユーザーが AD パスワードを忘れた場合、エンドユーザーは AD 管理者に連絡してパスワードをリセットする必要がありました。セルフサービスパスワードリセット機能により、エンドユーザーは管理者の介入なしにパスワードをリセットできます。

セルフサービスパスワードリセットを使用する利点には、次のようなものがあります。

- パスワードの自動変更メカニズムにより生産性が向上し、ユーザーがパスワードのリセットを待つ時間がなくなります。
- パスワードの自動変更メカニズムにより、管理者は他の重要なタスクに集中できます。

次の図は、パスワードをリセットするセルフサービスパスワードリセットフローを示しています。



セルフサービスパスワードリセットを使用するには、ユーザーを NetScaler 認証、承認、監査、または NetScaler Gateway 仮想サーバーのいずれかに登録する必要があります。

セルフサービスパスワードリセットには、次の機能があります。

- 新規ユーザーの自己登録。新規ユーザーとして自己登録できます。
- ナレッジベースの質問を構成します。管理者は、ユーザに対して一連の質問を設定できます。
- 代替電子メール **ID** 登録。登録時に別の電子メール ID を指定する必要があります。ユーザーがプライマリ電子メール ID のパスワードを忘れたため、OTP は代替電子メール ID に送信されます。

> 注:
>
> バージョン 12.1 ビルド 51.xx から、代替電子メール ID の登録をスタンドアロンとして行うことができます。代替の電子メール ID 登録のみを行うために、新しいログインスキーマ **AltEmailRegister.xml** が導入されました。以前は、代替電子メール ID の登録は、KBA 登録の実行中にのみ実行できました。

- 忘れたパスワードをリセットします。ユーザーはナレッジベースの質問に答えることでパスワードをリセットできます。管理者は、質問を設定して保存できます。

セルフサービスパスワードリセットでは、次の 2 つの新しい認証メカニズムが提供されます。

- 知識ベースの質問と回答。ナレッジベースの質問と回答のスキーマを選択する前に、NetScaler の認証、承認、監査、または NetScaler Gateway に登録する必要があります。
- 電子メール **OTP** 認証。OTP は、ユーザーがセルフサービスパスワードリセット登録時に登録した代替電子メール ID に送信されます。

> 注
>
> これらの認証メカニズムは、セルフサービスパスワードリセットのユースケースや、既存の認証メカニズムと同様の認証目的に使用できます。

前提条件

セルフサービスパスワードリセットを構成する前に、次の前提条件を確認してください。

- NetScaler 機能リリース 12.1、ビルド 50.28
- サポートされているバージョンは 2016、2012、および 2008 の AD ドメイン機能レベルです。
- NetScaler にバインドされた LDAPBind ユーザー名には、ユーザーの AD パスへの書き込みアクセス権が必要です。

  > 注
  >
  > セルフサービスパスワードリセットは、nFactor 認証フローでのみサポートされます。詳細については、「NetScaler による nFactor 認証」を参照してください。

制限事項

セルフサービスパスワードのリセットには、次のような制限があります。

- セルフサービスパスワードリセットは、LDAPS でサポートされています。セルフサービスパスワードリセットは、認証バックエンドが LDAP (LDAP プロトコル) の場合にのみ使用できます。
- 登録済みの代替電子メール ID がユーザーに表示されることはありません。
- ナレッジベースの質疑応答、電子メール OTP の認証と登録は、認証フローの最初の要素にはなりません。

- ネイティブプラグインおよび Receiver では、登録はブラウザ経由でのみサポートされます。
- セルフサービスパスワードのリセットに使用される証明書の最小サイズは 1024 バイトで、x.509 標準に従う必要があります。
- セルフサービスパスワードリセットでは RSA 証明書のみがサポートされています。

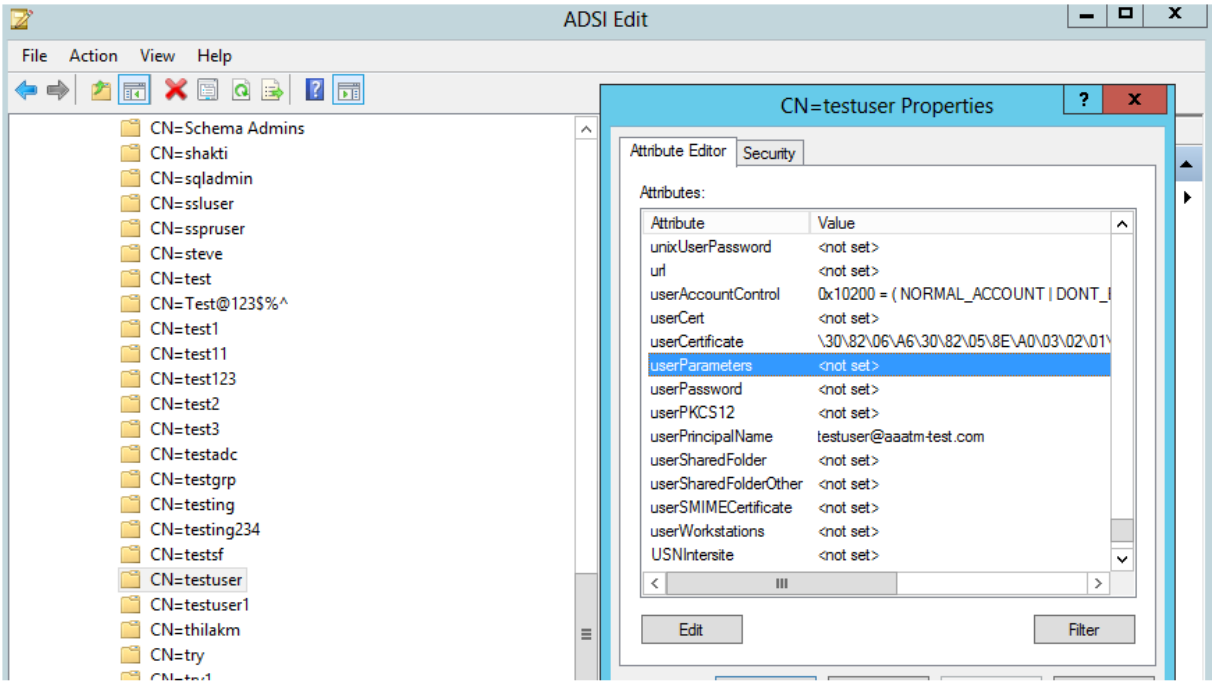
アクティブディレクトリ設定

NetScaler のナレッジベースの質問と回答、および電子メール OTP は、AD 属性を使用してユーザーデータを保存します。質問と回答を代替電子メール ID とともに保存するように AD 属性を設定する必要があります。NetScaler アプライアンスは、それを AD ユーザーオブジェクトに設定された KB 属性に保存します。AD 属性を設定する場合は、次の点を考慮してください。

- AD 属性は最大 32K の値をサポートする必要があります。
- 属性タイプは‘DirectoryString’でなければなりません。
- 1 つの AD 属性を、ナレッジベースの質問と回答、および代替電子メール ID に使用できます。
- ネイティブ OTP およびナレッジベースの質問と回答、または代替電子メール ID の登録には、単一の AD 属性を使用できません。
- NetScaler LDAP 管理者は、選択した AD 属性への書き込みアクセス権を持っている必要があります。

既存の AD 属性を使用することもできます。ただし、使用する予定の属性が他のケースでは使用されないようにしてください。たとえば、userParameters は AD ユーザー内の既存の属性であり、これを使用できます。この属性を確認するには、次の手順を実行します。

1. [**ADSI] > [**ユーザーの選択**]** に移動します。
2. 右クリックして、属性リストまでスクロールダウンします。
3. **cn=TestUser** プロパティウィンドウペインで、**UserParameters** 属性が設定されていないことを確認できます。



セルフサービスパスワードリセット登録

セルフサービスパスワードリセットソリューションを NetScaler アプライアンスに実装するには、以下を実行する必要があります。

- セルフサービスパスワードリセット (ナレッジベースの質問と回答/電子メール ID) 登録
- ユーザーログオンページ (パスワードのリセット用。ナレッジベースの質問と回答、電子メールの OTP 検証、最終的なパスワードリセット係数が含まれます)。

定義済みの質問カタログのセットが JSON ファイルとして提供されます。管理者は、NetScaler GUI を使用して質問を選択し、セルフサービスパスワードリセット登録ログインスキーマを作成できます。次のオプションのいずれかを選択できます。

- システム定義の質問を最大 4 つ選択します。
- ユーザーが 2 つの質問と回答をカスタマイズできるオプションを提供します。

CLI からデフォルトのナレッジベースの質問 **JSON** ファイルを表示するには

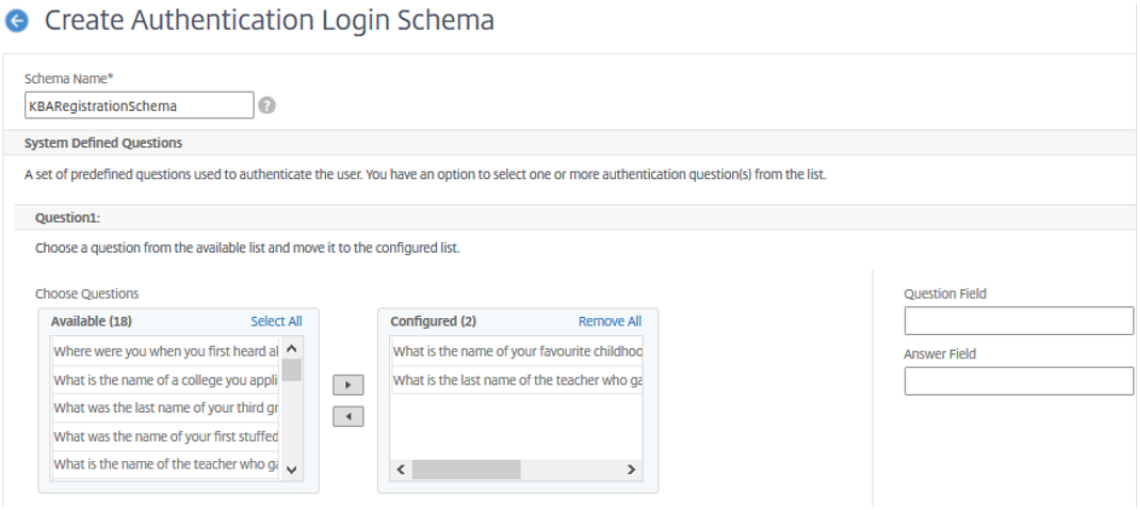
```
root@ns# cd /nsconfig/loginschema/LoginSchema/
root@ns# cat KBQuestions.json
[
    {"question":"What is the last name of the teacher who gave you your first failing
grade?"},
    {"question":"What is the name of your favourite childhood friend?"},
    {"question":"Where were you when you first heard about 9/11?"},
    {"question":"What is the name of a college you applied to but didn't attend?"},
    {"question":"What was the last name of your third grade teacher?"},
    {"question":"What was the name of your first stuffed animal?"},
    {"question":"What is the name of the teacher who gave you your first A?"},
    {"question":"What is the name of the city where you got lost?"},
    {"question":"In what city or town did your mother and father meet?"},
    {"question":"What was your most hated food as a child?"},
    {"question":"What was your most favourite food as a child?"},
    {"question":"What is your favourite website?"},
    {"question":"What is your most disliked website?"},
    {"question":"What is your dream job?"},
    {"question":"Why did the chicken cross the road?"},
    {"question":"Name your first boss."},
    {"question":"What is the name of your favorite school teacher?"},
    {"question":"What is the name of your favorite actor or actress?"},
    {"question":"What is the title of your favorite movie?"},
    {"question":"In what city or town did you spend most of your youth?"}
]
```

注

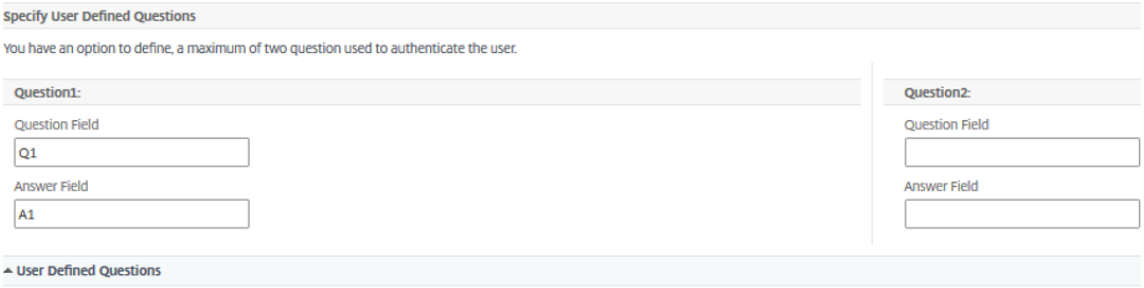
- NetScaler Gateway には、デフォルトでシステム定義の質問セットが含まれています。管理者は「KbQuestions.json」ファイルを編集して、選択した質問を含めることができます。
- システム定義の質問は英語でのみ表示され、これらの質問では言語ローカリゼーションのサポートは利用できません。

GUI を使用してナレッジベースの質問と回答の登録ログインスキーマを完了するには

1. [セキュリティ] **> [AAA** ―アプリケーショントラフィック] **> [**ログインスキーマ] に移動します。
2. [ログインスキーマ] ページで、[プロファイル] をクリックします。
3. [**KBA** 登録ログインスキーマの追加] をクリックします。
4. **[**認証ログインスキーマの作成] ページで、[スキーマ名] フィールドに名前を指定します。



5. 任意の質問を選択し、[設定済み] リストに移動します。
6. [ユーザ定義の質問（**User Defined** Questions）] セクションでは、Q1 および A1 フィールドに質問と回答を入力できます。
7. [メール登録] セクションで、[代替メールの登録] オプションをオンにします。OTP を受け取るには、ユーザー登録ログオンページから代替電子メール **ID** を登録します。



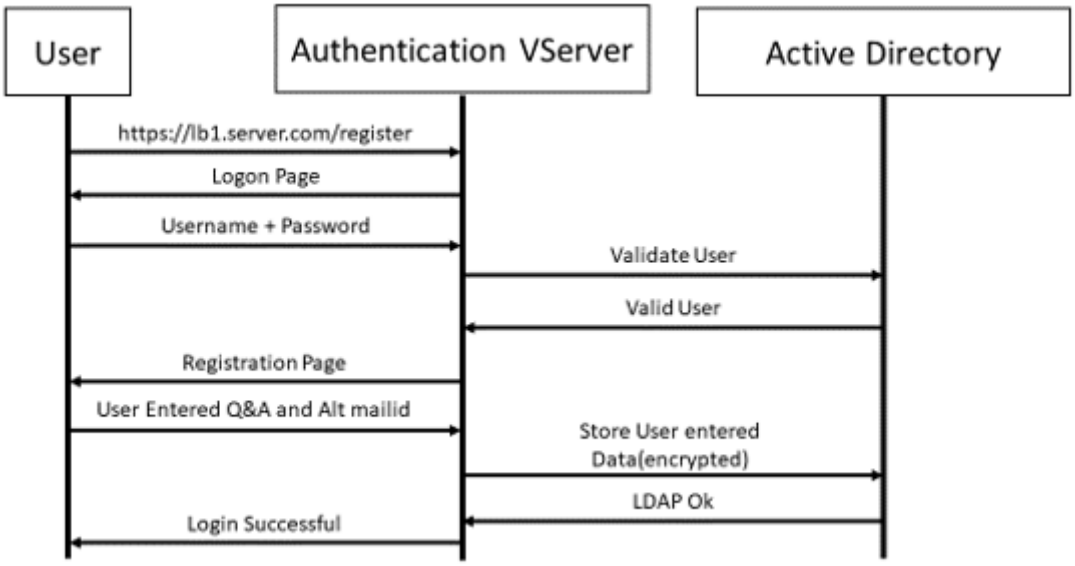
8. [作成] をクリックします。ログインスキーマが生成されると、登録プロセス中に設定されたすべての質問がエンドユーザに表示されます。

CLI を使用してユーザー登録と管理のワークフローを作成する

設定を開始する前に、次のことが必要です。

- 認証仮想サーバに割り当てられた IP アドレス
- 割り当てられた IP アドレスに対応する FQDN
- 認証仮想サーバーのサーバー証明書

デバイスの登録と管理ページを設定するには、認証仮想サーバーが必要です。次の図に、ユーザー登録を示します。



認証仮想サーバーを作成するには

1. 認証仮想サーバを設定します。SSL タイプで、認証仮想サーバーをポータル・テーマにバインドする必要があります。

```
> add authentication vserver <vServerName> SSL <ipaddress> <port>
> bind authentication vserver <vServerName> [-portaltheme<string>]
```

2. SSL 仮想サーバ証明書とキーのペアをバインドします。

```
> bind ssl vserver <vServerName> certkeyName <string>
```

例:

```
> add authentication vserver authvs SSL 1.2.3.4 443
> bind authentication vserver authvs -portaltheme RFWebUI
> bind ssl vserver authvs -certkeyname c1
```

LDAP ログオンアクションを作成するには

```
    > add authentication ldapAction <name> {
-serverIP <ipaddr|ipv6\_addr|> \[-serverPort <port>] \[-ldapBase <BASE
    > ] \[-ldapBindDn <AD USER>] \[-ldapBindDnPassword <PASSWORD>] \[-
    ldapLoginName <USER FORMAT>]
```

> 注
>
> 第 1 要素として任意の認証ポリシーを設定できます。

例:


```
> add authentication ldapAction ldap_logon_action -serverIP 1.2.3.4
    -serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -
   ldapBindDn administrator@ctxnsdev.com -ldapBindDnPassword
   PASSWORD -ldapLoginName samAccountName -serverport 636 -sectype
   SSL -KBAttribute userParameters
```

LDAP ログオン用の認証ポリシーを作成するには

```
> add authentication policy <name> <rule> [<reqAction]
```

例:

```
> add authentication policy ldap_logon -rule true -action
   ldap_logon_action
```

ナレッジベースの質問と回答の登録アクションを作成するには

ldapActionに 2 つの新しいパラメータが導入されました。KBA 認証 (登録と検証) 用のKBAttributeおよびユーザーの代替電子メール ID の登録用のalternateEmailAttr。

```
> add authentication ldapAction <name> {
-serverIP  <ipaddr|ipv6\_addr|> \[-serverPort <port>] \[-ldapBase <
   BASE> ] \[-ldapBindDn <AD USER>] \[-ldapBindDnPassword <PASSWORD>]
   \[-ldapLoginName <USER FORMAT>] \[-KBAttribute <LDAP ATTRIBUTE>]
   \[-alternateEmailAttr <LDAP ATTRIBUTE>]
```

例:

```
> add authentication ldapAction ldap1 -serverIP 1.2.3.4 -sectype
   ssl -serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -
   ldapBindDn administrator@ctxnsdev.com -ldapBindDnPassword
   PASSWORD -ldapLoginName samAccountName -KBAttribute
   userParameters -alternateEmailAttr userParameters
```

ユーザー登録と管理画面を表示する

「KBARegistrationSchema.xml」ログインスキーマは、エンドユーザーにユーザー登録ページを表示するために使用されます。次の CLI を使用して、ログインスキーマを表示します。

```
> add authentication loginSchema <name> -authenticationSchema <string>
```

例:

```
> add authentication loginSchema kba_register -authenticationSchema /
   nsconfig/loginschema/LoginSchema/KBARegistrationSchema.xml
```


ユーザー登録と管理画面の表示には、URL と LDAP 属性の 2 通りの表示方法が推奨されます。

URL を使う

URL パスに‘/register’(https://lb1.server.com/registerなど) が含まれている場合、URL を使用してユーザー登録ページが表示されます。

登録ポリシーを作成してバインドするには

```
> add authentication policylabel user_registration -loginSchema
    kba_register
> add authentication policy ldap1 -rule true -action ldap1
> bind authentication policylabel user_registration -policy ldap1 -
    priority 1
```

URL に‘**/register**’が含まれている場合に、認証ポリシーを認証、承認、監査仮想サーバーにバインドするには

```
> add authentication policy ldap_logon -rule "http.req.cookie.value(\"
    NSC_TASS\").contains(\"register\")" -action ldap_logon
> bind authentication vserver authvs -policy ldap_logon -nextfactor
    user_registration -priority 1
```

証明書を **VPN** グローバルにバインドするには

```
bind vpn global -userDataEncryptionKey c1
```

> 注
>
> - AD 属性に保存されているユーザーデータ (KB Q&A および登録済みの代替電子メール ID) を暗号化するには、証明書をバインドする必要があります。
> - 証明書の有効期限が切れた場合は、新しい証明書をバインドして、登録をもう一度実行する必要があります。

属性の使用

認証ポリシーを認証、承認、および監査仮想サーバーにバインドして、ユーザーがすでに登録されているかどうかを確認できます。このフローでは、ナレッジベースの質問と回答の登録係数より前の前述のポリシーは、KBA 属性が設定された LDAP である必要があります。これは、AD ユーザーが登録されているか、AD 属性を使用していないかを確認するためです。

> 重要
>
> ルール” AAA.USER.ATTRIBUTE(“kba_registered”).EQ(“0”)” は、新しいユーザにナレッジベースの質問と回答、および代替の電子メールへの登録を強制します。

ユーザーがまだ登録されていないかどうかを確認する認証ポリシーを作成するには

```
> add authentication policy switch_to_kba_register -rule "AAA.USER.
    ATTRIBUTE(\"kba_registered\").EQ(\"0\")" -action NO_AUTHN
> add authentication policy first_time_login_forced_kba_registration -
    rule true -action ldap1
```

登録ポリシーラベルを作成し、**LDAP** 登録ポリシーにバインドするには

```
> add authentication policylabel auth_or_switch_register -loginSchema
    LSCHEMA_INT
> add authentication policylabel kba_registration -loginSchema
    kba_register

> bind authentication policylabel auth_or_switch_register -policy
    switch_to_kba_register -priority 1 -nextFactor kba_registration
> bind authentication policylabel kba_registration -policy
    first_time_login_forced_kba_registration -priority 1
```

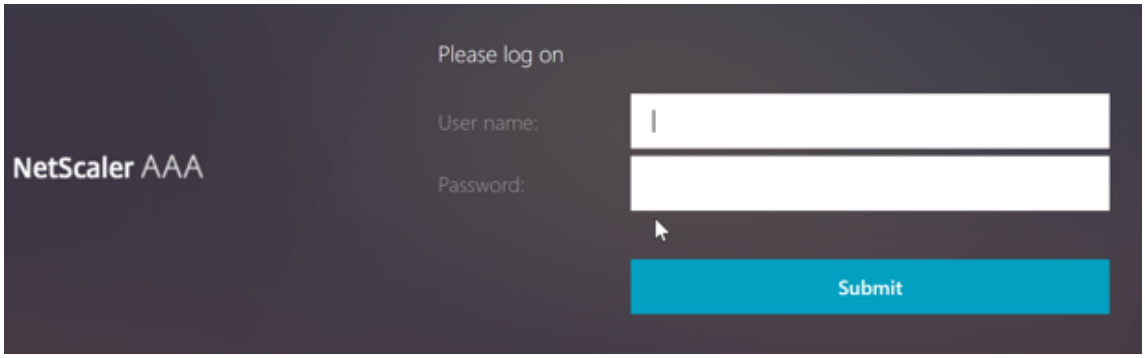
認証ポリシーを認証、承認、監査仮想サーバーにバインドするには

```
bind authentication vserver authvs -policy ldap_logon -nextfactor
    auth_or_switch_register -priority 2
```

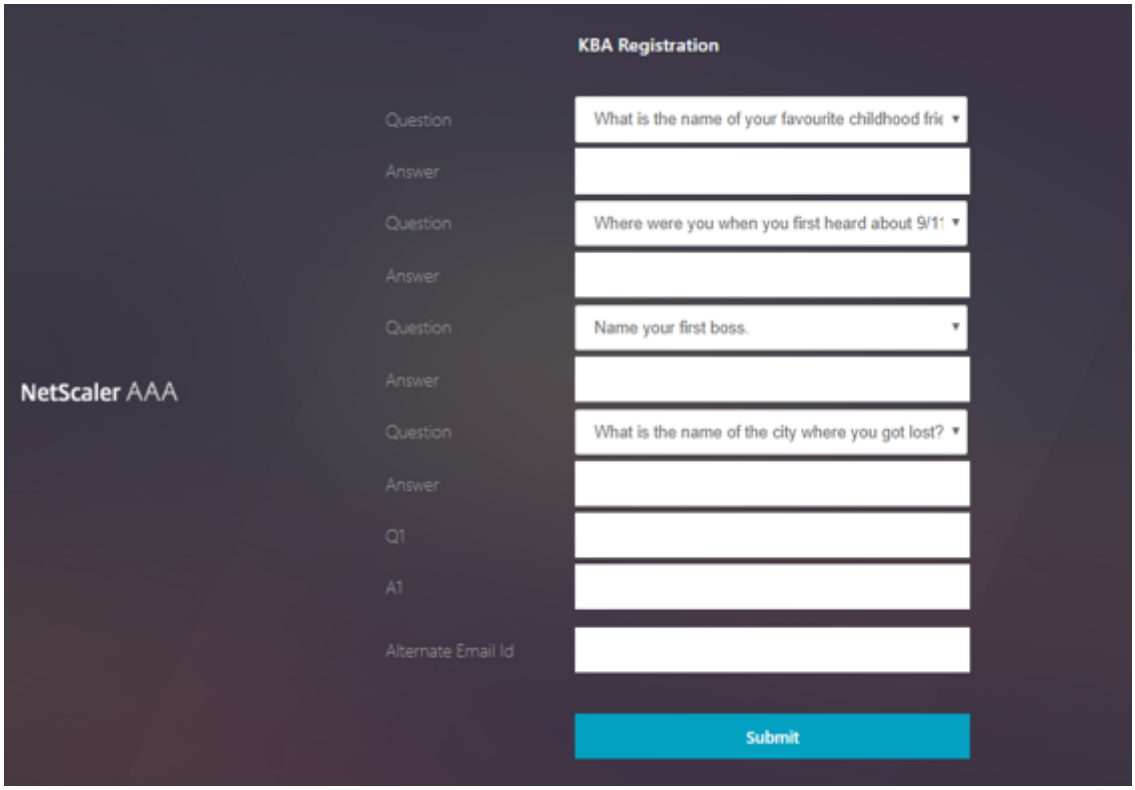
ユーザー登録と管理の検証

前のセクションで説明した手順をすべて設定したら、次の UI 画面が表示されます。

1. ロードバランサー仮想サーバの URL を入力します（例:https://lb1.server.com)。ログオン画面が表示されます。



2. ユーザ名とパスワードを入力します。[**Submit**] をクリックします。[ユーザー登録] 画面が表示されます。



3. ドロップダウンリストから希望する質問を選択し、[**Answer**] を入力します。
4. [**Submit**] をクリックします。ユーザー登録成功画面が表示されます。

ユーザーログオンの設定ページ

この例では、管理者は最初の要素が LDAP ログオン (エンドユーザーがパスワードを忘れた) であると想定しています。その後、ユーザーはナレッジベースの質問と回答の登録と電子メール ID OTP 検証に従い、最後にセルフサービスパスワードリセットを使用してパスワードをリセットします。

セルフサービスパスワードリセットには、どの認証メカニズムでも使用できます。強固なプライバシーを確保し、不正なユーザーパスワードのリセットを防ぐため、ナレッジベースの質問と回答、OTP、あるいはその両方を行うことをお勧めします。

ユーザーログオンページの設定を開始する前に、次のことが必要です:

- ロードバランサ仮想サーバの IP
- ロードバランサ仮想サーバに対応する FQDN
- ロードバランサーのサーバー証明書

CLI を使用してロードバランサ仮想サーバを作成する

内部 Web サイトにアクセスするには、バックエンドサービスの前にある LB 仮想サーバーを作成し、認証ロジックを認証仮想サーバーに委任する必要があります。

```
> add lb vserver lb1 SSL 1.2.3.162 443 -persistenceType NONE -
   cltTimeout 180 -AuthenticationHost otpauth.server.com -
   Authentication ON -authnVsName authvs

> bind ssl vserver lb1 -certkeyname c1
```

負荷分散でバックエンドサービスを表すには、次のようにします。

```
> add service iis_backendsso_server_com 1.2.3.4 HTTP 80

> bind lb vserver lb1 iis_backendsso_server_com
```

最初のポリシーとして認証を無効にした **LDAP** アクションの作成

```
> add authentication ldapAction ldap3 -serverIP 1.2.3.4 -serverPort 636
    -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
   administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
   ldapLoginName samAccountName -authentication disabled

> add authentication policy ldap3 -rule aaa.LOGIN.VALUE("passwdreset").
   EQ("1") -action ldap3
```

ナレッジベースの質問と回答の検証アクションを作成する

セルフサービスパスワードリセットフローでナレッジベースの質問と回答を検証するには、認証を無効にして LDAP サーバを設定する必要があります。

```
> add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP
   > -serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
   ldapBindDnPassword <PASSWORD> -ldapLoginName <USER FORMAT> -
   KBAttribute  <LDAP ATTRIBUTE> - alternateEmailAttr <LDAP ATTRIBUTE>
   -authentication DISABLED
```

例:

```
> add authentication ldapAction ldap2 -serverIP 1.2.3.4 -serverPort 636
    -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
   administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
   ldapLoginName samAccountName -KBAttribute userParameters -
   alternateEmailAttr userParameters -authentication disabled
```


CLI を使用してナレッジベースの質問と回答を検証するための認証ポリシーを作成するには

```
add authentication policy kba_validation -rule true -action ldap2
```

E メール検証アクションを作成する

セルフサービスパスワードリセット登録の一環として、ユーザーの電子メール ID または代替電子メール ID が必要なため、LDAP は電子メール検証要素に先行する必要があります。

> 注:
>
> 電子メール OTP ソリューションを機能させるには、SMTP サーバーでログインベースの認証が有効になっていることを確認してください。

ログインベースの認証が有効になっていることを確認するには、SMTP サーバーで次のコマンドを入力します。ログインベースの認証が有効になっている場合は、出力に **AUTH LOGIN** というテキストが太字で表示されます。

```
root@ns# telnet <IP address of the SMTP server><Port number of the
    server>
ehlo
```

例:

```
root@ns# telnet 10.106.3.66 25
Trying 10.106.3.66...
Connected to 10.106.3.66.
Escape character is '^]'.
220 E2K13.NSGSanity.com Microsoft ESMTP MAIL Service ready at Fri, 22
    Nov 2019 16:24:17 +0530
ehlo
250-E2K13.NSGSanity.com Hello [10.221.41.151]
250-SIZE 37748736
250-PIPELINING
250-DSN
250-ENHANCEDSTATUSCODES
250-STARTTLS
250-X-ANONYMOUSTLS
250-AUTH LOGIN
250-X-EXPS GSSAPI NTLM
250-8BITMIME
250-BINARYMIME
250-CHUNKING
250 XRDST
```

ログインベースの認証を有効にする方法については、https://support.microfocus.com/kb/doc.php?id=7020367を参照してください。

CLI を使用して電子メールアクションを設定するには

```
add authentication emailAction emailact -userName sender@example.com -
    password <Password> -serverURL "smtps://smtp.example.com:25" -
    content "OTP is $code"
```

例:

```
add authentication emailAction email -userName testmail@gmail.com -
    password 298
    a34b1a1b7626cd5902bbb416d04076e5ac4f357532e949db94c0534832670 -
    encrypted -encryptmethod ENCMTHD_3 -serverURL "smtps
    ://10.19.164.57:25" -content "OTP is $code" -emailAddress "aaa.user.
    attribute(\"alternate_mail\")"
```

> 注
>
> 構成の「EmailAddress」パラメータは PI 式です。そのため、セッションのデフォルトのユーザー電子メール ID、または既に登録済みの代替電子メール ID のいずれかを使用するように設定されます。

GUI を使用して電子メール **ID** を設定するには

1. [セキュリティ] **>** [**AAA**-アプリケーショントラフィック] **>** [ポリシー] **>** [認証] **>** [詳細ポリシー] **>** [アクション] [追加] をクリックします。
2. [認証メールアクションの作成] ページで詳細を入力し、[作成] をクリックします。

CLI を使用して **E** メール検証用の認証ポリシーを作成するには

```
add authentication policy email_validation -rule true -action email
```

パスワードリセット係数の認証ポリシーを作成するには

```
add authentication policy ldap_pwd -rule true -action ldap_logon_action
```

ログインスキーマによる **UI** の提示

パスワードをリセットするためのセルフサービスパスワードリセット用の LoginSchema は 3 つあります。次の CLI コマンドを使用して、3 つのログインスキーマを表示します。

```
root@ns# cd  /nsconfig/loginschema/LoginSchema/
root@ns# ls -ltr | grep -i password
-r--r--r--  1 nobody  wheel  2088 Nov 13 08:38
    SingleAuthPasswordResetRem.xml
-r--r--r--  1 nobody  wheel  1541 Nov 13 08:38
    OnlyUsernamePasswordReset.xml
-r--r--r--  1 nobody  wheel  1391 Nov 13 08:38 OnlyPassword.xml
```


CLI を使用して **1** つの認証パスワードリセットを作成するには

```
> add authentication loginSchema lschema_password_reset -
    authenticationSchema "/nsconfig/loginschema/LoginSchema/
    SingleAuthPasswordResetRem.xml"

> add authentication loginSchemaPolicy lpol_password_reset -rule true -
    action lschema_password_reset
```

ポリシーラベルを使用して、ナレッジベースの質問と回答と **E** メールによる **OTP** 検証ファクタを作成

最初の要素が LDAP ログオンの場合、次のコマンドを使用して、次の要素に対する知識ベースの質問と回答および電子メールの OTP ポリシーラベルを作成できます。

```
> add authentication loginSchema lschema_noschema -authenticationSchema
    noschema

> add authentication policylabel kba_validation -loginSchema
    lschema_noschema

> add authentication policylabel email_validation -loginSchema
    lschema_noschema
```

ポリシーラベルによるパスワードリセット係数の作成

次のコマンドを使用して、ポリシーラベルからパスワードリセット係数を作成できます。

```
> add authentication loginSchema lschema_noschema -authenticationSchema
    noschema

> add authentication policylabel password_reset -loginSchema
    lschema_noschema

> bind authentication policylabel password_reset -policyName ldap_pwd -
    priority 10 -gotoPriorityExpression NEXT
```

次のコマンドを使用して、ナレッジベースの質問と回答と電子メールポリシーを以前に作成したポリシーにバインドします。

```
> bind authentication policylabel email_validation -policyName
    email_validation -nextfactor password_reset -priority 10 -
    gotoPriorityExpression NEXT

> bind authentication policylabel kba_validation -policyName
    kba_validation -nextfactor email_validation -priority 10 -
    gotoPriorityExpression NEXT
```


フローをバインドする

LDAP Logon の認証ポリシーの下で LDAP ログオンフローが作成されている必要があります。このフローでは、最初の LDAP ログオンページに表示される [パスワードを忘れた場合] リンクをクリックし、次に KBA 検証、OTP 検証、最後にパスワードリセットページの順にクリックします。

```
bind authentication vserver authvs -policy ldap3 -nextfactor
    kba_validation -priority 10 -gotoPriorityExpression NEXT
```

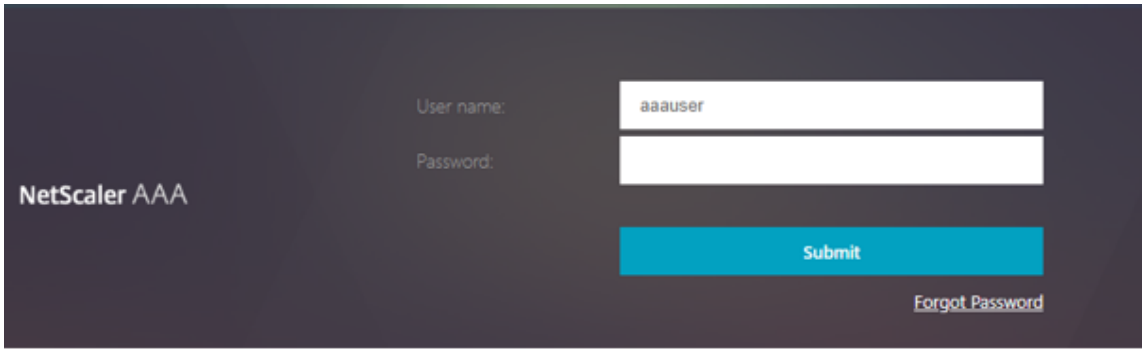
すべての **UI** フローをバインドするには

```
bind authentication vserver authvs -policy lpol_password_reset -
    priority 20 -gotoPriorityExpression END
```

パスワードをリセットするユーザーログオンワークフロー

ユーザーがパスワードをリセットする必要がある場合のユーザーログオンワークフローを次に示します。

1. ロードバランサー仮想サーバの URL を入力します（例:https://lb1.server.com）。ログオン画面が表示されます。



2. [パスワードを忘れた場合] をクリックします。検証画面には、AD ユーザーに対して登録された最大 6 つの質問と回答のうち 2 つの質問が表示されます。
3. 質問に答えて、[ログオン] をクリックします。登録済みの代替電子メール ID で受信した OTP を入力する必要がある電子メール OTP 検証画面が表示されます。
4. 電子メール OTP を入力します。電子メール OTP の検証に成功すると、パスワードのリセットページが表示されます。
5. 新しいパスワードを入力し、新しいパスワードを確認します。[**Submit**] をクリックします。パスワードのリセットに成功すると、[パスワードリセット成功] 画面が表示されます。

これで、リセットパスワードを使用してログオンできます。

トラブルシューティング

NetScaler には、セルフサービスパスワードリセットを使用する際に直面する可能性のある基本的な問題のいくつかをトラブルシューティングするオプションがあります。次のセクションは、特定の領域で発生する可能性がある問題のトラブルシューティングに役立ちます。

NS ログ

ログを分析する前に、`set syslogparams -loglevel DEBUG` コマンドを使用してログレベルをデバッグするように設定することをお勧めします。このパラメータを有効にすると、ログが`ns.log`ファイルに出力されます。

登録

次のメッセージは、ユーザー登録が成功したことを示します。

```
"ns_aaa_insert_hash_keyValue_entry key:kba_registered value:1"
Nov 14 23:35:51 <local0.debug> 10.102.229.76 11/14/2018:18:05:51 GMT
    0-PPE-1 : default SSLVPN Message 1588 0 :  "
    ns_aaa_insert_hash_keyValue_entry key:alternate_mail value:
    eyJ2ZXJzaW9uIjoiMSIsICJraWQiOiIxbk1oWjN0T2NjLVVvZUx6NDRwZFhxdS01dTA9IiwgImtleS
    ==.oKmvOalaOJ3a9z7BcGCSegNPMw=="
```

知識ベースの質問と回答の検証

次のメッセージは、ナレッジベースの質問と回答の検証に成功したことを示します。

```
"NFactor: Successfully completed KBA Validation, nextfactor is email"
```

電子メール ID の検証

次のメッセージは、パスワードのリセットが成功したことを示します。

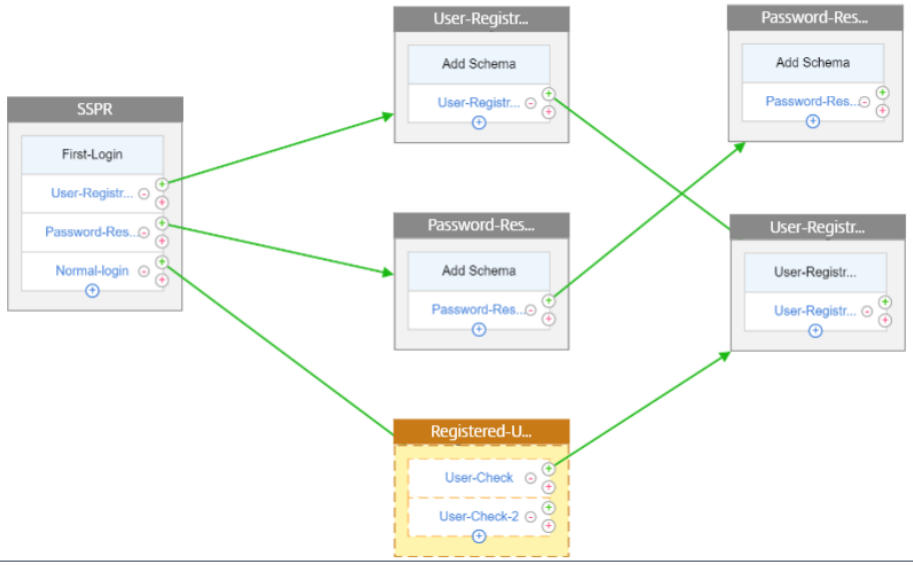
```
"NFactor: Successfully completed email auth, nextfactor is pwd_reset"
```

nFactor ビジュアライザーを使用した SSPR の設定

nFactor ビジュアライザーを使用して SSPR 設定を開始する前に、次の LDAP サーバーを追加する必要があります:

1. ユーザ認証に対して認証が有効になっていて、AD 属性が指定されている標準 LDAP サーバ。

2. 認証なしでユーザパラメータを抽出するための LDAP サーバ。
3. 認証なしの SSL でパスワードをリセットするための LDAP サーバ。また、ユーザー詳細の保存に使用する AD 属性は、このサーバーで定義する必要があります。
4. 認証が有効で、AD 属性が指定された、ユーザー登録用の LDAP サーバー。
5. 次の図に、フロー全体を示します:

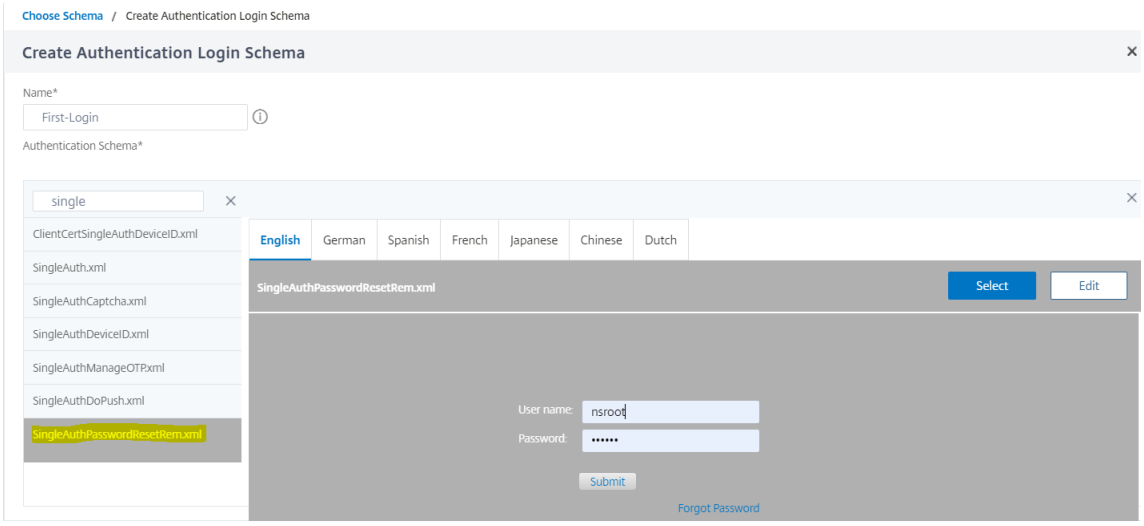


6. 次の CLI コマンドを使用して、証明書をグローバルにバインドします。

```
bind vpn global -userDataEncryptionKey Wildcard
```

LDAP サーバが追加されたので、ビジュアライザを使用して **nFactor** の設定を続行します

1. [セキュリティ **] > [AAA] > [**アプリケーショントラフィック **] > [nFactor Visualizer] > [nFactor** フロー] に移動し、[追加] をクリックして、ボックス内の + アイコンをクリックします。
2. フローに名前を付けます。
3. 既定のスキーマとして機能する **[**スキーマの追加] をクリックします。[ログインスキーマ] ページで [追加] をクリックします。
4. スキーマに名前を付けたら、スキーマを選択します。右上隅の「選択」(**Select**) をクリックして、スキーマを選択します。



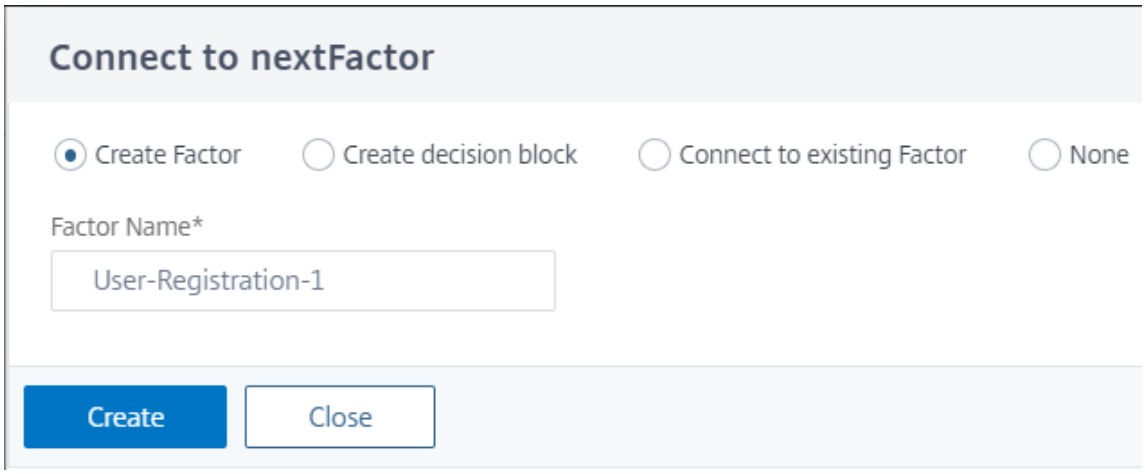
5. 「作成」をクリックして「**OK**」をクリックします。

デフォルトのスキーマを追加したら、次の 3 つのフローを設定する必要があります。

- ユーザー登録: 明示的なユーザー登録の場合
- パスワードリセット: パスワードリセット用
- 通常ログイン + 登録済みユーザーチェック: ユーザーが登録され、正しいパスワードを入力した場合、そのユーザーはログインしています。ユーザーが登録されていない場合は、登録ページに移動します。

ユーザー登録　スキーマを追加した後、残ったところから続けてみましょう。

1. [**Add Policy**] をクリックすると、ユーザーが明示的に登録しようとしているかどうかがチェックされます。
2. [作成] をクリックし、[追加] をクリックします。
3. 緑色の「+」アイコンをクリックして、ユーザー登録フローに次の認証要素を追加します。



4. ［作成］をクリックします。
5. [ユーザー登録 1 係数のポリシーの追加] をクリックします。

6. 認証ポリシーを作成します。このポリシーは、登録ページにリダイレクトする前に、ユーザー情報を抽出して検証します。
7. [作成] をクリックし、[追加] をクリックします。
8. 緑色の「+」アイコンをクリックしてユーザー登録用の別の要素を作成し、「作成」をクリックします。[スキーマの追加] をクリックします。

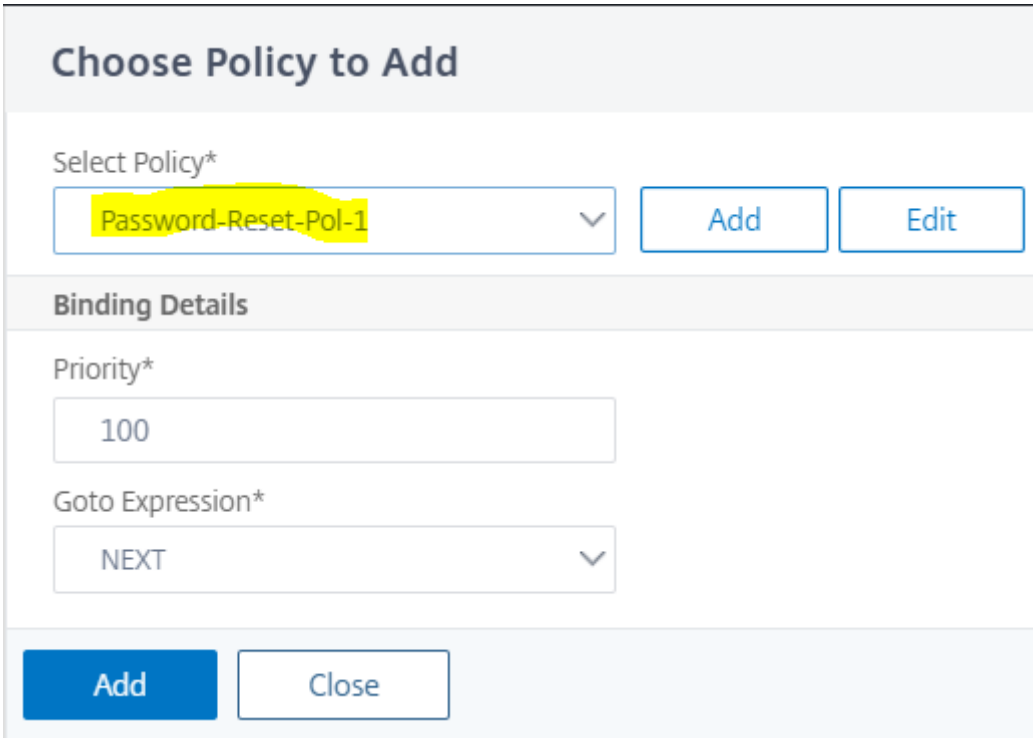
9. ユーザー登録用の認証ログインスキーマを作成します。
10. [ポリシーの追加] をクリックし、認証ポリシーを作成します。
11. [作成] をクリックし、[追加] をクリックします。

パスワードリセット

1. 青色の「+」アイコンをクリックして、親 SSPR ファクターに別のポリシー (パスワードリセットフロー) を追加します。

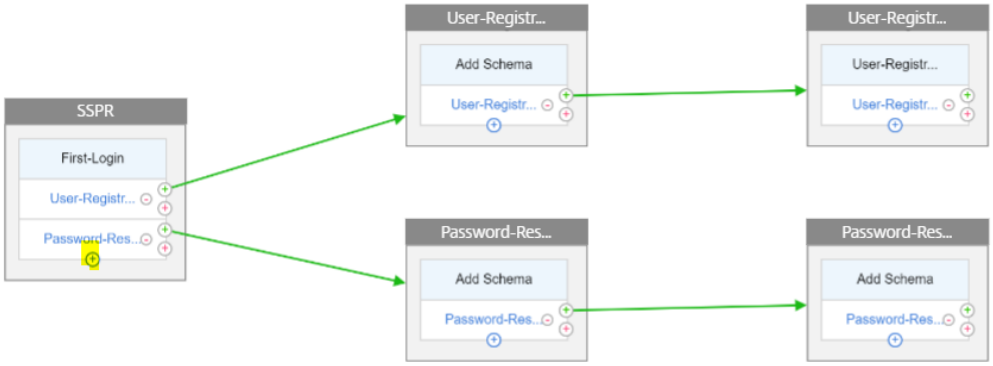
2. 「認証ポリシーの選択」ページで、「追加」をクリックして認証ポリシーを作成するか、ドロップダウンリストから既存のポリシーを選択します。このポリシーは、ユーザーがログインページで [パスワードを忘れた場合] をクリックするとトリガーされます。
3. [追加] をクリックします。
4. パスワードリセット認証ポリシーの緑色の「+」アイコンをクリックして、次の要素を追加します。

5. [作成] をクリックします。
6. [**Add policy**] をクリックして、以前に作成したファクターの認証ポリシーを作成します。この係数は、ユーザーを検証するためのものです。
7. [追加] をクリックします。
8. 緑色の「+」アイコンをクリックして、パスワードファクタフローの次のファクタを追加します。これにより、パスワードをリセットするための回答が検証されます。[作成] をクリックします。
9. [**Add Policy**] をクリックして、ファクタの認証ポリシーを追加します。
10. ドロップダウンリストから同じ認証ポリシーを選択し、[追加] をクリックします。

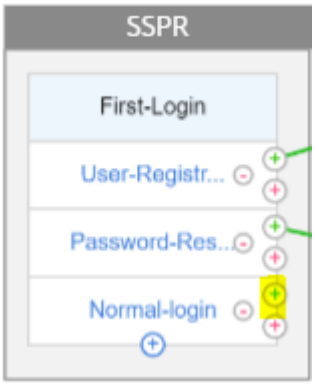


通常ログイン + 登録ユーザーチェック

1. 青い「+」アイコンをクリックして、親 SSPR ファクターに別の認証ポリシー (通常のログインフロー) を追加します。



2. [**Add**] をクリックして、通常のユーザログイン用の認証ポリシーを作成します。
3. [作成] をクリックし、[追加] をクリックします。



4. 以前に作成したポリシーの緑色の「+」アイコンをクリックして次の要素を追加し、「デシジョンブロックの作成」オプションを選択し、デシジョンブロックの名前を指定します。
5. [作成] をクリックします。
6. [**Add Policy**] をクリックして、この決定要因の認証ポリシーを作成します。
7. [作成] をクリックして [追加] をクリックします。これは、ユーザーが登録されているかどうかをチェックします。
8. デシジョンブロックタイルの緑色の「+」アイコンをクリックして登録ポリシーに移動し、「既存の要素に接続」オプションを選択します。

9. 「接続先」ドロップダウンリストから登録係数を選択し、「作成」をクリックします。
10. 次に、デシジョンブロックタイルの青い「+」アイコンをクリックして、デシジョンブロックに別のポリシーを追加します。このポリシーは、登録ユーザーが認証を完了するためのものです。

11. [**Add Policy**] をクリックして、認証ポリシーを作成します。
12. [作成] をクリックし、[追加] をクリックします。

認証中のポーリング

August 15, 2023

NetScaler リリースビルド 13.0.79.64 以降、NetScaler アプライアンスを多要素認証中にポーリングメカニズムを構成できるようになりました。

NetScaler アプライアンスでポーリングが構成されている場合、エンドポイント（Web ブラウザーやアプリなど）は、設定された間隔で認証中にアプライアンスをポーリング（プローブ）して、送信された認証リクエストのステータスを取得できます。

NetScaler アプライアンスでの認証中にエンドポイントが TCP 接続を切断した場合に認証を処理するようにポーリングを構成できます。

注意事項

- ポーリング設定は、LDAP、RADIUS、および TACACS 認証方式でサポートされています。
- クライアントは、第 2 要素以降から認証要求をプローブできます。

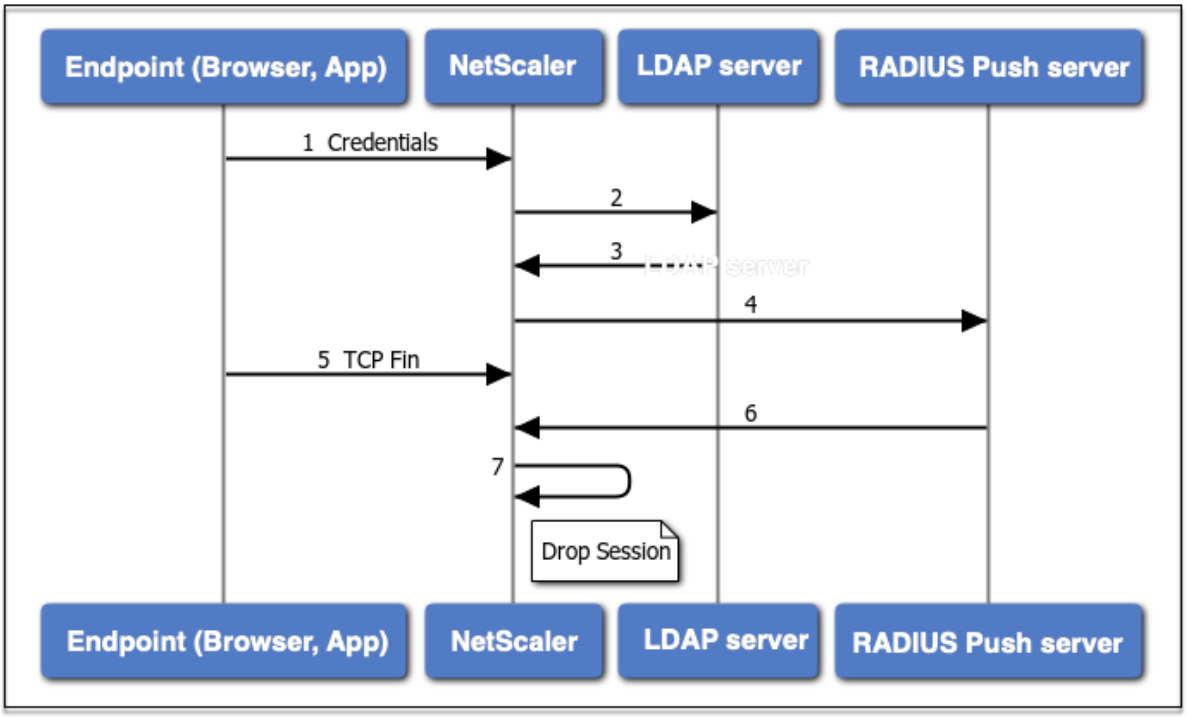
ポーリングを設定する理由

認証中にアプリ（ログインアプリと認証アプリなど）を切り替えると、エンドポイントが NetScaler アプライアンスとの接続を失い、認証フローが中断することがあります。ポーリングを設定すると、この認証のブレークを回避できます。

ポーリングメカニズムを理解する

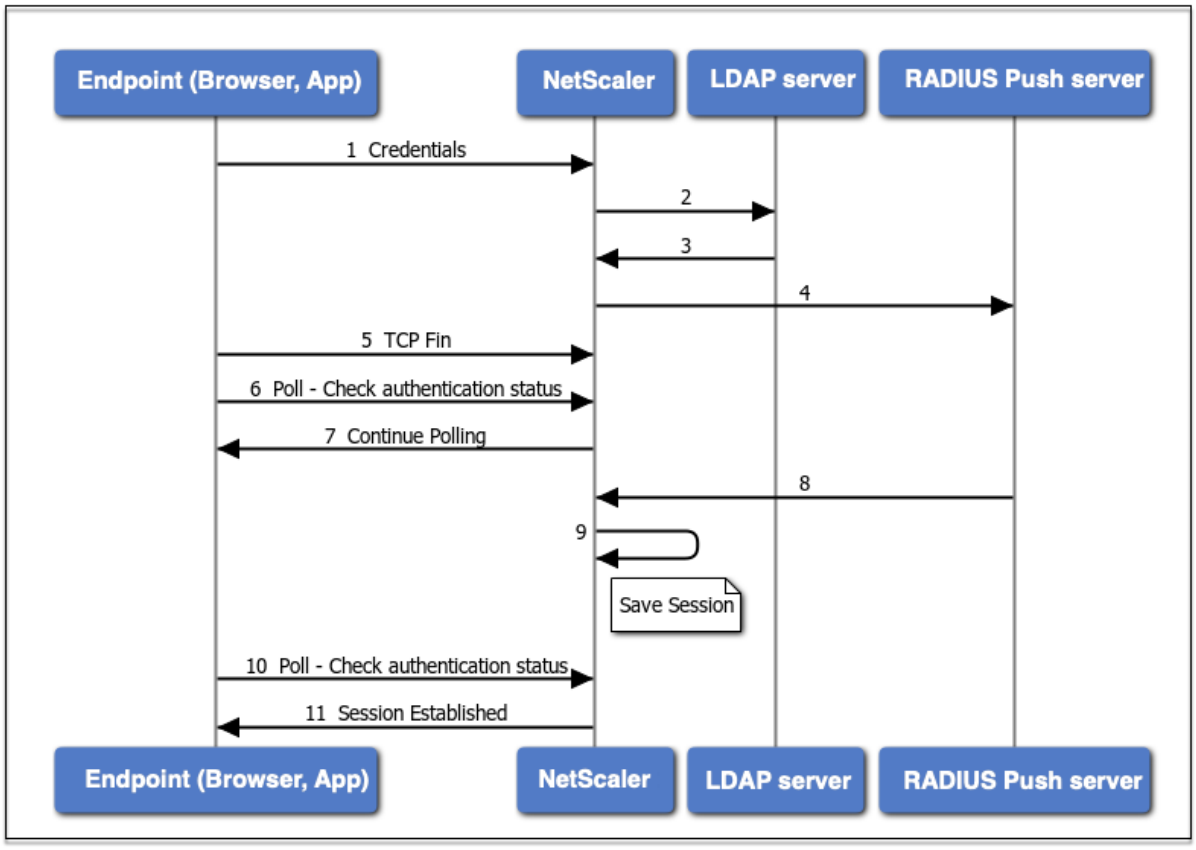
次に、ポーリングが設定されていない認証中のイベントのフローの例を示します。

ポーリングメカニズムにより、NetScaler アプライアンスは、まれにエンドポイントで TCP 接続がリセットされた場合でも、認証プロセスを再開しなくても、エンドポイントでの継続的な認証を再開できます。



1. エンドポイント (アプリまたは Web ブラウザ) は、認証情報を使用して認証します。
2. ユーザー名とパスワードは、既存の第 1 要素ディレクトリ (LDAP/Active Directory) に対して検証されます。
3. 正しいクレデンシャルが指定されている場合、認証は次の要素に移行します。
4. この時点で、NetScaler アプライアンスは RADIUS プッシュサーバーに要求を送信します。
5. NetScaler アプライアンスが RADIUS サーバーからの応答を待っている間、エンドポイントは TCP 接続を切断します。
6. NetScaler は RADIUS プッシュサーバーから応答を受け取ります。
7. クライアントの TCP 接続が見つからないため、NetScaler アプライアンスはセッションをドロップし、ログインは失敗します。

次に、ポーリングが設定された認証時のイベントのフローの例を示します。



1. エンドポイント (アプリまたは Web ブラウザ) は、認証情報を使用して認証します。
2. ユーザー名とパスワードは、既存の第 1 要素ディレクトリ (LDAP/Active Directory) に対して検証されます。
3. 正しいクレデンシャルが指定されている場合、認証は次の要素に移行します。
4. この時点で、NetScaler アプライアンスは RADIUS プッシュサーバーに要求を送信します。
5. NetScaler アプライアンスが RADIUS サーバーからの応答を待っている間、エンドポイントは TCP 接続を切断します。
6. エンドポイントは NetScaler アプライアンスにポーリング（プローブ）を送信して認証ステータスを確認します。
7. NetScaler アプライアンスは RADIUS サーバーからの応答がないため、エンドポイントにポーリングの継続を要求します。
8. NetScaler アプライアンスは、RADIUS プッシュサーバーから応答を受け取ります。
9. クライアント TCP 接続が見つからない場合、ADC はセッション状態を保存します。
10. エンドポイントは再びポーリングして、認証ステータスを確認します。
11. NetScaler アプライアンスがセッションを確立し、ログインが成功します。

CLI を使用したポーリングの設定

次に、CLI 設定の例を示します。

第 **1** 要素を構成する

```
add authentication ldapAction ldap-new -serverIP 10.106.40.65 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword 2
    f63d3659103464a4fad0ade65e2ccfd4e8440e36ddff941d29796af03e01139 -
    encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -
    groupAttrName memberof -subAttributeName CN -secType SSL -
    alternateEmailAttr userParameters

add authentication Policy ldap-new -rule true -action ldap-new

bind authentication vserver avs -policy ldap-new -priority 1 -
    nextFactor rad_factor
<!--NeedCopy-->
```

第 **2** 要素を構成する

```
add authentication radiusAction rad1 -serverIP 10.102.229.120 -radKey 1
    b1613760143ce2371961e9a9eb5392c86a4954a62397f29a01b5d12b42ce232 -
    encrypted -encryptmethod ENCMTHD_3

add authentication Policy rad -rule true -action rad1
<!--NeedCopy-->
```

Poll.xml ログインスキーマを構成する

```
add authentication loginSchema polling_schema -authenticationSchema
    LoginSchema/Poll.xml

add authentication policylabel rad_factor -loginSchema polling_schema

bind authentication policylabel rad_factor -policyName rad -priority 1
    -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

GUI を使用したポーリングの設定

GUI を使用した多要素認証の設定手順の詳細については、nFactor 認証の設定を参照してください。

以下は、セカンドファクター以降のポーリング用に NetScaler を構成するために必要な大まかな手順の例です。

1. LDAP など、認証の最初の要素を作成します。
2. RADIUS など、認証の 2 番目の要素を作成します。
3. NetScaler (/nsConfig/loginSchema/LoginSchema/) にある **Poll.xml** を第 2 ファクタのログインスキーマとして追加します。

セッションとトラフィックの管理

August 15, 2023

セッションの設定

認証、承認、および監査プロファイルを構成したら、セッション設定を構成してユーザーセッションをカスタマイズします。セッション設定は次のとおりです。

- セッションタイムアウト。

  ユーザーが自動的に切断され、イントラネットにアクセスするために再度認証が必要になるまでの期間を制御します。

- デフォルトの認証設定。

  NetScaler アプライアンスが特定の認証ポリシーのないコンテンツへのアクセスをデフォルトで許可するか拒否するかを決定します。

- シングルサインオンの設定。

  NetScaler アプライアンスが認証後にユーザーをすべての Web アプリケーションに自動的にログオンさせるか、ユーザーを Web アプリケーションのログオンページに渡して各アプリケーションの認証を行うかを決定します。

- クレデンシャルインデックスの設定。

  NetScaler アプライアンスがシングルサインオンにプライマリ認証資格情報を使用するかセカンダリ認証情報を使用するかを決定します。

セッション設定を構成するには、次の 2 つの方法のいずれかを使用できます。ユーザーアカウントまたはグループごとに異なる設定が必要な場合は、カスタムセッション設定を構成するユーザーアカウントまたはグループごとにプロファイルを作成します。また、特定のプロファイルを適用する接続を選択するポリシーを作成し、そのポリシーをユーザーまたはグループにバインドします。また、プロファイルを適用するトラフィックを処理する認証仮想サーバにポリシーをバインドすることもできます。

すべてのセッションに同じ設定を適用する場合、または特定のプロファイルとポリシーが設定されていないセッションのデフォルト設定をカスタマイズする場合は、グローバルセッション設定を構成するだけです。

セッションプロファイル

ユーザーセッションをカスタマイズするには、まずセッションプロファイルを作成します。セッションプロファイルを使用すると、任意のセッションパラメータのグローバル設定を上書きできます。

> 注
>
> 「セッションプロファイル」と「セッションアクション」という用語は同じ意味です。

コマンドラインインターフェイスを使用してセッションプロファイルを作成するには

コマンドプロンプトで次のコマンドを入力して、セッションプロファイルを作成し、構成を確認します。

```
add tm sessionAction <name> [-sessTimeout <mins>] [-
   defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
   ssoCredential ( PRIMARY | SECONDARY )] [-ssoDomain <string>][-
   httpOnlyCookie ( YES | NO )] [-persistentCookie ( ENABLED | DISABLED
    )] [-persistentCookieValidity <minutes>]

show tm sessionAction <name>
<!--NeedCopy-->
```

例

```
> add tm sessionAction session-profile -sessTimeout 30 -
   defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションプロファイルを変更するには

コマンドプロンプトで次のコマンドを入力して、セッションプロファイルを変更し、構成を確認します。

```
set tm sessionAction <name> [-sessTimeout <mins>] [-
   defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
   ssoCredential ( PRIMARY | SECONDARY )] [-ssoDomain <string>][-
   httpOnlyCookie ( YES | NO )] [-persistentCookie ( ENABLED | DISABLED
    )] [-persistentCookieValidity <minutes>]

show tm sessionAction
<!--NeedCopy-->
```

例

```
> set tm sessionAction session-profile -sessTimeout 30 -
   defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
```



```
        Authorization action : ALLOW
        Session timeout: 30 minutes
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションプロファイルを削除するには

コマンドプロンプトで次のコマンドを入力して、セッションプロファイルを削除します。

```
rm tm sessionAction <name>
<!--NeedCopy-->
```

構成ユーティリティを使用してセッションプロファイルを構成するには

1. [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**セッション] に移動します。
2. [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**セッション] に移動します。
3. 詳細ペインで、「プロファイル」タブをクリックします。
4. 「プロファイル」タブで、次のいずれかを実行します。
   - 新しいセッションプロファイルを作成するには、[追加（**Add**）**]** をクリックします。
   - 既存のセッションプロファイルを変更するには、プロファイルを選択し、[編集**]** をクリックします。
5. [TM セッションプロファイルの作成] または [TM セッションプロファイルの構成] ダイアログで、パラメータの値を入力または選択します。
   - name*：actionName（以前に設定されたセッションアクションでは変更できません。）
   - セッションタイムアウト：sesTimeout
   - Web アプリケーションへのシングルサインオン-SSO
   - デフォルトの認可アクション：defaultAuthorizationAction
   - クレデンシャルインデックス：SSOCredential
   - シングルサインオンドメイン—SSODomain
   - httpOnlyCookie —httpOnlyCookie
   - パーシステントクッキーの有効化-パーシステントクッキー
   - パーシステントクッキーの有効性—パーシステントクッキーの有効性
6. [作成] または [**OK**] をクリックします。作成したセッションプロファイルが [セッションポリシーおよびプロファイル] ペインに表示されます。

セッションポリシー

1 つ以上のセッションプロファイルを作成したら、セッションポリシーを作成し、ポリシーをグローバルにバインドするか、認証仮想サーバにバインドして有効にします。

コマンドラインインターフェイスを使用してセッションポリシーを作成するには

コマンドプロンプトで次のコマンドを入力して、セッションポリシーを作成し、構成を確認します。

```
-  add tm sessionPolicy <name> <rule> <action>
-  show tm sessionPolicy <name>
<!--NeedCopy-->
```

例

```
> add tm sessionPolicy session-pol "URL == /\*.png" session-profile
Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/\*.png'
        Action: session-profile
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションポリシーを変更するには

コマンドプロンプトで次のコマンドを入力して、セッションポリシーを変更し、構成を確認します。

```
-  set tm sessionPolicy <name> [-rule <expression>] [-action <action>]
-  show tm sessionPolicy <name>
<!--NeedCopy-->
```

例

```
> set tm sessionPolicy session-pol "URL == /\*.png" session-profile
 Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/\*.png'
        Action: session-profile
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションポリシーをグローバルにバインドするには

コマンドプロンプトで次のコマンドを入力して、セッションポリシーをグローバルにバインドし、構成を確認します。

```
bind tm global -policyName <policyname> [-priority <priority>]
<!--NeedCopy-->
```

例

```
> bind tm global -policyName session-pol
 Done
```



```
3
4  > show tm sessionPolicy session-pol
5  1)      Name: session-pol       Rule: URL == '/*.png'
6          Action: session-profile
7          Policy is bound to following entities
8          1) TM GLOBAL    PRIORITY : 0
9   Done
10
11 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションポリシーを認証仮想サーバーにバインドするには

コマンドプロンプトで次のコマンドを入力して、セッションポリシーを認証仮想にバインドし、構成を確認します。

```
1 bind authentication vserver <name> -policy <policyname> [-priority <
     priority>]
2 <!--NeedCopy-->
```

例

```
1 bind authentication vserver auth-vserver-1 -policyName Session-Pol-1 -
     priority 1000
2 Done
3 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して認証仮想サーバーからセッションポリシーをバインド解除するには

コマンドプロンプトで次のコマンドを入力して、認証仮想サーバーからセッションポリシーをバインド解除し、構成を確認します。

```
1 unbind authentication vserver <name> -policy <policyname>
2 <!--NeedCopy-->
```

例

```
1 unbind authentication vserver auth-vserver-1 -policyName Session-Pol-1
2 Done
3 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用してグローバルにバインドされたセッションポリシーをバインド解除するには

コマンドプロンプトで次のコマンドを入力して、グローバルにバインドされたセッションポリシーをバインド解除します。


```
unbind tm global -policyName <policyname>
<!--NeedCopy-->
```

例

```
unbind tm global -policyName Session-Pol-1
Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションポリシーを削除するには

まず、グローバルからセッションポリシーをバインド解除し、コマンドプロンプトで次のコマンドを入力して、セッションポリシーを削除し、構成を確認します。

```
rm tm sessionPolicy <name>
<!--NeedCopy-->
```

例

```
rm tm sessionPolicy Session-Pol-1
Done

<!--NeedCopy-->
```

構成ユーティリティを使用してセッションポリシーを構成およびバインドするには

1. [セキュリティ] **> [AAA-**アプリケーショントラフィック] **> [**セッション] に移動します。
2. [セキュリティ] **> [AAA-**アプリケーショントラフィック] **> [**ポリシー] **> [**セッション] に移動します。
3. 詳細ウィンドウの [ポリシー] タブで、次のいずれかの操作を行います。
   - 新しいセッションポリシーを作成するには、[追加] をクリックします。
   - 既存のセッションポリシーを変更するには、ポリシーを選択し、[編集] をクリックします。
4. [セッションポリシーの作成] または [セッションポリシーの構成] ダイアログで、パラメータの値を入力または選択します。
   - name*: policyName（以前に設定されたセッションポリシーでは変更できません。）
   - リクエストプロファイル *—ActionName
   - 式 *-規則 (式を入力するには、まず [式] テキスト領域の下にある左端のドロップダウンリストで式のタイプを選択し、次に式テキスト領域に式を直接入力するか、[追加] をクリックして [式の追加] ダイアログボックスを開き、ドロップダウンを使用します。その中にリストして式を構成します。)
5. [作成] または [**OK**] をクリックします。作成したポリシーが、[**** セッションポリシーおよびプロファイル ****] ページの詳細ペインに表示されます。

6. セッションポリシーをグローバルにバインドするには、詳細ペインで、［アクション］ドロップダウンリストから［グローバルバインディング］を選択し、ダイアログに入力します。

   - グローバルにバインドするセッションポリシーの名前を選択します。
   - ［**OK**］をクリックします。

7. セッションポリシーを認証仮想サーバーにバインドするには、ナビゲーションペインで [仮想サーバー] をクリックし、そのポリシーをポリシーリストに追加します。

   - 詳細ウィンドウで、仮想サーバーを選択し、[編集] をクリックします。
   - 詳細領域の右側にある **[**詳細選択] で、[ポリシー] をクリックします。
   - ポリシーを選択するか、プラスアイコンをクリックしてポリシーを追加します。
   - 左側の [**Priority**] 列で、デフォルトの優先度を変更して、ポリシーが適切な順序で評価されるようにします。
   - ［**OK**］をクリックします。
     ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

グローバルセッション設定

セッションプロファイルおよびポリシーの作成に加えて、または作成する代わりに、グローバルセッション設定を構成できます。これらの設定は、セッション構成をオーバーライドする明示的なポリシーがない場合に、セッション構成を制御します。

コマンドラインインターフェイスを使用してセッション設定を構成するには

コマンドプロンプトで次のコマンドを入力して、グローバルセッション設定を構成し、構成を確認します。

```
set tm sessionParameter [-sessTimeout <mins>][-
    defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
    ssoCredential ( PRIMARY | SECONDARY )][-ssoDomain <string>][-
    httpOnlyCookie ( YES | NO )][-persistentCookie ( ENABLED | DISABLED
    )] [-persistentCookieValidity <minutes>]
<!--NeedCopy-->
```

例

```
> set tm sessionParameter -sessTimeout 30
  Done
> set tm sessionParameter -defaultAuthorizationAction DENY
  Done
> set tm sessionParameter -SSO ON
  Done
> set tm sessionParameter -ssoCredential PRIMARY
  Done
<!--NeedCopy-->
```


構成ユーティリティを使用してセッション設定を構成するには

1. セキュリティ **> AAA-**アプリケーショントラフィックに移動します。
2. 詳細ウィンドウの [設定] で、[グローバル設定の変更] をクリックします。
3. [グローバルセッション設定] ダイアログで、パラメータの値を入力または選択します。
    - セッションタイムアウト：sesTimeout
    - デフォルトの認可アクション：defaultAuthorizationAction
    - Web アプリケーションへのシングルサインオン：SSO
    - クレデンシャルインデックス：SSOCredential
    - シングルサインオンドメイン—SSODomain
    - httpOnlyCookie —httpOnlyCookie
    - パーシステントクッキーの有効化-パーシステントクッキー
    - パーシステントクッキーの有効性 (分) —パーシステントクッキーの有効性
    - ホームページ-ホームページ
4. ［**OK**］をクリックします。

トラフィック設定

保護されたアプリケーションにフォームベースまたは SAML シングルサインオン (SSO) を使用する場合は、その機能を [トラフィック] 設定で構成します。SSO を使用すると、ユーザーは 1 回ログオンして保護されているすべてのアプリケーションにアクセスできます。各アプリケーションにアクセスするために個別にログオンする必要はありません。

フォームベースの SSO では、一般的なポップアップウィンドウの代わりに、独自のデザインの Web フォームをサインオン方法として使用できます。したがって、会社のロゴやその他の情報をログオンフォームに表示することができます。SAML SSO を使用すると、1 つの NetScaler アプライアンスまたは仮想アプライアンスインスタンスが、最初のアプライアンスで認証されたユーザーに代わって別の NetScaler アプライアンスに対して認証されるように構成できます。

いずれかのタイプの SSO を設定するには、まずフォームまたは SAML SSO プロファイルを作成します。次に、トラフィックプロファイルを作成し、作成した SSO プロファイルにリンクします。次に、ポリシーを作成し、トラフィックプロファイルにリンクします。最後に、ポリシーをグローバルにバインドするか、認証仮想サーバにバインドして、設定を有効にします。

トラフィックプロファイル

少なくとも 1 つのフォームまたは SAML SSO プロファイルを作成したら、次にトラフィックプロファイルを作成する必要があります。

> 注:
>
> この機能では、「プロファイル」と「アクション」という用語は同じことを意味します。

コマンドラインインターフェイスを使用してトラフィックプロファイルを作成するには

コマンドプロンプトで入力します。

```
add tm trafficAction <name> [-appTimeout <mins>][-SSO ( ON | OFF ) [-
    formSSOAction <string>]][-persistentCookie ( ENABLED | DISABLED )][-
    InitiateLogout ( ON | OFF )]
<!--NeedCopy-->
```

例

```
add tm trafficAction Traffic-Prof-1  - appTimeout 10 -SSO ON -
    formSSOAction SSO-Prof-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションプロファイルを変更するには

コマンドプロンプトで入力します。

```
set tm trafficAction <name> [-appTimeout <mins>] [-SSO ( ON | OFF ) [-
    formSSOAction <string>]] [-persistentCookie ( ENABLED | DISABLED )]
    [-InitiateLogout ( ON | OFF )]
<!--NeedCopy-->
```

例

```
set tm trafficAction Traffic-Prof-1  - appTimeout 10 -SSO ON -
    formSSOAction SSO-Prof-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションプロファイルを削除するには

コマンドプロンプトで入力します。

```
rm tm trafficAction <name>
<!--NeedCopy-->
```

例

```
rm tm trafficAction Traffic-Prof-1
<!--NeedCopy-->
```


構成ユーティリティを使用してトラフィックプロファイルを構成するには

1. セキュリティ **> AAA-**アプリケーショントラフィック **>** トラフィックに移動します。
2. [セキュリティ **] > [AAA-**アプリケーショントラフィック **] > [**ポリシー **] > [**トラフィック] に移動します。
3. 詳細ペインで、「プロファイル」タブをクリックします。
4. 「プロファイル」タブで、次のいずれかを実行します。
   - 新しいトラフィックプロファイルを作成するには、[追加（**Add**）] をクリックします。
   - 既存のトラフィックプロファイルを変更するには、プロファイルを選択し、[編集（**Edit**）] をクリックします。
5. [トラフィックプロファイルの作成] または [トラフィックプロファイルの設定] ダイアログボックスで、パラメータの値を指定します。
   - name*：name（以前に設定されたセッションアクションでは変更できません）。
   - appTimeout —apptimeout
   - シングルサインオン：SSO
   - フォームの SSO アクション—FormsSOAction
   - SAML SSO アクション—samlssoAction
   - パーシステントクッキーの有効化-パーシステントクッキー
   - ログアウトの開始：InitiateLogout
6. [作成] または [**OK**] をクリックします。作成したトラフィックプロファイルは、[トラフィックポリシー]、[プロファイル]、および [フォーム SSO プロファイル] ペインまたは [SAML SSO プロファイル] ペインに適切に表示されます。

AAA.USER および **AAA.LOGIN** 式のサポート

AAA.USER 式は、既存の HTTP.REQ.USER 式を置き換えるために実装されました。AAA.USER 式は、Secure Web Gateway（SWG）やロールベースアクセス（RBA）メカニズムなど、HTTP 以外のトラフィックを処理する場合に適用できます。AAA.USER 式は HTTP.REQ.USER 式と同等です。

この式は、さまざまなアクションまたはプロファイル設定で使用できます。

コマンドプロンプトで入力します。

```
add tm trafficAction <name> [SSO (ON|OFF)] [-userExpression <string>]

add tm trafficAction <name> [SSO (ON|OFF)] [-passwdExpression <string>]

<!--NeedCopy-->
```

例

```
add tm trafficAction tm_act -SSO ON -userExpression "AAA.USER.NAME"

```



```
add tm trafficAction tm_act -SSO ON -userExpression "AAA.USER.PASSWD"

add tm trafficPolicy tm_pol true tm_act

bind lb vserver lb1 -policyName tm_pol -priority 2
<!--NeedCopy-->
```

> 注:
>
> HTTP.REQ.USER 式を使用すると、警告メッセージ”HTTP.REQ.USER は廃止されました。代わりに AAA.USER を使用する”とコマンドプロンプトに表示されます。

- **AAA.LOGIN** 式。LOGIN 式は、ログイン要求とも呼ばれるログイン前を表します。ログインリクエストは、NetScaler Gateway、SAML IdP、または OAuth 認証から行うことができます。NetScaler はポリシー構成から必要な属性を抽出します。AAA.LOGIN 式には属性が含まれ、次の条件に基づいて取得できます。
  - **AAA.LOGIN.USERNAME.** ユーザー名 (見つかった場合) は、現在のログイン要求から取得されます。非ログイン要求（認証、認可、および監査によって決定される）に適用された同じ式は、空の文字列になります。
  - **AAA.LOGIN.PASSWORD.** ユーザパスワード (見つかった場合) は、現在のログイン要求から取得されます。パスワードが見つからない場合、この式は空の文字列になります。
  - **AAA.LOGIN.PASSWORD2.** 2 番目のパスワード (見つかった場合) は、ログイン要求から取得されます。
  - **AAA.LOGIN.DOMAIN.** ドメイン情報は、ログイン要求からフェッチされます。
- **AAA.USER.ATTRIBUTE(“#”).** この式は、ユーザー属性を格納するために使用されます。ここで # は、整数 (1 から 16) または文字列値のいずれかになります。これらのインデックス値は、式 AAA.USER.ATTRIBUTE (「#」) を使用して使用できます。認証、承認、および監査モジュールはユーザーセッション属性を検索し、`AAA.USER.ATTRIBUTE("#")` はその特定の属性についてハッシュテーブルにクエリを実行します。たとえば、`Attributes("samaccountname")` が設定されている場合、`AAA.USER.ATTRIBUTE("samaccountname")` はハッシュマップを照会し、`samaccountname`に対応する値をフェッチします。

トラフィックポリシー

1 つ以上のフォーム SSO およびトラフィックプロファイルを作成したら、トラフィックポリシーを作成し、ポリシーをグローバルに、またはトラフィック管理仮想サーバにバインドして有効にします。

コマンドラインインターフェイスを使用してトラフィックポリシーを作成するには

コマンドプロンプトで入力します。

```
add tm trafficPolicy <name> <rule> <action>
<!--NeedCopy-->
```


例

```
add tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS("login=true")" Traffic-Prof-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してトラフィックポリシーを変更するには

コマンドプロンプトで入力します。

```
set tm trafficPolicy <name> <rule> <action>
<!--NeedCopy-->
```

例

```
set tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS("login=true")" Traffic-Prof-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してトラフィックポリシーをグローバルにバインドするには

コマンドプロンプトで入力します。

```
bind tm global -policyName <string> [-priority <priority>]
<!--NeedCopy-->
```

例

```
bind tm global -policyName Traffic-Pol-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してトラフィックポリシーを負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにバインドするには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
bind lb vserver <name> -policy <policyName> [-priority <priority>]

bind cs vserver <name> -policy <policyName> [-priority <priority>]
<!--NeedCopy-->
```

例

```
bind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1 -priority 1000
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用してグローバルにバインドされたトラフィックポリシーをバインド解除するには

コマンドプロンプトで入力します。

```
unbind tm global -policyName <policyname>
<!--NeedCopy-->
```

例

```
unbind tm global -policyName Traffic-Pol-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーからトラフィックポリシーをバインド解除するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
unbind lb vserver <name> -policy <policyname>

unbind cs vserver <name> -policy <policyname>
<!--NeedCopy-->
```

例

```
unbind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してトラフィックポリシーを削除するには

まず、グローバルからセッションポリシーをバインド解除し、コマンドプロンプトで次のように入力します。

```
rm tm trafficPolicy <name>
<!--NeedCopy-->
```

例

```
rm tm trafficPolicy Traffic-Pol-1
<!--NeedCopy-->
```

構成ユーティリティを使用してトラフィックポリシーを構成およびバインドするには

1. セキュリティ **>** **AAA-**アプリケーショントラフィック **>** トラフィックに移動します。
2. [セキュリティ **] > [AAA-**アプリケーショントラフィック **] > [**ポリシー **] > [**トラフィック] に移動します。
3. 詳細ウィンドウで、次のいずれかの操作を行います。

- 新しいセッションポリシーを作成するには、[追加] をクリックします。
- 既存のセッションポリシーを変更するには、ポリシーを選択し、[編集] をクリックします。

4. [トラフィックポリシーの作成] または [トラフィックポリシーの設定] ダイアログで、パラメータの値を指定します。
   - name*: policyName（以前に設定されたセッションポリシーでは変更できません。）
   - プロファイル *—アクション名
   - 式-規則 (式を入力するには、まず [式] テキスト領域の下にある左端のドロップダウンリストで式のタイプを選択し、次に式テキスト領域に式を直接入力するか、[追加] をクリックして [式の追加] ダイアログボックスを開き、その中のドロップダウンリストを使用して式を構築します。)
5. [作成] または [**OK**] をクリックします。作成したポリシーが、[**** セッションポリシーおよびプロファイル ****] ページの詳細ペインに表示されます。

フォームの **SSO** プロファイル

フォームベースの SSO を有効にして構成するには、最初に SSO プロファイルを作成します。

> 注
>
> - フォームが Javascript を含むようにカスタマイズされている場合、フォームベースのシングルサインオンは機能しません。
> - この機能では、「プロファイル」と「アクション」という用語は同じことを意味します。

コマンドラインインターフェイスを使用してフォーム **SSO** プロファイルを作成するには

コマンドプロンプトで入力します。

```
add tm formSSOAction <name> -actionURL <URL> -userField <string> -
    passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <
    string>] [-responsesize <positive_integer>][-nvtype ( STATIC |
    DYNAMIC )][-submitMethod ( GET | POST )]

show tm formSSOAction [<name>]
<!--NeedCopy-->
```

例

```
add tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-nameValuePair "loginID passwd" -responsesize "9096"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nvtype STATIC -submitMethod GET
 - sessTimeout 10 -defaultAuthorizationAction ALLOW
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用してフォーム **SSO** を変更するには

コマンドプロンプトで入力します。

```
set tm formSSOAction <name> -actionURL <URL> -userField <string> -
    passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <
    string>] [-responsesize <positive_integer>][-nvtype ( STATIC |
    DYNAMIC )][-submitMethod ( GET | POST )]
<!--NeedCopy-->
```

例

```
set tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nameValuePair "loginID passwd" -responsesize "9096"
-nvtype STATIC -submitMethod GET
 - sessTimeout 10 -defaultAuthorizationAction ALLOW
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してフォーム **SSO** プロファイルを削除するには

コマンドプロンプトで入力します。

```
rm tm formSSOAction <name>
<!--NeedCopy-->
```

例

```
rm tm sessionAction SSO-Prof-1
<!--NeedCopy-->
```

構成ユーティリティを使用してフォーム **SSO** プロファイルを構成するには

1. [セキュリティ] **>** **[AAA-**アプリケーショントラフィック] **>** **[**ポリシー] **>** **[**トラフィック] に移動します。
2. 詳細ウィンドウで、[フォーム **SSO** プロファイル] タブをクリックします。
3. [フォーム SSO プロファイル] タブで、次のいずれかの操作を行います。
   - 新しいフォーム SSO プロファイルを作成するには、[追加] をクリックします。
   - 既存のフォーム SSO プロファイルを変更するには、プロファイルを選択し、[編集] をクリックします。
4. 「フォーム **SSO** プロファイルの作成」または「フォーム **SSO** プロファイルの設定」ダイアログで、パラメーターの値を指定します。
   - name*: name（以前に設定されたセッションアクションでは変更できません）。
   - アクション URL*—actionURL
   - ユーザー名フィールド *—userField

- パスワードフィールド *—passField
- 式 * —ssoSuccessRule
- 名前と値のペア-ON, OFF
- レスポンスサイズ-responsesize
- 抽出-nvType
- 送信メソッド-submitMethod

5. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。作成したフォーム SSO プロファイルが、[トラフィックポリシー]、[プロファイル]、および [フォーム **SSO** プロファイル] ペインに表示されます。

SAML SSO プロファイル

SAML ベースの SSO を有効にして設定するには、まず SAML SSO プロファイルを作成します。

コマンドラインインターフェイスを使用して **SAML SSO** プロファイルを作成するには

コマンドプロンプトで入力します。

```
add tm samlSSOProfile <name> -samlSigningCertName <string> -
    assertionConsumerServiceURL <URL> -relaystateRule <expression> -
    sendPassword (ON | OFF) [-samlIssuerName <string>]
<!--NeedCopy-->
```

例

```
add tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example,
    Inc."  -assertionConsumerServiceURL "https://service.example.com" -
    relaystateRule "true"  -sendPassword "ON" -samlIssuerName "Example,
    Inc."
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **SAML SSO** を変更するには

コマンドプロンプトで入力します。

```
set tm samlSSOProfile <name> -samlSigningCertName <string> -
    assertionConsumerServiceURL <URL> -relaystateRule <expression> -
    sendPassword (ON | OFF) [-samlIssuerName <string>]
<!--NeedCopy-->
```

例

```
set tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example,
    Inc."  -assertionConsumerServiceURL "https://service.example.com" -
    relaystateRule "true"  -sendPassword "ON" -samlIssuerName "Example,
    Inc."
```



```
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **SAML SSO** プロファイルを削除するには

コマンドプロンプトで入力します。

```
rm tm samlSSOProfile <name>
<!--NeedCopy-->
```

例

```
rm tm sessionAction saml-SSO-Prof-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **SAML SSO** プロファイルを構成するには

1. [セキュリティ] **>** [**AAA-**アプリケーショントラフィック] **>** [ポリシー] **>** [トラフィック] に移動します。
2. 詳細ペインで、[**SAML SSO** プロファイル] タブをクリックします。
3. [**SAML SSO** プロファイル] タブで、次のいずれかの操作を行います。
   - 新しい SAML SSO プロファイルを作成するには、[追加] をクリックします。
   - 既存の SAML SSO プロファイルを変更するには、プロファイルを選択し、[**OpenEdit**] をクリックします。
4. [**SAML SSO** プロファイルの作成] または [**SAML SSO** プロファイルの構成] ダイアログボックスで、次のパラメータを設定します。
   - 名前 *
   - 署名証明書名 *
   - ACS URL*
   - リレーステートルール *
   - パスワードを送信
   - 発行者名
5. [作成] または [**OK**] をクリックし、[閉じる] をクリックします。作成した SAML SSO プロファイルが [トラフィックポリシー、プロファイル、および SAML SSO プロファイル] ペインに表示されます。

OWA 2010 のセッションタイムアウト

OWA 2010 の接続を、非アクティブ状態の一定期間後に強制的にタイムアウトできるようになりました。OWA は、タイムアウトを防ぐために、繰り返しキープアライブ要求をサーバーに送信します。接続を開いたままにしておくと、シングルサインオンを妨げることがあります。

コマンドラインインターフェイスを使用して **OWA 2010** を指定期間後に強制的にタイムアウトするには

コマンドプロンプトで、次のコマンドを入力します。

```
add tm trafficAction <actname> [-forcedTimeout <forcedTimeout> -
    forcedTimeoutVal <mins>]
<!--NeedCopy-->
```

<actname> の場合は、トラフィックポリシーの名前に置き換えます。<mins> の場合は、強制タイムアウトを開始するまでの分数に置き換えます。<forcedTimeout> の場合は、次の値のいずれかを代入します。

-**START**：タイマーがまだ開始されていない場合、強制タイムアウトのタイマーを開始します。実行中のタイマーが存在する場合、効果はありません。
-**STOP** —実行中のタイマーを停止します。実行中のタイマーが見つからない場合、は効果がありません。
-**RESET** —実行中のタイマーを再起動します。実行中のタイマーが見つからない場合、START オプションが使用されているかのようにタイマーを開始します。

```
add tm trafficPolicy <polname> <rule> <actname>
<!--NeedCopy-->
```

<polname> の場合は、トラフィックポリシーの名前に置き換えます。<rule> の場合は、NetScaler ADC 詳細ポリシーのルールを置き換えます。

```
bind lb vserver <vservername>  - policyName <name> -priority <number>
<!--NeedCopy-->
```

<vservername> の場合は、認証、認可、および監査トラフィック管理仮想サーバの名前を置き換えます。<priority> の場合は、ポリシーの優先度を指定する整数に置き換えます。

例

```
add tm trafficAction act-owa2010timeout -forcedTimeout RESET -
    forcedTimeoutVal 10
add tm trafficPolicy pol-owa2010timeout true act-owa2010timeout
bind lb vserver vs-owa2010 -policyName pol-owa2010timeout -priority 10
<!--NeedCopy-->
```

NetScaler Gateway のレート制限

April 15, 2024

NetScaler Gateway のレート制限機能を使用すると、NetScaler Gateway アプライアンス上の特定のネットワークエンティティまたは仮想エンティティの最大負荷を定義できます。NetScaler Gateway アプライアンスは認証されていないトラフィックをすべて消費するため、アプライアンスは頻繁にプロセス要求にさらされます。レート制限機能を使用すると、エンティティに関連するトラフィックのレートを監視し、トラフィックに基づいてリアルタイム

で予防措置を講じるように NetScaler Gateway アプライアンスを構成できます。NetScaler アプライアンスでのレート制限の仕組みについて詳しくは、「レート制限」を参照してください。

NetScaler には、予期しないレートが発生してもバックエンドサーバーを保護するレート制限機能があります。NetScaler の機能は、NetScaler Gateway が処理する認証されていないトラフィックを処理しなかったため、NetScaler Gateway には独自のレート制限機能が必要でした。これは、NetScaler Gateway アプライアンスがさらされているさまざまなソースからの予期しないリクエスト率を確認するために必要です。たとえば、認証されていない/ログイン/制御要求や、エンドユーザーまたはデバイスの検証のために公開される特定の API。

レート制限の一般的なユースケース

- URL からの 1 秒あたりのリクエスト数を制限します。
- リクエストがレート制限を超えた場合、特定のホストからのリクエストで受信した Cookie に基づいて接続をドロップします。
- 同じホスト（特定のサブネットマスク）から到着し、同じ宛先 IP アドレスを持つ HTTP 要求の数を制限します。

NetScaler Gateway のレート制限の設定

前提条件

設定済みの認証仮想サーバー。

注意事項

- 設定手順では、サンプルの制限 ID が設定されます。ストリームセレクタ、モードなどのサポートされているすべてのパラメータで同じように設定することができます。レート制限機能の完全な説明については、レート制限を参照してください。
- このポリシーは、次のように VPN 仮想サーバーにバインドすることもできます。次のコマンドを使用してポリシーをバインドするには、設定された VPN 仮想サーバーが必要です。

```
1  bind vpn vserver  -policy denylogin  -pri 1  -type aaa_request
2  <!--NeedCopy-->
```

- AAA_REQUEST は、レスポンダーポリシー用に新しく導入されたバインドポイントです。このバインドポイントで構成されたポリシーは、指定された仮想サーバーのすべての着信要求に適用されます。ポリシーは、他の処理の前に最初に非認証/制御トラフィックに対して処理されます。
- ポリシーを NetScaler Gateway 仮想サーバーにバインドすると、認証されていないリクエストを含め、NetScaler Gateway が消費するすべてのトラフィックの AAA_REQUEST バインドポイントでのレート制限が可能になります。

- ポリシーを認証仮想サーバレートにバインドすると、認証仮想サーバにヒットする非認証/制御要求が制限されます。

コマンドラインインターフェイスを使用してレート制限を構成するには、コマンドプロンプトで次のコマンドを入力します。

```
add limitIdentifier <limitIdentifier name> -threshold <positive_integer> -timeslice <positive_integer> -mode <mode type>
<!--NeedCopy-->
```

例:

```
add limitIdentifier limit_one_login -threshold 10 -timeslice 4294967290 -mode REQUEST_RATE
<!--NeedCopy-->
```

```
add responderaction denylogin respondwith  ‘ “HTTP/1.1 200 OK\r\n\r\n” + “Request is denied due to unusual rate” ’
<!--NeedCopy-->
```

```
add responder policy denylogin 'sys.check_limit("limit_one_login")' denylogin
<!--NeedCopy-->
```

```
bind authentication vserver <vserver name> -policy denylogin -pri 1 -type aaa_request
<!--NeedCopy-->
```

例:

```
bind authentication vserver authvserver -policy denylogin -pri 1 -type aaa_request
<!--NeedCopy-->
```

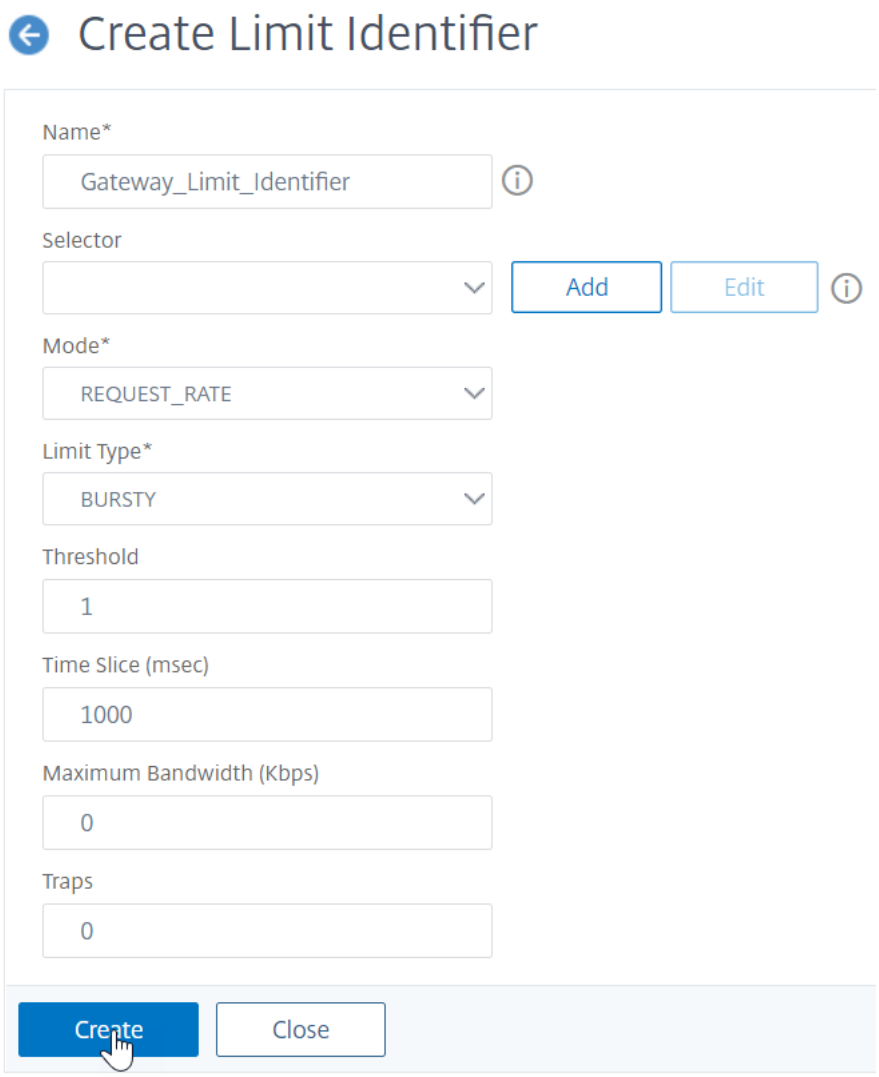
パラメータの説明

- **limitIdentifier**: レート制限識別子の名前。ASCII 文字またはアンダースコア (_) 文字で始まり、ASCII 英数字またはアンダースコア文字のみで構成する必要があります。予約語は使用できません。これは必須の議論です。最大長: 31
- **threshold-** 要求 (モードが REQUEST_RATE として設定されている) がタイムスライスごとに追跡される場合に、指定されたタイムスライスで許可される要求の最大数。接続 (モードが CONNECTION として設定されている) が追跡される場合、通過する接続の合計数になります。デフォルト値:1 最小値:1 最大値:4294967295
- **timeSlice-** 10 の倍数で指定された時間間隔（ミリ秒単位)。この間は、要求がしきい値を超えるかどうかをチェックするために追跡されます。引数は、モードが REQUEST_RATE に設定されている場合にのみ必要です。デフォルト値:1000 最小値:10 最大値:4294967295

- **mode-** 追跡するトラフィックのタイプを定義します。
  - REQUEST_RATE-リクエスト/タイムスライスを追跡します。
  - CONNECTION-アクティブなトランザクションを追跡します。

NetScaler GUI を使用してレート制限を設定するには:

1. 「**AppExpert**」**>**「レート制限」**>**「制限識別子」の順に選択し、「追加」をクリックして、CLI セクションで指定されている関連の詳細を指定します。



2. [**AppExpert**] **>** [レスポンダー] **>** [ポリシー] に移動します。[レスポンダーポリシー] ページで、[追加] をクリックします。
3. [レスポンダーポリシーの作成] ページで、制限識別子を持つレスポンダーアクションを含むレスポンダーポリシーを作成します。
4. レスポンダーアクションを作成するには、アクションの横にある [追加] をクリックします。

5. [レスポンダーアクションの作成] ページで、レスポンダーアクションの名前を入力します。
6. ドロップダウンメニューからタイプとして [返信する] を選択します。
7. エクスプレッションエディターで、レスポンスメッセージを設定します。例: `"HTTP/1.1 200 OK\r\n\r\n"+ "Request is denied due to unusual rate"`。
8. [**Create**] をクリックします。
9. レスポンダーポリシーを作成するには、[レスポンダーポリシーの作成] ページで、レスポンダーポリシーの名前を入力します。
10. 式エディターで、レスポンダーポリシーの条件を設定します。例: `'sys.check_limit("limit_one_login")'`。
11. [**Create**] をクリックします。
12. レスポンダーポリシーを認証仮想サーバーにバインドします。
    - [セキュリティ] **> [AAA** アプリケーショントラフィック] **> [**仮想サーバー] に移動します。
    - 仮想サーバを選択します。
    - ポリシーを追加します。
    - サーバーにバインドするレスポンダーポリシーを選択し、優先度を設定します。
    - タイプとして **AAA-REQUEST** を選択し、[続行] をクリックします。

> 注:
>
> VPN 仮想サーバーの AAA_REQUEST バインドポイントでレート制限を有効にすることもできます。

NetScaler Gateway にレート制限を適用する一般的なユースケースの構成

次に、一般的なユースケースを設定するコマンドの例を示します。

- URL からの 1 秒あたりのリクエスト数を制限します。

```
1  add stream selector ipStreamSelector http.req.url “client.ip.src
       ”
2
3  add ns limitIdentifier ipLimitIdentifier -threshold 4 -
       timeslice 1000 -mode request_rate -limitType smooth -
       selectorName ip StreamSelector
4
5  add responder policy ipLimitResponderPolicy “http.req.url.
       contains(\” myasp.asp\” ) && sys.check_limit(\”
       ipLimitIdentifier\” )” myWebSiteRedirectAction
6
7  bind authentication virtual server authvserver -policy denylogin
       -pri 1 -type aaa_request
8  <!--NeedCopy-->
```


- リクエストがレート制限を超えた場合、www.yourcompany.comからのリクエストで受信した Cookie に基づいて接続をドロップします。

```
add stream selector cacheStreamSelector  “http.req.cookie.value(\” mycookie\” )”  “client.ip.src.subnet(24)”

add ns limitIdentifier myLimitIdentifier  -Threshold 2  -timeSlice 3000  -selectorName reqCookieStreamSelector

add responder action sendRedirectURL redirect `"http://www.mycompany.com"` + http.req.url'

add responder policy rateLimitCookiePolicy

 “http.req.url.contains(\www.yourcompany.com) && sys.check_limit(\” myLimitIdentifier\” )”  sendRedirectUrl

<!--NeedCopy-->
```

- 同じホスト（サブネットマスクが 32）から着信し、同じ宛先 IP アドレスを持つ HTTP 要求の数を制限します。

```
add stream selector ipv6_sel  “CLIENT.IPv6.src.subnet(32)” CLIENT.IPv6.dst

add ns limitIdentifier ipv6_id  -imeSlice 20000  -selectorName ipv6_sel

add lb vserver ipv6_vip HTTP 3ffe:: 209 80  -persistenceType NONE  -cltTime

add responder action redirect_page redirect  “\” `http://redirectpage.com/\” "`

add responder policy ipv6_resp_pol  “SYS.CHECK_LIMIT(\” ipv6_id\” )”  redirect_page

bind responder global ipv6_resp_pol 5 END  -type DEFAULT
<!--NeedCopy-->
```

認証、承認、監査エンドポイントのバインドレスポンダーポリシー

パケットエンジンが要求を処理できるようにするには、AAA_REQUESTバインドポイントを使用してレスポンダーポリシーを認証仮想サーバーまたは VPN 仮想サーバーにバインドします。

- **VPN** 仮想サーバーの例**:** `bind vpn vserver vpnVs -policy resp_pol -priority 6 -type AAA_REQUEST`
- 認証仮想サーバーの例**:** `bind authentication vserver av_vs -policy resp_pol -priority 6 -type AAA_REQUEST`

Apache HTTP サーバーが要求を処理できるようにするには、バインドポイントREQUESTを使用してレスポンダーポリシーを認証仮想サーバーまたは VPN 仮想サーバーにバインドします。

例:

- **VPN** 仮想サーバーの例**:** `bind vpn vserver vpnVs -policy resp_pol -priority 6 -type REQUEST`
- 認証仮想サーバーの例**:** `bind authentication vserver av_vs -policy resp_pol -priority 6 -type REQUEST`

レスポンダーポリシーのバインドについて詳しくは、「レスポンダーポリシーのバインド」を参照してください。

アプリケーションリソースへのユーザーアクセスの承認

August 15, 2023

認証されたユーザーがアプリケーション内でアクセスできるリソースを制御できます。

そのためには、権限付与ポリシーを各ユーザーに関連付けます。この方法は、個別に、またはポリシーをユーザーのグループに関連付ける方法です。承認ポリシーでは以下を指定する必要があります。

- ルール。アクセスを許可する必要があるリソース。これは、基本式または高度な式を使用して指定できます。
- アクション。リソースへのアクセスを許可する必要があるか、拒否する必要があるか。

デフォルトでは、アプリケーション内のすべてのリソースへのアクセスはすべてのユーザーに拒否されます。ただし、このデフォルトの承認アクションを（セッションプロファイルでセッションパラメータを設定するか、グローバルセッションパラメータを設定して）すべてのユーザーへのアクセスを許可するように変更できます。

> 警告
>
> セキュリティを最適化するために、Citrix ではデフォルトの承認アクションを「拒否」から「許可」に変更しないことをお勧めします。代わりに、特定のリソースへのアクセスを必要とするユーザー向けに特定の承認ポリシーを作成することをお勧めします。

CLI を使用して承認を設定するには

1. 承認ポリシーを設定します。

   ns-cli-prompt> **add authorization policy <name> <rule> <action>**

2. ポリシーを適切なユーザーまたはグループに関連付けます。

   - ポリシーを特定のユーザーにバインドします。

     ns-cli-prompt> **bind aaa user <username> -policy** <policyname>

- ポリシーを特定のグループにバインドします。

  ns-cli-prompt> **bind aaa group <groupName> -policy** <policyname>

GUI ([設定] タブ) を使用して認証を設定するには

1. 承認ポリシーを作成します。

   [セキュリティ] **>** [**AAA-**アプリケーショントラフィック] **>** [ポリシー] **>** [許可] に移動し、[追加] をクリックし、必要に応じてポリシーを定義します。

2. ポリシーを適切なユーザーまたはグループに関連付けます。

   [セキュリティ] **>** [**AAA-**アプリケーショントラフィック] **>** [** ユーザまたはグループ **] に移動し、関連するユーザまたはグループを編集して認可ポリシーに関連付けます。

認証構成の例

ここでは、一部のアプリケーションリソースへのユーザーアクセスを許可する設定例をいくつか示します。これらは CLI コマンドであることに注意してください。GUI を使用して同様の設定を行うことができますが、式を引用符 (“) で囲んではいけません。

- “‘
  add authorization policy authzpol1 “HTTP.REQ.URL.SUFFIX.EQ(“gif”)” ALLOW
  “‘
  - ```
    bind aaa user user1 -policy authzpol1
    ```
- “‘
  add authorization policy authzpol2 “HTTP.REQ.URL.SUFFIX.EQ(“png”)” DENY
  “‘
  - ```
    bind aaa group group1 -policy authzpol2
    <!--NeedCopy-->
    ```

認証済みセッションの監査

August 15, 2023

認証されたセッションでトリガーされたすべてのイベントのログを保持するように NetScaler アプライアンスを構成できます。この情報を使用して、状態とステータス情報を監査し、ユーザーの履歴を時系列で表示できます。

そのためには、以下を指定する監査ポリシーを定義します。

- ログタイプ。ログはリモート（syslog）または NetScaler アプライアンス（nslog）にローカルに保存できます。
- ルール。ログが保存される条件。
- アクション。ログサーバーの詳細、およびログエントリを作成するためのその他の詳細。

この監査ポリシーは、ユーザーレベル、グループレベル、認証、承認、監査仮想サーバー、グローバルシステムレベルなど、さまざまなレベルで構成できます。ユーザーレベルで設定されたポリシーが最も優先されます。

> 注
>
> このトピックでは、syslog を使用する手順を詳しく説明します。nslog を使用するには、必要な変更を行います。

CLI を使用して Syslog 監査を設定するには

1. 関連するログ設定を使用して監査サーバーを構成します。

   ns-cli-prompt> **add audit syslogAction <name> <serverIP>** …

2. 監査サーバーを関連付けて監査ポリシーを設定します。

   ns-cli-prompt> **add audit syslogPolicy <name> <rule> <action>**

3. 監査ポリシーを次のエンティティのいずれかに関連付けます。

   - ポリシーを特定のユーザーにバインドします。

     ns-cli-prompt> **bind aaa user <userName>-policy** <policyname> …

   - ポリシーを特定のグループにバインドします。

     ns-cli-prompt> **bind aaa group <groupName>-policy** <policyname> …

   - ポリシーを認証、承認、および監査仮想サーバーにバインドします。

     ns-cli-prompt> **bind authentication vserver <name> -policy** <policyname> …

   - ポリシーを NetScaler アプライアンスにグローバルにバインドします。

     ns-cli-prompt> **bind tm global -policyName** <policyname> …

GUI (設定タブ) を使用して Syslog 監査を設定するには

1. 監査サーバーとポリシーを設定します。

   [セキュリティ] **> [AAA-**アプリケーショントラフィック] **> [**ポリシー] **> [**監査] **> [Syslog**] に移動し、関連するタブでサーバとポリシーを設定します。

2. ポリシーを次のいずれかに関連付けます。

- ポリシーを特定のユーザーにバインドします。

  [セキュリティ] **> [AAA-**アプリケーショントラフィック**] > [**ユーザ] に移動し、承認ポリシーを該当するユーザに関連付けます。

- ポリシーを特定のグループにバインドします。

  [セキュリティ] **> [AAA-**アプリケーショントラフィック**] > [**グループ] に移動し、承認ポリシーを関連するグループに関連付けます。

- ポリシーを認証、承認、および監査仮想サーバーにバインドします。

  [セキュリティ] **> [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] に移動し、承認ポリシーを関連する仮想サーバに関連付けます。

- ポリシーを NetScaler アプライアンスにグローバルにバインドします。

  [セキュリティ] **> [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**監査**] > [Syslog**] または [**Nslog**] に移動し、承認ポリシーを選択し、[アクション**] > [**グローバルバインディング**]** をクリックしてポリシーをグローバルにバインドします。

Active Directory フェデレーションサービスプロキシとしての **NetScaler**

August 15, 2023

Active Directory フェデレーションサービス (ADFS) は、Active Directory 認証を受けたクライアントがエンタープライズデータセンター外のリソースにシングルサインオン (SSO) できるようにする Microsoft のサービスです。ADFS サーバーファームにより、内部ユーザーは外部のクラウドホストサービスにアクセスできます。しかし、外部ユーザーが参加した瞬間に、外部ユーザーにリモート接続し、フェデレーション ID を介してクラウドベースのサービスにアクセスする方法を提供する必要があります。ほとんどの企業は、ADFS サーバを DMZ に公開したままにすることを望んでいません。そのため、ADFS プロキシは、リモートユーザー接続とアプリケーションアクセスにおいて重要な役割を果たします。

10 年以上にわたり、NetScaler アプライアンスはリモートユーザー接続とアプリケーションアクセスにおいて同様の役割を果たしてきました。NetScaler アプライアンスは、次のサービスを可能にする新しい ADFS 実装をサポートするための ADFS プロキシとして使用するソリューションとして推奨されています。

- 安全な接続。
- フェデレーション ID の認証と処理。

SAML IdP としての NetScaler について詳しくは、「SAML IdP としての NetScaler」を参照してください。

ADFS プロキシの利点

- DMZ の設置面積を縮小し、ほとんどの企業のニーズに応えます。

- エンドユーザーに SSO エクスペリエンスを提供します。
- 豊富な事前認証方法をサポートし、多要素認証を可能にします。
- アクティブクライアントとパッシブクライアントの両方をサポートします。

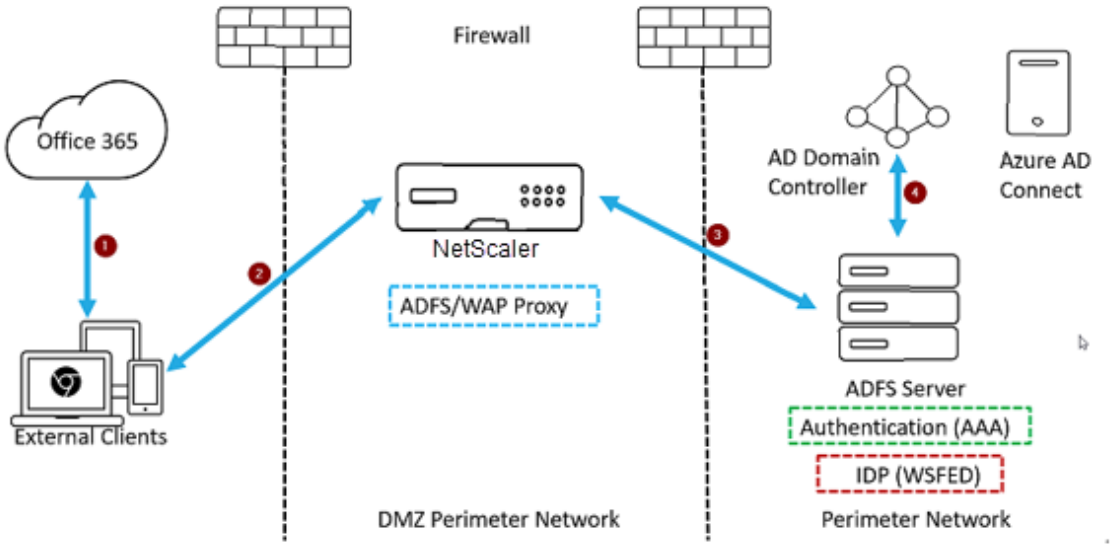
NetScaler を ADFS プロキシとして使用するための前提条件

NetScaler アプライアンスを ADFS プロキシとして構成する前に、次の前提条件が満たされていることを確認してください。

- 12.1 ビルド以降の NetScaler アプライアンス。
- ドメイン ADFS サーバー。
- ドメイン SSL 証明書。
- コンテンツスイッチング仮想サーバーの仮想 IP。
- NetScaler アプライアンスの負荷分散、SSL オフロード、コンテンツスイッチング、書き換え、認証、承認、監査のトラフィック管理機能を有効にします。

NetScaler アプライアンスを ADFS プロキシとして構成する

このユースケースを実現するには、NetScaler を DMZ ゾーンの ADFS プロキシとして構成します。ADFS サーバーは、バックエンドの AD ドメインコントローラーとともに構成されます。



1. Microsoft Office 365 にアクセスするクライアントリクエストは、ADFS プロキシとして展開された NetScaler にリダイレクトされます。
2. ユーザーの資格情報は ADFS サーバーに渡されます。
3. ADFS サーバーは、ドメインのオンプレミス AD で認証情報を認証します。

4. ADFS サーバーは、AD による認証情報の検証が成功すると、セッションを確立するために Microsoft Office365 に渡されるトークンを生成します。

ADFS プロキシとして構成する前に NetScaler アプライアンスを構成する場合の基本的な手順は次のとおりです。

NetScaler のコマンドプロンプトで、次のコマンドを入力します。

1. バックエンドの SSL プロファイルを作成し、SSL プロファイルで SNI を有効にします。SSLv3/TLS1 を無効にします。

   ```
   add ssl profile <new SSL profile> -sslprofileType backEnd -
   sniEnable ENABLED -ssl3 DISABLED -tls1 DISABLED -commonName <
   FQDN of ADFS>
   ```

2. サービスの SSLv3/TLS1 を無効にします。

   ```
   set ssl service <adfs service name> -sslProfile <SSL profile
   created in the above step>
   ```

3. バックエンドサーバーハンドシェイクの SNI 拡張を有効にします。

   - `set vpn parameter -backendServerSni ENABLED`
   - `set ssl parameter -denySSLReneg NONSECURE`

CLI を使用して **NetScaler** アプライアンスを **ADFS** プロキシとして構成する

以下のセクションは、構成手順を完了するための要件に基づいて分類されています。

ADFS サービスを設定するには

1. NetScaler for ADFS サーバーで ADFS サービスを構成します。

   ```
   add service <Domain_ADFS_Service> <ADFS Server IP> SSL 443 -gslb
   NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport
   YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP
    NO
   ```

   例

   add service CTXTEST_ADFS_Service 1.1.1.1 SSL 443 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO

2. コンテンツスイッチ仮想サーバーの FQDN を構成し、SNI を有効にします。

   ```
   set ssl service <Domain_ADFS_Service> -SNIEnable ENABLED -commonName
    <sts.domain.com>
   ```


例

set ssl service CTXTEST_ADFS_Service -SNIEnable ENABLED -commonName sts.ctxtest.com

ADFS 負荷分散仮想サーバーを構成するには

> 重要
>
> トラフィックをセキュリティで保護するには、ドメイン SSL 証明書 (SSL_CERT) が必要です。

1. ADFS 負荷分散仮想サーバーを設定します。

```
add lb vserver <Domain_ADFS_LBVS> SSL <IP_address> -persistenceType
 NONE -cltTimeout 180
```

例

```
add lb vserver CTXTEST_ADFS_LBVS SSL 192.168.1.0 -persistenceType
 NONE -cltTimeout 180
```

2. ADFS 負荷分散仮想サーバーを ADFS サービスにバインドします。

```
bind lb vserver <Domain_ADFS_LBVS> <Domain_ADFS_Service>
```

例

```
bind lb vserver CTXTEST_ADFS_LBVS CTXTEST_ADFS_Service
```

3. SSL 仮想サーバーの証明書とキーのペアをバインドします。

```
bind ssl vserver <Domain_ADFS_LBVS> -certkeyName <SSL_CERT>
```

例

```
bind ssl vserver CTXTEST_ADFS_LBVS -certkeyName ctxtest_newcert_2019
```

ドメイン用のコンテンツスイッチ仮想サーバーを構成するには

> 注:
>
> コンテンツスイッチング仮想サーバーには、パブリック IP に NAT された空いている仮想 IP (2.2.2.2 など) が 1 つ必要です。外部トラフィックと内部トラフィックの両方からアクセス可能でなければなりません。

1. 無料の VIP でコンテンツスイッチング仮想サーバーを作成します。

```
add cs vserver <Domain_CSVS> SSL <FREE VIP> 443 -cltTimeout 180 -
persistenceType NONE
```

例

```
add cs vserver CTXTEST_CSVS SSL 2.2.2.2 443 -cltTimeout 180 -
persistenceType NONE
```


2. コンテンツスイッチ仮想サーバーを負荷分散仮想サーバーにバインドします。

   ```
   bind cs vserver <Domain_CSVS> -lbvserver <Domain_ADFS_LBVS>
   ```

   例

   - `bind cs vserver CTXTEST_CSVS -lbvserver CTXTEST_ADFS_LBVS`
   - `set ssl vserver CTXTEST_CSVS -sessReuse DISABLED`

3. SSL 仮想サーバーの証明書とキーのペアをバインドします。

   ```
   bind ssl vserver <Domain_CSVS> -certkeyName <SSL_CERT>
   ```

   例

   ```
   bind ssl vserver CTXTEST_CSVS -certkeyName ctxtest_newcert_2019
   ```

サポートされているプロトコル

Microsoft が提供するプロトコルは、NetScaler アプライアンスとの統合において重要な役割を果たします。ADFS プロキシとしての NetScaler は、次のプロトコルをサポートしています。

- **WS** フェデレーション。詳細については、「Web サービスフェデレーションプロトコル」を参照してください。
- **ADFSPIP**。詳細については、「Active Directory フェデレーションサービスプロキシ統合プロトコルコンプライアンス」を参照してください。

> 注
>
> NetScaler アプライアンスは、ADFS プロキシとして展開した場合、デバイス証明書認証をサポートしません。

Web サービスフェデレーションプロトコル

August 15, 2023

Web サービスフェデレーション (WS-Federation) は、2 つのドメイン間に信頼関係がある場合に、ある信頼ドメインのセキュリティトークンサービス (STS) が別の信頼ドメインの STS に認証情報を提供できるようにするアイデンティティプロトコルです。

WS-フェデレーションの利点

WS フェデレーションはアクティブクライアントとパッシブクライアントの両方をサポートしますが、SAML IdP はパッシブクライアントのみをサポートします

- アクティブクライアントは、Outlook などの Microsoft ネイティブクライアントと Office クライアント (Word、PowerPoint、Excel、および OneNote) です。
- パッシブクライアントは、Google Chrome、Mozilla Firefox、Internet Explorer などのブラウザベースのクライアントです。

NetScaler を **WS** フェデレーションとして使用するための前提条件

NetScaler アプライアンスを ADFS プロキシとして構成する前に、以下を確認してください。

- Active Directory
- ドメイン SSL 証明書。
- ADFS サーバー上の NetScaler SSL 証明書と ADFS トークン署名証明書は同じである必要があります。

> 重要
>
> SAML IdP は WS フェデレーションプロトコルを処理できるようになりました。したがって、WS-Federation IdP を設定するには、実際に SAML IdP を設定する必要があります。WS-Federation について明示的に言及しているユーザーインターフェイスは表示されません。

ADFS プロキシおよび **WS-**フェデレーション **IdP** として構成されている場合に **NetScaler** がサポートする機能

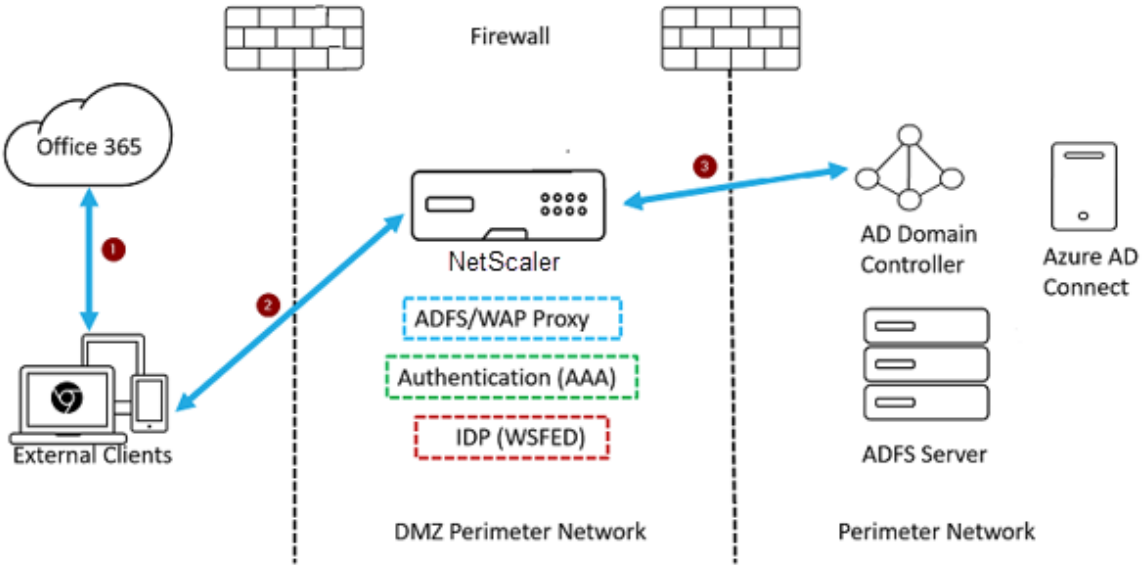
次の表は、ADFS プロキシおよび WS-Federation IdP として構成された場合に NetScaler アプライアンスがサポートする機能を示しています。

Features	NetScaler アプライアンスを ADFS プロキシとして構成する	WS-フェデレーション IdP としての NetScaler	ADFSPIP としての NetScaler ADC
負荷分散	はい	はい	はい
SSL 終了	はい	はい	はい
レート制限	はい	はい	はい
統合 (DMZ サーバの設置面積を削減し、パブリック IP を節約)	はい	はい	はい
Web アプリケーションファイアウォール（WAF）	はい	はい	はい
NetScaler アプライアンスへの認証オフロード	はい	はい (アクティブクライアントとパッシブクライアント)	はい

Features	NetScaler アプライアンスを ADFS プロキシとして構成する	WS-フェデレーション IdP としての NetScaler	ADFSPIP としての NetScaler ADC
シングルサインオン（SSO）	はい	はい (アクティブクライアントとパッシブクライアント)	はい
多要素（nFactor）認証	いいえ	はい (アクティブクライアントとパッシブクライアント)	はい
Azure 多要素認証	いいえ	はい (アクティブクライアントとパッシブクライアント)	はい
ADFS サーバーファームは避けられる	いいえ	はい	はい

NetScaler アプライアンスを WS-フェデレーション IdP として構成する

NetScaler を DMZ ゾーンの WS フェデレーション IdP（SAML IdP）として構成します。ADFS サーバーは、バックエンドの AD ドメインコントローラーとともに構成されます。



1. Microsoft Office365 へのクライアントリクエストは、NetScaler アプライアンスにリダイレクトされます。
2. ユーザーは、多要素認証の資格情報を入力します。
3. NetScaler は AD を使用して資格情報を検証し、NetScaler アプライアンス上でネイティブにトークンを生成します。資格情報は、アクセスのために Office365 に渡されます。

> 注
>
> F5 Networks のロードバランサーと比較すると、WS-Federation IdP サポートは NetScaler アプライアンスを介してネイティブに行われます。

CLI を使用して NetScaler アプライアンスを WS-フェデレーション IdP（SAML IdP）として構成する

以下のセクションは、構成手順を完了するための要件に基づいて分類されています。

LDAP 認証を設定してポリシーを追加するには

> 重要
>
> ドメインユーザーが会社のメールアドレスを使用して NetScaler アプライアンスにログオンするには、以下を構成する必要があります。

- NetScaler アプライアンスで LDAP 認証サーバーとポリシーを構成します。
- 認証、承認、および監査用の仮想 IP アドレスにバインドします (既存の LDAP 設定の使用もサポートされています)。

```
1 add authentication ldapAction <Domain_LDAP_Action> -serverIP <Active
     Directory IP> -serverPort 636 -ldapBase "cn=Users,dc=domain,dc=com"
     -ldapBindDn "cn=administrator,cn=Users,dc=domain,dc=com" -
     ldapBindDnPassword <administrator password> -encrypted -
     encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName
      memberOf -subAttributeName cn -secType SSL -ssoNameAttribute
     UserPrincipalName -followReferrals ON -Attribute1 mail -Attribute2
     objectGUID
2
3 add authentication Policy <Domain_LDAP_Policy> -rule true -action <
     Domain_LDAP_Action>
4 <!--NeedCopy-->
```

例

```
1 add authentication ldapAction CTXTEST_LDAP_Action -serverIP 3.3.3.3 -
     serverPort 636 -ldapBase "cn=Users,dc=ctxtest,dc=com" -ldapBindDn "
     cn=administrator,cn=Users,dc=ctxtest,dc=com" -ldapBindDnPassword
     xxxxxxxxxx -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName
     sAMAccountName -groupAttrName memberOf -subAttributeName cn -secType
      SSL -ssoNameAttribute UserPrincipalName -followReferrals ON -
     Attribute1 mail -Attribute2 objectGUID
2
3 add authentication Policy CTXTEST_LDAP_Policy -rule true -action
     CTXTEST_LDAP_Action
4 <!--NeedCopy-->
```


NetScaler を **WS-**フェデレーション **IdP** または **SAML IdP** として構成するには

トークン生成のための WS-Federation IdP (SAML IdP) アクションとポリシーを作成します。後で認証、承認、および監査仮想サーバーにバインドします。

```
add authentication samlIdPProfile <Domain_SAMLIDP_Profile> -
    samlIdPCertName <SSL_CERT> -assertionConsumerServiceURL "https://
    login.microsoftonline.com/login.srf" -samlIssuerName <Issuer Name
    for Office 365 in ADFS Server> -rejectUnsignedRequests OFF -audience
     urn:federation:MicrosoftOnline -NameIDFormat persistent -NameIDExpr
     "HTTP.REQ.USER.ATTRIBUTE(2).B64ENCODE" -Attribute1 IDPEmail -
    Attribute1Expr "HTTP.REQ.USER.ATTRIBUTE(1)"

add authentication samlIdPPolicy <Domain_SAMLIDP_Policy> -rule "HTTP.
    REQ.HEADER("referer").CONTAINS("microsoft") || true" -action <
    Domain_SAMLIDP_Profile>
<!--NeedCopy-->
```

例

```
add authentication samlIdPProfile CTXTEST_SAMLIDP_Profile -
    samlIdPCertName ctxtest_newcert_2019 -assertionConsumerServiceURL "
    https://login.microsoftonline.com/login.srf" -samlIssuerName "http
    ://ctxtest.com/adfs/services/trust/" -rejectUnsignedRequests OFF -
    audience urn:federation:MicrosoftOnline -NameIDFormat persistent -
    NameIDExpr "HTTP.REQ.USER.ATTRIBUTE(2).B64ENCODE" -Attribute1
    IDPEmail -Attribute1Expr "HTTP.REQ.USER.ATTRIBUTE(1)"

add authentication samlIdPPolicy CTXTEST_SAMLIDP_Policy -rule "HTTP.REQ
    .HEADER("referer").CONTAINS("microsoft") || true" -action
    CTXTEST_SAMLIDP_Profile
<!--NeedCopy-->
```

企業の資格情報を使用して **Office365** にログオンする従業員を認証するように、認証、承認、および監査仮想サーバーを構成するには

```
add authentication vserver <Domain_AAA_VS> SSL <IP_address>`
<!--NeedCopy-->
```

例

```
add authentication vserver CTXTEST_AAA_VS SSL 192.168.1.0

bind authentication vserver CTXTEST_AAA_VS -portaltheme RfWebUI
<!--NeedCopy-->
```

認証仮想サーバーとポリシーをバインドするには


```
bind authentication vserver <Domain_AAA_VS> -policy <
    Domain_SAMLIDP_Policy> -priority 100 -gotoPriorityExpression NEXT

bind authentication vserver <Domain_AAA_VS> -policy <Domain_LDAP_Policy
    > -priority 100 -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

例

```
bind authentication vserver CTXTEST_AAA_VS -policy
    CTXTEST_SAMLIDP_Policy -priority 100 -gotoPriorityExpression NEXT

bind authentication vserver CTXTEST_AAA_VS -policy CTXTEST_LDAP_Policy
    -priority 100 -gotoPriorityExpression NEXT

bind ssl vserver CTXTEST_AAA_VS -certkeyName ctxtest_newcert_2019
<!--NeedCopy-->
```

コンテンツスイッチングを設定するには

```
add cs action <Domain_CS_Action> -targetVserver <Domain_AAA_VS>

add cs policy <Domain_CS_Policy> -rule "is_vpn_url || http.req.url.
    contains("/adfs/ls") || http.req.url.contains("/adfs/services/trust"
    ) || -action <Domain_CS_Action>
<!--NeedCopy-->
```

例

```
add cs action CTXTEST_CS_Action -targetVserver CTXTEST_AAA_VS

add cs policy CTXTEST_CS_Policy -rule "is_vpn_url || http.req.url.
    contains("/adfs/ls") || http.req.url.contains("/adfs/services/trust"
    ) || -action CTXTEST_CS_Action
<!--NeedCopy-->
```

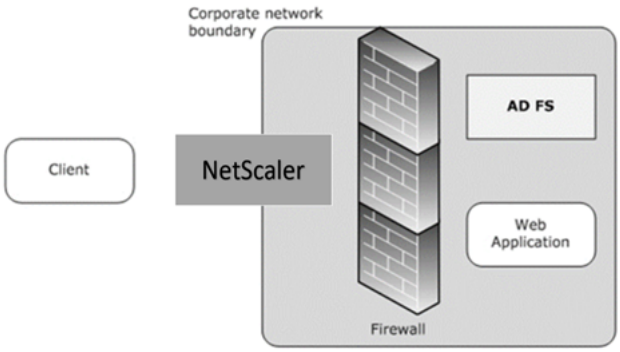
コンテンツスイッチ仮想サーバーをポリシーにバインドするには

```
bind cs vserver CTXTEST_CSVS -policyName CTXTEST_CS_Policy -priority
    100
<!--NeedCopy-->
```


Active Directory フェデレーションサービスプロキシ統合プロトコルへの準拠

January 9, 2024

Web アプリケーションプロキシの代わりにサードパーティプロキシを使用する場合は、ADFS と WAP の統合ルールを指定する MS-ADFSPIP プロトコルをサポートしている必要があります。ADFSPIP は Active Directory フェデレーションサービスを認証およびアプリケーションプロキシと統合し、企業ネットワークの境界の外側にあるクライアントが、企業ネットワークの境界内にあるサービスにアクセスできるようにします。



前提条件

プロキシサーバーと ADFS ファーム間の信頼を正常に確立するには、NetScaler アプライアンスの次の構成を確認してください。

- バックエンドの SSL プロファイルを作成し、SSL プロファイルで SNI を有効にします。SSLv3/TLS1 を無効にします。コマンドプロンプトで、次のコマンドを入力します:

```
add ssl profile <new SSL profile> -sniEnable ENABLED -ssl3
    DISABLED -tls1 DISABLED -commonName <FQDN of ADFS>
<!--NeedCopy-->
```

- サービスの SSLv3/TLS1 を無効にします。コマンドプロンプトで、次のコマンドを入力します:

```
set ssl profile <adfs service name> -sessReuse <ENABLED|DISABLED>
     -tls1 <ENABLED|DISABLED> -SNIEnable <ENABLED|DISABLED> -
    commonName <name> -denySSLReneg <YES|NO>
<!--NeedCopy-->
```

```
>**重要**
>
> 認証をADFSサーバーにオフロードする必要があるホームレルム検出（HRD）シナリオでは、NetScalerアプライアンスで認証とSSOの両方を無効にすることをお勧めします。

```



```
## 認証メカニズム

認証のイベントフローの概要を次に示します。

1.  **ADFSサーバーとの信頼の確立** — NetScalerサーバーは、クライアント証明書を登録することによってADFSサーバーとの信頼を確立します。信頼が確立されると、NetScalerアプライアンスは再起動後にユーザーの介入なしに信頼を再確立します。

    証明書の有効期限が切れたら、ADFS プロキシプロファイルを削除して再度追加して、信頼を再確立する必要があります。

1.  **公開エンドポイント** -NetScalerアプライアンスは、信頼確立後にADFSサーバー上の公開エンドポイントのリストを自動的に取得します。これらの公開されたエンドポイントは、ADFS サーバーに転送される要求をフィルタリングします。

1.  **クライアント要求へのヘッダーの挿入** — NetScalerアプライアンスがクライアント要求をトンネリングすると、ADFSサーバーへの送信中にADFSPIPに関連するHTTPヘッダーがパケットに追加されます。ADFS サーバーでは、これらのヘッダー値に基づいてアクセス制御を実装できます。次のヘッダーがサポートされています。

    - X-MSプロキシ
    - X-MS-エンドポイント-絶対パス
    - X-MS 転送クライアント IP
    - X-MSプロキシ
    - X-MS-ターゲット-ロール
    - X-MS-ADFS-プロキシ-クライアント-IP

1.  **エンドユーザートラフィックの管理** — エンドユーザートラフィックは、目的のリソースに安全にルーティングされます。

    > **注： **
    >
    >- NetScalerはフォームベースの認証を使用します。
    >
    >- NetScalerは、Active Directory フェデレーションサービスのプロキシ統合プロトコルコンプライアンスを使用したアプリケーションの公開をサポートしていません。

## ADFSサーバーをサポートするようにNetScalerを構成する

### 前提条件

- Context Switching（CS; コンテキストスイッチング）サーバをフロントエンドとして設定し、CS の背後に認証、許可、および監査サーバコマンドプロンプトで入力します:

    ```
    add cs vserver cs_adfs_proxy SSL a.b.c.d 443 -cltTimeout 180 -AuthenticationHost adfs.ns.com
    <!--NeedCopy-->
```


```

```
```
add cs action adfs_proxy_cs_act -targetVserver avs_adfs_proxy
<!--NeedCopy--> ```

```
add cs policy adfs_proxy_policy -rule is_vpn_url -action
    adfs_proxy_cs_act
<!--NeedCopy--> ```

```
bind cs vserver cs_adfs_proxy -policyName adfs_proxy_policy -priority
    100
<!--NeedCopy--> ```

```
bind cs vserver cs_adfs_proxy -lbvserver lb_adfs_proxy
<!--NeedCopy--> ```
```

- ADFS サービスを追加します。コマンドプロンプトで入力します:

```
add service adfs_service adfs_server SSL 443

set ssl service adfs_service -sslProfile
    ns_default_ssl_profile_backend

bind ssl service adfs_service -certkeyName adfs_trust_cert
<!--NeedCopy-->
```

```
set ssl profile ns_default_ssl_profile_backend -sessReuse ENABLED
     -tls1 DISABLED -SNIEnable ENABLED -commonName www.server.com
    -denySSLReneg NO
<!--NeedCopy-->
```

- 負荷分散仮想サーバーを追加します。コマンドプロンプトで入力します:

```
add lb vserver lb_adfs_proxy SSL 0.0.0.0 0 -AuthenticationHost
    adfs.ns.com -Authentication ON -adfsProxyProfile
    adfs_proxy_profile
<!--NeedCopy-->
```

- ADFS トラフィックポリシーと ADFS サービスを負荷分散仮想サーバーにバインドします。コマンドプロンプトで入力します:

  “

  bind lb vserver lb_adfs_proxy adfs_service

  bind lb vserver lb_adfs_proxy -policyName adfs_traffic_pol -priority 1 -type REQUEST

  “‘

- SNIP アドレスを設定します:


```
add ns ip p.q.r.s 255.255.255.0 -type SNIP
```

- トラフィックアクションとポリシーを設定します:

```
add tm trafficAction adfs_traffic_action -SSO ON -formSSOAction
    adfs_form_action

add tm formSSOAction adfs_form_action -actionURL "/adfs/ls" -
    userField UserName -passwdField Password -ssoSuccessRule "HTTP
    .RES.SET_COOKIE.EQ("MSISAuth")" -nameValuePair AuthMethod=
    FormsAuthentication -submitMethod POST

add tm trafficAction adfs_traffic_action -SSO ON -formSSOAction
    adfs_form_action

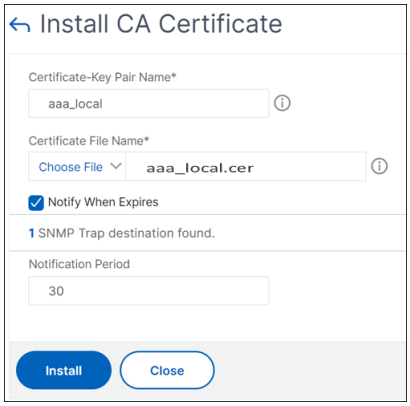
add tm trafficPolicy adfs_traffic_pol "HTTP.REQ.URL.EQ("/adfs/ls/
    ")" adfs_traffic_action
```

NetScaler が **ADFS** サーバーと連携するように構成するには、以下を実行する必要があります。

1. ADFS プロキシプロファイルで使用する SSL CertKey プロファイルキーを作成する
2. ADFS プロキシプロファイルを作成する
3. ADFS プロキシプロファイルを負荷分散仮想サーバーに関連付けます

GUI を使用して **ADFS** プロキシプロファイルで使用する秘密鍵を含む **SSL** 証明書を作成する

1. [トラフィック管理] **> [SSL] > [**証明書] **> [CA** 証明書] に移動します。
2. 詳細ペインで、[インストール] をクリックします。
   [証明書のインストール] ダイアログボックスで、証明書やキーファイル名などの詳細を入力し、[証明書バンドル] を選択します。
3. [インストール] をクリックし、[閉じる] をクリックします。



CLI を使用して **ADFS** プロキシプロファイルで使用するプライベートキーを含む **SSL** 証明書を作成する

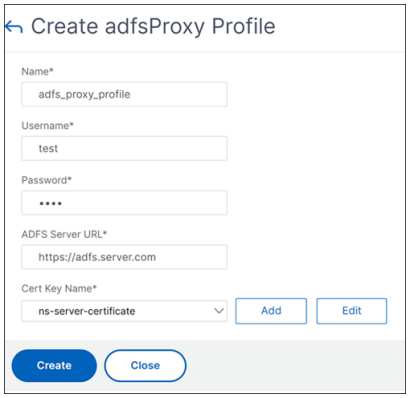
コマンドプロンプトで入力します:

```
add ssl certKey aaa_local -cert aaa_local.cer -key aaa_local.key
```

注: 証明書ファイルとキーファイルは NetScaler アプライアンスにある必要があります。

GUI を使用して **ADFS** プロキシプロファイルを作成する

1. [設定] **>** [セキュリティ] **>** [**AAA** アプリケーショントラフィック] **>** [**AdfsProxy** プロファイル] に移動します。
2. **AdfsProxy** プロファイルページで「追加」をクリックします。
3. 「**ADFSProxy** プロファイルの作成」ページで、次のパラメータを設定します:
   - 名前:ADFS プロキシプロファイルに名前を割り当てます。
   - ユーザー名: プロキシとして機能する NetScaler からの信頼要求の認証に使用されるディレクトリ内のアカウントの名前です。
   - パスワード: これはアカウントのパスワードです。
   - サーバー URL: ADFS サーバーの FQDN。
   - 証明書キー名: ADFS サーバーに登録されているプロキシの SSL 証明書。
4. [作成] をクリックします。



CLI を使用して **ADFS** プロキシプロファイルを作成する

コマンドプロンプトで入力します:

```
add authentication adfsProxyProfile <profile name> -serverUrl <https://
    <server FQDN or IP address>/> -username <adfs admin user name> -
    password <password for admin user> -certKeyName <name of the CertKey
     profile created above>
```



```
2
3 add dns addRec adfs.server.com 10.106.30.151
```

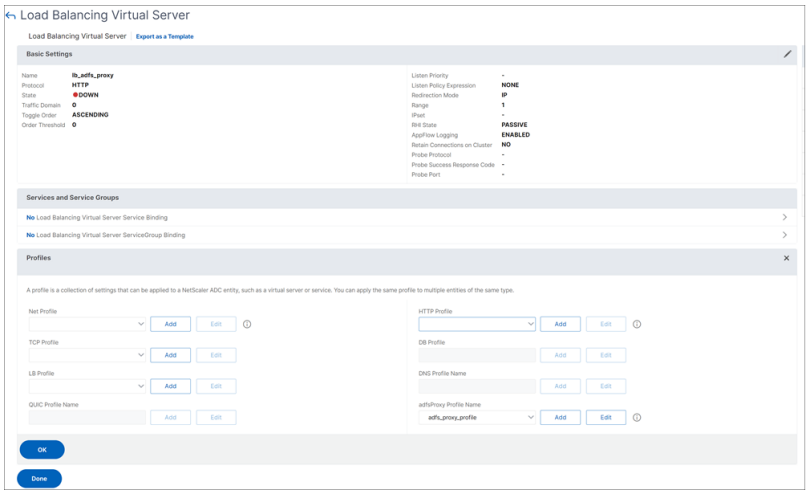
例:

```
1 add authentication adfsProxyProfile adfs_proxy_profile -userName test -
    password test -serverURL "https://adfs.server.com" -CertKeyName
    adfs_trust_cert
```

> 注:
>
> ADFS プロキシと ADFS サーバーは、証明書が ADFS サーバーの認証局によって署名されている場合にのみ信頼を確立できます。

GUI を使用して **ADFS** プロキシプロファイルを負荷分散仮想サーバーに関連付けます

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、SSL 仮想サーバーを選択します。
2. [編集] をクリックします。
3. 「詳細設定」で、「プロフィール」をクリックします。
4. 前に作成した ADFS プロファイルを選択します。
5. 「**OK**」をクリックし、「完了」をクリックします。



CLI を使用して **ADFS** プロキシプロファイルを負荷分散仮想サーバーに関連付けます

ADFS デプロイメントでは、2 つの仮想サーバーが使用されます。1 つはクライアントトラフィック用、もう 1 つはメタデータ交換用です。ADFS プロキシプロファイルは、ADFS サーバーのフロントエンドである負荷分散仮想サーバーに関連付ける必要があります。

コマンドプロンプトで入力します:


```
set ssl vserver lb_adfs_proxy -sslProfile
    ns_default_ssl_profile_frontend
```

ADFSPIP の信頼更新サポート

有効期限が近づいているか、既存の証明書が有効でない場合は、既存の証明書の信頼を更新できます。証明書の信頼更新は、NetScaler アプライアンスと ADFS サーバー間で信頼が確立された場合にのみ行われます。証明書の信頼性を更新するには、新しい証明書を提供する必要があります。

> 重要:
>
> 新しい証明書の信頼更新には、手動による介入が必要です。

次の例は、証明書信頼の更新に関係する手順を示しています。

1. NetScaler アプライアンスは、古い証明書（シリアル化された信頼証明書）と新しい証明書（シリアル化された代替証明書）の両方を POST リクエストで ADFS サーバーに送信して信頼を更新します。
2. 信頼が正常に更新されると、ADFS サーバーは 200 OK 成功を返して応答します。
3. 信頼の更新が成功すると、NetScaler アプライアンスは状態を「ESTABLISHED_RENEW_SUCCESS」として更新します。トラストの更新に失敗すると、状態は「ESTABLISHED_RENEW_FAILED」に更新され、NetScaler アプライアンスは古い証明書を使い続けます。

> 注:
>
> 証明書キーがすでに ADFS プロキシプロファイルにバインドされている場合は、証明書キーを更新できません。

CLI を使用して証明書の信頼更新を設定するには

コマンドプロンプトで入力します:

```
add ssl certKey <name> -cert <certificate file> -key <key file>
```

```
add authentication vserver <name> SSL <ipaddress> <port>
```

```
bind ssl vserver <virtual server name> -certkeyName <string>
```

例:

```
add ssl certKey adfs_trust_cert -cert "client/client_rsa_2048.pem" -key
    "client/client_rsa_2048.ky"
```

> 注:
>
> 証明書は ADFS サーバーの認証局が発行する必要があります。


```
add authentication vserver avs_adfs_proxy SSL 0.0.0.0
```

```
bind ssl vserver avs_adfs_proxy -certkeyName aaa_local
```

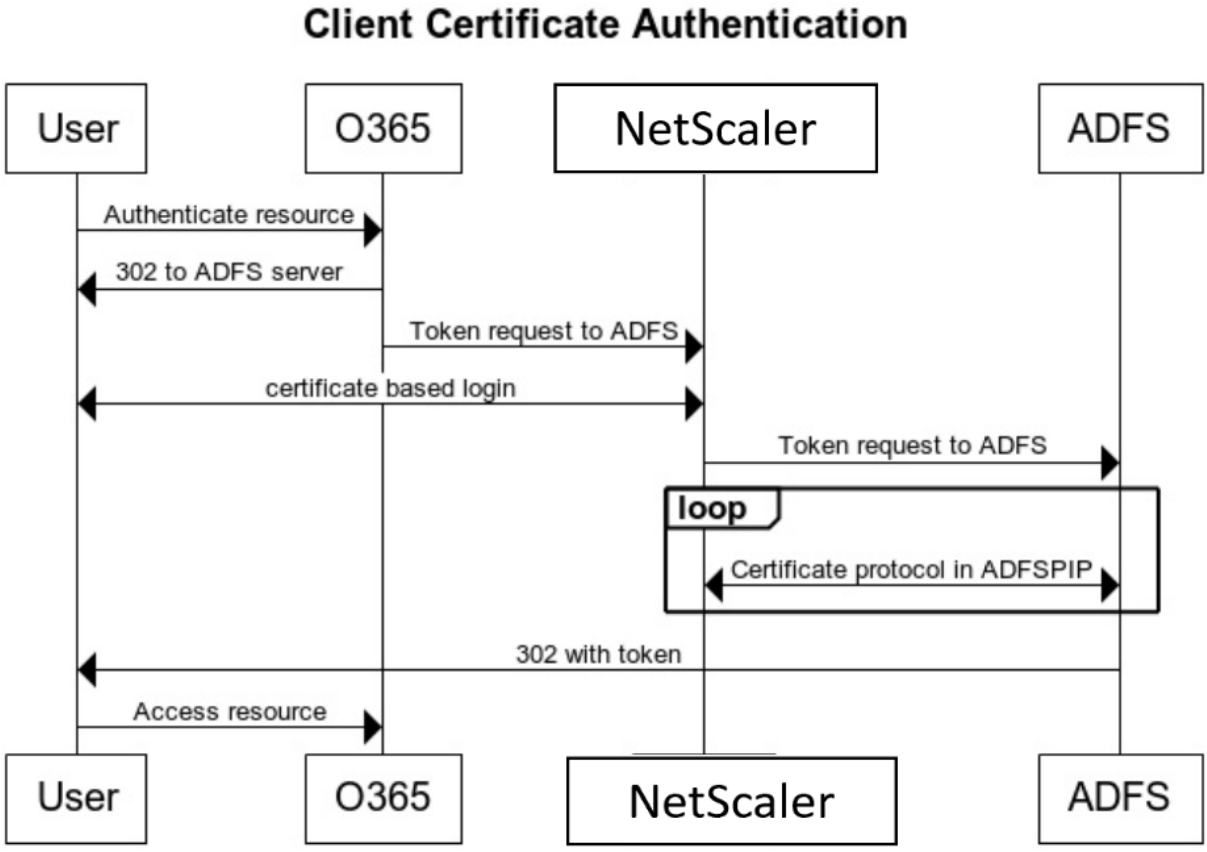
ADFS サーバーでのクライアント証明書ベースの認証

Microsoft は、Windows Server 2016 以降、プロキシサーバーを介して ADFS にアクセスするときにユーザーを認証する新しい方法を導入しました。これで、エンドユーザーは証明書を使用してログインできるようになり、パスワードの使用を回避できます。

エンドユーザーは、特に社内にいない場合に、プロキシ経由で ADFS にアクセスすることがよくあります。したがって、ADFS プロキシサーバーは ADFSPIP プロトコルによるクライアント証明書認証をサポートする必要があります。

NetScaler アプライアンスを使用して ADFS の負荷分散を行う場合、ADFS サーバーで証明書ベースの認証をサポートするには、ユーザーは証明書を使用して NetScaler アプライアンスにログインする必要もあります。これにより、NetScaler はユーザー証明書を ADFS に渡して、ADFS サーバーに SSO を提供できます。

次の図は、クライアント証明書認証フローを示しています。



クライアント証明書を使用して **ADFS** サーバの **SSO** を設定する

クライアント証明書を使用して ADFS サーバーの SSO を構成するには、まず NetScaler アプライアンスでクライアント証明書認証を構成する必要があります。次に、証明書認証ポリシーを認証、承認、および監査仮想サーバーにバインドする必要があります。

また、次の手順を実行する必要があります。

- ポート 49443 を持つ追加のコンテキストスイッチング仮想サーバを設定し、このコンテキストスイッチング仮想サーバが、前に作成したすべてのポートに対して開いている同じ負荷分散仮想サーバをポイントする必要があります。
- 認証するには、NetScaler アプライアンスでポート 49443 を開く必要があります。
- コンテキストスイッチングポリシーは、先に作成したポート 443 が開いているのと同じ負荷分散仮想サーバにバインドする必要があります。
- 前に作成したものと同じ SSL サービスを、負荷分散仮想サーバーにバインドする必要があります。
- バックエンドの SSL プロファイルを既に作成している場合は、そのプロファイルを使用する必要があります。

コマンドプロンプトで次を入力します:

```
bind ssl vserver <name> -certkeyName <string>

bind ssl vserver <vServerName> -certkeyName <string>

add authentication Policy <policy name> -rule <expression> -action <
    action name>

bind authentication vserver <name> -policy <name of the policy> -
    priority <integer>
```

例:

```
bind ssl vserver lb_adfs_proxy -certkeyName aaa_local

set ssl profile ns_default_ssl_profile_frontend -eRSA ENABLED -
    sessReuse ENABLED -denySSLReneg NONSECURE

bind ssl vserver cs_adfs_proxy -certkeyName aaa_local

add authentication Policy local_pol -rule true -action LOCAL

bind authentication vserver avs_adfs_proxy -policy local_pol -priority
    1
```

NetScaler アプライアンスでのクライアント証明書の構成については、「高度なポリシーを使用したクライアント証明書認証の構成」を参照してください。

‘‘‘

Citrix Cloud の ID プロバイダーとしてオンプレミスの NetScaler Gateway を使用する

December 8, 2023

Citrix Cloud では、ワークスペースにサインインする利用者を認証するための ID プロバイダーとしてオンプレミスの NetScaler Gateway を使用することをサポートしています。

NetScaler Gateway 認証を使用すると、次のことが可能になります。

- 引き続き、既存の NetScaler Gateway でユーザーを認証するため、Citrix Workspace 経由でオンプレミスの Virtual Apps and Desktops のリソースにアクセスできます。
- NetScaler Gateway の認証、承認、および監査機能を Citrix Workspace で使用してください。
- パススルー認証、スマートカード、セキュアトークン、条件付きアクセスポリシー、フェデレーションなどの機能を使用して、ユーザーが Citrix Workspace を通じて必要なリソースにアクセスできるようにします。

NetScaler Gateway 認証は、次の製品バージョンでの使用がサポートされています。

- NetScaler Gateway 13.0 41.20 アドバンスエディションまたはそれ以降
- NetScaler Gateway 12.1 54.13 アドバンスエディションまたはそれ以降

前提条件

- クラウドコネクタ-Citrix Cloud Connector ソフトウェアをインストールするには、少なくとも 2 台のサーバーが必要です。
- Active Directory -必要なチェックを実行します。
- NetScaler Gateway の要件
  - クラシックポリシーの非推奨のため、オンプレミスゲートウェイで高度なポリシーを使用します。
  - Citrix Workspace へのサブスクライバを認証するためにゲートウェイを構成する場合、ゲートウェイは OpenID Connect プロバイダーとして機能します。Citrix Cloud と Gateway 間のメッセージは OIDC プロトコルに準拠し、デジタル署名トークンが含まれます。したがって、これらのトークンに署名するための証明書を構成する必要があります。
  - クロック同期-ゲートウェイは NTP 時刻に同期する必要があります。

詳しくは、「前提条件」を参照してください。

オンプレミスの NetScaler Gateway で OAuth IdP ポリシーを作成する

> 重要:
>
> **Citrix Cloud >** ［アイデンティティおよびアクセス管理］**>** ［認証］タブで、クライアント **ID**、シークレット、およびリダイレクト **URL** を生成しておく必要があります。詳しくは、「オンプレミスの NetScaler Gateway を Citrix Cloud に接続する」を参照してください。

OAuth IdP 認証ポリシーの作成には、次のタスクが含まれます。

1. OAuth ID プロバイダープロファイルを作成する。
2. OAuth IdP ポリシーを追加します。
3. OAuth IdP ポリシーを認証仮想サーバーにバインドします。
4. 証明書をグローバルにバインドします。

CLI を使用した **OAuth IdP** プロファイルの作成

コマンドプロンプトで次を入力します:

```
add authentication OAuthIDPProfile <name> [-clientID <string>][-
    clientSecret ][-redirectURL <URL>][-issuer <string>][-audience <
    string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]

add authentication OAuthIdPPolicy <name> -rule <expression> [-action <
    string>] [-undefAction <string>] [-comment <string>][-logAction <
    string>]

add authentication ldapAction <name> -serverIP <IP> -ldapBase "dc=aaa,
    dc=local"

ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -
    ldapLoginName sAMAccountName

add authentication policy <name> -rule <expression> -action <string>

bind authentication vserver auth_vs -policy <ldap_policy_name> -
    priority <integer> -gotoPriorityExpression NEXT

bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -
    priority <integer> -gotoPriorityExpression END

bind vpn global -certkeyName <>
<!--NeedCopy-->
```

GUI を使用した **OAuth IdP** プロファイルの作成

1. [セキュリティ] **>** **[AAA-**アプリケーショントラフィック] **>** [ポリシー] **>** [認証] **>** [高度なポリシー] **>** **[OAuth ID** プロバイダー] に移動します。

2. **OAuth IDP** ページで、[プロファイル] タブを選択し、[追加] をクリックします。
3. OAuth IdP プロファイルを設定します。

   > 注:
   >
   > - **Citrix Cloud >** ［アイデンティティとアクセス管理］ **>** ［認証］ タブからクライアント **ID**、シークレット、およびリダイレクト **URL** の値をコピーして貼り付けて、Citrix Cloud への接続を確立します。
   > - 発行者名の例にゲートウェイ URL を正しく入力します。https://GatewayFQDN.com
   > - また、クライアント ID を [オーディエンス] フィールドにもコピーして貼り付けます。
   > - パスワードを送信: シングルサインオンサポートの場合、このオプションを有効にします。デフォルトでは、このオプションは無効になっています。

4. [認証 **OAuth IDP** プロファイルの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。

   - **Name**： 認証プロファイルの名前。英字、数字、またはアンダースコア (_) で始まる必要があります。名前には、英字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコアのみを含める必要があります。プロファイルの作成後は変更できません。
   - クライアント **ID**： SP を識別する一意の文字列。承認サーバーは、この ID を使用してクライアントの構成を推測します。最大文字数：127。
   - **Client Secret** - ユーザーと承認サーバーによって確立されたシークレット文字列。最大長:239
   - リダイレクト **URL** —コード/トークンのポスト先となる SP 上のエンドポイント。
   - [**Issuer Name**]：トークンを受け付けるサーバーの ID。最大文字数：127。例：https://GatewayFQDN.com
   - **Audience** —IdP によって送信されたトークンのターゲット受信者。このトークンは受信者によって確認されます。
   - スキュー時間—このオプションは、NetScaler が受信トークンで許容するクロックスキュー（分単位）を指定します。たとえば、SkewTime が 10 の場合、トークンは (現在の時刻-10) 分から (現在時刻 + 10) 分、つまり 20 分まで有効です。デフォルト値:5。
   - デフォルト認証グループ：nFactor フローで使用できる IdP によってこのプロファイルが選択されたときに、セッション内部グループリストに追加されるグループ。証明書パーティ関連の nFactor フローを識別するための認証ポリシーの式（AAA.USER.IS_MEMBER_OF (「xxx」)）で使用できます。最大文字数：63。

   このプロファイルのセッションにグループが追加され、ポリシーの評価が簡素化され、ポリシーのカスタマイズに役立ちます。このグループは、抽出されたグループに加えて認証が成功した場合に選択されるデフォルトのグループです。

- 依存パーティメタデータ **URL**：NetScaler IdP が構成中の依存パーティに関する詳細を取得できるエンドポイント。メタデータ応答には、RP 公開鍵の jwks_uri のエンドポイントが含まれている必要があります。最大長は 255 です。
- 更新間隔: 依存パーティメタデータが更新される間隔。デフォルトの間隔は 50 です。
- トークンの暗号化：このオプションを選択すると、NetScaler から送信されるトークンは暗号化されます。
- 署名サービス: データの署名に使用されるクラウドサービスの名前。これは、署名がクラウドにオフロードされた場合にのみ適用されます。
- 属性:ID トークンに挿入される属性の名前と値のペア。最大長は 1047 文字です。
- パスワードを送信:ID トークンで暗号化されたパスワードを送信するには、このオプションを選択します。

5. **[**ポリシー**]** をクリックし、**[**追加**]** をクリックします
6. [認証 **OAuth IDP** ポリシーの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。

   - **Name** - 認証ポリシーの名前。
   - **Action** - 以前に作成されたプロファイルの名前。
   - **Log Action** - 要求がこのポリシーに一致する場合に使用するメッセージログアクションの名前。必須の提出ではありません。
   - **Undefined-Result** アクション：ポリシー評価の結果が未定義（UNDEF）である場合に実行するアクション。必須フィールドではありません。
   - **Expression** - ポリシーが特定の要求に応答するために使用するデフォルトの構文式。例：true。
   - **Comments** - ポリシーに関するコメント。

> 注:
>
> **SendPassword** がオン（デフォルトではオフ）に設定されている場合、ユーザー資格情報は暗号化され、安全なチャネルを介して Citrix Cloud に渡されます。セキュリティで保護されたチャネルを介してユーザー資格情報を渡すと、起動時に Citrix Virtual Apps and Desktops への SSO を有効にできます。

OAuthIdP ポリシーと **LDAP** ポリシーを認証仮想サーバーにバインドする

1. 設定 **>** セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** アクション **> LDAP** に移動します。
2. [**LDAP** アクション] 画面で、[追加] をクリックします。
3. [認証 **LDAP** サーバーの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。

   - **Name** — LDAP アクションの名前
   - サーバ名**/**サーバ **IP** — LDAP サーバの FQDN または IP を指定します。
   - [セキュリティタイプ**]**、**[**ポート**]**、**[**サーバータイプ**]**、**[**タイムアウト] の適切な値を選択する
   - [認証**]** がオンになっていることを確認します

- ベース **DN**： LDAP 検索を開始する基本。例：`dc=aaa,dc=local`。
- 管理者バインド **DN:** LDAP サーバーへのバインドのユーザー名。例：`admin@aaa.local`。
- 管理者パスワード**/**パスワードの確認**:LDAP** をバインドするためのパスワード
- [接続のテスト] をクリックして、設定をテストします。
- サーバーログオン名属性**:**「**samAccountName**」を選択します。
- その他のフィールドは必須ではないため、必要に応じて設定できます。

4. 設定 **>** セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** ポリシーにナビゲートして下さい。
5. [認証ポリシー] 画面で、[追加] をクリックします。
6. [認証ポリシーの作成] ページで、次のパラメータの値を設定し、[作成**]** をクリックします。

   - **Name** — LDAP 認証ポリシーの名前。
   - [アクションタイプ] —[**LDAP**] を選択します。
   - [アクション**]** — LDAP アクションを選択します。
   - **Expression** —ポリシーが特定のリクエストに応答するために使用するデフォルトの構文式。例: true**。

NetScaler Gateway でのアクティブ**-**アクティブ **GSLB** 展開のサポート

August 15, 2023

OIDC プロトコルを使用して ID プロバイダー（IdP）として構成された NetScaler Gateway は、アクティブ-アクティブ GSLB 展開をサポートできます。

GSLB セットアップの設定について詳しくは、「GSLB のセットアップと設定の例」を参照してください。

> 重要：
>
> Citrix Cloud では、NetScaler Gateway を OAuth IdP として使用するアクティブ-アクティブ GSLB はサポートされていません。

接続プロキシを使用した多要素認証に対する **GSLB** アクティブ**/**アクティブのサポート

NetScaler リリース 13.1 ビルド 12.x 以降、接続プロキシを使用した多要素認証の GSLB アクティブ展開のサポートが追加されました。このサポートは、NetScaler Gateway と NetScaler の認証、承認、および監査シナリオに適用されます。接続プロキシは、認証が成功すると、要求を正しい GSLB サイトにルーティングするために使用されます。接続プロキシの永続性の詳細については、「接続プロキシ」を参照してください。

機能

GSLB サイト永続性 Cookie が認証レスポンスに挿入されます。この Cookie を使用して、NetScaler または NetScaler Gateway アプライアンスは、リクエストがローカルサイト向けかリモートサイト向けかを識別します。その後、要求はそれに応じてルーティングされます。

> 重要:
>
> - GSLB アクティブ/アクティブタイプのデプロイメントのみがサポートされます。
> - 親子トポロジはサポートされていません。
> - GSLB デプロイメントのパーシステンスタイプは「ConnectionProxy」として構成する必要があります。

SameSite Cookie 属性の構成サポート

December 8, 2023

SameSite 属性は、Cookie をクロスサイトコンテキストに使用できるか、同一サイトコンテキストにのみ使用できるかをブラウザに示します。また、アプリケーションがクロスサイトコンテキストでアクセスされることを意図している場合は、HTTPS 接続を介してのみアクセスできます。詳細については、RFC6265 を参照してください。

2020 年 2 月まで、SameSite 属性は NetScaler で明示的に設定されていませんでした。ブラウザはデフォルト値 (None) を使用しました。SameSite 属性を設定しなくても、NetScaler Gateway、認証、承認、監査の展開には影響しませんでした。

Google Chrome 80 などの特定のブラウザのアップグレードでは、Cookie のデフォルトのクロスドメイン動作に変更があります。SameSite 属性は、次のいずれかの値に設定できます。Google Chrome のデフォルト値は Lax に設定されています。他のブラウザの特定のバージョンでは、SameSite 属性のデフォルト値が None に設定されている場合があります。

- なし: 安全な接続でのみクロスサイトコンテキストで Cookie を使用するようブラウザに指示します。
- **Lax**: ブラウザが同じドメイン上のリクエストやクロスサイトのリクエストに Cookie を使用することを示します。クロスサイトでは、GET リクエストなどの安全な HTTP メソッドのみが Cookie を使用できます。たとえば、あるサブドメイン abc.example.com の GET リクエストは、GET を使用して別のサブドメイン xyz.example.com の Cookie を読み取ることができます。
  クロスサイトの場合、安全な HTTP メソッドはサーバーの状態を変更しないため、安全な HTTP メソッドのみが使用されます。詳しくは、「https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite#lax」を参照してください
- 厳格:Cookie は同じサイトコンテキストでのみ使用します。

Cookie に SameSite 属性がない場合、Google Chrome は SameSite = Lax の機能を想定しています。
その結果、ブラウザによる Cookie の挿入を必要とするクロスサイトコンテキストを持つ iframe 内のデプロイでは、

Google Chrome はクロスサイト Cookie を共有しません。その結果、Web サイト内の iframe が読み込まれないことがあります。

SameSite Cookie 属性を設定する

SameSite という名前の新しい Cookie 属性が VPN および認証、承認、および監査仮想サーバーに追加されます。この属性は、グローバルレベルおよび仮想サーバレベルで設定できます。

SameSite 属性を設定するには、次の操作を行う必要があります。

1. 仮想サーバーの SameSite 属性を設定する
2. クッキーをパッチセットにバインドする (ブラウザがクロスサイトクッキーをドロップした場合)

CLI を使用して SameSite 属性を設定する

仮想サーバーレベルで SameSite 属性を設定するには、次のコマンドを使用します。

```
set vpn vserver VP1 -SameSite [STRICT | LAX | None]
set authentication vserver AV1 -SameSite [STRICT | LAX | None]
<!--NeedCopy-->
```

グローバルレベルで SameSite 属性を設定するには、次のコマンドを使用します。

```
set aaa parameter -SameSite [STRICT | LAX | None]
set vpn parameter -SameSite [STRICT | LAX | None]
<!--NeedCopy-->
```

注：仮想サーバーレベルの設定は、グローバルレベル設定よりも優先されます。Citrix では、仮想サーバーレベルで SameSiteCookie 属性を設定することをお勧めします。

CLI を使用してクッキーをパッチセットにバインドする

ブラウザがクロスサイトクッキーをドロップした場合、その Cookie 文字列を既存の NS_cookies_SameSite パッチセットにバインドして、SameSite 属性が Cookie に追加されるようにすることができます。

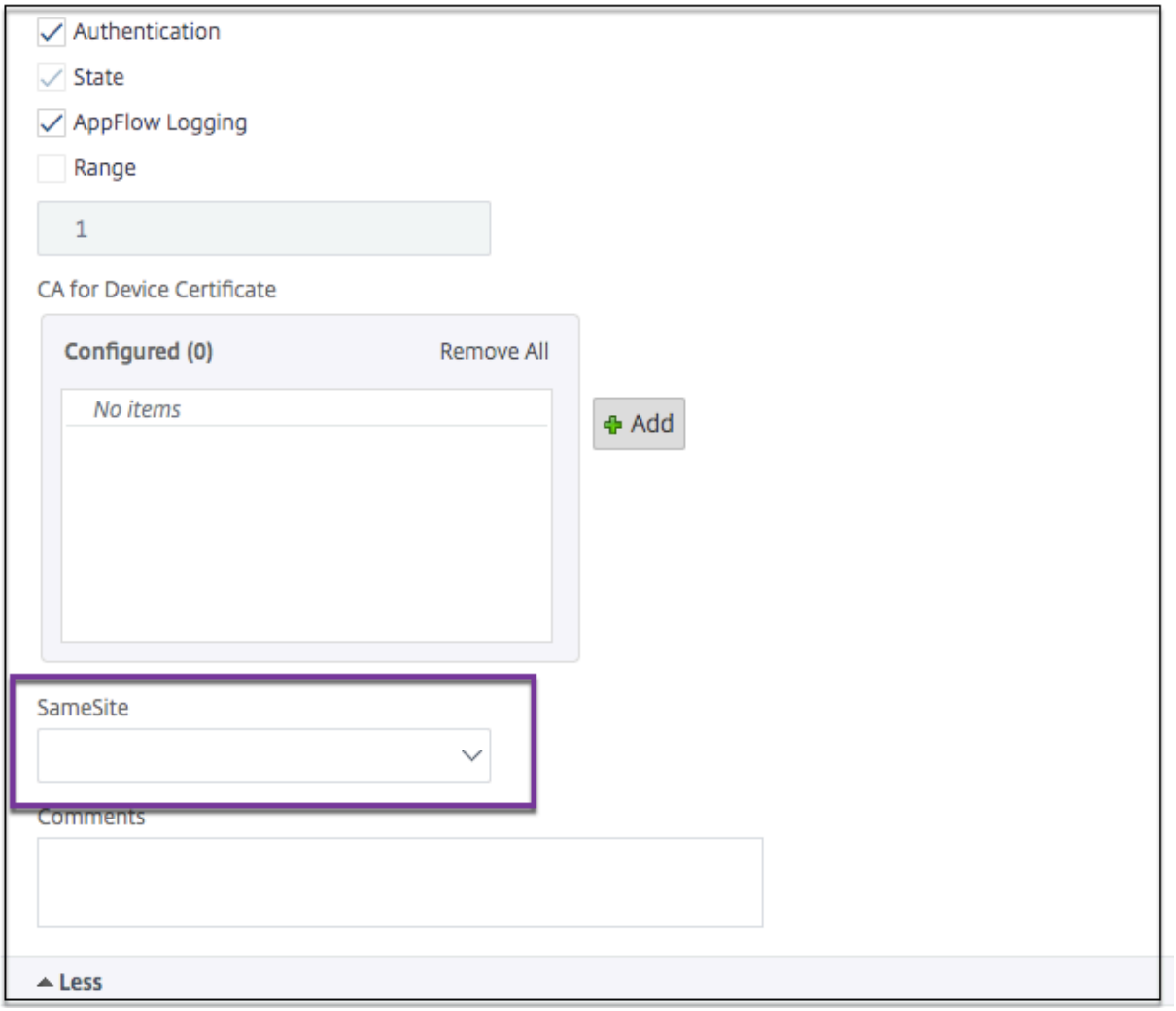
例:

```
bind patset ns_cookies_SameSite "NSC_TASS"
bind patset ns_cookies_SameSite "NSC_TMAS"
<!--NeedCopy-->
```

GUI を使用して sameSite 属性を設定する

仮想サーバーレベルで **SameSite** 属性を設定するには、次の手順を実行します。

1. [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] に移動します
2. 仮想サーバを選択し、[編集（Edit）] をクリックします。
3. [基本設定] セクションの [編集] アイコンをクリックし、[詳細] をクリックします。
4. **SameSite** で、必要に応じてオプションを選択します。



グローバルレベルで **SameSite** 属性を設定するには**:**

1. セキュリティ **> AAA** ―アプリケーショントラフィック **>** 認証設定の変更に移動します。

2. [**AAA** パラメータの設定] ページで、[**SameSite**] リストをクリックし、必要に応じてオプションを選択しま

す。

一般的に使用されるプロトコルの認証、承認、および監査の構成

August 15, 2023

NetScaler アプライアンスを認証、承認、および監査用に構成するには、NetScaler アプライアンスとクライアントのブラウザーで特定の設定を行う必要があります。設定は、認証、認可、および監査に使用されるプロトコルによって異なります。

NetScaler アプライアンスを Kerberos 認証用に構成する方法の詳細については、「Kerberos/NTLM による認証、承認、および監査の処理」を参照してください。

Kerberos/NTLM による認証、承認、監査の処理

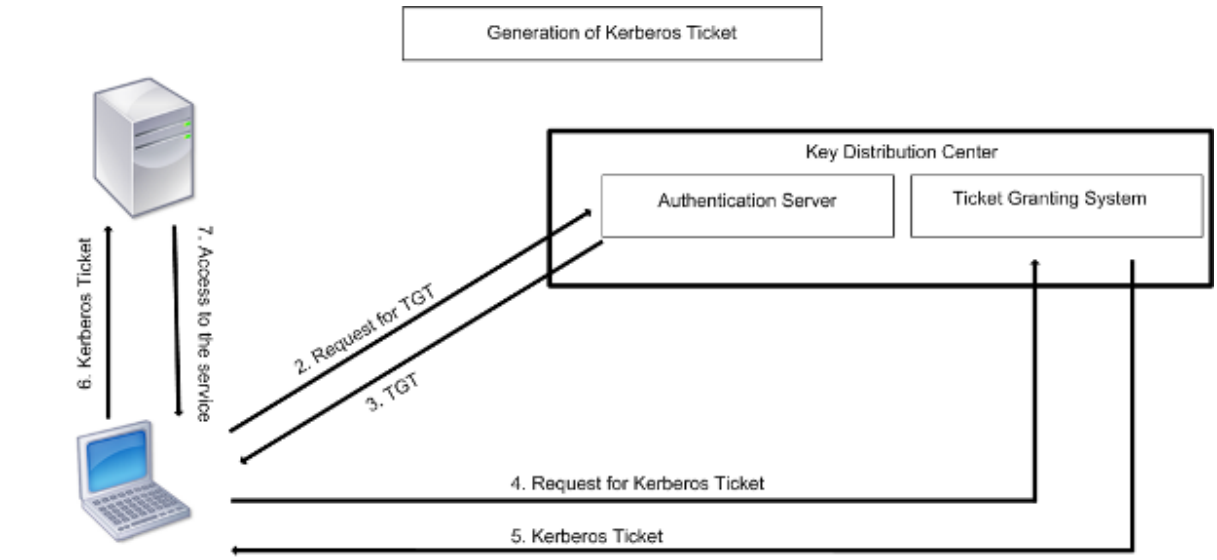
August 15, 2023

コンピュータネットワーク認証プロトコルである Kerberos は、インターネットを介した安全な通信を提供します。主にクライアント/サーバーアプリケーション向けに設計されており、クライアントとサーバーが相互の信頼性を確保できる相互認証を提供します。Kerberos は、キー配布センター (KDC) と呼ばれる信頼できるサードパーティを使用しています。KDC は、ユーザーを認証する認証サーバー (AS) とチケット交付サーバー (TGS) で構成されています。

ネットワーク上の各エンティティ (クライアントまたはサーバー) には、そのエンティティと KDC だけが知っている秘密鍵があります。このキーを知っているということは、エンティティが本物であることを意味します。ネットワーク上の 2 つのエンティティ間の通信では、KDC は Kerberos チケットまたはサービスチケットと呼ばれるセッションキーを生成します。クライアントは AS に特定のサーバーの認証情報を要求します。その後、クライアントはチケット交付チケット (TGT) と呼ばれるチケットを受け取ります。次に、クライアントは AS から受け取った TGT を使用してその身元を証明して TGS に連絡し、サービスを依頼します。クライアントがサービスを受ける資格がある場合、TGS はクライアントに Kerberos チケットを発行します。次に、クライアントは、Kerberos チケットを使用してサービスをホストしているサーバー (サービスサーバーと呼ばれる) に接続し、サービスの受信が許可されていることを証明します。Kerberos チケットの有効期限は設定可能です。クライアントは AS で自身を認証するのは 1 回だけです。物理サーバーに複数回接続する場合、AS チケットを再利用します。

次の図は、Kerberos プロトコルの基本的な機能を示しています。

図 1: ケルベロスの機能



Kerberos 認証には次の利点があります。

- より高速な認証。物理サーバーがクライアントから Kerberos チケットを取得すると、サーバーにはクライアントを直接認証するのに十分な情報があります。クライアント認証のためにドメインコントローラーに連絡する必要がないため、認証プロセスが高速になります。
- 相互認証。KDC がクライアントに Kerberos チケットを発行し、クライアントがそのチケットを使用してサービスにアクセスする場合、認証されたサーバーのみが Kerberos チケットを復号化できます。NetScaler アプライアンス上の仮想サーバーが Kerberos チケットを復号化できる場合は、仮想サーバーとクライアントの両方が認証されていると判断できます。したがって、サーバーの認証はクライアントの認証とともに行われます。
- Windows と Kerberos をサポートする他のオペレーティングシステム間のシングルサインオン。

Kerberos 認証には次のような欠点があります。

- Kerberos には厳しい時間要件があります。認証が失敗しないように、関係するホストの時計を Kerberos サーバーの時計と同期させる必要があります。この欠点は、ネットワークタイムプロトコルデーモンを使用してホストクロックの同期を維持することで軽減できます。Kerberos チケットには利用可能期間があり、設定することができます。
- Kerberos では、中央サーバーが継続的に利用可能である必要があります。Kerberos サーバーがダウンすると、誰もログオンできなくなります。複数の Kerberos サーバーとフォールバック認証メカニズムを使用することで、このリスクを軽減できます。
- 認証はすべて一元化された KDC によって制御されるため、ローカルワークステーションのユーザーのパスワードが盗まれるなど、このインフラストラクチャが侵害されると、攻撃者は任意のユーザーになりすますことができます。信頼できるデスクトップマシンまたはラップトップのみを使用するか、ハードウェアトークンによる事前認証を強制することで、このリスクをある程度軽減できます。

Kerberos 認証を使用するには、NetScaler アプライアンスと各クライアントで認証を構成する必要があります。

認証、承認、監査における **Kerberos** 認証の最適化

NetScaler アプライアンスは、Kerberos 認証中にシステムパフォーマンスを最適化および改善するようになりました。認証、承認、および監査デーモンは、同じユーザーに対する未処理の Kerberos 要求を記憶し、キー配布センター (KDC) への負荷を回避します。これにより、要求の重複を回避できます。

NetScaler がクライアント認証用に **Kerberos** を実装する方法

December 8, 2023

> 重要
>
> Kerberos/NTLM 認証は、NetScaler 9.3 nCore リリース以降でのみサポートされており、トラフィック管理仮想サーバーの認証、承認、監査にのみ使用できます。

NetScaler は、Kerberos 認証に関連するコンポーネントを以下の方法で処理します。

キー・ディストリビューション・センター **(KDC)**

Windows 2000 サーバ以降のバージョンでは、ドメインコントローラと KDC は Windows サーバの一部です。Windows Server が稼働している場合は、ドメインコントローラと KDC が設定されていることを示します。KDC は Active Directory サーバでもあります。

> 注
>
> すべての Kerberos インタラクションは Windows Kerberos ドメインコントローラーで検証されます。

認証サービスとプロトコルネゴシエーション

NetScaler アプライアンスは、認証、承認、および監査トラフィック管理認証仮想サーバーでの Kerberos 認証をサポートします。Kerberos 認証が失敗した場合、NetScaler は NTLM 認証を使用します。

デフォルトでは、Windows 2000 Server 以降の Windows Server バージョンでは、認証、承認、および監査に Kerberos を使用します。認証タイプとして NEGITEATE を使用して認証ポリシーを作成すると、NetScaler は認証、承認、および監査に Kerberos プロトコルを使用しようとします。クライアントのブラウザが Kerberos チケットを受信できない場合、NetScaler は NTLM 認証を使用します。このプロセスはネゴシエーションと呼ばれます。

次のいずれかの場合、クライアントは Kerberos チケットを受信できない可能性があります。

- Kerberos はクライアントではサポートされていません。
- Kerberos はクライアントで有効になっていません。
- クライアントは KDC のドメインとは別のドメインにあります。

- クライアントは KDC のアクセスディレクトリにアクセスできません。

Kerberos/NTLM 認証の場合、NetScaler は NetScaler アプライアンス上にローカルに存在するデータを使用しません。

承認

トラフィック管理仮想サーバーは、負荷分散仮想サーバーでもコンテンツスイッチング仮想サーバーでもかまいません。

監査

NetScaler アプライアンスは、以下の監査ログによる Kerberos 認証の監査をサポートしています。

- トラフィック管理のエンドユーザーアクティビティの完全な監査証跡
- SYSLOG およびハイパフォーマンス TCP ロギング
- システム管理者の完全な監査証跡
- すべてのシステムイベント
- スクリプト可能なログ形式

サポート環境

Kerberos 認証には、NetScaler 上の特定の環境は必要ありません。クライアント (ブラウザ) は Kerberos 認証をサポートする必要があります。

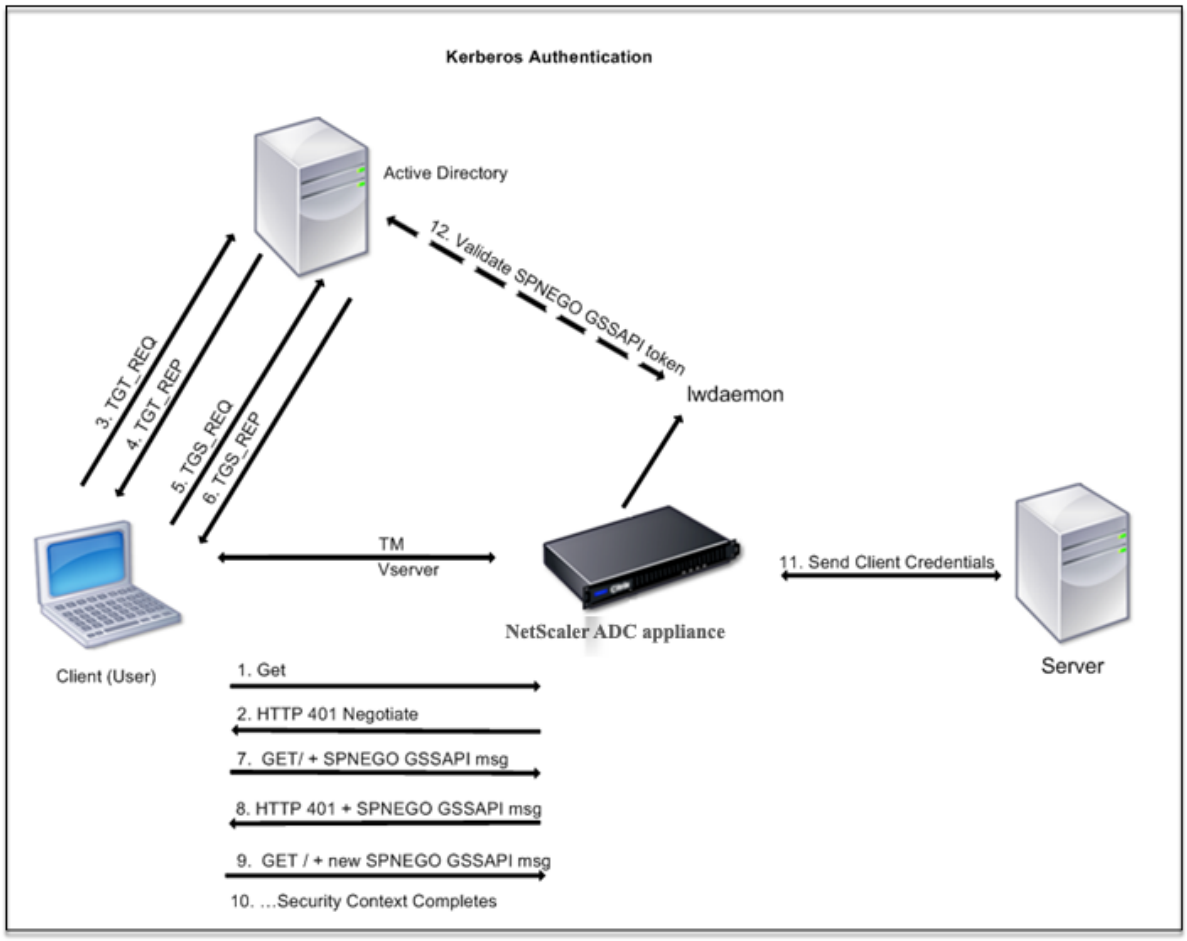
高可用性

高可用性セットアップでは、アクティブな NetScaler のみがドメインに参加します。フェイルオーバーの場合、NetScaler lwagent デーモンはセカンダリ NetScaler アプライアンスをドメインに参加させます。この機能には特別な設定は必要ありません。

Kerberos 認証プロセス

次の図は、NetScaler 環境での Kerberos 認証の一般的なプロセスを示しています。

図 1. NetScaler でのケルベロス認証プロセス



Kerberos 認証は次の段階で行われます。

クライアントは **KDC** に対して自分自身を認証します

1. NetScaler アプライアンスはクライアントから要求を受け取ります。
2. NetScaler アプライアンス上のトラフィック管理（負荷分散またはコンテンツスイッチング）仮想サーバーは、クライアントにチャレンジを送信します。
3. このチャレンジに応えるために、クライアントは Kerberos チケットを受け取ります。
    - クライアントは KDC の認証サーバーにチケット交付チケット (TGT) のリクエストを送信し、TGT を受信します。(図「Kerberos 認証プロセス」の 3、4 を参照してください。)
    - クライアントは TGT を KDC のチケット交付サーバーに送信し、Kerberos チケットを受け取ります。(図「Kerberos 認証プロセス」の 5、6 を参照してください。)

> 注
>
> クライアントに有効期限が切れていない Kerberos チケットが既にある場合は、上記の認証プロセスは必要ありません。さらに、SPNEGO をサポートする Web サービス、.NET、J2EE などのクライアントは、ターゲッ

> トサーバーの Kerberos チケットを取得し、SPNEGO トークンを作成し、HTTP リクエストを送信するときにそのトークンを HTTP ヘッダーに挿入します。クライアント認証プロセスは実行されません。

クライアントはサービスをリクエストします。

1. クライアントは、SPNEGO トークンと HTTP リクエストを含む Kerberos チケットを NetScaler 上のトラフィック管理仮想サーバーに送信します。SPNEGO トークンには必要な GSSAPI データが含まれています。
2. NetScaler アプライアンスは、クライアントと NetScaler 間のセキュリティコンテキストを確立します。NetScaler が Kerberos チケットで提供されたデータを受け入れられない場合、クライアントは別のチケットを取得するよう求められます。このサイクルは、GSSAPI データが許容範囲内になり、セキュリティコンテキストが確立されるまで繰り返されます。NetScaler 上のトラフィック管理仮想サーバーは、クライアントと物理サーバー間の HTTP プロキシとして機能します。

NetScaler アプライアンスが認証を完了します。

1. セキュリティコンテキストが完了すると、トラフィック管理仮想サーバーは SPNEGO トークンを検証します。
2. 有効な SPNEGO トークンから、仮想サーバーはユーザー ID と GSS 認証情報を抽出し、認証デーモンに渡します。
3. 認証が成功すると、Kerberos 認証が完了します。

NetScaler アプライアンスでのケルベロス認証の設定

August 16, 2023

このトピックでは、CLI と GUI を使用して NetScaler アプライアンスで Kerberos 認証を構成する詳細な手順を説明します。

CLI での **Kerberos** 認証の設定

1. 認証、承認、および監査機能を有効にして、アプライアンス上のトラフィックの認証を確実に行います。

   ns-cli-promp **ns** 機能 **AAA** を有効にする

2. キータブファイルを NetScaler アプライアンスに追加します。Kerberos 認証中にクライアントから受け取ったシークレットを復号するには、キータブファイルが必要です。1 つのキータブファイルには、NetScaler アプライアンス上のトラフィック管理仮想サーバーにバインドされているすべてのサービスの認証詳細が含まれています。

   まず、Active Directory サーバーでキータブファイルを生成してから、NetScaler アプライアンスに転送します。

- Active Directory サーバーにログオンし、次のコマンドを使用して Kerberos 認証用のユーザーを追加します。

```
net user <username> <password> /add
```

> 注
>
> [ユーザーのプロパティ] セクションで、[次回のログオン時にパスワードを変更] オプションが選択されておらず、[パスワードが期限切れにならない] オプションが選択されていることを確認します。

- HTTP サービスを上記のユーザーにマップし、keytab ファイルをエクスポートします。たとえば、Active Directory サーバーで次のコマンドを実行します。

```
ktpass /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM
   /pass <user password> /mapuser newacp\\dummy /ptype KRB5\
   _NT\_PRINCIPAL
```

> 注
>
> 複数のサービスに認証が必要な場合は、複数のサービスをマッピングできます。さらに多くのサービスをマップする場合は、サービスごとに上記のコマンドを繰り返します。出力ファイルには、同じ名前でも別の名前でもかまいません。

- unix **ftp** コマンドまたはその他の任意のファイル転送ユーティリティを使用して、キータブファイルを NetScaler アプライアンスに転送します。キータブファイルを NetScaler アプライアンスの/nsconfig/krb/ディレクトリにアップロードします。

3. NetScaler アプライアンスは、完全修飾ドメイン名（FQDN）からドメインコントローラーの IP アドレスを取得する必要があります。したがって、NetScaler ADC を DNS サーバーで構成することをお勧めします。

   ns-cli-promp **DNS** ネームサーバーを追加 <ip-address>

   > 注
   >
   > または、静的ホストエントリを追加するか、他の方法を使用して、NetScaler アプライアンスがドメインコントローラーの FQDN 名を IP アドレスに変換できるようにすることもできます。

4. 認証アクションを設定し、認証ポリシーに関連付けます。

   - ネゴシエートアクションを設定します。この構成により、Kerberos キー配布センター (KDC) として使用される Active Directory サーバーのアクション (プロファイル) が作成されます。プロファイルには、その AD KDC サーバーとの通信に必要なすべての構成データが含まれます。

     ns-cli-prompt> **add authentication negotiateAction** <name> -domain <domain name> -domainUser <domain user name> -domainUserPasswd <domain user password> -defaultAuthenticationGroup <default authentication group> -keytab <string> -NTLMPath <string>

注: ドメインユーザおよびドメイン名の設定については、クライアントに移動し、次の例に示すように klist コマンドを使用します。

クライアント: ユーザー名 @ AAA.LOCAL

サーバー:http/onprem_idp.AAA.local @ AAA.LOCAL

add authentication negotiateAction <name> -domain -domainUser <HTTP/onprem_idp.aaa.local>

- ネゴシエートポリシーを設定し、ネゴシエートアクションをこのポリシーに関連付けます。

  ns-cli-prompt> **add authentication negotiatePolicy** <name> <rule> <reqAction>

5. 認証仮想サーバを作成し、ネゴシエートポリシーを関連付けます。

   - 認証仮想サーバーを作成します。

     ns-cli-prompt> **add authentication vserver** <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>

   - ネゴシエートポリシーを認証仮想サーバーにバインドします。

     ns-cli-prompt> **bind authentication vserver** <name> -policy <negotiatePolicyName>

6. 認証仮想サーバーをトラフィック管理 (負荷分散またはコンテンツスイッチング) 仮想サーバーに関連付けます。

   ns-cli-prompt> **set lb vserver** <name> -authn401 ON -authnVsName <string>

   > 注
   >
   > コンテンツスイッチング仮想サーバーでも同様の構成を行うことができます。

7. 次の操作を行って、設定を確認します。

   - FQDN を使用して、トラフィック管理仮想サーバにアクセスします。たとえば、サンプル
   - CLI でセッションの詳細を表示します。

     ns-cli-promp **AAA** セッションを表示

GUI での Kerberos 認証の設定

1. 認証、認可、および監査機能を有効にします。

   [システム] > [設定] に移動し、[基本機能の構成] をクリックして、認証、承認、および監査機能を有効にします。

2. 前述の CLI 手順のステップ 2 で説明したように、keytab ファイルを追加します。

3. DNS サーバーを追加します。

   [トラフィック管理] **> [DNS] > [**ネームサーバー] に移動し、DNS サーバーの IP アドレスを指定します。

4. ネゴシエートアクションとポリシーを設定します。

   [セキュリティ] **> [AAA-**アプリケーショントラフィック] **> [**ポリシー] **> [**認証] **> [**高度なポリシー] > [ポリシー] に移動し、アクションタイプとして [ネゴシエート [**ADD**] をクリックして新しい認証ネゴシエートサーバーを作成するか、[**Edit**] をクリックして既存の詳細を設定します。

5. ネゴシエートポリシーを認証仮想サーバーにバインドします。

   [セキュリティ] **> [AAA-**アプリケーショントラフィック] **> [**仮想サーバ] に移動し、ネゴシエートポリシーを認証仮想サーバに関連付けます。

6. 認証仮想サーバーをトラフィック管理 (負荷分散またはコンテンツスイッチング) 仮想サーバーに関連付けます。

   [トラフィック管理] **> [**負荷分散] **> [**仮想サーバー] に移動し、関連する認証設定を指定します。

   > 注
   >
   > コンテンツスイッチング仮想サーバーでも同様の構成を行うことができます。

7. 上記の CLI 手順のステップ 7 で詳述した設定を確認します。

クライアントでの **Kerberos** 認証の設定

August 15, 2023

認証に Kerberos を使用するには、ブラウザで Kerberos サポートを設定する必要があります。Kerberos 準拠のブラウザならどれでも使用できます。Internet Explorer と Mozilla Firefox で Kerberos サポートを設定する手順は次のとおりです。他のブラウザについては、ブラウザのマニュアルを参照してください。

Internet Explorer で **Kerberos** 認証を構成するには

1. [ツール] メニューの [インターネットオプション] を選択します。
2. **[**** セキュリティ] タブの [ローカルイントラネット] をクリックし、[サイト] をクリックします。******
3. ローカルイントラネットダイアログボックスで、「イントラネットネットワークを自動的に検出する」オプションが選択されていることを確認し、「詳細設定」をクリックします。
4. ローカルイントラネットダイアログボックスで、NetScaler アプライアンス上のトラフィック管理仮想サーバーのドメインの Web サイトを追加します。指定されたサイトはローカルイントラネットサイトになります。
5. 「閉じる」または「**OK**」をクリックしてダイアログ・ボックスを閉じます。

Mozilla Firefox をケルベロス認証用に設定するには

1. お使いのコンピューターで Kerberos が適切に設定されていることを確認してください。
2. URL バーに about: config と入力します。
3. フィルターテキストボックスに「network.negotiate」と入力します。
4. network.negotiate-auth.delegation-uris を追加したいドメインに変更してください。
5. network.negotiate-auth.trusted-uris を追加したいドメインに変更してください。

> 注**:** Windows を実行している場合は、フィルターテキストボックスに sspi と入力し、network.auth.use-sspi オプションを False に変更する必要もあります。

Kerberos 認証を物理サーバーからオフロードする

August 15, 2023

NetScaler アプライアンスは、認証タスクをサーバーからオフロードできます。物理サーバーがクライアントからの要求を認証する代わりに、NetScaler はすべてのクライアント要求を認証してから、バインドされた物理サーバーに転送します。ユーザー認証は、Active Directory トークンに基づいています。

NetScaler と物理サーバー間の認証は行われず、認証のオフロードはエンドユーザーには意識されません。Windows コンピュータに最初にログオンした後、エンドユーザーはポップアップまたはログオンページに追加の認証情報を入力する必要はありません。

現在の NetScaler アプライアンスリリースでは、Kerberos 認証はトラフィック管理仮想サーバーの認証、承認、監査にのみ使用できます。Kerberos 認証は、NetScaler Gateway アドバンスドエディションアプライアンスの SSL VPN または NetScaler ADC アプライアンス管理ではサポートされていません。

Kerberos 認証には、NetScaler アプライアンスとクライアントブラウザーでの構成が必要です。

NetScaler アプライアンスで **Kerberos** 認証を構成するには

> 注
>
> 次の設定例で使用されるパスワードは単なる例であり、実際の設定パスワードではありません。

1. Active Directory にユーザーアカウントを作成します。ユーザーアカウントを作成するときは、[ユーザープロパティ] セクションで次のオプションを確認します。
   - [次回ログオン時にパスワードを変更する] オプションを選択していないことを確認します。
   - 必ず [パスワードは期限切れではない] オプションを選択してください。
2. AD サーバーの CLI コマンドプロンプトで、次のように入力します。

- ktpass -princ HTTP/kerberos.crete.lab.net@crete.lab.net -ptype KRB5_NT_PRINCIPAL -mapuser kerbuser@crete.lab.net -mapop set -pass Citrix1 -out C:\kerbtabfile.txt

> 注
>
> 上記のコマンドは必ず 1 行で入力してください。上記のコマンドの出力は C:\kerbtabfile.txt ファイルに書き込まれます。

3. セキュアコピー（SCP）クライアントを使用して、kerbtabfile.txt ファイルを NetScaler アプライアンスの/etc ディレクトリにアップロードします。
4. 次のコマンドを実行して、DNS サーバーを NetScaler アプライアンスに追加します。

   - add dns nameserver 1.2.3.4

   NetScaler アプライアンスは、DNS サーバーがないと Kerberos リクエストを処理できません。必ず、Microsoft Windows ドメインで使用されているものと同じ DNS サーバーを使用してください。
5. NetScaler のコマンドラインインターフェイスに切り替えます。
6. 次のコマンドを実行して、Kerberos 認証サーバーを作成します。

   - add authentication negotiateAction KerberosServer -domain “crete.lab.net” -domainUser kerbuser -domainUserPasswd Citrix1 -keytab /var/mykcd.keytab

   > 注
   >
   > : キータブが使用できない場合は、ドメイン、domainUser、および-domainUserPasswd のパラメータを指定できます。

7. 次のコマンドを実行して、ネゴシエーションポリシーを作成します。

   - `add authentication negotiatePolicy Kerberos-Policy "REQ.IP.DESTIP == 192.168.17.200"KerberosServer<!--NeedCopy-->`

8. 次のコマンドを実行して、認証仮想サーバーを作成します。

   - `add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.lab.net<!--NeedCopy-->`

9. 次のコマンドを実行して、Kerberos ポリシーを認証仮想サーバーにバインドします。

   - `bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100<!--NeedCopy-->`

10. 次のコマンドを実行して、SSL 証明書を認証仮想サーバーにバインドします。GUI NetScaler アプライアンスからインストールできるテスト証明書のいずれかを使用できます。次のコマンドを実行して、ServerTestCert サンプル証明書を使用します。

- `bind ssl vserver Kerb-Auth -certkeyName ServerTestCert<!--NeedCopy-->`

11. IP アドレス 192.168.17.200 を使用して HTTP 負荷分散仮想サーバーを作成します。

    NetScaler 9.3 リリースが 9.3.47.8 より古い場合は、コマンドラインインターフェイスから仮想サーバーを作成するようにしてください。

12. 次のコマンドを実行して、認証仮想サーバーを構成します。

    - `set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth<!--NeedCopy-->`

13. Web ブラウザのアドレスバーにホスト名 Example を入力します。

    Kerberos 認証がブラウザで設定されていないため、Web ブラウザに認証ダイアログボックスが表示されます。

    > 注
    >
    > Kerberos 認証には、クライアント上で特定の設定が必要です。クライアントがホスト名を解決できることを確認します。これにより、Web ブラウザが HTTP 仮想サーバーに接続します。

14. クライアントコンピュータの Web ブラウザで Kerberos を構成します。

    - Internet Explorer での構成については、「Kerberos 認証用の Internet Explorer の構成」を参照してください。
    - Mozilla Firefox での設定については、Internet Explorer の Kerberos 認証の設定を参照してください。

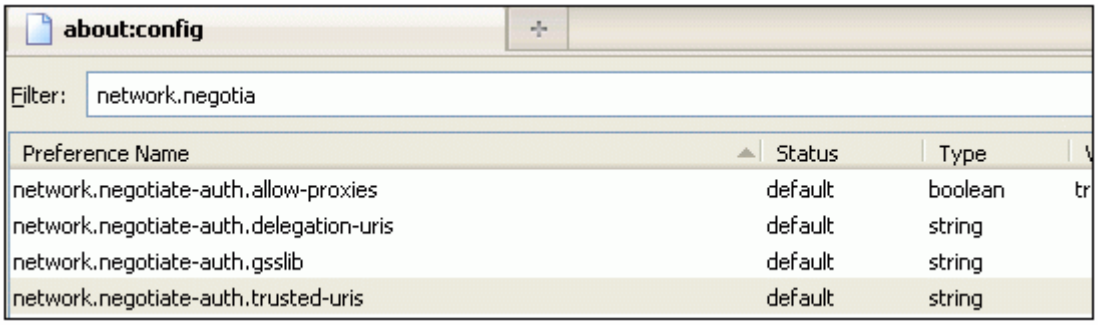
15. 認証なしでバックエンド物理サーバにアクセスできるかどうかを確認します。

Internet Explorer で **Kerberos** 認証を構成するには

1. [ツール] メニューの [インターネットオプション] を選択します。
2. [セキュリティ] タブを有効にします。
3. [セキュリティ設定を変更するゾーンを選択してください] セクションから [ローカルイントラネット] を選択します。
4. [サイト] をクリックします。
5. [詳細設定] をクリックします。
6. URL、例] を指定し、[追加] をクリックします。
7. **Internet Explorer** を再起動します。

Mozilla Firefox をケルベロス認証用に設定するには

1. ブラウザのアドレスバーに about: config と入力します。
2. 警告免責事項をクリックします。
3. [フィルタ] ボックスに「**network.negotiate-auth.trusted-URI** 」と入力します。
4. [**Network.negotiate-auth.trusted-URI]** をダブルクリックします。サンプル画面を以下に示します。



5. [文字列値の入力] ダイアログボックスで、www.crete.lab.net を指定します。
6. Firefox を再起動します。

認証と承認関連の問題のトラブルシューティング

August 15, 2023

エラーメッセージをローカライズする

NetScaler nFactor システムによって生成されたエラーメッセージをローカライズします

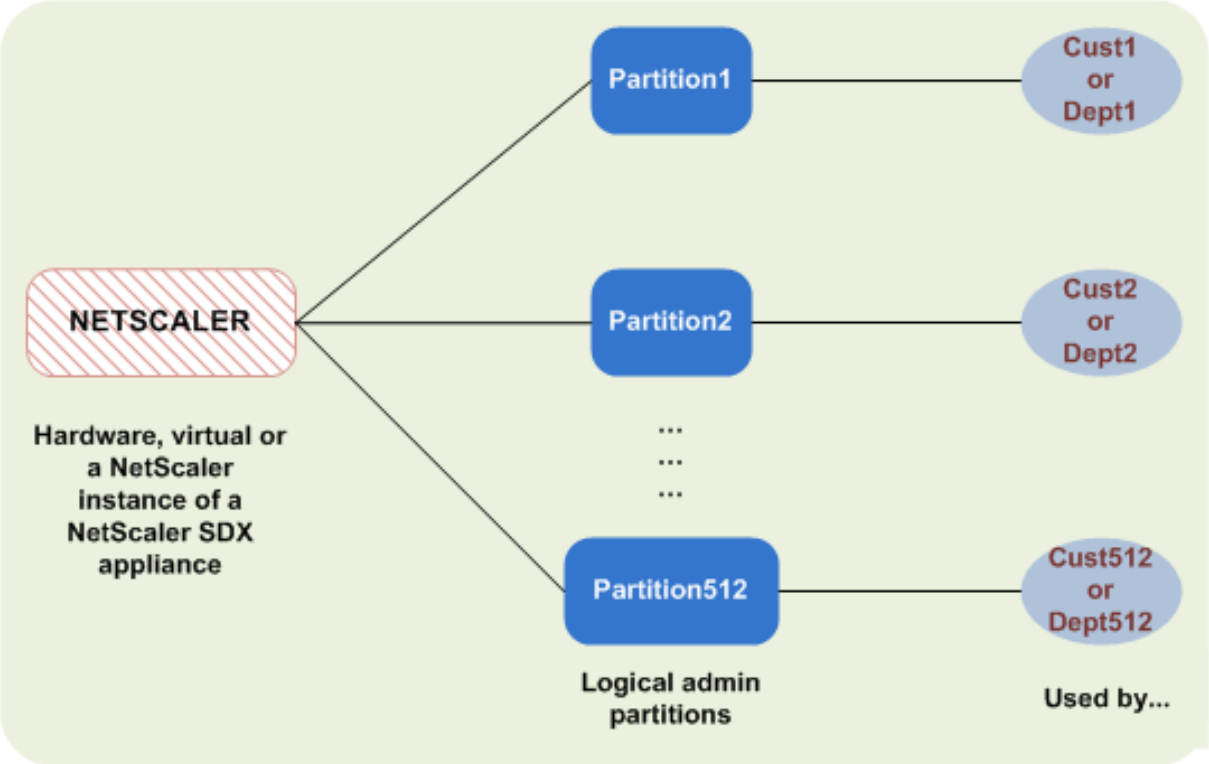
aaad.debug モジュールの認証に関する問題のトラブルシューティング

aaad.debug モジュールによる NetScaler および NetScaler Gateway の認証問題のトラブルシューティング

管理パーティション

August 15, 2023

NetScaler アプライアンスは、管理パーティションと呼ばれる論理エンティティに分割できます。各パーティションは、個別の NetScaler アプライアンスとして構成して使用できます。次の図は、さまざまな顧客や部門で使用されている NetScaler のパーティションを示しています。



パーティション化された NetScaler アプライアンスには、1 つのデフォルトパーティションと 1 つ以上の管理パーティションがあります。次の表に、2 つのパーティションタイプの詳細を示します。

> 注
>
> パーティション化されたアプライアンスでは、モード BridgeBPDU はデフォルトパーティションでのみ有効になり、管理パーティションでは有効にできません。

可用性:

NetScaler アプライアンスには、デフォルトパーティションと呼ばれる単一のパーティションが付属しています。NetScaler アプライアンスのパーティション分割後も、デフォルトパーティションは保持されます。

管理パーティションの構成の説明に従って、明示的に作成する必要があります。

パーティション数:

一つ

NetScaler アプライアンスには、1 つ以上（最大 512）の管理パーティションを設定できます。

ユーザーアクセスとロール:

パーティション固有のコマンドポリシーに関連付けられていないすべての *NetScaler* ユーザーは、デフォルトパーティションにアクセスして構成できます。いつものように、関連するコマンドポリシーは、ユーザーが実行できる操作を制限します。

ユーザーアクセスとロールは、そのパーティションのユーザーも指定する NetScaler スーパーユーザーによって作成されます。管理者パーティションにアクセスして構成できるのは、スーパーユーザーおよびパーティションの関連付けられたユーザーだけです。

> 注
>
> パーティションユーザーにはシェルアクセス権がありません。

ファイル構造:

デフォルトパーティション内のすべてのファイルは、デフォルトの NetScaler ファイル構造に保存されます。

たとえば、*/nsconfig* ディレクトリには *NetScaler* 構成ファイルが保存され、*/var/log/*ディレクトリには *NetScaler* ログが格納されます。

管理パーティション内のすべてのファイルは、管理パーティションの名前を持つディレクトリパスに格納されます。

たとえば、NetScaler 構成ファイル（ns.conf）はディレクトリに保存されます。/nsconfig/partitions/<partitionName> その他のパーティション固有のファイルは、/var/partitions/<partitionName> ディレクトリに格納されます。

管理パーティション内のその他のパスは次のとおりです。

- ダウンロードされたファイル: /var/partitions/<partitionName>/download/
- ログファイル: /var/partitions/<partitionName>/log/

> 注
>
> 現在、ロギングはパーティションレベルではサポートされていません。したがって、このディレクトリは空で、すべてのログが/var/log/ディレクトリに保存されます。

- SSL CRL 証明書関連ファイル: /var/partitions/<partitionName>/netscaler/ssl

利用可能なリソース:

すべての NetScaler リソース。

管理パーティションに明示的に割り当てられている NetScaler リソース。

ユーザーアクセスとロール

パーティション化された NetScaler アプライアンスを認証および承認する際、ルート管理者は 1 つ以上のパーティションにパーティション管理者を割り当てることができます。パーティション管理者は、他のパーティションに影響を与えずに、そのパーティションに対するユーザーを認証できます。パーティションユーザーは、SNIP アドレスを

使用してそのパーティションにのみアクセスする権限があります。ルート管理者とパーティション管理者の両方が、異なるアプリケーションへのアクセスをユーザーに許可することで、ロールベースのアクセス (RBA) を構成できます。

管理者とユーザーロールは、次のように記述できます。

ルート管理者。パーティション化されたアプライアンスに NSIP アドレスを使用してアクセスし、1 つ以上のパーティションへのユーザーアクセスを許可できます。管理者は、パーティション管理者を 1 つ以上のパーティションに割り当てることもできます。管理者は、NSIP アドレスを使用してデフォルトパーティションからパーティション管理者を作成するか、パーティションに切り替えて、ユーザーを作成し、SNIP アドレスを使用してパーティション管理者アクセスを割り当てることができます。

パーティション管理者。ルート管理者によって割り当てられた NSIP アドレスを使用して、指定されたパーティションにアクセスします。管理者は、パーティションのユーザーアクセスをパーティションにロールベースのアクセスを割り当てて、パーティション固有の設定を使用して外部サーバー認証を構成することもできます。

システムユーザー。NSIP アドレスを通してパーティションにアクセスします。ルート管理者によって指定されたパーティションとリソースにアクセスできます。

パーティションユーザー。SNIP アドレスを使用してパーティションにアクセスします。ユーザーアカウントはパーティション管理者によって作成され、ユーザーはパーティション内でのみリソースにアクセスできます。

確認事項

パーティションでロールベースのアクセスを提供する際に覚えておくべき点をいくつか挙げます。

1. NSIP アドレスを介して GUI にアクセスする NetScaler ユーザーは、デフォルトのパーティション認証構成を使用してアプライアンスにログオンします。
2. パーティション SNIP アドレスを使用して GUI にアクセスするパーティションシステムユーザは、パーティション固有の認証設定を使用してアプライアンスにログオンします。
3. パーティション内に作成されたパーティションユーザーは、NSIP アドレスを使用してログインできません。
4. パーティションにバインドされている NetScaler ユーザーは、パーティションの SNIP アドレスを使用してログインできません。
5. 外部認証サーバ（LDAP、RADIUS、TACACS など）を介して認証するシステムユーザは、SNIP アドレスを介してパーティションにアクセスする必要があります。

パーティション設定でのロールベースのアクセスを管理するためのユースケース

企業組織 www.example.com に複数のビジネスユニットがあり、ネットワーク内のすべてのインスタンスを管理する一元管理者がいるシナリオを考えてみましょう。ただし、各ビジネスユニットに排他的なユーザー権限と環境を提供したいと考えています。

パーティションアプライアンスでデフォルトのパーティション認証設定およびパーティション固有の設定で管理される管理者とユーザーを次に示します。

ジョン: ルート管理者

ジョージ: パーティション管理者

Adam: システムユーザー

Jane: パーティションユーザー

ジョンは、パーティション化された NetScaler アプライアンスのルート管理者です。John は、アプライアンス内のパーティション（P1、P2、P3、P4、P5 など）にわたってすべてのユーザーアカウントと管理ユーザーアカウントを管理します。John は、アプライアンスのデフォルトパーティションからエンティティへの詳細なロールベースのアクセスを提供します。John は、ユーザーアカウントを作成し、各アカウントにパーティションアクセスを割り当てます。組織内のネットワークエンジニアである George は、パーティション P2 で実行されているいくつかのアプリケーションに対して、ロールベースのアクセス権を持つことを好みます。ユーザー管理に基づいて、John は George のパーティション管理者ロールを作成し、P2 パーティション内の partition-admin コマンドポリシーに自分のユーザーアカウントを関連付けます。別のネットワークエンジニアであるアダムは、P2 で実行されているアプリケーションにアクセスすることを好みます。John は Adam のシステムユーザアカウントを作成し、ユーザアカウントを P2 パーティションに関連付けます。アカウントが作成されると、Adam はアプライアンスにログインして NSIP アドレスを介して NetScaler 管理インターフェイスにアクセスし、ユーザー/グループのバインディングに基づいてパーティション P2 に切り替えることができます。

別のネットワークエンジニアである Jane が、パーティション P2 でのみ実行されているアプリケーションに直接アクセスしたいとします。George（パーティション管理者）は、自分のパーティションユーザーアカウントを作成し、自分のアカウントを認可権限のコマンドポリシーに関連付けることができます。パーティション内に作成された Jane のユーザーアカウントが P2 に直接関連付けられるようになりました。これで、Jane は SNIP アドレスを介して NetScaler 管理インターフェイスにアクセスでき、他のパーティションに切り替えることはできません。

> 注
>
> Jane のユーザーアカウントがパーティション P2 のパーティション管理者によって作成された場合、管理者は（パーティション内で作成された）SNIP アドレスを介してのみ NetScaler 管理インターフェイスにアクセスできます。管理者は NSIP アドレスを介してインターフェイスにアクセスすることはできません。同様に、Adam のユーザーアカウントがデフォルトパーティションにルート管理者によって作成され、P2 パーティションにバインドされている場合です。管理者は、デフォルトパーティション（管理アクセスが有効になっている）に作成された NSIP アドレスまたは SNIP アドレスを介してのみ NetScaler 管理インターフェイスにアクセスできます。また、管理パーティションに作成された SNIP アドレスを介してパーティションインターフェイスにアクセスすることは許可されていません。

パーティション管理者の役割と責任を構成する

ルート管理者がデフォルトパーティションで実行する構成を次に示します。

管理パーティションとシステムユーザーの作成—ルート管理者は、アプライアンスのデフォルトパーティションに管理パーティションとシステムユーザーを作成します。その後、管理者はユーザーを異なるパーティションに関連付けます。1 つ以上のパーティションにバインドされている場合は、ユーザーバインディングに基づいて、あるパーティ

ションから別のパーティションに切り替えることができます。また、1 つ以上のバインドされたパーティションへのアクセスは、ルート管理者によってのみ許可されます。

システムユーザーを特定のパーティションのパーティション管理者として承認する—ユーザーアカウントを作成すると、ルート管理者は特定のパーティションに切り替え、そのユーザーをパーティション管理者として承認します。このためには、partition-admin コマンドポリシーをユーザアカウントに割り当てます。これで、ユーザーはパーティション管理者としてパーティションにアクセスし、パーティション内のエンティティを管理できます。

次に、管理パーティション内のパーティション管理者が実行する構成を示します。

管理パーティションでの SNIP アドレスの設定-パーティション管理者はパーティションにログオンし、SNIP アドレスを作成し、アドレスへの管理アクセスを提供します。

パーティション・コマンド・ポリシーを使用したパーティション・システム・ユーザーの作成とバインド-パーティション管理者はパーティション・ユーザーを作成し、ユーザー・アクセスの範囲を定義します。このためには、ユーザーアカウントをパーティションコマンドポリシーにバインドします。

パーティションコマンドポリシーを使用したパーティションシステムユーザーグループの作成とバインド-パーティション管理者は、パーティションユーザーグループを作成し、ユーザーグループアクセスの範囲を定義します。これは、ユーザグループアカウントをパーティションコマンドポリシーにバインドすることによって行われます。

外部ユーザの外部サーバ認証の設定（オプション）：この設定は、SNIP アドレスを使用してパーティションにアクセスする外部 TACACS ユーザを認証するために行われます。

次に、管理パーティション内のパーティションユーザーに対するロールベースのアクセスの構成で実行されるタスクを示します。

1. 管理パーティションの作成—管理パーティションにパーティションユーザーを作成する前に、まずパーティションを作成する必要があります。ルート管理者は、構成ユーティリティまたはコマンドラインインターフェイスを使用して、デフォルトパーティションからパーティションを作成できます。
2. デフォルトパーティションからパーティション P2 へのユーザーアクセスの切り替え：デフォルトパーティションからアプライアンスにアクセスするパーティション管理者は、デフォルトパーティションから特定のパーティションに切り替えることができます。たとえば、ユーザーバインドに基づいて P2 をパーティション分割します。
3. 管理アクセス権が有効なパーティションユーザーアカウントに SNIP アドレスを追加する-管理パーティションへのアクセスを切り替えたら。SNIP アドレスを作成し、そのアドレスへの管理アクセスを提供します。
4. パーティションコマンドポリシーを使用したパーティションシステムユーザーの作成とバインド-パーティション管理者である場合は、パーティションユーザーを作成し、ユーザーアクセスの範囲を定義できます。このためには、ユーザーアカウントをパーティションコマンドポリシーにバインドします。
5. partition コマンドポリシーを使用したパーティション・ユーザー・グループの作成とバインド-パーティション管理者の場合は、パーティション・ユーザー・グループを作成し、ユーザー・アクセス制御の範囲を定義できます。これは、ユーザグループアカウントをパーティションコマンドポリシーにバインドすることによって行われます。

外部ユーザの外部サーバ認証の設定（オプション）：この設定は、SNIP アドレスを使用してパーティションにアクセ

スする外部 TACACS ユーザを認証するために行われます。

管理パーティションを使用する利点

デプロイメントに管理パーティションを使用すると、次のメリットを利用できます。

- アプリケーションの管理所有権を顧客に委任できます。
- パフォーマンスと使いやすさを損なうことなく、ADC の所有コストを削減します。
- 不当な構成変更から保護します。パーティション化されていない NetScaler アプライアンスでは、他のアプリケーションの権限のあるユーザーが、アプリケーションに必要な構成を意図的または意図せずに変更する可能性があります。望ましくない動作につながる可能性があります。パーティション化された NetScaler アプライアンスでは、この可能性は低くなります。
- パーティションごとに専用 VLAN を使用して、異なるアプリケーション間のトラフィックを分離します。
- アプリケーションのデプロイを高速化し、拡張できるようにします。
- アプリケーションレベルまたはローカライズされた管理とレポート作成を許可します。

いくつかのケースを分析して、管理パーティションを使用できるシナリオを理解しましょう。

ユーザーケース **1:** エンタープライズネットワークでの管理パーティションの使用方法

Foo.com という会社が直面するシナリオを考えてみましょう。

- **Foo.com** には **1** つの **NetScaler** があります。
- 5 つの部門があり、各部門には NetScaler で展開する必要のあるアプリケーションが 1 つあります。
- 各アプリケーションは、異なるユーザーまたは管理者によって個別に管理する必要があります。
- 他のユーザーは、設定へのアクセスを制限する必要があります。
- アプリケーションまたはバックエンドは、IP アドレスなどのリソースを共有できる必要があります。
- グローバル IT 部門は、すべてのパーティションに共通でなければならない NetScaler レベルの設定を管理できなければなりません。
- アプリケーションは互いに独立している必要があります。一方のアプリケーションの構成エラーは、他方のアプリケーションに影響してはいけません。

パーティション化されていない NetScaler では、これらの要件を満たすことができません。ただし、NetScaler をパーティション化することで、これらすべての要件を満たすことができます。

アプリケーションごとにパーティションを作成し、必要なユーザをパーティションに割り当てて、各パーティションの VLAN を指定し、デフォルトパーティションでグローバル設定を定義するだけです。

ユースケース **2:** サービスプロバイダによる管理パーティションの使用方法

BigProvider というサービスプロバイダーが直面するシナリオを考えてみましょう。

- BigProvider には 5 つの顧客があります。3 つの小企業と 2 つの大企業です。
- **SmallBiz**、**SmallerBiz**、**StartupBiz** は、最も基本的な NetScaler ADC 機能のみを必要とします。
- ****BigBiz** と **LargeBiz** は大企業であり ** 、大量のトラフィックを引き付けるアプリケーションを持っています。彼らは、より複雑な NetScaler 機能のいくつかを使用したいと考えています。

分割しないアプローチでは、NetScaler 管理者は通常、NetScaler SDX アプライアンスを使用して、顧客ごとに NetScaler インスタンスをプロビジョニングします。

このソリューションは **BigBiz** と **LargeBiz** に適しています。なぜなら、**BigBiz** や **LargeBiz** のアプリケーションには、パーティション化されていない NetScaler アプライアンス全体の処理能力が衰えない必要があるからです。ただし、このソリューションは、SmallBiz、**SmallerBiz**、および ****StartupBiz**** のサービスにはそれほど費用対効果がない可能性があります。

したがって、**BigProvider** は次の解決策を決定します。

- NetScaler SDX アプライアンスを使用して、****BigBiz** および **LargeBiz** 専用の **Citrix**ADC インスタンスを起動する。
- 単一の NetScaler ADC を使用して、SmallBiz、**SmallerBiz**、****StartupBiz** にそれぞれ **1** つずつ **、3 つのパーティションに分割されます。

NetScaler 管理者（スーパーユーザー）は、これらの顧客ごとに管理パーティションを作成し、パーティションのユーザーを指定します。また、パーティションの NetScaler リソースを指定し、各パーティション宛てのトラフィックが使用する VLAN を指定します。

管理パーティションでの **NetScaler** 構成サポート

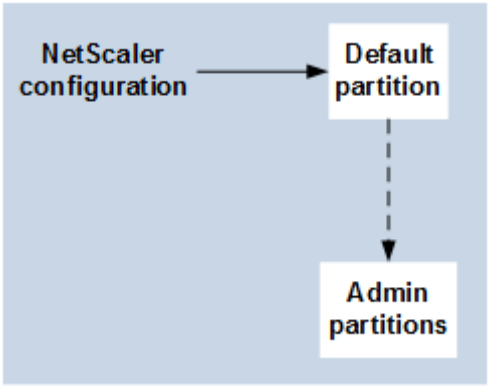
October 25, 2023

NetScaler の構成は、次の 3 種類の構成に分類できます。これは、Citrix の構成と構成が実行されるパーティションによって異なります。

> 注
>
> - 管理パーティションは NetScaler クラスターには設定できません。つまり、NetScaler クラスターはパーティション化できません。
> - NetScaler 14000 FIPS アプライアンスでは管理パーティションを設定できません。
> - ケース 3 は 、管理パーティションでサポートされていない NetScaler ADC 機能を示しています。
> - 負荷分散テンプレートは、管理パーティションではサポートされません。

ケース **1 (**グローバル構成**)**

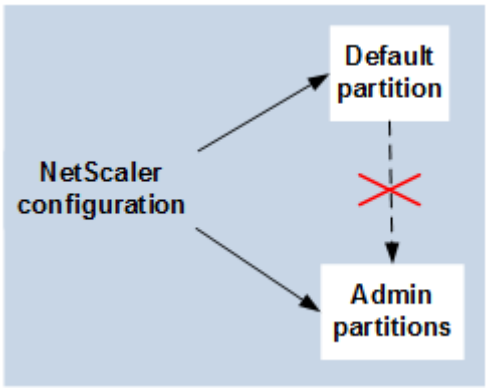
デフォルトパーティションでのみ実行でき、使用可能な設定、またはすべての管理パーティションに影響する設定。



- モニタ、TCP プロファイル、HTTP プロファイルなどの組み込みエンティティの更新。
- syslog、NSLOG、ウェブログ、コンテンツスイッチング、IPSEC、SIP、DHCP、サージ保護、TCP バッファリング、およびシステム収集のグローバルパラメータの更新。
- 高可用性（HA）構成
- インターフェイスと VLAN の変更
- ユーザー構成

ケース **2 (**パーティション固有の構成**)**

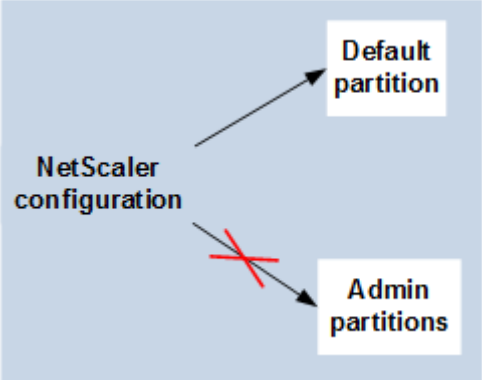
デフォルトパーティションと管理パーティションで独立して実行できる設定。これらの構成は、実行されているパーティションにのみ適用されます。



- パーティションのトラフィックレベルの統計情報の取得。
- パーティション管理者は、そのパーティションにバインドされている VLAN の IP バインディングを更新できます。ただし、インターフェイスバインディングを更新することはできません。
- NetScaler 構成のクリア。
- AppFlow、AppQoE、HTTP 圧縮、DNS、TCP、HTTP、暗号化、レスポンダ、書き換え、SSL の機能固有のパラメーター。
- 仮想サーバ、サービス、モニタなどの機能固有の構成。

ケース **3**

管理パーティションでは実行できない設定。これらの機能はデフォルトパーティションで設定できますが、管理パーティションには影響しません。



> 注:
>
> 特定のリリースの管理パーティションでサポートされている設定は、**Yes** とマークされます。

フィーチャコンポーネント	NetScaler 機能	NetScaler 12.1	NetScaler 13.0	NetScaler 13.1	NetScaler 14.1
ネットワーク	トラフィックドメイン	番号	番号	番号	番号
ポリシー	拡張性	はい	はい	はい	はい
負荷分散	DBS Autoscale	はい	はい	はい	はい
負荷分散	DNSSEC	はい	はい	はい	はい
負荷分散	Diameter	はい	はい	はい	はい
負荷分散	RTSP	番号	番号	番号	番号
負荷分散	Sure Connect	廃止済み	廃止済み	削除	削除
負荷分散	オートスケールサービスグループ	はい	はい	はい	はい
管理性	RBA 外部認証	はい	はい	はい	はい
管理性	RISE Cisco	番号	はい	はい	はい
管理性	ACI-Cisco	はい	はい	はい	はい
管理性	AppExpert	はい	はい	はい	はい
管理性	HDX Insight	番号	番号	番号	番号

フィーチャコンポーネント	NetScaler 機能	NetScaler 12.1	NetScaler 13.0	NetScaler 13.1	NetScaler 14.1
管理性	インサイト	番号	番号	番号	番号
VPN	Citrix CloudBridge Connector	番号	番号	番号	番号
VPN	NetScaler Gateway または SSL VPN	番号	番号	番号	番号
VPN	SSL VPN ICA プロキシ	番号	番号	番号	番号
VPN	NetScaler のウェブインターフェイス	番号	番号	番号	番号
SSL	SSL プロファイル	はい	はい	はい	はい
SSL	SSL-FIPS	番号	番号	番号	番号
SSL	外部 HSM	番号	番号	番号	番号
インフラ	キャッシュリダイレクト	番号	番号	番号	番号
インフラ	統合キャッシング	はい	はい	はい	はい
ネットワーク	VXLAN	はい	はい	はい	はい
ネットワーク	グレースフルシャットダウン	はい	はい	はい	はい
ネットワーク	LSN	番号	番号	番号	番号
ネットワーク	IPv6 レディロゴ	はい	はい	はい	はい
ネットワーク	vPath	はい	はい	はい	はい
負荷分散	Datastream	はい	はい	はい	はい
ログ	Web ログ	はい	はい	はい	はい
ネットワーク	L2 Param/L3 Param	はい	はい	はい	はい
ネットワーク	GRE トンネル	はい	はい	はい	はい
ローディングバランシング	スクリプト可能なモニタリング	はい	はい	はい	はい
負荷分散	GSLB か	はい	はい	はい	はい

フィーチャコンポーネント	NetScaler 機能	NetScaler 12.1	NetScaler 13.0	NetScaler 13.1	NetScaler 14.1
インフラ	接続ミラーリング	はい	はい	はい	はい
インフラ	FEO	はい	はい	はい	はい
インフラ	Ns トレース	はい	はい	はい	はい
負荷分散	優先度によるキューイング	廃止済み	廃止済み	削除	削除
ネットワーク	HDOSP	廃止済み	廃止済み	削除	削除
ネットワーク	ネットプロファイル	はい	はい	はい	はい
ネットワーク	ネットワーキング (制限付き機能)	はい	はい	はい	はい
ネットワーク	VRRP (制限付き機能)	はい	はい	はい	はい
ログ	監査ロギング (SYSLOG-TCP、Syslog サーバの LB、SNIP サポート、および Syslog の FQDN サポート)	はい	はい	はい	はい
VPN	NetScaler Gateway	番号	番号	番号	番号
VPN	AAA-TM	はい	はい	はい	はい
AppFlow	AppFlow	はい (IPFIX のみ)	はい	はい	はい
AppFw	アプリケーションファイアウォール	番号	番号	番号	番号
URL 変換	URL 変換	番号	番号	番号	番号
負荷分散	TCP バッファリング	番号	番号	番号	番号
ポリシー	OCSP レスポンダー	はい	はい	はい	はい
監査ログ	SYSLOG-TCP	はい	はい	はい	はい

フィーチャコンポーネント	NetScaler 機能	NetScaler 12.1	NetScaler 13.0	NetScaler 13.1	NetScaler 14.1
最適化	フロントエンド最適化	はい	はい	はい	はい
AppQoE	AppQoE	はい	はい	はい	はい

前の表では、管理パーティション設定の「制限付き機能」としての機能の一部を示します。次のセクションでは、一部の機能が制限付き機能として言及されている理由について説明します。

- **VRRP**。VRRP は、次の理由により、管理パーティションの [制限付き機能] です。
  - VRID の追加または削除は、デフォルトのパーティションコンテキストからのみ実行できます。ただし、VRID が作成されると、デフォルト以外のパーティション内で使用できます。
  - VRRP 機能は、専用 VLAN でのみサポートされます。
  - VRRP 機能は、管理パーティションで使用される共有 VLAN ではサポートされません。内部でブロックされています。設定中にエラーメッセージは表示されません。プロトコルは、デフォルトパーティションまたは任意の管理パーティションにバインドされた共有 VLAN（タグ付きまたはタグなし）でブロックされます。

> 重要
>
> VRRP を使用したアクティブ/アクティブ展開をサポートするには、メイン VIP とバックアップ VIP で同じ VRID を使用する必要があります。異なる VRD は使用できません。

- ネットワーキング。一部のネットワーク構成（L2 Param および L3 Param）は、パーティションコンテキストでサポートされていないか、有効ではありません。このような構成に遭遇すると、次のエラーメッセージが表示されます。「エラー: この設定オプションは、デフォルト以外のパーティションではサポートされていません。」

管理パーティションを構成する

August 15, 2023

> 重要
>
> - 管理パーティションの作成と構成は、スーパーユーザーのみに許可されます。
> - 特に指定がない限り、管理パーティションをセットアップするための設定は、デフォルトパーティションから行う必要があります。

NetScaler アプライアンスをパーティション化すると、1 つの NetScaler アプライアンスの複数のインスタンスが作成されます。各インスタンスには独自の設定があり、これらの各パーティションのトラフィックは他のパーティションから分離されます。これは、各パーティションに専用 VLAN または共有 VLAN を割り当てることによって行われます。

パーティション化された NetScaler には、1 つのデフォルトパーティションと作成される管理パーティションがあります。管理パーティションを設定するには、まず、関連するリソース（メモリ、最大帯域幅、および接続）を持つパーティションを作成する必要があります。次に、パーティションにアクセスできるユーザーと、パーティション上の各ユーザーの承認レベルを指定します。

パーティション化された NetScaler へのアクセスは、パーティション化されていない NetScaler へのアクセスと同じで、NSIP アドレスまたはその他の管理 IP アドレスを介してアクセスします。ユーザーとして、有効なログオン資格情報を入力すると、バインド先のパーティションに連れて行かれます。作成した構成は、そのパーティションに保存されます。複数のパーティションに関連付けられている場合は、関連付けられた最初のパーティションが表示されます。他のパーティションの 1 つにエンティティを構成する場合は、そのパーティションに明示的に切り替える必要があります。

適切なパーティションにアクセスすると、実行する構成はそのパーティションに保存され、そのパーティションに固有になります。

> 注
>
> - NetScaler のスーパーユーザーやその他のパーティション以外のユーザーは、デフォルトパーティションに移動します。
> - 512 パーティションのユーザーは同時にログインできます。

ヒント

SNIP を使用して（管理アクセスが有効になっている状態で）パーティション化された NetScaler アプライアンスに HTTPS 経由でアクセスするには、各パーティションにパーティション管理者の証明書があることを確認してください。パーティション内では、パーティション管理者は次の操作を行う必要があります。

1. 証明書を NetScaler に追加します。

   ```
   add ssl certKey ns-server-certificate -cert ns-server.cert-key ns
   -server.key
   ```

2. これはサービス`nshttps-<SNIP>-3009`にバインドされます。`<SNIP>`は SNIP アドレス（この場合は 100.10.10.1）に置き換える必要があります。

   ```
   bind ssl service nshttps-100.10.10.1-3009 -certkeyName ns-server-
   certificate
   ```


パーティションリソースの制限

パーティション化された NetScaler アプライアンスでは、ネットワーク管理者は、メモリ、帯域幅、接続制限などのパーティションリソースを無制限に設定したパーティションを作成できます。これは、パーティションリソースの値として Zero を指定することによって行われます。ゼロは、パーティション上でリソースが無制限であり、システム制限まで消費できることを示します。パーティションリソース構成は、トラフィックドメインの展開を管理パーティションに移行する場合や、特定の展開内のパーティションのリソース割り当て制限がわからない場合に便利です。

管理パーティションのリソース制限は次のとおりです。

1. パーティションメモリ。これは、パーティションに割り当てられる最大メモリです。パーティションを作成するときは、必ず値を指定してください。

   > 注
   >
   > NetScaler 12.0 以降では、パーティションを作成するときに、メモリ制限をゼロに設定できます。特定のメモリ制限を使用してパーティションがすでに作成されている場合は、制限を任意の値に減らすか、制限をゼロに設定できます。

   パラメータ:maxMemLimit

   パーティション内の最大メモリは MB 単位で割り当てられます。ゼロの値は、パーティション上のメモリが無制限であり、システム制限まで消費できることを示します。

   デフォルト値:10

2. パーティション帯域幅。パーティションに割り当てられる最大帯域幅。制限を指定する場合は、アプライアンスのライセンススループット内であることを確認してください。それ以外の場合、パーティションで使用される帯域幅は制限されません。指定された制限は、アプリケーションが必要とする帯域幅について責任があります。アプリケーションの帯域幅が指定された制限を超えると、パケットはドロップされます。

   > 注
   >
   > NetScaler 12.0 以降では、パーティションを作成できるときに、パーティションの帯域幅制限をゼロに設定できます。パーティションが特定の帯域幅で既に作成されている場合は、帯域幅を減らすか、制限をゼロに設定できます。

   パラメータ: 最大帯域幅

   パーティション内の最大帯域幅は Kbps 単位で割り当てられます。ゼロの値は、帯域幅が制限されていないことを示します。つまり、パーティションはシステム制限まで消費する可能性があります。

   デフォルト値:10240

   最大値:4294967295

3. パーティション接続。パーティションで開くことができる同時接続の最大数。この値は、パーティション内で予想される最大同時フローに対応する必要があります。パーティション接続は、パーティションクォータメモ

リから計算されます。以前は、接続はデフォルトのパーティションクォータメモリから計算されていました。これは、バックエンドのサーバー側の TCP 接続ではなく、クライアント側でのみ構成されます。この設定値を超えると、新しい接続を確立できません。

> 注
>
> NetScaler 12.0 以降では、開いている接続数がゼロに設定されたパーティションを作成できます。特定の数のオープン接続を持つパーティションをすでに作成している場合は、接続制限を減らすか、制限をゼロに設定できます。

パラメータ: 最大接続

パーティションで開くことができる同時接続の最大数。ゼロの値は、開いている接続の数に制限がないことを示します。

デフォルト値:1024

最小値:0

最大値:4294967295

管理パーティションを構成する

管理パーティションを設定するには、次のタスクを実行します。

CLI を使用して管理パーティションでアクセスするには

1. NetScaler アプライアンスにログオンします。
2. 正しいパーティションに入っているかどうかを確認します。コマンドプロンプトに、現在選択されているパーティションの名前が表示されます。
3. はいの場合は、次のステップに進みます。
4. いいえ、関連付けられているパーティションのリストを取得し、適切なパーティションに切り替えます。
   - `show system user <username>`
   - **`switch`** `ns partition <partitionName>`
5. これで、必要な構成をパーティション化されていない NetScaler と同じように実行できます。

GUI を使用して管理パーティションにアクセスするには

1. NetScaler アプライアンスにログオンします。
2. 正しいパーティションに入っているかどうかを確認します。GUI の上部バーには、現在選択されているパーティションの名前が表示されます。

- はいの場合は、次のステップに進みます。
- [いいえ] の場合は、[構成] **>** [システム] **>** [パーティション管理] ****[**パーティション] に移動し、切り替えるパーティションを右クリックし、**[**** 切り替え] を選択します。

3. これで、必要な構成をパーティション化されていない NetScaler と同じように実行できます。

管理者パーティションを追加する

ルート管理者は、デフォルトパーティションから管理パーティションを追加し、そのパーティションを VLAN 2 にバインドします。

CLI を使用して管理パーティションを作成するには

コマンドプロンプトで入力します。

```
1 add partition <partitionname>
```

ユーザーアクセスをデフォルトパーティションから管理パーティションに切り替える

これで、ユーザーアクセスをデフォルトパーティションからパーティション Par1 に切り替えることができます。

CLI を使用してユーザアカウントをデフォルトパーティションから管理パーティションに切り替えるには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 Switch ns partition <pname>
```

管理アクセスが有効になっているパーティションユーザーアカウントに **SNIP** アドレスを追加する

パーティションで、管理アクセスが有効な SNIP アドレスを作成します。

コマンドラインインターフェイスを使用して管理アクセスが有効になっているパーティションユーザーアカウントに **SNIP** アドレスを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
> add ns ip <ip address> <subnet mask> -mgmtAccess enabled
```


パーティションコマンドポリシーを使用してパーティションユーザーを作成してバインドする

パーティションで、パーティションシステムユーザーを作成し、partition-admin コマンドポリシーでユーザーをバインドします。

CLI を使用して **partition** コマンドポリシーを使用してパーティションシステムユーザーを作成してバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
> add system user <username> <password>

Done
```

partition コマンドポリシーを使用したパーティションユーザーグループの作成とバインド

パーティション Par1 で、パーティションシステムユーザーグループを作成し、パーティション管理、パーティション読み取り専用、パーティションオペレータ、パーティションネットワークなどのパーティションコマンドポリシーを使用してグループをバインドします。

コマンドラインインターフェイスを使用して、パーティションコマンドポリシーを使用してパーティションユーザーグループを作成し、バインドするには:

```
1 > add system group <groupName>
2 > bind system group <groupname> \(-userName | -policyName <cmdpolicy> <
    priority> | -partitionName)
```

外部ユーザの外部サーバ認証の設定

パーティション Par1 では、外部サーバ認証を設定して、SNIP アドレスを介してパーティションにアクセスする外部 TACACS ユーザを認証できます。

コマンドラインインターフェイスを使用して外部ユーザーの外部サーバー認証を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 > add authentication tacacsaction <name> -serverip <IP> -tacacsSecret <
    secret key> -authorization ON -accounting ON
2 > add authentication policy <policname> -rule true -action <name>
3 > bind system global <policyname> -priority <value>1
```

GUI を使用して、パーティション内のパーティションシステムユーザーアカウントを構成します

管理パーティションでパーティションユーザーアカウントを構成するには、パーティションユーザーまたはパーティションユーザーグループを作成し、パーティションコマンドポリシーをバインドする必要があります。また、外部ユ

ーザの外部サーバ認証を構成することもできます。

GUI を使用してパーティションにパーティションユーザーアカウントを作成するには

[システム] **>** [ユーザー管理] に移動し、[ユーザー] をクリックしてパーティションシステムユーザーを追加し、コマンドポリシー (partitionadmin/partition/partition-ad-only/パーティション演算子/パーティションネットワーク) にユーザーをバインドします。

GUI を使用してパーティションにパーティションのユーザーグループアカウントを作成するには

[システム] **>** [ユーザー管理] に移動し、[グループ] をクリックしてパーティションシステムのユーザーグループを追加し、ユーザーグループをコマンドポリシー (partitionadmin/partition/partition-read-only/パーティション演算子/パーティションネットワーク) にバインドします。

GUI を使用して外部ユーザの外部サーバ認証を設定するには

[システム] **>** [認証] **>** [基本アクション] に移動し、[**TACACS**] をクリックして、パーティションにアクセスする外部ユーザを認証するための TACACS サーバを設定します。

構成例

次の構成では、パーティションユーザーまたはパーティションユーザーグループを作成し、パーティションコマンドポリシーをバインドする方法を示します。また、外部ユーザを認証するための外部サーバ認証の設定方法についても説明します。

```
1  > add partition Par1
2  > switch ns partition Par1
3  > add ns ip 10.102.29.203 255.255.255.0 -mgmtAccessenabled
4  > add system user John Password
5  > bind system user Jane partition-read-only -priority 1
6  > add system group Retail
7  > bind system group Retail -policyname partition-network 1 (where 1 is
       the priority number)
8  > bind system group Retail  - username Jane
9  > add authentication tacacssaction tacuser  - serverip 10.102.29.200  -
       tacacsSecret Password  - authorization ON  - accounting ON
10 > add authentication policy polname  - rule true  - action tacacsAction
11 > bind system global polname  - priority 1
```

管理パーティション内のパーティションユーザーおよびパーティションユーザーグループのコマンドポリシー

管理パーティション内のユーザアカウントを認証するコマンド	管理パーティション内で使用可能なコマンドポリシー (組み込みポリシー)	ユーザーアカウントアクセスタイプ
add system user	パーティション管理	SNIP（管理アクセスが有効になっている場合）
add system group	パーティションネットワーク	SNIP（管理アクセスが有効になっている場合）
add authentication <action, policy>, bind system global <policy name>	パーティション-読み取り専用	SNIP（管理アクセスが有効になっている場合）
remove system user	パーティション管理	SNIP（管理アクセスが有効になっている場合）
remove system group	パーティション管理	SNIP（管理アクセスが有効になっている場合）
bind system cmdpolicy to system user; bind system cmdpolicy to system group	パーティション管理	SNIP（管理アクセスが有効になっている場合）

デフォルトの管理パーティションで **LACP** イーサネットチャネルを設定します

リンク集約制御プロトコル（LACP）を使用すると、複数のポートを単一の高速リンク（チャネルとも呼ばれる）に結合できます。LACP 対応アプライアンスは、チャネルを介して LACP データユニット（LACPDU）を交換します。

NetScaler アプライアンスのデフォルトパーティションで有効にできる LACP 構成モードは 3 つあります。

1. アクティブ。アクティブモードのポートは LACPDU を送信します。リンクアグリゲーションは、イーサネットリンクのもう一方の端が LACP アクティブモードまたはパッシブモードにある場合に形成されます。
2. パッシブ。パッシブモードのポートは、lacPDU を受信したときにのみ lacPDU を送信します。リンクアグリゲーションは、イーサネットリンクのもう一方の端が LACP アクティブモードにある場合に形成されます。
3. 無効化。リンクアグリゲーションは形成されません。

> 注
>
> デフォルトでは、リンクアグリゲーションはアプライアンスのデフォルトパーティションで無効になっています。

LACP は、イーサネットリンクによって接続されたデバイス間で LACPDU を交換します。これらのデバイスは、通常、アクターまたはパートナーと呼ばれます。

LACPDU データユニットには、次のパラメータが含まれています。

- LACP モード。アクティブ、パッシブ、または無効。
- LACP タイムアウト。パートナーまたは俳優がタイムアウトするまでの待機期間。指定可能な値: ロングとショート。デフォルト: 長い。
- ポートキー。異なるチャネルを区別する。キーが 1 の場合、LA/1 が作成されます。キーが 2 の場合、LA/2 が作成されます。指定可能な値:1 から 8 までの整数。4 ～8 はクラスタ CLAG 用です。
- ポートプライオリティ。最小値:1。最大値:65535 デフォルト:32768。
- システム優先度。このプライオリティをシステム MAC とともに使用して、パートナーとの LACP ネゴシエーション中にシステムを一意に識別するシステム ID を形成します。システムプライオリティを 1 および 65535 から設定します。デフォルト値は 32768 に設定されています。
- インターフェイス。NetScaler 10.1 アプライアンスではチャネルごとに 8 つのインターフェイスをサポートし、NetScaler 10.5 および 11.0 アプライアンスではチャネルごとに 16 インターフェイスをサポートします。

LACPDU を交換した後、アクターとパートナーは設定をネゴシエートし、集約にポートを追加するかどうかを決定します。

LACP の設定と確認

次のセクションでは、管理パーティションで LACP を設定して検証する方法について説明します。

CLI を使用して **NetScaler** アプライアンスの **LACP** を構成および検証するには

1. 各インターフェイスで LACP を有効にします。

   ```
   set interface <Interface_ID> -lacpMode PASSIVE -lacpKey 1<!--NeedCopy-->
   ```

   インターフェイスで LACP を有効にすると、チャネルが動的に作成されます。また、インターフェイスで LACP を有効にして lacpKey を 1 に設定すると、インターフェイスはチャネル LA/1 に自動的にバインドされます。

   > 注
   >
   > インターフェイスをチャネルにバインドすると、チャネルパラメータがインターフェイスパラメータよりも優先されるため、インターフェイスパラメータは無視されます。チャネルが LACP によって動的に作成される場合、チャネルに対して追加、バインド、アンバインド、または削除の各操作を実行できません。LACP によって動的に作成されたチャネルは、チャネルのすべてのインターフェイスで LACP を無効にすると、自動的に削除されます。

2. システムプライオリティを設定します。

   ```
   set lacp -sysPriority <Positive_Integer><!--NeedCopy-->
   ```

3. LACP が期待どおりに動作していることを確認します。

   “‘show interface


```
1 ```show channel<!--NeedCopy-->
```

show LACP<!--NeedCopy-->

> 注
>
> Cisco インターネットワークオペレーティングシステム（iOS）の一部のバージョンでは、switchport trunk native VLAN <VLAN_ID> コマンドを実行すると、Cisco スイッチに LACP PDU のタグが付けられます。これにより、Cisco スイッチと NetScaler ADC アプライアンス間の LACP チャネルに障害が発生します。ただし、この問題は、前の手順で設定したスタティックリンク集約チャネルには影響しません。

デフォルトパーティションのすべての管理パーティションの設定を保存する

管理者は、デフォルトパーティションからすべての管理パーティションの設定を一度に保存できます。

CLI を使用して、デフォルトパーティションからすべての管理パーティションを保存する

コマンドプロンプトで入力します。

```
save ns config -all
```

パーティションおよびクラスタベースのカスタムレポートのサポート

NetScaler GUI には、現在の表示パーティションまたはクラスターで作成されたカスタムレポートのみが表示されます。

以前は、NetScaler GUI では、区別するパーティションやクラスター名を指定せずに、カスタムレポート名をバックエンドファイルに直接保存していました。

GUI で現在のパーティションまたはクラスタのカスタムレポートを表示するには

- [レポート] タブに移動します。
- [カスタムレポート] をクリックして、現在のパーティションまたはクラスタで作成されたレポートを表示します。

OAuth IdP のパーティション設定で **VPN** グローバル証明書をバインドするサポート

パーティション設定で、OAuth IdP 展開用の VPN グローバルに証明書をバインドできるようになりました。

CLI を使用してパーティション設定で証明書をバインドするには

コマンドプロンプトで入力します。

```
bind vpn global [-certkeyName <string>] [-userDataEncryptionKey <string
    >]
```

管理パーティションの **VLAN** 構成

August 15, 2023

VLAN は、「専用」VLAN または「共有」VLAN としてパーティションにバインドできます。導入環境に基づいて、VLAN をパーティションにバインドして、そのネットワークトラフィックを他のパーティションから分離できます。

専用 **VLAN**：「共有」オプションが無効になっている 1 つのパーティションにのみバインドされた VLAN で、タグ付き VLAN である必要があります。たとえば、クライアント/サーバ配置では、セキュリティ上の理由から、システム管理者はサーバ側のパーティションごとに専用 VLAN を作成します。

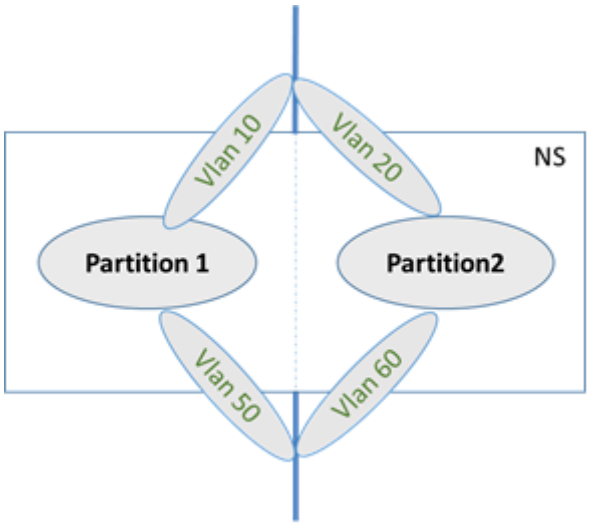
共有 **VLAN**：「共有」オプションが有効になっている複数のパーティションにバインドされた VLAN（全体で共有）。たとえば、クライアント/サーバ配置では、システム管理者がクライアント側のネットワークを制御できない場合、VLAN が作成され、複数のパーティションで共有されます。

共有 VLAN は複数のパーティションで使用できます。デフォルトパーティションに作成され、共有 VLAN を複数のパーティションにバインドできます。デフォルトでは、共有 VLAN はデフォルトパーティションに暗黙的にバインドされるため、明示的にバインドすることはできません。

> メモ
>
> - ハイパーバイザー（ESX、KVM、Xen、Hyper-V）プラットフォームに導入された NetScaler アプライアンスは、パーティション設定とトラフィックドメインの次の条件の両方を満たす必要があります。
>   - Enable the promiscuous mode, MAC changes, MAC spoofing, or forged transmit for shared VLANs with partition.
>   - Enable the VLAN with port group properties of the virtual switch, if the traffic is through a dedicated VLAN.
> - パーティショニング（マルチテナント）の NetScaler アプライアンスでは、システム管理者は特定のパーティションに流れるトラフィックを分離できます。これは、1 つ以上の VLAN を各パーティションにバインドすることによって行われます。VLAN は、1 つのパーティション専用にすることも、複数のパーティション間で共有することもできます。
> - 同じ NetScaler アプライアンスでホストされているパーティション間の内部ルーティングはサポートされていません。

専用 **VLAN**

パーティションに流れるトラフィックを分離するには、VLAN を作成し、それをパーティションに関連付けます。これにより、VLAN は関連付けられたパーティションにだけ認識され、VLAN を通過するトラフィックは、関連付けられたパーティションでのみ分類および処理されます。



特定のパーティションに専用 VLAN を実装するには、次の手順を実行します。

1. VLAN（V1）を追加します。
2. ネットワークインターフェースをタグ付きネットワークインターフェースとして VLAN にバインドします。
3. パーティション (P1) を作成します。
4. パーティション（P1）を専用 VLAN（V1）にバインドします。

CLI を使用して次の設定を行います

- VLAN を作成する

```
add vlan <id>
```

例

```
1      add vlan 100
```

- VLAN をバインドする

```
bind vlan <id> -ifnum <interface> -tagged
```

例

```
1      bind vlan 100  - ifnum 1/8 -tagged
```


- パーティションを作成する

```
Add ns partition <partition name> [-maxBandwidth <positive_integer
>][-maxConn <positive_integer>] [-maxMemLimit <positive_integer>]
```

例

```
1     Add ns partition P1  -maxBandwidth 200  -maxconn 50  -maxmemlimit
         90
2
3     Done
```

- パーティションを VLAN にバインドする

```
bind partition <partition-id> -vlan <id>
```

例

```
1     bind partition P1  -vlan 100
```

NetScaler GUI を使用して専用 VLAN を構成する

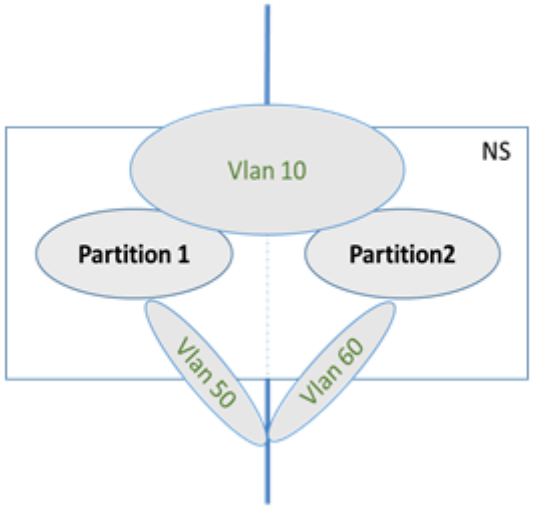
1. 設定 **>** システム **>** ネットワーク **> VLAN*** に移動し、VLAN を作成するために『**Add** 』をクリックして下さい。
2. **[VLAN** の作成（**Create VLAN** ）] ページで、次のパラメータを設定します。
   - VLAN ID
   - エイリアス名
   - 最大伝送ユニット
   - 動的ルーティング
   - IPv6 ダイナミックルーティング
   - パーティション共有
3. [インターフェイスバインディング（**Interface Bindings** ）] セクションで、1 つ以上のインターフェイスを選択し、VLAN にバインドします。
4. [**IP** バインディング] セクションで、1 つ以上の IP アドレスを選択し、VLAN にバインドします。
5. [**OK**] をクリックし、[完了] をクリックします。

共有 VLAN

共有 VLAN 設定では、各パーティションに MAC アドレスがあり、共有 VLAN で受信されるトラフィックは MAC アドレスで分類されます。レイヤ 3 VLAN はサブネットトラフィックを制限できるため、レイヤ 3 VLAN だけを推奨します。パーティション MAC アドレスは、共有 VLAN 配置にのみ適用され、重要です。

> 注
>
> NetScaler バージョン 12.1 ビルド 51.16 以降、パーティション化されたアプライアンスの共有 VLAN は動的ルーティングプロトコルをサポートします。

次の図は、VLAN（VLAN 10）が 2 つのパーティション間でどのように共有されるかを示しています。



共有 VLAN 設定を展開するには、次の手順を実行します。

1. 共有オプション「enabled」で VLAN を作成するか、既存の VLAN で共有オプションを有効にします。デフォルトでは、このオプションは「無効」です。
2. パーティションインターフェイスを共有 VLAN にバインドします。
3. パーティションを作成します。各パーティションには独自のパーティション MAC アドレスがあります。
4. パーティションを共有 VLAN にバインドします。

CLI を使用した共有 **VLAN** の設定

コマンドプロンプトで、次のいずれかのコマンドを入力して、VLAN を追加するか、既存の VLAN の共有パラメータを設定します。

```
add vlan <id> \[-sharing \(ENABLED | DISABLED)]

set vlan <id> \[-sharing \(ENABLED | DISABLED)]

add vlan 100  -sharing ENABLED

set vlan 100  -sharing ENABLED
```

CLI を使用してパーティションを共有 **VLAN** にバインドします

コマンドプロンプトで入力します。


```
bind partition <partition-id> -vlan <id>

bind partition P1 -vlan 100

add ns partition P1 -maxBandwidth 200 -maxconn 50 -maxmemlimit 90
   -partitionMAC<mac_addr

Done
```

CLI を使用してパーティションの **MAC** アドレスを構成する

```
set ns partition <partition name> [-partitionMAC<mac_addr>]

set ns partition P1 -partitionMAC 22:33:44:55:66:77
```

CLI を使用してパーティションを共有 **VLAN** にバインドします

```
bind partition <partition-id> -vlan <id>

bind partition <partition-id> -vlan <id>

bind partition P1 -vlan 100

bind partition P2 -vlan 100

bind partition P3 -vlan 100

bind partition P4 -vlan 100
```

NetScaler GUI を使用して共有 **VLAN** を構成する

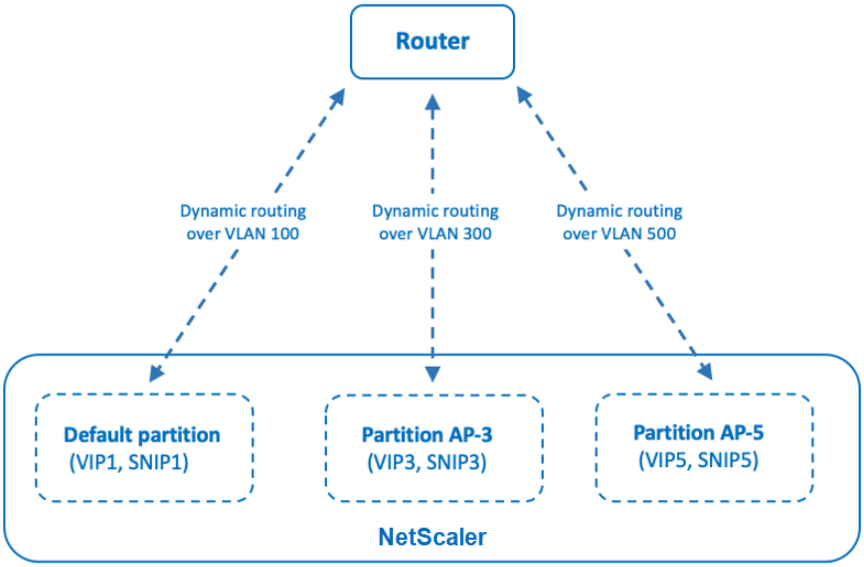
1. [構成] **>** [システム] **>** [ネットワーク] **>** [**VLAN**] に移動し、**VLAN** プロファイルを選択し、[編集] をクリックしてパーティション共有パラメータを設定します。
2. [**Create VLAN**] ページで [**Partitions Sharing**] チェックボックスをオンにします。
3. [**OK**]、[完了] の順にクリックします。

管理パーティション間の共有 **VLAN** を介したダイナミックルーティング

NetScaler アプライアンスの管理パーティションは、複数のテナントをホストする方法を提供します。

NetScaler バージョン 12.1 ビルド 51.16 以降、パーティション化されたアプライアンスの共有 VLAN は動的ルーティングプロトコルをサポートします。ルーティングは、管理パーティションに関連付けられた専用 VLAN または共有 VLAN で設定できます。

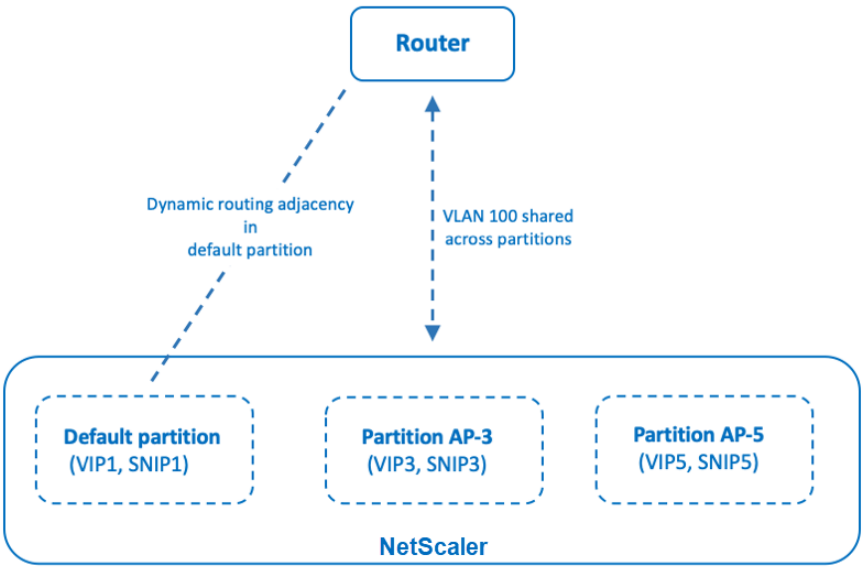
管理パーティションの専用 **VLAN**。専用 VLAN では、テナントのデータパスは 1 つ以上の VLAN を使用して識別されます。その結果、テナントの構成とデータパスの分離が厳密になります。VIP アドレスの健全性をアドバタイズするために、ダイナミックルーティングが各パーティションで有効になり、パーティションごとにルーティング隣接関係が確立されます。



Dynamic routing over a dedicated VLAN per partition

管理パーティション間の共有 **VLAN**。共有 VLAN では、デフォルト以外のパーティションに設定された VIP アドレスは、デフォルトパーティションで形成される単一の隣接またはピアリングを介してアドバタイズできます。デフォルト以外のパーティションの SNIP アドレスは、そのデフォルト以外のパーティション内のすべての VIP アドレス（**advertiseOnDefaultPartition** オプションを使用して設定）のネクストホップとして使用されます。設定された SNIP アドレスは、ルーティングアドバタイズメントでネクストホップ IP アドレスとしてマークされます。

NetScaler アプライアンスの管理パーティションの設定例を考えてみましょう。VLAN 100 はデフォルトパーティションとデフォルト以外のパーティション（AP-3 と AP-5）で共有されます。SNIP アドレス SNIP1 はデフォルトパーティションに追加され、SNIP3 は AP-3 に追加され、SNIP5 は AP-5 に追加されます。SNIP1、SNIP3、および SNIP5 は vlan-100 を介して到達可能です。VIP アドレス VIP1 はデフォルトパーティションに追加され、VIP3 は AP-3 に追加され、VIP5 は AP-5 に追加されます。VIP3 および VIP5 は、デフォルトパーティションで形成される単一の隣接またはピアリングを介してアドバタイズされます。



Dynamic routing over a shared VLAN across partitions

はじめに

デフォルト以外の管理パーティションの共有 VLAN 上でダイナミックルーティングを設定する前に、次のことを確認してください。

- ダイナミックルーティングは、デフォルトパーティションの共有 **VLAN** で設定されます。デフォルトパーティションの共有 VLAN でダイナミックルーティングを設定する手順は、次のとおりです。

  1. 共有 VLAN でダイナミックルーティングを有効にします。
  2. 動的ルーティングを有効にした状態で SNIP IP アドレスを追加します。この SNIP IP アドレスは、アップストリームとの動的ルーティングに使用されます。
  3. SNIP IP サブネットを共有 VLAN にバインドします。

- **1** つ以上の動的ルーティングプロトコルがデフォルトのパーティションに設定されています。詳細については、ダイナミックルーティングプロトコルの設定を参照してください。

構成の手順

デフォルト以外の管理パーティション内の共有 VLAN を介したダイナミックルーティングの設定は、次の手順で構成されます。

1. デフォルト以外のパーティションに **SNIP IP** アドレスを追加します。この SNIP IP アドレスは、デフォルトパーティションの動的ルーティングに使用されている SNIP IP アドレスと同じサブネットに存在する必要があります。

2. ダイナミックルーティングを使用して、デフォルト以外のパーティションで **VIP** アドレスをアドバタイズするための次のパラメータを設定または有効にします。

   - ホストルートゲートウェイ (HostRtgW)。このパラメータを、前の手順で追加した SNIP アドレスに設定します。
   - デフォルトパーティションでアドバタイズ（AdvertiseOnDefaultPartition)。このパラメーターを有効にします。

設定例

NetScaler アプライアンスの管理パーティション設定の例を考えてみましょう。このアプライアンスには、デフォルト以外の管理パーティション AP-3 が設定されています。共有 VLAN VLAN100 は AP-3 にバインドされています。次の設定例では、AP-3 で VLAN100 を介したダイナミックルーティングを設定します。

手順	設定例
デフォルトの管理パーティション	-
共有 VLAN100 で動的ルーティングを有効にします。	`set vlan 100 -dynamicRouting enabled`
動的ルーティングを有効にした状態で SNIP IP アドレス 192.0.2.10 を追加します。この SNIP IP アドレスは、アップストリームとの動的ルーティングに使用されます。	`add ns ip 192.0.2.10 255.255.255.0 -type SNIP -dynamicRouting enabled`
192.0.2.10 のサブネットを共有 VLAN 100 にバインドします。	`bind vlan 100 -IPAddress 192.0.2.10 255.255.255.0`
デフォルト以外の管理パーティション **AP-3**	-
SNIP IP アドレス 192.0.2.30 を追加します。この SNIP IP アドレスは、デフォルトパーティションの SNIP IP アドレス 192.0.2.10 と同じサブネットにあります。	`add ns ip 192.0.2.30 255.255.255.0 -type SNIP`
ダイナミックルーティングを使用して VIP アドレス 203.0.113.300 をアドバタイズする場合は、`advertiseOnDefaultPartition`パラメータを有効にし、`hostRtGw`パラメータを 192.0.2.30 に設定します。	`set ns ip 203.0.113.300 255.255.255.255 -hostRoute enabled -advertiseOnDefaultPartition enabled -hostRtGw 192.0.2.30`

管理パーティション全体の共有 **VLAN** を介した **IPv6** のダイナミックルーティング

IPv6 アドレスが管理パーティション内の共有 VLAN を介して動的にルーティングするには、`enable ns feature IPv6PT`コマンドおよび`set L3Param -ipv6DynamicRouting ENABLED`コマンドを

有効にする必要があります。次の設定例は、共有 VLAN 経由の IPv6 のダイナミックルーティングの設定に役立ちます。

設定例

次の設定例では、AP-3 で VLAN 100 を介したダイナミックルーティングを設定します。

手順	設定例
デフォルトの管理パーティション	-
共有 VLAN100 で動的ルーティングを有効にします。	`set vlan 100 -dynamicRouting enabled`
ダイナミックルーティングを有効にした状態で SNIP IP アドレス 2001: b: c: d: 1/64 を追加します。SNIP IP アドレスは、アップストリームとのダイナミックルーティングに使用されます。	`add ns ip6 2001:b:c:d::1/64 -type SNIP -dynamicRouting enabled`
2001: b: c: d: 1/64 のサブネットを共有 VLAN 100 にバインドします。	`bind vlan 100 -IPAddress 2001:b:c:d::1/64`
デフォルト以外の管理パーティション **AP-3**	-
SNIP IP アドレス 2001: b: c: d: 2/64 を追加します。この SNIP アドレスは、デフォルトパーティションの SNIP アドレス 2001: b: c: d: 2/64 と同じサブネット内にあります。	`add ns ip6 2001:b:c:d::2/64 -type SNIP`
ダイナミックルーティングを使用して VIP アドレス 2002: 1/128 をアドバタイズする場合は、`advertiseOnDefaultPartition` パラメータを有効にし、`ip6hostRtGw`パラメータを 2001: b: c: d: 2 に設定します。	`set ns ip6 2002::1/128 -hostRoute enabled -advertiseOnDefaultPartition enabled -ip6hostRtGw 2001:b:c:d::2`

管理パーティションに存在する VIP は、デフォルトパーティションの VTYSH でカーネルルートとして認識される必要があります。

```
> switch partition default
Done

>vtysh
ns#

ns# sh ipv6 route kernel

IPv6 routing table
```



```
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
   IA - OSPF inter area, E1 - OSPF external type 1,
   E2 - OSPF external type 2, I - IS-IS, B - BGP
Timers: Uptime

K      2002::1/128 via 2001:b:c:d::2, vlan0, 01:24:15
                                  >> on Default Partition, VIP : 2002::1
    present in AP known via SNIP6 : 2001:b:c:d::2 is present in AP as a
    Kernel Route
```

デフォルトパーティションで OSPFv3/bgp+ の下の「カーネルの再配布」オプションを使用して、アップストリームにアドバタイズできます。

```
ns# sh run router ipv6 ospf
!
router ipv6 ospf 1
redistribute kernel
!
```

NetScaler SDX アプライアンス上の管理パーティション付きの共有 VLAN

SDX アプライアンスでは、共有 VLAN で管理パーティションを使用する前に、管理サービスのユーザーインターフェイスを使用して PMAC アドレスを生成および設定する必要があります。管理サービスでは、次の方法によってパーティション MAC アドレスを生成できます。

- ベース MAC アドレスの使用
- カスタム MAC アドレスの指定
- MAC アドレスをランダムに生成する

> メモ
>
> - ランダムに生成される MAC アドレスは、高可用性以外の他の展開に使用されます。
> - パーティションの MAC アドレスを生成したら、管理パーティションを構成する前に NetScaler インスタンスを再起動する必要があります。SDX アプライアンスからパーティション MAC アドレスを生成する方法の詳細については、「SDX アプライアンスの NetScaler インスタンスで管理パーティションを構成するためのパーティション MAC アドレスの生成」を参照してください。

管理パーティションの VXLAN サポート

August 15, 2023

パーティション化された NetScaler アプライアンスでは、VLAN の構成と同様に、デフォルトパーティションに VXLAN を構成できます。VXLAN を設定したら、それを管理パーティションにバインドできます。または、VXLAN

がパーティションにバインドされている VLAN を拡張している場合、アプライアンスは VXLAN を同じブロードキャストドメインのパーティションにバインドします。これは、VXLAN をパーティションからアンバインドする VLAN のバインド解除に適用できます。

NetScaler アプライアンスで VXLAN がどのように機能するかについて詳しくは、「VXLAN」を参照してください。

また、パーティション化された NetScaler アプライアンスで VLAN がどのように機能するかについて詳しくは、「管理パーティショニング」を参照してください。

VXLAN を設定する前に覚えておくべきポイント

パーティション化された NetScaler アプライアンスで VXLAN を構成する前に、次の点に注意してください。

- VXLAN で VLAN を拡張する場合は、VLAN がパーティションにバインドされていることを確認してください。
- パーティション管理者だけが、管理パーティションの VXAN の IP と動的ルーティングを設定する必要があります。

共有 VXLAN はパーティション化されたアプライアンスではサポートされないため、VXLAN を共有 VLAN にタグ付けすることはできません。また、VXLAN にタグ付けされている VLAN を共有にすることもできません。

サポート可能な **VXLAN** 構成

サポート可能な VXLAN 構成は次のとおりです。

同じブロードキャストドメイン内の **VXLAN** を介した **VLAN** の拡張

次の CLI 手順は、VXLAN 上で VLAN を拡張したり、同じブロードキャストドメイン内でその逆を行ったりするのに役立ちます。

1. デフォルトパーティションに VLAN を追加する

```
1  add vlan <id>
```

2. 同じブロードキャストドメイン内の VXLAN 上で VLAN を拡張します。

```
1 add vxlan <vxlan id>  - vlan <id>
```

3. すべての BUM（ブロードキャスト不明マルチキャスト）トラフィックを伝送するようにピア `vtep` を構成します。

> 注
>
> `vtep` アドレスはマルチキャストアドレスにすることができます。


```
add bridgetable -mac <mac_addr> -vxlan <positive_integer> -vtep <
    ip_addr> [-vni <positive_integer>][-deviceVlan <
    positive_integer>]
```

4. IP アドレスを VXLAN にバインドします。

```
bind vxlan <id> \[-srcIP <ip\_addr>]\[-IPAddress <ip\_addr|ipv6\
    _addr|\*> \[<netmask>]]
```

5. VLAN を管理パーティションにバインドします。

```
bind partition <partition-id> -vxlan <id>

add vlan 3000

add vxlan 3000  - vlan 10

add bridgetable  - mac 00:00:00:00:00:00  - vxlan 3000 -vtep
    10.102.58.8  - vni 11

bind vxlan 3000  -  srcIP 10.102.101.15

bind partition p1  - vlan 10
```

管理パーティションの **SNMP** サポート

August 15, 2023

パーティション化された NetScaler アプライアンスは、SNMP インフラストラクチャを使用してパーティションのレート制限やパーティションリソース使用率の詳細の監視を行います。

管理パーティションのレート制限用の **SNMP** トラップ

パーティション化された NetScaler アプライアンスでは、PARTITION-RATE-LIMIT アラームによって 9 つの SNMP トラップが生成され、パーティションリソース（帯域幅、接続、メモリなど）が制限に達したか、正常に戻ったことを通知できます。

次の 9 つの SNMP トラップは次の場合に生成されます。

- パーティション接続のしきい値に達しました。パーティションのアクティブな接続数が上限パーセンテージを超えています。
- パーティション接続しきい値: 標準。アクティブな接続の数は、通常のしきい値のパーセンテージ以下です。
- パーティション **BW** のしきい値に達しました。パーティションの帯域幅使用率が高いしきい値に達しました。

- パーティション **MEM** のしきい値に達しました。パーティションの現在のメモリ使用量が、上限しきい値のパーセンテージを超えています。
- パーティションメモリしきい値 (標準)。パーティションの現在のメモリ使用量は、通常のしきい値のパーセンテージ以下になります。
- パーティションの **MEM** 制限を超えています。パーティションの現在のメモリ使用量がメモリ制限パーセンテージを超えています。
- パーティションの接続制限を超えました。パーティションのアクティブな接続の数が設定された制限を超えているため、新しい接続がドロップされます。
- パーティション接続/制限/標準。パーティションのアクティブな接続の数が設定された制限を下回り、パーティションが新しい接続を受け入れることができるようになりました。
- パーティション **BW** の制限を超えました。パーティションの現在の帯域幅使用量が、設定されている制限を超えています。

SNMP トラップのしきい値は設定できず、次のとおりです。

- 高いしきい値 = 80%（すべてのパーティションレート制限トラップに適用）
- 下限しきい値 = 60%（すべてのパーティションレート制限トラップに適用）
- メモリ制限 = 95% (パーティションメモリトラップにのみ適用)

パーティションレート制限アラームの設定

PARTITION-RATE-LIMIT アラームを特定のパーティションに設定し、SNMP トラップメッセージの生成を有効にします。

1. パーティションレート制限アラームを有効にする
2. パーティションレート制限アラームの設定
3. SNMP トラップ送信先の設定

CLI を使用してパーティションレート制限アラームを有効にするには

コマンドプロンプトで、次のコマンドを入力します。

```
enable snmp alarm PARTITION-RATE-LIMIT

show snmp alarm PARTITION-RATE-LIMIT
```

CLI を使用して **PARTITION-RATE-LIMIT** アラームを設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
set snmp alarm PARTITION-RATE-LIMIT \[-state \( ENABLED | DISABLED )]
    \[-severity <severity>] \[-logging \( ENABLED | DISABLED )]
```


CLI を使用して **SNMP** トラップの宛先を構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add snmp trap <trapClass> <trapDestination> \[-version <version>] \[-td
    <positive\_integer>] \[-destPort <port>] \[-communityName <string>]
    \[-srcIP <ip\_addr|ipv6\_addr>] \[-severity <severity>] \[-
   allPartitions \( ENABLED | DISABLED )]
```

GUI を使用して **PARTITION-Rate-Limit** アラームを設定するには

[システム] **> [SNMP] > [**アラーム] に移動し、[パーティションレート制限アラーム] を選択し、アラームパラメータを設定します。

GUI を使用して **SNMP** トラップの宛先を設定するには

[システム] **> [SNMP] > [**トラップ] に移動し、宛先デバイスの IP アドレスを指定します。

パーティションリソース使用率の **SNMP** 監視

SNMP を使用すると、NetScaler アプライアンス上のパーティションのリソース（帯域幅、接続、メモリなど）の使用率の詳細をリアルタイムで監視できます。そのためには、SNMP マネージャから SNMP リクエスト (SNMP GET、SNMP GET BULK、SNMP GETNEXT、SNMP WALK など) を送信します。

> 注
>
> パーティションリソースを監視するには、デフォルトパーティションに SNMP コミュニティを設定する必要があります。この場合、*PartitionTable* はデフォルトのパーティションに保持され、SNMP 通信はアプライアンスの NSIP アドレスを介して行われます。

NetScaler 管理者がアプライアンスのパーティション P1 の帯域幅使用量を知りたい場合を考えてみましょう。SNMP マネージャーは、対応する OID (PartitionCurrentBandwidth) の SNMP GET リクエストをアプライアンスの NSIP アドレスに送信することによってこの情報を取得します。デフォルトパーティションの SNMP エージェントは、P1 の現在の帯域幅使用状況を取得し、NSIP アドレスを介して SNMP マネージャーに送信します。

次の表は、パーティションテーブルに含まれる *SNMP* カウンタとその説明を示しています。

SNMP パラメータ	スナップフッド	説明
パーティション名	1.3.6.1.4.1.5951.4.1.1.88.1.1	パーティション名
パーティションの現在の帯域幅	1.3.6.1.4.1.5951.4.1.1.88.1.2	パーティションの現在の帯域幅使用量。

SNMP パラメータ	スナップフッド	説明
パーティションの現在の接続	1.3.6.1.4.1.5951.4.1.1.88.1.3	パーティションの現在のアクティブな接続数。
パーティションメモリ使用量 PCNT	1.3.6.1.4.1.5951.4.1.1.88.1.4	パーティションの現在のメモリ使用量 (パーセンテージ)。

管理パーティションの監査ログサポート

August 15, 2023

パーティション化された NetScaler アプライアンスでは、データセキュリティを強化するために、高度なポリシーを使用して管理パーティションに監査ログを構成できます。たとえば、特定のパーティションのログ (状態とステータス情報) を表示したい場合があります。パーティション内の権限レベルに基づいて、複数のユーザーがさまざまな機能セットにアクセスします。

確認事項

1. パーティションから生成された監査ログは、単一のログファイル (/var/log/ns.log) として保存されます。
2. 監査ログサーバー (syslog または ns log) のサブネットアドレスを、監査ログメッセージを送信するパーティションの送信元 IP アドレスとして設定します。
3. デフォルトパーティションは、デフォルトで NSIP を監査ログメッセージの送信元 IP アドレスとして使用します。
4. 監査ログメッセージを表示するには、「show audit messages」コマンドを使用します。

監査ログの構成の詳細については、「監査ログ用の NetScaler アプライアンスの構成」を参照してください。

パーティション化された **NetScaler** アプライアンスでの監査ログの設定

管理パーティションに監査ログを設定するには、次のタスクを実行します。

1. パーティションのサブネット IP アドレスを設定します。管理パーティションの IPv4 SNIP アドレス。
2. 監査ログ (syslog および ns ログ) アクションを設定します。監査アクションは、ログに記録するメッセージとそのメッセージを外部ログサーバに記録する方法を指定する情報の集まりです。
3. 監査ログ (syslog と ns ログ) ポリシーを設定します。監査ログポリシーは、ソースパーティションの syslog または ns ログサーバへのログメッセージを定義します。
4. 監査ログポリシーを SysGlobal エンティティと NSGlobal エンティティにバインドします。監査ログポリシーをシステムグローバルエンティティにバインドします。

5. 監査ログ統計を確認する。監査ログの統計情報を表示し、構成を評価します。

CLI を使用して次の設定を行います

1. パーティションのサブネット IP アドレスを作成

```
add ns ip <ip address> <subnet mask>
```

2. Syslog アクションを作成する

```
add audit syslogAction <name> <serverIP> [-serverPort <port>]
 -logLevel <logLevel> [-dateFormat (MMDDYYYY | DDMMYYYY )] [-
transport ( TCP | UDP )]
```

3. ns ログアクションの作成

```
add audit nslogAction <name> <serverIP> [-serverPort <port>] -
logLevel <logLevel> [-dateFormat (MMDDYYYY | DDMMYYYY )]
```

4. Syslog 監査ログポリシーの作成

```
add audit syslogpolicy syslog-pol1 true audit-action1
```

5. NS ログ監査ログポリシーの作成

```
add audit nslogpolicy nslog-pol1 true audit-action1
```

6. 監査ログポリシーを SyslogGlobal エンティティにバインドする

```
bind audit sysloggglobal -policyName <name> -priority <priority_integer
> -globalBindType SYSTEM_GLOBAL
```

7. 監査ログポリシーを NSLogGlobal エンティティにバインドする

```
bind audit nsloggglobal -policyName <name> -priority <priority_integer
> -globalBindType SYSTEM_GLOBAL
```

8. 監査ログの統計情報を表示する

```
stat audit -detail
```

例

```
1 add ns ip 10.102.1.1 255.255.255.0
2 add audit syslogAction syslog_action1 10.102.1.2 -logLevel
     INFORMATIONAL -dateFormat MMDDYYYY -transport UDP
3 add audit syslogpolicy syslog-pol1 true syslog_action1
4 bind audit sysloggglobal -policyName syslog-pol1 -priority 1 -
     globalBindType SYSTEM_GLOBAL
```


ログの保存

SYSLOG または NSLOG サーバがすべてのパーティションからログ情報を収集すると、ログメッセージとして ns.log ファイルに保存されます。ログメッセージには次の情報が含まれます。

- パーティション名。
- IP アドレス。
- タイムスタンプ。
- メッセージの種類
- 定義済みのログレベル (重大、エラー、通知、警告、情報、デバッグ、アラート、緊急)
- メッセージ情報。

共有 **VLAN** 構成用に設定された **PMAC** アドレスを表示する

August 15, 2023

共有 VLAN 設定でパーティション設定を使用するには、パーティション MAC（PMAC）アドレスと呼ばれる仮想 MAC アドレスが必要です。パーティションは、共有 VLAN での通信に PMAC アドレスを使用します。パーティションごとに一意の PMAC アドレスが設定され、そのパーティションにバインドされているすべての共有 VLAN で使用されます。非 SDX プラットフォーム（VPX または MPX）プラットフォームの場合、PMAC アドレスはユーザーが指定することも、NetScaler アプライアンスによって内部生成されたものでもかまいません。パーティションに PMAC アドレスが指定されていない場合、パーティションが最初の共有 VLAN にバインドされたときに内部的に生成されます。SDX プラットフォームの場合は、最初に SVM ツールから PMAC アドレスを設定してから、パーティションに割り当てる必要があります。

設定された PMAC のリストを表示するには、**show ns partitionMac** コマンドを使用します。このコマンドを使用すると、NetScaler CLI または GUI のいずれかで構成された PMAC を確認できます。このコマンドは、すべての PMAC アドレスと対応するパーティション（割り当てられている場合）を表示します。非 SDX プラットフォームの場合、このコマンドは、すべての PMAC アドレスとそれに対応するパーティションを表示します。これは、PMAC アドレスは必要に応じてパーティションに割り当てられるためです（パーティションが共有 VLAN をバインドしている場合）。ただし、SDX プラットフォームの場合、リストに未割り当ての PMAC が含まれている可能性があります。

SDX プラットフォーム用の PMAC を生成する方法については、「パーティション MAC アドレスの生成」トピックを参照してください。

NetScaler CLI を使用して **PMAC** を表示する

コマンドプロンプトで、次のコマンドを入力します。

```
show ns partitionMAC
```



```
Partition MAC Partition Name

1) f2:0c:64:da:f6:d7

2) b4:0c:43:da:f6:d2

3) a6:e7:b2:6c:48:e0

Done
```

NetScaler GUI を使用して **PMAC** アドレスを表示する

1. NetScaler アプライアンスにサインインし、[構成**] > [**システム**] > [**パーティション **MAC**] に移動します。
2. [パーティション MAC] ページには、PMAC とそのパーティションのリストが表示されます。

AppExpert

August 15, 2023

次のトピックでは、NetScaler ADC アプライアンスの AppExpert およびその他の機能に関する概念的なリファレンスと構成手順について説明します。

> 注
>
> ポリシー拡張機能の詳細については、「ポリシー拡張」を参照してください。

- アクション分析: 事前定義された条件に基づいて実行時の統計を収集します。ポリシーとともに使用すると、自動的にリアルタイムのトラフィック最適化のためのインフラストラクチャも提供されます。
- AppExpert アプリケーションとテンプレート: アプリケーション、アプリケーションテンプレート、NetScaler Gateway アプリケーション、およびエンティティテンプレートを使用して、Citrix® NetScaler® アプライアンスの構成手順を簡素化します。
- AppQoE: アプリケーションレベルのエクスペリエンスの品質（AppQoE）は、NetScaler ADC アプライアンスの既存のポリシーベースのセキュリティ機能を新しいキューイングメカニズムである均等化キューイングを利用する単一の統合機能に統合します。
- エンティティテンプレート: エンティティテンプレートを使用して、ポリシーや仮想サーバーなどの個々の NetScaler ADC エンティティを設定および構成する方法について説明します。エンティティテンプレートは、オブジェクトの仕様とデフォルトセットを提供します。
- HTTP コールアウト: ポリシー評価中に特定の基準が満たされたときに、NetScaler ADC アプライアンスが生成して外部アプリケーションに送信する HTTP リクエスト。

- パターンセット: 詳細ポリシーの評価中に文字列の一致を許可します。
- ポリシーと式: NetScaler ADC アプライアンスが実行する必要のある操作を決定するルール。
- レート制限: NetScaler ADC アプライアンス上の特定のネットワークエンティティまたは仮想エンティティの最大負荷を定義します。
- Responder: リクエストの送信元、リクエストの送信元、およびその他の基準に基づいて、セキュリティとシステム管理に影響します。
- 書き換え: NetScaler ADC アプライアンスによって処理される要求または応答の情報を書き換えます。
- 文字列マップ: デフォルトのポリシー構文を使用するすべての NetScaler ADC 機能でパターンマッチングを実行します。

アクション分析

August 15, 2023

Web サイトやアプリケーションのパフォーマンスは、最も頻繁に要求されるコンテンツの配信をどのように最適化するかにより決まります。キャッシュや圧縮などの方法は、クライアントへのサービス配信の高速化に役立ちますが、最も頻繁に要求されるリソースを特定し、それらのリソースをキャッシュまたは圧縮できるようにする必要があります。Web サイトやアプリケーショントラフィックに関するリアルタイム統計を集計すれば、最も頻繁に使用されるリソースを特定できます。リソースごとのアクセス頻度や消費帯域幅などの統計によって、サーバーパフォーマンスとネットワーク使用率を改善するために、それらのリソースをキャッシュまたは圧縮する必要があるかどうかを判断できます。応答時間やアプリケーションへの同時接続数などの統計は、サーバー側のリソースを強化する必要があるかどうかを判断するのに役立ちます。

Web サイトやアプリケーションが頻繁に更新されない場合、統計データを収集する製品を使用して、その統計を手動で分析し、コンテンツの配信を最適化できます。ただし、手動による最適化を行いたくない場合や、Web サイトやアプリケーションが本質的に動的である場合は、統計データを収集するだけでなく、統計に基づいてリソースの配信を自動的に最適化できるインフラストラクチャが必要です。NetScaler アプライアンスでは、この機能はアクション分析機能によって提供されます。この機能は単一の NetScaler アプライアンス上で実行され、定義した条件に従ってリアルタイム統計を収集します。NetScaler ポリシーと共に使用すると、この機能によって自動的なリアルタイムトラフィックの最適化に必要なインフラストラクチャも提供されます。

アクション分析機能を設定する場合、セレクタと呼ばれるエンティティで高度なポリシー式を設定して、URL や HTTP メソッドなどの統計データを収集する要求属性を指定します。次に、識別子を構成して、サンプリング間隔やサンプル数などの設定を構成します。また、セレクタと識別子のペアで指定されたとおりにアプライアンスがトラフィックを評価できるようにするポリシーも設定します。最後に、ポリシーをバインドポイントにバインドして、統計情報の収集を開始します。

アプライアンスには、組み込みセレクタ、識別子、およびレスポンダーポリシーのセットも用意されています。これらのポリシーを使用して、この機能の使用を開始できます。

アプライアンスは、次の統計情報を集計します。

- リクエストの数。
- 要求によって消費される帯域幅。
- 応答時間。
- 同時接続の数。

選択した属性でレコードの実行時にソートを実行するようにフィーチャを構成できます。統計データは、コマンドラインインターフェイスまたは構成ユーティリティの Stream Sessions ツールのいずれかを使用して表示できます。

セレクタを構成する

December 8, 2023

セレクタは、要求を識別するためのフィルタです。これは、クライアント IP アドレスや要求内の URL などの要求属性を識別する最大 5 つの個別の高度なポリシー式で構成されます。各式は複合ではない高度なポリシー式であり、他の式との AND 関係にあると見なされます。セレクター式の例をいくつか示します:

- `HTTP.REQ.URL`
- `CLIENT.IP.SRC`
- `HTTP.RES.BODY(1000).AFTER_STR("<string>").BEFORE_STR("<string>")"`
- `CLIENT.IP.SRC.SUBNET(24)`

セレクターは、レート制限とアクション分析の設定で使用されます。セレクターはレート制限設定ではオプションですが、アクション分析設定では必須です。

パラメータを指定する順序は重要です。たとえば、1 つのセレクターで IP アドレスとドメインを（この順序で）構成し、別のセレクターでドメインと IP アドレスを（逆の順序で）指定すると、NetScaler はこれらの値を一意と見なします。これにより、同じトランザクションが 2 回カウントされる可能性があります。また、複数のポリシーが同じセレクタを呼び出す場合、NetScaler は同じトランザクションを複数回カウントできます。

セレクタの式を変更すると、それを呼び出すポリシーが新しいポリシーラベルまたはバインドポイントにバインドされていると、エラーが発生することがあります。たとえば、myLimitSelector1 という名前のセレクタを作成し、それを myLimitId1 から呼び出して、dnsRateLimit1 という名前の DNS ポリシーから識別子を呼び出すとします。myLimitSelector1 の式を変更すると、dnsRateLimit1 を新しいバインドポイントにバインドするときにエラーが発生することがあります。回避策は、これらの式を呼び出すポリシーを作成する前に、これらの式を変更することです。

NetScaler アプライアンスには、最も一般的なユースケースに対応するセレクターが組み込まれています 。

また、選択したリクエスト属性を識別する式を使用してセレクタを設定することもできます。たとえば、特定のヘッダーで到着するリクエストのレコードを作成するとします。ヘッダーを評価するために、使用するセレクター

にHTTP.REQ.HEADER("<header_name>")を追加できます。

コマンドラインインターフェイスを使用してセレクタを設定するには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、セレクタを構成し、構成を確認します。

- add stream selector <name> <rule> ...
- show stream selector

例

```
> add stream selector myselector HTTP.REQ.URL CLIENT.IP.SRC
 Done
> show stream selector myselector
    Name: myselector
    Expressions:
        1) HTTP.REQ.URL
        2) CLIENT.IP.SRC
 Done
>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセレクタを変更または削除するには、次の手順を実行します。

- セレクターを変更するには、set stream selector コマンド、セレクターの名前、および式を含む規則パラメーターを入力します。保持する既存の式と、追加する新しい式を入力します。
- セレクターを削除するには、rm stream selector コマンドとセレクターの名前を入力します。

GUI を使用してセレクタを設定するには、次の手順を実行します。

1. [**AppExpert**] > [アクション分析**] > [**セレクター**]** に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います:
    - セレクタを作成するには、[追加] をクリックします。
    - セレクタを変更するには、セレクタを選択し、[編集] をクリックします。
3. 「セレクタの作成」または「セレクタの設定」ページで、次のパラメータを設定します。
    - Name: セレクターの名前を追加するには、名前フィールドに名前を入力します。名前は ASCII、英数字、またはアンダースコア文字で始まる必要があります。名前には、ASCII 英数字、アンダースコア、ハッシュ、ピリオド、スペース、コロン、アット文字、等号、ハイフンのみを含める必要があります。
    - Expressions. セレクタ設定に式を追加するには、[挿入] をクリックします。セレクター設定から式を削除するには、[式] ボックスで式を選択し、[削除] をクリックします。注:[式] ボックスに、有効なパラメータを入力します。たとえば、HTTP と入力します。次に、このパラメータの後にピリオドを入力します。ドロップダウンメニューが表示されます。このメニューの内容には、入力した最初のキーワードの後に続くキーワードが表示されます。このエクスプレフィックス内の次のキーワードを選択するには、ドロップダウンメニューで選択したキーワードをダブルクリックします。[式] テキストボックスには、

式プレフィックスの 1 番目と 2 番目のキーワード (HTTP.REQ など) の両方が表示されます。完全なエクスプレッションが形成されるまで、エクスプレッションコンポーネントの追加を続けます。

4. [挿入] をクリックします。
5. 非複合式を 5 つまで追加します。
6. [作成]、[閉じる] の順にクリックします。

ストリーム **ID** を構成する

December 8, 2023

ストリーム識別子を設定して、特定のセレクターによって識別されるリクエストから統計データを収集するためのパラメーターを指定します。識別子は、使用するセレクター、統計収集間隔、サンプル数、およびレコードをソートするフィールドを指定します。

NetScaler アプライアンスには、一般的な用途向けに次の組み込みストリーム識別子が含まれています。組み込みの識別子はすべて、サンプル数を 1、間隔を 1 分として指定します。さらに、REQUESTS 属性でデータをソートします。異なるのは、さまざまな組み込みセレクターに関連付けられている点だけです。各組み込み識別子は、同じ名前の組み込みセレクターに関連付けられています (たとえば、組み込み識別子
top_URL
は組み込みセレクター top_URL に関連付けられます)。組み込みの識別子は次のとおりです:

- Top_URL
- Top_CLIENTS
- Top_URL_CLIENTS_LBVSERVER
- Top_URL_CLIENTS_CSVSERVER
- Top_MSSQL_QUERY_DB_LBVSERVER
- Top_MYSQL_QUERY_DB_LBVSERVER

組み込みセレクタの詳細については、「セレクタの設定」を参照してください。

注: セレクタの文字列結果（HTTP.REQ.URL など）を格納するための最大長は 60 文字です。文字列 (URL など) の長さが 1000 文字で、そのうちの 50 文字で文字列を一意に識別できる場合は、式を使用して必要な 50 文字だけを抽出します。

組み込み識別子の構成は変更できません。ただし、任意の構成で識別子を作成できます。

コマンドラインインターフェイスを使用してストリーム識別子を設定するには

コマンドプロンプトで次のコマンドを入力してストリーム識別子を構成し、構成を確認します。

- `add stream identifier <name> <selectorName> [-interval <positive_integer>] [-SampleCount <positive_integer>] [-sort <sort>]`
- `show stream identifier <name>`

例

```
> add stream identifier myidentifier Top_URL -interval 10 -sampleCount
    100
 Done
<!--NeedCopy-->
```

GUI を使用してストリーム識別子を設定するには

1. **AppExpert** > アクション分析 > ストリーム識別子に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います:
   - ストリーム識別子を作成するには、[追加] をクリックします。
   - ストリーム識別子を変更するには、識別子を選択し、**[**編集**]** をクリックします。
3. 「ストリーム識別子の設定」ページで、次のパラメータを設定します。
   - Name
   - [セレクタ]
   - 間隔
   - サンプル数
   - 並べ替え
4. [作成] をクリックし、[閉じる] をクリックします。

統計の表示

December 8, 2023

収集された統計情報は、コマンドラインインターフェイスでは表形式で、構成ユーティリティではグラフ形式で表示できます。

次の表に、収集された統計情報を示します。

統計	stat ストリーム識別子<identifier name> コマンドの出力内の列名	説明
要求数	要求	<interval> 直近の分間にレコードが作成されたリクエストの数。

統計	stat ストリーム識別子<identifier name> コマンドの出力内の列名	説明
消費帯域幅	バンド **W**	<interval> 直近の数分間に受信されたリクエストによって消費された合計帯域幅。要求の合計帯域幅は、要求とその応答によって消費される帯域幅です。値は、次の上位または下位の整数値に四捨五入されます。したがって、期待値と若干異なる可能性があります。たとえば、リクエストの合計帯域幅消費が 2.2 KB の場合。リクエストの 1 つのインスタンスが 2 KB を消費したとして表示されることがあります。2 つのインスタンスは 4 KB を消費したとして表示されますが、3 つのインスタンスは 7 KB を消費したと見なされる場合があります。
応答時間	**RSP** タイム	<interval> 直近の数分間に受信したすべてのリクエストの平均応答時間。
同時接続	コネチカット州	現在開いている同時接続の総数。

コマンドラインを使用してストリーム識別子について収集された統計データを表示するには

コマンドプロンプトで入力します:

```
stat stream identifier <name> [<pattern> ...] [-detail] [-fullValues]
 [-ntimes <positive_integer>] [-logFile <input_filename>] [-sortBy <
sortBy> [<sortOrder>]
```

例

例 1 では、BandW 列の出力を降順にソートしています。例 2 では、例 1 の [**Req**] 列の出力を昇順でソートします。

例 1


```
> stat stream identifier myidentifier -sortBy BandW Descending -
    fullValues
Stream Session statistics
                      Req                  BandW
User1                 508                 125924
User2                5020                  12692
User3                2025                   4316

                  RspTime                   Conn
User1                5694                      0
User2                 109                      0
User3                   3                      0
 Done
<!--NeedCopy-->
```

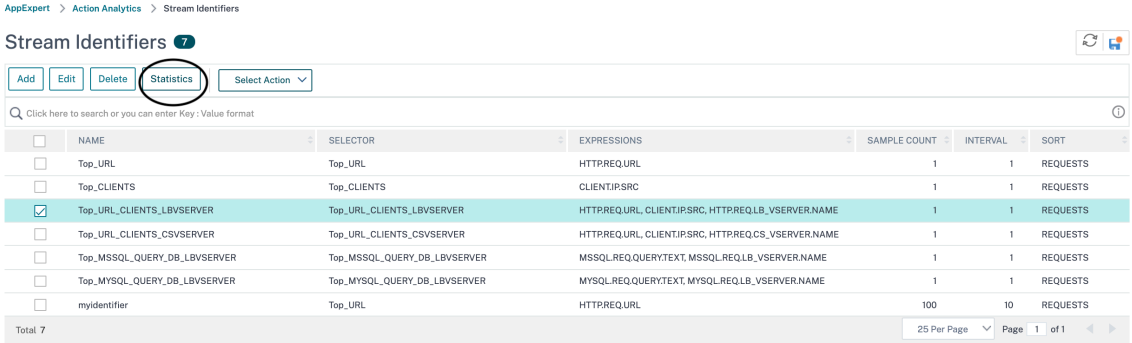
例 2

```
> stat stream identifier myidentifier -sortBy Req Ascending  -
    fullValues
Stream Session statistics
                      Req                  BandW
User1                 508                 125924
User3                2025                   4316
User2                5020                  12692

                  RspTime                   Conn
User1                5694                      0
User3                   3                      0
User2                 109                      0
 Done
<!--NeedCopy-->
```

GUI を使用してストリーム識別子について収集された統計データを表示するには

1. **AppExpert** > アクション分析 > ストリーム識別子に移動します。
2. セッションを表示するストリーム識別子を選択し、「統計」をクリックします。さまざまなセレクター式で収集された値に基づいて出力をグループ化する方法については、こちらをご覧ください。



属性値に基づくレコードのグループ化

August 15, 2023

特定の URL が全体およびクライアントごとにアクセスされた回数、クライアントごとの GET リクエストと POST リクエストの総数などの統計情報により、需要に合わせてリソースを拡張する必要があるか、配信用に最適化する必要があるかについての貴重な洞察が得られます。このような統計情報を取得するには、適切なセレクター式のセットを使用してから、stat stream identifier コマンドのパターンパラメーターを使用する必要があります。グループ化は、コマンドで指定されたパターンに基づいています。グループ化は、複数の式の値を同時に実行できます。

コマンドラインインターフェイスでは、任意のパターンを使用して出力をグループ化できます。構成ユーティリティでは、さまざまなセレクター式の値をドリルダウンするときに行う選択によってパターンが異なります。たとえば、式HTTP.REQ.URL、CLIENT.IP.SRC、HTTP.REQ.LB_VSERVER.NAMEがその順序にあるセレクタを考えてみましょう。統計ホームページには、これらの各式のアイコンが表示されます。CLIENT.IP.SRCのアイコンをクリックすると、出力はパターンに基づいてい ますか? で指定します。出力には、各クライアント IP アドレスの統計が表示されます。IP アドレスをクリックすると、* <IP address> ? と? <IP address> *のパターンに基づいて出力されます。ここで<IP address> は選択した IP アドレスです。結果の出力では、URL をクリックすると、使用されるパターンは<URL> <IP address> ?になります。

コマンドラインインターフェイスを使用してセレクター式の値のレコードをグループ化するには

コマンドプロンプトで次のコマンドを入力して、セレクター式に基づいてレコードをグループ化します。

```
stat stream identifier <name> [<pattern> ...]
```

以下の例では、別のパターンを使用して、このパターンが stat stream identifier コマンドの出力に与える影響を示しています。セレクター式は HTTP.REQ.URL と HTTP.REQ.HEADER (「ユーザーヘッダー」) の順序です。リクエストには、UserHeader という名前のカスタムヘッダーが含まれています。この例では、特定の統計値はグループ化の決定に応じて変化しますが、特定のフィールドの値の合計は同じままであることに注意してください。

例 1

次のコマンドでは、使用されるパターンは??。アプライアンスは、両方のセレクター式で収集された値に基づいて出力をグループ化します。行ヘッダーは、疑問符 (?) で区切られた式の値で構成されます。/mysite/mypage1.html というヘッダーのある行?Ed は、ユーザー Ed が /mysite/mypage1.html という URL に対して行ったリクエストの統計を表示します。

> 注:
>
> 次のコマンドを「?」ではなく「\?」で入力する必要があります。たとえば、セレクターが client.ip.src と client.tcp.srcport という式を使用する場合。セレクター用に収集された値に基づいて出力をグループ化する Stat コマンドは、「stat ストリーム識別子 myidentifier?」です。?-「全値」は以下の通りです。


```
> stat stream identifier myidentifier ? ? -fullValues
Stream Session statistics

                                          Req                  BandW
/mysite/mypage2.html?Grace                  1                   2553
/mysite/mypage1.html?Grace                  2                      4
/mysite/mypage1.html?Ed                     8                     16
/mysite/mypage2.html?Joe                    1                   2554
/mysite/mypage1.html?Joe                    5                     10
/mysite/?Joe                                1                      4

                                       RspTime                  Conn
/mysite/mypage2.html?Grace                   0                      0
/mysite/mypage1.html?Grace                   0                      0
/mysite/mypage1.html?Ed                      0                      0
/mysite/mypage2.html?Joe                     0                      0
/mysite/mypage1.html?Joe                     0                      0
/mysite/?Joe                                 6                      0
 Done
<!--NeedCopy-->
```

例 2

次のコマンドでは、使用されるパターンは *? です。アプライアンスは、2 番目の式 HTTP.REQ.HEADER (「UserHeader」) の累積値に基づいて出力をグループ化します。行には、Grace、Ed、Joe というユーザーによるすべてのリクエストの統計が表示されます。

> 注:
>
> 次のコマンドを「?」の代わりに「?」。

```
> stat stream identifier myidentifier * ?
Stream Session statistics
                   Req    BandW  RspTime     Conn
Grace                3     2557        0        0
Ed                   8       16        0        0
Joe                  7     2568        6        0
 Done
<!--NeedCopy-->
```

例 3

次のコマンドでは、使用されるパターンは?*、これはデフォルトのパターンです。出力は、最初のセレクター式で収集された値に基づいてグループ化されます。各行には、1 つの URL の統計情報が表示されます。

> 注:
>
> 次のコマンドを「?」の代わりに「?」。

```
> stat stream identifier myidentifier ? * -fullValues
Stream Session statistics
                                          Req                  BandW
```



```
/mysite/mypage2.html                          2                    5107
/mysite/mypage1.html                         15                      30
/mysite/                                      1                       4

                                         RspTime                    Conn
/mysite/mypage2.html                          0                       0
/mysite/mypage1.html                          0                       0
/mysite/                                      6                       0
 Done
<!--NeedCopy-->
```

例 4

次のコマンドでは、使用されるパターンは* *です。アプライアンスは、受信したすべてのリクエストについて、行のタイトルなしで 1 セットの集合統計を表示します。

```
> stat stream identifier myidentifier * *
Stream Session statistics
                  Req    BandW  RspTime     Conn
                   18     5141        6        0
 Done
<!--NeedCopy-->
```

例 5

次のコマンドでは、パターンは /mysite/mypage1.html * です。アプライアンスは、URL /mysite/mypage1.html に対して受信したすべてのリクエストについて、行タイトルなしの 1 セットの集合統計情報を表示します。

```
> stat stream identifier myidentifier /mysite/mypage1.html *
Stream Session statistics
                  Req    BandW  RspTime     Conn
                   15       30        0        0
 Done
<!--NeedCopy-->
```

ストリームセッションのクリア

August 15, 2023

ストリーム識別子に蓄積されたすべてのレコードをフラッシュできます。

コマンドラインインターフェイスを使用してストリームセッションをクリアするには

コマンドプロンプトで次のコマンドを入力してストリームセッションをクリアし、結果を確認します。

- ストリームセッションをクリア
- stat stream identifier

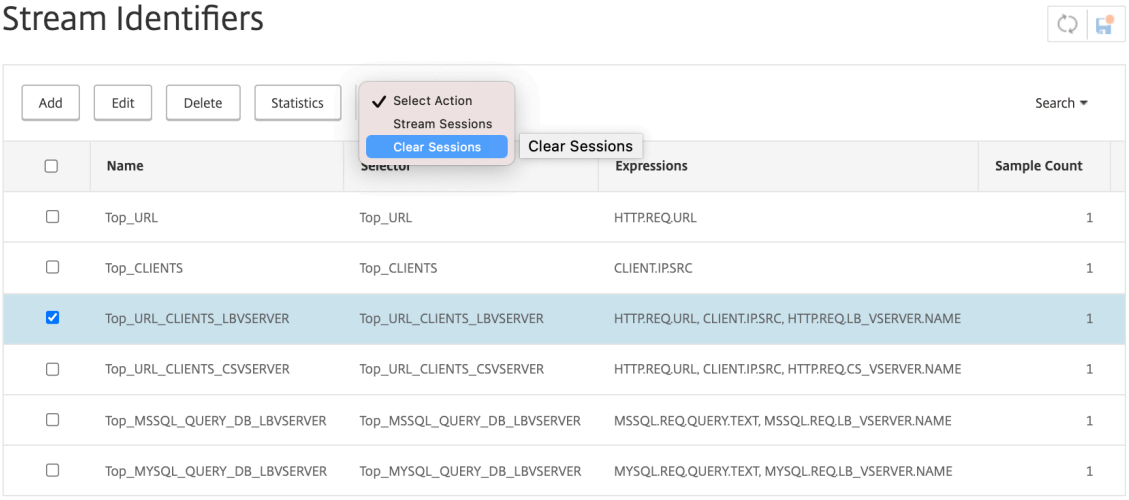
例

この例では、最初に stat stream identifier コマンドを使用しているため、clear stream session コマンドの結果を確認するために使用される stat stream identifier コマンドと比較できます。

```
>stat stream identifier myidentifier
Stream Session statistics
                    Req     BandW  RspTime      Conn
/aed....html          2         0         0         0
/                   636       303        12         0
 Done
>clear stream session myidentifier
 Done
>stat stream identifier myidentifier
 Done
<!--NeedCopy-->
```

GUI を使用してストリームセッションをクリアするには

1. **AppExpert** > アクション分析 > ストリーム識別子に移動します。
2. セッションをクリアするストリーム識別子を選択し、[セッションのクリア] をクリックします。



トラフィックを最適化するためのポリシーを構成する

August 15, 2023

アクション分析設定でセレクタと識別子のペアを有効にするには、統計情報を収集するトラフィックフロー内のポイントにペアを関連付ける必要があります。そのためには、詳細ポリシーを設定し、ポリシールールからストリーム識

別子を参照します。圧縮ポリシー、キャッシュポリシー、書き換えポリシー、アプリケーションファイアウォールポリシー、レスポンダーポリシー、およびアクションがブール式に基づくその他のポリシーを使用できます。

アクション分析機能には、データを収集および評価するための一連の高度なポリシー式と関数が導入されています。式 ANALYTICS.STREAM(<identifier_name>)は、使用する識別子を参照するために使用されます。COLLECT_STATS という式は、統計データの収集に使用されます。IS_TOP(<uint>)やIS_TOP_FREQUENTS(<uint>)などの関数は、リアルタイムのトラフィック最適化を自動的に決定するために使用されます。

- **IS_TOP (<number>)**。<number> 指定されたオブジェクトが要素の最上位にあるかどうかを検出します。たとえば、は上位 10 個の要素の中の要素です。複数の要素にカウントがある場合、それらは性質が似ていると見なされます。undef 条件を回避するには、ソート機能をオンにする必要があります。
- **IS_TOP_FREQUENTS ()**。指定されたオブジェクトが、最上位要素にある要素の最上位にあるかどうかを検出します。たとえば、上位のすべての要素の上位 50% の中の要素が維持されているかなどです。同じ値を持つ要素は、本質的に類似していると見なされます。undef 条件を回避するには、ソート機能をオンにする必要があります。

NetScaler アプライアンスがトラフィックからのデータ収集のみを行う必要があるのか、それともアクションも実行する必要があるのかを決定するのは、ポリシー構成です。アプライアンスが統計データのみを収集する必要がある場合は、ルールANALYTICS.STREAM(<identifier_name>).COLLECT_STATSとアクション NOOP を使用してポリシーを設定できます。NOOP ポリシーは、バインドポイントでプライオリティが最も高いポリシーである必要があります。統計情報を収集するだけの場合は、このポリシーで十分です。圧縮またはキャッシュする対象などのトラフィック最適化の決定は、統計データの手動による定期的な評価に基づいて行う必要があります。

統計情報の収集に加えて、アプライアンスがトラフィックに対してアクションを実行する必要がある場合は、目的のルールとアクションを持つ別のポリシーが後で評価されるように、NOOP ポリシーの gotoPriorityExpression パラメータを設定する必要があります。この 2 番目のポリシーには、ANALYTICS.STREAM(<identifier_name>) プレフィックスで始まるルールと、データを評価する関数が必要です。

次に、グローバルに設定およびバインドされる 2 つのレスポンダーポリシーの例を示します。ポリシー responder_stat_collection を使用すると、アプライアンスは識別子 myidentifier に基づいて統計情報を収集できます。ポリシー responder_notify は、収集されたデータを評価します。

例

```
1 > add responder action send_notification respondwith '"You are in the
     Top 10 list for bandwidth consumption"'
2  Done
3 > add responder policy responder_stat_collection' ANALYTICS.STREAM("
     myidentifier").COLLECT_STATS' NOOP
4 Done
5 > add responder policy responder_notify 'ANALYTICS.STREAM("myidentifier
     ").BANDWIDTH.IS_TOP(10)' send_notification
6  Done
```



```
> bind responder global responder_stat_collection 10 NEXT
 Done
> bind responder global responder_notify 20 END
 Done
<!--NeedCopy-->
```

ユーザーまたはクライアントデバイスごとの帯域幅消費量を制限する方法

August 15, 2023

Web サイト、アプリケーション、またはファイルホスティングサービスでは、すべてのユーザーが利用できるネットワークリソースとサーバーリソースには限りがあります。最も重要なリソースの 1 つは帯域幅です。一部のユーザーのみが帯域幅を大量に消費すると、ネットワークが混雑し、他のユーザーが利用できるリソースが低下する可能性があります。ネットワークの混雑を防ぐには、一時的なサービス拒否の手法を使用して、クライアントの帯域幅消費を制限する必要がある場合があります。たとえば、要求に至るまでの一定期間にあらかじめ設定された帯域幅値を超えた場合に、HTML ページでクライアント要求に応答するなどです。

一般に、帯域幅の使用量はクライアントデバイスごとまたはユーザーごとに調整できます。このユースケースは、1 時間の間にクライアントあたりの帯域幅消費量を 100 MB に制限する方法を示しています。このユースケースでは、ユーザー名を提供するカスタムヘッダーを使用して、1 時間にわたってユーザー 1 人あたりの帯域幅消費量を 100 MB に調整する方法も示しています。いずれの場合も、1 時間の移動期間における帯域幅消費量の追跡は、ストリーム識別子のインターバルパラメーターを 60 分に設定することで実現されます。ユースケースでは、制限を超えた HTML ページをインポートしてクライアントに送信する方法も示しています。HTML ページをインポートすると、これらのユースケースでのレスポンダーアクションの設定が簡単になるだけでなく、同じレスポンスを必要とするすべてのレスポンダーアクションの設定も簡単になります。

コマンドラインインターフェイスを使用してユーザーまたはクライアントデバイスごとの帯域幅消費を制限するには

コマンドラインインターフェイスで、次のタスクを実行して、クライアントまたはユーザーの帯域幅消費を制限するアクション分析を設定します。各ステップには、サンプルコマンドとその出力が含まれています。

1. 負荷分散構成を設定します。負荷分散仮想サーバー mysitevip を構成し、必要なすべてのサービスを構成します。サービスを仮想サーバーにバインドします。次の例では、10 個のサービスを作成し、そのサービスを mysitevip にバインドします。

```
> add lb vserver mysitevip HTTP 192.0.2.17 80
 Done
> add service service[1-10] 192.0.2.[240-249] HTTP 80
service "service1" added
service "service2" added
service "service3" added
.
.
```



```
.
service "service10" added
 Done
> bind lb vserver vserver1 service[1-10]
service "service1" bound
service "service2" bound
service "service3" bound
.
.
.
service "service10" bound
 Done
<!--NeedCopy-->
```

2. ストリームセレクタを設定します。以下のストリームセレクターのいずれかを設定します。

   - クライアントごとの帯域幅消費を制限するには、クライアントの IP アドレスを識別するストリームセレクターを設定します。

     ```
     > add stream selector myselector CLIENT.IP.SRC
      Done
     <!--NeedCopy-->
     ```

   - ユーザー名を提供する要求ヘッダーの値に基づいてユーザーごとの帯域幅消費を制限するには、ヘッダーを識別するストリームセレクターを設定します。次の例では、ヘッダーの名前は UserHeader です。

     ```
     > add stream selector myselector HTTP.REQ.HEADER( “UserHeader
         ” )
      Done
     <!--NeedCopy-->
     ```

3. ストリーム識別子を設定します。ストリームセレクターを使用するストリーム識別子を設定します。間隔パラメータを 60 分に設定します。

   ```
   > add stream identifier myidentifier myselector -interval 60 -
       sampleCount 1 -sort BANDWIDTH
    Done
   <!--NeedCopy-->
   ```

4. レスポンダアクションを設定します。帯域幅消費制限を超えたユーザーまたはクライアントに送信する HTML ページをインポートし、そのページをレスポンダーアクション crosssed_limits で使用します。

   ```
   > import responder htmlpage http://.1.1.1/stdpages/wait.html
       crossed-limits.html
   This operation may take some time, Please wait...

    Done
   > add responder action crossed_limits respondwithhtmlpage crossed-
       limits.html
    Done
   <!--NeedCopy-->
   ```


5. レスポンダポリシーを設定します。ANALYTICS.STREAM (「myidentifier」) .COLLECT_STATS というルールとアクション NOOP を使用してレスポンダーポリシー myrespol1 を設定します。次に、クライアントまたはユーザーが 100 MB の制限を超えているかどうかを判断するポリシー myrespol2 を設定します。ポリシー myrespol2 には、レスポンダーアクション crosssed_limits が設定されています。

```
> add responder policy myrespol1 'ANALYTICS.STREAM("myidentifier")
   .COLLECT_STATS' NOOP
 Done
> add responder policy myrespol2 'ANALYTICS.STREAM("myidentifier")
   .BANDWIDTH.GT(104857600)' crossed_limits
 Done
<!--NeedCopy-->
```

6. レスポンダーポリシーを負荷分散仮想サーバーにバインドします。統計データのみを収集するポリシー myrespol1 の方が優先度が高く、GOTO 式が NEXT である必要があります。

```
> bind lb vserver mysitevip -policyName myrespol1 -priority 1 -
   gotoPriorityExpression NEXT
 Done
> bind lb vserver mysitevip -policyName myrespol2 -priority 2 -
   gotoPriorityExpression END
 Done
<!--NeedCopy-->
```

7. 構成をテストします。複数のクライアントまたはユーザーからのテスト HTTP リクエストを負荷分散仮想サーバーに送信し、stat stream identifier コマンドを使用して、指定された識別子について収集された統計を表示して、構成をテストします。次の出力は、クライアントの統計を表示します。

```
> stat stream identifier myidentifier -sortBy BandW  - fullValues
Stream Session statistics
                            Req             BandW
192.0.2.30                  5000            3761
192.0.2.31                  29              2602
192.0.2.32                  25              51

                            RspTime         Conn
192.0.2.30                  2               0
192.0.2.31                  0               0
192.0.2.32                  0               0
 Done
>
<!--NeedCopy-->
```

AppExpert アプリケーション

> 警告アプリケーションテンプレート機能は廃止されました
>
> 。別の方法として、StyleBook を使用することもできます。詳細については、「StyleBooks 」および「Web

> アプリケーションファイアウォール StyleBook」を参照してください。

AppExpert アプリケーションは、NetScaler アプライアンスで設定した構成のコレクションです。AppExpert アプリケーションの管理は、GUI（GUI）によって簡素化され、各トラフィックサブセットを処理するためのアプリケーショントラフィックサブセット、および個別のセキュリティおよび最適化ポリシーのセットを指定できます。また、展開手順が 1 つのビューに統合されるため、クライアントのターゲット IP アドレスをすばやく構成し、ホストサーバーを指定できます。

AppExpert アプリケーションがセットアップされたら、アプリケーションが正しく動作していることを確認する必要があります。必要に応じて、要件に合わせて構成をカスタマイズできます。

さまざまなアプリケーションコンポーネント、統計情報、およびアプリケーションビジュアライザーのカウンタを表示して、定期的に構成を検証および監視できます。また、アプリケーションの認証、認可、監査（認証、認可、監査）のポリシーを構成することもできます。

AppExpert アプリケーションの用語

AppExpert アプリケーション機能で使用される用語と、その用語が使用されるエンティティの説明を以下に示します。

パブリックエンドポイント。NetScaler アプライアンスが関連する Web アプリケーションのクライアント要求を受信する IP アドレスとポートの組み合わせです。パブリックエンドポイントは、HTTP トラフィックまたはセキュア HTTP（HTTPS）トラフィックを受信するように設定できます。Web アプリケーションに対するすべてのクライアント要求は、パブリックエンドポイントに送信する必要があります。AppExpert アプリケーションには、複数のエンドポイントを割り当てることができます。

アプリケーションユニット。Web アプリケーショントラフィックのサブセットを処理し、関連付けられたコンテンツをホストする一連のサービスを負荷分散する AppExpert アプリケーションエンティティ。アプリケーションユニットが管理する必要があるトラフィックのサブセットは、ルールによって定義されます。また、各アプリケーションユニットは、管理する要求と応答に対して、独自のトラフィック最適化およびセキュリティポリシーのセットを定義します。これらのポリシーに関連する NetScaler サービスは、圧縮、キャッシュ、リライト、レスポンダー、およびアプリケーションファイアウォールです。

デフォルトでは、少なくとも 1 つのアプリケーションユニットを持つすべての AppExpert アプリケーションには、デフォルトのアプリケーションユニットが含まれていますが、これは削除できません。デフォルトのアプリケーションユニットは、要求を識別するための規則に関連付けられず、常にアプリケーションユニットの順序で最後に配置されます。これは、他のアプリケーションユニットに対して設定されたルールと一致しない要求を処理するための一連のポリシーを定義します。これにより、すべてのクライアント要求が確実に処理されます。

Service. Web アプリケーションインスタンスをホストするサーバーの IP アドレスと、アプリケーションがサーバー上でマップされるポートの組み合わせ（形式\<IP address\>:\<Port\>）。多くのリクエストを処理する Web アプリケーションは、複数のサーバーでホストされます。各サーバーは Web アプリケーションのインスタンスをホストすると言われ、Web アプリケーションの各インスタンスは NetScaler アプライアンス上のサービスによって表されます。

アプリケーションユニット規則。アプリケーションユニットのトラフィックサブセットの特性を定義する高度なポリシー式。次のルールの例は、4 つのイメージタイプで構成されるトラフィックサブセットを識別する高度なポリシー式です。

```
HTTP.REQ.URL.SUFFIX.EQ("bmp")|| HTTP.REQ.URL.SUFFIX.EQ("gif")|| HTTP.REQ.URL.SUFFIX.EQ("png")|| HTTP.REQ.URL.SUFFIX.EQ("jpg")
```

高度なポリシー式の詳細については、「ポリシーと式」を参照してください。

トラフィックサブセット。トラフィック最適化ポリシーとセキュリティポリシーの共通セットを必要とするクライアント要求のセット。トラフィックサブセットはアプリケーションユニットによって管理され、ルールによって定義されます。

AppExpert アプリケーションの仕組み

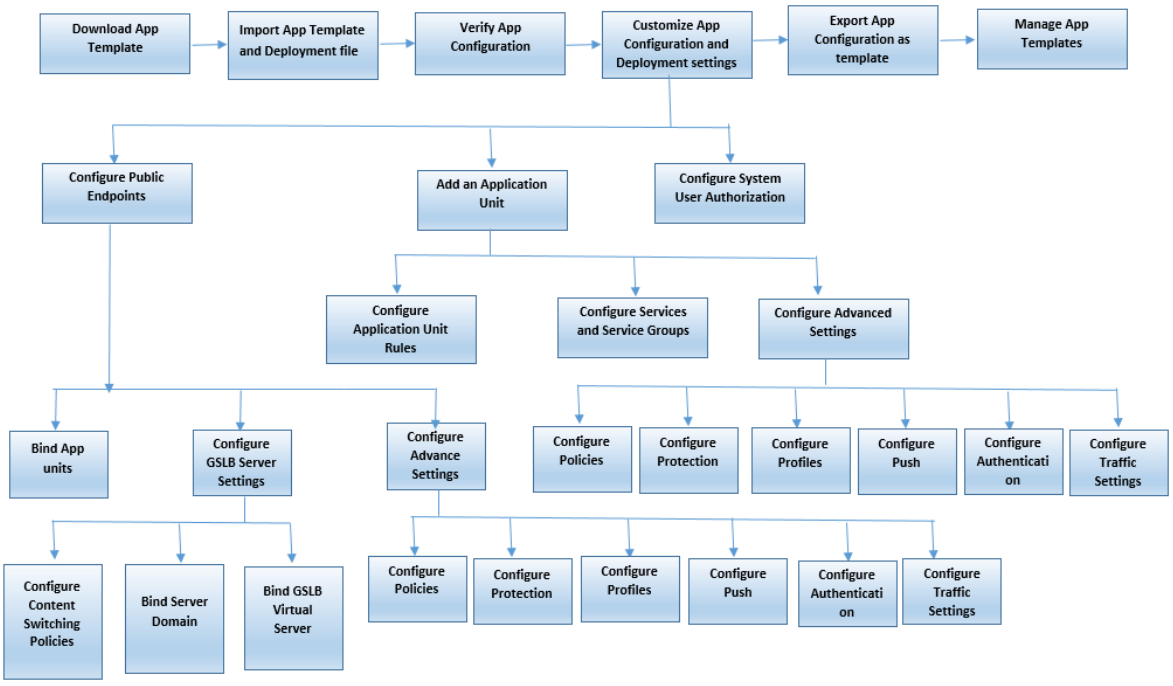
August 15, 2023

エンドポイントがクライアント要求を受信すると、NetScaler アプライアンスは最上位のアプリケーションユニットに設定されたルールに照らして要求を評価します。要求がこの規則を満たす場合、要求はアプリケーションユニットに設定されたポリシーによって処理され、サービスに転送されます。サービスの選択は、アプリケーションに対して構成されているサービス、およびアプリケーションユニットに対して構成されている負荷分散アルゴリズムや永続化方法などの設定によって異なります。

要求が規則を満たさない場合、要求は次に上位のアプリケーション単位の規則に対して評価されます。この順序で、要求は、要求が規則を満たすまで、各アプリケーション単位規則に対して評価されます。要求が設定されたルールのいずれも満たさない場合、既定のアプリケーション単位 (常に最後のアプリケーション単位) によって処理されます。

AppExpert アプリケーションには、複数のパブリックエンドポイントを設定できます。このような構成では、デフォルトでは、各アプリケーションユニットはすべてのパブリックエンドポイントによって受信された要求を処理し、アプリケーションに対して構成されているすべてのサービスの負荷を分散します。ただし、アプリケーションユニットがパブリックエンドポイントのサブセットからのトラフィックのみを処理し、AppExpert アプリケーション用に構成されているサービスのサブセットのみをロードバランシングするように指定できます。

次のフロー図は、組み込みアプリケーションテンプレートを使用するための AppExpert アプリケーションフローシーケンスを示しています。



テンプレートを使用せずにカスタマイズされたアプリケーションを作成する場合は、次の操作を行います。

1. カスタムアプリケーションを作成します。
2. アプリケーションと展開の設定を構成します。
3. 構成を新しいテンプレートファイルにエクスポートします (オプション)。
4. テンプレートファイルを、同様の AppExpert アプリケーション構成を必要とする他の NetScaler アプライアンスにインポートします。

構成をカスタマイズする

August 15, 2023

AppExpert アプリケーションが正しく動作していることを確認したら、要件に合わせて構成をカスタマイズできます。

AppExpert アプリケーション構成が正しく機能していることを確認したら、要件に合わせてアプリケーションと展開設定を構成できます。アプリケーションテンプレートと展開ファイルをインポートすると、使用可能な構成設定（アプリケーションユニット、アプリケーションユニットのルール、ポリシー、永続性設定、負荷分散方法、プロファイル、トラフィック設定など）がターゲットアプリケーションに自動的に入力されます。このアプリケーションでは、トラフィックサブセットごとに、パブリックエンドポイント、サービス、サービスグループなどの展開設定を構成できます。テンプレートに含まれていないトラフィックサブセットを AppExpert アプリケーションで管理する場合は、トラフィックサブセットのアプリケーションユニットを追加するか、既存のアプリケーションユニットを変更します。

構成をカスタマイズしたら、アプリケーションが管理する各トラフィックサブセットの評価順序を指定することもできます。

AppExpert アプリケーションの構成は、次の手順で構成されます。

1. パブリックエンドポイントの設定
2. アプリケーションユニットの設定
3. 評価順序の指定
4. Visualizer を使用したアプリケーション構成の表示

また、テンプレートが提供するポリシーを設定することもできます。AppExpert アプリケーションテンプレートに、リライトやアプリケーションファイアウォールなどの特定の NetScaler 機能のポリシーが含まれていない場合は、独自のポリシーを構成できます。

パブリックエンドポイントの設定

August 15, 2023

AppExpert アプリケーションのインポート時にパブリックエンドポイントを指定しなかった場合は、アプリケーションの作成後にパブリックエンドポイントを指定できます。AppExpert アプリケーションには、HTTP タイプのパブリックエンドポイントと HTTPS タイプのパブリックエンドポイントを 1 つ設定できます。

エンドポイントがアプリケーションに対してすでに構成されている場合は、AppExpert アプリケーションからエンドポイントの関連付けを解除し、不要になったエンドポイントを削除できます。パブリックエンドポイントと AppExpert アプリケーションの関連付けを解除すると、エンドポイントは関連付けられたアプリケーションユニットから自動的にバインド解除されますが、システムからは削除されません。

AppExpert アプリケーションのパブリックエンドポイントを設定するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、パブリックエンドポイントを構成するアプリケーションを右クリックし、[編集] をクリックします。
3. [アプリケーション] ページで、[パブリックエンドポイント] セクションに移動し、鉛筆アイコンをクリックします。
4. [パブリックエンドポイント] スライダーで、次のパラメーターを設定します。

    a) パブリックエンドポイントタイプ。ラジオボタンを選択して、エンドポイントタイプを定義します。
    b) [名前]。パブリックエンドポイントの名前。
    c) IP アドレス。パブリックエンドポイントの IP アドレス。
    d) ポート。パブリックエンドポイントのポート番号。
    e) プロトコル。プロトコルタイプとして [HTTP] または [HTTPS] を選択します。

5. [続行] をクリックします。

6. [アプリケーション単位] セクションで、一覧からアプリケーション単位を選択します。
7. [**Continue**] をクリックして、ポリシーとサーバーの詳細を設定します。
8. 「**OK**」をクリックし、「完了」をクリックします。
9. [閉じる] をクリックします。

「パブリックエンドポイントの設定」ダイアログボックスのパラメータの詳細については、「コンテンツの切り替え」を参照してください。

アプリケーションユニットのサービスとサービスグループを構成する

August 15, 2023

サービスまたはサービスグループを構成するときは、既存のサービスまたはサービスグループを変更するか、AppExpert アプリケーションに新しいサービスを追加します。アプリケーションテンプレートをインポートしたときに指定しなかったサービスまたはサービスグループを追加します。アプリケーションのインスタンスをホストするサーバーの数を増やすときも、サービスとサービスグループを追加します。アプリケーションユニットのサービスおよびサービスグループは、AppExpert アプリケーションのサービスまたはサービスグループを構成した後にのみ構成できます。

AppExpert アプリケーションのサービスまたはサービスグループを設定するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、アプリケーションを右クリックし、[編集**]** をクリックします。
3. [アプリケーション] ページで、アプリケーションユニットを選択し、[続行] をクリックします。
4. [サービスとサービスグループ] セクションで、次の操作を行います。

    a) [サービスバインディング] スライダーで、次のパラメーターを設定します。

        i. Service. リストから負荷分散サービスを選択するか、新しいサービスを作成します。
        ii. 重量。サービスの重み値を指定します。

    b) [バインド**]**、[完了**]** の順にクリックします。
    c) ServiceGroup Binding スライダで、次のパラメータを設定します。

        i. サービスグループ名。負荷分散サービスグループを選択するか、新しいサービスグループを作成します。
        ii. 「バインド」をクリックし、「完了」をクリックします。

    d) [完了] をクリックします。

5. [続行] をクリックして、他の構成を設定します。

アプリケーションユニットを作成する

August 15, 2023

Web アプリケーションの実装に固有であるか、テンプレートで定義されていないトラフィックサブセットのアプリケーションユニットを追加する必要がある場合があります。アプリケーションユニットを作成するときは、アプリケーションユニットのルールを構成する必要があります。

AppExpert アプリケーションのアプリケーションユニットを作成するには

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、アプリケーションユニットを追加するアプリケーションを右クリックし、[追加] をクリックします。
3. [アプリケーション] ページで、[アプリケーション単位] セクションに移動し、鉛筆アイコンをクリックします。

アプリケーションユニットのポリシー式を構成するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、アプリケーションユニットを追加するアプリケーションを右クリックし、[追加] をクリックします。
3. ［アプリケーション］ページで、［アプリケーションユニット］セクションに移動し、［**+** ］アイコンをクリックします。ユニットを作成し、ポリシー式を追加します。
4. 新しい式の形式を指定するには、次のいずれかを実行します。

    a) [規則] ボックスでポリシー式を構成することを指定するには、[クラシック構文] をクリックします。
    b) [ルール] ボックスで高度な式を構成することを指定するには、[高度なポリシー] をクリックします。
    c) [ルール] ボックスで、式を設定します。

5. ［**OK**］をクリックします。

アプリケーションユニットルールの構成

August 15, 2023

アプリケーションユニットルールを設定して、特定のタイプのトラフィックを含めるか除外するかを設定できます。ルールを設定するときに、式の構文を定義することもできます。

アプリケーションユニットルールを構成するには、次の手順を実行します。

1. GUI のナビゲーションウィンドウで、[AppExpert] を展開し、[アプリケーション] をクリックします。
2. 詳細ウィンドウで、規則を変更するアプリケーションユニットを右クリックし、[開く] をクリックします。

3. [アプリケーションユニットの構成] ダイアログボックスで、次の操作を行います。

   a) 新しい式の形式を指定するには、次のいずれかを実行します。

      - [規則] ボックスで高度なポリシー式を構成することを指定するには、[クラシック構文] をクリックします。
      - [ルール] ボックスで高度な式を構成することを指定するには、[高度なポリシー] をクリックします。

   b) [ルール] ボックスで、式を設定します。

4. [**OK**] をクリックします。

アプリケーションユニットのポリシーの設定

August 15, 2023

AppExpert アプリケーションの場合は、圧縮、キャッシュ、書き換え、レスポンダー、およびアプリケーションファイアウォールのポリシーを構成できます。Citrix Community Web サイトからダウンロードするテンプレートは、最も一般的なアプリケーション管理要件を満たす一連のポリシーを提供します。これらのポリシーを微調整またはカスタマイズしたい場合があります。特定のアプリケーションユニットに提供されるポリシーのセットに特定の機能のポリシーが含まれていない場合は、その機能用の独自のポリシーを作成してバインドできます。

テンプレートを使用せずに AppExpert アプリケーションを作成する場合は、Web アプリケーションが必要とするすべてのポリシーを構成する必要があります。

GUI では、さまざまなアイコンを使用して、機能に対してポリシーが設定されているかどうかを示します。アプリケーションユニットでは、特定の機能に対してポリシーが設定されている場合は、その機能を表すアイコンが表示されます。たとえば、アプリケーションユニットに対して圧縮ポリシーが構成されている場合、アプリケーションユニットの [圧縮] 列に圧縮アイコンが表示されます。ポリシーが設定されていない機能については、プラス記号（+）を示すアイコンが表示されます。

注：アプリケーションユニットのポリシーを構成する場合、クラシックポリシーまたは詳細ポリシーのポリシーおよび式を構成する必要がある場合があります。さらに、高度なポリシーポリシーを構成するときに、Goto 式などのパラメータを指定し、ポリシーバンクを呼び出す必要がある場合があります。

両方の形式のポリシーと式を設定する方法については、「ポリシーと式」を参照してください。

圧縮ポリシーの構成

従来のポリシーまたは高度なポリシーを使用して圧縮を構成できますが、両方のタイプの圧縮ポリシーを同じアプリケーションユニットにバインドすることはできません。

アプリケーションユニットの圧縮ポリシーを構成するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ペインで、構成するアプリケーションユニットの行で、[圧縮] 列に表示されるアイコンをクリックします。
3. [圧縮ポリシーの構成] ダイアログボックスで、実行する構成タスクに応じて、次の 1 つまたは複数の操作を行います。

   - 高度なポリシー圧縮ポリシーを構成する場合は、[高度なポリシーに切り替える] をクリックします。クラシック圧縮ポリシーをバインドまたは構成する場合、および詳細ポリシービューを表示している場合は、[クラシック構文に切り替え] をクリックしてクラシックポリシービューに戻り、バインドされたクラシックポリシーの変更を開始するか、新しいクラシック圧縮ポリシーを作成してバインドします。

     重要: この設定は、ポリシーを挿入するときに表示されるポリシーも決定します。たとえば、[詳細ポリシー] ビューを表示している場合、[ポリシーの挿入] をクリックすると、[ポリシー名] 列に表示されるリストには、高度なポリシーポリシーのみが含まれます。両方のタイプのポリシーをアプリケーションユニットにバインドすることはできません。
   - クラシックポリシーを構成する場合は、ポリシーをリクエスト時に評価するか、レスポンス時に評価するかに応じて、[Request] または [Response] をクリックします。

     アプリケーションユニットには、要求時間および応答時間のクラシック圧縮ポリシーの両方を設定できます。すべての要求時間ポリシーを評価した後、一致するものが見つからない場合、アプライアンスは応答時間ポリシーを評価します。
   - アプリケーションユニットに既にバインドされている圧縮ポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[圧縮ポリシーの構成] ダイアログボックスで、ポリシーを変更し、[OK] をクリックします。

     圧縮ポリシーの変更の詳細については、「圧縮」を参照してください。
   - ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
   - ポリシーに割り当てられているプライオリティを変更するには、プライオリティ値をダブルクリックし、新しい値を入力します。
   - 割り当てられた優先度を再生成するには、「優先度を再生成する」をクリックします。
   - 新しいポリシーを挿入するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列に表示されるリストで [新しいポリシー] をクリックします。次に、[圧縮ポリシーの作成] ダイアログボックスで、ポリシーを構成し、[作成] をクリックします。

     圧縮ポリシーの変更の詳細については、「圧縮」を参照してください。
   - 高度なポリシー式を設定する場合は、次の手順を実行します。
     - [式に移動] 列で、[式を移動] を選択します。
     - [Invoke] 列で、現在のポリシーが TRUE と評価された場合に呼び出すポリシーバンクを指定します。
4. [変更を適用] をクリックし、[閉じる] をクリックします。

キャッシングポリシーの設定

キャッシュポリシーを構成するには、高度なポリシーポリシーと式のみを使用できます。

アプリケーションユニットのキャッシュポリシーを構成するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ペインで、構成するアプリケーションユニットの行で、[キャッシュ] 列に表示されるアイコンをクリックします。
3. [キャッシュポリシーの構成] ダイアログボックスで、実行する構成タスクに応じて、次の 1 つまたは複数の操作を行います。

    - ポリシーをリクエスト時に評価するか、レスポンス時に評価するかに応じて、[リクエスト] または [レスポンス] をクリックします。

      アプリケーションユニットには、要求時間と応答時間の両方のキャッシングポリシーを設定できます。すべての要求時間ポリシーを評価した後、一致するものが見つからない場合、アプライアンスは応答時間ポリシーを評価します。
    - アプリケーションユニットに既にバインドされているキャッシュポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[キャッシュポリシーの構成] ダイアログボックスで、ポリシーを変更し、[**OK**] をクリックします。

      キャッシュポリシーの変更については、統合キャッシュを参照してください。
    - ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
    - ポリシーに割り当てられているプライオリティを変更するには、プライオリティ値をダブルクリックし、新しい値を入力します。
    - 割り当てられた優先度を再生成するには、「優先度を再生成する」をクリックします。
    - 新しいポリシーを挿入するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列に表示されるリストで [新しいポリシー] をクリックします。次に、**[**キャッシュポリシーの作成] ダイアログボックスで、ポリシーを構成し、[作成] をクリックします。

      キャッシュポリシーの変更については、統合キャッシュを参照してください。
    - [式に移動] 列で、[式を移動] を選択します。
    - [Invoke] 列で、現在のポリシーが TRUE と評価された場合に呼び出すポリシーバンクを指定します。
4. [変更を適用] をクリックし、[閉じる] をクリックします。

書き換えポリシーの設定

書き換えポリシーを構成するには、高度なポリシーポリシーと式のみを使用できます。

アプリケーションユニットの書き換えポリシーを構成するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ペインで、構成するアプリケーションユニットの行で、[書き換え] 列に表示されるアイコンをクリックします。
3. [書き換えポリシーの構成] ダイアログボックスで、実行する構成タスクに応じて、次の 1 つまたは複数の操作を行います。

    - ポリシーをリクエスト時に評価するか、レスポンス時に評価するかに応じて、[リクエスト] または [レスポンス] をクリックします。

      アプリケーションユニットには、要求時間および応答時間の書き換えポリシーの両方を設定できます。すべての要求時間ポリシーを評価した後、一致するものが見つからない場合、アプライアンスは応答時間ポリシーを評価します。
    - アプリケーションユニットに既にバインドされている書き換えポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[書き換えポリシーの構成] ダイアログボックスで、ポリシーを変更し、[**OK**] をクリックします。

      書き換えポリシーの変更については、書き換えを参照してください。
    - ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
    - ポリシーに割り当てられているプライオリティを変更するには、プライオリティ値をダブルクリックし、新しい値を入力します。
    - 割り当てられた優先度を再生成するには、「優先度を再生成する」をクリックします。
    - 新しいポリシーを挿入するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列に表示されるリストで [新しいポリシー**]** をクリックします。次に、[書き換えポリシーの作成] ダイアログボックスで、ポリシーを構成し、[作成] をクリックします。

      書き換えポリシーの変更については、書き換えを参照してください。
    - [式に移動] 列で、[式を移動] を選択します。
    - [Invoke] 列で、現在のポリシーが TRUE と評価された場合に呼び出すポリシーバンクを指定します。
4. [変更を適用] をクリックし、[閉じる] をクリックします。

レスポンダーポリシーの設定

レスポンダーポリシーを構成するには、高度なポリシーポリシーと式のみを使用できます。

アプリケーションユニットのレスポンダーポリシーを構成するには:

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウの、構成するアプリケーションユニットの行で、[レスポンダー] 列に表示されるアイコンをクリックします。
3. [レスポンダーポリシーの構成] ダイアログボックスで、実行する構成タスクに応じて、次の 1 つまたは複数の操作を行います。

- アプリケーションユニットに既にバインドされているフィルタポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[レスポンダーポリシーの構成] ダイアログボックスでポリシーを変更し、[**OK**] をクリックします。
  レスポンダーポリシーの変更の詳細については、レスポンダーを参照してください。
- ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
- ポリシーに割り当てられているプライオリティを変更するには、プライオリティ値をダブルクリックし、新しい値を入力します。
- 割り当てられた優先度を再生成するには、「優先度を再生成する」をクリックします。
- 新しいポリシーを挿入するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列に表示されるリストで [新しいポリシー] をクリックします。次に、[レスポンダーポリシーの作成] ダイアログボックスでポリシーを構成し、[作成] をクリックします。
  レスポンダーポリシーの変更の詳細については、レスポンダーを参照してください。
- [式に移動] 列で、[式を移動] を選択します。
- [Invoke] 列で、現在のポリシーが TRUE と評価された場合に呼び出すポリシーバンクを指定します。

4. [変更を適用] をクリックし、[閉じる] をクリックします。

アプリケーションファイアウォールポリシーの設定

アプリケーションファイアウォールには、従来のポリシーポリシーと高度なポリシーポリシーと式の両方を設定できます。ただし、あるタイプのポリシーがすでにグローバルにバインドされている場合、またはアプライアンスで構成されている仮想サーバにバインドされている場合は、もう一方のタイプのポリシーをアプリケーションユニットにバインドすることはできません。たとえば、詳細ポリシーがすでにグローバルまたは仮想サーバにバインドされている場合、クラシックポリシーをアプリケーションユニットにバインドすることはできません。

アプリケーションユニットのアプリケーションファイアウォールポリシーを設定するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ペインで、構成するアプリケーションユニットの行で、[アプリケーションファイアウォール] 列に表示されるアイコンをクリックします。
3. [アプリケーションファイアウォールポリシーの構成] ダイアログボックスで、実行する構成タスクに応じて、次の 1 つまたは複数の操作を行います。

   - アプリケーションファイアウォールポリシーに設定する式の種類に応じて、[クラシック式] または [高度な式] をクリックします。
     重要: この設定は、ポリシーを挿入するときに表示されるポリシーも決定します。たとえば、[高度な式] を選択した場合、[ポリシーの挿入] をクリックすると、[ポリシー名] 列に表示されるリストには、高度なポリシーポリシーのみが含まれます。両方のタイプのポリシーをアプリケーションユニットにバインドすることはできません。このオプションは、いずれかのタイプのポリシーがすでにグローバルまたは仮想サーバにバインドされている場合は使用できません。

- アプリケーションユニットに既にバインドされているアプリケーションファイアウォールポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[アプリケーションファイアウォールポリシーの構成] ダイアログボックスで、ポリシーを変更し、[OK] をクリックします。

  アプリケーションファイアウォールポリシーの変更の詳細については、「ポリシー」を参照してください。
- ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
- ポリシーに割り当てられているプライオリティを変更するには、プライオリティ値をダブルクリックし、新しい値を入力します。
- 割り当てられた優先度を再生成するには、「優先度を再生成する」をクリックします。
- 新しいポリシーを挿入するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列に表示されるリストで [新しいポリシー] をクリックします。次に、**[**アプリケーションファイアウォールポリシーの作成] ダイアログボックスで、ポリシーを構成し、[作成] をクリックします。

  アプリケーションファイアウォールポリシーの変更の詳細については、「ポリシー」を参照してください。

4. [変更を適用] をクリックし、[閉じる] をクリックします。

アプリケーションユニットの設定

August 15, 2023

GUI を使用してアプリケーションユニットを構成するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] > [アプリケーションユニット] セクションに移動し、プラスアイコンをクリックして、トラフィックサブセットの新しいアプリケーションユニットを追加します。
2. [アプリケーション単位] スライダで、次のパラメータを設定します。

   - 名前
   - 式

   エクスプレッションを挿入するには、エクスプレッションコンポーネントを手動で追加するか、[エクスプレッションエディタ] リンクを使用します。式を手動で追加するには、セレクタコンポーネントを入力し、ピリオド (.) を入力して、次のコンポーネントを選択できるリストを表示します。たとえば、「HTTP」と入力し、ピリオドを入力します。ドロップダウンメニューが表示されます。このメニューの内容には、入力した最初のキーワードの後に続くキーワードが表示されます。ドロップダウンメニューからコンポーネントを選択します。[式 *] テキストボックスに、式に追加したコンポーネント (HTTP.REQ など) が表示されます。完全な表現が形成されるまで、コンポーネントを追加し続けます。

式の作成を支援したい場合は、[式エディタ] リンクを使用できます。[式エディタ] ページでは、ドロップダウンボックスからコンポーネントを選択して式を作成できます。コンポーネントを選択し、[完了] をクリックして、[アプリケーションユニット] ページに式を挿入します。

3. [**Continue**] をクリックして、サービスとサービスグループをバインドします。
4. [サービス] セクションをクリックして、仮想サービスを選択または追加し、アプリケーションユニットにバインドします。
5. [**Continue**] をクリックし、[**Service Group**] セクションをクリックして、仮想サービスグループを選択または追加し、アプリケーションユニットにバインドします。
6. 「****** バインドして続行 ****** 」をクリックして、アプリケーションユニットの詳細設定（ポリシー、メソッド、永続性、保護、プロファイル、プッシュ、認証、トラフィック設定など）を構成します。
7. 各セクションのプラスアイコンをクリックして、構成パラメータを設定します。
8. [**OK]**、[完了**]** の順にクリックします。

GUI を使用してアプリケーションのアプリケーションユニットを編集するには、次の手順を実行します。

[**AppExpert**] > [アプリケーション] に移動し、アプリケーションを選択して [編集] をクリックします。[アプリケーションユニット] セクションで、エンティティを選択し、編集アイコンをクリックして、アプリケーションユニット設定を変更します。

注: 既存のアプリケーションユニットの名前とルール式は変更できません。

NetScaler のビデオチュートリアルでは、NetScaler の機能を簡単かつ簡単に理解できます。アプリケーションユニットの構成方法については、https://www.youtube.com/watch?v=bJ5_i8fV2hc ビデオをご覧ください。

アプリケーションのパブリックエンドポイントの設定

August 15, 2023

GUI を使用してアプリケーションのパブリックエンドポイントを設定するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動し、アプリケーションエンティティを選択して [編集] をクリックします。
2. [パブリックエンドポイント] セクションで、[**+**] をクリックして新しいパブリックエンドポイントを設定します。
3. [パブリックエンドポイント] スライダーで、次のいずれかを実行します。

   a) [新規] をクリックして、新しいエンドポイントを作成します。
   b) [既存のパブリックエンドポイント] をクリックして、ドロップダウンリストからエンドポイントを選択します。

4. 次のエンドポイントパラメータを設定します。

    a) 名前
    b) IP アドレス
    c) プロトコル
    d) ポート

5. [続行] をクリックして、アプリケーションユニット、GSLB サーバーバインディング、ポリシー、プロファイル、プッシュ、トラフィック設定、認証などの追加設定を構成します。
6. [**OK]**、[完了**]** の順にクリックします。
7. [続行] をクリックし、[完了] をクリックします。

GUI を使用してアプリケーションのパブリックエンドポイントを編集するには、次の手順を実行します。

[**AppExpert**] > [アプリケーション] に移動し、アプリケーションを選択して [編集] をクリックします。[パブリックエンドポイント] セクションで、エンドポイントを選択し、ペンアイコンをクリックして、エンドポイントの設定を変更します。

GUI を使用してアプリケーションのパブリックエンドポイントを削除するには

[**AppExpert**] > [アプリケーション] > [パブリックエンドポイント**]** に移動し、ペンアイコンをクリックして、エンティティの横にある削除アイコンを表示します。

NetScaler のビデオチュートリアルでは、NetScaler の機能を簡単かつ簡単に理解できます。パブリックエンドポイントの設定方法については、https://www.youtube.com/watch?v=z4v-edQiVpw ビデオをご覧ください。

アプリケーション単位の評価順序の指定

August 15, 2023

アプリケーションユニットのルールは、GUI に配置された順序で評価されます。最上位のアプリケーションユニットに対して構成されるルールが常に最初に設定され、その後に 2 番目に上位のアプリケーションユニットに対して設定されたルールが続きます。既定のアプリケーション単位は常に最後に評価されます。

要求がアプリケーションユニットに対して構成されたルールと一致すると、その要求はアプリケーションユニットによって処理され、それ以上マッチングは実行されません。したがって、2 つ以上のアプリケーションユニットのトラフィックサブセットが重複する場合、アプリケーションユニットの評価の順序が重要な要素になります。2 つ以上のアプリケーションユニットのトラフィックサブセットが重複する場合、着信要求がアプリケーションユニットのルールと照合される順序を指定する必要があります。

アプリケーション単位の評価順序を指定するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動し、アプリケーションを選択して [編集] をクリックします。[アプリケーションユニット] セクションで、鉛筆アイコンをクリックし、アプリケーションユニットの名前の左

側にあるチェックボックスの上にカーソルを置きます。チェックボックスの横に表示されるアイコンをクリックし、マウスを押したままにして、アプリケーションを優先リスト内の新しい場所まで上下にドラッグします。

アプリケーションユニットの永続性グループの設定

August 15, 2023

AppExpert アプリケーションのアプリケーションユニットの永続性グループを構成できます。AppExpert アプリケーションのコンテキストでは、永続性グループは、共通の永続性設定を適用する目的で 1 つのエンティティとして扱うことができるアプリケーションユニットのグループです。アプリケーションをアプリケーションテンプレートファイルにエクスポートすると、持続性グループの設定が含まれ、AppExpert アプリケーションをインポートするときにアプリケーションユニットに自動的に適用されます。

GUI を使用してアプリケーションの永続性グループを設定するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2.「アプリケーションビュー」ダイアログ・ボックスで、永続性グループを構成するアプリケーション・ユニットのアプリケーション名をクリックし、「持続性グループの構成」をクリックします。
3.「永続性グループの構成」ダイアログ・ボックスで、次のいずれかを実行します。
   - 永続性グループを追加するには、[追加] をクリックします。
   - 持続性グループを変更するには、「開く」をクリックします。
4.「永続性グループの作成」または「永続性グループの構成」ダイアログ・ボックスで、次のパラメータを設定します。
   - Group Name：永続性グループの名前。NetScaler アプライアンスが永続性グループをアプリケーション構成の一部として認識するには、AppExpert アプリケーションの名前を永続性グループの名前にプレフィックスとして含める必要があります。したがって、デフォルトでは、アプライアンスの [グループ名] ボックスにプレフィックスが表示され、そのプレフィックスを削除することはできません。接頭辞の後に任意の名前を入力します。
   - [Persistence]：仮想サーバーの永続性のタイプ。SOURCEIP を選択した場合は、[IPv4 Netmask] ボックスに、永続性セッションを作成するときにアプライアンスが考慮する必要があるビット数を指定するネットワークマスクを入力します。COOKIEINSERT を選択した場合は、「Cookie ドメイン」ボックスと「Cookie 名」ボックスで、Set-Cookie ディレクティブで送信するドメイン属性と Cookie の名前をそれぞれ指定します。
   - タイムアウト-永続セッションが有効な期間。
   - Backup Persistence：グループのバックアップ永続性のタイプ。
   - Backup Timeout：バックアップの永続性が有効になる期間（分単位）。
   - アプリケーションユニット-アプリケーションユニットを永続性グループに追加するには、[使用可能なアプリケーションユニット] ボックスでアプリケーションユニットをクリックし、[追加] をクリックし

ます。持続性グループからアプリケーションユニットを削除するには、[構成済みのアプリケーションユニット] ボックスでアプリケーションユニットをクリックし、[削除] をクリックします。

5. [**OK**] をクリックします。

アプリケーションビジュアライザーを使用した **AppExpert** アプリケーションの表示とエンティティの設定

August 15, 2023

ビジュアライザー機能は、アプリケーションの構成をグラフィカルに表示します。これには、パブリックエンドポイントの名前、パブリックエンドポイントに割り当てられたアプリケーションユニット、およびアプリケーションにバインドされたポリシーとサービスの数が含まれます。ビジュアライザーを使用して、AppExpert アプリケーションの構成の概要を視覚的に把握し、表示されたエンティティの一部を構成できます。デフォルトでは、ビジュアライザーには、選択したアプリケーションのアプリケーションユニット、サービス、およびモニターが表示されます。

アプリケーションビジュアライザーを使用して AppExpert アプリケーションを表示するには

1. [**AppExpert**] > [アプリケーション] に移動し、アプリケーションエンティティを選択して [ビジュアライザー] をクリックします。

ユーザー認証、承認、および監査の設定

August 15, 2023

ユーザーとグループの承認を構成して、AppExpert アプリケーションへのアクセスを有効にすることができます。権限を設定する AAA ユーザまたはグループがまだ作成されていない場合は、AppExpert から作成し、アプリケーションアクセスの権限を設定できます。

設定ユーティリティを使用してアプリケーションの AAA ユーザと AAA ユーザグループを設定するには

1. [**AppExpert**] > [アプリケーション] に移動し、アプリケーションエンティティを選択して [編集] をクリックします。
2. [詳細設定] セクションで、[承認] をクリックし、承認されたユーザーとユーザーグループを構成します。
3. 許可されたユーザをアプリケーションにバインドするには、[**AAA** ユーザ] セクションをクリックします。
4. **AAA User** スライダで、パラメータを設定します。
5. [続行] をクリックし、[詳細設定] セクションの [承認ポリシー] をクリックします。
6. [承認ポリシー] スライダーで、承認ポリシーをアプリケーションにバインドします。
7. [続行] をクリックし、[詳細設定**]** セクションの **[**** 承認グループ]** セクションをクリックします。

8. [**AAA Group Binding**] スライダで、認可ユーザグループをアプリケーションにバインドします。
9. [続行] をクリックし、[詳細設定] セクションの [ポリシー] をクリックします。
10. ポリシースライダーで、監査 **Syslog** または **AuditNSLog** ポリシーをアプリケーションにバインドします。
11. [続行] をクリックし、[完了] をクリックします。

GUI を使用してアプリケーションの AAA ユーザと AAA ユーザグループを編集するには、次の手順を実行します。

AppExpert > アプリケーション > 詳細設定に移動し、認証をクリックします。次に、編集アイコンをクリックし、ユーザーまたはユーザーグループの承認設定の値を指定します。

GUI を使用して AAA ユーザと AAA ユーザグループを削除するには、次の手順を実行します。

[**AppExpert**] > [アプリケーション] に移動し、アプリケーションを選択して [編集] をクリックします。[アプリケーション] ページで、[** 詳細設定 **] をクリックし、[承認] をクリックします。エンティティの横にある削除アイコンをクリックします。

NetScaler アプリケーションの監視

August 15, 2023

AppExpert アプリケーションをカスタマイズしたら、アプリケーション統計を表示して、アプリケーションとそのすべてのエンティティが正しく動作していることを確認できます。アプリケーションビジュアライザーを使用して、ポリシーや仮想サーバーなどの特定のエンティティに関連する統計を監視することもできます。

また、さまざまなエンティティのヒットカウンタを定期的に表示して、カウンタが更新されていることを確認することもできます。

アプリケーション統計を表示する

「アプリケーション」ノードでは、アプリケーションを選択し、そのアプリケーションの「統計」ページを表示できます。[Statistics] ページでは、パブリックエンドポイントとアプリケーションユニットの状態と状態を監視し、次の統計情報を表示できます。

- 各パブリックエンドポイントとアプリケーションユニットの 1 秒あたりの要求と応答。
- 着信トラフィックと発信トラフィックの各エンドポイントでの 1 秒あたりのバイト数。
- アプリケーションユニットのヒットカウンタと、各アプリケーションユニットのクライアントおよびサーバ接続数。
- アプリケーションユニットにバインドされているサービスの統計。

[Statistics] ページでは、CPU 使用率、メモリ使用量、およびシステムログも表示できます。

アプリケーションの統計を表示するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、統計を表示するアプリケーションをクリックし、[統計] をクリックします。

アプリケーションビジュアライザーを使用したアプリケーションの監視

アプリケーションビジュアライザーを使用して、vserver が特定の時点で 1 秒あたりに受信する要求の数と、書き換え、レスポンダー、およびキャッシュの各ポリシーについて、特定の時点における 1 秒あたりのヒット数を監視できます。

ビジュアライザーで vserver、書き換えポリシー、レスポンダーポリシー、およびキャッシュポリシーの統計情報を表示するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、統計情報を表示するアプリケーションを選択し、[ビジュアライザー] をクリックします。
3. [アプリケーションビジュアライザー] ウィンドウで、次の操作を行います。

    - 統計を表示するには、[統計を表示] をクリックします。

      統計情報は、ビジュアライザーの各ノードに表示されます。この情報はリアルタイムでは更新されないため、手動で更新する必要があります。
    - 統計情報を更新するには、[統計情報の更新] をクリックします。

ヒットを表示する

さまざまな AppExpert アプリケーションエンティティに提供されるヒットカウンターにより、パブリックエンドポイントとアプリケーションユニットの機能を監視できます。アプリケーションの場合、[ヒット（Hits）] ダイアログボックスには、構成された各パブリックエンドポイントによって受信されたリクエストの総数が表示されます。アプリケーションユニットの場合、[ヒット] ダイアログボックスには、アプリケーションユニットが各パブリックエンドポイントから処理したリクエストの数と合計ヒット数が表示されます。ヒットカウンタの表示手順については、設定の確認とテストを参照してください。

アプリケーションを削除する

August 15, 2023

アプリケーションとそのアプリケーションユニットが不要になった場合は、削除できます。AppExpert アプリケーションを削除しても、バックエンドサービスは削除されず、そのアプリケーションが使用していたパブリックエンドポイントは、他のアプリケーションが使用できるようになります。

アプリケーションを削除する場合、他の場所で使用されていないバインドされたポリシーとアクションを削除するかどうかを指定するように求められます。

GUI を使用してアプリケーションのアプリケーションユニットを削除するには、次の手順を実行します。

[**AppExpert**] > [アプリケーション] に移動し、アプリケーションを選択して [編集] をクリックします。[アプリケーションユニット] セクションで、エンティティの横にある削除アイコンをクリックします。

アプリケーションの認証、承認、および監査を構成する

August 15, 2023

アプライアンスに設定するアプリケーションの認証、承認、監査（AAA）を設定できます。アプリケーションに対して構成された認証ポリシーは、ユーザーまたはグループがアプリケーションにアクセスしようとしたときに適用する認証の種類を定義します。外部認証を使用する場合、ポリシーは外部認証サーバも指定します。アプリケーションに設定された承認ポリシーは、特定のユーザーまたはグループがアプリケーションにアクセスできるかどうかを指定します。監査ポリシーは、監査ログの種類、ログ記録が実行されるレベル、およびその他の監査サーバー設定を定義します。認証ポリシーと監査ポリシーは、従来のポリシー形式を使用します。

認証ポリシー、承認ポリシー、および監査ポリシーは、任意の順序で構成できます。ただし、アプリケーションに AAA を設定する前に、アプリケーションのパブリックエンドポイントを設定する必要があります。

アプリケーションの認証を構成するには、認証 FQDN、認証仮想サーバー、サーバー証明書、および認証ポリシーとセッションポリシーを指定します。認証ポリシーは、アプリケーションに対して指定された認証仮想サーバーに自動的にバインドされます。

AppExpert アプリケーションの認証を設定するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
    a) [追加] をクリックして、新しいアプリケーションの認証を追加します。
    b) 既存のアプリケーションを変更するには、[編集] をクリックします。
3. [アプリケーション] ページで、アプリケーションユニットを選択します。
4. [アプリケーションユニット] スライダページで、[詳細設定] セクションから [認証] をクリックします。
5. [認証] セクションで、次のように認証タイプを選択します。
    a) フォームベース認証
    b) 401 ベースの認証
    c) なし
6.「**OK**」をクリックし、「完了」をクリックします。

アプリケーション認証を構成する

ユーザーとグループの承認を構成して、AppExpert アプリケーションへのアクセスを有効にすることができます。権限を設定する AAA ユーザまたはグループがまだ作成されていない場合は、AppExpert から作成し、アプリケーションアクセスの権限を設定できます。

AAA ユーザまたはグループが AppExpert アプリケーションにアクセスするための権限を設定するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、ユーザーまたはグループのアクセスを構成する AppExpert アプリケーションをクリックします。
3. [アプリケーション] ページで、[詳細設定] セクションから [承認] をクリックします。
4. 次のいずれかを行います:
   - 権限を設定する AAA ユーザまたはグループがすでに [グループ/ユーザ] ツリーにある場合は、ユーザまたはグループを [グループ/ユーザ] ツリーからアプリケーションツリーの [ユーザ] ノードまたは [グループ] ノードにドラッグします。次に、ユーザーまたはグループを右クリックして、[許可] をクリックします。
   - 権限を設定する AAA ユーザまたはグループがアプライアンスで設定されていない場合は、アプリケーションツリーで [ユーザ] または [グループ] を右クリックし、[追加] をクリックします。[AAA グループの作成] または [AAA ユーザの作成] ダイアログボックスで、値を入力して [作成] をクリックし、[閉じる] をクリックします。

     ユーザーまたはグループは、許可に設定された権限で作成されます。権限設定を変更するには、グループまたはユーザーを右クリックし、権限設定をクリックします。
5. [完了] をクリックし、[閉じる] をクリックします。

アプリケーション監査を構成する

アプリケーションの監査ポリシーを構成するときは、ログメッセージの送信先となるサーバー、ログに記録されるメッセージの形式、およびログレベルを指定する必要があります。オプションで、ログ機能や日付形式など、他の設定を構成できます。監査ポリシーは、すべての AppExpert アプリケーションのパブリックエンドポイントに自動的にバインドされます。

アプリケーションの監査ポリシーを構成するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、監査ポリシーを構成するアプリケーションをクリックします。
3. [アプリケーションユニット] スライダページで、[ポリシー] セクションの [+] アイコンをクリックして、監査ポリシーを設定します。
4. [ポリシー] スライダページで、ポリシータイプとして [Syslog 監査] または [NSLOG 監査] を選択し、[続行] をクリックします。

5. [ポリシーバインディング] セクションで、次のパラメーターを設定します。

   a) バインドするポリシーを選択します。バインドするポリシーがない場合は、[+] をクリックして新しいポリシーを作成します。

   b) 新しい監査ポリシーを作成するには、[ポリシー名] で [新しいポリシー**]** をクリックし、 [ポリシー] ページで次の操作を行います。

      i. [名前] ボックスに、ポリシーの名前を入力します。
      ii. [名前] ボックスには、サーバー名の先頭に必要な文字列が既に含まれています。文字列は変更できません。
      iii. [監査タイプ] リストから、監査タイプ (SYSLOG または NSLOG) を選択します。
      iv. 指定する監査サーバーが既に [サーバー] ボックスの一覧に表示されている場合は、一覧からサーバーを選択し、サーバー設定を変更する場合は [変更] をクリックします。[監査サーバーの構成] ダイアログボックスで、必要に応じて設定を変更し、[OK] をクリックします。[監査サーバーの構成] ダイアログボックスの設定の詳細については、「認証されたセッションの監査」を参照してください。
      v. 新しい監査サーバーを構成する場合は、[新規] をクリックし、[監査サーバーの作成] ダイアログボックスでサーバーの名前を入力し、サーバーの IP アドレス、ポート番号、およびその他の設定を必要に応じて指定します。終了したら、「**OK**」をクリックします。
      vi. [作成] をクリックします。

   c) 作成した新しい監査ポリシーの優先順位を変更するには、[優先度] で、優先度を変更するポリシーごとに、優先順位の値をダブルクリックし、新しい優先度の値を入力します。

   d) 優先度を再生成するには、[優先度を再生成する] をクリックします。

   e) ポリシーのバインドを解除するには、ポリシーをクリックし、[ポリシーのバインド解除] をクリックします。

   f) ポリシーを変更するには、ポリシーをクリックし、次に [ポリシーの変更**]** をクリックします。

6. [変更を適用**]** をクリックし、 [閉じる] をクリックします。

アプリケーションの **AAA** の無効化

アプリケーションに AAA を設定したら、そのアプリケーションの AAA 設定を無効にできます。アプリケーションの AAA を無効にしても、設定は失われません。設定を再適用する場合は、アプリケーションの AAA を有効にできます。

アプリケーションの AAA を有効または無効にするには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、AAA を有効または無効にするアプリケーションをクリックし、次のいずれかを実行します。
3. アプリケーションの AAA を無効にするには、[**AAA** をオフにする**]** をクリックします。
4. アプリケーションで AAA を有効にするには、[**AAA** をオンにする] をクリックします。

カスタム **NetScaler** アプリケーションのセットアップ

August 15, 2023

NetScaler アプライアンスで管理する Web アプリケーションに AppExpert アプリケーションテンプレートがない場合、または利用可能な AppExpert アプリケーションテンプレートが要件に合わない場合は、テンプレートなしで AppExpert アプリケーションを作成できます。

テンプレートなしで AppExpert アプリケーションを作成するには、まずアプリケーションとアプリケーションユニットを作成する必要があります。次に、パブリックエンドポイント、サービス、およびサービスグループを設定します。最後に、アプリケーショントラフィックの評価方法と処理方法を決定するポリシーを設定します。

アプリケーションユニットとアプリケーションユニットを作成し、ポリシーを構成したら、事前に構築された AppExpert アプリケーションテンプレートを使用してアプリケーションを構成する場合と同様に、構成を検証して正しく動作することをテストする必要があります。次に、アプリケーションを監視して、アプリケーションとそのエンティティが正しく動作していることを確認する必要があります。

アプリケーションを作成する

AppExpert アプリケーションを作成すると、アプリケーションユニットを追加できるコンテナがアプライアンスによって作成されます。デフォルトのアプリケーションユニットは、最初のアプリケーションユニットを作成するまで作成されません。

GUI を使用して AppExpert アプリケーションを作成するには

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、[アプリケーション] を右クリックし、[追加] をクリックします。
3.「アプリケーションの作成」ダイアログ・ボックスの「名前」にアプリケーションの名前を入力し、「**OK**」をクリックします。

アプリケーションユニットの作成

Web アプリケーションに関連するトラフィックのサブセットごとに、アプリケーションユニットを作成する必要があります。

GUI を使用して AppExpert アプリケーションのアプリケーションユニットを作成するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、アプリケーションユニットを追加するアプリケーションを右クリックし、[追加] をクリックします。
3. ［作成］をクリックします。

AppExpert アプリケーションのパブリックエンドポイントの設定

必要なアプリケーションユニットをすべて作成したら、クライアントが NetScaler アプライアンスを介して Web アプリケーションにアクセスできるように、1 つ以上のパブリックエンドポイントを構成する必要があります。

GUI を使用して AppExpert アプリケーションのパブリックエンドポイントを設定するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、パブリックエンドポイントを構成するアプリケーションを右クリックし、[パブリックエンドポイントの構成] をクリックします。
3. アプリケーションの [パブリックエンドポイントの選択] ダイアログボックスで、次のいずれかを実行します。
   - 必要なエンドポイントがダイアログボックスに一覧表示されている場合は、対応するチェックボックスをオンにします。
   - すべてのパブリックエンドポイントを指定する場合は、[**Activate All**] をクリックします。
   - エンドポイントと AppExpert アプリケーションの関連付けを解除する場合は、対応するチェックボックスをオフにします。
   - 新しいパブリックエンドポイントを作成する場合は、[追加] をクリックします。次に、[パブリックエンドポイントの作成] ダイアログボックスでエンドポイント設定を構成し、[**OK**] をクリックします。

     [パブリックエンドポイントの作成] ダイアログボックスでは、エンドポイントの名前、IP アドレス、ポート、およびプロトコルのみを指定できます。パブリックエンドポイントを作成した後で、追加のエンドポイント設定を指定できます。追加のエンドポイント設定を指定するには、エンドポイントを作成した後、[パブリックエンドポイントの選択] ダイアログボックスでエンドポイントをクリックし、[開く] をクリックします。次に、[パブリックエンドポイントの構成] ダイアログボックスで追加の設定を指定し、[**OK**] をクリックします。

     [パブリックエンドポイントの作成**]** および **[**** パブリックエンドポイントの構成 **] ダイアログボックスのパラメータの詳細については、「コンテンツの切り替え」を参照してください。
   - パブリックエンドポイントを変更する場合は、エンドポイントをクリックし、[開く] をクリックします。次に、[パブリックエンドポイントの構成] ダイアログボックスで、エンドポイントの設定を変更し、[**OK**] をクリックします。

     パブリックエンドポイントの設定ダイアログボックスのパラメータの詳細については、「コンテンツスイッチング」を参照してください。
4. [閉じる] をクリックします。

アプリケーションユニットのパブリックエンドポイントの設定

アプリケーションユニットの場合は、AppExpert アプリケーションテンプレートから作成されたアプリケーションのパブリックエンドポイントを指定するのと同じ方法でパブリックエンドポイントを指定します。アプリケーション

ユニットのエンドポイントのサブセットを指定する方法の詳細については、「アプリケーションユニットのエンドポイントの設定」を参照してください。

GUI を使用してアプリケーションユニットのエンドポイントを設定するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、パブリックエンドポイントを指定するアプリケーションユニットを右クリックし、[パブリックエンドポイントの構成] をクリックします。
3. アプリケーションユニットの [パブリックエンドポイントの選択] ダイアログボックスで、次のいずれかを実行します。
   - アプリケーションユニットのエンドポイントを初めて指定する場合は、アプリケーションユニットにバインドしないエンドポイントに対応するチェックボックスをオフにします。
   - ダイアログボックスに一覧表示されているが、現在はアプリケーションユニットにバインドされていないエンドポイントを指定する場合は、対応するチェックボックスをオンにします。
4. [**OK**] をクリックします。

AppExpert アプリケーションのサービスとサービスグループの設定

サービスとサービスグループは、AppExpert アプリケーションのサービスとサービスグループを構成した後にのみ、アプリケーションユニットで使用可能になります。したがって、アプリケーションユニットのサービスを構成する前に、AppExpert アプリケーションのサービスとサービスグループを構成する必要があります。AppExpert アプリケーション用に構成するすべてのサービスおよびサービスグループは、同じプロトコル (HTTP または HTTPS) を使用する必要があります。テンプレートから作成されない AppExpert アプリケーションのサービスとサービスグループを構成する手順は、テンプレートから作成されたアプリケーションの手順と同じです。

GUI を使用して AppExpert アプリケーションのサービスまたはサービスグループを構成するには、次の手順を実行します。

1. [**AppExpert**] > [アプリケーション] に移動します。
2. 詳細ウィンドウで、サービスまたはサービスグループを構成するアプリケーションを右クリックし、[バックエンドサービスの構成] をクリックします。
3. [バックエンドサービスの設定] ダイアログボックスで、次のいずれかを実行します。
   - サービスを構成するには、[サービス] タブをクリックします。
   - サービスグループを設定するには、**** サービスグループタブをクリックします ** 。
4. [サービス] タブまたは [サービスグループ] タブで、次のいずれかを実行します。
   - 必要なサービスまたはサービスグループがタブに表示されている場合は、対応するチェックボックスをオンにします。
   - すべてのサービスまたはサービスグループを指定する場合は、[すべてアクティブ化] をクリックします。

- 新しいサービスまたはサービスグループを作成する場合は、[追加] をクリックします。次に、[サービスの作成] ダイアログボックスまたは [サービスグループの作成] ダイアログボックスで、サービスまたはサービスグループの設定をそれぞれ構成し、[作成] をクリックします。
- サービスを変更する場合は、サービスをクリックし、[開く] をクリックします。次に、[サービスの構成] ダイアログボックスまたは [サービスグループの作成] ダイアログボックスで、サービスまたはサービスグループの設定をそれぞれ構成し、[**OK**] をクリックします。

[サービスの作成]、[サービスの構成]、および [サービスグループの作成] の各ダイアログボックスの設定については、「負荷分散」を参照してください。

アプリケーションユニットのサービスおよびサービスグループの設定

サービスとサービスグループを設定したら、アプリケーションユニットごとにサービスとサービスグループを設定する必要があります。ただし、各バックエンドサービスが Web アプリケーションに関連するすべてのコンテンツをホストしている場合は、この手順は必要ありません。アプリケーションユニットに関連付けられたコンテンツがバックエンドサーバーのサブセットでのみホストされている場合は、アプリケーションユニットのサービスとサービスグループを構成します。

GUI を使用してアプリケーションユニットのサービスまたはサービスグループを設定するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、サービスまたはサービスグループを構成するアプリケーションユニットを右クリックし、[バックエンドサービスの構成] をクリックします。
3. [バックエンドサービスの設定] ダイアログボックスで、次のいずれかを実行します。
   - サービスを構成するには、[サービス] タブをクリックします。
   - サービスグループを設定するには、サービスグループタブをクリックします。
4. サービス または サービスグループ] タブで、次のいずれかの操作を行います。
   - アプリケーションユニットに設定しないサービスまたはサービスグループに対応するチェックボックスをオフにします。アプリケーションユニットに設定するサービスまたはサービスグループに対応するチェックボックスがオンになっていることを確認します。次に、[Weight] 列で、構成済みの各サービスに割り当てる重みを指定します。
   - すべてのサービスまたはサービスグループを指定するには、[すべてアクティブ化] をクリックします。
5. [メソッド] タブと [持続性] タブと [詳細] タブで、目的のパラメータを指定します。
6. [**OK**] をクリックします。

ポリシーの設定

テンプレートを使用せずに作成された AppExpert アプリケーションのポリシーを構成する手順は、テンプレートから作成された AppExpert アプリケーションの手順と同じです。詳細については、「アプリケーションユニットのポリ

シーの設定」を参照してください。

NetScaler Gateway アプリケーション

August 15, 2023

Citrix® NetScaler® アプライアンスを介して Web アプリケーションを管理するように AppExpert アプリケーションを構成する場合、アプリケーションユニットのセットを作成し、各ユニットのトラフィック最適化とセキュリティポリシーのセットを構成します。各アプリケーションユニットに設定するポリシー（圧縮、キャッシング、書き換えなどの機能のポリシー）は、そのユニット専用のトラフィックを評価します。これらのポリシーに加えて、アプリケーション全体の Access Gateway ポリシーを構成して、Access Gateway 経由でアクセスする場合のアプリケーショントラフィックを最適化することもできます。Access Gateway アプリケーション機能を使用すると、AppExpert アプリケーションの Access Gateway ポリシー (承認、トラフィック、クライアントレスアクセス、および TCP 圧縮) を構成できます。AppExpert アプリケーションの NetScaler Gateway ポリシーを構成したら、作成した AppExpert アプリケーションテンプレートにポリシー構成を含めることができます。

イントラネットサブネット、ファイル共有、およびその他のネットワークリソースに対して NetScaler Gateway ポリシーを構成することもできます。最後に、ユーザーが NetScaler Gateway のホームページからアクセスできるようにしたい場合は、AppExpert アプリケーションと特定のリソースのブックマークを作成できます。

NetScaler Gateway アプリケーション機能のエンティティは、GUI のみを使用して構成できます。

NetScaler Gateway アプリケーションの仕組み

GUI の [アプリケーション] ノードで AppExpert アプリケーションを作成すると、対応する Access Gateway アプリケーションが [Access Gateway アプリケーション] ノードに自動的に作成されます。さらに、AppExpert アプリケーションの構成済みパブリックエンドポイントを使用するルールが、Access Gateway アプリケーションエントリに対して自動的に作成されます。AppExpert アプリケーションに複数のエンドポイントが構成されている場合、ルールには構成されているすべてのパブリックエンドポイントが含まれます。NetScaler アプライアンスはこのルールを使用して、構成済みの Access Gateway ポリシーを AppExpert アプリケーションのパブリックエンドポイントで受信したトラフィックに適用します。AppExpert アプリケーションのパブリックエンドポイントで受信したトラフィックは、最初に NetScaler Gateway ポリシーと照らし合わせて評価され、次に AppExpert アプリケーションのアプリケーションユニットに設定されたポリシーと照らし合わせて評価されます。

Access Gateway アプリケーションのクライアントレスアクセスポリシー用に作成される規則は、AppExpert アプリケーション用に構成されたパブリックエンドポイントも使用する高度な式です。そのため、AppExpert アプリケーションの NetScaler Gateway ポリシーを構成する前に、AppExpert アプリケーションのパブリックエンドポイントを構成する必要があります。

NetScaler Gateway 構成をアプリケーションテンプレートに含める場合、IP アドレスやポート情報などの展開固有の情報と、この情報から作成されたルールはテンプレートに含まれません。

ファイル共有の **NetScaler** 構成の仕組み

NetScaler アプライアンスでは、組織のネットワークでホストされているファイル共有の認証ポリシーを構成できます。

ファイル共有を作成するときは、ファイル共有の名前とファイル共有へのネットワークパスを指定します。ネットワークパスでは、サーバーの名前またはサーバーの IP アドレスのいずれかを指定できます。ファイル共有パスのコンポーネントを使用するルールが、ファイル共有に対して自動的に作成されます。このルールにより、アプライアンスはファイル共有サーバーでホストされているファイルの要求を識別できます。ファイル共有に対して構成されている承認ポリシーは、受信要求に適用されます。

ファイル共有の NetScaler 構成は、AppExpert アプリケーションテンプレートに保存できません。

イントラネットサブネットの **NetScaler** 構成の仕組み

ネットワークの一部を形成するイントラネットサブネットについては、NetScaler アプライアンスで承認、トラフィック、TCP 圧縮のポリシーを構成できます。イントラネットサブネットを追加するときは、イントラネットサブネットの IP アドレスとネットマスクを指定します。これら 2 つのパラメーターを使用するルールは、イントラネットサブネットに対して自動的に作成されます。アプライアンスは、宛先 IP アドレスとネットマスクがそれぞれサブネットの IP アドレスとネットマスクに設定されている要求に、構成済みのポリシーを適用します。

イントラネットサブネットの NetScaler 構成は、AppExpert アプリケーションテンプレートに保存できません。

他のリソースカテゴリの仕組み

[その他のリソース] カテゴリでは、任意のルールを使用して、任意のネットワークリソースの Access Gateway ポリシーを構成できます。ネットワークリソースの要求を処理するように NetScaler アプライアンスを構成する場合、ネットワークリソースに関連する要求を識別するクラシック表現を構成します。[その他のリソース] で、ネットワークリソースの承認、トラフィック、クライアントレスアクセス、および TCP 圧縮ポリシーを設定できます。NetScaler アプライアンスは、構成されたルールに一致するすべてのリクエストに、構成済みの NetScaler Gateway ポリシーを適用します。

「その他のリソース」にあるネットワークリソースの NetScaler 構成は、AppExpert アプリケーションテンプレートに保存できません。

エンティティの命名規則

NetScaler Gateway アプリケーション機能では、この機能で作成する一部のエンティティに命名規則が適用されます。たとえば、イントラネットサブネットのトラフィックポリシー用に作成するプロファイルの名前は、常にイントラネットサブネットの名前とそれに続くアンダースコア (_) で構成される文字列で始まります。エンティティに指定した名前が、この文字列に追加されます。サブネットの名前が「subnet1」の場合、プロファイルの名前は「subnet1_」

で始まります。このような命名規則が必要な場合 (エンティティの名前を入力するテキストボックスなど)、ユーザーインタフェースは、エンティティ名の開始に使用する文字列を自動的に挿入し、変更することはできません。

イントラネットサブネットの追加

August 15, 2023

ネットワークに構成されているイントラネットサブネットにバインドされるトラフィックの承認ポリシーとトラフィックポリシーを指定できます。これらのポリシーのルールは、サブネットに指定したパラメータを使用して自動的に作成されます。

GUI を使用してイントラネットサブネットを構成するには、次の手順を実行します。

1. GUI のナビゲーションペインで「**AppExpert**」を展開し、「Access Gateway アプリケーション」をクリックします。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - イントラネットサブネットを追加するには、[イントラネットサブネット] をクリックし、[追加] をクリックします。
   - イントラネットサブネットを変更するには、イントラネットサブネットをクリックし、[開く] をクリックします。
3. [イントラネットサブネットの作成] または [イントラネットサブネットの構成] ダイアログボックスで、次の操作を行います。
   a) [名前] ボックスに、追加するイントラネットサブネットの名前を入力します。このパラメーターは、既存のイントラネットサブネットでは変更できません。
   b) [IP アドレス] ボックスに、イントラネットサブネットの IP アドレスを入力します。
   c) [ネットマスク] ボックスに、イントラネットサブネットに使用されるネットマスクを入力します。
   d)「作成」または「**OK**」をクリックし、「閉じる」をクリックします。

他のリソースを追加する

August 15, 2023

[その他のリソース] に追加するネットワークリソースについては、リソースに関連付けられたトラフィックのサブセットを識別する高度なポリシー式を構成する必要があります。

GUI を使用して他のリソースのリソースを構成するには、次の手順を実行します。

1. GUI のナビゲーションペインで「AppExpert」を展開し、「**Access Gateway** アプリケーション」をクリックします。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - リソースを追加するには、[その他のリソース] をクリックし、[追加] をクリックします。
   - リソースを変更するには、リソースをクリックし、[開く] をクリックします。
3. [リソースの作成] または [リソースの構成] ダイアログボックスで、次の操作を行います。
   a) [名前] ボックスに、追加するリソースの名前を入力します。このパラメータは、既存のリソースでは変更できません。
   b) [ルール] ボックスに、追加するリソースに関連付けられているトラフィックのサブセットを識別するルールを入力します。
   または、[構成] をクリックし、[式の作成] ダイアログボックスでルールを作成します。
   c)「作成」または「**OK**」をクリックし、「閉じる」をクリックします。

承認ポリシーの設定

August 15, 2023

AAA ユーザーおよびグループがリソースにアクセスするための NetScaler Gateway 承認ポリシーを構成できます。

GUI を使用して AAA ユーザまたはグループがリソースにアクセスする権限を設定するには、次の手順を実行します。

1. GUI のナビゲーションペインで「AppExpert」を展開し、「**Access Gateway** アプリケーション」をクリックします。
2. 詳細ペインの [Authorization] 列で、AAA ユーザおよびグループの認可ポリシーを設定するアプリケーション、ファイル共有、イントラネットサブネット、またはリソースのアイコンをクリックします。
3. 次のいずれかを行います:
   - 権限を設定する AAA ユーザまたはグループがすでに [グループ/ユーザ] ツリーにある場合は、ユーザまたはグループを [グループ/ユーザ] ツリーから<application name>ツリーの [ユーザ] ノードまたは [グループ] ノードにドラッグします。次に、ユーザーまたはグループを右クリックして、[許可] をクリックします。
   - 権限を設定する AAA ユーザまたはグループがアプライアンスで設定されていない場合は、<application name> ツリーで [ユーザ] または [グループ] を右クリックし、[追加] をクリックします。**[**AAA** グループの作成]** または **[AAA** ユーザの作成] ダイアログボックスで、値を入力して **[**作成] をクリックし、[閉じる] をクリックします。**

ユーザーまたはグループは、許可に設定された権限で作成されます。権限設定を変更するには、グループまたはユーザーを右クリックし、権限設定をクリックします。

4. [閉じる] をクリックします。

トラフィックポリシーの設定

February 15, 2024

NetScaler Gateway アプリケーションノードのリソースに設定するトラフィックポリシーは、アプリケーションへのクライアント接続を制御します。リソースのルールを設定する必要はありません。リソースの作成時に自動的に作成されるルール。要求プロファイルとトラフィックポリシーを関連付けるだけで済みます。トラフィックプロファイルでは、プロトコル、アプリケーションタイムアウト、ファイルタイプの関連付けなどのパラメータを指定します。

リソースのトラフィックポリシーを構成するには

1. GUI のナビゲーションペインで「AppExpert」を展開し、「Access Gateway アプリケーション」をクリックします。
2. 詳細ウィンドウの [トラフィック] 列で、トラフィックポリシーを構成するアプリケーション、ファイル共有、イントラネットサブネット、またはリソースのアイコンをクリックします。
3. [トラフィックポリシーの構成] ダイアログボックスで、次の操作を行います。
   - 既存のトラフィックポリシーを指定するには、[ポリシーの挿入（**Insert Policy**）] をクリックし、[ポリシー名（Policy Name）] 列でポリシーの名前をクリックします。
   - 新しいポリシーを構成するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列の [新しいポリシー] をクリックします。[トラフィックポリシーの作成] ダイアログボックスの [名前] ボックスで、アンダースコア (_) の後にポリシーの名前を入力します。次に、[要求プロファイル] で既存の要求プロファイルを選択するか、[新規] をクリックして新しい要求プロファイルを構成します。既存のプロファイルを選択し、[変更] をクリックしてプロファイルを変更することもできます。
     トラフィックポリシーまたはプロファイルの構成について詳しくは、「NetScaler Gateway」を参照してください。
   - 挿入したポリシーを変更するには、「ポリシー名」列でポリシー名をクリックし、「ポリシーの変更」をクリックします。関連付けられているプロファイルのみを変更するには、[プロファイル] 列でプロファイルの名前をクリックし、[プロファイルの変更] をクリックします。
   - ポリシーに割り当てられた優先度を再生成するには、[** 優先度の再生成 **] をクリックします。
   - ポリシーの新しいプライオリティ値を指定するには、[Priority] 列で、割り当てられているプライオリティをダブルクリックし、必要な値を入力します。
   - ポリシーのバインドを解除するには、ポリシーをクリックし、[ポリシーのバインド解除] をクリックします。
4. [変更を適用] をクリックし、[閉じる] をクリックします。

クライアントレスアクセスポリシーの設定

February 15, 2024

NetScaler アプライアンスのリソースに対してクライアントレスアクセスを構成すると、エンドユーザーは NetScaler Gateway クライアントソフトウェアを使用せずにリソースにアクセスできます。ユーザーは、Web ブラウザを使用して Outlook Web Access などのリソースにアクセスできます。リソースのクライアントレスアクセスを設定するには、クライアントレスアクセスプロファイルに関連付けられたクライアントレスアクセスポリシーを設定します。

NetScaler Gateway アプリケーションノードのリソースにクライアントレスアクセスポリシーを構成するには:

1. GUI のナビゲーションペインで「**AppExpert**」を展開し、「**Access Gateway** アプリケーション」をクリックします。
2. 詳細ウィンドウの [クライアントレスアクセス] 列で、クライアントレスアクセスポリシーを構成するアプリケーション、ファイル共有、イントラネットサブネット、またはリソースのアイコンをクリックします。
3. [クライアントレスアクセスポリシーの設定] ダイアログボックスで、次の手順を実行します。

    - 既存のクライアントレスアクセスポリシーを指定するには、[ポリシーの挿入（**Insert Policy**）] をクリックし、[ポリシー名（Policy Name）] 列でポリシーの名前をクリックします。
    - 新しいクライアントレスアクセスポリシーを設定するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列で [新しいポリシー] をクリックします。[クライアントレスアクセスポリシーの作成] ダイアログボックスの [名前] ボックスで、アンダースコア (_) の後にポリシーの名前を入力します。次に、[プロファイル] で既存のプロファイルを選択するか、[新規] をクリックして新しいプロファイルを設定します。既存のプロファイルを選択し、[変更] をクリックしてプロファイルを変更することもできます。

      クライアントレスアクセスポリシーまたはプロファイルの構成の詳細については、「NetScaler Gateway」を参照してください。
    - 挿入したポリシーを変更するには、「ポリシー名」列でポリシー名をクリックし、「ポリシーの変更」をクリックします。関連付けられているプロファイルのみを変更するには、[プロファイル] 列でプロファイルの名前をクリックし、[プロファイルの変更] をクリックします。
    - ポリシーの新しいプライオリティ値を指定するには、[Priority] 列で、割り当てられているプライオリティをダブルクリックし、必要な値を入力します。
    - ポリシーのバインドを解除するには、ポリシーをクリックし、[ポリシーのバインド解除] をクリックします。

4. [変更を適用] をクリックし、[閉じる] をクリックします。

TCP 圧縮ポリシーの設定

February 15, 2024

アプリケーションの TCP 圧縮ポリシーを構成して、アプリケーションのパフォーマンスを向上させることができます。TCP 圧縮により、ネットワーク遅延が減少し、帯域幅要件が軽減され、伝送速度が向上します。TCP 圧縮ポリシーを設定するときは、圧縮アクションをポリシーに関連付けます。圧縮アクションでは、圧縮タイプとして [圧縮]、[GZIP]、[減圧]、または [圧縮なし] のいずれかを指定します。圧縮ポリシーと圧縮アクションの詳細については、「NetScalerGateway」を参照してください。

NetScaler Gateway アプリケーションノード内のリソースに TCP 圧縮ポリシーを構成するには

1. GUI のナビゲーションペインで「**AppExpert**」を展開し、「**Access Gateway** アプリケーション」をクリックします。
2. 詳細ウィンドウの [TCP 圧縮] 列で、TCP 圧縮ポリシーを構成するアプリケーション、ファイル共有、イントラネットサブネット、またはリソースのアイコンをクリックします。
3. [**TCP** 圧縮ポリシーの構成] ダイアログボックスで、次の操作を行います。

    - 既存の TCP 圧縮ポリシーを指定するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列でポリシーの名前をクリックします。
    - 新しい TCP 圧縮ポリシーを作成するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列の [新しいポリシー] をクリックします。[TCP 圧縮ポリシーの作成] ダイアログボックスの [ポリシー名] ボックスで、アンダースコア (「_」) の後にポリシーの名前を入力します。次に、[アクション] で既存のアクションを選択するか、[新規] をクリックして新しいアクションを設定します。[View] をクリックして、構成済みの圧縮タイプを表示することもできます。

      TCP 圧縮ポリシーまたはアクションの構成の詳細については、「NetScaler Gateway、NetScaler Gateway のアドバンスドエディション」を参照してください。
    - 挿入したポリシーを変更するには、「ポリシー名」列でポリシー名をクリックし、「ポリシーの変更」をクリックします。
    - ポリシーに割り当てられた優先度を再生成するには、[****** 優先度の再生成 ******] をクリックします。
    - ポリシーの新しいプライオリティ値を指定するには、[Priority] 列で、割り当てられているプライオリティをダブルクリックし、必要な値を入力します。
    - ポリシーのバインドを解除するには、ポリシーをクリックし、[ポリシーのバインド解除] をクリックします。

4. [変更を適用] をクリックし、 [閉じる] をクリックします。

ブックマークを設定する

February 15, 2024

資格のあるユーザーが利用できる内部アプリケーションまたはリソースのブックマークを設定できます。その後、ブックマークをユーザー、ユーザーグループ、または仮想サーバーにグローバルにバインドし、アクセスインターフェイスでそのユーザーに対して有効にすることができます。作成したブックマークリンクは、エンタープライズ Web サイトの下の Web サイトペインに表示されます。

詳細については、「Web リンクの作成と適用 」トピックを参照してください。

AppQoE

August 15, 2023

アプリケーションレベルのエクスペリエンス品質（AppQoE）は、NetScaler アプライアンスの複数の既存のポリシーベースのセキュリティ機能を 1 つの統合機能に統合し、新しいキューイングメカニズムであるフェアキューイングを活用します。均等化キューイングは、負荷分散された Web サーバーおよびアプリケーションへの要求をサービスレベルではなく仮想サーバーレベルで管理します。これにより、負荷分散後の個別のストリームとしてではなく、負荷分散の前にウェブサイトまたはアプリケーションへのすべての要求のキューイングを 1 つのグループとして処理できます。

- 単純な過負荷。どのサーバーでも、どのサーバーでも、一度に限られた数の接続しか受け付けられません。保護された Web サイトまたはアプリケーションが一度に受信する要求が多すぎる場合、サージ保護機能はオーバーロードを検出し、サーバーがそれらを受け入れることができるまで余分な接続をキューに入れます。AppQoE 機能は、要求したリソースが使用できないことをユーザーに通知する代替 Web ページを表示します。
- サービス拒否 **(DOS)** 攻撃。公開されているリソースは、そのサービスを停止させ、正当なユーザーによるアクセスを拒否することを目的とする攻撃に対して脆弱です。サージ保護機能は、他のタイプの高負荷に加えて、DOS 攻撃の管理に役立ちます。さらに、HTTP サービス拒否保護機能は、Web サイトに対する DOS 攻撃、疑わしい攻撃者にチャレンジを送信し、クライアントが適切な応答を送信しない場合に接続をドロップします。

NetScaler オペレーティングシステムの最新バージョンまでは、これらの機能はサービスレベルで実装されていました。つまり、各サービスに独自のキューが割り当てられていました。サービスレベルのキューは機能しますが、いくつかの欠点もあります。そのほとんどは、キューイングに依存する保護機能を実装する前に、NetScaler アプライアンスが要求の負荷分散を行う必要があるためです。キューイングの前に保護機能を実装すると、さまざまな利点があります。そのいくつかを以下に示します。

- サービスが状態遷移しても、接続はサービスレベルのキューにあるため、フラッシュされません。

- サービス拒否攻撃など、高負荷の期間中、負荷分散の前に HTTP DoS が実行されるため、ロードバランサが対処しなければならない前に、ロードバランサからの不要なトラフィックまたは優先度の低いトラフィックを検出して迂回できます。

AppQoE は、均等なキューイングの実装に加えて、共通の目標を達成するためにそれぞれ異なるツールセットを提供する一連の機能を統合しています。つまり、ネットワークリソースを過剰または不適切な需要から保護します。これらの機能を共通のフレームワークに組み込むと、より簡単に設定および実装できます。

AppQoE を有効にする

August 15, 2023

AppQoE を設定するには、まず機能を有効にする必要があります。

コマンドラインを使用して **AppQoE** を有効にするには

コマンドプロンプトで、次のコマンドを入力します。

- enable ns feature appqoe
- show ns feature

例:

```
> enable ns feature appqoe
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  ON
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 ...
 1)     AppQoE                          AppQoE              ON
 Done
<!--NeedCopy-->
```

GUI を使用して **AppQoE** を有効にするには

1. [システム] > [設定] に移動します。
2. 詳細ウィンドウで、「拡張機能の設定」をクリックします。
3.「拡張機能の設定」ダイアログで、「**AppQoE** 」チェックボックスを選択します。
4. [**OK**] をクリックします。

AppQoE アクション

August 15, 2023

AppQoE 機能を有効にした後、要求を処理するための 1 つ以上のアクションを設定する必要があります。

> 重要:
>
> アクションの作成には特定の個別のパラメーターは必要ありませんが、少なくとも 1 つのパラメーターを含める必要があります。含めないと、アクションを作成できません。

コマンドラインを使用して AppQoE アクションを設定するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appqoe action <name> [-priority <priority>] [-respondWith (ACS|NS)[<customfile>] [-altContentSvcName <string>] [-altContentPath <string>] [-maxConn <positive_integer>] [-delay <usecs>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-dosTrigExpression <expression>] [-dosAction (**SimpleResponse** | **HICResponse**)]`
- `show appqoe action`

例

中プライオリティキューと最低プライオリティキューにそれぞれ 10 と 1000 のポリシーキュー深度でプライオリティキューイングを設定するには:

```
> add appqoe action appqoe-act-basic-prhigh -priority HIGH
 Done

> add appqoe action appqoe-act-basic-prmedium -priority MEDIUM -
    polqDepth 10
 Done

> add appqoe action appqoe-act-basic-prlow -priority LOW -polqDepth
    1000
 Done

> show appqoe action

1.       Name: appqoe-act-basic-prhigh
        ActionType: PRIORITY_QUEUING
        Priority: HIGH
        PolicyQdepth: 0
        Qdepth: 0

```



```
1.       Name: appqoe-act-basic-prmedium
         ActionType: PRIORITY_QUEUING
         Priority: MEDIUM
         PolicyQdepth: 10
         Qdepth: 0

1.       Name: appqoe-act-basic-prlow
         ActionType: PRIORITY_QUEUING
         Priority: LOW
         PolicyQdepth: 1000
         Qdepth: 0
Done
<!--NeedCopy-->
```

コマンドラインを使用して既存の **AppQoE** アクションを変更するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appqoe action <name> [-priority <priority>] [-altContentSvcName <string>] [-altContentPath <string>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-maxConn <positive_integer>] [-delay <usecs>] [-dosTrigExpression <expression>] [-dosAction ( SimpleResponse | HICResponse )]`
- `show appqoe action`

コマンドラインを使用して **AppQoE** アクションを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `rm appqoe action <name>`
- `show appqoe action`

AppQoE アクションを設定するためのパラメーター

- 名を入力します。新しいアクションの名前、または変更する既存のアクションの名前。名前は、文字、数字、またはアンダースコア記号で始まり、1 から文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) で構成できます。
- 優先度。リクエストが割り当てられるプライオリティキュー。保護対象の Web サーバーまたはアプリケーションに負荷がかかり、追加のリクエストを受け入れることができない場合に、リソースが使用可能になったときに待機中のリクエストを処理する順序を指定します。選択肢は以下のとおりです。

1. 高い。リソースが使用可能になり次第、リクエストを処理します。
2. ミディアム。高優先度キュー内のすべての要求を満たした後で要求を処理します。

3. 低い。優先度が HIGH と MEDIUM のキューにあるすべてのリクエストを満たした後でリクエストを処理します。
4. 最低。優先度の高いキューにあるすべてのリクエストを処理した後にのみ、リクエストを処理します。

優先度が設定されていない場合、NetScaler アプライアンスはデフォルトで最も優先順位の低いキューに要求を割り当てます。

- で応答。指定されたしきい値に達したときに、指定されたレスポンダーアクションを実行するように NetScaler を構成します。次のいずれかの設定で使用する必要があります。
  - **ACS:** 代替コンテンツサービスのコンテンツを提供します。しきい値:MaxConn (最大接続数) または遅延。
  - **NS:** NetScaler からの組み込みレスポンスを提供します。しきい値:MaxConn (最大接続数) または遅延。
  - アクションなし**:** 代替コンテンツは配信されません。MaxConn (最大接続数) または遅延のしきい値に達した場合に、接続を最も優先順位の低いキューに割り当てます。
- 代替コンテンツ/VC 名。-ResponseWith ACS が指定されている場合、代替コンテンツサービスの名前。通常は代替コンテンツをホストする Web サーバへの絶対 URL です。

代替コンテンツパス。If -responseWith (ACS	代替コンテンツへのパスである NS) が指定されています。

-
- ロークデプス。このアクションに関連するポリシーキューのポリシーキュー深度のしきい値。このアクションに関連するポリシーキュー内の接続数が指定された数まで増えると、後続のリクエストは最も低いポリシーキューに割り当てられます。最小値:1 最大値:4,294,967,294
- プリクデプス。指定されたプライオリティキューのポリシーキュー深度のしきい値。現在のアクションに関連するポリシーがバインドされている仮想サーバー上の指定されたキュー内のリクエストの数が指定された数まで増えると、後続のリクエストは最も優先順位の低いキューに割り当てられます。最小値:1 最大値:4,294,967,294
- MaxConn。ポリシー・ルールに一致するリクエストに対して開くことができる接続の最大数。最小値:1 最大値:4,294,967,294
- ディレイ。ポリシー・ルールに一致するリクエストの遅延閾値（マイクロ秒単位）。一致するリクエストがしきい値より長く遅延している場合、
  NetScaler アプライアンスは指定されたアクションを実行します。アクションが指定されていない場合、アプライアンスは要求を最も優先順位の低いキューに割り当てます。最小値:1 最大値:599999,999
- 文字列表現。DoS アクションをトリガーするオプションの第 2 レベルのチェックを追加します。
- DS アクション。アプライアンスが、自身または保護対象サーバーが DoS 攻撃を受けていると判断した場合に実行するアクション。設定可能な値: シンプルレスポンス、HIC レスポンス。

これらの値は、HTTP-DDoS 攻撃を軽減するために受信リクエストの信頼性を検証するための HTTP チャレンジレスポンスメソッドを指定します。

HTTP チャレンジレスポンスの生成と検証プロセスでは、AppQoE は Cookie を使用してクライアントのレスポンスを検証し、クライアントが本物であることを確認します。チャレンジを送信すると、NetScaler アプライアンスは 2 つのクッキーを生成します。

ヘッダー Cookie (_DOSQ)。NetScaler アプライアンスが応答を確認できるように、クライアント固有の情報が含まれます。

ボディ Cookie (_DOSH)。クライアントマシンを検証するために使用される情報。クライアントのブラウザ (HIC の場合はユーザ) が、この Cookie の値を計算します。NetScaler アプライアンスは、その値を期待値と比較して、クライアントを検証します。

_DOSH 値を計算するためにアプライアンスがクライアントに送信する情報は、DoS アクション構成に基づいています。

1. シンプルなレスポンス：この場合、NetScaler アプライアンスは値を分割し、最終的な値を組み合わせる JavaScript コードを生成します。元の値を計算できるクライアントマシンは本物と見なされます。
2. HicResponse：この場合、NetScaler アプライアンスは 2 つの 1 桁の数字を生成し、それらの番号の画像を生成します。次に、バックパッチフレームワークを使用して、アプライアンスはそれらの画像を base64 文字列として挿入します。

制限事項

1. これは簡単な CAPTCHA の実装ではないので、この用語は使われていません。
2. 検証番号は、NetScaler が生成した数値に基づいており、120 秒間変化しません。この番号は動的またはクライアント固有でなければなりません。

構成ユーティリティを使用して **AppQoE** アクションを構成するには

1. **[**** アプリエキスパート**] > [**AppQoE****] > [アクション] に移動します。**
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいアクションを作成するには、[追加] をクリックします。
   - 既存のアクションを変更するには、アクションを選択し、[編集] をクリックします。
3. 「**AppQoE** アクションの作成」または「**AppQoE** アクションの設定」画面で、パラメータの値を入力または選択します。ダイアログボックスの内容は、「AppQoE アクションを構成するためのパラメーター」で説明されているパラメーターに次のように対応します (アスタリスクは必須パラメーターを示します)。
   - 名前—名前
   - アクションタイプ—次の式で応答

- 優先度—優先度
- ポリシーキューの深さ: POLQ の深さ
- キューの深さ—PRIQ の深さ
- DOS アクション—ディスコクション

4. [作成] または [**OK]** をクリックします。

AppQoE パラメータ

August 15, 2023

AppQoE パラメータでは、AppQoE セッションのセッション寿命、カスタマイズされた応答を含むファイルの名前、およびキューに入れることができるクライアント接続の数を設定します。

コマンドラインを使用して **AppQoE** パラメータ設定を構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appqoe parameter [-sessionLife <secs>] [-avgwaitingclient <positive_integer>] [-MaxAltRespBandWidth <positive_integer>] [-dosAttackThresh <positive_integer>]`
- `show appqoe parameter`

AppQoE パラメーターを構成するためのパラメーター

- sessionLife

  代替コンテンツを表示した後、アプライアンスが同じコンテンツを再度表示するまでに待機する秒数。デフォルト値:300 最大最小値:1 最大値:4,294,967,294

- avgwaitingclient

  サービス待機キューに入ることができるクライアント要求の平均数。デフォルト値:1000000 最大値:4,294,967,294

- MaxAltRespBandWidth

  代替応答の送信時に消費する最大帯域幅。最大数に達すると、アプライアンスは帯域幅消費量が減少するまで代替コンテンツの送信を停止します。デフォルト値:100 最小値:1 最大値:4,294,967,294

- dosAtckThrsh

  サービス拒否攻撃の閾値。アプライアンスが DoS 保護対策で応答するまでにキューで待機している必要がある接続の数。デフォルト値:2000 最小値:0 最大値:4,294,967,294

GUI を使用して AppQoE パラメータ設定を構成するには

1. 「**AppExpert** 」>「**AppQoE** 」に移動します。
2. 詳細ペインで、「**AppQoE** パラメーターの設定」をクリックします。
3. 「**AppQoE** パラメータの設定」画面で、パラメータの値を入力または選択します。ダイアログ・ボックスの内容は、「AppQoE パラメータを構成するためのパラメータ」で説明されているパラメータに次のように対応しています（アスタリスクは必須パラメータを示します）。

   - セッション寿命 (秒)
   - sessionLife
   - 平均待機クライアント: avgwaitingclient
   - 代替応答帯域幅制限 (Mbps) —代替応答の最大帯域幅
   - DOS 攻撃のしきい値—DoS 攻撃のスレッシュ

4. [**OK**] をクリックします。

AppQoE ポリシー

August 15, 2023

AppQoE を実装するには、特定のキューにキューに入れる接続を区別する方法を NetScaler に指示するポリシーを少なくとも 1 つ構成する必要があります。

コマンドラインを使用して AppQoE ポリシーを構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add appqoe policy <name> -rule <expression> -action <string>
```

例:

次の例では、「Android」を含む User-Agent ヘッダーのリクエストを選択し、中優先度のキューに割り当てます。これらのリクエストは、Google Android オペレーティングシステムを実行しているスマートフォンやタブレットから送信されます。

```
> add appqoe action appqoe-act-primd -priority MEDIUM
Done
> add appqoe policy appqoe-pol-primd -rule "HTTP.REQ.HEADER("User-Agent
    ").CONTAINS("Android")" -action appqoe-act-primd
Done
> sh appqoe policy appqoe-pol-primd
        Name: appqoe-pol-primd
        Rule: HTTP.REQ.HEADER("User-Agent").CONTAINS("Android")
        Action: appqoe-act-primd
```



```
 9          Hits: 0
10
11  Done
12  <!--NeedCopy-->
```

AppQoE ポリシーを設定するためのパラメータ

- 名を入力します。AppQoE ポリシーの名前。名前は、文字、数字、またはアンダースコア記号で始まり、1 ～ 127 文字の英数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) で構成できます。アクションの種類を識別しやすい名前を選択してください。
- 規則。アプライアンスに処理すべき接続を指定する NetScaler の式です。
- アクション。接続がポリシーに一致したときに実行される AppQoE アクション。

構成ユーティリティを使用して **AppQoE** ポリシーを構成するには

1. **[**** アプリエキスパート**] > [**AppQoE****] > [ポリシー] に移動します。****
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - ポリシーを作成するには、[**Add**] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[編集] をクリックします。
3. ポリシーを作成する場合は、「**AppQoE** ポリシーの作成」ダイアログの「名前」テキストボックスに、新しいポリシーの名前を入力します。

   名前は、文字、数字、またはアンダースコア記号で始まり、1 ～127 文字の英数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) で構成できます。このポリシーの目的と効果がわかりやすい名前を選択してください。

   既存のポリシーを変更する場合は、この手順をスキップしてください。既存のポリシーの名前は変更できません。
4. アクションドロップダウンリストで、ポリシーが接続と一致したときに実行する AppQoE アクションを選択します。プラス (+) をクリックして **AppQoE** アクションの追加ダイアログを開き、新しいアクションを追加します。
5. ルールテキストボックスに、ポリシー表現を直接入力するか、「新規」をクリックしてポリシー表現を作成します。「新規」をクリックした場合は、次の手順を実行します。

   a) [式の作成] ダイアログボックスで、[追加] をクリックします。
   1 エクスプレッションの追加ダイアログボックスで、「よく使うエクスプレッション」ドロップダウンリストから一般的なエクスプレッションを選択するか、「エクスプレッションの作成」ドロップダウンリストを使用して、フィルタリングするトラフィックを定義するエクスプレッションを作成します。
   独自の式を作成する場合は、まず「式を作成」領域の左側にある最初のドロップダウンリストから最初の用語を選択します。このリストの選択肢は以下のとおりです。

- HTTP
- システム
- クライアント
- サーバー
- 分析
- テキスト

デフォルトの選択は HTTP です。最初のドロップダウンリストで選択したら (またはデフォルトをそのまま使用して)、その右側のドロップダウンリストから式内の次の用語を選択できます。そのリストとそれに続く他のリストの用語は、以前に選択した内容に応じて変わります。リストには、有効な選択肢となる用語のみが表示されます。式が完成するまで用語の選択を続けてください。

a) 必要な式を作成したら、「**OK**」をクリックします。エクスプレッションがエクスプレッションテキストボックスに追加されます。

6. [作成] をクリックします。この式は「ルール」テキストボックスに表示されます。

負荷分散仮想サーバーのエンティティテンプレート

March 20, 2024

> 警告
>
> : エンティティテンプレート機能は、NetScaler 13.0 ビルド 82.x 以降から廃止され、代替としてスタイルブックを使用することをお勧めします。詳細については、「スタイルブック 」トピックを参照してください。

エンティティテンプレートは、NetScaler アプライアンスで負荷分散仮想サーバーテンプレートを作成するための情報の集まりです。負荷分散仮想サーバーに設定する仕様とデフォルトのセットを提供します。デフォルトのセットを定義するテンプレートを使用すると、いくつかの構成手順を省略しながら、同様の構成を必要とする複数の仮想サーバーをすばやく構成できます。

負荷分散仮想サーバーの詳細をテンプレートファイルにエクスポートすることで、エンティティテンプレートを作成できます。これは NetScaler GUI でのみ実行できます。NetScaler GUI を使用して、エンティティテンプレートのエクスポート、インポート、管理を行います。エンティティテンプレートを他の管理者と共有したり、アプライアンスまたはマシンにローカルに保存されたテンプレートを管理したりできます。アプライアンスまたはローカルコンピューターからエンティティテンプレートをインポートすることもできます。

テンプレートを作成する前に、負荷分散仮想サーバーの構成に精通している必要があります。

負荷分散仮想サーバーテンプレート

負荷分散エンティティテンプレートは、NetScaler アプリケーションテンプレートの作成と同じ方法で作成されます。負荷分散仮想サーバーをテンプレートファイルにエクスポートすると、次の 2 つのファイルが自動的に作成されます。

- 負荷分散仮想サーバーテンプレートファイル。負荷分散仮想サーバーに設定されたパラメーターの値を格納する XML 要素を含みます。このファイルには、バインドされたポリシーに関する情報を保存するための XML 要素も含まれています。
- デプロイファイル。サービス、サービスグループ、設定済み変数などのデプロイメント固有の情報を格納する XML 要素が含まれます。
  テンプレートファイルおよびデプロイファイルでは、構成情報の各ユニットは、そのユニットタイプ用の特定の XML 要素にカプセル化されています。たとえば、負荷分散方法のパラメーター LBMethod は、<lbmethod>および</lbmethod>タグ内にカプセル化されています。

> 注:
>
> 負荷分散仮想サーバーをエクスポートしたら、構成情報を NetScaler アプライアンスにインポートする前に、要素の追加、要素の削除、既存の要素の変更を行うことができます。

負荷分散仮想サーバーテンプレートの仕組み

負荷分散仮想サーバーのテンプレートを作成するときは、サーバーのデフォルト値を指定します。読み取り専用にする値、表示しない値、およびユーザーが構成できる値を指定します。また、テンプレートインポートウィザードを構成するページも設定します。入力したすべての情報と設定は、テンプレートファイルに保存されます。
ユーザーがテンプレートを NetScaler アプライアンスにインポートすると、テンプレート用に構成したさまざまなページが GUI によってユーザーに表示されます。GUI には読み取り専用のパラメータ値が表示され、ユーザに設定可能なパラメータの値を指定するよう求められます。ユーザーが指示に従うと、アプライアンスは設定値を使用してエンティティを作成します。
[トラフィック管理] ノードから、負荷分散仮想サーバーのエンティティテンプレートを作成または変更できます。
仮想サーバーの詳細をテンプレートにエクスポートするには、テンプレートの次のオプションと設定を指定する必要があります。

- パラメータのデフォルト値。
- デフォルト値がユーザーに表示されるかどうか。
- デフォルト値をユーザーが変更できるかどうか。
- ページ名、テキスト、使用可能なパラメータを含む、エンティティインポートウィザードのページ数。
- テンプレートを作成する対象のエンティティにバインドする必要があるエンティティ。

たとえば、負荷分散仮想サーバーテンプレートを作成するときに、テンプレートから作成した仮想サーバーにバインドするポリシーを指定できます。ただし、テンプレートにはバインディング情報のみが含まれます。バインドされたエンティティは含まれません。エンティティテンプレートを別の NetScaler アプライアンスにインポートする場合、

バインドが成功するには、バインドされたエンティティがインポート時にアプライアンスに存在している必要があります。バインドされたエンティティがターゲットアプライアンスに存在しない場合、（テンプレートが設定された）エンティティはバインディングなしで作成されます。バインドされたエンティティのサブセットのみがターゲットアプライアンスに存在する場合、それらはテンプレートから作成されたエンティティにバインドされます。
負荷分散仮想サーバーのテンプレートをエクスポートすると、エンティティの構成設定がテンプレートに表示されます。バインドされたエンティティはすべてデフォルトで選択されますが、必要に応じてバインディングを変更できます。既存のエンティティに基づいていないテンプレートの場合と同様に、バインディング情報のみが含まれており、エンティティは含まれません。テンプレートを既存の構成設定とともに保存することも、その設定をテンプレートの新しい構成を作成するための基礎として使用することもできます。

負荷分散仮想サーバーテンプレートでの変数の設定

負荷分散仮想サーバーテンプレートは、設定された負荷分散パラメーターおよびインバウンドポリシーとアクションの変数宣言をサポートします。変数を宣言する機能により、事前設定された値を、テンプレートをインポートする環境に適した値に置き換えることができます。

例として、テンプレートを作成する負荷分散仮想サーバーにバインドされたポリシーに設定された次の式を考えてみましょう。この式は、HTTP リクエストの受け入れ言語ヘッダーの値を評価します。

```
HTTP.REQ.HEADER("Accept-Language").CONTAINS("en-us")
```

インポート時にヘッダーの値を設定できるようにするには、文字列 en-us を変数として指定できます。

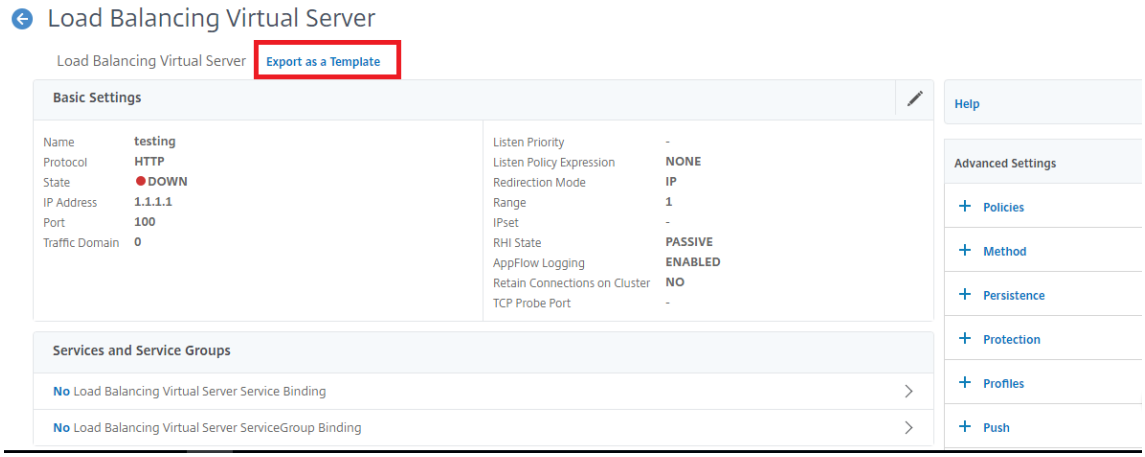
変数を作成したら、次の操作を実行できます。

- 既存の変数にもっと文字列を割り当てます。文字列の変数を作成した後、同じ式または異なる式の他の部分を選択し、変数に割り当てることができます。変数に割り当てる文字列は同じである必要はありません。インポート時に、変数に割り当てられたすべての文字列は、指定した値に置き換えられます。
- 変数に割り当てられている 1 つまたは複数の文字列を表示します。
- 変数を使用するすべてのエンティティとパラメータのリストを表示します

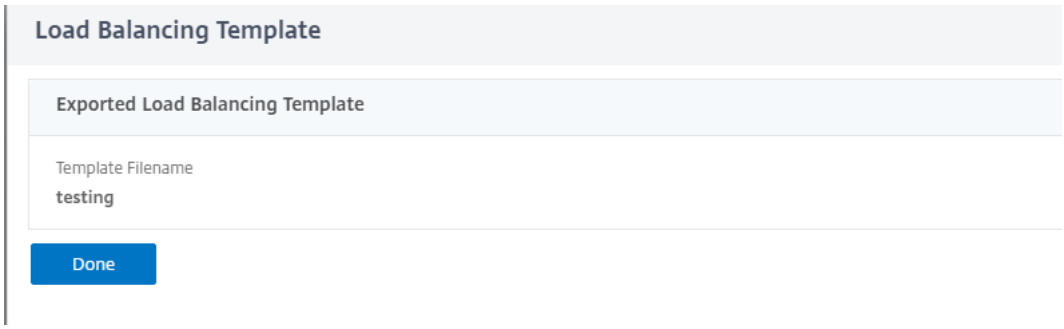
負荷分散仮想サーバーテンプレートで変数を設定するには

NetScaler GUI を使用して負荷分散仮想サーバーテンプレートの変数を構成するには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します
2. 詳細ウィンドウで、テンプレートファイルにエクスポートする仮想サーバーを右クリックし、[追加] をクリックします。
3. [負荷分散仮想サーバーの作成] ページで、仮想サーバーのパラメータを設定します。負荷分散仮想サーバーの構成の詳細については、「負荷分散の仕組み」を参照してください。
4. 負荷分散仮想サーバーのパラメータを設定したら、[**Done**] をクリックします。



5. サーバーの詳細をテンプレートファイルとしてエクスポートするには、上部の [テンプレートとしてエクスポート] リンクをクリックします。
6. 負荷分散テンプレートの作成ページで、テンプレート設定を入力します。
7. [完了] をクリックします。



負荷分散仮想サーバーテンプレートの変更

テンプレートに設定されているパラメーター、バインディング、ページのみを変更できます。テンプレートの作成時に指定したテンプレートの名前と場所は変更できません。NetScaler アプライアンスには、負荷分散仮想サーバーテンプレートを変更するオプションはありません。

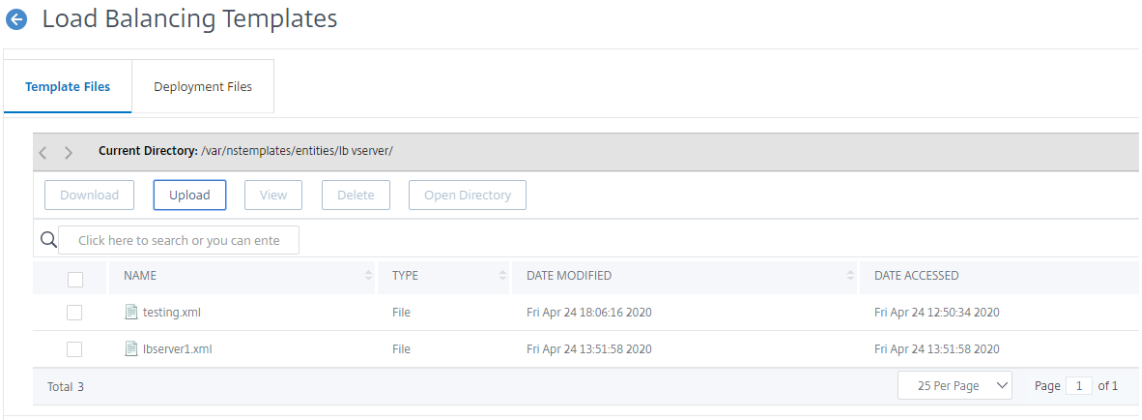
NetScaler GUI を使用して負荷分散仮想サーバーを変更するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 負荷分散仮想サーバーページで、エンティティパラメーターを変更します。
3. [完了] をクリックします。
4.「テンプレートとしてエクスポート」リンクをクリックします。
5. 変更された変更は、負荷分散仮想サーバーのテンプレートファイルで使用できるようになりました。
6.「エクスポートされた負荷分散テンプレート」ページで、「完了」をクリックします。

負荷分散仮想サーバーテンプレートの管理

NetScaler GUI を使用して、負荷分散仮想サーバーのテンプレートファイルと展開ファイルを整理できます。

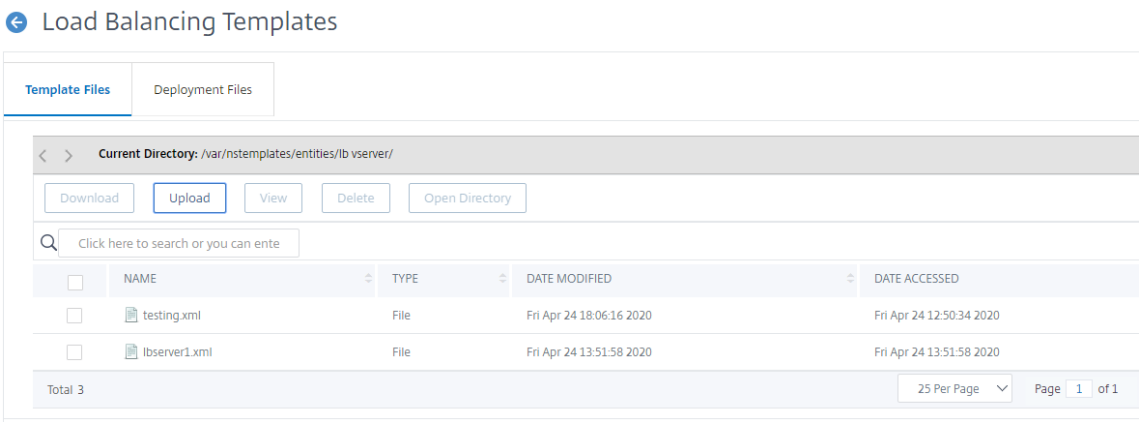
1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 「仮想サーバー」ページで、「テンプレートの管理」アクションを選択します。
3. 「負荷分散テンプレート」ページで、「テンプレートファイル」タブをクリックします。
4. テンプレートファイルタブページでは、アプライアンステンプレートフォルダからテンプレートをアップロードしたり、アプライアンステンプレートフォルダにテンプレートをダウンロードしたりできます。



5. [閉じる] をクリックします。

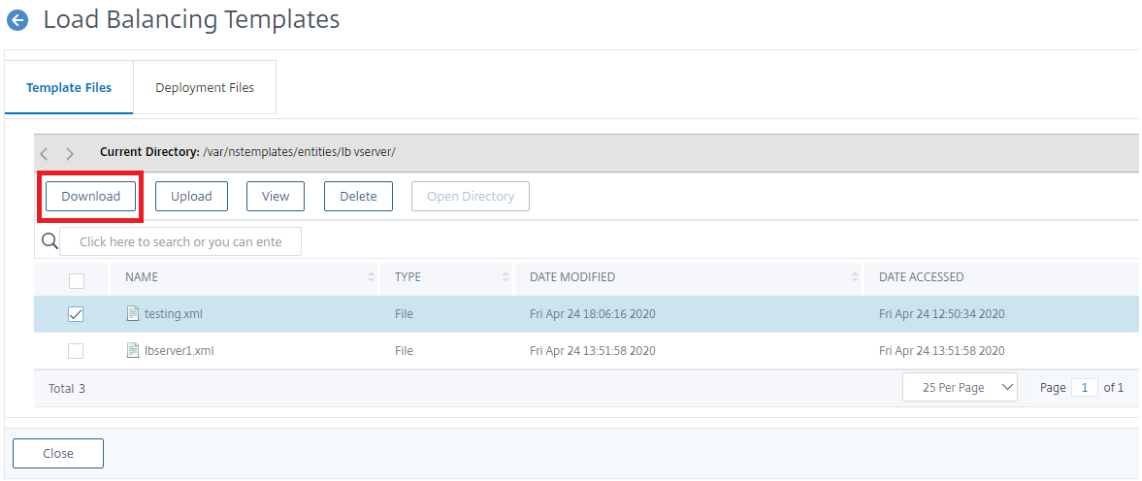
NetScaler GUI を使用して負荷分散仮想サーバーエンティティテンプレートをアップロードするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 「仮想サーバー」ページで、「アクションの選択」をクリックし、「テンプレートの管理」を選択します。
3. 「負荷分散テンプレート」ページで、「テンプレートファイル」タブをクリックします。
4. 「テンプレートファイル」タブページで、「アップロード」をクリックしてテンプレートをアップロードします。
5. [閉じる] をクリックします。



NetScaler GUI を使用して負荷分散仮想サーバーエンティティテンプレートをダウンロードするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2.「仮想サーバー」ページで、「アクションの選択」をクリックし、「テンプレートの管理」を選択します。
3.「負荷分散テンプレート」ページで、「テンプレートファイル」タブをクリックします。
4.「テンプレートファイル」タブページで、テンプレートファイルを選択して「ダウンロード」をクリックします。
5. [閉じる] をクリックします。



負荷分散仮想サーバーテンプレートと展開テンプレートの例

以下は、「Lbvip」という負荷分散仮想サーバーから作成されたテンプレートファイルの例です。

```
COPY

<?xml version="1.0" encoding="UTF-8" ?>
  <template>
    <template_info>
      <entity_name>Lbvip</entity_name>
      <version_major>10</version_major>
      <version_minor>0</version_minor>
      <build_number>40.406</build_number>
    </template_info>
    <entitytemplate>
      <lbvserver_list>
        <lbvserver>
          <name>Lbvip</name>
          <servicetype>HTTP</servicetype>
          <ipv46>0.0.0.0</ipv46>
          <ipmask>*</ipmask>
          <port>0</port>
          <range>1</range>
          <persistencetype>NONE</persistencetype>
```



```
            <timeout>2</timeout>
            <persistencebackup>NONE</persistencebackup>
            <backuppersistencetimeout>2</backuppersistencetimeout>
            <lbmethod>LEASTCONNECTION</lbmethod>
            <persistmask>255.255.255.255</persistmask>
            <v6persistmasklen>128</v6persistmasklen>
            <pq>OFF</pq>
            <sc>OFF</sc>
            <m>IP</m>
            <datalength>0</datalength>
            <dataoffset>0</dataoffset>
            <sessionless>DISABLED</sessionless>
            <state>ENABLED</state>
            <connfailover>DISABLED</connfailover>
            <clttimeout>180</clttimeout>
            <somethod>NONE</somethod>
            <sopersistence>DISABLED</sopersistence>
            <sopersistencetimeout>2</sopersistencetimeout>
            <redirectportrewrite>DISABLED</redirectportrewrite>
            <downstateflush>DISABLED</downstateflush>
            <gt2gb>DISABLED</gt2gb>
            <ipmapping>0.0.0.0</ipmapping>
            <disableprimaryondown>DISABLED</disableprimaryondown>
            <insertvserveripport>OFF</insertvserveripport>
            <authentication>OFF</authentication>
            <authn401>OFF</authn401>
            <push>DISABLED</push>
            <pushlabel>none</pushlabel>
            <l2conn>OFF</l2conn>
            <appflowlog>DISABLED</appflowlog>
            <icmpvsrresponse>PASSIVE</icmpvsrresponse>
            <lbvserver_cmppolicy_binding_list>
              <lbvserver_cmppolicy_binding>
                <name>Lbvip</name>
                <policyname>NOPOLICY-COMPRESSION</policyname>
                <priority>100</priority>
                <gotopriorityexpression>END</gotopriorityexpression>
                <bindpoint>REQUEST</bindpoint>
              </lbvserver_cmppolicy_binding>
            </lbvserver_cmppolicy_binding_list>
          </lbvserver>
        </lbvserver_list>
      </entitytemplate>
    </template>
<!--NeedCopy-->
```

デプロイファイルの例　前の例の仮想サーバーに関連付けられたデプロイファイルは次のとおりです。COPY

```
<?xml version="1.0" encoding="UTF-8" ?>
  <template_deployment>
    <template_info>
```



```
        <entity_name>Lbvip</entity_name>
        <version_major>10</version_major>
        <version_minor>0</version_minor>
        <build_number>40.406</build_number>
      </template_info>
      <service_list>
        <service>
        <ip>1.2.3.4</ip>
        <port>80</port>
        <servicetype>HTTP</servicetype>
        </service>
      </service_list>
      <servicegroup_list>
        <servicegroup>
          <name>svcgrp</name>
          <servicetype>HTTP</servicetype>
          <servicegroup_servicegroupmember_binding_list>
            <servicegroup_servicegroupmember_binding>
              <ip>1.2.3.90</ip>
              <port>80</port>
            </servicegroup_servicegroupmember_binding>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.8.0</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.8.1</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.9.0</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
        </servicegroup_servicegroupmember_binding_list>
      </servicegroup>
    </servicegroup_list>
</template_deployment>

<!--NeedCopy-->
```

HTTP コールアウト

August 15, 2023

特定のタイプの要求の場合、またはポリシー評価中に特定の基準が満たされた場合は、ポリシー評価を一時的に停止し、サーバから情報を取得し、取得した情報に応じて特定のアクションを実行することができます。また、特定の種類の要求を受け取ったときに、Web サーバーでホストされているデータベースまたはコンテンツを更新したい場合があります。HTTP コールアウトを使用すると、これらすべてのタスクを実行できます。

HTTP コールアウトは、ポリシー評価中に特定の基準が満たされたときに NetScaler アプライアンスが生成して外部アプリケーションに送信する HTTP または HTTPS リクエストです。サーバーから取得した情報は、高度なポリシー式で分析でき、適切なアクションを実行できます。HTTP コンテンツスイッチング、TCP コンテンツスイッチング、書き換え、レスポンダ、およびトークンベースの負荷分散方式の HTTP コールアウトを設定できます。

HTTP コールアウトを設定する前に、コールアウトの送信先となるサーバ上のアプリケーションを設定する必要があります。*HTTP* コールアウトエージェントと呼ばれるアプリケーションは、必要な情報で HTTP コールアウト要求に応答するように設定する必要があります。HTTP コールアウトエージェントは、NetScaler アプライアンスがコールアウトを送信するデータを提供する Web サーバーとしても使用できます。HTTP コールアウトへの応答の形式が、呼び出し間で変更されないようにする必要があります。

HTTP コールアウトエージェントをセットアップしたら、NetScaler アプライアンスで HTTP コールアウトを構成します。最後に、コールアウトを呼び出すには、適切な NetScaler 機能の詳細ポリシーにコールアウトを含めてから、ポリシーを評価したいバインドポイントにポリシーをバインドします。

HTTP コールアウトを設定したら、コールアウトが正しく動作していることを確認する設定を確認する必要があります。

HTTP コールアウトの仕組み

August 15, 2023

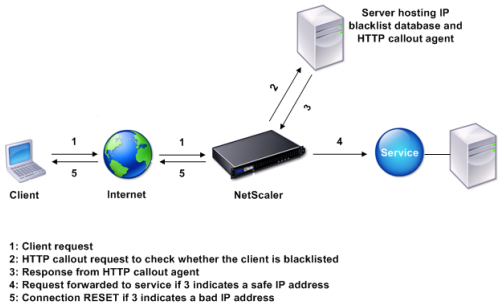
NetScaler アプライアンスがクライアント要求を受信すると、アプライアンスはその要求をさまざまなバインドポイントにバインドされたポリシーと照らし合わせて評価します。この評価中、アプライアンスは HTTP コールアウト表現`SYS.HTTP_CALLOUT(<name>)`を検出すると、ポリシー評価を一時的に停止し、指定した HTTP コールアウトに設定されたパラメータを使用して HTTP コールアウトエージェントに要求を送信します。応答を受信すると、アプライアンスは応答の指定された部分を検査し、HTTP コールアウトエージェントからの応答の評価がそれぞれ TRUE または FALSE のどちらに評価されるかに応じて、アクションを実行するか、次のポリシーを評価します。たとえば、HTTP コールアウトがレスポンダーポリシーに含まれている場合、応答の評価が TRUE と評価されると、アプライアンスはレスポンダーポリシーに関連付けられたアクションを実行します。

HTTP コールアウトの設定が正しくないか不完全な場合、またはコールアウトが再帰的に呼び出された場合、アプライアンスは UNDEF 条件を発生させ、未定義のヒットカウンタを更新します。

次の図は、グローバルにバインドされたレスポンダーポリシーから呼び出される HTTP コールアウトの動作を示しています。HTTP コールアウトは、受信リクエストに関連付けられているクライアントの IP アドレスを含むように設定されます。NetScaler アプライアンスがクライアントから要求を受信すると、アプライアンスはコールアウト要求を生成してコールアウトサーバーに送信します。コールアウトサーバーは、ブラックリストに登録された IP アドレスのデータベースと、クライアントの IP アドレスがデータベースにリストされているかどうかをチェックする HTTP コールアウトエージェントをホストします。HTTP コールアウトエージェントはコールアウトリクエストを受信し、クライアントの IP アドレスがリストされているかどうかを確認し、NetScaler アプライアンスが評価する応答を送信します。応答からクライアントの IP アドレスがブラックリストに登録されていないことが示された場合、アプライ

アンスは応答を設定されたサービスに転送します。クライアントの IP アドレスがブラックリストに登録されている場合、アプライアンスはクライアント接続をリセットします

図 1: HTTP コールアウトエンティティモデル



HTTP リクエストとレスポンスの形式に関する注記

August 15, 2023

NetScaler アプライアンスは、HTTP コールアウトリクエストの有効性をチェックしません。そのため、HTTP コールアウトを設定する前に、HTTP リクエストの形式を知っておく必要があります。HTTP コールアウトを設定するには、HTTP コールアウトエージェントからの応答を評価する式を設定する必要があるため、HTTP 応答の形式も知っておく必要があります。

このセクションには次のセクションが含まれます。

- HTTP リクエストのフォーマット
- HTTP レスポンスのフォーマット

HTTP リクエストのフォーマット

HTTP リクエストには一連の行が含まれており、各行末にはキャリッジリターンとラインフィード（<CR><LF> or \r\nどちらかで表されます）が付いています。

リクエストの最初の行 (メッセージ行) には、HTTP メソッドとターゲットが含まれます。たとえば、次の例に示すように、GET リクエストのメッセージ行には GET というキーワードと、取得するオブジェクトを表す文字列が含まれます。

```
1 GET /mysite/mydirectory/index.html HTTP/1.1\r\n
2 <!--NeedCopy-->
```

リクエストの残りの部分には、必須の Host ヘッダーと、該当する場合はメッセージ本文を含む HTTP ヘッダーが含まれます。

リクエストは銀行回線（追加 <CR><LF> or \r\n）で終わります。

リクエストの例を以下に示します。

```
Get /mysite/index.html HTTP/1.1\r\n
Host: 10.101.101.10\r\n
Accept: \*/\*\r\n
\r\n
<!--NeedCopy-->
```

HTTP レスポンスのフォーマット

HTTP 応答には、ステータスメッセージ、応答 HTTP ヘッダー、および要求されたオブジェクト、または要求されたオブジェクトが処理できない場合はエラーメッセージが含まれます。

レスポンスの例を以下に示します。

```
HTTP/1.1 200 OK\r\n
Content-Length: 55\r\n
Content-Type: text/html\r\n
Last-Modified: Wed, 12 Aug 1998 15:03:50 GMT\r\n
Accept-Ranges: bytes\r\n
ETag:  “04f97692cbd1:377” \r\n
Date: Thu, 19 Jun 2008 19:29:07 GMT\r\n
\r\n
<55-character response>
<!--NeedCopy-->
```

HTTP コールアウトの設定

HTTP コールアウトを設定するときは、リクエストのタイプ（HTTP または HTTPS）、送信先、およびリクエストの形式を指定します。応答の期待される形式、および最後に分析する応答の部分。

宛先には、HTTP コールアウトエージェントの IP アドレスとポートを指定します。または、負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーを使用して、HTTP コールアウト要求を管理します。

最初のケースでは、HTTP コールアウト要求は HTTP コールアウトエージェントに直接送信されます。2 番目のケースでは、HTTP コールアウト要求は、指定された仮想サーバーの仮想 IP アドレス (VIP) に送信されます。仮想サーバーは、クライアント要求を処理するのと同じ方法で要求を処理します。たとえば、多数のコールアウトが生成されることが予想される場合、HTTP コールアウトエージェントのインスタンスを複数のサーバーで構成し、これらのインスタンスを (サービスとして) 負荷分散仮想サーバーにバインドし、HTTP コールアウト設定で負荷分散仮想サーバーを指定できます。次に、負荷分散仮想サーバーは、負荷分散アルゴリズムによって決定された設定済みインスタンスの負荷を分散します。

HTTP コールアウト要求の形式については、HTTP コールアウト要求の個々の属性（属性ベースの HTTP コールアウト）を指定するか、HTTP コールアウト要求全体を高度なポリシー式（式ベースの HTTP コールアウト）として指定できます。

HTTP コールアウト要求の形式については、HTTP コールアウト要求の個々の属性（属性ベースの HTTP コールアウト）を指定するか、HTTP コールアウト要求全体を高度なポリシー式（式ベースの HTTP コールアウト）として指定できます。

詳細については、「ポリシー httpCallout」を参照してください。

パラメーター	説明
名前	コールアウトの名前 (最大 127 文字)
IP アドレスとポート (IP アドレス/ポート) または仮想サーバー名 (vserver)	コールアウトの送信先サーバーの IPv4 または IPv6 アドレス、またはワイルドカード、およびコールアウトの送信先サーバー上のポート、またはワイルドカード。または、サービスタイプが HTTP の負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーの名前。
HTTP メソッド (httpMethod)	HTTP メソッド (httpMethod)。このコールアウトが送信する HTTP リクエストで使用されるメソッド。有効な値:GET または POST。デフォルト:GET。
ホスト式 (HostExpr)	ホスト式 (HostExpr)。Host ヘッダーを構成するための高度なテキスト式。最大長:255。式はリテラル値でも、値を導き出す高度な式でもかまいません。 例：“10.101.10.11”，“http.req.header(“Host”)”
URL ステム式 (urlStemExpr)	URL ステム式 (urlStemExpr) URL ステムを生成するための高度な文字列式です。最大長:8191. 式には、リテラル文字列または値を導出する式を指定できます。 例:”“/mysite/index.html”” “http.req.url”
HTTP ヘッダー (ヘッダー)	HTTP ヘッダー (ヘッダー)。HTTP コールアウトリクエストに HTTP ヘッダーとその値を挿入するための高度なテキスト式。すべてのヘッダーに値を指定します。ヘッダー名を文字列として指定し、ヘッダー値を高度な式として指定します。ヘッダーをスペースで区切って指定します。-headers cip (client.ip.src) hdr (http.req.header (「HDR」)) など。ヘッダーの数は 8 にすることができます。

パラメーター	説明
サーバーに送信する式ベースの要求 (fullReqExpr)	NetScaler が 8191 文字までの高度な表現として送信する HTTP リクエストとまったく同じです。このパラメータを指定する場合は、HttpMethod、HostExpr、urlStemExpr、ヘッダー、およびパラメーターの引数を省略する必要があります。リクエスト式は、コールアウトが使用されるフィーチャによって制約されます。たとえば、HTTP.RES 式は、要求時ポリシーバンクまたは TCP コンテンツスイッチングポリシーバンクでは使用できません。
サーバーに送信する式ベースの要求 (BodyExpr)	リクエストの本文を生成するための高度な文字列式。式には、リテラル文字列または値を導出する式 (client.ip.src など) を含めることができます。-fullReqExpr と相互に排他的です。
パラメーター	コールアウトが送信する HTTP リクエストにクエリパラメータを挿入するための高度な式。設定するすべてのパラメータの値を指定します。コールアウトリクエストが GET メソッドを使用する場合、これらのパラメーターは URL に挿入されます。コールアウトリクエストが POST メソッドを使用する場合、これらのパラメータは POST 本文に挿入されます。クエリパラメータ名を文字列として、値を高度な式として設定します。パラメータ値は URL エンコードされます。パラメータをスペースで区切って指定します 例 パラメータ name1 (「name1」) name2 (http.req.header (「hdr」))。最大 8 つのパラメータを設定できます。
戻り値の型 (ReturnType)	ターゲットアプリケーションがコールアウトへの応答で返すデータのタイプ。有効な値:TEXT: 戻り値をテキスト文字列として扱います。NUM: 戻り値を数値として扱います。BOOL: 戻り値をブール値として扱います。注意: 戻り値の型は、設定後に変更できません。

パラメーター	説明
レスポンスからデータを抽出する式 (ResultExpr)	HTTP コールアウトへの応答から HTTP.RES オブジェクトを抽出する高度な式。最大長は 8191 です。この式の操作は、戻り値の型と一致する必要があります。たとえば、戻り値の種類のテキストを構成する場合、結果式はテキストベースの式である必要があります。戻り値の型が num の場合、結果式 (resultExpr) は次のような数値を返す必要があります。「http.res.body (10000) .length」注: 場合によっては、戻り値の型を TEXT に設定し、サーバーから送信される結果が 16 KB を超えると、結果式が NULL を返すことがあります。たとえば、結果が 16 KB を超える連結文字列になった場合などです。
スキーム	コールアウトサーバーのスキームのタイプ。例: HTTP, https
cacheForsecs	コールアウト応答がキャッシュされる期間（秒）。キャッシュされた応答は、「calloutContentGroup」という名前の統合キャッシュコンテンツグループに保存されます。デュレーションが設定されていない場合、通常のキャッシュ構成を使用してキャッシュしない限り、コールアウト応答はキャッシュされません。このパラメータは、これらの応答に適用される通常のキャッシュ設定よりも優先されます。

注: アプライアンスはリクエストの有効性をチェックしません。リクエストが有効なリクエストであり、機密情報が含まれていないことを確認する必要があります。正しくないまたは不完全な HTTP コールアウト構成は、アクションに関連付けられていない実行時 UNDEF 条件になります。UNDEF 条件は、未定義のヒットカウンタを更新するだけで、誤って設定された HTTP コールアウトをトラブルシューティングできます。ただし、アプライアンスは HTTP コールアウトリクエストを解析して、特定の NetScaler 機能をコールアウト用に構成できるようにします。これにより、HTTP コールアウトが自身を呼び出す可能性があります。コールアウトの再帰とその回避方法については、「HTTP コールアウトの再帰を回避する」を参照してください。

最後に、HTTP 要求属性を使用するか、式を使用して HTTP コールアウト要求の形式を定義するかに関係なく、HTTP コールアウトエージェントからの応答の形式と、評価する応答の部分を指定する必要があります。レスポンスタイプには、ブール値、数値、またはテキストを使用できます。この戻り値の型だけに基づいて、コールアウト応答でさらにエクスプレッションメソッドを使用できます。戻り値の型が数値の場合、コールアウト応答で数値ベースの式を使用できます。評価する応答の部分は、式によって指定されます。たとえば、応答にテキストが含まれるように指定した場合、`HTTP.RES.BODY(<unit>)` を使用して、アプライアンスがコールアウトエージェントからの応答の最初の <unit> バイトのみを評価するように指定できます。

コマンドラインでは、最初に
add コマンドを使用して HTTP コールアウトを作成します。コールアウトを追加すると、デフォルト値の GET に設定された HTTP メソッドを除き、すべてのパラメータがデフォルト値の NONE に設定されます。次に、set コマンドを使用してコールアウトのパラメータを設定します。set コマンドは、両方のタイプのコールアウト（属性ベースとエクスプレッションベース）を設定するために使用します。違いは、2 種類のコールアウトを設定するために使用されるパラメータにあります。したがって、次のコマンドライン手順には、属性ベースのコールアウトを設定するための set コマンドと、式ベースのコールアウトを設定するための set コマンドが含まれます。構成ユーティリティでは、これらの構成タスクはすべて 1 つのダイアログボックスで実行されます。

注:HTTP コールアウトをポリシーに入れる前に、戻り値のタイプを除くすべての設定済みパラメータを変更できます。HTTP コールアウトがポリシー内にあると、コールアウトで設定されている式を完全に変更することはできません。たとえば、クライアント.IP.SRC に HTTP.REQ ヘッダー (「マイバル」) を変更することはできません。式に渡される演算子と引数は変更できます。たとえば、HTTP.REQ.HEADER("myVal1")をHTTP.REQ.HEADER("myVal2")にまたはHTTP.REQ.HEADER("myVal")をHTTP.REQ.HEADER("myVal").AFTER_STR(<string>)に変更できます。set コマンドが失敗した場合は、HTTP コールアウトを作成します。

HTTP コールアウトの設定には、高度なポリシー式の設定が含まれます。高度なポリシー式の設定の詳細については、「高度なポリシー式の設定: はじめに」を参照してください。

コマンドラインインターフェイスを使用して **HTTP** コールアウトを構成するには

コマンドプロンプトで、次の操作を行います。

HTTP コールアウトを作成します。

```
add policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port<
    port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod (
    GET | POST )] [-hostExpr <expression>]  [-urlStemExpr <expression>]
    [-headers <name(value)> ...] [-parameters <name(value)> ...] [-
    bodyExpr <expression>] [-fullReqExpr <expression>] [-scheme ( http |
     https )] [-resultExpr <expression>] [-cacheForSecs <secs>] [-
    comment <string>]

<!--NeedCopy-->
```

例:

```
add policy httpCallout mycallout -vserver lbv1 -returnType num -
    httpMethod GET -hostExpr 'http.req.header("Host")'-urlStemExpr "http
    .req.url" -parameters Name("My Name") -headers Name("MyHeader")-
    resultExpr "http.res.body(10000).length"

<!--NeedCopy-->
```

HTTP コールアウトの設定を変更します。


```
set policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr|\*>] [-
    port <port|\*>] [-vServer <string>] [-returnType <returnType>] [-
    httpMethod ( GET | POST )] [-hostExpr <string>] [-urlStemExpr <
    string>] [-headers <name(value)> ...] [-parameters <name(value)>
    ...] [-resultExpr <string>]

<!--NeedCopy-->
```

例:

```
> set policy httpCallout mycallout -vserver lbv1 -returnType num -
    httpMethod GET -hostExpr 'http.req.header("Host")'-urlStemExpr "http
    .req.url" -parameters Name("My Name") -headers Name("MyHeader") -
    resultExpr "http.res.body(10000).length"
<!--NeedCopy-->
```

FullReqExpr パラメータを使用して HTTP コールアウトを構成します。

```
set policy httpCallout <name> [-vServer <string>] [-returnType <
    returnType>] [-fullReqExpr <string>] [-resultExpr <string>]
<!--NeedCopy-->
```

例:

```
> set policy httpCallout mycallout1 -vserver lbv1 -returnType num
    fullReqExpr q{
 "GET " + http.req.url + "HTTP/" + http.req.version.major + "." + http.
    req.version.minor.sub(1) + "r\nHost:10.101.10.10\r\nAccept: \*/\*\r
    \n\r\n" }


<!--NeedCopy-->
```

HTTP コールアウトの設定を確認します。

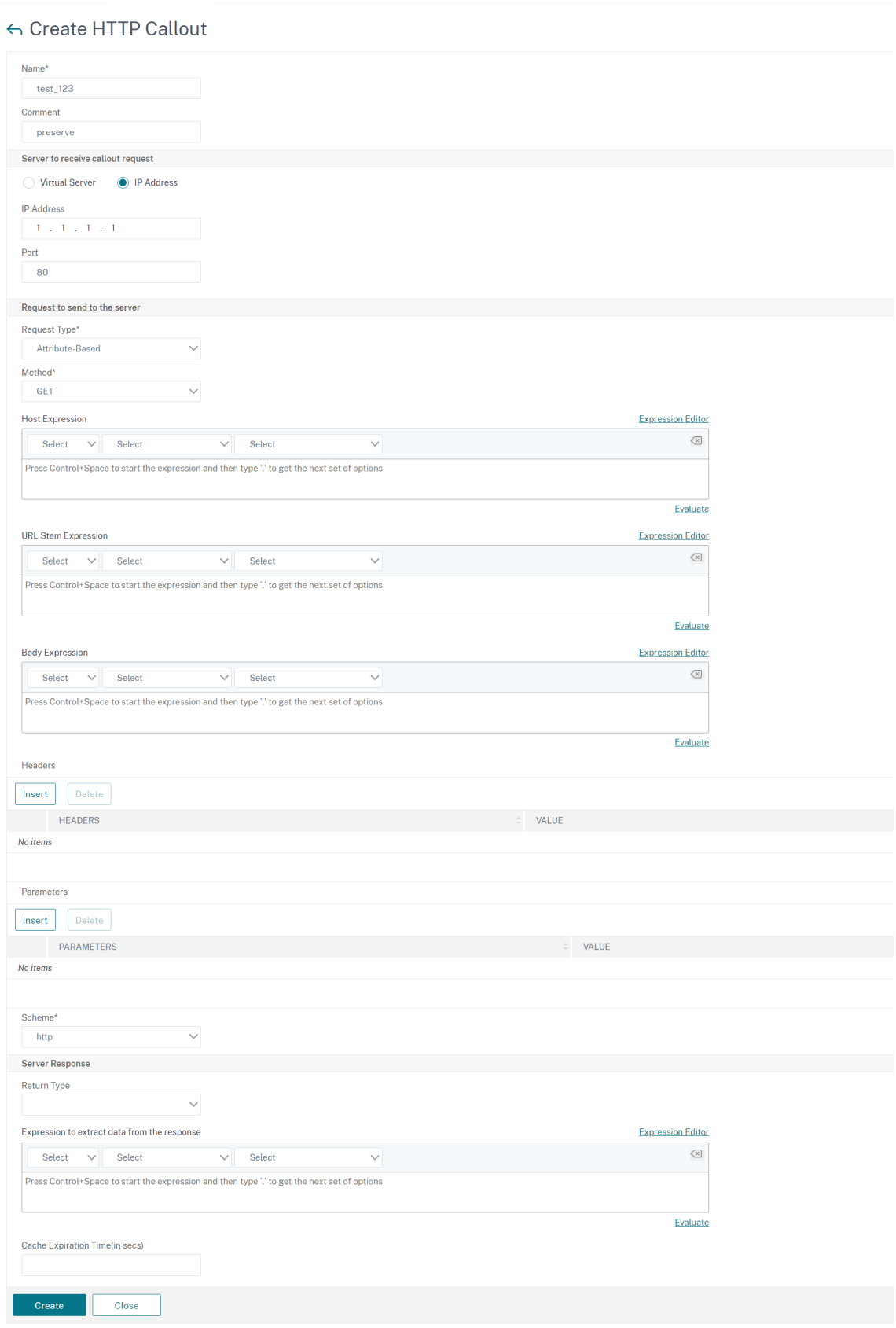
```
show policy httpCallout `<name>`

sh policy httpCallout mycallout1
> Name: mycallout1
>Vserver: lbv1 (UP)
Effective Vserver state: UP
Return type: TEXT
Scheme: HTTP
Full REQ expr: "GET " + http.req.url + "HTTP/" + http.req.version.major
    + "." + http.req.version.minor.sub(1)+ "r\nHost:10.101.10.10\r\
    nAccept: \*/\*\r\n\r\n"
Result expr: http.res.body(100)
Hits: 0
Undef Hits: 0
Done
>

<!--NeedCopy-->
```


構成ユーティリティを使用して **HTTP** コールアウトを構成するには

1. **AppExpert** > **HTTP Callouts** に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [**HTTP** コールアウトの作成] ダイアログボックスで、HTTP コールアウトのパラメータを設定します。パラメータの説明については、チェックボックスの上にマウスカーソルを合わせます。
4. [作成] をクリックし、[閉じる] をクリックします。



設定の確認

August 15, 2023

HTTP コールアウトが正しく動作するには、すべての HTTP コールアウトパラメータとコールアウトに関連するエンティティが正しく設定されている必要があります。NetScaler アプライアンスは HTTP コールアウトパラメーターの有効性をチェックしませんが、バインドされたエンティティ、つまり HTTP コールアウトの送信先のサーバーまたは仮想サーバーの状態を示します。次の表は、アイコンの一覧とアイコンが表示される条件を示しています。

アイコン	それを示します
	HTTP コールアウトエージェントをホストするサーバー、または HTTP コールアウトの送信先となる負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーの状態は UP です。
	HTTP コールアウトエージェントをホストするサーバー、または HTTP コールアウトの送信先となる負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーの状態が OUT OF SERVICE です。
	HTTP コールアウトエージェントをホストするサーバー、または HTTP コールアウトの送信先となる負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーの状態は DOWN です。

表 1. HTTP コールアウトにバインドされたエンティティの状態を示すアイコン

HTTP コールアウトが正しく機能するには、アイコンが常に緑色でなければなりません。アイコンが緑色でない場合は、HTTP コールアウトの送信先となるコールアウトサーバーまたは仮想サーバーの状態を確認します。アイコンが緑色なのに HTTP コールアウトが期待どおりに機能しない場合は、コールアウトに設定されているパラメータを確認してください。

また、HTTP コールアウトの呼び出し元のポリシーと一致するテストリクエストを送信し、ポリシーと HTTP コールアウトのヒットカウンターを確認し、NetScaler アプライアンスがクライアントに送信する応答を検証することで、構成を検証することもできます。

注:HTTP コールアウトは、再帰的に 2 回呼び出すことがあります。この場合、アプライアンスによって生成されるコールアウトごとに、ヒットカウンタが 2 カウントずつ増加します。hits カウンタが正しい値を表示するには、HTTP コールアウトを 2 回目に呼び出さないように設定する必要があります。HTTP コールアウトの再帰を回避する方法の詳細については、「HTTP コールアウトの再帰を回避する」を参照してください。

HTTP コールアウトのヒットカウンタを表示するには

1. **AppExpert** > **HTTP Callouts** に移動します。
2. 詳細ペインで、ヒットカウンターを表示したい HTTP コールアウトをクリックし、詳細領域にヒット数を表示します。

HTTP コールアウトの呼び出し

August 15, 2023

HTTP コールアウトを設定したら、詳細ポリシールールに`SYS.HTTP_CALLOUT(<name>)`式を含めることによってコールアウトを呼び出します。この式で、`<name>`は、呼び出す HTTP コールアウトの名前です。

コールアウト式で高度なポリシー式演算子を使用して、応答を処理し、適切なアクションを実行できます。HTTP コールアウトエージェントからの応答の戻り値の型によって、応答に使用できる演算子のセットが決まります。分析する応答の一部がテキストの場合、テキスト演算子を使用して応答を分析できます。たとえば、CONTAINS (<string>) 演算子を使用して、次の例のように、レスポンスの指定された部分に特定の文字列が含まれているかどうかを確認できます。

```
SYS.HTTP_CALLOUT(mycallout).contains("Good IP address")
<!--NeedCopy-->
```

レスポンダーポリシーで前述の式を使用する場合は、適切なレスポンダーアクションを構成できます。

同様に、評価する応答の一部が数値である場合は、GT (int) などの数値演算子を使用できます。応答にブール値が含まれている場合は、ブール演算子を使用できます。

注:HTTP コールアウトは自身を再帰的に呼び出すことができます。HTTP コールアウトの再帰を回避するには、HTTP コールアウト式を再帰を防止する高度なポリシー式と組み合わせます。HTTP コールアウトの再帰を回避する方法については、「HTTP コールアウトの再帰を回避する」を参照してください。

以前に生成されたコールアウトを評価した後にコールアウトを呼び出すポリシーを設定することで、HTTP コールアウトをカスケードすることもできます。このシナリオでは、1 つのポリシーがコールアウトを呼び出した後、NetScaler アプライアンスがコールアウトを解析してからコールアウトをコールアウトサーバーに送信すると、2 つ目のポリシーセットがコールアウトを評価して追加のコールアウトを呼び出し、さらにコールアウトが 3 つ目のポリシーセットで評価されるというようになります。このような実装を次の例で説明します。

まず、myCallout1 という名前の HTTP コールアウトを設定し、次に myCallout1 を呼び出すようにレスポンダーポリシー Pol1 を構成します。次に、2 番目の HTTP コールアウト myCallout2 とレスポンダーポリシー Pol2 を設定できます。myCallout1 を評価し、myCallout2 を呼び出すように Pol2 を設定するとします。両方のレスポンダーポリシーをグローバルにバインドします。

HTTP コールアウトの再帰を避けるために、myCallout1 は「Request1」という一意のカスタム HTTP ヘッダーで設定されています。Pol1 は、高度なポリシー式を使用して HTTP コールアウトの再帰を回避するように設定されています。

```
HTTP.REQ.HEADER("Request1").EQ("Callout Request").NOT.
<!--NeedCopy-->
```

Pol2 は同じ詳細ポリシー式を使用しますが、.NOT 演算子は除外されるため、NetScaler ADC アプライアンスが解析しているときにポリシーが MyCallout1 を評価します。myCallout2 は「Request2」と呼ばれる独自のヘッダーを識別し、Pol2 には myCallout2 が自身を再帰的に呼び出さないようにする高度なポリシー式が含まれています。

例:

```
> add policy httpCallout myCallout1

Done

> set policy httpCallout myCallout1 -IPAddress 10.102.3.95 -port 80 -
    returnType TEXT -hostExpr
 ""10.102.3.95"" -urlStemExpr ""/cgi-bin/check_clnt_from_database.pl""
     -headers Request1
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.
    RES.BODY(100)"

Done

> add responder policy Pol1 "HTTP.REQ.HEADER("Request1").EQ("Callout
    Request").NOT &&
SYS.HTTP_CALLOUT(myCallout1).CONTAINS("IP Matched")" RESET

Done

> bind responder global Pol1 100 END -type OVERRIDE

Done

> add policy httpCallout myCallout2

Done

> set policy httpCallout myCallout2 -IPAddress 10.102.3.96 -port 80 -
    returnType TEXT -hostExpr
""10.102.3.96"" -urlStemExpr ""/cgi-bin/
    check_clnt_location_from_database.pl"" -headers Request2
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.
    RES.BODY(200)"

 Done

> add responder policy Pol2 "HTTP.REQ.HEADER("Request2").EQ("Callout
    Request").NOT &&
```



```
31  HTTP.REQ.HEADER("Request1").EQ("Callout Request") && SYS.HTTP_CALLOUT(
       myCallout2).CONTAINS
32 ("APAC")" RESET
33
34  Done
35
36 > bind responder global Pol2 110 END -type OVERRIDE
37
38 Done
39 <!--NeedCopy-->
```

HTTP コールアウト再帰の回避

August 15, 2023

NetScaler アプライアンスは HTTP コールアウトリクエストの有効性をチェックしませんが、リクエストを HTTP コールアウトエージェントに送信する前にリクエストを一度解析します。この解析により、アプライアンスはコールアウトリクエストを他の着信リクエストと同様に扱うことができるため、コールアウトリクエストを処理するようにいくつかの便利な NetScaler 機能（統合キャッシュなど）を構成できます。

ただし、この解析中、HTTP コールアウト要求は同じポリシーを選択し、それ自体を再帰的に呼び出すことができます。アプライアンスは再帰呼び出しを検出し、未定義（UNDEF）条件を発生させます。ただし、再帰呼び出しにより、ポリシーおよび HTTP コールアウト選択カウンタは、それぞれ 1 カウントではなく、2 カウントずつ増加します。

コールアウトが自身を呼び出さないようにするには、HTTP コールアウト・リクエストの一意の特性を少なくとも 1 つ特定し、この特性を持つすべてのリクエストを、コールアウトを呼び出すポリシールールによって処理されないようにする必要があります。そのためには、ポリシールールに別の Advanced ポリシー式を含めます。`SYS.HTTP_CALLOUT(<name>)`式は、コールアウト式が評価される前に評価されるように、式の前に式を付ける必要があります。次に例を示します：

```
1 <Expression that prevents callout recursion> OR SYS.HTTP_CALLOUT(<name
     >)
2 <!--NeedCopy-->
```

この方法でポリシールールを設定すると、アプライアンスが要求を生成して解析すると、複合ルールは FALSE と評価され、コールアウトは 2 回目に生成されず、選択カウンタが正しく増分されます。

HTTP コールアウト要求に一意の特性を割り当てる 1 つの方法は、コールアウトを設定するときに一意のカスタム HTTP ヘッダーを含めることです。以下は、「MyCallout」と呼ばれる HTTP コールアウトの例です。コールアウトは、クライアントの IP アドレスがブラックリストに登録された IP アドレスのデータベースに存在するかどうかをチェックする HTTP 要求を生成します。コールアウトには「Request」というカスタムヘッダーが含まれ、値「Callout Request」に設定されています。グローバルにバインドされたレスポンダーポリシー「Pol1」は HTTP コールアウトを呼び出しますが、Request ヘッダーがこの値に設定されているすべてのリクエストを除外します。こ

れにより、myCallout の 2 回目の呼び出しが防止されます。2 回目の呼び出しを防ぐ式は、HTTP.REQ.HEADER(“Request”).EQ(“Callout Request”).NOT です。

例:

```
> add policy httpCallout myCallout
 Done

> set policy httpCallout myCallout -IPAddress 10.102.3.95 -port 80 -
   returnType TEXT -hostExpr ""10.102.3.95"" -urlStemExpr ""/cgi-bin/
   check_clnt_from_database.pl"" -headers Request("Callout Request") -
   parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(100)"
Done

> add responder policy Pol1 "HTTP.REQ.HEADER("Request").EQ("Callout
   Request").NOT && SYS.HTTP_CALLOUT(myCallout).CONTAINS("IP Matched")"
    RESET
Done

> bind responder global Pol1 100 END -type OVERRIDE
Done
<!--NeedCopy-->
```

> 注:
> リクエスト URL に HTTP コールアウト用に設定されたステム式が含まれているかどうかを確認する式を設定することもできます。このソリューションを実装するには、HTTP コールアウトエージェントが HTTP コールアウトにのみ応答でき、アプライアンスを介して送信される他の要求には応答できないようにします。HTTP コールアウトエージェントが他のクライアント要求を処理するアプリケーションまたは Web サーバーである場合、そのような式により、アプライアンスはそれらのクライアント要求を処理できません。代わりに、前述のように一意のカスタムヘッダーを使用します。

HTTP コールアウト応答のキャッシュ

August 15, 2023

コールアウトを使用する際のパフォーマンスを向上させるため、統合キャッシュ機能を使用してコールアウト応答をキャッシュできます。レスポンスは、指定された期間、CalloutContentGroup という名前の統合キャッシュコンテンツグループに保存されます。

注: コールアウト応答をキャッシュするには、統合キャッシュ機能が有効になっていることを確認してください。

コマンドラインインターフェイスを使用してキャッシュ期間を設定するには

コマンドプロンプトで入力します。


```
set policy httpCallout <name> -cacheForSecs <secs>
```

例:

```
> set httpcallout httpcallout1 -cacheForSecs 120
<!--NeedCopy-->
```

構成ユーティリティを使用してキャッシュ期間を設定するには

1. **AppExpert > HTTP** コールアウトに移動します。
2. 詳細ペインで、キャッシュ期間を設定する HTTP コールアウトを選択し、[開く] をクリックします。
3. 「**HTTP** コールアウトの設定」ダイアログで、キャッシュの有効期限を指定します。
4. 正しい期間を入力したことを確認し、「**OK**」をクリックします。

ユースケース**:IP** ブラックリストを使用したクライアントのフィルタリング

August 15, 2023

HTTP コールアウトは、管理者によってブラックリストに登録されているクライアントからの要求をブロックするために使用できます。クライアントのリストには、一般に知られているブラックリスト、組織で管理しているブラックリスト、またはその両方の組み合わせを使用できます。

NetScaler アプライアンスは、クライアントの IP アドレスを事前に構成されたブラックリストと照合し、IP アドレスがブラックリストに登録されている場合はトランザクションをブロックします。IP アドレスがリストにない場合、アプライアンスはトランザクションを処理します。

この設定を実装するには、次のタスクを実行する必要があります。

1. NetScaler アプライアンスでレスポンダーを有効にします。
2. NetScaler アプライアンスで HTTP コールアウトを作成し、外部サーバーやその他の必須パラメーターの詳細を使用して構成します。
3. HTTP コールアウトへの応答を分析するようにレスポンダーポリシーを設定し、ポリシーをグローバルにバインドします。
4. リモートサーバに HTTP コールアウトエージェントを作成します。

レスポンダーの有効化

Responder を使用する前に Responder を有効にする必要があります。

GUI を使用してレスポンダーを有効にするには

1. レスポンダーライセンスがインストールされていることを確認します。
2. 構成ユーティリティで、「AppExpert」を展開し、「レスポンダー」を右クリックして、「レスポンダー機能を有効にする」をクリックします。

NetScaler アプライアンスでの **HTTP** コールアウトの作成

次の表に示すパラメータ設定を使用して、HTTP コールアウト HTTP_Callout を作成します。HTTP コールアウトの作成の詳細については、「HTTP コールアウト PDF の設定 」を参照してください。

レスポンダーポリシーを構成し、グローバルにバインドする

HTTP コールアウトを設定したら、コールアウトの設定を確認し、コールアウトを呼び出すレスポンダポリシーを設定します。
[ポリシー] サブノードでレスポンダーポリシーを作成し、
レスポンダーポリシーマネージャーを使用してグローバルにバインドできますが、このデモでは、
レスポンダーポリシーマネージャーを使用してレスポンダーポリシーを作成し、ポリシーをグローバルにバインドします。

レスポンダーポリシーを作成し、を使用してグローバルにバインドするには

1. [**AppExpert**] > [レスポンダー] に移動します。
2. 詳細ペインの [ポリシーマネージャ] で、[**** ポリシーマネージャ **] をクリックします。
3. [レスポンダーポリシーマネージャー] ダイアログボックスで、[グローバルを上書き] をクリックします。
4. [ポリシーの挿入] をクリックし、[ポリシー名] の [新しいポリシー] をクリックします。
5. [レスポンダーポリシーの作成] ダイアログボックスで、次の操作を行います。
   a) [名前] に「**PolicyResponder1**」と入力します。
   b)「アクション」で、「リセット」を選択します。
   c)「未定義の結果アクション」で、「グローバル未定義の結果アクション」を選択します。
   d) [式] に、次の高度なポリシー式を入力します。

```
"HTTP.REQ.HEADER("Request").EQ("Callout Request").NOT && SYS.
   HTTP_CALLOUT(HTTP_Callout).CONTAINS("IP Matched")"
<!--NeedCopy-->
```

   e) [作成] をクリックし、[閉じる] をクリックします。
6. [変更を適用] をクリックし、[閉じる] をクリックします。

リモートサーバーでの **HTTP** コールアウトエージェントの作成

次に、NetScaler アプライアンスからのコールアウト要求を受信して適切に応答する HTTP コールアウトエージェントをリモートコールアウトサーバー上に作成する必要があります。HTTP コールアウトエージェントは、デプロイメントごとに異なるスクリプトであり、サポートされるデータベースのタイプやスクリプト言語など、サーバーの仕様を念頭に置いて記述する必要があります。

次に、指定された IP アドレスが IP ブラックリストの一部であるかどうかを検証するコールアウトエージェントの例を示します。エージェントは Perl スクリプト言語で記述されており、MYSQL データベースを使用しています。

次の CGI スクリプトは、コールアウトサーバー上の指定された IP アドレスをチェックします。

```
#!/usr/bin/perl -w
print "Content-type: text/html\n\n";
        use DBI();
        use CGI qw(:standard);
#Take the Client IP address from the request query
        my $ip_to_check = param('cip');
# Where a MYSQL database is running
        my $dsn = 'DBI:mysql:BAD_CLIENT:localhost';
# Database username to connect with
        my $db_user_name =  'dbuser' ;
# Database password to connect with
        my $db_password = 'dbpassword';
        my ($id, $password);
# Connecting to the database
        my $dbh = DBI->connect($dsn, $db_user_name, $db_password);
        my $sth = $dbh->prepare(qq{
  select * from bad_clnt  }
 );
        $sth->execute();
        while (my ($ip_in_database) = $sth->fetchrow_array()) {

        chomp($ip_in_database);
# Check for IP match
        if ($ip_in_database eq $ip_to_check) {

              print "\n IP Matched\n";
                                                $sth->finish();
                                          exit;
                 }

        }

       print "\n IP Failed\n";
       $sth->finish();
       exit;
<!--NeedCopy-->
```


ユースケース: コンテンツを動的に取得および更新するための ESI サポート

August 15, 2023

Edge Side Inincludes (ESI) は、エッジレベルの動的 Web コンテンツアセンブリのマークアップ言語です。ネットワークエッジで集約、組み立て、配信できるキャッシュ可能およびキャッシュ不可能な Web ページコンポーネントを記述する単純なマークアップ言語を定義することにより、動的な Web ベースのアプリケーションを高速化するのに役立ちます。NetScaler アプライアンスで HTTP コールアウトを使用すると、ESI 構成を読み取り、コンテンツを動的に集約またはアセンブルできます。

この設定を実装するには、次のタスクを実行する必要があります。

1. NetScaler アプライアンスでリライトを有効にします。
2. アプライアンスで HTTP コールアウトを作成し、外部サーバの詳細およびその他の必須パラメータを使用して設定します。
3. ESI コンテンツをコールアウトレスポンス本文に置き換える書き換えアクションを設定します。
4. リライトポリシーを設定して、アクションが実行される条件を指定し、リライトポリシーをグローバルにバインドします。

書き換えを有効にする

NetScaler アプライアンスで使用する前に、書き換えを有効にする必要があります。次の手順では、書き換え機能を有効にする手順について説明します。

GUI を使用して書き換えを有効にするには

1. 書き換えライセンスがインストールされていることを確認します。
2. 構成ユーティリティで、[AppExpert] を展開し、[書き換え] を右クリックし、[書き換え機能の有効化] をクリックします。

NetScaler アプライアンスでの HTTP コールアウトの作成

HTTP コールアウトの作成の詳細については、「HTTP コールアウトの設定」を参照してください。
パラメータ値の詳細については、http-Callout-2 pdf のパラメータと値を参照してください 。

書き換えアクションの設定

書き換えアクション Action-Rewrite-1 を作成し、ESI コンテンツをコールアウトレスポンス本文に置き換えます。次の表に示すパラメータ設定を使用します。

表 2. アクションリライト-1 のパラメータと値

パラメーター	値
名前	Action-Rewrite-1
種類	置換
ターゲットテキスト参照を選択する式	“HTTP.RES.BODY(500).AFTER_STR (\” \”).BEFORE_STR (\” \”)”
置換テキストの文字列式	“SYS.HTTP_CALLOUT(HTTP-Callout-2)”

構成ユーティリティを使用して書き換えアクションを構成するには

1. **[**AppExpert**] > [書き換え] > [アクション] に移動します。**
2. 詳細ペインで、[追加] をクリックします。
3. [書き換えアクションの作成] ダイアログボックスの [名前] に **Action-Rewrite-1** と入力します。
4. 「タイプ」で「置換」を選択します。
5. [ターゲットテキスト参照を選択する式] に、次の高度なポリシー式を入力します。

```
"HTTP.RES.BODY(500).AFTER_STR("<example>").BEFORE_STR("<example>")
   "
<!--NeedCopy-->
```

6. 置換テキストの [文字列式] に、次の文字列式を入力します。

```
"SYS.HTTP_CALLOUT(HTTP-Callout-2)"
<!--NeedCopy-->
```

7. [作成] をクリックし、[閉じる] をクリックします。

書き換えポリシーを作成してグローバルにバインドする

次の表に示すパラメータ設定を使用して、書き換えポリシー Policy-Rewrite-1 を作成します。[Policies] サブノードで書き換えポリシーを作成し、
書き換えポリシーマネージャを使用してグローバルにバインドできます。または、
書き換えポリシーマネージャを使用して、これらの両方のタスクを同時に実行することもできます。このデモでは、Rewrite Policy Manager を使用して両方のタスクを実行します。

表 3. ポリシー書き換え-1 のパラメータと値

パラメーター	値
名前	Policy-Rewrite-1
アクション	Action_Rewrite-1
未定義の結果アクション	-グローバル未定義の結果アクション-
式	“HTTP.REQ.HEADER(“Name”).CONTAINS (“Callout”).NOT”

構成ユーティリティを使用して書き換えポリシーを構成し、グローバルにバインドするには

1. **AppExpert** > [書き換え] に移動します。
2. 詳細ペインの [ポリシーマネージャ] で、[** ポリシーマネージャの書き換え **] をクリックします。
3. [ポリシーマネージャの書き換え] ダイアログボックスで、[グローバルを上書き] をクリックします。
4. [ポリシーの挿入] をクリックし、[ポリシー名] 列の [新しいポリシー] をクリックします。
5. [書き換えポリシーの作成] ダイアログボックスで、次の操作を行います。1
   . [名前] に「Policy-Rewrite-1」と入力します。

   a) [アクション] で、[アクション-書き換え-1] を選択します。
   b)「未定義の結果アクション」で、「グローバル未定義の結果アクション」を選択します。
   c) [式] に、次の高度なポリシー式を入力します。

```
"HTTP.REQ.HEADER("Name").CONTAINS("Callout").NOT"
<!--NeedCopy-->
```

   a) [作成] をクリックし、[閉じる] をクリックします。
6. [変更を適用] をクリックし、[閉じる] をクリックします。

使用事例**:** アクセス制御と認証

August 15, 2023

セキュリティの高いゾーンでは、クライアントがリソースにアクセスする前に、ユーザーを外部から認証することが必須です。NetScaler アプライアンスでは、HTTP コールアウトを使用して、提供された資格情報を評価することでユーザーを外部認証できます。この例では、クライアントがリクエスト内の HTTP ヘッダーを介してユーザー名とパスワードを送信していることを前提としています。ただし、URL または HTTP 本文から同じ情報を取得することはできます。

この設定を実装するには、次のタスクを実行する必要があります。

1. NetScaler アプライアンスのレスポンダー機能を有効にします。
2. アプライアンスで HTTP コールアウトを作成し、外部サーバの詳細およびその他の必須パラメータを使用して設定します。
3. レスポンダーポリシーを設定して応答を分析し、ポリシーをグローバルにバインドします。
4. リモートサーバにコールアウトエージェントを作成します。

レスポンダーの有効化

レスポンダー機能は、NetScaler アプライアンスで使用する前に有効にする必要があります。

構成ユーティリティを使用してレスポンダーを有効にするには

1. レスポンダーライセンスがインストールされていることを確認してください。
2. 構成ユーティリティで、「AppExpert」を展開し、「レスポンダー」を右クリックして、「レスポンダー機能を有効にする」をクリックします。

NetScaler アプライアンスでの **HTTP** コールアウトの作成

次の表に示すパラメータ設定を使用して、HTTP コールアウトである HTTP-Callout-3 を作成します。HTTP コールアウトの作成の詳細については、「HTTP コールアウトの設定」を参照してください。

表 1. HTTP コールアウト 3 のパラメータと値

パラメーター	値	名前
名前	Policy-Responder-3	

パラメーター

値

名前

HTTP-Callout-3

コールアウトリクエストを受信するサーバー:

IP アドレス

10.103.9.95

ポート

80

サーバーに送信するリクエスト:

方法

GET

ホスト表現

10.102.3.95

URL ステムエクスプレッション

“/cgi-bin/authenticate.pl”

ヘッダー:

名前

リクエスト

価値表現

コールアウトリクエスト

パラメーター:

名前

ユーザー名

価値表現

HTTP.REQ.HEADER(“Username”).VALUE(0)

名前

パスワード

価値表現

HTTP.REQ.HEADER(“Password”).VALUE(0)

サーバーの応答:

返品タイプ

テキスト

応答からデータを抽出する式

HTTP.RES.BODY(100)

応答を分析するためのレスポンダーポリシーの作成

コールアウトサーバーからの応答を確認し、送信元 IP アドレスがブラックリストに登録されている場合は接続をリセットするレスポンダーポリシー Policy-Responder-3 を作成します。次の表に示すパラメータ設定を使用してポ

リシーを作成します。ポリシーサブノードでレスポンダーポリシーを作成し、
レスポンダーポリシーマネージャーを使用してグローバルに
バインドすることもできますが、このデモでは
レスポンダーポリシーマネージャーを使用してレスポンダーポリシーを作成し、ポリシーをグローバルにバインドします。

表 2. ポリシーレスポンダー 3 のパラメータと値

パラメーター	値
名前	Policy-Responder-3
アクション	RESET
未定義の結果アクション	-グローバル未定義の結果アクション-
式	「HTTP.REQ.HEADER (\”リクエスト\」) .EQ (\ “コールアウトリクエスト\」) .NOT && SYS.HTTP_CALLOUT (HTTP-Callout-3) .CONTAINS (\ “認証失敗\」)」

レスポンダーポリシーを作成し、構成ユーティリティを使用してグローバルにバインドするには

1. [**AppExpert**] > [レスポンダー] に移動します。
2. 詳細ペインの [ポリシーマネージャー] で、[レスポンダーポリシーマネージャー] をクリックします。
3. レスポンダーポリシーマネージャーダイアログボックスで、「グローバルオーバーライド」をクリックします。
4. [ポリシーの挿入] をクリックし、[ポリシー名] 列の [新しいポリシー] をクリックします。
5. [レスポンダーポリシーの作成] ダイアログボックスで、次の操作を行います。
   a) [名前] に「ポリシーレスポンダー-3」と入力します。
   b) [アクション] で [リセット] を選択します。
   c)「未定義結果アクション」で、「グローバル未定義結果アクション」を選択します。
   d) Expression テキストボックスに、次のように入力します。

```
"HTTP.REQ.HEADER("Request").EQ("Callout Request").NOT && SYS.
   HTTP_CALLOUT(HTTP-Callout-3).CONTAINS("Authentication Failed")"
<!--NeedCopy-->
```

   a) **Create**、**Close** の順にクリックします。
6. [変更を適用] をクリックし、[閉じる] をクリックします。

リモートサーバーでの **HTTP** コールアウトエージェントの作成

次に、リモートコールアウトサーバーに HTTP コールアウトエージェントを作成する必要があります。HTTP コールアウトエージェントは、NetScaler アプライアンスからコールアウト要求を受け取り、適切に応答します。コールアウトエージェントはデプロイメントごとに異なるスクリプトであり、データベースの種類やサポートされているスクリプト言語など、サーバーの仕様を念頭に置いて作成する必要があります。

以下は、指定されたユーザー名とパスワードが有効かどうかを検証するコールアウトエージェントの擬似コードの例です。エージェントは、任意のプログラミング言語で実装できます。疑似コードは、コールアウトエージェントを開発する際のガイドラインとしてのみ使用してください。プログラムに追加の機能を組み込むことができます。

提供されたユーザー名とパスワードを疑似コードを使用して確認するには

1. リクエストに入力されたユーザー名とパスワードを受け入れ、適切な形式にします。
2. すべての有効なユーザー名とパスワードを含むデータベースに接続します。
3. 提供された認証情報をデータベースと照合してください。
4. HTTP コールアウトの要求に応じて応答をフォーマットします。
5. 応答を NetScaler アプライアンスに送信します。

使用事例**:OWA** ベースのスパムフィルタ

August 15, 2023

迷惑メールフィルターとは、既知または信頼できる送信元以外からのメールや、不適切な内容を含むメールを動的にブロックする機能です。スパムフィルタリングには、特定の種類のメッセージがスパムであることを示す関連ビジネスロジックが必要です。NetScaler アプライアンスが HTTP プロトコルに基づいて Outlook Web Access（OWA）メッセージを処理する場合、HTTP コールアウトを使用してスパムをフィルタリングできます。

HTTP コールアウトを使用して受信メッセージの任意の部分を抽出し、メッセージが正当なものかスパムかを判断するためのルールが設定された外部コールアウトサーバーで確認できます。スパムメールの場合、セキュリティ上の理由から、NetScaler アプライアンスはメールがスパムとしてマークされていることを送信者に通知しません。

次の例では、メールの件名にリストされているさまざまなキーワードについて非常に基本的なチェックを行っています。これらのチェックは、実稼働環境ではより複雑になる可能性があります。

この設定を実装するには、次のタスクを実行する必要があります。

1. NetScaler アプライアンスのレスポンダー機能を有効にします。
2. NetScaler アプライアンスで HTTP コールアウトを作成し、外部サーバーやその他の必須パラメーターの詳細を使用して構成します。
3. レスポンダーポリシーを作成してレスポンスを分析し、ポリシーをグローバルにバインドします。
4. リモートサーバにコールアウトエージェントを作成します。

レスポンダーの有効化

NetScaler アプライアンスで使用するには、レスポンダー機能を有効にする必要があります。

GUI を使用してレスポンダーを有効にするには

1. レスポンダーライセンスがインストールされていることを確認してください。
2. 構成ユーティリティで、「AppExpert」を展開し、「レスポンダー」を右クリックして、「レスポンダー機能を有効にする」をクリックします。

NetScaler アプライアンスでの **HTTP** コールアウトの作成

次の表に示すパラメータ設定を使用して、HTTP コールアウトである HTTP-Callout-4 を作成します。HTTP コールアウトの作成の詳細については、「HTTP コールアウトの設定」を参照してください。

詳細については、HTTP-Callout-4 pdf のパラメータと値を参照してください 。

レスポンダアクションの作成

レスポンダーアクション、アクションレスポンダー 4 を作成します。次の表に示すパラメータ設定を使用してアクションを作成します。

パラメーター	値
名前	Action-Responder-4
種類	で応答
ターゲット	「” HTTP/1.1 200 OK\ r\ n サーバー:Microsoft-IIS/6.0\ r\ nX-Powered-by: ASP.NET\ r\ n コンテンツ長:0\ r\ nMS-WebStorage: 6.5.6944\ r\ n キャッシュコントロール: キャッシュなし\ r\ n\ r\ n “」

表 2. アクションレスポンダー 4 のパラメーターと値

構成ユーティリティを使用してレスポンダーアクションを作成するには

1. **[**AppExpert**] > [レスポンダー] > [アクション] に移動します。**
2. 詳細ペインで、[追加] をクリックします。

3. [レスポンダーアクションの作成] ダイアログボックスの [名前] に「**Action-Responder-4**」と入力します。
4. [タイプ] の [返信先] をクリックします。
5. 「ターゲット」に、次のように入力します。

```
""HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By:
   ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\
   nCache-Control: no-cache\r\n\r\n""
<!--NeedCopy-->
```

6. [作成] をクリックし、[閉じる] をクリックします。

HTTP コールアウトを呼び出すレスポンダーポリシーの作成

リクエスト本文をチェックするレスポンダーポリシー Policy-Responder-4 を作成し、本文に「*subject*」という単語が含まれている場合は、HTTP コールアウトを呼び出してメールを確認します。次の表に示すパラメータ設定を使用してポリシーを作成します。ポリシーサブノードでレスポンダーポリシーを作成し、
レスポンダーポリシーマネージャーを使用して
グローバルにバインドすることもできますが、
このデモではレスポンダーポリシーマネージャーを使用してレスポンダーポリシーを作成し、グローバルにバインドします。

パラメーター	値
名前	Policy-Responder-4
アクション	Action-Responder-4
未定義の結果アクション	-グローバル未定義の結果アクション-
式	“HTTP.REQ.BODY(1000).CONTAINS(“urn:schemas:httpmail:subject”) && SYS.HTTP_CALLOUT(HTTP-Callout-4)”

構成ユーティリティを使用してレスポンダーポリシーを作成するには

1. [**AppExpert**] > [レスポンダー] に移動します。
2. 詳細ペインの [ポリシーマネージャー] で、[レスポンダーポリシーマネージャー] をクリックします。
3. レスポンダーポリシーマネージャーダイアログボックスで、「グローバルオーバーライド」をクリックします。
4. [ポリシーの挿入] をクリックし、[ポリシー名] 列の [新しいポリシー] をクリックします。
5. [レスポンダーポリシーの作成] ダイアログボックスで、次の操作を行います。
   a) [名前] に「ポリシーレスポンダー **4**」と入力します。
   b) 「アクション」で、「アクションレスポンダー **4**」をクリックします。

c)「未定義結果アクション」で、「グローバル未定義結果アクション」をクリックします。

d) **Expression** テキストボックスに、次のように入力します。

```
"HTTP.REQ.BODY(1000).CONTAINS("urn:schemas:httpmail:subject")
   && SYS.HTTP_CALLOUT(HTTP-Callout-4)"
<!--NeedCopy-->
```

e) **Create**、**Close** の順にクリックします。

6. [変更を適用] をクリックし、[閉じる] をクリックします。

リモートサーバーでの **HTTP** コールアウトエージェントの作成

次に、リモートコールアウトサーバーに HTTP コールアウトエージェントを作成する必要があります。HTTP コールアウトエージェントは、NetScaler アプライアンスからコールアウト要求を受信し、それに応じて応答します。コールアウトエージェントはデプロイメントごとに異なるスクリプトであり、データベースの種類やサポートされているスクリプト言語など、サーバーの仕様を念頭に置いて作成する必要があります。

次の擬似コードは、一般的にスパムメールを示していると理解されている単語のリストをチェックするコールアウトエージェントを作成する手順を示しています。エージェントは、任意のプログラミング言語で実装できます。疑似コードは、コールアウトエージェントを開発する際のガイドラインとしてのみ使用してください。プログラムに追加の機能を組み込むことができます。

疑似コードを使用して迷惑メールを識別するには

1. NetScaler アプライアンスから提供されたメールの件名を受け入れます。
2. 電子メールの件名のチェック対象となるすべての用語を含むデータベースに接続します。
3. メールの件名の単語をスパムワードリストと照合してください。
4. HTTP コールアウトの要求に応じて応答をフォーマットします。
5. 応答を NetScaler アプライアンスに送信します。

ユースケース**:** 動的コンテンツの切り替え

August 15, 2023

このユースケースでは、HTTP コールアウトを使用して要求の転送先となる負荷分散仮想サーバーの名前を取得することで、コンテンツを動的に切り替えることができます。

1. コンテンツスイッチ仮想サーバーを追加します。

```
add cs vserver cs_vserver1 HTTP 10.102.29.196 80
<!--NeedCopy-->
```


2. HTTP コールアウトを作成します。

```
add policy httpCallout http_callout1
<!--NeedCopy-->
```

3. HTTP ヘッダー「X-CLIENT-IP」にクライアント IP アドレスを含む要求からの負荷分散仮想サーバーの名前で応答するように HTTP コールアウトを設定します。

```
> set policy httpCallout http_callout1 -IPAddress 10.217.14.23 -
    port 80 -returnType TEXT -hostExpr ""www.get-lbvip.com"" -
    urlStemExpr ""/index.html"" -headers X-CLIENT-IP(CLIENT.IP.SRC)
     -resultExpr "HTTP.RES.BODY(1000).AFTER_STR("<lbvip>").
    BEFORE_STR("<lbvip")"
<!--NeedCopy-->
```

4. コールアウト応答を取得するようにコンテンツスイッチングアクションを設定します。

```
add cs action cs_action1 -targetVserverExpr 'SYS.HTTP_CALLOUT(
    http_callout1)'
<!--NeedCopy-->
```

> 注:
>
> 負荷分散仮想サーバーをコンテンツスイッチ仮想サーバーにバインドして、次の点を考慮する必要があります。

- コールアウトの解決先となる負荷分散仮想サーバーが使用できない。
- コールアウトを実行した結果生じる UNDEF 条件。

```
> bind cs vserver cs_vserver1 -lbvserver default_lbvip
<!--NeedCopy-->
```

5. コンテンツスイッチングポリシーを設定します。

```
add cs policy cs_policy1 -rule true -action cs_action1
<!--NeedCopy-->
```

6. コンテンツスイッチングポリシーをコンテンツスイッチング仮想サーバーにバインドします。

```
bind cs vserver cs_vserver1 -policyName cs_policy1 -priority 10
<!--NeedCopy-->
```

パターンセットとデータセット

August 15, 2023

多数の文字列パターンに対する文字列照合操作のポリシー式は、長くて複雑になる傾向があります。このような複雑な式の評価によって消費されるリソースは、処理サイクル、メモリ、および構成サイズの点で重要です。パターンマッチングを使用すると、より単純でリソース消費の少ない式を作成できます。

照合するパターンのタイプに応じて、次のいずれかの機能を使用してパターンマッチングを実装できます。

- パターンセットは、デフォルトの構文ポリシー評価時に文字列マッチングに使用されるインデックス付きパターンの配列です。パターンセットの例: イメージタイプ {svg、bmp、PNG、GIF、tiff、jpg}。
- データセットは、パターンセットの特殊な形式です。これは、型数（整数）、IPv4 アドレス、または IPv6 アドレスのパターンの配列です。

patsetとdatasetの違いは、datasetでは境界条件を比較する点です。たとえば、入力文字列が 1.1.1.11 で、1.1.1.1 パターンが IPv4 タイプのpatsetとdatasetにバインドされていると仮定すると、リクエストに IP アドレスが存在するかどうかを確認するようにpatsetとデータセットが設定されます。評価後、patsetは入力に 1.1.1.1 が存在するが、dataset評価が false であることを返します。これは、IP アドレスが他の IP アドレスの一部ではない境界チェックインが原因です。つまり、束縛されたパターンの後には整数があってはならないということです。

多くの場合、パターンセットとデータセットのどちらでも使用できます。ただし、数値データまたは IPv4 アドレスと IPv6 アドレスに特定の一致が必要な場合は、データセットを使用する必要があります。

> 注:
>
> - パターンセットとデータセットは、デフォルトの構文ポリシーでのみ使用できます。
> - リリース 13.1 ビルド 42.x 以降では、50000 個のパターンをパターンセットにバインドできます。パターンセットファイルでは、パターンセットにバインドできるパターンは 10000 個だけです。また、パターンセットをストリーミングで使用する場合、そのパターンセットにバインドできるパターンは 5000 個だけです。ストリーミング用のパターンセットは、リライトアクションの検索パラメータ、HTTP 本文、または TCP ペイロードベースの式で使用されます。

パターンセットとデータセットでの文字列マッチングの仕組み

August 15, 2023

パターンセットまたはデータセットには一連のパターンが含まれ、各パターンには固有のインデックスが割り当てられます。ポリシーをパケットに適用すると、評価する文字列が式で識別され、オペレータは、一致するものが見つかるか、すべてのパターンが比較されるまで、その文字列をパターンセットまたはデータセットで定義されているパターンと比較します。次に、演算子はその機能に応じて、一致するパターンが見つかったかどうかを示すブール値、または文字列に一致するパターンのインデックスのいずれかを返します。

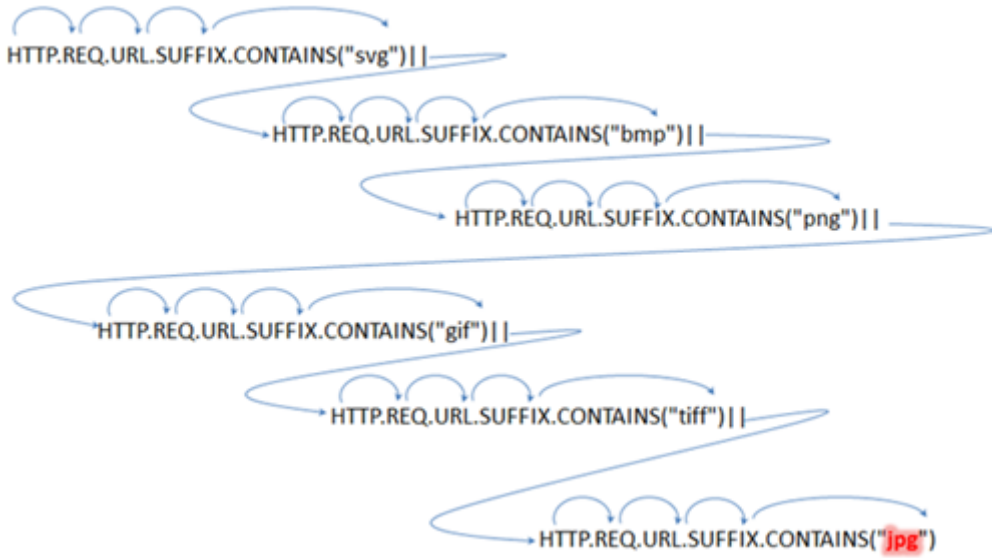
注意: このトピックでは、パターンセットの仕組みについて説明します。データセットも同じように機能します。パターンセットとデータセットの唯一の違いは、セットで定義されているパターンのタイプです。

文字列マッチングにパターンをどのように使用できるかを理解するには、次の使用事例を検討してください。

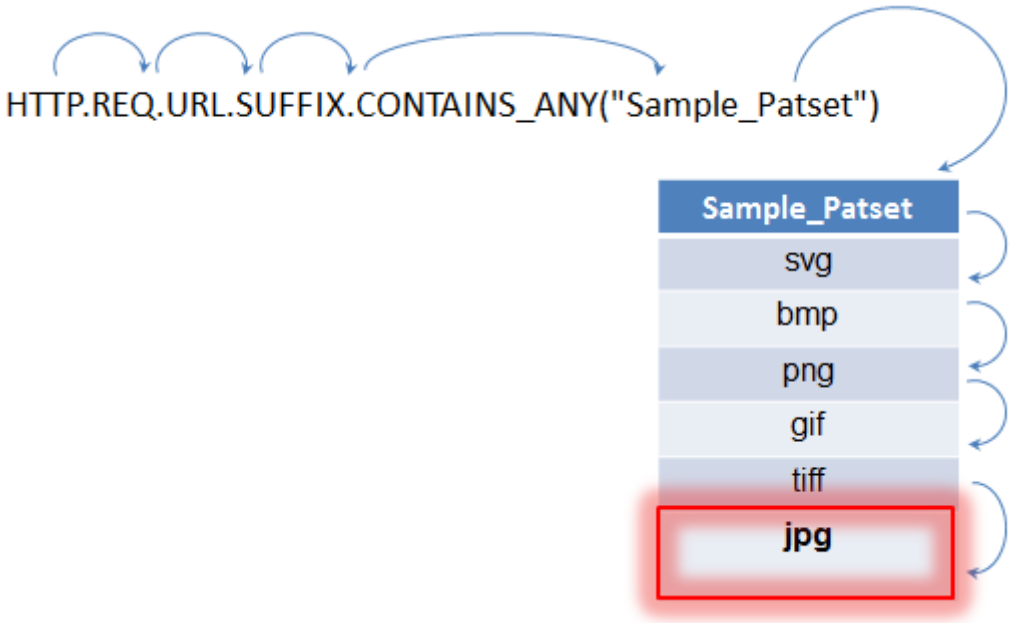
URL サフィックス (ターゲットテキスト) に画像ファイル拡張子が含まれているかどうかを確認する必要があります。パターンセットを使用しない場合、次のように複雑な式を定義する必要があります。

```
1 HTTP.REQ.URL.SUFFIX.CONTAINS("svg") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("
    bmp") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("png") ¦¦
2 HTTP.REQ.URL.SUFFIX.CONTAINS("gif") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("
    tiff") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("jpg")
3 <!--NeedCopy-->
```

URL に上記の複合式を含むサフィックスが「jpg」の場合、NetScaler アプライアンスはリクエストが jpg 画像を参照しているかどうかを判断するために、複合式全体をあるサブ式から次のサブ式へと順番に繰り返し処理する必要があります。次の図は、プロセスの手順を示しています。



複合式に何百ものサブ式が含まれる場合、上記の処理はリソースを大量に消費します。別の方法としては、次の図に示すように、パターンセットを呼び出す式があります。



前述のようにポリシーを評価する際、オペレータ (CONTAINS_ANY) は、一致するものが見つかるまで、リクエストで識別された文字列をパターンセットで定義されているパターンと比較します。Sample_Patset 式を使用すると、6 つのサブ式による複数回の反復が 1 回に減ります。

複数の OR 演算で文字列マッチングを実行する複合式を設定する必要がなくなるため、パターンセットまたはデータセットは構成を簡素化し、要求と応答の処理を高速化します。

パターンセットの構成

August 15, 2023

パターンセットを設定するには、パターンとして使用する文字列を指定する必要があります。これらの各パターンに固有のインデックス値を手動で割り当てることも、インデックス値を自動的に割り当てることもできます。

> 注:
>
> パターンセットでは大文字と小文字が区別されます (大文字と小文字を無視する式を指定しない限り)。そのため、たとえば文字列パターン「product1」は文字列パターン「Product1」と同じではありません。

インデックス値について覚えておくべきポイント:

- 同じインデックス値を複数のパターンにバインドすることはできません。
- 自動的に割り当てられるインデックス値は、パターンセット内の既存のパターンの最大インデックス値よりも 1 つ大きい数値です。たとえば、パターンセット内の既存のパターンの最大インデックス値が 104 の場合、次に自動的に割り当てられるインデックス値は 105 です。

- 最初のパターンにインデックスを指定しない場合、インデックス値 1 がそのパターンに自動的に割り当てられます。
- 1 つ以上のパターンが削除または変更されても、インデックス値は自動的に再生成されません。たとえば、セットにインデックスが 1 ～5 の 5 つのパターンが含まれていて、インデックスが 3 のパターンが削除されても、パターンセット内の他のインデックス値は自動的に 1 ～4 の値を生成するように再生成されません。
- パターンに割り当てることができる最大インデックス値は 4294967290 です。その値がセット内のパターンにすでに割り当てられている場合は、新しく追加されたパターンに手動でインデックス値を割り当てる必要があります。現在使用されている値よりも低い未使用のインデックス値を自動的に割り当てることはできません。

コマンドラインインターフェイスを使用してパターンセットを設定します

コマンドプロンプトで、次の操作を行います。

1. パターンセットを作成します。

   ```
   add policy patset <name>
   ```

例:

```
add policy patset samplepatset
```

1. パターンをパターンセットにバインドします。

   ```
   bind policy patset <name> <string> [-index <positive_integer>][-
   charset ( ASCII | UTF_8 )] [-comment <string>]
   ```

例:

```
bind policy patset samplepatset product1 -index 1 -comment short
description about the pattern bound to the pattern set
```

注意: パターンセットにバインドしたいすべてのパターンについて、この手順を繰り返します。

1. 設定を確認します。

```
show policy patset <name>
```

設定ユーティリティを使用してパターンセットを設定します

1. 「**AppExpert** 」>「パターンセット」に移動します。
2. 詳細ペインで、[追加] をクリックして [パターンセットの作成] ダイアログボックスを開きます。
3. 名前テキストボックスにパターンセットの名前を指定します。
4. 「パターンを指定」で、最初のパターンを入力し、オプションで次のパラメータの値を指定します。

   - バックスラッシュをエスケープ文字として扱う-このチェックボックスをオンにすると、パターンに含まれるバックスラッシュ文字はすべてエスケープ文字として処理されます。

- インデックス-1 から 4294967290 までの範囲でユーザーが割り当てたインデックス値。

5. 正しい文字を入力したことを確認し、[追加] をクリックします。
6. 手順 4 と 5 を繰り返してパターンを追加し、「作成」をクリックします。

ファイルベースのパターンセットの設定

NetScaler アプライアンスはファイルベースのパターンセットをサポートしています。

CLI を使用してファイルベースのパターンセットを設定する

コマンドプロンプトで、次のコマンドを入力します。

- 新しいパターンセットファイルを NetScaler アプライアンスにインポートします。

```
import policy patsetfile <src> <name> -delimiter <char> -charset
   <ASCII | UTF_8>
<!--NeedCopy-->
```

例:

```
import policy patsetfile local:test.csv clientids_list -
   delimiter ,
<!--NeedCopy-->
```

ローカルデバイス、HTTP サーバー、または FTP サーバーからファイルをインポートできます。ローカルデバイスからファイルを追加するには、ファイルが/var/tmpの場所にある必要があります。

- パターンセットファイルをパケットエンジンに追加します。

```
add policy patsetfile <patset filename>
<!--NeedCopy-->
```

例:

```
add policy patsetfile clientids_list
<!--NeedCopy-->
```

- NetScaler アプライアンス上の既存のパターンセットファイルを更新します。

```
update policy patsetfile <patset filename>
<!--NeedCopy-->
```

例:

```
update policy patsetfile clientids_list
<!--NeedCopy-->
```

- パターンをパターンセットにバインドします。


```
add policy patset <patset name> -patsetfile <patset filename>
<!--NeedCopy-->
```

例:

```
add policy patset clientid_patset -patsetfile clientids_list
<!--NeedCopy-->
```

- 設定を確認します。

```
show policy patsetfile clientids_list

Name: clientids_list
Patset Name: clientid_patset
Number of Imported Patterns: 8
Number of Bound Patterns: 8
(All the patterns bound successfully)

Done
<!--NeedCopy-->
```

GUI を使用してファイルベースのパターンセットを設定

1. **AppExpert->** パターンセットファイルに移動します。
2. [インポート] ペインで、[インポート] をクリックします。
3. 「ポリシーパッチセットファイルの設定」ページで、インポートするファイルを選択して「**OK**」をクリックします。
4. インポートしたファイルを選択して、[追加] をクリックします。
5. 「ポリシー・パット・セット・ファイルの作成」ページで詳細を入力し、「作成」をクリックしてポリシー・パターン・セットを追加します。

データセットの構成

August 15, 2023

データセットを構成するには、パターンとして処理する文字列を指定し、タイプ (数値、IPv4 アドレス、または IPv6 アドレス) を割り当て、データセット範囲を構成する必要があります。パターンには一意のインデックス値を手動で割り当てることも、インデックス値を自動的に割り当てることもできます。データセットは HTTP や 7 層プロトコルとは関係ありません。テキストまたは文字列に対してのみ機能します。Num、ULONG、IPv4、IPv6、MAC、DOUBLE など、さまざまなタイプのデータセットがあります。タイプを選択し、指定したタイプに基づいてデータセットの範囲を定義できます。

> 注:
>
> ポリシーデータセットでは大文字と小文字が区別されます (大文字と小文字を無視する式を指定しない限り)。したがって、たとえば MAC アドレス ff: ff: ff: ff: ff: ff: ff は MAC アドレス FF: FF: FF: FF: FF: FF と同じではありません。

データセットのインデックス値に適用されるルールは、パターンセットに似ています。インデックス値の詳細については、パターンセットの構成を参照してください。

データセットの構成

データセットを設定するには、次の手順を実行します。

1. ポリシーデータセットの追加
2. パターンをポリシーデータセットにバインドする
3. ポリシー式を追加する
4. ポリシー設定の検証

ポリシーデータセットの追加

コマンドプロンプトで、次の操作を行います。

```
add policy dataset <name> <type>
```

例:

```
add policy dataset ds1 ipv4 -comment numbers
```

パターンをデータセットにバインドする

コマンドプロンプトで入力します。

```
bind policy dataset <name> <value> [-index <positive_integer>] [-
endRange <string>] [-comment <string>]
```

例:

```
bind policy dataset ds1 1.1.1.1 -endRange 1.1.1.10 -comment short
description about the pattern bound to the data set
```

> 注:
>
> データセットにバインドするすべてのパターンに対して、この手順を繰り返す必要があります。1 つのデータセットにバインドできるパターンは 5000 個までです。
>
> また、データセット範囲は、データセットにバインドされた他の範囲と重複してはならず、データセットにバイ

> ンドされた単一の値を含めることはできません。範囲が重複するデータセットをバインドすると、エラーが発生します。

例:

```
1 add policy dataset ip_set ipv4
2  Done
3 bind policy dataset ip_set 2.2.2.25
4  Done
5 bind policy dataset ip_set 2.2.2.20 -endRange 2.2.2.30
6 ERROR: The range overlaps an existing range or includes a value bound
     to the dataset.
7 <!--NeedCopy-->
```

データセットにバインドされた範囲で、その値がデータセットにバインドされた単一の値と等しいか、小さい値と大きい値 (小さい値 <= 値 && 値 <-上限値) の間にある場合、その値はデータセット内にあると見なされます。

ポリシーデータセットでポリシー式を使用する

コマンドプロンプトで入力します。

```
add policy expression exp1 http.req.body(100).contains_any("ds1")
```

ここで、
式は、データセット ds1 にバインドされたパターン (または範囲内のパターン) が HTTP リクエスト本文の最初の 100 バイトに存在するかどうかをチェックします。

データセット設定の検証

コマンドプロンプトで入力します。

```
show policy dataset ds1
> show policy dataset ds1
```

例:

```
1         Dataset:        ds1
2          Type:   IPV4
3 1)       Bound Dataset Range from: 1.1.1.1       through: 1.1.1.10
            Index:  1
4 <!--NeedCopy-->
```

構成ユーティリティを使用してデータセットを構成する

ポリシーデータセットを設定するには、以下の手順に従います。

1. **AppExpert >** データセットに移動します。

2. 詳細ウィンドウの [データセット] で、[追加] をクリックします。
3. **[**データセットの構成] ページで、次のパラメーターを設定します。

    a) [名前]。ポリシーデータセットの名前。
    b) 種類。データセットにバインドする値の型。

    データセットの構成

4. [挿入] をクリックして、特定のタイプのデータセット値をバインドします。

    a) 価値。データセットに関連付けられた、指定された型の値。
    b) インデックス。データセットのインデックス値。
    c) 終了範囲。データセットエントリ。これは<value>から<end_range>への範囲です。
    d) [コメント]。データセットに関する簡単な説明。

    データセットバインディング

5. [** 挿入して閉じる **] をクリックします。
6. コメントを入力します。
7. **[**作成**]** して **[**閉じる**]** をクリックします。

ポリシーデータセットの **IPv4** および **IPv6** アドレスにおける **CIDR** サブネット表記

IPv4 および IPv6 アドレスのポリシーデータセットでは、バインドされた値を CIDR 表記法を使用してサブネットにすることができます。CIDR 表記法では、サブネットのアドレスと範囲を指定します。CIDR 表記<address>/<n>。<address> はサブネット内の最初のアドレス、<n> はサブネットの範囲を定義するサブネットマスクに設定された左端のビット数を指定する整数です。

たとえば、192.128.0.0/10 は、アドレス 192.129.0.0 から始まり、マスク 0xFFC0000 (255.192.0.0) を持つ IPv4 サブネットを表します。

例**:**

```
add policy dataset ds1 ipv4
bind policy dataset ds1 192.128.0.0/10
show policy dataset ds1
    Dataset: ds1
    Type: IPV4
Bound Dataset Value: 192.128.0.0/10 Index: 1 Comment: Subnet range from
    192.128.0.0 through 192.191.255.255

<!--NeedCopy-->
```

このデータセットを式で使用する例を次に示します。

```
add responder policy resp_ipv4_pol client.ip.src.typecast_text_t.
   equals_any("ds1") drop
```



```
<!--NeedCopy-->
```

IPv6 サブネットの例**:**

IPv6 サブネットの例としては 2001: db 8:123:: /56 があります。このサブネットはアドレス 2001: db 8:123:: から始まり、マスク FFFF: FFFF: FFFF: FF00::

```
add policy dataset ds2 ipv6
bind policy dataset ds2 2001:db8:123::/56
show policy dataset ds2
    Dataset: ds2
    Type: IPV61
Bound Dataset Value: 2001:db8:123::/56 Index: 1 Comment: Subnet range
   from 2001:db8:123:: through 2001:db8:123:ff:ffff:ffff:ffff:ffff

<!--NeedCopy-->
```

サブネットの開始アドレスは、サブネットマスクでマスクされた指定アドレスによって決定されます。指定したアドレスが結果の開始アドレスと一致しない場合、警告が発行されます。

例**:**

```
bind policy dataset ds1 192.168.0.0/10
Warning: Starting subnet address masked using subnet mask to create new
    starting address [192.128.0.0]
show policy dataset ds1
    Dataset: ds1
    Type: IPV4
Bound Dataset Value:192.168.0.0/10 Index: 1 Comment: Subnet range from
   192.128.0.0 through 192.191.255.255

<!--NeedCopy-->
```

このデータセットを式で使用する例を次に示します。

```
add responder policy resp_ipv6_pol client.ipv6.src.typecast_text_t.
   equals_any("ds2") drop
<!--NeedCopy-->
```

パターンセットとデータセットの使用

August 15, 2023

パターンセットまたはデータセットを引数として取る高度なポリシー式を使用して、文字列照合操作を実行できます。

使用法は次のとおりです。


```
<text>.<operator>("<name>")
<!--NeedCopy-->
```

各項目の意味は次のとおりです。

- <text>は、パケット内の文字列を識別する式です。例: HTTP.REQ.HEADER(“Host”).
- <operator>は、パターンセットタイプ表 pdf で説明されている演算子の 1 つです。

使用例については、サンプル使用法を参照してください。

使用サンプル

August 15, 2023

式でのパターンセットの使用方法を理解するために、「imagetypes」という名前のパターンセットの例を考えてみましょう。

パターン	インデックス値
svg	1
bmp	2
png	3
gif	4
TIFF	5
jpg	6

表 1. Pattern set “imagetypes”

例 **1:** HTTP リクエストのサフィックスが「imagetypes」パターンセットで定義されているファイル拡張子の 1 つであるかどうかを判断します。

- 式。HTTP.REQ.URL.SUFFIX.EQUALS_ANY(“imagetypes”)
- サンプル **URL**。http://www.example.com/homepageicon.jpg
- 結果。TRUE

例 **2:** HTTP リクエストのサフィックスが「imagetypes」パターンセットで定義されているファイル拡張子のいずれかであるかどうかを判断し、そのパターンのインデックスを返します。

- 式。HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”)
- サンプル **URL**。http://www.example.com/mylogo.png

- 結果。 4 (パターン「gif」のインデックス値)

例 **3:** パターンのインデックス値を使用して、URL サフィックスが指定されたインデックス値の範囲内にあるかどうかを判断します。

- 式。HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).GE(3) && HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).LE(5)
- サンプル **URL**。 `http://www.example.com/mylogo.png`
- 結果。TRUE (gif ファイルタイプのインデックス値は 4 です。)

例 **4:** ファイル拡張子 bmp、jpg、png には 1 つのポリシーセットを実装し、gif、tiff、および svg ファイルには別のポリシーセットを実装します。

一致したパターンのインデックスを返す式を使用して、Web アプリケーションのトラフィックサブセットを定義できます。コンテンツスイッチング仮想サーバーのコンテンツスイッチングポリシーでは、次の 2 つの式を使用できます。

- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).LE(3)
- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).GE(4)

変数

August 15, 2023

変数は、情報をトークンの形式で保存する名前付きオブジェクトです。これらのトークンは、NetScaler ADC アプライアンス上のさまざまなトランザクション内およびトランザクション全体で、内部計算とポリシー処理に使用されます。

NetScaler ADC アプライアンスは、次のタイプの変数の作成をサポートしています。

- シングルトン変数。ulong と text (最大サイズ) のいずれかのタイプの 1 つの値を指定できます。ulong 型は符号なし 64 ビット整数、text 型はバイト列、max-size はシーケンス内の最大バイト数です。
- 変数をマップします。マップにはキーに関連付けられた値が格納されます。各キーと値のペアはマップエントリと呼ばれます。各エントリのキーは、マップ内で一意です。マップは次のように指定されます。

  マップ (キータイプ、値タイプ、最大値)。

  各項目の意味は次のとおりです。

  - *key_type* はキーのデータタイプです。テキストタイプ (最大サイズ) です。
  - *value_type* はマップの値のデータタイプです。ロングタイプでもテキストタイプ (最大サイズ) でもかまいません。
  - *max-values* は、マップに含めることができるエントリの最大数です。ウロン型です。

これらの変数の値は、ポリシーアクションで呼び出す必要がある割り当てを使用して設定されます。

変数スコープ

マップ変数またはシングルトン変数にはグローバルスコープを設定できます。あるいは、シングルトン変数のスコープを 1 つのトランザクションに制限することもできます。

- グローバルスコープ変数 -グローバルスコープ（デフォルト）の変数にはインスタンスが 1 つしかなく、そのインスタンスは NetScaler ADC アプライアンスのすべてのコアとクラスタまたは HA 構成のすべてのノードで同じ値を持ちます。グローバル変数値は、明示的に削除されるか、有効期限が切れるか、スタンドアロンアプライアンスが再起動されるか、クラスターまたは HA 構成のすべてのノードが再起動されるまで存在します。
- トランザクションスコープ変数 -トランザクションスコープを持つ変数には、NetScaler ADC アプライアンスによって処理されるトランザクションごとに、独自の値を持つ個別のインスタンスがあります。トランザクション処理が完了すると、トランザクション変数の値は削除されます。

> 注：トランザクションスコープ変数は、NetScaler リリース 10.5.e 以降で使用できます。

変数の構成と使用

August 15, 2023

最初に変数を作成してから値を割り当てるか、変数に対して実行する操作を指定する必要があります。これらの操作を実行した後は、割り当てをポリシーアクションとして使用できます。

注: 一度設定すると、変数の設定を変更またはリセットすることはできません。変数を変更する必要がある場合は、変数とその変数へのすべての参照 (式と代入) を削除する必要があります。その後、変数を新しい設定で再追加したり、参照 (式と代入) を再度追加したりできます。

コマンドラインインターフェイスを使用して変数を設定するには

1. 変数を作成します。

```
add ns variable <name> -type <string> [-scope global] [-ifFull ( undef
    | lru )] [-ifValueTooBig ( undef | truncate )] [-ifNoValue ( undef |
     init )] [-init <string>] [-expires <positive_integer>] [-comment <
    string>]
<!--NeedCopy-->
```

注: コマンドパラメータの説明については、マニュアルページ「man add ns variable」を参照してください。

例 1:「my_counter」という名前の ulong 変数を作成し、それを 1 に初期化します。


```
add ns variable my_counter  - type ulong -init 1
<!--NeedCopy-->
```

例 2:「user_privilege_map」という名前のマップを作成します。マップには、最大長が 15 文字のキーと最大長 10 文字のテキスト値、最大 10000 エントリが含まれます。

```
add ns variable user_privilege_map -type map(text(15),text(10),10000)
<!--NeedCopy-->
```

注: マップに期限切れでないエントリが 10000 個含まれている場合、新しいキーの割り当てでは、最も使用頻度の低いエントリの 1 つが再使用されます。デフォルトでは、存在しないキーの値を取得しようとする式は、空のテキスト値を初期化します。

値を割り当てるか、変数に対して実行する操作を指定します。これは課題を作成することによって行われます。

```
add ns assignment <name> -variable <expression> [-set <expression> | -
    add <expression> | -sub <expression> | -append <expression> | -clear
    ] [-comment <string>]
<!--NeedCopy-->
```

注: 変数は、変数セレクター ($) を使用して参照されます。そのため、
$variable1 はテキスト変数や ulong 変数を参照するのに使われます。同様に、
$variable2 [キー表現**]** はマップ変数を参照するために使用されます。

例 1:「inc_my_counter」という名前の割り当てを定義すると、自動的に「my_counter」変数に 1 が追加されます。

```
add ns assignment inc_my_counter -variable $my_counter -add 1
<!--NeedCopy-->
```

例 2:「set_user_privilege」という名前の割り当てを定義して、「user_privilege_map」変数に、「get_user_privilege」HTTP コールアウトによって返される値を含むクライアントの IP アドレスのエントリを追加します。

```
add ns assignment set_user_privilege -variable $user_privilege_map[
    client.ip.src.typecast_text_t] -set sys.http.callout(
    get_user_privilege)
<!--NeedCopy-->
```

注: そのキーのエントリが既に存在する場合、値は置き換えられます。それ以外の場合は、キーと値の新しいエントリが追加されます。以前の user_privilege_map の宣言によると、マップにすでに 10000 個のエントリがある場合、最も使用頻度の低いエントリの 1 つが新しいキーと値に再利用されます。

1. ポリシー内の変数割り当てを呼び出します。

   マップ変数を操作できる関数は 2 つあります。

   - **$name.valueExists(key-expression).** キーエクスプレッションで選択されたマップに値がある場合は true を返します。それ以外の場合は false を返します。この関数は、マップエントリが存在する場

合は有効期限と LRU 情報を更新しますが、値が存在しない場合は新しいマップエントリを作成しません。

- **$name.valueCount.** 変数が現在保持している値の数を返します。これはマップ内のエントリの数です。シングルトン変数の場合、変数が初期化されていない場合は 0、そうでない場合は 1 です。

例**:** 圧縮ポリシーを使用して「set_user_privilege」という名前のアサインメントを呼び出します。

```
add cmp policy set_user_privilege_pol -rule $user_privilege_map.
    valueExists(client.ip.src.typecast_text_t).not -resAction
    set_user_privilege
<!--NeedCopy-->
```

レスポンス側に **HTTP** ヘッダーを挿入するユースケース

次の例は、シングルトン変数の例を示しています。

テキスト型のシングルトン変数を追加します。この変数は最大 100 バイトのデータを保持できます。

```
add ns variable http_req_data -type text(100) -scope transaction
<!--NeedCopy-->
```

HTTP リクエストデータを変数に保存するために使用される割り当てアクションを追加します。

```
add ns assignment set_http_req_data -variable $http_req_data -set http.
    req.body(100)
<!--NeedCopy-->
```

変数から値を取得する HTTP ヘッダを挿入する書き換えアクションを追加します。

```
add rewrite action act_ins_header insert_http_header user_name
    $http_req_data.after_str("user_name").before_str("password")
<!--NeedCopy-->
```

リクエスト時に評価し、データを保存するための割り当てアクションを実行するリライトポリシーを追加します。このポリシーに該当すると、代入アクションが実行され、データが ns 変数 (http_req_data) に保存されます。

```
add rewrite policy pol_set_variable true set_http_req_data

bind rewrite global pol_set_variable 10 -type req_dEFAULT
<!--NeedCopy-->
```

応答時間を評価する書き換えポリシーを追加し、応答に HTTP ヘッダーを追加します。

```
add rewrite policy pol_ins_header true act_ins_header

bind rewrite global pol_ins_header 10 -type res_dEFAULT
<!--NeedCopy-->
```


割り当てアクション

NetScaler アプライアンスでは、ポリシールールが true と評価されると、ポリシーにバインドされた割り当てアクションがトリガーされます。アクションは変数内の値を更新し、それ以降のポリシー・ルール評価で使用できます。これにより、同じ変数を更新して、同じ機能での今後のポリシー評価に使用できます。以前は、関連する割り当てアクションのポリシーが true と評価された場合、アプライアンスは機能内のすべてのポリシーを評価した後にのみ割り当てアクションを実行していました。そのため、割り当てアクションによって設定された変数値は、その後の機能内のポリシールール評価では使用できません。

この機能は、NetScaler アプライアンス上のクライアントのアクセスリストを制御するユースケースの方がよく理解できます。アクセスの決定は別のウェブサービスによって行われ、リクエスト`GET /client-access?<client-IP-address>`の本文に「BLOCK」または「ALLOW」を含む応答が返されます。HTTP コールアウトは、受信リクエストに関連付けられているクライアントの IP アドレスを含むように設定されます。NetScaler アプライアンスがクライアントから要求を受信すると、アプライアンスはコールアウト要求を生成してコールアウトサーバーに送信します。コールアウトサーバーは、ブラックリストに登録された IP アドレスのデータベースと、クライアントの IP アドレスがデータベースにリストされているかどうかをチェックする HTTP コールアウトエージェントをホストします。HTTP コールアウトエージェントはコールアウトリクエストを受信し、クライアントの IP アドレスがリストされているかどうかを確認し、応答を送信します。レスポンスは、ステータスコード 200、302 で、本文には「ブロック」または「許可」があります。ステータスコードに基づいて、アプライアンスはポリシー評価を実行します。ポリシー評価が true の場合、割り当てアクションがすぐにトリガーされ、アクションによって変数に値が設定されます。アプライアンスはこの変数値を使用して設定し、同じモジュールでの今後のポリシー評価に使用します。

割り当てアクションの設定のユースケース

以下の手順に従って割り当てアクションを設定し、後続のポリシーで変数を使用してください。

1. アクセスの決定は別の Web サービスによって行われ、リクエストの本文には BLOCK または ALLOW を含む応答が返されます。

   ```
   GET /url-service>/url-allowed?<URL path>
   ```

2. URL へのアクセス決定を格納するマップ変数を設定します。

   ```
   add ns variable url_list_map -type 'map(text(1000),text(10)
   ,10000)'
   ```

3. HTTP コールアウトを設定して Web サービスにアクセス要求を送信します。

   ```
   add policy httpCallout url_list_callout -vserver url_vs -returnType
    TEXT -urlStemExpr '"/url-allowed?" + HTTP.REQ.URL.PATH'-resultExpr
    'HTTP.RES.BODY(10)'
   ```

4. 割り当てアクションを設定して、コールアウトを呼び出してアクセス決定を取得し、URL のマップエントリに割り当てます。


```
add ns assignment client_access_assn -variable '$client_access_map
[CLIENT.IP.SRC.TYPECAST_TEXT_T]'-set SYS.HTTP_CALLOUT(client_access_callou
)
```

5. URL リクエストがブロックされた場合に 403 レスポンスを送信するようにレスポンダーアクションを設定します。

```
add responder action url_list_block_act respondwith '"HTTP/1.1
403 Forbidden\\r\\n\\r\\n"'
```

6. URL のマップエントリがまだ設定されていない場合は、レスポンダーポリシーを設定して設定します。即時アクション拡張では、このポリシーが評価されるときにマップエントリの値が設定されます。機能強化前は、すべてのレスポンダーポリシーが評価され、別のウェブサービスによって決定されるまで、割り当ては行われませんでした。

```
add responder policy url_list_assn_pol '!$url_list_map.VALUEEXISTS
(HTTP.REQ.URL.PATH)'url_list_assn
```

7. マップエントリの値が BLOCK の場合、URL へのアクセスをブロックするレスポンダーポリシーを設定します。即時アクションの強化により、前のポリシーで設定されたマップエントリをこのポリシーで使用できます。拡張前は、マップエントリはこの時点ではまだ未設定でした。

```
add responder policy client_access_block_pol '$client_access_map[
CLIENT.IP.SRC.TYPECAST_TEXT_T] == "BLOCK"'client_access_block_act
```

8. レスポンダーポリシーを仮想サーバーにバインドします。注: ポリシーをグローバルにバインドすることはできません。別の仮想サーバーで HTTP コールアウトを実行したくないからです。

```
bind lb vserver vs -policyName client_access_assn_pol -priority
10 -gotoPriorityExpression NEXT -type REQUEST
bind lb vserver vs -policyName client_access_block_pol -priority
20 -gotoPriorityExpression END -type REQUEST
```

設定ユーティリティを使用して変数を設定するには

1. 「**AppExpert**」>「**NS** 変数」に移動して、変数を作成します。
2. 「**AppExpert**」>「**NS** 割り当て」に移動して、変数に値を割り当てます。
3. 割り当てをアクションとして構成する適切な機能領域に移動します。

ユースケース: ユーザー権限のキャッシュ

August 15, 2023

このユースケースでは、ユーザー権限 (「GOLD」、「SILVER」など) を外部 Web サービスから取得する必要があります。

このユースケースを実現するには、次の操作を実行してください

HTTP コールアウトを作成して、外部 Web サービスからユーザー権限を取得します。

```
add policy httpcallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port <
    port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod (
    GET | POST )] [-hostExpr <string>] [-urlStemExpr <string>] [-headers
     <name(value)> ...] [-parameters <name(value)> ...] [-bodyExpr <
    string>][-fullReqExpr <string>] [-scheme ( http | https )] [-
    resultExpr <string>] [-cacheForSecs <secs>] [-comment <string>]

add policy httpcallout get_user_privilege -ipaddress 10.217.193.84 -
    port 80 -returnType text -httpMethod GET -hostExpr '"/
    get_user_privilege"' -resultExpr 'http.res.body(5)'
<!--NeedCopy-->
```

権限を変数に保存します。

```
add ns variable <name> -type <string> [-scope ( global | transaction )
    ][-ifFull ( undef | lru )] [-ifValueTooBig ( undef | truncate )][-
    ifNoValue ( undef | init )] [-init <string>] [-expires <
    positive_integer>] [-comment <string>]

add ns variable user_privilege_map -type map(text(15),text(10),10000) -
    expires 1200

add ns assignment set_user_privilege -variable $user_privilege_map[
    client.ip.src] -set sys.http_callout(get_user_privilege)
<!--NeedCopy-->
```

ポリシーを作成して、クライアントの IP アドレスのエントリが既にキャッシュされているかどうかを確認します。存在しない場合は、HTTP コールアウトを呼び出して、クライアントのマップエントリを設定します。

```
add cmp policy <name> -rule <expression> -resAction <string>

add cmp policy set_user_privilege_pol -rule $user_privilege_map.
    valueExists(client.ip.src).not -resAction set_user_privilege>
<!--NeedCopy-->
```

クライアントのキャッシュされた権限エントリが「GOLD」の場合に圧縮するポリシーを作成します。

```
add cmp policy <name> -rule <expression> -resAction <string>

add cmp policy compress_if_gold_privilege_pol -rule '
    $user_privilege_map[client.ip.src].eq("GOLD")' -resAction compress
<!--NeedCopy-->
```

圧縮ポリシーをグローバルにバインドします。


```
bind cmp global <policyName> [-priority <positive_integer>] [-state (
    ENABLED | DISABLED )] [-gotoPriorityExpression <expression>] [-type
    <type>] [-invoke (<labelType> <labelName>) ]

bind cmp global set_user_privilege_pol -priority 10 NEXT

bind cmp global compress_if_gold_privilege_pol -priority 20 END
<!--NeedCopy-->
```

ユースケース: セッション数の制限

August 15, 2023

このユースケースでは、要件はアクティブなバックエンドセッションの数を制限することです。展開では、各セッションログインの URL にはログインがあり、各セッションログアウトには URL にログアウトがあります。ログインが成功すると、バックエンドは一意の 10 文字の値を持つセッション IDCookie を設定します。

このユースケースを実現するには、次の操作を実行します。

1. アクティブな各セッションを格納できるマップ変数を作成します。マップのキーはセッション ID です。変数の有効期限は 600 秒 (10 分) に設定されています。

   ```
   > add ns variable session_map -type map(text(10),ulong,100) -
     expires 600
   <!--NeedCopy-->
   ```

2. map 変数に次の代入を作成します。

   - セッション ID のエントリを作成し、その値を 1 に設定します (この値は使用されません)。

     ```
     > add ns assignment add_session -variable '$session_map[http.
       req.cookie.value("sessionid")]' -set 1
     <!--NeedCopy-->
     ```

   - セッション ID のエントリを解放します。これにより、session_map の値カウントが暗黙的に減少します。

     ```
     > add ns assignment delete_session -variable '$session_map[
       http.req.cookie.value("sessionid")]' -clear
     <!--NeedCopy-->
     ```

3. 次のレスポンダーポリシーを作成します。

   - HTTP リクエストにそのセッション ID のマップエントリが存在するかどうかを確認する。マップエントリが存在しない場合、add_session 割り当てが実行されます。


```
> add responder policy add_session_pol 'http.req.url.contains
    ("example") || $session_map.valueExists(http.req.cookie.
    value("abc"))' add_session
<!--NeedCopy-->
```

注:

add_session_pol ポリシーの

valueExists () 関数は、セッションのマップエントリへの参照としてカウントされるため、各リクエストはそのセッションの有効期限タイムアウトをリセットします。10 分経過してもセッションの要求が受信されない場合、セッションのエントリの割り当ては解除されます。

- セッションがいつログアウトしたかを確認する。delete_session 割り当てが実行されます。

```
add responder policy delete_session_pol "http.req.url.
    contains("Logout")" delete_session
<!--NeedCopy-->
```

- ログイン要求があるかどうか、およびアクティブなセッションの数が 100 を超えているかどうかを確認する。これらの条件が満たされると、セッション数を制限するために、ユーザーはサーバーがビジーであることを示すページにリダイレクトされます。

```
add responder action redirect_too_busy redirect "/too_busy.
    html"
add responder policy check_login_pol "http.req.url.contains("
    example") && $session_map.valueCount > 100"
    redirect_too_busy
<!--NeedCopy-->
```

4. レスポンダーポリシーをグローバルにバインドします。

```
bind responder global add_session_pol 30 next
bind responder global delete_session_pol 10
bind responder global check_login_pol 20
<!--NeedCopy-->
```

ポリシーと式

October 25, 2023

以下のトピックでは、NetScaler で高度なポリシーを構成するために必要な概念と参考情報を提供します。

NetScaler でサポートされているすべての高度なポリシー式については、「ポリシー式」を参照してください。

トピック	説明
ポリシーと式の概要	式、ポリシー、およびアクションの目的、およびさまざまな NetScaler ADC アプリケーションがそれらをどのように使用するかについて説明します。
高度なポリシーの設定	高度なポリシーの構造と、それらを個別に、またはポリシーバンクとして設定する方法について説明します。
高度な定義式の設定: はじめに	式の構文とセマンティクスについて説明し、式とポリシーの構成方法について簡単に説明します。
高度な式: テキストの評価	テキスト（HTTP POST 要求の本文やユーザ証明書の内容など）を操作するときに設定する式について説明します。
高度な式: 日付、時刻、および数値の操作	任意のタイプの数値データ（URL の長さ、クライアントの IP アドレス、HTTP 要求が送信された日時など）を操作するときに構成する式について説明します。
高度な式:HTTP、TCP、および UDP データの解析	IP アドレスと IPv6 アドレス、MAC アドレス、および HTTP および TCP トラフィックに固有のデータを解析するための式について説明します。
高度な式:SSL 証明書の解析	SSL トラフィックとクライアント証明書の式を設定する方法について説明します。たとえば、証明書または証明書の発行者の有効期限を取得する方法などです。
高度な表現:IP アドレスと MAC アドレス、スループット、VLAN ID	他の章で説明されていない他のクライアント関連またはサーバー関連のデータを処理するために使用できる式について説明します。
型キャストデータ	あるタイプのデータを別のタイプに変換する式について説明します。
正規表現	正規表現を高度な定義式の演算子の引数として渡す方法について説明します。
エクスプレッションリファレンス	高度なエクスプレッション引数のリファレンスです。
高度な表現とポリシーのまとめ例	独自の用途に合わせてカスタマイズできる高度な表現とポリシーの例 (クイックリファレンス形式とチュートリアル形式の両方)。
書き換え用の高度なポリシーのチュートリアル例	リライト機能で使用する高度なポリシーの例。
ポリシーのチュートリアル例	アプリケーションファイアウォールや SSL などの NetScaler 機能のポリシーの例。
Apache mod_rewrite ルールのアドバンスドポリシーへの移行	Apache HTTP Server の mod_rewrite エンジンを使用して作成された関数の例と、NetScaler の書き換えポリシーとレスポンダーポリシーに変換された後のこれらの関数の例。

ポリシーと式の概要

August 15, 2023

多くの NetScaler ADC 機能では、ポリシーによって機能がデータを評価する方法が制御されます。ポリシーは、ルールと呼ばれる論理式を使用してデータを評価し、評価に基づいて 1 つ以上のアクションを適用します。または、ポリシーでプロファイルを適用して、複雑なアクションを定義することもできます。

一部の NetScaler ADC 機能では、高度なポリシーを使用します。これは、古いクラシックポリシーよりも優れた機能を提供します。新しいリリースの NetScaler ADC ソフトウェアに移行し、高度なポリシーを使用する機能のクラシックポリシーを構成した場合は、ポリシーを高度なポリシーインフラストラクチャに手動で移行する必要があります。

高度なポリシーインフラストラクチャ

January 9, 2024

高度なポリシーインフラストラクチャを使用すると、多くのデータ（たとえば、HTTP リクエストの本文）を分析したり、ポリシールールでさまざまな操作を設定したりできます（たとえば、リクエストの本文のデータを HTTP ヘッダーに変換するなど）。ポリシーは、NetScaler 機能に関連する処理の特定のポイントにバインドする必要があります。バインドポイントは、ポリシーがいつ評価されるかを決定する要素の 1 つです。

高度なポリシーを使用する利点

高度なポリシーでは、クラスオブジェクトモデルに基づいて構築された強力な表現言語が使用され、さまざまな NetScaler 機能の動作をより簡単に構成できるようにするオプションがいくつか用意されています。高度なポリシーインフラストラクチャを使用すると、次のことが可能になります。

- レイヤ 2 ～7 のネットワークトラフィックのきめ細かな分析を実行します。
- HTTP または HTTPS リクエストまたはレスポンスのヘッダーまたは本体の任意の部分を評価します。
- 高度なポリシーインフラストラクチャがデフォルト、オーバーライド、および仮想サーバーレベルでサポートする複数のバインドポイントにポリシーをバインドします。
- パターンセット、ポリシーラベル、レート制限識別子、HTTP コールアウト、変数などの特別なツールを使用すると、複雑なユースケースに対してポリシーを効果的に設定できます。

また、構成ユーティリティは、高度なポリシーインフラストラクチャと式に対する堅牢な GUI サポートを拡張し、ネットワークプロトコルに関する知識が限られているユーザーでもポリシーをすばやく簡単に構成できるようにします。構成ユーティリティには、高度なポリシー用のポリシー評価機能も含まれています。この機能を使用すると、コミットする前に高度なポリシーを評価し、その動作をテストできるため、設定エラーのリスクが軽減されます。

詳細ポリシーの基本コンポーネント

次に、高度なポリシーの特性をいくつか示します。

- Name：各ポリシーには一意の名前があります。
- ルール。ルールは、NetScaler ADC 機能がトラフィックの一部または別のオブジェクトを評価できるようにする論理式です。たとえば、ルールにより、NetScaler ADC は、HTTP リクエストが特定の IP アドレスから発信されたかどうか、または HTTP リクエストの Cache-Control ヘッダーの値が「No-Cache」であるかどうかを判断できます。
- バインディング。NetScaler ADC が必要なときにポリシーを呼び出すことができるようにするには、ポリシーを 1 つ以上のバインドポイントに関連付ける（バインドする）。

  ポリシーは、グローバルにバインドすることも、仮想サーバにバインドすることもできます。詳細については、「ポリシーバインディングについて」を参照してください。
- 関連付けられたアクション。アクションは、ポリシーとは別のエンティティです。ポリシーの評価は、最終的に NetScaler ADC がアクションを実行します。

  たとえば、統合キャッシュ内のポリシーは、.png または.jpeg ファイルに対する HTTP 要求を識別できます。このポリシーに関連付けるアクションによって、これらのタイプの要求に対する応答がキャッシュから提供されることが決定されます。

一部の機能では、プロファイルと呼ばれるより複雑な一連の手順の一部としてアクションを設定します。

NetScaler ADC 機能の異なるポリシーの使用方法

NetScaler ADC は、操作にポリシーに依存するさまざまな機能をサポートしています。次の表は、NetScaler ADC 機能がポリシーをどのように使用するかをまとめたものです。

機能名	機能でのポリシーの使用方法
書き換え	提供する前に変更したいデータを特定するため。ポリシーは、データを変更するためのルールを提供します。たとえば、HTTP データを変更して、要求を新しいホームページ、新しいサーバー、または着信要求のアドレスに基づいて選択したサーバーにリダイレクトしたり、セキュリティ上の目的で応答のサーバー情報をマスクするようにデータを変更したりできます。URL トランスフォーマー関数は、URL を変換する必要があるかどうかを評価するために、HTTP トランザクションとテキストファイル内の URL を識別します。

機能名	機能でのポリシーの使用方法
レスポンダー	レスポンダー関数の動作を構成するには。レスポンダーポリシーは、1 つ以上の式で構成されるルールに基づいています。ルールは、リクエストがルールに一致した場合に実行されるアクションに関連付けられます。
コンテンツスイッチ	受信したリクエストの特性に基づいて、どのサーバーまたはサーバーグループが応答を処理するかを決定すること。要求特性には、デバイスタイプ、言語、Cookie、HTTP メソッド、コンテンツタイプ、および関連するキャッシュサーバが含まれます。
キャッシュリダイレクト	レスポンスがキャッシュまたはオリジンサーバーのどちらから提供されているかを判断する。
圧縮制御	どのタイプのトラフィックを圧縮する必要があるかを判断するため。
DNS	DNS 要求と応答のさまざまな部分を変更するには
VPN クライアントレスアクセス	NetScaler Gateway が認証、承認、監査、およびその他の機能を実行する方法を決定し、NetScaler Gateway を使用して一般的な Web アクセスの書き換えルールを定義する。
キャッシュ	キャッシュとオリジンサーバーのどちらから応答を返すかを決定するため。
URL 変換ポリシー	NetScaler が URL 変換プロファイルを使用して変換する必要がある要求と応答を選択すること。
アプリケーションファイアウォールポリシー	さまざまな種類の Web コンテンツに異なるフィルタールールを割り当てること。
承認	ウェブサイトの実際の構成に関する不必要な詳細を公開することなく、要求されたコンテンツへのアクセスを提供すること。
TM トラフィック	実行時のアプリケーショントラフィックの特性 (接続タイムアウト、シングルサインオン、ログアウトの開始など) を設定します。
TM セッション	ユーザーが承認、承認、およびアカウンティング仮想サーバーにログオンした後にユーザーセッションをカスタマイズする。

機能名	機能でのポリシーの使用方法
SSL ポリシー	リクエストに対して実行されるコントロールまたはデータアクションを定義すること。そのため、SSL ポリシーは、制御ポリシーとデータポリシーに分類できます。制御ポリシーは、クライアント認証の強制などの制御アクションを使用します。データポリシーは、リクエストへのデータの挿入などのデータアクションを使用します。
Autoscale	定義された条件に従って、仮想サーバーの数をシームレスかつ自動的にスケールアップまたはスケールダウンすること。
AppFlow	NetScaler がフローデータを収集ツールにエクスポートできるようにするため。多くの場合、ネットワーク分析やセキュリティ分析に使用されます。
コンテンツ最適化	クライアントとサーバー間のトランザクション時間を短縮し、帯域幅の消費量を減らすため。また、一部のタスクをオフロードして他のタスクをより効率的にすることで、サーバーのパフォーマンスを向上させることもできます。
スピルオーバー	NetScaler ルールを使用して、スピルオーバーが発生する条件を指定すること。ルールにより、さまざまな運用条件に合わせてスピルオーバーを柔軟に設定できます。
ICA	ICAP リクエストを動的に生成するには、ICAP レスポンスを受信し、コンテンツ検査データを記録します。
VPN セッション	NetScaler Gateway で、エンドポイント分析（EPA）を構成して、ユーザーデバイスが特定のセキュリティ要件を満たしているかどうかを確認し、それに応じてユーザーに内部リソースへのアクセスを許可します。
VPN トラフィック	NetScaler Gateway で、エンドポイント分析（EPA）を構成して、ユーザーデバイスが特定のセキュリティ要件を満たしているかどうかを確認し、それに応じてユーザーに内部リソースへのアクセスを許可します。
syslog	指定した Syslog サーバにどのメッセージを記録するかを定義します。
nslog	指定した nslog サーバにどのメッセージを記録するかを定義します。

機能名	機能でのポリシーの使用方法
ビデオ最適化検出	検出ポリシーをバインドできるユーザー定義のビデオ最適化検出ポリシーラベルを作成します。ポリシーラベルは、一連のポリシーを指定された順序で評価するためのツールです。ポリシーラベルを使用すると、次のポリシーを選択したり、別のポリシーラベルを呼び出したり、前のポリシーが TRUE または FALSE のどちらと評価されたかを確認してポリシー評価を完全に終了したりするようにビデオ最適化機能を設定できます。
トンネリング	トンネリングされたトラフィックに使用する圧縮タイプを定義します。
コンテンツ検査	NetScaler ADC がインターセプトして指定されたアクションを実行するリクエストを指定します。
VPN URL	種類に応じて Web サイトリンクまたはファイル共有リンクとしてアクセスインターフェイスに表示される外部または内部リソースへのブックマークリンクを作成します。
ボット	ポリシーをバインドできるユーザー定義のボットポリシーラベルを作成します。ポリシーラベルは、一連のポリシーを指定された順序で評価するためのツールです。ポリシーラベルを使用すると、前のポリシーが TRUE または FALSE のいずれと評価されたかを確認して次のポリシーを選択したり、別のポリシーラベルを呼び出したり、ポリシー評価を完全に終了したりするようにレスポンダー機能を構成できます。
VPN イントラネットアプリケーションポリシー	NetScaler Gateway 経由でアクセスできるようにするイントラネットアプリケーションを定義すること。
SmartAccess	機能のステータス (デフォルトまたは無効) を指定する ICA アクセスプロファイルを作成する。
負荷分散	管理対象の負荷分散されたサーバー間でクライアント接続を分散する方法を定義します。

アクションとプロファイルについて

ポリシー自体は、データに対してアクションを実行しません。ポリシーは、トラフィックを評価するための読み取り専用ロジックを提供します。ポリシー評価に基づいて操作を実行する機能を有効にするには、アクションまたはプロファイルを設定し、それらをポリシーに関連付けます。

> 注:
>
> アクションとプロファイルは特定の機能に固有です。アクションとプロファイルをフィーチャーに割り当てる方法については、各機能のドキュメントを参照してください。

アクションについて

アクションは、ポリシー内の式の評価に応じて、NetScaler ADC が実行する手順です。たとえば、ポリシー内の式がリクエスト内の特定の送信元 IP アドレスと一致する場合、このポリシーに関連付けられているアクションによって、接続が許可されるかどうかが決まります。

NetScaler ADC が実行できるアクションの種類は、機能に固有です。たとえば、Rewrite では、アクションによってリクエスト内のテキストを置き換えたり、リクエストの宛先 URL を変更したりできます。統合キャッシュでは、アクションによって HTTP 応答がキャッシュまたはオリジンサーバーのどちらから提供されるかが決まります。

NetScaler ADC 機能の中には、アクションが事前定義されているものもあれば、構成可能なものもあります。場合によっては (リライトなど)、関連するポリシールールの設定に使用するのと同じタイプの式を使用してアクションを設定します。

> 注:
>
> 機能、プロトコル、方向、およびエンティティのすべての組み合わせが有効であるとは限りません。

プロファイルについて

一部の NetScaler ADC 機能では、プロファイル、またはアクションとプロファイルの両方をポリシーに関連付けることができます。プロファイルは、機能が複雑な機能を実行できるようにする設定の集まりです。たとえば、アプリケーションファイアウォールでは、XML データのプロファイルで、不正な XML 構文や SQL インジェクションの証拠がないかデータを調べるなど、複数のスクリーニング操作を実行できます。

ポリシーバインディングについて

ポリシーは、ポリシーを呼び出すことができるエンティティに関連付けられるか、エンティティにバインドされます。たとえば、すべての仮想サーバーに適用される要求時の評価にポリシーをバインドできます。特定のバインドポイントにバインドされたポリシーのコレクションは、ポリシーバンクを構成します。

以下は、ポリシーのさまざまなタイプのバインドポイントの概要です。

- リクエスト時間グローバル。ポリシーは、要求時に機能のすべてのコンポーネントで使用できます。
- レスポンス・タイム・グローバル。ポリシーは、応答時に機能のすべてのコンポーネントで使用できます。
- 要求時間、仮想サーバ固有。ポリシーは、特定の仮想サーバーの要求時処理にバインドできます。たとえば、要求時間ポリシーをキャッシュリダイレクト仮想サーバーにバインドして、特定の要求がキャッシュの負荷分

散仮想サーバーに転送され、他の要求がオリジンの負荷分散仮想サーバーに送信されるようにすることができます。

- 応答時間、仮想サーバ固有。ポリシーは、特定の仮想サーバーの応答時間処理にバインドすることもできます。
- ユーザー定義のポリシーラベル。高度なポリシーインフラストラクチャでは、ポリシーラベルを定義し、そのポリシーラベルの下で関連するポリシーのセットを集めることで、ポリシー（ポリシーバンク）のカスタムグループを設定できます。
- その他のバインドポイント。追加のバインドポイントを利用できるかどうかは、高度なポリシーの種類と、関連する NetScaler 機能の仕様によって異なります。

高度なポリシーバインディングの詳細については、「高度なポリシーを使用するバインドポリシー 」を参照してください。

ポリシーの評価順序について

NetScaler の機能は、機能のポリシーの評価や選択したアクションの実行など、特定の順序で処理されます。詳細については、「パケットフロー」を参照してください。

メッセージ処理の任意の時点で、次の組み合わせに応じてポリシー評価が実行されます。

- プロトコル (HTTP、SIP TCP、Diameter など)
- 方向 (要求または応答)
- 機能 (リライト、レスポンダー、ボットなど)

組み合わせを混同することはできません。ポリシーは、バンク（ポリシーラベルまたはバインドポイントとも呼ばれます）と呼ばれるポリシーのグループで次の順序で評価されます。

1. グローバルオーバーライド
2. 特定の LB 仮想サーバーが使用されています
3. 特定の CS 仮想サーバーが使用されている場合
4. グローバルデフォルト

銀行内では、保険契約は優先順位が最も低いものから最も高いものへと評価されます。ポリシー・ルールが「false」と評価された場合、評価は自動的に同じ銀行で次に番号が付けられた優先度に移されます。同じ銀行にポリシー・ルールがない場合、評価は順序内の次の銀行の最初のポリシーに委ねられます。ポリシーがなくなると、ポリシー評価は終了します。ポリシールールが true と評価されると、対応するアクションまたはプロファイルが記憶され、後で実行できるようになります。

ポリシーが true と評価されると、「GoToPriorityExpression」の値がチェックされます。「GoToPriorityExpression」が「END」の場合、ポリシー評価は停止します。「NEXT」の場合、(上記の) 次のポリシーが評価されます。式の場合は、その式が評価され、その優先度を持つポリシーが次に選択されます。

> 注:
>
> デフォルトの「優先度定義式」は「END」です。ただし、すべてのアクションを実行できる一部の機能で

> は、「GoToPriorityExpression」値を明示的に指定することが推奨されます。

ポリシー評価が停止すると、機能はアクションまたはプロファイルの順序付きリストを実行します。これらの機能は、すべてのアクション (たとえば、書き換え) を実行するか、1 つのアクション (たとえば、レスポンダー、ボット) を実行します。1 つしか実行できない機能に複数のアクションまたはプロファイルが関連付けられている場合、標準では最後のアクションまたはプロファイルが実行されます。アクションまたはプロファイルが選択されていない場合、機能はデフォルトのアクションを実行します。

トラフィックフローに基づく評価の順序

一部のポリシーは、他のポリシーの結果に影響します。以下はその例です。

- 統合キャッシュから応答が提供される場合、他の NetScaler ADC 機能の一部は、応答または応答を開始した要求を処理しません。
- アプリケーションファイアウォールが着信要求を拒否した場合、他の機能では要求を処理できません。
- Responder によって実行されるほとんどのアクションは、それ以降の処理を停止します。
- Rewrite によって実行される Drop アクションと Reset アクションは、それ以降の処理を停止します。

高度なポリシー式

August 15, 2023

ポリシーの最も基本的な構成要素の 1 つは、そのルールです。ポリシールールは、ポリシーでトラフィックを分析できるようにする論理式です。ポリシーの機能のほとんどは、その式から派生しています。

式は、トラフィックまたはその他のデータの特性を 1 つ以上のパラメータおよび値で照合します。たとえば、式を使用すると、NetScaler ADC で次のことを実行できます。

- 要求に証明書が含まれているかどうかを判断します。
- TCP 要求を送信したクライアントの IP アドレスを確認します。
- HTTP リクエストに含まれるデータ（一般的なスプレッドシートやワープロアプリケーションなど）を識別します。
- HTTP リクエストの長さを計算します。

高度なポリシー式について

高度なポリシーインフラストラクチャを使用する機能では、高度な式も使用されます。詳細ポリシーを使用する機能の詳細については、表「NetScaler 機能、ポリシータイプ、ポリシーの使用状況」を参照してください。

高度なポリシー式には、他にもいくつかの用途があります。ポリシールールで高度な式を設定することに加えて、次の状況で高度な式を構成します。

- 統合キャッシュ:

  高度なポリシー式を使用して、統合キャッシュ内のコンテンツグループのセレクタを設定します。

- 負荷分散:

  高度なポリシー式を使用して、負荷分散に TOKEN メソッドを使用する負荷分散仮想サーバーのトークン抽出を構成します。

- 書き換え:

  高度なポリシー式を使用して、書き換えアクションを構成します。

- レートベースのポリシー:

  さまざまなサーバへのトラフィックのレートを制御するポリシーを設定するときに、高度なポリシー式を使用して制限セレクタを設定します。

次に、高度なポリシー式の簡単な例をいくつか示します。

- HTTP リクエスト URL に含まれる文字数は 500 文字以下です。

```
http.req.url.length \<= 500
```

- HTTP リクエストには、500 文字未満のクッキーが含まれています。

```
http.req.cookie.length \< 500
```

- HTTP リクエスト URL には特定のテキスト文字列が含まれています。

```
http.req.url.contains(".html")
```

NSPEPI ツールを使用したポリシー式の変換

January 11, 2024

> 注:
>
> NSPEPI と事前設定チェックツールは、公開 GitHub からダウンロードできます。詳細については、ツールをダウンロード、インストール、使用するための詳細な手順については、GitHub [NSPEPI ページと README ページを参照してください](https://github.com/citrix/ADC-scripts/blob/master/nspepi/README.md)。お客様には、GitHub で利用可能なツールを使用して、最も完全で最新のバージョンを使用することをお勧めします。

従来のポリシーベースの機能は、NetScaler 12.0 ビルド 56.20 以降では廃止されました。別の方法として、NetScaler では、NSPEPI ツールでサポートされる機能に高度なポリシーインフラストラクチャを使用することを推奨しています。サポートされている機能のリストについては、「nspepi 変換ツールで処理されるコマンドまたは機能」を参照してください。NSPEPI ツールでサポートされていない機能は、引き続きクラシックポリシーをサポートします。サポートされていない機能のリストについては、「NSPEPI ツールのサポートされていない機能」を参照してください。

このnspepiツールは、次のことを実行できます:

1. クラシックポリシー式を高度なポリシー式に変換します。
2. 特定のクラシックポリシーとそのエンティティバインディングを高度なポリシーおよびバインディングに変換します。
3. いくつかの非推奨の機能を、対応する非推奨の機能に変換します。
4. 従来のフィルタコマンドを高度なフィルタコマンドに変換します。

> 注:
>
> nspepiツールが ns.conf 構成ファイルを正常に変換すると、変換されたファイルは、接頭辞「new_」が付いた新しいファイルとして表示されます。変換された設定ファイルにエラーまたは警告がある場合は、変換プロセスの一環として手動で修正する必要があります。変換したら、テスト環境でファイルをテストし、それを使用して実際の ns.conf 構成ファイルを置き換える必要があります。テスト後、新しく変換または修正された ns.conf 構成ファイル用にアプライアンスを再起動する必要があります。

クラシックポリシーまたは式のみをサポートする機能は廃止され、対応する非推奨の機能に置き換えることができます。

> 注:
>
> 旧バージョンのnspepiツールに関する情報は PDF 形式で入手できます。詳細については、「PDF 12.1-51.16 より前の nspepi ツールを使用した従来のポリシー変換 」を参照してください。

変換の警告とエラーファイル

変換にツールを使用する前に、注意すべき警告はほとんどありません。

1. すべての警告とエラーがコンソールに出力されます。設定ファイルが保存されている場所に警告ファイルが作成されます。
2. 警告およびエラーファイルの名前は、入力ファイルと同じですが、ファイル名に接頭辞「warn_」が追加されています。式の変換中 (-e を使用する場合)、警告はカレントディレクトリに「warn_expr」という名前で表示されます。

> 注:
>
> このファイルは、日付/時刻スタンプとログレベルを含む標準のログファイル形式です。ツールが複数回実行されるので、ファイルの以前のインスタンスは「.1」、「.2」などの接尾辞で保持されます。最大で 10 個のインス

> タンスが保持されます。

変換されたファイル形式

設定ファイル (「-f」を使用) を変換する場合、変換されたファイルは、同じ名前でプレフィックス「new_」の入力設定ファイルが存在するディレクトリと同じディレクトリに置かれます。

nspepi 変換ツールによって処理されるコマンドまたは機能

自動変換処理中に処理されるコマンドは次のとおりです。

- 次のクラシックポリシーとその式は、高度なポリシーと式に変換されます。変換には、エンティティバインディングとグローバルバインディングが含まれます。

1. add appfw policy
2. add cmp policy
3. add cr policy
4. add cs policy
5. SSL ポリシーの追加
6. add filter action
7. add filter policy
8. 負荷分散、コンテンツスイッチング、キャッシュリダイレクト、およびグローバルへのポリシーバインディングをフィルタします。

- 「Add lb 仮想サーバー」で設定されたルールパラメーターは、クラシック式から高度な式に変換されます。
- 「追加 ns HttpProfile」または「ns httpProfile を設定」コマンドで設定された SPDY パラメータが「-http2 ENABLED」に変更されます。
- 名前付き式 (「ポリシー式の追加」コマンド)。各クラシック名前付きポリシー式は、接頭辞として「nspepi_adv_」が設定された、対応する Advanced 名前付き式に変換されます。さらに、変換されたクラシック式での名前付き式の使用は、対応する Advanced 名前付き式に変更されます。さらに、すべての名前付き式には 2 つの名前付き式があります。1 つは Classic で、もう 1 つは Advanced です (以下を参照)。
- トンネルトラフィックポリシー変換がサポートされています。
- CMP、CR、およびトンネルでの組み込みのクラシックポリシーバインディングの処理。
- Patclass フィーチャは Pat セットフィーチャに変換されます。
- 「書き換えアクションを追加」コマンドの「-pattern」パラメータは、「-search」パラメータを使用するように変換されます。
- 高度な定義式の Q および S プレフィックスは、非推奨の同等の高度な式に変換されます。これらの式は、高度な定義式を使用できるすべてのコマンドで見ることができます。

例:


```
add policy expression classic_expr ns_true
Converts to:
add policy expression classic_expr ns_true
add policy expression nspepi_adv_classic_expr TRUE
<!--NeedCopy-->
```

- 「set cmp パラメータ」コマンドで設定された policyType パラメータが削除されます。デフォルトでは、ポリシータイプは「Advanced」です。

従来のフィルタコマンドを高度なフィルタコマンドに変換する

このnspepiツールは、追加、バインドなどの従来のフィルタ操作に基づくコマンドを高度なフィルタコマンドに変換できます。

ただし、nspepiツールは次のフィルターコマンドをサポートしていません。

1. add filter action <action Name> FORWARD <service name>
2. add filter action <action name> ADD prebody
3. add filter action <action name> ADD postbody

> 注:
>
> 1. ns.conf に既存の書き換え機能またはレスポンダ機能があり、それらのポリシーがGOTO式ENDまたはUSER_INVOCATION_RESULTとしてグローバルにバインドされ、バインドタイプがREQ_XまたはRES_Xである場合、バインドフィルタコマンドは部分的に変換され、コメントアウトされます。手動変換ではエラーが表示されます。
> 2. 既存の書き換え機能またはレスポンダー機能があり、そのポリシーがGOTO - ENDまたはUSER_INVOCATION_RESULTを使用して HTTPS タイプの仮想サーバー (負荷分散、コンテンツスイッチング、キャッシュリダイレクトなど) にバインドされている場合、ツールはバインドフィルターコマンドを部分的に変換してからコメントアウトします。手動変換には警告が表示されます。

例

以下は入力例です。

```
add lb vserver v1 http 1.1.1.1 80 -persistenceType NONE -cltTimeout
    9000
add cs vserver csv1 HTTP 1.1.1.2 80 -cltTimeout 180 -persistenceType
    NONE
add cr vserver crv1 HTTP 1.1.1.3 80 -cacheType FORWARD
add service svc1 1.1.1.4 http 80
add filter action fact_add add 'header:value'
add filter action fact_variable add 'H1:%%HTTP.TRANSID%%'
add filter action fact_prebody add prebody
add filter action fact_error_act1 ERRORCODE 200 "<HTML>Good URL</HTML>"
```



```
add filter action fact_forward_act1 FORWARD svc1
add filter policy fpol_add_res -rule ns_true -resAction fact_add
add filter policy fpol_error_res -rule ns_true -resAction
    fact_error_act1
add filter policy fpol_error_req -rule ns_true -reqAction
    fact_error_act1
add filter policy fpol_add_req -rule ns_true -reqAction fact_add
add filter policy fpol_variable_req -rule ns_true -reqAction
    fact_variable
add filter policy fpol_variable_res -rule ns_true -resAction
    fact_variable
add filter policy fpol_prebody_req -rule ns_true -reqAction
    fact_prebody
add filter policy fpol_prebody_res -rule ns_true -resAction
    fact_prebody
add filter policy fpol_forward_req -rule ns_true -reqAction
    fact_forward_act1
bind lb vserver v1 -policyName fpol_add_res
bind lb vserver v1 -policyName fpol_add_req
bind lb vserver v1 -policyName fpol_error_res
bind lb vserver v1 -policyName fpol_error_req
bind lb vserver v1 -policyName fpol_variable_res
bind lb vserver v1 -policyName fpol_variable_req
bind lb vserver v1 -policyName fpol_forward_req
bind cs vserver csv1 -policyName fpol_add_req
bind cs vserver csv1 -policyName fpol_add_res
bind cs vserver csv1 -policyName fpol_error_res
bind cs vserver csv1 -policyName fpol_error_req
bind cr vserver crv1 -policyName fpol_add_req
bind cr vserver crv1 -policyName fpol_add_res
bind cr vserver crv1 -policyName fpol_error_res
bind cr vserver crv1 -policyName fpol_error_req
bind cr vserver crv1 -policyName fpol_forward_req
bind filter global fpol_add_req
bind filter global fpol_add_res
bind filter global fpol_error_req
bind filter global fpol_error_res
bind filter global fpol_variable_req
bind filter global fpol_variable_res
bind filter global fpol_variable_res -state DISABLED
bind filter global fpol_prebody_req
bind filter global fpol_forward_req
After conversion, warning/error messages will be displayed for manual
    effort.
Warning files:
cat warn_<input file name>:
2019-11-07 17:13:34,724: ERROR - Conversion of [add filter action
    fact_prebody add prebody] not supported in this tool.
2019-11-07 17:13:34,739: ERROR - Conversion of [add filter action
    fact_forward_act1 FORWARD svc1] not supported in this tool.
2019-11-07 17:13:38,042: ERROR - Conversion of [add filter policy
    fpol_prebody_req -rule ns_true -reqAction fact_prebody] not
    supported in this tool.
```



```
2019-11-07 17:13:38,497: ERROR - Conversion of [add filter policy
    fpol_prebody_res -rule ns_true -resAction fact_prebody] not
    supported in this tool.
2019-11-07 17:13:39,035: ERROR - Conversion of [add filter policy
    fpol_forward_req -rule ns_true -reqAction fact_forward_act1] not
    supported in this tool.
2019-11-07 17:13:39,060: WARNING - Following bind command is commented
    out because state is disabled. Advanced expressions only have a
    fixed ordering of the types of bindings without interleaving, except
     that global bindings are allowed before all other bindings and
    after all bindings. If you have global bindings in the middle of non
    -global bindings or any other interleaving then you will need to
    reorder all your bindings for that feature and direction. Refer to
    nspepi documentation. If command is required please take a backup
    because comments will not be saved in ns.conf after triggering 'save
     ns config': bind filter global fpol_variable_res -state DISABLED


<!--NeedCopy-->
```

次に、出力例を示します。変換されたすべてのコマンドにはコメントが付きます。

```
cat new_<input file name>
add rewrite action fact_add insert_http_header header ""value""
add filter action fact_prebody add prebody
add filter action fact_forward_act1 FORWARD svc1
add filter policy fpol_prebody_req -rule ns_true -reqAction
    fact_prebody
add filter policy fpol_prebody_res -rule ns_true -resAction
    fact_prebody
add filter policy fpol_forward_req -rule ns_true -reqAction
    fact_forward_act1
bind lb vserver v1 -policyName fpol_forward_req
bind cr vserver crv1 -policyName fpol_forward_req
#bind filter global fpol_variable_res -state DISABLED
bind filter global fpol_prebody_req
bind filter global fpol_forward_req
add rewrite action nspepi_adv_fact_variable insert_http_header H1 HTTP.
    RES.TXID
add rewrite action fact_variable insert_http_header H1 HTTP.REQ.TXID
add responder action fact_error_act1 respondwith "HTTP.REQ.VERSION.
    APPEND(" 200 OK\r
nConnection: close\r
nContent-Length: 21\r\n\r
n<HTML>Good URL</HTML>")"
add rewrite action nspepi_adv_fact_error_act1 replace_http_res "HTTP.
    REQ.VERSION.APPEND(" 200 OK\r
nConnection: close\r
nContent-Length: 21\r\n\r
n<HTML>Good URL</HTML>")"
add rewrite policy fpol_add_res TRUE fact_add
add rewrite policy fpol_error_res TRUE nspepi_adv_fact_error_act1
add responder policy fpol_error_req TRUE fact_error_act1
```



```
add rewrite policy fpol_add_req TRUE fact_add
add rewrite policy fpol_variable_req TRUE fact_variable
add rewrite policy fpol_variable_res TRUE nspepi_adv_fact_variable
set cmp parameter -policyType ADVANCED
bind rewrite global fpol_add_req 100 NEXT -type REQ_DEFAULT
bind rewrite global fpol_variable_req 200 NEXT -type REQ_DEFAULT
bind rewrite global fpol_add_res 100 NEXT -type RES_DEFAULT
bind rewrite global fpol_error_res 200 NEXT -type RES_DEFAULT
bind rewrite global fpol_variable_res 300 NEXT -type RES_DEFAULT
bind responder global fpol_error_req 100 END -type REQ_DEFAULT
bind lb vserver v1 -policyName fpol_add_res -type RESPONSE -priority
    100 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_error_res -type RESPONSE -priority
    200 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_variable_res -type RESPONSE -
    priority 300 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_add_req -type REQUEST -priority 100
     -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_variable_req -type REQUEST -
    priority 200 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_error_req -type REQUEST -priority
    100 -gotoPriorityExpression END
bind cs vserver csv1 -policyName fpol_add_req -type REQUEST -priority
    100 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_add_res -type RESPONSE -priority
    100 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_error_res -type RESPONSE -
    priority 200 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_error_req -type REQUEST -priority
     100 -gotoPriorityExpression END
bind cr vserver crv1 -policyName fpol_add_req -type REQUEST -priority
    100 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_add_res -type RESPONSE -priority
    100 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_error_res -type RESPONSE -
    priority 200 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_error_req -type REQUEST -priority
     100 -gotoPriorityExpression END

<!--NeedCopy-->
```

既存の書き換えまたはレスポンダーポリシーバインディングに **goto** 式 **END** または **USE_INNVOCATION** がある場合、従来のフィルタコマンドを高度な機能コマンドに変換します

この変換では、書き換えポリシーが 1 つ以上の仮想サーバーにバインドされ、サーバーに END または USE_INVOCATION_RESULT がある場合、ツールはコマンドをコメントアウトします。

例

次に、入力コマンドの例を示します:

```
COPY
add filter policy fpol1 -rule ns_true -resAction reset
add filter policy fpol2 -rule ns_true -reqAction reset
add rewrite policy pol1 true NOREWRITE
add rewrite policylabel pl http_res
bind rewrite policylabel pl pol1 1
bind rewrite global NOPOLICY 1 USE_INVOCATION_RESULT -type RES_DEFAULT
    -invoke policylabel pl
add responder policy pol2 true NOOP
add responder policylabel pl -policylabeltype HTTP
bind responder policylabel pl pol2 1
bind responder global NOPOLICY 1 USE_INVOCATION_RESULT -type
    REQ_DEFAULT -invoke policylabel pl
bind lb vserver v1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind cs vserver csv1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind lb vserver v1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cs vserver csv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cr vserver crv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind lb vserver v1_http -policyName fpol1
bind cs vserver csv1_http -policyName fpol1
bind lb vserver v2_http -policyName fpol2
bind cs vserver csv2_http -policyName fpol2
bind cr vserver crv2_http -policyName fpol2
bind filter global fpol1 -priority 100
bind filter global fpol2 -priority 100
<!--NeedCopy-->
```

次に、出力コマンドの例を示します:

```
COPY
add rewrite policy pol1 true NOREWRITE
add rewrite policylabel pl http_res
bind rewrite policylabel pl pol1 1
add responder policy pol2 true NOOP
add responder policylabel pl -policylabeltype HTTP
bind responder policylabel pl pol2 1
add rewrite policy fpol1 TRUE RESET
add responder policy fpol2 TRUE RESET
#bind lb vserver v1_http -policyName fpol1 -type RESPONSE
#bind cs vserver csv1_http -policyName fpol1 -type RESPONSE
#bind rewrite global fpol1 100 -type RES_DEFAULT
#bind lb vserver v2_http -policyName fpol2 -type REQUEST
#bind cs vserver csv2_http -policyName fpol2 -type REQUEST
#bind cr vserver crv2_http -policyName fpol2 -type REQUEST
```



```
#bind responder global fpol2 100 -type REQ_DEFAULT
bind rewrite global NOPOLICY 1 USE_INVOCATION_RESULT -type RES_DEFAULT
    -invoke policylabel pl
bind responder global NOPOLICY 1 USE_INVOCATION_RESULT -type
    REQ_DEFAULT -invoke policylabel pl
bind lb vserver v1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind lb vserver v1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cs vserver csv1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind cs vserver csv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cr vserver crv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST-

<!--NeedCopy-->
```

nspepi ツールを実行する

以下は、nspepiツールを実行するためのコマンドライン例です。このツールは、シェルのコマンドラインから実行されます（そのためには、NetScaler「CLI」に「シェル」コマンドを入力する必要があります）。変換を実行するには、「-f」または」-e」のいずれかを指定する必要があります。「-d」の使用は、Citrix 担当者がサポート目的で分析することを意図しています。

```
usage: nspepi [-h] (-e <classic policy expression> | -f <path to ns
    config file>)[-d] [-v] [-V

Convert classic policy expressions to advanced policy expressions and
    deprecated commands to non-deprecated
commands.

optional arguments:
-h, --help show this help message and exit
-e <classic policy expression>, --expression <classic policy expression
    >
convert classic policy expression to advanced policy
expression (maximum length of 8191 allowed)
-f <path to ns config file>, --infile <path to ns config file>
convert netscaler config file
-d, --debug log debug output
-v, --verbose show verbose output
-V, --version show program's version number and exit
<!--NeedCopy-->
```

使用例**:**

1. `nspepi -e "req.tcp.destport == 80"`
2. `nspepi -f /nsconfig/ns.conf`

次に、CLI を使用したnspepiツールの実行例をいくつか示します。

—**e** パラメータの出力例**:**

```
root@ns# nspepi -e "req.http.header foo == "bar""
"HTTP.REQ.HEADER("foo").EQ("bar")"
<!--NeedCopy-->
```

-f パラメータの出力例**:**

```
root@ns# cat sample.conf
add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180 -
    originUSIP OFF
add cr policy cr_pol1 -rule ns_true
bind cr vserver cr_vs -policyName cr_pol1
<!--NeedCopy-->
```

-f パラメータを指定して **nspepi** を実行中:

```
nspepi -f sample.conf
<!--NeedCopy-->
```

変換されたコンフィグは新しいファイルnew_sample.confで使用できます。
warn_sample.confファイルをチェックして、生成された可能性のある警告またはエラーがないか確認します。

-f パラメータと-v パラメータの出力例

```
nspepi -f sample.conf -v
INFO - add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180
     -originUSIP OFF
INFO - add cr policy cr_pol1 -rule TRUE -action ORIGIN
INFO - bind cr vserver cr_vs -policyName cr_pol1 -priority 100 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

変換されたコンフィグは新しいファイルnew_sample.confで使用できます。
warn_sample.confファイルをチェックして、生成された可能性のある警告またはエラーがないか確認します。

変換された設定ファイル**:**

```
root@ns# cat new_sample.conf
add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180 -
    originUSIP OFF
add cr policy cr_pol1 -rule TRUE -action ORIGIN
set cmp parameter -policyType ADVANCED
bind cr vserver cr_vs -policyName cr_pol1 -priority 100 -
    gotoPriorityExpression END -type REQUEST

<!--NeedCopy-->
```


エラーや警告のない設定例の出力例**:**

```
nspepi -f sample_2.conf
<!--NeedCopy-->
```

変換されたコンフィグは新しいファイル`new_sample_2.conf`で使用できます。
`warn_sample_2.conf`ファイルをチェックして、生成された可能性のある警告またはエラーがないか確認します。

警告付きの設定例の出力例**:**

```
root@ns# cat sample_2.conf
add policy expression security_expr "req.tcp.destport == 80" -
    clientSecurityMessage "Not allowed"
set cmp parameter -policyType CLASSIC
add cmp policy cmp_pol1 -rule ns_true -resAction COMPRESS
add cmp policy cmp_pol2 -rule ns_true -resAction COMPRESS
add cmp policy cmp_pol3 -rule TRUE -resAction COMPRESS
bind cmp global cmp_pol1
bind cmp global cmp_pol2 -state DISABLED
bind cmp global cmp_pol3 -priority 1 -gotoPriorityExpression END -type
    RES_DEFAULT
bind lb vserver lb_vs -policyName cmp_pol2
root@ns#
<!--NeedCopy-->
```

-f パラメータを指定した **nspepi** の実行例**:**

```
root@ns# nspepi -f sample_2.conf
ERROR - Error in converting expression security_expr : conversion of
    clientSecurityMessage based expression is not supported.
WARNING - Following bind command is commented out because state is
    disabled. Advanced expressions only have a fixed ordering of the
    types of bindings without interleaving, except that global bindings
    are allowed before all other bindings and after all bindings. If you
     have global bindings in the middle of non-global bindings or any
    other interleaving then you will need to reorder all your bindings
    for that feature and direction. Refer to nspepi documentation. If
    command is required please take a backup because comments will not
    be saved in ns.conf after triggering 'save ns config': bind cmp
    global cmp_pol2 -state DISABLED
Warning - Bindings of advanced CMP policies to cmp global are commented
     out, because initial global cmp parameter is classic but advanced
    policies are bound. Now global cmp parameter policy type is set to
    advanced. If commands are required please take a backup because
    comments will not be saved in ns.conf after triggering 'save ns
    config'. Advanced expressions only have a fixed ordering of the
    types of bindings without interleaving, except that global bindings
    are allowed before all other bindings and after all bindings. If you
     have global bindings in the middle of non-global bindings or any
    other interleaving then you will need to reorder all your bindings
    for that feature and direction. Refer to nspepi documentation.
root@ns#
```



```
<!--NeedCopy-->
```

変換されたファイル:

```
root@ns# cat new_sample_2.conf
add policy expression security_expr "req.tcp.destport == 80" -
    clientSecurityMessage "Not allowed"
set cmp parameter -policyType ADVANCED
add cmp policy cmp_pol1 -rule TRUE -resAction COMPRESS
add cmp policy cmp_pol2 -rule TRUE -resAction COMPRESS
add cmp policy cmp_pol3 -rule TRUE -resAction COMPRESS
#bind cmp global cmp_pol2 -state DISABLED
#bind cmp global cmp_pol3 -priority 1 -gotoPriorityExpression END -type
     RES_DEFAULT
bind cmp global cmp_pol1 -priority 100 -gotoPriorityExpression END -
    type RES_DEFAULT
bind lb vserver lb_vs -policyName cmp_pol2 -priority 100 -
    gotoPriorityExpression END -type RESPONSE
root@ns#
<!--NeedCopy-->
```

警告ファイル:

```
root@ns# cat warn_sample_2.conf
2019-02-28 06:20:10,590: ERROR - Error in converting expression
    security_expr : conversion of clientSecurityMessage based expression
     is not supported.
2019-02-28 06:20:12,187: WARNING - Following bind command is commented
    out because state is disabled. Advanced expressions only have a
    fixed ordering of the types of bindings without interleaving, except
     that global bindings are allowed before all other bindings and
    after all bindings. If you have global bindings in the middle of non
    -global bindings or any other interleaving then you will need to
    reorder all your bindings for that feature and direction. Refer to
    nspepi documentation. If command is required please take a backup
    because comments will not be saved in ns.conf after triggering 'save
     ns config': bind cmp global cmp_pol2 -state DISABLED
2019-02-28 06:20:12,191: WARNING - Bindings of advanced CMP policies to
     cmp global are commented out, because initial global cmp parameter
    is classic but advanced policies are bound. Now global cmp parameter
     policy type is set to advanced. If commands are required please
    take a backup because comments will not be saved in ns.conf after
    triggering 'save ns config'. Advanced expressions only have a fixed
    ordering of the types of bindings without interleaving, except that
    global bindings are allowed before all other bindings and after all
    bindings. If you have global bindings in the middle of non-global
    bindings or any other interleaving then you will need to reorder all
     your bindings for that feature and direction. Refer to nspepi
    documentation.
root@ns#
<!--NeedCopy-->
```


バインディングの優先順位

高度なポリシーでは、グローバルと非グローバルの間、および異なるバインディングタイプ間の優先度による任意のインターリーブは許可されません。このようなクラシックポリシーの優先順位のインターリーブに依存する場合は、高度なポリシールールに準拠し、希望する動作になるように優先順位を調整する必要があります。
詳細ポリシーの優先順位は、バインドポイントに対してローカルです。バインドポイントは、プロトコル、機能、方向、およびエンティティの一意の組み合わせです（エンティティは、特定の仮想サーバ、ユーザ、グループ、サービス、およびグローバルオーバーライドまたはグローバルデフォルトのいずれかです)。ポリシーの優先順位は、バインドポイント全体では従いません。

特定のプロトコル、機能、方向性について、詳細ポリシーを評価する順序は次のとおりです。

- グローバルオーバーライド。
- (現在の) 認証、承認、および監査ユーザー。
- 認証、認可、および監査グループ（ユーザーがメンバーである）の重み順です。複数のグループの重みが同じ場合、順序は定義されません。
- 要求を受信したか、コンテンツスイッチングが選択された LB 仮想サーバー。
- コンテンツスイッチング仮想サーバー、要求を受信したキャッシュリダイレクト仮想サーバー。
- 負荷分散によって選択されたサービス。
- グローバルデフォルト。

各バインドポイント内では、ポリシーは、番号の小さいものから高い番号の優先順位で評価されます。ポリシーは、使用されているプロトコルとメッセージの受信元の方向についてのみ評価されます。

手動での優先順位の再設定が必要な従来のポリシーバインディング

ここでは、ニーズを満たすために手動で優先順位を再設定する必要があるクラシックポリシーバインディングのタイプをいくつか示します。これらはすべて、特定のフィーチャと方向に対するものです。

- 上記のエンティティタイプリストの方向とは逆のプライオリティ番号が増加するクラシックプライオリティ。たとえば、負荷分散仮想サーバーバインディングよりも低いコンテンツスイッチング仮想サーバーバインディングです。
- 認証、認可、および監査グループの重みの順序以外の数が増加する従来の優先順位。
- 一部の非グローバルプライオリティよりも小さく、同じグローバルプライオリティのクラシックグローバルプライオリティは、他の非グローバルプライオリティ（つまり、非グローバルプライオリティのセグメント、続いて 1 つ以上のグローバル、その後に非グローバルプライオリティ）よりも大きくなります。

NSPEPI および **check_invalid_config** ツールは、**CentOS** および **Ubuntu** システムで実行できます

以下のモジュールは、これらのツールを使用するための前提条件です。

- Python
- Perl
- Python pip モジュール
- Python 用プレイモジュール
- .pm を Perl 用に Switch.pm

Python 3 がインストールされている場合は、`"ln -s /usr/bin/python3 /usr/bin/python"`などのソフトリンクを作成します。

Python の pip モジュール、Python 用の PLY モジュール、および Perl 用の Switch.pm を CentOS にインストールするには、次のコマンドを実行します。

- `sudo yum install -y perl-Switch`
- `sudo yum install python-pip`
- `sudo yum install python-ply`

Python の pip モジュール、Python 用の PLY モジュール、および Perl 用の Switch.pm を Ubuntu にインストールするには、以下のコマンドを実行します。

- `sudo apt install libswitch-perl`
- `sudo apt install python-ply`
- `sudo apt install python-pip or sudo apt install python3-pip`

NSPEPI ツールがサポートされていない機能

January 11, 2024

このトピックでは、NSPEPI ツールでサポートされていない機能について説明します。

nspepi 変換ツールで処理されないコマンドまたは機能

以下は、自動変換プロセスの一部として処理されないコマンドです。

- クライアントセキュリティ式は処理されません。
- 認証
- 承認
- VPN
- Syslog
- Nslog
- ファイルベースの Classic 式は処理されません。

> 注:
>
> patClass/Filter のようないくつかの機能では、コマンドの構文が変更されます。コマンドポリシーがある場合は、お客様の要件に応じてコマンドポリシーを変更する必要がある場合があります。

クラシックポリシーを次の機能の高度なポリシーに変換するには、NetScaler カスタマーサポートにお問い合わせください:

- SureConnect (SC)
- プライオリティキューイング（PQ）
- HTTP サービス拒否 (HDOS)
- HTML インジェクション

バインディングの優先順位

高度なポリシーでは、グローバルと非グローバルの間、および異なるバインディングタイプ間の優先度による任意のインターリーブは許可されません。従来のポリシーの優先順位をこのように相互に組み合わせる場合は、アドバンスド・ポリシー・ルールに準拠し、望ましい動作が得られるように優先順位を調整する必要があります。
詳細ポリシーの優先順位は、バインドポイントに対してローカルです。バインドポイントは、プロトコル、機能、方向、およびエンティティの一意の組み合わせです（エンティティは、特定の仮想サーバ、ユーザ、グループ、サービス、およびグローバルオーバーライドまたはグローバルデフォルトのいずれかです）。ポリシーの優先順位は、バインドポイント全体では従いません。

特定のプロトコル、機能、方向性について、詳細ポリシーを評価する順序は次のとおりです。

- グローバルオーバーライド。
- ユーザーの認証、承認、および監査 (現在)。
- 認証グループ、承認グループ、監査グループ (ユーザーが所属している) を、重要度の高い順に並べています。2 つ以上のグループの重みが同じ場合、順序は定義されません。
- 要求が受信された LB 仮想サーバー、または CS によって選択された LB 仮想サーバー。
- コンテンツスイッチング仮想サーバー、要求を受信したキャッシュリダイレクト仮想サーバー。
- 負荷分散によって選択されたサービス。
- グローバルデフォルト。

認可ポリシーの評価では、次の順序になります。

- システムのオーバーライド。
- 要求が受信されたか、CS によって選択された負荷分散仮想サーバー。
- 要求を受信したコンテンツスイッチング仮想サーバー。
- システムデフォルト。

各バインドポイント内では、ポリシーは、番号の小さいものから高い番号の優先順位で評価されます。ポリシーは、使用されているプロトコルとメッセージの受信元の方向についてのみ評価されます。

警告

次のシナリオは、nspepiツール内の警告を示しています:

- 負荷分散仮想サーバーのルール式がブール式の場合、同等の高度な式は文字列形式のブール値になります。その結果、ルールをpersistenceTypeまたはlbMethodに使用すると機能が変わります。機能の変更を避けるため、コマンドはkeywords ruleおよびpersistenceTypeを削除して変更します。
- バインディングコマンドの state フィールドが DISABLED の場合。状態が無効の場合、コマンドは使用されていません。state パラメーターは詳細設定ではサポートされていません。そのため、この構成を変換すると、機能が変わります。コマンドが必要な場合は、save ns configトリガー後ns.confにコメントが保存されないため、バックアップを取ってください。

CMP フィーチャー変換の警告**:**

- グローバル cmp パラメータポリシータイプが CLASSIC に設定されていて、詳細ポリシーがグローバルにバインドされている場合。変換しないと、グローバルポリシータイプが CLASSIC に設定されているため、制限付きの高度なポリシーは評価されません。変換後、ポリシータイプは ADVANCED に変換されます。そのため、既存のグローバルアドバンストバインディングをコメントアウトしないと、これらのバインディングが評価され、機能が変更される可能性があります。
- グローバル cmp パラメータのポリシータイプが ADVANCED に設定されている場合、クラシックポリシーはグローバルにバインドされます。変換しないと、グローバルポリシータイプが ADVANCED であるため、これらのグローバルクラシックバインディングは評価されません。そのため、機能を維持するために、変換された設定をコメントアウトします。コメントアウトしないと、変換された高度なポリシーが評価され、機能が変更される可能性があります。

> 注:
>
> -state オプションを無効にした従来のポリシーバインディングはすべてコメントアウトされます。-state オプションは、高度なポリシーバインディングでは使用できません。

NSPEPI ツールの制限事項

次のシナリオでは、nspepiツールでエラーが発生します:

- 式を変換するときに問題がある場合
- 指定されたポリシー式で-ClientSecurityMessage パラメータが使用されている場合、このパラメータは拡張ポリシー式ではサポートされていないためです
- 負荷分散仮想サーバーのルール式が複雑な式で、CONTENT ベースの式が複数ある場合
- CMP フィーチャー変換のエラーは、次のシナリオで発生します。
  - 従来のポリシーと高度なポリシーの両方がグローバルに結びついています
  - クラシック・ポリシーがバインドされ、CMP パラメータがアドバンス・ポリシーになる
  - 高度なポリシーはバインドされており、CMP パラメータはクラシックです

  - クラシックポリシーは仮想サーバーにバインドされ、高度なポリシーはグローバルサーバーにバインドされます
  - 詳細ポリシーは仮想サーバーにバインドされ、クラシックポリシーはグローバルサーバーにバインドされます
  - クラシックポリシーは仮想サーバーにバインドされ、クラシックポリシーとアドバンスポリシーの両方がグローバルサーバーにバインドされます。
  - 高度なポリシーは仮想サーバーにバインドされ、クラシックポリシーと高度なポリシーの両方がグローバルサーバーにバインドされます。

- 従来の名前付きエクスプレッションの名前がコールアウトエンティティ名と同じ場合
- クラシックエクスプレッション名がアドバンスエクスプレッションに対して無効な場合
- 変換された式の長さが 1499 文字を超える場合
- クラシックエクスプレッションにクライアントセキュリティエクスプレッションまたはファイルベースのエクスプレッションがある場合

手動での優先順位の再設定が必要な従来のポリシーバインディング

ここでは、ニーズを満たすために手動で優先順位を再設定する必要があるクラシックポリシーバインディングのタイプをいくつか示します。これらはすべて、特定のフィーチャと方向に対するものです。

- 上記のエンティティタイプリストの方向とは逆のプライオリティ番号が増加するクラシックプライオリティ。たとえば、コンテンツスイッチ仮想サーバーバインディングは、負荷分散仮想サーバーバインディングよりも低くなります。
- 認証、承認、および監査グループをインターリーブする従来の優先順位。あるグループの一部が他のグループの前にあり、さらに別のパートはその他のグループの一部の後にあります。
- 認証、認可、および監査グループの重みの順序以外の数が増加する従来の優先順位。
- 一部の非グローバルプライオリティよりも小さく、同じグローバルプライオリティのクラシックグローバルプライオリティは、他の非グローバルプライオリティ（つまり、非グローバルプライオリティのセグメント、続いて 1 つ以上のグローバル、その後に非グローバルプライオリティ）よりも大きくなります。

事前設定チェックツール

August 15, 2023

> 注:
>
> NSPEPI と事前設定チェックツールは、パブリック GitHub からダウンロードできます。詳細については、GitHub NEPEPI ページおよび GitHub 事前設定ページを参照して 、ツールをダウンロードする詳細な手順を参照してください。お客様には、GitHub で利用可能なツールを使用して、最も完全で最新のバージョンを使用

> することをお勧めします。

NetScaler 12.1、13.0、および 13.1 バージョンでは、無効または削除された機能が機能構成で引き続き使用されているかどうかを確認するための事前検証ツールを使用できます。NetScaler 13.1 バージョンで削除されたコマンドにコマンドまたはパラメーターが含まれている場合、ツールは`nsconfig`ファイルを検証します。検証結果に削除されたコマンドまたは無効なコマンドの使用が示された場合は、アプライアンスをアップグレードする前に、まず Citrix が推奨する代替手段に設定を変更する必要があります。

また、このツールは、クラシックポリシーをサポートしない機能設定で使用されるクラシックポリシー式の使用を検証します。手動で修正することも、`nspepi`ツールを使用することもできます。

このツールは、次の使用法を検証します。

1. コンテンツスイッチング、キャッシュリダイレクト、AppFW、SSL、および CMP 機能の従来のポリシー式。
2. フィルタ機能 (コンテンツフィルタとも呼ばれる)-アクション、ポリシー、およびバインディング
3. HTTP プロファイルで SPDY、Sure Connect（SC)、プライオリティキューイング（PQ)、HTTP サービス拒否（DoS)、HTML インジェクションの機能がある。
4. 負荷分散永続性ルールのクラシック式。
5. 書き換えアクションの「パターン」および「BypassSafetyCheck」パラメータ。
6. 「patclass」設定エンティティ。
7. 高度な定義式で引数なしの「HTTP.REQ.BODY」。
8. 高度な定義式の Q および S プレフィックス。
9. cmp パラメータ設定用の「policyType」パラメータ。

UNIX シェルで事前再検証ツールを実行する

コマンドプロンプトで入力します。

```
check_invalid_config <config_file>
<!--NeedCopy-->
```

例**:**

```
root@ns# check_invalid_config/nsconfig/ns.conf
```

ここで、構成ファイルは NetScaler ADC 構成ファイルです。ファイルは、`ns.conf`などの保存済み設定から取得する必要があります。

検証エラーを含むサンプル出力

以下は、NetScaler ADC バージョン 13.1 でエラーが発生した構成ファイルの出力例です。

```
add cmp policy cmp_pol -rule ns_true -resAction GZIP
add cs policy cs_pol_2 -rule ns_true
add cs policy cs_pol_3 -domain www.abc.com
```



```
add cs policy cs_pol_4 -url "/abc"
add rewrite action act_1 replace_all "http.req.body(1000)" http.req.url
    -pattern abcd
add rewrite action act_123 replace_all http.req.url ""aaaa"" -pattern
   abcd
add responder action ract respondwith "Q.URL + Q.HEADER("abcd")"
add appfw policy aff_pol_1 "http.req.body.length.gt(10)" APPFW_BYPASS
add appfw policy aff_pol ns_true APPFW_BYPASS

<!--NeedCopy-->
```

これらのエラーが発生した場合は、nspepiアップグレードツールを使用して設定を変換するか、手動で設定を変換できます。詳細については、nspepi ツールのトピックを参照してください 。

> 注:
>
> このnspepiツールは、NetScaler ADC バージョン 12.1、13.0 およびそれ以降のバージョンでのみ実行できます。

検証エラーのないサンプル出力

次に、削除または無効な設定がない設定ファイルの出力例を示します。

```
root@ns# check_invalid_config /var/tmp/new_ns.conf
No issue detected with the configuration.
root@ns#
<!--NeedCopy-->
```

従来のポリシー廃止に関するよくある質問

August 29, 2023

- **NetScaler 12.0** 以降のリリースで廃止された従来のポリシーは何ですか**?**

  「非推奨ポリシー 」表に記載されているすべての機能と機能は、NetScaler ADC リリース 12.0 ビルド 56.20 から廃止されます。廃止された機能とポリシーの詳細については、次の表（PDF 形式）を参照してください。

  - 非推奨のポリシーとその代替ポリシーについては、表 1 。
  - 表 2 非推奨の NetScaler ADC 機能とその代替機能と構成の詳細を示します。

- 従来のポリシーベースの機能と機能を高度なポリシーに変換するにはどうすればよいですか。

  NetScaler ADC 独自のnspepiツールを使用して、コマンド、式、構成を変換できます。nspepi ツールは、NetScaler ADC 構成のすべてのクラシック式を高度なポリシー式に変換するのに役立ちます。このnspepiツールの詳細については、「NSPEPI ツールを使用したポリシー式の変換」を参照してください。

- クラシックポリシーベースの機能および機能が廃止されるのはどのリリースからですか。

  NetScaler 12.0 は 56.20 以降をビルドします。

- **NetScaler ADC** アプライアンスから削除された非推奨のポリシーベースの機能はどのリリースからですか**?**

  NetScaler ADC バージョン 13.1 以降。詳細については、「非推奨のポリシーテーブル」を参照してください。

- クラシックポリシーベースの機能をサポートしないビルドにアプライアンスをアップグレードするときに従うべき手順は何ですか。

  NetScaler リリース 13.1 以降、従来のポリシーベースの機能はサポートされていません。NetScaler 13.1 以降のリリースにアップグレードする前に、`nspepi`ツールを実行して`ns.conf`ファイルを変換することをお勧めします。`nspepi`ツールについて詳しくは、「NSPEPI ツールを使用したポリシー表現の変換」を参照してください。

- **NetScaler ADC** アプライアンスで非推奨の機能がサポートされる期間はどれくらいですか?

  NetScaler は、NetScaler リリース 13.0 以降のリリースでのクラシックポリシーとその使用法をサポートしません。

  12.0 ビルド 56.20 以降、従来のポリシーと式は非推奨 (使用は推奨されず、削除されません) されました。ポリシーと式は、リリース 13.0 のすべてのビルドで以前と同じようにすべての場所で引き続き機能します。ただし、NetScaler 13.1 以降のリリースでは、クラシックポリシーベースの特定の機能が削除されました。

- 構成ファイルを変換した後、アプライアンスを再起動する必要がありますか。

  はい、`ns.config`ファイルの変換に成功したら、NetScaler ADC インスタンスを再起動する必要があります。

先に進む前に

August 15, 2023

式とポリシーを構成する前に、関連する NetScaler ADC 機能とデータの構造を次のように理解してください。

- 関連する機能に関するドキュメントをお読みください。
- 設定するデータのタイプをデータストリームで調べます。

構成するトラフィックまたはコンテンツのタイプに対してトレースを実行することもできます。これにより、式で指定する必要があるパラメータと値、およびこれらのパラメータと値に対する操作が理解できます。

注: NetScaler ADC は、機能内で高度なポリシーをサポートしています。同じフィーチャーに両方のタイプを含めることはできません。過去数回のリリースで、一部の NetScaler ADC 機能は、ポリシーと式の使用から高度なポリシーと式に移行されました。関心のある機能が Advanced ポリシー形式に変更されている場合は、古い情報を手動で移行する必要がある場合があります。次に、ポリシーを移行する必要があるかどうかを判断するためのガイドラインを示します。

- リリース 9.0 より前のバージョンの統合キャッシュ機能でクラシックポリシーを設定し、バージョン 9.0 以降にアップグレードしても、影響はありません。すべてのレガシーポリシーは、高度なポリシー形式に移行されます。
- その他の機能については、機能が Advanced ポリシーに移行されている場合は、従来のポリシーと式を Advanced 構文に手動で移行する必要があります。

高度なポリシーインフラストラクチャを設定

August 15, 2023

DNS、リライト、レスポンダー、統合キャッシュなどのさまざまな NetScaler 機能や、NetScaler Gateway のクライアントレスアクセス機能に関する高度なポリシーを作成できます。ポリシーは、これらの機能の動作を制御します。

ポリシーを作成するときは、名前、ルール (式)、機能固有の属性、およびデータがポリシーと一致した場合に実行されるアクションを割り当てます。ポリシーを作成したら、ポリシーをグローバルにバインドするか、仮想サーバーの要求時処理または応答時間処理のいずれかにバインドすることで、いつ呼び出されるかを決定します。

同じバインドポイントを共有するポリシーは、ポリシーバンクと呼ばれます。たとえば、仮想サーバにバインドされているすべてのポリシーは、仮想サーバのポリシーバンクを構成します。ポリシーをバインドするときは、銀行内の他のポリシーと比較して、ポリシーをいつ呼び出されるかを指定する優先度を割り当てます。優先度を割り当てるだけでなく、Goto 式を指定することで、銀行内のポリシーの評価順序を任意に設定できます。

組み込みのバインドポイントまたは仮想サーバに関連付けられているポリシーバンクに加えて、ポリシーラベルを設定できます。ポリシーラベルは、任意の名前で識別されるポリシーバンクです。ポリシーラベルとその中のポリシーを、グローバルまたは仮想サーバ固有のポリシーバンクから呼び出します。ポリシーラベルまたは仮想サーバポリシーバンクは、複数のポリシーバンクから呼び出すことができます。

一部の機能では、ポリシーマネージャを使用してポリシーを設定およびバインドできます。

ポリシーで使用される識別子の名前に関する規則

August 15, 2023

名前付き式、HTTP コールアウト、パターンセット、およびレート制限機能の識別子の名前は、ASCII アルファベットまたはアンダースコア (_) で始まる必要があります。残りの文字は ASCII 英数字でもアンダースコア (_) でもかまいません。

これらの識別子の名前は、次の予約語で始まってはいけません。

- Alt、TRUE、FALSE という単語、または Q または S の 1 文字の識別子。

- 特殊構文インジケーター RE (正規表現の場合) または XP (XPath 表現の場合)。
- エクスプレッションプレフィックス。現在のところ、次のようになっています。
  - クライアント
  - 延長
  - HTTP
  - サーバー
  - システム
  - ターゲット
  - テキスト
  - URL
  - MYSQL
  - MSSQL

さらに、これらの識別子の名前は、ポリシーインフラストラクチャで使用される列挙定数の名前と同じにすることはできません。たとえば、識別子の名前を IGNORECASE、YEAR、LATIN2_CZECH_CS (MySQL 文字セット) にすることはできません。

注: NetScaler アプライアンスは、これらの単語と列挙定数を使用して識別子を大文字と小文字を区別せずに比較します。たとえば、識別子の名前を TRUE、True、または true で始めることはできません。

ポリシーを作成または変更する

August 15, 2023

すべてのポリシーには、いくつかの共通要素があります。ポリシーを作成するには、少なくとも、ポリシーに名前を付け、ルールを設定する必要があります。さまざまな機能のポリシー設定ツールには重複する部分がありますが、相違点もあります。アクションとポリシーの関連付けなど、特定の機能に対するポリシーの設定の詳細については、その機能のドキュメントを参照してください。

ポリシーを作成するには、まずポリシーの目的を決定することから始めます。たとえば、イメージファイルの HTTP リクエストや SSL 証明書を含むクライアントリクエストを識別するポリシーを定義したい場合があります。ポリシーに適用する情報の種類を知るだけでなく、ポリシーが分析するデータの形式も知っておく必要があります。

次に、ポリシーがグローバルに適用できるのか、それとも特定の仮想サーバーに適用されるのかを判断します。また、ポリシーが評価される順序（ポリシーをバインドする方法によって決まります）が、設定しようとしているポリシーに与える影響についても考慮してください。

CLI を使用してポリシーを作成する

コマンドプロンプトで次のコマンドを入力してポリシーを作成し、構成を確認します。


```
-  add responder|dns|cs|rewrite|cache policy <policyName> -rule <
    expression> [<feature-specific information>]

-  show rewrite policy <name>
<!--NeedCopy-->
```

例 **1**:

```
add rewrite policy "pol_remove-ae" true "act_remove-ae"
Done
> show rewrite policy pol_remove-ae
            Name: pol_remove-ae
            Rule: true
            RewriteAction: act_remove-ae
            UndefAction: Use Global
            Hits: 0
            Undef Hits: 0
            Bound to: GLOBAL RES_OVERRIDE
            Priority: 90
            GotoPriorityExpression: END
Done
<!--NeedCopy-->
```

例 **2**:

```
add cache policy BranchReportsCachePolicy -rule q{
 http.req.url.query.value("actionoverride").contains("branchReport s")
    }
  -action cache
Done
show cache policy BranchReportsCachePolicy
            Name: BranchReportsCachePolicy
            Rule: http.req.url.query.value("actionoverride").contains("
               branchReports")
            CacheAction: CACHE
            Stored in group: DEFAULT
            UndefAction: Use Global
            Hits: 0
            Undef Hits: 0
Done
<!--NeedCopy-->
```

注**:** コマンド・ラインでは、ポリシー・ルール（式）内の引用符（q）をエスケープするか、q デリミタで区切る必要があります。詳細については、「高度なポリシー式の構成: はじめに」を参照してください。

GUI を使用したポリシーの作成または変更

1. ナビゲーションウィンドウで、ポリシーを構成する機能の名前を展開し、[**Policies**] をクリックします。たとえば、[コンテンツの切り替え]、[統合キャッシュ]、[**DNS**]、[書き換え]、または [レスポンダー] を選択できます。

2. 詳細ウィンドウで [追加] をクリックするか、既存のポリシーを選択して [開く] をクリックします。ポリシー設定ダイアログボックスが表示されます。
3. 次のパラメータに値を指定します。(アスタリスクは必須パラメータを示します。括弧内の用語については、「ポリシーを作成または変更するためのパラメータ」の対応するパラメータを参照してください。)
4. [作成] をクリックし、[閉じる] をクリックします。
5. [保存] をクリックします。ポリシーが追加されました。
   注: ポリシーを作成したら、設定ペインのポリシーエントリをクリックしてポリシーの詳細を表示できます。強調表示および下線付きの詳細は、対応するエンティティ (名前付き式など) へのリンクです。

ポリシー設定の例

August 15, 2023

これらの例は、ポリシーとそれに関連するアクションをコマンドラインインターフェイスで入力する方法を示しています。構成ユーティリティでは、式は統合キャッシュ機能または書き換え機能の機能構成ダイアログボックスの式ウィンドウに表示されます。

以下は、キャッシュポリシーの作成例です。キャッシュポリシーのアクションは組み込まれているため、ポリシーとは別に設定する必要がないことに注意してください。

```
1   add cache policy BranchReportsCachePolicy -rule q{
2  http.req.url.query.value("actionoverride").contains("branchReports") }
3   -action cache
4 <!--NeedCopy-->
```

書き換えポリシーとアクションの例を以下に示します。

```
1  add rewrite action myAction1 INSERT_HTTP_HEADER "myHeader" "
      valueForMyHeader"
2  add rewrite policy myPolicy1 "http.req.url.contains("myURLstring")"
      myAction1
3 <!--NeedCopy-->
```

注**:** コマンド・ラインでは、ポリシー・ルール（式）内の引用符（q）をエスケープするか、q デリミタで区切る必要があります。詳細については、「高度なポリシー式の構成: はじめに」を参照してください。

ポリシーマネージャを使用したポリシーの設定とバインド

August 15, 2023

> 警告:
>
> NetScaler 12.0 ビルド 56.20 以降では、従来のポリシー式はサポートされなくなりました。また、高度なポリシーを使用することをお勧めします。詳しくは、「高度なポリシー」を参照してください。

一部のアプリケーションでは、NetScaler ADC 構成ユーティリティに専用の Policy Manager が用意されており、ポリシーバンクの構成を簡素化できます。また、使用されていないポリシーとアクションを検索して削除することもできます。

Policy Manager は現在、書き換え、統合キャッシュ、レスポンダー、および圧縮機能で使用できます。

このセクションの手順に相当するキーボード操作を次に示します。

- Policy Manager でセルを編集するには、Tab キーでセルに移動して [F2] をクリックするか、キーボードのスペースバーを押します。
- ドロップダウンメニューのエントリを選択するには、Tab キーを使用してエントリに移動し、スペースバーを押してドロップダウンメニューを表示し、上方向キーと下方向キーを使用して目的のエントリに移動し、スペースバーをもう一度押してエントリを選択します。
- ドロップダウンメニューの選択をキャンセルするには、Esc キーを押します。
- ポリシーを挿入するには、Tab キーで挿入ポイントの上にある行に移動し、Ctrl キーを押しながら Insert キーを押すか、[ポリシーの挿入] をクリックします。
- ポリシーを削除するには、Tab キーでポリシーを含む行に移動し、Delete キーを押します。

注：ポリシーを削除すると、NetScaler ADC はバンク内の他のポリシーの Goto Expression 値を検索します。これらの Goto Expression 値のいずれかが、削除されたポリシーのプライオリティレベルと一致する場合、それらは削除されます。

ポリシーマネージャを使用してポリシーバインディングを構成する

1. ナビゲーションペインで、ポリシーを構成する機能をクリックします。選択肢は、レスポンダー、統合キャッシュ、書き換え、または圧縮です。
2. 詳細ペインで、[ポリシーマネージャ] をクリックします。
3. ポリシーバインディングの設定を完了する前の任意の時点で、[高度なポリシー（Advanced Policy）] を使用するポリシーのバインドを設定する場合は、[Switch to Advanced Policy] ボタンをクリックします。
4. レスポンダー以外の機能の場合、バインドポイントを指定するには、[要求] または [応答] をクリックし、要求時間または応答時のバインドポイントのいずれかをクリックします。オプションは、[グローバルオーバーライド]、[LB 仮想サーバ]、[CS 仮想サーバ]、[デフォルトグローバル]、または [ポリシーラベル] です。レスポンダーを構成している場合、要求と応答のフロータイプは使用できません。
5. このバインドポイントにポリシーをバインドするには、[ポリシーの挿入] をクリックし、以前に構成したポリシー、NOPOLICY ラベル、または [新しいポリシー] オプションを選択します。選択するオプションに応じて、次の選択肢があります。

- 新しいポリシー**:**「ポリシーの作成または変更」の説明に従ってポリシーを作成し、ポリシーバンク内の各エントリの形式に記載されているように、プライオリティレベル、GoTo 式、およびポリシー呼び出しを設定します。
- 既存のポリシー、**NOPOLICY**、または NOPOLICY\<feature name\>:ポリシーバンクの各エントリの表で説明されているように、プライオリティレベル、GoTo 式、およびポリシー呼び出しを設定します。**NOPOLICY** またはNOPOLICY\<feature name\> オプションは、詳細ポリシーを使用するポリシーに対してのみ使用できます。

6. 前述の手順を繰り返して、このポリシーバンクにエントリを追加します。
7. エントリのプライオリティレベルを変更するには、次のいずれかを実行します。

   - エントリの [Priority] フィールドをダブルクリックし、値を編集します。
   - ポリシーをクリックし、テーブル内の別の行にドラッグします。
   - [優先度を再生成] をクリックします。

   3 つのいずれの場合も、他のすべてのポリシーのプライオリティレベルは、必要に応じて変更され、新しい値に対応します。整数値を持つ Goto 式も自動的に更新されます。たとえば、プライオリティ値を 10 から 100 に変更すると、Goto 式の値が 10 のすべてのポリシーが値 100 に更新されます。
8. テーブルの行のポリシー、アクション、またはポリシーバンクの呼び出しを変更するには、エントリの右側にある下矢印をクリックし、次のいずれかの操作を行います。

   - ポリシーを変更するには、別のポリシー名を選択するか、[新しいポリシー] を選択して、ポリシーの作成または変更の手順に従います。
   - 「次へ移動」式を変更するには、「次」、「終了」、「USE_INVOCATION_RESULT」を選択するか、または複数を選択して、このポリシーバンク内の別のエントリの優先度レベルを返す式を入力します。
   - 呼び出しを変更するには、既存のポリシーバンクを選択するか、[新しいポリシーラベル] をクリックして、ポリシーをポリシーラベルにバインドするの手順に従います。
9. このバンクからポリシーまたはポリシーラベルの呼び出しをバインド解除するには、ポリシーまたはポリシーラベルを含む行の任意のフィールドをクリックし、[ポリシーのバインド解除] をクリックします。
10. 完了したら、[変更を適用] をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常にバインドされたことが示されます。

ポリシーマネージャを使用して未使用のポリシーを削除する

1. ナビゲーションペインで、ポリシーバンクを設定する機能をクリックします。選択肢は、[レスポンダー]、[統合キャッシュ]、または [書き換え] です。
2. 詳細ペインで、[<Feature Name> ポリシーマネージャー] をクリックします。
3. [機能名] > [ポリシーマネージャ] ダイアログボックスで、[クリーンアップ設定] をクリックします。
4. [クリーンアップ構成] ダイアログボックスで、削除する項目を選択し、[削除] をクリックします。
5. [削除] ダイアログボックスで、[はい] をクリックします。

6. ［閉じる］をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常に削除されたことが示されます。

ポリシーのバインド解除

August 15, 2023

ポリシーを再割り当てしたり削除したりする場合は、まずそのバインディングを削除する必要があります。

CLI を使用して、統合されたキャッシュ、書き換え、または圧縮の高度なポリシーをグローバルにバインド解除する

コマンドプロンプトで次のコマンドを入力して、統合キャッシュ、書き換え、または圧縮の詳細ポリシーをグローバルにバインド解除し、構成を確認します。

```
1 -  unbind cache|rewrite|cmp global <policyName> [-type req_override|
     req_default|res_override|res_default] [-priority <positiveInteger>]
2
3 -  show cache|rewrite|cmp global
4 <!--NeedCopy-->
```

例:

```
1  > unbind cache global_nonPostReq
2  Done
3  > show cache global
4      1)      Global bindpoint: REQ_DEFAULT
5              Number of bound policies: 1
6
7      2)      Global bindpoint: RES_DEFAULT
8              Number of bound policies: 1
9
10 Done
11 <!--NeedCopy-->
```

優先度は、NOPOLICY という名前の「ダミー」ポリシーにのみ必要です。

CLI を使用してレスポンダーポリシーをグローバルにバインド解除する

コマンドプロンプトで次のコマンドを入力して、レスポンダーポリシーをグローバルにバインド解除し、構成を確認します。

```
1 -  unbind responder global <policyName> [-type override|default] [-
     priority <positiveInteger>]
```



```
2
3  -  show responder global
4  <!--NeedCopy-->
```

例:

```
1  > unbind responder global pol404Error
2  Done
3  > show responder global
4      1)      Global bindpoint: REQ_DEFAULT
5              Number of bound policies: 1
6  Done
7  <!--NeedCopy-->
```

優先度は、NOPOLICY という名前の「ダミー」ポリシーにのみ必要です。

CLI を使用して DNS ポリシーをグローバルにバインド解除する

コマンドプロンプトで次のコマンドを入力して DNS ポリシーをグローバルにバインド解除し、構成を確認します。

```
1  -  unbind responder global <policyName>
2
3  -  unbind responder global
4  <!--NeedCopy-->
```

例:

```
1  unbind dns global dfgdfg
2  Done
3  show dns global
4      Policy name : dfgdfggfhg
5              Priority : 100
6              Goto expression : END
7  Done
8  <!--NeedCopy-->
```

CLI を使用して仮想サーバーから高度なポリシーをバインド解除する

コマンドプロンプトで次のコマンドを入力して、高度なポリシーを仮想サーバーからバインド解除し、構成を確認します。

```
1  -  unbind cs vserver <name> -policyName <policyName> [-priority <
       positiveInteger>] [-type REQUEST|RESPONSE]
2
3  -  show lb vserver <name>
4  <!--NeedCopy-->
```

例:


```
unbind cs vserver vs-cont-switch  -policyName pol1
Done
> show cs vserver vs-cont-switch
            vs-cont-switch (10.102.29.10:80) - HTTP Type: CONTENT
            State: UP
            Last state change was at Wed Aug 19 08:56:55 2009 (+18 ms)
            Time since last state change: 0 days, 02:47:55.750
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            State Update: DISABLED
            Default:        Content Precedence: RULE
            Vserver IP and Port insertion: OFF
            Case Sensitivity: ON
            Push: DISABLED  Push VServer:
            Push Label Rule: none
Done
<!--NeedCopy-->
```

優先度は、NOPOLICY という名前の「ダミー」ポリシーにのみ必要です。

GUI を使用して、統合されたキャッシュ、レスポンダー、書き換え、または圧縮の詳細ポリシーをグローバルにバインド解除する

1. ナビゲーションペインで、バインドを解除するポリシーが適用されている機能 (たとえば、統合キャッシュ) をクリックします。
2. 詳細ペインで、[<Feature Name> ポリシーマネージャー] をクリックします。
3. 「ポリシーマネージャ」ダイアログ・ボックスで、バインドを解除したいポリシーのバインド・ポイント（たとえば、Advanced Global）を選択します。
4. バインドを解除するポリシー名をクリックし、[ポリシーのバインド解除] をクリックします。
5. [変更を適用] をクリックします。
6. [閉じる］をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常にバインド解除されたことが示されます。

GUI を使用して DNS ポリシーをグローバルにバインド解除する

1. [トラフィック管理] > [**DNS**] > [ポリシー] に移動します。
2. 詳細ウィンドウで、[グローバルバインド] をクリックします。
3. 「グローバル・バインディング」ダイアログ・ボックスで、ポリシーを選択し、「ポリシーのバインド解除」をクリックします。
4. [**OK**］をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常にバインド解除されたことが示されます。

GUI を使用して、負荷分散またはコンテンツスイッチング仮想サーバーから高度なポリシーをバインド解除します

1. [トラフィック管理] に移動し、[負荷分散] または [コンテンツスイッチング] を展開し、[仮想サーバー] をクリックします。
2. 詳細ペインで、ポリシーをバインド解除する仮想サーバーをダブルクリックします。
3. 「ポリシー」タブの「アクティブ」列で、バインドを解除するポリシーの横にあるチェックボックスをオフにします。
4. [**OK**] をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常にバインド解除されたことが示されます。

ポリシーラベルの作成

December 8, 2023

ポリシーバンクを設定する組み込みのバインドポイントに加えて、ユーザー定義のポリシーラベルを設定してポリシーを関連付けることもできます。

ポリシーラベル内では、ポリシーをバインドし、ポリシーラベルのポリシーバンク内の他のポリシーと比較して各ポリシーの評価順序を指定します。NetScaler では、次のように任意の評価順序を定義することもできます。

- 「goto」表現を使うと、現在のエントリに続いて評価されるバンク内の次のエントリを指すことができます。
- ポリシーバンクのエントリを使用して別のバンクを呼び出すことができます。

各機能によって、ポリシーラベルにバインドできるポリシーのタイプ、ラベルをバインドできる負荷分散仮想サーバーのタイプ、およびラベルを呼び出すことができるコンテンツスイッチング仮想サーバーのタイプが決まります。たとえば、TCP ポリシーラベルは TCP 負荷分散仮想サーバーにのみバインドできます。HTTP ポリシーをこのタイプのポリシーラベルにバインドすることはできません。また、TCP ポリシーラベルは TCP コンテンツスイッチング仮想サーバーからのみ呼び出すことができます。

新しいポリシーラベルを設定したら、組み込みのバインドポイント用に 1 つ以上のバンクからそのポリシーラベルを呼び出すことができます。

CLI を使用してキャッシュポリシーラベルを作成する

コマンドプロンプトで次のコマンドを入力してキャッシュポリシーラベルを作成し、構成を確認します。

```
1 -  add cache policylabel <labelName> -evaluates req|res
2
3 -  show cache policylabel<labelName>
4 <!--NeedCopy-->
```


例:

```
> add cache policylabel lbl-cache-pol -evaluates req
Done

> show cache policylabel lbl-cache-pol
            Label Name: lbl-cache-pol
            Evaluates: REQ
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

CLI を使用してコンテンツスイッチングポリシーラベルを作成する

コマンドプロンプトで次のコマンドを入力して、コンテンツスイッチングポリシーラベルを作成し、構成を確認します。

```
-  add cs policylabel <labelName> http|tcp|rtsp|ssl

-  show cs policylabel <labelName>
<!--NeedCopy-->
```

例:

```
> add cs policylabel lbl-cs-pol http
Done
> show cs policylabel lbl-cs-pol
            Label Name: lbl-cs-pol
            Label Type: HTTP
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

CLI を使用して書き換えポリシーラベルを作成する

コマンドプロンプトで次のコマンドを入力して書き換えポリシーラベルを作成し、構成を確認します。

```
-  add rewrite policylabel <labelName> http_req|http_res|url|text|
   clientless_vpn_req|clientless_vpn_res

-  show rewrite policylabel <labelName>
<!--NeedCopy-->
```

例:

```
> add rewrite policylabel lbl-rewrt-pol http_req
Done
```



```
3
4  > show rewrite policylabel lbl-rewrt-pol
5              Label Name: lbl-rewrt-pol
6              Transform Name: http_req
7              Number of bound policies: 0
8              Number of times invoked: 0
9  Done
10 <!--NeedCopy-->
```

CLI を使用してレスポンダーポリシーラベルを作成する

コマンドプロンプトで次のコマンドを入力して、レスポンダーポリシーラベルを作成し、構成を確認します。

```
1  -  add responder policylabel <labelName>
2
3  -  show responder policylabel <labelName>
4  <!--NeedCopy-->
```

例:

```
1  > add responder policylabel lbl-respndr-pol
2  Done
3
4  > show responder policylabel lbl-respndr-pol
5              Label Name: lbl-respndr-pol
6              Number of bound policies: 0
7              Number of times invoked: 0
8  Done
9  <!--NeedCopy-->
```

注: ポリシーバンクからこのポリシーラベルを呼び出してください。詳細については、「ポリシーをポリシーラベルにバインドする」セクションを参照してください。

GUI を使用してポリシーラベルを作成する

1. ナビゲーションペインで、ポリシーラベルを作成したい機能を展開し、「ポリシーラベル」をクリックします。たとえば、書き換えポリシーラベルを作成するには、**[AppExpert] > [**書き換え**]** に移動し、**[**ポリシーラベル**]** をクリックします。
2. 詳細ペインで、[追加] をクリックします。
3. 「名前」ボックスに、このポリシーラベルの固有の名前を入力します。
4. ポリシーラベルに機能固有の情報を入力します。たとえば、書き換えポリシーラベル固有の情報については、「書き換えポリシーラベルの設定」を参照してください。
5. [作成] をクリックします。
6. このポリシーラベルを呼び出すように、組み込みポリシーバンクのいずれかを設定します。詳細については、「ポリシーをポリシーラベルにバインドする 」セクションを参照してください。ステータスバーに、ポリシーラベルが正常に作成されたことを示すメッセージが表示されます。

ポリシーラベルの作成に関するその他の例については、以下を参照してください:

- 認証ポリシーラベルを作成する
- コンテンツスイッチングポリシーラベルの設定
- 統合キャッシュにポリシーラベルを設定

ポリシーをポリシーラベルにバインドする

組み込みのバインドポイントにバインドされているポリシーバンクと同様に、ポリシーラベルの各エントリはポリシーラベルにバインドされたポリシーです。グローバルまたは仮想サーバにバインドされているポリシーと同様に、ポリシーラベルにバインドされている各ポリシーは、現在のエントリが処理された後に評価されるポリシーバンクまたはポリシーラベルを呼び出すこともできます。次の表は、ポリシーラベルのエントリをまとめたものです。

- **Name**: ポリシーの名前、またはポリシーを評価せずに別のポリシーバンクを呼び出す場合は、「ダミー」ポリシー名 NOPOLICY。

  ポリシーバンクでは NOPOLICY を複数回指定できますが、名前付きポリシーは 1 回しか指定できません。

- **Priority**: 整数。この設定は Goto エクスプレッションで使用できます。
- 「表現」に移動してください。この銀行で次に評価する方針を決定します。次の値のいずれかを指定できます。

  - **[**次へ**]**。次に優先度の高いポリシーに移動します。
  - 終わり。評価を停止します。
  - 使用 _ 呼び出し _ 結果。このエントリが別のポリシーバンクを呼び出す場合に適用されます。呼び出されたバンクの最後の Goto の値が END の場合、評価は停止します。最後の Goto が END 以外の場合は、現在のポリシーバンクが NEXT を実行します。
  - 正の数**:** 次に評価されるポリシーの優先度番号。
  - 数値表現。次に評価されるポリシーのプライオリティ番号を生成する式。

  Goto は政策銀行でしか前進できない。

  Goto 式を省略すると、END を指定するのと同じになります。

- 呼び出しタイプ。ポリシーバンクの種類を指定します。値は次のいずれかになります。

  - 仮想サーバーをリクエストします。仮想サーバーに関連付けられている要求時ポリシーを呼び出します。
  - レスポンスサーバー。仮想サーバーに関連付けられた応答時間ポリシーを呼び出します。
  - ポリシーラベル。銀行のポリシーラベルで示されているように、別のポリシーバンクを呼び出します。

- 呼び出し名。［呼び出しタイプ］に指定した値に応じて、仮想サーバーまたはポリシー・ラベルの名前。

ポリシーラベルまたは仮想サーバポリシーバンクの設定

August 15, 2023

ポリシーを作成し、ポリシーをバインドしてポリシーバンクを作成したら、ラベルまたはポリシーバンク内でポリシーの追加設定を行うことができます。たとえば、外部ポリシーバンクの呼び出しを設定する前に、そのポリシーバンクの設定が完了するまで待つ必要がある場合があります。

このセクションでは、以下のトピックについて説明します:

- ポリシーラベルの設定
- 仮想サーバーのポリシーバンクの設定

ポリシーラベルの設定

ポリシーラベルは、一連のポリシーと、他のポリシーラベルおよび仮想サーバ固有のポリシーバンクの呼び出しで構成されます。Invoke パラメータを使用すると、ポリシーラベルまたは仮想サーバ固有のポリシーバンクを他のポリシーバンクから呼び出すことができます。専用の NoPolicy エントリを使用すると、式 (ルール) を処理せずに外部バンクを呼び出すことができます。NoPolicy エントリは、ルールを含まない「ダミー」ポリシーです。

NetScaler コマンドラインからポリシーラベルを構成する場合は、以下のコマンド構文の詳細に注意してください。

- gotoPriorityExpression は、表 2 の説明に従って構成されます。詳細ポリシーを使用したバインドポリシーの「ポリシーバンク内のエントリ」セクションの「ポリシーバンク内のエントリ」のポリシーバンク内の各エントリの形式。
- type 引数は必須です。これは、この引数がオプションである従来のポリシーをバインドするのとは異なります。
- ポリシーラベルを呼び出すのと同じ方法を使用して、仮想サーバーにバインドされているポリシーバンクを呼び出すことができます。

CLI を使用してポリシーラベルを設定する

コマンドプロンプトで次のコマンドを入力してポリシーラベルを設定し、構成を確認します。

```
1 -  bind cache|rewrite|responder policylabel <policylabelName> -
     policyName <policyName> -priority <priority> [-
     gotoPriorityExpression <gotopriorityExpression>] [-invoke reqvserver
     |resvserver|policylabel <policyLabelName>|<vserverName>]
2
3 -  show cache|rewrite|responder policylabel <policylabelName>
4 <!--NeedCopy-->
```


例:

```
bind cache policylabel _reqBuiltinDefaults -policyName  _nonGetReq -
   priority 100
Done
show cache policylabel _reqBuiltinDefaults
            Label Name: _reqBuiltinDefaults
            Evaluates: REQ
            Number of bound policies: 3
            Number of times invoked: 0
    1)      Policy Name: _nonGetReq
            Priority: 100
            GotoPriorityExpression: END
    2)      Policy Name: _advancedConditionalReq
            Priority: 200
            GotoPriorityExpression: END

    3)      Policy Name: _personalizedReq
            Priority: 300
            GotoPriorityExpression: END
Done
<!--NeedCopy-->
```

CLI を使用して **NOPOLICY** エントリを含むリライトポリシーバンクからポリシーラベルを呼び出す

コマンドプロンプトで次のコマンドを入力して、NOPOLICY エントリを含む Rewrite ポリシーバンクからポリシーラベルを呼び出し、構成を確認します。

```
-  bind rewrite global <policyName> <priority> <gotoPriorityExpression>
    -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke
   reqvserver|resvserver|policylabel <policyLabelName>|<vserverName>

-  show rewrite global
<!--NeedCopy-->
```

例:

```
> bind rewrite global NOPOLICY 100 -type REQ_DEFAULT -invoke
   policylabel lbl-rewrt-pol
Done
> show rewrite global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

    2)      Global bindpoint: REQ_OVERRIDE
            Number of bound policies: 1
Done
<!--NeedCopy-->
```


CLI を使用して統合キャッシュポリシーバンクからポリシーラベルを呼び出す

コマンドプロンプトで次のコマンドを入力して、統合キャッシュポリシーバンクからポリシーラベルを呼び出し、構成を確認します。

```
1 -  bind cache global NOPOLICY -priority <priority> -
     gotoPriorityExpression <gotopriorityExpression> -type REQ_OVERRIDE|
     REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke reqvserver|resvserver|
     policylabel <policyLabelName>|<vserverName>
2
3 -  show cache global
4 <!--NeedCopy-->
```

例:

```
1 bind cache global NOPOLICY -priority 100 -gotoPriorityExpression END -
     type REQ_DEFAULT -invoke policylabel lbl-cache-pol
2 Done
3 > show cache global
4      1)      Global bindpoint: REQ_DEFAULT
5              Number of bound policies: 2
6
7      2)      Global bindpoint: RES_DEFAULT
8              Number of bound policies: 1
9
10 Done
11 <!--NeedCopy-->
```

CLI を使用してレスポンダーポリシーバンクからポリシーラベルを呼び出す

コマンドプロンプトで次のコマンドを入力して、レスポンダーポリシーバンクからポリシーラベルを呼び出し、構成を確認します。

```
1 -  bind responder global NOPOLICY <priority> <gotopriorityExpression> -
     type OVERRIDE|DEFAULT -invoke vserver|policylabel <policyLabelName
     >|<vserverName>
2
3 -  show responder global
4 <!--NeedCopy-->
```

例:

```
1 > bind responder global NOPOLICY 100 NEXT -type DEFAULT -invoke
     policylabel lbl-respndr-pol
2 Done
3 > show responder global
4      1)      Global bindpoint: REQ_DEFAULT
5              Number of bound policies: 2
6
7 Done
```



```
<!--NeedCopy-->
```

GUI を使用してポリシーラベルを設定する

1. ナビゲーションペインで、ポリシーラベルを設定する機能を展開し、「ポリシーラベル」をクリックします。選択肢は、統合キャッシュ、リライト、またはレスポンダーです。
2. 詳細ペインで、設定するラベルをダブルクリックします。
3. このポリシーラベルに新しいポリシーを追加する場合は、[ポリシーの挿入] をクリックし、[ポリシー名] フィールドで [新しいポリシー] を選択します。ポリシーの追加の詳細については、「ポリシーを作成または変更する」を参照してください。ポリシーバンクを起動し、呼び出しの前に規則を評価しない場合は、[Insert Policy] をクリックし、[Policy Name] フィールドで [NOPOLICY] を選択します。
4. このポリシーラベルの各エントリについて、以下を設定します。

   - ポリシー名**:**

     これは、このバンクに挿入したポリシー名、新しいポリシー、または NOPOLICY エントリによってすでに決定されています。

   - 優先度**:**

     銀行内での評価の絶対順序を決定する数値、または Goto 式と組み合わせて使用される数値。

   - 式**:**

     ポリシールール。ポリシー式については、次の章で詳しく説明します。概要については、「高度なポリシー式の設定: はじめに」を参照してください。

   - アクション**:**

     このポリシーが TRUE と評価された場合に実行されるアクション。

   - **Goto** 表現**:**

     オプションです。次に評価する保険や政策銀行を決定する際の優先度レベルを補うために使用されます。Goto 式で使用可能な値の詳細については、表 2 を参照してください。詳細ポリシーを使用したバインドポリシーの「ポリシーバンク内のエントリ」セクションの「ポリシーバンク内のエントリ」のポリシーバンク内の各エントリの形式。

   - 呼び出し**:**

     オプションです。別のポリシーバンクを呼び出します。

5. [**OK**] をクリックします。ステータスバーに、ポリシーラベルが正常に設定されたことを示すメッセージが表示されます。

仮想サーバーのポリシーバンクの設定

仮想サーバーのポリシーバンクを構成できます。ポリシーバンクには個別のポリシーを含めることができ、ポリシーバンクの各エントリは、オプションでポリシーラベルまたは別の仮想サーバに設定したポリシーバンクを呼び出すことができます。ポリシーラベルまたはポリシーバンクを呼び出す場合は、ポリシー名の代わりに NOPOLICY「ダミー」エントリを選択することで、式（ルール）をトリガーせずに呼び出すことができます。

CLI を使用して仮想サーバーのポリシーバンクにポリシーを追加します

コマンドプロンプトで次のコマンドを入力して、仮想サーバーポリシーバンクにポリシーを追加し、構成を確認します。

```
1 -  bind lb|cs vserver <virtualServerName> <serviceType> [-policyName <
     policyName>] [-priority <positiveInteger>] [-gotoPriorityExpression
     <expression>] [-type REQUEST|RESPONSE]
2
3 -  show lb|cs vserver <virtualServerName>
4 <!--NeedCopy-->
```

例**:**

```
1  add lb vserver vs-cont-sw TCP
2  Done
3  show lb vserver vs-cont-sw
4              vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
5              State: DOWN
6              Last state change was at Wed Aug 19 10:04:02 2009 (+279 ms)
7              Time since last state change: 0 days, 00:02:14.420
8              Effective State: DOWN
9              Client Idle Timeout: 9000 sec
10             Down state flush: ENABLED
11             Disable Primary Vserver On Down : DISABLED
12             No. of Bound Services :  0 (Total)       0 (Active)
13             Configured Method: LEASTCONNECTION
14             Mode: IP
15             Persistence: NONE
16             Connection Failover: DISABLED
17 Done
18 <!--NeedCopy-->
```

CLI を使用して **NOPOLICY** エントリを含む仮想サーバポリシーバンクからポリシーラベルを呼び出す

コマンドプロンプトで次のコマンドを入力して、NOPOLICY エントリを含む仮想サーバーポリシーバンクからポリシーラベルを呼び出し、構成を確認します。

```
1 -  bind lb|cs vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
     NOPOLICY-CACHE|NOPOLICY-RESPONDER -priority <integer> -type REQUEST|
```



```
    RESPONSE -gotoPriorityExpression <gotoprioritыExpression> -invoke
    reqVserver|resVserver|policyLabel <vserverName>|<labelName>

-  show lb vserver
<!--NeedCopy-->
```

例**:**

```
> bind lb vserver vs-cont-sw -policyname NOPOLICY-REWRITE -priority 200
    -type REQUEST -gotoPriorityExpression NEXT -invoke policyLabel lbl-
    rewrt-pol
Done
<!--NeedCopy-->
```

GUI を使用して仮想サーバポリシーバンクを設定

1. 左側のナビゲーションペインで、必要に応じて **[**トラフィック管理**] > [**** 負荷分散**]**、[**** **** トラフィック管理 ****] > [**** コンテンツスイッチング]、[トラフィック管理] > [**SSL** オフロード]、[セキュリティ] > [**AAA-**アプリケーショントラフィック]、または [**0 NetScaler Gateway]** を展開し、**[**仮想サーバー**]** をクリックします。**
2. 詳細ペインで、構成する仮想サーバーを選択し、「開く」をクリックします。
3. 「仮想サーバーの構成」ダイアログ・ボックスで、「ポリシー」タブをクリックします。
4. このバンクに新しいポリシーを作成するには、仮想サーバーのポリシーバンクに追加するポリシーの種類またはポリシーラベルのアイコンをクリックし、[ポリシーの挿入] をクリックします。ポリシー・ルールを評価せずにポリシー・ラベルを呼び出す場合は、NOPOLICY「ダミー」ポリシーを選択することに注意してください。
5. このポリシーバンクの既存のエントリを設定するには、次のように入力します。

   - 優先度**:**

     銀行内での評価の絶対順序を決定する数値、または Goto 式と組み合わせて使用される数値。

   - 式**:**

     ポリシールール。ポリシー式については、次の章で詳しく説明します。概要については、「高度なポリシー式の設定: はじめに」を参照してください。

   - アクション**:**

     このポリシーが TRUE と評価された場合に実行されるアクション。

   - **Goto** 表現**:**

     オプションです。次のポリシーまたはポリシーバンクの評価を決定します。Goto 式の指定可能な値の詳細については、「詳細ポリシーを使用してポリシーをバインドする」の「ポリシーバンク内のエントリ」を参照してください。

- 呼び出し**:**

  オプションです。別のポリシーバンクを呼び出すには、呼び出すポリシーラベルまたは仮想サーバポリシーバンクの名前を選択します。

6. [**OK**] をクリックします。ステータスバーに、ポリシーが正常に構成されたことを示すメッセージが表示されます。

ポリシーラベルまたは仮想サーバポリシーバンクの起動または削除

August 15, 2023

一度しかバインドできないポリシーとは異なり、ポリシーラベルまたは仮想サーバーのポリシーバンクは、それを呼び出すことで何度でも使用できます。呼び出しは次の 2 つの場所から実行できます。

- ポリシーバンクの名前付きポリシーのバインディングから。
- ポリシーバンクの NOPOLICY「ダミー」エントリのバインディングから。

通常、ポリシーラベルは、それが呼び出されたポリシーと同じタイプである必要があります。たとえば、レスポンダーポリシーからレスポンダーポリシーラベルを呼び出します。

注: コマンドラインでポリシーバンクのグローバル NOPOLICY エントリをバインドまたはバインド解除する場合、ある NOPOLICY エントリを別の NOPOLICY エントリと区別する優先順位を指定します。

CLI を使用して書き換えまたは統合キャッシュポリシーラベルを呼び出す

コマンドプロンプトで、次のコマンドのいずれかを入力してリライトまたは統合キャッシュポリシーラベルを呼び出し、構成を確認します。

```
1 -  bind cache global <policy> -priority <positive_integer> [-
     gotoPriorityExpression <expression>] -type REQ_OVERRIDE|REQ_DEFAULT|
     RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel
     <label_name>
2
3 -  bind rewrite global<policy> -priority <positive_integer> [-
     gotoPriorityExpression <expression>] -type REQ_OVERRIDE|REQ_DEFAULT|
     RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel
     <label_name>
4
5 -  show cache global|show rewrite global
6 <!--NeedCopy-->
```

例**:**


```
> bind cache global _nonPostReq2 -priority 100 -type req_override -
    invoke
     policylabel lbl-cache-pol
Done
> show cache global
     1)      Global bindpoint: REQ_DEFAULT
             Number of bound policies: 2

     2)      Global bindpoint: RES_DEFAULT
             Number of bound policies: 1

     3)      Global bindpoint: REQ_OVERRIDE
             Number of bound policies: 1

Done
<!--NeedCopy-->
```

CLI を使用してレスポンダーポリシーラベルを呼び出す

コマンドプロンプトで次のコマンドを入力してレスポンダーポリシーラベルを呼び出し、構成を確認します。

```
-  bind responder global <policy_Name> <priority_as_positive_integer>
    [<gotoPriorityExpression>] -type REQ_OVERRIDE|REQ_DEFAULT|OVERRIDE|
    DEFAULT -invoke vserver|policylabel <label_name>

-  show responder global
<!--NeedCopy-->
```

例:

```
> bind responder global pol404Error1 300 -invoke policylabel lbl-
    respndr-pol
Done
> show responder global
     1)      Global bindpoint: REQ_DEFAULT
             Number of bound policies: 2

Done
<!--NeedCopy-->
```

CLI を使用して仮想サーバポリシーバンクを呼び出す

コマンドプロンプトで次のコマンドを入力して仮想サーバーポリシーバンクを呼び出し、構成を確認します。

```
-  bind lb vserver <vserver_name> -policyName <policy_Name> -priority <
    positive_integer> [-gotoPriorityExpression <expression>] -type
    REQUEST|RESPONSE -invoke reqvserver|resvserver|policylabel <
    policy_Label_Name>

```



```
-  bind lb vserver <vserver_name>
<!--NeedCopy-->
```

例:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 100
Done

> show lb vserver lbvip
            lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
            State: DOWN
            Last state change was at Wed Jul 15 05:54:24 2009 (+166 ms)
            Time since last state change: 28 days, 06:37:49.250
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none

    1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority:
       100   Hits: 0

    2)      Policy : pol-ssl Priority:0
    3)      Policy : ns_cmp_msapp Priority:100
    4)      Policy : cf-pol Priority:1      Inherited
Done
<!--NeedCopy-->
```

CLI を使用して書き換えまたは統合キャッシュポリシーラベルを削除する

コマンドプロンプトで次のコマンドのいずれかを入力して、書き換えまたは統合キャッシュのポリシーラベルを削除し、構成を確認します。

```
-  unbind rewrite global <policyName> -priority <positiveInteger> -type
    REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT

-  unbind cache global <policyName> -priority <positiveInteger> -type
   REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT

-  show rewrite global|show cache global
<!--NeedCopy-->
```

例:


```
> unbind rewrite global NOPOLICY -priority 100 -type REQ_OVERRIDE
> show rewrite global
Done
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

CLI を使用してレスポンダーポリシーラベルを削除する

コマンドプロンプトで次のコマンドを入力してレスポンダーポリシーラベルを削除し、構成を確認します。

```
-  unbind responder global <policyName> -priority <positiveInteger> -
   type OVERRIDE|DEFAULT

-  show responder global
<!--NeedCopy-->
```

例:

```
> unbind responder global NOPOLICY -priority 100 -type REQ_DEFAULT
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

CLI を使用して仮想サーバーポリシーラベルを削除する

コマンドプロンプトで次のコマンドのいずれかを入力して仮想サーバーのポリシーラベルを削除し、構成を確認します。

```
-  unbind lb vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
   NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority <
   positiveInteger>

-  unbind cs vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
   NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority <
   positiveInteger>

-  show lb vserver|show cs vserver
<!--NeedCopy-->
```

例:


```
> unbind lb vserver lbvip -policyName ns_cmp_msapp -priority 200
Done
> show lb vserver lbvip
            lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
            State: DOWN
            Last state change was at Wed Jul 15 05:54:24 2009 (+161 ms)
            Time since last state change: 28 days, 06:47:54.600
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none

    1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority:
        100   Hits: 0

    1)      Policy : pol-ssl Priority:0
    2)      Policy : cf-pol Priority:1      Inherited
Done
<!--NeedCopy-->
```

GUI を使用してポリシーラベルまたは仮想サーバポリシーバンクを呼び出す

1. ポリシーをグローバルにバインドする、ポリシーを仮想サーバーにバインドする、またはポリシーをポリシーラベルにバインドするで説明されているように、ポリシーをバインドします。または、ポリシー名の代わりに NOPOLICY の「ダミー」エントリを入力することもできます。これは、ポリシーバンクを評価する前にポリシーを評価したくない場合に行います。
2. [Invoke] フィールドで、トラフィックがバインドされたポリシーと一致するかどうかを評価するポリシーラベルまたは仮想サーバポリシーバンクの名前を選択します。ステータスバーのメッセージは、ポリシーラベルまたは仮想サーバーポリシーバンクが正常に呼び出されたことを示します。

GUI を使用してポリシーラベル呼び出しを削除する

1. ポリシーを開き、Invoke フィールドをクリアします。ポリシーのバインドを解除すると、ラベルの呼び出しも削除されます。ステータスバーのメッセージは、ポリシーラベルが正常に削除されたことを示します。

高度なポリシー表現の設定: はじめに

August 15, 2023

詳細ポリシーでは、詳細ポリシー表現に入力した情報に基づいてデータが評価されます。高度なポリシー表現は、データ要素（HTTP ヘッダー、ソース IP アドレス、NetScaler システム時間、POST 本文データなど）を分析します。一部の NetScaler 機能では、ポリシーで高度なポリシー表現を構成するだけでなく、ポリシーのコンテキスト外でも高度なポリシー表現を構成します。

高度なポリシー表現を作成するには、分析するデータを識別するプレフィックスを選択し、そのデータに対して実行する操作を指定します。たとえば、オペレーションでは、データの一部を指定したテキスト文字列と照合したり、テキスト文字列を HTTP ヘッダーに変換したりできます。他の操作では、返された文字列を文字列のセットまたは文字列パターンと照合します。複合式を構成するには、ブール演算子と算術演算子を指定し、括弧を使用して評価順序を制御します。

高度なポリシー表現には、従来の表現を含めることもできます。頻繁に使用する式に名前を付けると、式を繰り返し作成する必要がなくなります。

ポリシーやその他のいくつかのエンティティには、NetScaler が通過するトラフィックのパケットを評価したり、NetScaler システム自体からデータを抽出したり、外部アプリケーションに要求（「コールアウト」）を送信したり、別のデータを分析したりするために NetScaler が使用するルールが含まれています。ルールはトラフィックと比較される論理表現の形式をとり、最終的に TRUE または FALSE の値を返します。

ルールの要素自体が TRUE または FALSE、文字列、または数値を返すことができます。

詳細ポリシー表現を設定する前に、ポリシーまたは他のエンティティが評価するデータの特性を理解する必要があります。たとえば、統合キャッシュ機能を使用する場合、ポリシーによってキャッシュに保存できるデータが決まります。統合キャッシュでは、NetScaler が受信する HTTP リクエストとレスポンスの URL、ヘッダー、その他のデータを知る必要があります。この知識があれば、実際のデータと一致するポリシーを構成し、NetScaler が HTTP トラフィックのキャッシュを管理できるようにすることができます。この情報は、ポリシーで設定する必要のある式のタイプを決定するのに役立ちます。

高度なポリシー表現の基本要素

August 15, 2023

高度なポリシー表現は、少なくともプレフィックス（またはプレフィックスの代わりに使用される単一の要素）で構成されます。ほとんどの式では、プレフィックスで識別されるデータに対して実行される操作も指定されています。最大 1,499 文字の式を次のようにフォーマットします。

```
<prefix>.<operation> [<compound-operator> <prefix>.<operation>. . .]
```


各項目の意味は次のとおりです。

- <prefix>

  エクスプレッションを始めるためのアンカーポイントです。

  プレフィックスは、データ単位を識別するピリオドで区切られたキーです。たとえば、次のプレフィックスは、Content-Type という名前のヘッダーが存在するかどうかの HTTP リクエストを調べます。

  http.req.header(“Content-Type”)

  プレフィックスを単独で使用して、プレフィックスが識別するオブジェクトの値を返すこともできます。

- <operation>

  プレフィックスで識別されるデータに対して実行される評価を指定します。

  たとえば、次の式を考えてみましょう。

  http.req.header(“Content-Type”).eq(“text/html”)

  この式では、演算子コンポーネントは次のとおりです。

  eq(“text/html”)

  この演算子により、NetScaler ADC は Content-Type ヘッダーを含む HTTP リクエストを評価し、特にこのヘッダーの値が文字列「text/html」と等しいかどうかを判断します。詳細については、「操作」を参照してください。

- <compound-operator>

  は、複数のプレフィックスまたはプレフィックス.operation 要素から複合式を形成するブール演算子または算術演算子です。

  たとえば、次の式を考えてみましょう。

  http.req.header (「コンテンツタイプ」) .eq (「テキスト/html」) & http.req.url.contains (「.html」)

接頭辞

エクスプレフィックスは個別のデータを表します。たとえば、エクスプレフィックスは HTTP URL、HTTP クッキー ヘッダー、または HTTP POST リクエストの本文に含まれる文字列を表すことができます。式プレフィックスは、次のようなさまざまなデータ型を識別して返すことができます。

- TCP/IP パケット内のクライアント IP アドレス
- NetScaler システム時間
- HTTP 経由の外部コールアウト
- TCP または UDP レコードタイプ

ほとんどの場合、式のプレフィックスは次のいずれかのキーワードで始まります。

- クライアント:
  - 次の例のように、リクエストを送信しているクライアントまたはレスポンスを受信しているクライアントの特性を識別します。
  - client.ip.dst というプレフィックスは、要求または応答の宛先 IP アドレスを指定します。
  - client.ip.src というプレフィックスは、送信元 IP アドレスを指定します。
- HTTP:
  - 次の例のように、HTTP リクエストまたはレスポンス内の要素を識別します。
  - プレフィックス http.req.body (integer) は、HTTP リクエストの本文を、整数で指定された文字位置までの複数行のテキストオブジェクトとして指定します。
  - プレフィックス http.req.header (「header_name」) は、header_name で指定されている HTTP ヘッダーを指定します。
  - プレフィックス http.req.url は URL エンコード形式の HTTP URL を指定します。
- サーバー:

  要求を処理しているか、応答を送信しているサーバー内の要素を識別します。
- システム:

  トラフィックを処理している NetScaler の特性を識別します。

  注**:** DNS ポリシーは SYS、CLIENT、SERVER オブジェクトのみをサポートしていることに注意してください。

  さらに、NetScaler Gateway では、クライアントレス VPN 機能は次の種類のプレフィックスを使用できます。
- テキスト:

  リクエストまたはレスポンス内の任意のテキスト要素を識別します。
- ターゲット:

  接続のターゲットを識別します。
- URL:

  HTTP リクエストまたはレスポンスの URL 部分の要素を識別します。

一般的な経験則として、どの式のプレフィックスでも自己完結型の式にすることができます。たとえば、次のプレフィックスは、文字列引数 (引用符で囲まれた) で指定された HTTP ヘッダーの内容を返す完全な式です。

```
http.res.header.("myheader")
```

または、プレフィックスと簡単な操作を組み合わせて TRUE と FALSE の値を決定することもできます。たとえば、次の例では TRUE または FALSE の値を返します。

```
http.res.header.("myheader").exists
```


次の例のように、式内の個々のプレフィックスや複数のプレフィックスに対して複雑な演算を行うこともできます。

```
http.req.url.length + http.req.cookie.length <= 500
```

どのエクスプレッションプレフィックスを指定できるかは、NetScaler の機能によって異なります。次の表では、対象となるエクスプレッションプレフィックスを機能ごとにまとめています。

機能	機能で使用されるエクスプレッションプレフィックスのタイプ
DNS	SYS, CLIENT, SERVER
レスポンダー・イン・プロテクション機能	HTTP、システム、クライアント
コンテンツスイッチ	HTTP、システム、クライアント
書き換え	HTTP、システム、クライアント、サーバー、URL、テキスト、ターゲット、VPN
統合キャッシング	HTTP, SYS, CLIENT, SERVER
NetScaler Gateway、クライアントレスアクセス	HTTP、システム、クライアント、サーバー、URL、テキスト、ターゲット、VPN

表 1. NetScaler のさまざまな機能で使用できるエクスプレッションプレフィックスの種類

注: 機能で使用できるエクスプレッションプレフィックスの詳細については、その機能のドキュメントを参照してください。

単一要素表現

最も単純なタイプのアドバンスポリシー表現には、1 つの要素が含まれています。この要素には次のいずれかを使用できます。

- true. 高度なポリシー表現には、true という値だけで構成できます。このタイプの式は常に TRUE の値を返します。ポリシーアクションを連鎖させたり、Goto 表現をトリガーしたりするのに便利です。
- 偽。高度なポリシー表現には、単に false という値だけで構成できます。このタイプの式は常に FALSE の値を返します。
- 複合式のプレフィックス。たとえば、プレフィックス HTTP.REQ.HOSTNAME はホスト名を返す完全な式で、HTTP.REQ.URL は URL を返す完全な式です。プレフィックスを演算や追加のプレフィックスと組み合わせて使用して複合式を作成することもできます。

オペレーション

ほとんどの式では、プレフィックスで識別されるデータに対する操作も指定します。たとえば、次のプレフィックスを指定したとします。

http.req.url

このプレフィックスは、HTTP リクエストの URL を抽出します。この式のプレフィックスでは、式に演算子を使用する必要はありません。ただし、HTTP リクエスト URL を処理する式を設定すると、URL の特定の特性を分析する操作を指定できます。いくつかの可能性を以下に示します。

- URL で特定のホスト名を検索します。
- URL 内の特定のパスを検索します。
- URL の長さを評価してください。
- URL でタイムスタンプを示す文字列を検索し、GMT に変換します。

以下は、Server という名前の HTTP ヘッダーを識別するプレフィックスと、ヘッダー値の IIS という文字列を検索する操作の例です。

```
http.res.header("Server").contains("IIS")
```

ホスト名を識別するプレフィックスと、名前の値として「www.mycompany.com」という文字列を検索する操作の例を以下に示します。

```
http.req.hostname.eq("www.mycompany.com")
```

エクスプレッションプレフィックスの基本操作

次の表では、エクスプレフィックスに対して実行できる基本的な操作のいくつかを説明しています。

操作	か否かを決める
CONTAINS()	オブジェクトは と一致します。以下は例です:http.req.header (「キャッシュコントロール」) .contains (「キャッシュなし」)
EXISTS	特定のアイテムがオブジェクトに存在します。以下がその例です:http.res.header (「MyHDR」) .exists
EQ()	特定の非数値がオブジェクトに存在します。以下がその例です:http.req.method.eq (投稿)
EQ()	特定の数値がオブジェクトに存在します。以下は例です:client.ip.dst.eq (10.100.10.100)
LT()	オブジェクトの値が特定の値よりも小さい。以下がその例です:http.req.content_length.lt (5000)
GT()	オブジェクトの値が特定の値よりも大きい。以下がその例です:http.req.content_length.gt (5)

次の表は、使用可能な操作のいくつかをまとめたものです。

操作タイプ	説明
テキスト操作	個々の文字列や文字列のセットをターゲットの任意の部分と照合します。ターゲットは、文字列全体、文字列の先頭、または文字列の先頭と末尾の間にあるテキストの任意の部分です。たとえば、「XYZSomeText」から文字列「XYZ」を抽出できます。または、HTTP ヘッダー値をさまざまな文字列の配列と比較することもできます。テキストを別の種類のデータに変換することもできます。例としては、文字列を整数値に変換したり、URL 内のクエリ文字列からリストを作成したり、文字列を時間値に変換したりします。
数値演算	数値演算には、算術演算子の適用、コンテンツの長さの評価、リスト内の項目数、日付、時刻、および IP アドレスが含まれます。

複合高度なポリシー式

August 15, 2023

高度なポリシー式は、ブール演算子または算術演算子とアトミック演算で構成できます。次の複合式には、ブール値 AND があります。

```
http.req.hostname.eq("mycompany.com")&& http.req.method.eq(post)
```

次の式は、2 つのターゲットの値を加算し、その結果を 3 番目の値と比較します。

```
http.req.url.length + http.req.cookie.length \<= 500
```

複合式には、任意の数の論理演算子と算術演算子を使用できます。

次の式は、HTTP リクエストの長さを評価します。この式は URL と Cookie に基づいています。
この式は、ヘッダー内のテキストを評価します。また、この 2 つの結果に対してブール値 AND も表示されます。

```
http.req.url.length + http.req.cookie.length \<= 500 && http.req.
header.contains("some text")
```

括弧を使用して、複合式での評価の順序を制御できます。

複合式のブール値

複合式は、次の演算子を使用して設定します。

- &&.

  この演算子は論理 AND です。式を TRUE と評価するには、すべてのコンポーネントが TRUE と評価される必要があります。

  例:

  http.req.url.hostname.eq(“myHost”) && http.req.header(“myHeader”).exists

- ||.

  この演算子は論理 OR です。式のいずれかのコンポーネントが TRUE と評価された場合、式全体が TRUE になります。

- !.

  P 式に論理 NOT を指定します。

NetScaler ADC 構成ユーティリティでは、［式の追加］ダイアログボックスに AND、NOT、OR 演算子が表示されることがあります。ただし、これらの複合式は限定的に使用できます。演算子 &&、||、！を使用することをお勧めしますブール論理を使用する複合式を構成するには。

複合式の括弧

括弧を使用して、式の評価順序を制御できます。以下はその例です:

```
http.req.url.contains("myCompany.com")|| (http.req.url.hostname.eq("myHost")&& http.req.header("myHeader").exists)
```

次は別の例です。

```
(http.req.header("Content-Type").exists && http.req.header("Content-Type").eq("text/html"))|| (http.req.header("Transfer-Encoding").exists || http.req.header("Content-Length").exists)
```

文字列の複合演算

次の表では、文字列データに対して複合演算を設定するために使用できる演算子について説明します。

文字列値を生成する操作	説明
str + str	演算子の左側の式の値と右側の値を連結します。 例:http.req.hostname + http.req.url.protocol
str + num	演算子の左側にある式の値と、右側の数値を連結します。 例:http.req.hostname + http.req.url.content_length

文字列値を生成する操作	説明
num + str	演算子の左側にある式の数値を、右側の文字列値と連結します。例:http.req.url.content_length + http.req.url.hostname
str + ip	演算子の左側にある式の文字列値を、右側の IP アドレス値と連結します。例:http.req.hostname + 10.00.000.00
IP + str	演算子の左側にある式の IP アドレス値と右側の文字列値を連結します。例:client.ip.dst + http.req.url.hostname
str1 ALT str2	string1 の評価によって undef 例外が発生するか、結果がヌル文字列である場合は string2 を使用します。それ以外の場合は string1 を使用し、string2 は決して評価しません。例:http.req.hostname alt client.ip.src

TRUE または FALSE の結果を生成する文字列に対する演算	説明
str == str	演算子の両側の文字列が同じかどうかを評価します。以下は例です:http.req.header (「myheader」) == http.res.header (「myheader」)
str <= str	演算子の左側の文字列が右側の文字列と同じか、アルファベット順で先行するかを評価します。
str >= str	演算子の左側の文字列が右側の文字列と同じであるか、アルファベット順に続くかを評価します。
str < str	演算子の左側の文字列が、アルファベット順で右側の文字列より前にあるかどうかを評価します。
str > str	演算子の左側の文字列が、アルファベット順に右側の文字列の後に続くかどうかを評価します。
str !!= str	演算子の両側の文字列が異なるかどうかを評価します。

文字列に対する論理演算	説明
bool && bool	この演算子は論理 AND です。複合式のコンポーネントを評価する場合、および、結合されたすべてのコンポーネントが TRUE と評価される必要があります。以下は例である。http.req.method.eq (GET) && http.req.url.query.contains (「viewReport && my_pagelabel」)
bool || bool	この演算子は論理 OR です。複合式のコンポーネントを評価するときに、OR に属する式のいずれかのコンポーネントが TRUE と評価されると、式全体が TRUE になります。以下は、例です。http.req.url.contains (「.js」) || http.res.header。(「コンテンツタイプ」)。(「javascript」) を含む
bool	式に対して論理 NOT を実行します。

数値の複合演算

複合数値式を設定できます。たとえば、次の式は、HTTP ヘッダーの長さと URL の長さの合計を表す数値を返します。

```
http.req.header.length + http.req.url.length
```

次の表では、数値データの複合式を構成するために使用できる演算子について説明します。

数値の算術演算	説明
num + num	演算子の左側の式の値を、右側の式の値に加算します。以下は例である。http.req.content_length + http.req.url.length
num –num	演算子の右側の式の値を、左側の式の値から減算します。
num*num	演算子の左側の式の値と、右側の式の値を乗算します。次に例を示します。client.interface.rxthroughput* 9
num / num	演算子の左側の式の値を、右側の式の値で割ります。
num % num	モジュロ、または演算子の左側の式の値を、右側の式の値で除算した剰余を計算します。たとえば、値「15 mod 4」は 3、「12 mod 4」の値は 0 です。

数値の算術演算	説明
~number	数値のビット単位の論理否定を適用した後の数値を返します。次の例では、数値.expression が 12 (バイナリ 1100) を返すと仮定しています:~numeric.expression。～演算子を適用した結果は、-11（バイナリ 1110011、合計 32 ビット、すべてが左側）です。演算子を暗黙的に適用する前の 32 ビット未満のすべての戻り値は、32 ビット幅になるように左にゼロがあることに注意してください。
number ^ number	等しい長さの 2 つのビットパターンを比較し、各 number 引数の対応するビットの各ペアに対して XOR 演算を実行します。ビットが異なる場合は 1 を返し、同じ場合は 0 を返します。整数引数と現在の数値にビット単位の排他的論理和を適用した後の数値を返します。ビット単位の比較の値が同じ場合、戻り値は 0 です。次の例では、数値. 式 1 は 12 (バイナリ 1100) を返し、数値. 式 2 は 10 (バイナリ 1010) を返します。数値. 式 1 ^ 数値. 式 2 ^ 演算子を式全体に適用した結果は 6 (バイナリ 0110) です。演算子を暗黙的に適用する前の 32 ビット未満のすべての戻り値は、32 ビット幅になるように左にゼロがあることに注意してください。
数値 | 数値	数値にビット単位の OR を適用した後の数値を返します。ビット単位の比較のいずれかの値が 1 の場合、戻り値は 1 です。次の例では、数値. 式 1 が 12 (バイナリ 1100) を返し、数値. 式 2 が 10 (バイナリ 1010) を返すと仮定しています。数値. 式 1 | 数値. 式 2 | 演算子を式全体に適用した結果は 14 (バイナリ 1110) です。演算子を暗黙的に適用する前の 32 ビット未満のすべての戻り値は、32 ビット幅になるように左にゼロがあることに注意してください。

数値の算術演算	説明
number & number	等しい長さの 2 つのビットパターンを比較し、対応するビットの各ペアに対してビット単位の AND 演算を実行します。両方のビットに 1 の値が含まれている場合は 1 を返し、いずれかのビットが 0 の場合は 0 を返します。次の例では、数値. 式 1 が 12 (バイナリ 1100) を返し、数値. 式 2 は 10 (バイナリ 1010) を返すと仮定しています。数値. 式 1 と数値. 式 2 式全体が 8 (バイナリ 1000) と評価されます。演算子を暗黙的に適用する前の 32 ビット未満のすべての戻り値は、32 ビット幅になるように左にゼロがあることに注意してください。
num « num	右側の number 引数のビット数によって、数値値のビット単位の左シフトの後に数値を返します。シフトされるビット数は 32 を法とする整数であることに注意してください。次の例では、numeric.expression1 が 12 (バイナリ 1100) を返し、numeric.expression2 が 3 を返すと仮定しています。numeric.expression1 « numeric.expression2 LSHIFT 演算子を適用した結果は 96 (バイナリ 1100000) になります。演算子を適用する前に、32 ビット未満のすべての値が返されることに注意してください。暗黙的に左にゼロを持たせて、32 ビット幅にします。
num » num	整数引数のビット数による数値値のビット単位の右シフト後の数値を返します。シフトされるビット数は 32 を法とする整数であることに注意してください。次の例では、数値. 式 1 が 12 (バイナリ 1100) を返し、数値. 式 2 が 3 を返すと仮定しています。数値. 式 1 » 数値. 式 2 RSHIFT 演算子を適用した結果は 1 (バイナリ 0001) です。演算子を暗黙的に適用する前の 32 ビット未満のすべての戻り値は、32 ビット幅になるように左にゼロがあることに注意してください。

TRUE または FALSE の結果を生成する数値演算子	説明
——–	——————————————————————————
num == num	演算子の左側の式の値が、右側の式の値と等しいかどうかを判断します。
num! = num	演算子の左側の式の値が右側の式の値と等しくないかどうかを判定します。|
num > num	演算子の左側の式の値が右側の式の値より大きいかどうかを判定する。
num < num	演算子の左側の式の値が右側の式の値より小さいかどうかを判定する。

| num >= num | 演算子の左側の式の値が右側の式の値以上であるかどうかを判定する。|
| num <= num | 演算子の左側の式の値が右側の式の値以下かどうかを判定する |

ポリシーインフラストラクチャのデータ型の関数

NetScaler ADC ポリシーインフラストラクチャでは、次の数値データ型がサポートされています。

- 整数 (32 ビット)
- 符号なしロング (64 ビット)
- ダブル (64 ビット)

単純な式は、これらのすべてのデータ型を返すことができます。また、算術演算子と論理演算子を使用してこれらのデータ型の値を評価または返す複合式を作成することもできます。また、ポリシー式では、これらの値をすべて使用できます。符号なし long 型のリテラル定数は、文字列 ul を数値に追加することで指定できます。double 型のリテラル定数には、ピリオド (.)、指数、またはその両方が含まれます。

算術演算子、論理演算子、および型推進　複合式では、倍精度浮動小数点型および符号なし長整数型に対して、次の標準的な算術演算子と論理演算子を使用できます。

- +、-、*、/

%, ~, ^, &, 、<><, and > (ダブルには適用されません)

-

- ==、!=、>、<、>=、<=

これらの演算子はすべて、C プログラミング言語と同じ意味を持ちます。

整数型、符号なし long、および double 型のオペランド間の混合演算のすべての場合において。型昇格は、同じ型のオペランドに対して操作を行うために行われます。この演算では、優先順位の高いオペランドに下位優先タイプが昇格されます。優先順位の順序（高い順）は次のとおりです。

- Double
- 符号なしロング
- 整数

したがって、数値の結果を返す操作は、操作に含まれる最も高いタイプの結果を返します。

たとえば、オペランドが整数型で符号なし長型の場合、整数オペランドは自動的に符号なし長型に変換されます。この型変換は、単純な式で行われます。式の接頭辞によって識別されるデータの型が、関数に引数として渡されるデータの型と一致しません。HTTP.REQ.CONTENT_LENGTH.DIV (3ul) のオペレーションでは、接頭辞 HTTP.REQ.CONTENT_LENGTH は、符号なしロングになる整数を返します。符号なし長整数型:DIV () 関数の引数

として渡されるデータ型で、符号なし長除算が実行されます。同様に、引数は式で昇格することができます。たとえば、HTTP.REQ.HEADER (「myHeader」) .TYPECAST_DOUBLE_AT.DIV (5) は整数 5 を倍精度型に昇格させ、倍精度除算を行います。

ある型のデータを別の型のデータにキャストする式については、データの型キャストを参照してください。

エクスプレッションで文字セットを指定してください

August 15, 2023

NetScaler アプライアンスのポリシーインフラストラクチャは、ASCII および UTF-8 文字セットをサポートしています。デフォルトの文字セットは ASCII です。式を設定するトラフィックが ASCII 文字のみで構成されている場合は、式に文字セットを指定する必要はありません。アプライアンスでは、バイナリ文字を含むすべての文字列および文字リテラルを使用できます。ただし、UTF-8 文字セットでは、文字列と文字リテラルが有効な UTF-8 である必要があります。

```
CLIENT.TCP.PAYLOAD(100).CONTAINS("\xff\x02")
```

式では、式の特定の位置に SET_CHAR_SET () 関数を導入する必要があります。その後、指定された文字セットでデータ処理を実行する必要があります。<string> たとえば、HTTP.REQ.BODY (1000) .AFTER_REGEX (re/次の例/) .BEFORE_REGEX (re/前の例/) .CONTAINS_ANY (「ギリシャ語 _ アルファベット」) という式に、パターンセット「Greek_Alphabet」に格納されている文字列が UTF-8 の場合は、SET_CHAR_SET (UTF) を含める必要があります CONTAINS_ANY (「」) 関数の直前にある関数 (_8)。次のようになります。

```
HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX
(re/In the preceding example/).SET_CHAR_SET(UTF_8).CONTAINS_ANY("
Greek_ alphabet")
```

SET_CHAR_SET () 関数は、その式の後のほうで文字セットを変更する別の SET_CHAR_SET () 関数によってオーバーライドされない限り、式で以降のすべての処理 (つまり、後続のすべての関数) に文字セットを設定します。<int> したがって、特定の単純な式に含まれるすべての関数が UTF-8 を対象としている場合は、テキストを識別する関数 (たとえば、HEADER (「<name>」) や BODY () 関数) の直後に SET_CHAR_SET (UTF_8) 関数を含めることができます。上の最初の段落に続く 2 番目の例では、AFTER_REGEX () 関数と BEFORE_REGEX () 関数に渡される ASCII 引数を UTF-8 文字列に変更すると、次のように BODY (1000) 関数の直後に SET_CHAR_SET (UTF_8) 関数を含めることができます。

```
HTTP.REQ.BODY(1000).SET_CHAR_SET(UTF_8).AFTER_REGEX(re/Bücher/).
BEFORE_REGEX(re/Wörterbuch/).CONTAINS_ANY("Greek_alphabet")
```

UTF-8 文字セットは ASCII 文字セットのスーパーセットなので、文字セットを UTF-8 に変更しても、ASCII 文字セット用に設定された式は引き続き期待どおりに機能します。

文字セットが異なる複合表現

複合式で、式の 1 つのサブセットが ASCII 文字セットのデータを処理するように設定され、残りの式が UTF-8 文字セットのデータを処理するように構成されている場合、式を個別に評価する際には、個々の式に指定された文字セットが考慮されます。ただし、複合式を処理する場合、演算子を処理する直前に、アプライアンスは返された ASCII 値の文字セットを UTF-8 にプロモートします。たとえば、次の複合式では、最初の単純な式は ASCII 文字セットのデータを評価し、2 番目の単純な式は UTF-8 文字セットのデータを評価します。

```
HTTP.REQ.HEADER("MyHeader")== HTTP.REQ.BODY(10).SET_CHAR_SET(UTF_8)
```

ただし、複合式を処理する場合、「等しい」ブール演算子を評価する直前に、NetScaler アプライアンスは HTTP.REQ.HEADER（「MyHeader」）から返される値の文字セットを UTF-8 にプロモートします。

次の例の最初の簡単な式は、ASCII 文字セットのデータを評価します。ただし、NetScaler アプライアンスが複合式を処理する場合、2 つの単純な式の結果を連結する直前に、アプライアンスは HTTP.REQ.BODY (10) から返される値の文字セットを UTF-8 にプロモートします。

```
HTTP.REQ.BODY(10)+ HTTP.REQ.HEADER("MyHeader").SET_CHAR_SET(UTF_8)
```

そのため、複合式はデータを UTF-8 文字セットで返します。

トラフィックの文字セットに基づいて文字セットを指定します

トラフィックの特性に基づいて文字セットを UTF-8 に設定できます。評価対象のトラフィックの文字セットが UTF-8 かどうかわからない場合は、最初の式で UTF-8 トラフィックをチェックし、後続の式で文字セットを UTF-8 に設定する複合式を設定できます。以下は、リクエストの最初の 1000 バイトに UTF-8 文字列 Bücher が含まれているかどうかを確認する前に、最初にリクエストの Content-Type ヘッダーの「charset」の値が「UTF-8」であるかどうかをチェックする複合式の例です。

```
HTTP.REQ.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).TYPECAST_NVLIST_T
('=', '; ', '"').VALUE("charset").EQ("UTF-8")&& HTTP.REQ.BODY(1000).
SET_CHAR_SET(UTF_8).CONTAINS("Bücher")
```

評価対象のトラフィックの文字セットが UTF-8 であることが確実な場合は、例の 2 番目の式で十分です。

エクスプレッション内の文字リテラル、文字列リテラル

式の評価中、現在の文字セットが ASCII であっても、それぞれ一重引用符 (') と引用符 (「」) で囲まれた文字リテラルと文字列リテラルは UTF-8 文字セットのリテラルと見なされます。特定の式で、関数が ASCII 文字セットの文字または文字列リテラルを操作していて、リテラルに非 ASCII 文字を含めると、エラーが返されます。

> 注記:

> 高度なポリシー表現の文字列リテラルは、ポリシー表現と同じ長さになりました。式は 1499 バイトまたは 8191 バイトの長さにすることができます。

16 進形式と 8 進形式の値

式を設定する場合、8 進数および 16 進数の形式で値を入力できます。ただし、16 進数または 8 進数の各バイトは UTF-8 バイトと見なされます。無効な UTF-8 バイトは、値を手動で入力したか、クリップボードから貼り付けたかにかかわらずエラーになります。たとえば、「\ xce\ x20」は、「c8」の後に「20」を付けることはできないため、無効な UTF-8 文字です (マルチバイト UTF-8 文字列の各バイトには上位ビットが設定されている必要があります)。無効な UTF-8 文字のもう 1 つの例は、「\ xce\ xa9」です。これは、16 進文字が空白文字で区切られているためです。

UTF-8 文字列を返す関数

`text>.XPATH` and `<text>.XPATH_JSON` 関数だけが常に UTF-8 文字列を返します。次の MYSQL ルーチンは、プロトコル内のデータに応じて、どの文字セットを返すかを実行時に決定します。

- `MYSQL_CLIENT_T.USER`
- `MYSQL_CLIENT_T.DATABASE`
- `MYSQL_REQ_QUERY_T.COMMAND`
- `MYSQL_REQ_QUERY_T.TEXT`
- `MYSQL_REQ_QUERY_T.TEXT(<unsigned int>)`
- `MYSQL_RES_ERROR_T.SQLSTATE`
- `MYSQL_RES_ERROR_T.MESSAGE`
- `MYSQL_RES_FIELD_T.CATALOG`
- `MYSQL_RES_FIELD_T.DB`
- `MYSQL_RES_FIELD_T.TABLE`
- `MSQL_RES_FIELD_T.ORIGINAL_TABLE`
- `MYSQL_RES_FIELD_T.NAME`
- `MYSQL_RES_FIELD_T.ORIGINAL_NAME`
- `MYSQL_RES_OK_T.MESSAGE`
- `MYSQL_RES_ROW_T.TEXT_ELEM(<unsigned int>)`

UTF-8 の端末接続設定

ターミナル接続（PuTTY などを使用）を使用して NetScaler アプライアンスへの接続をセットアップする場合、データ転送用の文字セットを UTF-8 に設定する必要があります。

高度なポリシー表現における最小関数と最大関数

高度なポリシー表現は、以下の最小関数と最大機能をサポートします。

1. (<expression1>.max(<expression2>) -2 つの値の最大値を返します。
2. (<expression1>.min(<expression2>) -2 つの値の最小値を返します。

ポリシーで高度なポリシー式を構成する

August 15, 2023

1 つのポリシーに最大 1,499 文字の高度なポリシー式を設定できます。高度なポリシー式のユーザーインターフェイスは、式を設定している機能と、ポリシーまたは別の用途のどちらに式を設定するかによって、ある程度異なります。

コマンドラインで式を設定するときは、引用符 (「..」または‘.’) を使用して式を区切ります。式内では、バックスラッシュ () を使用して追加の引用符をエスケープします。たとえば、式で引用符をエスケープする標準的なメソッドを以下に示します。

```
"\"abc\""
```

```
'\"abc\"'
```

また、コマンドラインで疑問符やその他のバックスラッシュをエスケープするには、バックスラッシュを使用する必要があります。たとえば、式 http.req.url.contains (「?」) 疑問符を解析するには、バックスラッシュが必要です。疑問符を入力すると、コマンドラインにバックスラッシュ文字が表示されないことに注意してください。一方、バックスラッシュをエスケープすると（たとえば、式’http.req.url.contains (「\\ http」) ‘)、エスケープ文字がコマンドラインにエコーされます。

エントリを読みやすくするために、式全体の引用符をエスケープできます。式の先頭に、エスケープシーケンス「q」と、次の特殊文字のいずれかを入力します。/{<

~$^+=&%@# ポリシーで高度なポリシー式を構成する

---

1 つのポリシーに最大 1,499 文字の高度なポリシー式を設定できます。高度なポリシー式のユーザーインターフェイスは、式を設定している機能と、ポリシーまたは別の用途のどちらに式を設定するかによって、ある程度異なります。

コマンドラインで式を設定するときは、引用符 (「..」または‘.’) を使用して式を区切ります。式内では、バックスラッシュ () を使用して追加の引用符をエスケープします。たとえば、式で引用符をエスケープする標準的なメソッドを以下に示します。

<!JEKYLL@5180@0>

<!JEKYLL@5180@1>

また、コマンドラインで疑問符やその他のバックスラッシュをエスケープするには、バックスラッシュを使用する必要があります。たとえば、式 http.req.url.contains (「?」) 疑問符を解析するには、バックスラッシュが必要です。

疑問符を入力すると、コマンドラインにバックスラッシュ文字が表示されないことに注意してください。一方、バックスラッシュをエスケープすると（たとえば、式’http.req.url.contains (「\\ http」) ‘)、エスケープ文字がコマンドラインにエコーされます。

?.

次のように、式の最後に特殊文字のみを入力します。

```
q@http.req.url.contains("sometext") && http.req.cookie.exists@

q~http.req.url.contains("sometext") && http.req.cookie.exists~
<!--NeedCopy-->
```

{区切り文字を使用する式は} で閉じられることに注意してください。

一部の機能 (たとえば、統合キャッシュとレスポンダー) では、ポリシーの構成ダイアログボックスに式を構成するための第 2 のダイアログボックスが表示されます。このダイアログでは、式の設定中の各ポイントで使用可能な選択肢を示すドロップダウンリストから選択できます。これらの設定ダイアログを使用する場合は算術演算子を使用できませんが、他のほとんどの高度なポリシー式機能を使用できます。算術演算子を使用するには、式を自由形式で記述します。

CLI を使用して高度なポリシー構文規則を構成する

コマンドプロンプトで次のコマンドを入力して、詳細ポリシールールを構成し、構成を確認します。

1. `add cache|dns|rewrite|cs policyName **-rule** expression featureSpecificPa **-action**`
2. `show cache|dns|rewrite|cs policyName`
   次に、キャッシングポリシーの設定例を示します。

   例:

```
> add cache policy pol-cache -rule http.req.content_length.le(5) -
   action INVAL
 Done

> show cache policy pol-cache
        Name: pol-cache
        Rule: http.req.content_length.le(5)
        CacheAction: INVAL
        Invalidate groups: DEFAULT
        UndefAction: Use Global
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```


GUI を使用して高度なポリシー式を設定します

1. ナビゲーションウィンドウで、ポリシーを構成する機能の名前をクリックします。たとえば、[統合キャッシュ]、[レスポンダー]、[DNS]、[書き換え]、または [コンテンツスイッチング] を選択し、[ポリシー] をクリックします。
2. [追加] をクリックします。
3. ほとんどのフィーチャでは、[条件式] フィールドをクリックします。コンテンツスイッチの場合は、[構成] をクリックします。
4. [接頭辞] アイコン (家) をクリックし、ドロップダウンリストから最初の式の接頭辞を選択します。たとえば、レスポンダーでは、オプションは HTTP、SYS、およびクライアントです。次の適用可能なオプションのセットがドロップダウンリストに表示されます。
5. 次のオプションをダブルクリックして選択し、ピリオド (.) を入力します。ここでも、適用可能なオプションのセットが別のドロップダウンリストに表示されます。
6. 入力フィールド (括弧で示される) が表示されるまで、オプションの選択を続けます。入力フィールドが表示されたら、括弧内に適切な値を入力します。たとえば、[GT (int) (より大きい、整数形式)] を選択した場合は、括弧内に整数を指定します。テキスト文字列は引用符で区切られます。次に例を示します。

   ```
   HTTP.REQ.BODY(1000).BETWEEN("this","that")
   ```

複合式の 2 つの部分の間に演算子を挿入するには、[演算子] アイコン (シグマ) をクリックし、演算子のタイプを選択します。以下は、Boolean OR () (2 本の縦棒で示される) を使用して設定された式の例です。

---

7. `HTTP.REQ.URL.EQ("www.mycompany.com")||HTTP.REQ.BODY(1000).BETWEEN("this","that")`
8. 名前付き式を挿入するには、[追加] アイコン (プラス記号) の横にある下向き矢印をクリックし、名前付き式を選択します。
9. ドロップダウンメニューを使用して式を構成し、組み込み式を挿入するには、[追加] アイコン (プラス記号) をクリックします。[式の追加] ダイアログボックスは、メインダイアログボックスと同様に機能しますが、オプションを選択するためのドロップダウンリストが表示され、かっこではなくデータ入力用のテキストフィールドが表示されます。このダイアログボックスには、よく使用する式を挿入する [よく使用する式] ドロップダウンリストもあります。式の追加が完了したら、「**OK**」をクリックします。
10. 入力が終わったら、[作成] をクリックします。ステータスバーにメッセージが表示され、ポリシー式が正常に設定されたことが示されます。

GUI を使用して高度なポリシー式をテストする

1. ナビゲーションウィンドウで、ポリシーを構成する機能の名前をクリックし (たとえば、[統合キャッシュ]、[レスポンダー]、[DNS]、[書き換え]、または [コンテンツスイッチング] を選択できます)、[ポリシー] をクリ

ックします。

2. ポリシーを選択し、[**Open**] をクリックします。
3. 式をテストするには、[評価] アイコン (チェックマーク) をクリックします。
4. [式エバリュエーター] ダイアログボックスで、式に一致する [フロータイプ] を選択します。
5. 「**HTTP** リクエストデータ」または「**HTTP** レスポンスデータ」フィールドに、式を使用して解析する HTTP リクエストまたはレスポンスを貼り付け、「評価」をクリックします。完全な HTTP リクエストまたはレスポンスを指定する必要があり、ヘッダーと本文は空白行で区切る必要があります。HTTP ヘッダーをトラップするプログラムの中には、レスポンスもトラップしないものがあります。ヘッダーのみをコピーして貼り付ける場合は、ヘッダーの最後に空白行を挿入して、完全な HTTP リクエストまたはレスポンスを形成します。
6. [閉じる] をクリックして、このダイアログボックスを閉じます。

名前付き高度なポリシー式の設定

August 15, 2023

複数のポリシーで同じ式を複数回入力する代わりに、名前付き式を設定し、ポリシーで式を使用する場合はいつでも名前を参照できます。たとえば、次の名前付き式を作成できます。

- この式:

  ```
  http.req.body(100).contains("this")
  ```

- その表現:

  ```
  http.req.body(100).contains("that")
  ```

その後、これらの名前付き式をポリシー式で使用できます。たとえば、次の例は、上記の例に基づく有効な表現です。

このエクスプレッション

高度なポリシー式の名前は、関数の接頭辞として使用できます。名前付きエクスプレッションは、単純エクスプレッションまたは複合エクスプレッションのいずれかになります。この関数は、名前付き式によって返されるデータの型を操作できる関数でなければなりません。

例 **1:** プレフィックスとしての単純な名前付き式

テキスト文字列を識別する次の単純な名前付き式は、<string\ > テキストデータを処理する AFTER_STR (「\」) 関数のプレフィックスとして使用できます。

```
HTTP.REQ.BODY(1000)
```


式の名前がtop1KBの場合、HTTP.REQ.BODY (1000) .AFTER_STR (「ユーザー名」) の代わりにTop1KB.AFTER_STR (「ユーザー名」) を使用できます。

例 2: プレフィックスとしての複合名前付き式

basic_header_value という名前の複合式を作成して、リクエスト内のユーザー名、コロン (:)、およびユーザーのパスワードを次のように連結できます。

```
add policy expression basic_header_value "HTTP.REQ.USER.NAME + \":\"
+ HTTP.REQ.USER.PASSWD"
```

次に、次の例に示すように、書き換えアクションで式の名前を使用できます。

```
add rewrite action insert_b64encoded_authorization insert_http_header
 authorization '"Basic " + basic_header_value.b64encode'
```

この例では、カスタムヘッダーの値を構築するために使用される式で、名前付き複合式によって返される文字列にB64 エンコーディングアルゴリズムが適用されます。

名前付き式 (単独で、または関数の接頭辞として) を使用して、書き換えで置換ターゲットのテキスト式を作成することもできます。

CLI を使用して名前付き詳細ポリシー式を設定します

コマンドプロンプトで次のコマンドを入力して、名前付き式を構成し、構成を確認します。

```
1 -  add policy expression \<name\>\<value\>
2
3 -  show policy expression \<name\>
4 <!--NeedCopy-->
```

例:

```
1 > add policy expression myExp "http.req.body(100).contains("the other")
     "
2 Done
3
4 > show policy expression myExp
5     1)      Name: myExp  Expr: "http.req.body(100).contains("the other"
          )"  Hits: 0 Type : ADVANCED
6 Done
7 <!--NeedCopy-->
```

式には最大 1,499 文字を使用できます。

GUI を使用して名前付き式を構成する

1. ナビゲーションウィンドウで、[**AppExpert**] を展開し、[式] をクリックします。
2. [高度な定義式] をクリックします。
3. [追加] をクリックします。
4. 式の名前と説明を入力します。
5. 高度なポリシー式の構成で説明されているプロセスを使用して、式を設定します。ステータスバーにメッセージが表示され、ポリシー式が正常に設定されたことが示されます。

ポリシーのコンテキスト外での高度なポリシー表現の設定

August 15, 2023

以下を含む多くの関数では、ポリシーの一部ではない高度なポリシー式が必要になる場合があります。

- 統合キャッシュセレクター:

  セレクターの定義では、複数の非複合式 (セレクト) を定義します。各セレクションレットは、他のセレクトレットと暗黙の論理論理和関係にあります。

- 負荷分散:

  負荷分散仮想サーバーの負荷分散の TOKEN メソッドの式を設定します。

- 書き換えアクション:

  式は、設定する書き換えアクションの種類に応じて、書き換えアクションの場所と実行する書き換えの種類を定義します。たとえば、DELETE アクションはターゲットエクスプレッションのみを使用します。REPLACE アクションは、ターゲット式と式を使用して置換テキストを設定します。

- レートベースのポリシー:

  高度なポリシー式を使用してリミットセレクターを構成します。これらのセレクターは、さまざまなサーバーへのトラフィック速度を制限するポリシーを設定するときに使用できます。セレクターの定義には、最大 5 つの非複合式 (セレクトレット) を定義します。各セレクションレットは、他のセレクションレットと暗黙の論理和になっています。

CLI を使用してポリシーの外部に高度なポリシー表現を設定します（キャッシュセレクタの例）

コマンドプロンプトで次のコマンドを入力して、ポリシーの外に高度なポリシー表現を設定し、構成を確認します。

```
1 -  add cache selector <selectorName> <rule>
2 -  show cache selector <selectorName>
3 <!--NeedCopy-->
```


例:

```
> add cache selector mainpageSelector "http.req.cookie.value("ABC_def")
   "
    "http.req.url.query.value("_ghi")"selector "mainpageSelector" added
Done
> show cache selector mainpageSelector
            Name: mainpageSelector
            Expressions:
                    1) http.req.cookie.value("ABC_def")
                    2) http.req.url.query.value("_ghi")
Done
<!--NeedCopy-->
```

次に、ポリシーでの高度なポリシー式の設定で説明されているように、より読みやすい q 区切り文字を使用する同等のコマンドを示します。

```
> add cache selector mainpageSelector2 q~http.req.cookie.value("ABC_def
   ")~
    q~http.req.url.query.value("_ghi")~selector "mainpageSelector2"
       added
Done
> show cache selector mainpageSelector2
            Name: mainpageSelector2
            Expressions:
                    1) http.req.cookie.value("ABC_def")
                    2) http.req.url.query.value("_ghi")
Done
<!--NeedCopy-->
```

高度なポリシー式: テキストの評価

August 15, 2023

要求または応答のテキストを評価する高度なポリシー式を使用してポリシーを設定できます。高度なポリシーテキスト表現には、HTTP ヘッダー内の文字列照合を実行する単純な式から、テキストをエンコードおよびデコードする複雑な式までさまざまです。テキスト表現は、大文字と小文字を区別するか、または区別しないか、スペースを使用するか無視するかを設定できます。また、テキスト式とブール演算子を組み合わせて複雑なテキスト式を構成することもできます。

式プレフィックスと演算子を使用して、HTTP 要求、HTTP 応答、VPN およびクライアントレス VPN データを評価できます。ただし、テキスト表現プレフィックスは、トラフィックのこれらの要素の評価に限定されません。

テキスト式について

August 15, 2023

NetScaler アプライアンスを経由するテキストを処理するためのさまざまな式を構成できます。次に、高度なポリシー式を使用してテキストを解析する方法の例をいくつか示します。

- 特定の HTTP ヘッダーが存在することを確認します。

  たとえば、リクエストを特定のサーバーに送信するために、特定の Accept-Language ヘッダーを含む HTTP リクエストを識別できます。

- 特定の HTTP URL に特定の文字列が含まれていることを判別します。

  たとえば、特定の URL に対するリクエストをブロックしたい場合があります。文字列は、別の文字列の先頭、中間、または末尾に出現する可能性があることに注意してください。

- 特定のアプリケーションに送信される POST 要求を識別します。

  たとえば、キャッシュされたアプリケーションデータを更新する目的で、データベースアプリケーションに送信されるすべての POST 要求を識別できます。

HTTP リクエストとレスポンスのデータストリームを表示するための専用ツールがあることに注意してください。これらのツールを使用して、データストリームを表示できます。

テキストに対する操作について

テキストベースの式は、データの要素を識別するための少なくとも 1 つの接頭辞と、通常はその接頭辞に対する操作 (必ずしもそうではありませんが) で構成されます。テキストベースの操作は、リクエストまたはレスポンスの任意の部分に適用できます。テキストに対する基本的な操作には、さまざまなタイプの文字列一致が含まれます。

たとえば、次の式はヘッダー値と文字列を比較します。

```
http.req.header("myHeader").contains("some-text")
```

次の式は、リクエストでファイルタイプを照合する例です。

```
http.req.url.suffix.contains("jpeg")
```

```
http.req.url.suffix.eq("jpeg")
```

前の例では、contains 演算子は部分一致を許可し、eq 演算子は完全一致を検索します。

評価する前に文字列を書式設定する他の操作も利用できます。たとえば、テキスト操作を使用して、引用符や空白を取り除いたり、文字列をすべて小文字に変換したり、文字列を連結したりできます。

> 注意：パターンに基づいてマッチングを実行したり、あるタイプのテキストフォーマットを別のタイプに変換したりするには、
>
> 複雑な操作を使用できます。

詳しくは、次のトピックを参照してください:

- パターンセットとデータセット。
- 正規表現。
- データを型キャストします。

テキスト式の複合と優先順位

さまざまな演算子を適用して、テキストの接頭辞または式を組み合わせることができます。たとえば、次の式は、各プレフィクスの戻り値を連結します。

```
http.req.hostname + http.req.url
```

次に、論理 AND を使用する複合テキスト式の例を示します。この式の両方のコンポーネントは、式と一致する要求に対して TRUE である必要があります。

```
http.req.method.eq(post)&& http.req.body(1024).startswith("destination
=")
```

> メモ:
>
> コンパウンドの演算子の詳細については、「複合高度なエクスプレッション」を参照してください。

テキスト式のカテゴリ

設定できるテキスト式の主なカテゴリは次のとおりです。

- HTTP ヘッダー、HTTP URL、および HTTP リクエストの POST ボディ内の情報。

  詳細については、「HTTP リクエストとレスポンスのテキストの式プレフィックス」を参照してください。

- VPN またはクライアントレス VPN に関する情報。

  詳細については、「VPN およびクライアントレス VPN の式プレフィックス」を参照してください。

- TCP ペイロード情報。

  TCP ペイロード式の詳細については、「高度なポリシー式:HTTP、TCP、および UDP データの解析」を参照してください。

- SSL (セキュアソケットレイヤー) 証明書のテキスト。

SSL および SSL 証明書データのテキスト式については、「高度なポリシー式:SSL 証明書の解析」および「[SSL 証明書の日付の式](/ja-jp/citrix-adc/13-1/appexpert/policies-and-expressions/adv-policy-exp-working-with-dates-times-and-numbers/exp-for-ssl-certificate-date.html)」を参照してください。

> 注:
>
> POST リクエストの本文などのドキュメント本体を解析すると、パフォーマンスに影響を与える可能性があります。ドキュメント本文を評価するポリシーのパフォーマンスへの影響をテストできます。

テキスト式のガイドライン

パフォーマンスの観点からは、通常、式でプロトコル対応関数を使用するのが最善です。たとえば、次の式はプロトコル対応関数を使用します。

```
HTTP.REQ.URL.QUERY
```

前の式は、文字列解析に基づく次の同等の式よりも優れたパフォーマンスを発揮します。

```
HTTP.REQ.URL.AFTER_STR("?")
```

最初のケースでは、式は具体的に URL クエリを参照します。2 番目のケースでは、式は疑問符が最初に出現するデータをスキャンします。

また、次の式のように、テキストの構造化解析によるパフォーマンス上の利点もあります。

```
HTTP.REQ.HEADER("Example").TYPECAST_LIST_T(',').GET(1)
```

(型キャストの詳細については、データの型キャストを参照してください。カンマ区切りのデータを収集してリストに構造化する型キャスト式は、通常、次の非構造化同等のものよりも優れています。

```
HTTP.REQ.HEADER("Example").AFTER_STR(",").BEFORE_STR(",")
```

最後に、非構造化テキスト式は通常、正規表現よりもパフォーマンスが優れています。たとえば、次に示すのは非構造化テキスト式です。

```
HTTP.REQ.HEADER("Example").AFTER_STR("more")
```

前の式は、通常、正規表現を使用する次の同等のものよりも優れたパフォーマンスを提供します。

```
HTTP.REQ.HEADER("Example").AFTER_REGEX(re/more/)
```

正規表現の詳細については、「正規表現」を参照してください。

HTTP リクエストとレスポンスのテキストの式プレフィックス

August 15, 2023

HTTP リクエストまたはレスポンスには、通常、ヘッダー、ヘッダー値、URL、POST 本文などのテキストが含まれます。HTTP リクエストまたはレスポンスで、これらのテキストベースの項目の 1 つ以上を操作するように式を設定できます。

パラメーターについて詳しくは、「NetScaler 高度なポリシー式リファレンス」を参照してください。

高度な式を使用して設定する方法の詳細については、次のトピックを参照してください。

- 複合高度なポリシー式
- 高度なポリシー式:IP アドレスと MAC アドレス、スループット、VLAN ID
- 高度なポリシー式:SSL の解析
- 高度なポリシー式: 日付、時刻、数字の操作
- 高度なポリシー表現の基本要素
- 高度なポリシー式: テキストの評価
- 高度なポリシー式:HTTP、TCP、および UDP データの解析
- デフォルトの構文式とポリシーの概要例

VPN とクライアントレス VPN のエクスプレッションプレフィックス

August 15, 2023

アドバンスドポリシーエンジンには、VPN またはクライアントレス VPN データの解析に固有のプレフィックスが用意されています。このデータには以下が含まれます。

- VPN トラフィックのホスト名、ドメイン、URL。
- VPN トラフィックのプロトコル。
- VPN トラフィックのクエリ。

これらのテキスト要素は、多くの場合 URL および URL の構成要素です。これらの要素にテキストベースの操作を適用するだけでなく、URL の解析に固有の操作を使用してこれらの要素を解析できます。詳細については、「URL のセグメントを抽出するための式」を参照してください。

VPN 式プレフィクスの詳細については、VPN 式表を参照してください。

テキストの基本操作

August 15, 2023

テキストに対する基本的な操作には、文字列の一致、文字列の長さの計算、大文字と小文字の区別を制御する操作が含まれます。式の引数として渡される文字列には空白を含めることができますが、文字列は 255 文字を超えることはできません。

文字列比較関数

次の表は、関数が TRUE または FALSE のブール値を返す基本的な文字列照合操作の一覧です。

機能	説明
<text>.CONTAINS(<string>)	ターゲットに<string>が含まれる場合、ブール型 TRUE 値を返します。例: http.req.url.contains(".jpeg")
<text>.EQ(<string>)	ターゲットが<string>と完全に一致する場合、ブール型 TRUE 値を返します。たとえば、次の式は、ホスト名が「myhostabc」の URL に対してブール値 TRUE を返します。http.req.url.hostname.eq("myhostabc")
<text>.STARTSWITH(<string>)	ターゲットが<string>で始まる場合は、ブール型 TRUE 値を返します。たとえば、次の式は、ホスト名が「myhostabc」の URL に対してブール値 TRUE を返します。http.req.url.hostname.startswith("myhost")
<text>.ENDSWITH(<string>)	ターゲットが\ で終わっている場合、ブール値 TRUE を返します <string\ >。たとえば、次の式は、ホスト名が「myhostabc」の URL に対してブール値 TRUE を返します。http.req.url.hostname.endswith("abc")
<text>.NE(<string>)	接頭辞が文字列引数と等しくない場合に、ブール型 TRUE 値を返します。接頭辞が文字列以外の値を返す場合、関数の引数は接頭辞によって返される値の文字列表現と比較されます。この関数は、ASCII 文字セットと UTF-8 文字セットの両方で、およびSET_TEXT_MODE(IGNORECASE)orSET_TEXT_MODE(NOIGNORECASE)で使用できます。
<text>.GT(<string>)	プレフィックスが文字列引数よりもアルファベット順で大きい場合は、ブール型 TRUE 値を返します。接頭辞が文字列以外の値を返す場合、関数の引数は接頭辞によって返される値の文字列表現と比較されます。関数は、SET_TEXT_MODE(IGNORECASE)またはSET_TEXT_MODE(NOIGNORECASE)、および ASCII 文字セットと UTF-8 文字セットの両方で使用できます。

機能	説明
<text>.GE(<string>)	プレフィックスがアルファベット順で文字列引数より大きいか等しい場合は、ブール型 TRUE 値を返します。接頭辞が文字列以外の値を返す場合、関数の引数は接頭辞によって返される値の文字列表現と比較されます。関数は、SET_TEXT_MODE(IGNORECASE) またはSET_TEXT_MODE(NOIGNORECASE)、および ASCII 文字セットと UTF-8 文字セットの両方で使用できます。
<text>.LT(<string>)	接頭辞が文字列引数よりアルファベット順に小さい場合は、ブール型 TRUE 値を返します。接頭辞が文字列以外の値を返す場合、関数の引数は接頭辞によって返される値の文字列表現と比較されます。関数は、SET_TEXT_MODE(IGNORECASE) またはSET_TEXT_MODE(NOIGNORECASE)、および ASCII 文字セットと UTF-8 文字セットの両方で使用できます。
<text>.LE(<string>)	プレフィックスがアルファベット順で文字列引数より小さいか等しい場合は、ブール型 TRUE 値を返します。接頭辞が文字列以外の値を返す場合、関数の引数は接頭辞によって返される値の文字列表現と比較されます。関数は、SET_TEXT_MODE(IGNORECASE) またはSET_TEXT_MODE(NOIGNORECASE)、および ASCII 文字セットと UTF-8 文字セットの両方で使用できます。

文字列の長さを計算する

この<text>.LENGTHオペレーションは、文字列の文字数 (バイト数ではない) に等しい数値を返します。

```
<text>.LENGTH
```

たとえば、特定の長さを超えるリクエスト URL を特定したい場合があります。次に、この例を実装する式を示します。

```
HTTP.REQ.URL.LENGTH < 500
```

文字列内の文字または要素の数を取った後、それらに数値演算を適用することができます。詳細については、「高度なポリシー式: 日付、時刻、および数値の操作」を参照してください。

テキストの大文字小文字の考慮、無視、変更

次の関数は、文字列内の文字の大文字小文字 (大文字または小文字) を操作します。

機能	説明
<text>.SET_TEXT_MODE(IGNORECASE| NOIGNORECASE)	この関数は、すべてのテキスト操作で大文字と小文字の区別をオンまたはオフにします。
|<text>.TO_LOWER| 最大 2 キロバイト (KB) のテキストブロックのターゲットを小文字に変換します。ターゲットが 2 KB を超える場合は UNDEF を返します。たとえば、文字列「abcd:」は「abcd:」に変換されます。
|<text>.TO_UPPER| ターゲットを大文字に変換します。ターゲットが 2 KB を超える場合は UNDEF を返します。たとえば、文字列「ABCD:」は「ABCD:」に変換されます。.

文字列から特定の文字を取り除く

STRIP_CHARS (<string>) 関数を使用すると、高度なポリシー式のプレフィックス (入力文字列) によって返されるテキストから特定の文字を削除できます。引数で指定した文字のすべてのインスタンスが、入力文字列から削除されます。結果の文字列には、文字列をパターンセットと照合するために使用されるメソッドなど、任意のテキストメソッドを使用できます。

たとえば、式CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_")では、プレフィックス CLIENT.UDP.DNS.DOMAIN によって返されるドメイン名から、ピリオド（.）、ハイフン（-）、およびアンダースコア（_）がSTRIP_CHARS(<string>)関数ですべて取り除かれます。返されるドメイン名が「a.dom_ai_n-name」の場合、この関数は文字列「adomainname」を返します。

次の例では、結果の文字列を「listofdomains」というパターンセットと比較します。

```
CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_").CONTAINS_ANY("listofdomains")
```

注:STRIP_CHARS(<string>) 関数によって返される文字列の書き換えは実行できません。

次の関数は、指定された文字列入力の先頭と末尾から一致する文字を取り除きます。

機能	説明
<text>.STRIP_START_CHARS(s)	入力文字列の先頭から一致しない最初の文字が見つかるまで、一致する文字を取り除き、文字列の残りの部分を返します。削除したい文字は、引用符で囲んだ 1 つの文字列として指定する必要があります。たとえば、ヘッダーの名前が testLang で、*:/en_us:* がその値である場合、*HTTP.RES.HEADER(“TestLang”).STRIP_START_CHARS(“:”)* は、ヘッダーの値の先頭から最初の不一致文字 e が見つかるまで、指定された文字を取り除き、次の文字として返します。
<text>.STRIP_END_CHARS(s)	入力文字列の末尾から最初に見つかった一致しない文字までの一致する文字を取り除き、文字列の残りの部分を返します。削除したい文字は、引用符で囲んだ 1 つの文字列として指定する必要があります。たとえば、ヘッダーの名前が testLang で、*:/en_us:* がその値である場合、*HTTP.RES.HEADER(“TestLang”).STRIP_START_CHARS(“:”)* は、ヘッダーの値の最後から最初の不一致文字 s が見つかるまで、指定された文字を取り除き、:/_en_us を文字列として返します。

文字列を別の文字列に追加する

APPEND () 関数を使用すると、引数の文字列表現を、前の関数によって返される値の文字列表現に追加できます。上記の関数は、数値、符号なし長整数型、倍精度浮動小数点型、時刻値、IPv4 アドレス、または IPv6 アドレスを返す関数です。引数には、テキスト文字列、数値、符号なし長整数型、倍精度浮動小数点型、時刻値、IPv4 アドレス、または IPv6 アドレスを使用できます。結果の文字列値は、+ 演算子を使用して取得した文字列値と同じです。

テキストに対する複雑な操作

August 15, 2023

単純な文字列照合に加えて、特定の文字列ではなく、文字列の長さとテキストブロックのパターンを調べる式を設定できます。

テキストベースの操作では、次の点に注意してください。

- 文字列引数をとる操作では、文字列は 255 文字を超えることはできません。
- 式に文字列を指定するときは、空白を含めることができます。

文字列の長さに対する操作

次の操作では、文字列を文字カウントで抽出します。

文字カウント操作	説明
<text>.TRUNCATE(<count>)	ターゲットの末尾を<count>の文字数で切り捨てた後の文字列を返します。文字列全体が<count>より短い場合、何も返されません。
<text>.TRUNCATE(<character>, <count>)	<count>で指定された文字数だけ<character>の後のテキストを切り捨てた後の文字列を返します。
<text>.PREFIX(<character>,<count>)	<character>の出現回数が最も多い<count>ターゲット内で最も長いプレフィクスを選択します。
<text>.SUFFIX(<character>,<count>)	<character>の出現回数が最も多い<count>ターゲット内で最も長いサフィックスを選択します。たとえば、次のレスポンス本文を考えてみましょう peninsula。次の式は、sula: http.res.body(100).suffix('n',0)の値を返します。次の式が戻ります insula: http.res.body(100).suffix('n',1)。次の式は、peninsula: http.res.body(100).suffix('n',2)の値を返します。次の式は、peninsula: http.res.body(100).suffix('n',3)の値を返します。
<text>.SUBSTR(<starting_offset>, <length>)	ターゲットオブジェクトから<length>文字数を含む文字列を選択します。<starting_offset>の後の文字列の抽出を開始します。オフセットの後の文字数が<length>引数の値より少ない場合は、残りの文字をすべて選択します。
<text>.SKIP(<character>, <count>)	<character>の出現回数が最も多い<count>最長のプレフィックスをスキップした後に、ターゲットから文字列を選択します。

文字列の一部に対する操作

いずれかの操作を使用して大きな文字列のサブセットを抽出する方法については、文字列演算表を参照してください。

2 つの文字列の英数字の順序を比較する操作

COMPARE 操作は、2 つの異なる文字列の最初の不一致文字を検査します。この操作は、辞書の用語を並べ替えるときに使用される方法である辞書順に基づいています。

この演算は、比較された文字列の最初の一致しない文字の ASCII 値間の算術差を返します。次の相違点は例です。

- 「abc」と「and」の差は-1 です（3 番目のペアワイズ文字比較に基づく）。
- 「@」と「abc」の差は-33 です。
- 「1」と「abc」の差は-47 です。

COMPARE 操作の構文を以下に示します。

```
<text>.COMPARE(<string>)
```

テキストを表すバイト文字列から整数を抽出する

テキストを表すバイト文字列をバイト列として扱い、シーケンスから 8 ビット、16 ビット、または 32 ビットを抽出し、抽出したビットを整数に変換する方法については、整数抽出表を参照してください 。

テキストをハッシュ値に変換する

HASH 関数を使用して、テキスト文字列をハッシュ値に変換できます。この関数は、演算の結果として 31 ビットの正の整数を返します。式の形式は次のとおりです。

```
<text>.HASH
```

この関数は、大文字と小文字と空白を無視します。たとえば、演算後、2 つの文字列 Ab c と a bc は同じハッシュ値を生成します。

Base64 エンコーディングアルゴリズムを適用してテキストをエンコードおよびデコードする

次の 2 つの関数は、Base64 エンコーディングアルゴリズムを適用してテキスト文字列をエンコードおよびデコードします。

機能	説明
text.B64ENCODE	Base64 エンコーディングアルゴリズムを適用して、テキスト文字列 (text で指定) をエンコードします。
text.B64DECODE	Base64 デコードアルゴリズムを適用して、Base64 でエンコードされた文字列 (テキストで指定) をデコードします。テキストが B64 エンコード形式でない場合、この操作は UNDEF を生成します。

EXTEND 関数を使用して、書き換えアクションの検索を絞り込みます

EXTEND 関数は、パターンまたはパターンセットを指定し、HTTP パケットの本体をターゲットとする書き換えアクションで使用されます。パターン一致が見つかると、EXTEND 関数は、一致する文字列の両側の定義済みのバイト数だけ検索範囲を拡張します。その後、正規表現を使用して、この拡張領域の一致に対して書き換えを実行できます。EXTEND 関数で設定された書き換えアクションは、正規表現のみを使用して HTTP 本文全体を評価する書き換えアクションよりも速く書き換えを実行します。

EXTEND 関数の形式は EXTEND (m, n) です。ここで、m と n は、検索の範囲が一致パターンの前後に拡張されるバイト数です。一致が見つかると、新しい検索範囲は、一致する文字列の直前の m バイト、文字列自体、および文字列の後の n バイトで構成されます。正規表現を使用して、この新しい文字列の一部を書き換えることができます。

EXTEND 関数は、関数を使用する書き換えアクションが次の要件を満たしている場合にのみ使用できます。

- 検索は、パターンまたはパターンセット (正規表現ではない) を使用して実行されます。
- 書き換えアクションは、HTTP パケットの本体のみを評価します。

また、EXTEND 関数は、次のタイプの書き換えアクションでのみ使用できます。

- replace_all
- insert_after_all
- delete_all
- insert_before_all

たとえば、本文の最初の 1000 バイトの”http://exampleurl.com/”と”http://exampleurl.au/”のすべてのインスタンスを削除できます。これを行うには、文字列 exampleurl のすべてのインスタンスを検索し、一致が見つかったときに文字列の両側で検索範囲を拡張し、正規表現を使用して拡張領域で書き換えを実行する書き換えアクションを設定できます。次の例では、検索の範囲を、一致する文字列の左に 20 バイト、右に 50 バイト拡張します。

```
add rewrite action delurl_example delete_all 'HTTP.REQ.BODY(1000)'-
search exampleurl -refineSearch 'extend(20,50).regex_select(re#http
://exampleurl.(com|au)#)'
```

テキストを **16** 進数形式に変換する

次の関数は、テキストを 16 進数形式に変換し、結果の文字列を抽出します。

```
<text>.BLOB_TO_HEX(<string>)
```

たとえば、この関数はバイト文字列「abc」を「61:62:63」に変換します。

テキストの暗号化と復号化

高度なポリシー式では、ENCRYPT 関数と DECRYPT 関数を使用してテキストを暗号化および復号化できます。特定の NetScaler アプライアンスまたは高可用性（HA）ペアで ENCRYPT 機能によって暗号化されたデータは、同

じ NetScaler アプライアンスまたは HA ペア上の DECRYPT 機能による復号化を目的としています。アプライアンスは、RC4、DES3、AES128、AES192、および AES256 暗号化方式をサポートしています。暗号化に必要なキー値は、ユーザーが指定できません。暗号化方式が設定されると、アプライアンスは指定された方式に適したランダムなキー値を自動的に生成します。デフォルトの方法は AES256 暗号化です。これは最も安全な暗号化方法であり、Citrix が推奨する暗号化方法です。

暗号化方式を変更する場合、またはアプライアンスが現在の暗号化方式に対して新しいキー値を生成する場合を除き、暗号化を設定する必要はありません。

注**:** XML ペイロードを暗号化および復号化することもできます。XML ペイロードを暗号化および復号化する関数については、XML ペイロードの暗号化と復号化を参照してください。

暗号化の構成

起動時に、アプライアンスは set ns encryptionParams コマンドをデフォルトで AES256 暗号化方式で実行し、AES256 暗号化に適したランダムに生成されたキー値を使用します。また、アプライアンスはキー値を暗号化し、暗号化されたキー値とともにコマンドを NetScaler 構成ファイルに保存します。したがって、AES256 暗号化方式は、ENCRYPT 関数と DECRYPT 関数に対してデフォルトで有効になっています。設定ファイルに保存されているキー値は、アプライアンスが再起動するたびにコマンドを実行しても、再起動後も保持されます。

暗号化方式を変更する場合、またはアプライアンスで現在の暗号化方式に対して新しいキー値を生成する場合は、set ns encryptionParams コマンドを手動で実行するか、構成ユーティリティを使用できます。CLI を使用して暗号化方式を変更するには、「例 **1:** 暗号化方式の変更」に示すように、**method** パラメータのみを設定します。アプライアンスで現在の暗号化方式に対して新しいキー値を生成する場合は、「例 **2**: 現在の暗号化方式に対する新しいキー値の生成」に示すように、**method** パラメータを現在の暗号化方式に設定し、**KeyValue** パラメータを空の文字列（”“）に設定します。新しいキー値を生成したら、設定を保存する必要があります。設定を保存しない場合、アプライアンスは次に再起動するまで新しく生成されたキー値のみを使用し、その後、保存された設定のキー値に戻ります。

GUI を使用した暗号化の設定

1. [システム] > [設定**]** に移動します。
2. [設定] 領域で、[暗号化パラメータの変更] をクリックします。
3. [暗号化パラメータの変更] ダイアログボックスで、次のいずれかの操作を行います。
    - 暗号化方式を変更するには、[方式] ボックスの一覧で、使用する暗号化方式を選択します。
    - 現在の暗号化方式の新しいキー値を生成するには、[選択した方法の新しいキーを生成] をクリックします。
4. [**OK**] をクリックします。

暗号化関数と復号化関数を使用する

ENCRYPT 関数と DECRYPT 関数は、テキストを返す任意の式プレフィックスとともに使用できます。たとえば、クッキー暗号化のリライトポリシーで ENCRYPT 関数と DECRYPT 関数を使用できます。次の例では、書き換えアクションは、バックエンドサービスによって設定される myCookie という名前の Cookie を暗号化し、クライアントから返されたときに同じ Cookie を復号化します。

```
add rewrite action my-cookie-encrypt-action replace "HTTP.RES.
    SET_COOKIE.COOKIE("MyCookie").VALUE(0)" "HTTP.RES.SET_COOKIE.COOKIE(
    "MyCookie").VALUE(0).ENCRYPT"

add rewrite action my-cookie-decrypt-action replace "HTTP.REQ.COOKIE.
    VALUE("MyCookie")" "HTTP.REQ.COOKIE.VALUE("MyCookie").DECRYPT"
<!--NeedCopy-->
```

暗号化と復号化のポリシーを設定したら、設定を保存してポリシーを有効にします。

サードパーティ暗号化用の暗号化キーを構成する

高度なポリシー式では、ENCRYPT 関数と DECRYPT 関数を使用して、リクエストまたはレスポンスのテキストを暗号化および復号化できます。アプライアンス（スタンドアロン、高可用性、またはクラスタ）の ENCRYPT 機能によって暗号化されたデータは、同じアプライアンスによる DECRYPT 機能によって復号化されることを意図しています。アプライアンスは、RC4、DES、Triple-DES、AES92、AES256 暗号化方式をサポートしており、これらの各方式はデータの暗号化と復号化の両方に秘密鍵を使用します。これらの方法のいずれかを使用して、自己暗号化とサードパーティ暗号化の 2 つの方法でデータを暗号化および復号化できます。

アプライアンス（スタンドアロン、高可用性、またはクラスタ）の自己暗号化機能は、ヘッダー値を評価してデータを暗号化してから復号化します。これを理解する一例として、HTTP クッキーの暗号化があります。この式は、ヘッダーを評価し、送信応答の Set-Cookie ヘッダー内の HTTP Cookie 値を暗号化し、クライアントからの後続の着信要求の Cookie ヘッダーで返されたときに Cookie 値を復号化します。キー値はユーザー設定できません。代わりに、set ns encryptionParams コマンドで暗号化方式が設定されている場合、アプライアンスは設定された方式のランダムなキー値を自動的に生成します。デフォルトでは、このコマンドは AES256 暗号化方式を使用します。これは高度にセキュリティ保護された方法であり、この方法をお勧めします。

サードパーティの暗号化機能は、サードパーティアプリケーションを使用してデータを暗号化または復号化します。たとえば、クライアントはリクエスト内のデータを暗号化し、アプライアンスはバックエンドサーバーに送信する前にデータを復号化したり、その逆を行ったりします。これを実行するには、アプライアンスとサードパーティアプリケーションが秘密キーを共有する必要があります。アプライアンスでは、暗号化キーオブジェクトを使用して秘密キーを直接設定できます。キー値は、より強力な暗号化のためにアプライアンスによって自動的に生成されます。アプライアンスとサードパーティアプリケーションの両方がデータの暗号化と復号化に同じキーを使用できるように、サードパーティ製アプライアンスで同じキーを手動で設定します。

注: サードパーティの暗号化を使用して、XML ペイロードを暗号化および復号化することもできます。XML ペイロードを暗号化および復号化する関数については、「XML ペイロードの暗号化と復号化」を参照してください。

暗号メソッド

暗号方式は、平文のバイト列を暗号文バイト列に変換する暗号化機能と、暗号文を平文に戻す復号機能の 2 つの機能を提供する。暗号方式は、鍵と呼ばれるバイトシーケンスを使用して、暗号化と復号化を実行します。暗号化と復号に同じ鍵を使用する暗号方式は、対称と呼ばれます。暗号化と復号化に異なるキーを使用する暗号方式は非対称です。非対称暗号の最も注目すべき例は公開鍵暗号法である。公開鍵暗号では、暗号化のために誰でも利用できる公開鍵と、復号機のみが知っている秘密鍵を使用する。

優れた暗号方式は、鍵を持っていない場合、暗号文を解読（「クラック」）することが不可能になります。「実行不可能」とは、サイファーテキストをクラックすると、価値よりも多くの時間とコンピューティングリソースがかかることを意味します。コンピュータがより強力で安価になるにつれて、以前はクラックが不可能であった暗号がより実現可能になります。また、時間が経つにつれて、暗号メソッド（またはその実装）に欠陥が見つかり、クラッキングが容易になります。したがって、古い暗号方式よりも新しい暗号方式が優先されます。一般に、長い長さのキーは、短いキーよりもセキュリティが優れていますが、暗号化と復号化の時間が長くなります。

暗号方式は、ストリーム暗号またはブロック暗号を使用できます。RC4 は、ほとんどセキュリティで保護されたストリーム暗号であり、レガシーアプリケーションにのみ使用されます。ブロック暗号にはパディングを含めることができます。

ストリーム暗号

ストリーム暗号方式は、個々のバイトに対して動作します。NetScaler- アプライアンスで使用できるストリーム暗号は、128 ビット（16 バイト）のキー長を使用する RC4 だけです。与えられたキーに対して、RC4 はバイトの擬似乱数シーケンスを生成し、キーストリームを呼び出す。キーストリームは、暗号文を生成するために平文と X 論理変換される。RC4 はもはや安全とは見なされず、レガシーアプリケーションで必要な場合にのみ使用すべきです。

ブロック暗号

ブロック暗号方式は、固定されたバイトブロックで動作します。NetScaler アプライアンスには、データ暗号化標準（DES）と高度暗号化標準（AES）の 2 つのブロック暗号が用意されています。DES は 8 バイトのブロックサイズを使用し、（NetScaler アプライアンスでは）キーの長さに 2 つの選択肢があります。64 ビット（8 バイト）（56 ビットがデータ、8 ビットがパリティです）、トリプル DES は 192 ビット（24 バイト）のキー長です。AES のブロックサイズは 16 バイトで、（NetScaler では）キーの長さは 128 ビット（16 バイト）、192 ビット（24 バイト）、256 ビット（32 バイト）の 3 つから選択できます。

パディング

ブロック暗号のプレーンテキストがブロックの整数でない場合は、より多くのバイトのパディングが必要になることがあります。例えば、平文が「xyzzy」（16 進数 78797a7a79）であるとする。8 バイトの Triple-DES ブロックの場合、8 バイトを作成するには、この値をパディングする必要があります。パディングスキームは、復号化関数が復

号後の元の平文の長さを決定できるようにする必要があります。現在使用されているパディングスキームをいくつか次に示します (n は追加されたバイト数です)。

- PKCS7: n バイトの値を n 個ずつ加算します。たとえば、78797a7a79030303。これは、OpenSSL と ENCRYPT () ポリシー関数で使用されるパディングスキームです。PKCS5 のパディング方式は PKCS7 と同じです。
- ANSI X.923: n-1 個のゼロバイトと値 n の最後のバイトを加算します。たとえば、78797a7a79000003。
- ISO 10126: n-1 個のランダムなバイトと値 n の最後のバイトを加算する。たとえば、78797a7a79xxxx03 と指定します。xx には任意のバイト値を指定できます。DECRYPT () ポリシー関数はこのパディングスキームを受け入れ、PKCS7 および ANSI X.923 スキームを受け入れることもできます。
- ISO/IEC 7816-4:0x80 バイトと n-1 ゼロバイトを追加します。たとえば、78797a7a79800000。これは OneAndZeroS パディングとも呼ばれます。
- ゼロ:n 個のゼロバイトを追加します。例：78797a7a79000000。これは NUL バイトを含まないプレーンテキストでのみ使用できます。

パディングが使用され、平文がブロックの整数である場合、復号関数が元の平文の長さを明確に決定できるように、通常、余分なブロックが追加されます。PCKS7 と 8 バイトブロックの場合、これは 0808080808080808 になります。

動作モード

ブロック暗号にはさまざまな動作モードがあり、平文の複数のブロックをどのように暗号化するかを指定します。一部のモードは、暗号化プロセスを開始するために使用される平文とは別のデータのブロックである初期化ベクトル (IV) を使用します。暗号化ごとに異なる IV を使用することをお勧めします。そうすれば、同じ平文でも異なる暗号文が生成されます。IV は秘密である必要はないので、暗号文の前に付加されます。モードには次のものが含まれます。

- 電子コードブック (ECB): 平文の各ブロックは個別に暗号化されます。IV は使用されません。プレーンテキストが暗号ブロックサイズの倍数でない場合は、パディングが必要です。同じ平文と鍵は、常に同じ暗号文を生成します。このため、ECB は他のモードよりも安全性が低いと考えられ、レガシーアプリケーションにのみ使用する必要があります。
- 暗号ブロック連鎖（CBC）: 平文の各ブロックは、暗号化される前に、前の暗号文ブロック、または最初のブロックの IV と XOR 化されます。プレーンテキストが暗号ブロックサイズの倍数でない場合は、パディングが必要です。これは NetScaler の暗号化パラメータ方式で使用されるモードです。
- Cipher Feedback (CFB): 前の暗号文ブロック、または最初のブロックの IV は暗号化され、出力は現在の平文ブロックと XOR され、現在の暗号文ブロックが作成されます。フィードバックは、1 ビット、8 ビット、または 128 ビットです。平文は暗号文と XOR されるので、パディングは不要である。
- 出力フィードバック (OFB): キーストリームは、暗号を連続的に IV に適用し、キーストリームブロックを平文で XOR することによって生成されます。パディングは必要ありません。

サードパーティ暗号化用の暗号化キーを構成する　次に、暗号化キーの設定で実行される設定タスクを示します。

1. 暗号化キーを追加する。指定された暗号方式の暗号キーを、指定されたキー値で設定します。
2. 暗号化キーの変更。設定済みの暗号化キーのパラメータを編集できます。
3. 暗号化キーの設定解除。設定済みの暗号化キーのパラメータをデフォルト値に設定します。名前を持つ EncryptionKey 値が存在する必要があります。パディングを DEFAULT (メソッドで決定) に設定し、既存の IV を削除します。これにより、ENCRYPT () はランダムな IV を生成します。既存のコメントを削除します。メソッドとキー値はリセットできません。
4. 暗号化キーを削除する。設定されている暗号化キーを削除します。キーには参照を設定できません。
5. 暗号化キーを表示します。設定された暗号化キーまたはすべての設定済みキーのパラメータを表示します。名前を省略すると、キー値は表示されません。

CLI を使用して暗号化キーを追加する　コマンドプロンプトで入力します。

```
add ns encryptionKey <name> -method <method> [-keyValue <keyvalue>][-
padding (OFF | ON)] [-iv <hexstring>] -keyValue <keyvalue> [-comment
<string>]
```

各項目の意味は次のとおりです。

```
1 <method> = ( NONE | RC4 | DES3 | AES128 | AES192 | AES256 | DES | DES-
     CBC | DES-CFB | DES-OFB | DES-ECB | DES3-CBC | DES3-CFB | DES3-OFB |
      DES3-ECB | AES128-CBC | AES128-CFB | AES128-OFB | AES128-ECB |
     AES192-CBC | AES192-CFB | AES192-OFB | AES192-ECB | AES256-CBC |
     AES256-CFB | AES256-OFB | AES256-ECB ) <hexstring> = hex-encoded
     byte sequence
2 <!--NeedCopy-->
```

上記の暗号化方式は、CBC をデフォルトの動作モードとする動作モードを指定します。したがって、DES、DES2、AES128、AES192、AES256 方式は、DES-CBC、DES3-CBC、AES128-CBC、AES192-CBC、AES256-CBC 方式と同等である。

CLI を使用して暗号化キーを変更する　コマンドプロンプトで入力します。

```
set ns encryptionKey <name> [-method <method>] [-keyValue <keyvalue>]
 [-padding ( OFF | ON )] [-iv <string>] [-comment <string>]
```

CLI を使用して暗号化キーを設定解除する　コマンドプロンプトで入力します。

```
unset ns encryptionKey <name> [-padding] [-iv] [-comment]
```

CLI を使用して暗号化キーを削除する　コマンドプロンプトで入力します。

```
rm ns encryptionKey <name>
```


CLI を使用して暗号化キーを表示する　コマンドプロンプトで入力します。

例**:**

```
show ns encryptionKey [<name>]

add ns encryptionKey my_key -method aes256 -keyValue 26
    ea5537b7e0746089476e5658f9327c0b10c3b4778c673a5b38cee182874711  - iv
    c2bf0b2e15c15004d6b14bcdc7e5e365
set ns encryptionKey my_key -keyValue
    b8742b163abcf62d639837bbee3cef9fb5842d82d00dfe6548831d2bd1d93476
unset ns encryptionKey my_key -iv
rm ns encryptionKey my_key
show ns encryptionKey my_key
Name: my_key
Method: AES256
Padding: DEFAULT
Key Value: (not disclosed)
<!--NeedCopy-->
```

GUI を使用して暗号化キーを追加する　[システム] > [暗号化キー] に移動し、[追加] をクリックして暗号化キーを作成します。

GUI を使用して暗号化キーを変更する　[システム] > [暗号化キー] に移動し、[編集] をクリックして、設定された暗号化キーのパラメータを変更します。

GUI を使用して暗号化キーを削除する　[システム] > [暗号化キー] に移動し、[削除] をクリックします。

サードパーティ暗号化用の暗号化および復号化機能

以下は、サードパート暗号化に使用される ENCRYPT 関数です。

```
ENCRYPT (encryptionKey, out_encoding)
```

各項目の意味は次のとおりです。

アプライアンスの入力データは、暗号化されるテキストです。

encryptionKey: 暗号化方法、秘密キーの値、およびその他の暗号化パラメータを提供するために、設定された暗号化キーオブジェクトを指定するオプションの文字列パラメータ。省略した場合、このメソッドは set ns encryptionParams コマンドに関連付けられた自動生成されたキー値を使用します。

out_encoding: この値は、出力のエンコード方法を指定します。省略すると、BASE64 エンコーディングが使用されます。

入力**:**


```
BASE64: original PEM base64-encoding: 6 bits (0..63) encoded as one
    ASCII character:
                  0..23 = 'A'..'Z', 24..51 = 'a'..'z', 52..61 = '0'..'9
                      ', 62 = '+', 63 = '/', '=' = pad byte.
       BASE64URL:  URL and Filename safe base64-encoding: same as BASE64
            except 62 = '-', 63 = '_'
       HEX_UPPER:  Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'A'..'F
           '.
       HEX_LOWER:  Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'a'..'f
           '.
       HEX_COLONS: Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'A'..'F
           '; ':' between each hex byte. Matches BLOB_TO_HEX() output
           format
       HEX:  For input, accepts HEX_UPPER, HEX_LOWER, and HEX_COLONS
           format. For output, produces HEX_LOWER format
<!--NeedCopy-->
```

出力: 出力は、指定されたメソッドとキーを使用して暗号化され、指定された出力エンコーディングを使用してエンコードされたテキストです。IV を必要とするブロックメソッドおよびモードの暗号化されたテキストの前に生成された IV を挿入し、encryptionKey に IV が指定されていないか、または encryptionKey が省略されます。

以下は、サードパーティの復号化に使用される DECRYPT 関数です。

```
DECRYPT(encryptionKey, in_encoding)
```

各項目の意味は次のとおりです。

入力データは、指定されたメソッドを使用した暗号化されたテキストであり、指定された入力エンコーディングを使用してエンコードされたキーです。このテキストは、IV を必要とするブロックメソッドおよびモードの暗号化されたテキストの前に生成された IV を含むことが想定され、encryptionKey に IV が指定されていないか、または encryptionKey が省略されます。

encryptionKey: 暗号化方式、秘密鍵、およびその他の暗号化パラメータを提供するために、設定された encryptionKey オブジェクトを指定するオプションの文字列パラメータ。省略すると、encryptionParams 設定に関連付けられたメソッドと自動生成されたキーが使用されます。

in_encoding-入力のエンコード方法を指定するオプションの列挙パラメータ。これらの値は、ENCRYPT の out_encoding と同じです。省略すると、BASE64 エンコーディングが想定されます。

出力データは、エンコードされていない復号化されたテキストです。

バリアントとオプションのパラメータ　以下に、これらの関数のバリアントとオプションのパラメータを示します。

バリアント	説明
ENCRYPT	encryptionParams コマンドと BASE64 出力エンコーディングパラメータを使用します。

バリアント	説明
ENCRYPT (out_encoding)	EncryptionParams を使用し、出力エンコーディングパラメータを指定します。
ENCRYPT(encryptionKey)	指定した encryptionKey および BASE64 出力エンコーディングパラメータを使用します。
ENCRYPT(encryptionKey, out_encoding)	指定した encryptionKey と出力エンコーディングパラメーターを使用します。
DECRYPT	encryptionParams コマンドと BASE64 入力エンコーディングパラメータを使用します。
DECRYPT (out_encoding)	encryptionParams コマンドと指定された入力エンコーディングパラメータを使用します。
DECRYPT(encryptionKey)	指定した encryptionKey および BASE64 入力エンコーディングパラメータを使用します。
DECRYPT(encryptionKey, out_encoding)	指定した encryptionKey と入力エンコーディングパラメータを使用します。

HMAC キーの設定

NetScaler アプライアンスは、メッセージ送信者とメッセージ受信者の間で共有される秘密鍵を使用して入力テキストのダイジェストメソッドまたはハッシュを計算するハッシュメッセージ認証コード（HMAC）機能をサポートしています。ダイジェスト方式 (RFC 2104 手法から派生したもの) は、送信者を認証し、メッセージの内容が変更されていないことを確認します。たとえば、クライアントが共有 HMAC キーを含むメッセージを NetScaler アプライアンスに送信すると、高度な（PI）ポリシー表現は HMAC 関数を使用して選択したテキストのハッシュベースのコードを計算します。次に、受信者が秘密鍵を含むメッセージを受信すると、HMAC を元の HMAC と比較して再計算し、メッセージが改ざんされているかどうかを判断します。HMAC 機能は、スタンドアロンアプライアンス、および高可用性構成またはクラスタ内のアプライアンスによってサポートされます。これを使用することは、暗号化キーの設定に似ています。

add ns hmackey コマンドと set ns hmackey コマンドには、HMAC 計算に使用するダイジェスト方式と共有秘密鍵を指定するパラメータが含まれています。

HMAC キーを設定するには、次の手順を実行する必要があります。

1. HMAC キーの追加。指定されたキー値で HMAC キーを設定します。
2. HMAC キーの変更。設定された HMAC キーのパラメータを変更します。ダイジェストメソッドは、キー値の長さがダイジェストによって決定されないため、キー値を変更することなく変更できます。ただし、ダイジェストを変更するときは、新しいキーを指定することをお勧めします。
3. HMAC キーの設定を解除します。設定された HMAC キーのパラメータをデフォルト値に設定します。という名前の HMAckEy オブジェクトが存在する必要があります。設定解除できるパラメータはコメントのみです。コメントは削除されます。

4. HMAC キーを削除する。設定済みのキーを削除します。キーには参照を設定できません。
5. HMAC キーを表示します。設定された HMAC AC キーまたはすべての設定済みキーのパラメータを表示します。名前を省略すると、キー値は表示されません。

一意でランダムな **HMAC** キーを構成する

一意の HMAC キーを自動的に生成できます。アプライアンスがクラスタ構成の場合、HMAC キーはプロセスの開始時に生成され、すべてのノードとパケットエンジンに配布されます。これにより、HMAC キーは、クラスタ内のすべてのパケットエンジンとすべてのノードで同じになります。

コマンドプロンプトで入力します。

```
add ns hmacKey <your_key> -digest <digest> -keyValue <keyvalue>
```

例**:**

```
add ns hmacKey <name> -digest sha1 -keyValue AUTO
```

各項目の意味は次のとおりです。

- 名前の構文は正しく、既存のキーの名前と重複しません。
- 「AUTO」KeyValue を設定コマンドで使用すると、既存の encrytionKey オブジェクトと HMAckEy オブジェクトの新しいキーを生成できます。

> 注:
>
> 自動キー生成は、NetScaler アプライアンスがキーを使用してデータを暗号化および復号化する場合、または HMAC キーを生成して検証する場合に便利です。キー値自体は表示時にすでに暗号化されているため、生成されたキー値を取得して他の当事者が使用することはできません。

例**:**

```
add ns hmacKey my_hmac_key -digest sha1 -keyValue 0c753c6c5ef859189cacdf95b506
```

上記の暗号化方式は、CBC をデフォルトの動作モードとする動作モードを指定します。したがって、DES、DES2、AES128、AES192、AES256 方式は、DES-CBC、DES3-CBC、AES128-CBC、AES192-CBC、AES256-CBC 方式と同等である。

CLI を使用した **HMAC** キーの変更

このコマンドは、HMAC キーに設定されたパラメータを変更します。キー値の長さはダイジェストによって決定されないため、キー値を変更せずにダイジェストを変更できます。ただし、ダイジェストを変更するときは、新しいキーを指定することをお勧めします。コマンドプロンプトで入力します。


```
set ns hmacKey <name> [-digst <digest>] [-keyValue <keyvalue>]
[-comment <string>]

<!--NeedCopy-->
```

CLI を使用した HMAC キーの設定解除

このコマンドは、HMAC キー用に設定されたパラメータをデフォルト値に設定します。という名前の HMAckEy オブジェクトが存在する必要があります。設定解除できる唯一のパラメータは comment オプションで、これは削除されます。コマンドプロンプトで入力します。

```
unset ns hmacKey <name> -comment
```

CLI を使用して HMAC キーを削除する

このコマンドは、設定された hmac キーを削除します。キーは参照を持つことができません。コマンドプロンプトで入力します。

```
rm ns hmacKey <name>
```

CLI を使用して HMAC キーを表示する

コマンドプロンプトで入力します。

```
show ns encryptionKey [<name>]

add ns hmacKey my_hmac_key -digest sha1 -keyValue 0
    c753c6c5ef859189cacdf95b506d02c1797407d
set ns hmacKey my_hmac_key -keyValue
    f348c594341a840a1f641a1cf24aa24c15eb1317
rm ns hmacKey my_hmac_key
show ns hmacKey my_hmac_key
          Name: my_hmac_key
     Digest: SHA1
     Key Value: (not disclosed)
<!--NeedCopy-->
```

高度なポリシー式: 日付、時刻、数字の操作

August 15, 2023

NetScaler アプライアンスが処理するほとんどの数値データは、日付と時刻で構成されています。アプライアンスは、日付と時刻の処理に加えて、HTTP 要求と応答の長さなどの他の数値データを処理します。このデータを処理するために、数値を処理する高度なポリシー表現を設定できます。

数値式は、数値を返す式のプレフィックスと、数値に対して演算を実行できる演算子で構成されますが、常にそうであるとは限りません。数値を返す式のプレフィックスの例としては SYS.TIME.DAY、、、HTTP.REQ.CONTENT_LENGTH、HTTP.RES.BODY.LENGTH. Numeric および演算子は、データを数値形式で返す任意のプレフィックス式で使用できます。たとえば、GT(<**int**>) この演算子は HTTP.REQ.CONTENT_LENGTH など、整数を返す任意のプレフィックス式で使用できます。

式内の日付と時刻の形式

August 15, 2023

日付と時刻（NetScaler のシステム時刻や SSL 証明書内の日付など）を扱うポリシーで高度なポリシー表現を構成する場合は、次のように時間形式を指定します。

```
GMT|LOCAL [<yyyy>] [<month>] [<d>] [<h>] [<m>] [<s>]
```

各項目の意味は次のとおりです：

- <yyyy> は GMT またはローカル時刻の 4 桁の西暦の後のものです。
- <month> は月を表す 3 文字の略語です。たとえば、Jan、12 月です。
- <d> は曜日、または日付を表す整数です。

  曜日を月曜日、火曜日などに指定することはできません。月の特定の日を表す整数を指定するか、その月の第 1 平日、第 2 平日、第 3 曜日などの日付を指定します。曜日を指定する例を以下に示します。

  - Sun_1 はその月の最初の日曜日です。
  - Sun_3 はその月の第 3 日曜日です。
  - Wed_3 は毎月第 3 水曜日です。
  - 30 は月の正確な日付の一例です。
- <h> は時間です。たとえば、10h です。
- <s> は秒数で、たとえば 30 秒です。

次の式の例は、日付が 2008 年 1 月から 2009 年 1 月の間で、GMT を基準とした場合に当てはまります。

```
http.req.date.between(GMT 2008 Jan, GMT 2009 Jan)
```

次の式例は、GMT を基準とした暦年の 3 月と 3 月に続くすべての月に当てはまります。

```
sys.time.ge(GMT 2008 Mar)
```


日付と時刻を指定する場合は、大文字と小文字が区別され、エントリ間の空白の正確な数を保持する必要があることに注意してください。

```
**Note:**

In an expression that requires two time values, both must use GMT or
    both must use LOCAL. You cannot mix the two in an expression.

Unlike when you use the SYS.TIME prefix in an advanced policy
    expression, if you specify SYS.TIME in a rewrite action, the
    NetScaler returns a string in conventional date format (for example,
     Sun, 06 Nov 1994 08:49:37 GMT). For example, the following rewrite
    action replaces the http.res.date header with the NetScaler system
    time in a conventional date format:

add rewrite action sync_date replace http.res.date sys.time
```

NetScaler システム時間の表現

August 15, 2023

SYS.TIME エクスプレフィックスは、NetScaler のシステム時間を抽出します。特定のイベントが特定の時間に発生したのか、NetScaler システム時刻に従って特定の時間範囲内で発生したのかを示す式を構成できます。

次の表では、SYS.TIME プレフィックスを使用して作成できる式について説明します。

- **SYS.TIME.BETWEEN(<time1>, <time2>):**

  <time2> 戻り値が より後で <time1> より前の場合は、ブール値 TRUE を返します。

  <time1>,<time2> 引数は以下のようにフォーマットします。

  - 両方とも GMT であるか、両方がローカルである必要があります。
  - は <time2> 以降でなければなりません <time1>。

  たとえば、現在の時刻が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が月の第 1 日曜日の場合、次のように指定できます。

  - sys.time.between(GMT 2004, GMT 2006)
  - sys.time.between(GMT 2004 Jan, GMT 2006 Nov)
  - sys.time.between(GMT 2004 Jan, GMT 2006)
  - sys.time.between(GMT 2005 May Sun_1, GMT 2005 May Sun_3)
  - sys.time.between(GMT 2005 May 1, GMT May 2005 1)
  - sys.time.between(LOCAL 2005 May 1, LOCAL May 2005 1)

- **SYS.TIME.DAY:**

その月の現在の日を 1 から 31 までの数値で返します。

- **SYS.TIME.EQ(<time>):**

  現在の時間が <time> 引数と等しい場合は、ブール値 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 年 5 月 1 日 10 時 15 分 30 秒で、その日が月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

  - sys.time.eq (GMT 2005) (この例では TRUE です。)
  - sys.time.eq (グリニッジ標準時 2005 年 12 月) (この例では FALSE)
  - sys.time.eq (LOCAL 2005 年 5 月) (この例では、現在のタイムゾーンに応じて TRUE または FALSE と評価されます。)
  - sys.time.eq (グリニッジ標準時 10 時間) (この例では TRUE)
  - sys.time.eq (グリニッジ標準時 10 時間 30 秒) (この例では TRUE です)
  - sys.time.eq (グリニッジ標準時 5 月 10 日) (この例では TRUE)
  - sys.time.eq (GMT Sun) (この例では TRUE)
  - sys.time.eq (GMT May Sun_1) (この例では TRUE)

- **SYS.TIME.NE(<time>):**

  現在の時間が <time> 引数と等しくない場合は、ブール値 TRUE を返します。

- **SYS.TIME.GE(<time>):**

  <time> 現在の時刻が と等しいかそれより後の場合は、ブール値 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 年 5 月 1 日 10 時 15 分 30 秒で、その日が月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

  - sys.time.ge (GMT 2004) (この例では TRUE です。)
  - sys.time.ge (グリニッジ標準時 2005 年 1 月) (この例では TRUE です)
  - sys.time.ge (LOCAL 2005 年 5 月) (この例では TRUE か FALSE か。現在のタイムゾーンによって異なります。)
  - sys.time.ge (グリニッジ標準時 8 時間) (この例では TRUE)
  - sys.time.ge (GMT 30 分) (この例では FALSE)
  - sys.time.ge (グリニッジ標準時 5 月 10 日) (この例では TRUE)
  - sys.time.ge (グリニッジ標準時 5 月 10 時 0 分) (この例では TRUE)
  - sys.time.ge (GMT Sun) (この例では TRUE)
  - sys.time.ge (GMT May Sun_1) (この例では TRUE)

- **SYS.TIME.GT(<time>):**

  時間値が <time> 引数より後の場合は、ブール値 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 年 5 月 1 日 10 時 15 分 30 秒で、その日が月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

- sys.time.gt (GMT 2004) (この例では TRUE です)
- sys.time.gt (グリニッジ標準時 2005 年 1 月) (この例では TRUE です)
- sys.time.gt (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
- sys.time.gt (GMT 8 時間) (この例では TRUE)
- sys.time.gt (GMT 30 分) (この例では FALSE)
- sys.time.gt (グリニッジ標準時 5 月 10 日) (この例では FALSE)
- sys.time.gt (グリニッジ標準時 5 月 10 時 0 分) (この例では TRUE です)
- sys.time.gt (GMT Sun) (この例では FALSE)
- sys.time.gt (GMT May Sun_1) (この例では FALSE)

- **SYS.TIME.HOURS:**

  現在の時間を 0 から 23 までの整数で返します。

- **SYS.TIME.LE(<time>):**

  <time> 現在の時間値が 引数の前か等しい場合は、ブール値 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 年 5 月 1 日 10 時 15 分 30 秒で、その日が月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

  - sys.time.le (GMT 2006) (この例では TRUE です)
  - sys.time.le (グリニッジ標準時 2005 年 12 月) (この例では TRUE)
  - sys.time.le (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
  - sys.time.le (GMT 8 時間) (この例では FALSE)
  - sys.time.le (GMT 30 分) (この例では TRUE)
  - sys.time.le (グリニッジ標準時 5 月 10 日) (この例では TRUE)
  - sys.time.le (グリニッジ標準時 6 月 11 日) (この例では TRUE です)
  - sys.time.le (グリニッジ標準時、水曜日) (この例では TRUE)
  - sys.time.le (GMT May Sun_1) (この例では TRUE)

- **SYS.TIME.LT(<time>):**

  <time> 現在の時間値が 引数の前にある場合は、ブール値 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 年 5 月 1 日 10 時 15 分 30 秒で、その日が月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

  - sys.time.lt (GMT 2006) (この例では TRUE です)
  - sys.time.lt.time.lt (グリニッジ標準時 2005 年 12 月) (この例では TRUE)
  - sys.time.lt (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
  - sys.time.lt (GMT 8 時間) (この例では FALSE)
  - sys.time.lt (GMT 30 分) (この例では TRUE)
  - sys.time.lt (グリニッジ標準時 5 月 10 日) (この例では FALSE)
  - sys.time.lt (グリニッジ標準時 6 月 11 日) (この例では TRUE です)

  - sys.time.lt (グリニッジ標準時、水曜日) (この例では TRUE)
  - sys.time.lt (GMT May Sun_1) (この例では FALSE)

- **SYS.TIME.MINUTES:**

  現在の分を 0 から 59 までの整数で返します。

- **SYS.TIME.MONTH:**

  現在の月を抽出し、1 (1 月) から 12 (12 月) までの整数を返します。

- **SYS.TIME.RELATIVE_BOOT:**

  前回または予定された再起動に最も近いまでの秒数を計算し、整数を返します。

  最も近い起動時間が過去の場合、整数は負になります。将来の場合、整数は正です。

- **SYS.TIME.RELATIVE_NOW:**

  現在の NetScaler システム時刻と指定された時刻の間の秒数を計算し、その差を示す整数を返します。

  指定された時間が過去の場合は整数は負になり、未来の場合は整数は正になります。

- **SYS.TIME.SECONDS:**

  現在の NetScaler システム時間から秒を抽出し、その値を 0 から 59 までの整数として返します。

- **SYS.TIME.WEEKDAY:**

  現在の平日を 0 (日曜日) から 6 (土曜日) までの値で返します。

- **SYS.TIME.WITHIN (<time1>, <time2>):**

  内の時間要素 (<time1> たとえば、日や時間) を省略すると、その範囲で最も低い値を持つと見なされます。<time2> の要素を省略すると、その要素はその範囲の中で最大の値を持つと見なされます。

  時間の要素の範囲は、1 ～12 か月目、1 ～31 日目、平日 0 ～6 時、0 ～23 時、分 0 ～59 分、秒 0 ～59 です。年を指定する場合は、と <time1> の両方で指定する必要があります <time2>。

  たとえば、時刻が GMT 2005 年 5 月 10 日 10 時 15 分 30 秒で、その月の第 2 火曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

  - sys.time.within (GMT 2004、GMT 2006) (この例では TRUE)
  - sys.time.within (GMT 2004 年 1 月、GMT 2006 年 3 月) (FALSE、5 月は 1 月から 3 月の範囲外です。)
  - sys.time.within (GMT 2 月、GMT、GMT) (TRUE、5 月は 2 月から 12 月の範囲です。)
  - sys.time.within (GMT Sun_1、GMT Sun_3) (TRUE、2 番目の火曜日は第 1 日曜日と第 3 日曜日の間です。)
  - sys.time.within (GMT 2005 年 5 月 1 日 10 時、GMT 2005 年 5 月 1 日 17 時) (この例では TRUE)
  - sys.time.within (ローカル 2005 年 5 月 1 日、ローカル 2005 年 5 月 1 日) (NetScaler システムのタイムゾーンに応じて TRUE または FALSE)

- **SYS.TIME.YEAR:**

  現在のシステム時刻から年を抽出し、その値を 4 桁の整数として返します。

SSL 証明書の日付を表す表現

August 15, 2023

SSL 証明書の有効期間は、次のプレフィックスを含む式を設定することで決定できます。

```
CLIENT.SSL.CLIENT_CERT
```

次の式例では、特定の有効期限を証明書の情報と照合しています。

```
client.ssl.client_cert.valid_not_after.eq(GMT 2009)
```

次の表は、SSL 証明書の時間ベースの操作を示しています。必要な式を取得するには、最初の列の式の証明書をプレフィックス式「CLIENT.SSL.CLIENT_CERT」に置き換えます。

- <certificate>.VALID_NOT_AFTER:

  証明書の有効期限が切れる前の最終日を返します。戻り値の形式は、1970 年 1 月 1 日 GMT からの秒数 (0 時間、0 分、0 秒) です。

- **<certificate> <time1> <time2>.BETWEEN (\,)** の後には無効です**:**

  <time2> 証明書の有効期間が <time1> と 引数の間の場合は、ブール型 TRUE 値を返します。<time1> と <time2> の両方を完全に指定する必要があります。以下はその例です。

  グリニッジ標準時 1995 年 1 月は完全に指定されています。

  GMT Jan は完全には指定されていません

  GMT 1995 20 は完全に指定されていません。

  GMT Jan Mon_2 が完全に指定されていません。

  <time1><time1> と <time2> 引数は両方とも GMT か LOCAL でなければならず、は <time2> より大きくなければなりません。

  たとえば、GMT 2005 年 5 月 1 日 10 時 15 分 30 秒で、その月の第 1 日曜日の場合、次のように指定できます (評価結果は括弧内にあります)。

  - . . .between(GMT 2004, GMT 2006) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006) (TRUE)
  - . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)
  - . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)

- .. .between (ローカル 2005 年 5 月 1 日、ローカル 2005 年 5 月 1 日) (NetScaler システムのタイムゾーンに応じて TRUE または FALSE)

- **<certificate>.VALID_NOT_AFTER.DAY:**

  証明書が有効な月の最終日を抽出し、その日付に応じて 1 ～31 の数値を返します。

- **<certificate>.VALID_NOT_AFTER.EQ(<time>):**

  時間が <time> 引数と等しい場合は、ブール値 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 年 5 月 1 日 10 時 15 分 30 秒で、その日が月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧内にあります)。

  - . .eq (GMT 2005) (TRUE)
  - . . .eq(GMT 2005 Dec) (FALSE)
  - . .eq (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
  - . . .eq(GMT 10h) (TRUE)
  - . . .eq(GMT 10h 30s) (TRUE)
  - . . .eq(GMT May 10h) (TRUE)
  - . . .eq(GMT Sun) (TRUE)
  - . . .eq(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.GE(<time>):**

  <time> 時間値が引数 以上の場合は、ブール値 TRUE を返します。

  たとえば、時間値が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が 2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧内にあります)。

  - . . .ge(GMT 2004) (TRUE)
  - . . .ge(GMT 2005 Jan) (TRUE)
  - . .ge (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
  - . . .ge(GMT 8h) (TRUE)
  - . . .ge(GMT 30m) (FALSE)
  - . . .ge(GMT May 10h) (TRUE)
  - . . .ge(GMT May 10h 0m) (TRUE)
  - . . .ge(GMT Sun) (TRUE)
  - . . .ge(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.GT(<time>):**

  <time> 時間値が引数 より大きい場合は、ブール値 TRUE を返します。

  たとえば、時間値が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が 2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧内にあります)。

  - . . .gt(GMT 2004) (TRUE)

- . . .gt(GMT 2005 Jan) (TRUE)
- . .gt (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
- . . .gt(GMT 8h) (TRUE)
- . . .gt(GMT 30m) (FALSE)
- . . .gt(GMT May 10h) (FALSE)
- . . .gt(GMT Sun) (FALSE)
- .. .gt (GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_AFTER.HOURS:**

  証明書が有効な直近の 1 時間を抽出し、その値を 0 ～23 の整数で返します。

- **<certificate>.VALID_NOT_AFTER.LE(<time>):**

  <time> 時間が 引数の前か等しい場合は、ブール値 TRUE を返します。

  たとえば、時間値が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が 2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧内にあります)。

  - . .le (GMT 2006) (TRUE)
  - . . .le(GMT 2005 Dec) (TRUE)
  - .. .le (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
  - . . .le(GMT 8h) (FALSE)
  - . . .le(GMT 30m) (TRUE)
  - . . .le(GMT May 10h) (TRUE)
  - . . .le(GMT Jun 11h) (TRUE)
  - . . .le(GMT Wed) (TRUE)
  - . . .le(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.LT(<time>):**

  <time> 引数の前にある場合は、ブール値 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が月の第 1 日曜日の場合、次のように指定できます。

  - . .lt (GMT 2006) (TRUE)
  - . . .lt(GMT 2005 Dec) (TRUE)
  - . .lt (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
  - . . .lt(GMT 8h) (FALSE)
  - . . .lt(GMT 30m) (TRUE)
  - . . .lt(GMT May 10h) (FALSE)
  - . . .lt(GMT Jun 11h) (TRUE)
  - . . .lt(GMT Wed) (TRUE)
  - .. .lt (GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_AFTER.MINUTES:**

  証明書が有効である直前分を抽出し、その値を 0 ～59 の整数で返します。

- **<certificate>.VALID_NOT_AFTER.MONTH:**

  証明書が有効な最後の月を抽出し、その値を 1 (1 月) から 12 (12 月) までの整数として返します。

- **<certificate>.VALID_NOT_AFTER.RELATIVE_BOOT:**

  前回または予定された再起動に最も近いまでの秒数を計算し、整数を返します。最も近い起動時間が過去の場合、整数は負になります。将来の場合、整数は正です。

- **<certificate>.VALID_NOT_AFTER.RELATIVE_NOW;**

  現在のシステム時刻と指定された時刻の間の秒数を計算し、整数を返します。時間が過去の場合、整数は負です。未来の場合、整数は正です。

- **<certificate>.VALID_NOT_AFTER.SECONDS:**

  証明書が有効であることを示す最後の 1 秒を抽出し、その値を 0 ～59 の整数で返します。

- **<certificate>.VALID_NOT_AFTER.WEEKDAY:**

  証明書が有効な最後の平日を抽出します。0 (日曜日) から 6 (土曜日) までの数値を返して、時間値の曜日を指定します。

- **<certificate>.VALID_NOT_AFTER.WITHIN(<time1>, <time2>):**

  <time2> 時間が と の要素で定義されているすべての範囲内にある場合 <time1>、ブール値 TRUE を返します。

  から時間の要素を省略すると <time1>、その要素はその範囲内で最小の値であると見なされます。から要素を省略すると <time2>、その要素はその範囲の中で最大の値を持つと見なされます。<time1> に年を指定する場合は、で指定する必要があります <time2>。

  時間の要素の範囲は、1 ～12 か月目、1 ～31 日目、平日 0 ～6 日、時間 0 ～23 分、0 ～59 分、秒 0 ～59 です。結果が TRUE になるには、時間内の各要素が\, で指定した対応する範囲に存在する必要があります <time1> <time2>。

  たとえば、時刻が GMT 2005 年 5 月 10 日 10 時 15 分 30 秒で、その月の第 2 火曜日の場合、次のように指定できます (評価結果は括弧内にあります)。

  - . . .within(GMT 2004, GMT 2006) (TRUE)
  - .. .within (GMT 2004 年 1 月、GMT 2006 年 3 月) (FALSE、5 月は 1 月から 3 月の範囲外です。)
  - .. .within (GMT 2 月、GMT) (TRUE、5 月は 2 月から 12 月の範囲内)
  - .. .within (GMT Sun_1、GMT Sun_3) (TRUE、第 2 火曜日は第 1 日曜日から第 3 日曜日までの範囲内)
  - .. .within (GMT 2005 年 5 月 1 日 10 日、GMT 2005 年 5 月 1 日 17 時) (TRUE)
  - .. .within (ローカル 2005 年 5 月 1 日、ローカル 2005 年 5 月 1 日) (NetScaler システムのタイムゾーンに応じて TRUE または FALSE)

- **<certificate>.VALID_NOT_AFTER.YEAR:**

  証明書が有効な最後の年を抽出し、4 桁の整数を返します。

- **<certificate>.VALID_NOT_BEFORE:**

  クライアント証明書が有効になる日付を返します。

  戻り値の形式は、1970 年 1 月 1 日 GMT からの秒数 (0 時間、0 分、0 秒) です。

- **<certificate>.VALID_NOT_BEFORE.BETWEEN(<time1>, <time2>):**

  時間値が 2 つの時間引数の間にある場合は、ブール値 TRUE を返します。<time1> と <time2> の両方の引数を完全に指定する必要があります。

  以下はその例です。

  グリニッジ標準時 1995 年 1 月は完全に指定されています。
  GMT Jan は完全には指定されていません。
  GMT 1995 20 は完全には指定されていません。
  GMT 1 月 2 日 (月) は完全には指定されていません。
  <time1> 時間引数は両方とも GMT または両方とも LOCAL でなければならず、は <time2> より大きくなければなりません。

  たとえば、時間値が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が 2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧内にあります)。

  - . . .between(GMT 2004, GMT 2006) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006) (TRUE)
  - . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)
  - . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)
  - .. .between (ローカル 2005 年 5 月 1 日、ローカル 2005 年 5 月 1 日) (NetScaler システムのタイムゾーンに応じて TRUE または FALSE)

- **<certificate>.VALID_NOT_BEFORE.DAY:**

  証明書が有効な月の最終日を抽出し、その日を表す 1 ～31 の数値として返します。

- **<certificate>.VALID_NOT_BEFORE.EQ(<time>):**

  時間が <time> 引数と等しい場合は、ブール値 TRUE を返します。

  たとえば、時間値が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が 2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧内にあります)。

  - . .eq (GMT 2005) (TRUE)
  - . . .eq(GMT 2005 Dec) (FALSE)
  - .. .eq (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)

- . . .eq(GMT 10h) (TRUE)
- . . .eq(GMT 10h 30s) (TRUE)
- . . .eq(GMT May 10h) (TRUE)
- . . .eq(GMT Sun) (TRUE)
- . . .eq(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.GE(<time>):**

  時間が <time> 引数より大きい (後) か等しい場合は、ブール値 TRUE を返します。

  たとえば、時刻の値が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が 2005 年 5 月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内にあります)。

  - . . .ge(GMT 2004) (TRUE)
  - . . .ge(GMT 2005 Jan) (TRUE)
  - . .ge (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
  - . . .ge(GMT 8h) (TRUE)
  - . . .ge(GMT 30m) (FALSE)
  - . . .ge(GMT May 10h) (TRUE)
  - . . .ge(GMT May 10h 0m) (TRUE)
  - . . .ge(GMT Sun) (TRUE)
  - . . .ge(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.GT(<time>):**

  <time> 引数の後に時間がある場合はブール値 TRUE を返します。

  たとえば、時刻の値が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が 2005 年 5 月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内にあります)。

  - . . .gt(GMT 2004) (TRUE)
  - . . .gt(GMT 2005 Jan) (TRUE)
  - . .gt (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
  - . . .gt(GMT 8h) (TRUE)
  - . . .gt(GMT 30m) (FALSE)
  - . . .gt(GMT May 10h) (FALSE)
  - .. .gt (グリニッジ標準時 5 月 10 日午後 0 時) (TRUE)
  - . . .gt(GMT Sun) (FALSE)
  - .. .gt (GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_BEFORE.HOURS:**

  証明書が有効な直近の 1 時間を抽出し、その値を 0 ～23 の整数で返します。

- **<certificate>.VALID_NOT_BEFORE.LE(<time>)

  <time> 時間が 引数の前か等しい場合は、ブール値 TRUE を返します。

たとえば、時間値が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が 2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧内にあります)。

- . .le (GMT 2006) (TRUE)
- . . .le(GMT 2005 Dec) (TRUE)
- .. .le (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
- .. .le (GMT 8 時間) (FALSE)
  。- . .le (GMT 30m) (TRUE)
- . . .le(GMT May 10h) (TRUE)
- . . .le(GMT Jun 11h) (TRUE)
- . . .le(GMT Wed) (TRUE)
- . . .le(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.LT(<time>):**

  <time> 引数の前にある場合は、ブール値 TRUE を返します。

  たとえば、時間値が GMT 2005 年 5 月 1 日 10 時 15 分 30 分で、その日が 2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧内にあります)。

  - . .lt (GMT 2006) (TRUE)
  - . . .lt(GMT 2005 Dec) (TRUE)
  - . .lt (ローカル 2005 年 5 月) (現在のタイムゾーンに応じて TRUE または FALSE)
  - . . .lt(GMT 8h) (FALSE)
  - . . .lt(GMT 30m) (TRUE)
  - . . .lt(GMT May 10h) (FALSE)
  - . . .lt(GMT Jun 11h) (TRUE)
  - . . .lt(GMT Wed) (TRUE)
  - .. .lt (GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_BEFORE.MINUTES:**

  証明書が有効であることを示す最後の瞬間を抽出します。現在の分を 0 から 59 までの整数で返します。

- **<certificate>.VALID_NOT_BEFORE.MONTH:**

  証明書が有効な最後の月を抽出します。現在の月を 1 (1 月) から 12 (12 月) までの整数で返します。

- **<certificate>.VALID_NOT_BEFORE.RELATIVE_BOOT:**

  前回または予定されている NetScaler の再起動に最も近い時点までの秒数を計算し、整数を返します。最も近い起動時間が過去の場合は整数は負になり、将来の場合は整数は正になります。

- **<certificate>.VALID_NOT_BEFORE.RELATIVE_NOW:**

  現在の NetScaler システム時刻と指定した時刻の間の秒数を整数で返します。指定した時刻が過去の場合、整数は負になります。将来の場合、整数は正です。

- **<certificate>.VALID_NOT_BEFORE.SECONDS:**

  証明書が有効である最後の 1 秒を抽出します。現在の秒数を 0 から 59 までの整数で返します。

- **<certificate>.VALID_NOT_BEFORE.WEEKDAY:**

  証明書が有効な最後の平日を抽出します。平日を 0 (日曜日) から 6 (土曜日) までの数値で返します。

- **<certificate>.VALID_NOT_BEFORE.WITHIN(<time1>, <time2>):**

  時間の各要素が\, <time1><time2> 引数で定義された範囲内に存在する場合、ブール値 TRUE を返します。

  から時間の要素を省略すると <time1>、その要素はその範囲内で最小の値であると見なされます。から時間の要素を省略すると <time2>、その要素はその範囲内で最大の値を持つと見なされます。に年を指定する場合は <time1>、で指定する必要があります <time2>。時間の要素の範囲は、1 ～12 か月目、1 ～31 日目、平日 0 ～6 日、時間 0 ～23 分、0 ～59 分、秒 0 ～59 です。

  たとえば、時刻が GMT 2005 年 5 月 10 日 10 時 15 分 30 秒で、その月の第 2 火曜日の場合、次のように指定できます (評価結果は括弧内にあります)。

  - . . .within(GMT 2004, GMT 2006) (TRUE)
  - .. .within (GMT 2004 年 1 月、GMT 2006 年 3 月) (FALSE、5 月は 1 月から 3 月の範囲外です。)
  - .. .within (GMT 2 月、GMT、GMT) (TRUE、5 月は 2 月から 12 月の範囲です)
  - .. .within (GMT Sun_1、GMT Sun_3) (TRUE、第 2 火曜日は第 1 日曜日と第 3 日曜日の間です。)
  - .. .within (GMT 2005 年 5 月 1 日 10 日、GMT 2005 年 5 月 1 日 17 時) (TRUE)
  - .. .within (ローカル 2005 年 5 月 1 日、ローカル 2005 年 5 月 1 日) (NetScaler システムのタイムゾーンに応じて TRUE または FALSE)

- **<certificate>.VALID_NOT_BEFORE.YEAR:**

  証明書が有効な最後の年を抽出します。現在の年を 4 桁の整数で返します。

HTTP リクエストとレスポンスの日付の式

August 15, 2023

次の式プレフィックスは、HTTP Date ヘッダーの内容をテキストまたは日付オブジェクトとして返します。これらの値は次のように評価できます。

- 数字として。HTTP Date ヘッダーの数値は、1970 年 1 月 1 日からの秒数の形式で返されます。

  たとえば、http.req.date.mod (86400) という式は、一日の始まりからの秒数を返します。これらの値は、他の非日付の数値データと同じ演算を使用して評価できます。詳細については、日付と時刻以外の数値データの式接頭辞を参照してください。

- HTTP ヘッダーとして。日付ヘッダーは、他の HTTP ヘッダーと同じ操作を使用して評価できます。

  詳細については、「高度なポリシー式: HTTP、TCP、および UDP データの解析」を参照してください。

- テキストとして。日付ヘッダーは、他の文字列と同じ操作を使用して評価できます。

詳細については、「高度なポリシー式: テキストの評価」を参照してください。

前	説明
HTTP.REQ.DATE	HTTP Date ヘッダーの内容をテキストまたは日付オブジェクトとして返します。認識される日付形式は、RFC822 です。Sun, 06 Jan 1980 08:49:37 GMT, RFC850. Sunday, 06-Jan-80 09:49:37 GMT, and ASCTIME. Sun Jan 6 08:49:37 1980.
HTTP.RES.DATE	HTTP Date ヘッダーの内容をテキストまたは日付オブジェクトとして返します。認識される日付形式は、RFC822 です。Sun, 06 Jan 1980 8:49:37 GMT, RFC850. Sunday, 06-Jan-80 9:49:37 GMT, and ASCTIME. Sun Jan 6 08:49:37 1980.

曜日を文字列として、短い形式と長い形式で生成します

August 15, 2023

関数`WEEKDAY_STRING_SHORT`と`WEEKDAY_STRING`は、それぞれ短い形式と長い形式で、曜日を文字列として生成します。返される文字列は常に英語です。これらの関数で使用されるプレフィックスは、曜日を整数形式で返す必要があり、プレフィックスによって返される値の許容範囲は 0 ～6 です。したがって、許容範囲内の整数を返す任意のプレフィックスを使用できます。戻り値がこの範囲にない場合、またはメモリ割り当てに失敗した場合、UNDEF 条件が発生します。

関数の説明は次のとおりです。

機能	説明
<prefix>.WEEKDAY_STRING_SHORT	曜日をショートフォーマットで返します。短縮形は常に 3 文字で、最初の文字は大文字で、残りの文字は小文字です。たとえば、SYS.TIME.WEEKDAY.WEEKDAY_STRING_SHORT は、WEEKDAY 関数によって返される値が 0 の場合は Sun を返し、プレフィックスによって返される値が 6 の場合は Sat を返します。
<prefix>.WEEKDAY_STRING	曜日をロングフォーマットで返します。長い形式は常に頭文字が大文字で、残りの文字は小文字です。たとえば、WEEKDAY 関数によって返される値が 0 の場合は日曜日を返し、プレフィックスによって返される値が 6 の場合は Saturday SYS.TIME.WEEKDAY.WEEKDAY_STRING を返します。

日付と時刻以外の数値データの式プレフィックス

August 15, 2023

時間どおりに動作する式を設定できるだけでなく、次のタイプの数値データにも式を設定できます。

- HTTP リクエストの長さ、リクエスト内の HTTP ヘッダーの数など。

  詳細については、「日付以外の数値 HTTP ペイロードデータの式」を参照してください。

- IP アドレスと MAC アドレス。

  詳細については、「IP アドレスと IP サブネットの式」を参照してください。

- インターフェイス ID およびトランザクションスループットレートに関するクライアントおよびサーバデータ。

  詳細については、「数値クライアントおよびサーバーデータの式」を参照してください。

- 日付以外のクライアント証明書の数値データ。

  証明書の有効期限までの日数や暗号化キーのサイズなど、これらのプレフィクスの詳細については、「SSL 証明書の数値データのプレフィックス」を参照してください。

数値をテキストに変換

August 15, 2023

次の関数は、式のプレフィックスによって返される数値からバイナリ文字列を生成します。これらの関数は、バイナリデータの置換文字列として TCP 書き換え機能で特に有用です。TCP 書き換え機能の詳細については、書き換えを参照してください。

すべての関数は、テキスト型の値を返します。一部の関数がパラメーターとして受け入れるエンディアンは、LITTLE_ENDIAN または BIG_ENDIAN のいずれかです。

機能	説明
.SIGNED8_STRING	数値を表す 8 ビットの符号付きバイナリ文字列を生成します。値が範囲外の場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).GET_SIGNED8(16).SUB(3).SIGNED8_STR
.UNSIGNED8_STRING	数値を表す 8 ビットの符号なしバイナリ文字列を生成します。値が範囲外の場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).GET_UNSIGNED8(31).ADD(3).UNSIGNED
.SIGNED16_STRING()	数値を表す 16 ビットの符号付きバイナリ文字列を生成します。値が範囲外の場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).SKIP(12).GET_SIGNED16(0, BIG_ENDIAN).SUB(4).SIGNED16_STRING(BIG_ENDIAN)
.UNSIGNED16_STRING()	数値を表す 16 ビットの符号なしバイナリ文字列を生成します。値が範囲外の場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).GET_UNSIGNED16(47, LITTLE_ENDIAN).ADD(7).UNSIGNED16_STRING(LITTLE_ENDIAN)
.SIGNED32_STRING()	数値を表す 32 ビットの符号付きバイナリ文字列を生成します。例: HTTP.REQ.BODY(100).AFTER_STR(“delim”).GET_SIGNED32(0, BIG_ENDIAN).SUB(1).SIGNED32_STRING(BIG_ENDIAN)
.UNSIGNED8_STRING	数値を表す 8 ビットの符号なしバイナリ文字列を生成します。値が範囲外の場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).GET_UNSIGNED8(24).TYPECAST_UNSIG

機能	説明
.UNSIGNED16_STRING()	数値を表す 16 ビットの符号なしバイナリ文字列を生成します。値が範囲外の場合、undef 条件が発生します。 例:HTTP.REQ.BODY (100) .GET_UNSIGNED16 (23, LITTLE_ENDIAN) .TYPECAST_UNSIGNED_LONG_AT.ADD (10) .UNSIGNED16_STRING (LITTLE_ENDIAN)
.UNSIGNED32_STRING()	数値を表す 32 ビットの符号なしバイナリ文字列を生成します。値が範囲外の場合、undef 条件が発生します。 例: HTTP.REQ.BODY(100).AFTER_STR(“delim2”).GET_UNSIGNED32(0, BIG_ENDIAN).ADD(2).UNSIGNED32_STRING(BIG_ENDIAN)

仮想サーバーベースの式

August 15, 2023

`SYS.VSERVER("<vserver-name>")`式プレフィックスを使用すると、仮想サーバーを識別できます。このプレフィックスを持つ次の関数を使用して、指定した仮想サーバーに関連する情報を取得できます。

- **THROUGHPUT.** 仮想サーバーのスループットを Mbps (メガビット/秒) で返します。返される値は符号なし長整数です。

  使用法**:** SYS.VSERVER(“vserver”).THROUGHPUT

- **CONNECTIONS.** 仮想サーバーによって管理されている接続の数を返します。返される値は符号なし長整数です。

  使用法**:** SYS.VSERVER(“vserver”).CONNECTIONS

- **STATE.** 仮想サーバーの状態を返します。返される値は、UP、DOWN、または OUT_OF_SERVICE です。したがって、これらの値の 1 つを EQ () 演算子の引数として渡して、ブール値 TRUE または FALSE になる比較を実行できます。

  使用法**:** SYS.VSERVER(“vserver”).STATE

- **HEALTH.** 指定された仮想サーバーの UP 状態にあるサービスの割合を返します。返される値は整数です。

  使用法**:** SYS.VSERVER(“vserver”).HEALTH

- **RESPTIME.** 応答時間をマイクロ秒数を表す整数で返します。応答時間は、仮想サーバーにバインドされたすべてのサービスからの平均 TTFB（最初のバイトまでの時間）です。

使用法**:** SYS.VSERVER(“vserver”).RESPTIME

- **SURGECOUNT.** 仮想サーバーのサージキューにある要求の数を返します。返される値は整数です。

  使用法**:** SYS.VSERVER (「vserver」) .SURGECOUNT

例 **1**:

次の書き換えポリシーは、負荷分散仮想サーバー lbvServer の接続数が 10000 を超えると、書き換え処理を中止します。

```
add rewrite policy norewrite_pol sys.vserver("LBvserver").connections
.gt(10000)norewrite
```

例 **2**:

次の書き換えアクションは、カスタムヘッダー TP を挿入します。このヘッダーの値は、仮想サーバ lbvServer の全体です。

```
add rewrite action tp_header insert_http_header TP SYS.VSERVER("
LBvserver").THROUGHPUT
```

例 **3:**

次の監査ログメッセージアクションは、仮想サーバにバインドされたサービスの平均 TTFB を newnslog ログファイルに書き込みます。

```
add audit messageaction log_vserver_resptime_act INFORMATIONAL "\"NS
 Response Time to Servers:\" + sys.vserver(\"ssllb\").resptime + \"
millisec\""-logtoNewnslog YES
```

高度なポリシー式：**HTTP**、**TCP**、および **UDP** データの解析

August 15, 2023

高度なポリシー式を設定して、HTTP 要求または応答のペイロードを評価できます。HTTP 接続に関連付けられたペイロードには、HTTP ヘッダー（標準ヘッダーまたはカスタムヘッダー）、本文、接続 URL が含まれます。また、TCP または UDP パケットでペイロードを評価して処理することもできます。たとえば、HTTP 接続の場合、特定の HTTP ヘッダーが存在するかどうか、または URL に特定のクエリパラメーターが含まれているかどうかを確認できます。

式を設定して URL エンコーディングを変換し、後続の評価のために HTML または XML の「安全な」コーディングを適用できます。XPATH および JSON プレフィックスを使用して、XML ファイルと JSON ファイルの日付を評価することもできます。

また、テキストベースおよび数値の Advanced ポリシー式を使用して、HTTP 要求および応答データを評価することもできます。詳細については、「高度なポリシー式: テキストの評価 」および「高度なポリシー式: 日付、時刻、および数値の操作」を参照してください。

着信 IP パケット内のプロトコルを識別するための表現

August 15, 2023

次の表は、着信パケットのプロトコルを識別するために使用できる式の一覧です。

式	説明
CLIENT.IP.PROTOCOL	クライアントから送信された IPv4 パケットのプロトコルを識別します。
CLIENT.IPV6.PROTOCOL	クライアントから送信された IPv6 パケットのプロトコルを識別します。
SERVER.IP.PROTOCOL	サーバーから送信される IPv4 パケットのプロトコルを識別します。
SERVER.IPV6.PROTOCOL	サーバーから送信される IPv6 パケットのプロトコルを識別します。

プロトコル関数への引数

インターネット・アサインド・ナンバーズ・オーソリティ (IANA) のプロトコル番号を PROTOCOL 関数に渡すことができます。たとえば、着信パケットのプロトコルが TCP かどうかを調べたい場合は、CLIENT.IP.PROTOCOL.EQ (6) を使用できます。ここで、6 は IANA が割り当てた TCP のプロトコル番号です。一部のプロトコルでは、プロトコル番号の代わりに列挙値を渡すことができます。たとえば、CLIENT.IP.PROTOCOL.EQ (6) の代わりに CLIENT.IP.PROTOCOL.EQ (TCP) を使用できます。次の表は、列挙値を使用できるプロトコルと、PROTOCOL 関数で使用できる対応する列挙値を示しています。

プロトコル	列挙値
伝送制御プロトコル (TCP)	TCP
ユーザーデータグラムプロトコル (UDP)	UDP
インターネット制御メッセージプロトコル (ICMP)	ICMP
IPv4 および IPv6 で認証サービスを提供するための IP 認証ヘッダー (AH)	AH
カプセル化セキュリティペイロード (ESP) プロトコル	ESP
汎用ルーティングカプセル化 (GRE)	GRE
IP 内の IP カプセル化プロトコル	IPIP
IPv6 用インターネット制御メッセージプロトコル (ICMPv6)	ICMPv6

プロトコル	列挙値
IPv6 のフラグメントヘッダー	FRAGMENT

ユースケースシナリオ

プロトコル表現は、リクエストベースのポリシーとレスポンスベースのポリシーの両方で使用できます。これらの式は、負荷分散、WAN 最適化、コンテンツスイッチング、書き換え、リッスンポリシーなど、NetScaler のさまざまな機能で使用できます。この式を EQ () や NE () などの関数と組み合わせて使用すると、ポリシー内のプロトコルを識別してアクションを実行できます。

エクスプレッションの使用例は次のとおりです。

- Branch Repeater の負荷分散構成では、ワイルドカード仮想サーバーのリッスンポリシーで表現を使用できます。たとえば、ワイルドカード仮想サーバーに CLIENT.IP.PROTOCOL.EQ (TCP) というリッスンポリシーを設定して、仮想サーバーが TCP トラフィックのみを処理し、TCP 以外のトラフィックをすべてブリッジするようにすることができます。リッスンポリシーの代わりにアクセスコントロールリストを使用することもできますが、リッスンポリシーの方が処理されるトラフィックをより適切に制御できます。
- ANY タイプのコンテンツスイッチング仮想サーバーでは、受信パケットのプロトコルに基づいて要求を切り替えるコンテンツスイッチングポリシーを構成できます。たとえば、すべての TCP トラフィックを 1 つの負荷分散仮想サーバーに、TCP 以外のトラフィックをすべて別の負荷分散仮想サーバーに送信するようにコンテンツスイッチングポリシーを構成できます。
- クライアントベースの式を使用して、プロトコルに基づいてパーシステンスを設定できます。たとえば、CLIENT.IP.PROTOCOL を使用して、着信 IPv4 パケットのプロトコルに基づいて永続性を構成できます。

HTTP ヘッダーとキャッシュ制御ヘッダーの式

August 15, 2023

HTTP トラフィックを評価する一般的な方法の 1 つは、要求または応答のヘッダーを調べることです。ヘッダーは、次のようなさまざまな機能を実行できます。

- 送信者に関するデータを含むクッキーを提供します。
- 送信されているデータの種類を特定します。
- データが移動したルート (Via ヘッダー) を特定します。

> 注
>
> 操作を使用してヘッダーとテキストデータの両方を評価する場合、ヘッダーベースの操作は常にテキストベースの操作よりも優先されます。たとえば、AFTER_STR 操作をヘッダーに適用すると、現在のヘッダータイプ

> のすべてのインスタンスに対するテキストベースの AFTER_STR 操作がオーバーライドされます。

HTTP ヘッダーのプレフィックス

HTTP ヘッダーを抽出する式プレフィクスの 「HTTP ヘッダーのプレフィックス」テーブル。

HTTP ヘッダーの操作

HTTP ヘッダーのプレフィックスで指定できる操作については 、「HTTP ヘッダーの操作」表を参照してください。

キャッシュ制御ヘッダーのプレフィックス

以下のプレフィックスは、特に Cache-Control ヘッダーに適用されます。

HTTP ヘッダープレフィックス	説明
HTTP.REQ.CACHE_CONTROL	HTTP リクエストのキャッシュコントロールヘッダーを返します。
HTTP.RES.CACHE_CONTROL	HTTP レスポンスでキャッシュコントロールヘッダーを返します。

キャッシュ制御ヘッダーの操作

HTTP ヘッダーの操作はどれも Cache-Control ヘッダーに適用できます。

さらに、以下の操作は特定のタイプのキャッシュコントロールヘッダーを識別します。これらのヘッダータイプについては、RFC 2616 を参照してください。

HTTP ヘッダーオペレーション	説明
`Cache-Control header.NAME(<integer>)`	<integer>で指定された名前/値リスト内の n 番目のコンポーネントに対応する Cache-Control ヘッダーの名前をテキスト値として返します。名前と値のコンポーネントのインデックスは 0 から始まります。整数の引数で指定された値がリスト内のコンポーネントの数よりも大きい場合、長さがゼロのテキストオブジェクトが返されます。<integer> 次に例を示します。 `http.req.cache_control.name(3).contains("some_text")`

HTTP ヘッダーオペレーション	説明
Cache-Control header.IS_INVALID	リクエストまたはレスポンスに Cache-Control ヘッダーが存在しない場合は、ブール値 TRUE を返します。次に例を示します。 `http.req.cache_control.is_invalid`
`Cache-Control header.IS_PRIVATE`	キャッシュコントロールヘッダーの値が Private の場合、ブール値 TRUE を返します。次に例を示します。 `http.req.cache_control.is_private`
Cache-Control header.IS_PUBLIC	キャッシュコントロールヘッダーの値が Private の場合、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_public
Cache-Control header.IS_NO_STORE	キャッシュコントロールヘッダーの値が No-Store の場合、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_no_store
Cache-Control header.IS_NO_CACHE	キャッシュコントロールヘッダーの値が No-Cache の場合は、ブール値 TRUE を返します。例は次のとおりです。http.req.cache_control.is_no_cache
Cache-Control header.IS_MAX_AGE	キャッシュコントロールヘッダーの値が Max-Age の場合、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_max_age
Cache-Control header.IS_MIN_FRESH	キャッシュコントロールヘッダーの値が Min-Fresh の場合、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_min_fresh
Cache-Control header.IS_MAX_STALE	キャッシュコントロールヘッダーの値が Max-Sale の場合、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_max_stale
Cache-Control header.IS_MUST_REVALIDATE	キャッシュコントロールヘッダーの値が Must-Revalidate の場合は、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_must_revalidate
Cache-Control header.IS_NO_TRANSFORM	キャッシュコントロールヘッダーの値が No-Transform の場合は、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_no_transform
Cache-Control header.IS_ONLY_IF_CACHED	キャッシュ制御ヘッダーの値が「キャッシュされた場合のみ」の場合は、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_only_if_cached

HTTP ヘッダーオペレーション	説明
Cache-Control header.IS_PROXY_REVALIDATE	キャッシュコントロールヘッダーの値が Proxy-Revalidate の場合は、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_proxy_revalidate
Cache-Control header.IS_S_MAXAGE	キャッシュコントロールヘッダーの値が S-Maxage の場合、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_s_maxage
Cache-Control header.IS_UNKNOWN	キャッシュコントロールヘッダーのタイプが不明な場合は、ブール値 TRUE を返します。例は次のとおりです。 http.req.cache_control.is_unknown
Cache-Control header.MAX_AGE	キャッシュコントロールヘッダー Max-Age の値を返します。このヘッダーがないか無効な場合は、0 が返されます。以下に例を示します。
Cache-Control header.MAX_STALE	キャッシュコントロールヘッダー Max-Sale の値を返します。このヘッダーがないか無効な場合は、0 が返されます。例は次のとおりです。 http.req.cache_control.max_stale.le (3)
Cache-Control header.MIN_FRESH	キャッシュコントロールヘッダー Min-Fresh の値を返します。このヘッダーがないか無効な場合は、0 が返されます。以下に例を示します。
Cache-Control header.S_MAXAGE	キャッシュコントロールヘッダー S-Maxage の値を返します。このヘッダーが存在しないか無効な場合は、0 が返されます。以下に例を示します。

URL のセグメントを抽出するための式

August 15, 2023

URL と URL の一部 (ホスト名など)、または URL パスのセグメントを抽出できます。たとえば、次の式は、URL からイメージファイルのサフィックスを抽出することによって、イメージファイルの HTTP 要求を識別します。

```
http.req.url.suffix.eq("jpeg")|| http.req.url.suffix.eq("gif")
```

URL のほとんどの式はテキストで動作し、HTTP リクエストとレスポンスのテキストの式プレフィックスで説明されています。このセクションでは、GET オペレーションについて説明します。GET 操作では、次のプレフィックスを付けるとテキストが抽出されます。

- HTTP.REQ.URL.PATH
- VPN.BASEURL.PATH
- VPN.CLIENTLESS_BASEURL.PATH

次の表では、HTTP URL のプレフィックスを説明しています。

URL プレフィックス	説明
HTTP.REQ.URL.PATH.GET()	URL パスからスラッシュ (「/」) で区切られたリストを返します。たとえば、次の URL を考えてみます。<http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1>。次の式は、この URL から dir1 を返します。<http.req.url.path.get(1)> 次の式は dir2 を返します。http.req.url.path.get (2)
HTTP.REQ.URL.PATH.GET_REVERSE()	URL パスから、パスの末尾から始まるスラッシュ (「/」) で区切られたリストを返します。たとえば、次の URL を考えてみます。<http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1>。次の式は、この URL から index.html を返します。<http.req.url.path.get_reverse(0)>。次の式は、ディレクトリ 3 を返します。

日付以外の **HTTP** ステータスコードと数値の **HTTP** ペイロードデータの式

August 15, 2023

次の表は、日付以外の HTTP データ内の数値のプレフィックスをまとめたものです。

前	説明
HTTP.REQ.CONTENT_LENGTH	HTTP リクエストの長さを数値で返します。例は次のとおりです。http.req.content_length < 500
HTTP.RES.CONTENT_LENGTH	HTTP レスポンスの長さを数値で返します。以下は例です:http.res.content_length <= 1000
HTTP.RES.STATUS	レスポンスステータスコードを返す

前	説明
HTTP.RES.IS_REDIRECT	レスポンスコードがリダイレクトに関連付けられている場合は Boolean TRUE を返します。リダイレクト応答コードは次のとおりです。300 (複数選択可)、301 (永久移動)、302 (見つかりました)、303 (その他を参照)、305 (プロキシを使用)、307 (一時リダイレクト)。注: ステータスコード 304 は、リダイレクト HTTP 応答ステータスコードとはみなされません。ステータスコード 306 は使用されていません。

SIP エクスプレッション

August 15, 2023

NetScaler Advanced のポリシー表現言語には、セッション開始プロトコル（SIP）接続で動作する多数の式が含まれています。これらの表現は、リクエスト/レスポンスベースで動作する、サポートされているすべてのプロトコルのポリシーで使用することを目的としています。これらの式は、コンテンツスイッチング、レート制限、レスポンダ、および書き換えポリシーで使用できます。

レスポンダーポリシーで使用される SIP 表現には、特定の制限が適用されます。SIP 負荷分散仮想サーバーでは、DROP、NOOP、または RESPONDWITH アクションのみが許可されます。レスポンダーポリシーは、負荷分散仮想サーバー、オーバーライドグローバルバインドポイント、デフォルトのグローバルバインドポイント、または sip_udp ポリシーラベルにバインドできます。

SIP プロトコルで使用されるヘッダー形式は HTTP プロトコルで使用されるものと似ているため、新しい表現の多くは HTTP の類似表現と外観と機能によく似ています。各 SIP ヘッダーは、SIP メソッド、URL、およびバージョンを含む行と、その後に続く HTTP ヘッダーのように見える一連の名前と値のペアで構成されます。

以下は、その下にあるエクスプレッションテーブルで参照されている SIP ヘッダーのサンプルです。

```
INVITE sip:16@www.sip.com:5060;transport=udp SIP/2.0
Record-Route: <sip:200.200.100.22;lr=on>
Via: SIP/2.0/UDP 200.200.100.22;branch=z9hG4bK444b.c8e103d1.0;rport
   =5060;
    received=10.102.84.18
Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;
    received=10.102.84.160
From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53
   cc0185
To: "16" <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185
Call-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180
Max-Forwards: 69CSeq: 101 INVITE
```



```
User-Agent: Cisco-CP7940G/8.0
Contact: <sip:12@10.102.84.180:5060;transport=udp>
Expires: 180
Accept: application/sdp
Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,UPDATE
Supported: replaces,join,norefersub
Content-Length: 277
Content-Type: application/sdp
Content-Disposition: session;handling=optiona
<!--NeedCopy-->
```

SIP リファレンステーブル

次の表には、SIP ヘッダーを処理する式のリストが含まれています。最初の表には、リクエストヘッダーに適用される式が含まれています。ほとんどのレスポンスベースの式は、対応するリクエストベースの式とほぼ同じです。対応するリクエスト式から応答式を作成するには、式の最初の 2 つのセクションを SIP.REQ から SIP.RES に変更し、その他の明らかな調整を行います。2 番目の表には、レスポンス固有のレスポンス式が含まれ、リクエストに対応するものはありません。次の表の任意の要素を単独で完全な式として使用することも、さまざまな演算子を使用してこれらの式要素を他の要素と組み合わせてより複雑な式を作成することもできます。

SIP リクエストエクスプレッション

式	説明
SIP.REQ.METHOD	SIP リクエストのメソッドで動作します。サポートされている SIP リクエストの方法は、ACK、BYE、CANCEL、INFO、INVITE、メッセージ、通知、オプション、PRACK、公開、参照、登録、購読、更新です。この式はテキストクラスから派生したものなので、テキストに適用できるすべての操作がこのメソッドに適用できます。たとえば、INVITE sip: 16 @10 .102.84. 181:5060; transport=udp SIP/2.0 の SIP リクエストの場合、この表現は INVITE を返します。
SIP.REQ.URL	SIP リクエスト URL で動作します。この式はテキストクラスから派生したものなので、テキストに適用できるすべての操作がこのメソッドに適用できます。たとえば、INVITE sip: 16 @10 .102.84. 181:5060; transport=udp SIP/2.0 の SIP リクエストの場合、この式は sip: 16 @10 .102.84. 181:5060; transport=udp を返します。

式	説明
SIP.REQ.URL.PROTOCOL	URL プロトコルを返します。たとえば、SIP URL が 16@www.sip.com: 5060、transport=udp の場合、この式は sip を返します。
SIP.REQ.URL.HOSTNAME	SIP URL のホスト名部分を返します。たとえば、SIP: 16@www.sip.com: 5060、transport=udp の SIP URL の場合、この式は www.sip.com: 5060 を返します。
SIP.REQ.URL.HOSTNAME.PORT	SIP URL ホスト名のポート部分を返します。ポートが指定されていない場合、この式はデフォルトの SIP ポート 5060 を返します。たとえば、SIP ホスト名が www.sip.com: 5060 の場合、この式は 5060 を返します。
SIP.REQ.URL.HOSTNAME.DOMAIN	SIP URL ホスト名のドメイン名部分を返します。ホストが IP アドレスの場合、この式は誤った結果を返します。たとえば、SIP ホスト名が www.sip.com: 5060 の場合、この式は sip.com を返します。SIP ホスト名が 192.168.43. 15:5060 の場合、この式はエラーを返します。
SIP.REQ.URL.HOSTNAME.SERVER	ホストのサーバー部分を返します。たとえば、SIP ホスト名が www.sip.com: 5060 の場合、この式は www を返します。
SIP.REQ.URL.USERNAME	@ 文字の前のユーザー名を返します。たとえば、sip: 16@www.sip.com: 5060、transport=udp という SIP URL の場合、この式は 16 を返します。
SIP.REQ.VERSION	リクエストの SIP バージョン番号を返します。たとえば、INVITE sip: 16 @10 .102.84. 181:5060; transport=udp SIP/2.0 の SIP リクエストの場合、この式は SIP/2.0 を返します。
SIP.REQ.VERSION.MAJOR	メジャーバージョン番号 (ピリオドの左側の数字) を返します。たとえば、SIP バージョン番号が SIP/2.0 の場合、この式は 2 を返します。
SIP.REQ.VERSION.MINOR	マイナーバージョン番号 (ピリオドの右側にある番号) を返します。たとえば、SIP バージョン番号が SIP/2.0 の場合、この式は 0 を返します。

式	説明
SIP.REQ.CONTENT_LENGTH	コンテンツレングスヘッダーの内容を返します。この式は.sip_header_t クラスから派生したものなので、SIP ヘッダーに使用できるすべての操作を使用できます。たとえば、SIP コンテンツ長ヘッダーが「コンテンツ長:277」の場合、この式は 277 を返します。
SIP.REQ.TO	To ヘッダーの内容を返します。<sip: 16@sip_example.com > たとえば、SIP To ヘッダーが To:”16”<sip: 16@sip_example.com >; tag=00127f54ec85a6d90cc14f45-53cc0185 の場合、この式は”16”; tag=00127f54ec85a6d90cc14f45-53cc0185 を返します。
SIP.REQ.TO.ADDRESS	sip_url オブジェクトにある SIP URI を返します。SIP URI で使用できるすべての操作を使用できます。たとえば、SIP To ヘッダーが To:「16”<sip: 16@sip_example.com >; tag=00127f54ec85a6d90cc14f45-53cc0185 の場合、この式は sip: 16@sip_example.com を返します。
SIP.REQ.TO.DISPLAY_NAME	To ヘッダーの表示名部分を返します。たとえば、SIP To ヘッダーが To:”16”<sip: 16@sip_example.com >; tag=00127f54ec85a6d90cc14f45-53cc0185 の場合、この式は 16 を返します。
SIP.REQ.TO.TAG	TO ヘッダーの「タグ」名の値のペアから「タグ」値を返します。たとえば、SIP To ヘッダーが To:”16”<sip: 16@sip_example.com >; tag=00127f54ec85a6d90cc14f45-53cc0185 の場合、この式は 00127f54ec85a6d90cc14f45-53cc0185 を返します。
SIP.REQ.FROM	From ヘッダーの内容を返します。たとえば、SIP From ヘッダーが From:” 12” <sip: 12@sip_example.com >; tag=00127f54ec85a6d90cc14f45-53cc0185 の場合、この式は sip: 12@sip_example.com を返します。
SIP.REQ.FROM.ADDRESS	sip_url オブジェクトにある SIP URI を返します。SIP URI で使用できるすべての操作を使用できます。たとえば、SIP From ヘッダーが From:”12”<sip: 12@sip_example.com >; tag=00127f54ec85a6d90cc14f45-53cc0185 の場合、この式は sip: 12@sip_example.com を返します。

式	説明
SIP.REQ.FROM.DISPLAY_NAME	To ヘッダーの表示名部分を返します。たとえば、SIP From ヘッダーが From:”12”<sip: 12@sip_example.com >; tag=00127f54ec85a6d90cc14f45-53cc0185 の場合、この式は 12 を返します。
SIP.REQ.FROM.TAG	TO ヘッダーの「タグ」の名前と値のペアから「タグ」値を返します。たとえば、SIP From ヘッダーが From:”12”<sip:12@sip_example.com>; tag=00127f54ec85a6d90cc14f45-53cc0185 の場合、この式は 00127f54ec85a6d90cc14f45-53cc0185 を返します。
SIP.REQ.VIA	完全な Via ヘッダーを返します。リクエストに複数の Via ヘッダーがある場合は、最後の Via ヘッダーを返します。たとえば、サンプルの SIP ヘッダーにある 2 つの Via ヘッダーについて、次の式は Via: SIP/2.0/UDP 10.102.84. 180:5060; branch=Z9HG4BK03E76D0B; rport=5060; received=10.102.84.160 を返します。
SIP.REQ.VIA.SENTBY_ADDRESS	リクエストを送信したアドレスを返します。たとえば、Via ヘッダー Via: SIP/2.0/UDP 10.102.84. 180:5060、branch=Z9HG4BK03E76D0B、rport=5060、received=10.102.84.160 の場合、この式は 10.102.84.180 を返します。
SIP.REQ.VIA.SENTBY_PORT	リクエストを送信したポートを返します。たとえば、Via ヘッダー Via: SIP/2.0/UDP 10.102.84. 180:5060、branch=Z9HG4BK03E76D0B、rport=5060、received=10.102.84.160 の場合、この式は 5060 を返します。
SIP.REQ.VIA.RPORT	レポート名と値のペアから値を返します。たとえば、Via ヘッダー Via: SIP/2.0/UDP 10.102.84. 180:5060、branch=Z9HG4BK03E76D0B、rport=5060、received=10.102.84.160 の場合、この式は 5060 を返します。
SIP.REQ.VIA.BRANCH	ブランチ名と値のペアから値を返します。たとえば、Via ヘッダー Via: SIP/2.0/UDP 10.102.84. 180:5060、branch=Z9HG4BK03E76D0B、rport=5060、received=10.102.84.160 の場合、この式は Z9HG4BK03E76D0B を返します。

式	説明
SIP.REQ.VIA.RECEIVED	受け取った名前と値のペアから値を返します。たとえば、Via ヘッダー Via: SIP/2.0/UDP 10.102.84. 180:5060、branch=Z9HG4BK03E76D0B、rport=5060、received=10.102.84.160 の場合、この式は 10.102.84.160 を返します。
SIP.REQ.CALLID	Callid ヘッダーの内容を返します。この式は.sip_header_t クラスから派生したものなので、SIP ヘッダーに使用できるすべての操作を使用できます。たとえば、Call-ID: 00127f54-ec850017-0e46f5b9-5ec149c2 @10 .102.84.180 の SIP Callid ヘッダーの場合、この式は 00127f54-ec850017-0e46f5b9-5ec149c2 @10 .102.84.180 を返します。
SIP.REQ.CSEQ	CSEQ の CSEQ 番号を整数で返します。たとえば、SIP CSEQ ヘッダーが CSeq: 101 INVITE の場合、この式は 101 を返します。
SIP.REQ.HEADER()	指定された SIP ヘッダーを返します。、目的のヘッダーの名前に置き換えてください。たとえば、SIP From ヘッダーを返すには、SIP.REQ.HEADER (「From」) と入力します。
SIP.REQ.HEADER().INSTANCE()	指定した SIP ヘッダーの指定されたインスタンスを返します。同じ SIP ヘッダーのインスタンスが複数発生する可能性があります。このような SIP ヘッダーの特定のインスタンス (たとえば、特定の Via ヘッダー) が必要な場合は、に数値を入力してそのヘッダーを指定できます。ヘッダーインスタンスは最後 (0) から最初まで一致します。つまり、SIP.REQ.HEADER (「Via」) .INSTANCE (0) は Via ヘッダーの最後のインスタンスを返し、SIP.REQ.HEADER (「Via」) .INSTANCE (1) は Via ヘッダーの最後のインスタンスを返す、という具合です。たとえば、サンプルの SIP ヘッダーで使用した場合、SIP.REQ.HEADER (「Via」) .INSTANCE (1) は Via: SIP/2.0/UDP 10.102.84. 180:5060; branch=Z9HG4BK03E76D0B; rport=5060 を返します。

式	説明
SIP.REQ.HEADER().VALUE()	指定した SIP ヘッダーの指定されたインスタンスの内容を返します。使い方は前の式とほぼ同じです。たとえば、前のテーブルエントリの SIP ヘッダーの例で使用した場合、SIP.REQ.HEADER (「Via」) .VALUE (1) は SIP/2.0/UDP 10.102.84. 180:5060; branch=Z9HG4BK03E76D0B; rport=5060 を返します。
SIP.REQ.HEADER().COUNT	特定のヘッダーのインスタンス数を整数で返します。たとえば、上記の SIP ヘッダーの例で使用した場合、SIP.REQ.HEADER (「Via」) .COUNT は 2 を返します。
SIP.REQ.HEADER().EXISTS	指定されたヘッダーが存在するかどうかに応じて、true または false のブール値を返します。たとえば、上記の SIP ヘッダーの例で使用した場合、SIP.REQ.HEADER (「Expires」) .EXISTS は true を返し、SIP.REQ.HEADER (「発信者 ID」) .EXISTS は false を返します。
SIP.REQ.HEADER().LIST	指定されたヘッダーのコンマで区切られたパラメータリストを返します。たとえば、上記の SIP ヘッダーの例で使用した場合、SIP.REQ.HEADER (「許可」) .LIST は ACK、BYE、CANCEL、INVITE、NOTIFERE、NOTIFERE、NOTIFERE、REFERE、Register、UPDATE を返します。.GET () という文字列を追加して、特定のリスト項目を選択できます。たとえば、上記のリストから最初の項目 (ACK) を取得するには、SIP.REQ.HEADER (「許可」) .LIST.GET (0) と入力します。2 番目の項目 (BYE) を抽出するには、SIP.REQ.HEADER (「許可」) .LIST.GET (1) と入力します。注: 指定されたヘッダーに名前と値のペアのリストが含まれている場合、名前と値のペア全体が返されます。

式	説明
SIP.REQ.HEADER().TYPECAST_SIP_HEADER_T(””)	にタイプキャストします。任意のテキストを.sip_header_t クラスにタイプキャストできます。その後、すべてのヘッダーベースの操作を使用できます。この操作を実行すると、で使用できるすべての操作を適用できます。たとえば、SIP.REQ.CONTENT_LENGTH.TYPECAST_SIP_HEADER_T という式は、Content-Length ヘッダーのすべてのインスタンスをタイプキャストします。この操作を実行すると、指定したヘッダーのすべてのインスタンスにすべてのヘッダー操作を適用できます。
SIP.REQ.HEADER().CONTAINS().	指定されたテキスト文字列が指定されたヘッダーのいずれかのインスタンスに存在する場合、true を返します。指定されたヘッダーのすべてのインスタンスで動作します。ヘッダーインスタンスは最後 (0) から最初まで一致します。
SIP.REQ.HEADER().EQUALS_ANY()	に関連付けられたいずれかのパターンが、指定されたヘッダーのいずれかのインスタンスの内容と一致する場合、true を返します。指定されたヘッダーのすべてのインスタンスで動作します。ヘッダーインスタンスは最後 (0) から最初まで一致します。
SIP.REQ.HEADER().CONTAINS_ANY()	に関連付けられたいずれかのパターンが、指定されたヘッダーのいずれかのインスタンスの内容と一致する場合、ブール値 true を返します。指定されたヘッダーのすべてのインスタンスで動作します。ヘッダーインスタンスは最後 (0) から最初まで一致します。
SIP.REQ.HEADER().CONTAINS_INDEX()	に関連付けられた一致するパターンのインデックスを返します。そのパターンが指定されたヘッダーのいずれかのインスタンスの内容と一致する場合です。指定されたヘッダーのすべてのインスタンスで動作します。ヘッダーインスタンスは最後 (0) から最初まで一致します。
SIP.REQ.HEADER().EQUALS_INDEX()	に関連付けられた一致するパターンのインデックスを返します。そのパターンが指定されたヘッダーのいずれかのインスタンスと一致する場合です。指定されたヘッダーのすべてのインスタンスで動作します。ヘッダーインスタンスは最後 (0) から最初まで一致します。

式	説明
SIP.REQ.HEADER().SUBSTR()	指定された文字列が指定されたヘッダーのいずれかのインスタンスに存在する場合、この式はその文字列を返します。たとえば、SIP ヘッダー Via: SIP/2.0/UDP 10.102.84. 180:5060; branch=Z9HG4BK03E76D0B; rport=5060; received=10.102.84.160”の SIP.REQ.HEADER (「Via」) .SUBSTR (“rport=5060”) は「rport=5060」.sisi を返します p.req.Header (「Via」) .SUBSTR (「rport=5061”) は空の文字列を返します。
SIP.REQ.HEADER().AFTER_STR()	指定された文字列が指定されたヘッダーのいずれかのインスタンスに存在する場合、この式はその文字列の直後の文字列を返します。たとえば、SIP ヘッダー Via: SIP/2.0/UDP 10.102.84. 180:5060; branch=Z9HG4BK03E76D0B; rport=5060; received=10.102.84.160 の場合、SIP.REQ.HEADER (「Via」) .AFTER_STR (「rport=」) という表現は 5060 を返します。
SIP.REQ.HEADER().REGEX_MATCH()	指定された正規表現 (regex) が指定されたヘッダーのいずれかのインスタンスと一致する場合、true を返します。正規表現は、re 正規表現 の形式で指定する必要があります。正規表現の長さは 1499 文字を超えることはできません。PCRE 正規表現ライブラリに準拠している必要があります。PCRE 正規表現構文のドキュメントについては、http://www.pcre.org/pcre.txtを参照してください。pcrepattern のマニュアルページには、PCRE 正規表現を使用してパターンを指定する方法に関する有用な情報もあります。この表現でサポートされている正規表現構文には、PCRE とはいくつか違いがあります。バックリファレンスは許可されていません。再帰的な正規表現は避けるべきです。うまくいくものもあれば、そうでないものもあります。ドット (.) メタ文字は改行と一致します。ユニコードはサポートされていません。set_text_Mode (IGNORECASE) は (?) をオーバーライドします i) 正規表現で指定された内部オプション。

式	説明
SIP.REQ.HEADER().REGEX_SELECT()	指定された正規表現が指定されたヘッダーのいずれかのインスタンス内の任意のテキストと一致する場合、この式はテキストを返します。たとえば、SIP ヘッダー Via: SIP/2.0/UDP 10.102.84. 180:5060; branch=Z9HG4BK03E76D0B; rport=5060; received=10.102.84.160 の場合、SIP.REQ.HEADER (「Via」) .REGEX_SELECT (「received= [0-9] {1,3}」という表現になります。[0-9] {1,3}。9] {1,3}. [0-9] {1,3}. [0-9] {1,3}」) が受け取った返品 =10.102.84.160。
SIP.REQ.HEADER().AFTER_REGEX()	指定された正規表現が指定されたヘッダーのいずれかのインスタンスのテキストと一致する場合、この式はそのテキストの直後の文字列を返します。たとえば、SIP ヘッダー Via: SIP/2.0/UDP 10.102.84. 180:5060、branch=Z9HG4BK03E76D0B、rport=5060、received=10.102.84.160 の場合、SIP.REQ.HEADER (「Via」) .AFTER_REGEX (「received=」) という表現は 10.102.84.160 を返します。
SIP.REQ.HEADER().BEFORE_REGEX()	指定された正規表現が指定されたヘッダーのいずれかのインスタンスのテキストと一致する場合、この式はそのテキストの直前の文字列を返します。たとえば、SIP ヘッダー Via: SIP/2.0/UDP 10.102.84. 180:5060; branch=Z9HG4BK03E76D0B; rport=5060; received=10.102.84.160 の場合、SIP.REQ.HEADER (「Via」) .BEFORE_REGEX (「[0-9] {3}. [0-9] {3}. [0-9] {3}. [0-9] {1,3}. [0-9] {1,3}. [0-9] {1,3}」) は received= を返します。
SIP.REQ.FULL_HEADER	終端の CR/LF を含む SIP ヘッダー全体を返します。
SIP.REQ.IS_VALID	リクエスト形式が有効な場合は true というブール値を返します。
SIP.REQ.BODY()	指定された長さまでのリクエスト本文を返します。指定された長さがリクエスト本文の長さより大きい場合、この式はリクエスト本文全体を返します。
SIP.REQ.LB_VSERVER	現在のリクエストを処理している負荷分散仮想サーバー (LB vserver) の名前を返します。
SIP.REQ.CS_VSERVER	現在のリクエストを処理しているコンテンツスイッチ仮想サーバー (CS vserver) の名前を返します。

SIP レスポンスエクスプレッション

式	説明
SIP.RES.STATUS	SIP 応答ステータスコードを返します。たとえば、応答の最初の行が SIP/2.0 100 Trying の場合、この式は 100 を返します。
SIP.RES.STATUS_MSG	SIP 応答ステータスメッセージを返します。たとえば、応答の最初の行が SIP/2.0 100 Trying の場合、この式は Trying を返します。
SIP.RES.IS_REDIRECT	レスポンスコードがリダイレクトの場合は true を返します。
SIP.RES.METHOD	CSeq ヘッダ内の要求メソッド文字列から抽出された応答メソッドを返します。

HTTP、HTML、XML エンコーディングと「安全な」文字の操作

August 15, 2023

以下の操作は、リクエストまたはレスポンスの HTML データと POST ボディの XML データのエンコーディングで動作します。

- **\ .HTML_XML_SAFE:** 次の例のように、特殊文字を **XML** セーフフォーマットに変換します
  <text>。

  左向きの山括弧 (<) は < に変換されます
  < 右向きの山括弧 () は >
  アンパサンド (&) に変換されます &
  この操作は、クロスサイトスクリプティング攻撃から保護します。変換されたテキストの最大長は 2048 バイトです。これは読み取り専用の操作です。

  変換を適用すると、エクスプレッションで指定する追加の演算子が、選択したテキストに適用されます。次に例を示します。

  http.req.url.query.html_xml_safe. contains(“myQueryString”)

- **\ .HTTP_HEADER_SAFE:** 入力テキストのすべての改行 **(** ‘**\ n**’ **)** 文字を’ **%0A** ‘に変換して、入力を **HTTP** ヘッダーで安全に使用できるようにします
  <text>。

  この操作により、応答分割攻撃を防ぐことができます。

変換されたテキストの最大長は 2048 バイトです。これは読み取り専用の操作です。

- **\ .HTTP_URL_SAFE:** 安全ではない **URL** 文字を ‘**%xx**’ 値に変換します
  <text>。ここで、「xx」は入力文字を 16 進数で表現したものです。たとえば、アンパサンド (&) は URL セーフエンコーディングでは%26 と表されます。変換されたテキストの最大長は 2048 バイトです。これは読み取り専用の操作です。

  URL セーフ文字は次のとおりです。他のすべては安全ではありません。

  - 英数字:a-z、A-Z、0-9
  - Asterix: “*”
  - アンパサンド:「&」
  - アットサイン:「@」
  - コロン:「:」
  - カンマ:「,」
  - ドル:「$」
  - Dot: “.”
  - 等しい:「=」
  - Exclamation mark: “!”
  - ハイフン:「-」
  - 開括弧と閉じ括弧:「(「,「)」
  - パーセント:「%」
  - プラス:「+」
  - セミコロン:「;」
  - 一重引用符:「’」
  - スラッシュ:「/」
  - 疑問符:「?」
  - チルダ:「~」
  - アンダースコア:「_」

- **<text>.MARK_SAFE:**

  いかなる種類のデータ変換も適用せずに、テキストを安全とマークします。

.SET_TEXT_MODE(URLENCODED NOURLENCODED)

- バイトストリーム内のすべての%HH エンコーディングを変換します。この操作は文字に対して動作します (バイトではありません)。デフォルトでは、1 バイトは ASCII エンコーディングの文字を表します。ただし、URLENCODED モードを指定した場合、3 バイトで 1 文字を表すことができます。

  次の例では、PREFIX (3) 操作によってターゲットの最初の 3 文字が選択されます。

  ```
  http.req.url.hostname.prefix(3)
  ```


次の例では、NetScaler はターゲットから最大 9 バイトを選択できます。

```
http.req.url.hostname.set_text_mode(urlencoded).prefix(3)
```

.SET_TEXT_MODE(PLUS_AS_SPACE NO_PLUS_AS_SPACE):

- プラス文字 (+) の処理方法を指定します。PLUS_AS_SPACE オプションはプラス文字を空白に置き換えます。たとえば、「ハロー + ワールド」というテキストは「ハローワールド」になります。NO_PLUS_AS_SPACE オプションを指定すると、プラス文字はそのまま残ります。

.SET_TEXT_MODE(BACKSLASH_ENCODED NO_BACKSLASH_ENCODED):

- <text> で表されるテキストオブジェクトに対してバックスラッシュデコードを実行するかどうかを指定します。

  BACKSLASH_ENCODED が指定されている場合、SET_TEXT_MODE 演算子はテキストオブジェクトに対して次の操作を実行します。

  - 出現する「\ XXX」はすべて文字「Y」に置き換えられます (XXX は 8 進法の数字を表し、Y は XXX と同等の ASCII を表します)。このタイプのエンコーディングの 8 進値の有効範囲は 0 ～377 です。たとえば、エンコードされたテキスト「http\ 72//」と「http\ 072//」は両方ともに<http://>にデコードされます。ここで、コロン (:) は ASCII で 8 進数値「72」に相当します。
  - 出現する「\xHH」はすべて文字「Y」に置き換えられます（HH は 16 進法の数字を表し、Y は HH と同等の ASCII を表します)。たとえば、エンコードされたテキスト「http\ x3a//」は<http://>にデコードされます。ここで、コロン (:) は ASCII で 16 進値の「3a」に相当します。
  - 出現する「\ uWWxx」はすべて文字シーケンス「YZ」に置き換えられます (WW と XX は 2 つの異なる 16 進値を表し、Y と Z はそれぞれ WW と XX の ASCII 値を表します)。たとえば、エンコードされたテキスト「http%u3a2f/」と「http%u003a//」は両方とも<http://>にデコードされます。ここで、「3a」と「2f」は 2 つの 16 進値で、コロン (:) とフォワードスラッシュ (「/」) はそれぞれの ASCII 文字を表します。
  - 「\ b」、「\ n」、「\ t」、「\ f」、「\ r」はすべて、対応する ASCII 文字に置き換えられます。

  NO_BACKSLASH_ENCODED が指定されている場合、テキストオブジェクトではバックスラッシュデコードは実行されません。

.SET_TEXT_MODE(BAD_ENCODE_RAISE_UNDEF NO_BAD_ENCODE_RAISE_UNDEF):

- <text>URLENCODED または BACKSLASH_ENCODED モードのいずれかが設定されていて、で表されるテキストオブジェクトで指定されたエンコーディングモードに対応する不正なエンコーディングが見つかった場合に、関連する未定義のアクションを実行します。

<text>NO_BAD_ENCODE_RAISE_UNDEF が指定されている場合、で表されるテキストオブジェクトに不正なエンコーディングが発生しても、関連する未定義のアクションは実行されません。

TCP、UDP、および VLAN データの式

August 15, 2023

TCP と UDP のデータは、文字列または数字の形式をとります。TCP および UDP データの文字列値を返す式プレフィクスについては、任意のテキストベースの操作を適用できます。詳細については、「高度なポリシー式: テキストの評価」を参照してください。

送信元ポートなどの数値を返す式プレフィクスについては、算術演算を適用できます。詳細については、「式の接頭辞の基本操作 」および「数値の複合演算」を参照してください。

次の表は、クライアントから TCP および UDP データを抽出するプレフィックスをまとめたものです。

GET オペレーション	説明
CLIENT.TCP.PAYLOAD(<integer>)	TCP ペイロードデータを文字列として返します。ペイロードの最初の文字から始まり、<integer> 引数の文字数まで続きます。このプレフィックスには任意のテキストベースの操作を適用できます。
CLIENT.TCP.SRCPORT	現在のパケットの送信元ポートの ID を数値で返します。
CLIENT.TCP.DSTPORT	現在のパケットの宛先ポートの ID を数値で返します。
CLIENT.TCP.OPTIONS	クライアントによって設定された TCP オプションを返します。TCP オプションの例としては、最大セグメントサイズ (MSS)、ウィンドウスケール、選択的確認応答 (SACK)、およびタイムスタンプオプションなどがあります。このプレフィックスには、カウント、タイプ ()、タイプ名 () 演算子を使用できます。サーバによって設定される TCP オプションについては、SERVER.TCP.OPTIONS プレフィクスを参照してください。
CLIENT.TCP.OPTIONS.COUNT	クライアントが設定した TCP オプションの数を返します。

GET オペレーション	説明
CLIENT.TCP.OPTIONS.TYPE()	型 (またはオプションの種類) が引数として指定されている TCP オプションの値を返します。値は、ビッグエンディアン形式 (またはネットワークバイトオーダー) のバイト文字列として返されます。パラメータ: タイプ-タイプ値
CLIENT.TCP.OPTIONS.TYPE_NAME()	列挙定数が引数として指定されている TCP オプションの値を返します。引数として渡すことができる列挙定数は、REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, MAXSEG です。これらの列挙定数の代わりに TCP オプションの種類を指定するには、CLIENT.TCP.OPTIONS.TYPE () を使用してください。他の TCP オプションについては、CLIENT.TCP.OPTIONS.TYPE () を使用する必要があります。パラメータ:m-TCP オプション列挙定数
CLIENT.TCP.REPEATER_OPTION.EXISTS	リピーター TCP オプションが存在する場合は、ブール値 TRUE を返します。
CLIENT.TCP.REPEATER_OPTION.IP	リピータ TCP オプションからブランチリピータの IPv4 アドレスを返します。
CLIENT.TCP.REPEATER_OPTION.MAC	リピータ TCP オプションからブランチリピータの MAC アドレスを返します。
CLIENT.UDP.DNS.DOMAIN	DNS ドメイン名を返します。
CLIENT.UDP.DNS.DOMAIN.EQ(“”)	ドメイン名が 引数と一致する場合、ブール値 TRUE を返します。比較では、大文字と小文字は区別されません。以下は例です:client.udp.dns.domain.eq (「www.mycompany.com」)
CLIENT.UDP.DNS.IS_AAAAREC	レコードタイプが AAAA の場合、ブール値 TRUE を返します。これらのタイプのレコードは、フォワードルックアップでの IPv6 アドレスを示します。
CLIENT.UDP.DNS.IS_ANYREC	レコードタイプがなんであれ、ブール値 TRUE を返します。
CLIENT.UDP.DNS.IS_AREC	レコードがタイプ A の場合は、ブール型 TRUE を返します。タイプ A のレコードはホストアドレスを提供します。
CLIENT.UDP.DNS.IS_CNAMEREC	レコードが CNAME 型の場合、ブール値 TRUE を返します。リソースを識別するために複数の名前を使用するシステムでは、1 つの正規名といくつかのエイリアスがあります。CNAME は正規名を提供します。

GET オペレーション	説明
CLIENT.UDP.DNS.IS_MXREC	レコードのタイプが MX (メールエクスチェンジャー) の場合は、ブール値 TRUE を返します。この DNS レコードは、優先順位とホスト名を記述します。同じドメイン名の MX レコードは、ドメイン内のメールサーバーと各サーバーの優先度を指定します。
CLIENT.UDP.DNS.IS_NSREC	レコードが NS 型の場合、ブール値 TRUE を返します。これは、ホスト名と関連する A レコードを含むネームサーバーレコードです。これにより、NS レコードに関連付けられているドメイン名を検索できます。
CLIENT.UDP.DNS.IS_PTRREC	レコードが PTR 型の場合、ブール値 TRUE を返します。これはドメイン名ポインタであり、ドメイン名を IPv4 アドレスに関連付けるためによく使用されます。
CLIENT.UDP.DNS.IS_SOAREC	レコードが SOA 型の場合、ブール値 TRUE を返します。これが権限記録の始まりです。
CLIENT.UDP.DNS.IS_SRVREC	レコードが SRV 型の場合、ブール値 TRUE を返します。これは MX レコードのより一般的なバージョンです。
CLIENT.UDP.DSTPORT	現在のパケットの UDP 宛先ポートの数値 ID を返します。
CLIENT.UDP.SRCPORT	現在のパケットの UDP 送信元ポートの数値 ID を返します。
CLIENT.UDP.LENGTH	現在のパケットの UDP 長の数値 ID を返します。
CLIENT.UDP.CHECKSUM	現在のパケットの UDP チェックサムの数値 ID を返します。
CLIENT.UDP.PAYLOAD	現在のパケットの UDP ペイロードを返します。
CLIENT.UDP.RADIUS	現在のパケットの RADIUS データを返します。
CLIENT.UDP.RADIUS.ATTR_TYPE()	引数として指定された属性タイプの値を返します。
CLIENT.UDP.RADIUS.USERNAME	RADIUS ユーザー名を返します。
CLIENT.TCP.MSS	現在の接続の最大セグメントサイズ (MSS) を数値で返します。
CLIENT.VLAN.ID	現在のパケットが NetScaler に入った VLAN の数値 ID を返します。

次の表は、サーバーから TCP および UDP データを抽出するプレフィックスをまとめたものです。

GET オペレーション	説明
SERVER.TCP.DSTPORT	現在のパケットの宛先ポートの数値 ID を返します。
SERVER.TCP.SRCPORT	現在のパケットの送信元ポートの数値 ID を返します。
SERVER.TCP.OPTIONS	サーバーによって設定された TCP オプションを返します。TCP オプションの例としては、最大セグメントサイズ (MSS)、ウィンドウスケール、選択的確認応答 (SACK)、およびタイムスタンプオプションなどがあります。このプレフィックスには、カウント、タイプ ()、タイプ名 () 演算子を使用できます。クライアントによって設定される TCP オプションについては、CLIENT.TCP.OPTIONS プレフィクスを参照してください。
SERVER.TCP.OPTIONS.COUNT	サーバーが設定した TCP オプションの数を返します。
SERVER.TCP.OPTIONS.TYPE()	型 (またはオプションの種類) が引数として指定されている TCP オプションの値を返します。値は、ビッグエンディアン形式 (またはネットワークバイトオーダー) のバイト文字列として返されます。パラメータ: タイプ-タイプ値
SERVER.TCP.OPTIONS.TYPE_NAME()	列挙定数が引数として指定されている TCP オプションの値を返します。引数として渡すことができる列挙定数は、REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, MAXSEG です。これらの列挙定数の代わりに TCP オプションの種類を指定するには、CLIENT.TCP.OPTIONS.TYPE () を使用してください。他の TCP オプションについては、CLIENT.TCP.OPTIONS.TYPE () を使用する必要があります。パラメータ:m-TCP オプション列挙定数
SERVER.VLAN	現在のパケットが NetScaler に入った VLAN 上で動作します。
SERVER.UDP.DSTPORT	現在のパケットの UDP 宛先ポートの数値 ID を返します。
SERVER.UDP.SRCPORT	現在のパケットの UDP 送信元ポートの数値 ID を返します。
SERVER.UDP.LENGTH	現在のパケットの UDP 長の数値 ID を返します。
SERVER.UDP.CHECKSUM	現在のパケットの UDP チェックサムの数値 ID を返します。
SERVER.UDP.PAYLOAD	現在のパケットの UDP ペイロードを返します。

GET オペレーション	説明
SERVER.VLAN.ID	現在のパケットが NetScaler に入った VLAN の数値 ID を返します。

DNS メッセージを評価し、そのキャリアプロトコルを識別するための式

August 15, 2023

DNS 要求と応答は、それぞれ DNS.REQ と DNS.RES で始まる式を使用して評価できます。DNS メッセージの送信に使用されているトランスポート層プロトコルを特定することもできます。

次の関数は DNS クエリの内容を返します。

機能	説明
DNS.REQ.QUESTION.DOMAIN	DNS クエリの質問セクションのドメイン名 (QNAME フィールドの値) を返します。ドメイン名はテキスト文字列として返され、EQ ()、NE ()、およびテキストを処理するその他の関数に渡すことができます。
DNS.REQ.QUESTION.TYPE	DNS クエリ内のクエリタイプ (QTYPE フィールドの値) を返します。このフィールドは、ネームサーバが照会されるリソースレコードのタイプ (A、NS、CNAME など) を示します。戻り値は、EQ () および NE () 関数を使用して、A、AAAA、NS、SRV、PTR、CNAME、SOA、MX、ANY のいずれかの値と比較できます。メモ:TYPE 関数では EQ () および NE () 関数のみを使用できます。 例:DNS.REQ.QUESTION.TYPE.EQ (MX)

次の関数は DNS レスポンスの内容を返します。

機能	説明
DNS.RES.HEADER.RCODE	DNS レスポンスのヘッダーセクションのレスポンスコード (RCODE フィールドの値) を返します。RCODE 関数では、EQ () および NE () 関数のみを使用できます。指定できる値は、NOERROR、FORMERR、SERVFAIL、NXDOMAIN、NOTIMP、および拒否です。

機能	説明
DNS.RES.QUESTION.DOMAIN	DNS レスポンスの質問セクションにドメイン名 (QNAME フィールドの値) を返します。ドメイン名はテキスト文字列として返され、EQ ()、NE ()、およびテキストを処理するその他の関数に渡すことができます。
DNS.RES.QUESTION.TYPE	DNS レスポンスの質問セクションにクエリタイプ (QTYPE フィールドの値) を返します。このフィールドは、レスポンスに含まれるリソースレコードのタイプ (A、NS、CNAME など) を示します。戻り値は、EQ () および NE () 関数を使用して、A、AAAA、NS、SRV、PTR、CNAME、SOA、MX、ANY のいずれかの値と比較できます。TYPE 関数では、EQ () および NE () 関数のみを使用できます。例:DNS.RES.QUESTION. TYPE.EQ (SOA)

次の関数はトランスポート層プロトコル名を返します。

機能	説明
DNS.REQ.TRANSPORT	DNS クエリの送信に使用されたトランスポート層プロトコルの名前を返します。返される値は TCP と UDP です。TRANSPORT 関数で使用できるのは EQ () 関数と NE () 関数のみです。例:DNS.REQ.TRANSPORT.EQ (TCP)
DNS.RES.TRANSPORT	DNS 応答に使用されたトランスポート層プロトコルの名前を返します。返される値は TCP と UDP です。TRANSPORT 関数で使用できるのは EQ () 関数と NE () 関数のみです。例:DNS.RES.TRANSPORT.EQ (TCP)

次の関数は、クエリに DNS ECS オプションが含まれている場合と含まれていない場合に、一致した場所の名前を返します。

機能	説明
DNS.REQ.OPT.ECS.IP.MATCHES_LOCATION	DNS ECS オプションを使用して、クエリで使用された一致した場所の名前を返します。例: (DNS.REQ.OPT.ECS.IP.MATCHES_LOCATION(".CH…..")

機能	説明
client.IP.SRC.MATCHES_LOCATION	DNS ECS オプションを指定せずにクエリで使用された、一致した場所の名前を返します。例: (client.IP.SRC.MATCHES_LOCATION(“.CH…..”)
DNS.REQ.OPT.ECS.IP.MATCHES_LOCATION または client.ip.src.matches_LOCATION	DNS トラフィックにクエリで ECS オプションがある場合とない場合がある場合に、ポリシーで使用される共通の式。例: “(((DNS.REQ.OPT.ECS.IP.MATCHES_LOCATION(“.CH…..”).typecast_text_t) ALT (client.IP.SRC.MATCHES_LOCATION(“.CH…..”).typecast_text_t)).eq(“true”))”

XPath と HTML、XML、または JSON エクスプレッション

August 15, 2023

高度なポリシーインフラストラクチャは、HTML、XML、および JavaScript オブジェクト表記 (JSON) ファイルからデータを評価および取得するための式をサポートしています。これにより、HTML、XML、または JSON ドキュメント内の特定のノードを検索したり、ファイルにノードが存在するかどうかを確認したり、XML コンテキスト内のノード (たとえば、特定の親を持つノードや特定の値を持つ特定の属性を持つノード) を検索したり、それらのノードの内容を返したりすることができます。さらに、書き換え式では XPath 式を使用できます。

XPath 用の高度なポリシー表現の実装は、HTML または XML テキストを指定する高度なポリシー表現プレフィックス (「HTTP.REQ.BODY」など) と、XPath 式を引数とする XPATH 演算子で構成されています。

HTML ファイルは、主に自由形式のタグとテキスト要素のコレクションです。XPath 式を引数として取る XPATH_HTML 演算子を使用して HTML ファイルを処理できます。JSON ファイルは、名前と値のペアのコレクション、または順序付けられた値のリストです。XPATH 式を引数として取る XPATH_JSON 演算子を使用して JSON ファイルを処理できます。

- **<text>.XPATH(xpathex):**

  XML ファイルを操作し、ブール値を返します。

  たとえば、次のエクスプレッションは、XML ファイルの最初の 1000 バイト以内に「Creator」というノードが「Book」ノードの下にある場合に、ブール値 TRUE を返します。

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%boolean(//Book/creator)%)
  ```

  パラメーター:

  xpathex-XPath ブーリアンエクスプレッション

- **<text>.XPATH(xpathex):**

  XML ファイルを操作し、データ型「double」の値を返します。

  たとえば、次の式は、文字列が XML ファイルの最初の 1000 バイトにある場合、文字列「36」（価格値）をデータ型「double」の値に変換します。

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%number(/Book/price)%)
  ```

  パラメーター:

  xpathex-XPath 数値表現

例:

```
    <Book>
    <creator>
        <Person>
            <name>Milton</name>
        </Person>
    </creator>
    <title>Paradise Lost</title>
    </Book>
<!--NeedCopy-->
```

- **<text>.XPATH(xpathex):**

  XML ファイルを操作し、ノードセットまたは文字列を返します。ノードセットは、標準の XPath 文字列変換ルーチンを使用して対応する文字列に変換されます。

  たとえば、次のエクスプレッションは、本文の最初の 1000 バイトで「/book/Creator」(ノードセット) で囲まれているすべてのノードを選択します。

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%/Book/creator%)
  ```

  パラメーター:

  xpathex-XPath 式

- **<text>.XPATH_HTML(xpathex)**

  HTML ファイルを操作し、テキスト値を返します。

  たとえば、次の式は HTML ファイルに対して動作し、title HTML 要素が最初の 1000 バイトに見つかった場合、<title\></title\> タグで囲まれたテキストを返します。

  ```
  HTTP.REQ.BODY(1000).XPATH_HTML(xp%/html/head/title%)
  ```

  パラメーター:

  xpathex-XPath テキスト表現

- **<text>.XPATH_HTML_WITH_MARKUP(xpathex)**

HTML ファイルを操作して、ドキュメントの選択した部分全体を含む文字列を返します。これには、要素タグを囲むマークアップなどのマークアップも含まれます。

次の式は HTML ファイル上で動作し <\ title>、マークアップを含むタグ内のすべてのコンテンツを選択します。

```
HTTP.REQ.BODY(1000).XPATH_HTML_WITH_MARKUP( xp%/html/head/title%)
```

式によって選択された HTML 本文の部分には、今後の処理のためにマークが付けられます。

パラメーター:

xpathex-XPath 式

- **<text\ >.XPATH_JSON (xpathhex)**

  JSON ファイルを操作し、ブール値を返します。

  たとえば、次の JSON ファイルを考えてみます。

  { “Book” :{ “creator” :{ “person” :{ “name” :’ <name>’ }}, “title” :’ <title>’ }}

  次の式は JSON ファイルを操作し、JSON ファイルの最初の 1000 バイトに「creator」という名前の親ノードが「Book」である場合に、ブール値 TRUE を返します。

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON(xp%boolean(/Book/creator)%)
  ```

  パラメーター:

  xpathex-XPath ブーリアンエクスプレッション

- **<text>.XPATH_JSON(xpathex)**

  JSON ファイルを操作し、データ型「double」の値を返します。

  たとえば、次の JSON ファイルを考えてみます。

  { “Book” :{ “creator” :{ “person” :{ “name” :’ <name>’ }}, “title” :’ <title>’ , “price” :” 36” }}

  次の式は JSON ファイルを操作し、JSON ファイルの最初の 1000 バイトに文字列が存在する場合、文字列「36」をデータ型「double」の値に変換します。

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON(xp%number(/Book/price)%)
  ```

  パラメーター:

  xpathex-XPath 数値表現

- **<text>.XPATH_JSON(xpathex)**

  JSON ファイルを操作し、ノードセットまたは文字列を返します。ノードセットは、標準の XPath 文字列変換ルーチンを使用して対応する文字列に変換されます。

  たとえば、次の JSON ファイルを考えてみます。

  { “Book” :{ “creator” :{ “person” :{ “name” :’ <name>’ }}, “title” :’ <title>’ }}

<name\ ><title\ > 次の式は、JSON ファイル本体の最初の 1000 バイトで「/Book」(ノードセット) で囲まれているすべてのノードを選択し、対応する文字列値 (「」) を返します。

```
HTTP.REQ.BODY(1000).XPATH_JSON(xp%/Book%)
```

パラメーター:

xpathex-XPath 式

- **<text>.XPATH_JSON_WITH_MARKUP(xpathex)**

  XML ファイルを操作して、結果を囲む要素タグなどのマークアップを含む、結果ノードの文書全体を含む文字列を返します。

  たとえば、次の JSON ファイルを考えてみます。

  { “Book” :{ “creator” :{ “person” :{ “name” :’ <name>’ } }, “title” :’ <title>’ } }

  <name\ > 次の式は JSON ファイルに対して処理を行い、本文の最初の 1000 バイト、つまり「作成者:{person: {name: ‘\’ }}」で「/book/Creator」で囲まれているすべてのノードを選択します。

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON_WITH_MARKUP(xp%/Book/creator%)
  ```

  式によって選択された JSON 本文の部分には、今後の処理のためにマークが付けられます。

  パラメーター:

  xpathex-XPath 式

- **<text\ >\ .XPATH_WIT_MARKUP (xpathhex):**

  XML ファイルを操作して、結果を囲む要素タグなどのマークアップを含む、結果ノードの文書全体を含む文字列を返します。

  たとえば、次の式は XML ファイルを操作し、本文の最初の 1000 バイトで「/book/Creator」で囲まれたすべてのノードを選択します。

  ```
  HTTP.REQ.BODY(1000).XPATH_WITH_MARKUP(xp%/Book/creator%)
  ```

  式によって選択された JSON 本文の部分には、今後の処理のためにマークが付けられます。

  パラメーター:

  xpathex-XPath 式

XML ペイロードの暗号化と復号化

August 15, 2023

高度なポリシー式で XML_ENCRYPT () 関数と XML_DECRYPT () 関数を使用して、それぞれ XML データを暗号化および復号化できます。これらの関数は、”http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/ “で定

義されている W3C XML 暗号化標準に準拠しています。XML_ENCRYPT () と XML_DECRYPT () は、XML 暗号化仕様のサブセットをサポートしています。サブセットでは、データの暗号化はバルク暗号方式（RC4、DES3、AES128、AES192、または AES256）を使用し、バルク暗号キーの暗号化には RSA 公開キーが使用されます。

注: ペイロード内のテキストを暗号化および復号化する場合は、ENCRYPT 関数と DECRYPT 関数を使用する必要があります。これらの関数の詳細については、テキストの暗号化と復号化を参照してください。

XML_ENCRYPT () 関数と XML_DECRYPT () 関数は、テキストの ENCRYPT コマンドおよび DECRYPT コマンドで使用される暗号化/復号化サービスに依存しません。暗号方式は XML_ENCRYPT () 関数の引数として明示的に指定されます。この`XML_DECRYPT()`関数は、指定された暗号メソッドに関する情報を`<xenc:EncryptedData>`要素から取得します。次に、XML 暗号化および復号化関数の概要を示します。

- 暗号化された入力テキストと暗号化キーを含む`XML_ENCRYPT(<certKeyName>, <method>[, <flags>])**. Returns an <xenc:EncryptedData>`要素。暗号化キー自体は RSA を使用して暗号化されます。
- `XML_DECRYPT(<certKeyName>)`。input`<xenc:EncryptedData>`要素から復号化されたテキストを返します。これには、暗号方式と RSA 暗号化キーが含まれます。

注: この`<xenc:EncryptedData>`要素は、W3C XML 暗号化仕様で定義されています。

次に、引数の説明を示します。

- **certKeyName:** XML_ENCRYPT () 用の RSA 公開キーまたは XML_DECRYPT () 用の RSA プライベートキーを持つ X.509 証明書を選択します。証明書キーは、`add ssl certKey`コマンドによって事前に作成されている必要があります。
- **method:** XML データの暗号化に使用する暗号方式を指定します。可能な値: RC4、DES3、AES128、AES192、AES256。
- **flags :** `XML_ENCRYPT()`によって生成される`<xenc:EncryptedData>`要素に含める次のオプションのキー情報 (`<ds:KeyInfo>`) を指定するビットマスク。
  - **1** -certKeyName にキー名エレメントを含めます。要素は`<ds:KeyName>`です。
  - **2** -証明書の RSA 公開キーを持つ KeyValue 要素を含めます。要素は`<ds:KeyValue>`です。
  - **4** -証明書のシリアル番号と発行者 DN を含む X509IssuerSerial エレメントを含めます。要素は`<ds:X509IssuserSerial>`です。
  - **8** -X509SubjectName エレメントを証明書のサブジェクト DN に含めます。要素は`<ds:X509SubjectName>`です。
  - **16** -証明書全体に X509Certificate エレメントを含めます。要素は`<ds:X509Certificate>`です。

XML_ENCRYPT () 関数と XML_DECRYPT () 関数を式で使用する

XML 暗号化機能では、SSL 証明書とキーのペアを使用して、キー暗号化用の X.509 証明書（RSA 公開キー付き）、キー復号用の RSA プライベートキーを提供します。したがって、式で XML_ENCRYPT () 関数を使用する前に、SSL

証明書とキーのペアを作成する必要があります。次のコマンドは、X.509 証明書 my-cert.pem、およびプライベートキーファイル my-key.pem を持つ SSL 証明書とキーのペア my-certkey を作成します。

```
add ssl certKey my-certkey -cert my-cert.pem -key my-key.pem -passcrypt
 kxPeMRYnitY=
```

次の CLI コマンドは、XML コンテンツを暗号化および復号するための書き換えアクションとポリシーを作成します。

```
1  add rewrite action my-xml-encrypt-action replace "HTTP.RES.BODY(10000).
       XPATH_WITH_MARKUP(xp%/%)" "HTTP.RES.BODY(10000).XPATH_WITH_MARKUP(xp
       %/%).XML_ENCRYPT("my-certkey", AES256, 31)"
2
3  add rewrite action my-xml-decrypt-action replace "HTTP.REQ.BODY(10000).
       XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)" "HTTP.REQ.BODY(10000).
       XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%).XML_DECRYPT("my-certkey"
       )"
4
5  add rewrite policy my-xml-encrypt-policy "HTTP.REQ.URL.CONTAINS("xml-
       encrypt")" my-xml-encrypt-action
6
7  add rewrite policy my-xml-decrypt-policy "HTTP.REQ.BODY(10000).XPATH(xp
       %boolean(//xenc:EncryptedData)%)" my-xml-decrypt-action
8
9  bind rewrite global my-xml-encrypt-policy 30
10
11 bind rewrite global my-xml-decrypt-policy 30
12 <!--NeedCopy-->
```

上記の例では、書き換えアクション my-xml-encrypt-action は、AES-256 バルク暗号化方式と my-certkey の RSA 公開キーを使用してバルク暗号化キーを暗号化することにより、リクエスト内の XML ドキュメント全体 (XPATH_WITH_MARKUP (xp%/%)) を暗号化します。このアクションは、文書を、暗号化されたデータと暗号化されたキーを含む<xenc:EncryptedData>要素で置き換えます。31 で表されるフラグには、すべてのオプション<ds:KeyInfo>要素が含まれます。

アクション my-xml-decrypt-action は、レスポンスの最初の<xenc:EncryptedData>要素 (XPATH_WITH_MARKUP (XP%//XENC: encryptedData%)) を復号化します。これには、次の CLI コマンドを使用して xenc XML 名前空間を事前に追加する必要があります。

```
add ns xmlnamespace xenc http://www.w3.org/2001/04/xmlenc#
```

my-xml-decrypt-action アクションは、my-certkey 内の RSA 秘密キーを使用して暗号化されたキーを復号化し、要素で指定されたバルク暗号化方式を使用して暗号化されたコンテンツを復号化します。最後に、このアクションは、暗号化されたデータ要素を復号化されたコンテンツに置き換えます。

書き換えポリシー my-xml-encrypt-policy は、xml-encrypt を含む URL のリクエストに my-xml-encrypt-action を適用します。このアクションは、NetScaler アプライアンスで構成されたサービスからの応答全体を暗号化します。

書き換えポリシー my-xml-decrypt-policy は、<xenc:EncryptedData>要素（(XPATH (XP%//XENC:

encryptedData%) は空でない文字列を返す）を含む要求に my-xml-decrypt-action を適用します。このアクションは、NetScaler アプライアンスで構成されたサービス宛のリクエスト内の暗号化されたデータを復号化します。

高度なポリシー式:SSL の解析

August 15, 2023

SSL 証明書と SSL クライアント hello メッセージを解析するための高度なポリシー式があります。

SSL 証明書を解析する

X.509 Secure Sockets Layer (SSL) クライアント証明書を評価するには、高度なポリシー式を使用できます。クライアント証明書は、ユーザーの ID を認証するために使用できる電子文書です。クライアント証明書には、(少なくとも) バージョン情報、シリアル番号、署名アルゴリズム ID、発行者名、有効期間、サブジェクト (ユーザー) 名、公開鍵、および署名が含まれます。

SSL 接続とクライアント証明書内のデータの両方を調べることができます。たとえば、強度の低い暗号を使用する SSL 要求を特定の負荷分散仮想サーバーファームに送信できます。次のコマンドは、要求内の暗号強度を解析し、40 以下の暗号強度を照合するコンテンツスイッチングポリシーの例です。

```
add cs policy p1 -rule "client.ssl.cipher_bits.le(40)"
```

もう 1 つの例として、要求にクライアント証明書が含まれるかどうかを決定するポリシーを設定できます。

```
add cs policy p2 -rule "client.ssl.client_cert exists"
```

また、クライアント証明書内の特定の情報を検査するポリシーを構成することもできます。たとえば、次のポリシーは、証明書の有効期限が 1 日以上前であることを検証します。

```
add cs policy p2 -rule "client.ssl.client_cert exists && client.ssl.
client_cert.days_to_expire.ge(1)"
```

JA3 フィンガープリントの使用例:

```
add ssl policy ja3_pol -rule "CLIENT.SSL.JA3_FINGERPRINT.EQ(bb4c15a90e93a25ddc
)"-action reset
```

あるいは、patset での JA3 フィンガープリントの使用例:

```
1 add policy patset pat1
2 bind policy patset pat1 bb4c15a90e93a25ddc16274395bce4c6 -index 1
3 bind policy patset pat1 cd3c15a90e93a25ddc16274395bce6b4 -index 2
4 add ssl policy ssl_ja3_pol -rule CLIENT.SSL.JA3_FINGERPRINT.
     contains_any("pat1") -action reset
5 <!--NeedCopy-->
```


> 注
>
> 証明書の日付と時刻の解析については、「式内の日付と時刻のフォーマット」および 「SSL 証明書の日付の式」を参照してください。

テキストベースの **SSL** および証明書データのプレフィックス

次の表に、SSL トランザクションとクライアント証明書でテキストベースの項目を識別する式プレフィックスについて説明します。

表 1. SSL およびクライアント証明書データのテキストまたはブール値を返すプレフィックス

前	説明
CLIENT.SSL.CLIENT_CERT	現在の SSL トランザクションの SSL クライアント証明書を返します。
CLIENT.SSL.CLIENT_CERT.TO_PEM	SSL クライアント証明書をバイナリ形式で返します。
CLIENT.SSL.CIPHER_EXPORTABLE	SSL 暗号がエクスポート可能な場合は、ブール型 (Boolean) の TRUE を返します。
CLIENT.SSL.CIPHER_NAME	SSL 接続から呼び出された場合は SSL Cipher の名前を返し、非 SSL 接続から呼び出された場合は NULL 文字列を返します。
CLIENT.SSL.IS_SSL	現在の接続が SSL ベースの場合は、ブール型 (Boolean) の TRUE を返します。
CLIENT.SSL.JA3_FINGERPRINT	設定された JA3 フィンガープリントがクライアント hello メッセージの JA3 フィンガープリントと一致する場合、ブール型 TRUE を返します。注**:** この式は、リリース 13.1 ビルド 12.x 以降で使用できます。

SSL 証明書の数値データのプレフィックス

次の表では、SSL 証明書の日付以外の数値データを評価するプレフィックスについて説明します。これらのプレフィックスは、[式接頭辞の基本操作と数値の複合演算で説明されている操作で使用できま す](/ja-jp/citrix-adc/13-1/appexpert/policies-and-expressions/adv-policy-expressions-getting-started/compound-advanced-policy-expressions.html)。

表 2. SSL 証明書の日付以外の数値データを評価するプレフィックス

前	説明
CLIENT.SSL.CLIENT_CERT.DAYS_TO_EXPIRE	証明書が有効な日数を返します。期限切れの証明書の場合は -1 を返します。
CLIENT.SSL.CLIENT_CERT.PK_SIZE	証明書で使用されている公開鍵のサイズを返します。
CLIENT.SSL.CLIENT_CERT.VERSION	証明書のバージョン番号を返します。接続が SSL ベースでない場合は、ゼロ (0) を返します。
CLIENT.SSL.CIPHER_BITS	暗号鍵のビット数を返します。接続が SSL ベースでない場合は 0 を返します。
CLIENT.SSL.VERSION	SSL プロトコルのバージョンを表す数値 (0) を返します。トランザクションは SSL ベースではありません:0x002。トランザクションは SSLv2:0x300 です。トランザクションは SSLv3:0x301 です。トランザクションは TLSv1:0x302 です。トランザクションは TLS 1.1:0x303 です。トランザクションは TLS 1.2:0x304 です。トランザクションは TLS 1.3 です。

> 注
>
> 証明書の有効期限に関連する式については、「SSL 証明書日付の式」を参照してください。

SSL 証明書の式

SSL 証明書を解析するには、次のプレフィックスを使用する式を設定します。

CLIENT.SSL.CLIENT_CERT

ここでは、証明書に対して構成できる式について説明します。ただし、証明書の有効期限を調べる式は除きます。時間ベースの操作については、「高度なポリシー式: 日付、時刻、および数字の操作」を参照してください。

次の表に、CLIENT.SSL.CLIENT_CERT プレフィックスに指定できる操作を示します。

表 3. CLIENT.SSL.CLIENT_CERT プレフィックスで指定できる操作

SSL 証明書の操作	説明
`<certificate>.EXISTS`	クライアントが SSL 証明書を持っている場合は、ブール型 TRUE を返します。

SSL 証明書の操作	説明
<certificate>.ISSUER	証明書内の発行者の識別名 (DN) を名前と値のリストとして返します。等号 (「=」) は名前と値の区切り文字で、スラッシュ (「/」) は名前と値のペアを区切る区切り文字です。返される DN の例を以下に示します。/C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycompany.com
<certificate>.ISSUER.IGNORE_EMPTY_ELEMENTS	Issuer を返し、名前/値リスト内の空の要素は無視します。例えば, 次を考えてみましょう: Cert-Issuer: /c=in/st=kar//l=bangelore //o=mycompany/ou=sales/ /emailAddress=myuserid@mycompany.com。次の Rewrite アクションは、前の発行者の定義に基づいて 6 のカウントを返します。sh rewrite action insert_ssl_header Name: insert_ssl Operation: insert_http_header Target:Cert-Issuer Value:CLIENT.SSL.CLIENT_CERT.ISSUER.COUNT。ただし、値を次のように変更すると、返されるカウントは 9 です。CLIENT.SSL.CLIENT_CERT.ISSUER.IGNORE_EMPTY_ELEMENTS.COUNT
<certificate>. SERIALNUMBER	証明書のシリアル番号を、先頭にゼロを含まない大文字の 16 進文字列で返します。たとえば、証明書のシリアル番号が 04daa1e44bd2e7769638a0058b4964bd の場合、次の式はシリアル番号との照合に役立ちますCLIENT.SSL.CLIENT_CERT.SERIALNUMBER.SET_TEXT_MODE(IGNORECASE).CONTAINS(\"04daa1e44bd2e7769638a0058b4964bd\")

SSL クライアントを解析するこんにちは

SSL クライアントの hello メッセージを解析するには、次のプレフィックスを使用する式を設定します。

前	説明
CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXCODE	式で指定された 16 進コードを、クライアント hello メッセージで受け取った暗号スイートの 16 進コードと照合します。
CLIENT.SSL.CLIENT_HELLO.CLIENT_VERSION	クライアントの hello メッセージヘッダーで受け取ったバージョン。
CLIENT.SSL.CLIENT_HELLO.IS_RENEGOTIATE	クライアントまたはサーバーがセッション再ネゴシエーションを開始した場合は true を返します。
CLIENT.SSL.CLIENT_HELLO.IS_REUSE	client-hello メッセージで受信した 0 以外のセッション ID に基づいて、アプライアンスが SSL セッションを再利用する場合は true を返します。
CLIENT.SSL.CLIENT_HELLO.IS_SCSV	シグナリング暗号スイート値 (SCSV) 機能がクライアントの hello メッセージでアドバタイズされている場合は true を返します。フォールバック SCSV の 16 進コードは 0x5600 です。
CLIENT.SSL.CLIENT_HELLO.IS_SESSION_TICKET	長さが 0 以外のセッションチケット拡張が client-hello メッセージでアドバタイズされた場合、true を返します。
CLIENT.SSL.CLIENT_HELLO.LENGTH	クライアントの hello メッセージヘッダーで受信した長さ。
CLIENT.SSL.CLIENT_HELLO.SNI	クライアントの hello メッセージのサーバ名拡張で受け取ったサーバ名を返します。
CLIENT.SSL.CLIENT_HELLO.ALPN.HAS_NEXTPROTOCOL	クライアントの hello メッセージで受信した ALPN 拡張のアプリケーションプロトコルが、式で指定されたプロトコルと一致する場合に true を返します。

これらの式は CLIENTHELLO_REQ バインドポイントで使用できます。詳細については、SSL ポリシーバインディングを参照してください。

高度なポリシー式:IP アドレスと MAC アドレス、スループット、VLAN ID

August 15, 2023

IPv4 および IPv6 アドレス、MAC アドレス、IP サブネット、有用なクライアントおよびサーバデータ（インターフェイスポート（Rx、Tx、および RxTx）のスループットレート、およびパケットの受信に使用される VLAN の ID を返す高度なポリシー式プレフィクスを使用できます。その後、さまざまな演算子を使用して、これらの式プレフィックスによって返されるデータを評価できます。

IP アドレスと IP サブネットの式

高度なポリシー式を使用して、インターネットプロトコルバージョン 4 (IPv4) またはインターネットプロトコルバージョン 6 (IPv6) 形式のアドレスとサブネットを評価できます。IPv6 アドレスおよびサブネットの式プレフィックスには、プレフィックスに IPv6 が含まれます。IPv4 アドレスおよびサブネットの式プレフィックスには、プレフィックスに IP が含まれます。次に、要求が特定の IPv4 サブネットから発信されたかどうかを識別する式の例を示します。

```
client.ip.src.in_subnet(147.1.0.0/16)
<!--NeedCopy-->
```

次に、パケットの受信元のサブネットを調べ、ホストヘッダーで書き換えアクションを実行する書き換えポリシーの例を 2 つ示します。これら 2 つのポリシーが設定されている場合、実行される書き換えアクションは要求のサブネットによって異なります。これら 2 つのポリシーは、IPv4 アドレス形式の IP アドレスを評価します。

```
- add rewrite action URL1-rewrite-action replace "http.req.header("Host
    ")" ""www.mycompany1.com""
- add rewrite policy URL1-rewrite-policy "http.req.header("Host").
    contains("www.test1.com") && client.ip.src.in_subnet(147.1.0.0/16)"
    URL1-rewrite-action
- add rewrite action URL2-rewrite-action replace "http.req.header("Host
    ")" ""www.mycompany2.com""
- add rewrite policy URL2-rewrite-policy "http.req.header("Host").
    contains("www.test2.com") && client.ip.src.in_subnet(10.202.0.0/16)"
     URL2-rewrite-action
<!--NeedCopy-->
```

> 注
>
> 上記の例は、NetScaler ADC コマンドラインインターフェイス（CLI）で入力するコマンドです。したがって、各引用符の前にバックスラッシュ（\）を付ける必要があります。詳細については、「ポリシーでの高度なポリシー式の設定」を参照してください。

IPV4 アドレスおよび IP サブネットのプレフィクス

次の表では、IPv4 アドレス、サブネット、および IPv4 アドレスのセグメントを返すプレフィックスについて説明します。IPv4 アドレスに固有の数値演算子と演算子を、これらのプレフィックスとともに使用できます。数値演算の詳細については、「式接頭辞の基本操作」および「数値の複合演算」を参照してください。

表 1. IP アドレスと MAC アドレスを評価するプレフィクス

前	説明
CLIENT.IP.SRC	現在のパケットの送信元 IP を IP アドレスまたは数値で返します。

前	説明
CLIENT.IP.DST	現在のパケットの宛先 IP を IP アドレスまたは数値で返します。
SERVER.IP.SRC	現在のパケットの送信元 IP を IP アドレスまたは数値で返します。
SERVER.IP.DST	現在のパケットの宛先 IP を IP アドレスまたは数値で返します。

IPV4 アドレスの操作

IPV4 操作のプレフィクス表では、IPv4 アドレスを返すプレフィクスで使用できる演算子について説明します。

IPv6 式について

IPv6 アドレス形式では、古い IPv4 形式よりも柔軟性が高くなります。IPv6 アドレスは 16 進数形式です (RFC 2373)。次の例では、例 1 は IPv6 アドレス、例 2 は IPv6 アドレスを含む URL、例 3 には IPv6 アドレスとポート番号が含まれています。

例 **1**:

```
9901:0ab1:22a2:88a3:3333:4a4b:5555:6666
<!--NeedCopy-->
```

例 **2**:

```
http://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]/
<!--NeedCopy-->
```

例 **3:**

```
https://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]:8080/
<!--NeedCopy-->
```

例 3 では、IP アドレスとポート番号（8080）をブラケットで区切っています。

「+」演算子は、IPv6 式を他の式と組み合わせる場合にのみ使用できることに注意してください。出力は、個々の式から返される文字列値の連結です。IPv6 式では、他の算術演算子は使用できません。次の構文は例です。

```
client.ipv6.src + server.ip.dst
<!--NeedCopy-->
```

たとえば、クライアントの送信元 IPv6 アドレスが`ABCD:1234::ABCD`で、サーバの宛先 IPv4 アドレスが`10.100.10.100`の場合、前の式は`"ABCD:1234::ABCD10.100.10.100"`を返します。

NetScaler ADC アプライアンスは、IPv6 パケットを受信すると、未使用の IPv4 アドレス範囲から一時 IPv4 アドレスを割り当て、パケットの送信元アドレスをこの一時アドレスに変更します。応答時に、発信パケットの送信元アドレスは元の IPv6 アドレスに置き換えられます。

> 注
>
> IPv6 式は、ブール値の結果を生成する式以外の任意の式と組み合わせることができます。

IPv6 アドレスの式プレフィックス

次の表に示す式プレフィックスによって返される IPv6 アドレスは、テキストデータとして扱うことができます。たとえば、client.ipv6.dst というプレフィックスは、宛先 IPv6 アドレスをテキストとして評価できる文字列として返します。

次の表に、IPv6 アドレスを返す式プレフィクスを示します。

表 3. テキストを返す IPv6 式プレフィックス

前	説明
CLIENT.IPV6	現在のパケットでの IPv6 アドレスで動作します。
CLIENT.IPV6.DST	IP ヘッダーの宛先フィールドの IPv6 アドレスを返します。
CLIENT.IPV6.SRC	IP ヘッダーの送信元フィールドの IPv6 アドレスを返します。以下に例を示します。`client.ipv6.src.in_subnet(2007::2008/64)` `client.ipv6.src.get1.le(2008)`
SERVER.IPV6	現在のパケットでの IPv6 アドレスで動作します。
SERVER.IPV6.DST	IP ヘッダーの宛先フィールドの IPv6 アドレスを返します。
SERVER.IPV6.SRC	IP ヘッダーの送信元フィールドの IPv6 アドレスを返します。以下に例を示します。`server.ipv6.src.in_subnet(2007::2008/64)` `server.ipv6.src.get1.le(2008)`

IPv6 プレフィクスの操作

次の表に、IPv6 アドレスを返すプレフィックスで使用できる演算子を示します。

表 4. IPv6 アドレスを評価する操作

IPv6 オペレーション	説明
<ipv6>.EQ(<IPv6_address>	IP アドレスの値が<IPv6_address>引数と同じ場合は、ブール型 TRUE を返します。次に例を示します。 client.ipv6.dst.eq(ABCD:1234::ABCD)
<ipv6>.GET1. . .GET8	IPv6 アドレスのセグメントを数値で返します。次の式例では、IPv6 アドレス 1000:1001: CD 10:0000:0000:89 AB: 4567: CDEF: client.ipv6.dst.get5 extracts 0000からセグメントを取得します。これは、アドレスの 5 番目のビットセットです。client.ipv6.dst.get6 extracts 89AB. client.ipv6.dst.get7 extracts 4567.これらのセグメントに対して数値演算を実行できます。IPv6 アドレス全体を取得する場合、数値演算は実行できないことに注意してください。これは、CLIENT.IPV6.SRC など、IPv6 アドレス全体を返す式は、アドレスをテキスト形式で返すためです。
<ipv6>.IN_SUBNET(<subnet>)	IPv6 アドレス値が<subnet>引数で指定されたサブネットにある場合、ブール型 TRUE を返します。次に例を示します。 client.ipv6.dst.eq(1000:1001:CD10:0000:0000:89AB:4567:CDEF/60)
<ipv6>.IS_IPV4	これが IPv4 クライアントである場合はブール値 TRUE を返し、そうでない場合はブール値 FALSE を返します。
<ipv6>.SUBNET(<n>)	引数として指定されたサブネットマスクを適用した後に IPv6 アドレスを返します。サブネットマスクには 0 ～ 128 の値を指定できます。たとえば、次のようになります: CLIENT.IPV6.SRC.SUBNET(24)

MAC アドレスの式

MAC アドレスは ##: ##: ##: ##: ##: ##: ## の形式でコロンで区切られた 16 進数値で構成されます。各「#」は、0 ～9 の数字または A ～F の文字を表します。DAdvanced ポリシー式のプレフィクスおよび演算子は、送信元および宛先 MAC アドレスを評価する際に使用できます。

MAC アドレスのプレフィックス

次の表に、MAC アドレスを返すプレフィクスを示します。

表 5. MAC アドレスを評価するプレフィクス

前	説明
client.ether.dstmac	イーサネットヘッダーの宛先フィールドの MAC アドレスを返します。
client.ether.srcmac	イーサネットヘッダーの送信元フィールドの MAC アドレスを返します。

MAC アドレスの操作

次の表に、MAC アドレスを返すプレフィックスで使用できる演算子を示します。

表 6. MAC アドレスでの操作

前	説明
<mac address>.EQ(<address>)	MAC アドレス値が<address>引数と同じ場合は、ブール型 TRUE を返します。
<mac address>.GET1. . .GET4	GET 操作で指定された MAC アドレスのセグメントから抽出された数値を返します。たとえば、MAC アドレスが 12:34:56:78:9 a: bc の場合、次の例は 34 を返します。 client.ether.dstmac.get2

クライアントとサーバーの数値データの式

次の表に、スループット、ポート番号、VLAN ID など、クライアントおよびサーバの数値データを操作するためのプレフィクスを示します。

表 7. クライアントとサーバーの数値のデータを評価するプレフィックス

前	説明
interface.rxthroughput	直前 7 秒間の raw 受信トラフィックスループットを 1 秒あたりのキロバイト数 (KBps) で表す整数を返します。
client.interface.txthroughput	過去 7 秒間の raw 送信トラフィックスループットを Kbps 単位で表す整数を返します。

前	説明
client.interface.rxtxthroughput	直前 7 秒間の生の送受信トラフィックスループットを Kbps 単位で表す整数を返します。
server.interface.rxthroughput	過去 7 秒間の raw 受信トラフィックスループットを Kbps 単位で表す整数を返します。
server.interface.txthroughput	過去 7 秒間の raw 送信トラフィックスループットを Kbps 単位で表す整数を返します。
server.interface.rxtxthroughput	直前 7 秒間の生の送受信トラフィックスループットを Kbps 単位で表す整数を返します。
server.vlan.id	現在のパケットが NetScaler ADC に入った VLAN の数値 ID を返します。
client.vlan.id	現在のパケットが NetScaler ADC に入った VLAN の数値 ID を返します。

高度なポリシー表現**:** ストリーム分析関数

August 15, 2023

<identifier_name> ストリームアナリティクスの式は ANALYTICS.STREAM () プレフィックスで始まります。次のリストは、このプレフィックスで使用できる関数を示しています。

- **COLLECT_STATS**

  ポリシーに照らして評価されたリクエストから統計データを収集し、リクエストごとにレコードを作成します。

- **REQUESTS**

  指定されたレコードグループに存在するリクエストの数を返します。返される値は unsigned long 型です。

- **BANDWIDTH**

  指定されたレコードグループの帯域幅統計を返します。返される値は unsigned long 型です。

- **RESPTIME**

  指定されたレコードグループの応答時間統計を返します。返される値は unsigned long 型です。

- **CONNECTIONS**

  指定されたレコードグループに存在する同時接続の数を返します。返される値は unsigned long 型です。

- **IS_TOP (n)**

指定したレコードグループの統計値が上位 n 個のグループに含まれる場合は、ブール値 TRUE を返します。それ以外の場合は、ブール値 FALSE を返します。

- **CHECK_LIMIT**

  指定されたレコードグループの統計があらかじめ設定された制限に達した場合は、ブール値 TRUE を返します。それ以外の場合は、ブール値 FALSE を返します。

高度なポリシー表現:DataStream

August 15, 2023

NetScaler アプライアンスのポリシーインフラストラクチャには、アプライアンスをアプリケーションサーバーのファームとそれに関連するデータベースサーバーの間に展開する場合のデータベースサーバートラフィックの評価と処理に使用できる式が含まれています。

このセクションでは、以下のトピックについて説明します:

- MySQL プロトコルのエクスプレッション
- Microsoft SQL サーバー接続を評価するための表現

MySQL プロトコルのエクスプレッション

次の式は、MySQL データベースサーバーに関連するトラフィックを評価します。ポリシーではリクエストベースの表現（MYSQL.CLIENT と MYSQL.REQ で始まる式）を使用してコンテンツスイッチング仮想サーバーのバインドポイントでリクエストの切り替えを決定し、応答ベースの式（MYSQL.RES で始まる式）を使用してユーザー設定のヘルスモニターに対するサーバーの応答を評価できます。

- **MYSQL.CLIENT.** MySQL 接続のクライアントプロパティを操作します。
- **MYSQL.CLIENT.CAPABILITIES.** 認証中にクライアントがハンドシェイク初期化パケットのケイパビリティフィールドに設定したフラグのセットを返します。設定されているフラグの例としては、CLIENT_FOUND_ROWS、CLIENT_COMPRESS、CLIENT_SSL などがあります。
- **MYSQL.CLIENT.CHAR_SET.** クライアントが使用する文字セットに割り当てられた列挙定数を返します。<m> <m> このプレフィックスには、比較の結果を示すブール値を返す EQ () 演算子と NE () 演算子が使用されます。文字セット列挙定数は次のとおりです。

  - LATIN2_CZECH_CS
  - DEC8_SWEDISH_CI
  - CP850_GENERAL_CI
  - GREEK_GENERAL_CI

- LATIN1_GERMAN1_CI
- HP8_ENGLISH_CI
- KOI8R_GENERAL_CI
- LATIN1_SWEDISH_CI
- LATIN2_GENERAL_CI
- SWE7_SWEDISH_CI
- ASCII_GENERAL_CI
- CP1251_BULGARIAN_CI
- LATIN1_DANISH_CI
- HEBREW_GENERAL_CI
- LATIN7_ESTONIAN_CS
- LATIN2_HUNGARIAN_CI
- KOI8U_GENERAL_CI
- CP1251_UKRAINIAN_CI
- CP1250_GENERAL_CI
- LATIN2_CROATIAN_CI
- CP1257_LITHUANIAN_CI
- LATIN5_TURKISH_CI
- LATIN1_GERMAN2_CI
- ARMSCII8_GENERAL_CI
- UTF8_GENERAL_CI
- CP1250_CZECH_CS
- CP866_GENERAL_CI
- KEYBCS2_GENERAL_CI
- MACCE_GENERAL_CI
- MACROMAN_GENERAL_CI
- CP852_GENERAL_CI
- LATIN7_GENERAL_CI
- LATIN7_GENERAL_CS
- MACCE_BIN
- CP1250_CROATIAN_CI
- LATIN1_BIN
- LATIN1_GENERAL_CI
- LATIN1_GENERAL_CS
- CP1251_BIN
- CP1251_GENERAL_CI
- CP1251_GENERAL_CS
- MACROMAN_BIN
- CP1256_GENERAL_CI

- CP1257_BIN
- CP1257_GENERAL_CI
- ARMSCII8_BIN
- ASCII_BIN
- CP1250_BIN
- CP1256_BIN
- CP866_BIN
- DEC8_BIN
- GREEK_BIN
- HEBREW_BIN
- HP8_BIN
- KEYBCS2_BIN
- KOI8R_BIN
- KOI8U_BIN
- LATIN2_BIN
- LATIN5_BIN
- LATIN7_BIN
- CP850_BIN
- CP852_BIN
- SWE7_BIN
- UTF8_BIN
- GEOSTD8_GENERAL_CI
- GEOSTD8_BIN
- LATIN1_SPANISH_CI
- UTF8_UNICODE_CI
- UTF8_ICELANDIC_CI
- UTF8_LATVIAN_CI
- UTF8_ROMANIAN_CI
- UTF8_SLOVENIAN_CI
- UTF8_POLISH_CI
- UTF8_ESTONIAN_CI
- UTF8_SPANISH_CI
- UTF8_SWEDISH_CI
- UTF8_TURKISH_CI
- UTF8_CZECH_CI
- UTF8_DANISH_CI
- UTF8_LITHUANIAN_CI
- UTF8_SLOVAK_CI
- UTF8_SPANISH2_CI

  - UTF8_ROMAN_CI
  - UTF8_PERSIAN_CI
  - UTF8_ESPERANTO_CI
  - UTF8_HUNGARIAN_CI
  - INVAL_CHARSET

- **MYSQL.CLIENT.DATABASE.** クライアントがデータベースサーバーに送信する認証パケットで指定されたデータベースの名前を返します。これはデータベース名属性です。
- MYSQL.CLIENT.USER. クライアントがデータベースに接続しようとしているユーザー名 (認証パケット内) を返します。これはユーザー属性です。
- **MYSQL.REQ.** MySQL リクエストに対して操作します。
- **MYSQL.REQ.COMMAND.** リクエスト内のコマンドのタイプに割り当てられた列挙定数を識別します。<m> <m> このプレフィックスには、比較の結果を示すブール値を返す EQ () 演算子と NE () 演算子が使用されます。列挙定数の値は次のとおりです。

  - SLEEP
  - QUIT
  - INIT_DB
  - QUERY
  - FIELD_LIST
  - CREATE_DB
  - DROP_DB
  - REFRESH
  - SHUTDOWN
  - 統計情報
  - PROCESS_INFO
  - CONNECT
  - PROCESS_KILL
  - デバッグ
  - PING
  - 時間
  - DELAYED_INSERT
  - CHANGE_USER
  - BINLOG_DUMP
  - TABLE_DUMP
  - CONNECT_OUT
  - REGISTER_SLAVE
  - STMT_PREPARE
  - STMT_EXECUTE
  - STMT_SEND_LONG_DATA

  - STMT_CLOSE
  - STMT_RESET
  - SET_OPTION
  - STMT_FETCH

- **MYSQL.REQ.QUERY.** MySQL リクエスト内のクエリを識別します。
- **MYSQL.REQ.QUERY.COMMAND.** MySQL クエリの最初のキーワードを返します。
- **MYSQL.REQ.QUERY.SIZE.** リクエストクエリのサイズを整数形式で返します。SIZE メソッドは、HTTP リクエストまたはレスポンスの長さを返す CONTENT_LENGTH メソッドに似ています。
- **MYSQL.REQ.QUERY.TEXT.** クエリ全体をカバーする文字列を返します。
- **MYSQL.REQ.QUERY.TEXT(<n>).** MySQL クエリの最初の n バイトを文字列で返します。This is similar to HTTP.BODY(<n>).

  パラメーター:

  n-返されるバイト数

- **MYSQL.RES.** MySQL レスポンスに対して操作を行います。
- **MYSQL.RES.ATLEAST_ROWS_COUNT(<i>).** 応答の行数が i 行以上であるかどうかを確認し、結果を示すブール値 TRUE または FALSE を返します。

  パラメーター:

  i-行の数

- **MYSQL.RES.ERROR.** MySQL エラーオブジェクトを識別します。エラーオブジェクトには、エラー番号とエラーメッセージが含まれます。
- **MYSQL.RES.ERROR.MESSAGE.** サーバーのエラーレスポンスから取得したエラーメッセージを返します。
- **MYSQL.RES.ERROR.NUM.** サーバーのエラーレスポンスから取得したエラー番号を返します。
- **MYSQL.RES.ERROR.SQLSTATE.** サーバーのエラー応答の SQLSTATE フィールドの値を返します。MySQL サーバーは、エラー番号の値を SQLSTATE 値に変換します。
- **MYSQL.RES.FIELD(<i>).** i^{th に対応するパケットを識別します} サーバーの応答内の個々のフィールド。各フィールドパケットは、関連付けられたカラムのプロパティを記述します。パケットカウント (i) は 0 から始まります。

  パラメーター:

  i-パケット番号

- **MYSQL.RES.FIELD(<i>).CATALOG.** フィールドパケットのカタログプロパティを返します。
- **MYSQL.RES.FIELD(<i>).CHAR_SET.** 列の文字セットを返します。<m> <m> このプレフィックスには、比較の結果を示すブール値を返す EQ () 演算子と NE () 演算子が使用されます。

- **MYSQL.RES.FIELD(<i>).DATATYPE.** 列のデータ型を表す列挙定数を返します。これはカラムのタイプ (enum_field_type とも呼ばれる) 属性です。<m> <m> このプレフィックスには、比較の結果を示すブール値を返す EQ () 演算子と NE () 演算子が使用されます。さまざまなデータ型に指定できる値は次のとおりです。
  - 小数
  - 小さな
  - ショート
  - 長い
  - フロート
  - ダブル
  - NULL
  - タイムスタンプ
  - ロングロング
  - INT24
  - 日付
  - 時間
  - 日時
  - 年
  - 新しい日付
  - VARCHAR (MySQL 5.0 の新機能)
  - ビット (MySQL 5.0 の新機能)
  - 新しい十進数 (MySQL 5.0 の新機能)
  - 列挙型の
  - セット
  - TINY_BLOB
  - MEDIUM_BLOB
  - LONG_BLOB
  - BLOB
  - VAR_STRING
  - ストリング
  - ジオメトリ
- **MYSQL.RES.FIELD(<i>).DB.** フィールドパケットのデータベース識別子 (db) 属性を返します。
- **MYSQL.RES.FIELD(<i>).DECIMALS.** タイプが DECIMAL または NUMERIC の場合、小数点以下の位置の数を返します。これはフィールドパケットの小数点属性です。
- **MYSQL.RES.FIELD(<i>).FLAGS.** フィールドパケットの flags プロパティを返します。指定できる 16 進数のフラグ値は次のとおりです。
  - 0001: NULL フラグなし
  - 0002: PRI_KEY_FLAG

  - 0004: ユニークキーフラグ
  - 0008: マルチキーフラグ
  - 0010: ブロッブ・フラッグ
  - 0020: 署名なしフラグ
  - 0040: ゼロフィルフラグ
  - 0080: バイナリフラグ
  - 0100: 列挙型フラグ
  - 0200: オートインクリメント・フラグ
  - 0400: タイムスタンプ・フラグ
  - 0800: セット・フラグ

- **MYSQL.RES.FIELD(<i>).LENGTH.** 列の長さを返します。これはフィールドパケットの長さ属性の値です。返される値は、実際の値よりも大きい場合があります。たとえば、VARCHAR (2) 列のインスタンスは、1 文字しか含まれていない場合でも 2 の値を返す場合があります。
- **MYSQL.RES.FIELD(<i>).NAME.** 列識別子 (AS 句がある場合はそれに続く名前) を返します。これはフィールドパケットの名前属性です。
- **MYSQL.RES.FIELD(<i>).ORIGINAL_NAME.** 元の列識別子 (AS 句がある場合はその前) を返します。これはフィールドパケットの org_name 属性です。
- **MYSQL.RES.FIELD(<i>).ORIGINAL_TABLE.** カラムの元のテーブル識別子 (AS 句がある場合はその前) を返します。これはフィールドパケットの org_table 属性です。
- **MYSQL.RES.FIELD(<i>).TABLE.** 列のテーブル識別子 (AS 句がある場合はその後) を返します。これはフィールドパケットのテーブル属性です。
- **MYSQL.RES.FIELDS_COUNT.** レスポンスに含まれるフィールドパケットの数 (OK パケットの field_count 属性) を返します。
- **MYSQL.RES.OK.** データベースサーバーから送信された OK パケットを識別します。
- **MYSQL.RES.OK.AFFECTED_ROWS.** 挿入、更新、または DELETE クエリによって影響を受ける行の数を返します。これは OK パケットの affected_rows 属性の値です。
- **MYSQL.RES.OK.INSERT_ID.** OK パケットの unique_id 属性を識別します。現在の MySQL ステートメントまたはクエリで自動インクリメント ID が生成されない場合、unique_id の値、つまり式によって返される値は 0 になります。
- **MYSQL.RES.OK.MESSAGE.** OK パケットのメッセージプロパティを返します。
- **MYSQL.RES.OK.STATUS.** OK パケットの server_status 属性内のビット文字列を識別します。クライアントはサーバーステータスを使用して、現在のコマンドが実行中のトランザクションの一部であるかどうかを確認できます。server_status ビット文字列のビットは、次のフィールドに対応します (指定された順序で)。

  - トランザクション中
  - AUTO_COMMIT

- その他の結果
- マルチクエリ
- 不正なインデックスが使用されました
- インデックスは使用されていません
- カーソルが存在する
- 最後に表示した行
- データベースがドロップされました
- バックスラッシュエスケープなし

- **MYSQL.RES.OK.WARNING_COUNT.** OK パケットの warning_count 属性を返します。
- **MYSQL.RES.ROW(<i>).** i^{th に対応するパケットを識別します} データベースサーバーの応答内の個々の行。

  パラメーター:

  i-行番号

- **MYSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>).** j^{th 番目かどうかをチェックします} i^{th 番目のカラム} テーブルの行が NULL です。C の慣習に従い、インデックス i と j はどちらも 0 から始まります。^{したがって、行 i と列 j は実際には (i+1) 番目です} row and the (j+1)th 列、それぞれ。

  パラメーター:

  i-行番号

  j-列番号

- **MYSQL.RES.ROW(<i>).IS_NULL_ELEM(j).** j^{th 番目かどうかをチェックします} i^{th 番目のカラム} テーブルの行が NULL です。C の慣習に従い、インデックス i と j はどちらも 0 から始まります。^{したがって、行 i と列 j は実際には (i+1) 番目です} row and the (j+1)th 列、それぞれ。

  パラメーター:

  i-行番号

  j-列番号

- **MYSQL.RES.ROW(<i>).NUM_ELEM(<j>).** j^{th 番目の整数値を返します} i^{th 番目のカラム} テーブルの行。C の慣習に従い、インデックス i と j はどちらも 0 から始まります。^{したがって、行 i と列 j は実際には (i+1) 番目です} row and the (j+1)th 列、それぞれ。

  パラメーター:

  i-行番号

  j-列番号

- **MYSQL.RES.ROW(<i>).TEXT_ELEM(j).** j^{th 番目の文字列を返します} i^{th 番目のカラム} テーブルの行。C の慣習に従い、インデックス i と j はどちらも 0 から始まります。^{したがって、行 i と列 j は実際には (i+1) 番目です} row and the (j+1)th 列、それぞれ。

  パラメーター:

  i-行番号

  j-列番号

- **MYSQL.RES.TYPE.** レスポンスタイプの列挙定数を返します。その値には、エラー、OK、および RESULT_SET を指定できます。<m> <m> このプレフィックスには、比較の結果を示すブール値を返す EQ () 演算子と NE () 演算子が使用されます。

Microsoft SQL サーバー接続を評価するための表現

次の式は、Microsoft SQL Server データベースサーバーに関連するトラフィックを評価します。ポリシーではリクエストベースの表現（MSSQL.CLIENT と MSSQL.REQ で始まる式）を使用してコンテンツスイッチング仮想サーバーのバインドポイントでリクエストの切り替えを決定し、応答ベースの式（MSSQL.RES で始まる式）を使用してユーザー設定のヘルスモニターに対するサーバーの応答を評価できます。

式	説明
MSSQL.CLIENT.CAPABILITIES	Login7 認証パケットの OptionFlags1、OptionFlags2、OptionFlags3、TypeFlags の各フィールドを、この順序で 4 バイトの整数として返します。各フィールドは 1 バイトの長さで、クライアント機能のセットを指定します。
MSSQL.CLIENT.DATABASE	クライアントデータベースの名前を返します。返される値はテキスト型です。
MSSQL.CLIENT.USER	クライアントの認証に使用したユーザー名を返します。返される値はテキスト型です。
MSSQL.REQ.COMMAND	Microsoft SQL Server データベースサーバーに送信される要求内のコマンドのタイプを識別する列挙定数を返します。返される値はテキスト型です。列挙定数の値の例としては、クエリ、レスポンス、RPC、アテンションなどがあります。この式では、比較の結果を示すブール値を返す EQ () 演算子と NE () 演算子を使用します。
MSSQL.REQ.QUERY.COMMAND	SQL クエリの最初のキーワードを返します。返される値はテキスト型です。
MSSQL.REQ.QUERY.SIZE	リクエスト内の SQL クエリのサイズを返します。返される値は数値です。

式	説明
MSSQL.REQ.QUERY.TEXT	SQL クエリ全体を文字列として返します。返される値はテキスト型です。
MSSQL.REQ.QUERY.TEXT()	SQL クエリの最初の n バイトを返します。返される値はテキスト型です。パラメータ:n-バイト数
MSSQL.REQ.RPC.NAME	リモートプロシージャコール (RPC) リクエストで呼び出されているプロシージャの名前を返します。名前は文字列として返されます。
MSSQL.REQ.RPC.IS_PROCID	リモートプロシージャコール (RPC) リクエストにプロシージャ ID と RPC 名のどちらが含まれているかを示すブール値を返します。戻り値が True の場合はリクエストにプロシージャ ID が含まれていることを示し、FALSE の戻り値はリクエストに RPC 名が含まれていることを示します。
MSSQL.REQ.RPC.PROCID	リモートプロシージャコール (RPC) リクエストのプロシージャ ID を整数で返します。
MSSQL.REQ.RPC.BODY 注**:10.1** より前のリリースでは使用できません。	SQL リクエストの本文を、カンマで区切られた「a=b」句で表されるパラメータ形式の文字列として返します。ここで、「a」は RPC パラメータ名、「b」はその値です。
MSSQL.REQ.RPC.BODY (n) 注**:10.1** より前のリリースでは使用できません。	SQL リクエストの本文の一部を、カンマで区切られた「a=b」句で表されるパラメータ形式の文字列として返します。ここで、「a」は RPC パラメータ名、「b」はその値です。パラメータはリクエストの最初の「n」バイトからのみ返され、SQL ヘッダーはスキップされます。完全な名前と値のペアのみが返されます。
MSSQL.RES.ATLEAST_ROWS_COUNT(i)	レスポンスに i 行以上あるかどうかをチェックします。返される値は、ブール値 (TRUE) または FalseValue (偽値) です。パラメータ:i-行数
MSSQL.RES.DONE.ROWCOUNT	INSERT、UPDATE、または DELETE クエリによって影響を受けた行の数を返します。返される値は unsigned long 型です。
MSSQL.RES.DONE.STATUS	Microsoft SQL Server データベースサーバーから送信された DONE トークンからステータスフィールドを返します。返される値は数値です。
MSSQL.RES.ERROR.MESSAGE	Microsoft SQL Server データベースサーバーから送信された ERROR トークンからエラーメッセージを返します。これは ERROR トークンの MsgText フィールドの値です。返される値はテキスト型です。

式	説明
MSSQL.RES.ERROR.NUM	Microsoft SQL Server データベースサーバーから送信された ERROR トークンからエラー番号を返します。これは ERROR トークンの Number フィールドの値です。返される値は数値です。
MSSQL.RES.ERROR.STATE	Microsoft SQL Server データベースサーバーから送信された ERROR トークンからエラー状態を返します。これは ERROR トークンの State フィールドの値です。返される値は数値です。
MSSQL.RES.FIELD(*).DATATYPE*	サーバー応答の i 番目のフィールドのデータ型を返します。このプレフィックスには、比較の結果を示すブール値を返す EQ () 関数と NE () 関数を使用します。たとえば、次の式では、DATATYPE 関数が応答の 3 番目のフィールドの日時の値を返した場合、ブール値 TRUE が返されます。MSSQL.RES.FIELD (<2>) .DATATYPE.EQ (日時) パラメーター:i-行番号
MSSQL.RES.FIELD(*).LENGTH*	サーバー応答の i 番目のフィールドの最大長を返します。返される値は数値です。パラメータ:i-行番号
MSSQL.RES.FIELD(*).NAME*	サーバー応答の i 番目のフィールドの名前を返します。返される値はテキスト型です。パラメータ:i-行番号
MSSQL.RES.ROW(*).DOUBLE_ELEM()*	テーブルの i 番目の行の j 列目から double 型の値を返します。値が二重値でない場合、UNDEF 条件が発生します。C の慣習に従い、インデックス i と j はどちらも 0 (ゼロ) から始まります。したがって、行 i と列 j は実際には、それぞれ (i + 1) 番目の行と (j + 1) 番目の列です。パラメータ:i-行番号 j-列番号
MSSQL.RES.ROW(*).NUM_ELEM(j)*	テーブルの i 行目の j 列目から整数値を返します。値が整数値でない場合、UNDEF 条件が発生します。C の慣習に従い、インデックス i と j はどちらも 0 (ゼロ) から始まります。したがって、行 i と列 j は実際には、それぞれ (i + 1) 番目の行と (j + 1) 番目の列です。パラメータ:i-行番号 j-列番号
MSSQL.RES.ROW(*).IS_NULL_ELEM(j)*	テーブルの i 番目の行の j 列目が NULL かどうかをチェックし、結果を示すブール値 TRUE または FALSE を返します。C の慣習に従い、インデックス i と j はどちらも 0 (ゼロ) から始まります。したがって、行 i と列 j は実際には、それぞれ (i + 1) 番目の行と (j + 1) 番目の列です。パラメータ:i-行番号 j-列番号

式	説明
MSSQL.RES.ROW(*).TEXT_ELEM(j)*	テーブルの i 行目の j 列目からテキスト文字列を返します。C の慣習に従い、インデックス i と j はどちらも 0 (ゼロ) から始まります。したがって、行 i と列 j は実際には、それぞれ (i + 1) 番目の行と (j + 1) 番目の列です。パラメータ:i-行番号 j-列番号
MSSQL.RES.TYPE	応答タイプを識別する列挙定数を返します。返り値として指定できるのは、エラー、OK、および RESULT_SET です。この式では、比較の結果を示すブール値を返す EQ () 演算子と NE () 演算子を使用します。

タイプキャスティングデータ

August 15, 2023

リクエストとレスポンスからあるタイプ (テキストや整数など) のデータを抽出し、それを別のタイプのデータに変換できます。たとえば、文字列を抽出し、その文字列を時間形式に変換できます。また、HTTP リクエスト本文から文字列を抽出して HTTP ヘッダーのように扱うことも、あるタイプのリクエストヘッダーから値を抽出して別のタイプのレスポンスヘッダーに挿入することもできます。

データを型キャストしたら、新しいデータ型に適した任意の操作を適用できます。たとえば、HTTP ヘッダーにテキストをタイプキャストする場合、HTTP ヘッダーに適用可能な任意の操作を戻り値に適用できます。

型キャストデータの詳細については、「型キャスト操作」の PDF を参照してください。

正規表現

August 15, 2023

CONTAINS ("<string>") または EQ ("<string>") 演算子で実行する操作よりも複雑な文字列照合操作を実行する場合は、正規表現を使用します。Citrix® NetScaler® アプライアンスのポリシーインフラストラクチャには、テキスト照合の引数として正規表現を渡すことができる演算子が含まれています。正規表現を扱う演算子の名前には、文字列 REGEX が含まれます。引数として渡す正規表現は、"<http://www.pcre.org/pcre.txt>."で説明されている正規表現の構文に準拠している必要があります。正規表現の詳細については、"<http://www.regular-expressions.info/quickstart.html>"および"<http://www.silverstones.com/thebat/Regex.html>."を参照してください。

正規表現で機能する演算子のターゲットテキストは、テキストまたは HTTP ヘッダーの値のいずれかです。次に、正規表現演算子を使用してテキストを操作する高度なポリシー式の形式を示します。

```
<text>.<regex_operator>(re<delimiter><regex_pattern><delimiter>)
```

この文字列<text>は、パケット内のテキスト文字列（HTTP.REQ.URL など）を識別する高度なポリシー式のプレフィックスを表します。文字列<regex_operator>は正規表現演算子を表します。正規表現は常に文字列 re で始まります。<delimiter>で表される一対の一致する区切り文字は、正規表現を表す文字列<regex_pattern>を囲みます。

次の式例では、HTTP パケット内の URL に文字列 *.jpeg （* はワイルドカード）が含まれているかどうかを調べ、結果を示すブール値 TRUE または FALSE を返します。正規表現は、区切り文字として動作する一対のスラッシュマーク (/) で囲まれます。

```
http.req.url.regex_match(re/.<asterisk>\.jpeg/)
```

正規表現演算子を組み合わせて、検索の範囲を定義または絞り込むことができます。たとえば、<text>.AFTER_REGEX(reregex_pattern1).BEFORE_REGEX(reregex_pattern2)は文字列照合の対象が regex_pattern1 パターンと regex_pattern2 パターン間のテキストであることを指定します。正規表現演算子で定義されたスコープでは、テキスト演算子を使用できます。たとえば、CONTAINS ("<string>") 演算子を使用して、定義されたスコープに文字列 abc が含まれているかどうかを確認できます。

```
<text>.AFTER_REGEX(re/regex_pattern1).BEFORE_REGEX(re/regex_pattern2
/).CONTAINS("abc")
```

> 注
>
> 正規表現を評価するプロセスは、本質的に単純な文字列引数で動作する CONTAINS ("<string>") や EQ ("<string>") などの演算子の場合よりも時間がかかります。正規表現は、要件が他の演算子の範囲を超えている場合にのみ使用してください。

正規表現の基本的な特徴

August 15, 2023

NetScaler アプライアンスで定義されている正規表現の注目すべき特徴は次のとおりです。

- 正規表現は必ず「re」という文字列で始まり、その後に使用する正規表現を囲む区切り文字 (区切り文字と呼ばれる) の 2 組が続きます。

たとえば、re# # <regex_pattern> は番号記号 (#) を区切り文字として使用します。

- 正規表現は 1499 文字を超えることはできません。
- 数字の照合は、文字列\ d (バックスラッシュの後に d が続く) を使用して実行できます。

- 空白は\ s (バックスラッシュの後に s が続く) で表すことができます。
- 正規表現には空白を含めることができます。

NetScaler 構文と PCRE 構文の違いは次のとおりです。

- NetScaler では、正規表現でのバックリファレンスは許可されていません。
- 再帰的な正規表現は使用しないでください。
- ドットのメタ文字も改行文字と一致します。
- Unicode はサポートされていません。
- 操作 SET_TEXT_MODE (IGNORECASE) は、(? i）正規表現の内部オプション。

正規表現の演算

August 16, 2023

次の表では、正規表現を操作する演算子について説明します。特定の高度なポリシー式で正規表現演算子によって実行される操作は、式のプレフィックスがテキストヘッダーまたは HTTP ヘッダーを識別するかどうかによって異なります。ヘッダーを評価する操作は、指定されたヘッダータイプのすべてのインスタンスのテキストベースの操作をオーバーライドします。演算子を使用する場合は、<text> をテキストを識別するために構成する高度なポリシー式のプレフィックスに置き換えます。

正規表現操作	説明
.BEFORE_REGEX()	引数に一致する文字列の前にあるテキストを選択します。正規表現がターゲットのどのデータとも一致しな
.AFTER_REGEX()	引数に一致する文字列に続くテキストを選択します。正規表現がターゲットのどのテキストとも一致しない
.REGEX_SELECT()	引数に一致する文字列を選択します。正規表現がターゲットと一致しない場合、長さ 0 のテキストオブジェ
.REGEX_MATCH()	ターゲットが最大 1499 文字の\ 引数に一致する場合は TRUE を返します。正規表現は次の形式でなければ

高度なポリシー式とポリシーの概要例

August 15, 2023

次の表に、独自の高度なポリシー式の基礎として使用できる高度なポリシー式の例を示します。

表 1. 高度なポリシー式の例

エクスプレッションタイプ	サンプル式
HTTP リクエストで使用されているメソッドを見てください。	`http.req.method.eq(post)http.req.method.eq(get)`
HTTP 要求（req）または応答（res）のキャッシュコントロールまたはプラグマヘッダ値を確認します。	`http.req.header("Cache-Control").contains("no-store") http.req.header("Cache-Control").contains("no-cache") http.req.header("Pragma").contains("no-cache") http.res.header("Cache-Control").contains("private") http.res.header("Cache-Control").contains("public") http.res.header("Cache-Control").contains("must-revalidate")http.res.header("Cache-Control").contains ("proxy-revalidate") http.res.header("Cache-Control").contains("max-age")`
要求 (req) または応答 (res) にヘッダーが存在するかどうかを確認します。	`http.req.header("myHeader").exists http.res.header("myHeader").exists`
ファイル拡張子に基づいて HTTP リクエストで特定のファイルタイプを探します。	`http.req.url.contains(".html") http.req.url.contains(".cgi")http.req.url.contains(".asp")http.req.url.contains(".exe")http.req.url.contains(".cfm")http.req.url.contains(".ex")http.req.url.contains(".shtml")http.req.url.contains(".htx")http.req.url.contains("/cgi-bin/")http.req.url.contains("/exec/")http.req.url.contains("/bin/")`
HTTP リクエストで特定のファイルタイプ以外のものを探します。	`http.req.url.contains(".png").not ; http.req.url.contains(".jpeg").not`

エクスプレッションタイプ	サンプル式
Content-Type ヘッダーに基づいて、HTTP 応答で送信されるファイルのタイプを確認します。	`http.res.header("Content-Type").contains("text")http.res.header("Content-Type").contains "application/msword")http.res.header("Content-Type").contains("vnd.ms-excel")http.res.header("Content-Type").contains("application/vnd.ms-powerpoint"); http.res.header("Content-Type").contains("text/css"); http.res.header("Content-Type").contains("text/xml"); http.res.header("Content-Type").contains("image/")`
この応答に有効期限ヘッダーが含まれているかどうかを確認します。	`http.res.header("Expires").exists`
レスポンスで Set-Cookie ヘッダーを確認します。	`http.res.header("Set-Cookie").exists`
応答を送信したエージェントを確認します。	`http.res.header("User-Agent").contains("Mozilla/4.7")http.res.header("User-Agent").contains("MSIE")`
リクエストの本文の最初の 1024 バイトが文字列「some text」で始まるかどうかを確認します。	`http.req.body(1024).contains("some text")`

次の表に、一般的に使用される関数のポリシー設定とバインディングの例を示します。

表 2. 高度なポリシー式とポリシーの例

目的	例
書き換え機能を使用して、HTTP レスポンスの本文のhttp://with https://のオカランスを置き換えます。	add rewrite action httpRewriteAction replace_all http.res.body(50000)"\"https://\""-search http://add rewrite policy demo_rep34312 "http.res.body(50000).contains(\"http://\")" httpRewriteAction
HTTP 本文の最初の 1000 バイトの「abcd」のすべての出現箇所を”1234”に置き換えます。	add rewrite action abcdTo1234Action replace_all "http.req.body(1000)""\"1234\""-search abcd add rewrite policy abcdTo1234Policy "http.req.body(1000).contains(\"abcd\")" abcdTo1234Action bind rewrite global abcdTo1234Policy 100 END -type REQ_OVERRIDE
HTTP バージョンを 1.0 にダウングレードして、サーバーが HTTP 応答をチャンクしないようにします。	add rewrite action downgradeTo1.0Action replace http.req.version.minor "\"0\""add rewrite policy downgradeTo1.0Policy "http.req.version.minor.eq(1)"downgradeTo1.0Action bind lb vserver myLBVserver -policyName downgradeTo1.0Policy -priority 100 -gotoPriorityExpression NEXT -type REQUEST

目的	例
すべての応答で HTTP または HTTPS プロトコルへの参照を削除します。これにより、ユーザーの接続が HTTP の場合はリンクが HTTP を使用して開かれ、ユーザーの接続が HTTPS の場合は HTTPS を使用してリンクが開かれます。	add rewrite action remove_http_https replace_all " http.res.body(1000000). set_text_mode(ignorecase)""\"//\" "-search "re~https?://|HTTPS?://~ "add rewrite policy remove_http_https **true** remove_http_https bind lb vserver test_vsvr -policyName remove_http_https -priority 20 -gotoPriorityExpression NEXT -type RESPONSE
すべての URL の http: のインスタンスを https: に書き換えます。	add responder action httpToHttpsAction redirect "\" https://\" + http.req.hostname + http.req.url"add responder policy httpToHttpsPolicy "!CLIENT.SSL. IS_SSL"httpToHttpsAction bind responder global httpToHttpsPolicy 1 END -type OVERRIDE
URL A から URL B にリダイレクトするように URL を変更します。この例では、パスに「file5.html」が追加されます。	add responder action appendFile5Action redirect \"http ://\" + http.req.hostname + http. req.url + \"/file5.html\""add responder policy appendFile5Policy "http.req.url. eq(\"/testsite\")" appendFile5Action bind responder global appendFile5Policy 1 END - type OVERRIDE

目的	例
外部 URL を内部 URL にリダイレクトします。	`add rewrite action act_external_to_internal REPLACE 'http.req.hostname.server'"www.my.host.com"'add rewrite policy pol_external_to_internal 'http.req.hostname.server.eq("www.external.host.com")' act_external_to_internal bind rewrite global pol_external_to_internal 100 END -type REQ_OVERRIDE`
クエリ文字列を持つリクエストを www.webn.example.com にリダイレクトします。値 n は、クエリ文字列のサーバーパラメータ (server=5 など) から派生します。	`add rewrite action act_redirect_query REPLACE q#http.req.header("Host").before_str(".example.com")'"Web"+ http.req.url.query.value("server")# add rewrite policy pol_redirect_query q#http.req.header("Host").eq("www.example.com")&& http.req.url.contains("?")'act_redirect_query#`

目的	例
URL からの 1 秒あたりのリクエスト数を制限します。	`add ns limitSelector ip_limit_selector http.req.url "client.ip.src"add ns limitIdentifier ip_limit_identifier -threshold 4 -timeSlice 3600 -mode request_rate -limitType smooth -selectorName ip_limit_selector add responder action my_Web_site_redirect_action redirect "\"http://www.mycompany.com/\""add responder policy ip_limit_responder_policy "http.req.url.contains(\"myasp.asp\")&& sys.check_limit (\"ip_limit_identifier\")" my_Web_site_redirect_action bind responder global ip_limit_responder_policy 100 END -type default`
クライアントの IP アドレスを確認しますが、要求を変更せずに要求を渡します。	`add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER ("x-forwarded-for").EXISTS HTTP.REQ.HEADER ("client-ip").EXISTS'NOREWRITE bind rewrite global check_client_ip_policy 100 END`

目的	例
リクエストから古いヘッダーを削除し、NS-Client ヘッダーを挿入します。	`add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS'del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC'add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS HTTP.REQ.HEADER("client-ip").EXISTS'insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END`
要求から古いヘッダーを削除し、NS-Client ヘッダーを挿入し、挿入されたヘッダーの値に古いヘッダーのクライアント IP 値と NetScaler ADC アプライアンスの接続 IP アドレスが含まれるように、「ヘッダーの挿入」アクションを変更します。この例では、最後のセット書き換えアクションを除いて、前の例を繰り返していることに注意してください。	高度なポリシー式とポリシーの概要例

次の表に、独自の高度なポリシー式の基礎として使用できる高度なポリシー式の例を示します。

表 1. 高度なポリシー式の例

エクスプレッションタイプ	サンプル式
HTTP リクエストで使用されているメソッドを見てください。	<!JEKYLL@5180@0>
HTTP 要求（req）または応答（res）のキャッシュコントロールまたはプラグマヘッダ値を確認します。	<!JEKYLL@5180@1> <!JEKYLL@5180@2> <!JEKYLL@5180@3> <!JEKYLL@5180@4> <!JEKYLL@5180@5> <!JEKYLL@5180@6> <!JEKYLL@5180@7>
要求 (req) または応答 (res) にヘッダーが存在するかどうかを確認します。	<!JEKYLL@5180@8>
ファイル拡張子に基づいて HTTP リクエストで特定のファイルタイプを探します。	<!JEKYLL@5180@9>
HTTP リクエストで特定のファイルタイプ以外のものを探します。	<!JEKYLL@5180@10>
Content-Type ヘッダーに基づいて、HTTP 応答で送信されるファイルのタイプを確認します。	<!JEKYLL@5180@11>
この応答に有効期限ヘッダーが含まれているかどうかを確認します。	<!JEKYLL@5180@12>
レスポンスで Set-Cookie ヘッダーを確認します。	<!JEKYLL@5180@13>
応答を送信したエージェントを確認します。	<!JEKYLL@5180@14>
リクエストの本文の最初の 1024 バイトが文字列「some text」で始まるかどうかを確認します。	<!JEKYLL@5180@15>

次の表に、一般的に使用される関数のポリシー設定とバインディングの例を示します。

表 2. 高度なポリシー式とポリシーの例

目的	例
書き換え機能を使用して、HTTP レスポンスの本文の <!JEKYLL@5180@16> のオカランスを置き換えます。	<!JEKYLL@5180@17>
HTTP 本文の最初の 1000 バイトの「abcd」のすべての出現箇所を”1234”に置き換えます。	<!JEKYLL@5180@18>
HTTP バージョンを 1.0 にダウングレードして、サーバーが HTTP 応答をチャンクしないようにします。	<!JEKYLL@5180@19>
すべての応答で HTTP または HTTPS プロトコルへの参照を削除します。これにより、ユーザーの接続が HTTP の場合はリンクが HTTP を使用して開かれ、ユーザーの接続が HTTPS の場合は HTTPS を使用してリンクが開かれます。	<!JEKYLL@5180@20>

目的	例
すべての URL の http: のインスタンスを https: に書き換えます。	<!JEKYLL@5180@21>
URL A から URL B にリダイレクトするように URL を変更します。この例では、パスに「file5.html」が追加されます。	<!JEKYLL@5180@22>
外部 URL を内部 URL にリダイレクトします。	<!JEKYLL@5180@23>
クエリ文字列を持つリクエストを www.webn.example.com にリダイレクトします。値 n は、クエリ文字列のサーバーパラメータ (server=5 など) から派生します。	<!JEKYLL@5180@24>
URL からの 1 秒あたりのリクエスト数を制限します。	<!JEKYLL@5180@25>
クライアントの IP アドレスを確認しますが、要求を変更せずに要求を渡します。	<!JEKYLL@5180@26>
リクエストから古いヘッダーを削除し、NS-Client ヘッダーを挿入します。	<!JEKYLL@5180@27>

目的	例
	add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS' del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC' add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS HTTP.REQ.HEADER("client-ip").EXISTS' insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END set rewrite action insert_ns_client_header -stringBuilderExpr 'HTTP.REQ.HEADER("x-forwarded-for").VALUE(0) + ""+ HTTP.REQ.HEADER("client-ip").VALUE(0) + ""+ CLIENT.IP.SRC'

書き換え用の高度なポリシーポリシーのチュートリアル例

August 15, 2023

書き換え機能を使用すると、HTTP ヘッダーの任意の部分を変更できます。また、応答の場合は HTTP 本文を変更できます。この機能を使用すると、不要な HTTP ヘッダーの削除、内部 URL のマスキング、Web ページのリダイレクト、クエリやキーワードのリダイレクトなど、いくつかの便利なタスクを実行できます。

次の例では、まず書き換えアクションと書き換えポリシーを作成します。次に、ポリシーをグローバルにバインドします。

このドキュメントでは、次の詳細について説明します。

- 外部 URL を内部 URL にリダイレクトする
- クエリのリダイレクト
- HTTP から HTTPS への書き換え
- 不要なヘッダーの削除
- Web サーバーのリダイレクトを減らす
- サーバヘッダーのマスキング
- プレーンテキストを URL エンコードされた文字列に変換し、反対の方法

コマンドおよび構文の説明の詳細については、Rewrite コマンドリファレンスページを参照してください 。

外部 **URL** を内部 **URL** にリダイレクトする

この例では、外部 URL を内部 URL にリダイレクトする書き換えアクションと書き換えポリシーを作成する方法について説明します。書き換えを実行する act_external_to_internal というアクションを作成します。次に、pol_external_to_internal というポリシーを作成します。

CLI を使用して外部 **URL** を内部 **URL** にリダイレクトするには

- 書き換えアクションを作成するには、コマンドプロンプトで次のように入力します。

```
add rewrite action act_external_to_internal REPLACE “http.req.hostname
.server” “\” host_name_of_internal_Web_server\” ”
```

- 書き換えポリシーを作成するには、NetScaler ADC コマンドプロンプトで次のように入力します。

```
add rewrite policy pol_external_to_internal "http.req.hostname.server
.eq(\"host_name_of_external_Web_server\")"act_external_to_internal
```

- ポリシーをグローバルにバインドします。

構成ユーティリティを使用して外部 **URL** を内部 **URL** にリダイレクトするには

1. **[**AppExpert**] > [書き換え] > [アクション] に移動します。**
2. 詳細ペインで、[追加] をクリックします。
3. [書き換えアクションの作成] ダイアログボックスで、act_external_to_internal という名前を入力します。
4. HTTP サーバのホスト名を内部サーバ名に置き換えるには、[タイプ（Type）] リストボックスから [置換（Replace）] を選択します。
5. ヘッダー名フィールドに「**Host**」と入力します。
6. 置換テキストフィールドの文字列式に、Web サーバーの内部ホスト名を入力します。
7. [作成] をクリックし、[閉じる] をクリックします。
8. ナビゲーションペインで、[**Policies**] をクリックします。

9. 詳細ペインで、[追加] をクリックします。
10. 「名前」フィールドに「pol_external_to_internal」と入力します。このポリシーは、Web サーバーへの接続を検出します。
11. [アクション] ドロップダウンメニューで、アクション act_external_to_internal を選択します。
12. エクスプレッションエディタで、次のエクスプレッションを作成します。

```
HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")
<!--NeedCopy-->
```

1. 新しいポリシーをグローバルにバインドします。

クエリのリダイレクト

この例では、クエリを適切な URL にリダイレクトする書き換えアクションと書き換えポリシーを作成する方法について説明します。この例では、リクエストに**www.example.com**に設定された Host ヘッダーと **string /query.cgi?server=5** という文字列の GET メソッドが含まれていることを前提としています。リダイレクトは、ホストヘッダーからドメイン名を、クエリ文字列から番号を抽出し、ユーザーのクエリをサーバー **Web5.example.com** にリダイレクトします。**Web5.example.com** では、残りのユーザーのクエリが処理されます。

> 注:
>
> 次のコマンドは複数の行に表示されますが、改行せずに 1 行で入力する必要があります。

CLI を使用してクエリを適切な **URL** にリダイレクトするには

- HTTP サーバのホスト名を内部サーバ名に置き換える act_redirect_query という名前の書き換えアクションを作成するには、次のように入力します。

```
add rewrite action act_redirect_query REPLACE http.req.header("Host")
.before_str(".example.com")'"Web" + http.req.url.query.value("server
")'
```

- pol_redirect_query という名前の書き換えポリシーを作成するには、NetScaler ADC コマンドプロンプトで次のコマンドを入力します。このポリシーは、クエリ文字列を含む Web サーバーへの接続を検出します。このポリシーは、クエリ文字列を含まない接続には適用しないでください。

```
add rewrite policy pol_redirect_query 'http.req.header("Host").eq(www
.example.com)&& http.req.url.contains("?")'act_redirect_query
```

- 新しいポリシーをグローバルにバインドします。

この書き換えポリシーは、非常に特殊なものであり、他の書き換えポリシーの前に実行する必要があるため、高い優先度を割り当てることをお勧めします。優先度 1 を割り当てると、最初に評価されます。

HTTP から **HTTPS** への書き換え

この例では、文字列「HTTP」で始まるすべての URL を検索し、その文字列を「https」に置き換えるように Web サーバーレスポンスを書き換える方法について説明します。サーバーを HTTP から HTTPS に移動した後に Web ページを更新する必要がないようにするために使用できます。

CLI を使用して **HTTP URL** を **HTTPS** にリダイレクトするには

- 文字列「HTTP」のすべてのインスタンスを文字列「https」に置き換える act_replace_http_with_https という名前の書き換えアクションを作成するには、次のコマンドを入力します。

```
add rewrite action act_replace_http_with_https replace_all 'http.res.
body(100)'"https"'-search text("http")
```

- Web サーバへの接続を検出する pol_replace_http_with_https という名前の書き換えポリシーを作成するには、次のコマンドを入力します。

```
add rewrite policy pol_replace_http_with_https TRUE act_replace_http_with_http
 NOREWRITE
```

- 新しいポリシーをグローバルにバインドします。

この書き換え操作のトラブルシューティングについては、「ケーススタディ:HTTP リンクを HTTPS に変換するための書き換えポリシーが機能しない」を参照してください。

不要なヘッダーの削除

この例では、書き換えポリシーを使用して不要なヘッダーを削除する方法について説明します。具体的には、次のヘッダーを削除する例を示します。

- エンコーディングヘッダーを受け入れます。HTTP 応答から Accept Encoding ヘッダーを削除すると、応答の圧縮が防止されます。
- コンテンツの場所ヘッダー。HTTP レスポンスから Content Location ヘッダーを削除すると、サーバーがセキュリティ侵害を許容する可能性のある情報をハッカーに提供することを防ぎます。

HTTP 応答からヘッダーを削除するには、書き換えアクションと書き換えポリシーを作成し、ポリシーをグローバルにバインドします。

CLI を使用して適切な書き換えアクションを作成するには

コマンドプロンプトで、次のいずれかのコマンドを入力して、Accept Encoding ヘッダーを削除して応答の圧縮を防止するか、Content Location ヘッダーを削除します。

- `add rewrite action "act_remove-ae"delete_http_header "Accept-Encoding"`
- `add rewrite action "act_remove-cl"delete_http_header "Content-Location"`

CLI を使用して適切な書き換えポリシーを作成するには

コマンドプロンプトで、次のいずれかのコマンドを入力して、Accept Encoding ヘッダーまたは Content Location ヘッダーを削除します。

- `add rewrite policy "pol_remove-ae"true "act_remove-ae"`
- `add rewrite policy "pol_remove-cl"true "act_remove-cl"`

CLI を使用してポリシーをグローバルにバインドするには

コマンドプロンプトで、必要に応じて次のコマンドのいずれかを入力し、作成したポリシーをグローバルにバインドします。

- `bind rewrite global pol_remove_ae 100`
- `bind rewrite global pol_remove_cl 200`

Web サーバーのリダイレクトを減らす

この例では、書き換えポリシーを使用して、ホームページへの接続と、末尾がスラッシュ (/) で終わる他の URL をサーバーの既定のインデックスページに変更し、リダイレクトを防止し、サーバーの負荷を軽減する方法について説明します。

CLI を使用してデフォルトのホームページを含めるようにディレクトリレベルの **HTTP** 要求を変更するには

- スラッシュで終わる URL を変更して、デフォルトのホームページ index.html を含むように変更する action-default-homepage という名前の書き換えアクションを作成するには、次のように入力します。

```
add rewrite action "action-default-homepage"replace http.req.url.path
 "\"/index.html\""
```

- ホームページへの接続を検出して新しいアクションを適用する policy-default-homepage という名前の書き換えポリシーを作成するには、次のように入力します。

```
add rewrite policy "policy-default-homepage"q\\#http.req.url.path.EQ(
"/")"action-default-homepage"\\#
```

- 新しいポリシーをグローバルにバインドして有効にします。

サーバヘッダーのマスキング

この例では、書き換えポリシーを使用して、Web サーバーからの HTTP 応答の Server ヘッダー内の情報をマスクする方法について説明します。このヘッダーには、ハッカーがウェブサイトを侵害するために使用できる情報が含まれています。ヘッダーをマスクしても、熟練したハッカーがサーバーに関する情報を見つけることを妨げることはありませんが、Web サーバーのハッキングが難しくなり、ハッカーが保護されていないターゲットを選択するように促します。

CLI からの応答で **Server** ヘッダーをマスクするには

1. Server ヘッダーの内容を無意味な文字列に置き換える act_mask-server という名前の書き換えアクションを作成するには、次のように入力します。

```
add rewrite action "act_mask-server"replace "http.RES.HEADER(\"Server\")""\"Web Server 1.0\""
```

1. すべての接続を検出する pol_mask-server という名前の書き換えポリシーを作成するには、次のように入力します。

```
add rewrite policy "pol_mask-server"true "act_mask-server"
```

1. 新しいポリシーをグローバルにバインドして有効にします。

プレーンテキストを **URL** エンコードされた文字列に変換する方法と反対の方法

次の式は、プレーンテキストを URL エンコードされた文字列に変換し、反対の方法で変換します。

1. URL_RESERVED_CHARS_SAFE（URL エンコードする文字列）。

例**:**

```
("abc def&123").URL_RESERVED_CHARS_SAFE
Output will be
 “abc%20def%26123”  which is url encoded.
<!--NeedCopy-->
```

1. SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE. (URL を文字列にエンコード)

例**:**

```
("abc%20def%26123").SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE
Output will be
 “abc def&123”
<!--NeedCopy-->
```


リライトとレスポンダーポリシーの例

August 15, 2023

書き換えポリシーとレスポンダーポリシーの例をいくつか示します。

例 **1:** コマンドラインインターフェイスを使用してローカル **Client-IP** ヘッダーを追加するには

```
add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.IP.SRC'
add rewrite policy pol_ins_client http.req.is_valid act_ins_client
bind rewrite global pol_ins_client 300 END

namem@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
< HTTP/1.1 200 OK
< Date: Tue, 10 Nov 2020 10:06:48 GMT
* Server Apache/2.2.15 (CentOS) is not blacklisted
< Server: Apache/2.2.15 (CentOS)
< Last-Modified: Thu, 20 Jun 2019 07:16:04 GMT
< ETag: "816c5-5-58bbc1e73cdd3"
< Accept-Ranges: bytes
< Content-Length: 5
< Content-Type: text/html; charset=UTF-8
< NS-Client: 10.102.1.98
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_namem@obelix:~$

<!--NeedCopy-->
```

例 **2: HTTP** サーバータイプをマスクする

```
add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("Server") ""Web Server 1.0""
add rewrite policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
namem@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
```



```
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
< HTTP/1.1 200 OK
< Date: Tue, 10 Nov 2020 10:15:42 GMT
* Server Web Server 1.0 is not blacklisted
< Server: Web Server 1.0
< Last-Modified: Thu, 20 Jun 2019 07:16:04 GMT
< ETag: "816c5-5-58bbc1e73cdd3"
< Accept-Ranges: bytes
< Content-Length: 5
< Content-Type: text/html; charset=UTF-8
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_namem@obelix:~$
<!--NeedCopy-->
```

例 **3: URL** が受信されたときに別の **URL** にリダイレクトして応答する

```
> add responder action act1 redirect ""www.google.com""
 Done
> add responder policy pol1 'HTTP.REQ.URL.CONTAINS("file")' act1
 Done
> bind responder global pol1 1
 Done
>

name:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
< HTTP/1.1 302 Found : Moved Temporarily
< Location: www.google.com
< Connection: close
< Cache-Control: no-cache
< Pragma: no-cache
<
* Closing connection 0
name@obelix:~$
<!--NeedCopy-->
```


例 **4:** 任意の式またはテキストにできるメッセージで応答する

```
add responder action act123 respondwith ""Please reach out to
   administrator""
add responder policy pol1 "HTTP.REQ.URL.CONTAINS("file")" act123
bind responder global pol1 100 END

name@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10..Responder Action and Policy:

>add responder action Redirect-Action redirect ""https://xyz.abc.com/
   dispatcher/SAML2AuthService?siteurl=wmav"" -responseStatusCode 302

>add responder policy Redirect-Policy "HTTP.REQ.HOSTNAME.CONTAINS("abc"
   )" Redirect-Action

Binding to LB Virtual Server:

>bind lb vserver Test1_SF -policyName Redirect-Policy -priority 100 -
   gotoPriorityExpression END -type REQUEST.
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
* Connection #0 to host 10.10.10.10 left intact
Please reach out to administratort_name@obelix:~$
<!--NeedCopy-->
```

例 **5: HTML** インポートされたページで応答する

```
import responder htmlpage http://10.10.10.10)/testsite/file5.html
   page112
add responder action act1 respondwithHtmlpage page1
add responder policy pol1 true act1
bind responder global pol1 100

name@obelix:~$ curl -v http://10.10.10.10)/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10)) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.102.58.140
> Accept: \*/\*
>
< HTTP/1.1 200 OK
< Content-Length: 5
< Content-Type: text/html
```



```
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_name@obelix:~$
<!--NeedCopy-->
```

例 **6:** レスポンダーポリシーを使用したホスト名に基づく **URL** のリダイレクト

```
Responder Action and Policy:

>add responder action Redirect-Action redirect ""https://xyz.abc.com/
    dispatcher/SAML2AuthService?siteurl=wmav"" -responseStatusCode 302

>add responder policy Redirect-Policy "HTTP.REQ.HOSTNAME.CONTAINS("abc"
    )" Redirect-Action

Binding to LB Virtual Server:

>bind lb vserver Test1_SF -policyName Redirect-Policy -priority 100 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

レート制限

August 15, 2023

レート制限機能を使用すると、NetScaler ADC アプライアンス上の特定のネットワークエンティティまたは仮想エンティティの最大負荷を定義できます。この機能を使用すると、エンティティに関連付けられたトラフィックのレートを監視し、トラフィックレートに基づいて予防措置をリアルタイムで実行するようにアプライアンスを設定できます。この機能は、アプライアンスに大量の要求を送信している敵対的なクライアントからネットワークが攻撃を受けている場合に特に役立ちます。クライアントへのリソースの可用性に影響するリスクを軽減し、アプライアンスが管理するネットワークおよびリソースの信頼性を向上させることができます。

仮想サーバ、URL、ドメイン、URL とドメインの組み合わせなど、仮想エンティティおよびユーザ定義エンティティに関連付けられたトラフィックのレートをモニタおよび制御できます。トラフィックレートが高すぎる場合はトラフィックレートを調整し、トラフィックレートに基づいて情報キャッシングを行い、トラフィックレートが事前定義された制限を超えた場合はトラフィックを特定の負荷分散仮想サーバーにリダイレクトできます。HTTP、TCP、および DNS 要求にレートベースのモニタリングを適用できます。

特定のシナリオのトラフィックレートをモニタするには、レート制限 *ID* を設定します。レート制限識別子は、タイムスライスと呼ばれる指定された期間内に許可される（特定のタイプの）要求または接続の最大数などの数値しきい値を指定します。

オプションで、ストリームセレクタと呼ばれるフィルタを設定し、識別子の設定時にレート制限 ID に関連付けることができます。オプションのストリームセレクタと制限 ID を設定したら、詳細ポリシーから制限 ID を呼び出す必要が

あります。識別子は書き換え、レスポンダ、DNS、統合キャッシュなど、識別子が有用である可能性がある任意の機能から呼び出すことができます。

レート制限 ID の SNMP トラップは、グローバルに有効または無効にできます。タイムスライスごとに複数のトラップを生成するように指定していない限り、各トラップには、レート制限 ID の設定済みデータ収集間隔（タイムスライス）の累積データが含まれます。SNMP トラップおよびマネージャの設定の詳細については、「SNMP」を参照してください。

ストリームセレクターの構成

August 15, 2023

トラフィックストリームセレクタは、アクセスを制限するエンティティを識別するためのオプションのフィルタです。セレクタは、要求または応答に適用され、レートストリーム識別子によって分析できるデータポイント（キー）を選択します。これらのデータポイントは、IP アドレス、サブネット、ドメイン名、TCP または UDP 識別子、URL 内の特定の文字列または拡張子など、トラフィックのほぼすべての特性に基づくことができます。

ストリームセレクタは、selectlets と呼ばれる個々の高度なポリシー式で構成されます。各 selectlet は、複合ではない高度なポリシー式です。トラフィックストリームセレクタには、selectlet と呼ばれる複合以外の式を最大 5 つ含めることができます。各 selectlet は、他の式との AND 関係にあると見なされます。次に、選択レットの例をいくつか示します。

```
http.req.url
http.res.body(1000>after_str("car_model").before_str("made_in")"
"client.ip.src.subnet(24)"
<!--NeedCopy-->
```

パラメータを指定する順序は重要です。たとえば、1 つのセレクターで IP アドレスとドメインを（この順序で）構成し、別のセレクターでドメインと IP アドレスを（逆の順序で）指定すると、NetScaler ADC はこれらの値を一意と見なします。これにより、同じトランザクションが 2 回カウントされる可能性があります。また、複数のポリシーが同じセレクタを呼び出す場合、NetScaler ADC は同じトランザクションを複数回カウントできます。

注: ストリームセレクタの式を変更すると、それを呼び出すポリシーが新しいポリシーラベルまたはバインドポイントにバインドされていると、エラーが発生することがあります。たとえば、myStreamSelector1 という名前のストリームセレクタを作成し、myLimitId1 から呼び出して、dnsRateLimit1 という名前の DNS ポリシーから識別子を呼び出すとします。MyStreamSelector1 の式を変更すると、dnsRateLimit1 を新しいバインドポイントにバインドするときにエラーが発生することがあります。回避策は、これらの式を呼び出すポリシーを作成する前に、これらの式を変更することです。

コマンドラインインターフェイスを使用してトラフィックストリームセレクタを設定するには

コマンドプロンプトで入力します。


```
add stream selector <name> <rule> ...
<!--NeedCopy-->
```

例**:**

```
add stream selector myStreamSel HTTP.REQ.URL CLIENT.IP.SRC
<!--NeedCopy-->
```

構成ユーティリティを使用してストリームセレクターを構成するには

[AppExpert] > [レート制限] > [セレクタ] に移動し、[追加] をクリックして、関連する詳細を指定します。

トラフィックレート制限 **ID** の構成

August 15, 2023

レート制限識別子は、特定の時間間隔内にトラフィック量が指定された値を超えているかどうかを確認します。特定の時間間隔内にトラフィック量が制限を超えた場合、識別子は「Boolean TRUE」を返します。ポリシー規則の複合 DAdvanced ポリシー式に制限識別子を含める場合は、ストリームセレクタを含める必要があります。を指定しない場合、制限識別子は複合式で識別されるすべての要求または応答に適用されます。

> 注:
>
> 文字列結果 (HTTP.REQ.URL など) を格納するための最大長は 60 文字です。文字列 (URL など) の長さが 1000 文字で、そのうちの 50 文字で文字列を一意に識別できる場合は、式を使用して必要な 50 文字を抽出できます。

コマンドラインインターフェイスからトラフィック制限 **ID** を設定するには

コマンドプロンプトで入力します。

```
add ns limitIdentifier <limitIdentifier> -threshold <positive_integer>
    -timeSlice <positive_integer> -mode <mode> -limitType ( BURSTY |
    SMOOTH ) -selectorName <string> -maxBandwidth <positive_integer> -
    trapsInTimeSlice <positive_integer>
<!--NeedCopy-->
```

引数の説明

limitIdentifier. レート制限 ID の名前。ASCII 文字またはアンダースコア (_) 文字で始まり、ASCII 英数字またはアンダースコア文字のみで構成する必要があります。予約語は使用できません。これは必須の議論です。最大長: 31

threshold. リクエスト (モードが REQUEST_RATE として設定されている) がタイムスライスごとに追跡される場合に、指定されたタイムスライスで許可されるリクエストの最大数。接続 (モードが CONNECTION として設定されている) が追跡される場合、通過する接続の合計数になります。デフォルト値:1 最小値:1 最大値:4294967295

TimeSlice。10 の倍数で指定された時間間隔（ミリ秒）。この間は、リクエストがしきい値を超えるかどうかをチェックするために追跡されます。この引数は、モードが REQUEST_RATE に設定されている場合にのみ必要です。デフォルト値:1000 最小値:10 最大値:4294967295

mode. 追跡するトラフィックのタイプを定義します。

1. REQUEST_RATE。リクエスト/タイムスライスを追跡します。
2. CONNECTION. アクティブなトランザクションを追跡します。

limitType. 制限のタイプを定義します。

- スムーズ: 設定したタイムフレームの各タイムスライスに負荷が均等に分散されます。一貫性のあるアプリケーショントラフィックに使用します。
- **Bursty**: 負荷が設定されたしきい値を下回った場合にリクエストをパススルーさせます。散発的なアプリケーショントラフィックに使用します。設定した時間枠内のいつでも負荷がピークに達すると便利です。

  たとえば、設定されている最大リクエスト数は 100 で、時間枠は 10 秒です。アプリケーションが最初の 1 秒間に 80 のリクエストを受け取った場合、これらの制限タイプの動作は異なります。バースト制限タイプでは、負荷が設定されたしきい値を下回っているため、リクエストは通過できます。ただし、スムーズリミットタイプでは 1 秒あたり 10 リクエストしか許可されません。そのため、超過負荷に対して設定されたアクションが適用されます。

selectorName. レート制限セレクタの名前。この引数が NULL の場合、レート制限は、仮想サーバーまたは NetScaler ADC によって受信されたすべてのトラフィック（制限識別子が仮想サーバーにバインドされているかグローバルにバインドされているかによって異なります）にフィルタリングなしで適用されます。最大長：**31**

maxBandwidth. 許可される最大帯域幅（kbps）。最小値:0 最大値:4294967287

例**:**

BURSTY モードでトラフィックレート制限 ID を設定します。

```
add ns limitIdentifier 100_request_limit -threshold 100 -timeSlice 1000
    -mode REQUEST_RATE -limitType BURSTY -selectorName
   limit_100_requests_selector -trapsInTimeSlice 30
<!--NeedCopy-->
```

SMOOTH モードでのトラフィックレート制限識別子の設定:

```
add ns limitidentifier limit_req -mode request_rate -limitType smooth -
   timeslice 1000 -Threshold 2000 -trapsInTimeSlice 200
<!--NeedCopy-->
```


構成ユーティリティを使用してトラフィック制限識別子を構成するには

[AppExpert] > [レート制限] > [制限識別子] に移動し、[追加] をクリックして、関連する詳細を指定します。

トラフィックレートポリシーの設定とバインド

August 15, 2023

レートベースのアプリケーションの動作を実装するには、適切な NetScaler ADC 機能でポリシーを構成します。この機能は、高度なポリシーをサポートしている必要があります。この機能がトラフィックレートを分析できるようにするには、ポリシー式に次の式プレフィクスを含める必要があります。

```
sys.check_limit(<limit_identifier>)
<!--NeedCopy-->
```

ここで、制限識別子は制限識別子の名前です。

ポリシー式は、少なくとも 2 つのコンポーネントを含む複合式である必要があります。

- レート制限 ID が適用されるトラフィックを識別する式。次に例を示します:

```
    http.req.url.contains("my_aspx.aspx").
<!--NeedCopy-->
```

- レート制限識別子を識別する式。たとえば、sys.check_limit (「my_limit_identifier」)。これは、ポリシー式の最後の式である必要があります。

コマンドラインインターフェイスを使用してレートベースポリシーを構成するには

コマンドプロンプトで次のコマンドを入力して、レートベースポリシーを構成し、構成を確認します。

```
add cache|dns|rewrite|responder policy <policy_name> -rule expression
    && sys.check_limit("<LimitIdentifierName>") [<feature-specific
    information>]
<!--NeedCopy-->
```

次に、レートベースのポリシールールの完全な例を示します。この例では、ポリシーに関連付けられているレスポンダアクション send_direct_url が設定されていることを前提としています。sys.check_limit パラメータは、ポリシー式の最後の要素である必要があります。

```
add responder policy responder_threshold_policy "http.req.url.contains(
    "myindex.html") && sys.check_limit("my_limit_identifier")"
    send_direct_url
<!--NeedCopy-->
```


ポリシーをグローバルに、または仮想サーバにバインドする方法については、「高度なポリシーポリシーのバインド」を参照してください。

構成ユーティリティを使用してレートベースのポリシーを構成するには

1. ナビゲーションウィンドウで、ポリシーを構成する機能 (たとえば、統合キャッシュ、書き換え、レスポンダー) を展開し、[ポリシー] をクリックします。
2. 詳細ペインで、［Add］をクリックします。[Name] に、ポリシーの一意の名前を入力します。
3. [式] にポリシー規則を入力し、式の最後のコンポーネントとして sys.check_limit パラメータが含まれていることを確認します。次に例を示します:

```
1 http.req.url.contains("my_aspx.aspx") && sys.check_limit("
     my_limit_identifier")
2 <!--NeedCopy-->
```

4. ポリシーに関する機能固有の情報を入力します。

   たとえば、ポリシーをアクションまたはプロファイルに関連付ける必要がある場合があります。詳細については、機能固有のドキュメントを参照してください。

5. ［Create］をクリックしてから、［Close］をクリックします。
6. ［保存］をクリックします。

トラフィックレートの表示

August 15, 2023

1 つ以上の仮想サーバーを経由するトラフィックがレートベースのポリシーと一致する場合、このトラフィックのレートを表示できます。レート統計は、レートベースポリシーのルールで指定した制限識別子に保持されます。複数のポリシーが同じ制限識別子を使用している場合は、特定の制限識別子を使用するすべてのポリシーへのヒット数によって定義されたトラフィックレートを表示できます。

コマンドラインインターフェイスを使用してトラフィックレートを表示するには

コマンドプロンプトで、次のコマンドを入力してトラフィックレートを表示します。

```
1 show ns limitSessions <limitIdentifier>
2 <!--NeedCopy-->
```

例:


```
sh limitsession myLimitSession
<!--NeedCopy-->
```

構成ユーティリティを使用してトラフィックレートを表示するには

1. 「AppExpert」>「レート制限」>「制限識別子」に移動します。
2. トラフィックレートを表示したい制限識別子を選択します。
3. 「セッションを表示」ボタンをクリックします。1 つ以上の仮想サーバーを経由するトラフィックが、この制限識別子を使用するレート制限ポリシーに一致した場合（かつヒット数がこの識別子に設定されたタイムスライス内にある場合）、セッションの詳細ダイアログ・ボックスが表示されます。それ以外の場合は、「セッションが存在しません」というメッセージが表示されます。

レートベースのポリシーのテスト

August 15, 2023

レートベースのポリシーをテストするには、レートベースのポリシーがバインドされている任意の仮想サーバーにトラフィックを送信できます。

タスクの概要**:** レートベースのポリシーのテスト

1. ストリームセレクター (オプション) とレート制限識別子 (必須) を設定します。次に例を示します:

```
add stream selector sel_subnet Q.URL "CLIENT.IP.SRC.SUBNET(24)"
add ns limitIdentifier k_subnet -Threshold 4 -timeSlice 3600 -mode
    REQUEST_RATE  -limittype smooth  -selectorName sel_subnet  -
   trapsInTimeSlice 8
<!--NeedCopy-->
```

2. レート制限識別子を使用するポリシーに関連付けるアクションを設定します。次に例を示します:

```
add responder action resp_redirect redirect ""http://response_site
   .com/""
<!--NeedCopy-->
```

3. sys.check_limit 表現プレフィックスを使用してレート制限識別子を呼び出すポリシーを設定します。たとえば、ポリシーでは、次のように特定のサブネットから届くすべてのリクエストにレート制限識別子を適用できます。

```
add responder policy resp_subnet "SYS.CHECK_LIMIT("k_subnet")"
   resp_redirect
<!--NeedCopy-->
```


4. ポリシーをグローバルにバインドするか、仮想サーバーにバインドします。次に例を示します:

```
bind responder global resp_subnet 6 END -type DEFAULT
<!--NeedCopy-->
```

5. ブラウザのアドレスバーで、テスト HTTP クエリを仮想サーバーに送信します。次に例を示します:

```
http://<IP of a vserver>/testsite/test.txt
<!--NeedCopy-->
```

6. NetScaler のコマンドプロンプトで、次のように入力します。

```
show ns limitSessions \<limitIdentifier\>
<!--NeedCopy-->
```

例

```
> sh limitsession k_subnet
1)      Time Remaining:       98 secs  Hits: 2
                             Action Taken: 0
        Total Hash:    1718618  Hash String: /test.txt
        IPs gathered:
                1) 10.217.253.0
        Active Transactions: 0
 Done
>
<!--NeedCopy-->
```

7. クエリを繰り返し、リミット識別子の統計をもう一度チェックして、統計が正しく更新されていることを確認します。

料金ベースのポリシーの例

August 15, 2023

このトピックでは、レートベースのポリシーの例をいくつか示します。

URL からのリクエスト数を制限する

次のコマンドを実行して、URL からの 1 秒あたりのリクエスト数を制限します。

```
add stream selector ipStreamSelector http.req.url "client.ip.src" add
    ns limitIdentifier ipLimitIdentifier -threshold 4 -timeSlice 1000 -
    mode request_rate -limitType smooth -selectorName ipStreamSelector

add responder action myWebSiteRedirectAction redirect ""http: //www.
    mycompany .com/""
```



```
4
5 add responder policy ipLimitResponderPolicy "http.req-url.contains("
     myasp.asp") && sys.check_limit("ipLimitIdentifier")"
     myWebSiteRedirectaction
6
7 bind responder global ipLimitResponderPolicy 100 END -type default
8 <!--NeedCopy-->
```

リクエスト **URL** のレスポンスをキャッシュする

リクエスト URL レートが 20000 ミリ秒あたり 5 を超える場合は、次のコマンドを実行してレスポンスをキャッシュします。

```
1 add stream selector cacheStreamSelector http.req.url add ns
     limitidentifier cacheRateLimitIdentifier -threshold 5 -timeSlice
     2000 -selectorName cacheStreamSelector
2
3 add cache policy cacheRateLimitPolicy -rule "http req.method.eq(get) &&
      sys.check_limit "cacheRateLimitIdentifier")" -action cache
4
5 bind cache global cacheRateLimitPolicy -priority 10
6 <!--NeedCopy-->
```

Cookie に基づいて接続をドロップする

リクエストがレート制限を超えた場合、次のコマンドを実行して、www.mycompany.comからのリクエストで受信した Cookie に基づいて接続を切断します。

```
1 add stream selector reqCookieStreamSelector "http req.cookie «value("
     mycookie")" "client.ip.src.subnet(24)"
2
3 add ns limitIdentifier myLimitIdentifier -Threshold 2 -timeSlice 3000 -
     selectorName reqCookieStreamSelector
4
5 add responder action sendRedirectUrl redirect '"http://www.mycompany.
     com" + http.req.url' -bypassSafetyCheck YES
6
7 add responder policy rateLimitCookiePolicy "http. req.url.contains("www
     .yourcompany.com") && sys check_limit("myLimitIdentifier")"
     sendRedirectUrl
8 <!--NeedCopy-->
```

特定の **IP** アドレスからの **DNS** パケットをドロップ

特定のクライアント IP アドレスと DNS ドメインからのリクエストがレート制限を超えた場合に、次のコマンドを実行して DNS パケットをドロップします。


```
add stream selector dropDNSStreamSelector client udp.dns.domain client.ip.src
add ns limitIdentifier dropDNSRateIdentifier -timeslice 20000 -mode request_rate -selectorName dropDNSStreamSelector -maxBandwidth 1 -trapsintimeslice 20

add dns policy dnsDropOnClientRatePolicy "sys check_limit ("dropDNSRateIdentifier")" -drop yes
<!--NeedCopy-->
```

同じホストからの **HTTP** リクエスト数を制限する

次のコマンドを実行して、同じホストから送信される、サブネットマスクが 32 で、宛先 IP アドレスが同じの HTTP 要求の数を制限します。

```
add stream selector ipv6_sel "CLIENT.IPv6.src.subne (32)" CLIENT.IPv6.dst Q.URL
add ns limitIdentifier ipvé_id -imeSlice 20000 -selectorName ipvé_sel
add lb vserver ipv6é_vip HTTP 3ffe::209 80 -persistenceType NONE -cltTimeout 180
add responder action redirect_page redirect ""http://redirectpage.com/""
add responder policy ipvé_resp_pol "SYS.CHECK_LIMIT("ipv6_id")" redirect_page
bind responder global ipv6_resp_pol 5 END -type DEFAULT
<!--NeedCopy-->
```

レートベースのポリシーのサンプルユースケース

August 15, 2023

以下のシナリオでは、グローバルサーバー負荷分散 (GSLB) におけるレートベースのポリシーの 2 つの用途について説明します。

- 最初のシナリオでは、DNS リクエストのレートが 1 秒あたり 1000 件を超える場合に、新しいデータセンターにトラフィックを送信するレートベースのポリシーの使用について説明します。
- 2 番目のシナリオでは、特定の期間にローカル DNS (LDNS) クライアントに 5 つ以上の DNS 要求が到着した場合、それ以上の要求は破棄されます。

トラフィックレートに基づくトラフィックのリダイレクト

このシナリオでは、近接ベースの負荷分散方法と、特定のリージョンの DNS リクエストを識別するレート制限ポリシーを設定します。レート制限ポリシーでは、1 秒あたり 1000 件の DNS リクエストのしきい値を指定します。DNS

ポリシーは、「Europe.GB.17.london.uk-East.isp-UK」というリージョンの DNS リクエストにレート制限ポリシーを適用します。DNS ポリシーでは、リクエスト 1001 から始まり、1 秒間隔の終わりまで続くレート制限のしきい値を超える DNS リクエストは、「NorthAmerica.us.tx.dallas.US-East.ISP-US」リージョンに関連付けられている IP アドレスに転送されます。「

次の構成は、このシナリオを示しています。

```
add stream selector DNSSelector1 client.udp.dns.domain

add ns limitIdentifier DNSLimitIdentifier1 -threshold 5 -timeSlice 1000
    -selectorName DNSSelector1

add dns policy DNSLimitPolicy1 "client.ip.src.matches_location("Europe.
   GB.17.London.\*.\*") &&
sys.check_limit("DNSLimitIdentifier1")" -preferredLocation "North
   America.US.TX.Dallas.\*.\*"

bind dns global DNSLimitPolicy1 5
<!--NeedCopy-->
```

トラフィックレートに基づく **DNS** リクエストのドロップ

次のグローバルサーバ負荷分散の例では、ドメインごとに特定の間隔で最大 5 つの DNS 要求を LDNS クライアントに転送して解決できるようにするレート制限ポリシーを設定します。このレートを超えるリクエストはすべてドロップされます。このタイプのポリシーは、NetScaler をリソースの悪用から保護するのに役立ちます。たとえば、このシナリオでは、接続の存続時間 (TTL) が 5 秒の場合、このポリシーにより LDNS はドメインを再クエリできなくなります。代わりに、NetScaler にキャッシュされているデータを使用します。

```
add stream selector LDNSSelector1 client.udp.dns.domain client.ip.src

add ns limitIdentifier LDNSLimitIdentifier1 -threshold 5 -timeSlice
   1000 -selectorName LDNSSelector1

add dns policy LDNSPolicy1 "client.udp.dns.domain.contains(".") && sys.
   check_limit("LDNSLimitIdentifier1")" -drop YES

bind dns global LDNSPolicy1 6

show gslb vserver gvip

gvip - HTTP     State: UP
Last state change was at Mon Sep  8 11:50:48 2008 (+711 ms)
Time since last state change: 1 days, 02:55:08.830
Configured Method: STATICPROXIMITY
BackupMethod: ROUNDROBIN
No. of Bound Services :  3 (Total)       3 (Active)
Persistence: NONE       Persistence ID: 100
Disable Primary Vserver on Down: DISABLED       Site Persistence: NONE
Backup Session Timeout: 0
```



```
Empty Down Response: DISABLED
Multi IP Response: DISABLED Dynamic Weights: DISABLED
Cname Flag: DISABLED
Effective State Considered: NONE
1.     site11_svc(10.100.00.00: 80)- HTTP State: UP    Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: Europe.GB.17.London.UK-East.ISP-UK
2.      site12_svc(10.101.00.100: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.TX.Dallas.US-East.ISP-US
3.      site13_svc(10.102.00.200: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.NJ.Salem.US-Mid.ISP-US
4.      www.gslbindia.com        TTL: 5 secn
Cookie Timeout: 0 min   Site domain TTL: 3600 sec
Done
<!--NeedCopy-->
```

トラフィックドメインのレート制限

April 15, 2024

トラフィックドメインのレート制限を設定できます。NetScaler 表現言語の次の式は、トラフィックドメインに関連するトラフィックを識別します。

- client.traffic_domain.id

特定のトラフィックドメイン、トラフィックドメインのセット、またはすべてのトラフィックドメインに関連するトラフィックのレート制限を設定できます。

NetScaler で IP レート制限を構成する方法について詳しくは、次のビデオを参照してください:

これは埋め込みビデオです。リンクをクリックしてビデオを見る

トラフィックドメインのレート制限を構成するには、構成ユーティリティまたは NetScaler コマンドラインを使用して、NetScaler アプライアンスで次の手順を実行します:

1. client.traffic_domain.id 表現を使用してレート制限の対象となるトラフィックを識別するストリームセレクターを設定します。
2. レート制限するトラフィックの最大しきい値などのパラメータを指定するレート制限識別子を設定します。また、このステップでは、ストリームセレクターをレートリミッターに関連付けます。

3. レート制限識別子を使用するポリシーに関連付けるアクションを設定します。
4. sys.check_limit 表現プレフィックスを使用してレート制限識別子を呼び出すポリシーを設定し、アクションをこのポリシーに関連付けます。
5. ポリシーをグローバルにバインドします。

NetScaler NS1 に ID が 10 と 20 の 2 つのトラフィックドメインが構成されている例を考えてみましょう。トラフィックドメイン 10 では、LB1-TD-1 はサーバー S1 と S2 の負荷分散を行うように設定され、LB2-TD1 はサーバー S3 と S4 の負荷分散を行うように設定されています。

トラフィックドメイン 20 では、LB1-TD-2 はサーバ S5 と S6 のロードバランシングを行うように設定され、LB2-TD2 はサーバ S7 と S8 のロードバランシングを行うように設定されています。

次の表は、設定例に含まれるトラフィックドメインのレート制限ポリシーの例を示しています。

目的	CLI コマンド
各トラフィックドメインのリクエスト数を 1 秒あたり 10 回に制限します。	add stream selector tdratelimit-1 CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier limitidf-1 -threshold 10 -selectorName tdratelimit-1 -trapsInTimeSlice 0 add responder policy ratelimit-pol “sys.check_limit(\ “limitidf-1\”)” DROP bind responder global ratelimit-pol 1
各トラフィックドメインのリクエスト数は、クライアントあたり 1 秒あたり 5 回に制限します。	add stream selector tdandclientip CLIENT.IP.SRC,CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td_limitidf -threshold 5 -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy tdratelimit-pol “sys.check_limit(\ “td_limitidf\”)” DROP bind responder global tdratelimit-pol 2
特定のトラフィックドメイン（たとえば、トラフィックドメイン 10）に送信されるリクエストの数を 3 秒ごとに 30 リクエストに制限します。	add stream selector tdratelimit CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td10_limitidf -threshold 30 -timeSlice 3000 -selectorName tdratelimit -trapsInTimeSlice 5 add responder policy td10ratelimit “client.traffic_domain.id==10 && sys.check_limit(\ “td10_limitidf\”)” DROP bind responder global td10ratelimit 3

目的	CLI コマンド
特定のトラフィックドメイン（たとえば、トラフィックドメイン 20）の接続数は、クライアントあたり 1 秒あたり 5 つに制限します。	add stream selector tdandclientip CLIENT.IP.SRC CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td20_limitidf -threshold 5 -mode CONNECTION -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy td20_ratelimit “client.traffic_domain.id==20 && sys.check_limit(\“td20_limitidf\”)” DROP bind responder global td20_ratelimit 4

パケットレベルでレート制限を設定する

August 15, 2023

ストリームセレクタとレスポンダポリシーを設定して、セレクタによって識別されるすべての接続を通過するパケットレベルで統計情報を収集できます。1 秒あたりのパケット数が、設定されたしきい値を超えると、ポリシーは設定されたアクション（RESET または DROP）を適用します。これらのポリシーは、すべてのタイプの仮想サーバに対して構成できます。すべてのサイズのパケットが考慮されます。

パケットレベルでレート制限を設定するには、次の作業を実行します

1. 負荷分散を有効にする
2. ストリームセレクターを追加する
3. ストリーム識別子の追加
4. レスポンダーポリシーの追加
5. 負荷分散仮想サーバの追加
6. レスポンダーポリシーのバインド

負荷分散機能を有効にするには

コマンドプロンプトで入力します。

```
enable ns feature lb
<!--NeedCopy-->
```


ストリームセレクターを追加するには

コマンドプロンプトで入力します。

```
1 add stream selector packetlimitselector client.ip.src client.tcp.
     srcport client.ip.dst client.tcp.dstport
2 <!--NeedCopy-->
```

ストリーム **ID** を追加するには

コマンドプロンプトで入力します。

```
1 add stream identifier packetlimitidentifier packetlimitselector -
     interval 1
2 <!--NeedCopy-->
```

ACK のみのパケットのトラッキングを有効にするには

コマンドプロンプトで入力します。

```
1 set stream identifier packetlimitidentifier  - trackAckOnlyPackets
     ENABLED
2 <!--NeedCopy-->
```

レスポンダーポリシーを追加するには

コマンドプロンプトで入力します。

```
1 add responder policy packet_rate_sessionpolicy "ANALYTICS.STREAM("
     packetlimitidentifier").COLLECT_STATS("PACKET_LIMIT", <
     max_threshold_PPS>, ACTION, 0/1)" NOOP
2 <!--NeedCopy-->
```

各項目の意味は次のとおりです。

- <max_threshold_PPS> は、接続を通過できる 1 秒あたりの最大パケット数です。
- ACTION は、ドロップまたはリセットが可能です。
- 0 または 1 は制限タイプを表し、0 は BURSTY 制限タイプ、1 は SMOOTH 制限タイプを表します。

例**:**

```
1 add responder policy packet_rate_sessionpolicy "ANALYTICS.STREAM("
     packetlimitidentifier").COLLECT_STATS("PACKET_LIMIT", 40, RESET, 0)"
      NOOP
2 <!--NeedCopy-->
```


負荷分散仮想サーバーを追加するには

コマンドプロンプトで入力します。

```
add lb vserver <name> <serviceType> <ip> <port>

add lb vserver Vserver-lb-1 HTTP 10.102.20.200 80
<!--NeedCopy-->
```

レスポンダーポリシーをバインドするには

セレクタとレスポンダポリシーを構成したら、ポリシーをグローバルにバインドすることも、特定の仮想サーバにバインドすることもできます。

コマンドプロンプトで、次のいずれかのコマンドを入力します。

```
bind responder global <policyName> <priority> [<gotoPriorityExpression
    >] [-type <type>] [-invoke  (<labelType>  <labelName>) ]
<!--NeedCopy-->
```

または

```
bind lb vserver <name>@  (-policyName <string>@ [-priority <
    positive_integer>]
<!--NeedCopy-->
```

例:

```
bind responder global packet_rate_sessionpolicy 101 END -type
    REQ_DEFAULT

bind responder global packet_rate_sessionpolicy 102 END -type

bind lb vserver v1 -policyname packet_rate_sessionpolicy -priority 10
<!--NeedCopy-->
```

レスポンダー

August 15, 2023

> 警告
>
> 従来のポリシーを使用したフィルター機能は廃止され、代替として、高度なポリシーインフラストラクチャで書き換え機能とレスポンダー機能を使用することをお勧めします。

今日の複雑な Web 構成では、表面上は似ているように見える HTTP リクエストに対して異なる応答が必要になることがよくあります。ユーザーがウェブページをリクエストしたときに、ユーザーの地理的位置、ブラウザーの仕様、ブラウザーが受け付ける言語、および優先順序に応じて、別のページを提供したい場合があります。リクエストが DDoS 攻撃を引き起こしている、またはハッキングを試みている IP アドレス範囲からのものである場合は、接続を切断する必要があります。

レスポンダーは TCP、DNS (UDP)、HTTP などのプロトコルをサポートしています。アプライアンスでレスポンダーを有効にすると、リクエストの送信者、送信元、およびセキュリティやシステム管理に影響するその他の基準に基づいてサーバーの応答を行うことができます。この機能はシンプルで迅速に使用できます。より複雑な機能の呼び出しを避けることで、複雑な処理を必要としない要求の処理に費やされる CPU サイクルと時間を削減できます。

財務情報などの機密データを処理する場合、クライアントが安全な接続を使用してサイトを閲覧するようにしたい場合は、`http://`の代わりに`https://`を使用してリクエストを安全な接続にリダイレクトできます。

レスポンダーを使用するには、次の操作を行います。

- アプライアンスのレスポンダー機能を有効にします。
- レスポンダーアクションを設定します。アクションには、カスタムレスポンスを生成したり、リクエストを別の Web ページにリダイレクトしたり、接続をリセットしたりすることができます。
- レスポンダーポリシーを設定します。ポリシーによって、アクションを実行する必要があるリクエスト (トラフィック) が決まります。
- 各ポリシーをバインドポイントにバインドして有効にします。バインドポイントとは、NetScaler アプライアンスがトラフィックを検査してポリシーと一致するかどうかを確認するエンティティのことです。たとえば、バインドポイントは負荷分散仮想サーバーである可能性があります。

どのポリシーにも一致しないリクエストにはデフォルトアクションを指定でき、そうでなければエラーメッセージを生成するアクションについてはセーフティチェックをバイパスできます。

NetScaler の書き換え機能は、NetScaler が処理する要求または応答の一部の情報を書き換えるのに役立ちます。次のセクションでは、この 2 つの機能の違いをいくつか示します。

[書き換え] オプションと [レスポンダー] オプションの比較

書き換え機能とレスポンダ機能の主な違いは次のとおりです。

Responder は、レスポンスまたはサーバーベースの式には使用できません。Responder は、クライアントパラメータに応じて、次のシナリオでのみ使用できます。

- 新しい Web サイトまたはウェブページへの HTTP リクエストのリダイレクト
- カスタムレスポンスで応答する
- 要求レベルで接続をドロップまたはリセットする

レスポンダーポリシーがある場合、NetScaler ADC はクライアントからの要求を調べ、該当するポリシーに従ってアクションを実行し、クライアントに応答を送信し、クライアントとの接続を閉じます。

書き換えポリシーがある場合、NetScaler ADC はクライアントからの要求またはサーバーからの応答を調べ、該当するポリシーに従ってアクションを実行し、トラフィックをクライアントまたはサーバーに転送します。

一般に、リクエストベースのパラメータに基づいてアプライアンスに接続をリセットまたはドロップさせたい場合は、レスポンダーを使用することをお勧めします。レスポンダーを使用してトラフィックをリダイレクトするか、カスタムメッセージで応答します。HTTP リクエストとレスポンスのデータを操作するには、書き換えを使用します。

レスポンダー機能の有効化

August 15, 2023

レスポンダー機能を使用するには、まずレスポンダー機能を有効にする必要があります。

NetScaler CLI を使用してレスポンダー機能を有効にするには:

コマンドプロンプトで次のコマンドを入力して、レスポンダー機能を有効にし、構成を確認します。

- `enable ns feature <feature>`
- `show ns feature`

例:

```
enable ns feature Responder
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 .
 .
 .
 19)     Responder                       RESPONDER            ON
 20)     NetScaler Push                  push                 OFF
 Done
>
<!--NeedCopy-->
```

GUI を使用してレスポンダ機能を有効にするには、次の手順を実行します。

1. ナビゲーションペインで、[システム] を展開し、[設定] をクリックします。
2. 詳細ウィンドウの [** モードと機能] で、[** 高度な機能の変更] をクリックします。
3. [拡張機能の構成] ダイアログボックスで、[レスポンダー] チェックボックスをオンにし、[**OK**] をクリックします。
4. [機能の有効化/無効化] ダイアログボックスで、[はい] をクリックします。ステータスバーに、機能が有効になっていることを示すメッセージが表示されます。

レスポンダーアクションの設定

August 15, 2023

レスポンダー機能を有効にしたら、要求を処理するための 1 つ以上のアクションを設定する必要があります。レスポンダーは、次のタイプのアクションをサポートします。

- **Respond with**. リクエストを Web サーバーに転送せずに、Target 式で定義されたレスポンスを送信します。(NetScaler ADC アプライアンスは、Web サーバーの代わりとして機能し、Web サーバーとして機能します)。このタイプのアクションは、単純な HTML ベースのレスポンスを手動で定義する場合に使用します。通常、応答アクションのテキストは、Web サーバーのエラーコードと簡潔な HTML ページで構成されます。
- **Respond with SQL OK**。Target 式で定義された指定された SQL OK レスポンスを送信します。このタイプのアクションは、SQL クエリに SQL OK レスポンスを送信するときに使用します。
- **Respond with SQL Error**。Target 式で定義された指定された SQL エラーレスポンスを送信します。この種類のアクションは、SQL クエリに SQL Error 応答を送信するときに使用します。
- **Respond with HTML page**。指定された HTML ページをレスポンスとして送信します。以前にアップロードされた HTML ページのドロップダウンリストから選択するか、新しい HTML ページをアップロードできます。このタイプのアクションは、インポートされた HTML ページをレスポンスとして送信するときに使用します。アプライアンスは、responsewithhtmlpage レスポンダアクションのカスタムヘッダーで応答します。最大 8 つのカスタムヘッダーを設定できます。インポートされた HTML ページは/var/download/responderディレクトリに保存されます。
- **Redirect**。要求を別の Web ページまたは Web サーバーにリダイレクトします。リダイレクトアクションは、DNS に存在するが、実際のウェブサーバーがない「ダミー」の Web サイトに送信されたリクエストを実際の Web サイトにリダイレクトできます。また、検索リクエストを適切な URL にリダイレクトすることもできます。通常、Redirect アクションのリダイレクトターゲットは完全な URL で構成されます。

CLI を使用してレスポンダーアクションを設定します。

指定されたレスポンダーアクションの現在の設定が表示されます。アクション名を指定しない場合は、NetScaler ADC アプライアンスで現在構成されているすべてのレスポンダーアクションのリストを省略した設定で表示します。

コマンドプロンプトで次のコマンドを入力して、レスポンダーアクションを構成し、構成を確認します。

- `add responder action <name> <type> <target>`
- `show responder action`

パラメーター:

- **Name**: レスポンダーアクションの名前。最大長: 127
- **type**。レスポンダーアクションのタイプ。これは:(respondwith) とすることができます。

- ターゲット。何で応答するかを指定する式。
- **htmlpage**。HTML ページで応答することを指定するオプション。
- **hits**. アクションが実行された回数。
- **referenceCount**. アクションへの参照の数。
- **undefHits**. アクションが UNDEF になった回数。
- **comment**. このレスポンダーアクションに関するあらゆる種類の情報。
- **builtin**. レスポンダーアクションが組み込まれているかどうかを決定するフラグ。

例:

```
Create a responder action that displays a “Not Found” error page for URLs that do not exist:

> add responder action act404Error respondWith '"HTTP/1.1 404 Not Found\r\n\r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."'
Done

> show responder action

1) Name: act404Error
Operation: respondwith
Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

Create a responder action that displays a “Not Found” error page for URLs that do not exist:

add responder action act404Error respondWith '"HTTP/1.1 404 Not Found\r\n\r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."'
Done
> show responder action

1) Name: act404Error
Operation: respondwith
Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```


CLI を使用して既存のレスポンダーアクションを変更します。

コマンドプロンプトで次のコマンドを入力して、既存のレスポンダーアクションを変更し、構成を確認します。

- set responder action <name> -target <string>
- show responder action

例:

```
set responder action act404Error  -target '"HTTP/1.1 404 Not Found\r\n\
   r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web
   server."'
 Done
> show responder action

1)      Name: act404Error
        Operation: respondwith
        Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE +
            " does not exist on the web server."
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 0
 Done

<!--NeedCopy-->
```

CLI を使用してレスポンダーアクションを削除します。

コマンドプロンプトで次のコマンドを入力して、レスポンダーアクションを削除し、構成を確認します。

- rm responder action <name>
- show responder action

例:

```
rm responder action act404Error
 Done

> show responder action
 Done

<!--NeedCopy-->
```

CLI を使用して htmlpage レスポンダーアクションでレスポンスにカスタムヘッダーを追加します。

NetScaler ADC アプライアンスは、responsewithhtmlpage レスポンダーアクションでカスタムヘッダーで応答できるようになりました。最大 8 つのカスタムヘッダーを設定できます。以前は、アプライアンスはContent-type:text/htmlおよびContent-Length:<value>静的ヘッダーのみで応答していました。

> 注:
>
> カスタムヘッダー設定では、「Content-Type」ヘッダー値を上書きすることもできます。

コマンドプロンプトで、次のコマンドを入力します。

```
add responder action <name> <type> (<target> | <htmlpage>)[-comment
<string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <
expression>] [-headers <name(value)> ...]
```

各項目の意味は次のとおりです。

名前**:** レスポンダーアクションの名前。文字、数字、またはアンダースコア文字 (_) で始まり、文字、数字、およびハイフン (-)、ピリオド (.) ハッシュ (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア文字のみを含める必要があります。レスポンダーポリシーの追加後に変更できます。

タイプ**:** レスポンダーアクションのタイプ。使用可能な設定は次のように機能します。

1. **respondwith <target>** -ターゲットとして指定された式でリクエストに応答します。
2. **respondwithhtmlpage** -アップロードされた HTML ページオブジェクトをターゲットとして指定してリクエストに応答します。
3. **redirect** -ターゲットとして指定された URL にリクエストをリダイレクトします。
4. **sqlresponse_ok-SQL OK** レスポンスを送信します。
5. **sqlresponse_error** -SQL エラーレスポンスを送信します。これは必須の議論です。指定可能な値: noop、respondwith、redirect、respondwithhtmlpage、sqlresponse_ok、sqlresponse_error

ターゲット**:** 何で応答するかを指定する式。通常は、リダイレクトポリシーの URL またはデフォルトの構文式です。リクエスト内の情報を参照する NetScaler ADC のデフォルト構文式に加えて、文字列ビルダ式にはテキストと HTML、および新しい行や段落を定義する単純なエスケープコードを含めることができます。各ストリングビルダーの式要素（NetScaler のデフォルト構文式または文字列）を二重引用符で囲みます。プラス (+) 文字を使用して要素を結合します。

htmlpage:respondwithhtmlpage ポリシーの場合、レスポンスとして使用する HTML ページオブジェクトの名前。最初にページオブジェクトをインポートする必要があります。最大長: 31

コメント**:** このレスポンダーアクションに関するあらゆる種類の情報。最大長: 255

responseStatusCode: HTTP レスポンスステータスコード (200、302、404 など) redirect actionタイプの
デフォルト値は 302 で、respondwithhtmlpageは 200 です。最小値:100 最大値:599

理由フレーズ**:** HTTP 応答の理由フレーズを指定する式。理由句は、引用符付きの文字列リテラルまたは PI 式です。たとえば、次のようになります: "Invalid URL: "+ HTTP.REQ.URL Maximum Length: 8191

ヘッダー**:** HTTP レスポンスに挿入する 1 つ以上のヘッダー。各ヘッダーは"name(expr),"のように指定されます。expr は実行時に評価され、指定されたヘッダーの値を提供する式です。レスポンダーアクションには最大 8 つのヘッダーを設定できます。

GUI を使用してレスポンダーアクションを設定します。

1. **AppExpert >** レスポンダー **>** アクションに移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。

    - アクションを作成するには、[追加] をクリックします。
    - 既存のアクションを変更するには、アクションを選択し、[開く] をクリックします。

3. アクションを作成するか、既存のアクションを修正するかに応じて、「作成」または「**OK**」をクリックします。
4. [閉じる] をクリックします。ステータスバーに、機能が有効になっていることを示すメッセージが表示されます。
5. レスポンダーアクションを削除するには、アクションを選択し、[削除] をクリックします。ステータスバーに、機能が無効になったことを示すメッセージが表示されます。

[数式の追加] ダイアログボックスを使用して式を追加します

1. [レスポンダーアクションの作成] または [レスポンダーアクションの構成] ダイアログボックスで、[追加] をクリックします。
2. [式の追加] ダイアログボックスの最初のリストボックスで、式の最初の用語を選択します。

    - HTTP HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。
    - SYS。1 つ以上の保護された Web サイト。リクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
    - CLIENT. 要求を送信したコンピュータ。リクエストの送信者の側面を調べる場合は、これを選択します。
    - ANALYTICS. リクエストに関連付けられた分析データ。リクエストメタデータを調べる場合は、これを選択します。
    - SIP. SIP 要求。SIP 要求の一部の側面を調べる場合は、これを選択します。選択すると、右端のリストボックスに、式の次の部分に適した用語がリストされます。

3. 2 番目のリストボックスで、式の 2 番目の用語を選択します。選択肢は、前のステップで行った選択によって異なり、コンテキストに適切です。2 番目の選択を行った後、[式の構築] ウィンドウの下のヘルプウィンドウ (空白) に、選択した用語の目的と使用法を説明するヘルプが表示されます。
4. 式が終了するまで、前のリストボックスの右側に表示されるリストボックスから用語を選択するか、値の入力を求めるテキストボックスに文字列または数値を入力します。

グローバル **HTTP** アクションの設定

HTTP 要求がタイムアウトしたときにレスポンダアクションを呼び出すように、グローバル HTTP アクションを設定できます。この機能を設定するには、最初に呼び出すレスポンダーアクションを作成する必要があります。次に、そのレスポンダーアクションでタイムアウトに応答するようにグローバル HTTP タイムアウトアクションを設定します。

CLI を使用してグローバル HTTP アクションを設定します。

コマンドプロンプトで、次のコマンドを入力します。

- `set ns httpProfile -reqTimeoutAction <responder action name>`
- `save ns config`

`<responder action name>`については、レスポンダーアクションの名前を置き換えてください。

HTML ページのインポートの設定

NetScaler ADC アプライアンスがカスタムメッセージで応答すると、HTML ファイルで応答できます。`import responder htmlpage`コマンドを使用してファイルをインポートし、このファイルを`add responder action <act name> respondwithhtmlpage <file name>`コマンドで使用することができます。NetScaler GUI を使用してファイルをインポートすることもできます。目的の HTML ページをアプライアンスフォルダにインポートし、レスポンダの実行時にページをアップロードできます。

CLI を使用して HTML ページをインポートする

コマンドプロンプトで入力します。

```
import responder htmlpage [<src>] <name> [-comment <string>] [-
overwrite][-CAcertFile <string>]
```

例:

```
import responder htmlpage http://www.example.com/page.html my-responder
-page -CAcertFile my_root_ca_cert
```

ここで、
CA 証明書はクライアント証明書の検証に使用されます。証明書は `import ssl certfile` CLI コマンドを使用するか、API または GUI を使用して同等のコマンドをインポートする必要があります。証明書名が設定されていない場合、証明書の検証にはデフォルトのルート CA 証明書が使用されます。

ローカルファイルシステムから HTML ページをインポートする

ローカルファイルシステムから HTML ページをインポートすることもできます。インポートするには、SCP またはその他の方法でファイルを /var/tmp/ ディレクトリにコピーし、「local:」キーワードを使用してインポートします。次に例を示します:

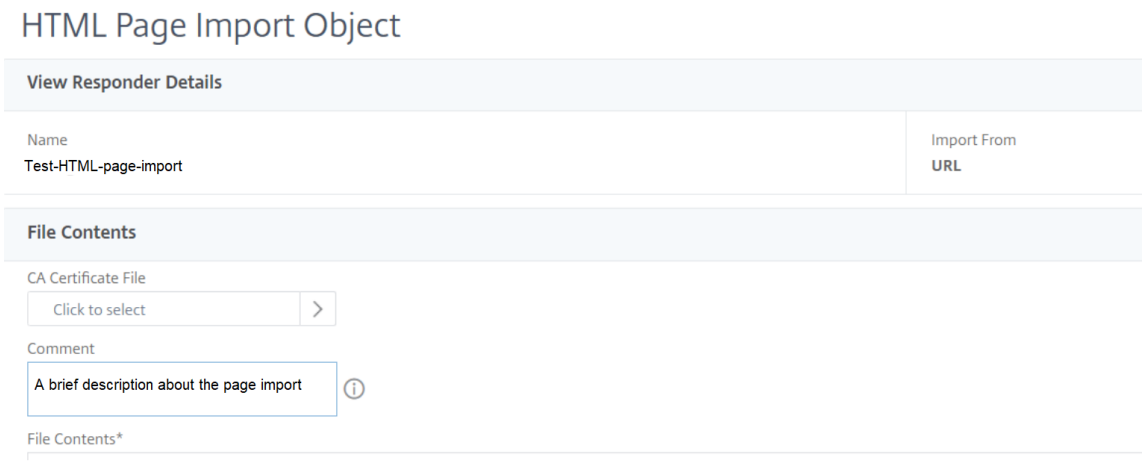
```
import responder htmlpage local:my_local_file.html my_local_file
```

`my_local_file.html`は「/var/tmp/」ディレクトリに存在します。

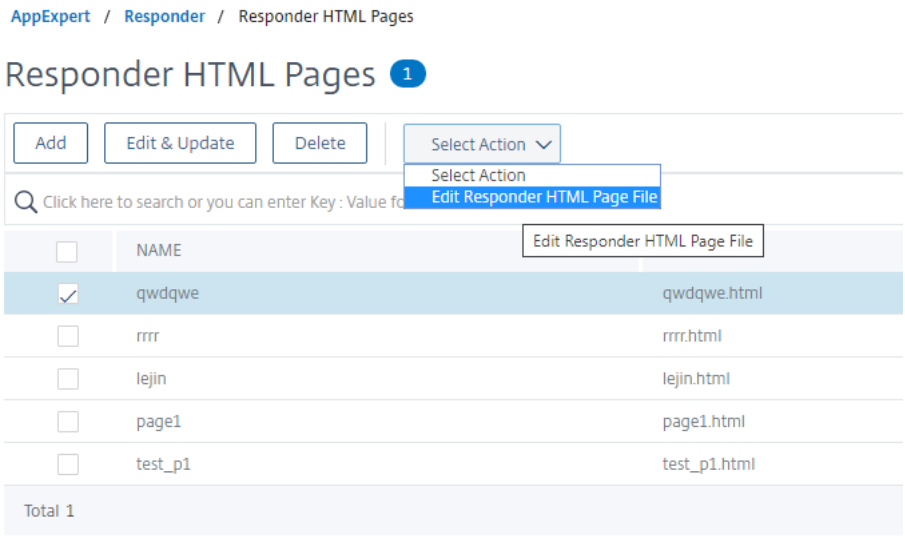
> 注意
>
> :「local:」キーワードは「/var/tmp/」ディレクトリ内のファイルのみを検索します。デフォルト以外のパーティションの場合は、/var/partitions/<partition name>/tmpにあるパーティション固有の tmp ディレクトリにファイルをコピーする必要があります。

GUI を使用して **HTML** ページをインポートする

1. [**AppExpert**] > [レスポンダー] > [**HTML** ページのインポート**]** に移動します。
2. [レスポンダー **HTML** のインポート] 詳細ウィンドウで、[追加] をクリックします。
3. **HTML** ページのインポートオブジェクトページで、次のパラメータを設定します。
   a) [名前]。HTML ページの名前。
   b) インポート元。ファイル、テキスト、またはテキストから読み込まれます。
   c) URL。HTML ファイルの URL の場所を入力するを選択します。
   d) [ファイル]。アプライアンスディレクトリから HTML ファイルを選択します。
   e) テキスト。HTML ファイルをテキストとして選択します。
4. [続行] をクリックします。
5. レスポンダーの HTML ページの詳細を確認します。
6. [完了] をクリックします。



HTML ページを編集するには、ファイルを選択し、[アクションの選択] ドロップダウンリストから [レスポンダー **HTML** ページファイルの編集] をクリックします。



レスポンダーポリシーの設定

August 29, 2023

レスポンダーアクションを構成したら、次にレスポンダーポリシーを構成して、NetScaler ADC アプライアンスが応答する要求を選択する必要があります。レスポンダーポリシーは、1 つ以上の式で構成されるルールに基づいています。ルールは、リクエストがルールに一致した場合に実行されるアクションに関連付けられます。

注：レスポンダーポリシーの作成と管理のために、GUI は NetScaler ADC コマンドプロンプトでは利用できない支援を提供します。

NetScaler ADC コマンドラインを使用してレスポンダーポリシーを構成するには：

コマンドプロンプトで入力します。

- `add responder policy <name> <expression> <action> [<undefaction>]-appFlowaction <actionName>`
- `show responder policy <name>`

例：

```
> add responder policyThree "CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)"
   RESET
 Done
> show responder policy policyThree

        Name: policyThree
        Rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
        Responder Action: RESET
        UndefAction: Use Global
```



```
        Hits: 0
        Undef Hits: 0
 Done
<!--NeedCopy-->
```

NetScaler ADC コマンドラインを使用して既存のレスポンダーポリシーを変更するには:

コマンドプロンプトで入力します。

- `set responder policy <name> [-rule <expression>] [-action <string>] [-undefAction <string>]`
- `show responder policy <name>`

コマンドラインを使用してレスポンダーポリシーを削除するには:

コマンドプロンプトで入力します。

- `rm responder policy <name>`
- `show responder policy`

例:

```
>rm responder policy pol404Error
 Done

> show responder policy
 Done
<!--NeedCopy-->
```

GUI を使用してレスポンダーポリシーを設定するには、次の手順を実行します。

1. [**AppExpert] > [レスポンダー] **[ポリシー] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - ポリシーを作成するには、[**Add**] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。
3. ポリシーを作成するか、既存のポリシーを変更するかに応じて、「作成」または「**OK**」をクリックします。
4. [閉じる] をクリックします。ステータスバーに、機能が設定されたことを示すメッセージが表示されます。

レスポンダーポリシーのバインド

August 15, 2023

ポリシーを有効にするには、NetScaler ADC を通過するすべてのトラフィックに適用されるようにグローバルにバインドするか、特定の仮想サーバーにバインドして、宛先 IP アドレスがその仮想サーバーの VIP である要求にのみポリシーを適用するようにする必要があります。

ポリシーをバインドするときは、そのポリシーにプライオリティを割り当てます。プライオリティによって、定義したポリシーが評価される順序が決まります。優先度は、任意の正の整数に設定できます。

NetScaler ADC オペレーティングシステムでは、ポリシーの優先順位は逆の順序で機能します。数値が大きいほど優先度は低くなります。たとえば、プライオリティが 10、100、1000 の 3 つのポリシーがある場合、プライオリティ 10 が割り当てられたポリシーが最初に実行され、次にポリシーにプライオリティ 100 が割り当てられ、最後にポリシーにオーダー 1000 が割り当てられます。レスポンダー機能では、リクエストが最初に一致するポリシーのみが実装され、一致する可能性のある追加のポリシーは実装されないため、意図した結果を得るにはポリシーの優先度が重要です。

ポリシーをグローバルにバインドするときに、各ポリシーの間隔を 50 または 100 に設定して優先度を設定することで、他のポリシーを任意の順序で追加する余地を十分に確保し、必要な順序で評価するように設定できます。これにより、既存のポリシーの優先度を再割り当てしなくても、いつでもポリシーを追加できます。

NetScaler ADC でのポリシーのバインドの詳細については、「ポリシーと式」を参照してください。

> 注:
>
> レスポンダーポリシーは TCP ベースの仮想サーバーにバインドされます。

NetScaler ADC コマンドラインを使用してレスポンダーポリシーをグローバルにバインドするには:

コマンドプロンプトで次のコマンドを入力して、レスポンダーポリシーをグローバルにバインドし、構成を確認します。

- `bind responder global <policyName> <priority> [<gotoPriorityExpression> [-type <type>] [-invoke (<labelType> <labelName>)]`
- `show responder global`

例:

```
1 > bind responder global poliError 100
2  Done
3 > show responder global
4 1)      Global bindpoint: REQ_DEFAULT
5         Number of bound policies: 1
6
7  Done
8 <!--NeedCopy-->
```

NetScaler ADC コマンドラインを使用してレスポンダーポリシーを特定の仮想サーバーにバインドするには:

コマンドプロンプトで入力します。

- `bind lb vserver <name> -policyname <policy_name> -priority <priority>`
- `show lb vserver vs-loadbal <name>`

例:


```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver
1)      vs-loadbal (10.102.29.20:80) - HTTP     Type: ADDRESS
        State: OUT OF SERVICE
        Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
        Time since last state change: 2 days, 00:58:03.260
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
2)      vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
        State: DOWN
        Last state change was at Wed Aug 19 10:03:46 2009 (+213 ms)
        Time since last state change: 2 days, 00:00:04.260
        Effective State: DOWN
        Client Idle Timeout: 9000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED
 Done
<!--NeedCopy-->
```

GUI を使用してレスポンダーポリシーをグローバルにバインドするには、次の手順を実行します。

1. **AppExpert> **** レスポンダー **>** ポリシーに移動します。**
2. [レスポンダーポリシー] ページで、レスポンダーポリシーを選択し、[ポリシーマネージャー] をクリックします。
3. [レスポンダーポリシーマネージャー] ダイアログボックスの [バインドポイント] メニューで、[既定のグローバル] を選択します。
4. [**Insert Policy**] をクリックして新しい行を挿入し、すべての非バインドレスポンダーポリシーのドロップダウンリストを表示します。
5. リストにあるポリシーの 1 つをクリックします。このポリシーは、グローバルにバインドされたレスポンダーポリシーの一覧に挿入されます。
6. [変更を適用] をクリックします。
7. [閉じる] をクリックします。ステータスバーに、構成が正常に完了したことを示すメッセージが表示されます。

GUI を使用してレスポンダーポリシーを特定の仮想サーバーにバインドするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. [負荷分散仮想サーバー] ページで、レスポンダーポリシーをバインドする仮想サーバーを選択し、[開く] をクリックします。
3. ［仮想サーバーの構成（負荷分散）］ダイアログボックスで、［ポリシー］タブを選択します。このタブには、NetScaler ADC アプライアンスに構成されているすべてのポリシーの一覧が表示されます。
4. この仮想サーバにバインドするポリシーの名前の横にあるチェックボックスをオンにします。
5. ［**OK**］をクリックします。ステータスバーに、構成が正常に完了したことを示すメッセージが表示されます。

レスポンダーポリシーのデフォルトアクションの設定

August 15, 2023

NetScaler アプライアンスは、リクエストがレスポンダーポリシーと一致しない場合、未定義イベント（UNDEF イベント）を生成します。次に、アプライアンスは未定義のイベントに割り当てられたデフォルトのアクションを実行します。デフォルトでは、アクションは負荷分散、コンテンツフィルタリングなどの次の機能にリクエストを転送します。このデフォルトの動作により、リクエストを特定のレスポンダーアクションをウェブサーバーに送信する必要がなくなります。また、クライアントは要求したコンテンツにアクセスできます。

ただし、NetScaler アプライアンスが保護する 1 つ以上の Web サイトが多数の無効または悪意のある要求を受け取った場合は、デフォルトのアクションをクライアント接続をリセットするか、要求を取り下げるように変更したい場合があります。このタイプの構成では、正当なリクエストと一致するレスポンダーポリシーを 1 つ以上作成し、それらのリクエストを元の宛先にリダイレクトするだけです。その後、NetScaler アプライアンスは、設定したデフォルトアクションで指定された他のリクエストをすべてブロックします。

未定義のイベントには、次のアクションのいずれかを割り当てることができます。

- ヌープ。NOOP アクションは応答側の処理を中止しますが、パケットフローは変更しません。これにより、アプライアンスはどのレスポンダーポリシーにも一致しない要求を引き続き処理し、別の機能が介入して要求をブロックまたはリダイレクトしない限り、最終的には要求された URL に転送します。このアクションは Web サーバーへの通常のリクエストに適しており、デフォルト設定です。

- リセット。未定義のアクションが RESET に設定されている場合、アプライアンスはクライアント接続をリセットし、Web サーバーとのセッションを再確立する必要があることをクライアントに通知します。このアクションは、存在しないウェブページを繰り返しリクエストする場合や、保護されている Web サイトをハッキングまたは調査しようとする可能性のある接続に対しても適切です。

- ドロップ。未定義のアクションが DROP に設定されている場合、アプライアンスはクライアントにまったく応答せずに要求をサイレントにドロップします。このアクションは、サーバーに対する DDoS 攻撃やその他の持続的な攻撃の一部であると思われるリクエストに適しています。

注**:** UNDEF イベントは、クライアントのリクエストに対してのみトリガーされます。応答に対して UNDEF イベントはトリガーされません。

NetScaler コマンドラインを使用して未定義のアクションを設定するには:

コマンドプロンプトで次のコマンドを入力して、未定義のアクションを設定し、構成を確認します。

- `set responder param -undefAction (RESET|DROP|NOOP)[-timeout <msecs>]`
- `show responder param`

各項目の意味は次のとおりです。

タイムアウト-すべてのポリシーと選択したアクションを中断なく処理できる最大時間（ミリ秒単位）。タイムアウトになると、評価によって UNDEF が生成され、それ以上の処理は実行されません。

最小値:1

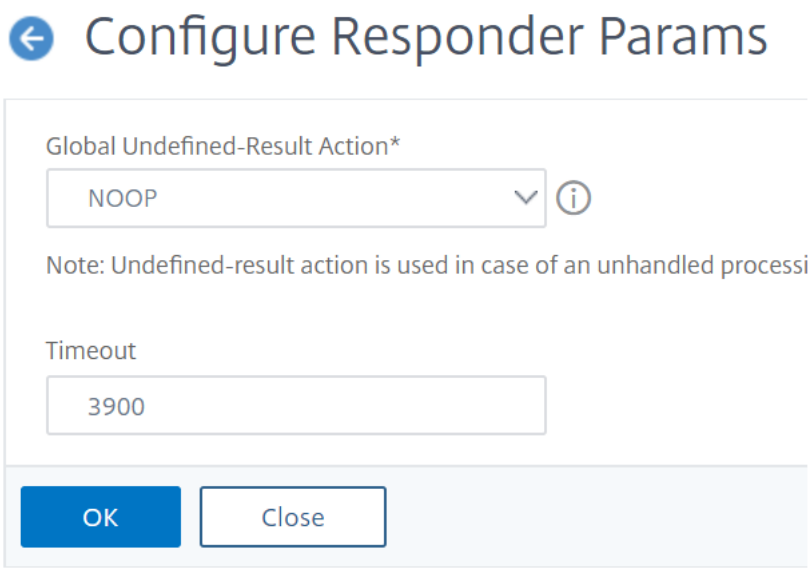
最大値:5000

例**:**

```
>set responder param -undefAction RESET -timeout 3900
 Done
> show responder param
Action Name: RESET
Timeout: 3900
 Done
>
<!--NeedCopy-->
```

GUI を使用して未定義のアクションを設定します

1. [**AppExpert**] > [レスポンダー] に移動し、[設定] の [レスポンダー設定の変更] リンクをクリックします。
2. 「レスポンダーパラメーターの設定」ページで、次のパラメーターを設定します。
   a) グローバル未定義結果アクション。レスポンダーのポリシーとアクションで未処理の処理例外が発生する場合は、未定義の結果のアクションが優先されます。[**NOOP**]、[リセット]、または [ドロップ] を選択します。
   b) タイムアウト。すべてのポリシーと選択したアクションを中断なく処理できる最大時間（ミリ秒単位）。タイムアウトになると、評価によって UNDEF が生成され、それ以上の処理は実行されません。
3. [**OK**] をクリックします。



。

レスポンダーのアクションとポリシーの例

August 15, 2023

レスポンダーのアクションとポリシーは強力で複雑ですが、比較的単純なアプリケーションから始めることができます。

例**:** 指定された **IP** からのアクセスのブロック

次の手順では、CIDR 222.222.0.0/16 から送信されたクライアントによる保護された Web サイトへのアクセスをブロックします。レスポンダは、要求された URL へのアクセスがクライアントに許可されていないことを示すエラーメッセージを送信します。

NetScaler ADC コマンドラインを使用してアクセスをブロックするには:

コマンドプロンプトで、次のコマンドを入力してアクセスをブロックします。

- add responder action act_unauthorized respond with “HTTP/1.1 403 Forbidden\r\n\r\n” + “Client: “+ CLIENT.IP.SRC + “is not authorized to access URL:”+ “HTTP.REQ.URL.HTTP_URL_SAFE”’
- add responder policy pol_un “CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)” act_unauthorized
- bind responder global pol_un 10

GUI を使用してアクセスをブロックするには、次の手順を実行します。

1. ナビゲーションウィンドウで [レスポンダー] を展開し、[アクション] をクリックします。
2. 詳細ペインで、[追加] をクリックします。

3. [レスポンダーアクションの作成] ダイアログボックスで、次の操作を行います。

   a) [名前] テキストボックスに act_unauthorized と入力します。
   b) [種類] で、[返信方法] を選択します。
   c) [ターゲット] テキスト領域に、次の文字列を入力します。「HTTP/1.1 403 禁止された\ r\ n\ r\ n」+「クライアント:” + CLIENT.IP.SRC + “は URL にアクセスする権限がありません:」+ HTTP.REQ.URL.HTTP_URL_SAFE
   d) [作成] をクリックし、[閉じる] をクリックします。
      構成した act_unauthorized という名前のレスポンダーアクションが [レスポンダーアクション] ページに表示されます。

4. ナビゲーションペインで、[**Policies**] をクリックします。
5. 詳細ペインで、[追加] をクリックします。
6. [レスポンダーポリシーの作成] ダイアログボックスで、次の操作を行います。

   a) [名前] テキストボックスに pol_unauthorized と入力します。
   b) [アクション] で [act_unauthorized] を選択します。
   c) [式] ウィンドウで、次のルールを入力します。CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)
   d) [作成] をクリックし、[閉じる] をクリックします。
      構成した pol_unauthorized という名前のレスポンダーポリシーが [レスポンダーポリシー] ページに表示されます。

7. レスポンダーポリシーのバインドで説明されているように、新しいポリシー pol_unauthorized をグローバルにバインドします。

例**:** クライアントを新しい **URL** にリダイレクトする

次の手順では、CIDR 222.222.0.0/16 内から保護された Web サイトにアクセスするクライアントを指定された URL にリダイレクトします。

NetScaler コマンドラインを使用してクライアントをリダイレクトするには:

コマンドプロンプトで次のコマンドを入力してクライアントをリダイレクトし、構成を確認します。

- add responder action act_redirect redirect "<http://www.example.com/404.html>"
- show responder action act_redirect
- add responder policy pol_redirect “CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)” act_redirect
- show responder policy pol_redirect
- bind responder global pol_redirect 10

例**:**

```
1 > add responder action act_redirect redirect `" http ://www.example.com
     /404.html "`
2 Done
```



```
3
4 > add responder policy pol_redirect "CLIENT.IP.SRC.IN_SUBNET
     (222.222.0.0/16)" act_redirect
5 Done
6 <!--NeedCopy-->
```

GUI を使用してクライアントをリダイレクトするには、次の手順を実行します。

1. **[**AppExpert**] > [レスポンダー] > [アクション] に移動します。**
2. 詳細ペインで、[追加] をクリックします。
3. [レスポンダーアクションの作成] ダイアログボックスで、次の操作を行います。

    a) [名前] テキストボックスに act_redirect と入力します。
    b) [タイプ] で [リダイレクト] を選択します
    c) [ターゲット] テキスト領域に、次の文字列を入力します。"<http://www.example.com/404.html>"
    d) [作成] をクリックし、[閉じる] をクリックします。
    act_redirect という名前のレスポンダーアクションが [レスポンダーアクション] ページに表示されます。

4. ナビゲーションペインで、[**Policies**] をクリックします。
5. 詳細ペインで、[追加] をクリックします。
6. [レスポンダーポリシーの作成] ダイアログボックスで、次の操作を行います。

    a) [名前] テキストボックスに pol_redirect と入力します。
    b) [アクション] で [act_redirect] を選択します。
    c) [式] ウィンドウで、次のルールを入力します。CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)
    d) [作成] をクリックし、[閉じる] をクリックします。
    構成した pol_redirect という名前のレスポンダーポリシーが [レスポンダーポリシー] ページに表示されます。

7. レスポンダーポリシーのバインドの説明に従って、新しいポリシー pol_redirect をグローバルにバインドします。

レスポンダーの **Diameter** サポート

August 15, 2023

レスポンダー機能が Diameter プロトコルをサポートするようになりました。HTTP および TCP リクエストと同様に Diameter リクエストに応答するようにレスポンダーを構成できます。たとえば、特定の Diameter オリジンからのリクエストに応答して、モバイルデバイス向けに拡張された Web ページへのリダイレクトを行うように Responder を設定できます。Diameter ameter ヘッダーと属性と値のペア（AVP）の検査をサポートする

NetScaler 式が多数追加されました。これらの表現は、インデックス、ID、または名前による特定の AVP の検索、各 AVP の情報の確認、および適切な応答の送信をサポートします。

Diameter リクエストに応答するようにレスポンダーを設定するには:

コマンドプロンプトで、次のコマンドを入力します。

- `add responder action <actname> RESPONDWITH "DIAMETER.NEW_REDIRECT (\"aaa://host.example.com\")"`

  <actname>は、新しいアクションの名前に置き換えてください。名前は 1 ～127 文字で、文字、数字、ハイフン (-) と下線 (_) を使用できます。aaa://host.example.com、接続のリダイレクト先となる Diameter ホストの URL に置き換えてください。

- add responder policy <polname> “diameter.req.avp(264).value.eq(“host1.example.net”)” <actname>

  <polname>は、新しいポリシーの名前に置き換えてください。<actname>と同様に、名前は 1 ～127 文字で、文字、数字、ハイフン (-) とアンダースコア (_) 記号を使用できます。host1.example.net には、リダイレクトするリクエストの発信元ホストの名前を代入します。<actname>は、先ほど作成したアクションの名前に置き換えてください。

- `bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST`

  <vservername>、ポリシーをバインドする負荷分散仮想サーバーの名前に置き換えてください。<polname>、作成したポリシーの名前に置き換えてください。<priority>については、ポリシーの代わりに優先度を使用してください。

例:

「host1.example.net」から送信された Diameter ameter リクエストに「host.example.com」にリダイレクトして応答するレスポンダーアクションとポリシーを作成するには、次のアクションとポリシーを追加し、図のようにポリシーをバインドできます。

```
1 > add responder action act_resp-dm-redirect RESPONDWITH "DIAMETER.
     NEW_REDIRECT("aaa://host.example.com")"
2 Done
3
4 > add responder pol_resp-dm-redirect "diameter.req.avp(264).value.eq("
     host1.example.net")" act_resp-dm-redirect
5 Done
6
7 > bind lb vserver vs1 -policyName pol_resp-dm-redirect -priority 10 -
     type REQUEST
8 Done
9 <!--NeedCopy-->
```


レスポンダーの **RADIUS** サポート

August 15, 2023

NetScaler の式言語には、RADIUS 要求から情報を抽出したり、RADIUS 要求を操作したりできる式が含まれています。これらの式により、レスポンダー機能を使用して RADIUS 要求に応答できます。レスポンダーのポリシーとアクションには、RADIUS リクエストに適切な、または関連する任意の表現を使用できます。使用可能な式を使用すると、RADIUS メッセージタイプを識別し、接続から任意の属性値ペア（AVP）を抽出し、その情報に基づいてさまざまな応答を送信できます。RADIUS 接続のすべての応答側ポリシーを呼び出すポリシーラベルを作成することもできます。

RADIUS 式を使用すると、要求の送信先の RADIUS サーバとの通信を必要としない単純な応答を作成できます。レスポンダーポリシーが接続と一致すると、NetScaler は RADIUS 認証サーバーに接続せずに適切な RADIUS レスポンスを作成して送信します。たとえば、RADIUS リクエストの送信元 IP アドレスがレスポンダーポリシーで指定されているサブネットからのものである場合、NetScaler はそのリクエストにアクセス拒否メッセージで応答することも、単にリクエストをドロップすることもできます。

ポリシーラベルを作成して、特定のタイプの RADIUS 要求を、それらの要求に適した一連のポリシーを通じてルーティングすることもできます。

注**:** 現在の RADIUS 式は RADIUS IPv6 属性では機能しません。

RADIUS をサポートする表現に関する NetScaler のドキュメントでは、RADIUS 通信の基本的な構造と目的に精通していることを前提としています。RADIUS の詳細については、RADIUS サーバのマニュアルを参照するか、RADIUS プロトコルの概要をオンラインで検索してください。

RADIUS のレスポンダポリシーの設定

以下の手順では、NetScaler コマンドラインを使用してレスポンダーのアクションとポリシーを構成し、ポリシーを RADIUS 固有のグローバルバインドポイントにバインドします。

レスポンダーアクションとポリシーを設定し、ポリシーをバインドするには:

コマンドプロンプトで、次のコマンドを入力します。

- `add responder action <actName> <actType>`
- `add responder policy <polName> <rule> <actName>`
- `bind responder policy <polName> <priority> <nextExpr> -type <bindPoint>` ここで、`<bindPoint>`は RADIUS 固有のグローバルバインドポイントの 1 つを表します。

レスポンダ用の **RADIUS** エクスプレッション

レスポンダー構成では、次の NetScaler 表現を使用して RADIUS 要求のさまざまな部分を参照できます。

接続タイプの識別**:**

- RADIUS.IS_CLIENT. 接続が RADIUS クライアント (要求) メッセージの場合は TRUE を返します。
- RADIUS.IS_SERVER. 接続が RADIUS サーバー (応答) メッセージの場合、TRUE を返します。

リクエスト表現**:**

- RADIUS.REQ.CODE. RADIUS リクエストタイプに対応する番号を返します。num_at クラスの派生関数です。たとえば、RADIUS アクセス要求は 1 を返します。RADIUS アカウンティング要求では 4 が返されます。
- RADIUS.REQ.LENGTH. ヘッダーを含む RADIUS リクエストの長さを返します。num_at クラスの派生関数です。
- RADIUS.REQ.IDENTIFIER. RADIUS リクエスト識別子を返します。これは、リクエストを対応するレスポンスと一致させるために各リクエストに割り当てられる番号です。num_at クラスの派生関数です。
- RADIUS.REQ.AVP(<AVP Code No>).VALUE。この AVP が最初に出現したときの値を text_t 型の文字列で返します。
- RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)。AVP の指定されたインスタンスを RAVP_t 型の文字列として返します。特定の RADIUS AVP が 1 つの RADIUS メッセージに複数回表示されることがあります。INSTANCE (0) は最初のインスタンスを返し、INSTANCE (1) は 2 番目のインスタンスを返す、というように、最大 16 個のインスタンスを返します。
- RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)。AVP の指定されたインスタンスの値を text_t 型の文字列として返します。
- RADIUS.REQ.AVP(<AVP code no>).COUNT。RADIUS 接続内の特定の AVP のインスタンス数を整数で返します。
- RADIUS.REQ.AVP(<AVP code no>).EXISTS。指定されたタイプの AVP がメッセージに存在する場合は TRUE を返し、存在しない場合は FALSE を返します。

レスポンス表現**:**

RADIUS レスポンス式は RADIUS リクエスト表現と同じですが、REQ が REQ に置き換わっている点が異なります。

AVP 値のタイプキャスト**:**

ADC は、RADIUS AVP 値をテキスト、整数、符号なし整数、ロング、符号なしロング、ipv4 アドレス、ipv6 アドレス、ipv6 プレフィックス、および時刻データ型にタイプキャストする式をサポートしています。構文は他の NetScaler タイプキャスト式と同じです。

例**:**

ADC は、RADIUS AVP 値をテキスト、整数、符号なし整数、ロング、符号なしロング、ipv4 アドレス、ipv6 アドレス、ipv6 プレフィックス、および時刻データ型にタイプキャストする式をサポートしています。構文は他の NetScaler タイプキャスト式と同じです。

```
1 RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at
```



```
<!--NeedCopy-->
```

AVP タイプエクスプレッション**:**

NetScaler は、RFC2865 および RFC2866 で説明されている割り当てられた整数コードを使用して RADIUS AVP 値を抽出する式をサポートしています。テキストエイリアスを使用して同じタスクを実行することもできます。次にいくつかの例を示します。

- RADIUS.REQ.AVP (1).VALUE or RADIUS.REQ.USERNAME.value. RADIUS ユーザー名の値を抽出します。
- RADIUS.REQ.AVP (4). VALUE or RADIUS.REQ. ACCT_SESSION_ID.value. メッセージから ACCT セッション ID AVP (コード 44) を抽出します。
- RADIUS.REQ.AVP (26). VALUE or RADIUS.REQ.VENDOR_SPECIFIC.VALUE. ベンダー固有の値を抽出します。

最も一般的に使用される RADIUS AVP の値も同じ方法で抽出できます。

RADIUS バインドポイント**:**

RADIUS 表現を含むポリシーには、4 つのグローバルバインドポイントを使用できます。

- RADIUS_REQ_OVERRIDE. プライオリティ/オーバーライドリクエストポリシーキュー。
- RADIUS_REQ_DEFAULT. 標準リクエストポリシーキュー。
- RADIUS_RES_OVERRIDE. プライオリティ/オーバーライドレスポンスポリシーキュー。
- RADIUS_RES_DEFAULT. 標準レスポンスポリシーキュー。

RADIUS レスポンダ固有の表現**:**

- RADIUS_RESPONDWITH. 指定した RADIUS 応答で応答します。応答は、RADIUS 式とその他の該当する式の両方を含む NetScaler 式で作成されます。
- RADIUS.NEW_ANSWER. 新しい RADIUS 応答をユーザに送信します。
- RADIUS.NEW_ACCESSREJECT. RADIUS 要求を拒否します。
- RADIUS.NEW_AVP. 指定された新しい AVP をレスポンスに追加します。

使用例

レスポンダ付きの RADIUS の使用例は次のとおりです。

特定のネットワークからの **RADIUS** 要求のブロック

特定のネットワークからの認証リクエストをブロックするようにレスポンダー機能を設定するには、まずリクエストを拒否するレスポンダーアクションを作成します。ブロックするネットワークからのリクエストを選択するポリシーでアクションを使用してください。以下を指定して、レスポンダポリシーを RADIUS 固有のグローバルバインドポイントにバインドします。

- 優先順位
- NextExpr の値として END を指定すると、このポリシーに一致するとポリシー評価が停止します。
- ポリシーを割り当てるキューに RADIUS_REQ_OVERRIDE を指定します。これにより、ポリシーがデフォルトキューに割り当てられる前に評価されます。

特定のネットワークからのログオンをブロックするようにレスポンダーを設定するには **

- `add responder action <actName> <actType>`
- `add responder policy <polName> <rule> <actName>`
- `bind responder global <polName> <priority> <nextExpr> -type <bindPoint>`

例:

```
> add responder action rspActRadiusReject respondwith radius.
   new_accessreject
Done

> add responder policy rspPolRadiusReject client.ip.src.in_subnet
   (10.224.85.0/24) rspActRadiusReject
Done

> bind responder global rspPolRadiusReject 1 END -type
   RADIUS_REQ_OVERRIDE
<!--NeedCopy-->
```

レスポンダー機能の **DNS** サポート

August 15, 2023

HTTP および TCP 要求と同様に DNS 要求に応答するようにレスポンダー機能を設定できます。たとえば、UDP 経由で DNS 応答を送信し、クライアントからの DNS 要求が TCP 経由で送信されるように設定できます。リクエスト内の DNS ヘッダーの検査をサポートする NetScaler の表現は多数あります。これらの式は特定のヘッダーフィールドを調べ、適切な応答を送信します。

- **DNS** エクスプレッション。レスポンダー構成では、次の NetScaler 表現を使用して DNS リクエストのさまざまな部分を参照できます。

式	説明
DNS.NEW_RESPONSE	リクエストに基づいて新しい空の DNS レスポンスを作成します。
DNS.NEW_RESPONSE <AA, TC, rcode>	指定されたパラメータに基づいて新しい DNS レスポンスを作成します。

- **DNS** バインドポイント。DNS 表現を含むポリシーでは、次のグローバルバインドポイントを使用できます。

バインドポイント	説明
DNS_REQ_OVERRIDE	プライオリティ/オーバーライドリクエストポリシーキュー。
DNS_REQ_DEFAULT	標準リクエストポリシーキュー。

デフォルトのバインドポイントに加えて、DNS タイプのポリシーラベルを作成し、それらに DNS ポリシーをバインドできます。

DNS のレスポンダーポリシーの設定

以下の手順では、NetScaler コマンドラインを使用してレスポンダーのアクションとポリシーを構成し、ポリシーをレスポンダー固有のグローバルバインドポイントにバインドします。

DNS リクエストに応答するようにレスポンダーを設定するには:

コマンドプロンプトで、次のコマンドを入力します。

1. `add responder action <actName> <actType>`

   <actname>は、新しいアクションの名前に置き換えてください。名前の長さは 1 ～127 文字で、文字、数字、ハイフン (-)、および下線 (_) 記号を使用できます。<*actType*>については、レスポンダーアクションタイプを *RespondWith* に置き換えてください。

2. `add responder policy <polName> <rule> <actName>`

   <polname>は、新しいポリシーの名前に置き換えてください。<actname>では、名前の長さは 1 ～127 文字で、文字、数字、ハイフン (-)、および下線 (_) 記号を使用できます。<actname>は、先ほど作成したアクションの名前に置き換えてください。

3. `bind responder policy <polName> <priority> <nextExpr> -type <bindPoint>`

   <bindPoint>には、レスポンダー固有のグローバルバインドポイントのいずれかを指定します。<polName>は、先ほど作成したポリシーの名前の代わりに使用してください。<priority>には、ポリシーの優先度を指定します。

設定例**-**すべての **DNS** 要求を **TCP** 経由で強制します。

すべての DNS リクエストを TCP 経由で強制するには、TC ビットと rcode を NOERROR に設定するレスポンダーアクションを作成します。

```
1 > add responder action resp_act_set_tc_bit respondwith DNS.NEW_RESPONSE
    (true, true, NOERROR)
```



```
Done

> add responder policy enforce_tcp  dns.REQ.TRANSPORT.EQ(udp)
   resp_act_set_tc_bit
Done

>bind lb vserver dns_udp  - policyName  enforce_tcp   -type request  -
   priority 100
Done
<!--NeedCopy-->
```

レスポンダーの **MQTT** サポート

August 15, 2023

レスポンダー機能は MQTT プロトコルをサポートします。受信した MQTT メッセージのパラメータに基づいてアクションを実行するようにレスポンダーポリシーを設定できます。

アクションは、新しい接続に対して次のいずれかで応答します。

- DROP
- RESET
- NOOP
- 新しい MQTT CONNACK レスポンスを開始するためのレスポンダーアクション。

MQTT のレスポンダーポリシーの設定

レスポンダー機能を有効にしたら、MQTT リクエストを処理するための 1 つ以上のアクションを設定する必要があります。次に、レスポンダーポリシーを設定します。レスポンダーポリシーをグローバルにバインドすることも、特定の負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにバインドすることもできます。

レスポンダーポリシーをグローバルにバインドするには、次のバインドポイントを使用できます。

- MQTT_REQ_DEFAULT
- MQTT_REQ_OVERRIDE
- MQTT_JUMBO_REQ_DEFAULT
- MQTT_JUMBO_REQ_OVERRIDE

レスポンダーポリシーをコンテンツスイッチングまたは負荷分散仮想サーバーにバインドするには、次のバインドポイントを使用できます。

- REQUEST
- MQTT_JUMBO_REQ (このバインドポイントはジャンボパケットにのみ使用されます)

CLI を使用して MQTT リクエストに応答するようにレスポンダーを設定するには

コマンドプロンプトで、次のコマンドを入力します。

レスポンダーアクションを設定します。

```
add responder action <actName> <actType>
<!--NeedCopy-->
```

- actnameは、新しいアクションの名前に置き換えてください。名前の長さは 1 ～127 文字で、文字、数字、ハイフン (-)、および下線 (_) 記号を使用できます。
- actTypeについては、レスポンダーアクションタイプを respond で置き換えてください。

例:

```
add responder action mqtt_connack_unsup_ver respondwith MQTT.
   NEW_CONNACK(132)
<!--NeedCopy-->
```

レスポンダーポリシーを設定します。NetScaler アプライアンスは、このレスポンダーポリシーで選択された MQTT リクエストに応答します。

```
add responder policy <polName> <rule> <actname>
<!--NeedCopy-->
```

- polnameは、新しいポリシーの名前に置き換えてください。
- actnameは、作成したアクションの名前の代わりに使用してください。

例:

```
add responder policy reject_lower_version "MQTT.HEADER.COMMAND.EQ(
   CONNECT) && MQTT.VERSION.LT(3)" mqtt_connack_unsup_ver
<!--NeedCopy-->
```

レスポンダーポリシーを特定の負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにバインドします。このポリシーは、宛先 IP アドレスがその仮想サーバーの VIP である MQTT リクエストにのみ適用されます。

```
bind lb vserver <name> -policyName <policy_name> -priority <priority>

bind cs vserver <name> -policyName <policy_name> -priority <priority>
<!--NeedCopy-->
```

- policy_nameは、作成したポリシーの名前の代わりに使用してください。
- priorityには、ポリシーの優先度を指定します。

例:

```
bind lb vserver lb1 -policyName reject_lower_version -priority 50

```



```
bind cs vserver mqtt_frontend_cs -policyName reject_lower_version -
    priority 5
<!--NeedCopy-->
```

使用例 **1:** ユーザー名またはクライアント **ID** に基づいてクライアントをフィルタリングする

管理者は、MQTT CONNECT メッセージ内のユーザー名またはクライアント ID に基づいて接続を拒否するように MQTT レスポンダーポリシーを設定できます。

クライアント **ID** に基づいてクライアントをフィルタリングするための設定例

```
add policy patset filter_clients
bind policy patset filter_clients client1

add responder action mqtt_connack_invalid_client respondwith MQTT.
    NEW_CONNACK(2)

add responder policy reject_clients "MQTT.HEADER.COMMAND.EQ(CONNECT) &&
     mqtt.connect.clientid.equals_any("filter_clients")"
    mqtt_connack_invalid_client

bind cs vserver mqtt_frontend_cs -policyName reject_clients -priority 5
<!--NeedCopy-->
```

使用例 **2:** ジャンボパケットを処理するための **MQTT** メッセージの最大メッセージ長を制限する

管理者は、メッセージの長さが特定のしきい値を超えた場合にクライアント接続を切断するように MQTT レスポンダーポリシーを設定したり、要件に基づいて必要なアクションを実行したりできます。

ジャンボパケットを処理するには、次のルールパターンのいずれかを含むレスポンダーポリシーがジャンボバインドポイントにバインドされます。

- MQTT.MESSAGE_LENGTH
- MQTT.COMMAND
- MQTT.FROM_CLIENT
- MQTT.FROM_SERVER

ジャンボバインドポイントにバインドされたポリシーは、ジャンボパケットについてのみ評価されます。

MQTT メッセージの最大メッセージ長を制限する設定例

```
set lb parameter -dropmqttjumbomessage no

```



```
3 add responder policy drop_large_message MQTT.MESSAGE_LENGTH.GT(100000)
     reset
4
5 bind cs vserver mqtt_frontend_cs -policyName drop_large_message -
     priority 10
6 <!--NeedCopy-->
```

この例では、dropmqttjumbomessage パラメータは NO に設定されています。したがって、ADC アプライアンスは、長さが 64,000 バイトを超え、1,000,000 バイト未満の長さのメッセージを処理します。長さが 1,00,000 バイトを超えるメッセージはリセットされます。

レスポンダーを使用して **HTTP** リクエストを **HTTPS** にリダイレクトする方法

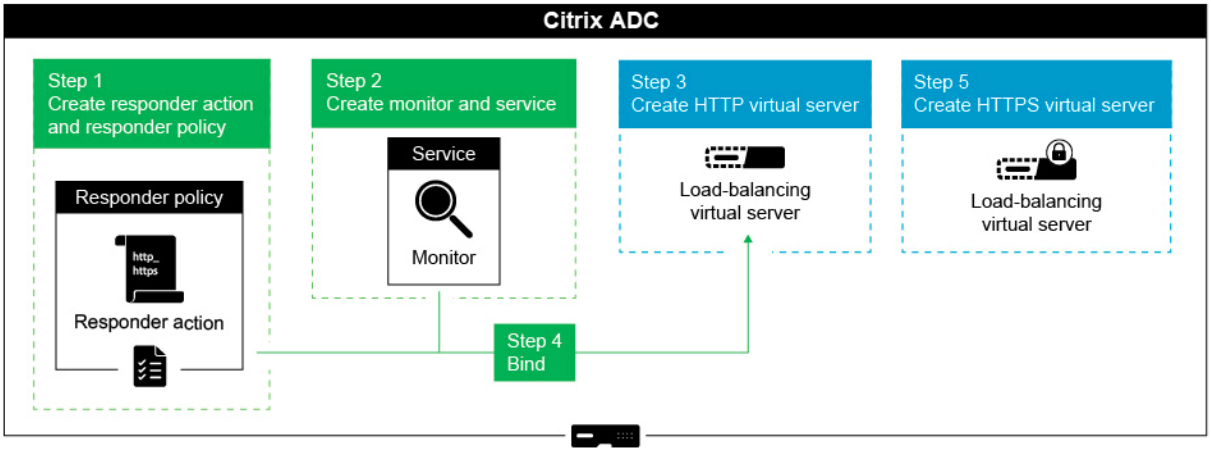
January 9, 2024

この記事では、負荷分散仮想サーバーの IP アドレスを使用してレスポンダー機能を構成し、クライアント要求を HTTP から HTTPS にリダイレクトする方法について説明します。

ユーザーが HTTP リクエストを送信して安全なウェブサイトにアクセスしようとするシナリオを考えてみましょう。リクエストをドロップする代わりに、安全なウェブサイトにリクエストをリダイレクトしたい場合があります。レスポンダー機能を使用すると、ユーザーがアクセスしようとするパスや URL クエリを変更せずに、リクエストを安全な Web サイトにリダイレクトできます。

NetScaler レスポンダーがリクエストを **HTTP** から **HTTPS** にリダイレクトする方法

次の図は、アプライアンスがリクエストをリダイレクトする方法の段階的なフローを示しています。



レスポンダー機能を NetScaler アプライアンスの負荷分散 VIP アドレスとともに構成して、クライアント要求を HTTP から HTTPS にリダイレクトするには、次の手順を実行します。

アプライアンスのレスポンダー機能を有効にする

アプライアンスでレスポンダー機能を有効にするには、[システム] > [設定] **> [** 拡張機能の構成] に移動し、[レスポンダー] を選択します。

レスポンダーアクションの作成

レスポンダーアクションを作成するには、次の手順を実行します:

1. [**AppExpert**] > [レスポンダー] > [アクション] に移動し、[追加] をクリックします。
2. 「名前」フィールドに、http_to_https_actn などの適切な名前を指定します。
3. タイプとして「リダイレクト」を選択します。
4. 「式」フィールドに、次の式を入力します:

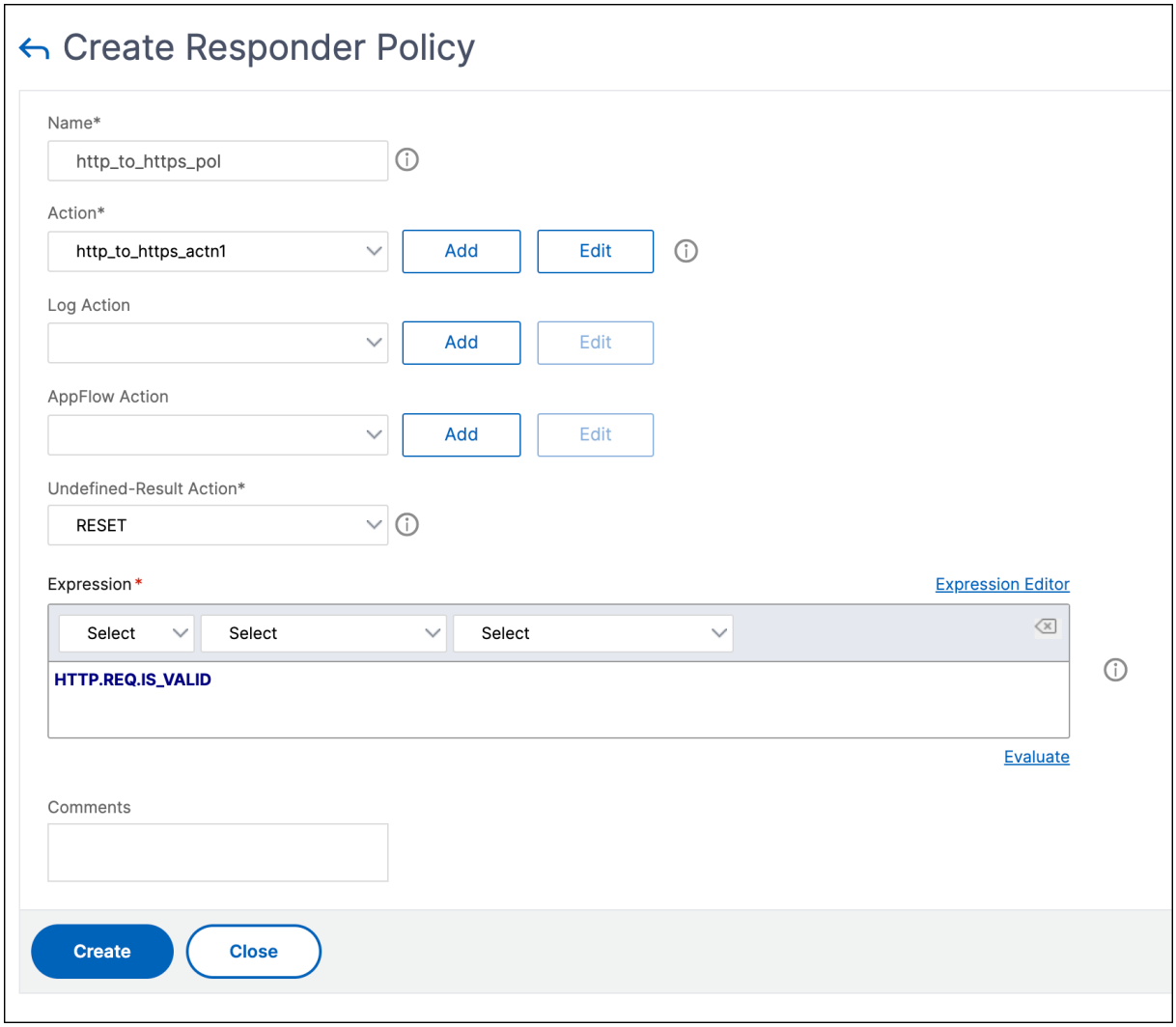
   ```
   "https://"+ HTTP.REQ.HOSTNAME.HTTP_URL_SAFE + HTTP.REQ.URL.
   PATH_AND_QUERY.HTTP_URL_SAFE。
   ```

5. [作成] をクリックします。

レスポンダーポリシーの作成

レスポンダーポリシーを作成するには、次の手順を実行します:

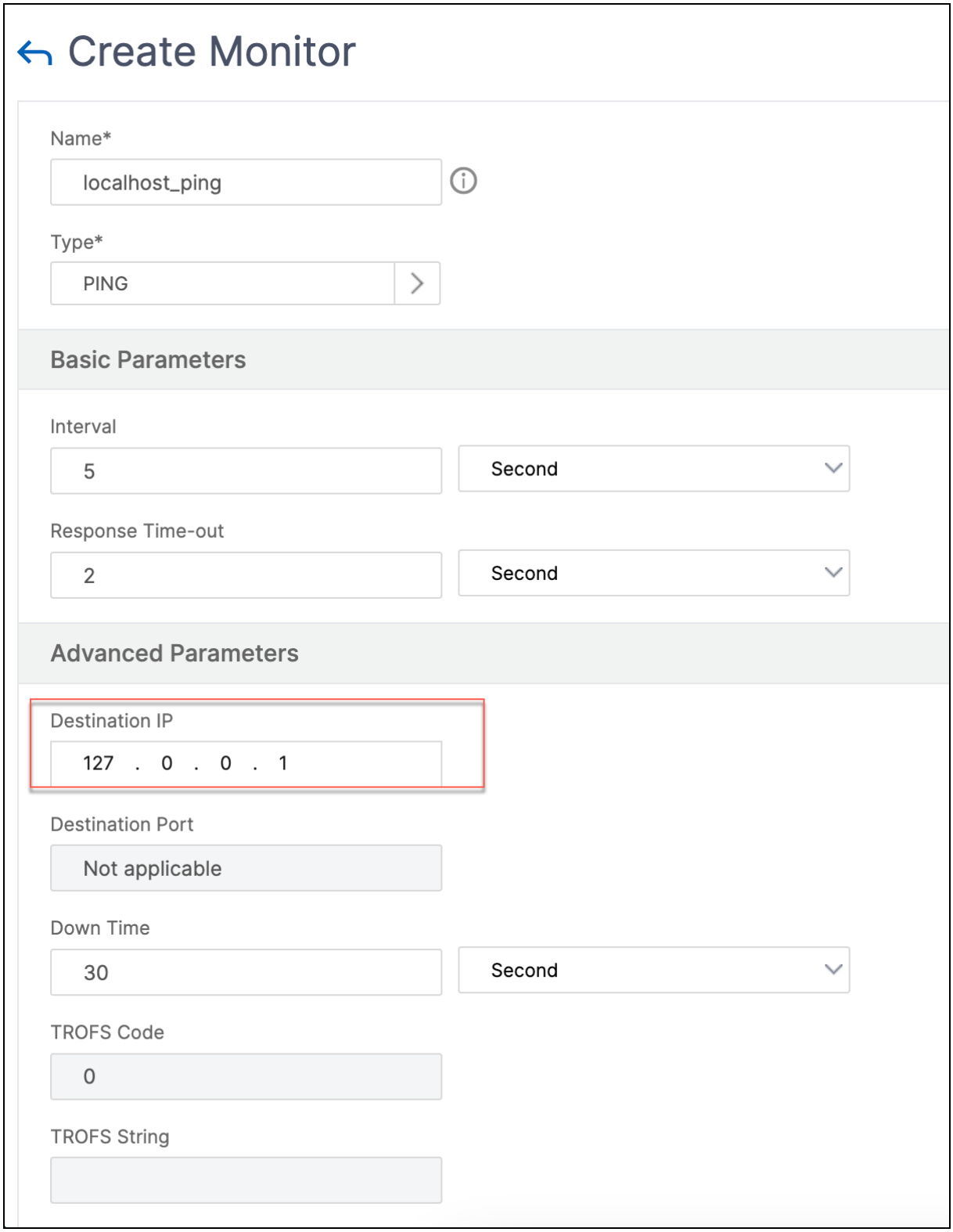
1. [**AppExpert**] > [レスポンダー] > [ポリシー] に移動し、[追加] をクリックします。
2. 「名前」フィールドに、http_to_https_pol などの適切な名前を指定します。
3. アクションリストから、作成したアクション名を選択します。
4. 「未定義のアクション」リストから、「リセット」を選択します。
5. 次のスクリーンショットに示すように、「式」フィールドに **HTTP.REQ.IS_VALID** 式を入力します。



モニターの作成

ステータスが常に UP とマークされているモニターを作成するには、次の手順を実行します:

1. [トラフィック管理] > [負荷分散] > [モニター] に移動し、[追加] をクリックします。
2. 「名前」フィールドに localhost_ping などの適切な名前を指定します。
3. 「宛先 **IP** 」フィールドで、IP アドレスとして 127.0.0.1 を指定します。



サービスを作成

サービスを作成するには、次の手順を実行します:

1. 負荷分散 > サービスに移動し、追加をクリックします。
2.「名前」フィールドに、Always_UP_Service などの適切な名前を指定します。
3.「サーバー」フィールドに、存在しない IP アドレスを指定します。
4. ポートフィールドに 80 を指定します。
5. 作成したモニターを「使用可能なモニター」リストから追加します。

負荷分散仮想サーバーの作成

1. [負荷分散] > [サービス] に移動し、[追加] をクリックします。
2.「名前」フィールドに適切な名前を指定します。
3. IP アドレスフィールドに Web サイトの **IP** アドレスを指定します。
4.「プロトコル」リストから「HTTP」を選択します。
5.「ポート」フィールドに「80」と入力します。
6. [ポリシー] タブをクリックします。
7. 作成したレスポンダーポリシーを Web サイトの HTTP 負荷分散 VIP アドレスにバインドします。
8. Web サイトの IP アドレスとポートが 443 の安全な負荷分散仮想サーバーを作成します。

アプライアンスのコマンドラインインターフェイスから前述の手順と同様の構成を作成するには、次のコマンドを実行します:

```
enable ns feature responder
add responder action http_to_https_actn redirect ""https://" + http.req
    .hostname.HTTP_URL_SAFE + http.REQ.URL.PATH_AND_QUERY.HTTP_URL_SAFE"
add responder policy http_to_https_pol HTTP.REQ.IS_VALID
    http_to_https_actn RESET
add lb monitor localhost_ping PING -LRTM ENABLED -destIP 127.0.0.1
add service Always_UP_service 1.2.3.4 HTTP 80 -gslb NONE -maxClient 0 -
    maxReq 0 -cip ENABLED dummy -usip NO -sp OFF -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP YES
bind lb monitor localhost_ping Always_UP_service
add lb vserver http_site.com HTTP 10.217.96.238 80 -persistenceType
    COOKIEINSERT -timeout 0 -cltTimeout 180
bind lb vserver http_site.com Always_UP_service
bind lb vserver http_site.com -policyName http_to_https_pol -priority 1
     -gotoPriorityExpression END
<!--NeedCopy-->
```

> 注:
>
> - リダイレクトが機能するには、ポート 80 の負荷分散リダイレクト仮想サーバーのステータスが UP にな

> っている必要があります。
> - HTTPS 仮想サーバーがアクティブでない場合、Web ブラウザーが正しくリダイレクトされないことがあります。
> - このリダイレクト設定により、複数のドメインが同じ IP アドレスにバインドされている状況にも対応できます。
> - クライアントが無効な HTTP 要求をリダイレクト仮想サーバに送信すると、アプライアンスは RESET メッセージ・コードを送信します。

トラブルシューティング

August 15, 2023

構成後にレスポンダー機能が期待どおりに機能しない場合は、いくつかの一般的なツールを使用して NetScaler リソースにアクセスし、問題を診断できます。

トラブルシューティング用リソース

最良の結果を得るには、次のリソースを使用して NetScaler アプライアンスの統合キャッシュの問題をトラブルシューティングしてください。

- ns.conf ファイル
- クライアントと NetScaler アプライアンスからの関連トレースファイル

上記のリソースに加えて、次のツールを使用するとトラブルシューティングが容易になります。

- iehttp ヘッダーまたは同様のユーティリティ
- NetScaler トレースファイル用にカスタマイズされた Wireshark アプリケーション

レスポンダー問題のトラブルシューティング

- 問題

  レスポンダー機能は設定されていますが、レスポンダーアクションが機能していません。

  解像度

  - この機能が有効になっていることを確認します。
  - いずれかのポリシーのヒットカウンタをチェックして、カウンタが増加しているかどうかを確認します。
  - ポリシーとアクションが正しく設定されていることを確認します。
  - アクションとポリシーが適切にバインドされていることを確認してください。

  - クライアントと NetScaler アプライアンスのパケットトレースを記録し、それらを分析して問題の原因を特定します。
  - クライアント上の IEHttpHeaters パケットトレースを記録し、HTTP リクエストとレスポンスを検証して問題への何らかの指摘を得てください。

- 問題

  メンテナンスページを作成する必要があります。

  解像度

1. サービスと仮想サーバーを設定します。
2. サービスがバインドされたバックアップ仮想サーバーを構成します。これにより、Web サイトのステータスは常に UP と表示されます。
3. バックアップ仮想サーバーをバックアップとして使用するようにプライマリ仮想サーバーを構成します。
4. 適切なターゲットでレスポンダーアクションを作成します。参考までに、次の例を示します。

```
add responder action sorry_page respondwith q{ "HTTP/1.0 200 OK"+
"\r\n\r\n"+ "<html><body>Sorry, this page is not available\</body
>\</html>"+ "\r\n"}
```

5. レスポンダーポリシーを作成し、アクションをそれにバインドします。
6. レスポンダポリシーをバックアップ仮想サーバにバインドします。

書き換え

August 15, 2023

> 警告:
>
> 従来のポリシーを使用したフィルター機能は廃止され、代替として、高度なポリシーインフラストラクチャで書き換え機能とレスポンダー機能を使用することをお勧めします。

書き換えとは、NetScaler ADC アプライアンスが処理する要求または応答の情報を書き換えることです。書き換えは、ウェブサイトの実際の設定に関する不必要な詳細を公開することなく、要求されたコンテンツへのアクセスを提供するのに役立ちます。書き換え機能が便利な状況は、次のとおりです。

- セキュリティを向上させるために、Citrix ADC はすべての`http://links`をレスポンス本文の`https://`に書き換えることができます。
- SSL オフロード展開では、応答内のセキュアでないリンクをセキュアリンクに変換する必要があります。書き換えオプションを使用すると、NetScaler ADC からクライアントへの送信応答にセキュリティで保護された

リンクがあることを確認するために、すべての `http://links`を`https://`に書き換えることができます。

- Web サイトにエラーページを表示する必要がある場合は、デフォルトの 404 エラーページの代わりにカスタムエラーページを表示できます。例えば、エラーページの代わりに Web サイトのホームページまたはサイトマップを表示すると、訪問者は Web サイトから離れるのではなく、サイトにとどまります。
- 新しい Web サイトを立ち上げるが、古い URL を使用する場合は、[書き換え] オプションを使用できます。
- サイト内のトピックに複雑な URL がある場合、シンプルで覚えやすい URL（「クール URL」とも呼ばれます）で書き直すことができます。
- デフォルトのページ名を Web サイトの URL に追加できます。たとえば、企業の Web サイトのデフォルトページが`http://www.abc.com/index.php`の場合、ユーザーがブラウザのアドレスバーに「abc.com」と入力すると、URL を「abc.com/index.php」に書き換えることができます。

書き換え機能を有効にすると、NetScaler ADC は HTTP 要求と応答のヘッダーと本文を変更できます。

HTTP リクエストとレスポンスを書き換えるには、構成する書き換えポリシーでプロトコル対応の NetScaler ADC ポリシー式を使用できます。HTTP 要求および応答を管理する仮想サーバーは、
HTTP または
SSL のタイプである必要があります。HTTP トラフィックでは、次のアクションを実行できます。

- リクエストの URL を変更する
- ヘッダーの追加、変更、削除
- 本文またはヘッダー内の特定の文字列を追加、置換、または削除します。

TCP ペイロードを書き換えるには、ペイロードをバイトの生のストリームと見なします。TCP 接続を管理する各仮想サーバーは、TCP または SSL_TCP のタイプである必要があります。TCP リライトという用語は、HTTP データではない TCP ペイロードの書き換えを指すために使用されます。TCP トラフィックでは、TCP ペイロードの任意の部分を追加、変更、または削除できます。

書き換え機能の使用例については、書き換えアクションとポリシーの例を参照してください。

[書き換え] オプションと [レスポンダー] オプションの比較

書き換え機能とレスポンダ機能の主な違いは次のとおりです。

Responder は、レスポンスまたはサーバーベースの式には使用できません。Responder は、クライアントパラメータに応じて、次のシナリオでのみ使用できます。

- 新しい Web サイトまたはウェブページへの HTTP リクエストのリダイレクト
- カスタムレスポンスで応答する
- 要求レベルで接続をドロップまたはリセットする

レスポンダーポリシーがある場合、NetScaler ADC はクライアントからの要求を調べ、該当するポリシーに従ってアクションを実行し、クライアントに応答を送信し、クライアントとの接続を閉じます。

書き換えポリシーがある場合、NetScaler ADC はクライアントからの要求またはサーバーからの応答を調べ、該当するポリシーに従ってアクションを実行し、トラフィックをクライアントまたはサーバーに転送します。

一般に、NetScaler ADC でクライアントまたは要求ベースのパラメータに基づいて接続をリセットまたはドロップする場合は、レスポンダーを使用することをお勧めします。レスポンダーを使用してトラフィックをリダイレクトするか、カスタムメッセージで応答します。HTTP リクエストとレスポンスのデータを操作するには、書き換えを使用します。

書き換えの仕組み

書き換えポリシーは、規則とアクションで構成されます。このルールは、書き換えが適用されるトラフィックを決定し、アクションによってが実行するアクションを決定します。複数の書き換えポリシーを定義できます。ポリシーごとに、バインドポイントと優先度を指定します。

バインドポイントとは、NetScaler ADC がトラフィックを検査して、書き換えポリシーを適用できるかどうかを検証するトラフィックフロー内のポイントを指します。ポリシーを特定の負荷分散またはコンテンツスイッチング仮想サーバーにバインドするか、NetScaler ADC によって処理されるトラフィック全体にポリシーを適用する場合は、ポリシーをグローバルにすることができます。これらのポリシーは、グローバルポリシーと呼ばれます。

ユーザー定義のポリシーに加えて、NetScaler ADC にはいくつかのデフォルトポリシーがあります。デフォルトポリシーを変更または削除することはできません。

ポリシーを評価するために、NetScaler ADC は次の順序に従います。

- グローバルポリシー
- 特定の仮想サーバにバインドされたポリシー
- 既定のポリシー

> 注:
>
> NetScaler ADC は、ポイントにバインドされている場合にのみ書き換えポリシーを適用できます。

NetScaler ADC は、次の手順で書き換え機能を実装します。

- NetScaler ADC アプライアンスは、グローバルポリシーをチェックし、個々のバインドポイントでポリシーをチェックします。
- 複数のポリシーがバインドポイントにバインドされている場合、NetScaler ADC は優先順位の順序でポリシーを評価します。プライオリティが最も高いポリシーが最初に評価されます。各ポリシーを評価した後、ポリシーが TRUE と評価されると、関連付けられたアクションが実行されるポリシーに関連付けられたアクションが追加されます。一致は、ポリシールールで指定された特性が評価されるリクエストまたはレスポンスの特性と一致する場合に発生します。

- どのポリシーでも、アクションに加えて、現在のポリシーが評価された後に評価される必要があるポリシーを指定できます。このポリシーは「式に移動」と呼ばれます。どのポリシーでも、式に移動（gotoPriorityExpr）が指定されている場合、NetScaler ADC は式に移動ポリシーを評価します。次に高い優先度を持つポリシーは無視されます。

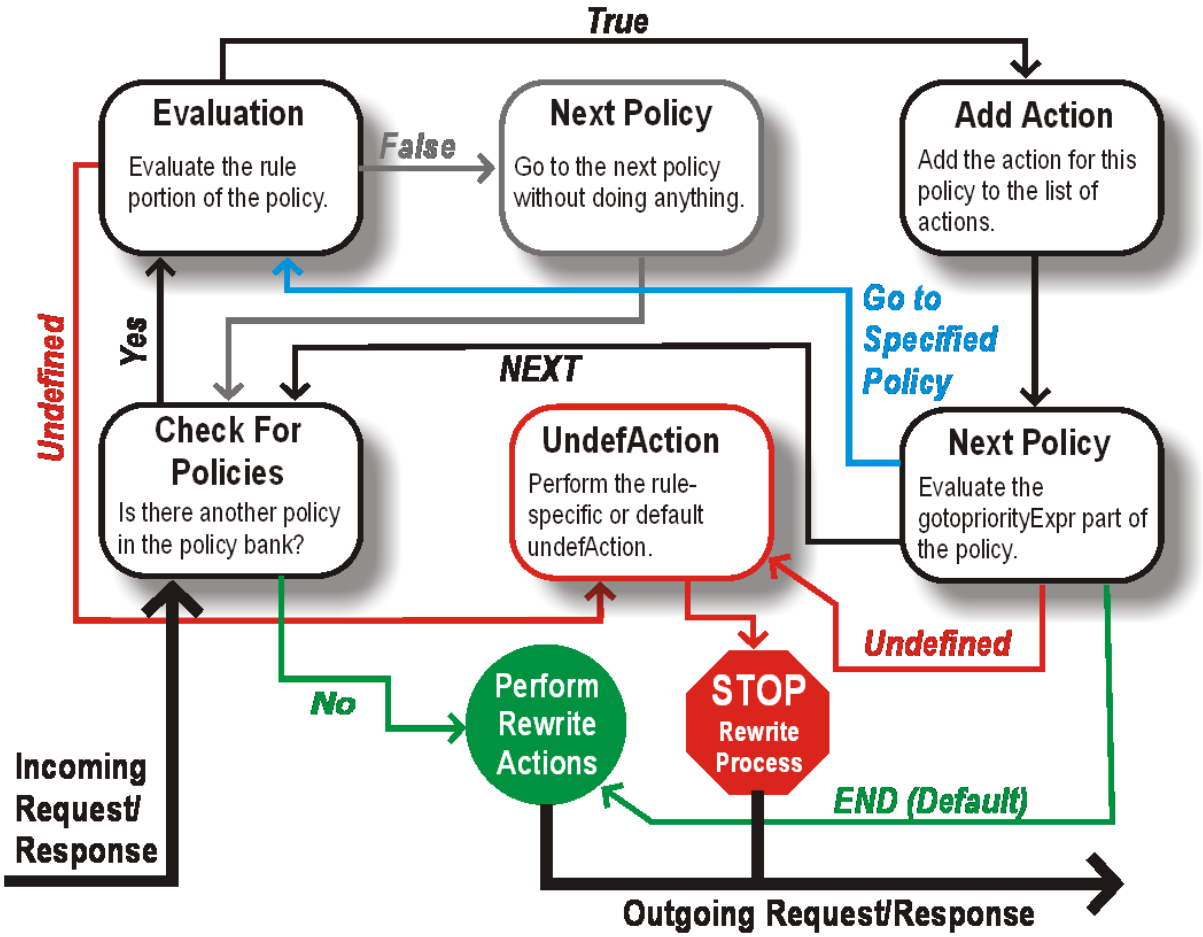
  ポリシーの優先順位を指定して、[式に移動（Go to Expression）] ポリシーを示すことができます。ポリシーの名前は使用できません。特定のポリシーを評価した後に NetScaler ADC が他のポリシーの評価を停止する場合は、[式に移動] を [終了] に設定できます。

- すべてのポリシーが評価された後、またはポリシーの「式に移動」が「終了」に設定されている場合、NetScaler ADC はアクションのリストに従ってアクションの実行を開始します。

書き換えポリシーの構成の詳細については、書き換えポリシーの構成および書き換えポリシーのバインドについては、「書き換えポリシーのバインド」を参照してください。

次の図は、書き換え機能を使用した場合に、NetScaler ADC が要求または応答をどのように処理するかを示しています。

図 1：書き換えプロセス



ポリシー評価

プライオリティが最も高いポリシーが最初に評価されます。NetScaler ADC は、一致が見つかったときに書き換えポリシーの評価を停止しません。NetScaler ADC で構成されているすべての書き換えポリシーを評価します。

- ポリシーが TRUE と評価された場合、NetScaler ADC は以下の手順に従います。
  - ポリシーの [式に移動] が [終了] に設定されている場合、NetScaler ADC は他のすべてのポリシーの評価を停止し、書き換えの実行を開始します。
  - gotoPriorityExpression は、’NEXT’、’END’、いくつかの整数、または ‘INVOCATION_LIST’ に設定できます。この値によって、次のプライオリティを持つポリシーが決まります。次の表は、式の各値に対して NetScaler ADC が実行するアクションを示しています。

式の値	アクション
NEXT	次のプライオリティを持つポリシーが評価されます。
END	ポリシーの評価は停止します。
<an integer>	指定された優先度を持つポリシーが評価されます。
INVOCATION_LIST	Goto NEXT または END は、呼び出しリストの結果に基づいて適用されます。

- ポリシーが FALSE と評価された場合、NetScaler ADC は優先順位の順に評価を続行します。
- ポリシーが UNDEFINED と評価された場合（エラーのために受信トラフィックで評価できない）、NetScaler ADC は UNDEFINED 条件（undeFaction と呼ばれる）に割り当てられたアクションを実行し、ポリシーのさらなる評価を停止します。

NetScaler ADC は、評価が完了した後にのみ、実際の書き換えを開始します。これは、TRUE と評価されるポリシーによって識別されるアクションのリストを参照し、書き換えを開始します。リスト内のすべてのアクションを実装すると、NetScaler ADC は必要に応じてトラフィックを転送します。

> 注:
>
> HTTP ヘッダーまたは本文、または TCP ペイロードの同じ部分で、競合または重複するアクションがポリシーで指定されていないことを確認します。このような競合が発生すると、NetScaler ADC は未定義の状況に遭遇し、書き換えを中止します。

書き換えアクション

NetScaler ADC アプライアンスで、本文内のテキストの追加、置換、削除、ヘッダーの追加、変更、削除、または書き換えアクションとしての TCP ペイロードの変更など、実行するアクションを指定します。書き換えアクションの詳細については、「書き換えアクションの設定」を参照してください。

次の表では、ポリシーが TRUE と評価されたときに NetScaler ADC が実行できる手順について説明します。

アクション	結果
Insert	ポリシーに指定された書き換えアクションが実行されます。
NOREWRITE	リクエストまたはレスポンスは書き換えられません。NetScaler ADC は、メッセージの一部を書き換えずにトラフィックを転送します。
RESET	接続は TCP レベルで中止されます。
DROP	メッセージはドロップされます。

> 注記:
>
> どのポリシーでも、アンダーアクション（ポリシーが UNDEFINED と評価されたときに実行されるアクション）を NOREWRITE、RESET、または DROP として設定できます。

書き換え機能を使用するには、次の手順を実行します。

- NetScaler ADC でこの機能を有効にします。
- 書き換えアクションを定義します。
- 書き換えポリシーを定義します。
- ポリシーをバインドポイントにバインドして、ポリシーを有効にします。

書き換えを有効にする

HTTP または TCP 要求または応答を書き換える場合は、NetScaler ADC アプライアンスの書き換え機能を有効にします。この機能が有効になっている場合、NetScaler ADC は指定されたポリシーに従って書き換えアクションを実行します。詳細については、「書き換えの仕組み」を参照してください。

コマンドラインインターフェイスを使用して書き換え機能を有効にするには

コマンドプロンプトで次のコマンドを入力して、書き換え機能を有効にし、構成を確認します。

- enable ns feature REWRITE
- show ns feature

例:

```
1 > enable ns feature REWRITE
2  Done
3 > show ns feature
4
5         Feature                         Acronym              Status
```



```
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 .
 .
 .
 1)     Rewrite                         REWRITE             ON
 .
 .
 1)     NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

GUI を使用して書き換え機能を有効にするには

1. ナビゲーションペインで、[システム] をクリックし、[設定] をクリックします。
2. 詳細ウィンドウの [モードと機能] で、[基本機能の構成] をクリックします。
3. [基本機能の構成] ダイアログボックスで、[書き換え] チェックボックスをオンにし、[**OK**] をクリックします。
4. [機能の有効化/無効化] ダイアログボックスで、[はい] をクリックします。ステータスバーに、選択した機能が有効になったことを示すメッセージが表示されます。

書き換えアクションの構成

> 警告
>
> 書き換えアクションのパターン機能は、NetScaler 12.0 ビルド 56.20 以降では廃止され、代替として、［検索書き換えアクション］パラメータを使用することをお勧めします。

書き換えアクションは、サーバーまたはクライアントに送信する前に要求または応答に加えられた変更を示します。

式は、次のことを定義しています。

- アクションタイプを書き換えます。
- 書き換えアクションの場所です。
- アクションの設定タイプを書き換えます。

たとえば、DELETE アクションはターゲット式のみを使用します。REPLACE アクションは、ターゲット式と式を使用して置換テキストを設定します。

書き換え機能を有効にした後は、組み込みの書き換えアクションで十分でない限り、1 つ以上のアクションを設定する必要があります。すべての組み込みアクションの名前は、文字列 ns_cvpn で始まり、その後に文字列とアンダースコア文字が続きます。組み込みアクションは、クライアントレス VPN 要求または応答の一部のデコード、JavaScript または XML データの変更など、便利で複雑なタスクを実行します。組み込みアクションは、表示、有効化、および無効化できますが、変更または削除することはできません。

> 注:
>
> HTTP 書き換えにのみ使用できるアクションタイプは、[書き換え アクションタイプ] 列で識別されます。

詳細については、「**Type** パラメータ」を参照してください。

コマンドラインインターフェイスを使用して書き換えアクションを作成する

コマンドプロンプトで次のコマンドを入力して、書き換えアクションを作成し、構成を確認します。

- `add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-search <expression>] [refineSearch <expression>] [-comment<string>]`
- `show rewrite action <name>`

詳細については、書き換えアクションタイプとその引数テーブルを参照してください 。

書き換え機能には、次の組み込みアクションがあります。

- noreWrite-リクエストまたはレスポンスを書き換えずにユーザーに送信します。
- RESET-ユーザーがリクエストを再送信できるように、接続をリセットしてユーザーのブラウザに通知します。
- DROP-ユーザーに応答を送信せずに接続をドロップします。

次のフロータイプの 1 つがすべてのアクションに暗黙的に関連付けられます。

- Request-アクションはリクエストに適用されます。
- Response-アクションは応答に適用されます。
- ニュートラル-アクションはリクエストとレスポンスの両方に適用されます。

名前

ユーザー定義の書き換えアクションの名前。文字、数字、またはアンダースコア文字 (_) で始まり、ハイフン (-)、ピリオド (.) ハッシュ (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア文字のみを含める必要があります。書き換えポリシーの追加後に変更できます。

型パラメータ

Type パラメータには、ユーザー定義の書き換えアクションのタイプが表示されます。

Type パラメータの値は次のとおりです。

- REPLACE <target> <string_builder_expr>。ターゲット文字列を文字列ビルダーの式に置き換えます。

例:

```
> add rewrite action replace_http_act replace http.res.body(100) '"new_replaced_data"'
Done
> sh rewrite action replace_http_act
Name: replace_http_act
Operation: replace
Target:http.res.body(100)
Value:"new_replaced_data"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_ALL <target> <string_builder_expr1> -(search)<s>-<target> で指定されたリクエストまたはレスポンスで、で定義されている文字列のすべての出現箇所を、<pattern_to_search>で定義されている文字列を<string_builder_expr>に置き換えます。検索オプションを使用して、置換する文字列を検索できます。

例:

```
> add policy patset pat_list_2
Done
> bind policy patset pat_list_2 "www.abc.com"
Done
> bind policy patset pat_list_2 "www.def.com"
Done
> add rewrite action refineSearch_act_31 replace_all "HTTP.RES.BODY(100000)" ""https://""-search "patset("pat_list_2")" -refineSearch "EXTEND(7,0).REGEX_SELECT(re#http://#)"
Done

> sh rewrite action refineSearch_act_31
Name: refineSearch_act_31
Operation: replace_all
Target:HTTP.RES.BODY(100000)
Refine Search:EXTEND(7,0).REGEX_SELECT(re#http://#)
Value:"https://"
Search: patset("pat_list_2")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```


- REPLACE_HTTP_RES <string_builder_expr>。完全な HTTP レスポンスを文字列ビルダー式で定義された文字列に置き換えます。

例:

```
> add rewrite action replace_http_res_act replace_http_res '"HTTP/1.1 200 OK\r\n\r\nSending from ADC"'
 Done
> sh rewrite action replace_http_res_act
Name: replace_http_res_act
Operation: replace_http_res
Target:"HTTP/1.1 200 OK
Sending from ADC"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_SIP_RES <target>。SIP 応答全体を、<target>で指定された文字列に置き換えます。

例:

```
> add rewrite action replace_sip_res_act replace_sip_res '"HTTP/1.1 200 OK\r\n\r\nSending from ADC"'
Done
> sh rewrite action replace_sip_res_act
Name: replace_sip_res_act
Operation: replace_sip_res
Target:"HTTP/1.1 200 OK
Sending from ADC"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_HTTP_HEADER <header_string> <contents_string_builder_expr>。header_stringで指定された HTTP ヘッダーと、contents_string_builder_exprで指定されたヘッダーコンテンツを挿入します。

例:

```
> add rewrite action ins_cip_header insert_http_header "CIP" "CLIENT.IP.SRC"
Done
> sh rewrite action ins_cip_header
Name: ins_cip_header
Operation: insert_http_header
Target:CIP
```



```
Value:CLIENT.IP.SRC
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE_HTTP_HEADER <target>。<target>で指定された HTTP ヘッダーを削除します。

例:

```
> add rewrite action del_true_client_ip_header delete_http_header "True
    -Client-IP"
Done
> sh rewrite action del_true_client_ip_header
Name: del_true_client_ip_header
Operation: delete_http_header
Target:True-Client-IP
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CORRUPT_HTTP_HEADER <target>。<target>で指定したすべての出現する HTTP ヘッダーのヘッダー名を破損した名前に置き換え、レシーバによって認識されないようにします。例:MY_HEADER は MHEY_ADER に変更されます。

例:

```
> add rewrite action corrupt_content_length_hdr corrupt_http_header "
    Content-Length"
Done
> sh rewrite action corrupt_content_length_hdr
Name: corrupt_content_length_hdr
Operation: corrupt_http_header
Target:Content-Length
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_BEFORE <string_builder_expr1> <string_builder_expr1>。<string_builder_expr1>で指定された文字列を検索し、その前の<string_builder_expr2>に文字列を挿入します。


```
> add rewrite action insert_before_ex_act insert_before http.res.body
    (100) '"Add this string in the starting"'
Done
> sh rewrite action insert_before_ex_act
Name: insert_before_ex_act
Operation: insert_before
Target:http.res.body(100)
Value:"Add this string in the starting"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_BEFORE_ALL <target> <string_builder_expr1> -(search)<string_builder_expr2>。<target>で指定されたリクエストまたはレスポンスで、で指定された文字列のすべての出現箇所を検索します。で指定された文字列を挿入します その前に。検索オプションを使用して文字列を検索できます。

例:

```
> add policy patset pat
 Done
> bind policy patset pat abcd
 Done
> add rewrite action refineSearch_act_1 insert_before_all http.res.body
    (10) 'target.prefix(10) + "refineSearch_testing"' -search patset("
    pat") -refineSearch extend(10,10)
 Done
> sh rewrite action refineSearch_act_1
Name: refineSearch_act_1
Operation: insert_before_all
Target:http.res.body(10)
Refine Search:extend(10,10)
Value:target.prefix(10) + "refineSearch_testing"
Search: patset("pat")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_AFTER <string_builder_expr1> <string_builder_expr2>。string_builder_
文字列の後に、string_builder_expr2で指定された文字列を挿入します。

例:

```
> add rewrite action insert_after_act insert_after http.req.body(100) '
    "add this string after 100 bytes"'
```



```
Done
> sh rewrite action insert_after_act
Name: insert_after_act
Operation: insert_after
Target:http.req.body(100)
Value:"add this string after 100 bytes"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_AFTER_ALL <target> <string_builder_expr1> -(search)<string_builder_expr2>。<target>で指定されたリクエストまたはレスポンスで、<string_builder_expr2>で指定された文字列のすべての出現箇所を検索し、その後に<string_builder_expr1>で指定された文字列を挿入します。検索機能を使用して文字列を検索できます。

例:

```
> add rewrite action refineSearch_act_2 insert_after_all http.res.body
    (100) '"refineSearch_testing"' -search text("abc") -refineSearch
    extend(0, 10)
Done
> sh rewrite action refineSearch_act_2
Name: refineSearch_act_2
Operation: insert_after_all
Target:http.res.body(100)
Refine Search:extend(0, 10)
Value:"refineSearch_testing"
Search: text("abc")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE <target>。target で指定された文字列を削除します。

例:

```
> add rewrite action delete_ex_act delete http.req.header("HDR")
Done
> sh rewrite action delete_ex_act
Name: delete_ex_act
Operation: delete
Target:http.req.header("HDR")
Hits: 0
Undef Hits: 0
```



```
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE_ALL <target> -(search)<string_builder_expr>。<target>で指定されたリクエストまたはレスポンスで、<string_builder_expr>で指定された文字列のすべてのオカレンスを検索して削除します。検索機能を使用して文字列を検索できます。

例:

```
>add rewrite action refineSearch_act_4 delete_all "HTTP.RES.BODY(50000)
   " -search text("Windows Desktops") -refineSearch "EXTEND(40,40).
   REGEX_SELECT(re#\s`\*`<AppData>.`\*`\s`\*`<\\/AppData>#)"
Done
> show REWRITE action refineSearch_act_4
Name: refineSearch_act_4
Operation: delete_all
Target:HTTP.RES.BODY(50000)
Refine Search:EXTEND(40,40).REGEX_SELECT(re#\s`\*`<AppData>.`\*`\s
   `\*`</AppData>#)
Search: text("Windows Desktops")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_DIAMETER_HEADER_FIELD <target> <field value>。リクエストまたはレスポンスで、<target>で指定されたヘッダーフィールドを変更します。Diameter.req.flags.SET(<flag>)または stringbuilderexpressionとしてのDiameter.req.flags.UNSET<flag>を使用して、フラグを設定または解除します。

例:

```
> add rewrite action replace_diameter_field_ex_act
   replace_diameter_header_field diameter.req.flags diameter.req.flags.
   set(PROXIABLE)
Done
> sh rewrite action replace_diameter_field_ex_act
Name: replace_diameter_field_ex_act
Operation: replace_diameter_header_field
Target:diameter.req.flags
Value:diameter.req.flags.set(PROXIABLE)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```


- REPLACE_DNS_HEADER_FIELD <target>。リクエストまたはレスポンスで、<target>で指定されたヘッダーフィールドを変更します。

例:

```
> add rewrite action replace_dns_hdr_act replace_dns_header_field dns.req.header.flags.set(AA)
Done
> sh rewrite action replace_dns_hdr_act
Name: replace_dns_hdr_act
Operation: replace_dns_header_field
Target:dns.req.header.flags.set(AA)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_DNS_ANSWER_SECTION <target>。応答の DNS 応答セクションを置き換えます。これは、A レコードと AAAA レコードにのみ適用されます。DNS.NEW_RRSET_AおよびNS.NEW_RRSET_AAAA式を使用して、新しい回答セクションを構成します。

例:

```
> add rewrite action replace_dns_ans_act replace_dns_answer_section DNS.NEW_RRSET_A("1.1.1.1", 10)
Done
> sh rewrite action replace_dns_ans_act
Name: replace_dns_ans_act
Operation: replace_dns_answer_section
Target:DNS.NEW_RRSET_A("1.1.1.1", 10)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_DECODE<target>。ターゲットによって指定されたパターンをクライアントレス VPN 形式でデコードします。

例:

```
> add rewrite action cvpn_decode_act_1 clientless_vpn_decode http.req.body(100)
Done
> sh rewrite action cvpn_decode_act_1
Name: cvpn_decode_act_1
Operation: clientless_vpn_decode
Target:http.req.body(100)
```



```
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_DECODE_ALL<target>-search<expression>。検索パラメータで指定されたすべてのパターンをクライアントレス VPN 形式でデコードします。

例:

```
> add rewrite action act1 clientless_vpn_decode_all http.req.body(100)
   -search text("abcd")
Done
> sh rewrite action act1
Name: act1
Operation: clientless_vpn_decode_all
Target:http.req.body(100)
Search: text("abcd")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_ENCODE<target>。ターゲットで指定されたパターンをクライアントレス VPN 形式でエンコードします。

例:

```
> add rewrite action cvpn_encode_act_1 clientless_vpn_encode http.req.
   body(100)
Done
> sh rewrite action cvpn_encode_act_1
Name: cvpn_encode_act_1
Operation: clientless_vpn_encode
Target:http.req.body(100)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_ENCODE_ALL<target>-search<expression>。指定されたすべてのパターンをクライアントレス VPN 形式でエンコードします。

例:


```
> add rewrite action act2 clientless_vpn_encode_all http.req.body(100)
   -search text("abcd")
Done
> sh rewrite action act2
Name: act1
Operation: clientless_vpn_encode_all
Target:http.req.body(100)
Search: text("abcd")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CORRUPT_SIP_HEADER<target>。<target>で指定したすべての SIP ヘッダーのヘッダー名を破損した名前に置き換え、受信者がそれを認識しないようにします。

例:

```
> add rewrite action corrupt_sip_hdr_act corrupt_sip_header SIP_HDR
Done
> sh rewrite action corrupt_sip_hdr_act
Name: corrupt_sip_hdr_act
Operation: corrupt_sip_header
Target:SIP_HDR
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_SIP_HEADER <header_string_builder_expr> <contents_string_builder_ex>。<header_string_builder_expr>で指定された SIP ヘッダーと、<contents_string_builder_ex>で指定されたヘッダーコンテンツを挿入します。

例:

```
> add rewrite action insert_sip_hdr_act insert_sip_header SIP_HDR '"
   inserting_sip_header"'
 Done
>sh rewrite action insert_sip_hdr_act
Name: insert_sip_hdr_act
Operation: insert_sip_header
Target:SIP_HDR
Value:"inserting_sip_header"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

```



```
<!--NeedCopy-->
```

- DELETE_SIP_HEADER<target>。<target>で指定された SIP ヘッダーを削除します。

例:

```
> add rewrite action delete_sip_hdr delete_sip_header  SIP_HDR
Done
> sh rewrite action delete_sip_hdr
Name: delete_sip_hdr
Operation: delete_sip_header
Target:SIP_HDR
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

ターゲットパラメータ

Target パラメータリクエストまたはレスポンスのどの部分を書き換えるかを指定する式を指定します。

StringBuilderExpr

StringBuilderExpr 指定された場所のリクエストまたは応答に挿入されるコンテンツを指定する式を指定します。この式は、指定された文字列を置き換えます。

例 **1.** クライアント **IP** を使用した **HTTP** ヘッダーの挿入:

```
> add rewrite action insertact INSERT_HTTP_HEADER "client-IP" CLIENT.IP
    .SRC
Done
> show rewrite action insertact
Name: insertact
Operation: insert_http_header
Target:Client-IP
Value:CLIENT.IP.SRC
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

例 **2. TCP** ペイロード内の文字列の置換（**TCP** 書き換え）:

```
> add rewrite action client_tcp_payload_replace_all REPLACE_ALL
```



```
  'client.tcp.payload(1000)' '"new-string"' -search text("old-string")
Done
> show rewrite action client_tcp_payload_replace_all
Name: client_tcp_payload_replace_all
Operation: replace_all
Target:client.tcp.payload(1000)
Value:"new-string"
Search: text("old-string")
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
>
<!--NeedCopy-->
```

リクエストまたはレスポンスの一部を検索して書き換え

検索機能は、リクエストまたはレスポンスで必要なパターンのすべてのインスタンスを検索するのに役立ちます。

検索機能は、次のアクションタイプで使用する必要があります。

- INSERT_BEFORE_ALL
- INSERT_AFTER_ALL
- REPLACE_ALL
- DELETE_ALL
- CLIENTLESS_VPN_ENCODE_ALL
- CLIENTLESS_VPN_DECODE_ALL

検索機能は、次のアクションタイプでは使用できません。

- INSERT_HTTP_HEADER
- INSERT_BEFORE
- INSERT_AFTER
- REPLACE
- 削除
- DELETE_HTTP_HEADER
- CORRUPT_HTTP_HEADER
- REPLACE_HTTP_RES
- CLIENTLESS_VPN_ENCODE
- CLIENTLESS_VPN_DECODE
- INSERT_SIP_HEADER
- DELETE_SIP_HEADER
- CORRUPT_SIP_HEADER
- REPLACE_DIAMETER_HEADER_FIELD

- REPLACE_DNS_ANSWER_SECTION
- REPLACE_DNS_HEADER_FIELD
- REPLACE_SIP_RES

次の検索タイプがサポートされています。

- テキスト-
  リテラル文字列例 ⊠ 検索テキスト (「hello」)
- 正規表現-リクエストまたはレスポンスの複数の文字列を照合するために使用されるパターン例 ⊠ 検索正規表現 (re~^hello*~)
- XPATH-XML を検索する XPATH 式。
  例 ⊠-search xpath(xp%/a/b%)
- JSON-JSON を検索するための XPATH 式。
  例 ⊠ 検索 xpath_json (xp%/a/b%)
  HTML-HTML を検索する XPATH
  式例 ⊠ 検索 xpath_html (xp%/html/body%)
  パッチセット-これはパッチセットエンティティにバインドされたすべてのパターンを検索します。
  例:-search patset(“patset1”)
- Datset-データセットエンティティにバインドされたすべてのパターンを検索します。
  例 ⊠-search dataset(“dataset1”)
- AVP-
  直径/Radius メッセージの例 ⊠ 検索 avp (999) で複数の AVP を照合するために使用される AVP 番号

検索結果を絞り込む

検索の絞り込み機能を使用して、検索結果を絞り込むための追加条件を指定できます。検索の絞り込み機能は、検索機能が使用されている場合にのみ使用できます。
検索の絞り込みパラメータは、常に「extend (m, n)」操作で始まります。ここで、’m’ は検索結果の左側に数バイトを指定し、’n’ は検索結果の右側に数バイトを指定して、選択範囲を拡張します。

設定されている書き換えアクションが次の場合:

```
> add rewrite action test_refine_search replace_all http.res.body(10) '
    " testing_refine_search" ' -search text("abc") -refineSearch extend
    (1,1)
And the HTTP response body is abcxxxx456.

<!--NeedCopy-->
```

次に、検索パラメータはパターン「abc」を検出し、refineSearch パラメータは一致したパターンの左側に余分な 1 バイト、右側に余分な 1 バイトをチェックするように設定されているためです。結果として置換されるテキストは abcx です。したがって、このアクションの出力は`testing_refine_searchxxx456`です。

例 **1: INSERT_BEFORE_ALL** アクションタイプで **[絞り込み]** 検索機能を使用する。

```
> add policy patset pat
Done
> bind policy patset pat abcd
Done
> add rewrite action refineSearch_act_1 insert_before_all http.res.body
    (10) 'target.prefix(10) + "refineSearch_testing"' -search patset("
    pat") -refineSearch extend(10,10)
Done
> sh rewrite action refineSearch_act_1
Name: refineSearch_act_1
Operation: insert_before_all
Target:http.res.body(10)
Refine Search:extend(10,10)
Value:target.prefix(10) + "refineSearch_testing"
Search: patset("pat")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

例 **2: INSERT_AFTER_ALL** アクションタイプで「絞り込み」サーチ機能を使用する。

```
> add rewrite action refineSearch_act_2 insert_after_all http.res.body
    (100) '"refineSearch_testing"' -search text("abc") -refineSearch
    extend(0, 10)
Done
> sh rewrite action refineSearch_act_2
Name: refineSearch_act_2
Operation: insert_after_all
Target:http.res.body(100)
Refine Search:extend(0, 10)
Value:"refineSearch_testing"
Search: text("abc")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

例 **3: REPLACE_ALL** アクションタイプで **[絞り込み]** 検索機能を使用する。

```
> add policy patset pat_list_2
Done
> bind policy patset pat_list_2 "www.abc.com"
Done
> bind policy patset pat_list_2 "www.def.com"
Done
> add rewrite action refineSearch_act_31 replace_all "HTTP.RES.BODY
```



```
    (100000)" ""https://"" -search "patset("pat_list_2")" -refineSearch
    "EXTEND(7,0).REGEX_SELECT(re#http://#)"
Done
> sh rewrite action refineSearch_act_31
Name: refineSearch_act_31
Operation: replace_all
Target:HTTP.RES.BODY(100000)
Refine Search:EXTEND(7,0).REGEX_SELECT(re#http://#)
Value:"https://"
Search: patset("pat_list_2")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

例 **4: DELETE_ALL** アクションタイプで **[**検索の絞り込み**]** 機能を使用する。

```
>add rewrite action refineSearch_act_4 delete_all "HTTP.RES.BODY(50000)
    " -search text("Windows Desktops") -refineSearch "EXTEND(40,40).
    REGEX_SELECT(re#\s*<AppData>.\*\s\*<\/AppData>#)"
> show REWRITE action refineSearch_act_4
Name: refineSearch_act_4
Operation: delete_all
Target:HTTP.RES.BODY(50000)
Refine Search:EXTEND(40,40).REGEX_SELECT(re#\s*<AppData>.\*\s\*</
    AppData>#)
Search: text("Windows Desktops")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
>
<!--NeedCopy-->
```

例 **5: CLIENTLESS_VPN_ENCODE_ALL** アクションタイプで「検索の絞り込み」機能を使用する。

,,,

> add rewrite action act2 clientless_vpn_encode_all http.req.body(100) -search text(“abcd”)
> Done
> sh rewrite action act2
> Name: act1
> Operation: clientless_vpn_encode_all
> Target:http.req.body(100)
> Search: text(“abcd”)
> Hits: 0
> Undef Hits: 0
> Action Reference Count: 0


```
Done
,,,
```

例 **6: CLIENTLESS_VPN_DECODE_ALL** アクションタイプで「検索の絞り込み」機能を使用する。

```
> add rewrite action act1 clientless_vpn_decode_all http.req.body(100)
   -search text("abcd")
Done
> sh rewrite action act1
Name: act1
Operation: clientless_vpn_decode_all
Target:http.req.body(100)
Search: text("abcd")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して既存の書き換えアクションを変更する

コマンドプロンプトで次のコマンドを入力して、既存の書き換えアクションを変更し、構成を確認します。

- `set rewrite action <name> [-target <expression>] [-stringBuilderExpr <expression>] [-search <expression>] [-refineSearch <expression>] [-comment <string>]`

コマンドプロンプトで次のコマンドを入力して、変更された構成を確認します。

- `show rewrite action <name>`

例:

```
> set rewrite action insertact -target "Client-IP"
 Done
> show rewrite action insertact

Name: insertact
Operation: insert_http_header   Target:Client-IP
Value:CLIENT.IP.SRC
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して書き換えアクションを削除する

コマンドプロンプトで次のコマンドを入力して、書き換えアクションを削除します。

```
rm rewrite action <name>
```

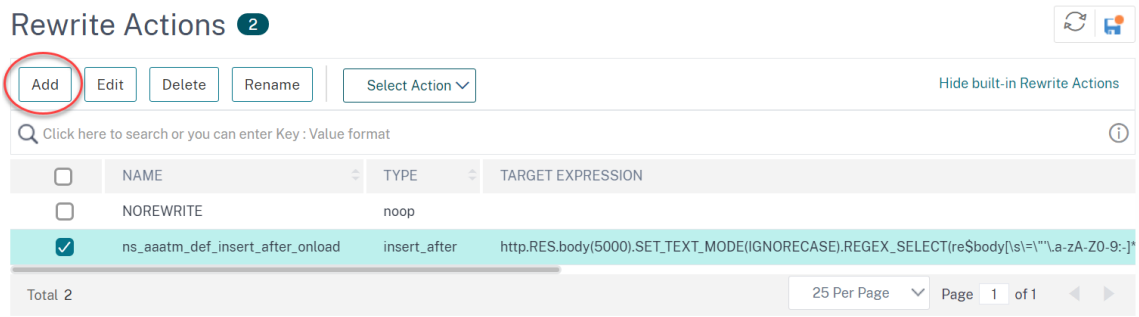
例:

```
> rm rewrite action insertact
Done

<!--NeedCopy-->
```

構成ユーティリティを使用して書き換えアクションを構成する

1. [**AppExpert**] > [**Rewrite**] > [**Actions**] の順に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - アクションを作成するには、[追加] をクリックします。
   - 既存のアクションを変更するには、アクションを選択し、[編集] をクリックします。
3. [作成] または [**OK**] をクリックします。アクションが正常に構成されたことを示すメッセージがステータスバーに表示されます。
4. 手順 2 ～4 を繰り返して、必要な数の書き換えアクションを作成または変更します。
5. [閉じる] をクリックします。



[数式の追加**]** ダイアログボックスを使用して式を追加します

1. [書き換えアクションの作成] または [書き換えアクションの設定] ダイアログボックスで、入力する引数 type のテキスト領域で、[追加] をクリックします。

2. [式の追加] ダイアログボックスの最初のリストボックスで、式の最初の用語を選択します。

    - HTTP HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。
    - SYS。保護されたウェブサイト。リクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
    - CLIENT. 要求を送信したコンピュータ。リクエストの送信者の側面を調べる場合は、これを選択します。

選択すると、右端のリストボックスに、式の次の部分に適した用語がリストされます。

1. 2 番目のリストボックスで、式の 2 番目の用語を選択します。選択肢は、前のステップで行った選択によって異なり、コンテキストに適切です。2 番目の選択を行った後、[式の構築] ウィンドウの下のヘルプウィンドウ (空白) に、選択した用語の目的と使用法を説明するヘルプが表示されます。
2. 式が終了するまで、前のリストボックスの右側に表示されるリストボックスから用語を選択するか、値の入力を求めるテキストボックスに文字列または数値を入力します。
   PI 式の言語およびレスポンダーポリシーの式の作成の詳細については、「ポリシーと式」を参照してください。

サンプル HTTP データに対してリライトアクションを使用した場合のエフェクトをテストする場合は、リライト式エバリュエータを使用できます。

TCP ペイロードを書き換え

TCP 書き換えのアクション内のターゲット式は、次のいずれかの式プレフィックスで開始する必要があります。

- **CLIENT.TCP.PAYLOAD.** クライアント要求の TCP ペイロードを書き換えるため。たとえば、CLIENT.TCP.PAYLOAD (10000) .AFTER_STR (“string1”) などです。
- **SERVER.TCP.PAYLOAD.** サーバー応答の TCP ペイロードを書き換えるため。たとえば、SERVER.TCP.PAYLOAD(1000).B64DECODE.BETWEEN(“string1” ,” string2”) など。

[書き換えアクションエバリュエータ**]** ダイアログボックスを使用して書き換えアクションを評価する

1. [書き換えアクション] の詳細ウィンドウで、評価する書き換えアクションを選択し、[評価] をクリックします。
2. [式エバリュエーターを書き換え] ダイアログボックスで、次のパラメータの値を指定します。(アスタリスクは必須パラメータを示します)。

   「書き換えアクション」(Rewrite Action)-評価する書き換えアクションがまだ選択されていない場合は、ドロップダウンリストから選択します。書き換えアクションを選択すると、「詳細」セクションに、選択した書き換えアクションの詳細が表示されます。
   「新規」(New)-「新規作成」(New) を選択して「書き換えアクションを作成」(Create Rewrite Action) ダイアログボックスを開

「変更」(Modify)-「修正」(Modify) を選択して「書き換えアクションを設定」(Configure) ダイアログボックスを開き、選択した書き換え

フロータイプ：選択した書き換えアクションを HTTP リクエストデータまたは HTTP 応答データのどちらでテストするかを指定します。デフォルトは [リクエスト] です。応答データでテストする場合は、[応答] を選択します。

HTTP 要求/応答データ *：書き換えアクションエバリュエーターがテストに使用する HTTP データを提供するスペースを提供します。データをウィンドウに直接貼り付けるか、[Sample] をクリックしてサンプル HTTP ヘッダーを挿入できます。

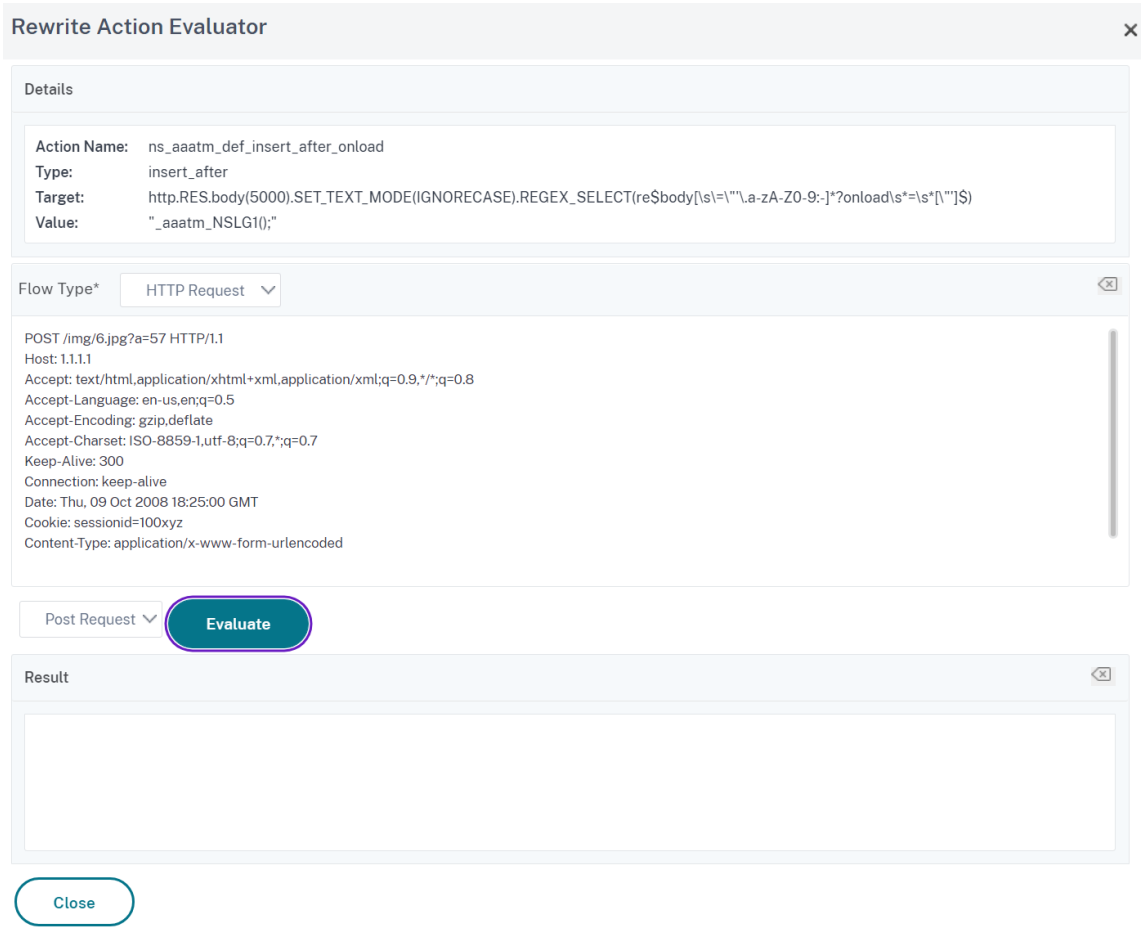
行末を表示-サンプル HTTP データの各行の末尾に UNIX スタイルの行末文字（\ n）を表示するかどうかを指定します。

[サンプル]：HTTP リクエスト/レスポンスデータウィンドウにサンプル HTTP データを挿入します。GET または POST データを選択できます。

[参照（Browse）]：ローカルブラウズウィンドウが開き、ローカルまたはネットワークロケーションのサンプル HTTP データを含むファイルを選択できます。

[クリア（Clear）]：[HTTP 要求/応答データ] ウィンドウから現在のサンプル HTTP データを消去します。

3. [評価] をクリックします。書き換えアクションエバリュエーターは、選択したサンプルデータに対する書き換えアクションの効果を評価し、[結果] ウィンドウで選択した [書き換え] アクションによって変更された結果を表示します。追加および削除は、ダイアログボックスの左下隅の凡例に示されているように強調表示されます。
4. すべてのアクションが希望する効果があると判断するまで、書き換えアクションを評価し続けます。
    - 選択した書き換えアクションを修正し、変更したバージョンをテストするには、[修正] をクリックして [書き換えアクションの構成] ダイアログボックスを開き、変更内容を保存し、もう一度 [評価] をクリックします。
    - 同じリクエストまたはレスポンスデータを使用して、別の書き換えアクションを評価するには、[書き換えアクション] ドロップダウンリストからそれを選択し、もう一度 [評価] をクリックします。
5. [閉じる] をクリックして、[式の書き換えエバリュエーター] を閉じ、[書き換え操作] ウィンドウに戻ります。
6. 書き換えアクションを削除するには、削除する書き換えアクションを選択し、「削除」をクリックし、プロンプトが表示されたら、「**OK**」をクリックして選択を確定します。



書き換えポリシーの構成

必要な書き換えアクションを作成したら、少なくとも 1 つの書き換えポリシーを作成して、NetScaler ADC アプライアンスに書き換える要求を選択する必要があります。

書き換えポリシーは、1 つ以上の式で構成される規則と、要求または応答が規則に一致した場合に実行される関連アクションで構成されます。HTTP リクエストとレスポンスを評価するためのポリシールールは、リクエストまたはレスポンスのほぼすべての部分に基づくことができます。

TCP ペイロード以外のデータの書き換えに TCP 書き換えアクションを使用することはできませんが、TCP 書き換えポリシーのポリシールールは、トランスポート層およびトランスポート層の下の層の情報に基づいて作成できます。

設定されたルールが要求または応答に一致すると、対応するポリシーがトリガーされ、それに関連付けられたアクションが実行されます。

> 注:
>
> コマンドラインインターフェイスまたは GUI を使用して、書き換えポリシーを作成および設定できます。コマンドラインインターフェイスと NetScaler ADC ポリシー式言語に精通していないユーザーは、通常、GUI の

> 使用がはるかに簡単になります。

コマンドラインインターフェイスを使用して新しい書き換えポリシーを追加するには

コマンドプロンプトで次のコマンドを入力して、新しい書き換えポリシーを追加し、構成を確認します。

- <add rewrite policy <name> <expression> <action> [<undefaction>]
- <show rewrite policy <name>

例 1. HTTP コンテンツの書き換え

```
> add rewrite policyNew "HTTP.RES.IS_VALID" insertact NOREWRITE
 Done
> show rewrite policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertact
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```

例 **2. TCP** ペイロードの書き換え **(TCP** 書き換え**):**

```
> add rewrite policy client_tcp_payload_policy CLIENT.IP.SRC.EQ
   (172.168.12.232) client_tcp_payload_replace_all
 Done
> show rewrite policy client_tcp_payload_policy
        Name: client_tcp_payload_policy
        Rule: CLIENT.IP.SRC.EQ(172.168.12.232)
        RewriteAction: client_tcp_payload_replace_all
        UndefAction: Use Global
        LogAction: Use Global
        Hits: 0
        Undef Hits: 0

 Done
>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して既存の書き換えポリシーを変更するには

コマンドプロンプトで次のコマンドを入力して、既存の書き換えポリシーを変更し、構成を確認します。

- <set rewrite policy <name> -rule <expression> -action <action> [<undefaction>]
- <show rewrite policy <name>

例**:**


```
> set rewrite policyNew -rule "HTTP.RES.IS_VALID" -action insertaction
 Done

> show rewrite policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertaction
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して書き換えポリシーを削除するには

コマンドプロンプトで次のコマンドを入力して、書き換えポリシーを削除します。

```
rm rewrite policy <name>
```

例:

```
> rm rewrite policyNew
Done
<!--NeedCopy-->
```

GUI を使用して書き換えポリシーを設定するには

1. [**AppExpert**] > [**Rewrite**] > [**Policies**] の順に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - ポリシーを作成するには、[Add] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。
3. [作成] または [**OK**] をクリックします。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。
4. 手順 2 ～4 を繰り返して、必要な数の書き換えアクションを作成または変更します。
5. [閉じる] をクリックします。書き換えポリシーを削除するには、削除する書き換えポリシーを選択し、「削除」をクリックし、プロンプトが表示されたら、「**OK**」をクリックして選択を確定します。

書き換えポリシーのバインド

書き換えポリシーを作成したら、それをバインドして有効にする必要があります。NetScaler ADC を通過するすべてのトラフィックにポリシーを適用する場合は、ポリシーをグローバルにバインドするか、ポリシーを特定の仮想サーバーまたはバインドポイントにバインドして、その仮想サーバーのみを転送するか、ポイントの着信トラフィックをそのポリシーにバインドできます。着信要求が書き換えポリシーと一致すると、そのポリシーに関連付けられたアクションが実行されます。

HTTP 要求および応答を評価するための書き換えポリシーは、HTTP または SSL タイプの仮想サーバーにバインドするか、REQ_OVERRIDE、
REQ_DEFAULT、RES_OVERRIDE、および RES_DEFAULT バインドポイントにバインドできます。TCP 書き換えの書き換えポリシーは、タイプ TCP または SSL_TCP の仮想サーバー、または OTHERTCP_REQ_OVERRIDE、OTHERTCP_REQ_DEFAULT、OTHERTCP_REQ_OVERRIDE、および
OTHERTCP_RES_DEFAULT バインドポイントにのみバインドできます。

> 注:
>
> OTHERTCP という用語は、NetScaler ADC アプライアンスのコンテキストで使用され、TCP パケットがカプセル化するプロトコルに関係なく、未加工のバイトストリームとして扱うすべての TCP または SSL_TCP 要求および応答を指します。

ポリシーをバインドするときは、そのポリシーにプライオリティを割り当てます。プライオリティによって、定義したポリシーが評価される順序が決まります。優先度は、任意の正の整数に設定できます。

NetScaler ADC オペレーティングシステムでは、ポリシーの優先度は逆の順序で動作します。数値が大きいほど、優先度は低くなります。たとえば、プライオリティが 10、100、1000 の 3 つのポリシーがある場合、プライオリティ 10 が割り当てられたポリシーが最初に適用され、次にポリシーにプライオリティ 100 が割り当てられ、最後にポリシーにオーダー 1000 が割り当てられます。

NetScaler ADC オペレーティングシステムの他のほとんどの機能とは異なり、書き換え機能は、要求がポリシーに一致した後も引き続きポリシーを評価および実装します。ただし、リクエストまたはレスポンスに対する特定のアクションポリシーの効果は、別のアクションの前で実行されるか後に実行されるかによって異なることがよくあります。優先度は、意図した結果を得るために重要です。

他のポリシーを任意の順序で追加するための十分な余裕を残し、バインドするときに各ポリシー間に 50 または 100 の間隔で優先順位を設定することで、希望の順序で評価されるように設定できます。これを実行すると、既存のポリシーのプライオリティを再割り当てすることなく、いつでもポリシーを追加できます。

書き換えポリシーをバインドする場合、ポリシーに goto 式 (gotoPriorityExpression) を割り当てるオプションもあります。goto 式には、goto 式を含むポリシーよりも高いプライオリティを持つ別のポリシーに割り当てられたプライオリティに一致する任意の正の整数を指定できます。goto 式をポリシーに割り当てて、要求または応答がポリシーと一致すると、NetScaler ADC は優先順位が goto 式と一致するポリシーにすぐに移動します。現在のポリシーよりも小さいが、goto 式のプライオリティ番号よりも高いプライオリティ番号を持つポリシーはすべてスキップされ、それらのポリシーは評価されません。

コマンドラインインターフェイスを使用して書き換えポリシーをグローバルにバインドするには

コマンドプロンプトで次のコマンドを入力して、書き換えポリシーをグローバルにバインドし、構成を確認します。

- `bind rewrite global <policyName> <priority> [<gotoPriorityExpression>] [-type <type>] [-invoke (<labelType> <labelName>)]`
- `show rewrite global`

例:


```
>bind rewrite global policyNew 10
 Done

> show rewrite global
1)      Global bindpoint: RES_DEFAULT
        Number of bound policies: 1

2)      Global bindpoint: REQ_OVERRIDE
        Number of bound policies: 1

 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して書き換えポリシーを特定の仮想サーバーにバインドするには

コマンドプロンプトで次のコマンドを入力して、書き換えポリシーを特定の仮想サーバーにバインドし、構成を確認します。

- `bind lb vserver <name>@ (<serviceName>@ [-weight <positive_integer>])| <serviceGroupName>@ | (-policyName <string>@ [-priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST | RESPONSE )] [-invoke (<labelType> <labelName>)] )`
- `show lb vserver <name>`

例:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 50
 Done
>
> show lb vserver lbvip
        lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jul 15 05:54:24 2009 (+226 ms)
        Time since last state change: 28 days, 01:57:26.350
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1)      Policy : ns_cmp_msapp Priority:50
2)      Policy : cf-pol Priority:1      Inherited
 Done
```



```
<!--NeedCopy-->
```

GUI を使用して書き換えポリシーをバインドポイントにバインドするには

1. **AppExpert** > 書き換え > ポリシーに移動します。
2. 詳細ペインで、グローバルにバインドする書き換えポリシーを選択し、[**Policy Manager**] をクリックします。
3. [ポリシーマネージャの書き換え] ダイアログボックスの [バインドポイント] メニューで、次のいずれかの操作を行います。

   a) HTTP 書き換えポリシーのバインディングを構成する場合は、[**HTTP**] をクリックし、要求ベースの書き換えポリシーと応答ベースの書き換えポリシーのどちらを構成するかに応じて、[要求] または [応答] をクリックします。

   b) TCP 書き換えポリシーのバインドを構成する場合は、[**TCP**] をクリックし、クライアント側の TCP 書き換えポリシーを構成するか、サーバー側の TCP 書き換えポリシーを構成するかに応じて、[クライアント] または [サーバー] をクリックします。

4. 書き換えポリシーをバインドするバインドポイントをクリックします。[**Rewrite Policy Manager**] ダイアログボックスには、選択したバインドポイントにバインドされているすべての書き換えポリシーが表示されます。
5. [**Insert Policy**] をクリックして新しい行を挿入し、使用可能なバインドされていない書き換えポリシーをすべて備えたドロップダウンリストを表示します。
6. バインドポイントにバインドするポリシーをクリックします。ポリシーは、バインドポイントにバインドされた書き換えポリシーのリストに挿入されます。
7. [優先度] 列では、優先度を任意の正の整数に変更できます。このパラメータの詳細については、「書き換えポリシーをバインドするためのパラメータ」の「priority」を参照してください。
8. 現在のポリシーが一致する場合、ポリシーをスキップして特定のポリシーに直接移動する場合は、[Goto Expression] カラムの値を、適用する次のポリシーのプライオリティと等しくなるように変更します。このパラメータの詳細については、「書き換えポリシーをバインドするためのパラメータ」の「gotoPriorityExpression」を参照してください。
9. ポリシーを変更するには、ポリシーをクリックし、[ポリシーの変更] をクリックします。
10. ポリシーのバインドを解除するには、ポリシーをクリックし、[ポリシーのバインド解除] をクリックします。
11. アクションを変更するには、[アクション] 列で変更するアクションをクリックし、[アクションの変更] をクリックします。
12. 呼び出しラベルを変更するには、[呼び出し] 列で、変更する呼び出しラベルをクリックし、[呼び出しラベルの変更] をクリックします。
13. 現在設定しているバインドポイントにバインドされているすべてのポリシーのプライオリティを再生成するには、[**Regenerate Priorities**] をクリックします。ポリシーは、他のポリシーと比較して既存の優先度を保持しますが、優先度は 10 の倍数で再番号付けされます。
14. [変更を適用] をクリックします。
15. [閉じる] をクリックします。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

GUI を使用して書き換えポリシーを特定の仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバーの詳細ウィンドウの一覧で、書き換えポリシーをバインドする仮想サーバーを選択し、[開く] をクリックします。
3. [仮想サーバーの構成 **(**負荷分散**)**] ダイアログボックスで、[ポリシー] タブを選択します。NetScaler ADC で構成されているすべてのポリシーがリストに表示されます。
4. この仮想サーバにバインドするポリシーの名前の横にあるチェックボックスをオンにします。
5. [**OK**] をクリックします。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

書き換えポリシーラベルの設定

単一のポリシーでサポートされるよりも複雑なポリシー構造を構築する場合は、ポリシーラベルを作成し、ポリシーと同様にバインドできます。ポリシーラベルは、ポリシーがバインドされるユーザー定義のポイントです。ポリシーラベルが呼び出されると、そのラベルにバインドされているすべてのポリシーが、設定したプライオリティの順序で評価されます。ポリシーラベルには、1 つまたは複数のポリシーを含めることができ、それぞれに独自の結果を割り当てることができます。ポリシー・ラベル内の 1 つのポリシーが一致すると、次のポリシーに進み、別のポリシー・ラベルまたは適切なリソースを呼び出すか、またはポリシー評価が即時に終了し、ポリシー・ラベルを呼び出したポリシーに制御が戻る場合があります。

書き換えポリシーラベルは、名前、ポリシーラベルに含まれるポリシーのタイプを示すトランスフォーム名、およびポリシーラベルにバインドされたポリシーのリストで構成されます。ポリシーラベルにバインドされている各ポリシーには、書き換えポリシーの設定で説明されているすべての要素が含まれます。

注: コマンドラインインターフェイスまたは GUI を使用して、書き換えポリシーラベルを作成および設定できます。コマンドラインインターフェイスと NetScaler ADC ポリシーインフラストラクチャ（PI）言語に精通していないユーザーは、通常、GUI の使用がはるかに簡単になります。

コマンドラインインターフェイスを使用して書き換えポリシーラベルを設定するには

書き換えポリシーラベルを追加するには、コマンドプロンプトで次のコマンドを入力します。

```
add rewrite policylabel <labelName> <transform>
```

たとえば、PollabelHttpResponses という名前の書き換えポリシーラベルを追加して、HTTP 応答で動作するすべてのポリシーをグループ化するには、次のように入力します。

```
add rewrite policy label polLabelHTTPResponses http_res
```

既存の書き換えポリシーラベルを変更するには、**NetScaler ADC** コマンドプロンプトで次のコマンドを入力します。

```
set rewrite policy <name> <transform>
```


> 注:
>
> set rewrite policy コマンドは、追加書き換えポリシーコマンドと同じオプションを取ります。

書き換えポリシーラベルを削除するには、**NetScaler ADC** コマンドプロンプトで次のコマンドを入力します。

```
rm rewrite policy<name>
```

たとえば、pollabelHttpResponses という名前の書き換えポリシーラベルを削除するには、次のように入力します。

```
rm rewrite policy polLabelHTTPResponses
```

GUI を使用して書き換えポリシーラベルを設定するには

1. **AppExpert** > 書き換え > ポリシーラベルに移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - ポリシーラベルを作成するには、[**Add**] をクリックします。
   - 既存のポリシーラベルを変更するには、ポリシーを選択し、[開く] をクリックします。
3. ポリシーラベルにバインドされているリストからポリシーを追加または削除します。
   - リストにポリシーを追加するには、[**Insert Policy**] をクリックし、ドロップダウンリストからポリシーを選択します。ポリシーを作成してリストに追加するには、リストで [New Policy] を選択し、書き換えポリシーの設定の手順に従います。
   - リストからポリシーを削除するには、そのポリシーを選択し、[ポリシーのバインド解除] をクリックします。
4. [Priority] 列の数値を編集して、各ポリシーの優先度を変更します。
   また、[優先順位の再生成] をクリックして、ポリシーの番号を自動的に再設定することもできます。
5. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。
   ポリシーラベルを削除するには、ポリシーラベルを選択し、[削除] をクリックします。ポリシーラベルの名前を変更するには、ポリシーラベルを選択し、[名前の変更] をクリックします。ポリシーの名前を編集し、[**OK**] をクリックして変更を保存します。

ストリーミング書き換えアクションでの **Content-Length** ヘッダーの動作

August 15, 2023

Content-Length ヘッダーは、HTTP リクエストまたはレスポンス内のメッセージの長さ (バイト単位) を示す方法の 1 つです。Content-Length ヘッダーとは別に、次のいずれかの方法でメッセージの長さを指定することもできます。

- チャンクエンコーディング

- FIN ターミネーション

ストリーミングプロセスでは、NetScaler は書き換えアクションを処理した後もデータを継続的に送信します。データは継続的に送信され、NetScaler には保持されないため、クライアントに送信されるメッセージの実際の長さは不明です。そのため、Content-Length ヘッダーの正しい値をレスポンスに記載することはできません。

ストリーミング処理をサポートするために、NetScaler の書き換え機能では、メッセージの長さを指定する方法が Content-Length ヘッダーから FIN ターミネーションに変換されます。変換の一環として、NetScaler はヘッダー名の最初の 4 文字を再配置することで Content-Length ヘッダーを壊します。

HTTP では、クライアントは理解できないヘッダーを無視することが予想されます。そのため、クライアントは破損した Content-Length ヘッダー名を認識できないため、ヘッダーを無視します。NetScaler のパフォーマンスを向上させるため、ヘッダーは削除されるのではなく破損されます。削除する代わりにヘッダー名を変更すると、同じバイトの順序が異なる場合でもチェックサムは変更されないため、チェックサムの再計算を回避できます。

たとえば、次の HTTP リクエストを考えてみましょう。

```
GET / HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml,
     image/gif, image/pjpeg, application/x-ms-xbap, application/vnd.ms-
    excel, application/vnd.ms-powerpoint, application/msword, /
Accept-Language: en-GB
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;
    Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR
     3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; CMDTDF; MS-RTC
    LM 8)
Accept-Encoding: gzip, deflate
Host: test.example.net
Connection: Keep-Alive
<!--NeedCopy-->
```

作業シナリオでは、この HTTP リクエストに対する NetScaler とバックエンドサーバー間の応答は次のようになります。

```
HTTP/1.1 200 OK
Content-Length: 10967
Connection: close
var SERVER_URL = 'https\x3a\x2f\x2ftest.example.net\x2f';
var WEB_SERVER_HOST = 'test.example.net';
<!--NeedCopy-->
```

ただし、動作していないシナリオでクライアントが NetScaler から受け取る応答は次のとおりです。`Content-Length` ヘッダーの名前が`ntcoent-Length`に変更されました。

```
HTTP/1.1 200 OK
ntCoent-Length: 10967
nnCoection: close
var SERVER_URL = 'https\x3a\x2f\x2ftest.example.net\x2f';
var WEB_SERVER_HOST = 'test.example.net';
<!--NeedCopy-->
```


一般に、クライアントアプリケーションは Content-Length ヘッダー、チャンクエンコーディング、FIN ターミネーションの 3 つのトランザクション方法すべてをサポートします。そのため、Content-Length ヘッダーから FIN ターミネーションに変換しても問題は発生しないはずです。ただし、この変更によりアプリケーションが動作しない場合は、ストリーミングプロセスを無効にする必要があります。

リライトポリシーでストリーミングプロセスを無効にする方法

次のいずれかの方法を使用して、リライトポリシーのストリーミングプロセスを無効にできます。

1. より高い優先度でバインドされている書き換えポリシーに関連する非ストリーミングアクションを追加します。アクションは、応答を変更しないような方法で行う必要があります。

   次に例を示します:

   ```
   add rewrite action non_stream_act replace_all HTTP.RES.BODY
   (1000000)HTTP.RES.FULL_HEADER -search text("pattern_which_will_not_match_i
   ")
   ```

   この書き換えアクションの本文の値は、現在のストリーミングアクションが動作している値よりも大きくなければなりません。

2. ストリーミング設定の代わりに、非ストリーミング設定を使用してください。

   > 注:
   >
   > ストリーミング処理から非ストリーミング処理に移行すると、NetScaler のパフォーマンスに影響する可能性があります。

   たとえば、ストリーミング構成は、次のように非ストリーミング構成に変換できます。

   ストリーミング設定**:**

   ```
   add rewrite action rw_act_1 replace_all HTTP.RES.BODY(1000) ""http
      "" -search text("http")

   add policy patset pat_list
   bind policy patset pat_list abcd
   bind policy patset pat_list defg

   add rewrite action rw_act_2 replace_all HTTP.RES.BODY(1000) ""
      replaced_data"" -search patset("pat_list")
   <!--NeedCopy-->
   ```

   非ストリーミング構成**:**

   ```
   add rewrite action rw_act_1 replace_all HTTP.RES.BODY(1000) ""http
      "" -search regex(re/http/)

   add rewrite action rw_act_1 replace_all HTTP.RES.BODY(1000) ""http
      "" -search regex(re/abcd|defg/)
   ```



```
<!--NeedCopy-->
```

書き換えアクションとポリシーの例

August 15, 2023

このセクションの例は、さまざまな便利なタスクを実行するようにリライトを設定する方法を示しています。例としては、Example Manufacturing Inc. のサーバールームがあります。Example Manufacturing Inc. は、自社の Web サイトを使用して販売、配送、カスタマーサポートのかなりの部分を管理している中規模の製造会社です。

Example Manufacturing には 2 つのドメインがあります。1 つはウェブサイトと顧客への電子メール用の example.com で、もう 1 つはイントラネット用の example.net です。お客様は、Example Web サイトを使用して、注文、見積もりの依頼、製品の調査、カスタマーサービスやテクニカルサポートへの連絡を行います。

Example の収益源の重要な部分として、Web サイトは迅速に対応し、顧客データの機密性を維持する必要があります。そのため、Example には複数の Web サーバーがあり、NetScaler アプライアンスを使用して Web サイトの負荷を分散し、Web サーバーとの間で送受信されるトラフィックを管理しています。

Example システム管理者は書き換え機能を使用して以下のタスクを実行します。

例 **1:** 古い **X-Forwarded-For** ヘッダーとクライアント **IP** ヘッダーの削除

Example Inc. は、受信リクエストから古い X-Forwarded-For ヘッダーとクライアント IP HTTP ヘッダーを削除します。

例 **2**: ローカルの **Client-IP** ヘッダーの追加

Example Inc. は、受信リクエストに新しいローカルクライアント IP ヘッダーを追加します。

例 **3**: 安全な接続と安全でない接続のタグ付け

Example Inc. は、受信リクエストに、接続が安全な接続かどうかを示すヘッダーをタグ付けします。

例 **4**: **HTTP** サーバータイプをマスク

Example Inc. は HTTP Server: ヘッダーを変更して、権限のないユーザーや悪意のあるコードがそのヘッダーを使用して使用している HTTP サーバーソフトウェアを特定できないようにします。

例 **5**: 外部 **URL** を内部 **URL** にリダイレクト

Example Inc. は、Web サイト上の URL を短く覚えやすくし、サイトのセキュリティを向上させるために、Web サーバーの実際の名前やサーバールームの構成に関する情報をユーザーから隠しています。

例 **6**: **Apache** 書き換えモジュール規則の移行

Example Inc. は、Apache リライトルールを NetScaler アプライアンスに移行し、Apache Perl ベースのスクリプト構文を NetScaler リライトルール構文に変換しました。

例 **7**：マーケティングキーワードのリダイレクト

Example Inc. のマーケティング部門は、会社の Web サイトで事前に定義された特定のキーワード検索用の簡略化された URL を設定します。

例 **8:** クエリをクエリされたサーバーにリダイレクトする。

Example Inc. は特定のクエリリクエストを適切なサーバーにリダイレクトします。

例 **9**：ホームページのリダイレクト

Example Inc. は最近、小規模な競合他社を買収し、買収した会社のホームページへのリクエストを自社の Web サイトのページにリダイレクトするようになりました。

例 **10**：ポリシーベースの **RSA** 暗号化

Example Inc. は、PEM RSA 公開鍵を使用して、HTTP の定義済みおよびユーザー定義のヘッダーまたは本文コンテンツを暗号化します。

これらの各タスクでは、システム管理者が書き換えアクションとポリシーを作成し、それらを NetScaler 上の有効なバインドポイントにバインドする必要があります。

例 **1**：古い **X-Forwarded-For** および **Client-IP** ヘッダーの削除

August 15, 2023

Example Inc. は、受信リクエストから古い X-Forwarded-For ヘッダーと Client-IP HTTP ヘッダーを削除して、表示される X-Forwarded-For ヘッダーがローカルサーバーによって追加されたヘッダーだけになるようにしたいと考えています。この構成は、NetScaler コマンドラインまたは構成ユーティリティを使用して実行できます。Example Inc. のシステム管理者は昔ながらのネットワークエンジニアで、可能な場合は CLI を使用することを好みますが、チームの新しいシステム管理者にその使用方法を説明できるように、構成ユーティリティのインターフェイスを理解していることを確認したいと考えています。

以下の例は、CLI と構成ユーティリティの両方を使用して各構成を実行する方法を示しています。この手順は、ユーザーが書き換えアクションの作成、書き換えポリシーの作成、およびバインドポリシーの基本をすでに知っていると仮定して省略されています。

- 書き換えアクションの作成の詳細については、「書き換えアクションの設定」を参照してください。
- 書き換えポリシーの作成の詳細については、「書き換えポリシーの設定」を参照してください。
- 書き換えポリシーのバインドの詳細については、「書き換えポリシーのバインド」を参照してください。

コマンドラインインターフェイスを使用して要求から古い **X-Forwarded** ヘッダーとクライアント **IP** ヘッダーを削除するには

コマンドプロンプトで、次のコマンドを次の順序で入力します。


```
add rewrite action act_del_xfor delete_http_header x-forwarded-for
add rewrite action act_del_cip delete_http_header client-ip
add rewrite policy pol_check_xfor 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS' act_del_xfor
add rewrite policy pol_check_cip 'HTTP.REQ.HEADER("client-ip").EXISTS' act_del_cip
bind rewrite global pol_check_xfor 100 200
bind rewrite global pol_check_cip 200 300
<!--NeedCopy-->
```

構成ユーティリティを使用してリクエストから古い **X-Forwarded** ヘッダーと **Client-IP** ヘッダーを削除するには

[書き換えアクションの作成] ダイアログボックスで、次の説明を使用して 2 つの書き換えアクションを作成します。

名前	種類	引数 (複数可)
act_del_xfor	delete_http_header	x-forwarded-for
act_del_cip	delete_http_header	client-ip

「書き換えポリシーの作成」ダイアログ・ボックスで、次の説明を含む 2 つの書き換えポリシーを作成します。

名前	式	アクション
pol_check_xfor	‘HTTP.REQ.HEADER(“x-forwarded-for”).EXISTS’	act_del_xfor
pol_check_cip	‘HTTP.REQ.HEADER(“client-ip”).EXISTS’	act_del_cip

両方のポリシーをグローバルにバインドし、次に示す優先順位と goto 式の値を割り当てます。

名前	優先度	**Goto** 式
pol_check_xfor	100	200
pol_check_cip	200	300

古い X-Forwarded-For および Client-IP の HTTP ヘッダーはすべて、受信リクエストから削除されるようになりました。

例 2: ローカルの Client-IP ヘッダーの追加

August 15, 2023

Example Inc. は、受信リクエストにローカルのクライアント IP HTTP ヘッダーを追加したいと考えています。この例には、同じ基本タスクの 2 つのわずかに異なるバージョンが含まれています。

コマンドラインインターフェイスを使用してローカル Client-IP ヘッダーを追加するには

コマンドプロンプトで、次のコマンドを次の順序で入力します。

```
add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.IP.SRC'
add rewrite policy pol_ins_client 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' act_ins_client
bind rewrite global pol_ins_client 300 END
<!--NeedCopy-->
```

構成ユーティリティを使用してローカル Client-IP ヘッダーを追加するには

[書き換えアクションの作成] ダイアログボックスで、次の説明を含む書き換えアクションを作成します。

名前	種類	引数 (複数可)
act_ins_client	insert_http_header	NS-クライアント ‘CLIENT.IP.SRC’

[書き換えポリシーの作成] ダイアログボックスで、次の説明を含む書き換えポリシーを作成します。

名前	式	アクション
pol_ins_client	‘HTTP.REQ.HEADER(“x-forwarded-for”).EXISTS || HTTP.REQ.HEADER(“client-ip”).EXISTS’	act_ins_client

ポリシーをグローバルにバインドし、以下に示す優先順位と goto 式の値を割り当てます。

名前	優先度	Goto 式
pol_ins_client	100	Next

例 3: 安全な接続と安全でない接続のタグ付け

August 15, 2023

Example Inc. は、受信リクエストに、接続が安全な接続であるかどうかを示すヘッダーでタグ付けしたいと考えています。これにより、NetScaler が接続を復号した後も、サーバーは安全な接続を追跡できます。

この構成を実装するには、まず次の表に示す値を使用して書き換えアクションを作成します。これらのアクションでは、ポート 80 への接続は安全でない接続として、ポート 443 への接続は安全な接続としてラベル付けされます。

アクション名	書き換えアクションのタイプ	ヘッダー名	値
Action-Rewrite-SSL_YES	INSERT_HTTP_HEADER	SSL	はい

アクション名	書き換えアクションのタイプ	ヘッダー名	値
Action-Rewrite-SSL_NO	INSERT_HTTP_HEADER	SSL	いいえ

次に、次の表に示す値を使用してリライトポリシーを作成します。これらのポリシーは、受信リクエストをチェックして、どのリクエストがポート 80 に送信され、どのリクエストがポート 443 に送信されるかを判断します。次に、ポリシーによって正しい SSL ヘッダーが追加されます。

ポリシー名	アクション名	未定義のアクション	式
Policy-Rewrite-SSL_YES	Action-Rewrite-SSL_YES	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(443)
Policy-Rewrite-SSL_NO	Action-Rewrite-SSL_NO	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(80)

最後に、書き換えポリシーを NetScaler にバインドし、最初のポリシーに 200、2 番目のポリシーに優先順位 300 を割り当て、両方のポリシーの goto 表現を END に設定します。

これで、ポート 80 への各受信接続に SSL: NO HTTP ヘッダーが追加され、ポート 443 への各受信接続には SSL: YES HTTP ヘッダーが追加されます。

例 4: HTTP サーバータイプをマスク

August 15, 2023

Example Inc. は、HTTP Server: ヘッダーを変更して、不正なユーザーや悪意のあるコードがヘッダーを使用して HTTP サーバーが使用するソフトウェアを識別できないようにしたいと考えています。

HTTP Server: ヘッダーを変更するには、次の表の値を使用して書き換えアクションと書き換えポリシーを作成します。

[アクション名]	書き換えアクションのタイプ	ターゲットリファレンスを選択する式	置換テキストの文字列式
Action-Rewrite-Server_Mask	REPLACE	HTTP.RES.HEADER(“Server”)	“Web Server 1.0”

ポリシー名	[アクション名]	未定義のアクション	式
Policy-Rewrite-Server_Mask	Action-Rewrite-Server_Mask	NOREWRITE	HTTP.RES.IS_VALID

コマンドの例:

```
> add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("Server")"\"Web Server 1.0\""

> add rewrite policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
```

次に、書き換えポリシーをグローバルにバインドし、優先度 100 を割り当て、ポリシーの Goto Priority 式を END に設定します。

HTTP Server: ヘッダーは、Example Inc. Web サイトで使用される実際の HTTP サーバーソフトウェアをマスキングして、「Web サーバー 1.0」と読み取るように変更されました。

例 5: 外部 URL を内部 URL にリダイレクトする

August 15, 2023

Example Inc. は、Web サーバーのセキュリティを向上させるために、実際のサーブルームの設定をユーザーから隠したいと考えています。

セキュリティを強化するには、次の表に示す値を使用して書き換えアクションを作成します。リクエストヘッダーの場合、テーブル内のアクションはwww.example.comを web.hq.example.net に変更します。レスポンスヘッダーの場合、アクションは逆の動作をして、web.hq.example.netをwww.example.comに変換します。

[アクション名]	書き換えアクションのタイプ	ターゲットリファレンスを選択する式	置換テキストの文字列式
Action-Rewrite-Request_Server_Replace	REPLACE	HTTP.REQ.HOSTNAME.SERVER	“Web.hq.example.net”
Action-Rewrite-Response_Server_Replace	REPLACE	HTTP.RES.HEADER(“Server”)	“www.example.com”

最初のポリシーは、受信するリクエストが有効かどうかをチェックします。有効な場合は、Action-Rewrite-Request_Server_Replace アクションが実行されます。2 番目のポリシーは、応答がサーバーweb.hq.example.netから発信されているかどうかを確認します。その場合、Action-Rewrite-Response_Server_Replace アクションが実行されます。

外部 URL をリダイレクトするための書き換えアクションとポリシーの例。

```
add rewrite action Action-Rewrite-Request_Server_Replace REPLACE HTTP
.REQ.HOSTNAME.SERVER '"Web.hq.example.net"'

add rewrite action Action-Rewrite-Response_Server_Replace REPLACE
HTTP.RES.HEADER("Server")'"www.example.com"'

add rewrite policy Rewrite-Request_Server_Replace HTTP.REQ.HOSTNAME
.SERVER.EQ("www.example.com")Action-Rewrite-Request_Server_Replace
NOREWRITE

add rewrite policy Rewrite-Response_Server_Replace HTTP.REQ.HEADER("
Server").EQ("Web.hq.example.net")Action-Rewrite-Response_Server_Replace
```

最後に、書き換えポリシーをバインドし、それぞれに優先度 500 を割り当てます。これは、書き換えポリシーが異なるポリシーバンクにあり、競合しないためです。両方のバインディングで goto 式を NEXT に設定します。

```
bind rewrite global Policy-Rewrite-Request_Server_Replace 500 END -
type REQ_DEFAULT

bind rewrite global Policy-Rewrite-Response_Server_Replace 500 END -
type RES_DEFAULT
```


リクエストヘッダーのすべてのwww.example.comのインスタンスがweb.hq.example.netに変更され、レスポンスヘッダーのすべてのweb.hq.example.netのインスタンスがwww.example.comに変更されるようになりました。

例 6: Apache 書き換えモジュール規則の移行

August 15, 2023

Example Inc. は現在、Apache rewrite モジュールを使用して Web サーバーに送信された検索リクエストを処理し、リクエスト URL の情報に基づいてそれらのリクエストを適切なサーバーにリダイレクトしています。Example Inc. は、これらのルールを NetScaler プラットフォームに移行することでセットアップを簡素化したいと考えています。

Example が現在使用しているいくつかの Apache リライトルールを以下に示します。これらのルールは、SiteID 文字列がない場合や SiteID 文字列がゼロ (0) の場合は検索リクエストを特別な結果ページにリダイレクトし、これらの条件に当てはまらない場合は標準結果ページにリダイレクトします。

以下は、現在の Apache 書き換えルールです。

- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} !SiteId= [OR]
- RewriteCond %{QUERY_STRING} SiteId=0
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ results2.html [P,L]
- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ /results.html [P,L]

これらの Apache リライトルールを NetScaler に実装するには、次の表の値を使用してリライトアクションを作成します。

[アクション名]	書き換えアクションのタイプ	ターゲットリファレンスを選択する式	置換テキストの文字列式
Action-Rewrite-Display_Results_NulSiteID	REPLACE	HTTP.REQ.URL	“/results2.html”
Action-Rewrite-Display_Results	REPLACE	HTTP.REQ.URL	“/results2.html”

次に、以下の表に示す値を使用してリライトポリシーを作成します。

ポリシー名	[アクション名]	未定義のアクション	式
Policy-Rewrite-Display_Results_NulSiteID	Action-Rewrite-Display_Results_NulSiteID	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MO “/search”) && (!HTTP.REQ.URL.QUERY.CONTAINS(“SiteId=”) || HTTP.REQ.URL.QUERY.CONTAINS(“SiteId=0”) || HTTP.REQ.URL.QUERY.SET_TEXT_M “Call-Name=DisplayResults”))
Policy-Rewrite-Display_Results	Action-Rewrite-Display_Results	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MO “/search”) || HTTP.REQ.URL.QUERY.SET_TEXT_M “Call-Name=DisplayResults”))

最後に、書き換えポリシーをバインドして、最初のポリシーに優先順位を 600、2 番目のポリシーに優先度 700 を割り当て、両方のバインディングの goto 式を NEXT に設定します。

NetScaler ADC は、Apache 書き換えモジュールのルールが移行される前の Web サーバーと同じようにこれらの検索要求を処理するようになりました。

例 7: マーケティングキーワードのリダイレクト

August 15, 2023

Example Inc. のマーケティング部門は、会社の Web サイトでの特定の定義済みキーワード検索に簡略化された URL を設定したいと考えています。これらのキーワードについては、以下に示すように URL を再定義する必要があります。

- 外部 URL:

```
http://www.example.com/\<marketingkeyword\>
```

- 内部 URL:

```
http://www.example.com/go/kwsearch.asp?keyword=\<marketingkeyword\>
```


マーケティングキーワードのリダイレクトを設定するには、次の表の値を使用してリライトアクションを作成します。

[アクション名]	書き換えアクションのタイプ	ターゲット位置を選択するための表現	置換テキストの文字列式
Action-Rewrite-Modify_URL	INSERT_BEFORE	HTTP.REQ.URL.PATH.GET(1)	“go/k-wsearch.aspkeyword=”|”

次に、次の表の値を使用して書き換えポリシーを作成します。

ポリシー名	[アクション名]	未定義のアクション	式
Policy-Rewrite-Modify_URL	Action-Rewrite-Modify_URL	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ(“www.example.com”)

最後に、書き換えポリシーをバインドし、優先度 800 を割り当てます。以前の書き換えポリシーとは異なり、条件に一致するリクエストにはこのポリシーを最後に適用する必要があります。このため、NetScaler 管理者は「優先度指定に移動」を「END」に設定します。

マーケティングキーワードを使用するリクエストはすべてキーワード検索 CGI ページにリダイレクトされ、そこで検索が実行され、残りのポリシーはすべてスキップされます。

例 8: クエリをクエリされたサーバーにリダイレクト

August 15, 2023

Example Inc. は、次に示すように、クエリ要求を適切なサーバーにリダイレクトしたいと考えています。

- <Request: GET /query.cgi?server=5HOST: www.example.com
- <Redirect URL: <http://web-5.example.com/>

このリダイレクトを実装するには、まず次の表の値を使用して書き換えアクションを作成します。

[アクション名]	書き換えアクションのタイプ	ターゲットリファレンスを選択する式	置換テキストの文字列式
Action-Rewrite-Replace_Hostheader	REPLACE	HTTP.REQ.HEADER(“Host”).BEFORE_STR(“.example.com”)	“server-“+ HTTP.REQ.URL.QUERY.VALUE(“web”)

次に、次の表の値を使用して書き換えポリシーを作成します。

ポリシー名	[アクション名]	未定義のアクション	式
Policy-Rewrite-Replace_Hostheader	Action-Rewrite-Replace_Hostheader	NOREWRITE	HTTP.REQ.HEADER("Host").EQ("www.example.com")

コマンドの例:

```
> add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("Server")"\"Web Server 1.0\""
Done

> add rewrite policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
Done
```

最後に、書き換えポリシーをバインドし、優先度 900 を割り当てます。このポリシーは、条件に一致するリクエストに適用される最後のポリシーである必要があるため、goto 式を END に設定します。

<http://www.example.com/query.cgi?server>=で始まる URL への着信要求は、クエリのサーバ番号にリダイレクトされます。

例 9: ホームページのリダイレクト

August 15, 2023

New Company, Inc. は最近、小規模な競合企業である Purchased Company を買収したので、次に示すように、Purchased Company のホームページを自社 Web サイトの新しいページにリダイレクトしたいと考えています。

- 古い URL: http://www.purchasedcompany.com/%5C*
- 新しい URL: http://www.newcompany.com/products/page.htm

リクエストを購入した会社のホームページにリダイレクトするには、次の表の値を使用してリライトアクションを作成します。

[アクション名]	書き換えアクションのタイプ	ターゲットリファレンスを選択する式	置換テキストの文字列式
Action-Rewrite-Replace_URLr	REPLACE	HTTP.REQ.URL.PATH_AND_QUERY	“/products/page.htm”
Action-Rewrite-Replace_Host	REPLACE	HTTP.REQ.HOSTNAME	“www.newcompany.com”

```
add rewrite action action-Rewrite-Replace_URLr REPLACE HTTP.REQ.URL.
    PATH_AND_QUERY  “/products/page.htm”

add rewrite action action-Rewrite-Replace_Host REPLACE HTTP.REQ.
    HOSTNAME  “www.newcompany.com”
<!--NeedCopy-->
```

次に、次の表の値を使用してリライトポリシーを作成します。

ポリシー名	[アクション名]	未定義のアクション	式
Policy-Rewrite-Replace-None	Action-Rewrite-Replace-None	NOREWRITE	!HTTP.REQ.HOSTNAME.SERVER.EQ(“www.purchasedcompany.com”)
Policy-Rewrite-Replace-Host	Action-Rewrite-Replace_Host	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ(“www.purchasedcompany.com”)

```
add rewrite policy Policy-Rewrite-Replace-None !HTTP.REQ.HOSTNAME.
    SERVER.EQ( “www.purchasedcompany.com” ) Action-Rewrite-Replace-None
    NOREWRITE

add rewrite policy Policy-Rewrite-Replace-Host HTTP.REQ.HOSTNAME.SERVER
    .EQ( “www.purchasedcompany.com” ) Action-Rewrite-Replace_Host
    NOREWRITE
<!--NeedCopy-->
```

最後に、書き換えポリシーをグローバルにバインドして、最初のポリシーに優先度を 100、2 番目のポリシーに優先度 200 を割り当てます。

```
bind rewrite global Policy-Rewrite-Replace-None 100

bind rewrite global Policy-Rewrite-Replace-Host 200
<!--NeedCopy-->
```

買収した会社の旧ウェブサイトへのリクエストが、新会社のホームページの正しいページにリダイレクトされるようになりました。

例 **10**：ポリシーベースの **RSA** 暗号化

August 15, 2023

RSA アルゴリズムは PKEY_ENCRYPT_PEM () 関数を使用して、HTTP の事前定義およびユーザー定義のヘッダーまたは本文コンテンツを暗号化します。この関数は RSA 公開鍵のみを受け入れ (秘密鍵は受け付けません)、暗号化されたデータは公開鍵の長さを超えることはできません。暗号化されるデータがキーの長さよりも短い場合、アルゴリズムは RSA_PKCS1 のパディング方式を使用します。

サンプルシナリオでは、この関数を B64ENCODE () 関数と書き換えアクションで使用して、HTTP ヘッダー値を RSA 公開鍵で暗号化された値に置き換えることができます。暗号化中のデータは、受信者によって RSA 秘密鍵を使用して復号化されます。

この機能は、書き換えポリシーを使用して実装できます。そのためには、次のタスクを完了する必要があります。

1. RSA 公開鍵をポリシー表現として追加します。
2. 書き換えアクションを作成します。
3. 書き換えポリシーを作成します。
4. リライトポリシーをグローバルとしてバインドします。
5. RSA 暗号化の検証

NetScaler コマンドインターフェイスを使用したポリシーベースの **RSA** 暗号化

NetScaler コマンドインターフェイスを使用してポリシーベースの RSA 暗号化を構成するには、次のタスクを完了します。

NetScaler コマンドインターフェイスを使用して **RSA** 公開鍵をポリシー表現として追加するには:

```
1 add policy expression pubkey '"-----BEGIN RSA PUBLIC KEY-----
     MIGJAoGBAKl5vgQEj73Kxp+9
     yn1v5gPR1pnc4oLM2a0kaWwBOsB6rzCIy6znwnvwCY1xRvQhRlJSAyJbloL7wZFIJ2FOR8Cz
     +8ZQWXU2syG+udi4EnWqLgFYowF9zK+o79az597eNPAjsHZ/C2oL/+6qY5a/
     f1z8bQPrHC4GpFfAEJhh/+NnAgMBAAE=-----END RSA PUBLIC KEY-----"'
2 <!--NeedCopy-->
```

NetScaler コマンドインターフェイスを使用して **HTTP** ヘッダーリクエストを暗号化するアクションを書き換えるには:

```
add rewrite action encrypt_act insert_http_header encrypted_data
HTTP.REQ.HEADER("data_to_encrypt").PKEY_ENCRYPT_PEM(pubkey).B64ENCODE
```

NetScaler コマンドインターフェイスを使用して書き換えポリシーを追加するには:


```
add rewrite policy encrypt_pol 'HTTP.REQ.HEADER("data_to_encrypt").
   EXISTS' encrypt_act
<!--NeedCopy-->
```

NetScaler コマンドインターフェイスを使用してリライトポリシーをグローバルにバインドするには:

```
bind rewrite global encrypt_pol 10 -type RES_DEFAULT
```

NetScaler コマンドインターフェイスを使用して **RSA** 暗号化を検証するには:

```
>curl -v -H "data_to_encrypt: Now is the time that tries men's souls"
   http://10.217.24.7/`

* About to connect() to 10.217.24.7 port 80 (#0)

*  Trying 10.217.24.7...

*  connected

*  Connected to 10.217.24.7 (10.217.24.7) port 80 (#0)

> GET / HTTP/1.1
> User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0
   OpenSSL/0.9.8y zlib/1.2.3
> Host: 10.217.24.7
> Accept: \*/\*
> data_to_encrypt: Now is the time that tries men's souls
>
< HTTP/1.1 200 OK
< Date: Mon, 09 Oct 2017 05:22:37 GMT
< Server: Apache/2.2.24 (FreeBSD) mod_ssl/2.2.24 OpenSSL/0.9.8y DAV/2
< Last-Modified: Thu, 20 Feb 2014 20:29:06 GMT
< ETag: "6bd9f2-2c-4f2dc5b570880"
< Accept-Ranges: bytes
< Content-Length: 44
< Content-Type: text/html
< encrypted_data: UliegKBJqZd7JdaC49XMLEK1+eQN2rEfevypW91gKvBVlaKM9N9/
   C2BKuztS99SE0xQaisidzN5IgeIcpQMn+
   CiKYVlLzPG1RuhGaqHYzIt6C8A842da7xE4OlV5SHwScqkqZ5aVrXc3EwtUksna7jOLr40aLeXnnB
   /DB11pUAE=
<
* Connection #0 to host 10.217.24.7 left intact
<html><body><h1>It works!</h1></body></html>* Closing connection #0

<!--NeedCopy-->
```

同じ暗号化データを使用してこの curl コマンドを後から実行すると、暗号化されたデータが実行ごとに異なることがわかります。これは、パディングが暗号化するデータの先頭にランダムなバイトを挿入するため、暗号化されたデータが毎回異なるためです。

```
>curl -v -H "data_to_encrypt: Now is the time that tries men's souls"
   http://10.217.24.7/`

```



```
< encrypted_data:
   DaOjtl1Pl4DlQKf58MMeL4cFwFvZwhjMqv5aUYM5Iyzk4UpwIYhpRvgTNu2lXEVc1H0tcR1EGC
   /ViQncLc4EbTurCWLbzjce3+fknnMmzF0lRT6ZZXWbMvsNFOxDA1SnuAgwxWXy/
   ooe9Wy6SYsL2oi1sr5wTG+RihDd9zP+P14=

>curl -v -H "data_to_encrypt: Now is the time that tries men's souls"
   http://10.217.24.7/

. . .

< encrypted_data: eej6YbGP68yHn48qFUvi+fkG+OiO8j3yYLScrRBU+
   TPQ8WeDVaWnDNAVLvL0ZYHHAU1W2YDRYb+8
   cdKHLpW36QbI6Q5FfBuWKZSI2hSyUvypTpCoAYcHXFv0ns+tRtg0EPNNj+
   lyGjKQWtFi6K8IXXISoDy42FblKIlaA7gEriY=
<!--NeedCopy-->
```

GUI によるポリシーベースの RSA 暗号化

GUI では、次のタスクを実行できます。

GUI を使用して **RSA** 公開鍵をポリシー表現として追加するには:

1. **NetScaler** アプライアンスにサインインし、［構成］**>**［AppExpert］**>**［高度な表現］に移動します。
2. 詳細ペインで [追加] をクリックして、RSA 公開鍵を高度なポリシー表現として定義します。
3. エクスプレッションの作成ページで、次のパラメータを設定します。

   a) エクスプレッション名。拡張エクスプレッションの名前。
   b) 式。RSA 公開鍵をエクスプレッションエディタを使用して高度なエクスプレッションとして定義します。
   c) [コメント]。表現の簡単な説明。

4. ［作成］をクリックします。

GUI を使用して **HTTP** ヘッダーリクエストを暗号化するアクションを書き換えるには:

1. **NetScaler** アプライアンスにサインインし、**[**** 構成] > [**AppExpert**] > [書き換え] > [アクション] に移動します。**
2. 詳細ウィンドウで、[追加] をクリックして書き換えアクションを追加します。
3.「書き換えアクションの作成」画面で、次のパラメータを設定します。

   a) [名前]。書き換えアクションの名前。
   b) 種類。アクションタイプを INSERT_HTTP_HEADER として選択します。
   c) アクションタイプを使用してヘッダーを挿入します。書き換えが必要な HTTP ヘッダーの名前を入力します。
   d) 式。アクションに関連するアドバンスポリシー表現の名前。
   e) [コメント]。書き換えアクションの簡単な説明。

4. ［作成］をクリックします。

GUI を使用してリライト・アドバンスド・ポリシーを追加するには:

1. **NetScaler** アプライアンスにサインインし、**[**** 構成] > [**AppExpert**] > [リライト] > [ポリシー] に移動します。******
2.「リライトポリシー」ページで、「追加」をクリックしてリライトポリシーを追加します。
3.「リライトポリシーの作成」ページで、次のパラメータを設定します。

   a) [名前]。リライトポリシーの名前。
   b) アクション。リクエストまたはレスポンスがこのリライトポリシーに一致した場合に実行するリライトアクションの名前。
   c) ログアクション。リクエストがこのポリシーに一致する場合に使用するメッセージログアクションの名前。
   d) 未定義の結果のアクション。ポリシー評価の結果が未定義の場合に実行するアクション。
   e) 式。アクションをトリガーする高度なポリシー表現の名前。
   f) [コメント]。書き換えアクションの簡単な説明。

4. [作成] をクリックします。

GUI を使用してリライトポリシーをグローバルにバインドするには:

1. **NetScaler** アプライアンスにサインインし、**[**** 構成] > [**AppExpert**] > [リライト] > [ポリシー] に移動します。******
2.「リライトポリシー」画面で、バインドするリライトポリシーを選択し、「ポリシーマネージャ」をクリックします。
3.「ポリシーマネージャーの書き換え」ページの「バインドポイント」セクションで、次のパラメータを設定します。

   a) バインドポイント。バインディングポイントをデフォルトグローバルとして選択します。
   b) プロトコル。プロトコルタイプを HTTP として選択します。
   c) 接続タイプ。接続タイプをリクエストとして選択します。
   d)「続行」をクリックすると、「ポリシーバインディング」セクションが表示されます。
   e)「ポリシー・バインディング」セクションで、リライト・ポリシーを選択し、バインド・パラメータを設定します。

4. [**Bind**] をクリックします。

例 **11:** パディング操作なしのポリシーベースの **RSA** 暗号化

August 15, 2023

PKEY_ENCRYPT_PEM_NO_PADDING () ポリシー関数は RSA 暗号化を実行する前に、パディング操作なしで RSA アルゴリズムを使用します。ポリシー関数は PKEY_ENCRYPT_PEM () 関数と同じように機能しますが、

RSA_PKCS1_PADDING の代わりに RSA_NO_PADDING メソッドを使用する点が異なります。pkey パラメータは、PEM でエンコードされた RSA 公開鍵を含むテキスト文字列です。PKEY_ENCRYPT_PEM () と同様に、キーにはポリシーエクスプレッションを使用できます。

この機能は、書き換えポリシーを使用して実装できます。そのためには、次のタスクを完了する必要があります。

1. RSA 公開鍵をポリシー表現として追加します。
2. 書き換えアクションを作成します。

NetScaler コマンドインターフェイスを使用したポリシーベースの **RSA** 暗号化

NetScaler コマンドインターフェイスを使用してポリシーベースの RSA 暗号化を構成するには、次のタスクを完了します。

NetScaler コマンドインターフェイスを使用してパディングポリシー表現なしの **RSA** 公開鍵を追加するには:

```
add expression rsa_pub_key_4096 '"-----BEGIN RSA PUBLIC KEY-----" + "
    MIICCgKCAgEArrwBldKd48xrpOSRPMrg+eNAO0ODU6t5b/WYQLdElqNv7WpefBrA" +
    "nwI2s619gEU1r4zoLqL7l5ALtt5Z+F0JBYfOzBzOky0GtEJ5iX5GP4QxT65J3nHH" +
     "4MTF3acmjvXxclmaKXEFlaVIzW7FTr3Luw/CnOjflAB403Q6F9VBVvQmOVYWnqoI"
    + "+0q1VIg6Q1pAcvdKBiOf85BBofE5EIbZ/1Jt0CdbsV568l+8ve7BnSUncFHoRR3O"
     + "/VfSsDuNWZf7n3RNMzxEuIA72UGPzNYFQzvcPOdzd0aN7jAXw0mgC/NSvKzGKHlo
    " + "mUYYBzlVQdDMZWnd6jSzsBRXSXxsNEy/
    RuXwplrA5epo7JdCoMkfeI4vUXm6MNr8" + "
    TQdFqIc1pdnOsbRf9ec62XbcfR7P8CDTsmLSaagx3rjenPdB+LTWKw2VUF+YONIg" +
    "jM3fyFef9ovVhLhS5HvMqFGs8P75W+d7BOIbIu3EngACiEJOpYSsETD4WgPK6Iyv" +
     "j6cxsLeYMtElTb0fBIIqysCHdmjF3M1lqdqp4dKs3+W798GJZYM5MxZKUzrBi0Xu"
    + "e7GtSh2aimsfQureUD+0z0RN2umeDsYcA1ghXMclDP+jLS1lnrv0Yvo+TKcm9b8G"
     + "uR/drbCrCsGyWFW+bsAu3AWz9S6TePurP5unRmNNvXpH5DRgsYl3d50CAwEAAQ
    ==" + "-----END RSA PUBLIC KEY-----"
<!--NeedCopy-->
```

NetScaler コマンドインターフェイスを使用してパディングポリシー表現なしの書き換えアクションを追加するには:

```
add rewrite action rsa_encrypt_act insertHttpHeader encrypted 'HTTP.
REQ.HEADER("plaintext").PKEY_ENCRYPT_PEM_NO_PADDING(rsa_pub_key_4096)
```

GUI によるパディングオプションなしのポリシーベースの **RSA** 暗号化

GUI では、次のタスクを実行できます。

GUI を使用してパディングなし操作の **RSA** 公開鍵をポリシー表現として追加するには**:**

1. **NetScaler** アプライアンスにサインインし、［構成］**>**［AppExpert］**>**［高度な表現］に移動します。
2. 詳細ペインで [追加] をクリックして、RSA 公開鍵を高度なポリシー表現として定義します。

3. エクスプレッションの作成ページで、次のパラメータを設定します。

    a) エクスプレッション名。拡張エクスプレッションの名前。
    b) 式。RSA 公開鍵をエクスプレッションエディタを使用して高度なエクスプレッションとして定義します。注: ポリシー表現の最大文字列長は 255 文字です。1024 ビットを超えるキーの場合は、キーを小さなチャンクに分割し、チャンクを「chunk1」+「chunk2」+…として連結する必要があります。
    c) [コメント]。表現の簡単な説明。

4. ［作成］をクリックします。

GUI を使用してアクションを追加**/**書き換えるには**:**

1. **NetScaler** アプライアンスにサインインし、**[**** 構成] > [**AppExpert**] > [書き換え] > [アクション] に移動します。******
2. 詳細ウィンドウで、[追加] をクリックして書き換えアクションを追加します。
3.「書き換えアクションの作成」画面で、次のパラメータを設定します。

    a) [名前]。書き換えアクションの名前。
    b) 種類。アクションタイプを INSERT_HTTP_HEADER として選択します。
    c) アクションタイプを使用してヘッダーを挿入します。書き換えが必要な HTTP ヘッダーの名前を入力します。
    d) 式。アクションに関連するアドバンスポリシー表現の名前。
    e) [コメント]。書き換えアクションの簡単な説明。

4. ［作成］をクリックします。

例 **12**: **NetScaler ADC** アプライアンスのクライアント要求でホスト名と **URL** を変更するように書き換えを構成する

August 15, 2023

NetScaler ADC アプライアンスの書き換え機能を使用して、クライアント要求で使用可能な URL を、バックエンドサーバーが理解できる別の URL に変換します。書き換え機能を使用すると、次のメリットが得られます。

- クライアントから要求されたリソースへの実際の URL を非表示にすることで、セキュリティを強化します。
- 不正なユーザーアクセスがネットワークリソースにアクセスするのを防ぎます。

現在の組織が別の組織に買収された例を考えてみましょう。管理者にとって、買収した組織のすべてのユーザーに新しいウェブアドレスを知らせるのは大変な作業になります。このシナリオでは、取得した組織の Web サイトに対するクライアントリクエストのホスト名や URL を変更するときに、書き換え機能を使用すると便利になります。ウェブサイトがメンテナンス中のときに、rewrite を使用してクライアントリクエストの URL を一時的に変更できます。

次のセクションでは、書き換え機能を使用してクライアントリクエストのホスト名と URL を変更する手順について説明します。

ユーザーが Web ブラウザに `http://www.example.com` URL を入力する例を考えてみましょう。Web サイト管理者は、NetScaler ADC アプライアンスにクライアント要求内の前述の URL を次のように変換させたいと考えています。`http://myexample.example.net.in/resource/inventory/s?t=112`

前述の例では、Web サイト管理者は NetScaler アプライアンスで「example.com」ドメイン名を「myexample.example.net.in」に、URL を「resource/inventory/s?」に置き換えることを望んでいます。t=12 インチ。

CLI を使用して、次の操作を実行します

1. SSH を使用して NetScaler ADC アプライアンスにログオンします。
2. 書き換えアクションを追加する。
   - `add rewrite action rewrite_doman_url_repalce_act replace HTTP.REQ.URL "\"http://myexample.example.net.in/resource/inventory/s?t=112\""`
3. 書き換えアクションの書き換えポリシーを追加します。
   - `add rewrite policy rewrite_domain_url_pol HTTP.REQ.HOSTNAME.EQ("www.example.com")rewrite_doman_url_repalce_act`
4. 書き換えポリシーを仮想サーバにバインドします。
   - `bind lb vserver rewrite_LB -policyName rewrite_domain_url_pol -priority 100 -gotoPriorityExpression END -type REQUEST`

URL 変換

August 15, 2023

URL 変換機能を使用すると、指定されたリクエストに含まれるすべての URL を、外部ユーザーが参照できる外部バージョンから、Web サーバーと IT スタッフだけが見る内部 URL に変更できます。ネットワーク構造をユーザーに公開することなく、ユーザー要求をシームレスにリダイレクトできます。また、ユーザーが覚えにくい複雑な内部 URL を、よりシンプルで覚えやすい外部 URL に変更することもできます。

> 注
>
> URL 変換機能を使用する前に、書き換え機能を有効にする必要があります。書き換え機能を有効にするには、書き換え機能の有効化を参照してください。

> URL 変換機能は、HTML レスポンス本文の URL を書き換え、JavaScript やその他の変数には適用されません。

URL 変換の設定を開始するには、それぞれが特定の変換を説明するプロファイルを作成します。各プロファイル内に、変換を詳細に説明するアクションを 1 つ以上作成します。次に、ポリシーを作成します。各ポリシーは、変換する HTTP リクエストのタイプを識別し、各ポリシーを適切なプロファイルに関連付けます。最後に、各ポリシーをグローバルにバインドして有効にします。

URL 変換プロファイルの構成

August 15, 2023

プロファイルは、特定の URL 変換を一連のアクションとして記述します。プロファイルは主にアクションのコンテナとして機能し、アクションが実行される順序を決定します。ほとんどの変換は、外部ホスト名とオプションパスを別の内部ホスト名とパスに変換します。ほとんどの便利な変換は単純で 1 つのアクションしか必要としませんが、複数のアクションを使用して複雑な変換を実行できます。

アクションを作成してからプロファイルに追加することはできません。最初にプロファイルを作成し、そのプロファイルにアクションを追加する必要があります。CLI では、アクションの作成とアクションの設定は別々のステップです。プロファイルの作成とプロファイルの設定は、CLI と構成ユーティリティの両方で別々の手順です。

NetScaler コマンドラインを使用して URL 変換プロファイルを作成するには

NetScaler のコマンドプロンプトで、次のコマンドを順番に入力して、URL 変換プロファイルを作成し、構成を確認します。その後、2 番目と 3 番目のコマンドを繰り返して追加のアクションを設定できます。

- `add transform profile <profileName> -type URL [-onlyTransformAbsURLinBody (ON|OFF)] \[-comment <comment>]`
- `add transform action <name> <profileName> <priority>`
- `set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainFrom <expression>] [-cookieDomainInto <expression>] [-state (ENABLED|DISABLED)] [-comment "<string>"]`
- `show transform profile <name>`

例:

```
> add transform profile shoppingcart -type URL
 Done
> add transform action actshopping shoppingcart 1000
```



```
 Done
> set transform action actshopping -priority 1000 -reqUrlFrom 'shopping
   .example.com' -reqUrlInto 'www.example.net/shopping' -resUrlFrom '
   www.example.net/shopping' -resUrlInto 'shopping.example.com' -
   cookieDomainFrom 'example.com' -cookieDomainInto 'example.net' -
   state ENABLED -comment 'URL transformation for shopping cart.'
 Done
> show transform profile shoppingcart
        Name: shoppingcart
                Type: URL       onlyTransformAbsURLinBody: OFF
        Comment:
        Actions:

1)              Priority 1000   Name: actshopping       ENABLED
 Done
<!--NeedCopy-->
```

NetScaler コマンドラインを使用して既存の URL 変換プロファイルまたはアクションを変更するには

NetScaler のコマンドプロンプトで、次のコマンドを入力して既存の URL 変換プロファイルまたはアクションを変更し、構成を確認します。

注: トランスフォームプロファイルの設定またはトランスフォームアクションの設定コマンドをそれぞれ使用してください。set transform profile コマンドは add transform profile コマンドと同じ引数を取ります。set transform action は初期設定に使用されたコマンドと同じです。

- `set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainInto <expression>] [-state (ENABLED|DISABLED)] [-comment "<string>"]`
- `show transform profile <name>`

例:

```
> set transform action actshopping -priority 1000 -reqUrlFrom '
   searching.example.net' -reqUrlInto 'www.example.net/searching' -
   resUrlFrom 'www.example.net/searching' -resUrlInto 'searching.
   example.com' -cookieDomainInto 'example.net' -state ENABLED -comment
    'URL transformation for searching cart.'
 Done
> show transform profile shoppingcart
        Name: shoppingcart
                Type: URL       onlyTransformAbsURLinBody: OFF
        Comment:
        Actions:

1)              Priority 1000   Name: actshopping       ENABLED
 Done
<!--NeedCopy-->
```


NetScaler コマンドラインを使用して **URL** 変換プロファイルとアクションを削除するには

まず、アクションごとに次のコマンドを 1 回入力して、そのプロファイルに関連付けられているすべてのアクションを削除します。

- rm transform action <name> プロファイルに関連するすべてのアクションを削除したら、次に示すようにプロファイルを削除します。
- rm トランスフォームプロファイル <name>

構成ユーティリティを使用して **URL** 変換プロファイルを作成するには

1. ナビゲーションペインで、[書き換え]、[URL 変換] の順に展開し、[プロファイル] をクリックします。
2. 詳細ペインで、[追加] をクリックします。
3. **URL** 変換プロファイルの作成ダイアログボックスで、パラメータの値を入力または選択します。ダイアログボックスの内容は、「URL 変換プロファイルを構成するためのパラメータ」で説明されているパラメータに次のように対応します (アスタリスクは必須パラメータを示します)。
    - 名前 * —名前
    - Comment—comment
    - レスポンス本文の絶対 URL のみを変換する—Absurlin 本文のみを変換する
4. [作成] をクリックし、[閉じる] をクリックします。プロファイルが正常に構成されたことを示すメッセージがステータスバーに表示されます。

構成ユーティリティを使用して **URL** 変換プロファイルとアクションを設定するには

1. ナビゲーションペインで、[書き換え]、[URL 変換] の順に展開し、[プロファイル] をクリックします。
2. 詳細ペインで、設定するプロファイルを選択し、「開く」をクリックします。
3. **URL** 変換プロファイルの設定ダイアログボックスで、次のいずれかを実行します。
    - 新しいアクションを作成するには、[追加] をクリックします。
    - 既存のアクションを変更するには、アクションを選択し、[開く] をクリックします。
4. パラメータの値を入力または選択して、「**URL** 変換アクションの作成」または「**URL** 変換アクションの変更」ダイアログボックスに入力します。ダイアログボックスの内容は、「URL 変換プロファイルを構成するためのパラメータ」で説明されているパラメータに次のように対応します (アスタリスクは必須パラメータを示します)。
    - アクション名 *—name
    - Comments—comment
    - Priority*—priority
    - URL のリクエスト元—リクエスト元

- URL 情報をリクエストする—リクエストする
- レスポンス URL 送信元—返信元
- レスポンス URL の宛先—戻る
- クッキードメインの送信元—クッキードメインの送信元
- クッキードメイン情報—クッキードメイン情報
- 有効—状態

5. 変更を保存します。
   - 新しいアクションを作成する場合は、「作成」をクリックし、「閉じる」をクリックします。
   - 既存のアクションを変更する場合は、「**OK**」をクリックします。
     ステータス・バーに、プロファイルが正常に構成されたことを示すメッセージが表示されます。
6. ステップ 3 からステップ 5 を繰り返して、追加のアクションを作成または変更します。
7. アクションを削除するには、アクションを選択し、[削除] をクリックします。メッセージが表示されたら、OK をクリックして削除を確定します。
8. 「**OK**」をクリックして変更を保存し、「URL 変換プロファイルの変更」ダイアログ・ボックスを閉じます。
9. プロファイルを削除するには、詳細ペインでプロファイルを選択し、[削除] をクリックします。メッセージが表示されたら、**OK** をクリックして削除を確定します。

URL 変換ポリシーの構成

August 15, 2023

URL 変換プロファイルを作成したら、次に URL 変換ポリシーを作成して、NetScaler がプロファイルを使用して変換する要求と応答を選択します。URL 変換では、各要求とそれに対する応答が 1 つの単位と見なされるため、URL 変換ポリシーは、要求が受信されたときにのみ評価されます。ポリシーが一致すると、NetScaler は要求と応答の両方を変換します。

注: リクエスト処理中に、URL 変換機能と書き換え機能を両方とも同じ HTTP ヘッダーで動作させることはできません。このため、リクエストに URL 変換を適用する場合は、URL 変換によって変更される HTTP ヘッダーがいずれの書き換えアクションによっても操作されないようにする必要があります。

NetScaler コマンドラインを使用して **URL** 変換ポリシーを構成するには

新しいポリシーを作成する必要があります。コマンドラインでは、既存のポリシーを削除することしかできません。NetScaler のコマンドプロンプトで、次のコマンドを入力して URL 変換ポリシーを構成し、構成を確認します。

- `<add transform policy <name> <rule> <profileName>`
- `<show transform policy <name>`

例:

```
> add transform policy polsearch  HTP.REQ.URL.SUFFIX.EQ("Searching")
   prosearching
 Done
> show transform policy polsearch
1)      Name: polsearch
        Rule: HTP.REQ.URL.SUFFIX.EQ("Searching")
        Profile: prosearching
        Priority: 0
        Hits: 0
 Done
<!--NeedCopy-->
```

NetScaler コマンドラインを使用して **URL** 変換ポリシーを削除するには

NetScaler のコマンドプロンプトで、次のコマンドを入力して URL 変換ポリシーを削除します。

```
rm transform policy <name>
```

例:

```
> rm transform policy polsearch
Done
<!--NeedCopy-->
```

構成ユーティリティを使用して **URL** 変換ポリシーを構成するには

1. ナビゲーションペインで、[リライト]、[URL 変換] の順に展開し、[ポリシー] をクリックします。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいポリシーを作成するには、[追加（**Add**）] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。
3. [**URL** 変換ポリシーの作成] または [**URL** 変換ポリシーの設定] ダイアログボックスで、パラメータの値を入力または選択します。ダイアログボックスの内容は、「URL 変換ポリシーを設定するためのパラメータ」で説明されているパラメータに次のように対応します (アスタリスクは必須パラメータを示します)。
   - name*-名前 (以前に設定したポリシーでは変更できません)
   - Profile*—profileName
   - エクスプレッション—ルール

   新しいポリシーのエクスプレッションの作成について不明な点がある場合は、Ctrl キーを押したままエクスプレッションテキストボックスにカーソルを置いたままスペースバーを押してください。式を作成するには、下記の説明に従って直接入力するか、「式の追加」(Add Expression) ダイアログボックスを使用できます。

4. 「プレフィックス」をクリックし、エクスプレッションのプレフィックスを選択します。

   選択肢は次のとおりです:

   - HTTP —HTTP プロトコルです。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。
   - SYS —保護された Web サイト。リクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
   - クライアント-要求を送信したコンピュータ。リクエストの送信者の側面を調べる場合は、これを選択します。
   - サーバー—要求が送信されたコンピューター。リクエストの受信者の何らかの側面を調べたい場合は、これを選択してください。
   - URL —リクエストの URL。リクエストの送信先の URL の何らかの側面を調べたい場合は、これを選択してください。
   - text —リクエスト内の任意のテキスト文字列。リクエスト内のテキスト文字列を調べたい場合は、これを選択してください。
   - ターゲット—リクエストのターゲット。リクエストターゲットの何らかの側面を調べたい場合は、これを選択してください。

   プレフィックスを選択すると、NetScaler は 2 つの部分からなるプロンプトウィンドウを表示し、上部に次の選択肢が表示され、下部には選択した選択肢の意味の簡単な説明が表示されます。選択は、選択したプレフィックスによって異なります。

5. 次の用語を選択してください。

プレフィックスとして HTTP を選択した場合、選択肢は HTTP リクエストを指定する REQ と HTTP レスポンスを指定する RES です。別のプレフィックスを選択した場合、選択肢はより多様になります。特定の選択肢に関するヘルプを表示するには、その選択肢を 1 回クリックすると、その選択に関する情報が下のプロンプトウィンドウに表示されます。

どの選択肢にするかが決まったら、それをダブルクリックしてエクスプレッションウィンドウに挿入します。

1. ピリオドを入力し、前のリストボックスの右側に表示されるリストボックスから続けて用語を選択します。式が完成するまで、表示されるテキストボックスに適切な文字列または数字を入力して値の入力を求めます。
2. 新しいポリシーを作成するのか、既存のポリシーを変更するのかに応じて、[Create] または [**OK**] をクリックします。
3. ［閉じる］をクリックします。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

[式の追加**]** ダイアログボックスを使用して式を追加するには

1. [レスポンダーアクションの作成] または [レスポンダーアクションの構成] ダイアログボックスで、[追加] をクリックします。
2. [式の追加] ダイアログボックスの最初のリストボックスで、式の最初の用語を選択します。
   - HTTP HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。
   - SYS。保護されている Web サイト。リクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
   - CLIENT. 要求を送信したコンピュータ。リクエストの送信者の側面を調べる場合は、これを選択します。
   - サーバー。要求の送信先のコンピュータ。リクエストの受信者の何らかの側面を調べたい場合は、これを選択してください。
   - URL。リクエストの URL。リクエストの送信先の URL の何らかの側面を調べたい場合は、これを選択してください。
   - テキスト。リクエスト内の任意のテキスト文字列。リクエスト内のテキスト文字列を調べたい場合は、これを選択してください。
   - ターゲット。リクエストのターゲット。リクエストターゲットの何らかの側面を調べたい場合は、これを選択してください。
     選択すると、右端のリストボックスに、式の次の部分に適した用語がリストされます。
3. 2 番目のリストボックスで、式の 2 番目の用語を選択します。選択肢は、前のステップで行った選択によって異なり、コンテキストに適切です。2 番目の選択を行った後、[式の構築] ウィンドウの下のヘルプウィンドウ (空白) に、選択した用語の目的と使用法を説明するヘルプが表示されます。
4. 式が終了するまで、前のリストボックスの右側に表示されるリストボックスから用語を選択するか、値の入力を求めるテキストボックスに文字列または数値を入力します。

グローバルにバインドされた **URL** 変換ポリシー

August 15, 2023

URL 変換ポリシーを設定したら、それらをグローバルまたはバインドポイントにバインドして有効にします。バインド後、URL 変換ポリシーと一致するリクエストまたはレスポンスは、そのポリシーに関連付けられたプロファイルによって変換されます。

ポリシーをバインドするときは、そのポリシーにプライオリティを割り当てます。プライオリティによって、定義したポリシーが評価される順序が決まります。優先度は、任意の正の整数に設定できます。NetScaler OS では、ポリシーの優先順位は逆の順序で機能します。数値が大きいほど、優先度は低くなります。

URL 変換機能では、リクエストが一致した最初のポリシーのみが実装され、一致する可能性のある追加のポリシーは実装されないため、意図した結果を得るにはポリシーの優先順位が重要です。最初のポリシーに低い優先度（1000 など）を与えた場合は、優先順位の高い他のポリシーが要求と一致しない場合にのみ実行するように NetScaler に指示します。最初のポリシーに高い優先度（1 など）を与える場合は、NetScaler にそのポリシーを最初に実行し、一致する可能性のある他のポリシーはすべてスキップするように指示します。ポリシーをグローバルにバインドするときに、各ポリシーの間に 50 または 100 の間隔で優先順位を設定することで、優先順位を再割り当てすることなく、他のポリシーを任意の順序で追加する余地を十分に確保できます。

注:URL 変換ポリシーを TCP ベースの仮想サーバーにバインドすることはできません。

NetScaler コマンドラインを使用して **URL** 変換ポリシーをバインドするには

NetScaler のコマンドプロンプトで次のコマンドを入力して、URL 変換ポリシーをグローバルにバインドし、構成を確認します。

- `bind transform global <policyName> <priority>`
- `show transform global`

例:

```
> bind transform global polisearching 100
 Done
> show transform global
1)      Policy Name: polisearching
        Priority: 100

 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して **URL** 変換ポリシーをバインドするには

1. ナビゲーションペインで、[書き換え]、[URL 変換] の順に展開し、[** ポリシー] をクリックします。
2. 詳細ペインで、[ポリシーマネージャ] をクリックします。
3. トランスフォームポリシーマネージャーダイアログボックスで、ポリシーをバインドするバインドポイントを選択します。選択肢は以下のとおりです。
   -グローバルオーバーライド。このバインドポイントにバインドされたポリシーは、**NetScaler** アプライアンス上のすべてのインターフェイスからのすべてのトラフィックを処理し、他のポリシーよりも先に適用されます。
   -LB 仮想サーバー。負荷分散仮想サーバーにバインドされたポリシーは、その負荷分散仮想サーバーによって処理されるトラフィックにのみ適用され、デフォルトのグローバルポリシーの前に適用されます。**LB** 仮想サーバーを選択したら、このポリシーをバインドする特定の負荷分散仮想サーバーも選択する必要があります。
   -CS 仮想サーバー。コンテンツスイッチング仮想サーバーにバインドされたポリシーは、そのコンテンツスイッチング仮想サーバーによって処理されるトラフィックにのみ適用され、デフォルトのグローバルポリシーの

前に適用されます。**CS** 仮想サーバーを選択したら、このポリシーをバインドする特定のコンテンツスイッチング仮想サーバーも選択する必要があります。

-デフォルトグローバル。このバインドポイントにバインドされたポリシーは、**NetScaler** アプライアンス上のすべてのインターフェイスからのすべてのトラフィックを処理します。

-ポリシーラベル。** ポリシーラベルにバインドされたポリシーは、ポリシーラベルがルーティングするトラフィックを処理します。ポリシーラベルは、このトラフィックにポリシーが適用される順序を制御します。

4. 「ポリシーを挿入」を選択して新しい行を挿入し、使用可能なすべてのバインドされていない URL 変換ポリシーを含むドロップダウンリストを表示します。
5. バインドするポリシーを選択するか、[New Policy] を選択して新しいポリシーを作成します。選択または作成したポリシーは、グローバルにバインドされた URL 変換ポリシーのリストに挿入されます。
6. バインディングをさらに調整します。
    - ポリシーの優先度を変更するには、フィールドをクリックして有効にし、新しい優先度を入力します。「優先順位を再作成」を選択して、優先順位を均等に再設定することもできます。
    - ポリシー表現を変更するには、そのフィールドをダブルクリックして [トランスフォームポリシーの設定（Configure Transform Policy）] ダイアログボックスを開き、ポリシー表現を編集できます。
    - Goto Expression を設定するには、Goto Expression 列の見出しにあるフィールドをダブルクリックしてドロップダウンリストを表示し、そこで式を選択できます。
    - Invoke オプションを設定するには、Invoke 列の見出しにあるフィールドをダブルクリックしてドロップダウンリストを表示し、そこで式を選択できます。
7. ステップ 3 ～6 を繰り返して、グローバルにバインドしたい URL 変換ポリシーを追加します。
8. [**OK**] をクリックして変更を保存します。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

リライト機能の **RADIUS** サポート

August 15, 2023

NetScaler の式言語には、要求や応答に含まれる RADIUS メッセージから情報を抽出したり操作したりできる式が含まれています。これらの式により、宛先に送信する前に、書き換え機能を使用して RADIUS メッセージの一部を変更できます。書き換えポリシーとアクションでは、RADIUS メッセージに適切な、または関連する任意の表現を使用できます。使用可能な式により、RADIUS メッセージタイプを識別したり、接続から任意の属性値ペア（AVP）を抽出したり、RADIUS AVP を変更したりできます。RADIUS 接続のポリシーラベルを作成することもできます。

リライトルールの新しい RADIUS 式は、さまざまな目的に使用できます。たとえば、次のことができます。

- シングルサインオン (SSO) を簡略化するには、RADIUS ユーザー名 AVP のドメイン\ 部分を削除してください。
- 電話会社の業務で使用する MSISDN フィールドなど、ベンダー固有の AVP を挿入して加入者情報を格納します。

ポリシーラベルを作成して、特定のタイプの RADIUS 要求を、それらの要求に適した一連のポリシーを通じてルーティングすることもできます。

> 注記:
>
> リライト用 RADIUS には次の制限があります。

- NetScaler は、書き換えられた RADIUS リクエストやレスポンスに再署名しません。RADIUS 認証サーバーが署名付き RADIUS メッセージを必要とする場合、認証は失敗します。
- 現在使用可能な RADIUS 式は RADIUS IPv6 属性では機能しません。

RADIUS をサポートする表現に関する NetScaler のドキュメントでは、RADIUS 通信の基本的な構造と目的に精通していることを前提としています。RADIUS の詳細については、RADIUS サーバのマニュアルを参照するか、RADIUS プロトコルの概要をオンラインで検索してください。

RADIUS の書き換えポリシーの設定

以下の手順では、NetScaler コマンドラインを使用して書き換えアクションとポリシーを構成し、ポリシーを書き換え固有のグローバルバインドポイントにバインドします。

書き換えアクションとポリシーを設定し、ポリシーをバインドするには:

コマンドプロンプトで、次のコマンドを入力します。

- `add rewrite action <actName> <actType>`
- `add rewrite policy <polName> <rule> <actName>`
- `bind rewrite policy <polName> <priority> <nextExpr> -type <bindPoint>`ここで、`<bindPoint>`はリライト固有のグローバルバインドポイントの 1 つを表します。

リライト用の **RADIUS** エクスプレッション

書き換え構成では、次の NetScaler 式を使用して RADIUS 要求または応答のさまざまな部分を参照できます。

接続タイプの識別:

- `RADIUS.IS_CLIENT`

  接続が RADIUS クライアント (要求) メッセージの場合は TRUE を返します。

- `RADIUS.IS_SERVER`

  接続が RADIUS サーバー (応答) メッセージの場合、TRUE を返します。

リクエスト表現:

- RADIUS.REQ.CODE

  RADIUS リクエストタイプに対応する番号を返します。num_at クラスの派生関数です。たとえば、RADIUS アクセス要求は 1 を返します。RADIUS アカウンティング要求では 4 が返されます。

- RADIUS.REQ.LENGTH

  ヘッダーを含む RADIUS リクエストの長さを返します。
  num_at クラスの派生関数です。

- RADIUS.REQ.IDENTIFIER

  RADIUS リクエスト識別子を返します。これは、リクエストを対応するレスポンスと一致させるために各リクエストに割り当てられる番号です。
  num_at クラスの派生関数です。

- RADIUS.REQ.AVP(<AVP Code No>).VALUE

  この AVP が最初に出現したときの値を
  text_t 型の文字列で返します。

- RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)

  AVP の指定されたインスタンスを RAVP_t 型の文字列として返します。特定の RADIUS AVP が 1 つの RADIUS メッセージに複数回表示されることがあります。INSTANCE (0) は最初のインスタンスを返し、INSTANCE (1) は 2 番目のインスタンスを返す、というように、最大 16 個のインスタンスを返します。

- RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)

  AVP の指定されたインスタンスの値を
  text_t 型の文字列として返します。

- RADIUS.REQ.AVP(<AVP code no>).COUNT

  RADIUS 接続内の特定の AVP のインスタンス数を整数で返します。

- RADIUS.REQ.AVP(<AVP code no>).EXISTS

  指定されたタイプの AVP がメッセージに存在する場合は TRUE を返し、存在しない場合は FALSE を返します。

レスポンス表現:

RADIUS レスポンス式は RADIUS リクエスト表現と同じですが、REQ が REQ に置き換わっている点が異なります。

AVP 値のタイプキャスト**:**

ADC は、RADIUS AVP 値をテキスト、整数、符号なし整数、ロング、符号なしロング、ipv4 アドレス、ipv6 アドレス、ipv6 プレフィックス、および時刻データ型にタイプキャストする式をサポートしています。構文は他の NetScaler タイプキャスト式と同じです。

例:

ADC は、RADIUS AVP 値をテキスト、整数、符号なし整数、ロング、符号なしロング、ipv4 アドレス、ipv6 アドレス、ipv6 プレフィックス、および時刻データ型にタイプキャストする式をサポートしています。構文は他の NetScaler タイプキャスト式と同じです。

```
RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at
<!--NeedCopy-->
```

AVP 型式:

NetScaler は、RFC2865 および RFC2866 で説明されている割り当てられた整数コードを使用して RADIUS AVP 値を抽出する式をサポートしています。テキストエイリアスを使用して同じタスクを実行することもできます。次にいくつかの例を示します。

- `RADIUS.REQ.AVP (1).VALUE or RADIUS.REQ.USERNAME.value`

  RADIUS ユーザー名の値を抽出します。

- `RADIUS.REQ.AVP (4). VALUE or RADIUS.REQ. ACCT\\_SESSION\\_ID. value`

  メッセージから ACCT セッション ID AVP (コード 44) を抽出します。

- `RADIUS.REQ.AVP (26). VALUE or RADIUS.REQ.VENDOR\\_SPECIFIC.VALUE`

  ベンダー固有の値を抽出します。

最も一般的に使用される RADIUS AVP の値も同じ方法で抽出できます。

RADIUS バインドポイント:

RADIUS 表現を含むポリシーには、4 つのグローバルバインドポイントを使用できます。

- `RADIUS_REQ_OVERRIDE`

  プライオリティ/オーバーライドリクエストポリシーキュー。

- `RADIUS_REQ_DEFAULT`

  標準リクエストポリシーキュー。

- `RADIUS_RES_OVERRIDE`

  プライオリティ/オーバーライドレスポンスポリシーキュー。

- `RADIUS_RES_DEFAULT`

  標準レスポンスポリシーキュー。

RADIUS リライト固有の表現**:**

- `RADIUS.NEW_AVP`

  指定した RADIUS AVP を文字列で返します。

- RADIUS.NEW_AVP_INTEGER32

  指定された RADIUS AVP を整数で返します。

- RADIUS.NEW_AVP_UNSIGNED32

  指定した RADIUS AVP を符号なし整数で返します。

- RADIUS.NEW_VENDOR_SPEC_AVP(<ID>, <definition>)

  指定された拡張ベンダー固有の AVP を接続に追加します。<ID>は、長い数字に置き換えてください。<definition>は、AVP のデータを含む文字列に置き換えてください。

- RADIUS.REQ.AVP_START

  RADIUS ヘッダーの終わりと AVP の開始点の間の位置を返します。書き換えアクションで使用されます。

  例:

```
1     add rewrite action insert1 insert_after radius.req.avp_start radius
          .new_avp(33, "NEW AVP")
2 <!--NeedCopy-->
```

- RADIUS.REQ.AVP_END

  RADIUS メッセージ内の RADIUS メッセージの最後 (つまり、すべての AVP の最後) の位置を返します。書き換えアクションを実行するときに使用されます。

  例:

```
1     add rewrite action insert2 insert_before radius.req.avp_end "radius
          .new_avp(33, "NEW AVP")"
2 <!--NeedCopy-->
```

- RADIUS.REQ.AVP_LIST

  RADIUS メッセージ内の AVP の開始位置と、ヘッダーを除く RADIUS メッセージの長さを返します。つまり、RADIUS メッセージ内のすべての AVP を返します。書き換えアクションを実行するために使用されます。

  例:

```
1     add rewrite action insert3 insert_before_all radius.req.avp_list "
          radius.new_avp(33, "NEW AVP")" -search "avp(33)"
2 <!--NeedCopy-->
```

RADIUS の有効なリライト・アクション・タイプ:

RADIUS エクスプレッションで使用できるリライトアクションタイプは次のとおりです。

- INSERT_AFTER
- INSERT_BEFORE
- INSERT_AFTER_ALL

- INSERT_BEFORE_ALL
- 削除
- DELETE_ALL
- REPLACE
- REPLACE_ALL

`INSERT_ actions` これらすべてを使用して、RADIUS AVP を RADIUS 接続に挿入できます。

使用例

書き換え機能を備えた RADIUS のユースケースを以下に示します。

ユーザー名 **AVP** の書き換え

RADIUS ユーザー名 AVP からドメイン\ 文字列を削除するように書き換え機能を設定するには、まず次の例に示すように、書き換え置換アクションを作成します。すべての RADIUS 要求を選択する書き換えポリシーでアクションを使用してください。ポリシーをグローバルバインドポイントにバインドします。その際、優先度を適切なレベルに設定して、ブロックポリシーまたは拒否ポリシーが最初に有効になるようにします。ただし、ブロックまたは拒否されていないすべてのリクエストは必ず書き直してください。Goto Expression (GoToPriorityExpr) を NEXT に設定してポリシー評価を続行し、ポリシーを RADIUS_REQ_DEFAULT キューにアタッチします。

例:

```
1 add rewrite action rwActRadiusDomainDel replace radius.req.user_name q/
     RADIUS.NEW_AVP(1,RADIUS.REQ.USER_NAME.VALUE.AFTER_STR(" "))/
2 add rewrite policy RadiusRemoveDomainPol true rwActRadiusDomainDel
3 <!--NeedCopy-->
```

> 注記:
>
> RADIUS の書き換えポリシーは、ゲートウェイ仮想サーバーには適用されません。ゲートウェイ仮想サーバーを負荷分散に使用する場合は、RADIUS を設定し、書き換えポリシーを RADIUS 負荷分散仮想サーバーにバインドする必要があります。

ベンダー固有の **AVP** の挿入

MSISDN フィールドの内容を含むベンダー固有の AVP を挿入するように書き換えアクションを設定するには、まず MSISDN フィールドをリクエストに挿入する書き換え INSERT アクションを作成します。すべての RADIUS 要求を選択する書き換えポリシーでアクションを使用してください。ポリシーをグローバルにバインドし、次の例に示すように、優先順位を適切なレベルに設定し、その他のパラメータを設定します。

例:


```
add rewrite action rwActRadiusInsMSISDN insert_after radius.req.
    avp_start RADIUS.NEW_VENDOR_SPEC_AVP(<VENDOR ID>, "RADIUS.NEW_AVP(<
    Attribute Code>, <MSISDN>)")
add rewrite policy rwPolRadiusInsMSISDN true rwActRadiusInsMSISDN
bind rewrite global rwPolRadiusInsMSISDN 100 NEXT -type
    RADIUS_REQ_DEFAULT
<!--NeedCopy-->
```

書き換えの **Diameter** サポート

August 15, 2023

リライト機能が Diameter プロトコルをサポートするようになりました。HTTP や TCP のリクエストやレスポンスと同じように Diameter のリクエストやレスポンスを変更するように Rewrite を設定できます。これにより、Rewrite を使用して Diameter リクエストのフローを管理し、必要な変更を加えることができます。たとえば、Diameter リクエストの「Origin-Host」値が不適切な場合は、Rewrite を使用して Diameter サーバーが許容できる値に置き換えることができます。

Diameter リクエストを変更するようにリライトを設定するには

直径リクエスト内のオリジンホストを別の値に置き換えるように書き換え機能を設定するには、コマンドプロンプトで次のコマンドを入力します。

- <add rewrite action <actname> replace “DIAMETER.REQ.AVP(264,\ “NetScaler.example.net\”)”
  For <actname>, substitute a name for your new action. 名前は 1 ～127 文字で、文字、数字、ハイフン (-) と下線 (_) を使用できます。NetScaler.Example.net の場合は、元のホスト名の代わりに使用したいホストオリジンに置き換えてください。
- add rewrite policy <polname> “diameter.req.avp(264).value.eq(\ “host.example.com\”)” <actname>
  For <polname>, substitute a name for your new policy. と同様に <actname>、名前は 1 ～127 文字で、文字、数字、ハイフン (-) と下線 (_) を使用できます。host.example.com には、変更するホストオリジンの名前に置き換えてください。<actname> の代わりに、作成したアクションの名前を使用してください。
- bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST
  For <vservername>, substitute the name of the load balancing virtual server to which you want to bind the policy. <polname> の代わりに、作成したポリシーの名前を使用してください。には <priority>、ポリシーの代わりに優先度を使用してください。

例:

「host.example.com」のすべての Diameter Host-Origin を「Netscaler.example.net」に変更する書き換えアクションとポリシーを作成するには、次のアクションとポリシーを追加し、図のようにポリシーをバインドできます。

```
> add rewrite action rw_act_replace_avp replace "diameter.req.avp(264)"
    "diameter.new.avp(264,"NetScaler.example.net")"
> add rewrite policy rw_diam_pol "diameter.req.avp(264).value.eq("
    client.realm2.net")" rw_act_replace_avp
> bind lb vserver vs1 -policyName rw_diam_pol -priority 10 -type
    REQUEST

Done
<!--NeedCopy-->
```

書き換え機能の **DNS** サポート

August 15, 2023

HTTP または TCP の要求と応答の場合と同様に、書き換え機能を設定して DNS 要求と応答を変更できます。rewrite を使用して DNS リクエストのフローを管理し、ヘッダーまたは回答セクションで必要な変更を加えることができます。たとえば、DNS 応答のヘッダーフラグに AA ビットが設定されていない場合、rewrite を使用して DNS 応答に AA ビットを設定し、それをクライアントに送信できます。

DNS エクスプレッション

書き換え構成では、次の NetScaler ADC 式を使用して、DNS 要求または応答のさまざまな部分を参照できます。

エクスプレッションと説明を参照してください。

DNS バインドポイント

DNS 表現を含むポリシーでは、次のグローバルバインドポイントを使用できます。

バインドポイント	説明
DNS_REQ_OVERRIDE	オーバーライドリクエストポリシーキュー。
DNS_REQ_DEFAULT	標準リクエストポリシーキュー。
DNS_RES_OVERRIDE	レスポンスポリシーキューをオーバーライドします。
DNS_RES_DEFAULT	標準レスポンスポリシーキュー。

デフォルトのバインドポイントに加えて、DNS_REQ または DNS_RES タイプのポリシーラベルを作成し、それらに DNS ポリシーをバインドできます。

DNS の書き換えアクションタイプ

- **replace_dns_answer_section**—このアクションにより、DNS 回答セクションが DNS ポリシーで定義されている式に置き換えられます。
- **replace_dns_header_field**—DNS リクエストのオペコードタイプをチェックします。DNS リクエストのオペコードタイプが指定されたオペコードタイプと一致するかどうかを示す True または False を返します。このアクションにより、DNS ヘッダーセクションが DNS ポリシーで定義されている式に置き換えられます。

DNS の書き換えポリシーの設定

以下の手順では、NetScaler コマンドラインを使用して書き換えアクションとポリシーを構成し、ポリシーを書き換え固有のグローバルバインドポイントにバインドします。

書き換えアクションとポリシーを設定し、ポリシーを **DNS** にバインドする

コマンドプロンプトで、次のコマンドを入力します。

1. `add rewrite action <actName> <actType>`

   <actname>は、新しいアクションの名前に置き換えてください。名前の長さは 1 ～127 文字で、文字、数字、ハイフン (-)、および下線 (_) 記号を使用できます。<actType>には、DNS 表現に用意されている書き換えアクションの種類を指定します。

2. `add rewrite policy <polName> <rule> <actName>`

   <polname>は、新しいポリシーの名前に置き換えてください。<actname>では、名前の長さは 1 ～127 文字で、文字、数字、ハイフン (-)、および下線 (_) 記号を使用できます。<actname>は、先ほど作成したアクションの名前に置き換えてください。

3. `bind rewrite global <polName> <priority> <gotoPriorityExpression> -type <bindPoint>`

   <polName>は、先ほど作成したポリシーの名前の代わりに使用してください。<priority>には、ポリシーの優先度を指定します。<bindPoint>を、リライト固有のグローバルバインドポイントの 1 つに置き換えてください。

例:

DNS リクエストの **AA** ビットを設定して仮想サーバーの負荷を分散します。

次のコマンドは、NetScaler アプライアンスが処理するすべてのクエリに対して権限のある DNS サーバーとして機能するように構成します。


```
add rewrite action set_aa replace_dns_header_field dns.req.header.flags
    .set(aa)
add rewrite policy pol !dns.req.header.flags.is_set(aa)  set_aa
bind rewrite global  pol  100  -type dns_res_override
<!--NeedCopy-->
```

回答とヘッダーセクションを変更します。

サーバーが NX ドメインで応答する場合、応答を指定した IP アドレスに置き換えるように書き換えアクションを設定できます。NOPOLICY-REWRITE を使用すると、式 (ルール) を処理せずに外部バンクを呼び出すことができます。このエントリは、ルールを含まないダミーポリシーですが、エントリをポリシーラベルまたは仮想サーバ固有のポリシーバンクに転送します。

```
add rewrite action set_aa_res replace_dns_header_field "dns.res.header.
    flags.set(aa)"
add rewrite action modify_nxdomain_res replace_dns_answer_section "dns.
    new_rrset_a("10.102.218.160",300)"
add rewrite policy set_res_aa true set_aa_res
add add rewrite policy modify_answer "dns.RES.HEADER.RCODE.EQ(nxdomain)
     && dns.RES.QUESTION.TYPE.EQ(A)"
modify_nxdomain_res
add rewrite policylabel MODIFY_NODATA dns_res
bind rewrite policylabel MODIFY_NODATA modify_answer 10 END
bind rewrite policylabel MODIFY_NODATA set_res_aa 11 END
bind lb vserver v1 -policyName NOPOLICY-REWRITE -priority 11 -
    gotoPriorityExpression END -type
RESPONSE -invoke policylabel MODIFY_NODATA
<!--NeedCopy-->
```

制限事項:

- 書き換えポリシーは、NetScaler アプライアンスが DNS プロキシサーバーとして構成されていてキャッシュミスがある場合にのみ評価されます。
- ヘッダーの Recursion Available (RA) フラグが YES に設定されている場合、RA フラグは書き換え時に変更されません。
- ヘッダーの RA フラグが YES に設定されている場合、書き換え操作に関係なくヘッダーの CD フラグが変更されます。

リライトに対する **MQTT** サポート

August 15, 2023

リライト機能は MQTT プロトコルをサポートしています。MQTT クライアント要求とサーバー応答のパラメータに基づいてアクションを実行するように、書き換えポリシーを設定できます。

MQTT の書き換えアクション

MQTT の rewrite アクションは、MQTT リクエストまたはレスポンスをサーバーまたはクライアントに送信する前に MQTT リクエストまたはレスポンスに加えられた変更を示します。

表現:

```
add rewrite action <name> <rewrite_type> <target> <rewrite_action>
```

MQTT の書き換えタイプ

使用するリライト式ルールのタイプに応じて、次の MQTT リライトタイプがサポートされます。

- replace_mqtt
- insert_before_mqtt
- insert_after_mqtt
- delete_mqtt
- insert_mqtt

MQTT の書き換えターゲット

次の例では、MQTT 書き換え機能がポリシー式を使用して、変更するリクエストの部分（ターゲット）と実行する変更（文字列式）を示します。

- replace_mqtt アクションタイプを使用して、接続パケット内のクライアント ID を書き換えます。

  ```
  add rewrite action rwact1 replace_mqtt MQTT.CONNECT.CLIENTID "\"
  xyz\""
  ```

- replace_mqtt アクションタイプを使用して、パブリッシュリクエスト内のトピックを書き換えます。

  ```
  add rewrite action rwact1 replace_mqtt MQTT.PUBLISH.TOPIC "\"
  testing/test123\""
  ```

- insert_mqtt アクションタイプを使用してプロパティを挿入するように書き直します。

  ```
  add rewrite action rwact1 insert_mqtt MQTT.NEW_PROPERTY("prop1",
  "test")
  ```

- delete_mqtt アクションタイプを使用してトピックを削除します。

  ```
  add rewrite action rwact2 delete_mqtt MQTT.SUBSCRIBE.TOPIC_FILTERS
  .TOPIC(1)
  ```


MQTT の書き換えアクション

MQTT の事前定義済み書き換えアクションは次のとおりです。

- MQTT.NEW_KEEPALIVE(interval)
- MQTT.NEW_PACKET_IDENTIFIER(packetID)
- MQTT.NEW_REASON_CODE(retCode)
- MQTT.NEW_PUBLISH(topic_name, payload)
- MQTT.NEW_CONNECT_USERNAME(username)
- MQTT.NEW_CONNECT_WILL_MESSAGE(will_topic, will_payload, will_Qos, will_retain)
- MQTT.NEW_TOPIC(topic, qos)
- MQTT.NEW_TOPIC(topic)
- MQTT.NEW_PROPERTY(key, value)

定義済みの書き換えアクションの例:

```
add rewrite action rwact1 replace_mqtt MQTT.CONNECT.KEEPALIVE MQTT.
NEW_KEEPALIVE(90)
```

ユーザー定義の書き換えアクションの例:

```
add rewrite action rwact1 replace_mqtt MQTT.CONNECT.USERNAME "\"user1
\""
```

MQTT の書き換えポリシー

MQTT の書き換えポリシーは、ルールとアクションで構成されます。ルールは書き換えが適用される MQTT トラフィックを決定し、アクションは NetScaler ADC アプライアンスが実行するアクションを決定します。

表現:

```
add rewrite policy <name> <rewrite_rule> <rewrite_action>
```

例:

```
add rewrite action insert_mqtt_username insert_mqtt MQTT.NEW_CONNECT_USERNAME
("user1")
```

```
add rewrite policy rewrite_mqtt_username "MQTT.COMMAND.EQ(CONNECT)&&
MQTT.CONNECT.USERNAME.LENGTH.EQUALS(0)insert_mqtt_username
```

MQTT のバインドポイント

書き換えポリシーは、グローバルにバインドすることも、特定の負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにバインドすることもできます。

グローバルバインドポイントは次のとおりです。

- MQTT_REQ_DEFAULT
- MQTT_REQ_OVERRIDE
- MQTT_RES_DEFAULT
- MQTT_RES_OVERRIDE

表現:

- bind rewrite global <policyName> <priority> [-type MQTT_REQ_OVERRIDE | MQTT_REQ_DEFAULT | MQTT_RES_OVERRIDE | MQTT_RES_DEFAULT]
- bind lb|cs vserver <virtualServerName> -policyName <policyName> -priority <positiveInteger> -type REQUEST|RESPONSE

例:

- bind rewrite global pol1 10 -type MQTT_REQ_DEFAULT
- add/bind lb vserver v1 -policyName pol1 -type reqUEST -priority 10

MQTT の書き換えポリシーを設定する

書き換えポリシーを構成するには、手順に従い、コマンドプロンプトでコマンドを入力します。

1. NetScaler ADC アプライアンスで書き換え機能を有効にします。

   ```
   enable ns feature REWRITE
   ```

2. 書き換えアクションを追加します。

   ```
   add rewrite action rwact1 replace_mqtt MQTT.CONNECT.KEEPALIVE MQTT.NEW_KEEPALIVE(10)
   ```

3. 書き換えポリシーを追加します。

   ```
   add rewrite policy pol1 MQTT.COMMAND.EQ(CONNECT)rwact1
   ```

4. MQTT 負荷分散仮想サーバーを構成します。

   ```
   add lb vserver v1 MQTT 1.1.1.1 1883
   ```

5. 書き換えポリシーをグローバルにバインドするか、特定の負荷分散仮想サーバーにバインドします。

   ```
   bind rewrite global pol1 10 -type MQTT_REQ_DEFAULT
   ```

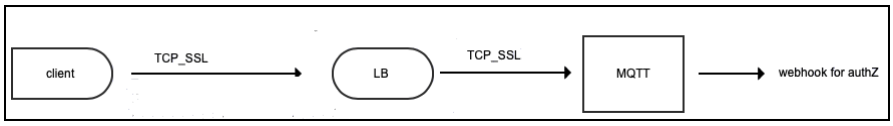
   ```
   add/bind lb vserver v1 -policyName pol1 -type REQUEST -priority 10
   ```


ユースケース **1: MQTT CONNECT** メッセージ内のユーザー名を証明書名に置き換えます

管理者は MQTT 書き換えポリシーを設定して、ユーザー名をクライアントの証明書名に置き換えることができます。

例を考えてみましょう。クライアント要求には、ユーザー名が「admin」というMQTT CONNECTメッセージが含まれています。このユーザー名は、クライアント証明書 (証明書名) から抽出されたシリアル番号 (16 桁) に置き換える必要があります。

次の図に、このワークフローを示します。



1. トランスポート制御プロトコル (TCP) 要求がロードバランサーに送信されます。
2. ロードバランサーでは、ユーザー名は証明書名に置き換えられます。
3. リクエストは MQTT ブローカーに転送されます。
4. この新しいユーザー名は、Webhook ペイロードによる認証に使用されます。

設定例:

```
add rewrite action mqtt_rw_unameact1 replace_mqtt MQTT.CONNECT.
USERNAME CLIENT.SSL.CLIENT_CERT.SERIALNUMBER

add rewrite policy mqtt_rw_uname_pol1 "MQTT.COMMAND.EQ(CONNECT)"
mqtt_rw_unameact1

bind cs vserver mqtt_frontend_cs -policyName mqtt_rw_uname_pol1 -
priority 10 -gotoPriorityExpression END -type REQUEST
```

ユースケース **2:** 新しい **TOPIC** へのサブスクリプションを提供する

管理者は新しい TOPIC へのサブスクリプションを提供できます。例を考えてみましょう。クライアント要求に TOPIC 1 へのサブスクリプションがあります。管理者は、新しい TOPIC 2 へのサブスクリプションを提供する書き換えポリシーを構成できます。サブスクリプションは前または後に挿入できます。

設定例:

- `add rewrite action act2 insert_before_mqtt MQTT.TOPIC_FILTERS.TOPIC(1)MQTT.NEW_TOPIC(topic2, 2)`
- `add rewrite policy policy2 “MQTT.COMMAND.EQ(SUBSCRIBE)&& MQTT.SUBSCRIBE. TOPIC_FILTERS.TOPIC.CONTAINS(\"test\")"act2`

文字列マップ

August 15, 2023

文字列マップを使用して、デフォルトのポリシー構文を使用するすべての NetScaler ADC 機能でパターンマッチングを実行できます。文字列マップは、キーと値のペアで構成される NetScaler ADC エンティティです。キーと値は、ASCII 形式または UTF-8 形式の文字列です。文字列比較では、2 つの新しい関数、`MAP_STRING(<string_map_name>)`および`IS_STRINGMAP_KEY(<string_map_name>)`を使用します。

文字列マップを使用するポリシー設定は、ポリシー式を使用して文字列照合を行うポリシー設定よりもパフォーマンスが優れており、多数のキーと値のペアで文字列照合を実行するために必要なポリシーの数が少なくて済みます。文字列マップも直感的で設定が簡単で、構成が小さくなります。

文字列マップの仕組み

文字列マップはパターンセットと構造が似ており（パターンセットはインデックス値の文字列へのマッピングを定義し、文字列マップは文字列から文字列へのマッピングを定義します）、文字列マップの構成コマンド（add、bind、unbind、remove、show などのコマンド）は構文的に設定に似ていますパターンセットのコマンド。また、パターンセット内のインデックス値の場合と同様に、文字列マップ内の各キーはマップ全体で一意でなければなりません。次の表に、url_string_map という文字列マップを示します。このマップには、URL がキーと値として含まれています。

キー	値
/url_1.html	`http://www.redirect_url_1.com/url_1.html`
/url_2.html	`http://www.redirect_url_2.com/url_2.html`
/url_3.html	`http://www.redirect_url_1.com/url_1.html`

表 1. 文字列マップ「url_string_map」

次の表に、文字列マップ内のキーとの文字列照合を可能にするために導入された 2 つの関数を示します。文字列のマッチングは、常にキーで実行されます。さらに、次の関数は、文字列マップ内のキーと、式の接頭辞によって返される完全な文字列との比較を実行します。説明の例は、前の例を示しています。

文字列マップ内のキーとの文字列マッチングを有効にするために導入された 2 つの関数の詳細については、文字列マップ関数表 pdf を参照してください。

文字列マップの設定

最初に文字列マップを作成し、それにキーと値のペアをバインドします。文字列マップは、コマンドラインインターフェイス（CLI）または設定ユーティリティから作成できます。

コマンドラインインターフェイスを使用して文字列マップを設定するには

コマンドプロンプトで、次の操作を行います。

1. 文字列マップを作成します。

```
add policy stringmap <name> -comment <string>
```

1. キーと値のペアを文字列マップにバインドします。

```
bind policy stringmap <name> <key> <value> [-comment <string>]
```

例**:**

```
1 bind policy stringmap url_string_map1 "/url_1.html" "http://www.
     redirect_url_1.com/url_1.html"
2 <!--NeedCopy-->
```

NetScaler GUI を使用して文字列マップを構成するには

[**AppExpert**] > [文字列マップ] に移動し、[追加] をクリックして、関連する詳細を指定します。

例**:** リダイレクトアクションを含むレスポンダーポリシー

次のユースケースには、リダイレクトアクションを含むレスポンダーポリシーが含まれます。以下の例では、最初の 4 つのコマンドが文字列マップ url_string_map を作成し、前の例で使用した 3 つのキーと値のペアをバインドします。マップを作成し、キーと値のペアをバインドしたら、クライアントを文字列マップ内の対応する URL または www.default.com にリダイレクトするレスポンダアクション（act_url_redirects）を作成します。また、要求された URL が url_string_map のいずれかのキーと一致するかどうかをチェックし、設定されたアクションを実行するレスポンダーポリシー（pol_url_redirects）も設定します。最後に、レスポンダーポリシーを、評価対象のクライアント要求を受信するコンテンツスイッチング仮想サーバーにバインドします。

```
add stringmap url_string_map

bind stringmap url_string_map /url_1.html http://www.redirect_url_1.
com/url_1.html

bind stringmap url_string_map /url_2.html http://www.redirect_url_2.
com/url_2.html

bind stringmap url_string_map /url_3.html http://www.redirect_url_1.
com/url_1.html
```


‘add responder action act_url_redirects redirect ‘HTTP.REQ.URL.MAP_STRING(“url_string_map”) ALT “www.default.com”’

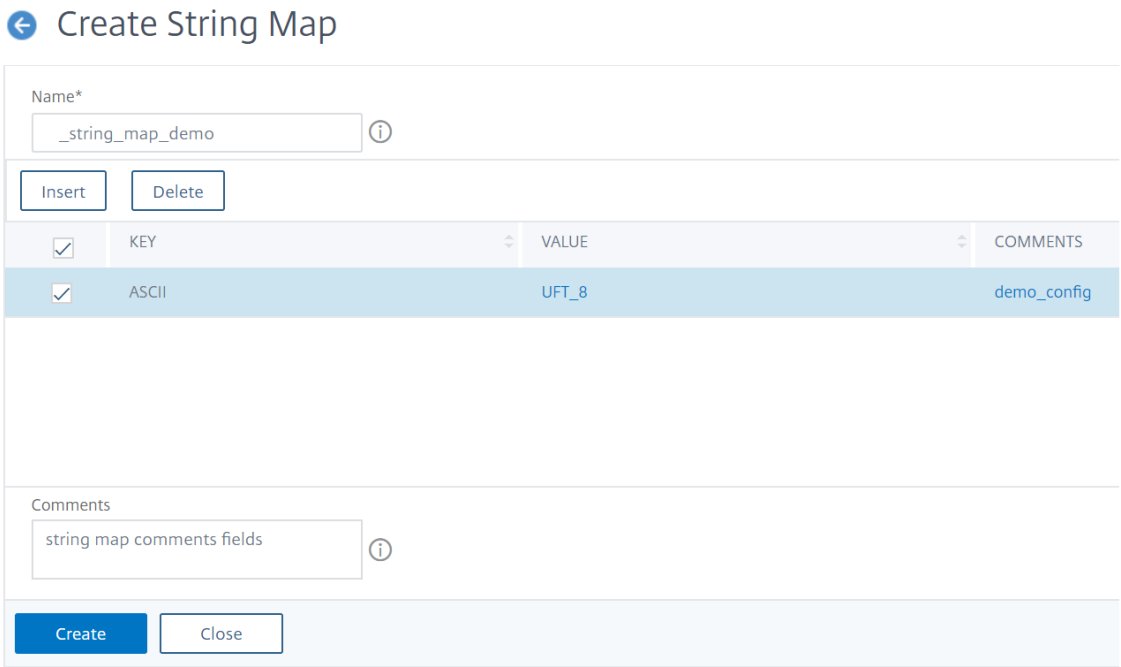
```
add responder policy pol_url_redirects TRUE act_url_redirects

bind cs vserver csw_redirect -policyname pol_url_redirects -priority
1 -type request
```

NetScaler GUI を使用して文字列マップを構成するには

ストリングマップを設定するには、以下の手順に従います。

1. ナビゲーションペインで、**AppExpert** を展開し、[文字列マップ] をクリックします。
2. 詳細ウィンドウで、[追加] をクリックします。
3. **[**文字列マップの作成] ページで、次のパラメータを設定します。
   - [名前]。文字列マップの名前。
   - キー値を設定します。文字列マップにバインドされた ASCII ベースのキー値エントリ
   - [コメント]。文字列マップにバインドされたキー値に関する簡単な説明。
4. **[**作成**]** して **[**閉じる**]** をクリックします。



URL セット

August 15, 2023

この機能により、100 万の URL をブラックリストに登録できます。このセクションには、以下のトピックが含まれています。

- Getting Started
- URL 評価における高度なポリシー表現の使用
- URL セットを設定する
- URL パターンセマンティクス
- ブラックリストに登録された URL カテゴリ

Getting Started

August 15, 2023

制限された Web サイトへのアクセスを防ぐために、NetScaler アプライアンスは専用の URL マッチングアルゴリズムを使用します。このアルゴリズムでは、ブロックされたエントリが最大 100 万 (1,000,000) の URL のリストを含むことができる URL セットを使用します。グローバル制限は 100 万エントリです。エントリ数が 100 万の URL セットを 1 つ追加することも、合計 100 万エントリを含む複数の URL セットを追加することもできます。

> 注:
>
> 多くの URL セットを使用することは避けてください。URL セットに使用可能なメモリに応じて、使用する URL セットの数を制限することをお勧めします。

各エントリには、URL カテゴリとカテゴリグループをインデックス付きパターンとして定義するメタデータを含めることができます。アプライアンスは、インターネット執行機関（政府の Web サイトを含む）またはインターネット組織が管理する機密性の高い URL セットを定期的にダウンロードすることもできます。URL セットが Web サイトからダウンロードされてアプライアンスにインポートされると、アプライアンスは URL セットを（これらの機関の要求に応じて）暗号化します。暗号化された URL セットは機密に保たれ、エントリは改ざんされません。

NetScaler アプライアンスは高度なポリシーを使用して、受信 URL をブロック、許可、リダイレクトする必要があるかどうかを判断します。これらのポリシーでは、高度な表現を使用して受信 URL をブラックリストに登録されたエントリと照合して評価します。エントリにはメタデータを含めることができます。メタデータがないエントリには、文字列の完全一致に基づいて URL を評価する式を使用できます。他の URL では、文字列が完全に一致するかどうかをチェックする式に加えて、URL のメタデータを評価する式を使用できます。

ISP/通信事業者向けの安全なインターネットアクセスポリシーのユースケース

URL セットを使用すると、ISP（ISP）または通信会社の顧客は、政府が義務付けている次のような安全なインターネットアクセスポリシーを実施できます。

1. 違法なインターネットサイト (児童虐待、薬物など) へのアクセスをブロックする
2. お子様向けセーフブラウジング

NetScaler アプライアンスを使用すると、インターネット執行機関や独立したインターネット組織が管理する URL セットを定期的にダウンロードできます。アプライアンスは定期的にリストをダウンロードし、安全に更新します。リストは機密の URL セットとして保存されるため、改ざんされたり、人間が読めたりすることはありません。定期的にダウンロードされる URL セットは、URL 評価用のブラックリストに登録されたセットとして機能します。

プライベート URL を設定していて、リストの内容は秘密にされ、ネットワーク管理者はリストに存在するブラックリストの URL を知らない場合。ポリシーが正しく設定され、正しいリストが参照されていることを確認するには、Canary URL を設定して URL セットに追加する必要があります。Canary URL を使用すると、管理者はアプライアンスを介してリクエストできます。プライベート URL セットを使用して、URL リクエストのたびにそのプライベート URL が検索されるようにします。

URL 評価のための高度なポリシー式

August 15, 2023

次の表は、URL セット内のエントリを含む受信 URL を評価するために使用できる式を示しています。

注:HTTP.REQ.URL は次のように使用されるように一般化されています <URL expression>

式	操作
<URL expression>URLSET_MATCHES_ANY	URL が URL セット内のいずれかのエントリと完全に一致する場合、TRUE と評価されます。
<URL expression>GET_URLSET_METADATA (<URLSET>)	GET_URLSET_METADATA () 式は、URL が URL セット内のいずれかのパターンと完全に一致する場合、関連するメタデータを返します。一致しなかった場合は、空の文字列が返されます。
<URL expression> .GET_URLSET_METADATA(<URLSET>).EQ(<METADATA>)	一致したメタデータが<METADATA>と等しい場合は TRUE と評価されます。

式	操作
<URL expression> .GET_URLSET_METADATA(<URLSET>).TYPECAST_LIST_T(‘,’).GET(0).EQ(<CATEGORY>)	一致したメタデータがカテゴリの先頭にある場合は TRUE と評価されます。このパターンは、メタデータ内の別々のフィールドをエンコードするために使用できますが、最初のフィールドのみと一致します。
HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL)	ホストパラメータと URL パラメータを結合します。このパラメータは、照合のため に<URL expression>として使用できます。

URL セットの構成

December 8, 2023

NetScaler プラットフォームで URL セットを構成し、URL を制限するには、次のタスクを実行します。

1. URL セットをインポートします (ダウンロードして暗号化)。NetScaler アプライアンスで URL セットをインポートすると、次のことが可能になることができます。

- URL ファイルをダウンロードします。
- アプライアンスにファイルを追加するには。
- ファイルを暗号化する。
  URL セットをシステムに追加するまで、ユーザーには表示されません。

セットは次の方法でダウンロードできます。

- リモートサーバーから URL セットを 1 回ダウンロードして、次のように指定します。http://myserver.com/file_with_urlset.csv
- ADC 内の/var/tmp/パスの下にファイルを追加し、以下の例のようにコマンドを使用します。

```
> shell cat /var/tmp/test_urlset.csv
example.com
google.com
> import policy urlset top10
k -url local:test_urlset.csv -delimiter "," -rowSeparator "n" -interval
    10 -privateSet -canaryUrl http://www.in.gr
Done

<!--NeedCopy-->
```

インポートされた URL セットは、データベース内の異なるカテゴリおよびカテゴリグループに分類されます。これは、URL セットファイルのメタデータにカテゴリが存在する場合にのみ有効です。

注: メタデータなしで URL パターンがある可能性があります。

ファイルをインポートしたら、ファイルのプロパティを更新、削除、または表示することができます。ファイルがアプライアンスにプッシュされた後、行を追加してエントリを変更できます。

インポートされたセットは、暗号化されたファイル形式で NetScaler ディレクトリに保存されます。インポートされたリストには、数百万の URL エントリが含まれています。次の’インポートされたリストには、最大 100 万の URL エントリを含めることができます。それ以外の場合、アプライアンスは値が制限を超えていることを示すエラーメッセージを返します。インポートされた URL セットにメタデータを含むブラックリストに登録されているエントリがある場合、インポート時にアプライアンスによってメタデータが検出されます。

URL セットをインポートしてアプライアンスに追加すると、着信 URL 評価時に正しい URL セットを識別するために、高度なポリシーで URL セットを使用できます。`HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URLSET_MATCHES_ANY(<URL set name>)`

1. NetScaler アプライアンスの URL セットの更新。ファイルをアプライアンスにプッシュしたら、この間隔でコマンドラインインターフェイスを使用して URL ファイルを手動で更新できます。
2. URL セットのエクスポート。URL セットのバックアップが必要な場合は、URL パターンのリストをエクスポートし、そのコピーを宛先 URL に保存できます。エクスポートする前に、URL セットがプライベートとしてマークされているかどうかを確認します。がプライベートとマークされている場合、URL セットはエクスポートできません。エクスポート機能は、プライベートセットでは機能しません。したがって、新しい URL セット`myurl`は、プライベートセットが定義されずにインポートされ、以下のようにローカルパス内の別のファイルにエクスポートされます。

```
> shell touch /var/tmp/test_urlset_export.csv
Done
> shell cat /var/tmp/test_urlset_export.csv
Done
> shell cat /var/tmp/test_urlset.csv
example.com
google.com
Done
> export urlset myurl -url local:test_urlset_export.csv

> import urlset myurl -url local:test_urlset.csv
Done
(a non-private urlset is imported)

<!--NeedCopy-->
```

1. URL セットの削除。ブラックリストに登録されたエントリの URL セットを削除する場合は、remove コマンドを使用して、NetScaler アプライアンスから URL セットを削除できます。
2. URL セットの表示。show コマンドを使用して、URL セットのプロパティを表示できます。

> 注: クエリ部分を含む URL は、インポート中に削除されます。

例:


```
show urlset
Name: top100 PatternCount: 100 Delimiter:  RowSeparator: Interval: 0
Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、メタを含む **URL** セットをインポートする

コマンドプロンプトで、次のように入力します:

```
import urlset <name> [-overwrite] [ - delimiter <character>] [-
   rowSeparator <character>] [-url] <url> [-interval <seconds>] [-
   privateSet] [-canaryUrl <URL>]
<!--NeedCopy-->
```

各項目の意味は次のとおりです。

区切り文字は、デフォルト値が 44 に設定された CSV ファイルレコードです。

RowSeparator は、デフォルト値は 10 に設定された CSV ファイルの行区切り文字です。

間隔は、URL セットの更新が行われる最も近い 15 分に四捨五入された時間間隔を秒単位で指定します。

CanaryUrl は、URL セットの内容が機密保持されるときのテストに使用される URL です。

例

```
import policy urlset -url local:test_urlset.csv -delimiter ","-
rowSeparator "n"-interval 10 -privateSet -canaryUrl http://www.in.
gr
```

インポートされた **URL** セットに対して明示的なサブドメイン一致を実行する

インポートされた URL セットに対して明示的なサブドメイン一致を実行できるようになりました。新しいパラメーター「SubdomainExactMatch」が「インポートポリシー URLSet」コマンドに追加されます。パラメーターを有効にすると、URL フィルタリングアルゴリズムは明示的なサブドメイン一致を実行します。たとえば、着信 URL が「news.example.com」で、URL セット内のエントリが「example.com」の場合、アルゴリズムは URL と一致しません。

コマンドプロンプトで入力します:

```
import policy urlset <name> [-overwrite] [-delimiter <character>][-
rowSeparator <character>] -url <URL> [-interval <secs>] [-privateSet
][-subdomainExactMatch] [-canaryUrl <URL>]
```

例:

```
import policy urlset forth_urlset -url local:test_urlset.csv -interval
 3600 -subdomainExactMatch
```


コマンドラインインターフェイスを使用して **URL** セットを表示するには

コマンドプロンプトで入力します:

```
show urlset <name>
```

例:

コマンドプロンプトで入力します:

```
1          URLset     Count
2          ------     -----
3  1)      top1k      100
4   Done
5
6  > show urlset top1k
7          Count      Delimiter  Interval   RowSeparator
8          -----      ---------  --------   ------------
9          100              ,        0          0x0a
10  Done
11 >
12
13 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用してインポートされた **URL** セットを表示するには

コマンドプロンプトで入力します:

```
show urlset -imported
```

例:

コマンドプロンプトで入力します:

```
1          URLset
2          ------
3  1)      top1k
4   Done
5  <!--NeedCopy-->
```

URL セットを表示するには コマンドラインインターフェイスを使用する

コマンドプロンプトで入力します:

```
show urlset <name>
```

コマンドラインインターフェイスを使用して **URL** セットをエクスポートするには

コマンドプロンプトで入力します:


```
export urlset <name> <url>
```

コマンドラインインターフェイスを使用して **URL** セットを追加するには

コマンドプロンプトで入力します:

```
add urlset <urlset_name>
```

コマンドラインインターフェイスを使用して **URL** セットを更新するには

コマンドプロンプトで入力します:

```
update urlset <name>
```

コマンドラインインターフェイスを使用して **URL set** コマンドを削除するには

コマンドプロンプトで入力します:

```
remove urlset <name>
```

例:

> 注:
>
> URLSet をインポートまたはエクスポートする前に、test_urlset_export.csvおよびtest_urlset.csvファイルが作成され、/var/tmpディレクトリの下で使用可能であることを確認する必要があります。

```
1  import policy urlset -url local:test_urlset.csv -delimiter "," -
       rowSeparator "n" -interval 10 -privateSet -overwrite  - canaryUrl
       http://www.in.gr
2
3  add policy urlset top10k
4
5  update policy urlset top10k
6
7  sh policy urlset
8
9  sh policy urlset top10k
10
11 export policy urlset urlset1 -url local:test_urlset_export.csv
12
13 import policy urlset top10k -url local:test_urlset.csv  - privateSet
14
15 add policy urlset top10k
16
17 update policy urlset top10k
18
```



```
show policy urlset top10k
<!--NeedCopy-->
```

インポートされた **URL** セットを表示する

追加された URL セットに加えて、読み込んだ URL セットも表示できるようになりました。これを行うには、「show url set」コマンドに新しいパラメータ「imported」が追加されます。このオプションを有効にすると、アプライアンスはインポートされたすべての URL セットを表示し、インポートされた URL セットが追加された URL セットと区別されます。

コマンドプロンプトで入力します:

```
show policy urlset [<name>] [-imported]
```

例:

```
show policy urlset -imported
```

GUI を使用して **URL** セットをインポートするには

AppExpert > **URL** セットに移動し、インポートをクリックして URL セットをダウンロードします。

GUI を使用して **URL** セットを追加するには

AppExpert > **URL** セットに移動し、[追加] をクリックして、ダウンロードした URL セットの URL セットファイルを作成します。

GUI を使用して **URL** セットを編集するには

AppExpert > **URL** セットに移動し、URL セットを選択し、編集をクリックして変更します。

GUI を使用して **URL** セットを更新するには

AppExpert > **URL** セットに移動し、URL セットを選択し、**URL** セットの更新をクリックして、ファイルに加えられた最新の変更で URL セットを更新します。

GUI を使用して **URL** セットをエクスポートするには

AppExpert > URL セットに移動し、URL セットを選択し、[**URL** セットのエクスポート] をクリックして、セット内の URL パターンを宛先 URL にエクスポートし、その場所に保存します。

URL パターンのセマンティクス

August 15, 2023

次の表に、フィルタリングするページのリストを指定するために使用する URL パターンを示します。たとえば、URL パターン`http://www.example.com/bar`は 1 つのページ`http://www.example.com/bar`に一致します。URL が www.example.com/bar で始まるすべてのページをカバーするには、末尾に「*」を明示的に追加する必要があります。

詳細については、「URL パターンメタデータマッピングテーブル 」を参照してください。

URL のカテゴリ

August 15, 2023

以下は、ブラックリストに登録されているカテゴリのリストです。

S.no	ブラックリストに載っているカテゴリー
1	違法行為
2	違法薬物
3	薬物
4	マリファナ
5	テロリズム/過激派
6	武器
7	誹謗/中傷
8	暴力/自殺
9	主義主張全般
10	アダルト/ポルノ
11	ヌード
12	性的サービス
13	アダルト検索/リンク
14	ハッキング/クラッキング
15	マルウェア

S.no	ブラックリストに載っているカテゴリー
16	リモートプロキシ
17	検索エンジンのキャッシュ
18	翻訳
19	出会い系
20	結婚式/結婚生活
21	相場
22	オンライン取引
23	保険
24	金融商品
25	ギャンブル全般
26	宝くじ
27	オンラインゲーム
28	ゲーム
29	オークション
30	ショッピング/小売
31	不動産
32	IT オンラインショッピング
33	Web 上でのチャット
34	インスタントメッセージ
35	Web ベースのメール
36	メールサービス加入
37	掲示板
38	IT 掲示板
39	個人の Web ページ/ブログ
40	ダウンロード
41	プログラムのダウンロード
42	ストレージサービス
43	ストリーミングメディア
44	雇用

S.no	ブラックリストに載っているカテゴリー
45	キャリアアップ
46	副業
47	猟奇描写
48	スペシャルイベント
49	人気トピック
50	アダルト雑誌/ニュース
51	喫煙
52	飲酒
53	アルコール製品
54	フェチ
55	性的表現（テキスト）
56	コスプレ/仮装
57	オカルト
58	家庭および家族
59	プロスポーツ
60	スポーツ全般
61	ライフイベント
62	旅行/観光
63	公的機関（観光）
64	公共交通
65	宿泊施設
66	ミュージック
67	ホロスコープ/占星術/運勢判断
68	芸能人/著名人
69	飲食/グルメ
70	娯楽/施設/アクティビティ
71	既成宗教
72	宗教
73	政治

S.no	ブラックリストに載っているカテゴリー
74	広告/バナー
75	懸賞/プレゼント
76	スパム
77	ニュース
78	自動車
79	ビジネスおよび商業
80	コンピューターおよびインターネット
81	教育
82	自治体
83	状況
84	インターネット電話
85	軍事
86	ピアツーピア/トレント
87	レジャーおよび趣味
88	リファレンス
89	検索エンジンおよびポータル
90	性教育
91	SMS および携帯電話サービス
92	モバイルアプリおよびパブリッシャー
93	スパイウェア
94	コンテンツ配信ネットワークおよびインフラストラクチャ
95	子供向けサイト
96	水着および下着
97	芸術および文化イベント
98	ホスティングサイト
99	慈善事業および非営利団体
100	写真検索および写真共有サイト
101	着信音

S.no	ブラックリストに載っているカテゴリー
102	ファッションおよび美容
103	モバイルアプリストア
104	パークドメイン
105	絵文字
106	移動体通信事業者
107	ボットネット
108	感染サイト
109	フィッシングサイト
110	キーロガー
111	モバイルマルウェア
112	コンテンツなし
113	農業
114	アーキテクチャ
115	各種協会/業界団体/労働組合
116	書籍・電子書籍
117	ボットネットから命令元への通信
118	DDNS
119	サポートされていない URL
120	法律
121	地域コミュニティ
122	その他
123	オンラインマガジン
124	ペット/獣医
125	著作権や商標権の侵害
126	プライベート IP アドレス
127	リサイクル/環境
128	科学
129	社会および文化
130	輸送サービスと貨物

S.no	ブラックリストに載っているカテゴリー
131	写真および映画
132	博物館および歴史
133	E ラーニング
134	ソーシャルネットワーク全般
135	Facebook
136	Facebook：投稿
137	Facebook：コメント
138	Facebook：友達
139	Facebook：写真のアップロード
140	Facebook：イベント
141	Facebook：アプリ
142	Facebook：チャット
143	Facebook：質問
144	Facebook：動画のアップロード
145	Facebook：グループ
146	Facebook：ゲーム
147	LinkedIn
148	LinkedIn：アップデート
149	LinkedIn：メール
150	LinkedIn：つながり
151	LinkedIn：求人情報
152	Twitter
153	Twitter：投稿
154	Twitter：メール
155	Twitter：フォロー
156	YouTube
157	YouTube：コメント
158	YouTube：動画のアップロード
159	YouTube：共有

S.no	ブラックリストに載っているカテゴリー
160	Instagram
161	Instagram：アップロード
162	Instagram：コメント
163	Instagram：ダイレクトメッセージ
164	Tumblr
165	Tumblr：掲示板
166	Tumblr：コメント
167	Tumblr：写真または動画のアップロード
168	Google+
169	Google+：投稿
170	Google+：コメント
171	Google+：写真のアップロード
172	Google+：動画のアップロード
173	Google+：動画チャット
174	Pinterest
175	Pinterest：ピン
176	Vine：アップロード
177	Vine：コメント
178	Vine：メッセージ
179	Ask.fm
180	Ask.fm：質問
181	Ask.fm：回答
182	Yik Yak
183	Yik Yak：投稿
184	Yik Yak：コメント
185	WordPress
186	WordPress：投稿
187	WordPress：アップロード

AppFlow

February 15, 2024

NetScaler アプライアンスは、データセンター内のすべてのアプリケーショントラフィックを一元的に制御します。これは、アプリケーションパフォーマンスの監視、分析、およびビジネスインテリジェンスアプリケーションにとって有効なフローとユーザーセッションレベルの情報を収集します。また、Web ページのパフォーマンスデータとデータベース情報も収集します。AppFlow は、RFC 5101 で定義されているオープンなインターネット技術タスクフォース（IETF）標準であるインターネットプロトコルフロー情報エクスポート（IPFIX）形式を使用して情報を送信します。IPFIX（Cisco 社製 NetFlow の標準化バージョン）は、ネットワークフロー情報を監視するために幅広く使用されています。AppFlow は、アプリケーションレベルの情報、Web ページのパフォーマンスデータ、およびデータベース情報を表す新しい情報エレメントを定義します。

トランスポートプロトコルとして UDP を使用して、AppFlow は フローレコードと呼ばれる収集されたデータを 1 つまたは複数の IPv4 コレクターに送信します。コレクターはフローレコードを集約し、リアルタイムレポートまたは履歴レポートを生成します。

AppFlow は、HTTP、SSL、TCP、SSL_TCP フロー、および HDX Insight フローのトランザクションレベルで可視性を提供します。監視対象のフロータイプのサンプリングとフィルタリングを行うことが可能です。

> 注:
>
> HDX Insight の詳細については、HDX Insightを参照してください。

AppFlow は、アクションとポリシーを使用して、選択したフローのレコードを特定のコレクターセットに送信します。AppFlow アクションは、AppFlow レコードを受け取るコレクタのセットを指定します。高度な式に基づくポリシーは、関連付けられた AppFlow アクションで指定されたコレクタにフローレコードが送信されるフローを選択するように設定できます。

フローのタイプを制限するために、仮想サーバーの AppFlow を有効にできます。AppFlow では、仮想サーバーの統計情報も提供しています。

また、AppFlow を特定のサービス向けに有効化してアプリケーションサーバーを表現し、そのアプリケーションサーバーへのトラフィックを監視することもできます。

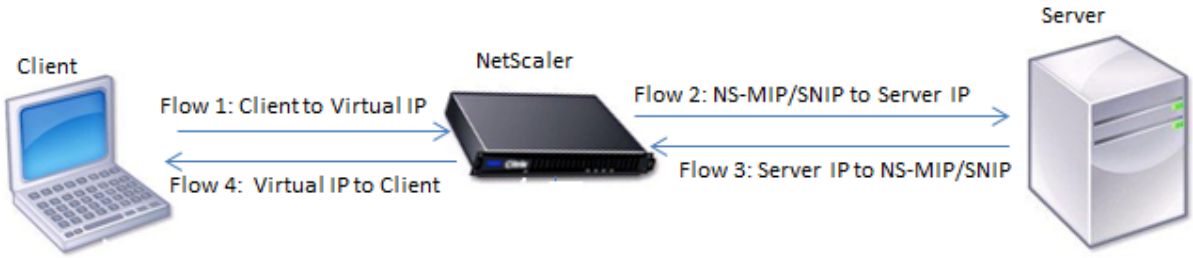
注：この機能は、NetScaler nCore ビルドでのみサポートされています。

AppFlow のしくみ

最も一般的な展開シナリオでは、受信トラフィックは NetScaler ADC アプライアンスの仮想 IP アドレス（VIP）に流れ、サーバーにロードバランスされます。送信トラフィックは、サーバーから NetScaler ADC 上のマッピングまたはサブネット IP アドレス、および VIP からクライアントに流れます。フローは、次の 5 つのタプル（sourceIP、sourcePort、destIP、destPort、およびプロトコル）によって識別される IP パケットの単方向コレクションです。

次の図は、AppFlow 機能の仕組みを示しています。

図 1: NetScaler フローシーケンス



図に示すように、トランザクションの各レッグのネットワークフロー識別子は、トラフィックの方向によって異なります。

フローレコードを形成するさまざまなフローは次のとおりです。

フロー 1: <Client-IP, Client-Port, VIP-IP, VIP-port, Protocol>

フロー 2: <NS-MIP/SNIP, NS-port, Server-IP, Server-Port, Protocol>

フロー 3: <Server-IP, Server-Port, NS-MIP/SNIP, NS-Port, Protocol>

フロー 4: <VIP-IP, VIP-port, Client-IP, Client-Port, Protocol>

コレクターがトランザクション内の 4 つのフローすべてをリンクできるように、AppFlow は各フローにカスタム transactionID 要素を追加します。HTTP などのアプリケーションレベルのコンテンツスイッチングでは、1 つのクライアント TCP 接続を、要求ごとに異なるバックエンド TCP 接続に負荷分散できます。AppFlow は、トランザクションごとに一連のレコードを提供します。

フローレコード

AppFlow レコードには、フローの開始と終了のタイムスタンプ、パケットカウント、バイトカウントなどの標準的な NetFlow または IPFIX 情報が含まれます。AppFlow レコードには、アプリケーションレベルの情報（HTTP URL、HTTP リクエストメソッド、レスポンスステータスコード、サーバーレスポンス時間、レイテンシーなど）も含まれます。Web ページのパフォーマンスデータ (ページの読み込み時間、ページのレンダリング時間、ページ滞在時間など)。データベース情報 (データベースプロトコル、データベース応答ステータス、データベース応答サイズなど)。IPFIX フローレコードは、フローレコードを送信する前に送信する必要があるテンプレートに基づいています。

テンプレート

AppFlow は、フローのタイプごとに 1 つずつ、一連のテンプレートを定義します。各テンプレートには、標準の情報エレメント (IE) とエンタープライズ固有の情報エレメント (EIE) のセットが含まれています。IPFIX テンプレートは、フローレコード内の情報要素（Internet Explorer）の順序とサイズを定義します。RFC 5101 で説明されているように、テンプレートは定期的にコレクタに送信されます。

テンプレートには、次の EIE を含めることができます。

- トランザクション ID

  アプリケーションレベルのトランザクションを識別する符号なし 32 ビットの数値。HTTP の場合、リクエストとレスポンスのペアに対応します。この要求と応答のペアに対応するすべてのフローレコードは、同じトランザクション ID を持ちます。最も一般的なケースでは、このトランザクションに対応するuniflowレコードが 4 つあります。NetScaler ADC がそれ自体で（統合キャッシュまたはセキュリティポリシーによって提供される）応答を生成する場合、このトランザクションのフローレコードは 2 つしかない可能性があります。

- ConnectionID

  レイヤ 4 接続（TCP または UDP）を識別する符号なし 32 ビットの数値。NetScaler ADC フローは双方向であり、フローの各方向に対して 2 つの別々のフローレコードがあります。この情報エレメントは、2 つのフローをリンクするために使用できます。

  NetScaler ADC の場合、connectionId は接続の進行状況を追跡するための接続データ構造の識別子です。たとえば、HTTP トランザクションでは、特定の connectionId に、その接続で行われた複数のリクエストに対応する複数の transactionID 要素がある場合があります。

- tcprTT

  TCP 接続で測定されたラウンドトリップ時間（ミリ秒単位）。これは、ネットワーク上のクライアントまたはサーバーのレイテンシーを決定するメトリックとして使用できます。

- httpRequestMethod

  トランザクションで使用される HTTP メソッドを示す 8 ビットの数値。番号からメソッドへのマッピングを含むオプションテンプレートが、テンプレートとともに送信されます。

- httpRequestSize

  リクエストのペイロードサイズを示す符号なし 32 ビットの数値。

- httpRequestURL

  クライアントによって要求された HTTP URL。

- HttpUserAgent

  Web サーバーへの受信要求のソース。

- httpResponseStatus

  応答ステータスコードを示す符号なし 32 ビットの数値。

- httpResponseSize

  応答サイズを示す符号なし 32 ビットの数値。

- HttpResponseTimeToFirstByte

  応答の最初のバイトを受信するのにかかった時間を示す符号なし 32 ビットの数値。

- httpResponseTimeToLastByte

  応答の最後のバイトを受信するのにかかった時間を示す符号なし 32 ビットの数値。

- flowFlags

  異なるフロー条件を示すために使用される符号なし 64 ビットフラグ。

Web ページのパフォーマンスデータの **EIE**

- clientInteractionStartTime

  ブラウザがレスポンスの最初のバイトを受信して、画像、スクリプト、スタイルシートなどのページのオブジェクトをロードする時刻。

- clientInteractionEndTime

  画像、スクリプト、スタイルシートなど、ページのすべてのオブジェクトをロードするために、ブラウザが応答の最後のバイトを受信した時刻。

- clientRenderStartTime

  ブラウザがページのレンダリングを開始する時刻。

- clientRenderEndTime

  ブラウザが埋め込みオブジェクトを含むページ全体のレンダリングを終了した時刻。

データベース情報の **EIE**

- dbProtocolName

  データベースプロトコルを示す符号なし 8 ビットの数値。有効な値は、MS SQL の場合は 1、MySQL の場合は 2 です。

- dbreqType

  トランザクションで使用されるデータベース要求メソッドを示す符号なし 8 ビットの数値。MS SQL の場合、有効な値は 1 がクエリ、2 がトランザクション、3 が RPC です。MySQL の有効な値については、MySQL のドキュメントを参照してください。

- dbreqString

  ヘッダーのないデータベース要求文字列を示します。

- dbrespStatus

  Web サーバーから受信したデータベース応答のステータスを示す符号なし 64 ビットの数値。

- dbResplength

  応答サイズを示す符号なし 64 ビットの数値。

- dbrespStatString

  Web サーバーから受信した応答ステータス文字列。

AppFlow 機能の構成

August 15, 2023

AppFlow は、他のほとんどのポリシーベースの機能と同じ方法で構成できます。まず、AppFlow 機能を有効にします。次に、フローレコードの送信先となるコレクタを指定します。その後、構成済みコレクターのセットであるアクションを定義します。次に、1 つ以上のポリシーを構成し、アクションを各ポリシーに関連付けます。このポリシーは、NetScaler ADC アプライアンスに、フローレコードが関連付けられたアクションに送信される要求を選択するように指示します。最後に、各ポリシーをグローバルに、または特定の仮想サーバーにバインドして、ポリシーを有効にします。

さらに、AppFlow パラメーターを設定して、テンプレートの更新間隔を指定し、httpURL、HttpCookie、および HttpReferer 情報のエクスポートを有効にすることができます。各コレクターで、エクスポーターのアドレスとして NetScaler ADC IP アドレスを指定する必要があります。

> 注
>
> NetScaler ADC をコレクターのエクスポーターとして構成する方法については、特定のコレクターのドキュメントを参照してください。

構成ユーティリティは、ユーザーがポリシーとアクションを定義するのに役立つツールを提供します。NetScaler ADC アプライアンスが特定のフローのレコードを一連のコレクタにエクスポートする方法を正確に決定します（アクション）。コマンドラインインターフェイスは、コマンドラインを好む経験豊富なユーザー向けに、対応する CLI ベースのコマンドセットを提供します。

AppFlow の有効化

AppFlow 機能を使用するには、まず AppFlow 機能を有効にする必要があります。

> 注
>
> AppFlow は、nCore NetScaler ADC アプライアンスでのみ有効にできます。

コマンドラインインターフェイスを使用して **AppFlow** 機能を有効にする

コマンドプロンプトで、次のコマンドのいずれかを入力します。


```
enable ns feature AppFlow

<!--NeedCopy-->
```

構成ユーティリティを使用して **AppFlow** 機能を有効にする

[システム] **>** [設定] に移動し、[高度な機能の構成] をクリックして、[**AppFlow**] オプションを選択します。

コレクターを指定する

コレクターは、NetScaler アプライアンスによって生成された AppFlow レコードを受信します。AppFlow レコードを送信するには、少なくとも 1 つのコレクターを指定する必要があります。デフォルトでは、コレクターは UDP ポート 4739 で IPFIX メッセージをリスンします。コレクターを構成するときに、デフォルトのポートを変更できます。同様に、デフォルトでは、NSIP は AppFlow トラフィックのソース IP として使用されます。コレクタを設定するときに、このデフォルトの送信元 IP を SNIP アドレスに変更できます。未使用のコレクターを削除することもできます。

コマンドラインインターフェイスを使用してコレクタを指定する

> 重要
>
> NetScaler ADC リリース 12.1 ビルド 55.13 以降、使用するコレクターのタイプを指定できます。`add appflow collector`コマンドに新しいパラメータ「Transport」が導入されました。デフォルトでは、コレクターは IPFIX メッセージをリッスンします。「Transport」パラメータを使用して、コレクタのタイプを`logstream`または`ipfix`またはリセットのいずれかに変更できます。構成の詳細については、例を参照してください。

コマンドプロンプトで次のコマンドを入力してコレクタを追加し、構成を確認します。

```
-  add appflow collector <name> -IPAddress <ipaddress> -port <
    port_number> -netprofile <netprofile_name> -Transport <Transport>

-  show appflow collector <name>

<!--NeedCopy-->
```

例

```
add appflow collector col1 -IPaddress 10.102.29.251 -port 8000 -
    netprofile n2 -Transport ipfix

<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して複数のコレクターを指定する

コマンドプロンプトで次のコマンドを入力して、同じデータを追加して複数のコレクタに送信します。

```
add appflow collector <collector1> -IPAddress <IP>

add appflow collector <collector2> -IPAddress <IP>

add appflow action <action> -collectors <collector1> <collector2>

add appflow policy <policy> true <action>

bind lbvserver <lbvserver> -policy <policy> -priority <priority>
<!--NeedCopy-->
```

構成ユーティリティを使用して **1** つ以上のコレクタを指定します

[システム] **>** **[AppFlow] >** [コレクター] に移動し、AppFlow コレクターを作成します。

AppFlow アクションを構成する

AppFlow アクションはセットコレクターであり、関連付けられた AppFlow ポリシーが一致した場合のフローレコードの送信先です。

コマンドラインインターフェイスを使用して **AppFlow** アクションを構成する

コマンドプロンプトで、次のコマンドを入力して AppFlow アクションを構成し、構成を確認します。

```
add appflow action <name> --collectors <string> ... [-
    clientSideMeasurements (Enabled|Disabled) ] [-comment <string>]

show appflow action

<!--NeedCopy-->
```

例

```
add appflow action apfl-act-collector-1-and-3 -collectors collector-1
    collecter-3

<!--NeedCopy-->
```

構成ユーティリティを使用して **AppFlow** アクションを構成する

[システム] **>** **[AppFlow] >** [アクション] に移動し、AppFlow アクションを作成します。

AppFlow ポリシーを構成する

AppFlow アクションを構成した後、AppFlow ポリシーを構成する必要があります。AppFlow ポリシーは、1 つ以上の式で構成される規則に基づいています。

> 注
>
> AppFlow ポリシーを作成および管理するために、構成ユーティリティは、コマンドラインインターフェイスでは利用できない支援を提供します。

コマンドラインインターフェイスを使用して AppFlow ポリシーを構成する

コマンドプロンプトで次のコマンドを入力し、AppFlow ポリシーを作成して構成を確認します:

```
add appflow policy <name> <rule> <action>

show appflow policy <name>

<!--NeedCopy-->
```

例:

```
add appflow policy apfl-pol-tcp-dsprt client.TCP.DSTPORT.EQ(22) apfl-
    act-collector-1-and-3

<!--NeedCopy-->
```

構成ユーティリティを使用して AppFlow ポリシーを構成する

[システム] > [AppFlow] > [ポリシー] に移動し、AppFlow ポリシーを作成します。

[数式の追加**]** ダイアログボックスを使用して式を追加します

1. [式の追加] ダイアログボックスの最初のリストボックスで、式の最初の用語を選択します。

   - HTTP
     HTTP プロトコル。HTTP プロトコルに関連するリクエストのいくつかの側面を調べる場合は、このオプションを選択します。
   - SSL

     ```
     保護された Web サイト。リクエストの受信者に関連するリクエストのいくつかの側面を調べる場合は、このオプションを選択します。 -
     CLIENT

     ```



```
4 The computer that sent the request. Choose the option if you want
      to examine some aspect of the sender of the request. 選択すると、右端のリストボックスに、式の次の部分に適した用語がリストされます。
```

2. 2 番目のリストボックスで、式の 2 番目の用語を選択します。選択肢は、前のステップで行った選択によって異なり、コンテキストに適切です。2 番目の選択を行った後、[式の構築] ウィンドウの下のヘルプウィンドウ (空白) に、選択した用語の目的と使用法を説明するヘルプが表示されます。
3. 式が終了するまで、前のリストボックスの右側に表示されるリストボックスから用語を選択するか、値の入力を求めるテキストボックスに文字列または数値を入力します。

AppFlow ポリシーをバインドする

ポリシーを有効にするには、ポリシーをグローバルにバインドして NetScaler ADC を通過するすべてのトラフィックに適用するか、特定の仮想サーバーにバインドして、その仮想サーバーに関連するトラフィックにのみポリシーを適用する必要があります。

ポリシーをバインドするときは、そのポリシーにプライオリティを割り当てます。プライオリティによって、定義したポリシーが評価される順序が決まります。優先度は、任意の正の整数に設定できます。

NetScaler ADC オペレーティングシステムでは、ポリシーの優先順位は逆の順序で機能します。数値が大きいほど優先度は低くなります。たとえば、優先順位が 10、100、1000 の 3 つのポリシーがある場合、優先度 10 が割り当てられたポリシーが最初に実行されます。その後、ポリシーに優先度 100 が割り当てられ、最後にポリシーに 1000 の順序が割り当てられました。

他のポリシーを任意の順序で追加する余地を十分に残しておき、希望する順序で評価するように設定できます。グローバルにバインドするときに、各ポリシー間に 50 または 100 の間隔で優先順位を設定することで実現できます。これにより、既存のポリシーの優先度を変更しなくても、いつでもポリシーを追加できます。

コマンドラインインターフェイスを使用して AppFlow ポリシーをグローバルにバインドする

コマンドプロンプトで次のコマンドを入力して、AppFlow ポリシーをグローバルにバインドし、構成を確認します。

```
bind appflow global <policyName> <priority> [<gotoPriorityExpression [-
    type <type>] [-invoke (<labelType> <labelName>)]

show appflow global

<!--NeedCopy-->
```

例

```
bind appflow global af_policy_lb1_10.102.71.190 1 NEXT -type
    REQ_OVERRIDE -invoke vserver google
```



```
2
3 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **AppFlow** ポリシーを特定の仮想サーバーにバインドする

コマンドプロンプトで次のコマンドを入力して、AppFlow ポリシーを特定の仮想サーバーにバインドし、構成を確認します。

```
1 bind lb vserver <name> -policyname <policy_name> -priority <priority>
2
3 <!--NeedCopy-->
```

例

```
1 bind lb vserver google -policyname af_policy_google_10.102.19.179 -
     priority 251
2
3 <!--NeedCopy-->
```

構成ユーティリティを使用して **AppFlow** ポリシーをグローバルにバインドする

[システム] **>** **[AppFlow]** に移動し、[**AppFlow** ポリシーマネージャー] をクリックし、関連するバインドポイント (デフォルトグローバル) と接続タイプを選択して、AppFlow ポリシーをバインドします。

構成ユーティリティを使用して **AppFlow** ポリシーを特定の仮想サーバーにバインドする

[トラフィック管理] **>** **[負荷分散]** **>** **[仮想サーバー]** に移動し、仮想サーバーを選択して [ポリシー] をクリックし、AppFlow ポリシーをバインドします。

仮想サーバーで **AppFlow** を有効にする

特定の仮想サーバーを経由するトラフィックのみを監視する場合は、その仮想サーバー専用に AppFlow を有効にします。AppFlow は、負荷分散、コンテンツスイッチング、キャッシュリダイレクト、SSL VPN、GSLB、および認証仮想サーバーに対して有効化できます。

コマンドラインインターフェイスを使用して仮想サーバーの **AppFlow** を有効にする

コマンドプロンプトで入力します。

```
1 set cs vserver <name> <protocol> <IPAddress> <port> -appflowLog ENABLED
2
3 <!--NeedCopy-->
```


例

```
set cs vserver Vserver-CS-1 HTTP 10.102.29.161 80 -appflowLog ENABLED

<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーの **AppFlow** を有効にする

[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを選択して [AppFlow ロギング] オプションを有効にします。

サービスに対して **AppFlow** を有効にする

負荷分散仮想サーバーにバインドされるサービスに対して AppFlow を有効にできます。

コマンドラインインターフェイスを使用してサービスの **AppFlow** を有効にする

コマンドプロンプトで入力します。

```
set service <name> -appflowLog ENABLED

<!--NeedCopy-->
```

例

```
set service ser -appflowLog ENABLED

<!--NeedCopy-->
```

構成ユーティリティを使用してサービスの **AppFlow** を有効にする

[トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、サービスを選択して [AppFlow ログ] オプションを有効にします。

AppFlow パラメーターを設定する

AppFlow パラメーターを設定して、コレクターへのデータのエクスポートをカスタマイズできます。

コマンドラインインターフェイスを使用して **AppFlow** パラメーターを設定する

> 重要
>
> - NetScaler ADC リリース 12.1 ビルド 55.13 から、SNIP の代わりに NSIP を使用して`Logstream`レ

> コードを送信できます。`set appflow param`コマンドに新しいパラメータ「LogStreamOvernSip」が導入されました。デフォルトでは、「logstreamOverNSIP」パラメーターは「無効」になっているため、「有効」にする必要があります。構成の詳細については、例を参照してください。
>
> - NetScaler ADC リリース 13.0 ビルド 58.x リリースから、AppFlow 機能で Web SaaS アプリケーションオプションを有効にできます。NetScaler Gateway サービスから Web アプリケーションまたは SaaS アプリケーションのデータ使用量を受け取ることができます。構成の詳細については、例を参照してください。

コマンドプロンプトで、次のコマンドを入力して AppFlow パラメーターを設定し、設定を確認します。

```
- set appflow param [-templateRefresh <secs>] [-appnameRefresh <secs>]
   [-flowRecordInterval <secs>] [-udpPmtu <positive_integer>] [-
  httpUrl ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-httpCookie ( \*\*
  ENABLED\*\* | \*\*DISABLED\*\* )] [-httpReferer ( \*\*ENABLED\*\* |
  \*\*DISABLED\*\* )] [-httpMethod ( \*\*ENABLED\*\* | \*\*DISABLED
  \*\* )] [-httpHost ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-
  httpUserAgent ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-
  httpXForwardedFor ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )][-
  clientTrafficOnly ( \*\*YES\*\* | \*\*NO\*\*)] [-
  webSaaSAppUsageReporting ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-
  logstreamOverNSIP ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )]

- show appflow Param

<!--NeedCopy-->
```

例

```
set appflow Param -templateRefresh 240 -udpPmtu 128 -httpUrl enabled -
   webSaaSAppUsageReporting ENABLED -logstreamOverNSIP ENABLED

<!--NeedCopy-->
```

構成ユーティリティを使用して **AppFlow** パラメーターを設定する

[システム] **> [AppFlow**] に移動し、[**AppFlow** 設定の変更] をクリックして、関連する AppFlow パラメーターを指定します。

加入者 **ID** の難読化のサポート

NetScaler ADC リリース 13.0 ビルド 35.xx 以降、AppFlow 構成は、レイヤー 4 またはレイヤー 7 の AppFlow レコードで MSISDN を難読化するための「subscriberIDObfuscation」アルゴリズムをサポートするように拡張されています。ただし、アルゴリズムを MD5 または SHA256 として構成する前に、まず AppFlow パラメーターとして有効にする必要があります。このパラメーターはデフォルトでは無効になっています。

CLI を使用して加入者 **ID** の難読化アルゴリズムを設定する

コマンドプロンプトで入力します。

```
set appflow param [-subscriberIdObfuscation ( ENABLED | DISABLED )  [-
    subscriberIdObfuscationAlgo ( MD5 | SHA256 )]]

<!--NeedCopy-->
```

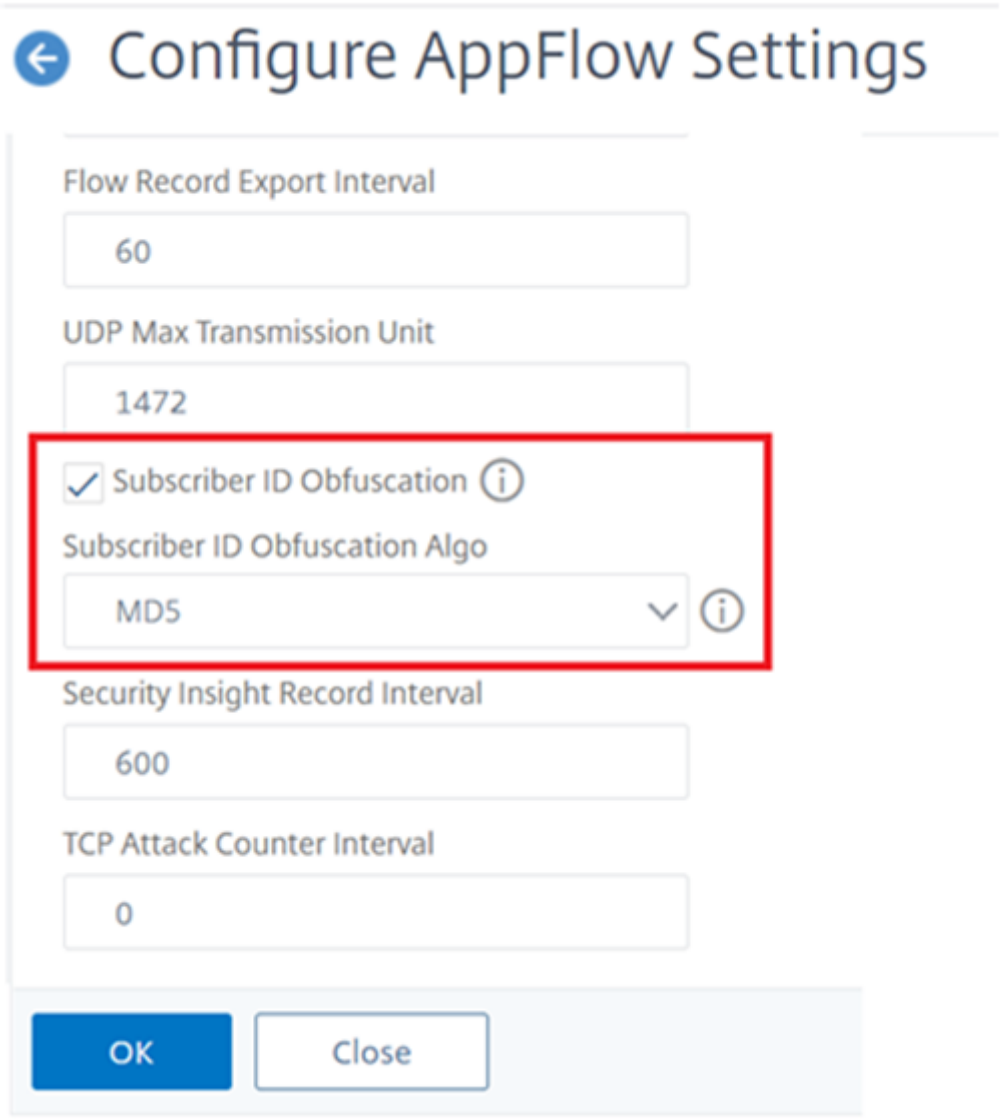
例

```
set appflow param  - subscriberIdObfuscation ENABLED  -
    subscriberIdObfuscationAlgo SHA256

<!--NeedCopy-->
```

GUI を使用して加入者 **ID** 難読化アルゴリズムを設定する

1. [システム**] > [AppFlow**] に移動します。
2. AppFlow の詳細ペインで、[設定] の [****AppFlow** 設定の変更 **] をクリックします。
3. [AppFlow 設定の構成] ページで、次のパラメーターを設定します。
   - 加入者 **ID** の難読化。L4/L7 AppFlow レコードで難読化 MSISDN のオプションを有効にします。
   - 購読者 **ID** 難読化アルゴ。アルゴリズムタイプとして [MD5] または [SHA256] を選択します。
4. 「**OK**」をクリックして「閉じる」をクリックします。



例:**DataStream** 用に **AppFlow** を構成する

次の例は、コマンドラインインターフェイスを使用して DataStream の AppFlow を構成する手順を示しています。

```
enable feature appflow

add db user sa password freebsd

add lbvserver lb0 MSSQL 10.102.147.97 1433 -appflowLog ENABLED

add service sv0 10.103.24.132 MSSQL 1433 -appflowLog ENABLED

bind lbvserver lb0 sv0

```



```
add appflow collector col0 -IPAddress 10.102.147.90

add appflow action act0 -collectors col0

add appflow policy pol0 "mssql.req.query.text.contains("select")" act0

bind lbvserver lb0 -policyName pol0 -priority 10

<!--NeedCopy-->
```

NetScaler ADC アプライアンスがデータベース要求を受信すると、アプライアンスは構成されたポリシーに対して要求を評価します。一致が見つかると、その詳細がポリシーで構成されている AppFlow コレクタに送信されます。

メトリクスコレクターの設定

メトリックコレクターは、NetScaler で有効化して、NetScaler からさまざまなエンドポイントにメトリックを収集およびエクスポートできるサービスです。メトリクスは Avro と Prometheus の 2 つの形式でエクスポートできます。エクスポートされたメトリックを処理および視覚化して、有意義な洞察を得ることができます。デフォルトでは、メトリクスコレクターは 30 秒ごとの時系列分析データのエクスポートをサポートします。ただし、時系列分析プロファイルデータをエクスポートする間隔を決定できるように、30 秒から 300 秒までの値に設定できます。

CLI を使用してメトリックコレクターを設定するには、次の手順を実行します。

1. 次のコマンドを使用して、IP アドレス、プロトコル、およびポートを使用してコレクタサービスを設定します。

   ```
   add service <metrics_service_name> <ip-address> <protocol> <port>
   ```

   例:

   ```
   add service metrics_service1 192.168.1.1 HTTP 5563
   ```

2. 分析時系列プロファイルを設定して、メトリックデータをコレクターサービスに送信します。コレクターサービス、メトリクスをエクスポートする頻度、および出力モードを指定します。

   ```
   set analytics profile ns_analytics_time_series_profile -collectors
        <metrics_service_name> -type timeseries -metrics ENABLED
       metricsExportFrequency <30-300> -outputMode <avro/prometheus>
   ```

   例:

   ```
   set analytics profile ns_analytics_time_series_profile -collectors
        metrics_service1 -type timeseries -metrics Enabled
       metricsExportFrequency 90 -outputMode prometheus --serveMode
       PUSH
   ```

   > 注:
   >
   > この例では、デフォルトの時系列プロファイル`ns_analytics_time_series_profile`を

> 使用しています。時系列プロファイルを作成する場合は、`add analytics profile` コマンドを使用できます。

この例では、メトリクスのエクスポート頻度は 90 秒に設定され、エクスポートモードは Prometheus に指定されています。

`show analytics profile <analytics-profile-name>` 次のコマンドを使用してメトリクスコレクタの設定を確認します。

```
show analytics profile ns_analytics_time_series_profile

    Name: ns_analytics_time_series_profile
    Collector: metrics_service1
    Profile-type: timeseries
        Output Mode: Prometheus
        Metrics: ENABLED
          Schema File: schema.json
          Metrics Export Frequency: 90
        Events: DISABLED
        Auditlog: DISABLED
        Serve mode: Pull
    Reference Count: 0
```

デバッグメトリックスコレクター

必要なデバッグログは、`/var/nslog/metricscollector.log`の場所に保存されます。

メトリックスファイルの生成

`metrics_<format>_log.*`ファイルは`/var/nslog/`フォルダーの場所に生成されます。

メトリクスコレクターでの動的スキーマのサポート

動的スキーマカウンターのサポートにより、カウンターのリストを含むスキーマファイルを、必要に応じて実行時に更新できます。デフォルトでは、`/var/metrics_conf/schema.json` ファイルにはカウンタのリストが設定されます。

> 注:
>
> メトリックコレクターのデフォルトスキーマファイル`/var/metrics_conf/schema.json` は、installns 手順で NetScaler ADC アプライアンスにインストールできます。

CLI を使用してカウンターをサブスクライブするようにメトリクスコレクターを構成する

コレクターサービスを構成して、メトリクスのエクスポートを開始します。

コマンドプロンプトで入力します。

```
set analytics profile ns_analytics_time_series_profile -metrics ENABLED
    -collectors <collector_name> -schemaFile  schema.json -outputMode <
   avro | prometheus>

<!--NeedCopy-->
```

> 注:
>
> schema.json はデフォルトの SchemaFile 構成です。

必要なカウンターのセットを含む新しいスキーマファイルは、CLI コマンドを使用してメトリックコレクターがエクスポートするように構成できます。スキーマファイルは/var/metrics_conf/の場所に存在する必要があります。

stats infra でサポートされるカウンタのリスト (reference_schema.json) をすべて含むスキーマファイルが、/var/metrics_conf/の場所に存在します。このファイルは、カウンターのカスタムリストを作成するための参照として使用できます。

CLI を使用してスキーマファイルを設定します

```
set analytics profile ns_analytics_time_series_profile -metrics ENABLED
    -collectors <collector name> -schemaFile <schema file_name> -
   outputMode <avro | prometheus>

<!--NeedCopy-->
```

必要なカウンタを含む新しいスキーマファイルは、前述の CLI コマンドを使用してメトリックコレクターがエクスポートできるように追加および設定できます。

統計インフラでサポートされているすべてのカウンタのリスト (reference_schema.json) を含む参照スキーマファイルが/var/metrics_conf/の場所にあります。このファイルは、カウンターのカスタムリストを作成するための参照として使用できます。

コマンドプロンプトで **CLI** 設定出力を確認します。

```
show analytics profile ns_analytics_time_series_profile

    Name: ns_analytics_time_series_profile
    Collector: <collector_name>
    Profile-type: timeseries
    Output Mode: avro
    Metrics: ENABLED
    Schema File: schema.json
    Events: ENABLED
    Auditlog: DISABLED
    Serve mode: Push
    Reference Count: 0
```



```
13
14 <!--NeedCopy-->
```

エクスポートされたカウンタのリストを更新する手順

次の手順では、エクスポートされたカウンタのリストを更新する手順について説明します。

1. カスタム/新しいスキーマファイルを更新します。
2. 使用する更新されたスキーマファイルの CLI 設定に表示される-metricsオプションを使用して、メトリクスを無効または有効にします。

複数の時系列プロファイルのサポート

メトリックコレクターは、NetScaler ADC アプライアンスで最大 3 つの時系列プロファイル構成をサポートします。各時系列を次のように構成できます。

- コレクター。
- エクスポートに必要なカウンタのセットを含むスキーマファイル。
- メトリックをエクスポートするデータ形式。
- メトリクスの監査ログとイベントを有効または無効にするオプション。

複数の時系列プロファイルがサポートされているため、メトリックコレクターは（構成されたスキーマファイルに基づいて）メトリックの異なるセットを異なるフォーマット（AVRO、Prometheus、Influx）のさまざまなコレクターに同時にエクスポートできます。

CLI を使用して時系列プロファイルを追加する

コマンドプロンプトで入力します。

```
1 add analytics profile <profile_name> -type timeseries
2 <!--NeedCopy-->
```

CLI を使用して時系列プロファイルを設定する

コマンドプロンプトで入力します。

```
1 set analytics profile <profile_name> -metrics <DISABLED|ENABLED> -
      auditlogs <DISABLED|ENABLED> -events <DISABLED|ENABLED> -collectors
      <collector_name> -schemaFile schema.json -outputMode <avro | influx
      | prometheus>
2
3 <!--NeedCopy-->
```


複数の時系列プロファイルをサポートするログファイルの命名規則

- Avro ログファイルは`metrics_avro_<profile_name>_log.*`として生成されます。
- Prometheus のログファイルは次のように生成されます。`metrics_prom_<profile_name>.log`。

> 注:
>
> - 設定したすべての時系列プロファイルでメトリクスを有効にできますが、イベントと監査ログは 1 つのプロファイルでのみ有効化できます。
> - 動的スキーマ機能は、バージョン 13.1 ビルド 23.16 以降でサポートされています。
> - 複数の時系列プロファイルは、バージョン 13.1 ビルド 33.6 以降でサポートされています。

Web ページのパフォーマンスデータを **AppFlow** コレクタにエクスポートする

August 15, 2023

EdgeSight Monitoring アプリケーションは、NetScaler ADC 環境で提供されるさまざまな Web アプリケーションのパフォーマンスを監視できる Web ページ監視データを提供します。これで、このデータを AppFlow コレクタにエクスポートして、ウェブページアプリケーションの詳細な分析を取得できます。IPFIX 標準に基づく AppFlow は、EdgeSight の監視だけよりもウェブアプリケーションのパフォーマンスに関するより具体的な情報を提供します。

EdgeSight Monitoring データを AppFlow コレクターにエクスポートするように、負荷分散とコンテンツスイッチング仮想サーバーの両方を構成できます。AppFlow エクスポート用に仮想サーバーを構成する前に、AppFlow アクションを EdgeSight モニタリングレスポンダーポリシーに関連付けます。

次のウェブページのパフォーマンスデータは AppFlow にエクスポートされます。

- ページの読み込み時間。ブラウザが応答の最初のバイトを受信し始めてからユーザーがページを操作し始めるまでの経過時間（ミリ秒単位）。この段階では、すべてのページコンテンツが読み込まれない可能性があります。
- ページレンダリング時間。ブラウザがレスポンスの最初のバイトを受け取ってから、すべてのページコンテンツがレンダリングされるか、ページ読み込みアクションがタイムアウトするまでの経過時間（ミリ秒）。
- ページでの滞在時間。ユーザーがページ上で費やした時間。あるページリクエストから次のページリクエストまでの時間を表します。

AppFlow は、インターネットプロトコルフロー情報 eXport (IPFIX) 形式を使用してパフォーマンスデータを送信します。これは、RFC 5101 で定義されているオープンなインターネット技術標準化委員会 (IETF) 標準です。AppFlow テンプレートは、次の企業固有の情報要素（EIE）を使用して情報をエクスポートします。

- クライアントのロード終了時刻。ブラウザがレスポンスの最後のバイトを受信し、画像、スクリプト、スタイルシートなどのページのすべてのオブジェクトをロードする時間。

- クライアントのロード開始時刻。ブラウザーがレスポンスの最初のバイトを受信して、画像、スクリプト、スタイルシートなどのページのオブジェクトをロードする時間。
- クライアントレンダリング終了時間。ブラウザが埋め込みオブジェクトを含むページ全体のレンダリングを終了した時刻。
- クライアントレンダリング開始時刻。ブラウザがページのレンダリングを開始した時刻。

Web ページのパフォーマンスデータを **AppFlow** コレクタにエクスポートするための前提条件

AppFlow アクションを AppFlow ポリシーに関連付ける前に、次の前提条件が満たされていることを確認します。

- AppFlow 機能が有効化され、設定されました。
- レスポンダー機能が有効になりました。
- EdgeSight モニタリング機能が有効になりました。
- EdgeSight Monitoring は、パフォーマンスデータを収集するアプリケーションのサービスにバインドされた負荷分散またはコンテンツスイッチング仮想サーバー上で有効になっています。

AppFlow アクションを **EdgeSight** モニタリングレスポンダーポリシーに関連付ける

Web ページのパフォーマンスデータを AppFlow コレクタにエクスポートするには、AppFlow アクションを EdgeSight モニタリングレスポンダーポリシーに関連付ける必要があります。AppFlow アクションは、トラフィックを受信するコレクタのセットを指定します。

CLI を使用して **AppFlow** アクションを **EdgeSight** モニタリングレスポンダーポリシーに関連付けるには

コマンドプロンプトで入力します。

```
set responder policy <name> -appflowAction <action_Name>
<!--NeedCopy-->
```

例

```
set responder policy pol -appflowAction actn
<!--NeedCopy-->
```

GUI を使用して **AppFlow** アクションを **EdgeSight** モニタリングレスポンダーポリシーに関連付けるには

1. **AppExpert> **** レスポンダー **>** ポリシーに移動します。**
2. 詳細ウィンドウで、EdgeSight Monitoring レスポンダーポリシーを選択し、[開く] をクリックします。
3. [レスポンダーポリシーの構成] ダイアログボックスの [**AppFlow** アクション] ドロップダウンリストで、Web ページのパフォーマンスデータを送信するコレクタに関連付けられた AppFlow アクションを選択します。
4. [**OK**] をクリックします。

EdgeSight 統計を **AppFlow** コレクタにエクスポートするように仮想サーバーを構成する

EdgeSight 統計情報を仮想サーバーから AppFlow コレクターにエクスポートするには、AppFlow アクションを仮想サーバーに関連付ける必要があります。

GUI を使用して負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーに **AppFlow** アクションを関連付けるには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。[トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動することもできます。
2. 詳細ペインで、1 つまたは複数の仮想サーバーを選択し、「**EdgeSight** 監視を有効にする」をクリックします。
3. [EdgeSight Monitoring を有効にする] ダイアログボックスで、[**EdgeSight** 統計情報を **Appflow** にエクスポート] チェックボックスをオンにします。
4. [AppFlow アクション] ドロップダウンリストから、**AppFlow** アクションを選択します。AppFlow アクションは、EdgeSight Monitoring 統計をエクスポートする AppFlow コレクタのリストを定義します。複数の負荷分散仮想サーバーを選択した場合、同じ AppFlow アクションがそれらにバインドされたレスポンダーポリシーに関連付けられます。必要に応じて、選択した負荷分散仮想サーバーごとに構成された AppFlow アクションを個別に変更できます。
5. [**OK**] をクリックします。

NetScaler 高可用性ペアでのセッションの信頼性

August 15, 2023

ICA セッション中にネットワークの中断またはデバイスのフェイルオーバーが発生した場合、セッションの再接続では、セッション画面の保持またはクライアントの自動再接続という 2 つのメカニズムのいずれかを使用できます。

セッションの信頼性。推奨モードは、ユーザーにとってスムーズなエクスペリエンスです。この中断は、短時間のネットワークの中断ではほとんど目立ちません。

クライアントの自動再接続。フォールバックオプションには、クライアントの再起動が含まれます。このメカニズムはユーザーにとって混乱を招き、常にサポートされているわけではありません。

HDX Insight が有効になっている場合、Receiver は ICA セッション画面の保持機能を使用して ICA セッションをシームレスに再接続できます。
この機能は、スタンドアロン構成と NetScaler ADC HA ペア構成の両方で、また NetScaler ADC フェイルオーバーが発生した場合でも機能します。

> 注:
>
> - NetScaler ADC アプライアンスは、ソフトウェアバージョン 11.1 ビルド 49.16 以降で実行されている

> 必要があります。
> - NetScaler ADC アプライアンスにアクティブな接続がある場合は、セッション画面の保持モードを有効または無効にしないでください。
> - 接続がまだアクティブなときにこの機能を有効または無効にすると、HDX Insight はフェイルオーバーの発生後にこれらのセッションの解析を停止します。その結果、セッションに関する情報が失われます。
> - 高可用性セットアップでのセッション画面の保持は、NetScaler ADC ソフトウェアバージョン 11.1 49.16 以降ではデフォルトで無効になっています。セッション画面の保持は、セットアップの両方のノードが同じビルド（たとえば、リリース 11.1 ビルド 53）を実行している場合にのみ、高可用性セットアップでサポートされます。つまり、両方のノードが異なるビルドを実行している場合（たとえば、一方のノードにリリース 11.1 のビルド 53 があり、もう一方のノードにリリース 11.1 のビルド 56 がある場合）、セッション画面の保持は高可用性セットアップではサポートされません。SSL VDA のセッションの信頼性は、次の条件が満たされた場合にサポートされます。
>   - The “EnableSRonHAFailover” parameter in the `set ica parameter` command must be YES.
>   - The HTTPS must be used instead of HTTP while configuring the virtual server.
> - HDX Insight を有効にすると、EnableSronhaFailover パラメーターが無効になっていても、高可用性フェイルオーバー後に基本暗号化アプリケーションとデスクトップが再接続されます。

CLI を使用してセッション画面の保持を設定するには、次の手順を実行します。

1. コマンドラインで、既定のシステム管理者の資格情報を使用してシステムにログオンします。
2. HA フェールオーバーでセッション画面の保持を有効にするには、プロンプトで次のように入力します。
   ```
   set ica parameter EnableSRonHAFailover YES
   ```
3. HA フェールオーバーでセッション画面の保持を無効にするには、プロンプトで次のように入力します。
   ```
   set ica parameter EnableSRonHAFailover NO
   ```

GUI を使用して **HA** フェールオーバーでセッション画面の保持を有効にするには、次の手順を実行します。

1. Web ブラウザで、HA ペアのプライマリ NetScaler ADC インスタンスの IP アドレスを入力します（例: http://192.168.100.1）。
2. [**User Name**] と [**Password**] に管理者の資格情報を入力します。
3. [構成] タブで、[システム] > [設定] に移動し、[**ICA** パラメータの変更] をクリックします。
4. [**ICA** パラメータの変更] セクションで、[高可用性フェイルオーバー時のセッション画面の保持] を選択します。
5. [**OK**] をクリックします。

制限事項

- この機能を有効にすると、この機能によって ICA 圧縮が無効になるため、帯域幅の消費が増加します。また、プライマリノードとセカンダリノード間の追加のトラフィックにより、それらの同期が維持されます。

- この機能はアクティブ-パッシブモードでのみサポートされています。アクティブ-アクティブモードは現在サポートされていません。
- HDX Insight が有効で、HA ノブのセッション画面の保持が［いいえ］に設定されている場合、NetScaler 高可用性フェイルオーバーシナリオでは ACR 再接続モードのみがサポートされます。HDX Insight が無効になっている場合、HA ノブはセッション画面の保持を無効にしません。

セッション再接続セマンティクスの 表は次のとおりです。

セッションはセマンティクスを再接続

ステータス	RONHA フェイルオーバーを有効にするはい	EnableSRonHAFailover No（デフォルト）
HDX Insight 有効	ICA セッションのセッション再接続は機能する	ICA セッションのセッション再接続が機能しない
HDX Insight 無効	ICA セッションのセッション再接続は機能する	ICA セッションのセッション再接続は機能する

> 注意事項

- ICA セッションのセッション画面の保持は、NetScaler Gateway でそのまま使用できます。
- 次の両方の条件が満たされている場合、ICA セッションのセッション画面の保持は機能しません。
  - HDX Insight が有効になっている
  - EnableSRonHAFailover が NO に設定されている
- HDX Insight が無効になっている場合、EnableSronhaFailover ノブを「はい」または「いいえ」に設定しても違いはありません。

NetScaler Web App Firewall

August 15, 2023

NetScaler Web App Firewall には、さまざまなアプリケーションセキュリティ要件を満たすように簡単に構成できるオプションが用意されています。一連のセキュリティチェックで構成される Web App Firewall プロファイルを使用すると、詳細なパケットレベルの検査を行うことで、要求と応答の両方を保護できます。各プロファイルには、基本保護または高度な保護を選択するオプションが含まれています。保護機能によっては、他のファイルを使用する必要がある場合があります。たとえば、XML 検証チェックには WSDL ファイルまたはスキーマファイルが必要な場合があります。プロファイルは、署名やエラーオブジェクトなどの他のファイルを使用することもできます。これら

のファイルはローカルで追加することも、事前にインポートして将来使用するためにアプライアンスに保存することもできます。

各ポリシーはトラフィックのタイプを識別し、そのトラフィックは、ポリシーに関連付けられたプロファイルに指定されたセキュリティチェック違反がないか検査されます。ポリシーには異なるバインドポイントを設定でき、それによってポリシーの範囲が決まります。たとえば、特定の仮想サーバーにバインドされたポリシーが呼び出され、その仮想サーバーを通過するトラフィックのみについて評価されます。ポリシーは、指定された優先順位の順に評価され、リクエストまたはレスポンスに最初に一致したポリシーが適用されます。

- Web App Firewall 保護の迅速な導入

  Web App Firewall セキュリティを迅速に展開するには、次の手順を使用できます。

  1. Web App Firewall プロファイルを追加し、アプリケーションのセキュリティ要件に適したタイプ (html、xml、JSON) を選択します。
  2. 必要なセキュリティレベル (基本または詳細) を選択します。
  3. 署名や WSDL などの必要なファイルを追加またはインポートします。
  4. ファイルを使用するようにプロファイルを設定し、その他の必要な変更をデフォルト設定に加えます。
  5. このプロファイルに Web App Firewall ポリシーを追加します。
  6. ポリシーをターゲットバインドポイントにバインドし、優先度を指定します。

- Web App Firewall エンティティ

  Profile: Web App Firewall プロファイルは、検索対象と何をすべきかを指定します。要求と応答の両方を検査して、チェックする必要のある潜在的なセキュリティ違反と、トランザクションの処理時に実行する必要のあるアクションを決定します。プロファイルは、HTML、XML、または HTML と XML のペイロードを保護できます。アプリケーションのセキュリティ要件に応じて、基本プロファイルまたは詳細プロファイルを作成できます。基本的なプロファイルは、既知の攻撃から保護できます。より高いセキュリティが必要な場合は、高度なプロファイルをデプロイしてアプリケーションリソースへのアクセスを制御し、ゼロデイ攻撃をブロックできます。ただし、基本プロファイルを変更して高度な保護を提供することも、その逆も可能です。複数のアクション (ブロック、ログ、学習、変換など) を選択できます。高度なセキュリティチェックでは、クライアント接続の制御と監視にセッション Cookie と隠しフォームタグを使用する場合があります。Web App Firewall プロファイルは、トリガーされた違反を学習し、緩和ルールを提案できます。

  基本保護: 基本プロファイルには、開始 URL および拒否 URL 緩和ルールの事前構成済みセットが含まれます。これらの緩和ルールは、どの要求を許可し、どの要求を拒否する必要があるかを決定します。受信したリクエストはこれらのリストと照合され、設定されたアクションが適用されます。これにより、ユーザーは緩和ルールの構成を最小限に抑えてアプリケーションを保護できます。開始 URL ルールは、強制的なブラウジングから保護します。ハッカーによって悪用される既知の Web サーバーの脆弱性は、デフォルトの URL の拒否ルールのセットを有効にすることで検出およびブロックできます。バッファオーバーフロー、SQL、クロスサイトスクリプティングなど、一般的に起動される攻撃も簡単に検出できます。

  Advanced Protections: 名前が示すように、高度な保護は、より高いセキュリティ要件を持つアプリケーションに使用されます。リラクゼーションルールは、特定のデータのみへのアクセスを許可し、残りのデータ

をブロックするように構成されています。このポジティブなセキュリティモデルは、基本的なセキュリティチェックでは検出されない可能性のある未知の攻撃を軽減します。すべての基本的な保護機能に加えて、高度なプロファイルではブラウジングの制御、Cookie のチェック、さまざまなフォームフィールドの入力要件の指定、フォームの改ざんやクロスサイトリクエスト偽造攻撃からの保護などにより、ユーザーセッションを追跡します。トラフィックを観察して適切な緩和策を講じるラーニングは、多くのセキュリティチェックでデフォルトで有効になっています。使いやすいが、高度な保護は、セキュリティを強化するだけでなく、より多くの処理を必要とし、パフォーマンスに影響を与えることができるキャッシュの使用を許可しないため、十分に考慮する必要があります。

インポート: インポート機能は、Web App Firewall プロファイルで外部ファイル、つまり外部または内部 Web サーバーでホストされているファイル、またはローカルマシンからコピーする必要がある場合に役立ちます。ファイルをインポートしてアプライアンスに保存すると、特に外部 Web サイトへのアクセスを制御する必要がある場合、コンパイルに時間がかかる場合、大きなファイルを HA デプロイメント間で同期する必要がある場合、または次の方法でファイルを再利用できる場合に便利です。複数のデバイス間でコピーします。次に例を示します:

- 外部 Web サーバーでホストされている WSDL は、外部 Web サイトへのアクセスをブロックする前にローカルにインポートできます。
- Cenzic などの外部スキャンツールで生成された大きな署名ファイルは、NetScaler ADC アプライアンスのスキーマを使用してインポートおよびプリコンパイルできます。
- カスタマイズされた HTML または XML エラーページは、外部 Web サーバーからインポートすることも、ローカルファイルからコピーすることもできます。

署名: シグニチャは、パターンマッチングを使用して悪意のある攻撃を検出し、トランザクションのリクエストとレスポンスの両方をチェックするように構成できるため、強力です。これらは、カスタマイズ可能なセキュリティソリューションが必要な場合に推奨されるオプションです。シグネチャの一致が検出されたときに実行するアクションには、複数の選択肢（ブロック、ログ、学習、変換など）を使用できます。Web App Firewall には、1,300 を超える署名ルールで構成される既定の署名オブジェクトが組み込まれており、自動更新機能を使用して最新のルールを取得することもできます。他のスキャンツールで作成されたルールもインポートできます。署名オブジェクトは、Web App Firewall プロファイルで指定されている他のセキュリティチェックと連携できる新しいルールを追加することでカスタマイズできます。シグニチャルールには複数のパターンを含めることができ、すべてのパターンが一致した場合にのみ違反のフラグを立てることができるため、誤検出を防ぐことができます。ルールのリテラル `fastmatch` パターンを注意深く選択すると、処理時間を大幅に最適化できます。

ポリシー: Web App ファイアウォールポリシーは、トラフィックをフィルタリングし、異なるタイプに分割するために使用されます。これにより、アプリケーションデータに対してさまざまなレベルのセキュリティ保護を実装する柔軟性が得られます。機密性の高いデータへのアクセスは高度なセキュリティチェック検査の対象となり、機密性の低いデータは基本レベルのセキュリティ検査で保護されます。ポリシーは、無害なトラフィックのセキュリティチェック検査をバイパスするように設定することもできます。セキュリティを高くするとより多くの処理が必要になるため、ポリシーを慎重に設計することで、パフォーマンスを最適化しながら必要なセキュリティを実現できます。ポリシーの優先度によって評価される順序が決まり、そのバインドポイン

トによって適用範囲が決まります。

ハイライト

1. さまざまなタイプのデータを保護し、さまざまなリソースに適切なレベルのセキュリティを実装しながら、パフォーマンスを最大限に引き出すことで、幅広いアプリケーションを保護できます。
2. セキュリティ設定を柔軟に追加または変更できます。基本保護と高度な保護を有効または無効にすることで、セキュリティチェックを強化または緩和できます。
3. HTML プロファイルを XML または Web2.0 (HTML+XML) プロファイルに変換できます。逆に、さまざまなタイプのペイロードに柔軟にセキュリティを追加できます。
4. 簡単に導入できるアクションで、攻撃をブロックしたり、ログで監視したり、統計情報を収集したり、一部の攻撃文字列を変換して無害化したりすることができます。
5. 受信したリクエストを検査することで攻撃を検出し、サーバーから送信された応答を検査することで機密データの漏洩を防ぐことができます。
6. トラフィックパターンから学習して、簡単に編集できる緩和ルールを提案し、例外を許容するように展開できます。
7. カスタマイズ可能なシグネチャの力を利用して特定のパターンに一致する攻撃をブロックし、ポジティブセキュリティモデルチェックを基本または高度なセキュリティ保護に柔軟に使用できるハイブリッドセキュリティモデル。
8. PCI-DSS 準拠に関する情報を含む、包括的な構成レポートが入手可能。

よくある質問と導入ガイド

August 15, 2023

Q: **NetScaler Web App Firewall** がアプリケーションのセキュリティ保護に適した選択肢であるのはなぜですか

次の機能により、NetScaler Web App Firewall は包括的なセキュリティソリューションを提供します。

- ハイブリッドセキュリティモデル： NetScaler ハイブリッドセキュリティモデルを使用すると、ポジティブセキュリティモデルとネガティブセキュリティモデルの両方を活用して、アプリケーションに最適な構成を作成できます。
  - ポジティブなセキュリティモデルは、バッファオーバーフロー、CGI-BIN パラメータ操作、フォーム/隠しフィールドの操作、強制的なブラウジング、クッキーまたはセッションポイズニング、壊れた ACL、クロスサイトスクリプティング（クロスサイトスクリプティング）、コマンドインジェクション、SQL インジェクション、エラートリガーセンシティブから保護します情報漏洩、暗号化の安全でない使用、

サーバーの設定ミス、バックドアとデバッグオプション、レートベースのポリシー適用、よく知られているプラットフォームの脆弱性、ゼロデイエクスプロイト、クロスサイトリクエストフォージェリ(CSRF)、クレジットカードおよびその他の機密データの漏洩。

- ネガティブセキュリティモデルは、リッチセットシグニチャを使用して、L7 および HTTP アプリケーションの脆弱性から保護します。Web App Firewall は、Cenzic、Qualys、Whitehat、および IBM が提供するいくつかのサードパーティ製スキャンツールと統合されています。組み込みの XSLT ファイルにより、ルールを簡単にインポートできます。これは、ネイティブ形式の Snort ベースのルールと組み合わせて使用できます。自動更新機能は、新しい脆弱性の最新の更新プログラムを取得します。

ポジティブ・セキュリティ・モデルは、誰がどのデータにアクセスできるかを完全に制御するオプションを提供するため、セキュリティに対する高いニーズを持つアプリケーションを保護するための好ましい選択肢となる場合があります。あなたが望むものだけを許可し、残りはブロックします。このモデルには組み込みのセキュリティチェック構成が含まれており、数回のクリックで展開できます。ただし、セキュリティが厳しくなるほど、処理のオーバーヘッドが大きくなることに注意してください。

カスタマイズしたアプリケーションには、ネガティブセキュリティモデルが適している場合があります。シグニチャを使用すると、複数の条件を組み合わせることができ、一致と指定されたアクションは、すべての条件が満たされた場合にのみトリガーされます。望まないものだけブロックしてあとは許可する。特定の高速一致パターンを指定の場所に配置すると、処理オーバーヘッドを大幅に削減してパフォーマンスを最適化できます。アプリケーションの特定のセキュリティニーズに基づいて独自のシグニチャルールを追加するオプションにより、独自のカスタムセキュリティソリューションを柔軟に設計できます。

- リクエストとレスポンス側の検出と保護：受信したリクエストを検査して疑わしい動作を検出し、適切なアクションを実行できます。また、レスポンスをチェックして、機密データの漏洩を検出して保護できます。
- **HTML**、**XML**、**JSON** ペイロードの豊富な組み込み保護セット**:** Web App Firewall は 19 種類のセキュリティチェックを提供します。そのうちの 6 つ (開始 URL や URL の拒否など) は、HTML データと XML データの両方に適用されます。5 つのチェック (フィールドの一貫性とフィールド形式など) は HTML に固有で、8 つ (XML 形式と Web サービスの相互運用性など) は XML ペイロードに固有です。この機能には、豊富なアクションとオプションが含まれています。たとえば、URL クロージャを使用すると、Web サイト内のナビゲーションを制御および最適化して、正当な URL をすべて許可する緩和ルールを設定することなく、強制的なブラウジングから保護できます。応答では、クレジットカード番号などの機密データを削除するか、または除外するかを選択できます。SOAP アレイ攻撃保護、XML サービス拒否 (XDoS)、WSDL スキャン防止、添付ファイルチェック、その他の任意の XML 攻撃のいずれであっても、アプリケーションが Web App Firewall によって保護されている場合、データを保護する鉄製のシールドがあるという安心感があります。シグニチャを使用すると、XPath-Expressions を使用してルールを設定して、JSON ペイロードのヘッダーだけでなく、本文の違反を検出できます。
- **GWT:** SQL、クロスサイトスクリプティング、フォームフィールドの一貫性チェック違反から保護するための Google Web Toolkit アプリケーションの保護をサポートします。
- **Java** を使用せずに、ユーザーフレンドリーなグラフィカルユーザーインターフェイス **(GUI):** 直感的な GUI と事前設定されたセキュリティチェックにより、いくつかのボタンをクリックするだけでセキュリティを簡単

に展開できます。ウィザードのプロンプトが表示され、プロファイル、ポリシー、シグニチャ、バインディングなどの必要な要素を作成するように指示されます。HTML5 ベースの GUI には、Java の依存関係はありません。そのパフォーマンスは、古い Java ベースのバージョンよりも大幅に優れています。

- 使いやすく自動化可能な **CLI:** GUI で使用できるほとんどの構成オプションは、コマンドラインインターフェイス (CLI) でも使用できます。CLI コマンドはバッチファイルで実行でき、簡単に自動化できます。
- **REST API** のサポート： NetScaler NITRO プロトコルは、Web App Firewall 構成を自動化し、セキュリティ違反の継続的な監視に関連する統計を収集するための豊富な REST API セットをサポートしています。
- 学習**:** Web App Firewall のトラフィックを監視してセキュリティを微調整することによって学習する機能は、非常にユーザーフレンドリーです。学習エンジンはルールを推奨するため、正規表現に習熟していない緩和を簡単に展開できます。
- **RegEx** エディタのサポート：正規表現は、ルールを統合して検索を最適化したいというジレンマに対する洗練されたソリューションを提供します。正規表現の力を活用して、URL、フィールド名、シグニチャパターンなどを設定できます。豊富な組み込みの GUI RegEx エディターは、式のクイックリファレンスを提供し、RegEx の精度を検証およびテストする便利な方法を提供します。
- カスタマイズされたエラーページ**:** ブロックされたリクエストは、エラー URL にリダイレクトできます。また、サポートされている変数と NetScaler の高度なポリシー（高度な PI 表現）を使用してカスタマイズされたエラーオブジェクトを表示し、クライアントにトラブルシューティング情報を埋め込むこともできます。
- **PCI-DSS**、統計、およびその他の違反レポート**:** 豊富なレポートセットにより、PCI-DSS コンプライアンス要件への対応、トラフィックカウンタに関する統計情報の収集、すべてのプロファイルまたは 1 つのプロファイルの違反レポートの表示を簡単に行うことができます。
- ログ記録とログからのクリックルール**:** 詳細ロギングは、ネイティブ形式と CEF 形式でもサポートされています。Web App Firewall には、syslog ビューアでターゲットログメッセージをフィルタリングする機能があります。ボタンをクリックするだけで、ログメッセージを選択し、対応する緩和ルールを展開できます。ログメッセージを柔軟にカスタマイズできるほか、Web ログの生成もサポートできます。詳細については、「Web App Firewall ログ 」トピックを参照してください。
- 違反ログをトレースレコードに含める**:** トレースレコードにログメッセージを含める機能により、リセットやブロックなどの予期しない動作を簡単にデバッグできます。
- クローン作成：便利なインポート/エクスポートプロファイルオプションを使用すると、1 つの NetScaler アプライアンスのセキュリティ構成を他の NetScaler アプライアンスにクローンできます。学習データのエクスポートオプションを使用すると、学習したルールを Excel ファイルに簡単にエクスポートできます。その後、申請する前に、アプリケーションの所有者によってレビューおよび承認を受けることができます。
- **AppExpert** テンプレート (構成設定のセット) は、Web サイトに適切な保護を提供するように設計できます。これらのクッキーカッターテンプレートをテンプレートにエクスポートすることで、同様の保護を他のアプライアンスに展開するプロセスを簡素化し、迅速化できます。

詳細については、AppExpert テンプレートのトピックを参照してください。

- セッションレスセキュリティチェック**:** セッションレスセキュリティチェックを展開すると、メモリフットプリントを削減し、処理を迅速化するのに役立ちます。
- 他の **NetScaler** 機能との相互運用性： Web App Firewall は、書き換え、URL 変換、統合キャッシュ、CVPN、レート制限などの他の NetScaler 機能とシームレスに機能します。
- ポリシーでの **PI** 式のサポート**:** 高度な PI 式の機能を活用して、アプリケーションのさまざまな部分に異なるレベルのセキュリティを実装するポリシーを設計できます。
- **IPv6** のサポート**:** Web App Firewall は、IPv4 プロトコルと IPv6 プロトコルの両方をサポートしています。
- 位置情報ベースのセキュリティ保護： NetScaler Advanced Policy（PI Expression）を使用してロケーションベースのポリシーを柔軟に構成できます。これを組み込みの位置データベースと組み合わせて使用すると、ファイアウォールの保護をカスタマイズできます。悪意のある要求の発信元を特定し、特定の地理的場所から発信された要求に対して必要なレベルのセキュリティチェック検査を実施できます。
- パフォーマンス**:** リクエストサイドストリーミングにより、パフォーマンスが大幅に向上します。フィールドが処理されるとすぐに、結果のデータはバックエンドに転送され、残りのフィールドの評価は続行されます。処理時間の改善は、大きな投稿を処理する場合に特に重要です。
- その他のセキュリティ機能**:** Web App Firewall には、データのセキュリティを確保するのに役立つその他のセキュリティ設定がいくつかあります。たとえば、機密フィールドを使用すると、ログメッセージ内の機密情報の漏洩をブロックできます。また、**HTML** コメントを削除すると、クライアントに転送する前に応答から HTML コメントを削除できます。フィールドタイプは、アプリケーションに送信されるフォームで許可される入力を指定するために使用できます。

Q: Web App Firewall を構成するには何が必要ですか?

以下を実行します：

- Web App Firewall プロファイルを追加し、アプリケーションのセキュリティ要件に適したタイプ（html、xml、web2.0）を選択します。
- 必要なセキュリティレベル (基本または詳細) を選択します。
- 署名や WSDL などの必要なファイルを追加またはインポートします。
- ファイルを使用するようにプロファイルを設定し、その他の必要な変更をデフォルト設定に加えます。
- このプロファイルの Web App Firewall ポリシーを追加します。
- ポリシーをターゲットバインドポイントにバインドし、優先度を指定します。

Q: 選択するプロファイルの種類はどのようにわかりますか?

Web App Firewall プロファイルは、HTML ペイロードと XML ペイロードの両方を保護します。アプリケーションのニーズに応じて、HTML プロファイルまたは XML プロファイルのいずれかを選択できます。アプリケーションが HTML データと XML データの両方をサポートしている場合は、Web2.0 プロファイルを選択できます。

Q: 基本プロファイルと詳細プロファイルの違いは何ですか**?** どちらが必要かを決めるにはどうすればいいですか

基本プロファイルとアドバンスプロファイルのどちらを使用するかは、アプリケーションのセキュリティ上の必要性によって異なります。基本プロファイルには、[開始 URL] と [URL の拒否] 緩和ルールの事前設定セットが含まれています。これらの緩和ルールは、許可される要求と拒否される要求を決定します。受信要求は事前設定されたルールと一致し、構成されたアクションが適用されます。ユーザーは、緩和ルールの最小構成でアプリケーションを保護できます。開始 URL ルールは、強制的なブラウジングから保護します。ハッカーによって悪用される既知の Web サーバーの脆弱性は、デフォルトの URL の拒否ルールのセットを有効にすることで検出およびブロックできます。バッファオーバーフロー、SQL、クロスサイトスクリプティングなどの一般的な攻撃も簡単に検出できます。

名前が示すように、高度な保護は、より高いセキュリティ要件を持つアプリケーション向けです。リラクゼーションルールは、特定のデータのみへのアクセスを許可し、残りのデータをブロックするように構成されています。このポジティブなセキュリティモデルは、基本的なセキュリティチェックでは検出されない可能性のある未知の攻撃を軽減します。すべての基本的な保護に加えて、高度なプロファイルは、ブラウジングの制御、クッキーのチェック、さまざまなフォームフィールドの入力要件の指定、フォームの改ざんやクロスサイトリクエストフォージェリ攻撃からの保護により、ユーザーセッションを追跡します。トラフィックを観察し、適切な緩和を推奨するラーニングは、多くのセキュリティチェックでデフォルトで有効になっています。高度な保護は使いやすくなりますが、より厳しいセキュリティを提供すると同時に、より多くの処理が必要になるため、十分な検討が必要です。一部の事前セキュリティチェックでは、キャッシュの使用が許可されていないため、パフォーマンスに影響する可能性があります。

基本プロファイルと詳細プロファイルのどちらを使用するかを決定する際には、次の点に注意してください。

- 基本プロファイルと詳細プロファイルは、テンプレートを開始したばかりです。基本プロファイルは、いつでも変更して高度なセキュリティ機能を展開できます。また、その逆も可能です。
- 高度なセキュリティチェックはより多くの処理を必要とし、パフォーマンスに影響を与える可能性があります。アプリケーションに高度なセキュリティが必要でない限り、基本プロファイルから始めて、アプリケーションに必要なセキュリティを強化することができます。
- アプリケーションで必要でない限り、すべてのセキュリティチェックを有効にするとは限りません。

Q: ポリシーとは何ですか**?** バインドポイントを選択し、優先度を設定するにはどうすればよいですか

Web App Firewall ポリシーを使用すると、トラフィックを論理グループに分類して、さまざまなレベルのセキュリティ実装を構成できます。ポリシーのバインドポイントを慎重に選択して、どのトラフィックがどのポリシーと照合されるかを決定します。たとえば、すべての着信リクエストで SQL/クロスサイトスクリプティング攻撃をチェックする場合は、汎用ポリシーを作成し、グローバルにバインドできます。または、機密データを含むアプリケーションをホストする仮想サーバーのトラフィックに、より厳格なセキュリティチェックを適用する場合は、その仮想サーバーにポリシーをバインドできます。

優先順位を慎重に割り当てることで、トラフィック処理が強化されます。より具体的なポリシーには高い優先順位を割り当て、汎用ポリシーには低い優先順位を割り当てます。数値が大きいほど、プライオリティは低くなります。プライオリティ 10 のポリシーは、プライオリティ 15 のポリシーよりも先に評価されます。

異なる種類のコンテンツに対して異なるレベルのセキュリティを適用できます。たとえば、画像やテキストなどの静的オブジェクトに対するリクエストは 1 つのポリシーを使用してバイパスでき、他の機密コンテンツのリクエストは第 2 のポリシーを使用して非常に厳しいチェックを受けることができます。

Q: アプリケーションを保護するためのルールを設定するにはどうすればよいですか**?**

Web App Firewall を使用すると、Web サイトに適したレベルのセキュリティを非常に簡単に設計できます。複数の Web App Firewall ポリシーを異なる Web App Firewall プロファイルにバインドして、アプリケーションにさまざまなレベルのセキュリティチェックインスペクションを実装できます。最初にログを監視して、検出されたセキュリティ上の脅威とトリガーされた違反を確認できます。緩和ルールを手動で追加するか、Web App Firewall の推奨学習済みルールを利用して、必要な緩和をデプロイして誤検出を回避できます。

NetScaler Web App Firewall は、**GUI** でのビジュアライザのサポートを提供し、ルール管理を非常に簡単にします。すべてのデータを 1 つの画面で簡単に表示し、ワンクリックで複数のルールに対してアクションを実行できます。ビジュアライザーの最大の利点は、いくつかのルールを統合するために正規表現を推奨することです。区切り文字とアクション URL に基づいてルールのサブセットを選択できます。ビジュアライザーのサポートは、1) 学習したルールと 2) 緩和ルールの表示に使用できます。

1. 学習したルールのビジュアライザーには、ルールを編集して緩和として展開するオプションがあります。ルールをスキップ (無視) することもできます。
2. 展開された緩和のビジュアライザーには、新しいルールを追加するか、既存のルールを編集するオプションがあります。また、ノードを選択し、緩和ビジュアライザーの「有効」または「無効」ボタンをクリックして、ルールのグループを有効または無効にすることもできます。

Q: 署名とは何ですか**?** 使用するシグニチャはどのようにしてわかりますか

シグニチャは、複数のルールを持つことができるオブジェクトです。各ルールは、指定した一連のアクションに関連付けることができる 1 つ以上のパターンで構成されます。Web App Firewall には、1,300 を超えるシグニチャルールで構成されるデフォルトのシグニチャオブジェクトが組み込まれており、自動更新機能を使用して新しい脆弱性に対する保護を取得することで、最新のルールを取得するオプションがあります。他のスキャンツールで作成されたルールもインポートできます。

シグニチャは、パターンマッチングを使用して悪意のある攻撃を検出し、トランザクションのリクエストとレスポンスの両方をチェックするように設定できるため、非常に強力です。これらは、カスタマイズ可能なセキュリティソリューションが必要な場合に推奨されるオプションです。シグニチャの一致が検出されたときに、複数のアクション選択肢（ブロック、ログ、学習、変換など）を使用できます。デフォルトのシグニチャは、web-cgi、web-coldfusion、web-FrontPage、web-IIS、web-php、ウェブクライアント、web-activex、ウェブシェルショック、ウェブストラットなど、さまざまなタイプのアプリケーションを保護するためのルールをカバーしています。アプリケーションのニーズに合わせて、特定のカテゴリに属するルールを選択して展開できます。

シグネチャの使用に関するヒント:

- デフォルトのシグニチャオブジェクトのコピーを作成し、それを変更して、必要なルールを有効にし、必要なアクションを設定することができます。
- シグニチャオブジェクトは、他のシグニチャルールと連携して機能する新しいルールを追加することでカスタマイズできます。
- シグニチャルールは、Web App Firewall プロファイルで指定されたセキュリティチェックと連動するように構成することもできます。違反を示す一致がシグニチャとセキュリティチェックによって検出された場合、より制限の厳しいアクションが強制されます。
- シグニチャルールは複数のパターンを持つことができ、すべてのパターンが一致した場合にのみ違反にフラグを付けるように設定することで、誤検出を回避できます。
- ルールのリテラル高速一致パターンを慎重に選択すると、処理時間を大幅に最適化できます。

Q: Web App Firewall は、他の NetScaler 機能と連携しますか

Web App Firewall は NetScaler アプライアンスに完全に統合されており、他の機能とシームレスに動作します。Web App Firewall と組み合わせて他の NetScaler セキュリティ機能を使用して、アプリケーションの最大限のセキュリティを構成できます。たとえば、**AAA-TM** を使用して、ユーザの認証、コンテンツへのアクセス許可の確認、無効なログイン試行を含むアクセスのログを記録できます。**Rewrite** は URL の変更、ヘッダーの追加、変更、削除に使用できます。**Responder** を使用すると、カスタマイズされたコンテンツをさまざまなユーザーに配信できます。ウェブサイトの最大負荷を定義するには、レート制限を使用してトラフィックを監視し、レートが高すぎる場合はレートをスロットルします。**HTTP** サービス拒否 **(DoS)** 保護は、実際の HTTP クライアントと悪意のある DoS クライアントを区別するのに役立ちます。Web App Firewall ポリシーを仮想サーバーにバインドすることで、セキュリティチェック検査の範囲を狭めることができます。また、負荷分散機能を使用して頻繁に使用されるアプリケーションを管理することで、ユーザーエクスペリエンスを最適化できます。画像やテキストなどの静的オブジェクトに対する要求は、そのようなコンテンツの帯域幅使用を最適化するために、** 統合されたキャッシュまたは圧縮を利用して ** 、セキュリティチェック検査をバイパスできます。

Q: ペイロードは、Web App Firewall およびその他の NetScaler 機能によってどのように処理されますか?

NetScaler アプライアンスの L7 パケットフローの詳細を示す図は、［機能の処理順序 ］セクションにあります。

Q: Web App Firewall デプロイに推奨されるワークフローは何ですか?

NetScaler Web App Firewall の最先端のセキュリティ保護を使用する利点がわかったので、セキュリティニーズに最適なソリューションを設計するのに役立つ追加情報を収集できます。Citrix は、次のことを実行することをお勧めします。

- 環境を知る**:** 環境を知ることで、ニーズに最適なセキュリティ保護ソリューション (署名、セキュリティチェック、またはその両方) を特定できます。設定を開始する前に、次の情報を収集する必要があります。

  - **OS**：どんな OS（MS Windows、Linux、BSD、Unix、その他）を持っていますか?
  - **Web** サーバー**:** どのウェブサーバー (IIS、Apache、または NetScaler エンタープライズサーバー) を実行していますか。
  - アプリケーション**:** アプリケーションサーバーで実行されているアプリケーションの種類 (ASP.NET、PHP、Cold Fusion、ActiveX、FrontPage、Struts、CGI、Apache Tomcat、ドミノ、WebLogic など)
  - カスタマイズされたアプリケーションまたは既製のアプリケーション（Oracle、SAP など）がありますか。どのバージョンを使用していますか?
  - **SSL: SSL** が必要ですか? その場合、証明書の署名に使用されるキーサイズ（512、1024、2048、4096）は何ですか。
  - **Traffic Volume:** アプリケーションの平均トラフィックレートはどれくらいですか。トラフィックに季節的または時間的特有の急上りがありますか?
  - サーバーファーム**:** サーバーはいくつありますか。負荷分散を使用する必要がありますか。
  - データベース：どのタイプのデータベース（MS-SQL、MySQL、Oracle、Postgres、SQLite、nosql、Sybase、Informix など）を使用していますか?
  - **DB Connectivity:** どのような種類のデータベース接続 (DSN、ファイルごとの接続文字列、単一ファイル接続文字列) があり、どのドライバが使用されていますか?

- セキュリティニーズの特定**:** 最大限のセキュリティ保護が必要なアプリケーションまたは特定のデータ、脆弱性の低いアプリケーション、セキュリティ検査を安全にバイパスできるアプリケーションを評価できます。これは、最適な構成を考案し、トラフィックを分離するための適切なポリシーとバインドポイントを設計する際に役立ちます。たとえば、画像、MP3 ファイル、ムービーなどの静的な Web コンテンツに対する要求のセキュリティ検査をバイパスするポリシーを構成し、動的コンテンツの要求に高度なセキュリティチェックを適用する別のポリシーを構成できます。複数のポリシーとプロファイルを使用して、同じアプリケーションの異なるコンテンツを保護できます。
- ライセンス要件： NetScaler は、負荷分散、コンテンツスイッチング、キャッシュ、圧縮、レスポンダー、リライト、コンテンツフィルタリングなどの豊富な機能を活用して、アプリケーションのパフォーマンスを最適化する統合ソリューションを提供します。必要な機能を特定すると、必要なライセンスを決定するのに役立ちます。
- **NetScaler** アプライアンスのインストールとベースライン：仮想サーバーを作成し、それを介してテストトラフィックを実行して、システムを通過するトラフィックの速度と量を把握します。この情報は、キャパシティ要件を特定し、適切なアプライアンス（VPX、MPX、または SDX）を選択するのに役立ちます。
- **Web App Firewall** を展開する**:** Web App Firewall ウィザードを使用して、簡単なセキュリティ構成を続行します。ウィザードでは、いくつかの画面が表示され、プロファイル、ポリシー、署名、およびセキュリティチェックを追加するように求められます。

  - プロファイル**:** プロファイルの意味のある名前と適切なタイプ (HTML、XML、または WEB 2.0) を選択します。ポリシーとシグニチャは、同じ名前を使用して自動生成されます。
  - **Policy**：自動生成されたポリシーには、すべてのトラフィックが選択され、グローバルにバインドされるデフォルトの式（true）があります。これは、使用する特定のポリシーを念頭に置いていない限り、出発点として適しています。

- 保護**:** このウィザードでは、ハイブリッドセキュリティモデルを利用できます。このモデルでは、さまざまな種類のアプリケーションを保護するための豊富なルールセットを提供する既定のシグニチャを使用できます。簡易編集モードでは、さまざまなカテゴリ (CGI、Cold Fusion、PHP など) を表示できます。1 つ以上のカテゴリを選択して、アプリケーションに適用可能な特定のルールセットを識別できます。選択したカテゴリのすべてのシグニチャルールを有効にするには、[**Action**] オプションを使用します。セキュリティを強化する前にトラフィックを監視できるように、ブロッキングが無効になっていることを確認します。[続行] をクリックします。[詳細な保護の指定] ウィンドウで、必要に応じて変更を加えて、セキュリティチェック保護を展開できます。ほとんどの場合、初期セキュリティ設定には基本的な保護で十分です。トラフィックをしばらく実行して、セキュリティ検査データの代表的なサンプルを収集します。
- セキュリティの強化**:** Web App Firewall をデプロイしてしばらくトラフィックを監視した後、緩和をデプロイしてからブロッキングを有効にすることで、アプリケーションのセキュリティの強化を開始できます。**Learning**、**Visualizer**、**Click to Deploy** のルールは、設定を簡単に微調整して適切なレベルのリラクゼーションを思いつくことができる便利な機能です。この時点で、ポリシー式を変更したり、追加のポリシーやプロファイルを構成したりして、さまざまなタイプのコンテンツに必要なセキュリティレベルを実装することもできます。
- デバッグ**:** アプリケーションの予期しない動作が発生した場合、Web App Firewall には、デバッグを簡単にするためのさまざまなオプションが用意されています。
  - ログ。正当なリクエストがブロックされた場合は、まず ns.log ファイルをチェックして、予期しないセキュリティー検査違反がトリガーされていないかどうかを確認します。
  - 機能を無効にします。違反は見られなくても、アプリケーションがリセットしたり、部分的な応答を送信したりするなど、予期しない動作が発生する場合は、Web App Firewall のデバッグ機能を無効にできます。問題が解決しない場合は、Web App Firewall が疑わしいものとして除外されます。
  - ログメッセージを含むレコードをトレースします。問題が Web App Firewall に関連しているように見え、より詳細な検査が必要な場合は、nstrace にセキュリティ違反メッセージを含めるオプションがあります。トレースで「Follow TCP stream」を使用すると、ヘッダー、ペイロード、対応するログメッセージなど、個々のトランザクションの詳細を同じ画面に表示できます。この機能の使用方法の詳細については、付録を参照してください。

NetScaler Web App Firewall の概要

October 25, 2023

NetScaler Web App Firewall は、機密のビジネス情報や顧客情報にアクセスする Web サイトへのセキュリティ侵害、データ損失、および不正な変更を防止します。そのために、リクエストとレスポンスの両方をフィルタリングし、悪意のあるアクティビティの証拠がないか調べ、そのようなアクティビティを示すリクエストをブロックします。サイトは、一般的なタイプの攻撃だけでなく、まだ未知の新しい攻撃からも保護されています。Web App Firewall

は、Web サーバーと Web サイトを不正アクセスから保護するだけでなく、レガシー CGI コードまたはスクリプト、Web フレームワーク、Web サーバーソフトウェア、およびその他の基盤となるオペレーティングシステムの脆弱性からも保護します。

NetScaler Web App Firewall は、スタンドアロンアプライアンスとして、または NetScaler ADC 仮想アプライアンス（VPX）の機能として利用できます。Web App Firewall のドキュメントでは、NetScaler ADC という用語は、そのプラットフォームが専用のファイアウォールアプライアンス、他の機能も構成されている NetScaler ADC、または NetScaler ADC VPX であるかどうかに関係なく、Web App Firewall が実行されているプラットフォームを指します。

Web App Firewall を使用するには、保護されている Web サイトに設定したルールに違反する接続をブロックするセキュリティ構成を少なくとも 1 つ作成する必要があります。作成するセキュリティ設定の数は、Web サイトの複雑さによって異なります。場合によっては、1 つの構成で十分な場合もあります。また、特にインタラクティブな Web サイト、データベースサーバーにアクセスする Web サイト、ショッピングカートのあるオンラインストアなどのケースでは、特定の種類の攻撃に対して脆弱ではないコンテンツに多大な労力を費やすことなく、機密データを最大限に保護するために、いくつかの異なる構成が必要になる場合があります。多くの場合、すべてのセキュリティ設定に影響するグローバル設定のデフォルトはそのままにしておくことができます。ただし、構成の他の部分と競合する場合や、カスタマイズしたい場合は、グローバル設定を変更できます。

Web アプリケーションセキュリティ

Web アプリケーションセキュリティは、HTTP および HTTPS プロトコルを使用して通信するコンピューターとプログラムのネットワークセキュリティです。これは、セキュリティ上の欠陥や弱点が多く存在する幅広い分野です。サーバーとクライアントの両方のオペレーティングシステムにはセキュリティ上の問題があり、攻撃に対して脆弱です。CGI、Java、JavaScript、PERL、PHP などの Web サーバーソフトウェアや Web サイト対応テクノロジーには、根本的な脆弱性があります。Web 対応アプリケーションと通信するブラウザやその他のクライアントアプリケーションにも脆弱性があります。訪問者とのやりとりを可能にするサイトを含め、最も単純な HTML 以外のテクノロジーを使用している Web サイトには、多くの場合、独自の脆弱性があります。

以前は、セキュリティ違反は単なる煩わしさであることが多かったが、今日ではそうなることはめったにない。たとえば、ハッカーが Web サーバーにアクセスし、Web サイトに不正な変更（改ざん）を加える（改ざん）する攻撃が一般的でした。通常、他のハッカーに自分のスキルを見せたり、対象となる個人や企業を当惑させたりする以外に動機のないハッカーによって立ち上げられました。しかし、現在のセキュリティ侵害のほとんどは、金銭への欲求によって動機付けられています。大多数は、機密性が高く潜在的に貴重な個人情報を取得すること、またはウェブサイトまたはウェブサーバーへの不正アクセスや制御を得ることのいずれかまたは両方を達成しようとします。

特定の形態のウェブ攻撃は、個人情報の取得に重点を置いています。これらの攻撃は、攻撃者が完全に制御できないほど安全な Web サイトに対しても発生することがよくあります。攻撃者が Web サイトから取得できる情報には、顧客の名前、住所、電話番号、社会保障番号、クレジットカード番号、医療記録、およびその他の個人情報が含まれます。攻撃者はこの情報を利用したり、他の人に売ったりすることができます。このような攻撃によって得られる情報の多くは法律によって保護されており、そのすべては慣習と期待によって保護されています。この種の違反は、個人情報が漏洩した顧客に重大な結果をもたらす可能性があります。せいぜい、これらの顧客は、他人がクレジットカー

ドを悪用したり、自分の名前で無許可のクレジットカードを開設したり、個人情報をあからさまに流用したり（個人情報の盗難）したりしないように警戒しなければなりません。最悪の場合、顧客は信用格付けの低下に直面したり、自分が関与していない犯罪行為のせいにされたりする可能性があります。

他の *Web* 攻撃は、*Web* サイトまたは *Web* サイトが動作するサーバー、あるいはその両方を制御（または侵害）することを目的としています。ウェブサイトやサーバーの制御権を獲得したハッカーは、それを使って不正なコンテンツをホストしたり、別のウェブサーバーでホストされているコンテンツのプロキシとして機能したり、SMTP サービスを提供して迷惑な大量のメールを送信したり、侵害された他のウェブサーバーでのそのような活動をサポートするための DNS サービスを提供したりする可能性があります。侵害されたウェブサーバーでホストされているほとんどのウェブサイトは、疑わしいビジネスやあからさまな詐欺行為を助長しています。たとえば、ほとんどのフィッシング Web サイトや児童搾取 Web サイトは、侵害された Web サーバーでホストされています。

ウェブサイトやウェブサービスをこれらの攻撃から守るには、識別可能な特徴を持つ既知の攻撃をブロックすることと、未知の攻撃を防ぐことの両方が可能な多層防御が必要です。未知の攻撃は、ウェブサイトやウェブサービスへの通常のトラフィックとは異なっているために検出されることがよくあります。

セキュリティチェックの詳細については、「セキュリティチェックの概要」を参照してください。

既知のウェブ攻撃

Web サイトの第一の防衛線は、存在することが知られており、Web セキュリティの専門家によって観察および分析された多数の攻撃からの保護です。HTML ベースの Web サイトに対する一般的な攻撃には、次の種類があります。

- バッファオーバーフロー攻撃。長い URL、長い Cookie、または長い情報を Web サーバーに送信すると、システムがハングアップしたり、クラッシュしたり、基盤となるオペレーティングシステムに不正にアクセスしたりします。バッファオーバーフロー攻撃は、不正な情報にアクセスしたり、Web サーバーを危険にさらしたり、あるいはその両方を行うために利用される可能性があります。
- クッキーセキュリティ攻撃。改ざんされた Cookie をウェブサーバーに送信すること。通常、偽造された認証情報を使用して不正なコンテンツにアクセスすることを期待します。
- 強制的なブラウジング。ホームページ上のハイパーリンクの付いた URL や Web サイト上のその他の一般的な開始 URL に移動せずに、Web サイトの URL に直接アクセスする。強制ブラウジングの個々の事例は、ユーザーが Web サイトのページをブックマークしたにもかかわらず、存在しないコンテンツやユーザーが直接アクセスしてはならないコンテンツに繰り返しアクセスしようとすると、Web サイトのセキュリティに対する攻撃と見なされることがよくあります。強制ブラウジングは通常、権限のない情報にアクセスするために使用されますが、バッファオーバーフロー攻撃と組み合わせてサーバーを危険にさらすこともあります。
- **Web** フォームのセキュリティ攻撃。不適切なコンテンツを Web フォームで Web サイトに送信すること。不適切なコンテンツには、修正された隠しフィールド、HTML、または英数字データ専用フィールドのコード、短い文字列のみを入力するフィールド内の長すぎる文字列、整数のみを受け入れるフィールドの英数字文字列、Web サイトがその Web フォームで受け取ることを想定していないさまざまなデータなどがあります。Web フォームセキュリティ攻撃は、通常はバッファオーバーフロー攻撃と組み合わせて、Web サイトから不正な情報を取得したり、Web サイトを完全に侵害したりするために利用されます。

特筆すべきは、Web フォームセキュリティに対する 2 種類の特殊な攻撃です。

- **SQL** インジェクション攻撃。SQL データベースに 1 つまたは複数のコマンドを実行させることを目的として、アクティブな SQL コマンドを Web フォームで、または URL の一部として送信すること。SQL インジェクション攻撃は通常、不正な情報を取得するために使用されます。
- クロスサイトスクリプティング攻撃。Web ページの URL またはスクリプトを使用して同一生成元ポリシーに違反すること。同一生成元ポリシーでは、スクリプトが別の Web サイトのプロパティを取得したり、別の Web サイトのコンテンツを変更したりすることを禁じています。スクリプトは Web サイトの情報を取得したり、ファイルを変更したりできるため、スクリプトが別の Web サイトのコンテンツにアクセスできるようにすると、攻撃者は不正な情報を取得したり、Web サーバーを侵害したり、あるいはその両方を行ったりする可能性があります。

XML ベースの Web サービスに対する攻撃は、通常、Web サービスに不適切なコンテンツを送信しようとする試み、もう 1 つは Web サービスのセキュリティを侵害しようとする攻撃の 2 つのカテゴリのうち少なくとも 1 つに分類されます。XML ベースの Web サービスに対する一般的な攻撃には、次の種類があります。

- 悪意のあるコードまたはオブジェクト。機密情報を直接取得したり、攻撃者に Web サービスや基盤となるサーバーを制御させたりする可能性のあるコードまたはオブジェクトを含む XML リクエスト。
- **XML** リクエストの形式が正しくありません。W3C XML 仕様に準拠していないため、安全ではない Web サービスのセキュリティを侵害する可能性のある XML リクエスト
- サービス拒否 **(DoS)** 攻撃。ターゲット Web サービスに負荷をかけ、正規ユーザーの Web サービスへのアクセスを拒否する目的で、大量に繰り返し送信される XML 要求。

XML Web サービスと Web 2.0 サイトは、標準的な XML ベースの攻撃に加えて、以下に説明するように SQL インジェクションやクロスサイトスクリプティング攻撃にも脆弱です。

- **SQL** インジェクション攻撃。SQL データベースにそのコマンドを実行させる目的で、XML ベースのリクエストでアクティブな SQL コマンドまたはコマンドを送信すること。HTML SQL インジェクション攻撃と同様に、XML SQL インジェクション攻撃は通常、不正な情報を取得するために使用されます。
- クロスサイトスクリプティング攻撃。XML ベースのアプリケーションに含まれるスクリプトを使用して、同一生成元ポリシーに違反すること。同一生成元ポリシーでは、どのスクリプトも別のアプリケーションのプロパティを取得したり、コンテンツを変更したりすることはできません。スクリプトは XML アプリケーションを使用して情報を取得したりファイルを変更したりできるため、別のアプリケーションのコンテンツへのスクリプトアクセスを許可すると、攻撃者は不正な情報を取得したり、アプリケーションを危険にさらしたり、あるいはその両方を行ったりする可能性があります。

既知の Web 攻撃は、通常、Web サイトのトラフィックを特定の特性（シグネチャ）でフィルタリングすることで阻止できます。これらの特徴は、特定の攻撃では必ず出現しますが、正規のトラフィックには絶対に出てはなりません。このアプローチには、必要なリソースが比較的少なく、誤検出のリスクも比較的少ないという利点があります。そのため、ウェブサイトやウェブサービスへの攻撃に対抗し、基本的な署名保護を設定する上で貴重なツールです。

未知のウェブ攻撃

ウェブサイトやアプリケーションに対する最大の脅威は、既知の攻撃ではなく、未知の攻撃によるものです。未知の攻撃のほとんどは 2 つのカテゴリのいずれかに分類されます。1 つは、セキュリティ会社がまだ効果的な防御策を開発していない新規攻撃です（ゼロデイ攻撃）。もう 1 つは、多くの Web サイトや Web サービスではなく、特定の Web サイトまたは Web サービスを狙った攻撃（スピア攻撃）です。これらの攻撃は、既知の攻撃と同様に、機密性の高い個人情報を取得し、ウェブサイトやウェブサービスを危険にさらしてさらなる攻撃に使用できるようにすること、あるいはその両方を目的としています。

ゼロデイ攻撃はすべてのユーザーにとって大きな脅威です。これらの攻撃は通常、既知の攻撃と同じタイプです。ゼロデイ攻撃には、SQL の注入、クロスサイトスクリプト、クロスサイトリクエストフォージェリ、または既知の攻撃に類似した別の種類の攻撃が含まれることがよくあります。通常、対象となるソフトウェア、Web サイト、または Web サービスの開発者が気付いていない、または発見したことのない脆弱性を標的にします。そのため、セキュリティ企業はこれらの攻撃に対する防御策を開発しておらず、たとえ開発したとしても、ユーザーはこれらの攻撃からの保護に必要なパッチを入手してインストールしたり、回避策を実行したりしていません。ゼロデイ攻撃が発見されてから防御が可能になるまでの時間（脆弱性ウィンドウ）は短くなっていますが、攻撃者は依然として、多くのウェブサイトやウェブサービスが攻撃に対する特定の保護策を欠いているため、何時間も、あるいは何日もかかる可能性があります。

スピア攻撃は大きな脅威ですが、特定のユーザーグループにとっては脅威です。スピア攻撃の一般的なタイプであるスピアフィッシングは、特定の銀行や金融機関の顧客、または（あまり一般的ではありませんが）特定の企業や組織の従業員を標的にします。他のフィッシングは、その銀行や金融機関の実際の通信に精通しているユーザーなら認識できる、大雑把に書かれた偽造品であることが多いですが、スピアフィッシングは文字どおりで説得力があります。それらには、見知らぬ人が知っていたり、入手できたりしてはならない個人固有の情報が含まれている場合があります。したがって、スピアフィッシング詐欺師は、要求された情報を提供するように標的を説得することができます。この情報を使用して、アカウントを略奪したり、他のソースから不正に入手した金銭を処理したり、さらに機密性の高い他の情報にアクセスしたりすることができます。

どちらのタイプの攻撃にも、通常は検出できる特定の特性があります。ただし、標準のシグネチャのように特定の特性を探す静的パターンを使用することはできません。この種の攻撃を検出するには、ヒューリスティックフィルタリングやポジティブセキュリティモデルシステムなど、より高度でリソースを大量に消費するアプローチが必要です。ヒューリスティックフィルタリングは、特定のパターンではなく、行動のパターンを対象としています。ポジティブセキュリティモデルシステムは、保護対象の Web サイトまたは Web サービスの通常の動作をモデル化し、その通常の使用モデルに当てはまらない接続をブロックします。URL ベースと Web フォームベースのセキュリティチェックでは、Web サイトの通常の使用状況をプロファイリングし、ヒューリスティックとポジティブセキュリティの両方を使用して異常なトラフィックや予期しないトラフィックをブロックして、ユーザーの Web サイトとのインタラクションを制御します。ヒューリスティックなセキュリティとポジティブなセキュリティの両方を適切に設計して導入すれば、シグネチャが見逃すほとんどの攻撃をキャッチできます。ただし、これらはシグニチャよりもかなり多くのリソースを必要とするため、誤検出を防ぐにはある程度の時間をかけて適切に設定する必要があります。そのため、これらは主要な防御線としてではなく、シグネチャのバックアップやその他のリソースをあまり消費しないアプローチとして使用されます。

シグネチャに加えてこれらの高度な保護を設定することで、ハイブリッドセキュリティモデルを作成できます。これにより、Web App Firewall は既知の攻撃と未知の攻撃の両方に対する包括的な保護を提供できます。

NetScaler Web App Firewall 仕組み

Web App Firewall をインストールすると、ポリシー、プロファイル、および署名オブジェクトで構成される初期セキュリティ設定を作成します。ポリシーは、フィルタリングするトラフィックを識別するルールであり、プロファイルは、トラフィックがフィルタリングされるときに許可またはブロックする動作のパターンとタイプを識別します。シグネチャと呼ばれる最も単純なパターンは、プロファイル内ではなく、プロファイルに関連付けられたシグネチャオブジェクトで指定されます。

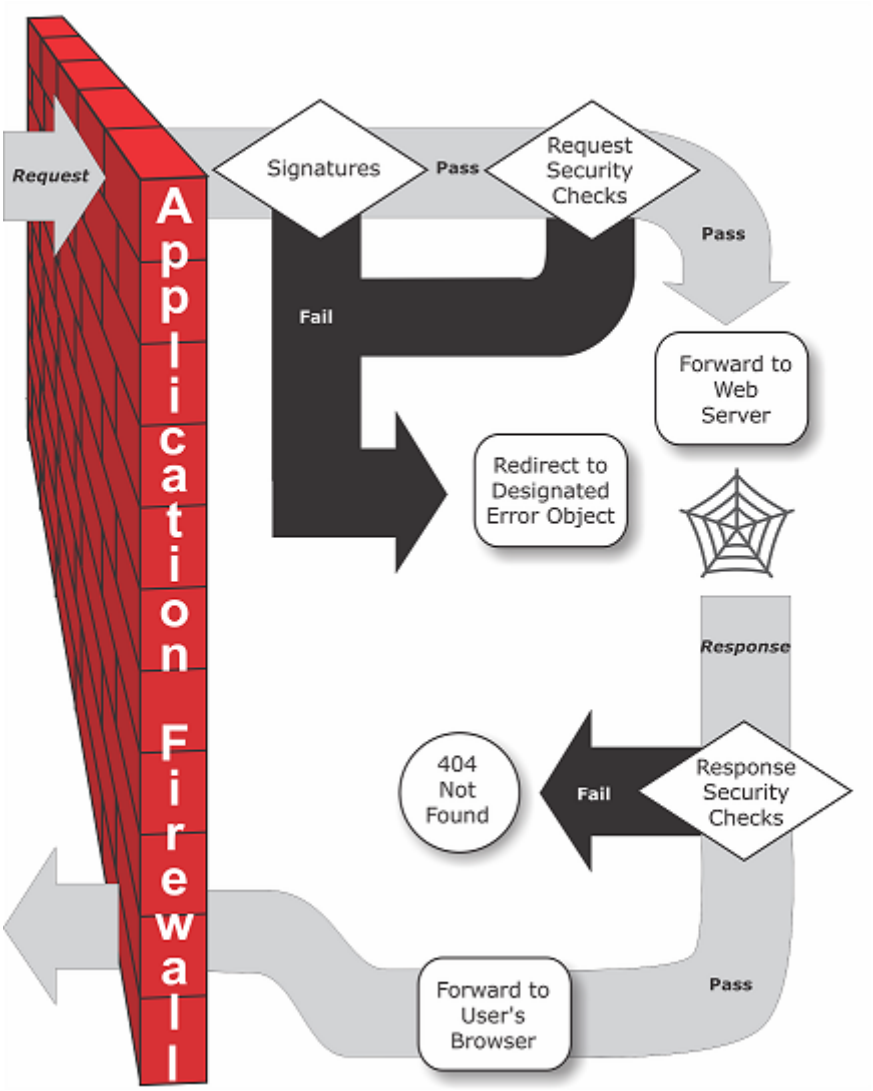
シグネチャは、既知の攻撃タイプと一致する文字列またはパターンです。Web App Firewall には 7 つのカテゴリに 1000 を超えるシグネチャが含まれており、それぞれが特定の種類のウェブサーバーや Web コンテンツへの攻撃を目的としています。NetScaler は、新しい脅威が特定されると、新しいシグネチャでリストを更新します。設定時に、保護する必要のあるウェブサーバーとコンテンツに適した署名カテゴリを指定します。シグネチャは、処理のオーバーヘッドを低く抑えながら、基本的な保護を良好に行います。アプリケーションに特別な脆弱性がある場合、またはシグネチャが存在しない攻撃を検出した場合は、独自のシグネチャを追加できます。

より高度な保護はセキュリティチェックと呼ばれます。セキュリティチェックは、保護対象の Web サイトや Web サービスに対する攻撃や脅威となる可能性のある、特定のパターンや動作の種類について、リクエストをより厳密にアルゴリズムに基づいて検査することです。たとえば、セキュリティを侵害する可能性のある特定の種類の操作を実行しようとするリクエストや、社会保障番号やクレジットカード番号などの機密性の高い個人情報を含む応答を特定できます。構成時に、保護する必要のあるウェブサーバーとコンテンツに適したセキュリティチェックを指定します。セキュリティチェックは制限が厳しいです。それらの多くは、設定時に適切な例外 (緩和) を追加しないと、正当な要求や応答をブロックする可能性があります。アダプティブラーニング機能を使えば、ウェブサイトの通常の使用状況を観察して推奨例外を作成するので、必要な例外を特定するのは難しくありません。

Web App Firewall は、レイヤー 3 ネットワークデバイス、またはサーバーとユーザー間のレイヤー 2 ネットワークブリッジとしてインストールできます。通常は会社のルーターまたはファイアウォールの背後に設置できます。保護する Web サーバーと、ユーザーがその Web サーバーにアクセスするハブまたはスイッチ間のトラフィックを傍受できる場所にインストールする必要があります。次に、リクエストをウェブサーバーに直接送信するのではなく Web App Firewall に送信し、ユーザーに直接送信するのではなく Web App Firewall に応答するようにネットワークを構成します。Web App Firewall は、内部ルールセットとユーザーによる追加や変更の両方を使用して、トラフィックを最終宛先に転送する前にフィルタリングします。有害であると検出したアクティビティをブロックまたはレンダリングし、残りのトラフィックを Web サーバに転送します。次の図は、フィルタリングプロセスの概要を示しています。

> 注:
>
> この図では、着信トラフィックへのポリシーの適用は省略されています。このスライドは、ポリシーがすべての要求を処理するセキュリティ設定を示しています。また、この設定では、シグニチャオブジェクトが設定され、プロファイルに関連付けられ、セキュリティチェックが設定されています。

図 1: Web App Firewall フィルタリングのフローチャート



図が示すように、ユーザーが保護された Web サイトの URL をリクエストすると、Web App Firewall は最初にそのリクエストを調べ、署名と一致しないことを確認します。要求が署名と一致する場合、NetScaler Web App Firewall はエラーオブジェクト（Web App Firewall アプライアンス上にあり、インポート機能を使用して構成できる Web ページ）を表示するか、指定されたエラー URL（エラーページ）に要求を転送します。署名はセキュリティチェックほど多くのリソースを必要としないため、セキュリティチェックを実行する前に署名によって検出された攻撃を検出して阻止することで、サーバーの負荷を軽減できます。

リクエストが署名検査に合格すると、Web App Firewall は有効になっているリクエストのセキュリティチェックを適用します。リクエストのセキュリティチェックは、リクエストがウェブサイトまたはウェブサービスに適しており、脅威となる可能性のある内容が含まれていないことを確認します。たとえば、セキュリティー検査では、リクエストが予期せぬタイプであるか、予期しないコンテンツをリクエストするか、予期せぬ悪意のある Web フォームデータ、SQL コマンド、またはスクリプトが含まれている可能性があることを示す標識がないか調べます。要求がセキュリティチェックに失敗した場合、Web App Firewall は要求をサニタイズしてから NetScaler ADC アプライアンス（ま

たは NetScaler ADC 仮想アプライアンス）に送り返すか、エラーオブジェクトを表示します。要求がセキュリティチェックに合格すると、NetScaler ADC アプライアンスに返送され、他の処理が完了し、保護された Web サーバーに要求が転送されます。

Web サイトまたは Web サービスがユーザーに応答を送信すると、Web App Firewall は有効になっている応答セキュリティチェックを適用します。レスポンスセキュリティチェックでは、機密性の高い個人情報の漏えい、Web サイトの改ざんの兆候、または存在してはならないその他のコンテンツがないかどうかをチェックします。応答がセキュリティチェックに失敗した場合、Web App Firewall は存在してはならないコンテンツを削除するか、応答をブロックします。応答がセキュリティチェックに合格すると、NetScaler ADC アプライアンスに返信され、ユーザーに転送されます。

NetScaler Web App Firewall 機能

Web App Firewall 基本的な機能は、ポリシー、プロファイル、シグニチャです。これらは、既知の Web 攻撃、不明な Web 攻撃、および WebApp Firewall しくみで説明されているハイブリッドセキュリティモデルを提供します。特に注意すべき点は、学習機能です。学習機能は、保護されたアプリケーションへのトラフィックを観察し、特定のセキュリティチェックに対して適切な構成設定を推奨しています。

インポート機能は、Web App Firewall にアップロードするファイルを管理します。これらのファイルは、Web App Firewall がさまざまなセキュリティチェックに使用したり、セキュリティチェックと一致する接続に応答したりするときに使用されます。

ログ、統計、およびレポート機能を使用して Web App Firewall のパフォーマンスを評価し、保護を強化する必要があるかどうかを特定できます。

NetScaler Web App Firewall ケーショントラフィックを変更する方法

NetScaler Web App Firewall は、以下を変更することにより、保護する Web アプリケーションの動作に影響を与えます。

- Cookies
- HTTP ヘッダー
- フォーム/データ

NetScaler Web App Firewall セッション **Cookie**

セッションの状態を維持するために、NetScaler Web App Firewall は独自のセッション Cookie を生成します。この Cookie は、Web ブラウザと NetScaler ADC Web アプリケーションファイアウォールの間でのみ渡され、Web サーバーには渡されません。ハッカーがセッション Cookie を変更しようとすると、Web App Firewall は要求をサーバーに転送する前に Cookie をドロップし、要求を新しいユーザーセッションとして扱います。セッション Cookie は、Web ブラウザが開いている限り存在します。Web ブラウザを閉じると、アプリケーションファイアウ

ォールのセッション Cookie は無効になります。セッションの状態には、クライアントがアクセスした URL とフォームの情報が保持されます。

設定可能な Web App Firewall セッション Cookie は`citrix_ns_id`です。

NetScaler ビルド 12.1 54 および 13.0 以降では、Cookie `citrix_ns_id`の整合性はセッションレスになり、アプライアンスによって生成されたセッション Cookie の追加は強制されません。Cookie の設定について詳しくは、「エンジン設定」を参照してください。

NetScaler Web App Firewall クッキー　多くのウェブアプリケーションは、ユーザーまたはセッション固有の情報を追跡するためにクッキーを生成します。この情報には、ユーザー設定やショッピングカートのアイテムなどがあります。Web アプリケーションの Cookie には、次の 2 つのタイプがあります。

- パーシステントクッキー -これらのクッキーはコンピューター上にローカルに保存され、次回サイトにアクセスしたときに再び使用されます。このタイプの Cookie には通常、ログオン、パスワード、設定などのユーザーに関する情報が含まれています。
- セッションクッキーまたはトランジェントクッキー -これらのクッキーはセッション中にのみ使用され、セッションが終了すると破棄されます。このタイプの Cookie には、ショッピングカートのアイテムやセッション認証情報などのアプリケーション状態情報が含まれます。

ハッカーは、アプリケーションの Cookie を変更したり盗んだりして、ユーザーセッションを乗っ取ったり、ユーザーになりすましたりする可能性があります。アプリケーションファイアウォールは、アプリケーションクッキーをハッシュし、デジタル署名付きのクッキーをさらに追加することで、このような試みを防ぎます。アプリケーションファイアウォールは、クッキーを追跡することで、クライアントブラウザとアプリケーションファイアウォールの間でクッキーが変更されたり、危険にさらされたりしないようにします。アプリケーションファイアウォールはアプリケーションクッキーを変更しません。

NetScaler Web App Firewall は、アプリケーションクッキーを追跡するために以下のデフォルトクッキーを生成します。

- パーシスタントクッキー: `citrix_ns_id_wlf`. 注意:wlf は永遠に生き続けるという意味です。
- セッションクッキーまたはトランジェントクッキー: `citrix_ns_id_wat`. 注: の略は一時的に動作します。
  アプリケーションクッキーを追跡するために、アプリケーションファイアウォールはパーシステントまたはセッションアプリケーションクッキーをグループ化し、すべてのクッキーをハッシュして署名します。したがって、アプリケーションファイアウォールは、すべての永続アプリケーション`wlf` Cookie を追跡する Cookie と、すべてのアプリケーションセッション`wat` Cookie を追跡する Cookie を 1 つ生成します。

次の表は、ウェブアプリケーションによって生成された Cookie に基づいてアプリケーションファイアウォールによって生成される Cookie の数と種類を示しています。

NetScaler Web App Firewall 前	変更後は以下の通り
1 つのパーシステント Cookie	パーシスタント Cookie: `citix_ns_id_wlf`
1 つのトランジェント Cookie	トランジェント Cookie: `citix_ns_id_wat`
複数のパーシステントクッキー、複数のトランジェントクッキー	1 つの永続 Cookie: `citrix_ns_id_wlf`、1 つの一時 Cookie: `citix_ns_id_wat`

NetScaler Web App Firewall では、アプリケーション Cookie を暗号化できます。アプリケーションファイアウォールには、アプリケーションから送信されたセッション Cookie を残りのアプリケーションファイアウォールのセッションデータと一緒に保存し、クライアントには送信しないことでプロキシするオプションもあります。クライアントがアプリケーションファイアウォールのセッション Cookie を含むリクエストをアプリケーションに送信すると、アプリケーションファイアウォールは、元のアプリケーションにリクエストを送信する前に、アプリケーションから送信された Cookie をリクエストに挿入し直します。アプリケーションファイアウォールでは、HttpOnly フラグや Secure フラグを Cookie に追加することもできます。

アプリケーションファイアウォールが **HTTP** ヘッダーに与える影響

HTTPS リクエストと HTTPS レスポンスはどちらも、ヘッダーを使用して 1 つ以上の HTTPS メッセージに関する情報を送信します。ヘッダーは一連の行で、各行には名前の後にコロンとスペース、値が続きます。たとえば、Host ヘッダーの形式は次のとおりです。

```
Host: www.citrix.com
```

ヘッダーフィールドには、リクエストヘッダーとレスポンスヘッダーの両方で使用されるものもあれば、リクエストまたはレスポンスにのみ適しているものもあります。アプリケーションファイアウォールは、アプリケーションのセキュリティを維持するために、1 つ以上の HTTPS 要求または応答の一部のヘッダーを追加、変更、または削除する場合があります。

NetScaler Web App Firewall によってドロップされたリクエストヘッダー　キャッシュに関連するリクエストヘッダーの多くは、セッションのコンテキスト内のすべてのリクエストを表示するためにドロップされます。同様に、リクエストに Web サーバーが圧縮された応答を送信できるようにするエンコーディングヘッダーが含まれている場合、アプリケーションファイアウォールはこのヘッダーを削除します。これにより、非圧縮サーバー応答の内容が Web App Firewall によって検査され、機密データがクライアントに漏洩するのを防ぎます。

アプリケーションファイアウォールは、次のリクエストヘッダーを削除します。

- 範囲—失敗したファイル転送または部分的なファイル転送からの回復に使用されます。
- If-Range —キャッシュにすでにオブジェクトの一部が含まれている場合に、クライアントがオブジェクトの一部を取得できるようにします (条件付き GET)。
- If-Modified-Since —このフィールドに指定された時間以降に要求されたオブジェクトが変更されていない場合、エンティティはサーバーから返されません。HTTP 304 変更されていないというエラーが表示されます。

- If-None-Match —最小限のオーバーヘッドで、キャッシュされた情報を効率的に更新できます。
- Accept-Encoding —特定のオブジェクトに使用できるエンコード方法 (gzip など)。

NetScaler Web App Firewall によって変更されたリクエストヘッダー　Web ブラウザが HTTP/1.0 以前のプロトコルを使用している場合、ブラウザは各応答を受信した後も TCP ソケット接続を継続的に開いたり閉じたりします。これにより、Web サーバーのオーバーヘッドが増え、セッション状態を維持できなくなります。HTTP/1.1 プロトコルでは、セッション中も接続を開いたままにできます。アプリケーションファイアウォールは、Web ブラウザが使用するプロトコルに関係なく、アプリケーションファイアウォールと Web サーバー間で HTTP/1.1 を使用するように次のリクエストヘッダーを変更します。
接続:keep-alive

NetScaler Web App Firewall によって追加されたリクエストヘッダー　アプリケーションファイアウォールはリバースプロキシとして機能し、セッションの元の送信元 IP アドレスをアプリケーションファイアウォールの IP アドレスに置き換えます。したがって、Web サーバーのログに記録されるすべての要求は、要求がアプリケーションファイアウォールから送信されたことを示しています。

NetScaler Web App Firewall によってドロップされた応答ヘッダー　アプリケーションファイアウォールは、クレジットカード番号の削除やコメントの削除などのコンテンツをブロックまたは変更する場合があり、その結果、サイズの不一致が生じる可能性があります。このようなシナリオを防ぐために、アプリケーションファイアウォールは次のヘッダーを削除します。

Content-Length —受信者に送信されるメッセージのサイズを示します。
アプリケーションファイアウォールによって変更されたレスポンスヘッダー

アプリケーションファイアウォールによって変更される応答ヘッダーの多くは、キャッシュに関連しています。HTTP (S) 応答のキャッシュヘッダーを変更して、Web ブラウザーがローカルキャッシュを使用せずに、常に最新データのリクエストを Web サーバーに送信するように強制する必要があります。ただし、ASP アプリケーションによっては、個別のプラグインを使用して動的コンテンツを表示するため、データを一時的にブラウザにキャッシュする機能が必要な場合があります。FFC、URL クロージャー、CSRF チェックなどの高度なセキュリティ保護が有効になっているときにデータを一時的にキャッシュできるように、アプリケーションファイアウォールは次のロジックを使用してサーバー応答のキャッシュ制御ヘッダーを追加または変更します。

- サーバーが Pragma: no-cache を送信した場合、アプリケーションファイアウォールは変更を行いません。
- クライアントリクエストが HTTP 1.0 の場合、アプリケーションファイアウォールは Pragma: no-cache を挿入します。
- クライアントリクエストが HTTP 1.1 で、Cache-Control: no-store が設定されている場合、アプリケーションファイアウォールは変更を行いません。
- クライアントリクエストが HTTP 1.1 で、サーバーレスポンスの Cache-Control ヘッダーにストアもキャッシュディレクティブもない場合、アプリケーションファイアウォールは変更を行いません。

- クライアント要求が HTTP 1.1 で、サーバーレスポンスに「キャッシュ制御ヘッダーなし」または「キャッシュ制御ヘッダーにストアなし」または「キャッシュなし」ディレクティブがない場合、アプリケーションファイアウォールは次のタスクを実行します。

1. キャッシュコントロールを挿入します。max-age=3、再検証が必要、非公開です。
2. X-Cache-Control-Orig = キャッシュコントロールヘッダーの元の値を挿入します。
3. 最終更新ヘッダーを削除します。
4. Etag の代わりになります。
5. サーバーから送信された有効期限ヘッダーの X-Expires-Orig= オリジナル値を挿入します。
6. Expires Header を変更し、Web ページの有効期限を過去に設定して、常に再度取得されるようにします。
7. Accept-Ranges を変更し、それを「なし」に設定します。

アプリケーションファイアウォールが StripComments、X-out/Remove SafeObject、xout、クレジットカードまたは URL 変換の削除などの応答を変更したときに、クライアントブラウザに一時的にキャッシュされたデータを置き換えるために、アプリケーションファイアウォールは次のアクションを実行します。

1. クライアントに転送する前に、Last-Modified をサーバーから削除します。
2. Etag をアプリケーションファイアウォールによって決定された値に置き換えます。

NetScaler Web App Firewall によって追加されたレスポンスヘッダー

- `Transfer-Encoding`: チャンク。このヘッダーは、応答を送信する前に応答の全長がわからなくても、情報をクライアントにストリーミングします。content-length ヘッダーが削除されているため、このヘッダーは必須です。
- `Set-Cookie`: アプリケーションファイアウォールによって追加された Cookie。
- `Xet-Cookie`: セッションが有効で、キャッシュ内の応答の有効期限が切れていない場合は、キャッシュから配信でき、セッションはまだ有効であるため、新しい Cookie を送信する必要はありません。このようなシナリオでは、セットクッキーは Xet-Cookie に変更されます。Web ブラウザ用。

フォームデータへの影響

アプリケーションファイアウォールは、サーバーから送信された元のフォームの内容を変更しようとする攻撃から保護します。また、クロスサイトリクエスト偽造攻撃からも保護できます。アプリケーションファイアウォールは、隠しフォームタグ as_fid をページに挿入することで実行されます。

例: `<input type="hidden"name="as_fid"value="VRgWq0I196Jmg/+LOY7C"/>`

隠しフィールド as_fid はフィールドの一貫性を保つために使用されます。Application Firewall はこのフィールドを使用して、隠しフィールドの名前と値のペアを含むフォームのすべてのフィールドを追跡し、サーバーから送信されたフォームのフィールドがクライアント側で変更されないようにします。また、CSRF チェックでは、この独自のフォームタグ as_fid を使用して、ユーザーが送信したフォームがこのセッションでユーザーに提供され、ハッカーがユーザーセッションを乗っ取ろうとしていないことを確認します。

セッションレスフォームチェック　アプリケーションファイアウォールには、セッションレスフィールドの一貫性を使用してフォームデータを保護するオプションもあります。これは、フォームに動的な隠しフィールドが多数あり、アプリケーションファイアウォールによるセッションあたりのメモリ割り当て量が多くなる可能性があるアプリケーションに役立ちます。セッションレスフィールドの一貫性チェックは、構成された設定に基づいて、POST リクエストのみ、または GET リクエストと POST リクエストの両方に別の隠しフィールド as_ffc_field を挿入することによって実行されます。アプリケーションファイアウォールは、フォームをクライアントに転送するときに GET メソッドを POST に変更します。その後、アプライアンスはメソッドをサーバーに送信するときに、メソッドを GET に戻します。as_ffc_field の値は、送信されるフォームの暗号化されたダイジェストを含むため、大きくなる可能性があります。以下は、セッションレスフォームチェックの例です。

```
<input type="hidden" name="as_ffc_field" value="CwAAAAVIGLD/
    luRRi1Wu1rbYrFYargEDcO5xVAxsEnMP1megXuQfiDTGbwk0fpgndMHqfMbzfAFdjwR+
    TOm1oT
+u+Svo9+NuloPhtnbkxGtNe7gB/o8GlxEcK9ZkIIVv3oIL/
    nIPSRWJljgpWgafzVx7wtugNwnn8/
    GdnhneLCJTaYU7ScnC6LexJDLisI1xsEeONWt8Zm
+vJTa3mTebDY6LVyhDpDQfBgI1XLgfLTexAUzSNWHYyloqPruGYfnRPw+
    DIGf6gGwn1BYLEsRHKNbjJBrKpOJo9JzhEqdtZ1g3bMzEF9PocPvM1Hpvi5T6VB
/YFunUFM4f+bD7EAVcugdhovzb71CsSQX5+qcC1B8WjQ==" />
<!--NeedCopy-->
```

HTML コメントストリッピング　アプリケーションファイアウォールには、応答に含まれるすべての HTML コメントをクライアントに送信する前に削除するオプションもあります。これはフォームだけでなく、すべての回答ページに影響します。アプリケーションファイアウォールは、「<!」の間に埋め込まれたテキストを見つけて削除します。-」と「->」コメントタグ。タグはそのままで、HTML ソースコードのその場所にコメントが存在したことを示すものです。他の HTML または JavaScript タグに埋め込まれたテキストは無視されます。
一部のアプリケーションは、コメントタグに JavaScript が誤って埋め込まれていると、正しく動作しないことがあります。Application Firewall によってコメントが削除される前と後のページソースコードを比較すると、削除されたコメントに必要な JavaScript が埋め込まれているかどうかを確認するのに役立ちます。

クレジットカード保護　アプリケーションファイアウォールには、応答のヘッダーと本文を検査し、応答をクライアントに転送する前にクレジットカード番号を削除または消去するオプションがあります。現在、アプリケーションファイアウォールは、アメリカンエキスプレス、ダイナースクラブ、ディスカバー、JCB、マスターカード、Visa などの主要なクレジットカードを保護しています。x-out アクションは Block アクションとは独立して動作します。

安全なオブジェクト保護　クレジットカード番号と同様に、Application Firewall Safe Object セキュリティチェックを使用して応答内の機密コンテンツを削除または消去することで、他の機密データの漏洩を防ぐことができます。

クロスサイトスクリプティングはアクションを変える　トランスフォームでクロスサイトスクリプティングを有効にすると、Web App Firewall `"<"into "%26lt;"and ">"into "%26gt;"` のリクエストが変更され

ます。Web App Firewall の CheckRequestHeaders 設定が有効になっている場合、Web App Firewall はリクエストヘッダーを検査し、ヘッダーと Cookie 内のこれらの文字も変換します。変換アクションは、サーバーから最初に送信された値をブロックまたは変換しません。クロスサイトスクリプティングには、Web App Firewall で許可されているデフォルトの属性とタグのセットがあります。拒否されたクロスサイトスクリプティングパターンのデフォルトリストも提供されています。これらは、シグネチャオブジェクトを選択し、GUI の「**SQL/**クロスサイトスクリプティングパターンの管理」ダイアログをクリックすることでカスタマイズできます。

SQL 特殊文字の変換　アプリケーションファイアウォールには、SQL 特殊文字に関する次のデフォルト変換ルールがあります。

ライセンス	変更後は以下の通り	トランスフォーメーション
’(一重引用符、つまり%27)	”	もう一つの一重引用符
\ (%5C のバックスラッシュ)	| 別のバックスラッシュが追加されました	
; (%3B のセミコロン)		落下しました

特殊文字の変換が有効で、checkRequestHeaders が ON に設定されている場合、特殊文字の変換はヘッダーとクッキーでも行われます。
注:User-Agent、Accept-Encoding などの一部のリクエストヘッダーには通常セミコロンが含まれており、SQL 変換の影響を受ける可能性があります。

Expect ヘッダーが破損する **NetScaler Web App Firewall** 動作

1. NetScaler が EXPECT ヘッダーを含む HTTP リクエストを受信するたびに、NetScaler はバックエンドサーバーに代わって EXPECT: 100-continue レスポンスをクライアントに送信します。
2. この動作は、要求をサーバーに転送する前に要求全体に対してアプリケーションファイアウォール保護を実行する必要があり、NetScaler は要求全体をクライアントから取得する必要があるためです。
3. `100 continue` 応答を受け取ると、クライアントはリクエストの残りの部分を送信してリクエストを完了します。
4. その後、NetScaler はすべての保護を実行し、要求をサーバーに転送します。
5. NetScaler がリクエスト全体を転送すると、最初のリクエストで受信した EXPECT ヘッダーは古くなり、NetScaler はこのヘッダーを破損してサーバーに送信します。
6. 要求を受信したサーバーは、破損しているヘッダーを無視します。

Web App Firewall 構成

August 15, 2023

NetScaler Web アプリファイアウォール（Web App Firewall）は、次のいずれかの方法を使用して構成できます。

- **Web App Firewall** ウィザード。設定プロセスを順を追って説明する一連の画面で構成されるダイアログボックス。
- **NetScaler Web** インターフェイス **AppExpert** テンプレート Web サイトを適切に保護するように設計された AppExpert テンプレート（構成設定のセット）。この AppExpert テンプレートには、多くの Web サイトを保護するための適切な Web App Firewall 構成設定が含まれています。
- **NetScaler GUI**。Web ベースの設定インターフェイス。
- **NetScaler** コマンドラインインターフェイス。コマンドライン設定インターフェイス。

Citrix では、Web App Firewall ウィザードを使用することをお勧めします。ほとんどのユーザーは、これが Web App Firewall を構成する最も簡単な方法であり、間違いを防ぐように設計されています。主に Web サイトの保護に使用する新しい NetScaler または VPX がある場合は、Web App Firewall だけでなくアプライアンス全体に適したデフォルト構成を提供するため、Web Interface AppExpert テンプレートの方が適している場合があります。GUI とコマンド・ライン・インタフェースはいずれも、経験豊かなユーザーを対象としています。主に、既存の構成を変更したり、詳細オプションを使用したりします。

Web App Firewall ウィザード

Web App Firewall ウィザードは、簡単な構成の各部分を設定するように求める複数の画面で構成されるダイアログボックスです。次に、Web App Firewall は、提供された情報から適切な構成要素を作成します。これは、Web App Firewall を設定する最も単純で、ほとんどの目的に最適な方法です。

ウィザードを使用するには、任意のブラウザで GUI に接続します。接続が確立されたら、Web App Firewall が有効になっていることを確認し、Web App Firewall ウィザードを実行して、構成情報の入力を求めます。ウィザードを初めて使用するときに、要求された情報をすべて入力する必要はありません。代わりに、デフォルト設定をそのまま使用し、比較的簡単な構成タスクをいくつか実行して重要な機能を有効にしてから、Web App Firewall が構成を完了するのに役立つ重要な情報を収集できるようにすることができます。

たとえば、処理するトラフィックを選択するルールを指定するようにウィザードから求められたら、デフォルトをそのまま使用してすべてのトラフィックを選択できます。シグニチャのリストが表示されたら、シグニチャの適切なカテゴリを有効にして、それらのシグニチャの統計情報の収集を有効にできます。この初期設定では、高度な保護 (セキュリティチェック) をスキップできます。ウィザードは、適切なポリシー、シグニチャオブジェクト、およびプロファイル (まとめてセキュリティ設定) を自動的に作成し、ポリシーをグローバルにバインドします。次に、Web App Firewall は、保護された Web サイトへの接続のフィルタリングを開始し、有効にした 1 つ以上のシグニチャと一致する接続をログに記録し、各シグニチャが一致する接続に関する統計を収集します。Web App Firewall が一部のトラフィックを処理したら、ウィザードを再度実行してログと統計を調べて、有効にしたシグニチャのいずれかが正当なトラフィックと一致するかどうかを確認できます。ブロックするトラフィックを識別しているシグニチャを特定したら、それらのシグニチャのブロックを有効にできます。ウェブサイトやウェブサービスが複雑でなく、SQL を使用せず、機密性の高い個人情報にアクセスできない場合は、この基本的なセキュリティ設定で十分な保護が得られるでしょう。

たとえば、ウェブサイトが動的な場合は、追加の保護が必要になる場合があります。スクリプトを使用するコンテンツには、クロスサイトスクリプティング攻撃に対する保護が必要な場合があります。ショッピングカート、多くのブログ、ほとんどのコンテンツ管理システムなど、SQL を使用する Web コンテンツには、SQL インジェクション攻撃に対する保護が必要な場合があります。社会保障番号やクレジットカード番号などの機密個人情報を収集するウェブサイトやウェブサービスでは、その情報が意図せずに漏洩するのを防ぐ必要がある場合があります。特定の種類の Web サーバーまたは XML サーバーソフトウェアには、そのソフトウェアに合わせた種類の攻撃からの保護が必要な場合があります。もう 1 つの考慮事項として、Web サイトまたは Web サービスの特定の要素には、他の要素とは異なる保護が必要になることがあります。Web App Firewall のログと統計を調べると、必要になる可能性のある追加の保護策を特定するのに役立ちます。

Web サイトと Web サービスに必要な高度な保護を決定したら、ウィザードを再度実行してそれらの保護を設定できます。特定のセキュリティチェックでは、チェックによって正当なトラフィックがブロックされないように、例外 (緩和) を入力する必要があります。手動で行うこともできますが、通常はアダプティブラーニング機能を有効にして、必要な緩和を推奨するほうが簡単です。ウィザードは必要なだけ何度でも使用して、基本的なセキュリティ構成を強化したり、追加のセキュリティ構成を作成したりできます。

このウィザードは、ウィザードを使用しなかった場合に手動で実行する必要のある一部のタスクを自動化します。ポリシー、シグニチャオブジェクト、プロファイルが自動的に作成され、設定名の入力を求められたときに指定した名前が割り当てられます。また、ウィザードは詳細な保護設定をプロファイルに追加し、シグニチャオブジェクトをプロファイルにバインドし、プロファイルをポリシーに関連付け、ポリシーをグローバルにバインドして有効にします。

ウィザードでは実行できないタスクがいくつかあります。ウィザードを使用してポリシーをグローバル以外のバインドポイントにバインドすることはできません。プロファイルを構成の特定の部分のみに適用する場合は、バインディングを手動で設定する必要があります。ウィザードでは、エンジン設定やその他の特定のグローバル設定オプションを設定することはできません。ウィザードでは保護の詳細設定を構成できますが、1 回のセキュリティチェックで特定の設定を変更する場合は、GUI の手動構成画面で変更する方が簡単です。

Web App Firewall ウィザードの使用方法の詳細については、「Web App Firewall ウィザード」を参照してください。

Citrix Web インターフェイス AppExpert テンプレート

AppExpert テンプレートは、複雑なエンタープライズアプリケーションを構成および管理するための、これまでとは異なるシンプルなアプローチです。GUI の AppExpert ディスプレイはテーブルで構成されています。アプリケーションは左端の列に表示され、そのアプリケーションに適用できる NetScaler 機能がそれぞれ右側の列に表示されます。(AppExpert インターフェイスでは、アプリケーションに関連する機能をアプリケーションユニットと呼びます)。AppExpert インターフェイスでは、各機能を個別に設定しなくても、アプリケーションごとに対象トラフィックを設定し、圧縮、キャッシュ、書き換え、フィルタリング、レスポンダー、Web App Firewall のルールを有効にします。

Web インターフェイス AppExpert テンプレートには、次の Web App Firewall シグネチャとセキュリティチェックのルールが含まれています。

- **URL** チェックを拒否します。セキュリティ上のリスクがあることがわかっているコンテンツ、または指定した他の URL への接続を検出します。
- バッファオーバーフローチェック。保護された Web サーバでバッファオーバーフローを引き起こす試みを検出します。
- クッキーの一貫性チェック。保護された Web サイトによって設定された Cookie への悪意のある変更を検出します。
- フォームフィールドの一貫性チェック。保護された Web サイト上の Web フォームの構造への変更を検出します。
- **CSRF** フォームタグチェックです。クロスサイトリクエストフォージェリ攻撃を検出します。
- **[**フィールド形式**]** チェック。保護された Web サイトの Web フォームにアップロードされた不適切な情報を検出します。
- **HTML SQL** インジェクションチェック。不正な SQL コードを挿入する試みを検出します。
- **HTML** クロスサイトスクリプティングチェック。クロスサイトスクリプティング攻撃を検出します。

AppExpert テンプレートのインストールと使用の詳細については、AppExpert アプリケーションとテンプレートを参照してください。

GUI

GUI は Web ベースのインターフェイスで、Web App Firewall 機能のすべての設定オプションにアクセスできます。これには、他の設定ツールやインターフェイスでは利用できない高度な設定や管理オプションも含まれます。具体的には、多くの高度な署名オプションは GUI でのみ設定できます。学習機能によって生成された推奨事項は GUI でのみ確認できます。GUI でのみグローバル以外のバインドポイントにポリシーをバインドできます。

GUI の説明については、「Web App Firewall 構成インターフェイス」を参照してください。GUI を使用して Web App Firewall を構成する方法の詳細については、GUI を使用した手動設定を参照してください。

GUI を使用して Web App Firewall を設定する手順については、GUI を使用した手動設定を参照してください。citrix-adc GUI の詳細については、「Web App Firewall 構成インターフェイス」を参照してください。

NetScaler コマンドラインインターフェイス

NetScaler コマンドラインインターフェイスは、FreeBSD bash シェルをベースにした修正された UNIX シェルです。コマンドラインインターフェイスから Web App Firewall を設定するには、他の Unix シェルと同様に、プロンプトでコマンドを入力して Enter キーを押します。Web App Firewall のほとんどのパラメーターとオプションは、NetScaler コマンドラインを使用して構成できます。例外はシグニチャ機能であり、その多くのオプションは GUI または Web App Firewall ウィザードを使用してのみ設定でき、その推奨事項は GUI でのみ確認できる学習機能です。

NetScaler ADC コマンドラインを使用して Web App Firewall を構成する方法については、「コマンドラインインターフェイスを使用した手動構成」を参照してください。

NetScaler Web App Firewall を有効にする

March 20, 2024

セキュリティ構成を作成する前に、アプライアンスで NetScaler Web App Firewall 機能を有効にする必要があります。

確認事項

- 専用の NetScaler Web App Firewall アプライアンスを構成する場合、または既存のアプライアンスをアップグレードする場合、この機能はすでに有効になっています。ここに記載されている手順のいずれも実行する必要はありません。
- 新しい NetScaler または VPX をお持ちの場合は、NetScaler Web App Firewall 機能を構成する前に有効にする必要があります。
- NetScaler または VPX を以前のバージョンからアップグレードする場合は、NetScaler Web App Firewall 機能を構成する前に、まず有効にする必要があります。

> 注:
>
> NetScaler または VPX を以前のバージョンからアップグレードする場合、NetScaler Web App Firewall を有効にする前にアプライアンスのライセンスを更新する必要がある場合があります。NetScaler の担当者または再販業者に問い合わせて、正しいライセンスを取得してください。

コマンドインターフェイスを使用して NetScaler Web App Firewall を有効にする

コマンドプロンプトで、次のコマンドを入力します:

```
enable ns feature AppFW
```

GUI を使用して Web App Firewall を有効にする

1. [システム] > [設定] に移動します。
2. 詳細ウィンドウで、「拡張機能の設定」をクリックします。
3. 「拡張機能の設定」ページで、「**NetScaler Web App**Firewall」を選択します。
4. [**OK**] をクリックします。

Web App Firewall ウィザード

August 15, 2023

ほとんどのウィザードとは異なり、NetScaler Web App Firewall Wizard は初期構成プロセスを簡略化するだけでなく、以前に作成した構成を変更したり、Web App Firewall の設定を維持したりできるように設計されています。一般的なユーザーは、ウィザードを複数回実行し、毎回一部の画面をスキップします。

Web App Firewall ウィザードは、プロファイル、ポリシー、および署名を自動的に作成します。

ウィザードを開く

Web App Firewall ウィザードを実行するには、GUI を開いて次の手順に従います。

1. [セキュリティ] > [アプリケーションファイアウォール] に移動します。
2. 詳細ペインの [はじめに] で、[アプリケーションファイアウォールウィザード] をクリックします。ウィザードが開きます。

GUI の詳細については、「Web App Firewall 設定インターフェイス」を参照してください。

ウィザードの画面

Web App Firewall ウィザードでは、次の画面が表形式で表示されます。

1. 名前を指定**:** この画面では、新しいセキュリティ設定を作成するときに、わかりやすい名前と適切なタイプ (HTML、XML、または WEB 2.0) をプロファイルに指定します。デフォルトのポリシーと署名は、同じ名前を使用して自動生成されます。

[プロファイル名]

名前は文字、数字、または下線記号で始めることができ、1 ～31 文字の数字、およびハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) で構成されます。新しいセキュリティ設定が保護するコンテンツが他のユーザーにわかりやすい名前を付けてください。

> 注:
>
> ウィザードはポリシーとプロファイルの両方にこの名前を使用するため、31 文字に制限されています。手動で作成したポリシーには、最大 127 文字の名前を付けることができます。

既存の構成を変更する場合は、「既存の構成を変更」を選択し、「名前」ドロップダウンリストで、変更する既存の構成の名前を選択します。

> 注:
>
> このリストには、グローバルまたはバインドポイントにバインドされているポリシーのみが表示されます。バインドされていないポリシーをアプリケーションファイアウォールウィザードを使用して変更することはできません。グローバルまたはバインドポイントに手動でバインドするか、手動で変更する必要があります。(手動で変更する場合は、GUI で) [アプリケーションファイアウォール] > [ポリシー] > [ファイアウォール] ペインでポリシーを選択し、[開く] をクリックします。

プロファイルの種類

また、この画面でプロファイルタイプを選択します。プロファイルタイプによって、設定できる高度な保護 (セキュリティチェック) の種類が決まります。特定の種類のコンテンツは特定の種類のセキュリティ脅威に対して脆弱ではないため、利用可能なチェックのリストを制限することで、設定時の時間を節約できます。Web App Firewall プロファイルの種類は次のとおりです。

- ウェブアプリケーション (HTML)。XML または Web 2.0 テクノロジーを使用しない HTML ベースの Web サイト。
- XML アプリケーション (XML、SOAP)。任意の XML ベースの Web サービス。
- ウェブ 2.0 アプリケーション (HTML、XML、REST)。Atom ベースのサイト、ブログ、RSS フィード、Wiki など、HTML と XML ベースのコンテンツを組み合わせた任意の Web 2.0 サイト。

注**:** Web サイトで使用されているコンテンツの種類が不明な場合は、Web 2.0 アプリケーションを選択して、あらゆる種類の Web アプリケーションコンテンツを確実に保護できます。

2. ルールを指定**:** この画面では、現在の構成で検査するトラフィックを定義するポリシールール (式) を指定します。ウェブサイトとウェブサービスを保護するための初期設定を作成する場合、デフォルト値の **true** をそのまま使用すると、すべての Web トラフィックが選択されます。

このセキュリティ設定で、アプライアンスを経由するすべての HTTP トラフィックではなく、特定のトラフィックを調べたい場合は、検査するトラフィックを指定するポリシールールを記述できます。ルールは、完全に機能するオブジェクト指向プログラミング言語である NetScaler ADC 表現言語で記述されています。

注: デフォルトの式の構文に加えて、下位互換性のために、NetScaler ADC オペレーティングシステムは、NetScaler Classic、nCore アプライアンスと仮想アプライアンスでの NetScaler ADC クラシック式構文をサポートしています。クラシック式は、NetScaler Cluster アプライアンスおよび仮想アプライアンスではサポートされていません。既存の構成を NetScaler ADC クラスターに移行する現在のユーザーは、従来の式を含むポリシーをデフォルトの式の構文に移行する必要があります。

- NetScaler ADC 式の構文を使用して Web App Firewall ルールを作成する簡単な説明と便利なルールのリストについては、「ファイアウォールポリシー」を参照してください。
- NetScaler ADC 式の構文でポリシールールを作成する方法の詳細については、「ポリシーと式」を参照してください。

4. 署名の選択: この画面では、ウェブサイトやウェブサービスを保護するために使用する署名のカテゴリを選択します。

これは必須の手順ではありません。必要であればスキップして、「ディーププロテクションの指定」画面に移動してください。Select Signatures 画面をスキップすると、プロファイルと関連するポリシーのみが作成され、署名は作成されません。

[新しい署名を作成] または **[**既存の署名を選択**]** を選択できます。

新しいセキュリティ設定を作成する場合、選択したシグニチャカテゴリは有効になり、デフォルトでは新しいシグニチャオブジェクトに記録されます。新しいシグネチャオブジェクトには、セキュリティ設定の名前として [名前の指定] 画面で入力したのと同じ名前が割り当てられます。

以前にシグニチャオブジェクトを設定していて、そのうちの 1 つを、作成中のセキュリティ設定に関連するシグニチャオブジェクトとして使用する場合は、「**Select Existing Signature** 」をクリックし、「Signature」リストからシグニチャオブジェクトを選択します。

既存のセキュリティ設定を変更する場合は、「既存の署名を選択」をクリックして、セキュリティ設定に別の署名オブジェクトを割り当てることができます。

「新しい署名を作成」をクリックすると、編集モードを「シンプル」または「アドバンス」として選択できます。

1. 署名保護の指定 **(**簡易モード**)**

シンプルモードでは、IIS（インターネットインフォメーションサーバー)、PHP、ActiveX などの一般的なアプリケーションの保護定義のプリセットリストを使用して、署名を簡単に設定できます。Simple モードのデフォルトカテゴリは以下のとおりです。

- CGI。PERL スクリプト、Unix シェルスクリプト、Python スクリプトなど、あらゆる言語の CGI スクリプトを使用する Web サイトへの攻撃に対する保護。
- Cold Fusion。アドビシステムズ ® ColdFusion® Web 開発プラットフォームを使用する Web サイトへの攻撃からの保護。
- 表紙。Microsoft® FrontPage® Web 開発プラットフォームを使用する Web サイトへの攻撃に対する保護。
- PHP。PHP オープンソースの Web 開発スクリプト言語を使用する Web サイトへの攻撃からの保護。
- クライアント側。Microsoft Internet Explorer、Mozilla Firefox、Opera ブラウザ、Adobe Acrobat Reader など、保護されている Web サイトへのアクセスに使用されるクライアントサイドツールへの攻撃からの保護。
- Microsoft IIS。Microsoft インターネットインフォメーションサーバー (IIS) を実行する Web サイトへの攻撃からの保護
- その他。Web サーバーやデータベースサーバーなど、他のサーバー側ツールへの攻撃に対する保護。

この画面では、「署名の選択」画面で選択した署名カテゴリに関連するアクションを選択します。設定できるアクションは以下のとおりです。

- ブロック
- ログ
- 統計情報

デフォルトでは、ログアクションと統計アクションは有効になっていますが、ブロックアクションは有効になっていません。アクションを設定するには、[設定] をクリックします。アクションドロップダウンリストを使用して、選択したすべてのカテゴリのアクション設定を変更できます。

1. 署名保護の指定 **(**詳細モード**)**

アドバンスモードでは、シグネチャ定義をより細かく制御でき、より多くの情報が得られます。シグニチャ定義を完全に制御する場合は、詳細モードを使用します。

この画面の内容は、「シグニチャオブジェクトの構成または変更」で説明されている「シグニチャオブジェクトの変更」(Modify Signatures Object)ダイアログボックスの内容と同じです。この画面では、[アクション] ドロップダウンリストまたは 3 つのドットが付いた円として表示されるアクションメニューをクリックして、アクションを構成できます。

7. ディーププロテクションの指定**:** この画面では、ウェブサイトやウェブサービスの保護に使用する高度な保護 (セキュリティチェックまたは単にチェックとも呼ばれます) を選択します。選択できるチェックは、「名前の指定」画面で選択したプロファイルタイプによって異なります。Web 2.0 アプリケーションプロファイルでは、すべてのチェックを使用できます。

詳細については、「セキュリティチェックの概要」および「高度なフォーム保護チェック」を参照してください。

有効化した高度な保護のアクションを設定します。設定できるアクションは次のとおりです。

- ブロック: 署名と一致する接続をブロックします。デフォルトでは、無効になっています。
- ログ: 後で分析できるように、署名と一致する接続を記録します。デフォルトで有効。
- 統計情報: シグネチャごとに、一致した接続の数を示す統計情報を保持し、ブロックされた接続の種類に関するその他の特定の情報を提供します。デフォルトでは、無効になっています。
- 学ぶ。このウェブサイトまたはウェブサービスへのトラフィックを観察し、このチェックに繰り返し違反する接続を使用して、チェックの推奨例外やチェックの新しいルールを生成してください。一部のチェックでのみ使用できます。学習機能の詳細については、学習機能の構成と使用、および学習のしくみ、例外（緩和）を設定する方法、またはチェックの学習ルールを展開する方法については、GUI を使用した手動設定を参照してください。

アクションを構成するには、チェックボックスをクリックして保護を選択し、[アクションの設定] をクリックして必要なアクションを選択します。必要に応じて他のパラメータを選択し、[**OK**] をクリックして [アクションの設定] ウィンドウを閉じます。

特定のチェックのすべてのログを表示するには、そのチェックを選択し、[ログ] をクリックして、[Web App Firewall ログ] の説明に従って Syslog Viewer を表示します。セキュリティチェックが保護された Web サイトまたは Web サービスへの正当なアクセスをブロックしている場合は、不要なブロックを示すログを選択し、[展開] をクリックして、そのセキュリティチェックの緩和を作成および実装できます。

アクション設定の指定が完了したら、[完了] をクリックしてウィザードを完了します。

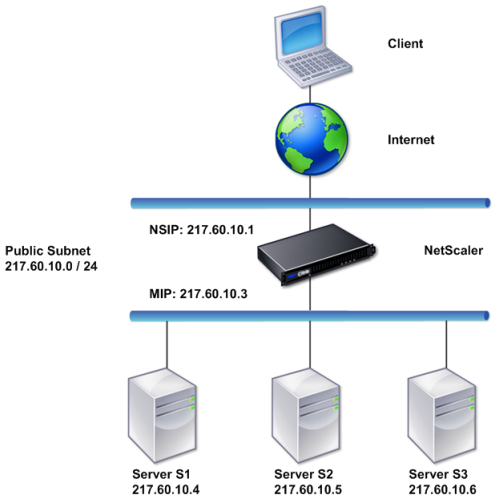
Web App Firewall ウィザードを使用して特定の種類の構成を実行する方法を示す 4 つの手順を次に示します。

新しい構成を作成する

次の手順に従って、アプリケーションファイアウォールウィザードを使用して新しいファイアウォール設定と署名オブジェクトを作成します。

1. [セキュリティ] > [アプリケーションファイアウォール] に移動します。

2. 詳細ペインの [はじめに] で、[** アプリケーションファイアウォール] をクリックします。ウィザードが開きます。



3. [名前の指定] 画面で、[** 新しい構成の作成] を選択します。
4. 「名前」フィールドに名前を入力して、「次へ」をクリックします。
5. [ルールを指定] 画面で、もう一度 [次へ] をクリックします。
6. 「署名の選択」画面で、編集モードとして「** 新しい署名と簡易署名の作成 ** 」を選択し、「次へ」をクリックします。
7. [署名保護の指定] 画面で、必要な設定を構成します。ブロッキングの対象となるシグニチャと、シグニチャのブロッキングを安全に有効にできるタイミングを判断する方法の詳細については、「シグニチャ」を参照してください。
8. [ディープ保護の指定] 画面で、[アクション 設定] で必要なアクションとパラメータを設定します。
9. 完了したら、[完了] をクリックしてアプリケーションファイアウォールウィザードを閉じます。

既存の構成を変更

既存の設定と既存のシグニチャカテゴリを変更するには、次の手順に従います。

1. [セキュリティ] > [アプリケーションファイアウォール] に移動します。
2. 詳細ペインの [はじめに] で、[アプリケーションファイアウォールウィザード] をクリックします。ウィザードが開きます。
3. 「名前の指定」画面で「既存の構成を変更」を選択し、「名前」ドロップダウンリストで、新しい構成で作成したセキュリティ構成を選択して、「次へ」をクリックします。

4. [ルールの指定] 画面で、[次へ] をクリックしてデフォルト値「true」のままにします。ルールを変更する場合は、「カスタムポリシー式を設定する」で説明されている手順に従います。
5. [署名の選択] 画面で、[既存の署名の選択] をクリックします。「既存の署名」ドロップダウンリストから適切なオプションを選択し、「次へ」をクリックします。「高度な署名保護」画面が表示されます。
   注**:** 既存の署名を選択した場合、「署名保護」のデフォルトの編集モードは「詳細」です。
6. [署名保護の指定] 画面で必要な設定を行い、[次へ] をクリックします。ブロッキングの対象となるシグニチャと、シグニチャのブロッキングを安全に有効にできるタイミングを判断する方法の詳細については、「シグニチャ」を参照してください。
7. [ディープ保護の指定] 画面で設定を構成し、[次へ] をクリックします。
8. 完了したら、[完了] をクリックして **Web App Firewall** ウィザードを閉じます。

署名なしで新しい設定を作成する

次の手順に従って、アプリケーションファイアウォールウィザードを使用して [署名の選択] 画面をスキップし、プロファイルと関連するポリシーのみを使用して署名なしで新しい構成を作成します。

1. [セキュリティ] > [アプリケーションファイアウォール] に移動します。
2. 詳細ペインの [はじめに] で、[アプリケーションファイアウォールウィザード] をクリックします。ウィザードが開きます。
3. 「名前の指定」画面で、「新規構成の作成」を選択します。
4. 「名前」フィールドに名前を入力して、「次へ」をクリックします。
5. 「ルールを指定」画面で、もう一度「次へ」をクリックします。
6. 「署名の選択」画面で、「スキップ」をクリックします。
7. [ディープ保護の指定] 画面で、[アクション 設定**]** で必要なアクションとパラメータを設定します。
8. 完了したら、[完了] をクリックしてアプリケーションファイアウォールウィザードを閉じます。

カスタムポリシー表現の設定

次の手順に従って、アプリケーションファイアウォールウィザードを使用して、特定のコンテンツのみを保護する特別なセキュリティ構成を作成します。この場合は、初期設定を変更する代わりに新しいセキュリティ設定を作成します。このタイプのセキュリティ設定にはカスタムルールが必要です。これにより、ポリシーは、選択した Web トラフィックにのみ設定を適用します。

1. [セキュリティ] > [アプリケーションファイアウォール] に移動します。
2. 詳細ペインの [はじめに] で、[アプリケーションファイアウォールウィザード] をクリックします。
3. 「名前の指定」画面で、「名前」テキストボックスに新しいセキュリティ構成の名前を入力し、「タイプ」ドロップダウンリストからセキュリティ構成の種類を選択して、「次へ」をクリックします。
4. 「ルールを指定」画面で、この Web アプリケーションに保護させたいコンテンツのみに一致するルールを入力します。****** よく使うエクスプレッションドロップダウンリストとエクスプレッションエディタを使用して ****** 、カスタムエクスプレッションを作成します。完了したら、[次へ] をクリックします。

5.「署名の選択」画面で編集モードを選択し、「次へ」をクリックします。
6. [署名保護の指定] 画面で、必要な設定を構成します。
7. [ディープ保護の指定] 画面で、[アクション 設定] で必要なアクションとパラメータを設定します。
8. 完了 したら、[完了] をクリックして アプリケーションファイアウォールウィザードを閉じます。

手動構成

August 15, 2023

グローバル以外のバインドポイントにプロファイルをバインドする場合は、バインドを手動で設定する必要があります。また、特定のセキュリティチェックでは、必要な例外を手動で入力するか、学習機能を有効にしてウェブサイトやウェブサービスが必要とする例外を生成する必要があります。これらのタスクの中には、Web App Firewall ウィザードを使用して実行できないものがあります。

Web App Firewall の動作に精通していて、手動設定を希望する場合は、シグニチャオブジェクトとプロファイルを手動で設定し、シグニチャオブジェクトをプロファイルに関連付け、設定する Web トラフィックと一致するルールを含むポリシーを作成し、ポリシーをプロファイルに関連付けることができます。次に、ポリシーをグローバルまたはバインドポイントにバインドして有効にすると、完全なセキュリティ構成が作成されました。

手動構成の場合は、GUI (グラフィカルインターフェイス) またはコマンドラインを使用できます。GUI を使用することをお勧めします。すべての構成タスクをコマンドラインで実行できるわけではありません。シグニチャの有効化や学習したデータの確認など、特定のタスクは GUI で実行する必要があります。他のほとんどのタスクは GUI で実行する方が簡単です。

構成の複製

GUI (GUI) またはコマンドラインインターフェイス (CLI) を使用して Web App Firewall を手動で構成すると、構成は /nsconfig/ns.conf ファイルに保存されます。そのファイル内のコマンドを使用して、別のアプライアンスに構成を複製できます。コマンドを 1 つずつ切り取って CLI に貼り付けることも、複数のコマンドを /var/tmp フォルダのテキストファイルに保存してバッチファイルとして実行することもできます。次に、別のアプライアンスの /nsconfig/ns.conf ファイルからコピーされたコマンドを含むバッチファイルの実行例を示します。

```
> batch -f /var/tmp/appfw_add.txt
```

> 警告:
>
> インポートコマンドは、ns.conf ファイルには保存されません。ns.conf ファイルからコマンドを実行して別のアプライアンスに構成をレプリケートする前に、構成で使用されるすべてのオブジェクト（シグニチャ、エラーページ、WSDL、スキーマなど）を、構成をレプリケートするアプライアンスにインポートする必要があります。ns.conf ファイルに保存された Web App Firewall プロファイルを追加するための add コマンドには、インポートされたオブジェクトの名前が含まれる場合がありますが、参照オブジェクトがそのアプライアンス

> 上に存在しない場合、別のアプライアンスで実行するとそのようなコマンドが失敗することがあります。

構成のレプリケーションのインポートまたはエクスポートの詳細については、「シグネチャエクスポート」および「一般的なインポートエクスポートのトピック」を参照してください。

NetScaler GUI を使用した手動構成

March 20, 2024

Web App Firewall 機能を手動で構成する必要がある場合は、NetScaler GUI の手順を使用することを Citrix では推奨しています。

署名オブジェクトを作成および設定するには

シグニチャを設定する前に、適切なデフォルトシグニチャオブジェクトテンプレートからシグニチャオブジェクトを作成する必要があります。コピーに新しい名前を割り当て、コピーを設定します。デフォルトのシグニチャオブジェクトを直接設定または変更することはできません。次の手順では、シグニチャオブジェクトを設定する基本的な手順を示します。詳細な手順については、シグニチャ機能の手動設定を参照してください。

1. **Security > NetScaler Web App Firewall > Signatures** に移動します。
2. 詳細ペインで、テンプレートとして使用する署名オブジェクトを選択し、**[追加]** をクリックします。

   選択肢は次のとおりです:

   - デフォルトのシグニチャ。シグニチャルール、SQL インジェクションルール、クロスサイトスクリプティングルールが含まれます。
   - **XPath** インジェクション。デフォルトシグネチャのすべての項目が含まれ、さらに XPath インジェクションルールも含まれます。

3. 署名オブジェクトの追加ダイアログボックスで、新しい署名オブジェクトの名前を入力して「**OK**」をクリックし、「閉じる」をクリックします。名前は、英字、数字、またはアンダースコア記号で始まり、1 ～31 個の英数字、およびハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、およびアンダースコア (_) 記号で構成できます。
4. 作成した署名オブジェクトを選択し、「開く」をクリックします。
5. 「署名オブジェクトの変更」ダイアログで、左側の「フィルター条件を表示」オプションを設定して、設定するフィルター項目を表示します。

   これらのオプションを変更すると、指定した結果が右側の「フィルタ結果」(Filtered Results) ウィンドウに表示されます。シグニチャのカテゴリの詳細については、「署名」を参照してください。

6. [**Filtered Results**] 領域で、該当するチェックボックスをオンまたはオフにして、シグニチャの設定を構成します。
7. 終了したら、[閉じる] をクリックします。

GUI を使用して **Web App Firewall** プロファイルを作成するには

Web App Firewall プロファイルを作成するには、いくつかの構成の詳細を指定するだけで済みます。

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **Web App Firewall** プロファイルの作成ダイアログボックスで、プロファイルの名前を入力します。

   名前は、文字、数字、またはアンダースコア記号で始まり、1 ～31 文字の英数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) で構成できます。
4. ドロップダウンリストからプロファイルタイプを選択します。
5. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して **Web App Firewall** プロファイルを構成するには

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ウィンドウで、構成するプロファイルを選択し、[編集] をクリックします。
3. [**Web App Firewall** プロファイルの構成] ダイアログボックスの [セキュリティチェック] タブで、セキュリティチェックを構成します。

   - チェックに対するアクションを有効または無効にするには、一覧でそのアクションのチェックボックスをオンまたはオフにします。
   - そのチェックに他のパラメータを設定するには、一覧でそのチェックの右端にある青色の山形をクリックします。表示されるダイアログボックスで、パラメータを設定します。これらはチェックごとに異なります。

     チェックを選択し、ダイアログボックスの下部にある [開く] をクリックして、そのチェックの [緩和の構成] ダイアログボックスまたは [規則の構成] ダイアログボックスを表示することもできます。これらのダイアログボックスもチェックごとに異なります。ほとんどの場合、[チェック] タブと [一般] タブがあります。チェックが緩和またはユーザー定義規則をサポートしている場合、「チェック」(Checks) タブには「追加」(Add) ボタンが表示されます。このボタンでは、チェックのリラクゼーションまたは規則を指定できる別のダイアログボックスが開きます。(緩和とは、特定のトラフィックを小切手から除外するルールです)。緩和が既に設定されている場合は、緩和を選択して [開く] をクリックして変更できます。

- チェックの学習済み例外またはルールを確認するには、チェックを選択し、[学習済み違反] をクリックします。[学習済み規則の管理] ダイアログボックスで、学習した例外または規則を順に選択します。
  - 例外またはルールを編集してリストに追加するには、[**Edit & Deploy**] をクリックします。
  - 例外またはルールを変更せずに受け入れるには、[**Deploy**] をクリックします。
  - リストから例外または規則を削除するには、[スキップ] をクリックします。
- 確認する例外または規則の一覧を更新するには、[**Refresh**] をクリックします。
- ラーニングビジュアライザーを開いて学習したルールを確認するには、ビジュアライザーをクリックします。
- チェックと一致した接続のログエントリを確認するには、チェックを選択し、[ログ] をクリックします。この情報を使用してどのチェックが攻撃に一致するかを判断し、それらのチェックをブロックできるようにすることができます。この情報を使用して、正当なトラフィックと一致するチェックを特定することもできます。これにより、正当な接続を許可するように適切な免除を構成できます。ログの詳細については、ログ、統計、およびレポートを参照してください。
- チェックを完全に無効にするには、リストで、そのチェックの右側にあるすべてのチェックボックスをオフにします。

4. [設定] タブで、プロファイル設定を構成します。
   - 以前に作成して設定したシグニチャのセットにプロファイルを関連付けるには、**[Common Settings]** で、[Signatures] ドロップダウンリストからそのシグニチャのセットを選択します。

     > 注:
     >
     > [共通設定] セクションを表示するには、ダイアログボックスの右側のスクロールバーを使用して下にスクロールする必要があります。

   - HTML または XML エラーオブジェクトを設定するには、該当するドロップダウンリストからオブジェクトを選択します。

     > 注:
     >
     > まず、インポートペインで使用するエラーオブジェクトをアップロードする必要があります。

   - デフォルトの XML コンテンツタイプを設定するには、「デフォルトリクエスト」テキストボックスと「デフォルトレスポンス」テキストボックスにコンテンツタイプ文字列を直接入力するか、「許可されたコンテンツタイプの管理」をクリックして、許可されるコンテンツタイプのリストを管理します。
5. 学習機能を使用する場合は、[学習] をクリックし、プロファイルの学習設定を構成します。詳細については、「機能の構成と学習機能」を参照してください。
6. 「**OK**」をクリックして変更を保存し、「プロファイル」ペインに戻ります。

Web App Firewall ルールまたは緩和の設定

このダイアログボックスでは、設定するセキュリティチェックに応じて 2 種類の情報を設定します。ほとんどの場合、セキュリティチェックに例外 (または緩和) を設定します。拒否 URL チェックまたはフィールドフォーマットチェックを設定する場合は、追加 (またはルール) を設定します。どちらのプロセスも同じです。

NetScaler GUI を使用して緩和ルールを構成するには

1. **[** セキュリティ**]** > **[NetScaler Web App Firewall]** > **[**プロファイル**]** に移動します。
2. 「プロファイル」ペインで、設定するプロファイルを選択し、「編集」をクリックします。
3. 「**Web App Firewall** プロファイルの設定」ページで、「詳細設定」セクションの「緩和ルール」をクリックします。「緩和規則」セクションには、Web App Firewall 緩和規則の全リストが記載されています。
4. 設定するセキュリティルールをクリックし、[編集] をクリックします。
5. URL 緩和ルールページには、このルールに設定できるアクションのリストと、既存の緩和またはルールのリストが含まれています。手動で緩和を追加していない場合や、学習エンジンが推奨する緩和を承認していない場合は、リストが空になる可能性があります。リストの下には、リストのリラクゼーションを追加、変更、削除、有効化、または無効化できるボタンが並んでいます。
6. リラクゼーションまたはルールを追加または変更するには、次のいずれかを実行します。

   - 新しいリラクゼーションを追加するには、「追加」をクリックします。
   - 既存のリラクゼーションを変更するには、変更するリラクゼーションを選択し、「開く」をクリックします。

   「開始 **URL** 緩和ルール」ページが表示されます。タイトル以外は、これらのダイアログボックスは同じです。
7. 以下の説明に従ってダイアログボックスに入力します。各チェックのダイアログボックスは異なります。以下のリストには、どのダイアログボックスにも表示される可能性のあるすべての要素が含まれています。

   - 「有効」チェックボックス-選択するとこの緩和またはルールをアクティブに使用し、オフにすると非アクティブになります。
   - 添付コンテンツタイプ—XML 添付ファイルのコンテンツタイプ属性。テキスト領域に、許可する XML 添付ファイルの Content-Type 属性と一致する正規表現を入力します。
   - アクション **URL**—テキスト領域に、Web フォームに入力されたデータの配信先の URL を定義する PCRE 形式の正規表現を入力します。
   - **Cookie**：テキスト領域に、Cookie を定義する PCRE 形式の正規表現を入力します。
   - フィールド名—Web フォームのフィールド名要素には、フィールド名、フォームフィールド、または同様の名前のラベルを付けることができます。テキスト領域に、フォームフィールドの名前を定義する PCRE 形式の正規表現を入力します。

- オリジン **URL** から-テキスト領域に、Web フォームをホストする URL を定義する PCRE 形式の正規表現を入力します。
- **From Action URL**—テキスト領域に、Web フォームに入力されたデータの配信先の URL を定義する PCRE 形式の正規表現を入力します。
- 名前—XML 要素または属性名。テキスト領域に、要素または属性の名前を定義する PCRE 形式の正規表現を入力します。
- **URL** —**URL** 要素には、アクション URL、拒否 URL、フォームアクション URL、フォームオリジン URL、開始 URL、または単に URL というラベルを付けることができます。テキスト領域に、URL を定義する PCRE 形式の正規表現を入力します。
- フォーマット—フォーマットセクションには、リストボックスやテキストボックスを含む複数の設定が含まれています。次のいずれかが表示されることがあります。
  - タイプ**-**タイプドロップダウンリストでフィールドタイプを選択します。新しいフィールドタイプ定義を追加するには、「管理」をクリックします。
  - 最小文字数—ユーザにこのフィールドへの入力を強制する場合は、最小文字数を表す正の整数を入力します。デフォルト:0 (フィールドを空白のままにできます)
  - 最大長—このフィールドのデータ長を制限するには、最大文字数を表す正の整数を入力します。デフォルト:65535
- 場所—緩和の対象となるリクエストの要素をドロップダウンリストから選択します。HTML セキュリティチェックでは、次の選択肢があります。
  - フォームフィールド-Web フォームのフォームフィールド。
  - ヘッダー-リクエストヘッダー。
  - クッキー-クッキーヘッダーを設定します。

  XML セキュリティチェックでは、次の選択肢があります。
  - エレメント-XML エレメント。
  - 属性-XML 属性。
- 添付ファイルの最大サイズ—XML 添付に使用できる最大サイズ (バイト単位)。
- コメント—テキスト領域にコメントを入力します。オプションです。

注: 正規表現を必要とする要素については、正規表現を入力するか、「正規表現トークン」メニューを使用して正規表現の要素と記号をテキストボックスに直接挿入するか、「正規表現エディタ」をクリックして「正規表現の追加」ダイアログボックスを開き、それを使用して表現を作成できます。

8. リラクゼーションまたはルールを削除するには、そのリラクゼーションまたはルールを選択し、「削除」をクリックします。
9. リラクゼーションまたはルールを有効にするには、そのリラクゼーションまたはルールを選択し、「有効にする」をクリックします。

10. リラクゼーションまたはルールを無効にするには、そのリラクゼーションまたはルールを選択し、「無効」をクリックします。
11. 統合された対話型グラフィック表示で、既存のすべての緩和の設定と関係を構成するには、[ビジュアライザー] をクリックし、表示ツールを使用します。

    > 注記**:**
    >
    > 「ビジュアライザー」ボタンは、すべてのチェック緩和ダイアログボックスに表示されるわけではありません。

12. このチェックの学習ルールを確認するには、[学習] をクリックし、学習機能を設定して使用するにはの手順を実行します。
13. [**OK**] をクリックします。

NetScaler GUI を使用して学習したルールを構成するには

1. **[** セキュリティ **] > [NetScaler Web App Firewall] > [**プロファイル**]** に移動します。
2. [プロファイル] ペインでプロファイルを選択し、[編集] をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、「****** 詳細設定から学習したルール」をクリックします。****** 学習ルールセクションには、現在のプロファイルで利用できる学習機能をサポートするセキュリティチェックのリストが表示されます。
4. 学習しきい値を設定するには、セキュリティチェックを選択し、**[**設定**]** をクリックします。
5. 「動的プロファイリングおよび学習ルールの設定」ページでは、設定を設定できます。詳細については、「動的プロファイル設定」を参照してください。

    - 最小数のしきい値。設定するセキュリティチェックの学習設定によっては、最小数のしきい値とは、監視する必要があるユーザーセッションの最小数、監視する必要のあるリクエストの最小数、または学習済み緩和が生成される前に特定のフォームフィールドを確認する必要がある最小回数を指す場合があります。デフォルト:1
    - 回数のしきい値のパーセンテージ。設定しているセキュリティチェックの学習設定に応じて、しきい値の割合は、セキュリティチェックに違反した観察されたユーザーセッションの合計の割合、リクエストの割合、またはフォームフィールドが特定のフィールドタイプと一致した回数の割合を指す場合があります。学習したリラクゼーションが生成されます。デフォルト:0

6. 学習したデータをすべて削除して学習機能をリセットし、最初から観測を再開する必要があるようにするには、「すべての学習データを削除」アクションを選択します。

    > 注:

> このボタンは、レビューされておらず、承認またはスキップされた学習済みの推奨事項のみを削除します。受け入れられて展開された学習済み緩和は削除されません。

7. 学習エンジンを特定の IP セットからのトラフィックに制限するには、[信頼できる学習クライアント] をクリックし、使用する IP アドレスをリストに追加します。

   a) [信頼できる学習クライアント] リストに IP アドレスまたは IP アドレス範囲を追加するには、[追加] をクリックします。
   b)「AppFirewall プロファイルからトラステッドクライアントへのバインディング」ページで、「追加」をクリックします。
   c)「有効」チェックボックスを選択して機能を有効にします。
   d)「トラステッド・ラーニング・クライアント」** ボックスに、IP アドレスまたは IP アドレス範囲を CIDR 形式で入力します。
   e)「コメント」テキスト領域に、この IP アドレスまたは IP アドレス範囲を説明するコメントを入力します。
   f) **[**作成**]** して **[**閉じる**]** をクリックします。

8. 既存の IP アドレスまたは範囲を変更するには、IP アドレスまたは IP 範囲をクリックし、[編集] をクリックします。名前以外は、表示されるダイアログボックスは「信頼できる学習クライアントの追加」ダイアログボックスと同じです。

9. IP アドレスまたは IP アドレス範囲を無効または有効にして、一覧に残すには、IP アドレスまたは範囲をクリックし、必要に応じて [無効化] または [有効にする] をクリックします。

10. IP アドレスまたは IP 範囲を完全に削除するには、IP アドレスまたは IP 範囲をクリックし、[削除] をクリックします。

11.「閉じる」をクリックすると、**NetScaler Web App Firewall** プロファイルページに戻ります。

NetScaler GUI を使用して **NetScaler Web App Firewall** ポリシーを作成するには

1. **[**セキュリティ] > [**NetScaler Web App Firewall**] > [ポリシー] に移動します。**
2. ポリシーページで、**NetScaler Web App Firewall** ポリシーリンクをクリックします。
3. **NetScaler Web App Firewall** ポリシーページで、「追加」をクリックします。
4. NetScaler Web App Firewall ポリシーの作成ページで、次のパラメーターを設定します。

   a) 名前。名前は、文字、数字、またはアンダースコア記号で始まり、1 ～128 文字の英数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) で構成できます。
   b) プロフィール。[プロファイル（Profile）] ドロップダウンリストから、このポリシーに関連付けるプロファイルを選択します。[新規] をクリックしてポリシーに関連付けるプロファイルを作成し、[変更] をクリックして既存のプロファイルを変更できます。
   c) 式。[式] テキスト領域で、ポリシーのルールを作成します。

d) ログアクション。ログアクションを追加するか、既存のログアクションを変更できます。
e) [コメント]。ポリシーに関する簡単な説明。

5.「作成」または「**OK**」をクリックし、「閉じる」をクリックします。

Web App Firewall ルール **(**式**)** を作成または構成するには

ポリシールールは、式とも呼ばれ、ポリシーに関連付けられたプロファイルを使用して Web App Firewall がフィルタリングする Web トラフィックを定義します。他の NetScaler ポリシールール（または表現）と同様に、Web App Firewall ルールは NetScaler 式の構文を使用します。この構文は強力で柔軟性があり、拡張可能です。この一連の命令で完全に記述するには複雑すぎます。次の手順を使用して単純なファイアウォールポリシールールを作成するか、ポリシー作成プロセスの概要として読むことができます。

1. まだ作成していない場合は、Web App Firewall ウィザードまたは NetScaler GUI で適切な場所に移動してポリシールールを作成します。

   - Web App Firewall ウィザードでポリシーを構成する場合は、ナビゲーションペインで［**NetScaler Web App Firewall** ウィザード］をクリックし、詳細ペインで［**NetScaler Web App Firewall** ウィザード］をクリックし、［ルールの指定］タブページに移動します。
   -「ルールを指定」ページで、ドロップダウンリストから式のプレフィックスを選択します。選択肢は次のとおりです:
   - **HTTP** HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。
   - **SYS**。1 つ以上の保護された Web サイト。リクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
   - **CLIENT**。要求を送信したコンピュータ。リクエストの送信者の側面を調べる場合は、これを選択します。
   - サーバー。要求の送信先のコンピュータ。リクエストの受信者の何らかの側面を調べたい場合は、これを選択してください。

   プレフィックスを選択すると、Web App Firewall に 2 つの部分からなるプロンプトウィンドウが表示され、次に選択可能な選択肢が一番下部に表示されます。

2. 次の用語を選択してください。

   プレフィックスとして HTTP を選択した場合、唯一の選択肢は REQ です。これは、要求と応答のペアを指定する REQ です。(Web App Firewall は、リクエストとレスポンスをそれぞれ別々ではなくユニットとして動作します)。別のプレフィックスを選択した場合、選択肢はより多様になります。特定の選択肢に関するヘルプを表示するには、その選択肢を 1 回クリックすると、その選択に関する情報が下のプロンプトウィンドウに表示されます。

   使用する用語が決まったら、ダブルクリックして [式] ウィンドウに挿入します。

3. 選択した期間の後にピリオドを入力します。次に、前の手順で説明したように、次の用語を選択するように求められます。用語で値を入力する必要がある場合は、適切な値を入力します。たとえば、HTTP.REQ.HEADER (「」) を選択した場合、引用符の間にヘッダー名を入力します。
4. 式が終了するまで、プロンプトから用語を選択し、必要な値を入力します。

   特定の目的のための式の例をいくつか挙げます。

   - 特定の **Web** ホスト。特定の Web ホストからのトラフィックを照合するには、次の手順を実行します。

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

shopping.example.com には、マッチングさせたいウェブホストの名前を代入してください。

- 特定の **Web** フォルダまたはディレクトリ。Web ホスト上の特定のフォルダまたはディレクトリからのトラフィックを照合するには、次の手順を実行します。

```
HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
```

www.example.com の場合は、ウェブホストの名前を代用してください。フォルダーの場合は、一致させたいコンテンツのフォルダーまたはパスに置き換えてください。たとえば、ショッピングカートが /solutions/orders というフォルダにある場合、その文字列をフォルダに置き換えます。

- 特定の種類のコンテンツ**:GIF** 画像。GIF 形式の画像を一致させるには:

```
HTTP.REQ.URL.ENDSWITH(".png")
```

他のフォーマットイメージを一致させるには、.png の代わりに別の文字列を置き換えます。

- 特定の種類のコンテンツ**:** スクリプト。CGI-BIN ディレクトリにあるすべての CGI スクリプトを一致させるには、次の手順を実行します。

```
HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
```

.js 拡張子を持つすべての JavaScript を照合するには、次の手順に従います。

```
HTTP.REQ.URL.ENDSWITH(".js")
```

ポリシー式の作成の詳細については、「ポリシーと式」を参照してください。

> 注:
>
> コマンドラインを使用してポリシーを構成する場合は、NetScaler 式内の二重引用符をエスケープしてください。たとえば、GUI で入力すると、次の式は正しいです。

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

ただし、コマンドラインで入力する場合は、代わりに次のように入力する必要があります。

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

```
1 ![Policy expression configuration](/en-us/citrix-adc/media/waf-rule.png
    )
```


[式の追加] ダイアログボックスを使用してファイアウォールルール (式) を追加するには

[式の追加] ダイアログボックス（式エディタとも呼ばれる）は、NetScaler 式の言語に慣れていないユーザーが、フィルタリングするトラフィックに一致するポリシーを構築するのに役立ちます。

1. まだ行っていない場合は、Web App Firewall ウィザードまたは Citrix ADC GUI で適切な場所に移動します。
   - **Web App Firewall** ウィザードでポリシーを構成する場合は、ナビゲーションペインで [**Web App Firewall**] をクリックし、詳細ペインで [**Web App Firewall** Wizard] をクリックして、[ルールの指定] 画面に移動します。
   - ポリシーを手動で構成する場合は、ナビゲーションペインで、[**Web App Firewall]**、**[**** ポリシー]、[ファイアウォール**]** の順に展開します。詳細ウィンドウで、ポリシーを作成するには、**[**** 追加] をクリックします。既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。
2. [ルールの指定] 画面の [**Web App Firewall** プロファイルの作成] ダイアログボックス、または [**Web App Firewall** プロファイルの構成] ダイアログボックスで、[追加] をクリックします。
3. [式の追加] ダイアログボックスの [式の作成] 領域の最初のリストボックスで、次のプレフィックスのいずれかを選択します。
   - **HTTP** HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。デフォルトの選択肢。
   - **SYS**。1 つ以上の保護された Web サイト。リクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
   - **CLIENT**。要求を送信したコンピュータ。リクエストの送信者の側面を調べる場合は、これを選択します。
   - サーバー。要求の送信先のコンピュータ。リクエストの受信者の何らかの側面を調べたい場合は、これを選択してください。
4. 2 番目のリストボックスで、次の用語を選択します。ダイアログボックスでは、コンテキストに有効な用語のみが含まれるようにリストが自動的に調整されるため、使用可能な用語は、前の手順で行った選択によって異なります。たとえば、前のリストボックスで [HTTP] を選択した場合、リクエストに対する唯一の選択肢は [REQ] です。Web App Firewall は、リクエストと関連する応答を 1 つのユニットとして扱い、両方をフィルタリングするため、個別にレスポンスを指定する必要はありません。2 番目の用語を選択すると、第 2 項の右に 3 番目のリストボックスが表示されます。[ヘルプ] ウィンドウには 2 番目の項の説明が表示され、[式のプレビュー] ウィンドウにはエクスプレッションが表示されます。
5. 3 番目のリストボックスで、次の用語を選択します。右側に新しいリストボックスが表示され、ヘルプウィンドウが新しい用語の説明が表示されます。エクスプレッションのプレビュー (Preview Expression) ウィンドウが更新され、指定したポイントまでのエクスプレッションが表示されます。
6. 条件の選択を続け、引数の入力を求められたら、式が完成します。既に用語を選択した後に間違えたり、式を変更したい場合は、単に別の用語を選択できます。式が変更され、変更した用語の後に追加した引数やその他の用語はクリアされます。
7. 式の作成が終了したら、「OK」をクリックして「式の追加」ダイアログボックスを閉じます。エクスプレッションがエクスプレッションテキストエリアに挿入されます。

NetScaler GUI を使用して **Web App Firewall** ポリシーをバインドするには

1. 次のいずれかを行います:

   - [セキュリティ] > [**Web App Firewall**] に移動し、詳細ペインで [アプリケーションファイアウォールポリシーマネージャー] をクリックします。
   - **[**** セキュリティ] > [**NetScaler Web App Firewall**] > [ポリシー] > [ファイアウォール] に移動し、[NetScaler Web App Firewall ポリシー] ペインで [ポリシーマネージャー] をクリックします。**

2. アプリケーションファイアウォールポリシーマネージャーダイアログで、ポリシーをバインドするバインドポイントをドロップダウンリストから選択します。選択肢は以下のとおりです。

   - グローバルオーバーライド。このバインドポイントにバインドされたポリシーは、NetScaler アプライアンス上のすべてのインターフェイスからのすべてのトラフィックを処理し、他のポリシーよりも先に適用されます。
   - **LB** 仮想サーバー。負荷分散仮想サーバーにバインドされたポリシーは、その負荷分散仮想サーバーによって処理されるトラフィックにのみ適用され、デフォルトのグローバルポリシーの前に適用されます。LB 仮想サーバーを選択したら、このポリシーをバインドする特定の負荷分散仮想サーバーも選択する必要があります。
   - **CS** 仮想サーバー。コンテンツスイッチング仮想サーバーにバインドされたポリシーは、そのコンテンツスイッチング仮想サーバーによって処理されるトラフィックにのみ適用され、デフォルトのグローバルポリシーの前に適用されます。CS 仮想サーバーを選択したら、このポリシーをバインドする特定のコンテンツスイッチング仮想サーバーも選択する必要があります。
   - デフォルトグローバル。このバインドポイントにバインドされたポリシーは、NetScaler アプライアンス上のすべてのインターフェイスからのすべてのトラフィックを処理します。
   - ポリシーラベル。ポリシーラベルにバインドされたポリシーは、ポリシーラベルがルーティングするトラフィックを処理します。ポリシーラベルは、このトラフィックにポリシーが適用される順序を制御します。
   - なし。ポリシーをどのバインドポイントにもバインドしないでください。

3. ［続行］をクリックします。既存の Web App Firewall ポリシーのリストが表示されます。
4. バインドするポリシーをクリックして選択します。
5. バインディングをさらに調整します。

   - ポリシーの優先度を変更するには、フィールドをクリックして有効にし、新しい優先度を入力します。「優先順位を再作成」を選択して、優先順位を均等に再設定することもできます。
   - ポリシー表現を変更するには、そのフィールドをダブルクリックして **Web App Firewall** ポリシーの設定ダイアログボックスを開き、ポリシー表現を編集できます。
   - Goto Expression を設定するには、**Goto Expression** 列の見出しにあるフィールドをダブルクリックしてドロップダウンリストを表示し、エクスプレッションを選択できます。
   - Invoke オプションを設定するには、Invoke 列の見出しにあるフィールドをダブルクリックしてドロップダウンリストを表示し、そこで式を選択できます。

6. ステップ 3 ～6 を繰り返して、グローバルにバインドするその他の Web App Firewall ポリシーを追加します。
7. [**OK**] をクリックします。ポリシーが正常にバインドされたことを示すメッセージがステータスバーに表示されます。

コマンドラインインターフェイスによる手動設定

August 15, 2023

> 注:
>
> Web App Firewall 機能を手動で構成する必要がある場合は、NetScaler GUI の手順を使用することを Citrix では推奨しています。

Web App Firewall の機能は、**NetScaler** のコマンドインターフェイスから構成できます。ただし、重要な例外があります。コマンドインターフェイスからシグニチャを有効にすることはできません。7 つのカテゴリにおよそ 1,000 のデフォルトシグネチャがあり、コマンドインターフェイスではタスクが複雑すぎます。コマンドラインから機能を有効または無効にしたり、パラメータを設定したりできますが、手動緩和は設定できません。適応型学習機能を設定してコマンドラインから学習を有効にすることはできますが、学習した緩和や学習したルールを確認して承認またはスキップすることはできません。コマンドラインインターフェイスは、NetScaler アプライアンスと Web App Firewall の使用に精通している上級ユーザーを対象としています。

NetScaler コマンドラインを使用して Web App Firewall を手動で構成するには、任意の Telnet またはセキュアシェルクライアントを使用して NetScaler コマンドラインにログオンします。

コマンドラインインターフェイスを使用してプロファイルを作成するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw profile <name> [-defaults ( basic | advanced )]`
- `set appfw profile <name> -type ( HTML | XML | HTML XML )`
- `save ns config`

例

次の例では、基本デフォルトで pr-basic という名前のプロファイルを追加し、プロファイルタイプに HTML を割り当てます。これは、HTML Web サイトを保護するためのプロファイルの適切な初期設定です。

```
1 add appfw profile pr-basic -defaults basic
2 set appfw profile pr-basic -type HTML
```



```
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してプロファイルを構成するには

コマンドプロンプトで、次のコマンドを入力します。

- set appfw profile <name> <arg1> [<arg2> ...]。<arg1>はパラメーターを表し、<arg2>は別のパラメーターか<arg1>のパラメーターに割り当てられる値を表します。特定のセキュリティチェックを構成するときに使用するパラメータの詳細については、高度な保護とそのサブトピックを参照してください。その他のパラメータについては、「縦断を作成するためのパラメータ」を参照してください。
- save ns config

例

次の例は、基本的な既定値で作成された HTML プロファイルを設定して、シンプルな HTML ベースの Web サイトの保護を開始する方法を示しています。この例では、ほとんどのセキュリティチェックでロギングと統計情報の管理を有効にしていますが、誤検出率が低く、特別な設定を必要としないチェックについてのみブロックを有効にします。また、安全でない HTML や安全でない SQL の変換を有効にすることで、攻撃を防ぎながらウェブサイトへのリクエストをブロックしません。ログと統計を有効にすると、後でログを確認して、特定のセキュリティチェックのブロックを有効にするかどうかを判断できます。

```
set appfw profile -startURLAction log stats
set appfw profile -denyURLAction block log stats
set appfw profile -cookieConsistencyAction log stats
set appfw profile -crossSiteScriptingAction log stats
set appfw profile -crossSiteScriptingTransformUnsafeHTML ON
set appfw profile -fieldConsistencyAction log stats
set appfw profile -SQLInjectionAction log stats
set appfw profile -SQLInjectionTransformSpecialChars ON
set appfw profile -SQLInjectionOnlyCheckFieldsWithSQLChars ON
set appfw profile -SQLInjectionParseComments checkall
set appfw profile -fieldFormatAction log stats
set appfw profile -bufferOverflowAction block log stats
set appfw profile -CSRFtagAction log stats
save ns config
<!--NeedCopy-->
```

ポリシーを作成および設定するには

コマンドプロンプトで、次のコマンドを入力します。

- add appfw policy <name> <rule> <profile>
- save ns config

例

次の例では、ホスト blog.example.com との間で送受信されるすべてのトラフィックをインターセプトするルールを使用して、pl-blog という名前のポリシーを追加し、そのポリシーをプロファイル pr-blog に関連付けます。

```
add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com")" pr-blog
<!--NeedCopy-->
```

Web App Firewall ポリシーをバインドするには

コマンドプロンプトで、次のコマンドを入力します。

- `bind appfw global <policyName> <priority>`
- `save ns config`

例

次の例では、pl-blog という名前のポリシーをバインドし、優先度 10 を割り当てます。

```
bind appfw global pl-blog 10
save ns config
<!--NeedCopy-->
```

PE ごとにセッション制限を設定するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw settings <session limit>`

例

次の例では、PE ごとにセッション制限を設定しています。

```
> set appfw settings -sessionLimit 500000`

Done

Default value:100000   Max value:500000 per PE
<!--NeedCopy-->
```


署名

August 15, 2023

Web App Firewall のシグネチャは、既知の攻撃からウェブサイトを保護するタスクを簡素化するための特定の設定可能なルールを提供します。シグニチャは、オペレーティングシステム、Web サーバー、Web サイト、XML ベースの Web サービス、またはその他のリソースに対する既知の攻撃のコンポーネントであるパターンを表します。事前設定された Web App Firewall の組み込みルールまたはネイティブルールが豊富に用意されているため、パターンマッチングの機能を活用して攻撃を検出し、アプリケーションの脆弱性から保護する、使いやすいセキュリティソリューションが提供されます。

独自の署名を作成することも、組み込みのテンプレートで署名を使用することもできます。Web App Firewall には、次の 2 つの組み込みテンプレートがあります。

- デフォルト署名: このテンプレートには、SQL インジェクションキーワード、SQL 特殊文字列、SQL 変換ルール、SQL ワイルド文字の全リストに加えて、1,300 を超える署名があらかじめ設定されています。また、クロスサイトスクリプティングの拒否パターン、クロスサイトスクリプティングの許可された属性とタグも含まれています。これは読み取り専用テンプレートです。このテンプレートの内容は表示できますが、追加、編集、削除はできません。使用するには、コピーを作成する必要があります。独自のコピーでは、トラフィックに適用するシグニチャルールを有効にし、シグニチャルールがトラフィックと一致したときに実行するアクションを指定できます。

Web App Firewall シグニチャは、Snortによって公開されているルールから得られます。Snort は、さまざまな攻撃やプローブを検出するためのリアルタイムトラフィック分析を実行できるオープンソースの侵入防御システムです。

- ***Xpath** インジェクションパターン: このテンプレートには、XPath (XML Path Language) インジェクション攻撃の検出に使用されるリテラルキーワードと PCRE キーワードと特殊文字列のセットがあらかじめ設定されています。

空白の署名: 組み込みの *Default Signatures テンプレートのコピーを作成するほかに、空白の署名テンプレートを使用して署名オブジェクトを作成できます。空白の署名オプションを使用して作成した署名オブジェクトにはネイティブの署名ルールはありませんが、*Default テンプレートと同様に、SQL/クロスサイトスクリプティングの組み込みエンティティがすべて含まれています。

外部形式の署名**:Web App Firewall** は外部形式の署名もサポートしています。NetScaler Web App Firewall でサポートされている XSLT ファイルを使用して、サードパーティのスキャンレポートをインポートできます。次のスキャンツールでは、外部形式のファイルをネイティブ形式に変換するための組み込み XSLT ファイルセットを使用できます。

- センジーク
- Web アプリ向けの高度なセキュリティ
- IBM AppScan エンタープライズ

- IBM AppScan 標準。
- Qualys
- Qualys Cloud
- Whitehat
- Hewlett Packard Enterprise WebInspect
- Rapid7 Appspider
- Acunetix

アプリケーションのセキュリティ保護

セキュリティが厳しくなると、処理のオーバーヘッドが増加します。シグネチャには、アプリケーションの保護を最適化するのに役立つ次のデプロイオプションが用意されています。

- ネガティブセキュリティモデル: ネガティブセキュリティモデルでは、あらかじめ設定された豊富なシグネチャルールセットを使用してパターンマッチングの機能を適用し、攻撃を検出し、アプリケーションの脆弱性から保護します。望まないものだけブロックしてあとは許可する。アプリケーションの特定のセキュリティニーズに基づいて独自の署名ルールを追加して、独自のカスタマイズされたセキュリティソリューションを設計できます。
- ハイブリッドセキュリティモデル: シグネチャを使用するだけでなく、ポジティブセキュリティチェックを使用して、アプリケーションに最適な構成を作成できます。署名を使用して不要なものをブロックし、許可されているものを強制する場合は確実なセキュリティチェックを行います。

署名を使用してアプリケーションを保護するには、署名オブジェクトを使用するように 1 つ以上のプロファイルを設定する必要があります。ハイブリッドセキュリティ構成では、署名オブジェクトの SQL インジェクションとクロスサイトスクリプティングパターン、および SQL 変換ルールは、署名ルールだけでなく、署名オブジェクトを使用する Web App Firewall プロファイルで設定されたポジティブセキュリティチェックでも使用されます。

Web App Firewall は、保護対象の Web サイトや Web サービスへのトラフィックを調べ、署名と一致するトラフィックを検出します。一致は、ルール内のすべてのパターンがトラフィックに一致する場合にのみトリガーされます。一致が発生すると、ルールに対して指定されたアクションが呼び出されます。リクエストがブロックされたときに、エラーページまたはエラーオブジェクトを表示できます。ログメッセージは、アプリケーションに対して実行されている攻撃を特定するのに役立ちます。統計を有効にすると、Web App Firewall は Web App Firewall の署名またはセキュリティチェックと一致するリクエストに関するデータを保持します。

トラフィックがシグニチャとポジティブセキュリティチェックの両方に一致する場合、2 つのアクションのうち、より厳しい制限が適用されます。たとえば、ブロックアクションが無効になっているシグネチャルールにリクエストが一致し、アクションがブロックされている SQL Injection ポジティブセキュリティチェックにも一致する場合、リクエストはブロックされます。この場合、リクエストは SQL インジェクションチェックによってブロックされても、署名違反が<not blocked>としてログに記録されることがあります。

カスタマイズ: 必要に応じて、独自のルールをシグニチャオブジェクトに追加できます。SQL/クロスサイトスクリプティングパターンをカスタマイズすることもできます。アプリケーションの特定のセキュリティニーズに基づいて独

自のシグニチャルールを追加するオプションにより、独自のカスタムセキュリティソリューションを柔軟に設計できます。望まないものだけブロックしてあとは許可する。特定の高速一致パターンを指定の場所に配置すると、処理オーバーヘッドを大幅に削減してパフォーマンスを最適化できます。SQL インジェクションとクロスサイトスクリプティングパターンを追加、変更、または削除できます。組み込みの正規表現エディターとエクスプレッションエディターにより、パターンを設定してその正確性を検証できます。

自動更新: 署名オブジェクトを手動で更新して最新の署名ルールを取得することも、自動更新機能を適用して Web App Firewall がクラウドベースの Web App Firewall 更新サービスから署名を自動的に更新できるようにすることもできます。

> 注:
>
> 自動更新中に新しいシグニチャルールが追加された場合、それらはデフォルトで無効になります。更新されたシグネチャを定期的に確認し、アプリケーションの保護に関連する新しく追加されたルールを有効にする必要があります。
>
> IIS サーバーで署名をホストするように CORS を設定する必要があります。
>
> NetScaler GUI から URL にアクセスすると、シグネチャ自動更新機能がローカル Web サーバーで動作しません。

はじめに

Citrix の署名を使用してアプリケーションを保護するのは簡単で、いくつかの簡単な手順で実行できます。

1. 署名オブジェクトを追加します。

- ウィザードを使用すると、プロファイルとポリシーの追加、署名の選択と有効化、署名とポジティブ・セキュリティ・チェックのアクションの指定など、Web App Firewall 構成全体を作成するように求めるプロンプトが表示されます。署名オブジェクトは自動的に作成されます。
- *Default Signatures テンプレートから署名オブジェクトのコピーを作成したり、空白のテンプレートを使用して独自のカスタマイズルールで署名を作成したり、外部形式の署名を追加したりできます。ルールを有効にして、適用するアクションを設定します。

1. このシグニチャオブジェクトを使用するように、ターゲット Web App Firewall プロファイルを設定します。
2. トラフィックを送信して機能を検証する

ハイライト

- デフォルト署名オブジェクトはテンプレートです。編集や削除はできません。これを使用するには、コピーを作成する必要があります。独自のコピーで、アプリケーションに必要なルールと各ルールの必要なアクションを有効にできます。アプリケーションを保護するには、この署名を使用するようにターゲットプロファイルを設定する必要があります。

- 署名パターンの処理にはオーバーヘッドがあります。すべてのシグニチャルールを有効にするのではなく、アプリケーションの保護に該当するシグニチャのみを有効にするようにしてください。
- シグネチャマッチをトリガーするには、ルール内のすべてのパターンが一致する必要があります。
- 独自にカスタマイズしたルールを追加して受信リクエストを検査し、SQL インジェクションやクロスサイトスクリプティング攻撃など、さまざまな種類の攻撃を検出できます。また、回答を検査するルールを追加して、クレジットカード番号などの機密情報の漏洩を検出し、ブロックすることもできます。
- 既存の署名オブジェクトのコピーを作成し、ルールや SQL/クロスサイトスクリプティングパターンを追加または編集して微調整することで、別のアプリケーションを保護できます。
- 自動更新を使用すると、Web App Firewall のデフォルトルールの最新バージョンをダウンロードできます。新しいアップデートの有無を継続的に監視する必要はありません。
- 署名オブジェクトは複数のプロファイルで使用できます。署名オブジェクトを使用するように 1 つ以上のプロファイルを設定した後でも、署名を有効または無効にしたり、アクション設定を変更したりできます。独自のカスタム署名ルールを手動で作成および変更できます。変更は、この署名オブジェクトを使用するように現在設定されているすべてのプロファイルに適用されます。
- シグネチャを設定して、HTML、XML、JSON、GWT などのさまざまなタイプのペイロードの違反を検出できます。
- 構成済みの署名オブジェクトをエクスポートして別の NetScaler アプライアンスにインポートすると、カスタマイズした署名ルールを簡単に複製できます。

シグネチャは、既知の脆弱性に関連するパターンです。署名保護を使用すると、これらの脆弱性を悪用しようとするトラフィックを特定し、具体的な対策を講じることができます。

署名はカテゴリ別に整理されています。アプリケーションの保護に適したカテゴリ内のルールのみを有効にすることで、パフォーマンスを最適化し、処理のオーバーヘッドを減らすことができます。

シグニチャ機能の手動設定

August 15, 2023

署名を使用して Web サイトを保護するには、ルールを確認し、適用するルールを有効にして設定する必要があります。ルールはデフォルトでは無効になっています。Web サイトで使用するコンテンツの種類に適用できるすべてのルールを有効にすることをお勧めします。

シグニチャ機能を手動で設定するには、ブラウザを使用して GUI に接続します。次に、組み込みテンプレート、既存の署名オブジェクト、またはファイルをインポートして、署名オブジェクトを作成します。次に、シグニチャオブジェクトの設定または変更の説明に従って、新しいシグニチャオブジェクトを設定します。

署名オブジェクトの追加と削除

August 15, 2023

Web App Firewall に新しい署名オブジェクトを追加するには、次の操作を行います。

- 組み込みテンプレートをコピーする。
- 既存のシグニチャオブジェクトをコピーする。
- 外部ファイルからシグニチャオブジェクトをインポートする。

シグニチャファイルには、CPU 使用率、最新の適用年、重大度の詳細が含まれます。シグニチャファイルが定期的に変更およびアップロードされるたびに、CPU 使用率、最新の年、および CVE 重大度を確認できます。これらの値を確認したら、アプライアンスのシグニチャを有効にするか無効にするかを決定できます。

テンプレートまたは既存のシグニチャオブジェクトをコピーするには、GUI を使用する必要があります。GUI またはコマンドラインを使用して、シグニチャオブジェクトをインポートできます。GUI またはコマンドラインを使用して、シグニチャオブジェクトを削除することもできます。

テンプレートからシグニチャオブジェクトを作成するには

1. **Security > NetScaler Web App Firewall > Signatures** に移動します。
2. 詳細ウィンドウで、テンプレートとして使用するシグニチャオブジェクトを選択します。

   選択肢は次のとおりです:

   - デフォルトのシグニチャ。シグニチャルール、SQL インジェクションルール、クロスサイトスクリプティングルールが含まれます。
   - **XPath** インジェクション。XPath インジェクションパターンが含まれます。
   - 既存のシグニチャオブジェクト。

   > 注意**:**
   >
   > テンプレートとして使用するシグニチャタイプを選択しない場合、Web App Firewall はシグニチャを最初から作成するように求めるプロンプトを表示します。

3. [追加] をクリックします。
4. [シグニチャオブジェクトの追加] ダイアログボックスで、新しいシグニチャオブジェクトの名前を入力し、[OK] をクリックします。名前は、英字、数字、またはアンダースコア記号で始まり、1 ～31 個の英数字、およびハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、およびアンダースコア (_) 記号で構成できます。
5. [閉じる] をクリックします。

ファイルをインポートしてシグニチャオブジェクトを作成するには

1. **Security > NetScaler Web App Firewall > Signatures** に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. シグニチャオブジェクトの追加（**Add Signatures Object** ）ダイアログボックスで、インポートするシグニチャの形式を選択します。
   - NetScaler ADC 形式の署名ファイルをインポートするには、［ネイティブ形式］タブを選択します。
   - 外部シグニチャ形式ファイルを読み込むには、[外部形式] タブを選択します。
4. シグニチャオブジェクトの作成に使用するファイルを選択します。
   - ネイティブの NetScaler ADC 形式の署名ファイルをインポートするには、［インポート］セクションで［ローカルファイルからインポート］または［URL からインポート］を選択して、ファイルへのパスまたは URL を入力または参照します。
   - Cenzic、IBM AppScan、Qualys、または Whitehat 形式のファイルをインポートするには、「XSLT」セクションで「組み込み XSLT ファイルを使用」、「ローカルファイルを使用」、または「URL から参照」を選択します。次に、[組み込み XSLT ファイルを使用] を選択した場合は、リストから適切なファイル形式を選択します。[ローカルファイルを使用] または [URL から参照] を選択した場合は、ファイルへのパスまたは URL を入力するか、参照して選択します。
5. [追加] をクリックし、[閉じる] をクリックします。

コマンドラインを使用してファイルをインポートしてシグニチャオブジェクトを作成するには

コマンドプロンプトで、次のコマンドを入力します。

- `import appfw signatures <src> <name> [-xslt <string>] [-comment <string>] [-overwrite] [-merge] [-sha1 <string>]`
- `save ns config`

例 **#1**

次の例では、signatures.xml という名前のファイルから署名オブジェクトを作成し、MySignatures という名前を割り当てます。

```
import appfw signatures local:signatures.xml MySignatures
save ns config
<!--NeedCopy-->
```


CLI を使用して個々の署名を追加するには

署名を ID またはカテゴリで選択し、アクションを設定できます。コマンドプロンプトで、次のコマンドを実行します。

```
import appfw signature <source> <name> [-sigRuleId| -sigCategory] [Rule
    -IDs | Category name] -Enabled [ON | OFF] [-Action LOG BLOCK]
<!--NeedCopy-->
```

- シグネチャー **ID** の使用例

  次の例では、シグニチャをルール ID で有効にし、ログアクションとブロックアクションを設定します。

  ```
  import appfw signature DEFAULT object_name -sigRuleId 1001 9882
      2000 1250 810 -Enabled ON -Action LOG BLOCK
  <!--NeedCopy-->
  ```

  次の例では、署名を有効化せずに ID で署名を追加しています。

  ```
  import appfw signature DEFAULT object_name -sigRuleId 810 -
      Enabled OFF
  <!--NeedCopy-->
  ```

- 署名カテゴリの使用例

  次の例では、web-misc カテゴリ別の署名を有効にし、ログアクションとブロックアクションを設定します。

  ```
  import appfw signature DEFAULT object_name -sigCategory web-misc
      -Enabled ON -Action LOG BLOCK
  <!--NeedCopy-->
  ```

  次の例では、web-misc 有効化せずにカテゴリ別に署名を追加します。

  ```
  import appfw signature DEFAULT object_name -sigCategory web-misc
      -Enabled OFF
  <!--NeedCopy-->
  ```

GUI を使用してシグニチャオブジェクトを削除するには

1. **Security > NetScaler Web App Firewall > Signatures** に移動します。
2. 詳細ペインで、削除するシグニチャオブジェクトを選択します。
3. [削除] をクリックします。

コマンドラインを使用してシグニチャオブジェクトを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- rm appfw signatures <name>

- save ns config

署名オブジェクトの設定または変更

March 20, 2024

署名オブジェクトを作成した後に構成するか、既存の署名オブジェクトを変更して署名カテゴリまたは特定の署名を有効または無効にしたり、署名が接続と一致したときの Web App Firewall の応答方法を設定します。

シグニチャオブジェクトを設定または変更するには

1. **Security > NetScaler Web App Firewall > Signatures** に移動します。
2. 詳細ペインで、設定するシグニチャオブジェクトを選択し、「開く」をクリックします。
3. 「署名オブジェクトの変更」ダイアログで、左側の「フィルター条件を表示」オプションを設定して、設定するフィルター項目を表示します。

   これらのオプションを変更すると、要求した結果が右側の「フィルター結果」ウィンドウに表示されます。

   - 選択したカテゴリの署名のみを表示するには、該当する署名カテゴリのチェックボックスをオンまたはオフにします。リリース 13.1 ビルド 48.x 以降、左側のパネルの CVE を使用して、選択した年に公開された脆弱性を表示できます。
     署名のカテゴリは次のとおりです。

名前	このシグニチャが保護する攻撃のタイプ
cgi	CGI スクリプト。Perl と UNIX のシェルスクリプトが含まれています。
クライアント	ブラウザとその他のクライアント
coldfusion	アドビシステムズの ColdFusion アプリケーションサーバーを使用する Web サイト。
frontpage	Microsoft の FrontPage サーバーを使用するウェブサイト。
IIS	Microsoft インターネットインフォメーションサーバー (IIS) を使用する Web サイト。
その他	その他の攻撃。
php	PHP を使用するウェブサイト
web-activex	ActiveX コントロールを含む Web サイト

名前	このシグニチャが保護する攻撃のタイプ
web-struts	java-ee ベースのアプレットである Apache ストラットを含むウェブサイト。
CVE	選択した年に公開された CVE を一覧表示します。

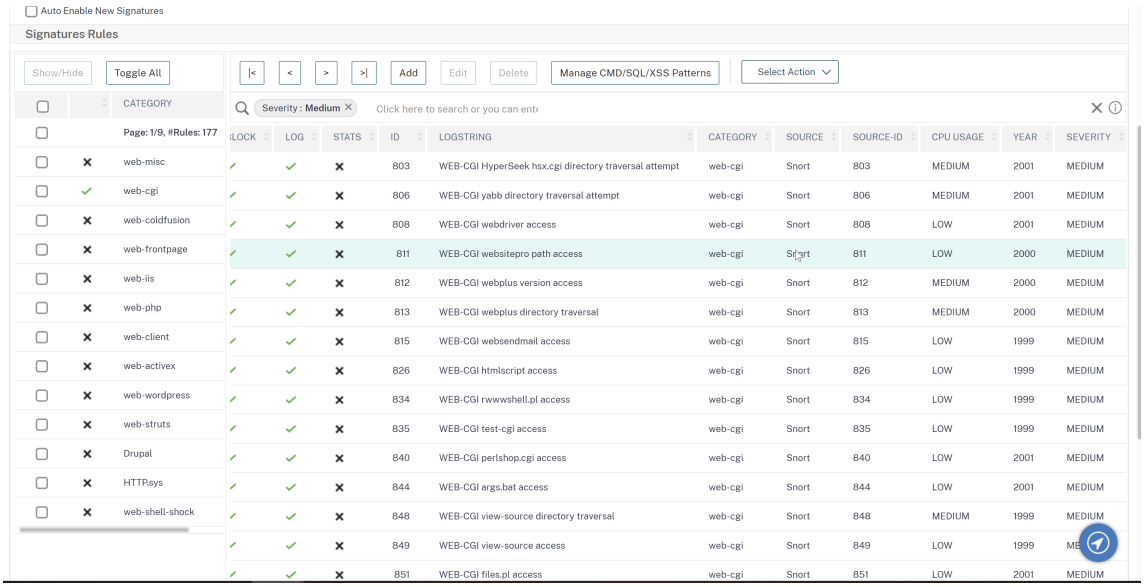
- 特定のチェックアクションが有効になっているシグニチャのみを表示するには、それらのアクションごとに [オン] チェックボックスをオンにし、他のアクションの [オン] チェックボックスをオフにし、[オフ] チェックボックスをすべてオフにします。特定のチェックアクションが無効になっているシグニチャのみを表示するには、それぞれの「オフ」チェックボックスを選択し、「オン」のチェックボックスをすべてオフにします。チェックアクションが有効か無効かに関係なく署名を表示するには、そのアクションの「オン」チェックボックスと「オフ」チェックボックスの両方をオンまたはオフにします。チェックアクションは以下のとおりです。

基準	説明
有効	署名は有効になっています。Web App Firewall は、トラフィックを処理するときに有効になっている署名のみをチェックします。
ブロック	このシグニチャと一致する接続はブロックされます。
ログ	このシグネチャに一致するすべての接続についてログエントリが生成されます。
統計情報	Web App Firewall は、このシグネチャと一致するすべての接続を、そのチェックで生成する統計情報に含めます。

- 結果ウィンドウに表示される詳細をさらに絞り込むには、結果ウィンドウの上にある検索バーを使用して次の手順を実行します:
  - 検索バーでフィルタリングするプロパティを選択します。
  - 値を入力し、Enter ボタンを押します。

  さらに、結果表示ウィンドウに既に表示されているコンテンツをフィルタリングし、入力された値に基づいて詳細を一覧表示します。

例: 次の画像では、左側の「表示フィルター基準」オプションのカテゴリとして Web-CGI が選択されています。Web-CGI 署名の詳細が右側の結果ウィンドウに表示されます。重要度に基づいて詳細をさらに絞り込むには、検索バーでプロパティとして重要度を選択し、値として「中」を入力します。重要度が中程度の Web-CGI シグニチャが結果ウィンドウに表示されます。



- すべての表示フィルタ条件をデフォルト設定にリセットしてすべてのシグニチャを表示するには、「すべて表示」をクリックします。

> 注:
>
> フィルタされた結果ウィンドウに表示される項目の数は 20 です。ページネーションは、左側の「フィルター条件を表示」オプションの上にあります。

1. 特定の署名の詳細については、署名を選択し、「その他」フィールドの青い二重矢印をクリックします。シグニチャルール脆弱性詳細メッセージボックスが表示されます。署名の目的に関する情報が含まれており、この署名が対処する 1 つまたは複数の脆弱性に関する外部の Web ベースの情報へのリンクも記載されています。外部リンクにアクセスするには、そのリンクの説明の左側にある青い二重矢印をクリックします。
2. 適切なチェックボックスをオンにして、署名の設定を構成します。
3. シグニチャオブジェクトにローカルシグニチャルールを追加する場合、または既存のローカルシグニチャルールを変更する場合は、シグネチャエディタ（SignaturesEditor）を参照してください。
4. SQL インジェクション、クロスサイトスクリプト、または Xpath インジェクションパターンが必要ない場合は、[OK] をクリックし、[閉じる] をクリックします。それ以外の場合は、詳細ペインの左下隅にある [SQL/クロスサイトスクリプティングパターンの管理] をクリックします。
5. 管理で SQL/cross-site [パターンのスクリプト] ダイアログボックスの [フィルターされた結果] ウィンドウで、構成するパターンのカテゴリとパターンに移動します。SQL インジェクションパターンの詳細については、HTML SQL インジェクションチェックを参照してください。クロスサイトスクリプティングパターンの詳細については、HTML クロスサイトスクリプティングチェックを参照してください。
6. 新しいパターンを追加するには:

   a) 新しいパターンを追加するブランチを選択します。
   b) [フィルター結果] ウィンドウの下部セクションのすぐ下にある [追加] ボタンをクリックします。

c) 署名アイテムの作成ダイアログボックスで、追加するパターンを要素テキストボックスに入力します。変換ルールブランチに変換パターンを追加する場合は、「要素」で、「From」テキストボックスに変更するパターンを入力し、「To」テキストボックスに変更前のパターンを変更するパターンを入力します。

d) [**OK**] をクリックします。

7. 既存のパターンを変更するには:

a) フィルター結果ウィンドウで、変更するパターンを含むブランチを選択します。

b) フィルター結果ウィンドウの下にある詳細ウィンドウで、変更するパターンを選択します。

c) [修正**]** をクリックします。

d) 署名項目の編集ダイアログボックスの「要素」テキストボックスで、パターンを変更します。変形パターンを変更する場合は、「要素」の「From」テキストボックスと「TO」テキストボックスで、どちらか一方または両方のパターンを変更できます。

e) [**OK**] をクリックします。

8. パターンを削除するには、削除するパターンを選択し、「フィルター結果」ウィンドウの下にある詳細ペインの下にある「削除」ボタンをクリックします。メッセージが表示されたら、[閉じる] をクリックして選択を確定します。

9. パターンカテゴリをクロスサイトスクリプティングブランチに追加するには:

a) パターンカテゴリを追加するブランチを選択します。

b) [フィルター結果] ウィンドウのすぐ下の [追加] ボタンをクリックします。

> 注**:** 現在、クロスサイトスクリプティングブランチに追加できるカテゴリは「patterns」という名前の 1 つだけです。そのため、「追加」をクリックした後は、デフォルトの選択である「patterns」をそのまま使用する必要があります。

c) [**OK**] をクリックします。

10. ブランチを削除するには、そのブランチを選択し、「フィルター結果」ウィンドウのすぐ下の「削除」ボタンをクリックします。メッセージが表示されたら、「**OK**」をクリックして選択を確定します。

> 注**:** デフォルトブランチを削除すると、そのブランチのすべてのパターンが削除されます。そうすることで、その情報を使用するセキュリティチェックを無効化できます。

11. SQL インジェクション、クロスサイトスクリプティング、および XPath インジェクションパターンの変更が終了したら、「**OK**」をクリックし、「閉じる」をクリックして「署名オブジェクトの変更」ダイアログ・ボックスに戻ります。

12. 任意の時点で OK をクリックして変更を保存し、シグニチャオブジェクトの設定が終了したら、「閉じる」をクリックします。

署名による **JSON** アプリケーションの保護

August 15, 2023

JavaScript オブジェクトノーテーション (JSON) は、JavaScript スクリプト言語から派生したテキストベースのオープンスタンダードです。JSON は、オブジェクトと呼ばれる単純なデータ構造や連想配列を人間が読める形式で表現する場合に適しています。XML の代替として機能し、主に Web アプリケーションとの通信用にシリアル化されたデータ構造を送信するために使用されます。JSON ファイルは通常、.json 拡張子を付けて保存されます。

JSON ペイロードは、通常、**application/json** として指定された MIME タイプで送信されます。JSON のその他の「標準」コンテンツタイプは次のとおりです。

- **application/x-javascript**
- **text/javascript**
- **text/x-javascript**
- **text/x-json**

NetScaler Web App Firewall 署名による **JSON** アプリケーションの保護

JSON リクエストを許可するために、次の show-command 出力に示すように、アプライアンスには JSON コンテンツタイプがあらかじめ設定されています。

```
> sh appfw jsonContentType
1)      JSONContenttypevalue:  "^application/json$" IsRegex:  REGEX
Done
<!--NeedCopy-->
```

NetScaler Web App Firewall は、次のコンテンツタイプの投稿本文のみを処理します。

- アプリケーション**/x-www-**フォーム**-URL** エンコード
- **multipart/form-data**
- テキスト**/x-gwt-rpc**

application/json（またはその他の許可されたコンテンツタイプ）を含む他のコンテンツタイプのヘッダーで受信されたリクエストは、ヘッダー検査後にバックエンドに転送されます。このようなリクエストの投稿本文は、SQL やクロスサイトスクリプティングなどのプロファイルのセキュリティチェックが有効になっていても、セキュリティチェック違反の検査は行われません。

JSON アプリケーションを保護して違反を検出するには、Web App Firewall シグネチャを使用できます。許可されたコンテンツタイプヘッダーを含むすべてのリクエストは、Web App Firewall によって処理され、署名が照合されます。独自にカスタマイズした署名ルールを追加して JSON ペイロードを処理し、さまざまなセキュリティチェック検査 (クロスサイトスクリプティング、SQL、フィールドの一貫性など) を実行したり、ヘッダーや投稿本文の違反を検出し、特定のアクションを実行したりできます。

> ヒント
>
> 他の組み込みデフォルトとは異なり、事前設定された JSON コンテンツタイプは CLI または GUI (GUI) を使用して編集または削除できます。JSON アプリケーションに対する正当なリクエストがブロックされ、コンテンツタイプ違反が発生する場合は、コンテンツタイプの値が正確に設定されていることを確認します。Web App Firewall がコンテンツタイプヘッダーを処理する方法の詳細については、「コンテンツタイプの保護」を参照してください。

コマンドラインインターフェイスを使用して **JSON** コンテンツタイプを追加または削除するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
add appfw jsonContentType ^application/json$ IsRegEx REGEX
rm appfw JSONContentType "^application/json$"
```

GUI を使用して **JSON** コンテンツタイプを管理するには

[セキュリティ] > [**Web App Firewall**] に移動し、[設定] セクションで [**JSON** コンテンツタイプの管理] を選択します。

Web App Firewall の **JSON** コンテンツタイプの設定パネルで、アプリケーションのニーズに合わせて JSON コンテンツタイプを追加、編集、または削除します。

JSON ペイロード内の攻撃を検出するための署名保護の設定

有効な JSON コンテンツタイプに加えて、JSON リクエストで検出されたときにセキュリティ違反であることを示すパターンを指定する署名を設定する必要があります。ブロックやログなどの指定されたアクションは、受信リクエストによってシグニチャルール内のすべてのターゲットパターンが一致したときに実行されます。

カスタマイズされた署名ルールを追加するには、Citrix では GUI を使用することをお勧めします。[システム] > [セキュリティ] > [**Web App Firewall**] > [署名] に移動します。ターゲットシグニチャオブジェクトをダブルクリックして、[**Web App Firewall Signatures**] パネルにアクセスします。[追加] ボタンをクリックして、アクション、カテゴリ、ログ文字列、ルールパターンなどを設定します。Web App Firewall は、許可されているすべてのコンテンツタイプのペイロードの署名一致を検査しますが、ルールに JSON 式を指定することで処理を最適化できます。新しいルールパターンを追加するときは、**Match** のドロップダウンオプションで **Expression** を選択し、JSON ペイロードからターゲットマッチ式を入力して、検査が必要な特定のリクエストを特定します。式は **TEXT** で始まる必要があります。プレフィックス。他のルールパターンを追加して、攻撃を特定するための追加のマッチパターンを指定できます。

次の例は、署名ルールを示しています。JSON ペイロードの POST 本文で、指定した XPATH_JSON 式と一致するクロスサイトスクリプトタグが検出されると、シグネチャマッチがトリガーされます。

JSON ペイロードのクロスサイトスクリプティングを検出する署名の例

```
<SignatureRule actions="log,stats" category="JSON" enabled="ON" id="
    1000001" severity="" source="" type="" version="1">

  <PatternList>

    <RequestPatterns>

      <Pattern>

        <Location area="HTTP_POST_BODY"/>

        <Match type="Expression">TEXT.XPATH_JSON(xp%/glossary/title%).
            CONTAINS("example glossary")</Match>

      </Pattern>

      <Pattern>

        <Location area="HTTP_METHOD"/>

        <Match type="LITERAL">POST</Match>

      </Pattern>

      <Pattern>

        <Location area="HTTP_POST_BODY"/>

        <Match type="CrossSiteScripting"/>

       </Pattern>

    </RequestPatterns>

  </PatternList>

  <LogString>Cross-site scripting violation detected in json payload</
      LogString>

  <Comment/>

</SignatureRule>
<!--NeedCopy-->
```

ペイロードの例

次のペイロードは、クロスサイトスクリプティングタグ **<Gotcha!!>** が含まれているため、シグニチャの一致をトリガーします。を選択します。


```
{
 "glossary": {
 "title": "example glossary","GlossDiv": {
 "title": "S","GlossList": {
 "GlossEntry": {
 "ID": "SGML","SortAs": "SGML","GlossTerm": "Standard Generalized
    Markup Language","Acronym": "SGML","Abbrev": "ISO 8879:1986","
    GlossDef": {
 "para": "A meta-markup language, used to create markup languages \*\*<
    Gotcha!!>\*\* such as DocBook.","GlossSeeAlso": ["GML", "XML"] }
 ,"GlossSee": "markup" }
  }
  }
  }
  }

<!--NeedCopy-->
```

ログメッセージの例

```
Aug 21 12:21:42 <local0.info> 10.217.31.239 08/21/2015:23:21:42 GMT ns
    0-PPE-1 : APPFW APPFW_SIGNATURE_MATCH 1471 0 :  10.217.253.62 990-
    PPE0 NtJnVMNnvPeQJnaUzXYW/GTvAQsA010 prof1 http://10.217.31.212/FFC/
    login_post.php Signature violation rule ID 1000001: cross-site
    scripting violation detected in json payload  <not blocked>
<!--NeedCopy-->
```

> 注
>
> クロスサイトスクリプトタグ（**<Gotcha!! >**）、シグネチャルールの一致はトリガーされません。

ハイライト

- JSON ペイロードを保護するには、Web App Firewall シグネチャを使用してクロスサイトスクリプティング、SQL、その他の違反を検出します。
- JSON コンテンツタイプがアプライアンスで許可されているコンテンツタイプとして設定されていることを確認します。
- ペイロードのコンテンツタイプが、設定された JSON コンテンツタイプと一致していることを確認してください。
- シグニチャ違反が発生するには、シグニチャルールで設定されたすべてのパターンが一致していることを確認してください。
- シグニチャルールを追加する場合、JSON ペイロードの Expression と一致するルールパターンが少なくとも 1 つ必要です。シグニチャルールの PI 式はすべて、プレフィックス TEXT. で始まり、ブール値でなければなりません。

ポリシーとシグネチャを使用して、アプリケーションまたは **JSON** コンテンツタイプを **SQL** とクロスサイトスクリプティングでエンコードされたペイロードで保護します

NetScaler Web App Firewall は、ポリシーと署名を使用してアプリケーションまたは JSON コンテンツタイプを保護できます。

ポリシーを使用してアプリケーションまたは **JSON** コンテンツタイプの **SQL** インジェクションを検査する

SQL インジェクションをサポートするために、次のポリシーを追加し、仮想サーバーにグローバルにバインドする必要があります。

```
add appfw policy sqli_1 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE
).SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re
#((((\\A)|(?<=[^a-zA-Z0-9_])))(select|insert|delete|update|drop|
create|alter|grant|revoke|commit|rollback|shutdown|union|intersect
|minus|case|decode|where|group|begin|join|exists|distinct|add|modify|
constraint|null|like|exec|execute|char|or|and|sp_sdidebug)((Z)|(?=[^a
-zA-Z0-9_]))#)APPFW_BLOCK

add appfw policy sqli_2 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE
).SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re
#((\\A)|(?<=[^a-zA-Z0-9_]))(xp_availablemedia|xp_cmdshell|xp_deletemail
|xp_dirtree|xp_dropwebtask|xp_dsninfo|xp_enumdsn|xp_enumerrorlogs|
xp_enumgroups|xp_enumqueuedtasks|xp_eventlog|xp_findnextmsg|xp_fixeddrives
|xp_getfiledetails|xp_getnetname|xp_grantlogin|xp_logevent|xp_loginconfig
|xp_logininfo|xp_makewebtask|xp_msver|xp_regread|xp_perfend|xp_perfmonitor
|xp_perfsample|xp_perfstart|xp_readerrorlog|xp_readmail|xp_revokelogin
|xp_runwebtask|xp_schedulersignal|xp_sendmail|xp_servicecontrol
|xp_snmp_getstate|xp_snmp_raisetrap|xp_sprintf|xp_sqlinventory|
xp_sqlregister|xp_sqltrace|xp_sscanf|xp_startmail|xp_stopmail|xp_subdirs
|xp_unc_to_drive)((Z)|(?=[^a-zA-Z0-9_]))#)APPFW_BLOCK

add appfw policy sqli_3 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE
).SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re
#((\\A)|(?<=[^a-zA-Z0-9_]))(sysobjects|syscolumns|MSysACEs|MSysObjects
|MSysQueries|MSysRelationships)((Z)|(?=[^a-zA-Z0-9_]))#)APPFW_BLOCK

add appfw policy sqli_4 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE
).SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re
#((\\A)|(?<=[^a-zA-Z0-9_]))(SYS\.USER_OBJECTS|SYS\.TAB|SYS\.USER_TABLES
|SYS\.USER_VIEWS|SYS\.ALL_TABLES|SYS\.USER_TAB_COLUMNS|SYS\.USER_CONSTRAINTS
|SYS\.USER_TRIGGERS|SYS\.USER_CATALOG|SYS\.ALL_CATALOG|SYS\.ALL_CONSTRAINTS
```



```
|SYS\.ALL_OBJECTS|SYS\.ALL_TAB_COLUMNS|SYS\.ALL_TAB_PRIVS|SYS\.
ALL_TRIGGERS|SYS\.ALL_USERS|SYS\.ALL_VIEWS|SYS\.USER_ROLE_PRIVS|
SYS\.USER_SYS_PRIVS|SYS\.USER_TAB_PRIVS)((Z)|(?=[^a-zA-Z0-9_]))#)
APPFW_BLOCK
```

署名を使用してアプリケーションまたは **JSON** コンテンツタイプを検査する

JSON content-type の SQL インジェクションをサポートするために、アプリケーションファイアウォールプロファイルの署名オブジェクトに次の署名ルールを追加できます。

> 注:
>
> ポストボディシグネチャは CPU を大量に消費します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Copyright 2013-2018 Citrix Systems, Inc. All rights reserved. -->
<SignaturesFile schema_version="6" version="0" minor_schema_version="0"
    >
     <Signatures>
         <SignatureRule id="4000000" enabled="ON" actions="log,block"
            category="sql" source="" severity="" type="" version="1"
            sourceid="" harmscore="">
             <PatternList>
                 <RequestPatterns>
                     <Pattern>
                         <Location area="HTTP_POST_BODY"/>
                         <Match type="Expression">TEXT.SET_TEXT_MODE(
                            IGNORECASE).SET_TEXT_MODE(URLENCODED).
                            DECODE_USING_TEXT_MODE.REGEX_MATCH(re#(((\A)
                            |(?<=[^a-zA-Z0-9_])))(select|insert|delete|
                            update|drop|create|alter|grant|revoke|commit
                            |rollback|shutdown|union|intersect|minus|
                            case|decode|where|group|begin|join|exists|
                            distinct|add|modify|constraint|null|like|
                            exec|execute|char|or|and|sp_sdidebug)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                     </Pattern>
                     <Pattern type="fastmatch">
                         <Location area="HTTP_METHOD"/>
                         <Match type="LITERAL">T</Match>
                     </Pattern>
                 </RequestPatterns>
             </PatternList>
             <LogString>sql Injection</LogString>
             <Comment/>
         </SignatureRule>
         <SignatureRule id="4000001" enabled="ON" actions="log,block"
            category="sql" source="" severity="" type="" version="1"
            sourceid="" harmscore="">
             <PatternList>
```



```
                    <RequestPatterns>
                        <Pattern>
                            <Location area="HTTP_POST_BODY"/>
                            <Match type="Expression">TEXT.SET_TEXT_MODE(
                                IGNORECASE).SET_TEXT_MODE(URLENCODED).
                                DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\A)
                                |(?<=[^a-zA-Z0-9_]))(xp_availablemedia|
                                xp_cmdshell|xp_deletemail|xp_dirtree|
                                xp_dropwebtask|xp_dsninfo|xp_enumdsn|
                                xp_enumerrorlogs|xp_enumgroups|
                                xp_enumqueuedtasks|xp_eventlog|
                                xp_findnextmsg|xp_fixeddrives|
                                xp_getfiledetails|xp_getnetname|
                                xp_grantlogin|xp_logevent|xp_loginconfig|
                                xp_logininfo|xp_makewebtask|xp_msver|
                                xp_regread|xp_perfend|xp_perfmonitor|
                                xp_perfsample|xp_perfstart|xp_readerrorlog|
                                xp_readmail|xp_revokelogin|xp_runwebtask|
                                xp_schedulersignal|xp_sendmail|
                                xp_servicecontrol|xp_snmp_getstate|
                                xp_snmp_raisetrap|xp_sprintf|xp_sqlinventory
                                |xp_sqlregister|xp_sqltrace|xp_sscanf|
                                xp_startmail|xp_stopmail|xp_subdirs|
                                xp_unc_to_drive)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                        </Pattern>
                        <Pattern type="fastmatch">
                            <Location area="HTTP_METHOD"/>
                            <Match type="LITERAL">T</Match>
                        </Pattern>
                    </RequestPatterns>
                </PatternList>
                <LogString>sql Injection</LogString>
                <Comment/>
            </SignatureRule>
            <SignatureRule id="4000002" enabled="ON" actions="log,block"
                category="sql" source="" severity="" type="" version="1"
                sourceid="" harmscore="">
                <PatternList>
                    <RequestPatterns>
                        <Pattern>
                            <Location area="HTTP_POST_BODY"/>
                            <Match type="Expression">TEXT.SET_TEXT_MODE(
                                IGNORECASE).SET_TEXT_MODE(URLENCODED).
                                DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\A)
                                |(?<=[^a-zA-Z0-9_]))(sysobjects|syscolumns|
                                MSysACEs|MSysObjects|MSysQueries|
                                MSysRelationships)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                        </Pattern>
                        <Pattern type="fastmatch">
                            <Location area="HTTP_METHOD"/>
                            <Match type="LITERAL">T</Match>
```



```
                    </Pattern>
                </RequestPatterns>
            </PatternList>
            <LogString>sql Injection</LogString>
            <Comment/>
        </SignatureRule>
        <SignatureRule id="4000003" enabled="ON" actions="log,block"
           category="sql" source="" severity="" type="" version="1"
           sourceid="" harmscore="">
            <PatternList>
                <RequestPatterns>
                    <Pattern>
                        <Location area="HTTP_POST_BODY"/>
                        <Match type="Expression">TEXT.SET_TEXT_MODE(
                           IGNORECASE).SET_TEXT_MODE(URLENCODED).
                           DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\A)
                           |(?<=[^a-zA-Z0-9_]))(SYS.USER_OBJECTS|SYS.
                           TAB|SYS.USER_TABLES|SYS.USER_VIEWS|SYS.
                           ALL_TABLES|SYS.USER_TAB_COLUMNS|SYS.
                           USER_CONSTRAINTS|SYS.USER_TRIGGERS|SYS.
                           USER_CATALOG|SYS.ALL_CATALOG|SYS.
                           ALL_CONSTRAINTS|SYS.ALL_OBJECTS|SYS.
                           ALL_TAB_COLUMNS|SYS.ALL_TAB_PRIVS|SYS.
                           ALL_TRIGGERS|SYS.ALL_USERS|SYS.ALL_VIEWS|SYS
                           .USER_ROLE_PRIVS|SYS.USER_SYS_PRIVS|SYS.
                           USER_TAB_PRIVS)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                    </Pattern>
                    <Pattern type="fastmatch">
                        <Location area="HTTP_METHOD"/>
                        <Match type="LITERAL">T</Match>
                    </Pattern>
                </RequestPatterns>
            </PatternList>
            <LogString>sql Injection</LogString>
            <Comment/>
        </SignatureRule>
    </Signatures>
</SignaturesFile>

<!--NeedCopy-->
```

署名オブジェクトの更新

August 15, 2023

Web App Firewall が現在の脅威から確実に保護できるように、シグネチャオブジェクトを頻繁に更新する必要があります。デフォルトの Web App Firewall 署名と、サポートされている脆弱性スキャンツールからインポートした署名の両方を定期的に更新する必要があります。

NetScaler は、Web App Firewall デフォルト署名を定期的に更新します。デフォルトのシグニチャは手動または自動で更新できます。いずれの場合も、アップデートにアクセスするための URL については、NetScaler の担当者または NetScaler リセラーに問い合わせてください。NetScaler ネイティブ形式の署名の自動更新は、［エンジン設定］ダイアログボックスと［署名自動更新設定］ダイアログボックスで有効にできます。

脆弱性スキャンツールのほとんどのメーカーは、定期的にツールを更新しています。ほとんどのウェブサイトも頻繁に変更されます。ツールを更新してウェブサイトを定期的に再スキャンし、生成された署名をファイルにエクスポートして、Web App Firewall の構成にインポートする必要があります。

> ヒント
>
> NetScaler コマンドラインから Web App Firewall の署名を更新する場合は、まずデフォルトの署名を更新してから、さらに更新コマンドを実行して、デフォルトの署名に基づく各カスタム署名ファイルを更新する必要があります。最初にデフォルトのシグニチャを更新しないと、バージョンの不一致エラーによりカスタムシグニチャファイルの更新が妨げられます。
>
> 注
>
> 以下は、サードパーティの署名オブジェクトを、ネイティブルールおよびユーザーが追加したルールを持つユーザー定義の署名オブジェクトとマージする場合に適用されます。
>
> バージョン 0 のシグネチャを新しいインポートされたファイルとマージしても、生成されたシグネチャはバージョン 0 のままです。
>
> つまり、インポートされたファイル内のすべてのネイティブ (またはビルトイン) ルールは、マージ後に無視されます。これは、バージョン 0 のシグネチャがマージ後もそのまま維持されるようにするためです。
>
> インポートしたファイルにネイティブ・ルールをマージ対象に含めるには、マージの前にまず既存のシグニチャをバージョン 0 から更新する必要があります。つまり、既存のシグニチャのバージョン 0 の性質を捨てる必要があるということです。
>
> NetScaler リリースのアップグレードが行われると、ファイル「default_signatures.xml」が新しいビルドに追加され、ファイル「updated_signature.xml」が古いビルドから削除されます。アップグレード後、署名の自動更新機能が有効になっている場合、アプライアンスは既存の署名を最新バージョンのビルドに更新し、「updated_signature.xml」ファイルを生成します。

コマンドラインを使用して **Web App Firewall** の署名をソースから更新するには

コマンドプロンプトで、次のコマンドを入力します。

- `update appfw signatures <name> [-mergedefault]`
- `save ns config`

例

次の例では、MySignatures という名前の署名オブジェクトを既定の署名オブジェクトから更新し、既定の署名オブジェクト内の新しい署名を既存の署名とマージします。このコマンドは、ユーザーが作成した署名や、承認された脆弱性スキャンツールなどの別のソースからインポートされた署名を上書きしません。

```
update appfw signatures MySignatures -mergedefault
save ns config
<!--NeedCopy-->
```

NetScaler 形式ファイルからの署名オブジェクトの更新

NetScaler は Web App Firewall ールの署名を定期的に更新します。Web App Firewall が最新のリストを使用するように、Web App Firewall のシグネチャを定期的に更新する必要があります。アップデートにアクセスするための URL については、NetScaler の担当者または NetScaler リセラーに問い合わせてください。

コマンドラインを使用して **NetScaler** 形式ファイルから署名オブジェクトを更新するには

コマンドプロンプトで、次のコマンドを入力します。

- `update appfw signatures <name> [-mergeDefault]`
- `save ns config`

GUI を使用して **NetScaler** 形式のファイルから署名オブジェクトを更新するには

1. [セキュリティ] > [**Web App Firewall**] > [署名] に移動します。
2. 詳細ペインで、更新するシグニチャオブジェクトを選択します。
3.「アクション」ドロップダウンリストで、「マージ」を選択します。
4.「署名オブジェクトの更新」ダイアログボックスで、次のオプションのいずれかを選択します。
   - **URL** からインポート: Web URL からシグニチャアップデートをダウンロードする場合は、このオプションを選択します。
   - ローカルファイルからインポート-ローカルハードドライブ、ネットワークハードドライブ、またはその他のストレージデバイス上のファイルからシグニチャアップデートをインポートする場合は、このオプションを選択します。
5. テキスト領域に URL を入力するか、ローカルファイルを入力または参照します。
6. [**Update**] をクリックします。更新ファイルがインポートされ、「署名の更新」ダイアログ・ボックスが「署名オブジェクトの変更」ダイアログ・ボックスとほぼ同じ形式に変わります。[**Update Signatures Object**] ダイアログボックスには、新規または変更されたシグニチャ規則、SQL インジェクションまたはクロスサイトスクリプティングパターン、および XPath インジェクションパターン（存在する場合）を含むすべてのブランチが表示されます。

7. 新しいシグニチャと変更されたシグニチャを確認して設定します。
8. 終了したら、「**OK**」をクリックし、「閉じる」をクリックします。

サポートされている脆弱性スキャンツールからのシグネチャオブジェクトの更新

> 注:
>
> ファイルからシグニチャオブジェクトを更新する前に、脆弱性スキャンツールからシグニチャをエクスポートしてファイルを作成する必要があります。

脆弱性スキャンツールから署名をインポートして更新するには

1. [セキュリティ] > [**Web App Firewall**] > [署名] に移動します。
2. 詳細ペインで、更新するシグニチャオブジェクトを選択し、「マージ」をクリックします。
3. 「署名オブジェクトの更新」ダイアログボックスの「外部形式」タブの「インポート」セクションで、次のオプションのいずれかを選択します。
   - **URL** からインポート： Web URL からシグニチャアップデートをダウンロードする場合は、このオプションを選択します。
   - ローカルファイルからのインポート-ローカルハードドライブ、ネットワークハードドライブ、またはその他のストレージデバイス上のファイルからシグニチャアップデートをインポートする場合は、このオプションを選択します。
4. テキスト領域に URL を入力するか、ローカルファイルへのパスを参照または入力します。
5. XSLT セクションで、次のオプションのいずれかを選択します。
   - ビルトイン **XSLT** ファイルを使用-ビルトイン XSLT ファイルを使用する場合は、このオプションを選択してください。
   - ローカル **XSLT** ファイルを使用**-**ローカルコンピューター上の XSLT ファイルを使用するには、このオプションを選択します。
   - **URL** から **XSLT** を参照**-**ウェブ **URL** から XSLT ファイルをインポートするには、このオプションを選択します。
6. 「ビルトイン XSLT ファイルを使用」を選択した場合は、ビルトイン XSLT ドロップダウンリストで、次のオプションから使用するファイルを選択します。
   - **Cenzic.**
   - **Deep_Security_for_Web_Apps.**
   - **Hewlett_Packard_Enterprise_WebInspect.**
   - **IBM-AppScan-Enterprise.**
   - **IBM-AppScan-Standard.**
   - **Qualys.**
   - **Whitehat.**

7. [**Update**] をクリックします。更新ファイルがインポートされ、[シグニチャの更新] ダイアログボックスは、「シグニチャオブジェクトの構成と変更」で説明されている「シグニチャオブジェクトの変更」ダイアログボックスとほぼ同じ形式に変更されます。[**Update Signatures Object**] ダイアログボックスには、新規または変更されたシグニチャ規則、SQL インジェクションまたはクロスサイトスクリプティングパターン、および XPath インジェクションパターン（存在する場合）を含むすべてのブランチが表示されます。
8. 新しいシグニチャと変更されたシグニチャを確認して設定します。
9. 終了したら、「**OK**」をクリックし、「閉じる」をクリックします。

署名の自動更新

August 15, 2023

Web アプリケーションファイアウォールの Signature Auto Update 機能を使用すると、ユーザーは最新のシグニチャを取得して、新しい脆弱性から Web アプリケーションを保護できます。自動更新機能は、最新の更新プログラムを取得するための継続的な手動介入を必要とせずに、より優れた保護を提供します。

シグニチャは 1 時間ごとに自動更新され、最新の更新の可用性を定期的にチェックする必要はありません。署名自動更新を有効にすると、NetScaler ADC アプライアンスは署名をホストしているサーバーに接続して、新しいバージョンが利用可能かどうかを確認します。

カスタマイズ可能な場所

最新のアプリケーションファイアウォールのシグニチャは Amazon でホストされ、最新のアップデートをチェックするためのデフォルトの署名 URL として設定されています。

ただし、ユーザは、これらのシグニチャマッピングファイルを内部サーバにダウンロードするオプションがあります。その後、ユーザーは別のシグニチャ URL パスを設定して、ローカルサーバからシグニチャマッピングファイルをダウンロードできます。自動更新機能を機能させるには、外部サイトにアクセスするように DNS サーバーを構成する必要があります。

署名の更新

appfw デフォルトシグニチャオブジェクトを使用して作成されたすべてのユーザー定義シグニチャオブジェクトのバージョンが 0 より大きい。シグニチャ自動更新を有効にすると、すべてのシグニチャが自動的に更新されます。

ユーザが Cenzic や Qualys などの外部フォーマットでシグニチャをインポートした場合、シグニチャはバージョンをゼロとしてインポートします。同様に、ユーザーが空のテンプレートを使用して署名オブジェクトを作成した場合は、ゼロバージョンの署名として作成されます。これらのシグニチャは自動的には更新されません。これは、使用されていないデフォルトシグニチャを管理するオーバーヘッドにユーザーが関心がない場合があるためです。

ただし、Web アプリケーションファイアウォールでは、これらのシグニチャを手動で選択して更新して、既存のルールにデフォルトのシグニチャルールを追加することもできます。シグニチャを手動で更新すると、バージョンが変更され、シグニチャは他のシグニチャとともに自動的に更新されます。

シグニチャ自動更新を構成する

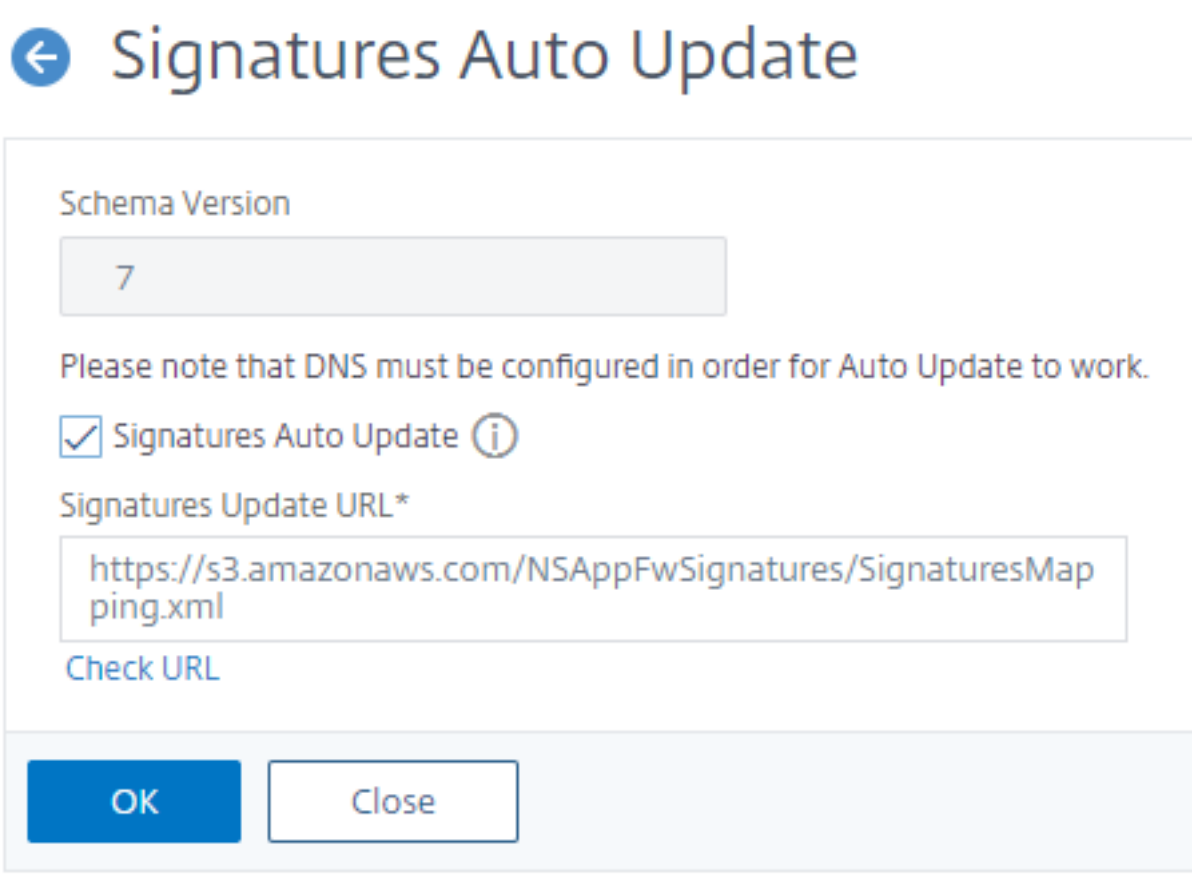
CLI を使用してシグニチャ自動更新機能を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
set appfw settings SignatureAutoUpdate on
set appfw settings SignatureUrl https://s3.amazonaws.com/
   NSAppFwSignatures/SignaturesMapping.xml
<!--NeedCopy-->
```

GUI を使用してシグニチャ自動更新を設定するには、次の手順を実行します。

1. **Security > NetScaler Web App Firewall > Signatures** に移動します。
2. 「アクション」から「設定を自動更新」を選択します。
3. [シグニチャの自動更新] オプションを有効にします。
4. 必要に応じて、シグニチャ更新 URL のカスタマイズされたパスを指定できます。[リセット] をクリックして、デフォルトの`s3.amazonaws.com server`にリセットします。
5. [**OK**] をクリックします。



シグニチャを手動で更新する

ゼロバージョンシグニチャまたはその他のユーザ定義シグニチャを手動で更新するには、まずデフォルトシグニチャの最新のアップデートを取得し、これを使用してターゲットユーザ定義シグニチャを更新する必要があります。

CLI から次のコマンドを実行して、シグニチャファイルを更新します。

```
update appfw signatures "*Default Signatures"
update appfw signatures cenzic  -mergedefault
<!--NeedCopy-->
```

> 注:
>
> `Default Signatures` 大文字と小文字は区別されます。前のコマンドの Cenzic は、更新されるシグニチャファイルの名前です。

インターネットにアクセスせずにデフォルトの署名をインポートする

最新のアップデートを入手するには、Amazon (AWS) サーバーを指すようにプロキシサーバーを設定することをお勧めします。ただし、NetScaler アプライアンスに外部サイトへのインターネット接続がない場合、ユーザーは更新

された署名ファイルをローカルサーバーに保存できます。その後、アプライアンスはローカルサーバからシグニチャをダウンロードできます。このシナリオでは、ユーザーは常に **Amazon** サイトをチェックして、最新のアップデートを入手する必要があります。**Citrix** 公開キーを使用して改ざんから保護して作成された対応する **sha1** ファイルに対して、署名ファイルをダウンロードして検証できます。

Signatures ファイルをローカルサーバにコピーするには、次の手順を実行します。

1. <MySignatures>などのローカルディレクトリを、ローカルサーバー上に作成します。
2. AWS サイトを開きます。
3. SignaturesMapping.xmlファイルを<MySignatures>フォルダにコピーします。

SignaturesMapping.xmlファイルを開くと、シグネチャ用のすべての xml ファイルと、サポートされているさまざまなバージョンの対応する sha1 ファイルが表示されます。次のスクリーンショットでは、そのようなペアの 1 つが強調表示されています。

1. <MySignatures>フォルダにサブディレクトリ<sigs>を作成します。
2. *.xml.sha1ファイルの対応する*.xml files listed in the <file>タグにリストされている<sha1> タグとSignaturesMapping.xmlファイルのすべてのペアを<sigs>フォルダにコピーします。次に、<sigs> フォルダにコピーされるサンプルファイルをいくつか示します。

```
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b86v3s3.xml
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b86v3s3.xml.
sha1
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b0v3s2.xml
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b0v3s2.xml.
sha1
```

> 注:
>
> <MySignatures> フォルダには任意の名前を付けて、任意の場所に置くことができます。ただし、サブディレクトリ<sigs>は、マッピングファイルがコピーされる<MySignatures>フォルダ内のサブディレクトリである必要があります。また、SignaturesMapping.xml に示すように、サブディレクトリ名<sigs>は正確な名前で、大文字と小文字が区別されていることを確認してください。すべてのシグネチャファイルとそれに対応する sha1 ファイルは、この<sigs>ディレクトリの下にコピーする必要があります。

ホストされている Amazon web サーバーからローカルサーバーにコンテンツをミラーリングした後、新しいローカルウェブサーバーへのパスを変更して、自動更新用の SignatureUrl に設定します。たとえば、アプライアンスのコマンドラインインターフェイスから次のコマンドを実行します。

```
1 set appfw settings SignatureUrl https://myserver.example.net/
     MySignatures/SignaturesMapping.xml
2 <!--NeedCopy-->
```

更新するシグニチャの数によっては、更新処理に数分かかる場合があります。更新操作が完了するまでに十分な時間を確保してください。

エラーに直面した場合「URL にアクセス中にエラーが発生しました!」設定中に、手順に従って解決してください。

1. コンテンツセキュリティポリシー (CSP) セキュリティが URL アクセスをブロックしないように、URLhttps://myserver.example.netを /netscaler/ns_gui/admin_ui/php/application/controllers/common/utils.phpに追加します。これらの設定はアップグレードでは保持されないことに注意してください。ユーザーはアップグレード後に再度追加する必要があります。

```
$configuration_view_connect_src = "connect-src 'self' https://app.pendo
    .io https://s3.amazonaws.comhttps://myserver.example.net;";
<!--NeedCopy-->
```

1. ユーザは、https://myserver.example.net/MySignatures/SignaturesMapping.xmlの次の CORS ヘッダーに応答するように Web サーバhttps://myserver.example.netを設定する必要があります。

```
Access-Control-Allow-Methods: GET
Access-Control-Allow-Origin: *
Access-Control-Max-Age: 3000
<!--NeedCopy-->
```

シグニチャを更新するためのガイドライン

シグニチャの更新時には、次のガイドラインが使用されます。

- シグニチャ更新の URL に同じバージョンまたは新しいバージョンの署名オブジェクトが含まれていると、シグニチャが更新されます。
- 各シグニチャルールは、ルール ID とバージョン番号に関連付けられます。たとえば、次のようになります: <SignatureRule id="803"version="16"…>
- 既存のものと同じ ID およびバージョン番号を持つ着信シグニチャファイルのシグニチャルールは、パターンやログ文字列が異なる場合でも無視されます。
- 新しい ID を持つシグニチャルールが追加されます。すべてのアクションと enabled フラグが新しいファイルから使用されます。

  > 注:
  >
  > 更新されたシグニチャを定期的に確認して、新しく追加されたルールを有効にし、アプリケーションの要件に従って他のアクション設定を変更する必要があります。

- 同じ ID を持つが、新しいバージョン番号を持つルールは、既存のルールを置き換えます。既存のルールからのすべてのアクションと有効フラグが保持されます。

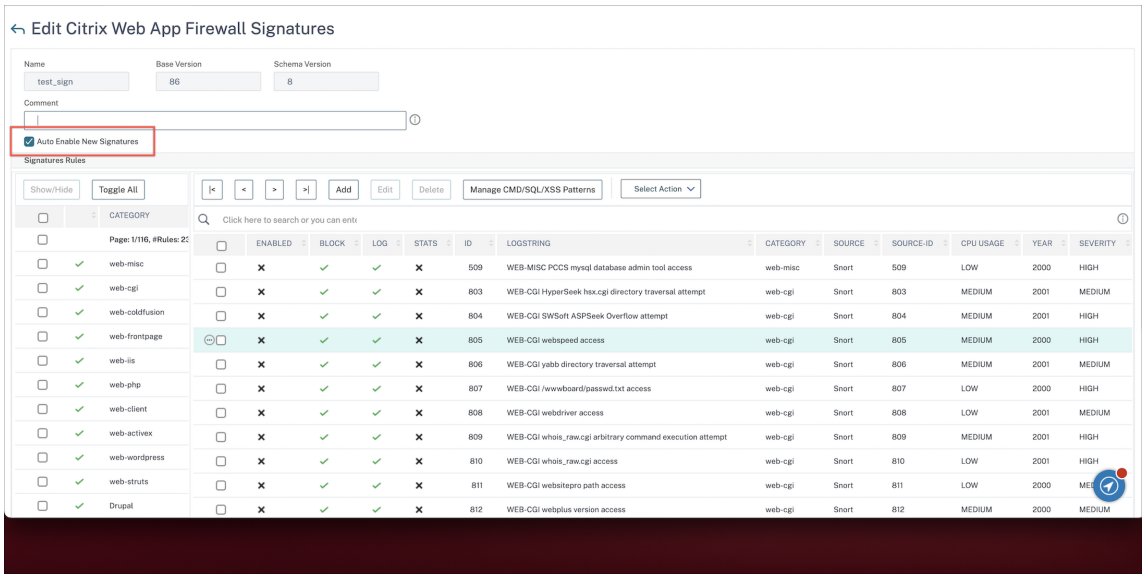
> ヒント：
>
> CLI からシグニチャを更新する場合は、まずデフォルトシグニチャを更新する必要があります。次に、デフォルトシグニチャに基づく各カスタムシグニチャファイルを更新するには、update コマンドを追加する必要があります。最初にデフォルトシグニチャを更新しないと、バージョンの不一致エラーによってカスタムシグニチャファイルの更新が防止されます。

新しい署名を自動有効にする

リリース 13.1 buid 27.x 以降では、[**Auto Enable New Signatures**] を選択して、新しい WAF シグネチャのデフォルトルールを更新後に自動的に有効にできます。

GUI を使用して新しい署名を自動有効にする

1. **Security > NetScaler Web App Firewall > Signatures** に移動します。
2. 署名を選択し、[編集] をクリックします。
3. [新しい署名を自動有効にする] を選択します。



CLI を使用して新しい署名を自動有効にする

コマンドプロンプトで入力します。

```
import appfw signatures <src> <name> [-xslt <string>] [-comment <
string>] [-overwrite] [-merge [-preservedefactions]] [-sha1 <string>]
 [-VendorType Snort] [-autoEnableNewSignatures ( ON | OFF )]
```


例:

```
import signatures http://www.example.com/ns/signatures.xml my-signature
 -autoEnableNewSignatures ON
```

Snort 規則の統合

March 20, 2024

Web アプリケーションへの悪意のある攻撃では、内部ネットワークを保護することが重要です。悪意のあるデータは、インターフェースレベルでウェブアプリケーションに影響を与えるだけでなく、悪意のあるパケットがアプリケーション層にも届きます。このような攻撃を克服するには、内部ネットワークを検査する侵入検知および防止システムを構成することが重要です。

Snort ルールはアプライアンスに統合され、アプリケーション層でデータパケット内の悪意のある攻撃を検査します。snot ルールをダウンロードして WAF シグニチャルールに変換できます。シグニチャは、DOS 攻撃、バッファオーバーフロー、ステルスポートスキャン、CGI 攻撃、SMB プローブ、OS フィンガープリンティングの試みなどの悪意のあるアクティビティを検出できるルールベースの設定になっています。Snort ルールを統合することで、インターフェースとアプリケーションレベルでセキュリティソリューションを強化できます。

Snort ルールを設定

設定は、最初に Snort ルールをダウンロードし、次に WAF シグニチャルールにインポートすることから始まります。ルールを WAF シグネチャに変換すると、そのルールを WAF セキュリティチェックとして使用できます。Snort ベースのシグニチャルールは、受信データパケットを検査して、ネットワークに悪意のある攻撃がないかどうかを検出します。

Snort ルールを WAF シグネチャに変換するための新しいパラメーター「vendorType」がインポートコマンドに追加されました。

パラメータ「vendorType」は、SNORT ルールに対してのみ SNORT で設定されます。

コマンドインターフェイスを使用して snort ルールをダウンロードする

Snort ルールは次の URL からテキストファイルとしてダウンロードできます。

```
https://www.snort.org/downloads/community/snort3-community-rules.tar.
gz
```


コマンドインターフェイスを使用して **Snort** ルールをインポートする

ダウンロード後、Snort ルールをアプライアンスにインポートできます。

コマンドプロンプトで入力します:

```
import appfw signatures <src> <name> [-xslt <string>] [-comment <
string>] [-overwrite] [-merge [-preservedefactions]] [-sha1 <string>]
 [-VendorType Snort]
```

例:

```
import appfw signatures http://www.example.com/ns/signatures.xml sig-
snort -comment “signatures from snort rules” -VendorType snort
```

引数**:**

SRC。インポートされた署名オブジェクトを保存する場所の URL (プロトコル、ホスト、パス、およびファイル名)。

> 注:
>
> インポートするオブジェクトが、アクセスにクライアント証明書認証を必要とする HTTPS サーバ上にある場合、インポートは失敗します。最大長の必須引数:2047

名前。NetScaler 上の署名オブジェクトに割り当てる名前。最大長の必須引数:31

[コメント]。署名オブジェクトに関する情報を保存する方法の説明。最大長:255
の上書き。同じ名前の既存の署名オブジェクトを上書きします。

マージ。既存の署名を新しい署名ルールと統合します。

保存された改ざん。シグニチャルールのデフアクションを保存します。

ベンダータイプ。WAF 署名を生成するサードパーティベンダー。指定できる値: スノート。

NetScaler GUI を使用して **Snort** ルールを構成する

Snort ルールの GUI 設定は、Cenzic、Qualys、Whitehat などの他の外部ウェブアプリケーションスキャナーの設定と似ています。

以下の手順に従って Snort を設定します:

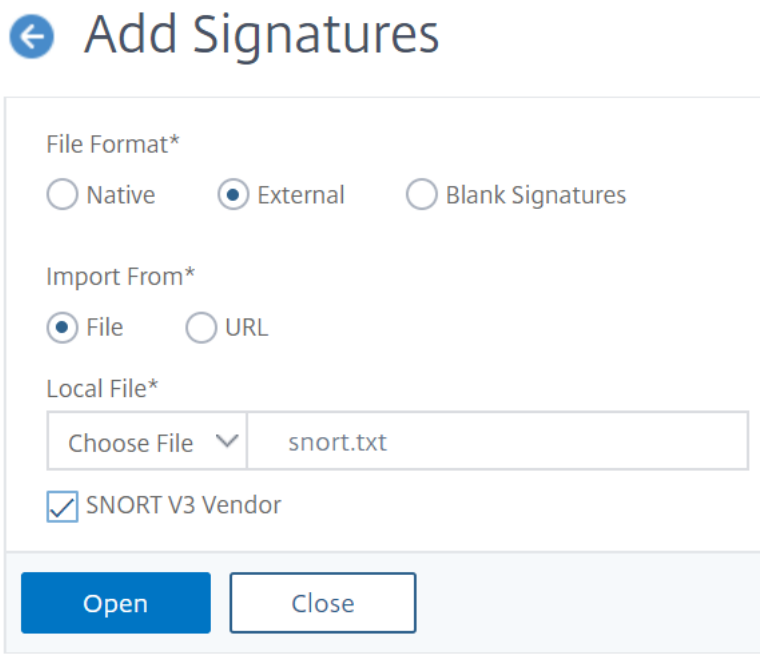
1. [** 構成] > [セキュリティ] > [NetScaler Web App Firewall] **[署名] に移動します。
2. 「署名」ページで、「追加」をクリックします。
3. 「署名の追加」ページで、次のパラメータを設定して Snort ルールを設定します。

    a) ファイル形式。ファイルフォーマットを外部として選択します。

b) からインポートします。インポートオプションを Snort ファイルまたは URL として選択し、URL を入力します。

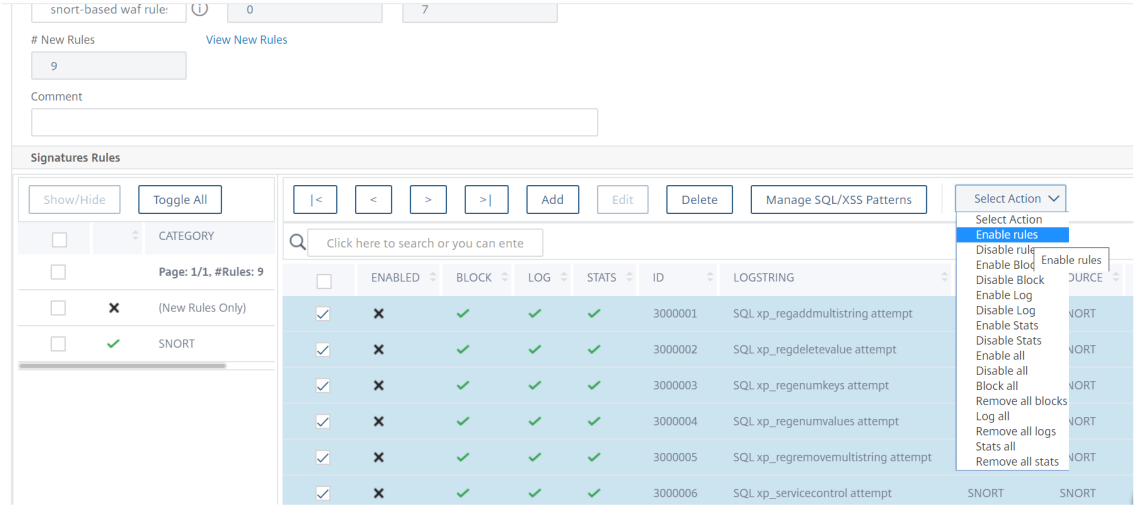
c) Snort V3 ベンダー。チェックボックスを選択して、ファイルまたは URL から Snort ルールをインポートします。

4. ［開く］をクリックします。



アプライアンスは Snort ルールを Snort ベースの WAF 署名ルールとしてインポートします。

ベストプラクティスとして、フィルタアクションを使用して、アプライアンスに WAF 署名ルールとしてインポートしたい SNORT ルールを有効にする必要があります。



5. 確認するには、「はい」をクリックします。

6. 選択したルールがアプライアンスで有効になります。
7. [**OK**] をクリックします。

署名オブジェクトのファイルへのエクスポート

August 15, 2023

署名オブジェクトをファイルにエクスポートして、別の NetScaler にインポートできるようにします。

署名オブジェクトをファイルにエクスポートするには

1. **Security > NetScaler Web App Firewall > Signatures** に移動します。
2. 詳細ペインで、設定するシグニチャオブジェクトを選択します。
3.「アクション」ドロップダウンリストで、「エクスポート」を選択します。
4.「署名オブジェクトのエクスポート」ダイアログ・ボックスの「ローカル・ファイル」テキスト・ボックスに、署名オブジェクトのエクスポート先となるファイルのパスと名前を入力するか、「ブラウズ」ダイアログを使用してパスと名前を指定します。
5. [**OK**] をクリックします。

署名を編集してルールを追加または変更する

August 15, 2023

ユーザー定義の署名を編集して、ルールを追加または変更できます。ローカル署名ルールは、Citrix のデフォルトシグニチャルールと同じ属性を持ち、同じように機能します。デフォルトシグニチャの場合と同様に、有効または無効にし、シグニチャアクションを設定します。

既存のシグネチャが一致しない既知の攻撃からウェブサイトやサービスを保護する必要がある場合は、ローカルルールを追加してください。たとえば、Web サーバーのログを調べて新しいタイプの攻撃を発見し、その特徴を判断したり、新しいタイプの攻撃に関するサードパーティの情報を入手したりすることができます。

シグニチャルールの中心となるのはルールパターンです。ルールが一致するように設計されている攻撃の特性をまとめて記述します。各パターンは、単純な文字列、PCRE 形式の正規表現、組み込みの SQL インジェクションまたはクロスサイトスクリプティングパターンで構成できます。

新しいパターンを追加したり、攻撃に合わせて既存のパターンを変更したりして、シグニチャルールを変更したい場合があります。たとえば、攻撃の変化について知ったり、ウェブサーバーのログを調べたり、サードパーティの情報からより良いパターンを判断したりできます。

ローカル署名ルールを追加または変更する

1. **Security > NetScaler Web App Firewall > Signatures** に移動します。
2. 詳細ペインで、編集するユーザー定義の署名を選択し、「編集」をクリックします。
3. 「署名ルール」セクションで、「追加」をクリックします。「署名規則」ウィンドウが表示されます。
4. 適切なチェックボックスを選択して、署名のアクションを設定します。

    - 有効。新しいシグニチャルールを有効にします。これを選択しない場合、この新しいシグニチャルールは設定に追加されますが、非アクティブになります。
    - ブロック。このシグニチャルールに違反する接続をブロックします。
    - ログ。この署名ルールの違反を NetScaler ログに記録します。
    - スタット。このシグネチャルールの違反を統計に含めます。
    - **Remove-** 応答から署名ルールに一致する情報を削除します。(応答ルールにのみ適用されます。)
    - **X-Out**。署名ルールに一致する情報を文字 X でマスクします（応答ルールにのみ適用)。
    - 重複を許可。このシグニチャオブジェクトでこのシグニチャルールの複製を許可します。

5. [**Category**] ドロップダウンリストから、新しいシグニチャルールのカテゴリを選択します。

    カテゴリを作成する場合は、[追加] をクリックします。詳細については、「署名ルールカテゴリの追加」を参照してください。

6. [**LogString**] テキストボックスに、ログで使用するシグニチャルールの簡単な説明を入力します。
7. 「コメント」テキストボックスに、コメントを入力します。（オプション）
8. 詳細オプションを変更するには、[詳細] をクリックします。

    a) このシグニチャルールを適用する前に HTML コメントを削除するには、[Strip Comments] ドロップダウンリストで [All] または [Exclude Script Tag] を選択します。
    b) CSRF リファラーヘッダーチェックを設定するには、「CSRF リファラーヘッダーチェック」ラジオボタン配列で、「存在する場合」または「常に」ラジオボタンのいずれかを選択します。
    c) このローカルシグニチャルールに割り当てられたルール ID を手動で変更するには、[Rule ID] テキストボックスで番号を変更します。ID は、ローカルシグニチャルールにまだ割り当てられていない 1000000 から 1999999 までの正の整数でなければなりません。
    d) 新しいシグニチャルールにバージョン番号を割り当てるには、バージョン番号テキストボックスで番号を変更します。
    e) ソース ID を割り当てるには、ソース ID テキストボックスの文字列を変更します。
    f) ソースを指定するには、「ソース」ドロップダウンリストから「ローカル」または「Snort」を選択するか、リストの右にある「追加」アイコンをクリックして新しいソースを追加します。
    g) このローカルシグニチャルールの違反にハームスコアを割り当てるには、ハームスコアテキストボックスに 1 ～10 の数字を入力します。

h) このローカルシグニチャルールに重要度評価を割り当てるには、「重要度」ドロップダウンリストで「高」、「中」、または「低」を選択するか、リストの右にある「追加」アイコンをクリックして新しい重要度評価を追加します。

i) このローカルシグニチャルールに違反タイプを割り当てるには、[Type] ドロップダウンリストで [Vulnerable] または [Warning] を選択するか、リストの右側にある [Add] アイコンをクリックして新しい違反タイプを追加します。

9. 「ルールパターン」で、「追加」をクリックしてパターンを追加します。既存のパターンを編集して編集することもできます。[編集] をクリックします。

   パターンの追加または編集の詳細については、「シグニチャルールパターン」を参照してください。

10. [**OK**] をクリックします。

シグネチャルールカテゴリの追加

シグニチャルールをカテゴリに配置すると、個々のシグニチャではなく、シグニチャグループのアクションを設定できます。次のような理由からそうしたいと思うかもしれません。

- 選択の容易さ。たとえば、特定のグループのすべてのシグネチャルールが、特定の種類の Web サーバソフトウェアまたはテクノロジに対する攻撃を防ぐと仮定します。保護されている Web サイトがそのソフトウェアやテクノロジーを使用している場合は、それらすべてを有効にする必要があります。有効になっていない場合は、いずれも有効にしないほうがよいでしょう。
- 初期設定のしやすさ。シグニチャのグループのデフォルトを 1 つずつではなく、カテゴリとして設定するのが最も簡単です。その後、必要に応じて個々の署名に変更を加えることができます。
- 継続的な構成の容易さ。特定のカテゴリに属するなど、特定の基準を満たすシグニチャだけを表示できる場合は、シグニチャの設定が簡単です。

シグネチャルールカテゴリの追加

December 8, 2023

カテゴリにシグニチャルールを追加すると、個々のシグニチャごとではなく、シグニチャのグループに対してアクションを設定できます。次のような理由からそうしたいと思うかもしれません:

- 選択の容易さ。たとえば、特定のグループのすべてのシグニチャルールが、特定のタイプの Web サーバソフトウェアまたはテクノロジに対する攻撃から保護するとします。保護されている Web サイトがそのソフトウェアやテクノロジーを使用している場合は、それらすべてを有効にする必要があります。有効になっていない場合は、いずれも有効にしないほうがよいでしょう。
- 初期設定のしやすさ。シグニチャのグループのデフォルトを 1 つずつではなく、カテゴリとして設定するのが最も簡単です。その後、必要に応じて個々の署名に変更を加えることができます。

- 継続的な構成の容易さ。特定のカテゴリに属するなど、特定の基準を満たすシグニチャだけを表示できる場合は、シグニチャの設定が簡単です。

1. [セキュリティ] > [**Web App Firewall**] > [署名] に移動します。
2. 詳細ペインで、設定するシグニチャオブジェクトを選択し、「開く」をクリックします。
3. **[**** 署名オブジェクトの変更] ダイアログボックスの画面中央の [フィルター結果] ウィンドウの下にある [追加]** をクリックします。
4. 「ローカル署名規則の追加」ダイアログで、「カテゴリ」ドロップダウンリストの右にあるアイコンをクリックします。
5. 「署名ルールカテゴリの追加」ダイアログボックスの「新規カテゴリ」テキストボックスに、新しい署名カテゴリの名前を入力します。名前は 1 ～64 文字で構成できます。
6. [**OK**] をクリックします。

署名ルールパターンを追加

August 16, 2023

シグネチャが一致する場合、パターンを追加するか、既存のパターンを変更して攻撃を特徴付ける文字列または式を指定できます。攻撃が示すパターンを検出するには、ウェブサーバーのログを調べることができます。ツールを使用してリアルタイムで接続データを観察したり、攻撃に関するサードパーティのレポートから文字列や式を取得したりできます。

> 重要
>
> : シグニチャルールに追加する新しいパターンは、既存のパターンと AND 関係になります。潜在的な攻撃がすべてのパターンをシグニチャと一致させたくない場合は、既存のシグニチャルールにパターンを追加しないでください。

各パターンは、単純な文字列、PCRE 形式の正規表現、組み込みの SQL インジェクションまたはクロスサイトスクリプティングパターンで構成できます。正規表現に基づくパターンを追加する前に、PCRE 形式の正規表現を理解していることを確認する必要があります。PCRE 表現は複雑で強力です。その仕組みを理解していないと、意図せず望んでいないものと一致するパターン（偽陽性）や、望んでいたものと一致しない（偽陰性 **）パターンを作成してしまう可能性があります。

デフォルト以外のコンテンツタイプのカスタム署名パターン

NetScaler Web App Firewall（WAF）は、正規化されたコンテンツを検査する新しい場所をサポートするようになりました。デフォルトでは、WAF はデフォルト以外のコンテンツタイプのエンコードされたペイロードをブロックしません。これらのコンテンツタイプがホワイトリストに登録され、設定済みのアクションが適用されていない場合、SQL およびクロスサイトスクリプティング保護チェックは、エンコードされたペイロード内の SQL 攻撃または

クロスサイトスクリプティング攻撃をフィルタリングしません。この問題を解決するには、ユーザーはこの新しい場所 (HTTP_CANON_POST_BODY) でカスタム署名ルールを作成して、エンコードされたペイロードにデフォルト以外のコンテンツタイプがないかを調べ、SQL 攻撃やクロスサイトスクリプティング攻撃を受けた場合は、投稿本文の正規化後にトラフィックをブロックします。

> 注:
>
> このサポートは HTTP リクエストにのみ適用されます。

PCRE 形式の正規表現にまだ慣れていない場合は、次のリソースを使用して基本を学んだり、特定の問題を解決したりすることができます。

- 「正規表現の習得」、第 3 版。著作権 (c) 2006 ジェフリー・フリードルオライリー・メディア、ISBN: 9780596528126。
- 「正規表現クックブック」。著作権 (c) 2009 年、ヤン・ゴイヴァールツとスティーブン・レヴィサンオライリー・メディア、ISBN: 9780596520687
- PCRE マニュアルページ/仕様
- PCRE マンページ/仕様
- ウィキペディア PCRE エントリ
- PCRE メーリングリスト

ASCII 以外の文字を PCRE 形式の正規表現でエンコードする必要がある場合、NetScaler ADC プラットフォームは 16 進数の UTF-8 コードのエンコードをサポートしています。詳細については、PCRE 文字エンコーディング形式を参照してください。

シグネチャルールパターンの設定

署名を編集するときに、ルールパターンを追加または編集できます。シグニチャルールを追加または変更するには、「シグニチャを編集してルールを追加または変更する」を参照してください。

- タイプ -パターンが一致させたい接続のタイプを選択します。
  - リクエスト -挿入された SQL コード、Web フォームへの攻撃、クロスサイトスクリプト、または不適切な URL などのリクエスト要素または機能を照合します。
  - レスポンス -クレジットカード番号やセーフオブジェクトなどのレスポンス要素や特徴と一致します。
- 場所 - このパターンで調べるエリアを選択してください。この領域には、HTTP リクエストまたはレスポンスのどの要素でこのパターンを調べるべきかが記載されています。選択したパターンタイプに応じて、オプションがエリアリストに表示されます。選択したパターンタイプによって異なります。

  リクエストパターンタイプには、HTTP リクエストに関連する項目が表示されます。
  - **HTTP_ANY**. HTTP 接続のすべての部分。
  - **HTTP_COOKIE**. 任意のクッキー変換が実行された後、HTTP リクエストヘッダー内のすべてのクッキー。

> 注
>
> :HTTP レスポンスの「Set-Cookie:」ヘッダーは検索しません。

- **HTTP_FORM_FIELD**. URL デコード、パーセントデコード、および余分な空白の削除後のフォームフィールドとその内容。<Location> タグを使用して、検索するフォームフィールド名のリストをさらに制限できます。
- **HTTP _HEADER**。クロスサイトスクリプティングまたは URL デコード変換後の HTTP ヘッダーの値部分。
- **HTTP _METHOD**。HTTP リクエストメソッド。
- **HTTP_URL**. UTF-* 文字セットへの変換、URL のデコード、空白の除去、相対 URL の絶対値への変換後の、HTTP ヘッダー内の URL の値部分（クエリポートまたはフラグメントポートは除く）。HTML エンティティのデコードは含まれません。
- **HTTP_ORIGIN_URL**. ウェブフォームのオリジン URL。
- **HTTP_POST _BODY**。HTTP 投稿本文とそれに含まれる Web フォームデータ。
- **HTTP_RAW_COOKIE**.「Cookie:」という名前の部分を含むすべての HTTP リクエストクッキー。注意:HTTP レスポンスの「Set-Cookie:」ヘッダーは検索しません。
- **HTTP _RAW _HEADER**。HTTP ヘッダー全体。個々のヘッダーは改行文字 (\ n) またはキャリッジリターン/ラインフィード文字列 (\ r\ n) で区切られています。

レスポンスタイプには、HTTP レスポンスに関連する項目が表示されます。

- **HTTP_RAW_RESP_HEADER**. URL 変換が行われた後のレスポンスヘッダーの名前と値の部分を含むレスポンスヘッダー全体、および完全なレスポンスステータス。HTTP_RAW_HEADER と同様に、個々のヘッダーは改行文字 (\ n) またはキャリッジリターン/ラインフィード文字列 (\ r\ n) で区切られます。
- **HTTP_RAW_SET_COOKIE**。URL 変換が実行された後の Set-Cookie ヘッダー全体

  > 注**:**
  >
  > URL 変換により、Set-Cookie ヘッダーのドメイン部分とパス部分の両方が変更される可能性があります。

- **HTTP_RAW_URL**. URL 変換が実行される前のリクエスト URL 全体 (クエリまたはフラグメント部分を含む)。
- **HTTP _RESP_HEADER**。URL 変換が実行された後のレスポンスヘッダ全体の値の部分。
- **HTTP_RESP_BODY**. HTTP レスポンスボディ
- **HTTP _SET_COOKIE**。HTTP レスポンスヘッダーのすべての「Set-Cookie」ヘッダー。

- **HTTP_STATUS_CODE**. HTTP ステータスコード。
- **HTTP_STATUS_MESSAGE**. HTTP ステータスメッセージ。

エリアリストからオプションを選択すると、選択したエリアのオプションが動的に変更されます。

- **Any.** フィールド名または URL をチェックします。
- リテラル。リテラル文字列を含むフィールド名または URL をチェックします。「リテラル」を選択すると、テキストボックスが表示されます。テキストボックスに必要なリテラル文字列を入力します。
- **PCRE**。PCRE 形式の正規表現と一致するフィールド名または URL をチェックします。この選択肢を選択すると、正規表現ウィンドウが表示されます。ウィンドウに正規表現を入力します。正規表現トークンを使用して一般的な正規表現要素をカーソルに挿入することも、「正規表現エディター」をクリックして正規表現エディターダイアログボックスを表示することもできます。このダイアログボックスでは、必要な正規表現をより簡単に作成できます。
- **Expression**: NetScaler のデフォルト式と一致するフィールド名または URL をチェックします。

- パターン - パターンは、照合するパターンを定義するリテラル文字列または PCRE 形式の正規表現です。リストからマッチタイプを選択します。
  - リテラル。リテラル文字列。
  - **PCRE**。PCRE 形式の正規表現。

    > 注
    >
    > PCRE を選択すると、パターンウィンドウの下にある正規表現ツールが有効になります。これらのツールは、他のほとんどのタイプのパターンには役に立ちません。

  - **Expression**: NetScaler ADC のデフォルト表現言語の式は、NetScaler ADC アプライアンスで Web App Firewall ポリシーを作成するための表現言語と同じです。NetScaler ADC 式言語は、もともとポリシールール用に開発されたものですが、柔軟性の高い汎用言語で、署名パターンの定義にも使用できます。

    エクスプレッションを選択すると、NetScaler エクスプレッションエディターがパターンウィンドウの下に表示されます。式エディタとその使用方法の詳細については、「[式の追加] ダイアログボックスを使用してファイアウォールルール (式) を追加するには」を参照してください。
  - **SQL** インジェクション。指定された場所に挿入された SQL を検索するように Web App Firewall に指示します。
  - **CrossSiteScripting**. Web App Firewall に、指定された場所でクロスサイトスクリプトを探すように指示します。
  - コマンドインジェクション。指定された場所に挿入された悪意のあるコマンドを探すように Citrix Web App Firewall に指示します。

- **SQL** インジェクション/グラマー。指定された場所に挿入された SQL 文法を探すように Citrix Web App Firewall に指示します。特に、`Select`や`From`などのよく使われる単語が HTTP リクエストで使われている場合はなおさらです。
- コマンドインジェクション/文法。指定された場所に挿入された悪意のあるコマンド文法を検索するように Citrix Web App Firewall に指示します。特に、HTTP リクエストで「Exit」などの一般的に使用される単語が使用されている場合。

さらに設定を行う場合は、以下を指定してください。

- オフセット。このパターンでマッチングを開始する前にスキップする文字数。このフィールドを使用して、最初の文字以外の任意の時点で文字列の検査を開始します。
- 深さ。一致するかどうかを調べる開始点から何文字か。このフィールドを使用して、大きな文字列の検索を特定の文字数に制限します。
- 最小長さ。検索する文字列は、指定されたバイト数以上の長さでなければなりません。短い文字列は一致しません。
- 最大長。検索する文字列の長さは、指定されたバイト数以下でなければなりません。長い文字列は一致しません。
- 検索方法。`fastmatch`ラベルの付いたチェックボックス。パフォーマンスを向上させるには、リテラルパターンでのみ`fastmatch`を有効にできます。

> 注
>
> 「署名ルールパターン」ペインで「**OK**」をクリックするまで、変更は保存されません。変更を破棄しない限り、[**OK**] をクリックせずにこれらのダイアログボックスを閉じないでください。

ルールをインポートしてマージする

December 8, 2023

署名エディタを使用して GUI からインポートおよびマージ操作を実行すると、新しいルール、更新されたルール、重複するルール、および無効なルールが表示されるようになりました。

署名エディターには、次の 4 つの新しい行が表示されます。

1. 新しい規則
2. 規則の更新
3. 重複ルール
4. ルールが無効です

「新規ルールのみ」フィルターと「更新済みルールのみ」フィルターの出力は、署名エディターの編集ウィンドウの「カテゴリ」フィルターペインにも表示されます。

GUI からファイルをインポートして、「新規」、「複製」、「無効」、「更新」のルールに対応するリンクを確認する必要があります。

署名ルールをインポートする手順:

1. NetScaler Web GUI で、「構成」**>**「セキュリティ」**>**「**NetScaler Web App Firewall** 署名」に移動します。「署名」ウィンドウで、「追加」をクリックします。[ファイル形式] > [ネイティブ]、**[**インポート元] > [**URL**] を選択し、[URL] フィールドに上記のリンクを追加します。URL にアクセスできない場合は、XML データをダウンロードできます。
2. [開く] をクリックすると、シグネチャファイルが開き、新しいルールと無効なルールのリンクが表示されます。
3. 3rdパーティシグニチャルールをインポートすると、インポートされた.xml ファイルに 90 個の新しいルールと 9 個の重複ルールが表示されます。URL にアクセスできない場合は、XML データをダウンロードできます。

高可用性デプロイとビルドのアップグレードにおけるシグネチャアップデート

August 15, 2023

シグニチャの更新はプライマリノードで行われます。シグネチャがプライマリノードで更新される間、同時に更新されたファイルはセカンダリノードと同期されます。

デフォルトシグニチャは常に最初に更新され、次に残りのユーザ定義シグニチャが更新されます。

Amazon AWS への接続

Amazon AWS への接続には、デフォルトルート NSIP が使用されます。SNIP を使用する特定のユースケースシナリオがあり、複数の SNIP がある場合は、ホスティングサイトから ARP 応答を最初に受信したものがルートを保持します。

バージョンアップグレード中のシグネチャ更新

アップグレードの場合、NS のシグニチャの基本バージョンが古い場合、* 新しいシグニチャバージョンが使用可能な場合は、デフォルトシグニチャが自動的に更新されます。

スキーマが変更された場合、バージョンがアップグレードされると、すべてのシグネチャオブジェクトのスキーマバージョンが更新されます。

ただし、ユーザ定義シグニチャの基本バージョンでは、リリース 10.5 とリリース 11.0 では動作が異なります。

リリース 10.5 では、デフォルトの署名のみが更新され、残りの署名の基本バージョンはビルドのアップグレード後も変更されませんでした。

リリース 11.0 では、この動作が変更されました。アプライアンスをアップグレードして新しいビルドをインストールすると、*Default 署名オブジェクトだけでなく、現在アプライアンスに存在する他のすべてのユーザー定義署名も更新され、ビルドのアップグレード後も同じバージョンになります。

10.5 と 11.0 の両方のリリースビルドで、自動更新が設定されている場合、*Default Signatures と 0 以外のすべてのバージョン署名は、リリースされた最新署名バージョンに自動的に更新され、同じ基本バージョンになります。

セキュリティチェックの概要

October 25, 2023

Web App Firewall の高度な保護（セキュリティチェック）は、保護されている Web サイトや Web サービスに対する複雑な攻撃や未知の攻撃を阻止するために設計された一連のフィルターです。セキュリティチェックでは、ヒューリスティック、ポジティブセキュリティ、およびその他の手法を使用して、シグネチャだけでは検出できない攻撃を検出します。セキュリティチェックを構成するには、Web App Firewall プロファイルを作成して構成します。このプロファイルは、使用するセキュリティチェックと、セキュリティチェックに失敗した要求または応答の処理方法を Web App Firewall に指示するユーザー定義設定の集まりです。プロファイルは、シグニチャオブジェクトとセキュリティ設定を作成するためのポリシーに関連付けられます。

Web App Firewall には 20 のセキュリティチェックがあり、対象となる攻撃の種類や構成の複雑さは大きく異なります。セキュリティチェックは次のカテゴリに分類されます。

- 一般的なセキュリティチェック。コンテンツを含まない、またはすべての種類のコンテンツに同等に適用される Web セキュリティのあらゆる側面に適用されるチェック。
- **HTML** セキュリティチェック。HTML のリクエストとレスポンスを調べるチェック。これらのチェックは、HTML ベースの Web サイト、および HTML と XML が混在するコンテンツを含む Web 2.0 サイトの HTML 部分に適用されます。
- **XML** セキュリティチェック。XML 要求と応答を調べるチェック。これらのチェックは、XML ベースの Web サービスと Web 2.0 サイトの XML 部分に適用されます。

セキュリティチェックは、オペレーティングシステムやウェブサーバーソフトウェアの脆弱性への攻撃、SQL データベースの脆弱性、Web サイトや Web サービスの設計とコーディングのエラー、機密情報をホストしたりアクセスできるサイトのセキュリティ保護の失敗など、さまざまな種類の攻撃から保護します。

すべてのセキュリティチェックには、Web App Firewall がチェックに一致する接続を処理する方法を制御する一連の構成オプション、つまりチェックアクションがあります。すべてのセキュリティチェックでは、3 つのチェックアクションを使用できます。これには、次の種類のアカウントがあります。

- ブロック。署名に一致する接続をブロックします。デフォルトでは、無効になっています。
- ログ。後で分析できるように、シグネチャに一致する接続をログに記録します。デフォルトで有効。
- 統計。シグニチャごとに、一致した接続の数を示す統計情報を保持し、ブロックされた接続の種類に関するその他の特定の情報を提供します。デフォルトでは、無効になっています。

4 つ目のチェックアクションである **Learn** は、チェックアクションの半分以上で使用できます。保護されている Web サイトまたは Web サービスへのトラフィックを監視し、セキュリティチェックに繰り返し違反する接続を使用して、チェックに対する推奨例外 (緩和) を生成したり、チェックの新しいルールを生成したりします。チェックアクションに加えて、特定のセキュリティチェックには、どの接続がそのチェックに違反しているかを判断するためにチェックが使用するルールを制御するパラメーターや、チェックに違反した接続に対する Web App Firewall の応答を構成するパラメーターがあります。これらのパラメータはチェックごとに異なり、各チェックのドキュメントで説明されています。

セキュリティチェックを構成するには、「Web App Firewall ウィザード」で説明されているように Web App Firewall ウィザードを使用するか、GUI を使用した手動構成の説明に従って、セキュリティチェックを手動で構成できます。緩和や規則を手動で入力したり、学習したデータを確認するなど、一部のタスクは、コマンドラインではなく GUI を使用してのみ実行できます。通常、ウィザードを使用するのが最適な構成方法ですが、ウィザードに精通していて、1 回のセキュリティチェック用に構成を調整したいだけの場合は、手動構成の方が簡単な場合もあります。

セキュリティチェックの設定にどの方法を使用するかにかかわらず、各セキュリティチェックでは特定のタスクを実行する必要があります。多くのチェックでは、そのセキュリティチェックのブロックを有効にする前に、正当なトラフィックがブロックされないように例外 (緩和) を指定する必要があります。これを手動で行うには、一定量のトラフィックがフィルタリングされた後にログエントリを確認し、必要な例外を作成します。ただし、通常は学習機能を有効にしてトラフィックを監視し、必要な例外を推奨するほうがはるかに簡単です。

Web App Firewall は、トランザクションの処理中にパケットエンジン (PE) を使用します。各パケットエンジンには 100 K セッションの制限があり、ほとんどの導入シナリオには十分です。ただし、Web App Firewall が大量のトラフィックを処理していて、セッションタイムアウトがより高い値に設定されている場合、セッションが蓄積される可能性があります。有効な Web App Firewall セッションの数が PE あたりの 100 K の制限を超えると、Web App Firewall のセキュリティチェック違反が Security Insight アプライアンスに送信されない可能性があります。セッションタイムアウトを小さい値に下げたり、セッションレス URL クロージャやセッションレスフィールドの一貫性を伴うセキュリティチェックにセッションレスモードを使用したりすると、セッションが蓄積されるのを防ぐのに役立つ場合があります。トランザクションに長いセッションが必要になる可能性があるシナリオでこれが実行可能なオプションでない場合は、より多くのパケットエンジンを搭載したハイエンドプラットフォームにアップグレードすることをお勧めします。

キャッシュされた AppFirewall のサポートが追加され、コアごとの CLI による最大セッション設定は 50K セッションに設定されます。

リクエストとレスポンスのセキュリティチェック

セキュリティチェックのリストは次のとおりです。

チェックをリクエストする

リクエストのセキュリティチェックは次のとおりです。

- 開始 URL
- URL を拒否する
- クッキーの一貫性
- Cookie ハイジャック
- バターオーバーフロー
- ポストボディリミット
- コンテンツタイプ
- コンテンツタイプ XML ペイロードを推測
- ファイルアップロードの種類
- フォームフィールドの一貫性
- フィールドの書式
- CSRF フォームタグ付け
- HTML クロスサイトスクリプティング
- HTML SQL インジェクション
- HTML コマンドインジェクション
- ブロックキーワード-XML
- XML 形式
- XML サービス拒否
- XML クロスサイトスクリプティング
- XML SQL インジェクション
- XML アタッチメント
- XML メッセージ検証
- JSON サービス拒否
- JSON クロスサイトスクリプティング
- JSON SQL インジェクション
- JSON コマンド・
- ブロックキーワード-JSON

レスポンスチェック

レスポンスセキュリティチェックは次のとおりです。

- クレジットカード
- セーフオブジェクト
- XML SOAP フォールトフィルタリング
- Web サービスの相互運用性の一部

トップレベルの保護

August 15, 2023

Web App Firewall の 4 つの保護機能は、一般的な種類のウェブ攻撃に対して特に効果的であるため、他のどの保護手段よりも一般的に使用されています。これには、次の種類のアカウントがあります。

- **HTML** クロスサイトスクリプティング。スクリプトが配置されているサイトとは異なる Web サイトのコンテンツにアクセスまたは変更しようとするスクリプトのリクエストとレスポンスを調べます。このチェックは、このようなスクリプトを検出すると、要求または応答を宛先に転送する前にスクリプトを無害にするか、接続をブロックします。
- **HTML SQL** インジェクション。フォームフィールドデータを含むリクエストを調べて、SQL データベースに SQL コマンドを挿入しようとするかどうかを調べます。このチェックは、挿入された SQL コードを検出すると、要求をブロックするか、または Web サーバーに要求を転送する前に、挿入された SQL コードを無害にします。

  注: 構成に次の両方の条件が当てはまる場合は、Web App Firewall が正しく構成されていることを確認する必要があります。

  - HTML クロスサイトスクリプティングチェックまたは HTML SQL インジェクションチェック (あるいはその両方) を有効にしている場合、
  - 保護された Web サイトは、ファイルのアップロードを受け入れるか、大きな POST 本文データを含むことができる Web フォームを含みます。

  このケースを処理するための Web App Firewall の構成の詳細については、「アプリケーションファイアウォールの構成」を参照してください。
- バッファオーバーフロー。リクエストを調べて、Web サーバー上でバッファオーバーフローを引き起こそうとする試みを検出します。
- クッキーの一貫性。ユーザーのリクエストで返された Cookie を調べて、Web サーバーがそのユーザーに設定した Cookie と一致することを確認します。変更された Cookie が見つかった場合、要求が Web サーバーに転送される前に、要求から取り除かれます。

バッファオーバーフローのチェックは簡単で、通常はただちにブロックを有効にできます。他の 3 つのトップレベルのチェックはかなり複雑で、トラフィックをブロックするために安全に使用するには設定が必要です。NetScaler では、これらのチェックを手動で構成するのではなく、学習機能を有効にして、必要な例外を生成できるようにすることを強くお勧めします。

HTML クロスサイトスクリプティングチェック

August 15, 2023

HTML クロスサイトスクリプティング (クロスサイトスクリプティング) チェックでは、クロスサイトスクリプティング攻撃の可能性について、ユーザーリクエストのヘッダーと POST 本文の両方を調べます。クロスサイトスクリプトが見つかった場合は、攻撃を無害化するようにリクエストを変更（変換）するか、リクエストをブロックします。

> 注:
>
> HTML クロスサイトスクリプティング (クロスサイトスクリプティング) チェックは、コンテンツタイプ、コンテンツの長さなどに対してのみ機能します。また、Web アプリケーションファイアウォールプロファイルで ‘checkRequestHeaders’ オプションが有効になっていることを確認します。

同じオリジンルールに違反する *HTML* クロスサイトスクリプティングスクリプトを使用することで、保護された *Web* サイトでのスクリプトの誤用を防ぐことができます。このルールでは、スクリプトが配置されているサーバー以外のサーバー上のコンテンツにアクセスしたり、コンテンツを変更したりしてはならないという規定があります。同一生成元ルールに違反するスクリプトはクロスサイトスクリプトと呼ばれ、スクリプトを使用して別のサーバー上のコンテンツにアクセスしたり変更したりすることをクロスサイトスクリプティングと呼びます。クロスサイトスクリプティングがセキュリティ上の問題である理由は、クロスサイトスクリプティングを許可する Web サーバーが、その Web サーバー上ではなく、攻撃者が所有および制御している別の Web サーバー上のスクリプトで攻撃される可能性があるためです。

残念ながら多くの企業では、同じオリジンルールに違反する JavaScript 拡張 Web コンテンツの大規模なインストールベースがあります。このようなサイトで HTML クロスサイトスクリプティングチェックを有効にする場合は、チェックが正当なアクティビティをブロックしないように、適切な例外を生成する必要があります。

Web App Firewall には、HTML クロスサイトスクリプティング保護を実装するためのさまざまなアクションオプションが用意されています。ブロック、ログ、統計、および学習アクションに加えて、送信されたリクエスト内のスクリプトタグをエンティティがエンコードすることにより、クロスサイトスクリプトを変換して攻撃を無害にするオプションもあります。[クロスサイトスクリプティングの完全な URL をチェック] パラメータを設定して、クエリパラメータだけでなく URL 全体を検査してクロスサイトスクリプティング攻撃を検出するかどうかを指定できます。**inspectQueryContentTypes** パラメータを設定して、特定のコンテンツタイプに対するクロスサイトスクリプティング攻撃のリクエストクエリ部分を検査できます。

リラクゼーションを展開すると、誤検出を回避できます。Web App Firewall 学習エンジンは、緩和ルールの設定に関する推奨事項を提供できます。

アプリケーションに最適化された HTML クロスサイトスクリプティング保護を設定するには、次のいずれかのアクションを設定します。

- ブロック—ブロックを有効にすると、リクエストでクロスサイトスクリプティングタグが検出されると、ブロックアクションがトリガーされます。

- [**Log**]：ログ機能を有効にすると、HTML クロスサイトスクリプティングチェックでは、実行したアクションを示すログメッセージが生成されます。ブロックを無効にすると、クロスサイトスクリプティング違反が検出されたヘッダーまたはフォームフィールドごとに個別のログメッセージが生成されます。ただし、要求がブロックされると、1 つのメッセージだけが生成されます。同様に、クロスサイトスクリプティングタグが複数のフィールドで変換された場合でも、変換操作ではリクエストごとに 1 つのログメッセージが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとしたことを示している可能性があります。
- [**Stats**]：有効にすると、統計機能は違反とログに関する統計情報を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当な要求がブロックされた場合は、新しい緩和ルールを構成する必要があるか、既存の緩和ルールを変更する必要があるかを再確認するために、構成を再確認する必要があります。
- 学習-どの緩和ルールがアプリケーションに適しているかわからない場合は、学習機能を使用して、学習データに基づいて HTML クロスサイトスクリプティングルールの推奨事項を生成できます。Web App Firewall 学習エンジンは、トラフィックを監視し、観測された値に基づいて学習の推奨事項を提供します。パフォーマンスを損なうことなく最適な効果を得るには、学習オプションを短時間有効にしてルールの代表的なサンプルを取得し、ルールを展開して学習を無効にすることをお勧めします。
- クロスサイトスクリプトの変換-有効にすると、Web App Firewall は HTML クロスサイトスクリプティングチェックに一致するリクエストに対して次の変更を加えます。
  - 左山かっこ (<) から HTML 文字エンティティに相当する (<)
  - 右山括弧 (>) を HTML 文字エンティティに相当 (>)

これにより、ブラウザが<script>などの安全でない HTML タグを解釈して悪質なコードを実行することがなくなります。リクエストヘッダーのチェックと変換の両方を有効にすると、リクエストヘッダーで見つかった特殊文字もすべて変更されます。保護された Web サイト上のスクリプトにクロスサイトスクリプティング機能が含まれているが、Web サイトがこれらのスクリプトに依存して正しく動作しない場合は、ブロックを無効にして変換を有効にしても問題ありません。この構成により、クロスサイトスクリプティング攻撃の可能性を阻止しながら、正当な Web トラフィックがブロックされなくなります。

- クロスサイトスクリプティングの完全な **URL** を確認します。完全な URL のチェックが有効な場合、Web App Firewall は URL のクエリ部分だけをチェックするのではなく、HTML クロスサイトスクリプティング攻撃について URL 全体を検査します。
- **[**リクエストヘッダー**]** を確認します。リクエストヘッダーチェックが有効な場合、Web App Firewall は URL だけでなく HTML クロスサイトスクリプティング攻撃のリクエストのヘッダーを検査します。GUI を使用する場合は、Web App Firewall プロファイルの [設定] タブでこのパラメーターを有効にできます。
- **QueryContentTypes** を検査します。リクエストクエリインスペクションが設定されている場合、App Firewall は特定のコンテンツタイプに対するクロスサイトスクリプティング攻撃に対するリクエストのクエリを検査します。GUI を使用する場合は、App Firewall プロファイルの [設定] タブでこのパラメータを構成できます。

> 重要:
>
> ストリーミングの変更の一環として、クロスサイトスクリプティングタグの Web App Firewall の処理が変更されました。この変更は 11.0 以降のビルドに適用されます。この変更は、リクエストサイドストリーミングをサポートする 10.5.e の拡張ビルドにも当てはまります。以前のリリースでは、開き括弧 (<), or close bracket (>)、または開き括弧と閉じ括弧 (<>) の両方が存在する場合、クロスサイトスクリプティング違反としてフラグが立てられていました。この動作は、リクエストサイドストリーミングのサポートを含むビルドで変更されました。閉じ括弧文字 (>) だけが攻撃と見なされなくなりました。開き括弧 (<) があってもリクエストはブロックされ、攻撃とみなされます。クロスサイトスクリプティング攻撃にはフラグが立てられます。

クロスサイトスクリプティングきめ細かなリラクゼーション

Web App Firewall には、クロスサイトスクリプティングインスペクションチェックから特定のフォームフィールド、ヘッダー、または Cookie を除外するオプションがあります。緩和ルールを設定することで、これらのフィールドの 1 つまたは複数のインスペクションを完全にバイパスできます。

Web App Firewall では、緩和ルールを微調整することで、より厳格なセキュリティを実装できます。アプリケーションには特定のパターンを許可する柔軟性が必要な場合がありますが、セキュリティインスペクションをバイパスするように緩和ルールを設定すると、ターゲットフィールドがクロスサイトスクリプティング攻撃パターンのインスペクションから除外されるため、アプリケーションが攻撃に対して脆弱になる可能性があります。クロスサイトスクリプティングのきめ細かなリラクゼーションには、特定の属性、タグ、パターンを許可するオプションがあります。残りの属性、タグ、パターンはブロックされます。たとえば、Web App Firewall には現在、125 個を超える拒否パターンがデフォルトで設定されています。ハッカーはクロスサイトスクリプト攻撃でこれらのパターンを使用する可能性があるため、Web App Firewall は潜在的な脅威としてフラグを立てます。特定の位置で安全と見なされる 1 つまたは複数のパターンをリラックスできます。その他の潜在的に危険なクロスサイトスクリプティングパターンは、ターゲットの場所で引き続きチェックされ、セキュリティチェック違反が引き続きトリガーされます。これで、より厳密な制御が可能になりました。

緩和で使用されるコマンドには、**[**値のタイプ**]** と **[**値の式] のオプションのパラメータがあります。値のタイプは空白のままにすることも、[タグ]、[属性]、または [パターン] を選択することもできます。値タイプを空白のままにすると、指定した URL の設定済みフィールドはクロスサイトスクリプティングチェックインスペクションから除外されます。値タイプを選択した場合は、値式を指定する必要があります。値式が正規表現かリテラル文字列かを指定できます。入力が許可リストと拒否リストと照合されると、緩和ルールで設定されている指定の式のみが除外されます。

Web App Firewall には、次のクロスサイトスクリプティングビルトインリストがあります。

1. クロスサイトスクリプティング許可される属性**: **abbr**、**accesskey**、**align**、**alt**、**axis**、**bgcolor**、ボーダー、セルパディング、セルなど、デフォルトで許可される属性は 52 種類あります。間隔、char、charoff、文字セットなど **
2. クロスサイトスクリプティング許可タグ: アドレス、**basefont**、**bgsound**、**big**、**blockquote**、**bg**、**br**、**caption**、**center** など、47 個のデフォルトで許可されているタグがあります。****** 引用、****dd**、**del** など

3. クロスサイトスクリプティング拒否パターン：**fsCommand**、**javascript:**、**onAbort**、onActivate** など、129 のデフォルト拒否パターンがあります。

> 警告
>
> Web App Firewall アクション URL は正規表現です。HTML クロスサイトスクリプティング緩和ルールを構成する場合、[名前] と [値の式] をリテラルまたは RegEx に指定できます。正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加するルールを正確に定義し、それ以外は何も定義していないことを確認してください。ワイルドカード、特にドットとアスタリスク (.*) メタ文字やワイルドカードの組み合わせを不注意に使用すると、意図していなかった Web コンテンツへのアクセスをブロックしたり、HTML クロスサイトスクリプティングチェックでブロックされていた攻撃を許可したりするなど、望ましくない結果になる可能性があります。

考慮すべきポイント**:**

- 値式はオプションの引数です。フィールド名には値式がない場合があります。
- フィールド名は複数の値式にバインドできます。
- 値式には値型を割り当てる必要があります。クロスサイトスクリプティング値のタイプには、1) タグ、2) 属性、3) パターンがあります。
- フィールド名と URL の組み合わせごとに複数の緩和ルールを設定できます。
- フォームフィールド名とアクション URL では、大文字と小文字は区別されません。

コマンドラインを使用した **HTML** クロスサイトスクリプティングチェックの設定

コマンドラインを使用して HTML クロスサイトスクリプティングのチェックアクションとその他のパラメータを設定するには

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して HTML クロスサイトスクリプティングチェックを設定できます。

- appfw プロファイルのトピックを設定します 。
- <name> -crossSiteScriptingAction (([block] [learn] [log] [stats]) | [**none**])
- [appfw プロファイルのトピックを設定します。
- <name> **-crossSiteScriptingTransformUnsafeHTML** (ON | OFF)
- appfw プロファイルのトピックを設定します。
- <name> -crossSiteScriptingCheckCompleteURLs (ON | OFF)
- appfw プロファイルのトピックを設定します。
-

HTML クロスサイトスクリプティングチェック

HTML クロスサイトスクリプティング (クロスサイトスクリプティング) チェックでは、クロスサイトスクリプティング攻撃の可能性について、ユーザーリクエストのヘッダーと POST 本文の両方を調べます。クロスサイトスクリプトが見つかった場合は、攻撃を無害化するようにリクエストを変更（変換）するか、リクエストをブロックします。

> 注:
>
> HTML クロスサイトスクリプティング (クロスサイトスクリプティング) チェックは、コンテンツタイプ、コンテンツの長さなどに対してのみ機能します。また、Web アプリケーションファイアウォールプロファイルで ‘checkRequestHeaders’ オプションが有効になっていることを確認します。

同じオリジンルールに違反する *HTML* クロスサイトスクリプティングスクリプトを使用することで、保護された *Web* サイトでのスクリプトの誤用を防ぐことができます。このルールでは、スクリプトが配置されているサーバー以外のサーバー上のコンテンツにアクセスしたり、コンテンツを変更したりしてはならないという規定があります。同一生成元ルールに違反するスクリプトはクロスサイトスクリプトと呼ばれ、スクリプトを使用して別のサーバー上のコンテンツにアクセスしたり変更したりすることをクロスサイトスクリプティングと呼びます。クロスサイトスクリプティングがセキュリティ上の問題である理由は、クロスサイトスクリプティングを許可する Web サーバーが、その Web サーバー上ではなく、攻撃者が所有および制御している別の Web サーバー上のスクリプトで攻撃される可能性があるためです。

残念ながら多くの企業では、同じオリジンルールに違反する JavaScript 拡張 Web コンテンツの大規模なインストールベースがあります。このようなサイトで HTML クロスサイトスクリプティングチェックを有効にする場合は、チェックが正当なアクティビティをブロックしないように、適切な例外を生成する必要があります。

Web App Firewall には、HTML クロスサイトスクリプティング保護を実装するためのさまざまなアクションオプションが用意されています。ブロック、ログ、統計、および学習アクションに加えて、送信されたリクエスト内のスクリプトタグをエンティティがエンコードすることにより、クロスサイトスクリプトを変換して攻撃を無害にするオプションもあります。[クロスサイトスクリプティングの完全な URL をチェック] パラメータを設定して、クエリパラメータだけでなく URL 全体を検査してクロスサイトスクリプティング攻撃を検出するかどうかを指定できます。**inspectQueryContentTypes** パラメータを設定して、特定のコンテンツタイプに対するクロスサイトスクリプティング攻撃のリクエストクエリ部分を検査できます。

リラクゼーションを展開すると、誤検出を回避できます。Web App Firewall 学習エンジンは、緩和ルールの設定に関する推奨事項を提供できます。

アプリケーションに最適化された HTML クロスサイトスクリプティング保護を設定するには、次のいずれかのアクションを設定します。

- ブロック—ブロックを有効にすると、リクエストでクロスサイトスクリプティングタグが検出されると、ブロックアクションがトリガーされます。
- [**Log**]: ログ機能を有効にすると、HTML クロスサイトスクリプティングチェックでは、実行したアクションを示すログメッセージが生成されます。ブロックを無効にすると、クロスサイトスクリプティング違反が検出されたヘッダーまたはフォームフィールドごとに個別のログメッセージが生成されます。ただし、要求がブロ

ックされると、1 つのメッセージだけが生成されます。同様に、クロスサイトスクリプティングタグが複数のフィールドで変換された場合でも、変換操作ではリクエストごとに 1 つのログメッセージが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとしたことを示している可能性があります。

- [**Stats**]: 有効にすると、統計機能は違反とログに関する統計情報を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当な要求がブロックされた場合は、新しい緩和ルールを構成する必要があるか、既存の緩和ルールを変更する必要があるかを再確認するために、構成を再確認する必要があります。
- 学習-どの緩和ルールがアプリケーションに適しているかわからない場合は、学習機能を使用して、学習データに基づいて HTML クロスサイトスクリプティングルールの推奨事項を生成できます。Web App Firewall 学習エンジンは、トラフィックを監視し、観測された値に基づいて学習の推奨事項を提供します。パフォーマンスを損なうことなく最適な効果を得るには、学習オプションを短時間有効にしてルールの代表的なサンプルを取得し、ルールを展開して学習を無効にすることをお勧めします。
- クロスサイトスクリプトの変換-有効にすると、Web App Firewall は HTML クロスサイトスクリプティングチェックに一致するリクエストに対して次の変更を加えます。
  - 左山かっこ (<) から HTML 文字エンティティに相当する (<)
  - 右山括弧 (>) を HTML 文字エンティティに相当 (>)

これにより、ブラウザが <!JEKYLL@5180@0> などの安全でない HTML タグを解釈して悪質なコードを実行することがなくなります。リクエストヘッダーのチェックと変換の両方を有効にすると、リクエストヘッダーで見つかった特殊文字もすべて変更されます。保護された Web サイト上のスクリプトにクロスサイトスクリプティング機能が含まれているが、Web サイトがこれらのスクリプトに依存して正しく動作しない場合は、ブロックを無効にして変換を有効にしても問題ありません。この構成により、クロスサイトスクリプティング攻撃の可能性を阻止しながら、正当な Web トラフィックがブロックされなくなります。

- クロスサイトスクリプティングの完全な **URL** を確認します。完全な URL のチェックが有効な場合、Web App Firewall は URL のクエリ部分だけをチェックするのではなく、HTML クロスサイトスクリプティング攻撃について URL 全体を検査します。
- **[**リクエストヘッダー**]** を確認します。リクエストヘッダーチェックが有効な場合、Web App Firewall は URL だけでなく HTML クロスサイトスクリプティング攻撃のリクエストのヘッダーを検査します。GUI を使用する場合は、Web App Firewall プロファイルの [設定] タブでこのパラメーターを有効にできます。
- **QueryContentTypes** を検査します。リクエストクエリインスペクションが設定されている場合、App Firewall は特定のコンテンツタイプに対するクロスサイトスクリプティング攻撃に対するリクエストのクエリを検査します。GUI を使用する場合は、App Firewall プロファイルの [設定] タブでこのパラメータを構成できます。

> 重要:
>
> ストリーミングの変更の一環として、クロスサイトスクリプティングタグの Web App Firewall の処理が変更されました。この変更は 11.0 以降のビルドに適用されます。この変更は、リクエストサイドストリーミングをサポートする 10.5.e の拡張ビルドにも当てはまります。以前のリリースでは、開き括弧 (<), or close bracket

> (>)、または開き括弧と閉じ括弧 (<>) の両方が存在する場合、クロスサイトスクリプティング違反としてフラグが立てられていました。この動作は、リクエストサイドストリーミングのサポートを含むビルドで変更されました。閉じ括弧文字 (>) だけが攻撃と見なされなくなりました。開き括弧 (<) があってもリクエストはブロックされ、攻撃とみなされます。クロスサイトスクリプティング攻撃にはフラグが立てられます。

クロスサイトスクリプティングきめ細かなリラクゼーション

Web App Firewall には、クロスサイトスクリプティングインスペクションチェックから特定のフォームフィールド、ヘッダー、または Cookie を除外するオプションがあります。緩和ルールを設定することで、これらのフィールドの 1 つまたは複数のインスペクションを完全にバイパスできます。

Web App Firewall では、緩和ルールを微調整することで、より厳格なセキュリティを実装できます。アプリケーションには特定のパターンを許可する柔軟性が必要な場合がありますが、セキュリティインスペクションをバイパスするように緩和ルールを設定すると、ターゲットフィールドがクロスサイトスクリプティング攻撃パターンのインスペクションから除外されるため、アプリケーションが攻撃に対して脆弱になる可能性があります。クロスサイトスクリプティングのきめ細かなリラクゼーションには、特定の属性、タグ、パターンを許可するオプションがあります。残りの属性、タグ、パターンはブロックされます。たとえば、Web App Firewall には現在、125 個を超える拒否パターンがデフォルトで設定されています。ハッカーはクロスサイトスクリプト攻撃でこれらのパターンを使用する可能性があるため、Web App Firewall は潜在的な脅威としてフラグを立てます。特定の位置で安全と見なされる 1 つまたは複数のパターンをリラックスできます。その他の潜在的に危険なクロスサイトスクリプティングパターンは、ターゲットの場所で引き続きチェックされ、セキュリティチェック違反が引き続きトリガーされます。これで、より厳密な制御が可能になりました。

緩和で使用されるコマンドには、**[**値のタイプ**]** と **[**値の式] のオプションのパラメータがあります。値のタイプは空白のままにすることも、[タグ]、[属性]、または [パターン] を選択することもできます。値タイプを空白のままにすると、指定した URL の設定済みフィールドはクロスサイトスクリプティングチェックインスペクションから除外されます。値タイプを選択した場合は、値式を指定する必要があります。値式が正規表現かリテラル文字列かを指定できます。入力が許可リストと拒否リストと照合されると、緩和ルールで設定されている指定の式のみが除外されます。

Web App Firewall には、次のクロスサイトスクリプティングビルトインリストがあります。

1. クロスサイトスクリプティング許可される属性**: **abbr**、**accesskey**、**align**、**alt**、**axis**、**bgcolor**、ボーダー、セルパディング、セルなど、デフォルトで許可される属性は 52 種類あります。間隔、char、charoff、文字セットなど **
2. クロスサイトスクリプティング許可タグ：アドレス、**basefont**、**bgsound**、**big**、**blockquote**、**bg**、**br**、**caption**、**center** など、47 個のデフォルトで許可されているタグがあります。**** 引用、****dd**、**del** など
3. クロスサイトスクリプティング拒否パターン：**fsCommand**、**javascript:**、**onAbort**、onActivate** など、129 のデフォルト拒否パターンがあります。

> 警告
>
> Web App Firewall アクション URL は正規表現です。HTML クロスサイトスクリプティング緩和ルールを構成する場合、[名前] と [値の式] をリテラルまたは RegEx に指定できます。正規表現は強力です。特に PCRE

> 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加するルールを正確に定義し、それ以外は何も定義していないことを確認してください。ワイルドカード、特にドットとアスタリスク (.*) メタ文字やワイルドカードの組み合わせを不注意に使用すると、意図していなかった Web コンテンツへのアクセスをブロックしたり、HTML クロスサイトスクリプティングチェックでブロックされていた攻撃を許可したりするなど、望ましくない結果になる可能性があります。

考慮すべきポイント**:**

- 値式はオプションの引数です。フィールド名には値式がない場合があります。
- フィールド名は複数の値式にバインドできます。
- 値式には値型を割り当てる必要があります。クロスサイトスクリプティング値のタイプには、1) タグ、2) 属性、3) パターンがあります。
- フィールド名と URL の組み合わせごとに複数の緩和ルールを設定できます。
- フォームフィールド名とアクション URL では、大文字と小文字は区別されません。

コマンドラインを使用した **HTML** クロスサイトスクリプティングチェックの設定

コマンドラインを使用して HTML クロスサイトスクリプティングのチェックアクションとその他のパラメータを設定するには

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して HTML クロスサイトスクリプティングチェックを設定できます。

- appfw プロファイルのトピックを設定します 。
- <!JEKYLL@5180@1>
- [appfw プロファイルのトピックを設定します。
- <!JEKYLL@5180@2>
- appfw プロファイルのトピックを設定します。
- <!JEKYLL@5180@3>
- appfw プロファイルのトピックを設定します。

- checkRequestHeaders \(ON | OFF)
- ‘ - CheckRequestQueryNonHtml (ON | OFF)‘
コマンドラインを使用して HTML クロスサイトスクリプティングチェック緩和ルールを構成するには
バインドを追加または削除するには、次のように bind または unbind コマンドを使用します。
- ‘bind appfw profile -crossSiteScripting [isRegex (REGEX | NOTREGEX)] [-location] [-valueType (Tag|Attribute|Pattern) [] [-isValueRegex (REGEX | NOTREGEX)]]‘
- ‘unbind appfw profile -crossSiteScripting [-location] [-valueType (Tag |Attribute|Pattern) []]‘
GUI を使用した HTML クロスサイトスクリプティングチェックの設定
GUI では、アプリケーションに関連付けられているプロファイルのペインで HTML クロスサイトスクリプティングチェックを設定できます。
GUI を使用して HTML クロスサイトスクリプティングチェックを構成または変更するには

1. [** アプリケーションファイアウォール **] > [** プロファイル **] に移動し、ターゲットプロファイルを強調表示して、[** 編集 **] をクリックします。
1. [** 詳細設定 **] ペインで、[** セキュリティチェック **] をクリックします。
セキュリティー検査テーブルには、すべてのセキュリティー検査に対して現在構成されているアクション設定が表示されます。設定には次の 2 つのオプションがあります。
a. HTML クロスサイトスクリプティングの「** ブロック **」、「** ログ **」、「** 統計 **」、および「** 学習 ** 」アクションを有効または無効にする場合は、テーブル内のチェックボックスをオンまたはオフにして、「**OK」** をクリックし、「** 保存して閉じる ** 」をクリックして **[セキュリティチェック **] ペイン。
b. このセキュリティー検査のオプションをさらに構成する場合は、[**HTML クロスサイトスクリプティング **] をダブルクリックするか、行を選択して [** アクションの設定 **] をクリックし、次のオプションを表示します。
** クロスサイトスクリプトの変換 **-安全でないスクリプトタグを変換します。
** 完全な URL でクロスサイトスクリプティングを確認する **-URL のクエリ部分だけをチェックするのではなく、完全な URL でクロスサイトスクリプト違反をチェックします。
上記の設定のいずれかを変更したら、「**OK」** をクリックして変更内容を保存し、「セキュリティチェック」(Security Checks) テーブルに戻ります。必要に応じて、他のセキュリティー検査の設定に進むことができます。［**OK** ］をクリックして［** セキュリティチェック ** ］セクションで行った変更をすべて保存し、［** 保存して閉じる ** ］ ** をクリックしてセキュリティチェックウィンドウを閉じます ** 。
[** 要求ヘッダーの確認 **] 設定を有効または無効にするには、[** 詳細設定 **] ウィンドウで [** プロファイル設定 **] をクリックします。[** 共通設定 **] で、[** 要求ヘッダーの確認 **] チェックボックスをオンまたはオフにします。** ［OK］ ** をクリックします。[** プロファイル設定] ペインの右上にある **X** アイコンを使用してこのセクションを閉じるか、このプロファイルの設定が完了したら ** [** 完了 **] をクリックして [** アプリケーションファイアウォール **] > [** プロファイル **] に戻ることができます。
[**HTML 以外のクエリを確認する **] 設定を有効または無効にするには、[** 詳細設定 **] ウィンドウで [** プロファイル設定 **] をクリックします。[** 共通設定 **] で、[**HTML 以外のリクエストクエリを確認する **] チェックボックスをオンまたはオフにします。** ［OK］ ** をクリックします。[** プロファイル設定] ペインの右上にある [X] アイコンを使用してこのセクションを閉じるか、このプロファイルの構成が完了したら ** [** 完了 **] をクリックして [**App Firewall] > [プロファイル **] に戻ることができます。
GUI を使用して HTML クロスサイトスクリプティング緩和規則を構成するには
1. [** アプリケーションファイアウォール **] > [** プロファイル **] に移動し、ターゲットプロファイルを強調表示して、[** 編集 **] をクリックします。
1. [** 詳細設定 **] ウィンドウで、[** 緩和規則 **] をクリックします。
1. 「緩和規則」テーブルで、「**HTML クロスサイトスクリプティング ** 」エントリをダブルクリックするか、エントリを選択して「** 編集 **」をクリックします。
1. [**HTML クロスサイトスクリプティング緩和規則] ダイアログボックスで、緩和規則の **[** 追加 **]、**[編集]、[**** 削除 **]、[** 有効化 **]、または [** 無効化 **] の操作を実行します。
> ** 注 **
>
> 新しいルールを追加する場合、[**** 値タイプフィールド] で [** タグ **]、[** 属性 **]、または [** パターン **] オプションを選択しない限り、[値式 ****] フィールドは表示されません。

ビジュアライザーを使用して HTML クロスサイトスクリプティング緩和ルールを管理するには
すべての緩和規則をまとめて表示するには、[緩和規則] テーブルの [**HTML クロスサイトスクリプティング] 行をハイライト表示し **、[** ビジュアライザー **] をクリックします。デプロイされたリラクゼーションのビジュアライザーには、[** 新しいルールを追加 **] または **[既存のルールを編集]** のオプションがあります。ノードを選択し、緩和ビジュアライザの対応するボタンをクリックして、ルールのグループ ** を有効 ** または ** 無効に ** することもできます。
GUI を使用してクロスサイトスクリプティングパターンを表示またはカスタマイズするには
GUI を使用して、クロスサイトスクリプティングが許可される属性または許可されたタグのデフォルトリストを表示またはカスタマイズできます。クロスサイトスクリプティング拒否パターンのデフォルトリストを表示またはカスタマイズすることもできます。
デフォルトの一覧は、[** アプリケーションファイアウォール **] > [** 署名 **] > [既定の署名 ****] で指定されています。シグニチャオブジェクトをプロファイルにバインドしない場合、Default Signatures オブジェクトで指定されているデフォルトのクロスサイトスクリプティング許可リストと拒否リストが、クロスサイトスクリプティングセキュリティチェック処理のプロファイルによって使用されます。デフォルトのシグネチャオブジェクトで指定された Tags、Attributes、Patterns は読み取り専用です。編集や修正はできません。これらを変更または変更する場合は、Default Signatures オブジェクトのコピーを作成して、ユーザー定義シグニチャオブジェクトを作成します。新しいユーザ定義シグニチャオブジェクトの許可リストまたは拒否リストを変更し、カスタマイズされた許可リストと拒否リストを使用するトラフィックを処理するプロファイルでこのシグニチャオブジェクトを使用します。
1. デフォルトのクロスサイトスクリプティングパターンを表示するには:
a. [** アプリケーションファイアウォール **] > [** 署名 **] に移動し、[** 既定の署名 **] を選択して [** 編集 **] をクリックします。次に、[**SQL/クロスサイトスクリプティングパターンの管理 **] をクリックします。
「**SQL/クロスサイトスクリプティングパスの管理 ** 」テーブルには、クロスサイトスクリプティングに関連する次の 3 つの行が表示されます。
‘xss/allowed/attribute‘
‘xss/allowed/tag‘
‘xss/denied/pattern‘
b. 行を選択して [** 要素の管理 **] をクリックすると、Web App Firewall ** クロスサイトスクリプティングチェックで使用される対応するクロスサイトスクリプティング要素 ** (タグ、属性、パターン) が表示されます。
1. ** クロスサイトスクリプティング要素をカスタマイズするには **:User-Defined signature オブジェクトを編集して、許可されたタグ、許可される属性、および拒否されたパターンをカスタマイズできます。新しいエントリを追加したり、既存のエントリを削除したりできます。
a. [** アプリケーションファイアウォール **] > [** 署名 **] に移動し、ターゲットの [ユーザー定義署名] を選択し、[** 編集 **] をクリックします。[**SQL/クロスサイトスクリプティングパターンの管理 ****] をクリックして、[SQL/クロスサイトスクリプティングパスの管理 **] テーブルを表示します。
b. 対象のクロスサイトスクリプティング行を選択します。
i. [** 要素を管理 **] をクリックして、** 対応するクロスサイトスクリプティング要素を追加 **、** 編集 ** 、** または削除します ** 。
ii. ** 選択した行を削除するには ** 、[削除] をクリックします。
> ** 警告： **

>
> デフォルトのクロスサイトスクリプティング要素を削除または変更したり、クロスサイトスクリプティングパスを削除して行全体を削除したりする前に、注意が必要です。シグニチャルールとクロスサイトスクリプティングセキュリティチェックは、これらの要素に基づいて攻撃を検出し、アプリケーションを保護します。クロスサイトスクリプティングエレメントをカスタマイズすると、編集中に必要なパターンが削除されると、アプリケーションがクロスサイトスクリプティング攻撃に対して脆弱になる可能性があります。
HTML クロスサイトスクリプティング (クロスサイトスクリプティング) 違反の学習
学習を有効にすると、NetScaler Web App Firewall 学習エンジンはトラフィックを監視し、クロスサイトスクリプティング URL 違反を学習します。クロスサイトスクリプティング URL ルールを定期的に検査し、フォールスポジティブのシナリオがないか展開できます。
> ** 注：**
>
> クラスタ構成では、クロスサイトスクリプティング URL ルールをデプロイするには、すべてのノードが同じバージョンである必要があります。
Web App Firewall は、学習構成の一部として、きめ細かい HTML クロスサイトスクリプティング学習を提供します。学習エンジンは、観測された値タイプ (タグ、属性、パターン) と、入力フィールドで観察された対応する値式に関するレコメンデーションを作成します。ブロックされたリクエストをチェックして、現在のルールが制限が厳しく、緩和する必要があるかどうかを判断するだけでなく、学習エンジンによって生成されたルールを確認して、違反を引き起こしている値タイプと値式を判断し、で対処する必要があります。リラクゼーションルール。
> ** 注：**
>
> Web App Firewall の学習エンジンでは、名前の最初の 128 バイトしか区別できません。フォームに、最初の 128 バイトに一致する名前のフィールドが複数ある場合、学習エンジンはそれらを区別できないことがあります。同様に、展開された緩和ルールによって、HTML クロスサイトスクリプティングインスペクションからこのようなすべてのフィールドが誤って緩和される可能性があります。
>
> ** ヒント：**
>
> 12 文字を超えるクロスサイトスクリプティングタグは、正しく学習または記録されません。
>
> 学習のためにより長いタグ長が必要な場合は、**as_crosssite scripting_allowed_tags_list** に、長さ 'x' の大きな非表示タグを追加できます。
HTML クロスサイトスクリプティングの学習プロセスにより、クロスサイトスクリプティング攻撃の誤検出が減少します。学習を有効にすると、リクエスト内のすべての違反を学習でき、複数のタグ、属性、またはパターンに繰り返しなく緩和を適用できる可能性があります。
たとえば、ペイロードに 15 個のカスタムタグがあり、それぞれ違反が発生した場合、一度に 1 つのタグに緩和を適用するプロセスを繰り返すのではなく、違反としてフラグが立てられたすべてのタグにきめ細かい緩和を適用できます。
** シナリオ 1：学習が有効、ブロックが有効：**
このシナリオでは、NetScaler ADC アプライアンスはカスタムタグ/属性/パターンのすべての違反を学習し、要求

がブロックされ、各違反がログに記録されます。この動作は、フォームフィールド、ヘッダー、または Cookie で特定された違反に対しても一貫しています。
** シナリオ 2：学習が有効でブロックが無効：**
このシナリオでは、NetScaler ADC アプライアンスはカスタムタグ/属性/パターンの違反を学習し、各違反がログに記録されます。リクエストはブロックされません。この動作は、フォームフィールド、ヘッダー、または Cookie で特定された違反に対しても一貫しています。
コマンドラインインターフェイスを使用して学習データを表示または使用するには
コマンドプロンプトで、次のコマンドのいずれかを入力します。
- ‘show appfw learningdata crossSiteScripting‘
- ‘rm appfw learningdata -crossSiteScripting [] [ ]‘
- ‘export appfw learningdata **crossSiteScripting*‘
カスタムタグをバイパスするようにクロスサイトスクリプティングのファイングレインリラクゼーションを構成する
Web App Firewall プロファイルでクロスサイトスクリプティング緩和を構成して、許可リストにないカスタムタグ/属性/パターンをバイパスできます。
コマンドプロンプトで、次のように入力します。
‘bind appfw profile p1 -crossSiteScripting -valueType ‘
** 例:**
‘bind appfw profile profile1 -crossSiteScripting formfield1 http://1.1.1.1 -valueType Tag tag1‘
GUI を使用して学習済みデータを表示または使用するには
1. [** アプリケーションファイアウォール **] > [** プロファイル **] に移動し、ターゲットプロファイルを強調表示して、[** 編集 **] をクリックします。
2. [** 詳細設定 **] ペインで、[** 学習済みルール **] をクリックします。「学習済み規則」(Learn Rules) テーブルで **HTML クロスサイトスクリプティングエントリを選択してダブルクリックすると ** 、学習済み規則にアクセスできます。テーブルには、[** フィールド名]**、[** アクション URL**]、[** 値の種類 **]、[** 値]**、および [** ヒット **] 列が表示されます学習したルールを展開したり、緩和ルールとして展開する前にルールを編集したりできます。ルールを破棄するには、ルールを選択して「スキップ」(**Skip**) ボタンをクリックします。一度に編集できるルールは 1 つだけですが、展開またはスキップするルールは複数選択できます。
また、「学習した規則」テーブルで「**HTML クロスサイトスクリプティング ** 」エントリを選択し、「** ビジュアライザー ** 」をクリックして、学習したすべての違反の統合ビューを表示することで、学習した緩和の要約ビューを表示することもできます。ビジュアライザを使用すると、学習したルールを簡単に管理できます。1 つの画面でデータの包括的なビューを表示し、1 回のクリックでルールのグループに対するアクションの実行を容易にします。ビジュアライザーの最大の利点は、正規表現を推奨して複数のルールを統合できることです。デリミタとアクション URL に基づいて、これらのルールのサブセットを選択できます。ドロップダウンリストから番号を選択すると、ビジュアライザーに 25、50、または 75 個のルールを表示できます。学習したルールのビジュアライザーには、ルールを編集して緩和として展開するオプションがあります。または、ルールをスキップして無視することもできます。
HTML クロスサイトスクリプティングチェックでのログ機能の使用
ログアクションを有効にすると、HTML クロスサイトスクリプティングのセキュリティチェック違反が **AppFW_crosssite スクリプティング違反として監査ログに記録されます ** 。Web App Firewall は、ネイティ

ブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。
コマンドラインを使用してログメッセージにアクセスするには
シェルに切り替えて、‘/var/log/‘ フォルダ内の ns.logs を末尾に付けて、HTML クロスサイトスクリプティング違反に関するログメッセージにアクセスします。
> ‘Shell‘
> ‘tail -f /var/log/ns.log | grep APPFW_cross-site scripting‘
CEF ログ形式のクロスサイトスクリプティングセキュリティチェック違反ログメッセージの例：
“‘
Jul 11 00:45:51 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|**APPFW_cross-site scripting**|6|src=10.217.253.62 geolocation=Unknown spt=4840 method=GET request=http://aaron.stratum8.net/FFC msg=**Cross-site script check failed for field abc=”Bad tag: def”** cn1=133 cn2=294 cs1=pr_ffc cs2=PPE1 cs3=eUljypvLa0BbabwfGVE52Sewg9U0001 cs4=ALERT cs5=2015 act=**not blocked**
“‘
変換アクションを示すネイティブログ形式のクロスサイトスクリプティングセキュリティチェック違反ログメッセージの例
“‘
Jul 11 01:00:28 10.217.31.98 07/11/2015:01:00:28 GMT ns 0-PPE-0 : default APPFW **APPFW_cross-site scripting** 132 0 : 10.217.253.62 392-PPE0 eUljypvLa0BbabwfGVE52Sewg9U0001 pr_ffc http://aaron.stratum8.net/FFC/login.php?login_name=%3CBOB%3E&passwd=&drinking_pref=on &text_area=&loginButton=ClickToLogin&as_sfid=AAAAAAVFqmYL68IGvkrcn2pzehjfIkm5E6EZ9FL8YLvIW_41AvAAT FC4llF **Cross-site script special characters seen in fields **
“‘
GUI を使用してログメッセージにアクセスする
GUI には、ログメッセージを分析するための便利なツール (Syslog Viewer) が含まれています。Syslog ビューアにアクセスするには、複数のオプションがあります。
- [** アプリケーションファイアウォール **] > [** プロファイル **] に移動し、ターゲットプロファイルを選択して [** セキュリティチェック **] をクリックします。[**HTML クロスサイトスクリプティング **] 行を強調表示し、[** ログ **] をクリックします。プロファイルの HTML クロスサイトスクリプティングチェックからログに直接アクセスすると、GUI によってログメッセージが除外され、これらのセキュリティチェック違反に関するログのみが表示されます。
- 「**NetScaler** 」>「** システム ** 」>「** 監査 ** 」の順に選択して、Syslog ビューアにアクセスすることもできます。** 監査メッセージセクションで ** 、**Syslog メッセージリンクをクリックして ** Syslog Viewer を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティチェック違反がトリガーされる可能性がある場合のデバッグに役立ちます。
- ** アプリケーションファイアウォール ** > ** ポリシー ** > ** 監査に移動します **。監査メッセージセクションで、**Syslog メッセージリンクをクリックして ** Syslog Viewer を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。
HTML ベースの Syslog ビューアには、関心のあるログメッセージのみを選択するためのさまざまなフィルタオプションがあります。**HTML クロスサイトスクリプティングチェックのログメッセージを選択するには ** 、[** モジュ

ール **] のドロップダウンリストオプションで [**APPFW**] を選択してフィルタします。**[Event Type]** リストには、選択内容をさらに絞り込むための豊富なオプションが用意されています。たとえば、**AppFW_Cross-site scripting** チェックボックスを選択して [**Apply**] ボタンをクリックすると、**HTML クロスサイトスクリプティングのセキュリティチェック違反に関するログメッセージのみが ** Syslog Viewer に表示されます。
特定のログメッセージの行にカーソルを置くと、[** モジュール **]、[** イベントタイプ **]、[イベント **ID**]、[** クライアント IP**] などの複数のオプションがログメッセージの下に表示されます。これらのオプションを選択すると、ログメッセージ内の対応する情報を強調表示できます。
**[クリックして展開 **] 機能は GUI でのみ使用できます。Syslog Viewer を使用すると、ログを表示するだけでなく、Web App Firewall セキュリティチェック違反のログメッセージに基づいて HTML クロスサイトスクリプティング緩和ルールを展開することもできます。この操作では、ログメッセージは CEF ログ形式である必要があります。クリックして展開機能は、ブロック (またはブロックしない) アクションによって生成されたログメッセージに対してのみ使用できます。変換操作に関するログメッセージの緩和ルールは展開できません。
Syslog Viewer から緩和ルールを展開するには、ログメッセージを選択します。選択した行の [**Syslog Viewer**] ボックスの右上隅にチェックボックスが表示されます。このチェックボックスをオンにし、[** アクション **] リストから緩和ルールを展開するオプションを選択します。**[** 編集とデプロイ ******]、[** デプロイ]、[すべてデプロイ **] は、**
[**Click to Deploy**] オプションを使用して展開された HTML クロスサイトスクリプティングルールには、きめ細かな緩和の推奨事項は含まれていません。
GUI を使用してクリックして展開機能を構成する
1. Syslog ビューアの [** モジュール **] オプションで [**APPFW**] を選択します。
1. 対応するログメッセージをフィルタリングするには **、[** イベントタイプ **] として [App_cross-Site] スクリプトを選択します ** 。
1. 展開するルールを識別するには、このチェックボックスをオンにします。
1. オプションの [** アクション ** （Action）] ドロップダウンリストを使用して、緩和ルールを展開します。
1. ルールが対応する [緩和ルール] セクションに表示されていることを確認します。
HTML クロスサイトスクリプティング違反の統計
統計アクションを有効にすると、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、HTML クロスサイトスクリプティングチェックのカウンタが増分します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分サイズは、構成された設定によって異なります。たとえば、ブロックアクションが有効になっている場合、3 つの HTML クロスサイトスクリプティング違反を含むページのリクエストでは、最初の違反が検出されたときにページがブロックされるため、統計カウンタが 1 つ増えます。ただし、ブロックが無効になっている場合、同じ要求を処理すると、違反ごとに個別のログメッセージが生成されるため、違反とログの統計カウンタが 3 ずつ増加します。
コマンドラインを使用して HTML クロスサイトスクリプティングチェック統計を表示するには
コマンドプロンプトで入力します。
‘> sh appfw stats‘
特定のプロファイルの統計情報を表示するには、以下のコマンドを使用します。
‘> **stat appfw profile** ‘
GUI を使用して HTML クロスサイトスクリプティングの統計情報を表示する

1. ** セキュリティ ** > ** アプリケーションファイアウォール > プロファイル > 統計に移動します **。
1. 右側のペインで、[** 統計 ** リンク] にアクセスします。
1. スクロールバーを使用して、HTML クロスサイトスクリプティング違反とログに関する統計情報を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

ハイライト

- **HTML クロスサイトスクリプティング攻撃保護の組み込みサポート **: NetScaler Web App Firewall は、受信したペイロード内の許可された属性とタグ、および拒否されたパターンの組み合わせを監視することにより、クロスサイトスクリプティング攻撃から保護します。クロスサイトスクリプティングチェックで使用される組み込みのデフォルト許可タグ、許可される属性、および拒否パターンは、すべて /netscaler/default_custom_settings.xml ファイルに指定されています。
- [**Customizatio**n]: タグ、属性、およびパターンのデフォルトリストを変更して、アプリケーションの特定のニーズに合わせてクロスサイトスクリプティングセキュリティチェックインスペクションをカスタマイズできます。デフォルトのシグニチャオブジェクトのコピーを作成するか、既存のエントリを変更するか、新しいシグニチャオブジェクトを追加します。このシグニチャオブジェクトをプロファイルにバインドして、カスタマイズした設定を利用します。
- ** ハイブリッドセキュリティモデル **-シグニチャとディープセキュリティ保護の両方で、プロファイルにバインドされたシグニチャオブジェクトで指定された SQL/Cross-Site スクリプティングパターンが使用されます。シグニチャオブジェクトがプロファイルにバインドされていない場合は、デフォルトのシグニチャオブジェクトに存在する SQL/Cross-Site スクリプティングパターンが使用されます。
- [**Transform**]: 変換操作について、次の点に注意してください。
  トランスフォーム操作は、他のクロスサイトスクリプティングアクション設定とは無関係に機能します。変換が有効で、ブロック、ログ、統計、学習がすべて無効になっている場合、クロスサイトスクリプティングタグは変換されます。ブロックアクションが有効になっている場合は、変換アクションよりも優先されます。
- ** きめ細かなリラクゼーションと学習 **。緩和ルールを微調整して、クロスサイトスクリプティング要素のサブセットをセキュリティチェックインスペクションから解放し、残りは検出します。学習エンジンは、観測されたデータに基づいて、特定の値のタイプと値の式を推奨しています。
- [**Click to Deploy**]: Syslog Viewer で 1 つまたは複数のクロスサイトスクリプティング違反ログメッセージを選択し、緩和ルールとして展開します。
- **Charset**: アプリケーションの必要性に基づいて、プロファイルのデフォルトの文字セットを設定する必要があります。デフォルトでは、プロファイル文字セットは英語 (US) (ISO-8859-1) に設定されています。指定された文字セットなしでリクエストを受信すると、Web App Firewall はそのリクエストを ISO-8859-1 であるかのように処理します。開き括弧文字 (<) or the close bracket character (>) は、他の文字セットでエンコードされている場合、クロスサイトスクリプティングタグとして解釈されません。たとえば、リクエストに UTF-8 文字列”%uff1cscript%uff1e**”** が含まれているが、リクエストページで文字セットが指定されていない場合、プロファイルのデフォルトの文字セットが Unicode として指定されていない限り、クロスサイトスクリプティング違反はトリガーされない可能性があります。

HTML SQL インジェクションチェック

March 20, 2024

多くの Web アプリケーションには、SQL を使用してリレーショナルデータベースサーバーと通信する Web フォームがあります。悪意のあるコードやハッカーが、安全ではない Web フォームを使用して SQL コマンドを Web サーバーに送信する可能性があります。Web App Firewall HTML SQL インジェクションチェックは、セキュリティを侵害する可能性のある不正な SQL コードのインジェクションに対する特別な防御を提供します。Web App Firewall は、ユーザーリクエストで不正な SQL コードを検出すると、リクエストを変換して SQL コードを非アクティブにするか、リクエストをブロックします。Web App Firewall は、1) POST 本文、2) ヘッダー、および 3) Cookie の 3 つの場所に挿入された SQL コードの要求ペイロードを調べます。注入された SQL コードのリクエストのクエリ部分を調べるには、特定のコンテンツタイプに対してアプリケーションファイアウォールプロファイル設定 'inspectQueryContentTypes' を構成してください。

キーワードと特殊文字のデフォルトのセットは、SQL 攻撃の起動に一般的に使用される既知のキーワードと特殊文字を提供します。新しいパターンを追加したり、デフォルトセットを編集して SQL チェックインスペクションをカスタマイズしたりできます。Web App Firewall には、SQL インジェクション保護を実装するためのさまざまなアクションオプションが用意されています。Web App Firewall プロファイルには、ブロック、ログ、統計、学習の各アクションに加えて、**SQL** 特殊文字を変換して攻撃を無害にするオプションも用意されています。

アクションに加えて、SQL インジェクション処理用に構成できるパラメーターがいくつかあります。**SQL** ワイルドカード文字をチェックできます。SQL インジェクションタイプを変更し、4 つのオプション (**sqlKeyword**、sqlSPLChar**、****sqlSplCharandKeyword**、**sqlSPLCharorKeyword**) のいずれかを選択して、ペイロードの処理時に SQL キーワードと SQL 特殊文字を評価する方法を指定できます。[**SQL Comments Handling]** パラメーターには、SQL インジェクション検出中に検査または除外する必要のあるコメントのタイプを指定するオプションがあります。

リラクゼーションを展開すると、誤検出を回避できます。Web App Firewall 学習エンジンは、緩和ルールの設定に関する推奨事項を提供できます。

アプリケーションに最適化された SQL インジェクション保護を構成するには、次のオプションを使用できます。

Block—入力が SQL インジェクションタイプの仕様と一致する場合にのみブロックアクションがトリガーされます。たとえば、**SQLSplCharANDKeyword** が SQL インジェクションタイプとして設定されている場合、入力で SQL 特殊文字が検出された場合でも、リクエストにキーワードが含まれていなくてもリクエストはブロックされません。SQL インジェクションタイプが **sqlSPLChar** または sqlSPLCharorKeyword** のいずれかに設定されている場合、このようなリクエストはブロックされます。

Log: ログ機能を有効にすると、SQL インジェクションチェックによって実行されるアクションを示すログメッセージが生成されます。ブロックアクションが無効になっている場合、SQL 違反が検出された入力フィールドごとに個別のログメッセージが生成されます。ただし、要求がブロックされると、1 つのメッセージだけが生成されます。同様に、SQL 特殊文字が複数のフィールドで変換された場合でも、変換操作に対してリクエストごとに 1 つのログメッセージが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ロ

グメッセージの数が大幅に増加すると、攻撃を開始しようとしたことを示している可能性があります。

[**Stats**]：有効にすると、統計機能は違反とログに関する統計情報を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当なリクエストがブロックされる場合、新しい緩和ルールを構成する必要があるか、既存の緩和ルールを変更する必要があるかを確認するために、構成を再確認しなければならない場合があります。

学習—どの SQL 緩和ルールがアプリケーションに適しているかわからない場合は、学習機能を使用して、学習したデータに基づいて推奨事項を生成できます。Web App Firewall 学習エンジンはトラフィックを監視し、観測された値に基づいて SQL 学習の推奨事項を提供します。パフォーマンスを損なうことなく最適な効果を得るには、学習オプションを短時間有効にしてルールの代表的なサンプルを取得し、ルールを展開して学習を無効にすることをお勧めします。

SQL 特殊文字の変換—Web App Firewall では、一重引用符 (‘)、バックスラッシュ (\ )、セミコロン (;) の 3 つの文字を SQL セキュリティチェック処理の特殊文字と見なします。SQL 変換機能は、HTML リクエストの SQL インジェクションコードを変更して、リクエストが無害になるようにします。変更された HTML リクエストはサーバーに送信されます。デフォルトの変換ルールはすべて /netscaler/default_custom_settings.xml ファイルに指定されています。

変換操作では、リクエストに以下の変更を加えることで、SQL コードが非アクティブになります。

- 一重引用符 (‘) から二重引用符 (「)。
- バックスラッシュ (\ ) をダブルバックスラッシュ (\ ) にします。
- セミコロン (;) は完全に削除されます。

この 3 つの文字 (特殊文字列) は、SQL Server にコマンドを発行するために必要です。SQL コマンドの前に特別な文字列を付けない限り、ほとんどの SQL サーバーではそのコマンドが無視されます。したがって、変換が有効な場合に Web App Firewall が実行する変更により、攻撃者はアクティブな SQL を挿入できなくなります。これらの変更が加えられた後、リクエストは保護された Web サイトに安全に転送されます。保護された Web サイト上の Web フォームが SQL 特殊文字列を正当に含むことができるが、Web フォームが正しく動作するために特殊文字列に依存しない場合、Web App Firewall が保護された Web サイトに提供する保護を低下させずに、ブロックを無効にして変換を有効にして、正当な Web フォームデータのブロックを防止できます。

変換操作は **SQL** インジェクションタイプの設定とは独立して機能します。変換が有効で、SQL インジェクションタイプが SQL キーワードとして指定されている場合、リクエストにキーワードが含まれていなくても SQL 特殊文字が変換されます。

> ヒント
>
> 通常、変換とブロッキングのどちらかを有効にしますが、両方を有効にすることはできません。ブロックアクションが有効になっている場合は、変換アクションよりも優先されます。ブロッキングを有効にしている場合、変換の有効化は冗長です。

SQL ワイルドカード文字の確認-ワイルドカード文字を使用して、SQL (SQL-SELECT) ステートメントの選択範囲を広げることができます。これらのワイルドカード演算子は、[いいね！] および [**NOT** いいね] 演算子と共に使用し

て、値を類似の値と比較できます。パーセント (%) およびアンダースコア (_) 文字は、ワイルドカードとしてよく使用されます。パーセント記号は、MS-DOS で使用されるアスタリスク (*) ワイルドカード文字に似ており、フィールド内の 0 文字、1 文字、または複数の文字に一致します。アンダースコアは MS-DOS の疑問符 (?) と似ています。ワイルドカード文字。これは、式の 1 つの数字または文字に一致します。

たとえば、次のクエリを使用して文字列検索を実行し、名前に D 文字が含まれるすべての顧客を検索できます。

「**%D%**」のような名前のカスタマーから ***** を選択してください。:

次の例では、演算子を組み合わせて、2 番目と 3 番目に 0 がある給与値をすべて検索します。

顧客から ***** を選択 **WHERE** 給与‘**_ 00%**’:

DBMS ベンダーによっては、演算子を追加してワイルドカード文字を拡張しています。NetScaler Web App Firewall は、これらのワイルドカード文字を挿入することによって開始される攻撃から保護できます。デフォルトの 5 つのワイルドカード文字は、パーセント (%)、アンダースコア (_)、キャレット (^)、開き角かっこ ([)、閉じ角かっこ (]) です。この保護は、HTML プロファイルと XML プロファイルの両方に適用されます。

デフォルトのワイルドカード文字は、***Default Signatures** で指定されたリテラルのリストです。

- <wildchar type=” LITERAL” >%</wildchar>
- <wildchar type=” LITERAL” >_</wildchar>
- <wildchar type=” LITERAL” >^</wildchar>
- <wildchar type=” LITERAL” >[</wildchar>
- <wildchar type=” LITERAL” >]</wildchar>

攻撃のワイルドカード文字は [^A-F] のように PCRE になります。Web App Firewall は PCRE ワイルドカードもサポートしていますが、ほとんどの攻撃をブロックするには、上記のリテラルワイルドカード文字で十分です。

> 注:
>
> SQL ワイルドカード文字のチェックは、SQL の特殊文字チェックとは異なります。誤検出を避けるため、このオプションは注意して使用する必要があります。

SQL インジェクションタイプを含むリクエストの確認—Web App Firewall には、アプリケーションの個々のニーズに基づいて、SQL インジェクションインスペクションに必要なレベルの厳格さを実装するための 4 つのオプションが用意されています。SQL 違反を検出するために、リクエストはインジェクションタイプの指定と照合されます。SQL インジェクションタイプには、次の 4 つのオプションがあります。

- **SQL** 特殊文字とキーワード: SQL 違反をトリガーするには、SQL キーワードと SQL 特殊文字の両方を入力に含める必要があります。この最も制限の少ない設定もデフォルト設定です。
- **SQL** 特殊文字-SQL 違反をトリガーするには、入力に少なくとも 1 つの特殊文字が含まれている必要があります。
- **SQL** キーワード: SQL 違反をトリガーするには、指定した SQL キーワードのうち少なくとも 1 つが入力に存在している必要があります。このオプションは十分に考慮せずに選択しないでください。誤検出を避けるため、入力にキーワードが含まれていないことを確認します。

- **SQL** 特殊文字またはキーワード：セキュリティチェック違反をトリガーするには、キーワードまたは特殊文字列のいずれかが入力に含まれている必要があります。

> ヒント：
>
> SQL 特殊文字を含む入力をチェックするように Web App Firewall を構成すると、Web アプリケーションファイアウォールは特殊文字を含まない Web フォームフィールドをスキップします。ほとんどの SQL サーバーは、前に特殊文字が付いていない SQL コマンドを処理しないため、このオプションを有効にすると、Web App Firewall の負荷を大幅に軽減し、保護された Web サイトを危険にさらすことなく処理を高速化できます。

SQL コメントの処理-デフォルトでは、Web App Firewall はすべての SQL コメントに注入された SQL コマンドをチェックします。ただし、多くの SQL サーバーでは、SQL 特殊文字が前に付いていても、コメント内の内容は無視されます。処理を高速化するために、SQL サーバーがコメントを無視する場合、挿入された SQL のリクエストを調べるときにコメントをスキップするように Web App Firewall を構成できます。SQL コメント処理オプションは次のとおりです。

- **ANSI**—UNIX ベースの SQL データベースで通常使用される ANSI 形式の SQL コメントをスキップします。例：
  - —(2 つのハイフン)-これは、2 つのハイフンで始まり、行末で終わるコメントです。
  - {}-中カッコ（中カッコはコメントを囲みます。{はコメントの前にあり、} はその後に続きます。中括弧は 1 行または複数行のコメントを区切ることができますが、コメントはネストできません)
  - `/**/: C style comments (Does not allow nested comments). Please note /*! <comment that begin with slash followed by asterisk and exclamation mark is not a comment > */`
  - MySQL Server は C スタイルのコメントのいくつかのバリアントをサポートしています。これらのコードを使用すると、MySQL 拡張を含むが、移植可能なコードを、次の形式のコメントを使用して記述できます。`/*! MySQL-specific code */`
  - . #: Mysql コメント: これは # 文字で始まるコメントです。
- ネスト: ネストされた SQL コメントをスキップします。このコメントは、Microsoft SQL Server で通常使用されます。たとえば、; —(2 つのハイフン)、**/* */** (ネストされたコメントを許可します)
-
- **[**すべてのコメントをチェック]: 何もスキップせずに、注入された SQL のリクエスト全体をチェックします。これがデフォルトの設定です。

> ヒント
>
> 通常、バックエンドデータベースが Microsoft SQL Server 上で実行されていない限り、[ネスト] または [ANSI/ネスト] オプションを選択しないでください。他のほとんどの種類の SQL Server ソフトウェアは、ネストされたコメントを認識しません。ネストされたコメントが、別の種類の SQL Server 宛ての要求に表示される場合は、そのサーバーのセキュリティ侵害の試みを示している可能性があります。

[**Check Request headers**]：フォームフィールドの入力の検査に加えて、HTML SQL インジェクション攻撃のリクエストヘッダーを調べる場合は、このオプションを有効にします。GUI を使用する場合は、Web App Firewall プロファイルの **[詳細設定]-> [プロファイル設定]** ペインでこのパラメーターを有効にすることができます。

> 注：
>
> Check Request ヘッダーフラグを有効にすると、**User-Agent** ヘッダーの緩和ルールを設定する必要がある場合があります。のような **SQL** キーワードや、SQL 特殊文字のセミコロン（**;**）が存在すると、偽陽性が発生し、このヘッダーを含む要求がブロックされる可能性があります。
> 警告
>
> リクエストヘッダーのチェックと変換の両方を有効にすると、ヘッダーで見つかった SQL 特殊文字も変換されます。受け入れ、受け入れ文字セット、受け入れエンコーディング、受け入れ言語、期待、およびユーザーエージェントヘッダーは、通常、セミコロン（;）が含まれています。Request ヘッダーのチェックと変換を同時に有効にすると、エラーが発生する場合があります。

inspectQueryContentTypes —特定のコンテンツタイプに対する SQL インジェクション攻撃のリクエストクエリ部分を調べる場合は、このオプションを設定します。GUI を使用する場合は、App Firewall プロファイルの [詳細設定]-> [プロファイル設定] ペインでこのパラメーターを構成できます。

SQL ファイングレインリラクゼーション

Web App Firewall では、SQL インジェクションインスペクションチェックから特定のフォームフィールド、ヘッダー、または Cookie を除外するオプションがあります。SQL インジェクションチェックのリラクゼーションルールを設定することで、これらのフィールドの 1 つ以上のインスペクションを完全にバイパスできます。

Web App Firewall では、緩和ルールを微調整することで、より厳格なセキュリティを実装できます。アプリケーションでは、特定のパターンを許可する柔軟性が要求される場合がありますが、セキュリティインスペクションをバイパスするように緩和規則を設定すると、ターゲットフィールドが SQL 攻撃パターンの検査から免除されるため、アプリケーションが攻撃に対して脆弱になる可能性があります。SQL のきめ細かい緩和は、特定のパターンを許可し、残りをブロックするオプションを提供します。たとえば、Web App Firewall には現在、100 を超える SQL キーワードのデフォルトセットがあります。ハッカーは SQL Injection 攻撃でこれらのキーワードを使用できるため、Web App Firewall は潜在的な脅威としてフラグを立てます。特定の場所で安全と見なされる 1 つ以上のキーワードをリラックスできます。潜在的に危険な SQL キーワードの残りの部分は、ターゲットの場所がチェックされ、セキュリティチェック違反が引き続きトリガーされます。これで、より厳密な制御が可能になりました。

リラクゼーションで使用されるコマンドには、[値タイプ] と **[**** 値式] のオプションパラメータがあります。値式が正規表現かリテラル文字列かを指定できます。値の型は空白のままにすることも、[キーワード]、[SpecialString]、または [WildChar]** を選択することもできます。

> 警告：
>
> 正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加する URL を正確に定義していることを確認し、それ以外は何も定義しないでくだ

> さい。ワイルドカード、特にドットアスタリスク（.*）メタ文字またはワイルドカードの組み合わせを不注意に使用すると、ブロックする意図がない Web コンテンツへのアクセスをブロックしたり、HTML SQL インジェクションチェックでブロックされた攻撃を許可するなど、望ましくない結果が生じる可能性があります。

考慮すべきポイント**:**

- 値式はオプションの引数です。フィールド名には値式がない場合があります。
- フィールド名は複数の値式にバインドできます。
- 値式には値型を割り当てる必要があります。SQL 値の型は、1) キーワード、2) SpecialString、または 3) WildChar のいずれかになります。
- フィールド名と URL の組み合わせごとに複数の緩和ルールを設定できます。

コマンドラインを使用した **SQL** インジェクションチェックの設定

コマンドラインを使用して SQL インジェクションアクションとその他のパラメーターを構成するには、次の手順を実行します。

コマンドラインインターフェイスでは、**set appfw profile** コマンドまたは **add appfw profile** コマンドのいずれかを使用して、SQL インジェクション保護を設定できます。ブロック、学習、ログ、統計アクションを有効にし、SQL インジェクション攻撃文字列で使用される特殊文字を変換して攻撃を無効にするかどうかを指定できます。ペイロードで検出する SQL 攻撃パターンの種類 (キーワード、ワイルドカード文字、特殊文字列) を選択し、Web App Firewall で SQL インジェクション違反のリクエストヘッダーも検査するかどうかを指定します。**unset appfw profile** コマンドを使用して、構成した設定をデフォルトに戻します。次のコマンドはそれぞれ 1 つのパラメータのみを設定しますが、1 つのコマンドに複数のパラメータを含めることができます。

- ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- <name> -SQLInjectionAction (([block] [learn] [log] [stats])| [none])
- ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- <name> -SQLInjectionTransformSpecialChars (**ON** | OFF)
- ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- <name> -**SQLInjectionCheckSQLWildChars** (**ON** |**OFF**)
- ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- **<name> -**SQLInjectionType** ([**SQLKeyword**] | [**SQLSplChar**] | [**SQLSplCharANDKeyword**] | [**SQLSplCharORKeyword**])
- ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。

- <name> -**SQLInjectionParseComments** ([**checkall**] | [**ansi|nested**] | [**ansinested**])
- **ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- <name> -CheckRequestHeaders (ON | OFF) ページの下部に表示されるパラメータの説明。
- <name> - CheckRequestQueryNonHtml (ON | OFF) ページの下部に表示されるパラメータの説明。

コマンドインターフェイスを使用して SQL インジェクション緩和ルールを設定するには

バインドを追加または削除するには、次のように bind または unbind コマンドを使用します。

- bind appfw profile <name> -SQLInjection <String> [isRegex(REGEX|NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor|SpecialString|Wildchar)[<valueExpression>][-isValueRegex (REGEX | NOTREGEX)]]
- unbind appfw profile <name> -SQLInjection <String> <formActionURL> [-location <location>] [-valueTyp (Keyword|SpecialString|Wildchar)[<valueExpression>]]

> 注:
>
> SQL キーワードと SQL 特殊文字のリストを含むビューシグニチャオブジェクトを表示することで、デフォルトのシグニチャファイルのコンテンツから SQL キーワードのリストを検索できます。

GUI を使用した SQL インジェクションのセキュリティー検査の設定

GUI では、アプリケーションに関連付けられたプロファイルのペインで SQL Injection セキュリティー検査を構成できます。

GUI を使用して SQL インジェクションチェックを構成または変更するには

1. [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ペインで、[セキュリティチェック] をクリックします。

セキュリティー検査テーブルには、すべてのセキュリティー検査に対して現在構成されているアクション設定が表示されます。設定には次の 2 つのオプションがあります。

a. HTML SQL インジェクションの [ブロック]、[ログ]、[統計]、および [学習] の各アクションを有効または無効にするには、テーブルのチェックボックスをオンまたはオフにして、[**OK**] をクリックし、[保存して閉じる] をクリックして [セキュリティチェック] ウィンドウを閉じます。

b. このセキュリティ・チェックのその他のオプションを構成する場合は、「HTML SQL インジェクション」をダブルクリックするか、行を選択して「アクションの設定」をクリックして、次のオプションを表示します。

SQL 特殊文字の変換：要求内の任意の SQL 特殊文字を変換します。

SQL ワイルドカード文字の確認—ペイロード内の SQL ワイルドカード文字を攻撃パターンと見なします。

「次を含むリクエストのチェック」—チェックする SQL インジェクションのタイプ (sqlKeyword、sqlSPLChar、sqlSPLCharandKeyword、または sqlSPLCharorKeyword)。

SQL コメントの処理-チェックするコメントのタイプ ([すべてのコメントをチェック]、[ANSI]、[ネスト]、または [ANSI/ネスト])。

上記の設定のいずれかを変更したら、「**OK**」をクリックして変更内容を保存し、「セキュリティチェック」(Security Checks) テーブルに戻ります。必要に応じて、他のセキュリティー検査の設定に進むことができます。[**OK**] をクリックして［セキュリティチェック］セクションで行った変更をすべて保存し、［保存して閉じる］をクリックしてセキュリティチェックウィンドウを閉じます。

GUI を使用して SQL インジェクション緩和ルールを構成するには

- [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを強調表示して [編集] をクリックします。
- [詳細設定] ウィンドウで、[緩和規則] をクリックします。
- 「緩和規則」テーブルで、「**HTML SQL Injection** 」エントリをダブルクリックするか、エントリを選択して「編集」をクリックします。
- 「**HTML SQL** インジェクション緩和規則」ダイアログで、緩和規則の「追加」、「編集」、「削除」、「有効化」、または「無効化」

> 注
>
> 新しいルールを追加すると、[値の型フィールド] で **[**キーワード**]** または **[**SpecialString**]** または **[**WildChar**]** オプションを選択しない限り、**[**値の ** 式 **] フィールドは表示されません。

ビジュアライザーを使用して SQL インジェクション緩和ルールを管理するには

すべての緩和ルールをまとめて表示するには、[**HTML SQL Injection**] 行をハイライト表示して [ビジュアライザー] をクリックします。デプロイされたリラクゼーションのビジュアライザーには、[新しいルールを追加] または **[**既存のルールを編集**]** のオプションがあります。ノードを選択し、緩和ビジュアライザの対応するボタンをクリックして、ルールのグループ を有効 または 無効に することもできます。

GUI を使用して射出パターンを表示またはカスタマイズする

GUI を使用して、射出パターンを表示またはカスタマイズできます。

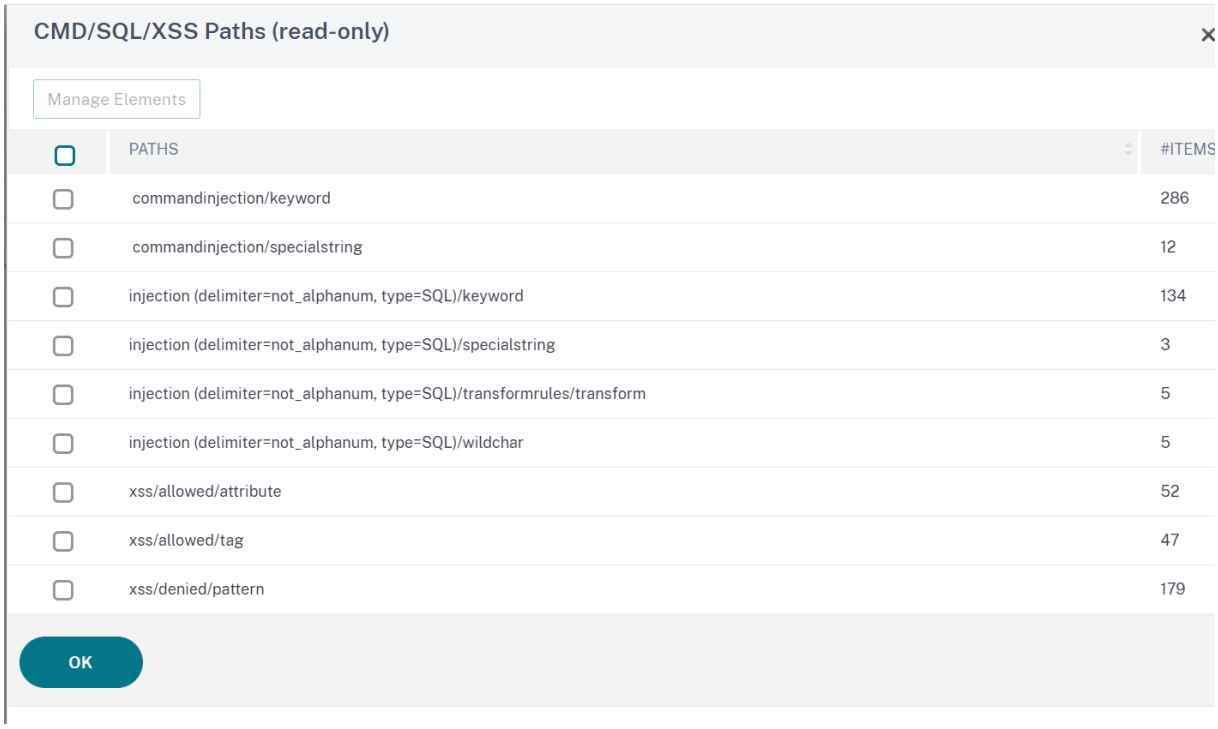
デフォルトの SQL パターンは、デフォルトのシグニチャファイルで指定されます。シグニチャオブジェクトをプロファイルにバインドしない場合、デフォルトシグニチャオブジェクトで指定されたデフォルトのインジェクションパ

ターンが、コマンドインジェクションセキュリティチェック処理のためにプロファイルによって使用されます。デフォルトのシグニチャオブジェクトで指定されている規則とパターンは読み取り専用です。編集や修正はできません。これらのパターンを変更または変更する場合は、デフォルトの SSignatures オブジェクトのコピーを作成して、ユーザー定義署名オブジェクトを作成します。新しいユーザ定義シグニチャオブジェクトのコマンドインジェクションパターンを変更し、これらのカスタマイズされたパターンを使用するトラフィックを処理しているプロファイルでこのシグニチャオブジェクトを使用します。

詳細については、「署名」を参照してください。

GUI を使用してデフォルトの射出パターンを表示するには、次の手順を実行します。

1. [アプリケーションファイアウォール] **>** [署名] に移動し、[***** デフォルトシグニチャ] を選択し、[編集] をクリックします。
2. [**CMD/SQL/XSS** パターンの管理] をクリックします。「**SQL/**クロスサイトスクリプティングパスの管理」テーブルには、CMD/SQL/XS インジェクションに関連するパターンが表示されます。



1. 行を選択し、[要素の管理] をクリックして、Web App Firewall コマンドインジェクションチェックで使用される対応する注入パターン (キーワード、特殊文字列、変換ルール、またはワイルドカード文字) を表示します。

SQL インジェクションチェックでの学習機能の使用

学習アクションが有効になると、Web App Firewall 学習エンジンはトラフィックを監視し、トリガーされた違反を学習します。学習したルールは定期的に検査できます。十分に検討した後、学習したルールを SQL インジェクション緩和ルールとして展開できます。

SQL インジェクション学習の拡張：NetScaler ソフトウェアのリリース 11.0 で、Web App Firewall の学習拡張機能が導入されました。細かい SQL インジェクション緩和をデプロイするために、Web App Firewall はきめ細かい SQL インジェクション学習を提供します。学習エンジンは、観測された値のタイプ（keyword、SpecialString、Wildchar）および入力フィールドで観測された対応する値式に関する推奨事項を作成します。ブロックされたリクエストをチェックして、現在のルールが制限が厳しく、緩和する必要があるかどうかを判断するだけでなく、学習エンジンによって生成されたルールを確認して、違反を引き起こしている値タイプと値式を判断し、で対処する必要があります。リラクゼーションルール。

> 重要
>
> Web App Firewall の学習エンジンでは、名前の最初の 128 バイトしか区別できません。フォームに、最初の 128 バイトに一致する名前のフィールドが複数ある場合、学習エンジンはそれらを区別できないことがあります。同様に、デプロイされた緩和ルールは、そのようなフィールドを SQL インジェクション検査から誤って緩和する可能性があります。
>
> （注）User-Agent ヘッダーの SQL チェックインをバイパスするには、次の緩和ルールを使用します。
>
> ```
> bind appfw profile your_profile_name -SQLInjection User-Agent ".*
> "-location HEADER
> ```

コマンドラインインターフェイスを使用して学習データを表示または使用するには

コマンドプロンプトで、次のコマンドのいずれかを入力します:

- `show appfw learningdata <profilename> SQLInjection`
- `rm appfw learningdata <profilename> -SQLInjection <string> <formActionURL> [<location>] [<valueType> <valueExpression>]`
- `export appfw learningdata <profilename> SQLInjection`

GUI を使用して学習済みデータを表示または使用するには

1. [アプリケーションファイアウォール**] > [**プロファイル] に移動し、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ペインで、[学習済みルール] をクリックします。「学習済みルール」テーブルで **HTML SQL Injection** エントリを選択してダブルクリックすると、学習したルールにアクセスできます。学習したルールを展開したり、緩和ルールとして展開する前にルールを編集したりできます。ルールを破棄するには、ルールを選択して「スキップ」(**Skip**) ボタンをクリックします。一度に編集できるルールは 1 つだけですが、展開またはスキップするルールは複数選択できます。

また、「学習済みルール」(Learned Rules) テーブルで「**HTML SQL Injection** 」エントリを選択し、「ビジュアライザー」(**Visualizer**) をクリックして、学習済みのすべての違反の統合ビューを表示することもできます。ビジュアライザを使用すると、学習したルールを簡単に管理できます。1 つの画面でデータの包括的なビューを表示し、1 回のクリックでルールのグループに対するアクションの実行を容易にします。ビジュアライザーの最大の利点は、正規表現を推奨して複数のルールを統合できることです。デリミタとアクション URL に基づいて、これらのルールの

サブセットを選択できます。ドロップダウンリストから番号を選択すると、ビジュアライザーに 25、50、または 75 個のルールを表示できます。学習したルールのビジュアライザーには、ルールを編集して緩和として展開するオプションがあります。または、ルールをスキップして無視することもできます。

SQL インジェクションチェックでログ機能を使用する

ログアクションを有効にすると、HTML SQL インジェクションのセキュリティチェック違反が **APPFW_SQL** 違反として監査ログに記録されます。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて **/var/log/** フォルダ内の **ns.logs** を末尾にして、SQL インジェクション違反に関連するログメッセージにアクセスします。

```
> Shell

# tail -f /var/log/ns.log | grep APPFW_SQL
```

リクエストが変換されたときの HTML SQL インジェクションログメッセージの例

```
1 Jun 26 21:08:41 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
     .0|APPFW|APPFW_SQL|6|src=10.217.253.62 geolocation=Unknown spt=54001
      method=GET request=http://aaron.stratum8.net/FFC/login.php?
     login_name=%27+or&passwd=and+%3B&drinking_pref=on&text_area=select
     +*+from+%5C+%3B&loginButton=ClickToLogin&as_sfid=AAAAAAXjnGN5gLH-
     hvhTOpIySEIqES7BjFRs5Mq0fwPp-3ZHDi5yWlRWByj0cVbMyy-
     Ens2vaaiULKOcUri4OD4kbXWwSY5s7I3QkDsrvIgCYMC9BMvBwY2wbNcSqCwk52lfE0k
     %3D&as_fid=feeec8758b41740eedeeb6b35b85dfd3d5def30c msg= Special
     characters seen in fields cn1=74 cn2=762 cs1=pr_ffc cs2=PPE1 cs3=9
     ztIlf9p1H7p6Xtzn6NMygTv/QM0002 cs4=ALERT cs5=2015 act=transformed
2 <!--NeedCopy-->
```

ポストリクエストがブロックされた場合の HTML SQL インジェクションログメッセージの例

```
1 Jun 26 21:30:34 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
     .0|APPFW|APPFW_SQL|6|src=10.217.253.62 geolocation=Unknown spt=9459
     method=POST request=http://aaron.stratum8.net/FFC/login_post.php msg
     =SQL Keyword check failed for field text_area="(')" cn1=78 cn2=834
     cs1=pr_ffc cs2=PPE1 cs3=eVJMMPtZ2XgylGrHjkx3rZLfBCI0002 cs4=ALERT
     cs5=2015 act=blocked
2 <!--NeedCopy-->
```

> 注
>
> 10.5.e ビルド (エンハンスメントビルド) と 11.0 以降のビルドでのストリーミング変更の一環として、入力データをブロック単位で処理するようになりました。RegEx パターンマッチングは、連続した文字列マッチングで 4K に制限されるようになりました。この変更により、SQL 違反ログメッセージには、以前のビルドとは異なる情報が含まれる場合があります。入力のキーワードと特殊文字は、多くのバイトで区切ることができます。

> データを処理するときに、入力値全体をバッファリングするのではなく、SQL キーワードと特殊文字列を追跡します。ログメッセージには、フィールド名に加えて、SQL キーワード、SQL 特殊文字、または SQL キーワードと SQL 特殊文字の両方が含まれるようになりました。これらの文字は、構成された設定によって決定されます。次の例に示すように、残りの入力はログメッセージに含まれなくなります。
>
> 例:
>
> 10.5 では、Web App Firewall が SQL 違反を検出すると、次に示すように、入力文字列全体がログメッセージに含まれることがあります。
>
> ```
> SQL Keyword check failed for field text=\"select a name from testbed1;(;)\".*<blocked>
> ```
>
> リクエストサイドストリーミングおよび 11.0 以降のビルドをサポートする 10.5.e の拡張ビルドでは、次に示すように、フィールド名、キーワード、および特殊文字（該当する場合）のみをログメッセージに記録します。
>
> ```
> SQL Keyword check failed for field **text="select(;)"<blocked>
> ```
>
> この変更は、application/x-www-form-urlencoded、マルチパート/フォームデータ、または text/x-gwt-rpc コンテンツタイプを含むリクエストに適用されます。**JSON** または **XML** ペイロードの処理中に生成されるログメッセージは、この変更の影響を受けません。

GUI を使用してログメッセージにアクセスするには

GUI には、ログメッセージを分析するための便利なツール (**Syslog Viewer**) が含まれています。Syslog ビューアにアクセスするには、複数のオプションがあります:

- [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを選択して [セキュリティチェック] をクリックします。**HTML SQL** インジェクション行を強調表示して、[ログ] をクリックします。プロファイルの HTML SQL インジェクションチェックから直接ログにアクセスすると、GUI によってログメッセージが除外され、これらのセキュリティチェック違反に関連するログのみが表示されます。
- 「**NetScaler** 」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。「監査メッセージ」セクションで、「**Syslog messages** 」リンクをクリックすると、Syslog Viewer が表示されます。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティチェック違反がトリガーされる可能性がある場合のデバッグに役立ちます。
- アプリケーションファイアウォール > ポリシー > 監査に移動します。監査メッセージセクションで、**Syslog** メッセージリンクをクリックして Syslog Viewer を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

HTML ベースの Syslog ビューアには、関心のあるログメッセージのみを選択するためのさまざまなフィルタオプションがあります。**HTML SQL** インジェクションチェックのログメッセージを選択するには、「モジュール」のドロップダウン・リスト・オプションで「**APPFW**」を選択してフィルタリングします。**[Event Type]** リストには、選択内容をさらに絞り込むための豊富なオプションが用意されています。たとえば、「**APPFW_SQL** 」チェック・ボックスを選択して「適用」ボタンをクリックすると、**SQL Injection** セキュリティ・チェック違反に関するログ・メッセージのみが Syslog Viewer に表示されます。

特定のログメッセージの行にカーソルを置くと、[モジュール]、[イベントタイプ]、[イベント **ID**]、[クライアント **IP**] などの複数のオプションがログメッセージの下に表示されます。これらのオプションを選択すると、ログメッセージ内の対応する情報を強調表示できます。

[クリックして展開] 機能は GUI でのみ使用できます。Syslog Viewer を使用すると、ログを表示するだけでなく、Web App Firewall セキュリティチェック違反のログメッセージに基づいて HTML SQL インジェクション緩和ルールを展開することもできます。この操作では、ログメッセージは CEF ログ形式である必要があります。クリックして展開機能は、ブロック (またはブロックしない) アクションによって生成されたログメッセージに対してのみ使用できます。変換操作に関するログメッセージの緩和ルールは展開できません。

Syslog Viewer から緩和ルールを展開するには、ログメッセージを選択します。選択した行の [**Syslog Viewer**] ボックスの右上隅にチェックボックスが表示されます。このチェックボックスをオンにし、[アクション] リストから緩和ルールを展開するオプションを選択します。**[**** 編集とデプロイ ****]**、**[**** デプロイ**]**、**[**すべてデプロイ] は、**

[Click to Deploy] オプションを使用してデプロイされる SQL インジェクションルールには、細粒度緩和の推奨事項は含まれません。

GUI の **[**クリックしてデプロイ**]** 機能を使用するには、次の手順を実行します。

1. Syslog Viewer で、[モジュール] オプションの [アプリケーションファイアウォール] を選択します。
2. 対応するログメッセージをフィルタするには、イベントタイプとして **APP_SQL** を選択します。
3. 展開するルールを識別するには、このチェックボックスをオンにします。
4. オプションの [アクション（Action）] ドロップダウンリストを使用して、緩和ルールを展開します。
5. ルールが対応する [緩和ルール] セクションに表示されていることを確認します。

SQL インジェクション違反の統計

統計アクションが有効になっている場合、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、SQL インジェクションチェックのカウンタが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分サイズは、構成された設定によって異なります。たとえば、ブロック・アクションが有効になっている場合、3 つの SQL インジェクション違反を含むページのリクエストでは、最初の違反が検出されるとすぐにページがブロックされるため、stats カウンタが 1 つずつ増加します。ただし、ブロックが無効になっている場合、同じ要求を処理すると、違反ごとに個別のログメッセージが生成されるため、違反とログの統計カウンタが 3 ずつ増加します。

コマンドラインを使用して **SQL** インジェクション・チェック統計を表示するには、次のステップを実行します。

コマンドプロンプトで入力します:

> **sh appfw** 統計

特定のプロファイルの統計情報を表示するには、以下のコマンドを使用します。

```
> stat appfw profile <profile name>
```


GUI を使用して HTML SQL インジェクションの統計情報を表示するには

1. システム > セキュリティ > アプリケーションファイアウォールに移動します。
2. 右側のペインで、[統計 リンク] にアクセスします。
3. スクロールバーを使用して、HTML SQL インジェクション違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

ハイライト

SQL インジェクションチェックについては、次の点に注意してください。

- **SQL** インジェクション保護の組み込みサポート-NetScaler Web App Firewall は、フォームパラメータ内の SQL キーワードと特殊文字の組み合わせを監視することで、SQL インジェクションから保護します。すべての SQL キーワード、特殊文字、ワイルドカード文字、およびデフォルトの変換ルールは、/netscaler/default_custom_settings.xml ファイルに指定されています。
- カスタマイズ: デフォルトのキーワード、特殊文字、ワイルドカード文字、および変換ルールを変更して、アプリケーションの特定のニーズに合わせて SQL セキュリティチェックインスペクションをカスタマイズできます。デフォルトのシグニチャオブジェクトのコピーを作成するか、既存のエントリを変更するか、新しいシグニチャオブジェクトを追加します。このシグニチャオブジェクトをプロファイルにバインドして、カスタマイズした設定を利用します。
- ハイブリッドセキュリティモデル-シグニチャとディープセキュリティ保護の両方で、プロファイルにバインドされたシグニチャオブジェクトで指定された SQL/Cross-Site スクリプティングパターンが使用されます。シグニチャオブジェクトがプロファイルにバインドされていない場合は、デフォルトのシグニチャオブジェクトに存在する SQL/Cross-Site スクリプティングパターンが使用されます。
- [**Transform**]: 変換操作について、次の点に注意してください。
  - 変換操作は、他の SQL インジェクションアクション設定とは独立して動作します。変換が有効で、ブロック、ログ、統計、学習がすべて無効になっている場合、SQL の特殊文字は変換されます。
  - SQL 変換が有効な場合、SQL 特殊文字が非ブロックモードで変換された後、ユーザー要求がバックエンドサーバーに送信されます。ブロックアクションが有効になっている場合は、変換アクションよりも優先されます。インジェクションタイプが SQL 特殊文字として指定され、ブロックが有効な場合、変換アクションにもかかわらずリクエストはブロックされます。
- [きめ細かな緩和と学習]: 緩和ルールを微調整して、SQL 要素のサブセットをセキュリティチェック検査から緩和し、残りは検出します。学習エンジンは、観測されたデータに基づいて、特定の値のタイプと値の式を推奨しています。
- [**Click to Deploy**]: syslog ビューアで 1 つまたは複数の SQL 違反ログメッセージを選択し、緩和ルールとして展開します。

HTML および JSON ペイロードの SQL 文法ベースの保護

March 20, 2024

NetScaler Web App Firewall は、HTTPおよびJSONペイロードでの SQL インジェクション攻撃を検出するためにパターン一致アプローチを使用します。このアプローチでは、事前定義された一連のキーワードと（または）特殊文字のセットを使用して攻撃を検出し、違反としてフラグを立てます。このアプローチは効果的ですが、多くの誤検出が発生し、1 つ以上の緩和ルールが追加される可能性があります。特に、HTTP または JSON リクエストで「Select」や「From」などの一般的に使用される単語が使用されている場合。HTMLおよびJSONペイロードの SQL 文法保護チェックを実装することで、誤検知を減らすことができます。

既存のパターンマッチアプローチでは、HTTP リクエストに事前定義されたキーワードまたは特殊文字が存在する場合、SQL インジェクション攻撃が識別されます。この場合、ステートメントは有効な SQL 文である必要はありません。しかし、文法ベースのアプローチでは、SQL インジェクション攻撃は、キーワードまたは特殊文字が SQL 文に存在するか、SQL 文の一部である場合にのみ検出され、誤検出シナリオが減少します。

SQL 文法ベースの保護使用シナリオ

HTTP リクエストに「チケットを選択してユニオンステーションで会おう」というステートメントを考えてみましょう。ステートメントは有効な SQL ステートメントではありませんが、既存のパターンマッチアプローチは、SQL インジェクション攻撃としてリクエストを検出します。これは、ステートメントが「Select」、「and」、「Union」などのキーワードを使用するためです。ただし、SQL 文法アプローチの場合、キーワードが有効な SQL 文に存在しないか、有効な SQL 文の一部ではないため、ステートメントは違反攻撃として検出されません。

文法ベースのアプローチは、JSONペイロードの SQL インジェクション攻撃を検出するように設定することもできます。緩和ルールを追加するには、既存の緩和ルールを再利用できます。「ValueType」「キーワード」のルールでは、細粒緩和ルールは、SQL 文法にも適用されます。JSON SQL 文法では、既存の URL ベースのメソッドを再利用できます。

CLI を使用して SQL 文法ベースの HTML 保護を設定します

SQL 文法ベースの検出を実装するには、Web App Firewall プロファイルで「sqLinjectionGrammer」パラメーターを構成する必要があります。デフォルトでは、パラメータは無効になっています。学習以外の既存の SQL インジェクションアクションはすべてサポートされています。アップグレード後に作成された新しいプロファイルは、SQL インジェクション文法をサポートしており、引き続きデフォルトのタイプを「特殊文字またはキーワード」とし、明示的に有効にする必要があります。

コマンドプロンプトで入力します：

```
1 add appfw profile <profile-name>  -SQLInjectionAction <action-name> -
     SQLInjectionGrammar ON/OFF
2 <!--NeedCopy-->
```


例:

```
add appfw profile profile1 -SQLInjectionAction Block -SQLInjectionGrammar
 ON
```

CLI を使用して HTML の SQL パターンマッチ保護と文法ベースの保護を設定します

文法ベースとパターンマッチの両方のアプローチを有効にした場合、アプライアンスは最初に文法ベースの検出を実行し、アクションタイプをブロックに設定した SQL インジェクション検出がある場合、リクエストはブロックされます（パターンマッチを使用して検出を確認せずに）。

コマンドプロンプトで入力します:

```
1 add appfw profile <profile-name>  - SQLInjectionAction <action-name> -
     SQLInjectionGrammar ON  - SQLInjectionType <Any action other than  '
     None' : SQLSplCharANDKeyword/ SQLSplCharORKeyword/ SQLSplChar/
     SQLKeyword>
2 <!--NeedCopy-->
```

例:

```
add appfw profile p1 -SQLInjectionAction block - SQLInjectionGrammar
ON -SQLInjectionType SQLSplChar
```

CLI を使用して HTML を文法ベースで保護する SQL インジェクションチェックのみを設定する

コマンドプロンプトで入力します:

```
1 add appfw profile <profile-name>  - SQLInjectionAction <action-name> -
     SQLInjectionGrammar ON  - SQLInjectionType None
2 <!--NeedCopy-->
```

例:

```
add appfw profile p1 -SQLInjectionAction block - SQLInjectionGrammar
ON -SQLInjectionType None
```

CLI を使用して HTML を SQL 文法ベースで保護するための緩和ルールをバインドする

アプリケーションでペイロード内の特定の「ELEMENT」または「ATTRIBUTE」のSQLインジェクションチェックをバイパスする必要がある場合は、緩和ルールを設定する必要があります。

> 注:

> ValueType「キーワード」を含む緩和ルールは、アプライアンスがSQL文法を使用して検出を実行する場合にのみ評価されます。

インジェクション検査緩和ルールSQLコマンドの構文は次のとおりです。コマンドプロンプトで入力します:

```
1 bind appfw profile <name> -SQLInjection <String> [isRegex(REGEX|
     NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor
     |SpecialString|Wildchar) [<valueExpression>][-isValueRegex (REGEX |
     NOTREGEX) ]]
2 <!--NeedCopy-->
```

例:

```
bind appfw profile p1 -sqlinjection abc http://10.10.10.10/
bind appfw profile p1 -sqlinjection 'abc[0-9]+'http://10.10.10.10/ -
isregex regEX
bind appfw profile p1 -sqlinjection 'name'http://10.10.10.10/ -valueType
 Keyword 'selec[a-z]+' -isvalueRegex regEX
```

GUI を使用して SQL 文法ベースの HTML 保護を設定します

GUI 手順を実行して、文法ベースの HTML SQL インジェクション検出を設定します。

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. 「プロファイル」ページで、「追加」をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、[詳細設定] の [セキュリティチェック] をクリックします。
4. [セキュリティチェック] セクションで、[**HTML SQL** インジェクションの設定] に移動します。
5. チェックボックスの近くにある実行可能アイコンをクリックします。
6. [アクションの設定] をクリックして、[**HMTL SQL** インジェクション設定] ページにアクセスします。
7. [**SQL** 文法を使用してチェックする] チェックボックスをオンにします。
8. [**OK**] をクリックします。

CLI を使用して JSON ペイロードの SQL 文法ベースの保護を設定します

JSON ペイロードの SQL 文法ベースの検出を実装するには、Web App Firewall プロファイルで「jsonSQLinjectionGrammer」パラメータを設定する必要があります。デフォルトでは、パラメータは無効になっています。学習以外の既存の SQL インジェクションアクションはすべてサポートされています。アップグレード後に作成された新しいプロファイルは、SQL インジェクション文法をサポートしており、引き続きデフォルトのタイプを「特殊文字またはキーワード」として使用するため、明示的に有効にする必要があります。

コマンドプロンプトで入力します:


```
add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
    action-name> -JSONSQLInjectionGrammar ON/OFF
<!--NeedCopy-->
```

例:

```
add appfw profile profile1 -type JSON -JSONSQLInjectionAction Block -
JSONSQLInjectionGrammar ON
```

CLI を使用して JSON ペイロードの SQL パターンマッチ保護と文法ベースの保護を設定します

文法ベースチェックとパターンマッチチェックの両方を有効にした場合、アプライアンスは最初に文法ベースの検出を実行し、アクションタイプをブロックに設定した SQL インジェクション検出がある場合、リクエストはブロックされます（パターンマッチを使用して検出を確認せずに）。

> 注:
>
> ValueType「キーワード」を持つ緩和ルールは、アプライアンスが SQL 文法を使用して検出を実行する場合にのみ評価されます。

コマンドプロンプトで入力します:

```
add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
    action-name> -JSONSQLInjectionGrammar ON  - JSONSQLInjectionType <Any
     action other than  'None' : SQLSplCharANDKeyword/
    SQLSplCharORKeyword/ SQLSplChar/ SQLKeyword>
<!--NeedCopy-->
```

例:

```
add appfw profile p1 -type JSON -JSONSQLInjectionAction block -
JSONSQLInjectionGrammar ON -JSONSQLInjectionType SQLSplChar
```

CLI を使用して JSON ペイロードの SQL インジェクションチェックのみの文法ベースの保護を設定します

コマンドプロンプトで入力します:

```
add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
    action-name> -JSONSQLInjectionGrammar ON  - JSONSQLInjectionType None
    `\
<!--NeedCopy-->
```

例:

```
add appfw profile p1 -type JSON -JSONSQLInjectionAction block -
JSONSQLInjectionGrammar ON -JSONSQLInjectionType None
```


CLI を使用して JSON ペイロードを SQL 文法ベースで保護するための URL ベースの緩和ルールをバインドする

アプリケーションでペイロード内の特定の「ELEMENT」または「ATTRIBUTE」のJSONコマンドインジェクションインスペクションをバイパスする必要がある場合は、緩和ルールを設定できます。
JSONコマンドインジェクションインスペクション緩和ルールの構文は次のとおりです。コマンドプロンプトで入力します:

```
1 bind appfw profile <profile name>  - JSONCMDURL <expression> -comment <
     string> -isAutoDeployed ( AUTODEPLOYED | NOTAUTODEPLOYED ) -state (
     ENABLED | DISABLED )
2 <!--NeedCopy-->
```

例:

```
bind appfw profile p1 -sqlinjection abc http://10.10.10.10/
bind appfw profile p1 -sqlinjection 'abc[0-9]+'http://10.10.10.10/ -
isregex regEX
bind appfw profile p1 -sqlinjection 'name'http://10.10.10.10/ -valueType
 Keyword 'selec[a-z]+' -isvalueRegex regEX
```

GUI を使用して JSON ペイロードの SQL 文法ベースの保護を設定します

GUI 手順を実行して、文法ベースの JSON SQL インジェクション検出を設定します。

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. 「プロファイル」ページで、「追加」をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、[詳細設定] の [セキュリティチェック] をクリックします。
4. [セキュリティチェック] セクションで、[**JSON SQL** インジェクション設定] に移動します。
5. チェックボックスの近くにある実行可能アイコンをクリックします。
6. [アクション設定] をクリックして、[**JSON SQL** インジェクション設定] ページにアクセスします。
7. [**SQL** 文法を使用してチェックする] チェックボックスをオンにします。
8. [**OK**] をクリックします。

HTML ペイロードのコマンドインジェクション文法ベースの保護

March 20, 2024

NetScaler Web App Firewall は、HTML ペイロードのコマンドインジェクション攻撃を検出するためにパターンマッチアプローチを使用します。このアプローチでは、事前定義されたキーワードと（または）特殊文字のセットを

使用して攻撃を検出し、違反としてフラグを立てます。このアプローチは効果的ですが、多くの誤検出が発生し、1つ以上の緩和ルールが追加される可能性があります。特に、HTTP リクエストで「Exit」などの一般的に使用される単語が使用されている場合。HTML ペイロードにコマンドインジェクションの文法ベースの保護チェックを実装することで、誤検出を減らすことができます。

パターンマッチアプローチでは、事前定義されたキーワードと（または）特殊文字が HTTP 要求に存在する場合、コマンドインジェクション攻撃が識別されます。この場合、ステートメントは有効なコマンドインジェクションステートメントである必要はありません。しかし、文法ベースのアプローチでは、コマンドインジェクション攻撃は、コマンドインジェクションステートメントにキーワードまたは特殊文字が存在する場合にのみ検出されます。したがって、誤検出のシナリオは減少します。

コマンドインジェクション文法ベースの保護の使用シナリオ

「出口に向かって急いで!」という声明を考えてみましょう。HTTP リクエストに存在します。このステートメントは有効なコマンドインジェクションステートメントではありませんが、pattern-match アプローチでは、キーワード「exit」によるコマンドインジェクション攻撃として要求が検出されます。しかし、コマンドインジェクションの文法ベースのアプローチでは、キーワードが有効なコマンドインジェクションステートメントに存在しないため、ステートメントは違反攻撃として検出されません。

CLI を使用してコマンドインジェクションの文法ベースの保護パラメータを構成する

コマンドインジェクションの文法ベースの検出を実装するには、Web App Firewall プロファイルで「cmdInjectionGrammar」パラメーターを構成する必要があります。デフォルトでは、パラメータは無効になっています。学習を除き、既存のコマンドインジェクションアクションはすべてサポートされています。アップグレード後に作成された新しいプロファイルは、コマンドインジェクショングラマーをサポートします。新しいプロファイルのデフォルトのタイプは「特殊文字またはキーワード」のままであり、コマンドインジェクショングラマーは明示的に有効にする必要があります。

コマンドプロンプトで入力します:

```
1 add appfw profile <profile-name>  - CMDInjectionAction <action-name> -
     CMDInjectionGrammar ON/OFF
2 <!--NeedCopy-->
```

例:

```
1 add appfw profile profile1  - CMDInjectionAction Block  -
     CMDInjectionGrammar ON
2 <!--NeedCopy-->
```


CLI を使用してコマンドインジェクションのパターン一致保護と文法ベースの保護を構成する

文法ベースとパターンマッチの両方のアプローチを有効にしている場合、アプライアンスは最初に文法ベースの検出を実行します。アクションタイプが「block」に設定されたコマンドインジェクションが検出された場合、リクエストはブロックされます（pattern-match を使用して検出を検証しません）。

コマンドプロンプトで入力します：

```
1 add appfw profile <profile-name>  -CMDInjectionAction <action-name> -
     CMDInjectionGrammar ON  -CMDInjectionType <Any action other than  '
     None' : CMDSplCharANDKeyword/ CMDSplCharORKeyword/ CMDSplChar/
     CMDKeyword>
2 <!--NeedCopy-->
```

例：

```
1 add appfw profile p1  -CMDInjectionAction block  -  CMDInjectionGrammar
     ON  -CMDInjectionType CMDSplChar
2 <!--NeedCopy-->
```

CLI を使用してコマンドインジェクションチェックを文法ベースの保護のみで構成する

コマンドプロンプトで入力します：

```
1 add appfw profile <profile-name>  -CMDInjectionAction <action-name> -
     CMDInjectionGrammar ON  -CMDInjectionType None
2 <!--NeedCopy-->
```

例：

```
1 add appfw profile p1  -CMDInjectionAction block  -  CMDInjectionGrammar
     ON  -CMDInjectionType None
2 <!--NeedCopy-->
```

CLI を使用したコマンドインジェクションの文法ベースの保護のバインド緩和ルール

アプリケーションで、HTML ペイロード内の特定の「ELEMENT」または「ATTRIBUTE」のコマンドインジェクションチェックをバイパスする必要がある場合は、緩和ルールを設定する必要があります。

> 注：
>
> valueType が「keyword」である緩和ルールは、アプライアンスがコマンドインジェクション文法を使用して検出を実行する場合にのみ評価されます。

コマンドインジェクションインスペクション緩和ルールには、次の構文があります。コマンドプロンプトで入力します：


```
bind appfw profile <name> -CMDInjection <String> [isRegex(REGEX|
    NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor
    |SpecialString|Wildchar) [<valueExpression>][-isValueRegex (REGEX |
    NOTREGEX) ]]
<!--NeedCopy-->
```

例:

```
bind appfw profile p1 -cmdinjection abc http://10.10.10.10/

bind appfw profile p1  - cmdinjection 'abc[0-9]+' http://10.10.10.10/ -
    isregex regEX

bind appfw profile p1  - cmdinjection 'name' http://10.10.10.10/ -
    valueType Keyword 'exi[a-z]+' -isvalueRegex regEX
<!--NeedCopy-->
```

GUI を使用してコマンドインジェクションの文法ベースの保護を構成する

次の手順を実行して、文法ベースの HTML コマンドインジェクション検出を設定します。

1. セキュリティ **> NetScaler Web App Firewall** プロファイル **>** プロファイルの順に移動します。
2. プロファイルを選択し、[編集] をクリックします。
3. [詳細設定] セクションに移動し、[セキュリティチェック] をクリックします。
4. [**HTML** コマンドインジェクション] チェックボックスをオンにして、[アクション設定] をクリックします。
5. [**CMD** 文法を使用してチェック] チェックボックスをオンにします。
6. 「要求を含むチェック」から「なし」を選択します。
7. 「**OK**」をクリックします。

HTML SQL インジェクション攻撃を処理するための緩和ルールと拒否ルール

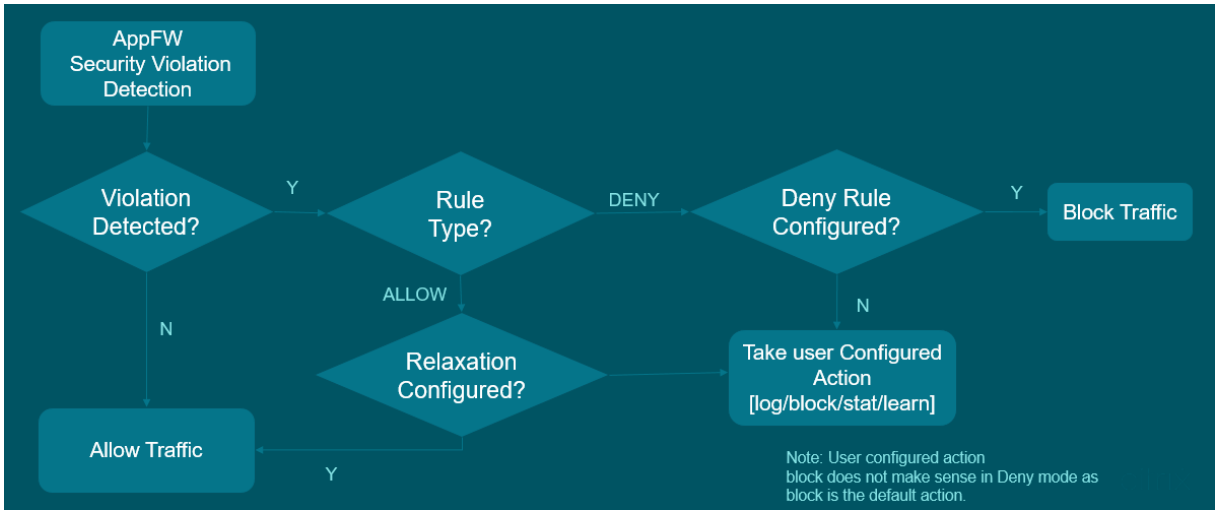
October 25, 2023

着信トラフィックがある場合、違反検出ロジックはトラフィック違反をチェックします。HTML SQL インジェクション攻撃が検出されない場合、トラフィックは通過できます。ただし、違反が検出された場合、緩和ルール（許可）と拒否ルールによって、違反の処理方法が定義されます。セキュリティチェックが許可モード（デフォルトモード）で設定されている場合、ユーザが緩和ルールまたは許可ルールを明示的に設定していない限り、検出された違反はブロックされます。

許可モードに加えて、セキュリティチェックを拒否モードで構成し、違反の処理に拒否ルールを使用することもできます。このモードでセキュリティチェックが設定されている場合、ユーザーが明示的に拒否ルールを設定している場合、検出された違反はブロックされます。拒否ルールが構成されていない場合は、ユーザー設定アクションが適用されます。

> 注:
>
> デフォルトでは、URL は正規表現です。

次の図は、動作モードの動作を許可および拒否する方法を説明しています。



1. 違反が検出されると、緩和ルール（許可）および拒否ルールによって、違反の処理方法が定義されます。
2. セキュリティチェックが拒否モードに設定されている場合（許可モードに設定されている場合は、ステップ 5 に進みます）、拒否ルールを明示的に設定していない限り、違反はブロックされます。
3. 違反が拒否ルールと一致する場合、アプライアンスはトラフィックをブロックします。
4. トラフィック違反がルールと一致しない場合、アプライアンスはユーザー定義アクション（ブロック、リセット、またはドロップ）を適用します。
5. セキュリティチェックが許可モードで構成されている場合、Web App Firewall モジュールは、許可ルールが構成されているかどうかを確認します。
6. 違反が許可ルールと一致する場合、アプライアンスはトラフィックをバイパスすることを許可し、それ以外の場合はブロックされます。

CLI を使用してセキュリティチェックイン、緩和、強制モードを設定する

コマンドプロンプトで入力します。

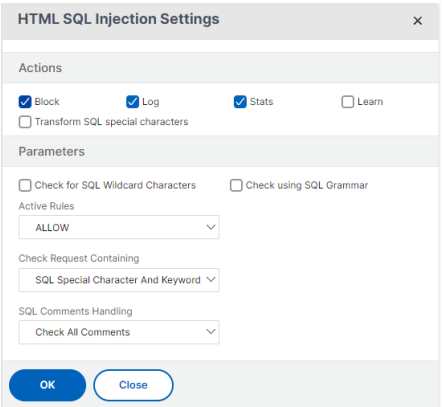
```
set appfw profile <name>   - SQLInjectionAction [block stats learn]  -
    SQLInjectionRuleType [ALLOW DENY]
<!--NeedCopy-->
```

例:


```
set appfw profile prof1 sqlInjectionAction block -sqlInjectionRuleType
 ALLOW DENY
```

GUI を使用してセキュリティチェックイン、緩和、強制モードを設定する

1. **[**** セキュリティ**] > [NetScaler Web App Firewall** とプロファイル] に移動します。******
2. [プロファイル**]** ページでプロファイルを選択し、[編集] をクリックします。
3. [**NetScaler Web App Firewall** プロファイル] ページで、[詳細設定] セクションに移動し、[セキュリティチェック] をクリックします。
4. 「セキュリティチェック」セクションで、「**HTML SQL** インジェクション設定」を選択し、「アクション設定」をクリックします。
5. **HTML** コマンドインジェクション設定ページで、HTML コマンドインジェクションのセキュリティチェックの一部として実行するアクションを選択し、パラメータを更新します。
6. [**OK**] をクリックします。



CLI を使用して緩和ルールと適用ルールを **Web** アプリケーションファイアウォールプロファイルにバインドする

コマンドプロンプトで入力します。

```
1 bind appfw profile <name> -SQLInjection <string> <formActionURL>
2 <!--NeedCopy-->
```

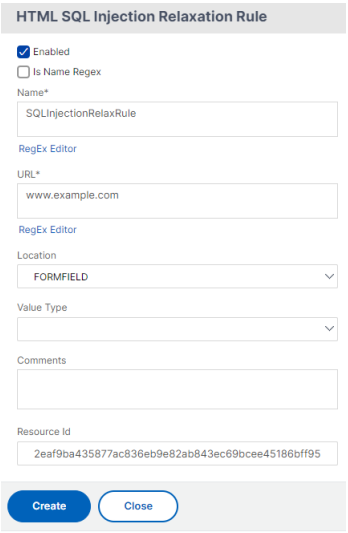
例**:**

```
bind appfw profile p1 -SQLInjection field_f1 "/login.php"-RuleType
ALLOW
```

```
bind appfw profile p2 -SQLInjection field_f1 "/login.php"-RuleType
ALLOW
```


GUI を使用して **Web** アプリケーションファイアウォールプロファイルに緩和ルールと適用ルールをバインドします

1. **[**** セキュリティ**] > [NetScaler Web App Firewall** とプロファイル] に移動します。******
2. [プロファイル**]** ページでプロファイルを選択し、[編集] をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、「詳細設定」セクションに移動し、「緩和ルール」をクリックします。
4. 「緩和ルール」セクションで、「**HTML SQL** インジェクション設定」を選択し、「編集」をクリックします。
5. 「**HTML SQL** インジェクション緩和ルール」ページで、「追加」をクリックします。
6. 必要な詳細を指定してください。
7. [作成] をクリックします。



HTML コマンドインジェクション保護チェック

March 20, 2024

HTML コマンドインジェクションチェックは、着信トラフィックにシステムセキュリティを壊したり、システムを変更したりする不正なコマンドがあるかどうかを調べます。検出されたトラフィックに悪意のあるコマンドが含まれている場合、アプライアンスは要求をブロックするか、設定されたアクションを実行します。

NetScaler Web App Firewall プロファイルが強化され、コマンドインジェクション攻撃に対する新しいセキュリティチェックが追加されました。コマンドインジェクションセキュリティチェックでトラフィックが検査され、悪意のあるコマンドが検出されると、アプライアンスは要求をブロックするか、設定されたアクションを実行します。

コマンドインジェクション攻撃では、攻撃者は NetScaler オペレーティングシステム上で不正なコマンドを実行することを目的としています。これを実現するために、攻撃者は脆弱なアプリケーションを使用してオペレーティング

システムコマンドを注入します。NetScaler アプライアンスは、アプリケーションが安全でないデータ（フォーム、Cookie、またはヘッダー）をシステムシェルに渡すと、インジェクション攻撃に対して脆弱になります。

コマンドインジェクション保護のしくみ

1. 受信リクエストの場合、WAF はトラフィックにキーワードまたは特殊文字がないか調べます。受信リクエストに、拒否されたキーワードや特殊文字のいずれにも一致するパターンがない場合、リクエストは許可されます。それ以外の場合、要求は設定されたアクションに基づいてブロック、ドロップ、またはリダイレクトされます。
2. キーワードや特殊文字をリストから除外したい場合は、緩和ルールを適用して特定の条件下でセキュリティチェックをバイパスできます。
3. ロギングを有効にすると、ログメッセージを生成できます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとしたことを示している可能性があります。
4. また、統計機能を有効にして、違反やログに関する統計データを収集することもできます。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当な要求がブロックされた場合は、新しい緩和ルールを構成する必要があるか、既存の緩和ルールを変更する必要があるかを再確認するために、構成を再確認する必要があります。

コマンドインジェクションチェックで拒否されたキーワードと特殊文字

コマンドインジェクション攻撃を検出してブロックするために、アプライアンスにはデフォルトのシグネチャファイルに一連のパターン（キーワードと特殊文字）が定義されています。コマンドインジェクションの検出中にブロックされるキーワードの一覧を次に示します。

```
    <commandinjection>
      <keyword type="LITERAL" builtin="ON">7z</keyword>
      <keyword type="LITERAL" builtin="ON">7za</keyword>
      <keyword type="LITERAL" builtin="ON">7zr</keyword>
…
</commandinjection>
<!--NeedCopy-->
```

シグネチャファイルに定義されている特殊文字は次のとおりです。

```
| ; & $ > < '\ ! >> #
```

CLI によるコマンドインジェクションチェックの設定

コマンドラインインターフェイスでは、set the profile コマンドまたは add the profile コマンドのいずれかを使用してコマンドインジェクション設定を構成できます。ブロック、ログ、統計の各アクションを有効にできます。また、ペイロードで検出したいキーワードと文字列を設定する必要があります。

コマンドプロンプトで入力します:

```
set appfw profile <profile-name> -cmdInjectionAction <action-name> -
CMDInjectionType <CMDInjectionType>]
```

> 注:
>
> デフォルトでは、コマンドインジェクションアクションは「なし」に設定されています。また、デフォルトのコマンドインジェクションタイプはCmdSplCharANDKeyWordとして設定されています。

例:

```
set appfw profile profile1 -cmdInjectionAction block -CMDInjectionType
 CmdSplChar
```

ここで、使用可能なコマンドインジェクションアクションは次のとおりです。

- None-コマンドインジェクション保護を無効にします。
- Log: セキュリティー検査のコマンドインジェクション違反をログに記録します。
- Block-コマンドインジェクションセキュリティー検査に違反するトラフィックをブロックします。
- Stats-コマンドインジェクションのセキュリティ違反に関する統計を生成します。

ここで、使用可能なコマンドインジェクションタイプは次のとおりです。

- SplChar コマンドです。特殊文字をチェックします
- コマンドキーワード。コマンドインジェクションキーワードをチェック
- CmdSplCharANDKeyWord. 特殊文字とコマンドインジェクションをチェックします。キーワードとブロックは、両方が存在する場合に限ります。
- CmdSplCharORKeyWord. 特殊文字とコマンドインジェクションのキーワードとブロックのいずれかが見つかった場合はチェックします。

コマンドインジェクション保護チェックの緩和ルールの設定

アプリケーションでペイロード内の特定の ELEMENT または ATTRIBUTE のコマンドインジェクションインスペクションをバイパスする必要がある場合は、緩和ルールを設定できます。

コマンドインジェクションインスペクション緩和ルールの構文は次のとおりです。

```
bind appfw profile <profile name> -cmdInjection <string> <URL> -
isregex <REGEX/NOTREGEX>
```

ヘッダーの正規表現の緩和ルールの例

```
bind appfw profile sample -CMDInjection hdr "http://10.10.10.10/"-
location heaDER -valueType Keyword '[a-z]+grep'-isvalueRegex REGEX
```


その結果、インジェクションはコマンドインジェクションチェックを免除し、「grep」のバリアントを含むヘッダーhdrを許可します。

クッキー内の正規表現として **valueType** を持つリラクゼーションルールの例

```
bind appfw profile sample -CMDInjection ck_login "http://10.10.10.10/
"-location cookie -valueType Keyword 'pkg[a-z]+'-isvalueRegex REGEX
```

NetScaler GUI を使用したコマンドインジェクションチェックの設定

コマンドインジェクションチェックを設定するには、次の手順を実行します。

1. [セキュリティ] **> [NetScaler Web App Firewall** とプロファイル] に移動します。
2. [プロファイル] ページでプロファイルを選択し、[編集] をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、「詳細設定」セクションに移動し、「セキュリティチェック」をクリックします。
4. 「セキュリティチェック」セクションで、「**HTML** コマンドインジェクション」を選択し、「アクション設定」をクリックします。
5. **HTML** コマンドインジェクション設定ページで、次のパラメータを設定します。
   a) アクション。コマンドインジェクションセキュリティチェックのために実行するアクションを 1 つ以上選択します。
   b) リクエストに含まれるものをチェックしてください。コマンドインジェクションパターンを選択して、受信リクエストにパターンがあるかどうかを確認します。
6. [**OK**] をクリックします。

GUI を使用してコマンドインジェクションパターンを表示またはカスタマイズする

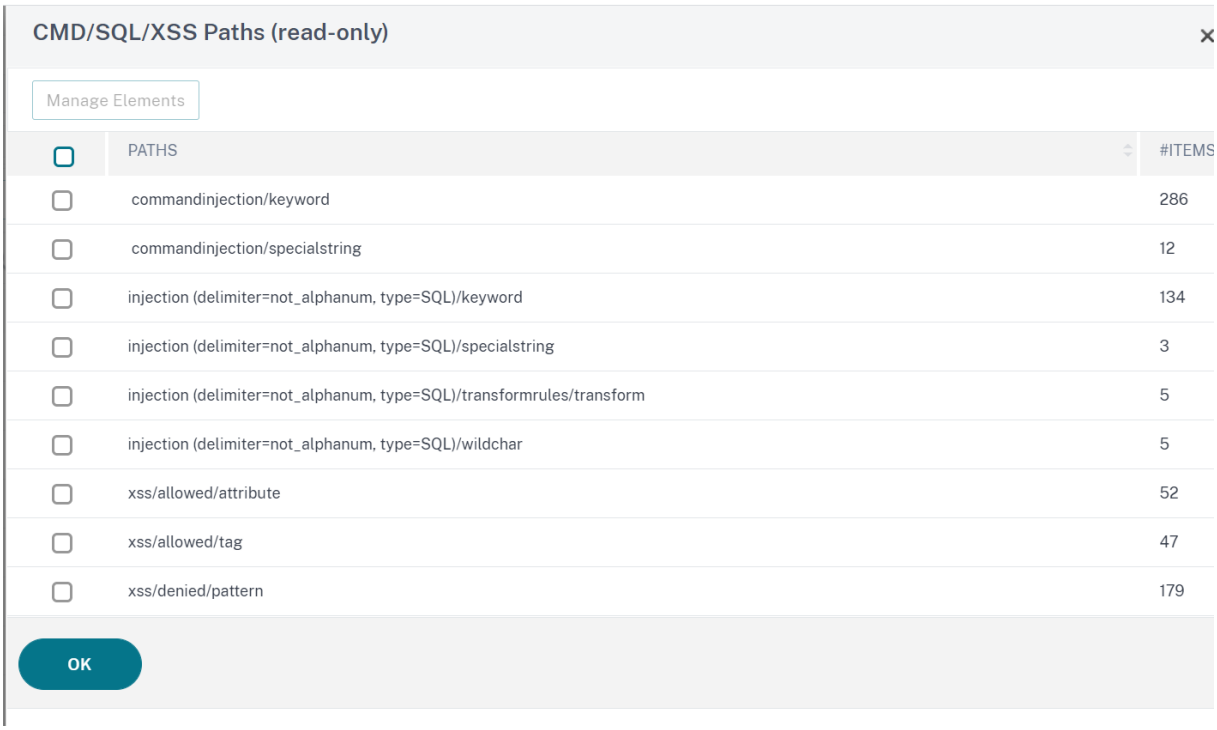
GUI を使用して、**HTML** コマンドインジェクションパターンを表示またはカスタマイズできます。

デフォルトのコマンド注入パターンは、デフォルトシグニチャファイルで指定されます。シグニチャオブジェクトをプロファイルにバインドしない場合、デフォルトシグニチャオブジェクトで指定されたデフォルトの HTML コマンド注入パターンが、コマンドインジェクションセキュリティチェック処理のためにプロファイルによって使用されます。デフォルトのシグニチャオブジェクトで指定されている規則とパターンは読み取り専用です。編集や修正はできません。これらのパターンを変更または変更する場合は、デフォルトの SSignatures オブジェクトのコピーを作成して、ユーザー定義署名オブジェクトを作成します。新しいユーザ定義シグニチャオブジェクトのコマンドインジェクションパターンを変更し、これらのカスタマイズされたパターンを使用するトラフィックを処理しているプロファイルでこのシグニチャオブジェクトを使用します。

詳細については、「署名」を参照してください。

GUI を使用してデフォルトのコマンドインジェクションパターンを表示するには、次の手順を実行します。

1. [アプリケーションファイアウォール**] > [**署名] に移動し、[***** デフォルトシグニチャ**]** を選択し、[編集] をクリックします。
2. [**CMD/SQL/XSS** パターンの管理] をクリックします。**CMD/SQL/XSS** パス **(**読み取り専用**)** テーブルには、CMD/SQL/XSSインジェクションに関連するパターンが表示されます。



1. 行を選択し、[要素の管理] をクリックして、Web App Firewall コマンドインジェクションチェックで使用される対応するコマンド注入パターン (キーワード、特殊文字列、変換ルール、またはワイルドカード文字) を表示します。

GUI を使用してコマンドインジェクションパターンをカスタマイズするには

ユーザー定義の署名オブジェクトを編集して、**CMD** キーワード、特殊文字列、およびワイルドカード文字をカスタマイズできます。新しいエントリを追加したり、既存のエントリを削除したりできます。コマンドインジェクション特殊文字列の変換ルールを変更できます。

1. [アプリケーションファイアウォール**] > [**署名**]** に移動し、ターゲットの [ユーザ定義署名] を選択し、[追加] をクリックします。[**CMD/SQL/XSS** パターンの管理] をクリックします。
2. [**CMD/SQL/XSS** パスの管理] ページで、ターゲットの CMD インジェクション行を選択します。
3. [要素を管理]、[コマンドインジェクション要素を追加]、または [削除] をクリックします。

> 警告:
>
> デフォルトのコマンドインジェクション要素を削除または変更する前に、注意が必要です。または、CMD パスを削除して行全体を削除する必要があります。シグニチャルールとコマンドインジェクションセキュリティチェックは、これらの要素に基づいてコマンドインジェクション攻撃を検出し、アプリケーションを保護します。SQL パターンをカスタマイズすると、編集中に必要なパターンが削除されると、アプリケーションがコマンドインジェクション攻撃に対して脆弱になる可能性があります。

コマンドインジェクショントラフィックおよび違反統計情報の表示

NetScaler Web App Firewall Statistics] ページには、セキュリティトラフィックとセキュリティ違反の詳細が表形式またはグラフ形式で表示されます。

コマンドインターフェイスを使用してセキュリティ統計情報を表示するには。

コマンドプロンプトで入力します:

```
stat appfw profile profile1
```

Appfw プロファイルのトラフィック統計	レート (/s)	合計
リクエスト	0	0
要求バイト数	0	0
レスポンス	0	0
送信バイト数	0	0
中止する	0	0
リダイレクト	0	0
長期平均応答時間 (ミリ秒)	–	0
最近の平均応答時間 (ミリ秒)	–	0

HTML/XML/JSON 違反の統計情報	レート (/s)	合計
開始 URL	0	0
URL を拒否する	0	0
リファラーヘッダー	0	0
バッファオーバーフロー	0	0
Cookie の整合性	0	0

HTML/XML/JSON 違反の統計情報	レート (/s)	合計
Cookie ハイジャック	0	0
CSRF フォームタグ	0	0
HTML クロスサイトスクリプティング	0	0
HTML SQL インジェクション	0	0
フィールド形式	0	0
フィールドの一貫性	0	0
クレジットカード	0	0
セーフオブジェクト	0	0
シグネチャ違反	0	0
コンテンツの種類	0	0
JSON サービス拒否	0	0
JSON SQL インジェクション	0	0
JSON クロスサイトスクリプティング	0	0
ファイルアップロードの種類	0	0
コンテンツタイプ XML ペイロードを推測	0	0
HTML CMD インジェクション	0	0
XML 形式	0	0
XML サービス拒否 (XDoS)	0	0
XML メッセージ検証	0	0
Web サービスの相互運用性	0	0
XML SQL インジェクション	0	0
XML クロスサイトスクリプティング	0	0
XML 添付ファイル	0	0
SOAP フォールト違反	0	0
XML ジェネリック違反	0	0
違反総数	0	0

HTML/XML/JSON ログ統計	レート (/s)	合計
URL ログの開始	0	0
URL ログの拒否	0	0
リファラーヘッダーログ	0	0
バッファオーバーフローログ	0	0
Cookie 整合性ログ	0	0
Cookie ハイジャックのログ	0	0
タグログからの CSRF	0	0
HTML クロスサイトスクリプティングログ	0	0
HTML クロスサイトスクリプティング変換ログ	0	0
HTML SQL インジェクションログ	0	0
HTML SQL 変換ログ	0	0
フィールド形式ログ	0	0
フィールド整合性ログ	0	0
クレジットカード	0	0
クレジットカード変換ログ	0	0
セーフオブジェクトログ	0	0
シグネチャログ	0	0
コンテンツタイプログ	0	0
JSON サービス拒否ログ	0	0
JSON SQL インジェクションログ	0	0
JSON クロスサイトスクリプティングログ	0	0
ファイルアップロードタイプログ	0	0
コンテンツタイプ XML ペイロードを推測 L	0	0
HTML コマンドインジェクションログ	0	0
XML 形式ログ	0	0
XML サービス拒否 (XDoS) ログ	0	0

HTML/XML/JSON ログ統計	レート (/s)	合計
XML メッセージ検証ログ	0	0
WSI ログ	0	0
XML SQL インジェクションログ	0	0
XML クロスサイトスクリプティングログ	0	0
XML 添付ファイルログ	0	0
SOAP フォールトログ	0	0
XML 汎用ログ	0	0
ログメッセージの総数	0	0

** サーバエラー応答統計レート (/s) > 合計 ** |
|–|-|-|
HTTP クライアントエラー (4xx Resp) | 0 | 0|
HTTP サーバエラー (5xx Resp) | 0 |

NetScaler GUI を使用した **HTML** コマンドインジェクション統計の表示

コマンドインジェクションの統計情報を表示するには、次の手順を実行します:

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ペインで Web App Firewall プロファイルを選択し、[統計] をクリックします。
3. **NetScaler Web App Firewall** 統計ページには、HTML コマンドインジェクショントラフィックと違反の詳細が表示されます。
4. 表形式ビューを選択するか、グラフィカルビューに切り替えて、データを表形式またはグラフィカル形式で表示できます。

HTML ペイロードのカスタムキーワードサポート

March 20, 2024

NetScaler リリース 13.1 ビルド 27.xx 以降、選択したキーワードを追加して、これらの構成済みキーワードが HTML ペイロードに存在するかどうかを確認できます。

SQL インジェクションとコマンドインジェクションには、受信リクエストで検索するキーワードまたはパターンの定義済みセットがあります。これらの事前定義されたキーワードセットは、要件に応じてすべてのキーワードをカバー

するとは限らず、誤検出の数が増加する可能性があります。この機能を使用すると、SQL インジェクションおよびコマンドインジェクションチェックでカバーされていないキーワードを追加して、誤検出を減らすことができます。

キーワードを追加した後、追加されたキーワードが着信要求で検出されるかどうかを確認するように NetScaler アプライアンスを構成できます。その後、次のいずれかのアクションを実行するように NetScaler アプライアンスを構成できます。

- **None** —アクションは実行されません。このアクションはデフォルトです。
- ログ -URL に一致し、設定されたキーワードを持つすべてのリクエストをログに記録します。
- ブロック—URL に一致し、キーワードが設定されている要求をすべてブロックします。
- **Stats** —URL に一致し、キーワードが設定されている各リクエストのログカウンターをインクリメントします。

CLI を使用してカスタムキーワードを追加する

CLI を使用してカスタムキーワードを追加する手順は次のとおりです。

1. Web アプリケーションファイアウォールプロファイルを設定し、カスタムキーワードが着信要求で検出されたときのアクションを定義します。

```
set appfw profile <profile-name>  - blockKeywordAction (block | log
    | stats | none)
<!--NeedCopy-->
```

デフォルトでは、-blockKeywordAction は none に設定されています。

例:

```
set appfw profile test_profile -blockKeywordAction none
<!--NeedCopy-->
```

2. Web アプリケーションファイアウォールプロファイルをカスタムキーワードにバインドします。

```
bind appfw profile <profile_name> -blockKeyword <keyword_name> -
    BlockKeywordType <literal|PCRE > -fieldName <field_name> -
    formURL <URL> -isFieldNameRegex <REGEX|NOTREGEX> -state <enable
    /disable> -comment <text>
<!--NeedCopy-->
```

例:

blockword をカスタムキーワードとして追加し、それを **test_profile** にバインドするには、以下のコマンドを実行します。

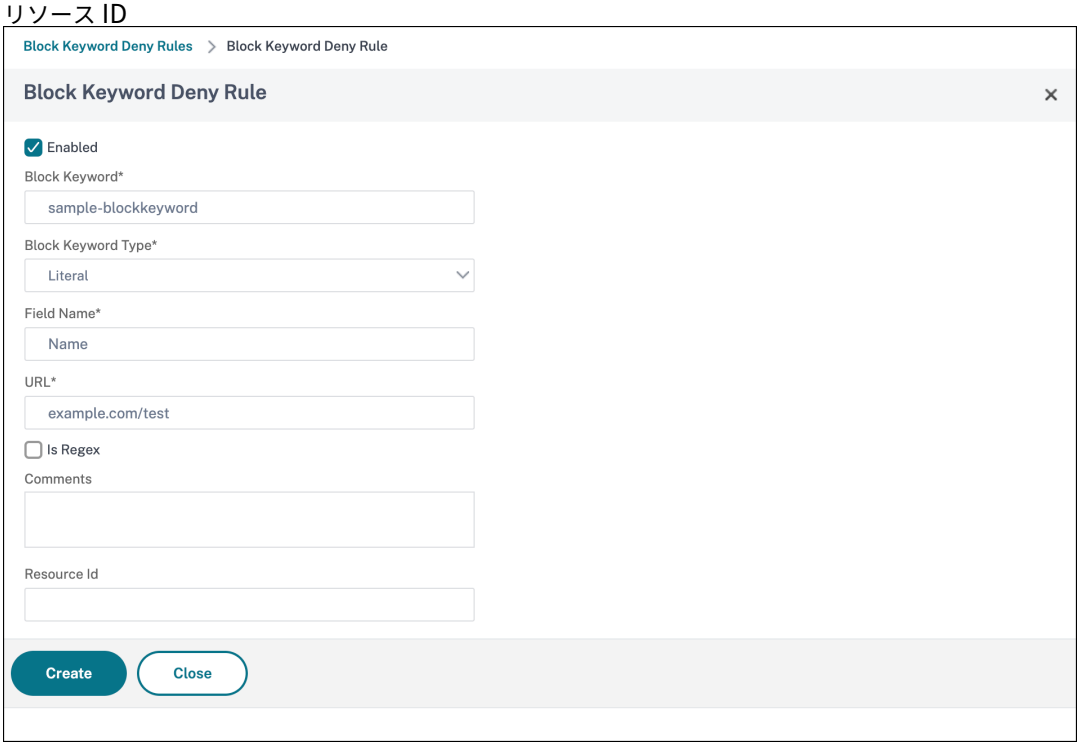
```
bind appfw profile test_profile -blockKeyword "blockword"
    BlockKeywordType literal -fieldName "firstname" -formURL "/
    signup.php" -state enable
<!--NeedCopy-->
```


> 注:
>
> URL パラメータには URL または FQDN を入力できます。FQDN は HTTP プロトコルと HTTPS プロトコルの両方をサポートしています。

GUI を使用してカスタムキーワードを追加する

1. セキュリティ **> NetScaler Web App Firewall** プロファイル **>** プロファイルの順に移動します。
2. プロファイルを選択し、[編集] をクリックします。
3. [詳細設定] セクションに移動し、[ルールを拒否] をクリックします。
4. [ブロックキーワード] を選択し、[編集] をクリックします。
5. [追加] をクリックし、次のパラメータを設定します:

   - 有効化
   - ブロックキーワード
   - ブロックキーワードタイプ
   - [フィールド名]
   - URL
   - 正規表現ですか
   - コメント
   - リソース ID



6. [**Create**] をクリックします。追加したカスタムキーワードは、[ブロックキーワード拒否ルール] ページに表示されます。

7. [詳細設定] セクションに移動し、[セキュリティチェック] をクリックします。
8. 「ブロックキーワード」を選択し、「アクション設定」をクリックします。
9. 必要なアクションを選択し、「**OK**」をクリックします。

CLI を使用してカスタムキーワードの統計情報を表示する

カスタムキーワードの統計を表示するには、コマンドプロンプトで次のコマンドを入力します。

```
stat appfw profile <profile name>
<!--NeedCopy-->
```

例

```
stat appfw profile test_profile
<!--NeedCopy-->
```

GUI でカスタムキーワードの統計情報を表示する

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ペインで **Web App Firewall** プロファイルを選択し、「統計」をクリックします。**NetScaler Web App Firewall** の統計ページには、カスタムキーワードのトラフィックと違反の詳細が表示されます。
3. 表形式ビューを選択するか、グラフィカルビューに切り替えて、データを表形式またはグラフィカル形式で表示できます。

XML 外部エンティティ (XXE) 攻撃からの保護

March 20, 2024

XML 外部エンティティ (XXE) 攻撃保護は、受信したペイロードに、Web アプリケーションが存在する信頼できるドメイン外のエンティティに関する不正な XML 入力がないかを調べます。XXE 攻撃は、外部エンティティへの参照を含む入力を含む XML ペイロードを解析する脆弱な XML パーサーを使用している場合に発生します。

NetScaler アプライアンスでは、XML パーサーが正しく構成されていない場合、この脆弱性を悪用した影響は危険です。これにより、攻撃者は Web サーバー上の機密データを読み取ることができます。サービス拒否攻撃などを実行します。したがって、XXE 攻撃からアプライアンスを保護することが重要です。Web アプリケーションファイアウォールは、コンテンツタイプが XML として識別される限り、XXE 攻撃からアプライアンスを保護できます。悪意のあるユーザーがこの保護メカニズムをバイパスするのを防ぐため、HTTP ヘッダーの「推論」コンテンツタイプが本文のコンテンツタイプと一致しない場合、WAF は受信要求をブロックします。このメカニズムにより、ホワイトリストに登録されたデフォルトまたはデフォルト以外のコンテンツタイプが使用されるときに XXE 攻撃保護バイパスが回避されます。

NetScaler アプライアンスに影響を与える可能性のある XXE 脅威には、次のようなものがあります。

- 機密データ漏えい
- サービス拒否 (DOS) 攻撃
- サーバー側の偽造要求
- ポートスキャン

XML 外部エンティティ (XXE) インジェクション保護を構成する

コマンドインターフェイスを使用して XML 外部エンティティ (XXE) チェックを構成するには:
コマンドラインインターフェイスで、Application fireall profile コマンドを追加または変更して **XXE** 設定を構成できます。ブロック、ログ、統計の各アクションを有効にできます。

コマンドプロンプトで入力します:

```
set appfw profile <name> [-inferContentTypeXmlPayloadAction <inferContentTypeX
 <block | log | stats | none>]
```

> 注:
>
> デフォルトでは、XXE アクションは「none」に設定されています。

例:

```
set appfw profile profile1 -inferContentTypeXmlPayloadAction Block
```

ここで、アクションタイプは次のとおりです。

ブロック: リクエスト内の URL に例外なくリクエストがブロックされます。

ログ:HTTP リクエストヘッダーとペイロードのコンテンツタイプが一致しない場合は、違反しているリクエストに関する情報をログメッセージに含める必要があります。

統計: コンテンツタイプの不一致が検出されると、その違反タイプに対応する統計がインクリメントされます。

なし: コンテンツタイプの不一致が検出されても、アクションは実行されません。None を他のアクションタイプと組み合わせることはできません。デフォルトアクションは None に設定されています。

NetScaler GUI を使用して XXE インジェクションチェックを構成する

XXE インジェクションチェックを設定するには、次の手順を実行します。

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. [プロファイル] ページでプロファイルを選択し、[編集] をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、「詳細設定」セクションに移動し、「セキュリティチェック」をクリックします。

4.「セキュリティチェック」セクションで、「コンテンツタイプの **XML** ペイロードを推定」を選択し、「アクション設定」をクリックします。

5.「推論コンテンツタイプ XML ペイロード設定」ページで、次のパラメータを設定します。

   a) アクション。XXE インジェクションのセキュリティチェックで実行するアクションを 1 つ以上選択します。

6.［**OK**］をクリックします。

XXE インジェクショントラフィックと違反統計の表示

NetScaler Web App Firewall Statistics] ページには、セキュリティトラフィックとセキュリティ違反の詳細が表形式またはグラフ形式で表示されます。

コマンドインターフェイスを使用してセキュリティ統計情報を表示するには。

コマンドプロンプトで入力します：

```
stat appfw profile profile1
```

NetScaler GUI を使用して **XXE** インジェクションの統計情報を表示する

XXE インジェクションの統計情報を表示するには、次の手順を実行してください：

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ペインで Web App Firewall プロファイルを選択し、[統計] をクリックします。
3. **NetScaler Web App Firewall** 統計ページには、XXE コマンドインジェクショントラフィックと違反の詳細が表示されます。
4. 表形式ビューを選択するか、グラフィカルビューに切り替えて、データを表形式またはグラフィカル形式で表示できます。

バッファオーバーフローチェック

August 15, 2023

バッファオーバーフローチェックは、Web サーバ上でバッファオーバーフローを引き起こす試みを検出します。Web App Firewall は、URL、Cookie、またはヘッダーが設定された長さよりも長いことを検出すると、バッファオーバーフローを引き起こす可能性があるためリクエストをブロックします。

バッファオーバーフローチェックは、処理できるよりも大きいデータ文字列を受信すると、クラッシュまたは予期せぬ動作をする、安全でないオペレーティングシステムまたは Web サーバソフトウェアに対する攻撃を防止します。

適切なプログラミング手法により、入力データをチェックしたり、長すぎる文字列を拒否または切り捨てたりすることで、バッファオーバーフローを防ぐことができます。ただし、多くのプログラムは受信データをすべてチェックしないため、バッファオーバーフローの影響を受けやすくなります。この問題は特に、古いバージョンの Web サーバーソフトウェアとオペレーティングシステムに影響しますが、その多くは引き続き使用されています。

バッファオーバーフローセキュリティチェックでは、ブロック、ログ、統計の各アクションを設定できます。さらに、次のパラメータを設定することもできます。

- **URL** の最大長。Web App Firewall がリクエストされた URL で許可する最大長。より長い URL を持つリクエストはブロックされます。指定できる値は 0 ～65535 です。デフォルト:1024
- クッキーの最大長。Web App Firewall がリクエストに含まれるすべての Cookie を許可する最大長です。より長いクッキーを使用したリクエストは、違反をトリガーします。指定できる値は 0 ～65535 です。デフォルト:4096
- ヘッダーの最大長。Web App Firewall が HTTP ヘッダーに使用できる最大長。長いヘッダーを持つリクエストはブロックされます。指定できる値は 0 ～65535 です。デフォルト:4096
- クエリ文字列の長さ。受信リクエストで許容されるクエリ文字列の最大長。クエリが長くなるリクエストはブロックされます。指定できる値は 0 ～65535 です。デフォルト:1024
- リクエストの合計長。着信要求に許可される最大要求長。長さが長いリクエストはブロックされます。指定できる値は 0 ～65535 です。デフォルト:24820

コマンドラインを使用してバッファオーバーフローセキュリティチェックを設定する

コマンドラインを使用してバッファオーバーフローのセキュリティチェックアクションとその他のパラメータを設定するには

コマンドプロンプトで入力します。

```
add appfw profile <name> -bufferOverflowMaxURLLength <positive_integer
> -bufferOverflowMaxHeaderLength <positive_integer> - bufferOverflowMaxCookieL
 <positive_integer> -bufferOverflowMaxQueryLength <positive_integer>
-bufferOverflowMaxTotalHeaderLength <positive_integer>
```

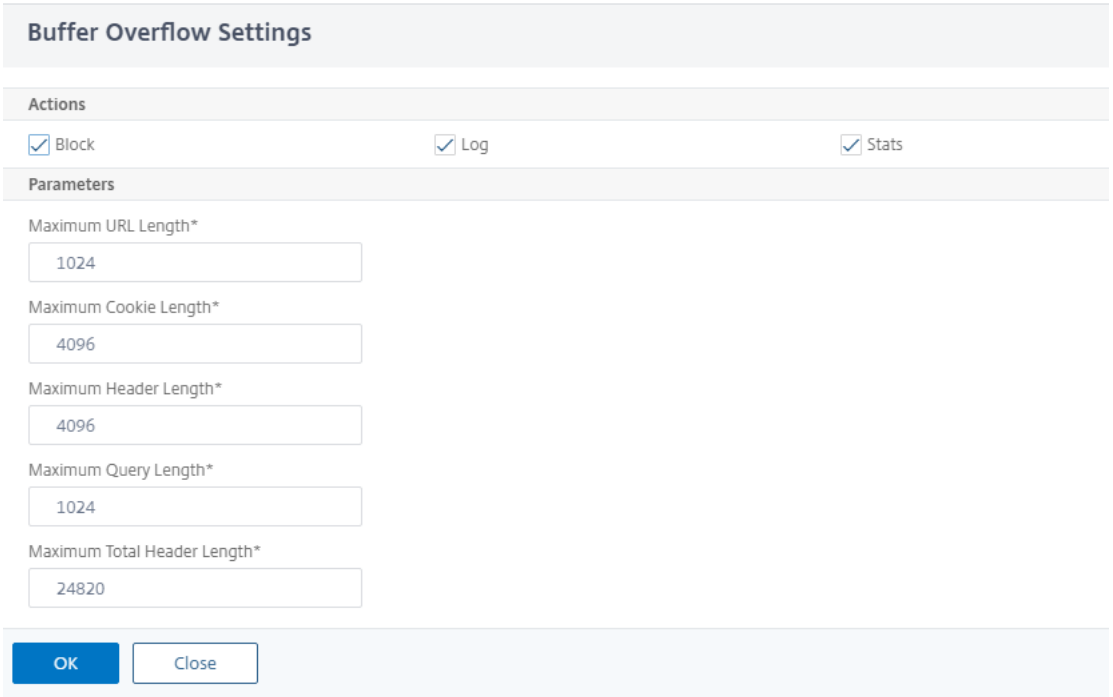
例**:**

```
add appfw profile profile1 -bufferOverflowMaxURLLength 7000 -bufferOverflowMaxH
 7250 - bufferOverflowMaxCookieLength 7100 -bufferOverflowMaxQueryLength
 7300 -bufferOverflowMaxTotalHeaderLength 7300
```

NetScaler GUI を使用してバッファオーバーフローセキュリティチェックを構成する

1. [セキュリティ] > [**Web App Firewall** とプロファイル] に移動します。
2. [プロファイル**]** ページでプロファイルを選択し、[編集] をクリックします。

3. [**NetScaler Web App Firewall** プロファイル] ページで、[詳細設定] セクションに移動し、[セキュリティチェック] をクリックします。
4. セキュリティチェックセクションで、「バッファオーバーフロー」を選択し、「アクション設定」をクリックします。
5. バッファオーバーフロー設定ページで、次のパラメータを設定します。
   a. アクション。コマンドインジェクションセキュリティチェックのために実行するアクションを 1 つ以上選択します。
   b. URL の最大長保護されている Web サイトの URL の最大長 (文字数)。より長い URL を持つリクエストはブロックされます。
   c. クッキーの最大長。保護された Web サイトに送信される Cookie の最大長 (文字数)。長い Cookie を含むリクエストはブロックされます。
   d. 最大ヘッダ長保護された Web サイトに送信されるリクエストの HTTP ヘッダーの最大長 (文字数)。長いヘッダーを持つリクエストはブロックされます。
   e. 最大クエリ長。保護された Web サイトに送信されるクエリ文字列の最大長 (バイト単位)。クエリ文字列が長いリクエストはブロックされます。
   f. ヘッダーの最大合計長。保護された Web サイトに送信されるリクエストの HTTP ヘッダーの合計長の最大長 (バイト単位)。これと HttpProfile の maxHeaderLen の最小値が使用されます。長さが長いリクエストはブロックされます。
6. 「**OK**」をクリックして「閉じる」をクリックします。



バッファオーバーフローセキュリティチェックでのログ機能の使用

** ログアクションを有効にすると、バッファオーバーフローのセキュリティチェック違反は **APPFW_BUFFEROVERFLOW_URL**、**APPFW_BUFFEROVERFLOW_COOKIE**、および ****APPFW_BUFFEROVERFLOW_** 違反として監査ログに記録されます。** Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

GUI を使用してログを確認する場合、Click-to-Deploy 機能を使用して、ログに示されている緩和を適用できます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて **/var/log/** フォルダー内の ns.logs を末尾に移動して、バッファオーバーフロー違反に関するログメッセージにアクセスします。

```
> \*\*Shell\*\*
> \*\*tail -f /var/log/ns.log | grep APPFW_BUFFEROVERFLOW\*\*
<!--NeedCopy-->
```

非ブロックモードでの BufferOverflowMaxCookieLength 違反を示す CEF ログメッセージの例

```
Oct 22 17:35:20 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|\*\*APPFW_BUFFEROVERFLOW_COOKIE\*\*|6|src=10.217.253.62
    geolocation=Unknown spt=41198 method=GET request=http://aaron.
    stratum8.net/FFC/sc11.html \*\*msg=Cookie header length(43) is
    greater than maximum allowed(16).\*\* cn1=119 cn2=465 cs1=
    owa_profile cs2=PPE1 cs3=wvOOOb+cJ2ZRbstZpyeNXIqLj7Y0001 cs4=ALERT
    cs5=2015 \*\*act=not blocked\*\*
<!--NeedCopy-->
```

非ブロックモードでの BufferOverflowMaxURL Length 違反を示す CEF ログメッセージの例

```
Oct 22 18:39:56 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|\*\*APPFW_BUFFEROVERFLOW_URL\*\*|6|src=10.217.253.62
    geolocation=Unknown spt=19171 method=GET request=http://aaron.
    stratum8.net/FFC/sc11.html \*\*msg=URL length(39) is greater than
    maximum allowed(20).\*\* cn1=707 cn2=402 cs1=owa_profile cs2=PPE0
    cs3=kW49GcKbnwKByByi3+jeNzfgWa80000 cs4=ALERT cs5=2015 \*\*act=not
    blocked\*\*
<!--NeedCopy-->
```

ブロックモードでの BufferOverflowMaxHeaderLength 違反を示すネイティブ形式のログメッセージの例

```
Oct 22 18:44:00 <local0.info> 10.217.31.98 10/22/2015:18:44:00 GMT ns
    0-PPE-2 : default APPFW \*\*APPFW_BUFFEROVERFLOW_HDR\*\* 155 0 :
    10.217.253.62 374-PPE2 khhBEeY4DB8V2D3H2sMLkXmfWnA0002 owa_profile
    \*\*Header(User-Agent) length(82) is greater than maximum allowed
    (10)\*\* : http://aaron.stratum8.net/ \*\*<blocked>\*\*
<!--NeedCopy-->
```

GUI を使用してログメッセージにアクセスするには

GUI には、ログメッセージを分析するための便利なツール (**Syslog Viewer**) が含まれています。Syslog ビューアにアクセスするには、複数のオプションがあります。

- [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを選択して [セキュリティチェック] をクリックします。[バッファオーバーフロー] 行を強調表示して [ログ] をクリックします。プロファイルのバッファオーバーフローセキュリティチェックから直接ログにアクセスすると、GUI はログメッセージをフィルタリングし、これらのセキュリティチェック違反に関連するログのみを表示します。
- 「**NetScaler** 」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。監査メッセージセクションで、**Syslog** メッセージリンクをクリックして Syslog Viewer を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティチェック違反がトリガーされる可能性がある場合のデバッグに役立ちます。
- アプリケーションファイアウォール > ポリシー > 監査に移動します。監査メッセージセクションで、**Syslog** メッセージリンクをクリックして Syslog Viewer を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

XML ベースの Syslog Viewer には、関心のあるログメッセージのみを選択するためのさまざまなフィルターオプションが用意されています。バッファオーバーフローチェックのログメッセージを選択するには、モジュールのドロップダウンリストオプションで ****APPFW** を選択してフィルタリングします。****** イベントタイプリストには、****** バッファオーバーフローセキュリティチェックに関するすべてのログメッセージを表示するための **APPFW_BUFFEROVERFLOW_URL**、**APPFW_BUFFEROVERFLOW_COOKIE**、**APPFW_BUFFEROVERFLOW_HDR** の 3 つのオプションがあります。**1** つまたは複数のオプションを選択して、選択をさらに絞り込むことができます。たとえば、****APPFW_BUFFEROVERFLOW_COOKIE** チェックボックスを選択して「****** 適用」ボタンをクリックすると、Cookie****** ヘッダーのバッファオーバーフローセキュリティチェック違反に関するログメッセージのみが Syslog ビューアに表示されます。特定のログメッセージの行にカーソルを置くと、モジュール、イベントタイプ、イベント **ID**、クライアント **IP** などの複数のオプションがログメッセージの下に表示されます。これらのオプションを選択すると、ログメッセージ内の対応する情報を強調表示できます。

Click-to-Deploy: GUI にはクリックしてデプロイする機能がありますが、これは現在 **URL** の長さ違反に関するバッファオーバーフローログメッセージでのみサポートされています。Syslog Viewer を使用すると、トリガーされた違反を確認できるだけでなく、ブロックされたメッセージの長さに基づいて、情報に基づいた判断を下すことができます。現在の値の制限が厳しすぎて誤検出が発生している場合は、メッセージを選択して展開し、現在の値をメッセージに表示されている URL 長の値に置き換えることができます。この操作では、ログメッセージは CEF ログ形式である必要があります。ログメッセージに緩和を適用できる場合は、行の **Syslog Viewer** ボックスの右端にチェックボックスが表示されます。チェックボックスを選択し、アクションリストからオプションを選択してリラクゼーションをデプロイします。**[**** 編集とデプロイ ****]**、**[**** デプロイ**]**、**[**すべてデプロイ] は、****APPFW_BUFFEROVERFLOW_URL** フィルタを使用すると、設定された **URL** の長さ違反に関連するすべてのログメッセージを分離できます。

個別のログメッセージを選択すると、[編集してデプロイ**]**、**[**デプロイ**]**、**[**** すべてデプロイ**]** の **3 **** つのアクションオプションすべてを使用できます。[編集してデプロイ] を選択すると、バッファオーバーフロー設定ダイアログが表示されます。リクエストで確認された新しい URL 長が [最大 **URL** 長**]** 入力フィールドに挿入されます。何も編集

せずに「閉じる」をクリックすると、現在の設定値は変更されません。「**OK**」ボタンをクリックすると、「最大 URL 長」の新しい値が以前の値に置き換わります。

> 注
>
> 表示されるバッファオーバーフロー設定ダイアログでは、ブロック、ログ、統計アクションのチェックボックスがオフになっているため、編集とデプロイオプションを選択した場合は再設定する必要があります。「**OK**」をクリックする前にこれらのチェックボックスがオンになっていることを確認してください。そうしないと、新しい URL の長さは設定されますが、アクションは「なし」に設定されます。

複数のログメッセージのチェックボックスを選択した場合、「Deploy」または「**DeployAll**」オプションを使用できます。デプロイされたログメッセージの URL の長さが異なる場合、設定された値は、選択したメッセージで確認されている中で最長の URL 長値に置き換えられます。ルールをデプロイしても、**BufferOverflowMaxURLLength** の値のみが変更されます。設定したアクションは保持され、変更されません。

GUI でクリック・トゥ・デプロイ機能を使用するには

1. Syslog ビューアの [モジュール] オプションで [**APPFW**] を選択します。
2. イベントタイプとして APPFW_BUFFEROVERFLOW_URL チェックボックスを有効にして、対応するログメッセージをフィルタリングします。
3. チェックボックスをオンにして、ルールを選択します。
4. オプションの「アクション」ドロップダウンリストを使用して、リラクゼーションをデプロイします。
5. [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを選択して [セキュリティチェック] をクリックして [バッファオーバーフロー] 設定ペインにアクセスし、[最大 **URL** 長] の値が更新されていることを確認します。

バッファオーバーフロー違反の統計

stats アクションを有効にすると、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、バッファオーバーフローセキュリティチェックのカウンタが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分サイズは、構成された設定によって異なります。たとえば、ブロックアクションが有効な場合、バッファオーバーフロー違反が 3 つあるページのリクエストでは、最初の違反が検出されるとページがブロックされるため、stats カウンタが 1 つ増えます。ただし、ブロックが無効になっている場合は、違反ごとに個別のログメッセージが生成されるため、同じリクエストを処理すると違反の統計カウンタが増えます。

コマンドラインを使用してバッファオーバーフローセキュリティチェックの統計情報を表示するには

コマンドプロンプトで入力します。

```
> sh appfw stats
```

特定のプロファイルの統計情報を表示するには、以下のコマンドを使用します。

```
> stat appfw profile <profile name>
```


GUI を使用してバッファオーバーフローの統計情報を表示するには

1. システム > セキュリティ > アプリケーションファイアウォールに移動します。
2. 右側のペインで、[統計 リンク] にアクセスします。
3. スクロールバーを使用して、バッファオーバーフロー違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

ハイライト

- バッファオーバーフローセキュリティチェックでは、URL、Cookie、およびヘッダーの最大長を制限するように制限を設定できます。
- ブロック、ログ、統計アクションにより、トラフィックを監視し、アプリケーションに最適な保護を設定できます。
- Syslog ビューアでは、バッファオーバーフロー違反に関するすべてのログメッセージをフィルタリングして表示できます。
- ****BufferOverflowMaxURLLength** 違反に対しては、クリックしてデプロイする機能がサポートされています ** 。個別のルールを選択して展開することも、複数のログメッセージを選択して、現在設定されている URL の最大長を微調整したり緩和したりすることもできます。選択したグループの URL の最大値が新しい値として設定され、現在違反としてフラグが付けられているこれらのリクエストがすべて許可されます。
- Web App Firewall は、受信したリクエストを検査する際に個々の Cookie を評価するようになりました。クッキーヘッダーで受信されたクッキーのいずれかの長さが、設定された **BufferOverflowMaxCookie-Length** を超えると、バッファオーバーフロー違反がトリガーされます。

> 重要
>
> リリース 10.5.e (59.13xx.e ビルドより前のいくつかの中間拡張ビルド) と 11.0 リリース (65.x より前のビルド) では、Web App Firewall の Cookie ヘッダーの処理が変更されました。これらのリリースでは、すべての Cookie が個別に評価され、Cookie ヘッダーで受信された 1 つの Cookie の長さが設定された bufferOverflowMaxCookieLength を超えると、バッファオーバーフロー違反がトリガーされます。この変更の結果、10.5 以前のリリースのビルドでブロックされたリクエストは許可される可能性があります。これは、Cookie の長さを決定するために Cookie ヘッダー全体の長さが計算されないためです。** 状況によっては、サーバーに転送される Cookie の合計サイズが許容値よりも大きく、サーバーが「400 Bad Request」で応答することがあります。
>
> この変更は元に戻されました。11.0 リリース 65.x およびそれ以降のビルドに加えて、10.5.e->59.13xx.e 以降の 10.5.e 拡張ビルドでの動作は、リリース 10.5 の非拡張ビルドと同様になりました。Cookie の長さを計算するときに、生の Cookie ヘッダー全体が考慮されるようになりました。Cookie の長さの決定には、前後のスペースと名前と値のペアを区切るセミコロン (;) 文字も含まれます。

Google ウェブツールキットのウェブアプリファイアウォールのサポート

August 15, 2023

注：この機能は NetScaler リリース 10.5.e で使用できます。

Google Web Toolkit（GWT）のリモートプロシージャコール（RPC）メカニズムを使用する Web サーバーは、GWT サポートを有効にするための特別な構成を必要とせずに、NetScaler Web App Firewall で保護できます。

GWT とは何か

GWT は、XMLHttpRequest や JavaScript の専門知識を持たないユーザーでも、複雑で高性能なウェブアプリケーションの構築と最適化に使用されます。このオープンソースの無料開発ツールキットは、小規模および大規模なアプリケーションの開発に広く使用されており、フライトやホテルなどの検索結果などのブラウザベースのデータを表示するために非常に頻繁に使用されます。GWT には、ほとんどのブラウザやモバイルデバイスで実行できる最適化された JavaScript スクリプトを作成するための Java API とウィジェットのコアセットが用意されています。GWT RPC フレームワークを使用すると、Web アプリケーションのクライアントコンポーネントとサーバーコンポーネントが HTTP 経由で Java オブジェクトを簡単に交換できます。GWT RPC サービスは、SOAP または REST に基づくウェブサービスとは異なります。これらは単純に、サーバーとクライアント上の GWT アプリケーションの間でデータを転送するための軽量な方法です。GWT は、メソッド呼び出しの引数と戻り値を交換する Java オブジェクトのシリアル化を処理します。

GWT を使用する人気の Web サイトについては、以下を参照してください。

https://www.quora.com/What-web-applications-use-Google-Web-Toolkit-%28GWT%29

GWT リクエストの仕組み

GWT RPC リクエストはパイプで区切られ、引数の数は可変です。HTTP POST のペイロードとして送信され、値は次のとおりです。

1. コンテンツタイプ = テキスト/x-gwt-rpc。Charset には任意の値を指定できます。
2. メソッド = POST

コンテンツタイプが「text/x-gwt-rpc」の場合、HTTP リクエストの GET リクエストと POST HTTP リクエストの両方が有効な GWT リクエストと見なされます。クエリ文字列が GWT リクエストの一部としてサポートされるようになりました。コンテンツタイプ「text/x-gwt-rpc」のリクエストクエリ部分を調べるには、アプリファイアウォールプロファイルの「InspectQueryContentTypes」パラメーターを「OTHER」に設定します。

次の例は、GWT リクエストの有効なペイロードを示しています。


```
5|0|8|http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com
    .test.client.TestService|testMethod|java.lang.String|java.lang.
    Integer| myInput1|java.lang.Integer/3438268394|1|2|3|4|2|5|6|7|8|1|
<!--NeedCopy-->
```

リクエストは次の 3 つの部分に分けることができます。

a) Header: 5|0|8|

上記のリクエストの最初の 3 桁5|0|8|は、それぞれ「バージョン、サブバージョン、テーブルのサイズ」を表しています。これらは正の整数でなければなりません。

b) 文字列テーブル**:**

```
http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com.
test.client.TestService|testMethod|java.lang.String|java.lang.Integer
|myInput1| java.lang.Integer/3438268394|
```

上記のパイプで区切られた文字列テーブルのメンバーには、ユーザー指定の入力が含まれています。これらの入力は Web App Firewall のチェック用に解析され、次のように識別されます。

- 1st : `http://localhost:8080/test/`

  これはリクエスト URL です。

- 2nd : 16878339`F02B83818D264AE430C20468`

  一意の HEX 識別子。この文字列に 16 進数以外の文字が含まれている場合、リクエストは形式が誤っていると見なされます。

- 3rd : `com.test.client.TestService`

  サービスクラス名

- 4th : `testMethod`

  サービスメソッド名

- 5th onwards: `java.lang.String`|`java.lang.Integer`|`myInput1`|`java.lang.Integer`/3438268394

  データ型とデータ。非プリミティブデータ型は次のように指定されます

  `<container>.<sub-cntnr>.name/<integer><identifier>`

c) Payload: 1|2|3|4|2|5|6|7|8|1|

ペイロードは、文字列テーブル内の要素への参照で構成されています。これらの整数値は、文字列テーブルの要素数より大きくすることはできません。

GWT アプリケーションの **Web** アプリケーションファイアウォール保護

Web App Firewall は GWT RPC リクエストを理解して解釈し、ペイロードにセキュリティチェック違反がないか検査し、指定されたアクションを実行します。

GWT リクエストの Web App Firewall ヘッダーと Cookie チェックは、他のリクエスト形式のものと同様です。適切な URL デコードと文字セット変換の後、文字列テーブルのすべてのパラメータが検査されます。GWT リクエストボディにはフィールド名は含まず、フィールド値のみが含まれます。Web App Firewall Field Format チェックを使用して、入力値を指定された形式と照合できます。このチェックを使用して入力の長さを制御することもできます。**** 入力に対するクロスサイトスクリプティング攻撃や **SQL** インジェクション攻撃は ** 、Web App Firewall で簡単に検出して阻止できます。

学習ルールと緩和ルール:GWT リクエストでは、緩和ルールの学習と展開がサポートされています。Web App Firewall ルールは <actionURL><gieldName> マッピングの形式です。GWT リクエスト形式にはフィールド名がないため、特別な処理が必要です。Web App Firewall は、学習したルールにダミーフィールド名を挿入し、緩和ルールとして展開できます。-IsRegex フラグは、GWT 以外のルールの場合と同じように機能します。

- アクション URL:

  RPC に応答する複数のサービスを同じ Web サーバー上で設定できます。HTTP リクエストには、RPC を処理する実際のサービスの URL ではなく、Web サーバーの URL が含まれます。そのため、HTTP リクエスト URL に基づいて緩和は適用されません。これは、その URL 上のすべてのサービスが対象フィールドに対して緩和されるためです。GWT リクエストの場合、Web App Firewall は、文字列テーブルの 4 番目のフィールドにある GWT ペイロードにある実際のサービスの URL を使用します。

- フィールド名:

  GWT リクエスト本文にはフィールド値のみが含まれているため、Web App Firewall は学習したルールを推奨するときに 1、2 などのダミーフィールド名を挿入します。

  GWT で学習したルールの例

```
POST /abcd/def/gh HTTP/1.1
Content-type: text/x-gwt-rpc
Host: 10.217.222.75
Content-length: 157

5|0|8|http://localhost:8080/acdtest/|16878339
   F02Baf83818D264AE430C20468|
com.test.client.TestService|testMethod|java.lang.String%3b|java.
   lang.Integer|onblur|

  The learn data will be as follows:
> sh learningdata pr1 crossSiteScripting
Profile: pr1    SecurityCheck: crossSiteScripting
1)  Url:    http://localhost:8080/acdtest/  >> From GWT Payload.
    Field:  10
    Hits:   1
 Done
```



```
16   <!--NeedCopy-->
```

GWT 緩和ルールの例

```
bind appfw profile pr1 -crossSiteScripting 1 abcd -isregex
NOTREGEX
```

ログメッセージ:Web App Firewall は、GWT リクエストで検出されたセキュリティチェック違反のログメッセージを生成します。不正な形式の GWT リクエストによって生成されたログメッセージには、簡単に識別できるように「GWT」という文字列が含まれます。

不正な形式の **GWT** リクエストのログメッセージの例**:**

```
Dec 5 21:48:02 <local0.notice> 10.217.31.247 12/05/2014:21:48:02 GMT
 ns 0-PPE-0 : APPFW Message 696 0 : "GWT RPC request with malformed
payload. <blocked>”
```

GWT リクエストと非 **GWT** リクエストの処理の違い:

同じペイロードでも、コンテンツタイプごとに異なる Web App Firewall セキュリティチェック違反が発生する可能性があります。次の例を考えてみましょう。

```
5|0|8|http://localhost:8080/acdtest/|16878339F02Baf83818D264AE430C20468
|com.test.client.TestService|testMethod|java.lang.String%3b|java.lang
.Integer|select|
```

コンテンツタイプ**:** アプリケーション**/x-www-form-URL** エンコード:

SQL インジェクションタイプが利用可能な 4 つのオプション (sqlSplCharandKeyword、sqlSplCharorKeyword、SQLKeyword、または sqlSplchar) のいずれかを使用するように構成されている場合、このコンテンツタイプで送信されたリクエストは SQL 違反になります。Web App Firewall は、上記のペイロードを処理する際、「&」をフィールド区切り文字、「=」を名前と値の区切り文字と見なします。これらの文字はどちらも投稿本文のどこにも表示されないため、コンテンツ全体が単一のフィールド名として扱われます。このリクエストのフィールド名には、SQL 特殊文字 (;) と SQL キーワード (select) の両方が含まれています。そのため、4 つの SQL インジェクションタイプのオプションすべてで違反が検出されます。

Content-type: text/x-gwt-rpc:

このコンテンツタイプで送信されたリクエストが SQL インジェクションタイプを sqlSplCharorKeyword、SQLKeyword、または SQLSplChar の 3 つのオプションのいずれかに設定した場合にのみ SQL 違反がトリガーされます。SQL インジェクションタイプがデフォルトのオプションである sqlSplCharandKeyword に設定されていても、違反は発生しません。Web App Firewall は、 | 縦棒を GWT リクエスト内の上記のペイロードのフィールド区切り文字と見なします。そのため、投稿本文はさまざまなフォームフィールド値に分割され、（前述の規則に従って）フォームフィールド名が追加されます。この分割により、SQL 特殊文字と SQL キーワードは別々のフォームフィールドの一部になります。

フォームフィールド 8: `java.lang.String%3b -\> %3b is the (;)char`

フォームフィールド 10: `select`

そのため、SQL インジェクションタイプが **sqlSplChar** に設定されている場合、フィールド 8 は SQL 違反を示します。**SQL** キーワードの場合、フィールド 10 は違反を示します。SQL Inject タイプに **sqlSplCharorKeyword** オプション（キーワードまたは特殊文字の有無を調べる）が設定されている場合、これら 2 つのフィールドのどちらかが違反を示している可能性があります。** デフォルトの **sqlSplCharAndKeyword** オプションでは違反は検出されません。これは、sqlSplChar と sqlKeyword の両方を含む値を持つ単一のフィールドはないためです。**

ヒント:

- GWT サポートを有効にするために特別な Web App Firewall を設定する必要はありません。
- コンテンツタイプは text/x-gwt-rpc でなければなりません。
- GWT ペイロードに適用されるすべての関連する Web App Firewall セキュリティチェックの緩和ルールを学習してデプロイすることは、サポートされている他のコンテンツタイプの場合と同じように機能します。
- POST リクエストのみが GWT で有効と見なされます。コンテンツタイプが text/x-gwt-rpc の場合、他のすべてのリクエストメソッドはブロックされます。
- GWT リクエストには、設定されているプロファイルの POST 本文制限が適用されます。
- セキュリティチェックのセッションレス設定は適用されないため無視されます。
- GWT ログメッセージでは CEF ログ形式がサポートされています。

Cookie 保護

August 15, 2023

クッキーは、ウェブサーバーからクライアントブラウザに送信される小さなパケットデータです。クッキーは、パスワード、ユーザー認証の詳細、資格情報などの機密データを HTTP 接続経由で伝送し、Web ブラウザーに保存します。したがって、情報を盗む攻撃者からクッキーを保護することは非常に重要です。

Cookie の整合性チェック**:** ユーザーのリクエストで返される Cookie を調べて、Web サーバーがそのユーザーに設定した Cookie と一致することを確認します。変更された Cookie が見つかった場合、要求が Web サーバーに転送される前に、要求から取り除かれます。詳細については、「Cookie の一貫性チェック」トピックを参照してください。

Cookie ハイジャック保護：ハイジャックとは、攻撃者が Cookie への不正アクセスを取得する状況を指します。許可されたアクセスから Cookie を保護するために、NetScaler Web App Firewall（WAF）は、WAF Cookie の一貫性検証とともにクライアントからの TLS 接続に挑戦します。新しいクライアント要求ごとに、アプライアンスは TLS 接続を検証し、要求内のアプリケーションとセッション Cookie の一貫性を検証します。詳細については、「Cookie ハイジャック保護」のトピックを参照してください。

SameSite cookie 属性： Set-Cookie HTTP 応答の `SameSite` 属性を使用すると、Cookie をファーストパーティまたは同じサイトのコンテキストに制限する必要があるかどうかを宣言できます。Cookie 設定は攻撃を軽減し、安全な Web 通信を提供します。詳細については、SameSite クッキー属性のトピックを参照してください。

Cookie 整合性チェック

August 15, 2023

Cookie の一貫性チェックは、ユーザーから返された Cookie を調べ、Web サイトがそのユーザーに対して設定した Cookie と一致することを確認します。変更された Cookie が見つかると、リクエストが Web サーバーに転送される前にリクエストから削除されます。また、Cookie の暗号化、Cookie のプロキシ、または Cookie へのフラグの追加によって、処理するすべてのサーバー Cookie を変換するように Cookie の一貫性チェックを構成することもできます。このチェックはリクエストとレスポンスに適用されます。

攻撃者は通常、以前に認証されたユーザーになりすまして機密性の高い個人情報にアクセスしたり、バッファオーバーフローを引き起こしたりするために Cookie を改ざんします。バッファオーバーフローチェックは、長い Cookie を使用してバッファオーバーフローを起こそうとする試みから保護します。Cookie の整合性チェックでは、最初のシナリオに焦点を当てます。

ウィザードまたは GUI を使用する場合、[Cookie
整合性チェックの変更] ダイアログボックスの [
全般] タブで、次のアクションを有効または無効にできます。

- ブロック
- ログ
- 使い方
- 統計
- トランスフォーム。有効にすると、Transform アクションによってすべての Cookie が次の設定で指定されているとおりに変更されます。
  - サーバーの **Cookie** を暗号化します。応答をクライアントに転送する前に、Cookie の一貫性チェック緩和リストに記載されているものを除き、Web サーバーによって設定された Cookie を暗号化します。暗号化された Cookie は、クライアントが後続の要求を送信すると復号化され、復号化された Cookie は保護された Web サーバに転送される前に要求に再挿入されます。次の暗号化タイプの 1 つを指定します。
    * なし。Cookie を暗号化または復号化しないでください。デフォルト。
    * 復号化のみ。暗号化された Cookie のみを復号化します。Cookie は暗号化しないでください。
    * セッションのみを暗号化します。セッション Cookie のみを暗号化します。永続的な Cookie は暗号化しないでください。暗号化された Cookie をすべて復号化します。
    * すべてを暗号化します。セッション Cookie と永続的な Cookie の両方を暗号化します。暗号化された Cookie をすべて復号化します。
      注:Cookie を暗号化する場合、Web App Firewall は Cookie に
      HttpOnly フラグを追加します。このフラグは、スクリプトが Cookie にアクセスしたり解析したりすることを防ぎます。したがって、このフラグは、スクリプトベースのウイルスやトロイの木馬が復号化された Cookie にアクセスし、その情報を使用してセキュリティを侵害することを防ぎ

ます。これは、[サーバーの Cookie の暗号化] パラメーター設定とは別に処理される [Cookie に追加するフラグ] パラメーター設定に関係なく行われます。

- プロキシサーバーの **Cookie**。Cookie の一貫性チェックの緩和リストに記載されているものを除き、Web サーバーによって設定されたすべての非永続的 (セッション) Cookie をプロキシします。Cookie は、既存の Web App Firewall セッション Cookie を使用してプロキシされます。Web App Firewall は、保護された Web サーバーによって設定されたセッション Cookie を取り除き、応答をクライアントに転送する前にローカルに保存します。クライアントが後続のリクエストを送信すると、Web App Firewall はセッション Cookie をリクエストに再挿入してから、保護された Web サーバーにリクエストを転送します。次のいずれかの設定を指定します。
  - なし。Cookie をプロキシしない。デフォルト。
  - セッションのみ。セッション Cookie のみプロキシする。永続的な Cookie をプロキシしない注:Cookie プロキシを有効にした後で無効にした場合 ([セッションのみ] に設定した後でこの値を [なし] に設定)、Cookie プロキシを無効にする前に確立されたセッションに対して Cookie プロキシが維持されます。したがって、Web App Firewall がユーザーセッションを処理している間は、この機能を安全に無効にできます。
- **Cookie** に追加するフラグ。変換中に Cookie にフラグを追加します。次のいずれかの設定を指定します。
  - なし。Cookie にフラグを追加しないでください。デフォルト。
  - **HTTP** のみ。httpOnly フラグをすべての Cookie に追加します。HttpOnly フラグをサポートするブラウザでは、このフラグが設定された Cookie にスクリプトからアクセスすることはできません。
  - セキュア。SSL 接続でのみ送信される Cookie に Secure フラグを追加します。Secure フラグをサポートするブラウザは、フラグ付きの Cookie をセキュアでない接続で送信しません。
  - **[**すべて**]**。すべての Cookie に HttpOnly フラグを追加し、SSL 接続でのみ送信される Cookie に Secure フラグを追加します。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して Cookie 整合性チェックを設定できます。

- `set appfw profile <name> -cookieConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]`
- `set appfw profile <name> -cookieTransforms ([**ON**] | [**OFF**])`
- `set appfw profile <name> -cookieEncryption ([**none**] | [**decryptOnly**] | [**encryptSession**] | [**encryptAll**])`
- `set appfw profile <name> -cookieProxying ([**none**] | [**sessionOnly**])`
- `set appfw profile <name> -addCookieFlags ([**none**] | [**httpOnly**] | [**secure**] | [**all**])`

Cookie 整合性チェックの緩和を指定するには、GUI を使用する必要があります。[Cookie 整合性チェックの変更] ダイアログボックスの [チェック] タブで、[追加] をクリックして [Cookie 整合性チェック緩和の追加] ダイアログボ

ックスを開くか、既存のリラクゼーションを選択して [開く] をクリックして [Cookie 整合性チェック緩和の変更] ダイアログボックスを開きます。どちらのダイアログボックスにも、リラクゼーションを構成するための同じオプションが表示されます。

Cookie 整合性チェック緩和の例を以下に示します。

- ログオンフィールド。次の式は、文字列 logon_ で始まり、その後に 2 文字以上 15 文字以下の文字または数字の文字列が続くすべての Cookie 名を除外します。

```
^logon_[0-9A-Za-z]{
2,15 }
$
<!--NeedCopy-->
```

- ログオンフィールド **(**特殊文字**)**。次の式では、文字列 türkçe-logon_ で始まり、その後に 2 文字以上 15 文字以下の文字または数字の文字列が続くすべての Cookie 名が除外されます。

```
^txC3xBCrkxC3xA7e-logon_[0-9A-Za-z]{
2,15 }
$
<!--NeedCopy-->
```

- 任意の文字列。sc-item_ という文字列にユーザーがショッピングカートに追加した商品の ID ([0-9a-zA-Z]+)、2 つ目のアンダースコア (_)、最後に希望する商品の数 ([1-9][0-9]?) が続くクッキーをユーザーが変更できるようにします。

```
^sc-item_[0-9A-Za-z]+_[1-9][0-9]?$
<!--NeedCopy-->
```

注意: 正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加する URL を正確に定義し、それ以外は何も定義していないことを確認してください。ワイルドカード、特にドットとアスタリスク（.*）メタキャラクタとワイルドカードの組み合わせを不注意に使用すると、意図しない Web コンテンツへのアクセスをブロックしたり、Cookie の一貫性チェックでそうしない攻撃を許可したりするなど、望ましくない結果が得られる可能性があります。ブロックされています。

Cookie ハイジャック対策

March 20, 2024

Cookie ハイジャック防止は、ハッカーによる Cookie 盗み攻撃を軽減します。セキュリティ攻撃では、攻撃者がユーザーセッションを乗っ取り、Web アプリケーションへの不正アクセスを取得します。ユーザーが銀行アプリケーションなどのウェブサイトを閲覧すると、ウェブサイトはブラウザとのセッションを確立します。セッション中、アプリケーションはログイン認証情報、ページ訪問などのユーザー詳細を Cookie ファイルに保存します。その後、Cookie ファイルが応答としてクライアントブラウザに送信されます。ブラウザは、アクティブなセッションを維持

するために Cookie を保存します。攻撃者はこれらの Cookie をブラウザの Cookie ストアから手動で盗むことも、ブラウザの不正な拡張機能を使って盗むこともできます。その後、攻撃者はこれらの Cookie を使用してユーザーの Web アプリケーションセッションにアクセスします。

Cookie 攻撃を軽減するために、NetScaler Web App Firewall（WAF）は、WAF Cookie 一貫性検証とともに、クライアントからの TLS 接続に挑戦します。新しいクライアント要求ごとに、アプライアンスは TLS 接続を検証し、要求内のアプリケーションとセッション Cookie の一貫性を検証します。攻撃者が犠牲者から盗まれたアプリケーション Cookie とセッション Cookie を混在させようとすると、Cookie の一貫性の検証は失敗し、設定された Cookie ハイジャックアクションが適用されます。Cookie の一貫性について詳しくは、「Cookie ーの一貫性チェック」を参照してください。

> 注:
>
> Cookie ハイジャック機能はロギングと SNMP トラップをサポートします。ロギングの詳細については ADM トピックを、SNMP 設定の詳細については SNMP トピックを参照してください。

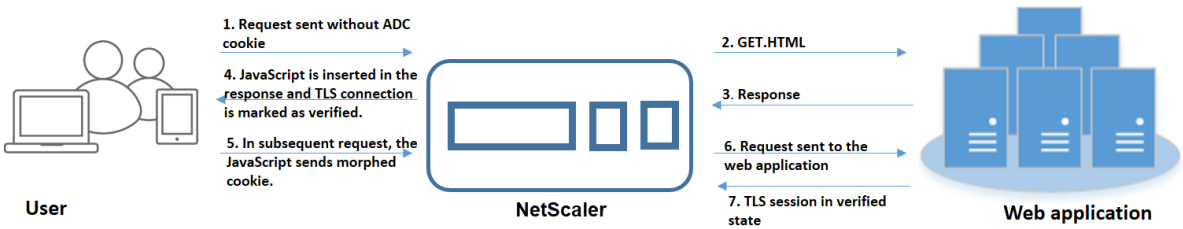
制限事項

- クライアントブラウザで JavaScript を有効にする必要があります。
- Cookie ハイジャック保護は、TLS バージョン 1.3 ではサポートされていません。
- ブラウザは SSL 接続を再利用しないため、Internet Explorer (IE) ブラウザのサポートが制限されています。リクエストに対して複数のリダイレクトが送信され、最終的に IE ブラウザで「最大リダイレクトを超えました」というエラーが発生します。

Cookie ハイジャック保護の仕組み

次のシナリオでは、NetScaler アプライアンスで Cookie ハイジャック保護がどのように機能するかを説明します。

シナリオ **1:** セッション **Cookie** なしで最初の **Web** ページにアクセスするユーザー

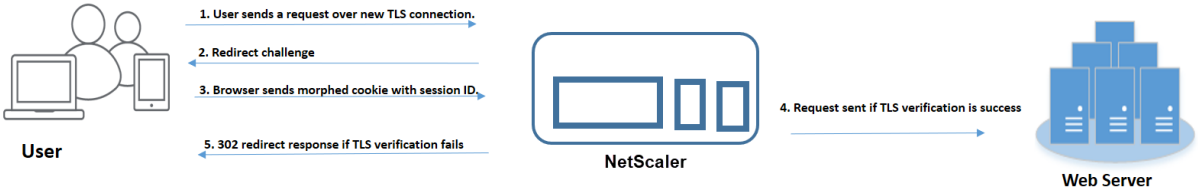


1. ユーザは Web アプリケーションへの認証を試み、要求に ADC セッション Cookie が含まれていない状態で最初の Web ページへのアクセスを開始します。
2. 要求を受信すると、アプライアンスはセッション Cookie ID を使用してアプリケーションファイアウォールセッションを作成します。
3. これにより、セッションの TLS 接続が開始されます。JavaScript はクライアントブラウザで送信および実行されないため、アプライアンスは TLS 接続を検証済みとしてマークし、チャレンジは不要です。

   > 注:
   >
   > 攻撃者がセッション Cookie を送信せずに被害者からすべてのアプリ Cookie ID を送信しようとしても、アプライアンスは問題を検出し、要求をバックエンドサーバーに転送する前に、要求に含まれるすべてのアプリ Cookie を削除します。バックエンドサーバーは、このリクエストをアプリ Cookie なしと見なし、その設定に従って必要に応じて処理します。

4. バックエンドサーバーが応答を送信すると、アプライアンスは応答を受信し、JavaScript セッショントークンとシード Cookie とともに転送します。その後、アプライアンスは TLS 接続を検証済みとしてマークします。
5. クライアントブラウザが応答を受け取ると、ブラウザは JavaScript を実行し、セッショントークンとシード Cookie を使用してモーフィングされた Cookie ID を生成します。
6. ユーザーが TLS 接続を介して後続のリクエストを送信すると、アプライアンスはモーフィングされた Cookie 検証をバイパスします。これは、TLS 接続がすでに検証されているためです。

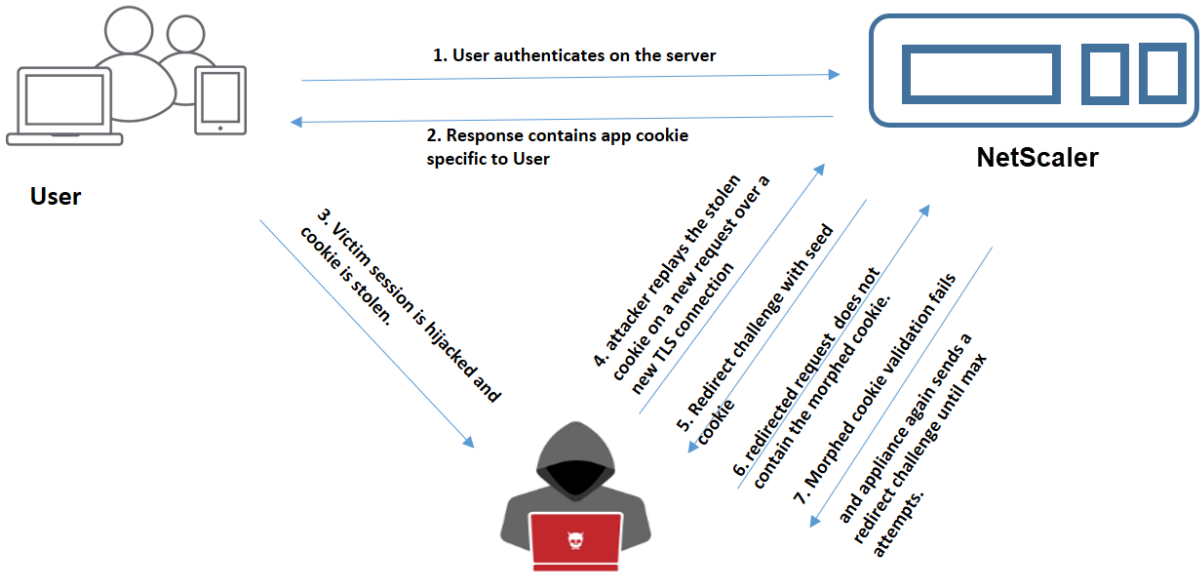
シナリオ **2:** セッション **Cookie** を使用して、新しい **TLS** 接続を介して連続する **Web** ページにアクセスするユーザー



1. ユーザーが新しい TLS 接続を介して連続ページの HTTP リクエストを送信すると、ブラウザはセッション Cookie ID とモーフィングされた Cookie ID を送信します。
2. これは新しい TLS 接続であるため、アプライアンスは TLS 接続を検出し、シード Cookie によるリダイレクト応答をクライアントに要求します。
3. クライアントは、ADC からの応答を受け取ると、セッションのトークンと新しいシード Cookie を使用してモーフィングされた Cookie を計算します。
4. 次に、クライアントは、この新しく計算されたモーフィングされた Cookie をセッション ID とともに送信します。

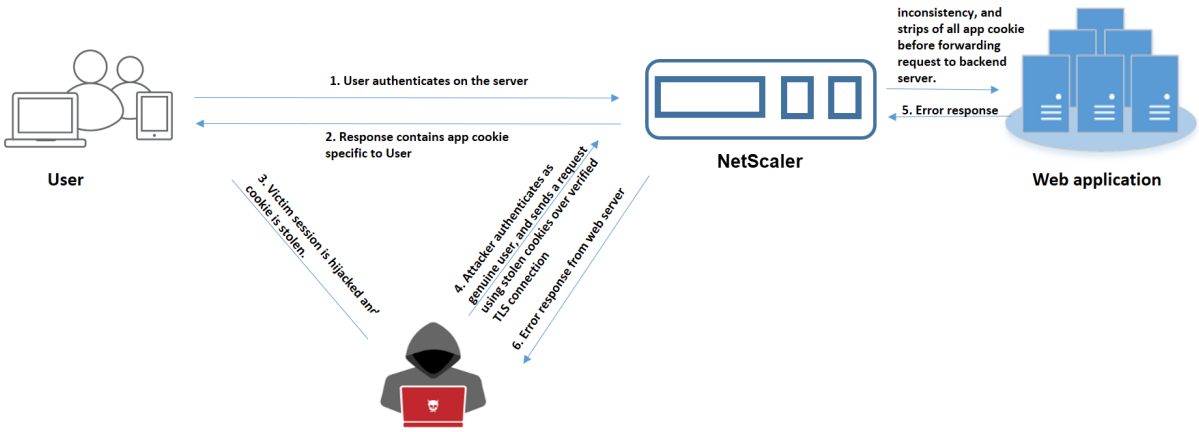
5. ADC アプライアンス内で計算されたモーフィングされた Cookie と、リクエストを介して送信された Cookie が一致すると、TLS 接続は検証済みとしてマークされます。
6. 計算されたモーフィングされた Cookie がクライアントのリクエストに含まれるものと異なる場合、検証は失敗します。その後、アプライアンスはチャレンジをクライアントに送り返し、適切にモーフィングされた Cookie を送信します。

シナリオ **3:** 攻撃者が認証されていないユーザーになりすます



1. ユーザーがウェブアプリケーションへの認証を行うと、攻撃者はさまざまな手法を使用して Cookie を盗み、再生します。
2. これは攻撃者からの新しい TLS 接続であるため、ADC は新しいシード Cookie と共にリダイレクトチャレンジを送信します。
3. 攻撃者は JavaScript を実行していないため、リダイレクトされたリクエストに対する攻撃者からの応答には、モーフィングされた Cookie は含まれていません。
4. その結果、ADC アプライアンス側でモーフィングされた Cookie 検証が失敗します。アプライアンスは再びリダイレクトチャレンジをクライアントに送信します。
5. モーフィングされた Cookie 検証の試行回数がしきい値制限を超えると、アプライアンスはステータスを Cookie Hijackking としてフラグ付けします。
6. 攻撃者が被害者から盗んだアプリケーション Cookie とセッション Cookie を混在させようとすると、Cookie の一貫性チェックは失敗し、アプライアンスは設定された Cookie ハイジャックアクションを適用します。

シナリオ **4:** 認証されたユーザーになりすます攻撃者



1. 攻撃者は、本物のユーザーとしてウェブアプリケーションへの認証を試み、被害者の Cookie を再生してウェブセッションにアクセスすることもできます。
2. ADC アプライアンスは、このようなりすまし攻撃者も検出します。攻撃者は検証済みの TLS 接続を使用して被害者の Cookie を再生しますが、ADC アプライアンスはリクエスト内のセッション Cookie とアプリケーション Cookie が一致しているかどうかを確認します。アプライアンスは、要求内のセッション Cookie を使用してアプリケーション Cookie の一貫性を検証します。リクエストには攻撃者のセッション Cookie と被害者のアプリ Cookie が含まれているため、Cookie の一貫性検証は失敗します。
3. その結果、アプライアンスは設定された Cookie ハイジャックアクションを適用します。設定されたアクションが「ブロック」に設定されている場合、アプライアンスはすべてのアプリケーション Cookie を削除し、要求をバックエンドサーバーに送信します。
4. バックエンドサーバーはアプリケーション Cookie ないリクエストを受け取り、攻撃者に「User not login」などのエラー応答を返します。

CLI を使用して Cookie ハイジャックを設定する

特定のアプリケーションファイアウォールプロファイルを選択し、Cookie ハイジャックを防止するアクションを 1 つ以上設定できます。

コマンドプロンプトで入力します:

```
set appfw profile <name> [-cookieHijackingAction <action-name> <block
 | log | stats | none>]
```

> 注:
>
> デフォルトでは、アクションは「なし」に設定されています。

例:

```
set appfw profile profile1 - cookieHijackingAction Block
```


ここで、アクションタイプは次のとおりです。

ブロック: このセキュリティチェックに違反する接続をブロックします。
ログ: このセキュリティチェックの違反をログに記録します。
統計: このセキュリティチェックの統計を生成します。
なし: このセキュリティチェックのすべてのアクションを無効にします。

NetScaler GUI を使用して **Cookie** ハイジャックを設定します

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. [プロファイル**]** ページでプロファイルを選択し、[編集] をクリックします。
3. [**NetScaler Web App Firewall** プロファイル］ページで、［詳細設定］セクションに移動し、［セキュリティチェック］をクリックします。
4. 「セキュリティチェック」セクションで「**Cookie** ハイジャック」を選択し、「アクション設定」をクリックします。
5. **Cookie** ハイジャック設定ページで、Cookie ハイジャックを防止するアクションを 1 つ以上選択します。
6. ［**OK**］をクリックします。

NetScaler GUI を使用して、**Cookie** 一貫性検証のための緩和ルールを追加します

Cookie の一貫性検証における誤検出を処理するために、Cookie の検証から除外できる Cookie の緩和ルールを追加できます。

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. [プロファイル**]** ページでプロファイルを選択し、[編集] をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、「詳細設定」セクションに移動し、「緩和ルール」をクリックします。
4. 「リラクゼーションルール」セクションで、「**Cookie** の一貫性」を選択し、「アクション」をクリックします。
5. **Cookie** の一貫性緩和ルールページで、次のパラメータを設定します。

    a) Enabled。緩和ルールを有効にするかどうかを選択します。
    b) Cookie 名は正規表現ですか. Cookie 名が正規表現であるかどうかを選択します。
    c) Cookie 名。Cookie の検証を免除できる Cookie ーの名前を入力します。
    d) 正規表現エディタ。このオプションをクリックすると、正規表現の詳細が表示されます。
    e) [コメント]。Cookie についての簡単な説明。

6. **[**作成**]** して **[**閉じる**]** をクリックします。

CLI を使用して Cookie ハイジャックのトラフィックと違反の統計情報を表示する

セキュリティトラフィックとセキュリティ違反の詳細を表形式またはグラフ形式で表示します。

セキュリティ統計を表示するには:

コマンドプロンプトで入力します:

```
stat appfw profile profile1
```

Appfw プロファイルのトラフィック統計	レート (/s)	合計
リクエスト	0	0
要求バイト数	0	0
レスポンス	0	0
送信バイト数	0	0
中止する	0	0
リダイレクト	0	0
長期平均応答時間 (ミリ秒)	–	0
最近の平均応答時間 (ミリ秒)	–	0

HTML/XML/JSON 違反統計	レート (/s)	合計
開始 URL	0	0
URL を拒否する	0	0
リファラーヘッダー	0	0
バッファオーバーフロー	0	0
Cookie の整合性	0	0
Cookie ハイジャック	0	0
CSRF フォームタグ	0	0
HTML クロスサイトスクリプティング	0	0
HTML SQL インジェクション	0	0
フィールド形式	0	0
フィールドの一貫性	0	0

HTML/XML/JSON 違反統計	レート (/s)	合計
クレジットカード	0	0
セーフオブジェクト	0	0
シグネチャ違反	0	0
コンテンツの種類	0	0
JSON サービス拒否	0	0
JSON SQL インジェクション	0	0
JSON クロスサイトスクリプティング	0	0
ファイルアップロードの種類	0	0
コンテンツタイプ XML ペイロードを推測	0	0
HTML CMD インジェクション	0	0
XML 形式	0	0
XML サービス拒否 (XDoS)	0	0
XML メッセージ検証	0	0
Web サービスの相互運用性	0	0
XML SQL インジェクション	0	0
XML クロスサイトスクリプティング	0	0
XML 添付ファイル	0	0
SOAP フォールト違反	0	0
XML ジェネリック違反	0	0
違反総数	0	0

HTML/XML/JSON ログ統計	レート (/s)	合計
URL ログの開始	0	0
URL ログの拒否	0	0
リファラーヘッダーログ	0	0
バッファオーバーフローログ	0	0
バッファオーバーフローログ	0	0

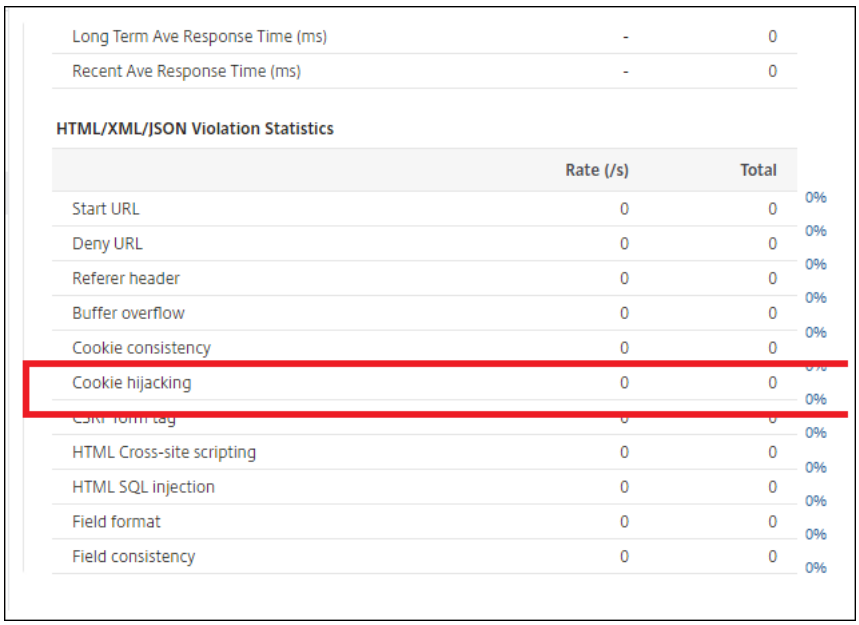
HTML/XML/JSON ログ統計	レート (/s)	合計
Cookie 整合性ログ	0	0
Cookie ハイジャックのログ	0	0
CSRF フォームタグログ	0	0
HTML クロスサイトスクリプティングログ	0	0
HTML クロスサイトスクリプティング変換ログ	0	0
HTML SQL インジェクションログ	0	0
HTML SQL 変換ログ	0	0
フィールド形式ログ	0	0
フィールド整合性ログ	0	0
クレジットカード	0	0
クレジットカード変換ログ	0	0
セーフオブジェクトログ	0	0
シグネチャログ	0	0
コンテンツタイプログ	0	0
JSON サービス拒否ログ	0	0
JSON SQL インジェクションログ	0	0
JSON クロスサイトスクリプティングログ	0	0
ファイルアップロードタイプログ	0	0
コンテンツタイプ XML ペイロードを推測 L	0	0
HTML コマンドインジェクションログ	0	0
XML 形式ログ	0	0
XML サービス拒否 (XDoS) ログ	0	0
XML メッセージ検証ログ	0	0
WSI ログ	0	0
XML SQL インジェクションログ	0	0
XML クロスサイトスクリプティングログ	0	0

HTML/XML/JSON ログ統計	レート (/s)	合計
XML 添付ファイルログ	0	0
SOAP フォールトログ	0	0
XML 汎用ログ	0	0
ログメッセージの総数	0	0

サーバーエラーレスポンス統計	レート (/s)	合計
HTTP クライアントエラー (4xx 回答)	0	0
HTTP サーバーエラー (5xx)	0	0

GUI を使用して Cookie ハイジャックのトラフィックと違反の統計情報を表示する

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ウィンドウで、**Web App Firewall** プロファイルを選択し、[統計] をクリックします。
3. **NetScaler Web App Firewall Statistics** ページには、Cookie ハイジャックトラフィックと違反の詳細が表示されます。
4. 表形式ビューを選択するか、グラフィカルビューに切り替えて、データを表形式またはグラフィカル形式で表示できます。



SameSite Cookie 属性

August 15, 2023

安全なウェブ通信のため、Google はSameSiteCookie 属性の使用を義務付けています。Google Chrome の新しいSameSiteポリシーに準拠することで、NetScaler アプライアンスはset-cookieヘッダーに設定されたSameSite属性を使用してサードパーティの Cookie を管理できます。Cookie 設定は攻撃を軽減し、安全な Web 通信を提供します。

2020 年 2 月まで、SameSite属性は Cookie に明示的に設定されていませんでした。ブラウザはデフォルト値を「None」に設定しました。ただし、Google Chrome 80 などの特定のブラウザのアップグレードでは、Cookie のデフォルトのクロスドメイン動作が変更されます。

Cookie 属性値の設定

SameSite 属性は次のいずれかの値に設定され、Google Chrome ブラウザではデフォルト値は「Lax」に設定されます。

[なし**]**。安全な接続でのみ、クロスサイトコンテキストのリクエストに Cookie を使用するようブラウザに指示します。

ラックス。同じサイトコンテキストでのリクエストに Cookie を使用するようにブラウザに指示します。クロスサイトコンテキストでは、GET リクエストなどの安全な HTTP メソッドのみが Cookie を使用できます。

厳密。Cookie は、ユーザーが明示的にドメインを要求している場合にのみ使用してください。

> 注:
>
> set-cookie（ファイアウォールセッション Cookie を含む）に SameSite属性があり、Web Application Firewall プロファイルでaddcookiesamesite属性フラグが有効になっている場合、SameSite属性はプロファイルで設定された値に従って上書きされます。

CLI を使用して Web App Firewall プロファイルの SameSite 属性を設定します

SameSite 属性を設定するには、次の手順を完了する必要があります。

1. SameSite Cookie 属性を有効にします。
2. appfw セッション Cookie のクッキー属性を設定します。

「Samesite」Cookie 属性を有効にする

コマンドプロンプトで入力します。


```
set appfw profile <profile-name> -insertCookieSameSiteAttribute ( ON
| OFF)
```

例**:**

```
set appfw profile p1 -insertCookieSameSiteAttribute ON
```

Web アプリケーションファイアウォールのセッション **Cookie** に同じサイト **Cookie** 属性値を設定

コマンドプロンプトで入力します。

```
set appfw profile <profile-name> - cookieSameSiteAttribute ( LAX |
NONE | STRICT )
```

例**:**

```
set appfw profile p1 - cookieSameSiteAttribute LAX
```

属性タイプがある場合、

[なし**]**。クッキー属性 SameSite は「なし」に設定され、すべての WAF クッキーとアプリケーションクッキーに対して安全とマークされています。

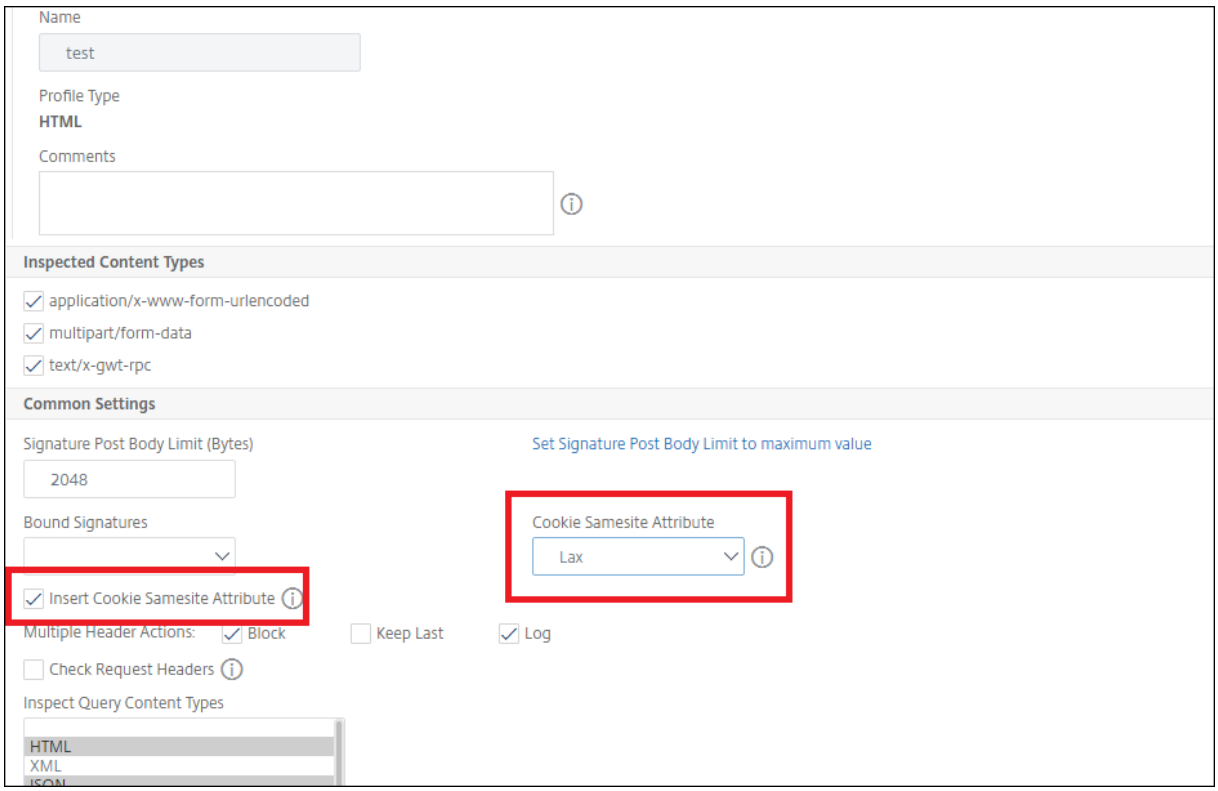
ラックス。クッキー属性 SameSite は、すべての WAF クッキーとアプリケーションクッキーで「Lax」に設定されています。

厳密。クッキー属性 SameSite は、すべての WAF クッキーとアプリケーションクッキーで「Lax」に設定されています。

GUI を使用して Web App Firewall プロファイルの SameSite Cookie 属性を設定します

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ペインで、プロファイルを選択し、[編集] をクリックします。
3. **NetScaler Web App Firewall**［プロファイル］ページで、［詳細設定］の［プロファイル設定］をクリックします。
4. **[**プロファイル設定**]** セクションで、次のパラメータを設定します。

   a. Cookie `Samesite` 属性を挿入します。チェックボックスを選択して Cookie `Samesite` 属性を有効にします。
   b. クッキーの同一サイト属性。ドロップダウンリストからオプションを選択し、`Samesite` Cookie の値を設定します。
5. [**OK**] をクリックし、[完了] をクリックします。



データ漏えい防止チェック

August 15, 2023

データ漏洩防止は、フィルタ応答をチェックし、クレジットカード番号や社会保障番号などの機密情報が許可されていない受信者に漏洩しないようにします。

クレジットカードチェック

August 15, 2023

クレジットカードを受け入れるアプリケーションを使用している場合、またはウェブサイトがクレジットカード番号を保存するデータベースサーバーにアクセスできる場合は、データ漏えい防止 (DLP) 対策を講じ、受け入れるクレジットカードの種類ごとに保護を設定する必要があります。

NetScaler Web App Firewall のクレジットカードチェックは、攻撃者がデータ漏えい防止の欠陥を悪用して顧客のクレジットカード番号を取得するのを防ぎます。簡単な設定手順に従うだけで、1）Visa、2）マスターカード、3）ディスカバー、4）アメリカンエキスプレス（アメックス）、5）JCB、6）ダイナースクラブのいずれかのクレジットカードを強制的に保護できます。

クレジットカードセキュリティチェックは、サーバーの応答を調べて対象のクレジットカード番号のインスタンスを特定し、そのような番号が見つかった場合は指定されたアクションを適用します。アクションとしては、クレジットカード番号の最後のグループを除くすべての桁を X アウトして応答を変換したり、指定された数を超えるクレジットカード番号が含まれている場合は応答をブロックしたりすることができます。両方を指定すると、ブロックアクションが優先されます。1 ページあたりに許可されるクレジットカードの最大数設定により、ブロックアクションがいつ呼び出されるかが決まります。デフォルト設定の 0 (ページにクレジットカード番号を入力できない) が最も安全ですが、最大 255 件まで許可できます。レスポンス内のどこで違反が検出され、ブロックアクションがトリガーされるかによっては、レスポンスで許可されているクレジットカードの最大数よりも少なくなる場合があります。

誤検出を防ぐために、緩和措置を適用して特定の番号をクレジットカードの小切手から免除することができます。たとえば、社会保障番号、注文番号、Google アカウント番号は、クレジットカード番号に似ている場合があります。クレジットカード検査用の応答 URL を処理する際に無視する数字列を指定するには、数字を個別に指定することも、正規表現を使用して指定することもできます。

どのクレジットカード番号を免除すればよいかわからない場合は、学習機能を使用して、学習したデータに基づいてレコメンデーションを生成できます。パフォーマンスを犠牲にすることなく最適な効果を得るには、このオプションを短時間有効にしてルールの代表的なサンプルを入手してから、緩和策を適用して学習を無効にするとよいでしょう。

ログ機能を有効にすると、クレジットカードチェックが実行したアクションを示すログメッセージが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増えている場合は、アクセスの試みが阻止された可能性があります。デフォルトでは、doSecureCreditCardLogging パラメータは ON なので、クレジットカード番号はセーフコマース (クレジットカード) 違反によって生成されるログメッセージには含まれません。

統計機能は、違反とログに関する統計を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。

アプリケーションを保護するためのクレジットカードセキュリティチェックを設定するには、このアプリケーションとの間で送受信されるトラフィックの検査を制御するプロファイルを設定します。

> 注記:
>
> SQL データベースにアクセスしない Web サイトでは、通常、クレジットカード番号などの機密の個人情報にはアクセスできません。

コマンドラインを使用してクレジットカードチェックを設定する

コマンドラインインターフェイスでは、set appfw profile コマンドまたは add appfw profile コマンドのいずれかを使用してクレジットカードチェックを有効にし、実行するアクションを指定できます。unset appfw profile コマンドを使用して、デフォルト設定に戻すことができます。緩和を指定するには、bind appfw コマンドを使用してクレジットカード番号をプロファイルにバインドします。

コマンドラインを使用してクレジットカードチェックを設定するには

次のように set appfw profile コマンドまたは add appfw profile コマンドのいずれかを使用します。

- `set appfw profile <name> -creditCardAction ( ([block][learn] [log][stats])| [none])`
- `set appfw profile <name> -creditCard (VISA | MASTERCARD | DISCOVER | AMEX | JCB | DINERSCLUB)`
- `set appfw profile <name> -creditCardMaxAllowed <integer>`
- `set appfw profile <name> -creditCardXOut ([ON] | [OFF])<name> -doSecureCreditCardLogging ([ON] | [OFF])`
- コマンドラインを使用してクレジットカード緩和ルールを設定するには

  bind コマンドを使用して、クレジットカード番号をプロファイルにバインドします。クレジットカード番号をプロファイルから削除するには、バインドコマンドで使用したのと同じ引数を指定して unbind コマンドを使用します。show コマンドを使用すると、プロファイルにバインドされたクレジットカード番号を表示できます。

- クレジットカード番号をプロフィールにバインドするには

  `bind appfw profile <profile-name> -creditCardNumber <any number/regex> “<url>”`

  例: バインド appfw プロファイル test_profile-クレジットカード番号 378282246310005 `http://www.example.com/credit\\_card\\_test.html`

  - クレジットカード番号をプロフィールからバインド解除するには

    `unbind appfw profile <profile-name> -creditCardNumber <credit card number / regex> <url>`

  - プロファイルに紐付けられたクレジットカード番号のリストを表示します。

    `show appfw profile <profile>`

GUI を使用してクレジットカードチェックを設定する

GUI では、アプリケーションに関連するプロファイルのペインでクレジットカードのセキュリティチェックを設定します。

GUI を使用してクレジットカードのセキュリティチェックを追加または変更するには

1. **Web App Firewall** > プロファイルに移動し、ターゲットプロファイルを強調表示して、「編集」をクリックします。
2. [詳細設定] ペインで、[セキュリティチェック] をクリックします。

セキュリティー検査テーブルには、すべてのセキュリティー検査に対して現在構成されているアクション設定が表示されます。設定には次の 2 つのオプションがあります。

a) クレジットカードのブロック、ログ、統計、学習アクションを有効または無効にするだけの場合は、表のチェックボックスをオンまたはオフにして、「**OK**」をクリックし、「****** 保存して閉じる ****** 」をクリックしてセキュリティチェックペインを閉じます。
b) このセキュリティチェックに追加のオプションを設定する場合は、「クレジットカード」をダブルクリックするか、行を選択して「アクション設定」をクリックすると、次のような追加オプションが表示されます。
   - アウト：最後のグループの数字を除く各桁を文字「X」に置き換えて、応答で検出されたすべてのクレジットカード番号をマスクします。
   - 1 ページあたりに許可されるクレジットカードの最大数-ブロックアクションをトリガーせずにクライアントに転送できるクレジットカードの数を指定します。
   - 保護されたクレジットカード。チェックボックスをオンまたはオフにして、クレジットカードの種類ごとに保護を有効または無効にします。
   - 「クレジットカード設定」ペインで「ブロック」、「ログ」、「統計」、「学習」のアクションを編集することもできます。
     上記の変更を行った後、「OK」をクリックして変更を保存し、「Security Checks」テーブルに戻ります。必要に応じて、他のセキュリティー検査の設定に進むことができます。「OK」をクリックして「セキュリティ・チェック」セクションで行ったすべての変更を保存し、「保存して閉じる」をクリックしてセキュリティ・チェック・ペインを閉じます。

3. [詳細設定] ペインで、[プロファイル設定] をクリックします。クレジットカード番号の安全なロギングを有効または無効にするには、「セキュア・クレジット・カード・ロギング」チェックボックスをオンまたはオフにします。(デフォルトでは選択されています)。

   「**OK**」をクリックして変更を保存します。

- GUI を使用してクレジットカード緩和ルールを設定するには

  1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
  2. [詳細設定**]** ペインで、[緩和規則] をクリックします。リラクゼーションルールテーブルにはクレジットカードのエントリがあります。ダブルクリックするか、この行を選択して [編集] をクリックすると、クレジットカード緩和ルールダイアログが表示されます。緩和ルールの [追加]、[編集]、[削除]、[有効化]、または [無効化] 操作を実行できます。

クレジットカード小切手での学習機能の使用

学習アクションが有効になると、Web App Firewall 学習エンジンはトラフィックを監視し、トリガーされた違反を学習します。学習したルールは定期的に検査できます。十分に検討したうえで、特定の数字列をクレジットカードのセキュリティチェックから除外したい場合は、学習したルールを緩和ルールとして適用できます。

- コマンドラインインターフェイスを使用して学習データを表示または使用するには

```
show appfw learningdata <profilename> creditCardNumber

rm appfw learningdata <profilename> -creditcardNumber <credit
card number> "<url>"

export appfw learningdata <profilename> creditCardNumber
```

- GUI を使用して学習済みデータを表示または使用するには

  1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
  2. [詳細設定] ペインで、[学習済みルール] をクリックします。「学習ルール」テーブルでクレジットカードのエントリを選択してダブルクリックすると、学習したルールにアクセスできます。学習したルールを展開したり、緩和ルールとして展開する前にルールを編集したりできます。ルールを破棄するには、ルールを選択して「スキップ」(**Skip**) ボタンをクリックします。一度に編集できるルールは 1 つだけですが、展開またはスキップするルールは複数選択できます。

  また、「学習ルール」テーブルで「クレジットカード」エントリを選択し、「ビジュアライザー」をクリックすると、学習したすべての違反をまとめて表示できます。ビジュアライザーを使用すると、学習したルールを非常に簡単に管理できます。1 つの画面でデータの包括的なビューを表示し、1 回のクリックでルールのグループに対するアクションの実行を容易にします。ビジュアライザーの最大の利点は、正規表現を推奨して複数のルールを統合できることです。デリミタとアクション URL に基づいて、これらのルールのサブセットを選択できます。ドロップダウンリストから番号を選択すると、ビジュアライザーに 25、50、または 75 個のルールを表示できます。学習したルールのビジュアライザーには、ルールを編集して緩和として展開するオプションがあります。または、ルールをスキップして無視することもできます。

クレジットカードチェックでのログ機能の使用

ログアクションを有効にすると、クレジットカードのセキュリティチェック違反は APPFW_SAFECOMMERCE または APPFW_SAFEMCOMMERCE_XFORM 違反として監査ログに記録されます。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

DoSecureCreditCardLogging のデフォルト設定は ON です。OFF に変更すると、クレジットカード番号と種類の両方がログメッセージに含まれます。

クレジットカードチェックの設定によっては、アプリケーションファイアウォールが生成するログメッセージには次の情報が含まれる場合があります。

- 応答がブロックされたか、ブロックされなかったか。
- クレジットカード番号が変換されました（X が出力されました）。変換されたクレジットカード番号ごとに個別のログメッセージが生成されるため、1 つの応答の処理中に複数のログメッセージが生成される可能性があります。

- 回答には、考えられるクレジットカード番号が最大数含まれていました。
- クレジットカード番号とそれに対応するタイプ。
- コマンドラインを使用してログメッセージにアクセスするには

  シェルに切り替えて、/var/log/ フォルダー内の ns.logs を末尾に移動して、クレジットカード違反に関するログメッセージにアクセスします。

  - Shell

tail -f /var/log/ns.log	grep SAFECOMMERCE

  -

- GUI を使用してログメッセージにアクセスするには

  1. GUI には、ログメッセージを分析するための非常に便利なツール (Syslog Viewer) が含まれています。Syslog Viewer にアクセスするにはいくつかの方法があります。ターゲットプロファイル > セキュリティチェックに移動します。「クレジットカード」行を強調表示して、「ログ」をクリックします。プロファイルのクレジットカードセキュリティチェックから直接ログにアクセスすると、ログメッセージが除外され、これらのセキュリティチェック違反に関連するログのみが表示されます。
  2. 「**NetScaler** 」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。監査メッセージセクションで、**Syslog** メッセージリンクをクリックして Syslog Viewer を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティチェック違反がトリガーされる可能性がある場合のデバッグに役立ちます。

     HTML ベースの Syslog ビューアには、関心のあるログメッセージのみを選択するためのさまざまなフィルタオプションがあります。クレジットカードのセキュリティチェック違反ログメッセージにアクセスするには、モジュールのドロップダウンオプションで [APPFW] を選択してフィルタリングします。[イベントタイプ] には、選択をさらに絞り込むための豊富なオプションセットが表示されます。たとえば、「APPFW_SAFECOMMERCE」と「APPFW_SAFECOMMERCE_XFORM」チェックボックスを選択して「適用」ボタンをクリックすると、クレジットカードのセキュリティチェック違反に関するログメッセージのみが Syslog ビューアに表示されます。

     特定のログメッセージの行にカーソルを置くと、Module や EventType などの複数のオプションがログメッセージの下に表示されます。これらのオプションのいずれかを選択して、ログ内の対応する情報を強調表示できます。

応答がブロックされていない場合のネイティブ形式のログメッセージの例

```
May 29 01:26:31 <local0.info> 10.217.31.98 05/29/2015:01:26:31 GMT ns
    0-PPE-0 :
default APPFW APPFW_SAFECOMMERCE 2181 0 :  10.217.253.62 1098-PPE0
```



```
4erNfkaHy0IeGP+nv2S9Rsdu77I0000 pr_ffc http://aaron.stratum8.net/FFC/
    CreditCardMind.html
Maximum number of potential credit card numbers seen <not blocked>
<!--NeedCopy-->
```

応答が変換された場合の CEF 形式のログメッセージの例

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE_XFORM|6|src
    =10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/
    CreditCardMind.html
msg=Transformed (xout) potential credit card numbers seen in server
    response
cn1=66 cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVl80002
cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

応答がブロックされた場合の CEF 形式のログメッセージの例。doSecureCreditCardLogging パラメータが無効になっているため、クレジットカードの番号と種類がログに表示されます。

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE|6|src
    =10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/
    CreditCardMind.html
msg=Credit Card number 4505050504030302 of type Visa is seen in
    response cn1=68
cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVl80002 cs4=
    ALERT cs5=2015
act=blocked
<!--NeedCopy-->
```

クレジットカード違反の統計

統計アクションを有効にすると、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、クレジットカードチェックに対応するカウンタが増えます。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンターのインクリメントは、構成された設定によって異なる場合があります。たとえば、ブロックアクションが有効で [クレジットカードの最大許容数] 設定が 0 の場合、20 個のクレジットカード番号を含むページのリクエストでは、最初のクレジットカード番号が検出されるとすぐに、ページがブロックされると、統計カウンタが 1 ずつ増えます。ただし、ブロックが無効で変換が有効な場合、クレジットカードの変換ごとに個別のログメッセージが生成されるため、同じリクエストを処理するとログの統計カウンタが 20 ずつ増えます。

- コマンドラインを使用してクレジットカードの統計情報を表示するには

  コマンドプロンプトで入力します。

  ```
  sh appfw stats
  ```


特定のプロファイルの統計情報を表示するには、以下のコマンドを使用します。

```
stat appfw profile <profile name>
```

GUI を使用してクレジットカードの統計情報を表示するには

1. [システム] > [セキュリティ] > [**Web App Firewall**] に移動します。
2. 右側のペインで、[統計 リンク] にアクセスします。
3. スクロールバーを使用すると、クレジットカード違反やログに関する統計が表示されます。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

ハイライト

クレジットカードのセキュリティチェックについては、次の点に注意してください。

- Web App Firewall を使用すると、クレジットカード情報を保護し、この機密データにアクセスしようとする試みを検出できます。
- クレジットカード保護チェックを使用するには、少なくとも 1 種類のクレジットカードとアクションを指定する必要があります。その後、チェックは HTML、XML、および Web 2.0 のプロファイルに適用されます。

sh appfw profile コマンドと grep for CreditCard の出力をパイプすると、クレジットカード固有の設定をすべて表示できます。たとえば、.sh.appfw プロファイル、my_profile などです。	grep CreditCard は、さまざまなパラメータの設定と、my_profile という名前の Web App Firewall プロファイルのクレジットカードチェックに関連する緩和ルールを表示します。

-
- クレジットカード番号のセキュリティチェックをバイパスせずに、特定の番号をクレジットカード検査から除外できます。
- リラクゼーションは、Web App Firewall で保護されているすべてのクレジットカードパターンで利用できます。GUI では、ビジュアライザーを使用して、緩和ルールの追加、編集、削除、有効化、または無効化操作を指定できます。
- Web App Firewall の学習エンジンは、送信トラフィックを監視して、検出された違反に基づいてルールを推奨できます。ビジュアライザーのサポートにより、学習したクレジットカードルールを GUI で管理することもできます。学習したルールは編集して展開することも、慎重に調べてからスキップすることもできます。
- 許可されるクレジットカードの数の設定は、各回答に適用されます。ユーザーセッション全体で確認されたクレジットカード番号の累積合計には関係ありません。
- 出力される X 桁の数は、クレジットカード番号の長さによって異なります。13 ～15 桁のクレジットカードでは、10 桁の X が出力されます。16 桁のクレジットカードの場合、12 桁の X が出力されます。アプリケーションでクレジットカード番号全体をレスポンスで送信する必要がない場合は、このアクションを有効にしてクレジットカード番号の数字を隠すことをお勧めします。

- X-out 操作はすべてのクレジットカードを変換し、許可されるクレジットカードの最大数の設定とは無関係に機能します。たとえば、レスポンスにクレジットカードが 4 枚あり、CreditCardMaxAllowed パラメータが 10 に設定されている場合、4 枚のクレジットカードすべてが X’D アウトになりますが、ブロックされません。クレジットカード番号が文書内に散らばっている場合、応答がブロックされる前に、X’d-out 番号を含む部分的な応答がクライアントに送信される可能性があります。
- 十分に検討する前に、doSecureCreditCardLogging パラメータを無効にしないでください。このパラメータをオフにすると、クレジットカード番号が表示され、ログメッセージからアクセスできるようになります。X-out アクションが有効になっていても、これらの数字はログに隠されません。ログをリモート Syslog サーバに送信していて、ログが漏えいした場合、クレジットカード番号が漏洩する可能性があります。
- クレジットカード違反のため応答ページがブロックされると、Web App Firewall はエラーページにリダイレクトされません。

セーフオブジェクトチェック

August 15, 2023

セーフオブジェクトチェックは、顧客番号、注文番号、国固有または地域固有の電話番号または郵便番号など、機密性の高いビジネス情報をユーザーが構成可能な保護を提供します。ユーザー定義の正規表現またはカスタムプラグインは、Web App Firewall にこの情報の形式を伝え、情報を保護するために使用するルールを定義します。ユーザーリクエストの文字列がセーフオブジェクト定義と一致する場合、Web App Firewall は、その特定のセーフオブジェクトルールの構成方法に応じて、レスポンスをブロックするか、保護された情報をマスクするか、またはユーザーに送信する前にレスポンスから保護された情報を削除します。

セーフオブジェクトチェックは、攻撃者が Web サーバーソフトウェアまたは Web サイトのセキュリティ上の欠陥を悪用して、会社のクレジットカード番号や社会保障番号などの機密性の高い個人情報を入手することを防ぎます。Web サイトがこれらの種類の情報にアクセスできない場合は、このチェックを構成する必要はありません。このような情報にアクセスできるショッピングカートやその他のアプリケーションがある場合、または Web サイトがそのような情報を含むデータベースサーバーにアクセスできる場合は、処理および保存する機密性の高い個人情報の種類ごとに保護を構成する必要があります。

> 注:
>
> SQL データベースにアクセスしないウェブサイトは、通常、機密性の高い個人情報にアクセスできません。

セーフオブジェクトチェックは、他のチェックとは異なります。作成する各セーフオブジェクト式は、クレジットカードチェックと同様に、その種類の情報に対する個別のセキュリティチェックと同等です。

GUI を使用してセーフオブジェクトチェックを構成する

> 注:
>
> セーフオブジェクトチェックは、GUI を使用してのみ設定する必要があります。コマンドラインインターフェイスはサポートされていません。

GUI を使用してセーフオブジェクトセキュリティー検査を追加するには、次の手順に従います。

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 必要なプロファイルを選択し、[編集] をクリックします。
3. [詳細設定] ウィンドウで、[緩和規則] をクリックします。
4. [セーフオブジェクト] を選択し、[編集] をクリックします。
5. [追加] をクリックして、以下を構成します。

   - 安全なオブジェクト名。新しいセーフオブジェクトの名前。名前は、英字、数字、またはアンダースコア記号で始めることができます。名前には、1 ～255 の英数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) 記号を使用できます。
   - アクション。ブロック、ログ、統計アクション、および次のアクションを有効または無効にします。
     - **X-Out**。セーフオブジェクト式と一致する情報を文字「X」でマスクします。
     - **Remove-** セーフオブジェクト式と一致する情報をすべて削除します。
   - 正規表現。セーフオブジェクトを定義する PCRE 互換の正規表現を入力します。正規表現は次のいずれかの方法で作成できます。
     - テキストボックスに正規表現を直接入力する
     - [正規表現トークン] メニューを使用して、正規表現の要素と記号をテキストボックスに直接入力する
     - 正規表現エディタを開き、それを使用して式を作成します。正規表現は ASCII 文字のみで構成されている必要があります。基本的な 128 文字の ASCII セットの一部ではない文字を切り取って貼り付けないでください。非 ASCII 文字を含める場合は、これらの文字を PCRE 16 進文字エンコード形式で手動で入力する必要があります。

     > 注:
     >
     > セーフオブジェクト式の先頭に開始アンカー (^) を使用したり、セーフオブジェクト式の最後に終了アンカー ($) を使用したりしないでください。これらの PCRE エンティティはセーフオブジェクト式ではサポートされていません。使用すると、式が意図したものと一致しなくなります。

   - 最大一致長。一致させる文字列の最大長を表す正の整数を入力します。たとえば、米国の社会保障番号を照合する場合は、このフィールドに番号 11 を入力します。これにより、正規表現で 9 つの数字と 2 つのハイフンを含む文字列に一致させることができます。カリフォルニア州の運転免許証番号と一致させたい場合は、数字の 8（8）を入力します。

> 注意：
>
> 最大一致長を指定しない場合、Web App Firewall は、セーフオブジェクト式に一致する文字列をフィルタリングするときに、既定値の 1 を使用します。その結果、ほとんどの安全なオブジェクト式はターゲット文字列と一致しません。

既存のエクスプレスを変更するには、必要な式を選択し、[開く] をクリックして、[セーフオブジェクトの変更] ダイアログボックスで式を構成します。

以下は、セーフオブジェクトチェックの正規表現の例です。

- 米国の社会保障番号 (SSN) のように見える文字列を探します。SSN は、次の文字を上記の順序で構成します。
  - 三つの数字 (最初の数字はゼロであってはならない)
  - ハイフン
  - あと 2 つの数字
  - 2 つ目のハイフン
  - あと 4 つの数字の文字列

```
[1-9][0-9]{
3,3 }
-[0-9]{
2,2 }
-[0-9]{
4,4 }

<!--NeedCopy-->
```

- カリフォルニア州の運転免許証 ID のように見える文字列を探します。文字で始まり、その後に正確に 7 つの数字の文字列が続きます。

```
[A-Za-z][0-9]{
7,7 }

<!--NeedCopy-->
```

- 顧客 ID のように見える文字列を探します。顧客 ID は、次の順序で構成されます。
  - 5 つの 16 進数文字の文字列 (すべての数字と A から F までの英文字)
  - ハイフン
  - 3 文字のコード
  - 2 つ目のハイフン
  - 10 個の数字の文字列

```
[0-9A-Fa-f]{
5,5 }
-[A-Za-z]{
3,3 }
```



```
-[0-9]{
10,10 }

<!--NeedCopy-->
```

> 注意:
>
> 正規表現は強力です。PCRE 形式の正規表現に慣れていない場合は、記述した正規表現を再確認してください。正規表現が、安全なオブジェクト定義として追加する文字列のタイプを正確に定義していることを確認してください。ワイルドカード、特にドットとアスタリスク (.*) メタキャラクタとワイルドカードの組み合わせを不注意に使用すると、ブロックする意図がなかった Web コンテンツへのアクセスをブロックするなど、望ましくない結果が生じる可能性があります。

高度なフォーム保護チェック

August 15, 2023

高度なフォーム保護チェックでは、Web フォームデータを検査して、Web サイト上の Web フォームを変更したり、予期しない種類や量のデータをフォームで Web サイトに送信したりして、攻撃者がシステムを侵害するのを防ぎます。

> 注:
>
> [セキュリティチェックからアップロードファイルを除外] が設定されていない場合は、**SQL**、クロスサイトスクリプティング、**FFC**、および **FieldFormat** 保護チェックが適用されます。

ファイルのアップロードは、フォーム送信の一部として送信されるコントロール 名 フィールドを持つフォーム要素でもあります。

詳細については、このページを参照してください。Forms

> 注
>
> フォームベースのチェックが有効になっている場合、フォーム保護はネストされたフォームを閉じます。これは、HTML 標準に準拠していることを確認するためです 。

フィールドフォーマットのチェック

August 15, 2023

フィールド形式チェックは、ユーザーが Web フォームで Web サイトに送信するデータを検証します。データの長さと種類の両方を調べて、データが表示されるフォームフィールドに適していることを確認します。Web App Firewall は、ユーザーリクエストで不適切な Web フォームデータを検出すると、リクエストをブロックします。

フィールドフォーマットチェックは、攻撃者が不適切な Web フォームデータを Web サイトに送信するのを防ぐことで、Web サイトやデータベースサーバーに対する特定の種類の攻撃を防ぎます。たとえば、特定のフィールドでユーザーが電話番号を入力する必要がある場合、フィールド形式チェックはユーザーが送信した入力を調べて、データが電話番号の形式と一致していることを確認します。特定のフィールドにファーストネームが必要な場合、フィールドフォーマットチェックにより、そのフィールド内のデータがファーストネームに適したタイプと長さであることを確認します。保護するように設定した各フォームフィールドに対して同じ処理を行います。

このチェックは HTML リクエストにのみ適用されます。XML 要求には適用されません。HTML プロファイルまたは Web 2.0 プロファイルでフィールド形式チェックを設定して、HTML ペイロードを検査してアプリケーションを保護できます。Web App Firewall は、Google Web Toolkit（GWT）アプリケーションのフィールドフォーマットチェック保護もサポートしています。

フィールドフォーマットチェックでは、1 つ以上のアクションを有効にする必要があります。Web App Firewall は送信された入力を調べ、指定されたアクションを適用します。

> 注
>
> フィールド形式のルールはルールを厳しくしています。学習したデータからそれらをリラクゼーションリストに追加すると、ブロッキングルールとして機能します。
>
> フィールドフォーマットのルールを緩和するには、特定の「フィールド名」をフィールドフォーマット緩和リストから削除してください。

デフォルトのフィールド形式を設定して、保護する各ウェブフォームの各フォームフィールドに必要なフィールドタイプとデータの最小長と最大長を指定できます。緩和ルールを適用して、特定のフォームの個々のフィールドのフィールド形式を設定できます。複数のルールを追加して、フィールド名、アクション URL、およびフィールド形式を指定できます。フィールド形式を指定して、さまざまなフォームフィールドでさまざまなタイプの入力を受け入れます。学習機能では、緩和ルールに関する推奨事項を提示できます。

フィールドフォーマットアクション-ブロック、ログ、統計、学習アクションを有効にできます。フィールドフォーマットチェック保護を有効にするには、これらのアクションのうち少なくとも 1 つを有効にする必要があります。

- ブロック。ブロックを有効にすると、入力が指定されたフィールド形式に従わない場合にブロックアクションがトリガーされます。ターゲットフィールドにルールが設定されている場合、入力は指定されたルールと照合されます。それ以外の場合は、デフォルトのフィールド形式仕様と照合されます。フィールドタイプまたは最小/最大長の指定に不一致があると、リクエストがブロックされます。
- ログ。ログ機能を有効にすると、フィールド形式チェックによって実行されたアクションを示すログメッセージが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増えている場合は、悪意を持って攻撃を仕掛けようとしている可能性があります。
- 統計。有効にすると、統計機能が違反とログに関する統計を収集します。統計カウンタが予期せず急増した場合は、アプリケーションが攻撃を受けている可能性があります。また、設定を見直して、指定したフィールド

形式の制限が厳しすぎないかどうかを確認する必要がある場合もあります。

- 学ぶ。どのフィールドタイプや最小長と最大長の値がアプリケーションに最適かわからない場合は、学習機能を使用して、学習したデータに基づいて推奨事項を生成できます。Web App Firewall 学習エンジンはトラフィックを監視し、観測された値に基づいて推奨されるフィールド形式を提供します。パフォーマンスを損なうことなく最適な効果を得るには、学習オプションを短時間有効にしてルールの代表的なサンプルを取得し、ルールを展開して学習を無効にすることをお勧めします。
  注:Web App Firewall の学習エンジンは、名前の最初の 128 バイトしか区別できません。フォームに、最初の 128 バイトに一致する名前のフィールドが複数ある場合、学習エンジンはそれらを区別できないことがあります。同様に、展開された緩和ルールによって、このようなすべてのフィールドが誤って緩和される可能性があります。

デフォルトフィールドフォーマット-アクションを設定するだけでなく、デフォルトのフィールドフォーマットを設定して、アプリケーションのすべてのフォームフィールドに必要なデータのタイプを指定できます。フィールドタイプをフィールドフォーマットタイプとして選択できます。[最小長] パラメータと [最大長] パラメータを使用して、許容される入力の長さを指定できます。フィールドタイプの代わりに、文字マップを使用してフィールドで許可される内容を指定できます (クラスター展開を除く)。

- フィールドタイプ—フィールドタイプは、割り当てられた優先順位値を割り当てる名前付きの式です。フィールドタイプ式は入力可能な値を指定し、送信されたデータと照合して、受信した値が許容値と一致しているかどうかを判断します。フィールドタイプは優先順位の順にチェックされます。数値が小さいほど優先度が高くなります。Web App Firewall では、独自のフィールドタイプを追加して、必要な優先順位を割り当てることができます。プライオリティ値の範囲は 0 ～64000 です。設定プロセスを簡略化するために、以下の組み込みフィールドタイプが用意されています。

```
> sh appfw fieldtype
1)      Name:  integer           Regex:  "^[+-]?[0-9]+$"
                Priority:  30            Comment:  Integer
                Builtin:  IMMUTABLE
2)      Name:  alpha             Regex:  "^[a-zA-Z]+$"
                Priority:  40            Comment:  "Alpha
                   characters"
             Builtin:  IMMUTABLE
3)      Name:  alphanum          Regex:  "^[a-zA-Z0-9]+$"
                Priority:  50            Comment:  "Alpha-numeric
                    characters"
                Builtin:  IMMUTABLE
4)      Name:  nohtml            Regex:  "^[^&<>]\*$"
                Priority:  60            Comment:  "Not HTML"
                Builtin:  IMMUTABLE
5)      Name:  any               Regex:  "^.\*$"
             Priority:  70            Comment:  Anything
                Builtin:  IMMUTABLE
    Done
>
<!--NeedCopy-->
```


> 注**:** 組み込みのフィールドタイプは不変です。これらを変更したり削除したりすることはできません。追加したフィールドタイプはすべて変更可能です。これらは編集または削除できます。

アプリケーションのすべてまたはほとんどのフォームフィールドで有効な入力を識別し、無効な入力を除外できる PCRE 式がある場合は、フィールドタイプをデフォルトのフィールド形式として設定すると便利な場合があります。たとえば、申請フォームのすべての入力に数字と文字のみが含まれていることが予想される場合は、組み込みのフィールドタイプ英数字をデフォルトのフィールドタイプとして使用することをお勧めします。入力にバックスラッシュ () やセミコロンなどの英数字以外の文字が含まれていると、違反になります。また、独自にカスタマイズしたフィールドタイプを追加し、それを使用してデフォルトのフィールド形式を構成することもできます。たとえば、小文字の「x」、「y」、「z」のみを許可する英文字にする場合は、正規表現「^ [x-z] +$」でカスタマイズされたフィールドタイプを設定できます。組み込みのフィールドタイプよりも高い優先度（低い優先度番号）を割り当てて、デフォルトのフィールドタイプとして使用できます。

- 最小長—明示的に設定されていない Web フォームのフォームフィールドに割り当てられるデフォルトの最小データ長。このパラメータはデフォルトで 0 に設定されているため、ユーザーはフィールドを空白のままにしておくことができます。これより大きい値に設定すると、ユーザーはフィールドへの入力を強制されます。

  注意: 最小長値が 0 で、フィールドタイプが整数、アルファまたは英数字の場合、最小長の設定にかかわらず、入力フィールドのいずれかが空のままになるとリクエストはブロックされます。これは、これらのフィールドタイプの正規表現に + 文字、つまり 1 つ以上の文字が含まれているためです。整数とアルファ文字を区別するには、少なくとも 1 文字が必要です。

- 最大長—明示的に設定されていない Web フォームのフォームフィールドに割り当てられるデフォルトの最大データ長。このパラメータはデフォルトで 65535 に設定されています。

  > 注**:** 文字とバイトの比較。フィールドフォーマットの最小長と最大長は、文字数ではなくバイト数を表します。1 バイトを超える文字表現を使用する言語では、最大値に設定されている数よりも少ない文字数で制限を超えることがあります。たとえば、2 バイト文字表現の場合、最大値の 9 では 4 文字までしか使用できません。
  > ヒント**:** GUI では、UTF-8 文字を 16 進数に変換せずに直接 GUI にカットアンドペーストできます。

- 文字マップ:Web App Firewall の学習エンジンでは、フィールドタイプを推奨するほかに、「文字マップを使用」というオプションを追加して、フォーマットチェックルールをデプロイできます。文字コード表は、特定のフォームフィールドで使用できるすべての文字のセットです。文字マップを使用して、特定の文字を許可または禁止するようにフィールド形式の仕様を微調整できます。フォームフィールドごとに個別の文字コードマップが生成されます。文字マップでは、英文字と数字の扱いが異なります。入力にアルファ文字が含まれている場合は、文字コード表の推奨 PCRE 式ですべてのアルファ文字 [a-za-z] を使用できます。同様に、いずれかの数字が含まれている場合は、[0-9] のすべての数字を使用できます。印刷できない文字は、x 構造体を使用して指定します。文字コード表の推奨対象となるのは、0 ～255 の値の 1 バイト文字だけです。

  文字コード表は、対応するフィールドタイプの推奨よりも具体的であり得ます。状況によっては、入力として使用できる文字セットをより厳密に制御できるため、文字マップの方が適している場合があります。展開されたキャラクタマップは、プレフィックス「CM」で始まり、その後に数字が続く文字列として表示されます。キ

ャラクターマップの優先度は 10000 から始まります。ユーザーが追加したフィールドタイプと同様に、文字コードマップを追加、編集、削除できます。デプロイされたルールで現在使用されているキャラクタマップは変更も削除もできません。

注: キャラクタマップはクラスタ展開ではサポートされていません。

> 注
>
> 組み込みのフィールドタイプを含むフィールドフォーマットルールを追加し、フィールドタイプの代わりに文字コードを使用して保存すると、変更は保存されず、ルールはフィールドタイプで引き続き表示されます。
>
> 文字マップが組み込み型のいずれかと一致する場合、新しい文字コードマップを作成する代わりにフィールドタイプが再利用されます。

コマンドラインを使用してフィールド形式チェックを設定する

コマンドラインインターフェイスでは、add appfw fieldtype コマンドを使用して新しいフィールドタイプを追加できます。set appfw profile コマンドまたは add appfw profile コマンドのいずれかを使用して、フィールド形式チェックを設定し、実行するアクションを指定できます。unset appfw profile コマンドを使用して、設定した設定をデフォルトに戻すことができます。フィールドフォーマットルールを指定するには、bind appfw コマンドを使用して、フィールドタイプをフォームのフィールドとアクション URL にバインドし、最小長と最大長の指定を行います。

コマンドラインを使用してフィールドタイプを追加、削除、または表示するには:

add コマンドを使用して、フィールドタイプを追加します。新しいフィールドタイプを追加するときは、名前、正規表現、優先度を指定する必要があります。コメントを追加することもできます。show コマンドを使用して、設定されているフィールドタイプを表示できます。remove コマンドを使用してフィールドタイプを削除することもできます。削除するには、フィールドタイプの名前のみが必要です。

```
add [appfw] fieldType <name> <regex> <priority> [-comment <string>]
```

各項目の意味は次のとおりです:

<regex> は正規表現です

<priority> は正の整数です

例:

```
1  add fieldtype "Cust_Zipcode" "^[0-9]{
2   5 }
3   [-][0-9]{
4   4 }
5   $" 4
6
7  -  show [appfw] fieldType [<name>]
8
9      Example: sh fieldtype
```



```
10
11      sh appfw fieldtype
12
13      sh appfw fieldtype cust_zipcode
14
15  -   `rm [appfw] fieldType <name>`
16
17      Example: rm fieldtype cusT_ziPcode
18
19      `rm appfw fieldtype cusT_ziPcode`
20  <!--NeedCopy-->
```

> **注:** 上に示したように、コマンドでの「appfw」の使用はオプションです。たとえば、「フィールドタイプを追加」または「appfw フィールドタイプを追加」はどちらも有効なオプションです。フィールドタイプの名前は、正規化により大文字と小文字が区別されません。上記の例に示されているように、Cust_Zipcode、cust_Zipcode、Cust_ZipCode は同じフィールドタイプを参照しています。

コマンドラインを使用してフィールドフォーマットチェックを設定するには

次のように set appfw profile コマンドまたは add appfw profile コマンドのいずれかを使用します。

- `set appfw profile <name> -fieldFormatAction (([block] [learn] [log] [stats])| [none])`
- `set appfw profile <name>-defaultFieldFormatType <string>`
- `set appfw profile <name> -defaultFieldFormatMinLength <integer>`
- `set appfw profile <name> -defaultFieldFormatMaxLength <integer>`

コマンドラインを使用してフィールド形式緩和ルールを設定するには

```
1  bind appfw profile <name> (-fieldFormat <string> <formActionURL>  <
       fieldType>
2  [-fieldFormatMinLength <positive_integer>]  [-fieldFormatMaxLength <
       positive_integer>]
3  [-isRegex ( REGEX | NOTREGEX )])
4  <!--NeedCopy-->
```

例:

```
1  bind appfw profile pr_ffc -fieldFormat "login_name" ".*/login.php"
       integer -fieldformatMinLength 3 -FieldformatMaxlength 6
2  <!--NeedCopy-->
```

GUI によるフィールドフォーマットのセキュリティチェックの設定

GUI では、フィールドタイプを管理できます。アプリケーションに関連付けられているプロファイルのペインで、フィールド形式のセキュリティチェックを設定することもできます。

GUI を使用してフィールドタイプを追加、変更、または削除するには

1. 「アプリケーションファイアウォール」ノードに移動します。「設定」で、「フィールドタイプの管理」をクリックして、「アプリケーションファイアウォールのフィールドタイプの設定」ダイアログボックスを表示します。
2. [追加] をクリックして、新しいフィールドタイプを追加します。このペインの指示に従い、「作成」をクリックします。ユーザーが追加したフィールドタイプがデプロイされたルールで現在使用されていない場合は、編集または削除することもできます。

GUI を使用してフィールドフォーマットのセキュリティチェックを追加または変更するには

1. [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ペインで、[セキュリティチェック] をクリックします。

   セキュリティー検査テーブルには、すべてのセキュリティー検査に対して現在構成されているアクション設定が表示されます。設定には次の 2 つのオプションがあります。

   a) フィールドフォーマットのブロック、ログ、統計、学習アクションを有効または無効にするだけの場合は、表のチェックボックスをオンまたはオフにして、「**OK**」をクリックし、「保存して閉じる」をクリックしてセキュリティチェックペインを閉じます。
   b) このセキュリティチェックに追加のオプションを設定する場合は、「フィールドフォーマット」をダブルクリックするか、行を選択して「アクション設定」をクリックすると、デフォルトフィールドフォーマットの次のオプションが表示されます。

      - フィールドタイプ-デフォルトのフィールドタイプとして設定するフィールドタイプを選択します。組み込みのフィールドタイプとユーザー定義のフィールドタイプを選択できます。展開されたキャラクターマップもリストに含まれており、選択できます。
      - 最小文字数-各フィールドに入力する必要がある最小文字数を指定します。設定可能な値は 0 ～ 65535 です。
      - 最大長—各フィールドに入力する必要がある最大文字数を指定します。設定可能な値は 1 ～65535 です。
        フィールドフォーマット設定ペインで、ブロック、ログ、統計、学習アクションを編集することもできます。

   上記の変更を行った後、「**OK**」をクリックして変更を保存し、「Security Checks」テーブルに戻ります。必要に応じて、他のセキュリティー検査の設定に進むことができます。[**OK**] をクリックして［セキュリティチェック］セクションで行った変更をすべて保存し、［保存して閉じる］をクリックしてセキュリティチェックウィンドウを閉じます。

GUI を使用してフィールドフォーマット緩和ルールを設定するには

1. [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ペインで、[緩和規則] をクリックします。緩和ルールテーブルには「フィールドフォーマット」エントリがあります。ダブルクリックするか、この行を選択して「編集」ボタンをクリックすると、フィールド

フォーマット緩和ルールダイアログにアクセスできます。緩和ルールの [追加]、[編集]、[削除]、[有効化]、または [無効化] 操作を実行できます。

すべての緩和ルールをまとめて表示するには、「フィールドフォーマット」行を強調表示して「ビジュアライザー」をクリックします。デプロイされたリラクゼーションのビジュアライザーには、[新しいルールを追加] または [既存のルールを編集] のオプションがあります。ノードを選択し、緩和ビジュアライザの対応するボタンをクリックして、ルールのグループ を有効 または 無効に することもできます。

フィールドフォーマットチェックでの学習機能の使用

学習アクションが有効になると、Web App Firewall 学習エンジンはトラフィックを監視し、トリガーされた違反を学習します。学習したルールは定期的に検査できます。十分に検討したら、学習したルールをフィールド形式緩和ルールとして展開できます。

フィールド形式学習の強化—Web App Firewall の学習拡張機能がリリース 11.0 で導入されました。以前のリリースでは、学習したフィールド形式のレコメンデーションがデプロイされると、Web App Firewall 学習エンジンは、新しいデータポイントに基づいて新しいルールを推奨する目的で有効なリクエストの監視を停止していました。これにより、セキュリティチェックによって処理された有効なリクエストに含まれる新しいデータが学習データベースに含まれないため、設定されているセキュリティ保護が制限されます。

違反はもはや学習と結びついていません。学習エンジンは、違反に関係なく、フィールドフォーマットを学習して推奨します。学習エンジンは、ブロックされたリクエストをチェックして、現在のフィールド形式が制限が厳しすぎて緩和する必要があるかどうかを判断するだけでなく、許可されたリクエストを監視して現在のフィールド形式が許容範囲を超えているかどうかを判断し、より制限の厳しいルールを適用することでセキュリティを強化できます。

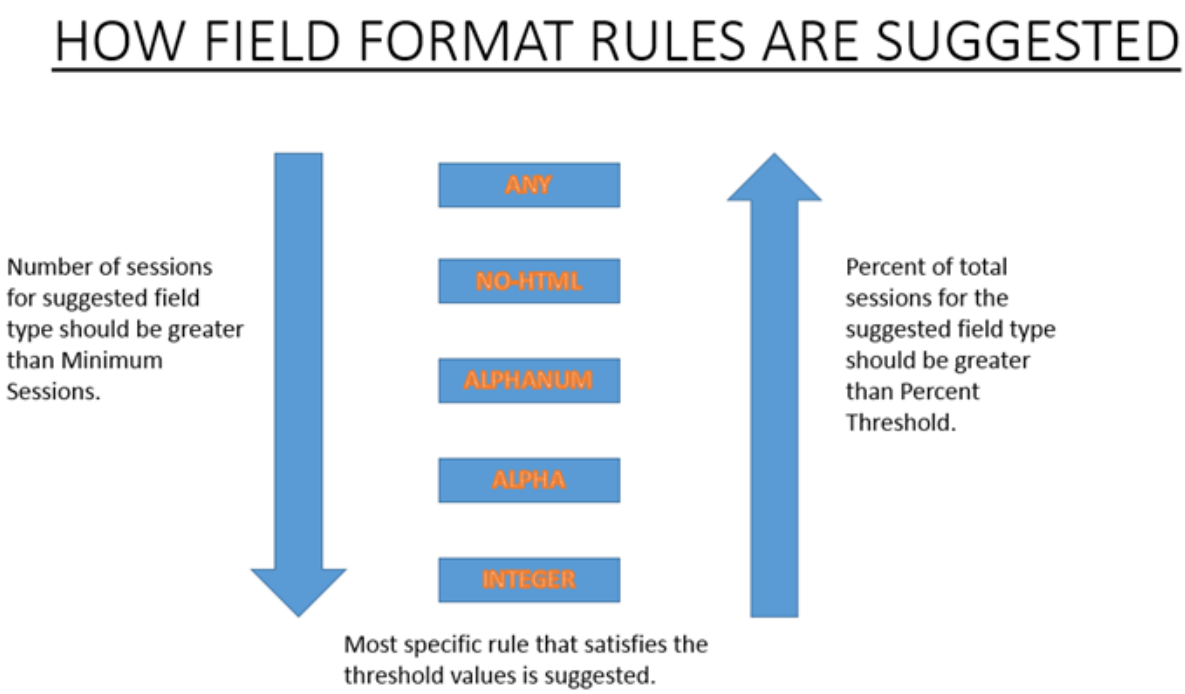
以下は、フィールドフォーマットの学習行動の概要です。

フィールド形式はバインドされていません—このシナリオでも動作は変わりません。すべての学習データは aslearn エンジンに送信されます。学習エンジンは、データセットに基づいてフィールド形式のルールを提案します。

フィールド形式は制限されています。以前のリリースでは、監視されたデータは違反の場合にのみ aslearn エンジンに送信されていました。学習エンジンは、データセットに基づいてフィールド形式のルールを提案します。11.0 リリースでは、違反が発生しなくても、すべてのデータが aslearn エンジンに送信されます。学習エンジンは、受信したすべての入力のデータセット全体に基づいてフィールド形式のルールを提案します。

学習強化のユースケース**:**

最初に学習したフィールドフォーマットのルールが少数のデータサンプルに基づいている場合、標準的でない値がいくつかあると、ターゲットフィールドに対して寛大すぎる推奨結果になる可能性があります。継続的な学習により、Web App Firewall はすべてのリクエストのデータポイントを観察して、学習した推奨事項の代表的なサンプルを収集できます。これは、適切な範囲値を持つ最適な入力形式を導入するためのセキュリティをさらに強化するのに役立ちます。



フィールド形式の学習では、フィールドタイプの優先度と、次の学習しきい値の設定を利用します。

- **FieldFormatMinThreshold**—学習済みリラクゼーションが生成されるまでに特定のフォームフィールドを観察する必要がある最小回数。デフォルト:1
- **FieldFormatPercentThreshold**—学習済み緩和が生成されるまでに、フォームフィールドが特定のフィールドタイプと一致した回数の割合。デフォルト:0。

フィールド形式ルールの推奨事項は、次の基準に基づいています。

- 推奨フィールドタイプ—推奨フィールドタイプは、既存のフィールドタイプに割り当てられた優先順位と指定されたフィールドフォーマットのしきい値によって決まります。優先順位によって、フィールドタイプが入力と照合される順序が決まります。数字が小さいほど、プライオリティが高くなります。たとえば、フィールドタイプ整数の方が優先度が高く (30)、フィールドタイプアルファヌム (50) よりも先に評価されます。しきい値によって、データポイントの代表的なサンプルを収集するために評価される入力の数が決まります。設定したフィールドタイプに適切な優先順位を割り当て、**FieldFormatPercentThreshold** パラメーターと **FieldFormatMinThreshold**** パラメーターに適切な学習設定値を設定することが、正しいフィールドフォーマットの推奨値を得るために不可欠です ** 。設定されたしきい値に基づいて、優先順位が最も高いフィールドタイプが最初に入力と照合されます。一致するものがあれば、他のフィールドタイプを考慮せずにこのフィールドタイプが候補として表示されます。たとえば、すべての入力に数値のみが含まれている場合は、整数、英字、および任意の 3 つのデフォルトフィールドタイプが一致します。ただし、整数の方が優先度が高いので推奨されます。
- 推奨される最小長と最大長—フィールドフォーマットの最小長と最大長の計算は、フィールドタイプの決定とは独立して行われます。フィールドフォーマットの長さの計算は、観測されたすべての入力の平均長に基づいています。この計算された平均の半分が最小値として推奨され、この平均値の 2 倍の値が最大値として推奨さ

れます。[最小長] の範囲は 0 ～65535 で、[最大長] の範囲は 1 ～65535 です。最小長の設定値は最大長を超えることはできません。

- スペース文字の処理—フィールドフォーマットチェックでは、フィールドフォーマットの長さをチェックするときに、すべてのスペース文字がカウントされます。入力処理中に、先頭または末尾のスペースは削除されず、入力文字列の途中にある複数の連続したスペースが 1 つのスペースに統合されなくなりました。

フィールドフォーマットの推奨事項を説明する例:

```
Total requests: 100
Number of Req with Field Type:
Int : 22             (22 int values)  -  22%
Alpha : 44                    (44 alpha values)  -  44%
Alphanum: 14             (14 + 44 + 22 = 80 alphanum values) = 80%
noHTML: 10               (80 + 10 = 90 noHTML values) = 90%
any : 10                      (90 + 10 = 100 any values) = 100%

% threshold                                  Suggested Field Type
0-22                 int
23-44                alpha
45-80                alphanum
81-90                noHTML
91-100               any
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して学習データを表示または使用するには

```
show appfw learningdata <profilename> FieldFormat
rm appfw learningdata <profilename> -fieldFormat <string>  <
   formActionURL>
export appfw learningdata <profilename> FieldFormat
<!--NeedCopy-->
```

GUI を使用して学習済みデータを表示または使用するには

1. [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ペインで、[学習済みルール] をクリックします。学習ルールテーブルの「フィールドフォーマット」エントリを選択してダブルクリックすると、学習したルールにアクセスできます。学習したルールを展開したり、緩和ルールとして展開する前にルールを編集したりできます。ルールを破棄するには、ルールを選択して「スキップ」(**Skip**) ボタンをクリックします。一度に編集できるルールは 1 つだけですが、展開またはスキップするルールは複数選択できます。

   また、「学習ルール」テーブルの「フィールドフォーマット」エントリを選択し、「ビジュアライザー」をクリックすると、学習したすべての違反をまとめて表示できます。ビジュアライザーを使用すると、学習したルールを非常に簡単に管理できます。1 つの画面でデータの包括的なビューを表示し、1 回のクリックでルールのグループに対するアクションの実行を容易にします。ビジュアライザーの最大の利点は、正規表現を推奨して複数のルールを統合できることです。デリミタとアクション URL に基づいて、これらのルールのサブセットを選択できます。ドロップダウンリストから番号を選択すると、ビジュアライザーに 25、50、または 75 個の

ルールを表示できます。学習したルールのビジュアライザーには、ルールを編集して緩和として展開するオプションがあります。または、ルールをスキップして無視することもできます。

フィールドフォーマットチェックでのログ機能の使用

ログアクションを有効にすると、フィールドフォーマットのセキュリティチェック違反が APPFW_FIELDFORMAT 違反として監査ログに記録されます。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて、/var/log/ フォルダー内の ns.logs を末尾に移動して、フィールドフォーマット違反に関するログメッセージにアクセスします。

- Shell
- `tail -f /var/log/ns.log | grep APPFW_FIELDFORMAT`

GUI を使用してログメッセージにアクセスするには

GUI には、ログメッセージを分析するための非常に便利なツール (Syslog Viewer) が含まれています。Syslog ビューアにアクセスするには、複数のオプションがあります。

- [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを選択して [セキュリティチェック] をクリックします。[フィールドフォーマット] 行を強調表示して、[ログ] をクリックします。プロファイルの **Field Formats** セキュリティチェックから直接ログにアクセスすると、ログメッセージが除外され、これらのセキュリティチェック違反に関連するログのみが表示されます。
- 「**NetScaler** 」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。監査メッセージセクションで、**Syslog** メッセージリンクをクリックして **Syslog Viewer** を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティチェック違反がトリガーされる可能性がある場合のデバッグに役立ちます。
- アプリケーションファイアウォール > ポリシー > 監査に移動します。監査メッセージセクションで、Syslog メッセージリンクをクリックして Syslog Viewer を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

  HTML ベースの Syslog ビューアには、関心のあるログメッセージのみを選択するためのさまざまなフィルタオプションがあります。フィールドフォーマットのセキュリティチェック違反ログメッセージにアクセスするには、モジュールのドロップダウンオプションで APPFW を選択してフィルタリングします。[イベントタイプ] には、選択をさらに絞り込むための豊富なオプションセットが表示されます。たとえば、**APPFW_FIELDFORMAT** チェックボックスを選択して「適用」ボタンをクリックすると、フィールドフォーマットのセキュリティチェック違反に関するログメッセージのみが Syslog Viewer に表示されます。

特定のログメッセージの行にカーソルを置くと、Module や EventType などの複数のオプションがログメッセージの下に表示されます。これらのオプションのいずれかを選択して、ログ内の対応する情報を強調表示できます。

リクエストがブロックされていない場合のネイティブ形式のログメッセージの例

```
Jun 10 22:32:26 <local0.info> 10.217.31.98 06/10/2015:22:32:26 GMT ns
    0-PPE-0 :
default APPFW APPFW_FIELDFORMAT 97 0 :  10.217.253.62 562-PPE0
x1MV+YnNGzQFM3Bsy2wti4bhXio0001 pr_ffc http://aaron.stratum8.net/FFC/
    login_post.php
Field format check failed for field passwd="65568888sz-*_" <not blocked
    >
Example of a CEF format log message when the request is blocked
Jun 11 00:03:51 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_FIELDFORMAT|6|src
    =10.217.253.62 spt=27076
method=POST requet=http://aaron.stratum8.net/FFC/maxlen_post.php msg=
    Field format check
failed for field text_area="" cn1=108 cn2=644 cs1=pr_ffc cs2=PPE0
cs3=GaUROfl1Nx1jJTvja5twH5BBqI0000 cs4=ALERT cs5=2015 act=blocked
<!--NeedCopy-->
```

フィールドフォーマット違反の統計

統計アクションを有効にすると、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、フィールドフォーマットチェックに対応するカウンタが増えます。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンターのインクリメントは、構成された設定によって異なる場合があります。たとえば、ブロックアクションが有効な場合、フィールドフォーマット違反が 3 つあるページのリクエストでは、最初のフィールドフォーマット違反が検出されるとすぐにページがブロックされるため、統計カウンタが 1 つ増えます。ただし、ブロックが無効になっている場合、同じリクエストを処理すると、フィールドフォーマット違反ごとに個別のログメッセージが生成されるため、違反とログの統計カウンタが 3 ずつ増えます。

コマンドラインを使用してフィールドフォーマットの統計情報を表示するには

コマンドプロンプトで入力します。

```
sh appfw stats
```

特定のプロファイルの統計情報を表示するには、以下のコマンドを使用します。

```
stat appfw profile <profile name>
```

GUI を使用してフィールドフォーマットの統計情報を表示するには

1. システム > セキュリティ > アプリケーションファイアウォールに移動します。
2. 右側のペインで、[統計 リンク] にアクセスします。
3. スクロールバーを使用して、フィールドフォーマットの違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

導入のヒント

- フィールド形式のアクションログ、学習、統計を有効にします。
- アプリケーションへのトラフィックの代表的なサンプルを実行したら、学習した推奨事項を確認します。
- 学習したルールのほとんどでフィールドタイプが推奨される場合は、そのフィールドタイプをデフォルトのフィールドタイプとして設定します。最小長と最大長については、これらのルールで推奨されている最も広い範囲を使用してください。
- 異なるフィールドタイプや異なる最小/最大長の方が適している他のフィールドにもルールをデプロイします。
- ブロッキングを有効にし、学習を無効にします。
- 統計とログを監視します。それでもかなりの数の違反が発生している場合は、ログメッセージを確認して、その違反がブロックされたはずの悪意のあるリクエストであるかどうかを確認することをお勧めします。有効なリクエストが違反としてフラグ付けされている場合は、設定したフィールド形式ルールを編集してさらに緩和するか、学習を再度有効にして新しいデータポイントに基づいて推奨事項を取得することができます。

> 注**:** 新しい学習推奨事項を入手することで、構成を微調整できます。

ハイライト

フィールドフォーマットのセキュリティチェックについては、次の点に注意してください。

- 保護—最適なフィールド形式ルールを設定することで、多くの攻撃から保護できます。たとえば、フィールドに整数のみを指定した場合、ハッカーはこのフィールドを使用して SQL インジェクション攻撃やクロスサイトスクリプティング攻撃を仕掛けることはできません。このような攻撃を開始するのに必要な入力は、設定されたフィールド形式の要件を満たさないためです。
- パフォーマンス—フィールドフォーマットルールの入力の最小許容長と最大長を制限できます。これにより、悪意のあるユーザーがサーバーに処理オーバーヘッドを追加しようとして過度に大きな入力文字列を入力したり、さらに悪いことに、スタックオーバーフローのためにサーバーがコアをダンプしたりするのを防ぐことができます。入力サイズを制限することで、正当なリクエストの処理に必要な時間を短縮できます。
- フィールドフォーマットの設定—フィールドフォーマットの保護を行うには、いずれかのアクション (ブロック、ログ、統計、学習) を有効にする必要があります。また、フィールド形式のルールを指定して、フォームフィールドに入力できる内容を特定することもできます。
- キャラクタマップの選択 **vs.** フィールドタイプ—文字マップとフィールドタイプはどちらも正規表現を使用します。ただし、文字コード表では、使用できる文字のリストを絞り込むことで、より具体的な式が得られます。たとえば、janedoe@citrix.com のような入力の場合、学習エンジンは「フィールドタイプ (.html)」ではなく「文字マップ []」を推奨する場合があります。@-za-Z] は、使用できる非アルファ文字のセットを絞り込むため、より具体的かもしれません。文字コード表オプションでは、アルファ文字の他に、ピリオド (.) とアットマーク (@) の 2 つの非英文字のみを使用できます。
- 継続的な学習—Web App Firewall は、すべての受信データ (違反データおよび許容される入力データ) を監視および考慮して、推奨ルール用の学習テーブルを作成します。ルールは、新しい受信データが到着すると改訂および更新されます。既にバインドされたフィールド形式のルールがある場合でも、そのフィールドには新しいフィールド形式のルールが提案されます。設定したフィールドフォーマットの制限が厳しすぎて、有効な

リクエストがブロックされている場合は、より緩和されたフィールドフォーマットを導入できます。同様に、現在のフィールドフォーマットが汎用的すぎる場合は、より制限の厳しいフィールドフォーマットを導入することで、セキュリティをさらに洗練し、強化することができます。

- ルールの上書き—フィールドと URL の組み合わせに対して既にルールが適用されている場合、ユーザは GUI を使用してフィールド形式を更新できます。既存のルールを置き換えるかどうかの確認を求めるダイアログボックスが表示されます。コマンドラインインターフェイスを使用している場合は、以前のバインディングを明示的にバインド解除してから、新しいルールをバインドする必要があります。
- 複数一致—複数のフィールド形式が特定のフィールド名とそのアクション URL と一致する場合、Web App Firewall はそのうちの 1 つを任意に選択して適用します。
- バッファ境界—フィールド値が複数のストリーミングバッファにまたがっていて、フィールド値のこの 2 つの部分の形式が異なる場合、「任意」に対応するフィールド形式が学習データベースに送信されます。
- フィールドフォーマット **vs.** フィールド整合性チェック—フィールド形式チェックとフィールド整合性チェックはどちらもフォームベースの保護チェックです。フィールドフォーマットチェックは、フォームフィールドの一貫性チェックとは異なるタイプの保護を提供します。フォームフィールドの一貫性チェックでは、ユーザーから返された Web フォームの構造が損なわれていないこと、HTML に設定されているデータ形式の制限が守られていること、非表示フィールドのデータが変更されていないことを確認します。これは、Web フォーム自体から派生したもの以外の Web フォームに関する特別な知識がなくても実行できます。フィールドフォーマットチェックでは、各フォームフィールドのデータが、手動で設定した特定のフォーマット制限と一致しているか、または学習機能が生成されて承認されたかを確認します。言い換えれば、フォームフィールドの一貫性チェックは一般的な Web フォームのセキュリティを強化し、フィールドフォーマットチェックは Web フォームの許可された入力に特定のルールを適用します。

フォームフィールドの一貫性チェック

August 15, 2023

フォームフィールドの一貫性チェックは、Web サイトのユーザーから返された Web フォームを調べ、Web フォームがクライアントによって不適切に変更されていないことを確認します。このチェックは、データの有無にかかわらず、Web フォームを含む HTML リクエストにのみ適用されます。XML 要求には適用されません。

フォームフィールドの一貫性チェックは、クライアントがフォームに入力して送信するときに、Web サイト上の Web フォームの構造に不正な変更を加えることを防ぎます。また、ユーザーが送信するデータが長さと種類に関する HTML の制限を満たしていること、および非表示フィールドのデータが変更されないことも保証されます。これにより、攻撃者が Web フォームを改ざんしたり、変更されたフォームを使用して Web サイトに不正にアクセスしたり、安全でないスクリプトを使用する問い合わせフォームの出力をリダイレクトして迷惑メールを一括送信したり、Web サーバーソフトウェアの脆弱性を悪用して Web サーバーや基盤となるオペレーティングシステムを制御したりすることを防ぎます。Web フォームは、多くの Web サイトの弱点であり、さまざまな攻撃を受けやすくなります。

フォームフィールドの一貫性チェックでは、次のすべてが検証されます。

- フィールドがユーザーに送信されると、チェックによってそのフィールドがユーザーによって返されるかどうかが確認されます。
- このチェックでは、HTML フィールドの長さとタイプが強制されます。

  > 注:
  >
  > - フォームフィールドの一貫性チェックは、データ型と長さに HTML の制限を適用しますが、それ以外の場合は Web フォーム内のデータを検証しません。フィールドフォーマットチェックを使用して、Web フォームの特定のフォームフィールドに返されたデータを検証するルールを設定できます。
  > - フォームフィールドの一貫性保護は、クライアントに送信される応答フォームに隠しフィールド「as_fid」を挿入します。クライアントがフォームを送信すると、同じ隠しフィールドが ADC によって削除されます。クライアントサイドの JavaScript がフォームフィールドでチェックサム計算を行い、バックエンドで同じチェックサムを検証すると、アプリケーションが壊れる可能性があります。このシナリオでは、アプリケーションファイアウォールのフォームフィールドの一貫性隠しフィールド「as_fid」をクライアント側の JavaScript チェックサム計算から解放することをお勧めします。

- Web サーバーがユーザーにフィールドを送信しない場合、このチェックではユーザーがそのフィールドを追加してデータを返すことはできません。
- フィールドが読み取り専用または非表示のフィールドの場合、チェックはデータが変更されていないことを確認します。
- フィールドがリストボックスまたはラジオボタンフィールドの場合、チェックは応答のデータがそのフィールドの値の 1 つに対応しているかどうかを検証します。

ユーザーから返された Web フォームが 1 つ以上のフォームフィールドの一貫性チェックに違反していて、その Web フォームがフォームフィールドの一貫性チェックに違反することを許可するように Web App Firewall を構成していない場合、リクエストはブロックされます。

ウィザードまたは GUI を使用する場合、[フォームフィールドの一貫性チェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ログ]、[学習]、および [統計] の各アクションを有効または無効にできます。

また、[全般] タブでセッションレスフィールドの一貫性を設定します。セッションレスフィールドの一貫性が有効になっている場合、Web App Firewall は Web フォーム構造のみをチェックし、セッション情報の維持に依存するフォームフィールドの一貫性チェックの部分は省略します。これにより、多くのフォームを使用する Web サイトのセキュリティ上のペナルティをほとんど発生させずに、フォームフィールドの一貫性チェックを高速化できます。すべての Web フォームでセッションレスフィールドの一貫性を使用するには、「オン」を選択します。HTTP POST メソッドで送信されたフォームにのみ使用するには、[postOnly] を選択します。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力してフォームフィールドの一貫性チェックを設定できます。

- `set appfw profile <name> -fieldConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]`

フォームフィールドの一貫性チェックの緩和を指定するには、GUI を使用する必要があります。フォームフィールドの一貫性チェックを修正ダイアログボックスの「チェック」タブで、「追加」をクリックして「フォームフィールドの一貫性チェックリラクゼーションを追加」ダイアログボックスを開くか、既存のリラクゼーションを選択して「開く」をクリックして、フォームフィールドの一貫性チェックリラクゼーションの変更ダイアログボックスを開きます。どちらのダイアログボックスでも、「GUI を使用した手動設定」で説明されているように、緩和の設定に同じオプションが用意されています。

フォームフィールドの一貫性チェック緩和の例を次に示します。

フォームフィールド名:

- UserType という名前のフォームフィールドを選択してください:

```
^UserType$
<!--NeedCopy-->
```

- 名前が UserType_ で始まり、文字または数字で始まり、1 ～21 の文字、数字、またはアポストロフィまたはハイフン記号で構成されるストリングが続くフォームフィールドを選択します。

```
^UserType_[0-9A-Za-z][0-9A-Za-z'-]{
0,20 }
$
<!--NeedCopy-->
```

- 名前が Turkish-userType_ で始まり、それ以外は前の式と同じフォームフィールドを選択してください。ただし、全体にトルコ語の特殊文字を含めることができます。

```
^T\xC3\xBCrk\xC3\xA7e-UserType_([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa
   -f])+$
<!--NeedCopy-->
```

> 注:
>
> サポートされている特殊文字の完全な説明と、それらを適切にエンコードする方法については、PCRE 文字エンコーディング形式を参照してください 。

- 文字または数字で始まり、文字または数字のみの組み合わせで構成され、文字列内の任意の場所に文字列 Num を含むフォームフィールド名を選択します。

```
^[0-9A-Za-z]\*Num[0-9A-Za-z]\*$
<!--NeedCopy-->
```

フォームフィールドアクション **URL**:

- 新しいクエリを除く `http://www.example.com/search.pl?` 、クエリの後に続く任意の文字列を含む URL を選択します。


```
^http://www[.]example[.]com/search[.]pl?[^?]*$
<!--NeedCopy-->
```

- http://www.example-español.comで始まり、大文字と小文字、数字、ASCII 以外の特殊文字、およびパス内の選択した記号で構成されるパスとファイル名を含む URL を選択してください。ñ 文字やその他の特殊文字は、UTF-8 文字セットの各特殊文字に割り当てられた 16 進コードを含むエンコードされた UTF-8 文字列として表現されます。

```
^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- 「/search.cgi?」という文字列を含むすべての URL を選択してください:

```
^[^?<>]\*/search[.]cgi?[^?<>]\*$
<!--NeedCopy-->
```

> 注意:
>
> 正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加する URL を正確に定義し、それ以外は何も定義していないことを確認してください。ワイルドカード、特にドットとアスタリスク（.*）メタキャラクタとワイルドカードの組み合わせを不注意に使用すると、意図しない Web コンテンツへのアクセスをブロックしたり、Cookie の一貫性チェックでそうしない攻撃を許可したりするなど、望ましくない結果が得られる可能性があります。ブロックされています。

CSRF フォームのタグ付けチェック

August 15, 2023

クロスサイトリクエストフォージェリ (CSRF) フォームのタグ付けチェックは、保護された Web サイトからユーザーに送信された各 Web フォームに一意で予測不可能な FormID をタグ付けし、ユーザーから返された Web フォームを調べて、指定された FormID が正しいことを確認します。このチェックは、クロスサイトリクエストフォージェリ攻撃から保護します。このチェックは、データの有無にかかわらず、Web フォームを含む HTML リクエストにのみ適用されます。XML 要求には適用されません。

CSRF フォームタグチェックは、攻撃者が独自の Web フォームを使用して、保護された Web サイトにデータを含む大量のフォーム応答を送信することを防ぎます。このチェックでは、Web フォームを詳細に分析する他の特定のセキュリティチェックと比較して、CPU の処理能力は比較的少なくて済みます。そのため、保護対象の Web サイトや Web App Firewall 自体のパフォーマンスを著しく低下させることなく、大量の攻撃を処理できます。

CSRF フォームタグチェックを有効にする前に、次の点に注意する必要があります。

- フォームのタグ付けを有効にする必要があります。CSRF チェックはフォームのタグ付けに依存し、それなしでは機能しません。
- そのプロファイルで保護されているフォームを含むすべての Web ページで、NetScaler 統合キャッシュ機能を無効にする必要があります。統合キャッシュ機能と CSRF フォームのタグ付けには互換性がありません。
- リファラーチェックを有効にすることを検討する必要があります。リファラーチェックは開始 URL チェックの一部ですが、クロスサイトリクエストの偽造を防ぐのであって、開始 URL の違反を防ぐことはできません。また、リファラーチェックは CSRF フォームのタグ付けチェックよりも CPU への負荷が少なくなります。リクエストがリファラーチェックに違反した場合、そのリクエストはすぐにブロックされるため、CSRF フォームタグチェックは呼び出されません。
- CSRF フォームのタグ付けチェックは、フォームオリジン URL とフォームアクション URL で異なるドメインを使用する Web フォームでは機能しません。たとえば、example.com と example.org は異なるドメインであるため、CSRF フォームのタグ付けでは、フォームオリジン URL が`http://www.example.com`でフォームアクション URL が`http://www.example.org/form`.pl の Web フォームを保護できません。

ウィザードまたは GUI を使用する場合は、「CSRF フォームのタグ付けチェックの変更」ダイアログボックスの「一般」タブで、ブロック、ログ、学習、および統計アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して CSRF フォームタグチェックを設定できます。

- `set appfw profile <name> -CSRFtagAction [**block**] [**log**] [**learn**] [**stats**] [**none**]`

CSRF フォームタグチェックの緩和を指定するには、GUI を使用する必要があります。「CSRF フォームのタグ付けチェックの変更」ダイアログ・ボックスの「チェック」タブで、「追加」をクリックして「CSRF フォームのタグ付けチェックの緩和」ダイアログ・ボックスを開くか、既存のリラクゼーションを選択して「開く」をクリックして「CSRF フォームのタグ付けチェックの変更」ダイアログ・ボックスを開きます。どちらのダイアログボックスにも、リラクゼーションを構成するための同じオプションが表示されます。

NetScaler Web App Firewall のセッション制限を 0 以下に設定するとアラートが生成されます。このような設定は、Web App Firewall セッションが正常に機能することを必要とする高度な保護チェック機能に影響するためです。

CSRF フォームのタグ付けチェック緩和の例を以下に示します。

注: 次の式は、フォームオリジン URL ロールとフォームアクション URL ロールの両方で使用できる URL 表現です。

- 新しいクエリを除いて、`http://www.example.com/search.pl?` クエリの後に任意の文字列で始まり、その文字列を含む URL を選択します。

```
1  ^http://www[.]example[.]com/search[.]pl?[^?]*$
2  <!--NeedCopy-->
```


- http://www.example-español.comで始まり、大文字と小文字、数字、ASCII 以外の特殊文字、およびパス内の選択した記号で構成されるパスとファイル名を含む URL を選択してください。ñ 文字やその他の特殊文字は、UTF-8 文字セットの各特殊文字に割り当てられた 16 進コードを含むエンコードされた UTF-8 文字列として表現されます。

```
1     ^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\x[0-9A-Fa
         -f][0-9A-Fa-f])
2 ([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*([0-9A-Za-z]|\x[0-9A-Fa-f
     ][0-9A-Fa-f])([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|
     php|s?html?)$
3 <!--NeedCopy-->
```

- 「/search.cgi?」という文字列を含むすべての URL を選択してください:

```
1     ^[^?<>]\*/search[.]cgi?[^?<>]\*$
2 <!--NeedCopy-->
```

> 重要
>
> 正規表現は強力です。PCRE 形式の正規表現に詳しくない場合は、作成した正規表現を再確認してください。例外として追加する URL を正確に定義していることを確認し、それ以外は何も定義しないでください。ワイルドカード、特にドットとアスタリスク (.*) メタ文字とワイルドカードの組み合わせを不注意に使用すると、ブロックするつもりのない Web コンテンツへのアクセスをブロックしたり、チェックでブロックされたはずの攻撃を許可したりするなど、望ましくない結果が生じる可能性があります。
>
> ヒント
>
> URL アクションの「開始 URL」で「EnableValidate」リファラーヘッダーが有効になっている場合は、リファラーヘッダーの URL も必ず StartURL に追加してください。
>
> 注
>
> NetScaler が appfw_session_limit に達し、CSRF チェックが有効になると、ウェブアプリケーションがフリーズします。
>
> Web アプリケーションがフリーズしないようにするには、次のコマンドを使用してセッションタイムアウトを減らし、セッション制限を増やします。
>
> From CLI: > set appfw settings -sessiontimeout 300
> From shell: root@ns# nsapimgr_wr.sh -s appfw_session_limit=200000
>
> appfw_session_limit に達したときに SNMP アラームをログに記録して生成すると、問題のトラブルシューティングとデバッグに役立ちます。

CSRF フォームのタグ付けチェック緩和の管理

August 15, 2023

CSRF フォームタグセキュリティチェックの例外 (または緩和) は、「クロスサイトリクエスト偽造タグチェックの緩和」ダイアログまたは「クロスサイトリクエスト偽造タグチェックの緩和」の変更ダイアログボックスで設定します。

GUI を使用して **CSRF** フォームのタグ付けチェック緩和を設定するには

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 「プロファイル」ペインで、構成するプロファイルを選択し、「開く」をクリックします。
3. 「**Web App Firewall** プロファイルの設定」ダイアログボックスで、「セキュリティチェック」タブをクリックします。「セキュリティチェック」タブには、Web App Firewall のセキュリティチェックのリストが含まれています。
4. CSRF 緩和を追加または変更するには、次のいずれかを実行します。
   - 新しいリラクゼーションを追加するには、「追加」をクリックします。
   - 既存のリラクゼーションを変更するには、変更するリラクゼーションを選択し、「開く」をクリックします。

   「クロスサイトリクエストフォージェリタギングチェック緩和の追加」または「クロスサイトリクエストフォージェリタギングチェック緩和の変更」ダイアログ・ボックスが表示されます。タイトル以外は、これらのダイアログボックスは同じです。
5. 以下の説明に従ってダイアログボックスに入力します。
   - 「有効」チェックボックス-選択するとこの緩和またはルールをアクティブに使用し、オフにすると非アクティブになります。
   - フォームオリジン **URL**—テキスト領域に、フォームをホストする URL を定義する PCRE 形式の正規表現を入力します。
   - フォームアクション **URL**—テキスト領域に、フォームに入力されたデータの配信先の URL を定義する PCRE 形式の正規表現を入力します。
   - コメント—テキスト領域にコメントを入力します。オプションです。

   > 注:
   >
   > 正規表現を必要とする要素については、正規表現を入力するか、「**Regex Tokens** 」メニューを使用し

> て正規表現の要素と記号をテキストボックスに直接挿入するか、「**Regex Editor** 」をクリックして「正規表現の追加」ダイアログボックスを開き、それを使用して正規表現を作成します。

6. [**OK**] をクリックします。「クロスサイトリクエストフォージェリタグチェックの追加」「緩和」または「クロスサイトリクエストのフォージェリタグチェックの緩和」ダイアログ・ボックスが閉じ、「クロスサイト・リクエスト・フォージェリー・タグ付け・チェックの変更」ダイアログ・ボックスに戻ります。
7. リラクゼーションまたはルールを削除するには、そのリラクゼーションまたはルールを選択し、[削除] をクリックします。
8. リラクゼーションまたはルールを有効にするには、そのリラクゼーションまたはルールを選択し、「有効にする」をクリックします。
9. リラクゼーションまたはルールを無効にするには、そのリラクゼーションまたはルールを選択し、「無効」をクリックします。
10. 統合された対話型グラフィック表示で、既存のすべての緩和の設定と関係を構成するには、[ビジュアライザー] をクリックし、表示ツールを使用します。
11. CSRF チェックの学習ルールを確認して設定するには、[学習] をクリックし、学習機能を設定して使用するにはの手順を実行します。
12. [**OK**] をクリックします。

URL 保護チェック

August 15, 2023

URL 保護チェックはリクエスト URL を調べて、攻撃者が複数の URL に積極的にアクセスしようとしたり (強制ブラウジング)、URL を使用して Web サーバーソフトウェアや Web サイトスクリプトの既知のセキュリティ脆弱性を引き起こしたりすることを防ぎます。

URL チェックを開始

August 15, 2023

開始 URL チェックは、受信リクエストの URL を調べ、URL が指定された基準を満たさない場合は接続試行をブロックします。条件を満たすには、「URL 閉鎖の強制」パラメーターが有効になっていない限り、URL が「開始 URL」リストのエントリと一致する必要があります。このパラメーターを有効にすると、ウェブサイト上のリンクをクリックしたユーザーは、そのリンクのターゲットに接続されます。

開始 URL チェックの主な目的は、ブックマークや外部リンクを介してウェブサイト上のランダムな URL に繰り返しアクセスしようとしたり（強制的にブラウジング）したり、URL を手動で入力してウェブサイトのその部分にアクセスするのに必要なページをスキップしてページにジャンプしたりすることを防ぐことです。強制ブラウジングを使用すると、バッファオーバーフローを引き起こしたり、ユーザーが直接アクセスすることを意図していないコンテンツを見つけたり、Web サーバーの安全な領域へのバックドアを見つけたりすることができます。Web App Firewall は、開始 URL として設定されている URL のみへのアクセスを許可することで、Web サイトに指定されたトラバーサルまたはロジックパスを強制します。

ウィザードまたは GUI を使用する場合、「開始 URL チェックの変更」ダイアログボックスの「一般」タブで、ブロック、ログ、統計、学習アクション、および次のパラメータを有効または無効にできます。

- **URL** の閉鎖を強制します。Web サイトの他のページにあるハイパーリンクをクリックして、Web サイト上の任意の Web ページにユーザーがアクセスできるようにします。ユーザーは、ハイパーリンクをクリックして、ホームページまたは指定されたスタートページからアクセスできる Web サイトの任意のページに移動できます。
  注意:URL クロージャ機能を使用すると、HTTP GET メソッドを使用して送信された Web フォームのアクション URL に任意のクエリ文字列を追加して送信できます。保護されている Web サイトがフォームを使用して SQL データベースにアクセスする場合は、SQL インジェクションチェックが有効になっていて、正しく設定されていることを確認してください。
- セッションレス **URL** の閉鎖。クライアントの観点から見ると、このタイプの URL クロージャは標準のセッション対応 URL クロージャとまったく同じように機能しますが、Cookie の代わりに URL に埋め込まれたトークンを使用してユーザーのアクティビティを追跡するため、使用するリソースが大幅に少なくなります。セッションレス URL クロージャが有効になっている場合、Web App Firewall は URL クロージャにあるすべての URL に「as_url_id」タグを追加します。
  注: セッションレス（セッションレス URL クロージャー）を有効にする場合は、通常の URL クロージャー（強制的な URL クロージャー）も有効にする必要があります。そうしないと、セッションレス URL クロージャが機能しません。
- リファラーヘッダーを検証します。リクエストの Referer ヘッダーに、別の Web サイトではなく、保護されている Web サイトの Web フォームデータが含まれていることを確認します。このアクションは、外部の攻撃者ではなく、自分の Web サイトが Web フォームの送信元であることを確認します。これにより、ヘッダーチェックよりも CPU 負荷の高いフォームのタグ付けを必要とせずに、クロスサイトリクエストフォージェリ (CSRF) を防ぐことができます。Web App Firewall は、ドロップダウンリストで選択したオプションに応じて、次の 4 つの方法のいずれかで HTTP Referer ヘッダーを処理できます。
  - オフ-リファラーヘッダーを検証しません。
  - **If-Present** —リファラーヘッダーが存在する場合は、リファラーヘッダーを検証します。無効な Referer ヘッダーが見つかった場合、リクエストはリファラーヘッダー違反を生成します。Referer ヘッダーが存在しない場合、リクエストによってリファラーヘッダー違反は発生しません。このオプションにより、Web App Firewall はリファラーヘッダーを含むリクエストに対してリファラーヘッダー検証を実行できますが、ブラウザーがリファラーヘッダーを設定していないユーザーや、そのヘッダーを削除するウェブプロキシやフィルターを使用するユーザーからのリクエストはブロックされません。

- 常に開始 **URL** を除く**-Referer** ヘッダーを常に検証します。Referer ヘッダーがなく、要求された URL が StartURL 緩和ルールで除外されていない場合、リクエストはリファラーヘッダー違反を生成します。Referer ヘッダーは存在するが無効な場合、リクエストはリファラーヘッダー違反を生成します。
- 最初のリクエスト以外は常に—リファラーヘッダーを常に検証します。リファラーヘッダーがない場合は、最初にアクセスされた URL のみが許可されます。他のすべての URL は、有効なリファラーヘッダーがないとブロックされます。Referer ヘッダーは存在するが無効な場合、リクエストはリファラーヘッダー違反を生成します。

開始 URL の 1 つである「セキュリティチェックからクロージャー **URL** を除外」設定は、「開始 URL チェックの変更」ダイアログボックスでは設定されませんが、プロファイルの「設定」タブで設定されます。この設定を有効にすると、Web App Firewall が URL 閉鎖基準を満たす URL に対してフォームベースのチェック（クロスサイトスクリプティングや SQL インジェクション検査など）をこれ以上実行しないように指示します。

> 注
>
> リファラーヘッダーチェックと開始 URL セキュリティチェックは同じアクション設定を共有していますが、開始 URL チェックに違反せずにリファラーヘッダーチェックに違反することは可能です。違いはログに表示され、リファラーヘッダーチェック違反は開始 URL チェック違反とは別にログに記録されます。

リファラーヘッダーの設定 (OFF、If-Present、AlwaysExceptStarUrls、AlwaysExceptFirstRequest) は、制限の少ないものから最も制限の厳しいものの順に並べられており、次のように機能します。

OFF:

- リファラーヘッダーがチェックされていません。

存在する場合**:**

- リクエストにはリファラーヘッダーがありません-> リクエストは許可されています。
- リクエストにはリファラーヘッダーがあり、リファラー URL は URL クロージャーにあります-> リクエストは許可されています。
- リクエストにはリファラーヘッダーがあり、リファラー URL は URL クロージャーに含まれていません -> リクエストはブロックされています。

スタート **URL** 以外は必ず**:**

- リクエストにはリファラーヘッダーがなく、リクエスト URL は開始 URL です。-> リクエストは許可されます。
- リクエストにはリファラーヘッダーがなく、リクエスト URL は開始 URL ではありません。-> リクエストはブロックされています。
- リクエストにはリファラーヘッダーがあり、リファラー URL は URL クロージャーにあります-> リクエストは許可されています。
- リクエストにはリファラーヘッダーがあり、リファラー URL は URL クロージャーに含まれていません -> リクエストはブロックされています。

最初のリクエスト以外は必ず**:**

- リクエストにはリファラーヘッダーがなく、セッションの最初のリクエスト URL です。-> リクエストは許可されます。
- リクエストにはリファラーヘッダーがなく、セッションの最初のリクエスト **URL** でもありません。-> リクエストはブロックされました。
- リクエストにはリファラーヘッダーがあり、セッションの最初のリクエスト URL か、URL クロージャーのどちらかです。-> リクエストは許可されます。
- リクエストにはリファラーヘッダーがあり、セッションの最初のリクエスト URL でも URL クロージャーでもありません-> リクエストはブロックされました。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して開始 URL チェックを設定できます。

- `set appfw profile <name> -startURLAction [block] [learn] [log] [stats] [none]`
- `set appfw profile <name> -startURLClosure ([ON] | [OFF])`
- `set appfw profile <name> -sessionlessURLClosure ([ON] | [OFF])`
- `set appfw profile <name> -exemptClosureURLsFromSecurityChecks ([ON] | [OFF)`
- `set appfw profile <name> -RefererHeaderCheck ([OFF] | [if-present] | [AlwaysExceptStartURLs] | [AlwaysExceptFirstRequest])`

開始 URL チェックのリラクゼーションを指定するには、GUI を使用する必要があります。開始 URL チェックを修正ダイアログボックスの「チェック」タブで、「追加」をクリックして「開始 URL チェック緩和を追加」ダイアログボックスを開くか、既存のリラクゼーションを選択して「開く」をクリックして「開始 URL チェック緩和を修正」ダイアログボックスを開きます。どちらのダイアログボックスにも、リラクゼーションを構成するための同じオプションが表示されます。

開始 URL チェック緩和の例を以下に示します。

- www.example.com のホームページへのアクセスをユーザーに許可します。

```
^http://www[.]example[.]com$
<!--NeedCopy-->
```

- ユーザーがすべての静的 HTML (.htm および.html)、サーバー解析された HTML (.htp および.shtml)、PHP (.php)、Microsoft の ASP (.asp) 形式のウェブページにアクセスすることを許可します。

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]\*/)\*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- www.example-español.com で非 ASCII 文字を含むパス名またはファイル名で Web ページにアクセスすることを許可します。


```
^http://www[.]example-espaxC3xB1ol[.]com/(([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*
([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

注：上記の式では、各文字クラスは文字列
x[0-9a-FA-F でグループ化されています][0-9A-Fa-f]。これは、適切に構築されたすべての文字エンコーディング文字列と一致しますが、UTF-8 文字エンコーディング文字列に関連付けられていない浮遊バックスラッシュ文字は使用できません。二重バックスラッシュ () はエスケープされたバックスラッシュで、Web App Firewall がリテラルバックスラッシュとして解釈するように指示します。バックスラッシュを 1 つだけ指定すると、Web App Firewall は、代わりに次の左角括弧 ([) を文字クラスを開く代わりにリテラル文字として解釈し、式が壊れてしまいます。

- www.example.com にあるすべての GIF (.png)、JPEG (.jpg および.jpeg)、PNG (.png) 形式のグラフィックにユーザーがアクセスできるようにしてください。

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]\*/)\*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](gif|jpe?g|png)$
<!--NeedCopy-->
```

- ユーザに CGI (.cgi) および PERL (.pl) スクリプトへのアクセスを許可しますが、CGI-BIN ディレクトリでのみアクセスできるようにします。

```
^http://www[.]example[.]com/CGI-BIN/[0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$
<!--NeedCopy-->
```

- ユーザーが docsarchive ディレクトリにある Microsoft Office およびその他のドキュメントファイルにアクセスできるようにします。

```
^http://www[.]example[.]com/docsarchive/[0-9A-Za-z][0-9A-Za-z_-.]*[.](doc|xls|pdf|ppt)$
<!--NeedCopy-->
```

> 注
>
> デフォルトでは、すべての Web App Firewall URL は正規表現と見なされます。

注意: 正規表現は強力です。特に PCRE 形式の正規表現にあまり詳しくない場合は、作成した正規表現をすべて再確認してください。例外として追加する URL を正確に定義し、それ以外は何も定義していないことを確認してください。ワイルドカード、特にドットとアスタリスク (
.*) メタ文字とワイルドカードの組み合わせを不注意に使用すると、意図していなかった Web コンテンツへのアクセスをブロックしたり、開始 URL チェックでブロックされたはずの攻撃を許可したりするなど、望ましくない結果になる可能性があります。

> ヒント
>
> URL 命名スキームの SQL キーワードの許可リストに *-and-* を追加できます。たとえば https://FQDN/bread-and-butterの例。

URL チェックを拒否

March 20, 2024

拒否 URL チェックは、ハッカーや悪意のあるコードがよくアクセスする URL への接続を検査してブロックします。このチェックには、ハッカーや悪意のあるコードの一般的なターゲットであり、正規のリクエストではほとんど出現しない URL のリストが含まれています。URL または URL パターンをリストに追加することもできます。Deny URL チェックは、Web サーバーソフトウェアや多くの Web サイトに存在することが知られているさまざまなセキュリティ上の弱点に対する攻撃を防ぎます。

拒否 URL チェックは開始 URL チェックよりも優先されるため、開始 URL の緩和によって通常であればリクエストの続行が許可される場合でも、悪意のある接続の試みは拒否されます。

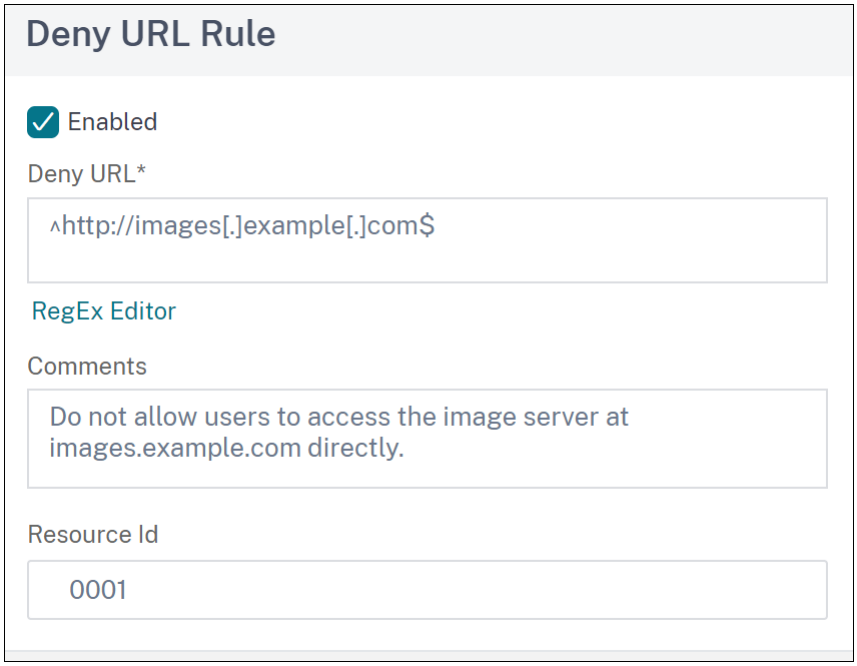
「拒否 URL チェックの変更」ダイアログ・ボックスの「一般」タブで、ブロック、ログ、統計の各アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して拒否 URL チェックを設定できます。

```
set appfw profile <name> -denyURLAction [**block**] [**log**] [**stats
    **] [**none**]
<!--NeedCopy-->
```

独自の拒否 URL を作成および構成できるのは、NetScaler GUI のみです。

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 拒否 URL を追加するプロファイルを選択し、[編集] をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、「詳細設定」セクションから「緩和ルール」を選択します。
4. [**URL** を拒否] を選択して [編集] をクリックします。
5. 「**URL** の拒否ルール」ページで、「追加」をクリックします。
6. 次の詳細を指定し、「作成」をクリックします。
   - 拒否 **URL** -拒否 URL を定義する正規表現。
   - コメント -エクスプレッションの説明。
   - リソース **ID** -拒否 URL ルールを識別するユニークな ID。



7. [閉じる] をクリックします。
8. **NetScaler Web App Firewall** プロファイルページで、「完了」をクリックします。

URL を拒否する表現の例を以下に示します。

- ユーザーが images.example.com のイメージサーバーに直接アクセスできないようにしてください。

```
^http://images[.]example[.]com$
<!--NeedCopy-->
```

- ユーザに CGI (.cgi) または PERL (.pl) スクリプトに直接アクセスさせないでください。

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$
<!--NeedCopy-->
```

- 以下は同じ拒否 URL で、非 ASCII 文字をサポートするように変更されています。

```
^http://www[.]example[.]com/(([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f
   ])
([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|x[0-9A-Fa
   -f][0-9A-Fa-f])
([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*[.](cgi|pl)$
<!--NeedCopy-->
```

> 注意:
>
> 正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。ブロックしたい URL やパターンを正確に定義していることを確認し、それ以外は何も定義してい

> ないことを確認してください。ワイルドカード、特にドットとアスタリスク (.*) のメタ文字とワイルドカードの組み合わせを不注意に使用すると、ブロックするつもりのない Web コンテンツへのアクセスがブロックされるなど、望ましくない結果になる可能性があります。

XML 保護チェック

August 15, 2023

XML 保護チェックは、すべての種類の XML ベースの攻撃に対する要求を調べます。

> 注意:
>
> XML セキュリティーチェックは、text/xml の HTTP コンテンツタイプヘッダーで送信されるコンテンツにのみ適用されます。content-type ヘッダーがないか、別の値に設定されている場合、すべての XML セキュリティチェックはバイパスされます。XML または Web 2.0 Web アプリケーションを保護する場合、それらのアプリケーションをホストする各 Web サーバーのウェブマスターは、適切な HTTP コンテンツタイプヘッダーが送信されていることを確認する必要があります。

XML 形式チェック

August 15, 2023

XML 形式チェックは、受信した要求の XML 形式を調べ、形式が正しくない要求や、形式が適切な XML 文書の XML 仕様の基準を満たさない要求をブロックします。これらの基準には次のようなものがあります。

- XML ドキュメントには、Unicode 仕様に一致する、適切にエンコードされた Unicode 文字のみが含まれている必要があります。
- XML マークアップで使用される場合を除き、<、>、& などの特殊な XML 構文文字を文書に含めることはできません。
- すべての開始タグ、終了タグ、および空要素タグは、欠落したり重複したりしないように、正しくネストされている必要があります。
- XML 要素タグでは大文字と小文字が区別されます。すべての開始タグと終了タグは完全に一致する必要があります。
- 1 つのルート要素には、XML 文書内の他のすべての要素が含まれている必要があります。

整形式の XML の基準を満たさない文書は、XML 文書の定義を満たしていません。厳密に言えば、XML ではありません。ただし、すべての XML アプリケーションと Web サービスが XML 整形式標準を適用しているわけではなく、形式が不適切な XML や無効な XML を正しく処理するとは限りません。形式が不適切な XML ドキュメントを不適切に処理すると、セキュリティ違反を引き起こす可能性があります。XML 形式チェックの目的は、悪意のあるユーザーが

形式が不適切な XML 要求を使用して XML アプリケーションまたは Web サービスのセキュリティを侵害するのを防ぐことです。

ウィザードまたは GUI を使用する場合、「XML 形式チェックの変更」ダイアログボックスの「一般」タブで、ブロック、ログ、および統計アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML Format Check を設定できます。

- `set appfw profile <name> -xmlFormatAction [**block**] [**log**] [**stats**] [**none**]`

XML 形式チェックに例外を設定することはできません。有効化または無効化のみが可能です。

XML サービス拒否チェック

August 15, 2023

XML サービス拒否 (XML DoS または XDoS) チェックでは、受信した XML 要求がサービス拒否 (DoS) 攻撃の特性と一致するかどうかを判断します。一致するものがあれば、それらのリクエストをブロックします。XML DoS チェックの目的は、攻撃者が XML リクエストを使用して Web サーバーや Web サイトにサービス拒否攻撃を仕掛けるのを防ぐことです。

ウィザードまたは GUI を使用する場合、「XML DoS チェックの変更」ダイアログボックスの「一般」タブで、ブロック、ログ、統計、および学習アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML サービス拒否チェックを設定できます。

- `set appfw profile <name> -xmlDoSAction [**block**] [**log**] [**learn**] [**stats**] [**none**]`

XML サービス拒否ルールを個別に設定するには、GUI を使用する必要があります。**[XML** サービス拒否チェック の変更] ダイアログボックスの [チェック] タブで、ルールを選択し、[開く] をクリックして、そのルールの **[XML** サービス拒否 の変更] ダイアログボックスを開きます。個々のダイアログボックスはルールによって異なりますが、シンプルです。ルールを有効または無効にすることしかできないものもあれば、テキストボックスに新しい値を入力して数値を変更できるものもあります。

> 注:
>
> サービス拒否攻撃に対するラーニングエンジンの予想される動作は、設定されたアクションに基づいています。アクションが「ブロック」に設定されている場合、エンジンは設定されたバインド値 +1 を学習し、違反がある

> と XML 解析が停止します。設定されたアクションが「ブロック」に設定されていない場合、エンジンは実際に受信した違反時間の値を学習します。

個々の XML サービス拒否ルールは次のとおりです。

- 最大要素深度。個々の要素の入れ子レベルの最大数を 256 に制限します。このルールが有効になっていて、Web App Firewall が最大許容レベル数を超える要素を含む XML リクエストを検出した場合、そのリクエストはブロックされます。レベルの最大数は 1 から 65,535 までの任意の値に変更できます。
- 要素名の最大長。各要素名の最大長を 128 文字に制限します。これには、展開された名前空間内の名前が含まれます。これには、次の形式の XML パスと要素名が含まれます。

```
1  {
2  http://prefix.example.com/path/ }
3  target_page.xml
4  <!--NeedCopy-->
```

ユーザーは、名前の最大長を 1 文字から 65,535 までの任意の値に変更できます。

- 最大要素数。1 つの XML ドキュメントにつき、1 つのタイプのエレメントの最大数を 65,535 に制限します。エレメントの最大数は、1 から 65,535 までの任意の値に変更できます。
- 子要素の最大数。個々の要素で使用できる子 (他の要素、文字情報、コメントを含む) の最大数を 65,535 人に制限します。子要素の最大数は、1 から 65,535 までの任意の値に変更できます。
- 最大属性数。個々の要素が持つことができる属性の最大数を 256 に制限します。属性の最大数は、1 から 256 までの任意の値に変更できます。
- 属性名の最大長。各属性名の最大長を 128 文字に制限します。属性名の最大長は、1 から 2,048 までの任意の値に変更できます。
- 属性値の最大長。各属性値の最大長を 2048 文字に制限します。属性名の最大長は、1 から 2,048 までの任意の値に変更できます。
- 最大文字データ長。各要素の最大文字データ長を 65,535 に制限します。長さは 1 から 65,535 までの任意の値に変更できます。
- 最大ファイルサイズ。各ファイルのサイズを 20 MB に制限します。最大ファイルサイズは任意の値に変更できます。
- 最小ファイルサイズ。各ファイルの長さは 9 バイト以上である必要があります。最小ファイルサイズは、さまざまなバイトを表す任意の正の整数に変更できます。
- エンティティ拡張の最大数。許可されるエンティティ拡張の数を指定された数に制限します。デフォルト:1024 です。
- エンティティの最大拡張深度。ネストされたエンティティ拡張の最大数を、指定された数以下に制限します。デフォルト:32。

- 名前空間の最大数。XML 文書内の名前空間宣言の数を、指定された数以下に制限します。デフォルトは 16 です。
- 名前空間 URI の最大長。各名前空間宣言の URL の長さを、指定された文字数以下に制限します。デフォルト:256。
- ブロック処理命令。リクエストに含まれる特別な処理命令をすべてブロックします。このルールには、ユーザーが変更できる値はありません。
- DTD をブロックします。リクエストに含まれるすべての文書型定義 (DTD) をブロックします。このルールには、ユーザーが変更できる値はありません。
- 外部エンティティをブロックします。リクエスト内の外部エンティティへの参照をすべてブロックします。このルールには、ユーザーが変更できる値はありません。
- SOAP アレイチェック。次の SOAP 配列チェックを有効または無効にします。
  - **SOAP** 配列の最大サイズ。接続がブロックされる前の XML リクエスト内のすべての SOAP 配列の最大合計サイズ。この値は変更できます。デフォルト:20000000。
  - **SOAP** アレイの最大ランク。接続がブロックされる前の XML リクエスト内の単一の SOAP 配列の最大ランクまたはサイズ。この値は変更できます。デフォルトは 16 です。

XML クロスサイトスクリプティングチェック

August 15, 2023

XML クロスサイトスクリプティングチェックは、XML ペイロードで発生する可能性のあるクロスサイトスクリプティング攻撃に対するユーザーリクエストを調べます。クロスサイトスクリプティング攻撃の可能性が見つかると、リクエストをブロックします。

保護された Web サービス上のスクリプトが悪用されて Web サービスのセキュリティが侵害されるのを防ぐため、XML クロスサイトスクリプティングチェックは、スクリプトが置かれているサーバー以外のサーバー上のコンテンツにアクセスしたり変更したりしてはならないという同じオリジンルールに違反するスクリプトをブロックします。同一生成元ルールに違反するスクリプトはクロスサイトスクリプトと呼ばれ、スクリプトを使用して別のサーバー上のコンテンツにアクセスしたり変更したりすることをクロスサイトスクリプティングと呼びます。クロスサイトスクリプティングがセキュリティ上の問題である理由は、クロスサイトスクリプティングを許可する Web サーバーが、その Web サーバー上ではなく、攻撃者が所有および制御している別の Web サーバー上のスクリプトで攻撃される可能性があるためです。

Web App Firewall には、XML クロスサイトスクリプティング保護を実装するためのさまざまなアクションオプションが用意されています。ブロック、ログ、統計の各アクションを設定することができます。

Web App Firewall XML クロスサイトスクリプティングチェックは、受信したリクエストのペイロードに対して実行され、攻撃文字列が複数行に分散していても識別されます。このチェックでは、****** 要素と属性値に含まれるクロス

サイトスクリプティング攻撃文字列を探します ** 。特定の条件下では、セキュリティチェックの検査をバイパスするための緩和措置を適用できます。ログと統計は、必要な緩和策を特定するのに役立ちます。

XML ペイロードの CDATA セクションは、スクリプトが CDATA セクションの外部では実行できないため、ハッカーにとって魅力的な領域となる可能性があります。CDATA セクションは、コンテンツ全体を文字データとして扱う場合に使用されます。HTML マークアップタグの区切り文字 **<**、**、**/** を使用しても、パーサーはコードを HTML 要素として解釈しません。次の例は、クロスサイトスクリプティング攻撃文字列を含む CDATA セクションを示しています。

```
1      <![CDATA[rn
2      <script language="Javascript" type="text/javascript">alert ("Got
          you")</script>rn
3      ]]>
4  <!--NeedCopy-->
```

アクションオプション

XML クロスサイトスクリプティングチェックがリクエスト内のクロスサイトスクリプティング攻撃を検出すると、アクションが適用されます。アプリケーションの XML クロスサイトスクリプティング保護を最適化するには、次のオプションを使用できます。

- ブロック—リクエストでクロスサイトスクリプティングタグが検出されると、ブロックアクションがトリガーされます。
- ログ：XML クロスサイトスクリプティングチェックによって実行されたアクションを示すログメッセージを生成します。ブロックを無効にすると、クロスサイトスクリプティング違反が検出された場所 (ELEMENT、ATTRIBUTE) ごとに個別のログメッセージが生成されます。ただし、要求がブロックされると、1 つのメッセージだけが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとしたことを示している可能性があります。
- 統計情報—違反やログに関する統計を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当なリクエストがブロックされる場合、新しい緩和ルールを構成する必要があるか、既存の緩和ルールを変更する必要があるかを確認するために、構成を再確認しなければならない場合があります。

リラクゼーションルール

アプリケーションで XML ペイロード内の特定の ELEMENT または ATTRIBUTE のクロスサイトスクリプティングチェックをバイパスする必要がある場合は、緩和ルールを設定できます。XML クロスサイトスクリプティングチェック緩和ルールには次のパラメータがあります。

- 名前—リテラル文字列または正規表現を使用して、要素または属性の名前を設定できます。次の式では、文字列 name_ で始まり、その後に大文字または小文字、または数字からなる文字列、つまり 2 文字以上で 15 文

字以下のすべての要素が除外されます。

```
^name_[0-9A-Za-z]{ 2,15 } $
```

> 注
>
> 大文字と小文字は区別されます。重複したエントリは許可されませんが、名前の大文字と場所の違いを使用して、類似のエントリを作成できます。たとえば、次の緩和規則はそれぞれ一意です。
>
> 1. XMLcross-site scripting: ABC IsRegex: NOTREGEX Location: ATTRIBUTE State: ENABLED
> 2. XMLcross-site scripting: ABC IsRegex: NOTREGEX Location: ELEMENT State: ENABLED
> 3. XMLcross-site scripting: abc IsRegex: NOTREGEX Location: ELEMENT State: ENABLED
> 4. XMLcross-site scripting: abc IsRegex: NOTREGEX Location: ATTRIBUTE State: ENABLED

- 場所—XML ペイロード内のクロスサイトスクリプティングチェック例外の場所を指定できます。デフォルトでは ELEMENT オプションが選択されています。属性に変更できます。
- コメント：これはオプションのフィールドです。この緩和ルールの目的を説明するには、最大 255 文字の文字列を使用できます。

> 警告
>
> 正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加したい名前とまったく同じ名前を定義し、それ以外は何も定義していないことを確認してください。正規表現を不注意に使用すると、意図していなかった Web コンテンツへのアクセスをブロックしたり、XML クロスサイトスクリプティングチェックでブロックされたはずの攻撃を許可したりするなど、望ましくない結果が生じる可能性があります。

コマンドラインを使用して **XML** クロスサイトスクリプティングチェックを設定する

コマンドラインを使用して XML クロスサイトスクリプティングチェックアクションとその他のパラメータを設定するには

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML クロスサイトスクリプティングチェックを設定できます。

```
> set appfw profile <name> -XMLcross-site scriptingAction (([block] [log] [stats])| [none])
```

コマンドラインを使用して XML クロスサイトスクリプティングチェック緩和ルールを設定するには

緩和ルールを追加して、特定の場所でのクロスサイトスクリプティングスクリプト攻撃インスペクションのインスペクションをバイパスできます。以下のように bind または unbind コマンドを使用して、緩和ルールバインディングを追加または削除します。

```
> bind appfw profile <name> -XMLcross-site scripting <string> [
isRegex (REGEX | NOTREGEX)] [-location ( ELEMENT | ATTRIBUTE )] -
comment <string> [-state ( ENABLED | DISABLED )]
```

```
> unbind appfw profile <name> -XMLcross-site scripting <String>
```

例:

```
> bind appfw profile test_pr -XMLcross-site scripting ABC
```

上記のコマンドを実行すると、次の緩和ルールが設定されます。ルールが有効になり、名前はリテラル (NOTREGEX) として扱われ、ELEMENT がデフォルトの場所として選択されます。

```
1)      XMLcross-site scripting:  ABC             IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

`> unbind appfw profile test_pr -XMLcross-site scripting abc`

ERROR: No such XMLcross-site scripting check

`> unbind appfw profile test_pr -XMLcross-site scripting ABC`

 Done
<!--NeedCopy-->
```

GUI を使用して XML クロスサイトスクリプティングチェックを設定する

GUI では、アプリケーションに関連するプロファイルのペインで XML クロスサイトスクリプティングチェックを設定できます。

GUI を使用して XML クロスサイトスクリプティングチェックを設定または変更するには

1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ペインで、[セキュリティチェック] をクリックします。

セキュリティー検査テーブルには、すべてのセキュリティー検査に対して現在構成されているアクション設定が表示されます。設定には次の 2 つのオプションがあります。

a) **XML** クロスサイトスクリプティングチェックのブロック、ログ、統計アクションを有効または無効にするだけの場合は、表のチェックボックスをオンまたはオフにして、「**OK**」をクリックし、「保存して閉じる」をクリックしてセキュリティチェックペインを閉じます。

b) **XML** クロスサイトスクリプティングをダブルクリックするか、行を選択して「アクション設定」をクリックすると、アクションオプションが表示されます。アクション設定のいずれかを変更したら、「**OK**」をクリックして変更を保存し、「セキュリティチェック」テーブルに戻ります。

必要に応じて、他のセキュリティー検査の設定に進むことができます。［**OK** ］をクリックして［セキュリティチェック］セクションで行った変更をすべて保存し、［保存して閉じる］をクリックしてセキュリティチェックウィンドウを閉じます。

GUI を使用して XML クロスサイトスクリプティング緩和ルールを設定するには

1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ウィンドウで、[緩和規則] をクリックします。
3. 「緩和規則」テーブルで、「XML クロスサイトスクリプティング」エントリをダブルクリックするか、エントリを選択して「編集」をクリックします。
4. **XML** クロスサイトスクリプティング緩和規則ダイアログボックスで、緩和規則の追加 **、** 編集、削除、有効化、または無効化の操作を実行します。

ビジュアライザーを使用して XML クロスサイトスクリプティング緩和ルールを管理するには

すべての緩和ルールをまとめて表示するには、「緩和ルール」テーブルの「XML クロスサイトスクリプティング」行を強調表示して、「ビジュアライザー」をクリックします。デプロイされたリラクゼーションのビジュアライザーには、[新しいルールを追加] または **[**既存のルールを編集**]** のオプションがあります。ノードを選択し、緩和ビジュアライザの対応するボタンをクリックして、ルールのグループ を有効 または 無効に することもできます。

GUI を使用してクロスサイトスクリプティングパターンを表示またはカスタマイズするには

GUI を使用して、クロスサイトスクリプティングが許可される属性または許可されたタグのデフォルトリストを表示またはカスタマイズできます。クロスサイトスクリプティング拒否パターンのデフォルトリストを表示またはカスタマイズすることもできます。

デフォルトのリストは、「**Web App Firewall** 」>「署名」>「** デフォルト署名 **」で指定されます。シグネチャオブジェクトをプロファイルにバインドしない場合、Default Signatures オブジェクトに指定されているデフォルトのクロスサイトスクリプティング許可リストと拒否リストがプロファイルによってクロスサイトスクリプティングセキュリティチェック処理に使用されます。デフォルトのシグネチャオブジェクトで指定された Tags、Attributes、Patterns は読み取り専用です。編集や修正はできません。これらを変更または変更する場合は、Default Signatures オブジェクトのコピーを作成して、ユーザー定義署名オブジェクトを作成します。新しいユーザー定義署名オブジェクトの許可リストまたは拒否リストを変更し、カスタマイズされた許可リストと拒否リストを使用したいトラフィックを処理しているプロファイルでこの署名オブジェクトを使用します。

署名の詳細については、を参照してください。http://support.citrix.com/proddocs/topic/ns-security-10-map/appfw-signatures-con.html

デフォルトのクロスサイトスクリプティングパターンを表示するには**:**

1. 「Web App Firewall」**>**「署名」に移動し、「*** デフォルト署名」を選択し、「編集」をクリックします。次に、**[**SQL/**クロスサイトスクリプティングパターンの管理] をクリックします。

「**SQL/**クロスサイトスクリプティングパスの管理」テーブルには、クロスサイトスクリプティングに関する次の **3** つの行が表示されます。

```
            xss/allowed/attribute

            xss/allowed/tag

            xss/denied/pattern
<!--NeedCopy-->
```

行を選択して「要素の管理」をクリックすると、**Web App Firewall** のクロスサイトスクリプティングチェックで使用される対応するクロスサイトスクリプティング要素 **(**タグ、属性、パターン**)** が表示されます。

クロスサイトスクリプティング要素をカスタマイズするには: ユーザー定義の署名オブジェクトを編集して、許可されるタグ、許可される属性、拒否パターンをカスタマイズできます。新しいエントリを追加したり、既存のエントリを削除したりできます。

1. **[**Web App Firewall] > [署名] に移動し、対象のユーザー定義署名を強調表示して、**[**編集**]** をクリックします。[**SQL/**クロスサイトスクリプティングパターンの管理**]** をクリックして、**[SQL/**クロスサイトスクリプティングパスの管理] テーブルを表示します。
2. ターゲットのクロスサイトスクリプティング行を選択します。

a)「要素の管理」をクリックして、対応するクロスサイトスクリプティング要素を追加、編集、または削除します。

b) 選択した行を削除するには、「削除」をクリックします。

> 警告
>
> デフォルトのクロスサイトスクリプティング要素を削除または変更したり、クロスサイトスクリプティングパスを削除して行全体を削除したりする場合は、十分に注意してください。シグネチャ、HTML クロスサイトスクリプティングセキュリティチェック、XML クロスサイトスクリプティングセキュリティチェックは、これらの Elements を利用して攻撃を検出し、アプリケーションを保護します。クロスサイトスクリプティングエレメントをカスタマイズすると、編集中に必要なパターンが削除されると、アプリケーションがクロスサイトスクリプティング攻撃に対して脆弱になる可能性があります。

XML クロスサイトスクリプティングチェックでのログ機能の使用

ログアクションを有効にすると、XML クロスサイトスクリプティングのセキュリティチェック違反が **AppFW_XML_Cross-Site** Script 違反として監査ログに記録されます。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて、/var/log/ フォルダー内の ns.logs を末尾に移動して、XML クロスサイトスクリプティング違反に関するログメッセージにアクセスします。


```
> \*\*Shell\*\*

> \*\*tail -f /var/log/ns.log | grep APPFW_XML_cross-site scripting\*\*
<!--NeedCopy-->
```

<blocked>アクションを示すネイティブログ形式の **XML** クロスサイトスクリプティングセキュリティチェック違反ログメッセージの例

```
Oct  7 01:44:34 <local0.warn> 10.217.31.98 10/07/2015:01:44:34 GMT ns
    0-PPE-1 : default APPFW APPFW_XML_cross-site scripting 1154 0 :
    10.217.253.69 3466-PPE1 - owa_profile http://10.217.31.101/FFC/login
    .html Cross-site script check failed for field script="Bad tag:
    script" <\*\*blocked\*\*>
<!--NeedCopy-->
```

<not blocked>アクションを示す CEF ログ形式の XML クロスサイトスクリプティングセキュリティチェック違反ログメッセージの例

```
Oct  7 01:46:52 <local0.warn> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|APPFW_XML_cross-site scripting|4|src=10.217.30.17
    geolocation=Unknown spt=33141 method=GET request=http://
    10.217.31.101/FFC/login.html msg=Cross-site script check failed for
    field script="Bad tag: script" cn1=1607 cn2=3538 cs1=owa_profile cs2
    =PPE0 cs4=ERROR cs5=2015 act=\*\*not blocked\*\*
<!--NeedCopy-->
```

GUI を使用してログメッセージにアクセスするには

GUI には、ログメッセージを分析するための便利なツール (**Syslog Viewer**) が含まれています。Syslog ビューアにアクセスするには、複数のオプションがあります。

- **Web App Firewall** > プロファイルに移動し、ターゲットプロファイルを選択して、セキュリティチェックをクリックします。「XML クロスサイトスクリプティング」行を強調表示して、「ログ」をクリックします。プロファイルの XML クロスサイトスクリプティングチェックから直接ログにアクセスすると、GUI はログメッセージをフィルタリングし、これらのセキュリティチェック違反に関連するログのみを表示します。
- 「**NetScaler**」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。監査メッセージセクションで、Syslog メッセージリンクをクリックして Syslog Viewer を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティチェック違反がトリガーされる可能性がある場合のデバッグに役立ちます。
- [**Web App Firewall**] > [ポリシー] > [監査] に移動します。監査メッセージセクションで、**Syslog** メッセージリンクをクリックして Syslog Viewer を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

XML ベースの Syslog Viewer には、関心のあるログメッセージのみを選択するためのさまざまなフィルターオプションが用意されています。**XML** クロスサイトスクリプティングチェックのログメッセージを選択するには、モジュールのドロップダウンオプションで ****APPFW** を選択してフィルタリングします。****[Event Type]** リストには、

選択内容をさらに絞り込むための豊富なオプションが用意されています。たとえば、**AppFW_XML_Cross-Site scripting** チェックボックスを選択して「**** 適用」ボタンをクリックすると、XML クロスサイトスクリプティングのセキュリティチェック違反に関するログメッセージのみが ** Syslog Viewer に表示されます。

特定のログメッセージの行にカーソルを置くと、[モジュール]、[イベントタイプ]、[イベント **ID**]、[クライアント **IP**] などの複数のオプションがログメッセージの下に表示されます。これらのオプションを選択すると、ログメッセージ内の対応する情報を強調表示できます。

XML クロスサイトスクリプティング違反の統計

stats アクションを有効にすると、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、XML クロスサイトスクリプティングチェックのカウンタが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分サイズは、構成された設定によって異なります。たとえば、ブロックアクションが有効な場合、XML クロスサイトスクリプティング違反が 3 つあるページのリクエストでは、最初の違反が検出されるとすぐにページがブロックされるため、stats カウンタが 1 つ増えます。ただし、ブロックが無効になっている場合、同じリクエストを処理すると、違反ごとに個別のログメッセージが生成されるため、違反とログの統計カウンタが 3 つ増えます。

XML クロスサイトスクリプティングの統計情報を表示するには、コマンドラインを使用して

コマンドプロンプトで入力します。

```
> **sh appfw stats**
```

特定のプロファイルの統計情報を表示するには、以下のコマンドを使用します。

```
> **stat appfw profile** <profile name>
```

GUI を使用して XML クロスサイトスクリプティング統計を表示するには

1. [システム] > [セキュリティ] > [**Web App Firewall**] に移動します。
2. 右側のペインで、[統計 リンク] にアクセスします。
3. スクロールバーを使用して、XML クロスサイトスクリプティング違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

XML SQL インジェクションチェック

August 15, 2023

XML SQL インジェクションチェックは、XML SQL インジェクション攻撃の可能性についてのユーザー要求を調べます。XML ペイロードに注入された SQL が見つかると、要求がブロックされます。

XML SQL 攻撃は、ソースコードを Web アプリケーションに注入して、それを有効な SQL クエリとして解釈して実行し、悪意を持ってデータベース操作を実行する可能性があります。たとえば、XML SQL 攻撃を仕掛けて、データベ

ースの内容に不正にアクセスしたり、保存されているデータを操作したりすることができます。XML SQL インジェクション攻撃はよくあるだけでなく、非常に有害でコストもかかります。

データベースユーザーの権限を区分化すると、データベースをある程度保護するのに役立ちます。すべてのデータベースユーザーには、SQL クエリを実行して他のタスクを実行できないように、目的のタスクを完了するために必要な権限のみを与える必要があります。たとえば、読み取り専用ユーザーには、データテーブルへの書き込みや操作を許可してはなりません。Web App Firewall の XML SQL インジェクションチェックは、すべての XML リクエストを検査して、セキュリティを侵害する可能性のある不正な SQL コードの注入に対する特別な防御策を提供します。Web App Firewall は、任意のユーザーの XML リクエストで不正な SQL コードを検出すると、そのリクエストをブロックできます。

NetScaler Web App Firewall は、SQL キーワードと特殊文字の有無を検査して、XML SQL インジェクション攻撃を特定します。デフォルトのキーワードと特殊文字のセットには、XML SQL 攻撃を仕掛けるのに一般的に使用される既知のキーワードと特殊文字が含まれます。Web App Firewall は、一重引用符 (‘)、バックスラッシュ ()、セミコロン (;) の 3 文字を SQL セキュリティチェック処理の特殊文字と見なします。新しいパターンを追加したり、デフォルトセットを編集して XML SQL チェックインスペクションをカスタマイズしたりできます。

Web App Firewall には、XML SQL インジェクション保護を実装するためのさまざまなアクションオプションが用意されています。リクエストをブロックしたり、検出された違反に関する詳細を含むメッセージを ns.log ファイルに記録したり、統計情報を収集して監視された攻撃の数を追跡したりできます。

アクションの他に、XML SQL インジェクション処理用に構成できるパラメータがいくつかあります。**SQL** ワイルドカード文字をチェックできます。XML SQL インジェクションタイプを変更し、4 つのオプション (**sqlKeyword**、**sqlSplCharar**、**sqlSplCharandKeyword****、**sqlSplCharorKeyword****) のいずれかを選択して、**XML** ペイロードを処理する際の **SQL** キーワードと SQL 特殊文字の評価方法を指定できます。XML **SQL** コメント処理パラメータには、XML SQL インジェクション検出時に検査または除外する必要があるコメントの種類を指定するオプションがあります。

リラクゼーションを展開すると、誤検出を回避できます。Web App Firewall XML SQL チェックは受信リクエストのペイロードに対して実行され、攻撃文字列が複数行に分散していても識別されます。このチェックでは、要素内の SQL インジェクション文字列と属性値が検索されます。特定の条件下では、セキュリティチェックの検査をバイパスするための緩和措置を適用できます。ログと統計は、必要な緩和策を特定するのに役立ちます。

アクションオプション

アクションは、XML SQL インジェクションチェックがリクエスト内の SQL インジェクション攻撃文字列を検出したときに適用されます。アプリケーションに最適化された XML SQL インジェクション保護を設定するには、次のアクションを使用できます。

ブロック—ブロックを有効にすると、入力が XML SQL インジェクションタイプの仕様と一致する場合にのみブロックアクションがトリガーされます。たとえば、**SQLSplCharAndKeyword** が **XML SQL** インジェクションタイプとして設定されている場合、ペイロードで **SQL** 特殊文字が検出された場合でも、リクエストにキーワード

が含まれていないリクエストはブロックされません。**XML SQL** インジェクションタイプが sqlSplchar または sqlSplCharorKeyword のいずれかに設定されている場合、このようなリクエストはブロックされます。

ログ：ログ機能を有効にすると、XML SQL インジェクションチェックが実行するアクションを示すログメッセージが生成されます。ブロックが無効になっている場合、XML SQL 違反が検出された場所 (**ELEMENT**、**ATTRIBUTE**) ごとに個別のログメッセージが生成されます。ただし、要求がブロックされると、1 つのメッセージだけが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとしたことを示している可能性があります。

[**Stats**]：有効にすると、統計機能は違反とログに関する統計情報を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当なリクエストがブロックされる場合、新しい緩和ルールを構成する必要があるか、既存の緩和ルールを変更する必要があるかを確認するために、構成を再確認しなければならない場合があります。

XML SQL パラメーター

ブロック、ログ、統計アクションに加えて、XML SQL インジェクションチェックには次のパラメータを設定できます。

XML SQL ワイルドカード文字の確認—ワイルドカード文字を使用すると、構造化クエリ言語 (SQL-SELECT) ステートメントの選択肢を広げることができます。これらのワイルドカード演算子を LIKE 演算子と **NOTLIKE** 演算子と組み合わせて使用すると、値を類似の値と比較できます。パーセント (%) およびアンダースコア (_) 文字は、ワイルドカードとしてよく使用されます。パーセント記号は、MS-DOS で使用されるアスタリスク (*) ワイルドカード文字に似ており、フィールド内の 0、1、または複数の文字に一致します。アンダースコアは、MS-DOS の疑問符 (?) に似ています。ワイルドカード文字。これは、式の 1 つの数字または文字に一致します。

たとえば、次のクエリを使用して文字列検索を実行し、名前に D 文字が含まれるすべての顧客を検索できます。

```
SELECT * from customer WHERE name like "%D%"
```

次の例では、演算子を組み合わせて、2 番目と 3 番目の文字が 0 の給与値をすべて検索します。

```
SELECT * from customer WHERE salary like '_00%
```

DBMS ベンダーによっては、演算子を追加してワイルドカード文字を拡張しています。NetScaler Web App Firewall は、これらのワイルドカード文字を挿入することによって開始される攻撃から保護できます。デフォルトのワイルドカード文字は、パーセント (%)、アンダースコア (_)、キャレット (^)、開始角括弧 ([)、および閉じ角括弧 (]) の 5 文字です。この保護は、HTML プロファイルと XML プロファイルの両方に適用されます。

デフォルトのワイルドカード文字は、***Default Signatures** で指定されたリテラルのリストです。

```
1 -  <wildchar type=" LITERAL" >%</wildchar>
2 -  <wildchar type=" LITERAL" >_</wildchar>
3 -  <wildchar type=" LITERAL" >^</wildchar>
4 -  <wildchar type=" LITERAL" >[</wildchar>
5 -  <wildchar type=" LITERAL" >]</wildchar>
6 <!--NeedCopy-->
```


攻撃のワイルドカード文字は [^A-F] のように PCRE になります。Web App Firewall は PCRE ワイルドカードもサポートしていますが、ほとんどの攻撃をブロックするには、上記のリテラルワイルドカード文字で十分です。

> 注
>
> XML SQL ワイルドカード文字チェックは XML SQL 特殊文字チェックとは異なります。誤検出を避けるため、このオプションは注意して使用する必要があります。

SQL インジェクションタイプを含むリクエストの確認—Web App Firewall には、アプリケーションの個々のニーズに基づいて、SQL インジェクションインスペクションに必要なレベルの厳格さを実装するための 4 つのオプションが用意されています。SQL 違反を検出するために、リクエストはインジェクションタイプの指定と照合されます。SQL インジェクションタイプには、次の 4 つのオプションがあります。

- **SQL** 特殊文字とキーワード：SQL 違反をトリガーするには、SQL キーワードと SQL 特殊文字の両方が検査対象の場所にある必要があります。この最も制限の少ない設定もデフォルト設定です。
- **SQL** 特殊文字—SQL 違反をトリガーするには、処理されたペイロード文字列に少なくとも 1 つの特殊文字が含まれている必要があります。
- **SQL** キーワード：SQL 違反をトリガーするには、指定した SQL キーワードのうち少なくとも 1 つが処理されたペイロード文字列に含まれている必要があります。このオプションは十分に考慮せずに選択しないでください。誤検出を避けるため、入力にキーワードが含まれていないことを確認します。
- **SQL** 特殊文字またはキーワード：セキュリティチェック違反をトリガーするには、キーワードまたは特殊文字列のいずれかがペイロードに含まれている必要があります。

> ヒント
>
> SQL 特殊文字オプションを選択すると、Web App Firewall は特殊文字を含まない文字列をスキップします。ほとんどの SQL サーバーは、前に特殊文字が付いていない SQL コマンドを処理しないため、このオプションを有効にすると、Web App Firewall の負荷を大幅に軽減し、保護された Web サイトを危険にさらすことなく処理を高速化できます。

SQL コメント処理—デフォルトでは、Web App Firewall は XML データ内のすべてのコメントを解析してチェックし、挿入された SQL コマンドを確認します。多くの SQL サーバーでは、SQL の特殊文字が先頭にあっても、コメント内の内容はすべて無視されます。処理を高速化するために、XML SQL サーバーがコメントを無視する場合、挿入された SQL のリクエストを調べるときにコメントをスキップするように Web App Firewall を構成できます。XML SQL コメント処理オプションは次のとおりです。

- **ANSI**—UNIX ベースの SQL データベースで通常使用される ANSI 形式の SQL コメントをスキップします。
- ネスト: ネストされた SQL コメントをスキップします。このコメントは、Microsoft SQL Server で通常使用されます。
-
- すべてのコメントを確認—何もスキップせずに、挿入された SQL のリクエスト全体をチェックします。これがデフォルトの設定です。

> ヒント
>
> バックエンドデータベースが Microsoft SQL Server で実行されている場合を除き、ほとんどの場合、ネストオプションまたは ANSI/ネストオプションを選択しないでください。他のほとんどの種類の SQL Server ソフトウェアは、ネストされたコメントを認識しません。ネストされたコメントが、別の種類の SQL Server 宛ての要求に表示される場合は、そのサーバーのセキュリティ侵害の試みを示している可能性があります。

リラクゼーションルール

アプリケーションで XML ペイロード内の特定の ELEMENT または ATTRIBUTE に対する XML SQL インジェクションインスペクションをバイパスする必要がある場合は、緩和ルールを設定できます。XML SQL インジェクション検査緩和ルールには次のパラメータがあります。

- 名前**:** リテラル文字列または正規表現を使用して、**** 要素または属性の名前を設定できます。次の式では、****PurchaseOrder_** という文字列で始まり、その後に長さが 2 文字以上 10 文字以下の数字の文字列が続く要素はすべて除外されます。

  コメント: 「発注書要素の XML SQL チェックを免除」

```
    XMLSQLInjection:  "PurchaseOrder_[0-9A-Za-z]{
 2,10 }
 "

    IsRegex:  REGEX          Location:  ELEMENT

    State:  ENABLED
<!--NeedCopy-->
```

注: 名前は大文字と小文字が区別されます。重複したエントリは許可されませんが、名前の大文字と場所の違いを使用して、類似のエントリを作成できます。たとえば、次の緩和ルールはそれぞれ異なります。

```
1)      XMLSQLInjection:  XYZ    IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

2)      XMLSQLInjection:  xyz    IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

3)      XMLSQLInjection:  xyz    IsRegex:  NOTREGEX

        Location:  ATTRIBUTE     State:  ENABLED

4)      XMLSQLInjection:  XYZ    IsRegex:  NOTREGEX

        Location:  ATTRIBUTE     State:  ENABLED
<!--NeedCopy-->
```


- 場所:XML ペイロード内の XML SQL インスペクション例外の場所を指定できます。デフォルトでは **ELEMENT** オプションが選択されています。属性に変更できます。
- コメント: これはオプションのフィールドです。この緩和ルールの目的を説明するには、最大 255 文字の文字列を使用できます。

> 警告
>
> 正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加したい名前とまったく同じ名前を定義し、それ以外は何も定義していないことを確認してください。正規表現を不注意に使用すると、意図していなかった Web コンテンツへのアクセスをブロックしたり、XML SQL インスペクションでブロックされたはずの攻撃を許可したりするなど、望ましくない結果が生じる可能性があります。

コマンドラインを使用して **XML SQL** インジェクションチェックを設定する

コマンドラインを使用して **XML SQL** インジェクションアクションとその他のパラメータを設定するには:

コマンドラインインターフェイスでは、**set appfw profile** コマンドまたは add appfw profile** コマンドのいずれかを使用して **XML SQL** インジェクション保護を設定できます。ブロック、ログ、統計アクションを有効にできます。ペイロードで検出したい **SQL** 攻撃パターンのタイプ **(**キーワード、ワイルドカード文字、特殊文字列**)** を選択します。****unset appfw profile** コマンドを使用して、構成した設定をデフォルトに戻します。次のコマンドはそれぞれ 1 つのパラメータのみを設定しますが、1 つのコマンドに複数のパラメータを含めることができます。

- set appfw profile <name> **-XMLSQLInjectionAction** ([block] [log] [stats])| [none])
- set appfw profile <name> -XMLSQLInjectionCheckSQLWildChars (ON | OFF)
- set appfw profile <name> -XMLSQLInjectionType ([SQLKeyword] | [SQLSplChar] | [SQLSplCharANDKeyword] | [SQLSplCharORKeyword])
- set appfw profile <name> -XMLSQLInjectionParseComments ([checkall] | [ansi|nested] | [ansinested])

コマンドラインを使用して SQL インジェクション緩和ルールを設定するには

バインドまたはバインド解除コマンドを使用して、次のように緩和ルールを追加または削除します。

```
1 -  bind appfw profile <name> -XMLSQLInjection <string> [isRegex (REGEX
     | NOTREGEX)] [-location ( ELEMENT | ATTRIBUTE )]  - comment <string>
     [-state ( ENABLED | DISABLED )]
2 -  unbind appfw profile <name> -XMLSQLInjection <String>
3 <!--NeedCopy-->
```

例:

```
1 > bind appfw profile test_profile -XMLSQLInjection "PurchaseOrder_[0-9A
     -Za-z]{
```



```
 2,15 }
 " -isregex REGEX -location ATTRIBUTE

> unbind appfw profile test_profile  -XMLSQLInjection "PurchaseOrder_
   [0-9A-Za-z]{
 2,15 }
 " -location ATTRIBUTE
<!--NeedCopy-->
```

GUI を使用して XMLSQL インジェクションセキュリティチェックを構成する

GUI では、アプリケーションに関連するプロファイルのペインで XML SQL インジェクションセキュリティチェックを設定できます。

GUI を使用して XML SQL インジェクションチェックを設定または変更するには

1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ペインで、[セキュリティチェック] をクリックします。

セキュリティー検査テーブルには、すべてのセキュリティー検査に対して現在構成されているアクション設定が表示されます。設定には次の 2 つのオプションがあります。

a. XML SQL インジェクションのブロック、ログ、統計アクションを有効または無効にするだけの場合は、テーブルのチェックボックスをオンまたはオフにして、「OK」をクリックし、「保存して閉じる」をクリックして「セキュリティチェック」ペインを閉じます。

b. このセキュリティーチェックに追加のオプションを設定する場合は、XML SQL インジェクションをダブルクリックするか、行を選択して「アクション設定」をクリックすると、次のオプションが表示されます。

SQL ワイルドカード文字の確認—ペイロード内の SQL ワイルドカード文字を攻撃パターンと見なします。

「次を含むリクエストのチェック」—チェックする SQL インジェクションのタイプ (sqlKeyword、sqlSPLChar、sqlSPLCharandKeyword、または sqlSPLCharorKeyword)。

SQL コメントの処理-チェックするコメントのタイプ ([すべてのコメントをチェック]、[ANSI]、[ネスト]、または [ANSI/ネスト])。

上記の設定のいずれかを変更したら、「**OK**」をクリックして変更内容を保存し、「セキュリティチェック」(Security Checks) テーブルに戻ります。必要に応じて、他のセキュリティー検査の設定に進むことができます。「**OK**」をクリックして「セキュリティ・チェック」セクションで行ったすべての変更を保存し、「保存して閉じる」をクリックしてセキュリティ・チェック・ペインを閉じます。

GUI を使用して XML SQL インジェクション緩和ルールを設定するには

1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。

2. [詳細設定] ウィンドウで、[緩和規則] をクリックします。
3.「緩和規則」テーブルで、**XML SQL** インジェクションのエントリをダブルクリックするか、エントリを選択して「編集」をクリックします。
4. **XML SQL** インジェクション緩和規則ダイアログボックスで、緩和規則の追加、編集、削除、有効化、または無効化の操作を実行します。

ビジュアライザーを使用して XML SQL インジェクション緩和ルールを管理するには

すべての緩和ルールをまとめて表示するには、緩和ルール表の **XML SQL** インジェクション行を強調表示して、「ビジュアライザー」をクリックします。デプロイされたリラクゼーションのビジュアライザーには、[新しいルールを追加] または **[**既存のルールを編集**]** のオプションがあります。ノードを選択し、緩和ビジュアライザの対応するボタンをクリックして、ルールのグループ を有効 または 無効に することもできます。

GUI を使用して **SQL** インジェクションパターンを表示またはカスタマイズするには:

GUI を使用して SQL パターンを表示またはカスタマイズできます。

デフォルトの SQL パターンは、**Web App Firewall** > 署名 **> ***** デフォルト署名で指定されます。** シグネチャオブジェクトをプロファイルにバインドしない場合、Default Signatures オブジェクトで指定されたデフォルト SQL パターンが、プロファイルの XML SQL インジェクションセキュリティチェック処理に使用されます。Default Signatures オブジェクトのルールとパターンは読み取り専用です。編集や修正はできません。これらのパターンを変更または変更する場合は、Default Signatures オブジェクトのコピーを作成して SQL パターンを変更して、ユーザー定義の署名オブジェクトを作成します。カスタマイズされた SQL パターンを使用するトラフィックを処理するプロファイルで、ユーザ定義のシグニチャオブジェクトを使用します。

詳細については、「署名」を参照してください。

デフォルトの **SQL** パターンを表示するには、次のステップを実行します。

a.「Web App Firewall」**>**「署名」に移動し、「******* デフォルト署名」を選択し、「編集」をクリックします。次に、**[**SQL/**クロスサイトスクリプティングパターンの管理] をクリックします。

「SQL/クロスサイトスクリプティングパスの管理」テーブルには、SQL インジェクションに関する次の 4 行が表示されます。

```
Injection (not_alphanum, SQL)/ Keyword

Injection (not_alphanum, SQL)/ specialstring

Injection (not_alphanum, SQL)/ transformrules/transform

Injection (not_alphanum, SQL)/ wildchar
<!--NeedCopy-->
```

b. 行を選択して「要素の管理」をクリックすると、Web App Firewall SQL インジェクションチェックで使用される対応する SQL パターン (キーワード、特殊文字列、変換ルール、またはワイルドカード文字) が表示されます。

SQL パターンをカスタマイズするには**:** ユーザー定義の署名オブジェクトを編集して、SQL キーワード、特殊文字列、およびワイルドカード文字をカスタマイズできます。新しいエントリを追加したり、既存のエントリを削除した

りできます。SQL 特殊文字列の変換ルールを変更できます。

a. **Web App Firewall> **** 署名に移動し、ターゲットのユーザー定義署名を強調表示して、「編集」をクリックします。**[**SQL/**クロスサイトスクリプティングパターンの管理]** をクリックして、**[SQL/**クロスサイトスクリプティングパスの管理] テーブルを表示します。

b. ターゲット SQL 行を選択します。

i. [要素の管理] をクリックして、対応する SQL 要素を追加、編集、または削除します。

ii. [削除**]** をクリックして、選択した行を削除します。

> 警告
>
> デフォルトの SQL 要素を削除または変更したり、SQL パスを削除して行全体を削除したりする場合は、十分に注意する必要があります。シグネチャルールと XML SQL インジェクションセキュリティチェックは、これらの要素を利用して SQL インジェクション攻撃を検出し、アプリケーションを保護します。SQL パターンをカスタマイズすると、編集中に必要なパターンが削除されると、アプリケーションが XML SQL 攻撃に対して脆弱になる可能性があります。

XML SQL インジェクションチェックでのログ機能の使用

ログアクションを有効にすると、**XML SQL** インジェクションのセキュリティチェック違反が **APPFW_XML_SQL** 違反として監査ログに記録されます。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

コマンドラインを使用してログメッセージにアクセスするには:

シェルに切り替えて、/var/log/ フォルダー内の ns.logs を末尾に移動して、XML クロスサイトスクリプティング違反に関するログメッセージにアクセスします。

```
> Shell

> tail -f /var/log/ns.log | grep APPFW_XML_SQL
<!--NeedCopy-->
```

GUI を使用してログメッセージにアクセスするには

GUI には、ログメッセージを分析するための便利なツール (Syslog Viewer) が含まれています。Syslog ビューアにアクセスするには、複数のオプションがあります。

- **Web App Firewall** > プロファイルに移動し、ターゲットプロファイルを選択して、セキュリティチェックをクリックします。**XML SQL** インジェクション行を強調表示して、「ログ」をクリックします。プロファイルの XML SQL インジェクションチェックから直接ログにアクセスすると、GUI はログメッセージをフィルタリングし、これらのセキュリティチェック違反に関連するログのみを表示します。
- **[**** システム] > [監査] に移動して Syslog ビューアにアクセスすることもできます。監査メッセージセクションで、****Syslog** メッセージリンクをクリックして Syslog Viewer を表示します。このビューアには、他

のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティチェック違反がトリガーされる可能性がある場合のデバッグに役立ちます。

- [**Web App Firewall**] > [ポリシー] > [監査] に移動します。[監査メッセージ] セクションで、[**Syslog** メッセージ] リンクをクリックして **Syslog** ビューアを表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

XML ベースの Syslog Viewer には、関心のあるログメッセージのみを選択するためのさまざまなフィルターオプションが用意されています。**XML SQL** インジェクションチェックのログメッセージを選択するには、モジュールのドロップダウンオプションで ****APPFW** を選択してフィルタリングします。****[Event Type]** リストには、選択内容をさらに絞り込むための豊富なオプションが用意されています。たとえば、**APPFW_XML_SQL** チェックボックスを選択して「****** 適用」ボタンをクリックすると、XML SQL****** インジェクションのセキュリティチェック違反に関するログメッセージのみが Syslog ビューアに表示されます。

特定のログメッセージの行にカーソルを置くと、モジュール、イベントタイプ、イベント **ID**、クライアント **IP** などの複数のオプションがログメッセージの下に表示されます。これらのオプションを選択すると、ログメッセージ内の対応する情報を強調表示できます。

XML SQL インジェクション違反の統計情報

stats アクションを有効にすると、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、**XML SQL** インジェクションチェックのカウンタが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分サイズは、構成された設定によって異なります。たとえば、ブロックアクションが有効な場合、3 つの **XML SQL** インジェクション違反を含むページのリクエストでは、最初の違反が検出されるとすぐにページがブロックされるため、stats カウンタが 1 つ増えます。ただし、ブロックが無効になっている場合、同じリクエストを処理すると、違反ごとに個別のログメッセージが生成されるため、違反とログの統計カウンタが 3 つ増えます。

XML SQL インジェクションの統計情報を表示するには、コマンドラインを使用して

コマンドプロンプトで入力します。

```
> sh appfw stats
```

特定のプロファイルの統計情報を表示するには、以下のコマンドを使用します。

```
> stat appfw profile <profile name>
```

GUI を使用して XML SQL インジェクションの統計を表示するには

1. [システム] > [セキュリティ] > [**Web App Firewall**] に移動します。
2. 右側のペインで、[統計 リンク] にアクセスします。
3. スクロールバーを使用して、**XML SQL** インジェクションの違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

XML 添付ファイルチェック

August 15, 2023

XML 添付ファイルチェックは、受信したリクエストに悪意のある添付ファイルがないかを調べ、アプリケーションのセキュリティを侵害する可能性のある添付ファイルを含むリクエストをブロックします。XML 添付ファイルチェックの目的は、攻撃者が XML 添付ファイルを使用してサーバーのセキュリティを侵害するのを防ぐことです。

ウィザードまたは GUI を使用する場合、「XML 添付ファイルチェックの変更」ダイアログボックスの「一般」タブで、「ブロック」、「学習」、「ログ」、「統計」、および「学習」の各アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML 添付ファイルチェックを設定できます。

- `set appfw profile <name> -xmlAttachmentAction [block] [learn] [log] [stats] [none]`

その他の XML 添付ファイルチェック設定は GUI で設定する必要があります。`Modify XML Attachment` チェックダイアログボックスの「チェック」タブでは、次の設定を構成できます。

- 添付ファイルの最大サイズ。指定した最大添付ファイルサイズ以下の添付ファイルを許可します。このオプションを有効にするには、まず [有効] チェックボックスをオンにし、`Size` テキストボックスに添付ファイルの最大サイズをバイト単位で入力します。
- 添付ファイルのコンテンツタイプ。指定されたコンテンツタイプの添付を許可します。このオプションを有効にするには、まず [有効] チェックボックスをオンにし、許可する添付ファイルの Content-Type 属性と一致する正規表現を入力します。
  - URL 式をテキストウィンドウに直接入力できます。その場合は、手動で入力する代わりに、`Regex Tokens` メニューを使用して便利な正規表現をカーソル位置に入力できます。
  - [Regex Editor] をクリックして`Add Regular Expression` ダイアログボックスを開き、それを使用して URL 式を作成できます。

Web サービスの相互運用性チェック

August 15, 2023

Web サービス相互運用性 (WS-I) チェックは、要求と応答の両方が WS-I 標準に準拠しているかどうかを調べ、この標準に準拠していない要求と応答をブロックします。WS-I チェックの目的は、他の XML と適切に相互作用しない可能性のある要求をブロックすることです。攻撃者は、相互運用性の不一致を利用して、XML アプリケーションに攻撃を仕掛ける可能性があります。

ウィザードまたは GUI を使用する場合、「Web サービス相互運用性チェックの変更」ダイアログボックスの「一般」タブで、「ブロック」、「ログ」、「統計」、および「学習」アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して Web サービスの相互運用性チェックを設定できます。

- `set appfw profile <name> -xmlWSIAction [block] ][log] [learn] [stats] [none]`

Web サービス相互運用性ルールを個別に設定するには、GUI を使用する必要があります。「Web サービス相互運用性チェックの変更」ダイアログ・ボックスの「チェック」タブで、ルールを選択し、「有効化」または「無効化」をクリックしてルールを有効または無効にします。[開く] をクリックして、そのルールの [Web サービス相互運用性の詳細] メッセージボックスを開くこともできます。メッセージボックスには、ルールに関する読み取り専用情報が表示されます。これらのルールを変更したり、その他の設定を変更したりすることはできません。

WS-I チェックは WS-I 基本プロファイル 1.0 に記載されているルールを使用します。WS-I は、相互運用可能な Web サービスソリューションを開発するためのベストプラクティスを提供します。WS-I チェックは SOAP メッセージに対してのみ実行されます。

各 WSI 標準ルールの説明を以下に示します。

規則	説明
BP1201	メッセージ本文は名前空間を持つ soap: envelope でなければなりません。
R1000	ENVELOPE が Fault である場合、soap:Fault 要素は、faultcode、faultstring、faultactor、detail 以外の要素の子を持つことはできません。
R1001	ENVELOPE が Fault である場合、soap:Fault 要素の子は、修飾されなければなりません。
R1003	受信者は、詳細要素にゼロを含む修飾属性または非修飾属性がいくつでも現れる障害メッセージを受け入れる必要があります。修飾属性の名前空間は、修飾された文書要素エンベロープの名前空間以外であれば何でもかまいません。
R1004	ENVELOPE に faultcode 要素が含まれる場合、その要素の内容は、SOAP 1.1 で定義されている障害コード (必要に応じて詳細要素に追加情報を提供) のいずれか、または障害の指定機関によって名前空間が制御されている Qname (優先順に) でなければなりません。
R1005	ENVELOPE は、その名前空間が修飾された文書要素 Envelope の名前空間と同じである要素のいずれかの soap を含んではいけません。

規則	説明
R1006	EnVELOPE は、soap:Body の子である任意の要素の soap:encodingStyle 属性を含んではいけません。
R1007	rpc リテラルバインディングで記述された ENVELOPE には、SOAP: Body の孫であるどの要素にも soap: EncodingStyle 属性が含まれていてはなりません。
R1011	EnVELOPE には、SOAPE: Body 要素に続く SOAPE: Envelope の要素の子があってはなりません。
R1012	メッセージは UTF-8 または UTF-16 としてシリアル化する必要があります。
R1013	soap:mustUnderstand 属性を含む ENVELOPE は、字句形式 0 と 1 のみを使用しなければなりません。
R1014	ENVELOPE の soap:Body 要素の子は、名前空間修飾でなければなりません。
R1015	文書要素が SOAP: Envelope ではないエンベロープに遭遇した場合、受信者はエラーを起こさなければなりません。
R1031	ENVELOPE に faultcode 要素が含まれている場合は、その要素の内容に SOAP 1.1 のドット表記を使用して障害の意味を絞り込んではいけません。
R1032	The soap:Envelope, soap:Header, and soap:ENVELOPE 内のボディ要素は、修飾された文書要素のエンベロープのそれと同じ名前空間内の属性を持つことはできません
R1033	EnVELOPE は名前空間宣言を含んではいけません: `xmlns:xml=http://www.w3.org/XML/1998/namespace.`
R1109	HTTP リクエストメッセージの SOAPAction HTTP ヘッダーフィールドの値は、引用符で囲まれた文字列でなければなりません。
R1111	INSTANCE は、障害ではないエンベロープを含む応答メッセージに対して 200 OK HTTP ステータスコードを使用すべきです。
R1126	レスポンスエンベロープが Fault の場合、インスタンスは 500 Internal Server Error HTTP ステータスコードを返さなければなりません。
R1132	HTTP リクエストメッセージには HTTP POST メソッドを使用する必要があります。

規則	説明
R1140	メッセージは HTTP/1.1 を使用して送信する必要があります。
R1141	メッセージは HTTP/1.1 または HTTP/1.0 のいずれかを使用して送信する必要があります。
R2113	エンベロープには、soapenc:arrayType 属性を含めることはできません。
R2211	rpc-literal バインディングで記述された ENVELOPE には、パートアクセサーに xsi: nil 属性の値が 1 または true があってはなりません。
R2714	一方向操作の場合、INSTANCE はエンベロープを含む HTTP 応答を返してはいけません。具体的には、HTTP 応答のエンティティ本文は空でなければなりません。
R2729	rpc リテラルバインディングで記述されたレスポンスである ENVELOPE には、対応する wsdl: operation 名の末尾に StringResponse が付いた名前のラッパー要素が必要です。
R2735	rpc-literal バインディングで記述された ENVELOPE は、パラメータと戻り値のパートアクセサー要素を名前空間に置かないでください。
R2738	ENVELOPE には、wsdl：input または wsdl：それを記述する wsdl：バインディングの wsdl：操作の wsdl: 出力で指定されたすべての soapbind を含める必要があります。
R2740	説明文の wsdl: binding には、既知のエラーをそれぞれ説明する soapbind: fault が含まれているはずです。
R2744	HTTP 要求メッセージには、対応する WSDL 記述内に存在する場合、soapbind:operation の soapAction 属性の値と等しい引用符で囲まれた値を持つ SOAPAction HTTP ヘッダーフィールドが含まれていなければなりません。

XML メッセージ検証チェック

August 15, 2023

XML メッセージ検証チェックでは、XML メッセージを含むリクエストが検証され、有効であるかどうかが確認され

ます。リクエストに無効な XML メッセージが含まれている場合、Web App Firewall はそのリクエストをブロックします。XML 検証チェックの目的は、攻撃者が特別に作成された無効な XML メッセージを使用してアプリケーションのセキュリティを侵害するのを防ぐことです。

ウィザードまたは GUI を使用する場合、「XML メッセージ検証チェックの変更」ダイアログボックスの「一般」タブで、「ブロック」、「ログ」、および「統計」アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML メッセージ検証チェックを設定できます。

- `set appfw profile <name> -xmlValidationAction [**block**] [**log**] [**stats**] [**none**]`

GUI を使用してその他の XML 検証チェック設定を構成する必要があります。「XML
メッセージ検証チェックの変更」ダイアログボックスの「
チェック」タブでは、次の設定を構成できます。

- **XML** メッセージの検証。XML メッセージを検証するには、次のいずれかのオプションを使用します。
  - **SOAP** エンベロープ。XML メッセージの SOAP エンベロープのみを検証します。
  - **WSDL**。XML SOAP WSDL を使用して XML メッセージを検証します。WSDL 検証を選択する場合、「WSDL オブジェクト」ドロップダウンリストで WSDL を選択する必要があります。Web App Firewall にまだインポートされていない WSDL に対して検証する場合は、「インポート」ボタンをクリックして「WSDL インポートの管理」ダイアログを開き、WSDL をインポートします。詳細については、WSDL を参照してください。
    * URL 全体を検証する場合は、[エンドポイントチェック] ボタン配列の [絶対] ラジオボタンを選択したままにします。URL のホストより後の部分だけを検証する場合は、「相対」ラジオボタンを選択します。
    * Web App Firewall に WSDL を厳密に適用し、WSDL で定義されていない追加の XML ヘッダーを許可しないようにするには、「WSDL で定義されていない追加のヘッダーを許可する」チェックボックスをオフにする必要があります。
      注意:「WSDL で定義されていないヘッダーを
      許可する」チェックボックスのチェックを外し、保護された XML アプリケーションまたは Web 2.0 アプリケーションが期待する XML ヘッダーや、クライアントが送信する XML ヘッダーが WSDL で定義されていない場合、保護されたサービスへの正当なアクセスをブロックする可能性があります。
  - **XML** スキーマ。XML スキーマを使用して XML メッセージを検証します。XML スキーマの検証を選択する場合、「XML スキーマオブジェクト」ドロップダウンリストで XML スキーマを選択する必要があります。Web App Firewall にまだインポートされていない XML スキーマに対して検証する場合は、「インポート」ボタンをクリックして「XML スキーマのインポートの管理」ダイアログを開き、WSDL をインポートします。詳細については、WSDL を参照してください。
- 応答の検証。デフォルトでは、Web App Firewall は応答の検証を試みません。保護対象のアプリケーションまたは Web 2.0 サイトからの応答を検証する場合は、「応答の検証」チェックボックスを選択します。これを

行うと、「要求検証で指定された XML スキーマを再使用」チェックボックスと「XML スキーマオブジェクト」ドロップダウンリストがアクティブになります。

- リクエストの検証で指定したスキーマを使用して応答の検証も行うには、「XML スキーマの再使用」チェックボックスをオンにします。
  注: このチェックボックスをオンにすると、「
  XML スキーマオブジェクト」ドロップダウンリストはグレー表示されます。
- 応答の検証に別の XML スキーマを使用する場合は、「XML スキーマオブジェクト」ドロップダウンリストを使用して、その XML スキーマを選択またはアップロードします。

XML SOAP 障害フィルタリングチェック

August 15, 2023

XML SOAP 障害フィルタリングチェックは、保護されている Web サービスからの応答を調べ、XML SOAP 障害を除外します。これにより、機密情報が攻撃者に漏洩するのを防ぎます。

ウィザードまたは GUI を使用する場合、「XML SOAP 障害フィルタリングチェックの変更」ダイアログボックスの「一般」タブで、ブロック、ログ、統計の各アクションと、応答をユーザーに転送する前に SOAP 障害を削除する削除アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML SOAP 障害フィルタリングチェックを設定できます。

```
set appfw profile <name> -XMLSOAPFaultAction [block] [log] [stats] [
none]
```

XML SOAP 障害フィルタリングチェックに例外を設定することはできません。有効化または無効化のみが可能です。

JSON 保護チェック

August 15, 2023

NetScaler Web App Firewall は、コンテンツレベルの DoS、SQL、またはクロスサイトスクリプティング攻撃から JSON アプリケーションを保護します。JSON リクエストに DoS 攻撃、SQL 攻撃、クロスサイトスクリプティング攻撃がある場合、配列や文字列などの JSON 構造に制限を設定してアプリケーションを保護する必要があります。

> 注:
>
> JSON セキュリティチェックは、JSON コンテンツタイプヘッダーで送信されるコンテンツにのみ適用されます。content-type ヘッダーがないか、別の値に設定されている場合、すべての JSON セキュリティチェックはバイパスされます。JSON アプリケーションを保護する場合、それらのアプリケーションをホストする各 Web サーバーのウェブマスターは、適切な JSON コンテンツタイプのヘッダーが送信されることを確認する必要があります。
>
> 学習機能は JSON SQL、クロスサイトスクリプティング、DOS コンテンツタイプをサポートしていません。

JSON サービス拒否保護チェック

March 20, 2024

JSON サービス拒否 (DoS) チェックは、受信 JSON リクエストを調べ、DoS 攻撃の特性と一致するデータがあるかどうかを検証します。リクエストに JSON 違反がある場合、アプライアンスはリクエストをブロックし、データをログに記録し、SNMP アラートを送信し、JSON エラーページも表示します。JSON DoS チェックの目的は、攻撃者が JSON リクエストを送信して JSON アプリケーションまたは Web サイトに DoS 攻撃を開始するのを防ぐことです。

クライアントが NetScaler アプライアンスに要求を送信すると、JSON パーサーは要求ペイロードを解析し、違反が検出された場合、アプライアンスは JSON 構造に制約を適用します。この制約により、JSON リクエストにサイズ制限が適用されます。その結果、JSON 違反が検出されると、アプライアンスはアクションを適用し、JSON エラーページで応答します。

JSON DoS ルール

アプライアンスが JSON リクエストを受信すると、JSON DOS 保護により、リクエストペイロード内の次の DoS パラメータにサイズ制限が適用されます。

1. 最大深度:JSON ドキュメントの最大ネスト (深さ)。このチェックは、階層の深さが過度に深いドキュメントから保護します。
2. 最大ドキュメント長:JSON ドキュメントの最大ドキュメント長。
3. 配列の最大長:JSON オブジェクトの配列の最大長。このチェックは、長い配列に対して保護します。
4. 文字列の最大長:JSON 内の文字列の最大長。このチェックは、長さの長い文字列から保護します。
5. 最大オブジェクトキーカウント:JSON オブジェクトの最大キー数。このチェックにより、大量のキーを持つオブジェクトから保護されます。
6. オブジェクトキーの最大長:JSON オブジェクトのキーの最大長。このチェックは、大きなキーを持つオブジェクトから保護します。

JSON の解析中に検証される JSON DoS ルールのリストを次に示します。

1. JSONmax コンテナの深さ。このチェックは jsonMaxContainerDepth チェックを設定することで有効にできます。デフォルトでは、このオプションは OFF です。
2. JSONmax コンテナの深さ。このチェックは、設定可能なオプション jsonMaxContainerDepthCheck によって有効または無効にでき、デフォルト値は jsonMaxContainerDepth オプションで変更できます。ただし、最大レベルは 1 ～127 の範囲の値に変更できます。デフォルト値:5、最小値:1、最大値:127
3. jsonMaxDocumentLength。このチェックは、jsonMaxDocumentLength チェックを設定することで有効にできます。デフォルトのオプションは OFF です。
4. jsonMaxDocumentLength。このチェックは、jsonMaxDocumentLength チェックを設定することで有効にできます。デフォルトの長さは 20000000 バイトに設定されています。最小値:1, 最大値:2147483647
5. jsonMaxObjectKeyCount。このルールは、JSON の最大オブジェクトキー数チェックがオンかオフかを検証します。可能な値: オン、オフ、デフォルト値:OFF
6. jsonMaxObjectKeyCount。このチェックは、jsonMaxObjectKeyCount チェックを設定することで有効にできます。このチェックでは、多数のキーを持つオブジェクトから保護され、デフォルト値は 1000 バイトに設定されています。最小値:0、最大値:2147483647
7. jsonMaxObjectKeyLength。このチェックは、jsonMaxObjectKeyLength チェックを設定することで有効にできます。このルールは、JSON オブジェクトキーの最大長チェックがオンかオフかを検証します。デフォルトではオフになっています。
8. jsonMaxObjectKeyLength。このチェックは、キーの長さが長いオブジェクトから保護します。デフォルト値は 128 です。最小値:1, 最大値:2147483647
9. jsonMaxArrayLength。このルールは、JSON 配列の最大長チェックが ON か OFF かを検証します。デフォルトではオフになっています。
10. jsonMaxArrayLength。このチェックは、長さが長いアレイから保護します。デフォルトでは、この値は 10000 に設定されています。最小値:1, 最大値:2147483647
11. JSONMaxStringLength。このチェックは、jsonMaxStringLength チェックを設定することで有効にできます。このチェックでは、JSON の最大文字列長が ON か OFF かを検証します。デフォルトではオフになっています。
12. JSONMaxStringLength。このチェックは、長さの長い文字列から保護します。デフォルトでは 1000000 に設定されています。最小値:1, 最大値:2147483647

JSON DoS 保護チェック

JSON DoS 保護を設定するには、次の手順を完了する必要があります。

1. JSON 用のアプリケーションファイアウォールプロファイルを追加します。

2. JSON DoS 設定のアプリケーションファイアウォールプロファイルを設定します。
3. アプリケーションファイアウォールプロファイルをバインドして JSON DoS 変数を設定します。

JSON DoS 保護用のアプリケーションファイアウォールプロファイルの追加

最初に、アプリケーションファイアウォールが JSON Web コンテンツを JSON DoS 攻撃から保護する方法を指定するプロファイルを作成する必要があります。
コマンドプロンプトで入力します:

```
add appfw profile <name> -type (HTML | XML | JSON)
```

> 注記:
>
> プロファイルタイプを JSON に設定すると、HTML や XML などの他のチェックは適用されません。

例 `add appfw profile profile1 -type JSON`

JSON DoS 保護用のアプリケーションファイアウォールプロファイルの設定

アプリケーションファイアウォールプロファイルに設定される 1 つ以上の JSON DoS アクションと JSON DoS エラーオブジェクトのプロファイルを設定する必要があります。
コマンドプロンプトで入力します:

```
set appfw profile <name> -JSONDoSAction [block] | [log] | [stats] | [none]
```

Block-このセキュリティー検査に違反する接続をブロックします。
Log - このセキュリティチェックの違反を記録します。
Stats-このセキュリティー検査の統計を生成します。
[なし]-このセキュリティー検査に対するすべてのアクションを無効にします。

> 注記:
>
> 1 つ以上のアクションを有効にするには、「set appfw profile-jsonDosAction」に続けて有効にするアクションを入力します。

例 `set appfw profile profile1 -JSONDoSAction block log stat`

アプリケーションファイアウォールプロファイルをバインドして **DoS** 変数を構成する

JSON DoS 保護を提供するには、アプリケーションファイアウォールプロファイルを JSON DoS 設定にバインドする必要があります。
コマンドプロンプトで入力します:


```
bind appfw profile <name> -JSONDoSURL <expression> [-JSONMaxContainerDepthChec
 ( ON | OFF )[-JSONMaxContainerDepth <positive_integer>]] [-JSONMaxDocumentLen
 ( ON | OFF )[-JSONMaxDocumentLength <positive_integer>]] [-JSONMaxObjectKeyCo
 ( ON | OFF )[-SONMaxObjectKeyCount <positive_integer>]] [-JSONMaxObjectKeyLen
 ( ON | OFF )[-JSONMaxObjectKeyLength <positive_integer>]] [-JSONMaxArrayLengt
 ( ON | OFF )[-JSONMaxArrayLength <positive_integer>]] [-JSONMaxStringLengthCh
 ( ON | OFF )[-JSONMaxStringLength <positive_integer>]]
```

例 `bind appfw profile profile1 -JSONDoSURL “.*” -JSONMaxContainerDepthCheck ON`

> 注記:
>
> JSON DoS チェックは、プロファイルタイプが JSON として選択された場合にのみ適用されます。また、JSON プロファイルの場合は、SQL、クロスサイトスクリプティング、フィールド形式、およびフォームフィールドシグネチャが Query パラメータに適用されます。

JSON エラーページのインポート

着信要求に DoS 攻撃があった場合、その要求をブロックすると、アプライアンスはエラーメッセージを表示します。
そのためには、JSON エラーページをインポートする必要があります。
コマンドプロンプトで入力します:

```
import appfw jsonerrorpage <src> <name> [-comment <string>] [-overwrite
]
```

各項目の意味は次のとおりです。

src。インポートされた JSON エラーオブジェクトを格納する場所の URL (プロトコル、ホスト、パス、名前)。

> 注記:
>
> インポートするオブジェクトが、アクセスにクライアント証明書認証を必要とする HTTPS サーバ上にある場合、インポートは失敗します。これは必須の議論です。最大長:2047

名前。NetScaler 上の JSON エラーオブジェクトに割り当てる名前。これは必須の議論です。最大長:31
コメント. JSON エラーオブジェクトに関する情報を保持するコメント。最大長:255
上書き。同じ名前の既存の JSON エラーオブジェクトを上書きします。

設定例

```
1 Add appfw prof profjson  - type JSON
2 Bind appfw prof profjson  - JSONDoSURL  “.*”  -
      JSONMaxDocumentLengthCheck ON  -JSONMaxDocumentLength 30 -
      JSONMaxContainerDepthCheck ON  -JSONMaxContainerDepth  3
      JSONMaxObjectKeyCountCheck ON  -JSONMaxObjectKeyCount  4 -
```



```
    JSONMaxObjectKeyLengthCheck  ON   -JSONMaxObjectKeyLength  10 -
    JSONMaxArrayLengthCheck ON  -JSONMaxArrayLength  5 -
    JSONMaxStringLengthCheck ON -JSONMaxStringLength 30
<!--NeedCopy-->
```

ペイロード、ログメッセージ、カウンタの例**:**

JSONmaxDocumentLength 違反　JSONMaxDocumentLength: 30

Payload: { “a” :” A” ,” b” :” B” ,” c” :” C” ,” d” :” D” ,” e” :” E” }

ログメッセージ**:**

```
Document Length exceeds 20000000 May 29 20:23:32 <local0.info>
    10.217.31.243 05/29/2019:20:23:32 GMT 0-PPE-0 : default APPFW
    APPFW_JSON_DOS_MAX_DOCUMENT_LENGTH 136 0 : 10.217.32.134 114-PPE0 -
    profjson http://10.217.30.120/forms/login.html Document exceeds
    maximum document length (30). cn1=30467 cn2=115 cs1=profjson cs2=
    PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

カウンター**:**

```
1 0 6 as_viol_json_dos
2 0 3 as_viol_json_dos_max_document_length
3 0 6 as_log_json_dos
4 0 3 as_log_json_dos_max_document_length
5 0 6 as_viol_json_dos_profile appfw__(profile1)
6 0 3 as_viol_json_dos_max_document_length_profile appfw__(profile1)
7 0 6 as_log_json_dos_profile appfw__(profile1)
8 0 3 as_log_json_dos_max_document_length_profile appfw__(profile1)
<!--NeedCopy-->
```

jsonMaxContainerDepth 違反　JSONMaxContainerDepth: 3

Payload: { “a” : {” b” : { “c” : { “d” : { ” e” : “f” }}}}}

ログメッセージ**:**

```
May 29 19:33:59 <local0.info> 10.217.31.243 05/29/2019:19:33:59 GMT 0-
    PPE-1 : default APPFW APPFW_JSON_DOS_MAX_CONTAINER_DEPTH 4626 0 :
    10.217.31.247 22-PPE1 - profjson http://10.217.30.120/forms/login.
    html Document at offset (15) exceeds maximum container depth (3).
    cn1=30466 cn2=113 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=
    blocked
<!--NeedCopy-->
```

カウンター**:**

```
36 20999 7 1 0 as_viol_json_dos
37 0 6 1 0 as_viol_json_dos_max_container_depth
38 0 7 1 0 as_log_json_dos
```



```
39 0 6 1 0 as_log_json_dos_max_container_depth
40 0 7 1 0 as_viol_json_dos_profile appfw__(profile1)
41 0 6 1 0 as_viol_json_dos_max_container_depth_profile appfw__(
    profile1)
42 0 7 1 0 as_log_json_dos_profile appfw__(profile1)
43 0 6 1 0 as_log_json_dos_max_container_depth_profile appfw__(profile1
    )
<!--NeedCopy-->
```

jsonMaxObjectKeycount 違反　jsonMaxObjectKeycount: 4

Payload: { “a” : ” A” , “b” : “B” , “c” : “C” , “d” : “D” ,” e” : “E” }

ログメッセージ**:**

```
May 30 19:42:41 <local0.info> 10.217.31.243 05/30/2019:19:42:41 GMT 0-
    PPE-1 : default APPFW APPFW_JSON_DOS_MAX_OBJECT_KEY_COUNT 457 0 :
    10.217.32.134 219-PPE1 - profjson http://10.217.30.120/forms/login.
    html Object at offset (41) that exceeds maximum key count (4). cn1
    =30468 cn2=118 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

カウンター**:**

```
94 119105 15 1 0 as_viol_json_dos
95 0 4 1 0 as_viol_json_dos_max_object_key_count
96 0 15 1 0 as_log_json_dos
97 0 4 1 0 as_log_json_dos_max_object_key_count
98 0 15 1 0 as_viol_json_dos_profile appfw__(profile1)
99 0 4 1 0 as_viol_json_dos_max_object_key_count_profile appfw__(
    profile1)
100 0 15 1 0 as_log_json_dos_profile appfw__(profile1)
101 0 4 1 0 as_log_json_dos_max_object_key_count_profile appfw__(
    profile1)
<!--NeedCopy-->
```

jsonMaxObjectKeyLength 違反　JSONMaxObjectKeyLength: 10

Payload: { “a” : ” A” , “b1234567890” : “B” , “c” : “C” , “d” : “D” ,” e” : “E” }

ログメッセージ**:**

```
May 31 20:26:10 <local0.info> 10.217.31.243 05/31/2019:20:26:10 GMT 0-
    PPE-1 : default APPFW APPFW_JSON_DOS_MAX_OBJECT_KEY_LENGTH 102 0 :
    10.217.32.134 89-PPE1 - profjson http://10.217.30.120/forms/login.
    html Object key(b1234567890) at offset (12) exceeds maximum key
    length (10). cn1=30469 cn2=118 cs1=profjson cs2=PPE0 cs4=ALERT cs5
    =2019 act=blocked
<!--NeedCopy-->
```

カウンター**:**


```
242172 6 1 0 as_viol_json_dos
0 1 1 0 as_viol_json_dos_max_object_key_length
10 0 5 1 0 as_log_json_dos
11 0 1 1 0 as_log_json_dos_max_object_key_length
12 0 6 1 0 as_viol_json_dos_profile appfw__(profile1)
13 0 1 1 0 as_viol_json_dos_max_object_key_length_profile appfw__(
   profile1)
14 0 5 1 0 as_log_json_dos_profile appfw__(profile1)
15 0 1 1 0 as_log_json_dos_max_object_key_length_profile appfw__(
   profile1)
<!--NeedCopy-->
```

JSONMaxArrayLength Violation JSONMaxArrayLength: 5

Payload: { “a” : ” A” , “c” :[” d” ,” e” ,” f” ,” g” ,” h” ,” i”],” e” :[“E” ,” e”]}

ログメッセージ**:**

```
May 29 20:58:39 <local0.info> 10.217.31.243 05/29/2019:20:58:39 GMT 0-
   PPE-1 : default APPFW APPFW_JSON_DOS_MAX_ARRAY_LENGTH 4650 0 :
   10.217.32.134 153-PPE1 -profjson http://10.217.30.120/forms/login.
   html Array at offset (37) that exceeds maximum array length (5). cn1
   =30469 cn2=120 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

カウンター**:**

```
36 182293 10 1 0 as_viol_json_dos
37 0 1 1 0 as_viol_json_dos_max_array_length
38 0 10 1 0 as_log_json_dos 39 0 1 1 0 as_log_json_dos_max_array_length
40 0 10 1 0 as_viol_json_dos_profile appfw__(profile1)
41 0 1 1 0 as_viol_json_dos_max_array_length_profile appfw__(profile1)
42 0 10 1 0 as_log_json_dos_profile appfw__(profile1)
43 0 1 1 0 as_log_json_dos_max_array_length_profile appfw__(profile1))
<!--NeedCopy-->
```

JSONmaxStringLength 違反 JSON 最大文字列長:10

Payload: { “a” : ” A” , “c” :” CcCcCcCcCcCcCcCcCc” ,” e” :[“E” ,” e”]}

ログメッセージ**:**

```
May 29 20:05:02 <local0.info> 10.217.31.243 05/29/2019:20:05:02 GMT 0-
   PPE-0 : default APPFW APPFW_JSON_DOS_MAX_STRING_LENGTH 134 0 :
   10.217.32.134 80-PPE0 - profjson http://10.217.30.120/forms/login.
   html String(CcCcCcCcCcCcCc) at offset (27) that exceeds maximum
   string length (10). n1=30470 cn2=122 cs1=profjson cs2=PPE0 cs4=ALERT
    cs5=2019 act=blocked
<!--NeedCopy-->
```

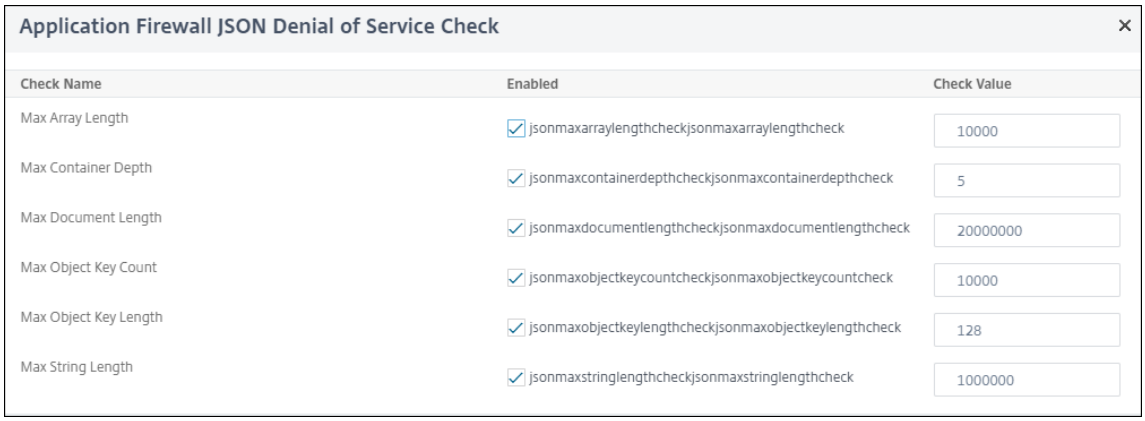
カウンター**:**


```
44 91079 3 1 0 as_viol_json_dos
45 0 1 1 0 as_viol_json_dos_max_string_length
46 0 3 1 0 as_log_json_dos
47 0 1 1 0 as_log_json_dos_max_string_length
48 0 3 1 0 as_viol_json_dos_profile appfw__(profile1)
49 0 1 1 0 as_viol_json_dos_max_string_length_profile appfw__(profile1)
50 0 3 1 0 as_log_json_dos_profile appfw__(profile1)
51 0 1 1 0 as_log_json_dos_max_string_length_profile appfw__(profile1
<!--NeedCopy-->
```

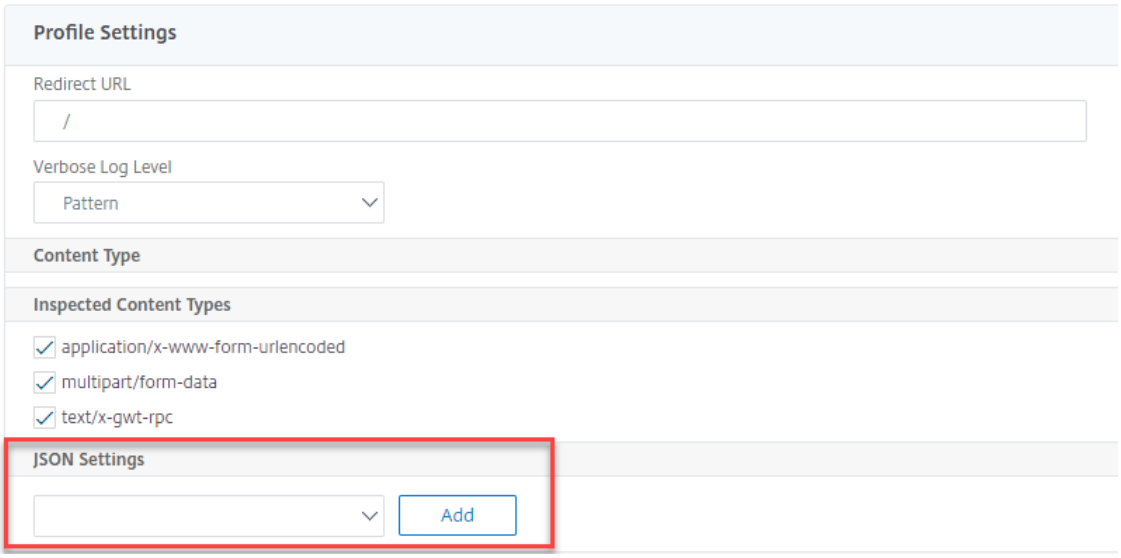
GUI を使用して **JSON DoS** プロテクションを設定する

JSON DoS 保護設定を設定するには、次の手順に従います。

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. 「プロファイル」ページで、「追加」をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、［詳細設定］の［セキュリティチェック］をクリックします。
4. [セキュリティチェック] セクションで、[**JSON** サービス拒否設定] に移動します。
5. チェックボックスの近くにある実行可能アイコンをクリックします。
6. [アクション設定] をクリックして、[**JSON** サービス拒否設定**]** ページにアクセスします。
7. JSON DoS アクションを選択します。
8. ［**OK**］をクリックします。
9. ［**NetScaler Web App Firewall** プロファイル］ページで、［詳細設定］の［緩和ルール］をクリックします。
10. [緩和ルール] セクションで、[**JSON** サービス拒否設定] を選択し、[編集] をクリックします。
11. アプリケーションファイアウォール **JSON** サービス拒否チェックで、JSON DoS 検証値を設定します。
12. ［**OK**］をクリックします。

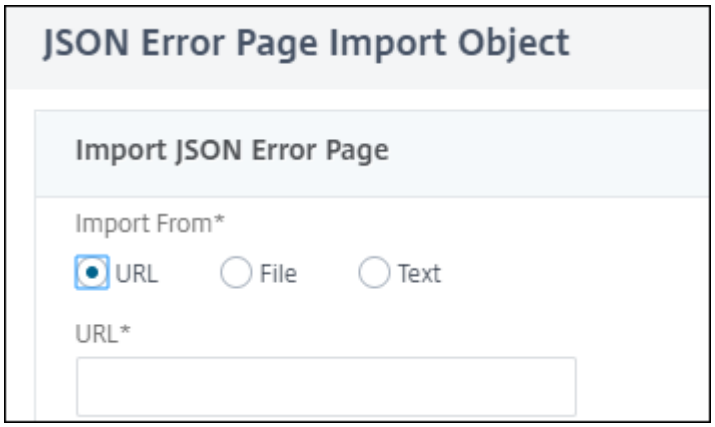


13. **NetScaler Web App Firewall**［プロファイル］ページで、［詳細設定］の［プロファイル設定］をクリックします。
14. **[**プロファイル設定**]** セクションで、[**JSON** エラー設定**]** サブセクションに移動して、**JSON DoS** エラーページを設定します。



15. **JSON** エラーページの「オブジェクトのインポート」ページで、次のパラメータを設定します。

a) からインポートします。エラーページをテキスト、ファイル、または URL としてインポートします。
b) URL。ユーザーをエラーページにリダイレクトする URL。
1 ファイル。JSON DoS エラーファイルとしてインポートするファイルを選択します。
c) テキスト。JSON ファイルの内容を入力します。
d) ［続行］をクリックします。
e) ファイル。ファイル名を入力します。
f) ファイルコンテンツ。エラーファイルの内容を追加します。
g) ［**OK**］をクリックします。



16. ［**OK**］をクリックします。

17. ［完了］をクリックします。

JSON SQL インジェクション保護

March 20, 2024

受信 JSON リクエストでは、部分的な SQL クエリ文字列またはコード内の不正なコマンドの形式で SQL インジェクションが行われることがあります。これにより、ウェブサーバーの JSON データベースからデータが盗まれることになります。このような要求を受信すると、アプライアンスはお客様のデータ保護要求をブロックします。

クライアントが JSON SQL 要求を NetScaler アプライアンスに送信し、JSON パーサーが要求ペイロードを解析し、SQL インジェクションが観察された場合、アプライアンスは JSON SQL コンテンツに制約を適用するシナリオを考えてみましょう。この制約により、JSON SQL リクエストにサイズ制限が適用されます。その結果、JSON SQL インジェクションが検出されると、アプライアンスはアクションを適用し、JSON SQL エラーページで応答します。

JSON SQL インジェクション保護

JSON SQL 保護を設定するには、次の手順を完了する必要があります。

1. アプリケーションファイアウォールプロファイルを JSON として追加します。
2. JSON SQL インジェクション設定のアプリケーションファイアウォールプロファイルの設定
3. アプリケーションファイアウォールプロファイルをバインドして JSON SQL アクションを設定します。

JSON タイプのアプリケーションファイアウォールプロファイルの追加

最初に、アプリケーションファイアウォールが JSON Web コンテンツを JSON SQL インジェクション攻撃から保護する方法を指定するプロファイルを作成する必要があります。
コマンドプロンプトで入力します：

```
add appfw profile <name> -type (HTML | XML | JSON)
```

> 注記:
>
> プロファイルタイプを JSON に設定すると、HTML や XML などの他のチェックは適用されません。

例

```
add appfw profile profile1 -type JSON
```


JSON SQL インジェクションの設定

JSON SQL インジェクション攻撃からアプリケーションを保護するには、1 つ以上の JSON SQL インジェクションアクションを設定する必要があります。
コマンドプロンプトで入力します:

```
set appfw profile <name> - JSONSQLInjectionAction [block] [log] [stats] [none]
```

SQL インジェクションアクションは次のとおりです。
ブロック-このセキュリティー検査に違反する接続をブロックします。
Log - このセキュリティチェックの違反を記録します。
Stats-このセキュリティー検査の統計を生成します。
[なし]-このセキュリティー検査に対するすべてのアクションを無効にします。

JSON SQL インジェクションタイプ

アプリケーションファイアウォールプロファイルで JSON SQL Injection タイプを構成するには、コマンドプロンプトで次のように入力します。

```
set appfw profile <name> - JSONSQLInjectionType <JSONSQLInjectionType>
```

例

```
set appfw profile profile1 -JSONSQLInjectionType SQLKeyword
```

利用可能な SQL インジェクションタイプは、
利用可能な SQL インジェクションタイプです。
SQLSplChar. SQL 特殊文字
SQLKeyword をチェックします。SQL キーワードをチェックします。
SQLSplCharANDKeyword. ブロックが見つかった場合は、ブロックとブロックの両方をチェックします。
SQLSplCharORKeyword. . SQL 特殊文字または spl キーワードが見つかった場合にブロックします。
指定可能な値は sqlSPLChar、sqlKeyword、sqlSPLCharor キーワード、sqlSPLCharand キーワードです。

> 注:1
> つ以上のアクションを有効にするには、「set appfw プロファイル-jsonSQLInjectionAction」に続けて有効にするアクションを入力します。

例

```
set appfw profile profile1 -JSONSQLInjectionAction block log stat
```


次に、ペイロード、対応するログメッセージ、および統計カウンタの例を示します。

```
Payload:
=======
{

  "test": "data",
  "username": "waf",
  "password": "select * from t1;",
  "details": {

    "surname": "test",
    "age": "23"
   }

 }

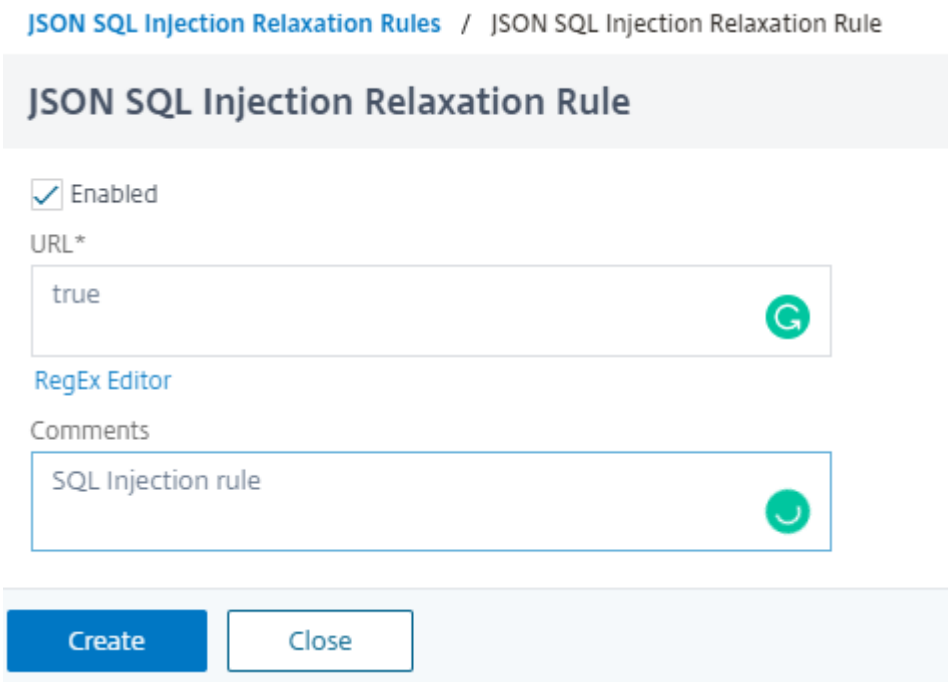

Log Message:
============
08/19/2019:08:49:46 GMT pegasus121 Informational 0-PPE-0 : default
   APPFW APPFW_JSON_SQL 6656 0 :  10.217.32.165 18402-PPE0 - profjson
   http://10.217.32.147/test.html SQL Keyword check failed for object
   value(with violation="select(;)") starting at offset(52) <blocked>
Counters:
========
   1  441083             1 as_viol_json_sql
   3       0             1 as_log_json_sql
   5       0             1 as_viol_json_sql_profile appfw__(profjson)
   7       0             1 as_log_json_sql_profile appfw__(profjson)
<!--NeedCopy-->
```

GUI を使用して **JSON SQL** インジェクション保護を設定する

JSON SQL インジェクション保護設定を設定するには、次の手順に従います。

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. 「プロファイル」ページで、「追加」をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、[詳細設定] の [セキュリティチェック] をクリックします。
4. [セキュリティチェック] セクションで、[**JSON SQL** インジェクション設定] に移動します。
5. チェックボックスの近くにある実行可能アイコンをクリックします。
6. [アクション設定] をクリックして、[**JSON SQL** インジェクション設定] ページにアクセスします。
7. **JSON SQL** インジェクションアクションを選択します。
8. [**OK**] をクリックします。

9. [**NetScaler Web App Firewall** プロファイル] ページで、[詳細設定] の [緩和ルール] をクリックします。
10. **[** 緩和ルール**]** セクションで、**[JSON SQL** インジェクション設定**]** を選択し、**[**編集**]** をクリックします。
11. [JSON SQL インジェクション緩和ルール] ページで、リクエストの送信先となる URL を入力します。この URL に送信されたすべてのリクエストはブロックされません。
12. [**Create**] をクリックします。



JSON SQL インジェクション保護のための細粒度緩和の設定

Web App Firewall には、JSON ベースの SQL インジェクションインスペクションチェックから特定の JSON キーまたは値を緩和するオプションがあります。細粒度緩和ルールを使用して JSON ペイロードを緩和する複数のオプションを設定できます。

以前は、JSON 保護チェックの緩和を設定する唯一の方法は URL 全体を指定することであり、URL 全体の検証はバイパスされていました。

JSON ベースの SQL セキュリティ保護により、次のことが緩和されます。

- キーネーム
- キーバリュー

JSON ベースの SQL 保護チェックでは、特定のパターンを許可し、残りをブロックする緩和を設定できます。たとえば、Web App Firewall には現在、100 を超える SQL キーワードのデフォルトセットがあります。ハッカーはこれらのキーワードを SQL インジェクション攻撃で使用できるため、Web App Firewall はすべてを潜在的な脅威としてフラグを立てます。特定の場所で安全と考えられる 1 つ以上のキーワードを緩和する場合は、セキュリティチェ

ックをバイパスして残りをブロックする緩和ルールを設定できます。リラクゼーションで使用されるコマンドには、値タイプと値表現のオプションパラメータがあります。値式が正規表現かリテラル文字列かを指定できます。値のタイプは空白のままにすることも、[キーワード] または [特殊文字列] を選択することもできます。

> 注:
>
> 正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加する URL を正確に定義していることを確認し、それ以外は何も定義しないでください。ワイルドカード、特にドットとアスタリスク (.*) のメタ文字やワイルドカードの組み合わせを不注意に使用すると、ブロックする意図のない Web コンテンツへのアクセスをブロックしたり、JSON SQL インジェクションチェックによってブロックされるような攻撃を許可したりするなど、望ましくない結果が生じる可能性があります。

考慮すべきポイント

- 値式はオプションの引数です。フィールド名には値式がない場合があります。
- 1 つのキー名を複数の値式にバインドできます。
- 値式には値型を割り当てる必要があります。値の型は、1) キーワード、2) SpecialString です。
- キー名または URL の組み合わせごとに複数の緩和ルールを設定できます。

コマンドインターフェイスを使用してコマンドインジェクション攻撃に対する **JSON** 細粒度緩和の設定

JSON ファイル粒度緩和ルールを構成するには、細粒度緩和エンティティを Web App Firewall プロファイルにバインドする必要があります。

コマンドプロンプトで入力します:

```
1 bind appfw profile <profile name> -jsoncmdURL <URL> -key <key name> -
     isregex <REGEX/NOTREGEX> -valueType <keyword/SpecialString> <value
     Expression> -isvalueRegex <REGEX/NOTREGEX>
2 <!--NeedCopy-->
```

例:

```
1 bind appfw profile appprofile1 -jsonsqlurl www.example.com -key
     stn_name  -isRegex NOTREGEX -valueType Keyword  “union”  -
     isvalueRegex NOTREGEX
2 <!--NeedCopy-->
```

GUI を使用して JSON ベースのコマンドインジェクション攻撃に対する細粒度緩和ルールを構成するには

1. [アプリケーションファイアウォール**] > [**プロファイル] に移動し、プロファイルを選択して [編集] をクリックします。
2. [詳細設定] ウィンドウで、[緩和規則] をクリックします。

3.「緩和ルール」セクションで、**JSON SQL** インジェクションレコードを選択し、「編集」をクリックします。

4. **JSON SQL** インジェクション緩和ルールスライダーで、「追加」をクリックします。

5. [**JSON SQL** インジェクション緩和規則] ページで、次のパラメータを設定します。

   a) 有効
   b) 名前は正規表現ですか
   c) キー名
   d) URL
   e) 値のタイプ
   f) コメント
   g) リソース ID

6.［**Create**］をクリックします。

JSON クロスサイトスクリプティング保護チェック

March 20, 2024

受信した JSON ペイロードに悪意のあるクロスサイトスクリプティングデータが含まれていると、WAF はリクエストをブロックします。次の手順では、CLI および GUI インターフェイスを使用してこれを設定する方法について説明します。

JSON クロスサイトスクリプティング保護の設定

JSON クロスサイトスクリプティング保護を設定するには、次の手順を完了する必要があります。

1. アプリケーションファイアウォールプロファイルを JSON として追加します。
2. JSON クロスサイトスクリプティングアクションを設定して、クロスサイトスクリプティングの悪意のあるペイロードをブロックする

JSON タイプのアプリケーションファイアウォールプロファイルの追加

最初に、アプリケーションファイアウォールが JSON Web コンテンツを JSON クロスサイトスクリプティング攻撃から保護する方法を指定するプロファイルを作成する必要があります。

コマンドプロンプトで入力します：

```
add appfw profile <name> -type (HTML | XML | JSON)
```


> 注記:
>
> プロファイルタイプを JSON に設定すると、HTML や XML などの他のチェックは適用されません。

例

```
add appfw profile profile1 -type JSON
```

JSON クロスサイトスクリプティング違反のサンプル出力

```
JSONcross-site scriptingAction: block log stats
Payload: {
 "username":"<a href="jAvAsCrIpT:alert(1)">X</a>","password":"xyz" }


Log message: Aug 19 06:57:33 <local0.info> 10.106.102.21
   08/19/2019:06:57:33 GMT  0-PPE-0 : default APPFW APPFW_JSON_cross-
   site scripting 58 0 :  10.102.1.98 12-PPE0 - profjson http://
   10.106.102.24/ Cross-site script check failed for object value(with
   violation="Bad URL: jAvAsCrIpT:alert(1)") starting at offset(12). <
   blocked>

Counters
   1  357000                  1 as_viol_json_xss
   3  0                       1 as_log_json_xss
   5  0                       1 as_viol_json_xss_profile appfw__(
      profjson)
   7  0                       1 as_log_json_xss_profile appfw__(
      profjson)

<!--NeedCopy-->
```

JSON クロスサイトスクリプティングの設定アクション

JSON クロスサイトスクリプティング攻撃からアプリケーションを保護するには、1 つ以上の JSON クロスサイトスクリプティングアクションを設定する必要があります。
コマンドプロンプトで入力します:

```
set appfw profile <name> - JSONcross-site scriptingAction [block] [
log] [stats] [none]
```

例

```
set appfw profile profile1 -JSONcross-site scriptingAction block
```

使用可能なクロスサイトスクリプティングアクションは、「
ブロック-このセキュリティチェックに違反する接続をブロックする」です。

Log - このセキュリティチェックの違反を記録します。
Stats-このセキュリティー検査の統計を生成します。
[なし]-このセキュリティー検査に対するすべてのアクションを無効にします。

> 注:1
> つ以上のアクションを有効にするには、「set appfw プロファイル-jsonCross-Site ScriptingAction」に続けて有効にするアクションを入力します。

例

```
set appfw profile profile1 -JSONSQLInjectionAction block log stat
```

GUI を使用して JSON クロスサイトスクリプティング (クロスサイトスクリプティング) 保護を設定する

以下の手順に従って、クロスサイトスクリプティング (クロスサイトスクリプティング) 保護を設定します。

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. 「プロファイル」ページで、「追加」をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、[詳細設定] の [セキュリティチェック] をクリックします。
4. [セキュリティチェック] セクションで、[**JSON** クロスサイトスクリプティング **(**クロスサイトスクリプティング**)** 設定] に移動します。
5. チェックボックスの近くにある実行可能アイコンをクリックします。
6. [アクション設定] をクリックして、[**JSON** クロスサイトスクリプティング設定**]** ページにアクセスします。
7. JSON クロスサイトスクリプティングアクションを選択します。
8. [**OK**] をクリックします。
9. [**NetScaler Web App Firewall** プロファイル] ページで、[詳細設定] の [緩和ルール] をクリックします。
10. [緩和ルール] セクションで [JSON クロスサイトスクリプティング設定] を選択し、[編集] をクリックします。
11. [**JSON** クロスサイトスクリプティング緩和ルール] ページで、[追加] をクリックして JSON クロスサイトスクリプティング緩和ルールを追加します。
12. リクエストの送信先の URL を入力します。この URL に送信されたすべてのリクエストはブロックされません。
13. [**Create**] をクリックします。

JSON ベースのクロスサイトスクリプティングのためのきめ細かい緩和の構成

Web App Firewall には、JSON ベースのクロスサイトスクリプティング (XSS) インスペクションチェックから特定の JSON キーまたは値を緩和するオプションがあります。細粒度緩和ルールを使用して JSON ペイロードを緩和する複数のオプションを設定できます。
以前は、JSON 保護チェックの緩和を設定する唯一の方法は URL 全体を指定することであり、URL 全体の検証はバ

イパスされていました。
JSON ベースの SQL セキュリティ保護により、次のことが緩和されます。

- キーネーム
- キーバリュー

JSON ベースのクロスサイトスクリプティング (XSS) 保護により、特定のパターンを許可し、残りをブロックする緩和を構成できます。たとえば、Web App Firewall には現在、100 を超える SQL キーワードのデフォルトセットがあります。ハッカーはこれらのキーワードを SQL インジェクション攻撃で使用できるため、Web App Firewall はすべてを潜在的な脅威としてフラグを立てます。特定の場所で安全と考えられる 1 つ以上のキーワードを緩和する場合は、セキュリティチェックをバイパスして残りをブロックする緩和ルールを設定できます。リラクゼーションで使用されるコマンドには、値タイプと値表現のオプションパラメータがあります。値式が正規表現かリテラル文字列かを指定できます。値のタイプは空白のままにすることも、[キーワード] または [特殊文字列] を選択することもできます。

> 注:
>
> 正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加する URL を正確に定義していることを確認し、それ以外は何も定義しないでください。ワイルドカード、特にドットとアスタリスク (.*) のメタ文字やワイルドカードの組み合わせを不注意に使用すると、ブロックする意図のない Web コンテンツへのアクセスをブロックしたり、JSON SQL インジェクションチェックによってブロックされるような攻撃を許可したりするなど、望ましくない結果が生じる可能性があります。

考慮すべきポイント

- 値式はオプションの引数です。フィールド名には値式がない場合があります。
- 1 つのキー名を複数の値式にバインドできます。
- 値式には値型を割り当てる必要があります。値の型は、タグ、属性、パターンです。
- キー名と URL の組み合わせごとに複数の緩和ルールを設定できます。

コマンドインターフェイスを使用して、クロスサイトスクリプティング **(XSS)** インジェクション攻撃に対する **JSON** の細粒度緩和の設定

JSON ファイル粒度緩和ルールを構成するには、細粒度緩和エンティティを Web App Firewall プロファイルにバインドする必要があります。

コマンドプロンプトで入力します:

```
1 bind appfw profile <profile name> -jsonxssURL <URL> -key <key name> -
     isregex <REGEX/NOTREGEX> -valueType <keyword/SpecialString> <value
     Expression> -isvalueRegex <REGEX/NOTREGEX>
2 <!--NeedCopy-->
```


例:

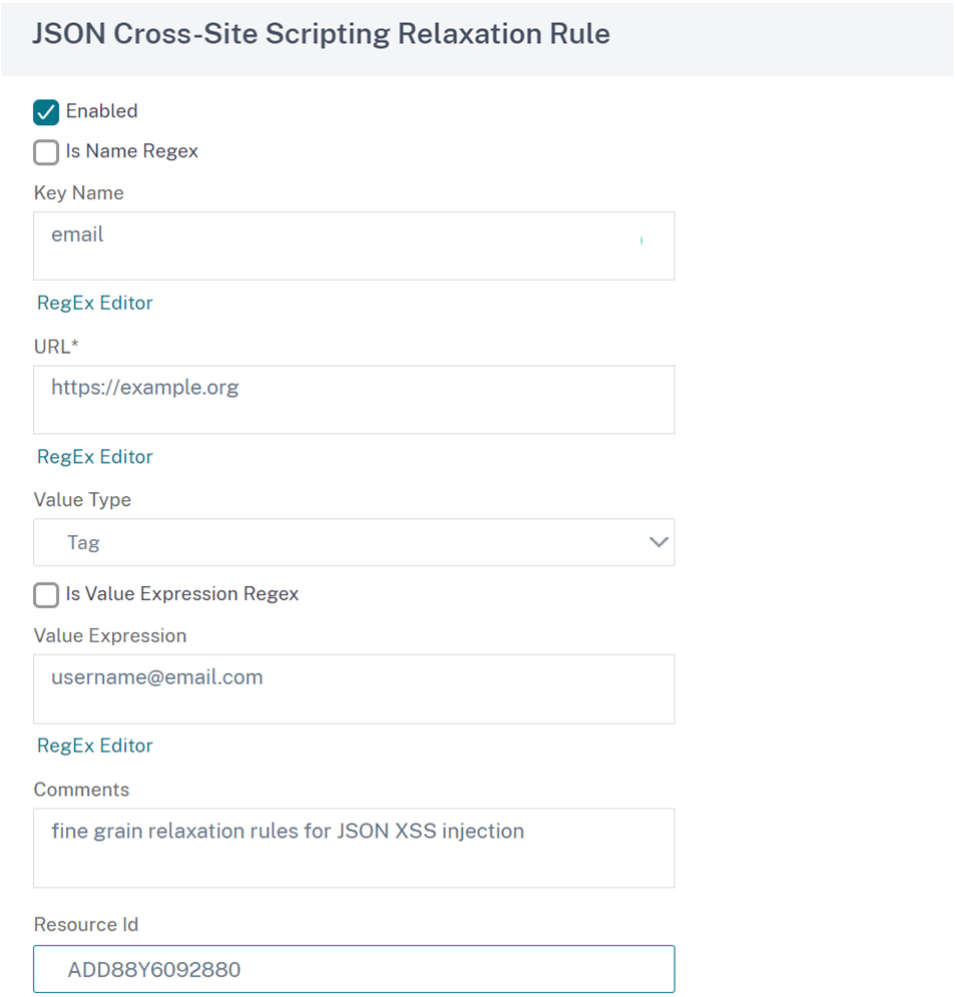
```
bind appfw profile appprofile1 -jsonxssurl www.example.com -key name  -
    isRegex NOTREGEX -valueType Tag  “sname”  -isvalueRegex NOTREGEX
<!--NeedCopy-->
```

GUI を使用して JSON ベースのクロスサイトスクリプティング (XSS) インジェクションの細粒度緩和ルールを構成するには

1. [アプリケーションファイアウォール**] > [**プロファイル] に移動し、プロファイルを選択して [編集] をクリックします。
2. [詳細設定] ウィンドウで、[緩和規則] をクリックします。
3. 「緩和ルール」セクションで、JSON SQL インジェクションレコードを選択し、「編集」をクリックします。
4. [**JSON** クロスサイトスクリプティング緩和規則] スライダーで、[追加] をクリックします。
5. [**JSON** クロスサイトスクリプティング緩和規則] ページで、次のパラメータを設定します。

    a) 有効
    b) 名前は正規表現ですか
    c) キー名
    d) URL
    e) 値のタイプ
    f) コメント
    g) リソース ID

6. [**Create**] をクリックします。



JSON コマンドインジェクション保護

March 20, 2024

JSON コマンドインジェクションチェックは、受信した JSON トラフィックを調べて、システムセキュリティを侵害したり、システムを変更したりする不正なコマンドがないか調べます。トラフィックを調べる際に、悪意のあるコマンドが検出されると、アプライアンスは要求をブロックするか、設定されたアクションを実行します。

コマンドインジェクション攻撃では、攻撃者は NetScaler オペレーティングシステムまたはバックエンドサーバーで不正なコマンドを実行しようとします。これを実現するために、攻撃者は脆弱なアプリケーションを使用してオペレーティングシステムコマンドを注入します。アプライアンスがセキュリティチェックを行わずに要求を転送するだけの場合、バックエンドアプリケーションはインジェクション攻撃に対して脆弱です。したがって、NetScaler アプライアンスが安全でないデータをブロックして Web アプリケーションを保護できるように、セキュリティチェックを構成することが非常に重要です。

コマンドインジェクション保護のしくみ

1. 受信した JSON リクエストに対して、WAF はトラフィックにキーワードや特殊文字がないか調べます。JSON リクエストに、拒否されたキーワードまたは特殊文字のいずれにも一致するパターンがない場合、リクエストは許可されます。それ以外の場合、要求は設定されたアクションに基づいてブロック、ドロップ、またはリダイレクトされます。
2. リストからキーワードまたは特殊文字を除外する場合は、特定の条件下でセキュリティチェックをバイパスする緩和ルールを作成できます。
3. ロギングを有効にすると、ログメッセージを生成できます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとしたことを示している可能性があります。
4. また、統計機能を有効にして、違反やログに関する統計データを収集することもできます。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当な要求がブロックされた場合は、新しい緩和ルールを構成する必要があるか、既存の緩和ルールを変更する必要があるかを再確認するために、構成を再確認する必要があります。

コマンドインジェクションチェックで拒否されたキーワードと特殊文字

JSON コマンドインジェクション攻撃を検出してブロックするために、アプライアンスではデフォルトのシグニチャファイルに一連のパターン（キーワードと特殊文字）が定義されています。コマンドインジェクションの検出中にブロックされるキーワードの一覧を次に示します。

```
<commandinjection>
      <keyword type="LITERAL" builtin="ON">7z</keyword>
      <keyword type="LITERAL" builtin="ON">7za</keyword>
      <keyword type="LITERAL" builtin="ON">7zr</keyword>
…
</commandinjection>

<!--NeedCopy-->
```

シグネチャファイルに定義されている特殊文字は次のとおりです。

```
| ; & $ > < '\ ! >> #
```

CLI を使用した JSON コマンドインジェクションチェックの設定

コマンドラインインターフェイスでは、set appfw profile コマンドを使用するか、appfw profile コマンドを追加して JSON コマンドインジェクション設定を構成できます。ブロック、ログ、統計の各アクションを有効にできます。また、ペイロードで検出するキーワードや文字列などのコマンドインジェクションタイプも設定する必要があります。

コマンドプロンプトで入力します：


```
set appfw profile <profile-name> -cmdInjectionAction <action-name> -
CMDInjectionType <CMDInjectionType>]
```

> 注:
>
> デフォルトでは、コマンドインジェクションアクションは「block log stats」に設定されています。また、デフォルトのコマンドインジェクションタイプはCmdSplCharANDKeyWordとして設定されています。アップグレード後、既存の Web App Firewall プロファイルのアクションは「なし」に設定されます。

例:

```
set appfw profile profile1 -JSONCMDInjectionAction block -JSONCMDInjectionType
 CmdSplChar
```

ここで、使用できる JSON コマンドインジェクションアクションは次のとおりです。

None-コマンドインジェクション保護を無効にします。
Log: セキュリティー検査のコマンドインジェクション違反をログに記録します。
Block-コマンドインジェクションセキュリティー検査に違反するトラフィックをブロックします。
Stats-コマンドインジェクションのセキュリティ違反に関する統計を生成します。

ここで、使用可能な JSON コマンドインジェクションタイプは次のとおりです。

Cmd SplChar -特殊文字をチェックする
CmdKeyWord -コマンドインジェクションをチェックするキーワード
CmdSplCharANDKeyWord -これはデフォルトのアクションです。アクションは特殊文字とコマンドインジェクションをチェックします。キーワードとブロックは、両方が存在する場合に限ります。
CmdSplCharORKeyWord -特殊文字とコマンドインジェクションキーワードとブロックのいずれかが見つかった場合にチェックします。

JSON コマンドインジェクション保護チェックのための緩和ルールの設定

アプリケーションでペイロード内の特定の ELEMENT または ATTRIBUTE に対する JSON コマンドインジェクションインスペクションをバイパスする必要がある場合は、緩和ルールを設定できます。

JSON コマンドのインジェクションインスペクション緩和規則の構文は次のとおりです。

```
bind appfw profile <profile name> -JSONCMDURL <expression> -comment
<string> -isAutoDeployed ( AUTODEPLOYED | NOTAUTODEPLOYED )-state (
ENABLED | DISABLED )
```

ヘッダーの正規表現の緩和ルールの例

```
bind appfw profile abc_json -jsoncmDURL http://1.1.1.1/hello.html
```


一方、以下では 1.1.1.1 でホストされているすべての URL からのリクエストが緩和されます。

```
bind appfw profile abc_json -jsoncmDURL http://1.1.1.1/*"
```

リラクゼーションを削除するには、’unbind’ を使います。

```
unbind appfw profile abc_json -jsoncmDURL  " http://1.1.1.1/*"
```

GUI を使用して JSON コマンドインジェクションチェックを設定する

JSON コマンドインジェクションチェックを設定するには、次の手順を実行します。

1. [セキュリティ] **> [NetScaler Web App Firewall** とプロファイル] に移動します。
2. [プロファイル] ページでプロファイルを選択し、[編集] をクリックします。
3. [**NetScaler Web App Firewall** プロファイル] ページで、[詳細設定] セクションに移動し、[セキュリティチェック] をクリックします。
4. **[** セキュリティチェック] セクションで、**[JSON** コマンドインジェクション] を選択し、**[** アクション] の設定をクリックします。
5. [**JSON** コマンドインジェクションの設定] ページで、次のパラメータを設定します。
   a) アクション。JSON コマンドインジェクションのセキュリティチェックに対して実行するアクションを 1 つ以上選択します。
   b) リクエストに含まれるものをチェックしてください。コマンドインジェクションパターンを選択して、受信リクエストにパターンがあるかどうかを確認します。
6. [**OK**] をクリックします。

コマンドインジェクショントラフィックおよび違反統計情報の表示

NetScaler Web App Firewall Statistics] ページには、セキュリティトラフィックとセキュリティ違反の詳細が表形式またはグラフ形式で表示されます。

コマンドインターフェイスを使用してセキュリティ統計情報を表示するには。

コマンドプロンプトで入力します:

```
stat appfw profile profile1
```

Appfw プロファイルのトラフィック統計	レート (/s)	合計
リクエスト	0	0
要求バイト数	0	0

Appfw プロファイルのトラフィック統計	レート (/s)	合計
レスポンス	0	0
送信バイト数	0	0
中止する	0	0
リダイレクト	0	0
長期平均応答時間 (ミリ秒)	–	0
最近の平均応答時間 (ミリ秒)	–	0

HTML/XML/JSON 違反の統計情報	レート (/s)	合計
開始 URL	0	0
URL を拒否する	0	0
リファラーヘッダー	0	0
バッファオーバーフロー	0	0
Cookie の整合性	0	0
Cookie ハイジャック	0	0
CSRF フォームタグ	0	0
HTML クロスサイトスクリプティング	0	0
HTML SQL インジェクション	0	0
フィールド形式	0	0
フィールドの一貫性	0	0
クレジットカード	0	0
セーフオブジェクト	0	0
シグネチャ違反	0	0
コンテンツの種類	0	0
JSON サービス拒否	0	0
JSON SQL インジェクション	0	0
JSON クロスサイトスクリプティング	0	0
ファイルアップロードの種類	0	0

HTML/XML/JSON 違反の統計情報	レート (/s)	合計
コンテンツタイプ XML ペイロードを推測	0	0
HTML CMD インジェクション	0	0
XML 形式	0	0
XML サービス拒否 (XDoS)	0	0
XML メッセージ検証	0	0
Web サービスの相互運用性	0	0
XML SQL インジェクション	0	0
XML クロスサイトスクリプティング	0	0
XML 添付ファイル	0	0
SOAP フォールト違反	0	0
XML ジェネリック違反	0	0
違反総数	0	0

HTML/XML/JSON ログ統計	レート (/s)	合計
URL ログの開始	0	0
URL ログの拒否	0	0
リファラーヘッダーログ	0	0
バッファオーバーフローログ	0	0
Cookie 整合性ログ	0	0
Cookie ハイジャックのログ	0	0
タグログからの CSRF	0	0
HTML クロスサイトスクリプティングログ	0	0
HTML クロスサイトスクリプティング変換ログ	0	0
HTML SQL インジェクションログ	0	0
HTML SQL 変換ログ	0	0
フィールド形式ログ	0	0

HTML/XML/JSON ログ統計	レート (/s)	合計
フィールド整合性ログ	0	0
クレジットカード	0	0
クレジットカード変換ログ	0	0
セーフオブジェクトログ	0	0
シグネチャログ	0	0
コンテンツタイプログ	0	0
JSON サービス拒否ログ	0	0
JSON SQL インジェクションログ	0	0
JSON クロスサイトスクリプティングログ	0	0
ファイルアップロードタイプログ	0	0
コンテンツタイプ XML ペイロードを推測 L	0	0
JSONCMD インジェクション	0	0
HTML コマンドインジェクションログ	0	0
XML 形式ログ	0	0
XML サービス拒否 (XDoS) ログ	0	0
XML メッセージ検証ログ	0	0
WSI ログ	0	0
XML SQL インジェクションログ	0	0
XML クロスサイトスクリプティングログ	0	0
XML 添付ファイルログ	0	0
SOAP フォールトログ	0	0
XML 汎用ログ	0	0
ログメッセージの総数	0	0

サーバエラーレスポンス統計レート (/s) | 合計 |
|–|-|-|
HTTP クライアントエラー (4xx Resp) | 0 | 0|
HTTP サーバエラー (5xx Resp) | 0 |

HTML/XML/JSON ログ統計	レート (/s)	合計
JSON コマンド注入ログ	0	0
XML 形式のログ	0	0

NetScaler GUI を使用した **JSON** コマンドインジェクションの統計情報の表示

コマンドインジェクションの統計情報を表示するには、次の手順を実行します:

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ペインで Web App Firewall プロファイルを選択し、[統計] をクリックします。
3. **NetScaler Web App Firewall** 統計ページには、JSON コマンドインジェクショントラフィックと違反の詳細が表示されます。
4. 表形式ビューを選択するか、グラフィカルビューに切り替えて、データを表形式またはグラフィカル形式で表示できます。

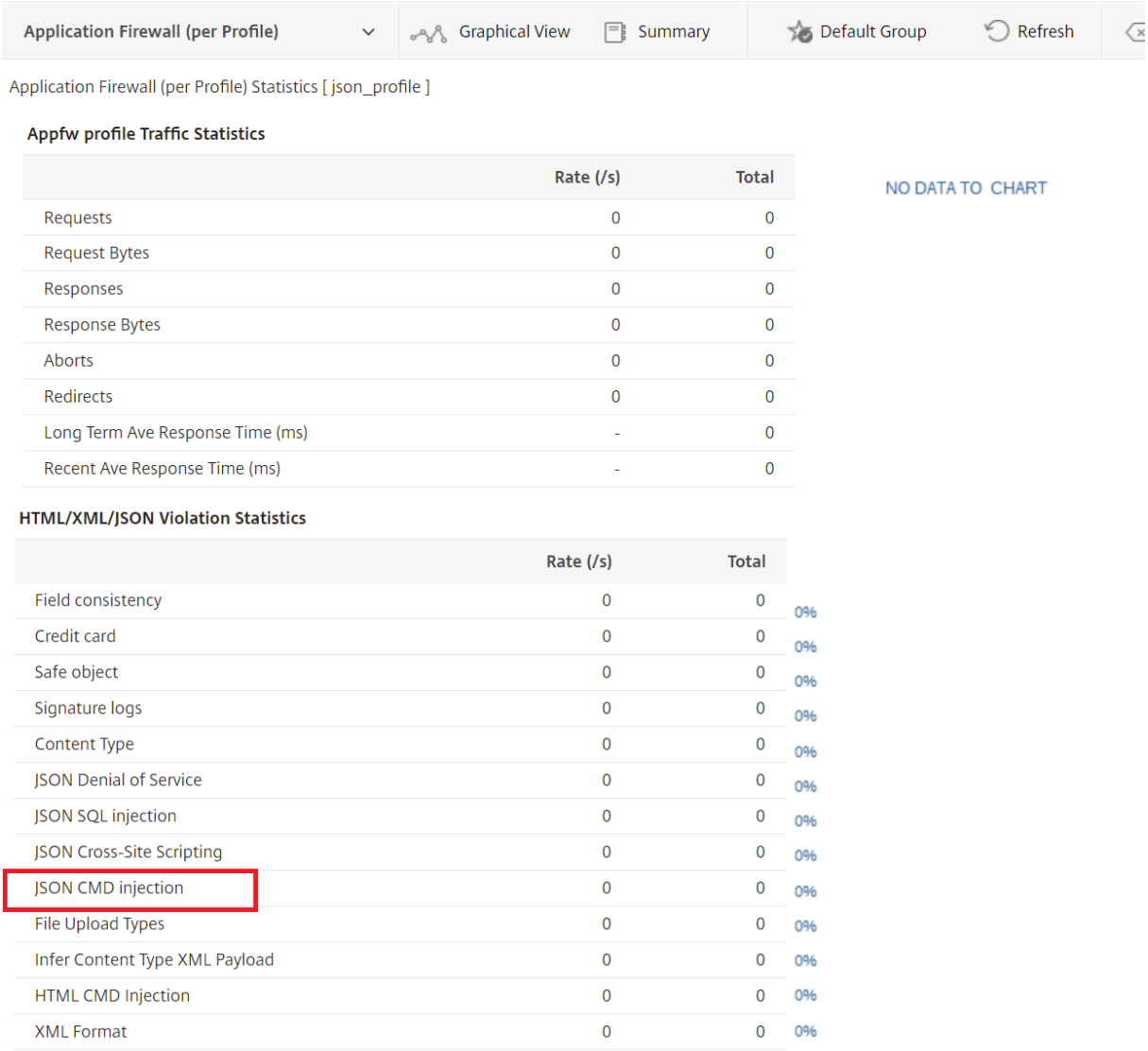
JSON コマンドのインジェクション

HTML/XML/JSON Log Statistics

	Rate (/s)	Total
Start URL logs	0	0
Deny URL logs	0	0
Field consistency logs	0	0
Credit cards	0	0
Credit card transform logs	0	0
Safe object logs	0	0
Signature logs	0	0
Content Type logs	0	0
JSON Denial of Service logs	0	0
JSON SQL injection logs	0	0
JSON Cross-Site Scripting logs	0	0
JSON CMD injection logs	0	0
File upload types logs	0	0
Infer Content Type XML Payload Logs	0	0

JSON CMD injection logs: X
Number of JSON Command Injection security check log messages generated by the Application Firewall.

JSON コマンド注入違反統計



JSON コマンドインジェクション用の微粒子緩和の設定

Web App Firewall には、JSON ベースのコマンドインジェクションチェックから特定の JSON キーまたは値を緩和するオプションがあります。細粒度緩和ルールを設定することで、1 つ以上のフィールドの検査を完全に回避できます。

以前は、JSON 保護チェックの緩和を設定する唯一の方法は URL 全体を指定することであり、URL 全体の検証はバイパスされていました。

JSON ベースのコマンドインジェクションセキュリティ保護により、次のことが緩和されます。

- キーネーム
- キーバリュー

JSON ベースのコマンドインジェクション保護により、特定のパターンを許可し、残りをブロックする緩和を設定できます。たとえば、Web App Firewall には現在、100 を超える SQL キーワードのデフォルトセットがあります。ハッカーはこれらのキーワードをコマンドインジェクション攻撃で使用できるため、Web App Firewall はすべてを潜在的な脅威としてフラグを立てます。特定の場所で安全と考えられる 1 つ以上のキーワードを緩和する場合は、セキュリティチェックをバイパスして残りをブロックする緩和ルールを設定できます。リラクゼーションで使用されるコマンドには、値タイプと値表現のオプションパラメータがあります。値式が正規表現かリテラル文字列かを指定できます。値のタイプは空白のままにすることも、[キーワード] または [特殊文字列] を選択することもできます。

> 注:
>
> 正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加する URL を正確に定義していることを確認し、それ以外は何も定義しないでください。ワイルドカード、特にドットとアスタリスク (.*) のメタ文字やワイルドカードの組み合わせを不注意に使用すると、ブロックする意図のない Web コンテンツへのアクセスをブロックしたり、JSON SQL インジェクションチェックによってブロックされるような攻撃を許可したりするなど、望ましくない結果が生じる可能性があります。

考慮すべきポイント

- 値式はオプションの引数です。フィールド名には値式がない場合があります。
- 1 つのキー名を複数の値式にバインドできます。
- 値式には値型を割り当てる必要があります。値の型は、1) キーワード、2) SpecialString です。
- キー名と URL の組み合わせごとに複数の緩和ルールを設定できます。

コマンドインターフェイスを使用してコマンドインジェクション攻撃に対する **JSON** 細粒度緩和の設定

JSON ファイル粒度緩和ルールを構成するには、細粒度緩和エンティティを Web App Firewall プロファイルにバインドする必要があります。

コマンドプロンプトで入力します:

```
1 bind appfw profile <profile name> -jsoncmdURL <URL> -key <key name> -
      valueType <keyword/SpecialString> <value Expression>
2 <!--NeedCopy-->
```

例:

```
bind appfw profile appprofile1 -jsoncmdurl www.example.com -key
blg_cnt -isRegex NOTREGEX -valueType Keyword “cat” -isvalueRegex
NOTREGEX
```

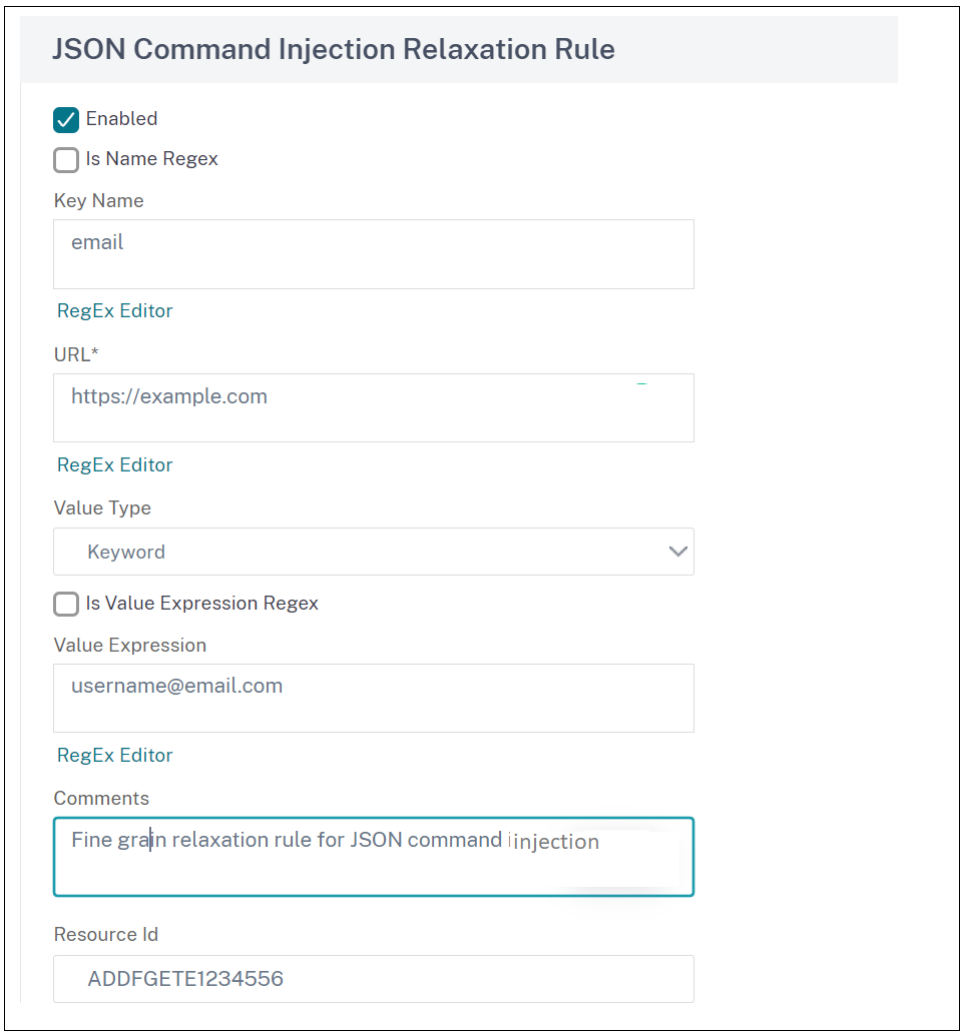
GUI を使用して JSON ベースのコマンドインジェクション攻撃に対する細粒度緩和ルールを構成するには

1. [アプリケーションファイアウォール] **>** [プロファイル] に移動し、プロファイルを選択して [編集] をクリックします。

2. [詳細設定] ウィンドウで、[緩和規則] をクリックします。
3. [緩和ルール] セクションで、**JSON** コマンドインジェクションレコードを選択し、[編集] をクリックします。
4. **JSON** コマンドインジェクション緩和規則スライダーで、「追加」をクリックします。
5. **[JSON** コマンドインジェクション緩和規則] ページで、次のパラメータを設定します。

    a) 有効
    b) 名前は正規表現ですか
    c) キー名
    d) URL
    e) 値のタイプ
    f) コメント
    g) リソース ID

6. [**Create**] をクリックします。



コンテンツタイプの管理

August 15, 2023

Web サーバーは、コンテンツタイプごとに MIME/タイプ定義を含む Content-Type ヘッダーを追加します。Web サーバーは、さまざまな種類のコンテンツを提供します。たとえば、標準 HTML には「text/html」の MIME タイプが割り当てられます。JPG 画像には、「画像/JPEG」または「画像/jpg」コンテンツタイプが割り当てられます。通常のウェブサーバーは、割り当てられた MIME/Type によって Content Type ヘッダーで定義されているさまざまなタイプのコンテンツを提供できます。

Web App Firewall のフィルタリングルールの多くは、特定のコンテンツタイプをフィルタリングするように設計されています。フィルタールールは HTML などの 1 種類のコンテンツに適用され、別の種類のコンテンツ (画像など) をフィルタリングする場合には不適切な場合がよくあります。そのため、Web App Firewall は、リクエストとレスポンスのコンテンツタイプをフィルタリングする前に、そのコンテンツタイプを特定しようとします。Web サーバーまたはブラウザが要求または応答に Content-Type ヘッダーを追加しない場合、Web App Firewall はデフォルトのコンテンツタイプを適用し、それに応じてコンテンツをフィルタリングします。

デフォルトのコンテンツタイプは通常「application/octet-stream」で、MIME/タイプ定義が最も一般的です。MIME/タイプは、ウェブサーバーが配信する可能性のあるあらゆるコンテンツタイプに適しています。ただし、Web App Firewall が適切なフィルタリングを選択するための情報はあまり提供されません。保護された Web サーバーが正確なコンテンツタイプヘッダーを追加するように設定されている場合は、その Web サーバーのプロファイルを作成し、それにデフォルトのコンテンツタイプを割り当てることができます。これは、フィルタリングの速度と精度の両方を向上させるためです。

特定のプロファイルに許可されるリクエストコンテンツタイプのリストを設定することもできます。この機能を設定すると、Web App Firewall が許可されているコンテンツタイプのいずれにも一致しないリクエストをフィルタリングすると、そのリクエストはブロックされます。

リクエストは必ず「application/x-www-form-urlencoded」、「multipart/form-data」、または「text/x-gwt-rpc」タイプのいずれかでなければなりません。Web App Firewall は、他のコンテンツタイプが指定されたリクエストをすべてブロックします。

> 注
>
> 許可される応答コンテンツタイプリストに「application/x-www-form-urlencoded」または「multipart/form-data」コンテンツタイプを含めることはできません。

コマンドラインインターフェイスを使用してデフォルトのリクエストコンテンツタイプを設定するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw profile <name> -requestContentType <type>`
- `save ns config`

例

次の例では、「text/html」コンテンツタイプを指定されたプロファイルのデフォルトとして設定しています。

```
set appfw profile profile1 -requestContentType "text/html"
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してユーザー定義のデフォルトリクエストコンテンツタイプを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `unset appfw profile <name> -requestContentType <type>`
- `save ns config`

例

次の例では、指定されたプロファイルのデフォルトのコンテンツタイプである「text/html」の設定を解除して、タイプを「application/octet-stream」に戻せるようにします。

```
unset appfw profile profile1 -requestContentType "text/html"
save ns config
<!--NeedCopy-->
```

> 注
>
> 処理には常に最後のコンテンツタイプヘッダーを使用し、バックエンドサーバーが 1 つのコンテンツタイプのみのリクエストを確実に受け取れるように残りのコンテンツタイプヘッダーがある場合は削除してください。
>
> バイパスされる可能性のあるリクエストをブロックするには、ルールを *HTTP.REQ.HEADER (「content-type」) .COUNT.GT (1)* 、プロファイルを *appfw_block* とする *Web App Firewall* ポリシーを追加します。
>
> Content-Type ヘッダーなしでリクエストが受信された場合、またはリクエストに Content-Type ヘッダーに値がない場合、Web App Firewall は設定された **RequestContentType** 値を適用し、それに応じてリクエストを処理します。

コマンドラインインターフェイスを使用してデフォルトの応答コンテンツタイプを設定するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw profile <name> -responseContentType <type>`
- `save ns config`

例

次の例では、「text/html」コンテンツタイプを指定されたプロファイルのデフォルトとして設定しています。

```
set appfw profile profile1 -responseContentType "text/html"
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してユーザー定義のデフォルト応答コンテンツタイプを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `unset appfw profile <name> -responseContentType <type>`
- `save ns config`

例

次の例では、指定されたプロファイルのデフォルトのコンテンツタイプである「text/html」の設定を解除して、タイプを「application/octet-stream」に戻せるようにします。

```
unset appfw profile profile1 -responseContentType "text/html"
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してコンテンツタイプを許可コンテンツタイプリストに追加するには

コマンドプロンプトで、次のコマンドを入力します。

- `bind appfw profile <name> -ContentType <contentTypeName>`
- `save ns config`

例

次の例では、指定したプロファイルの許可コンテンツタイプリストに「text/shtml」コンテンツタイプを追加します。

```
bind appfw profile profile1 -contentType "text/shtml"
save ns config
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して、許可されているコンテンツタイプリストからコンテンツタイプを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `unbind appfw profile <name> -ContentType <contentTypeName>`
- `save ns config`

例

次の例では、指定されたプロファイルの許可コンテンツタイプリストから「text/shtml」コンテンツタイプを削除します。

```
unbind appfw profile profile1 -contentType "text/shtml"
save ns config
<!--NeedCopy-->
```

URL エンコードされたコンテンツタイプとマルチパートフォームのコンテンツタイプを管理

NetScaler Web App Firewall では、フォームの URL エンコードおよびマルチパートフォームのコンテンツタイプを構成できるようになりました。コンテンツタイプの設定は XML および JSON リストに似ています。構成に基づいて、Web App Firewall はリクエストを分類し、URL エンコードされたコンテンツタイプやマルチパートフォームのコンテンツタイプがないかを検査します。

Web App Firewall プロファイルを Urlencoded コンテンツタイプと Multipart-Form コンテンツタイプで構成するには、
コマンドプロンプトで次のように入力します。

```
bind appfw profile p2 -contentType <string>
```

例:

```
bind appfw profile p2 -contentType UrlencodedFormContentType

bind appfw profile p2 -ContentType appfwmultipartform
```

GUI を使用してデフォルトコンテンツタイプと許可コンテンツタイプを管理するには

1. セキュリティ > **Web App Firewall** > プロファイルに移動します。
2. 詳細ウィンドウで、構成するプロファイルを選択し、[編集] をクリックします。「**Web App Firewall** プロファイルの設定」ダイアログ・ボックスが表示されます。
3. 「**Web App Firewall** プロファイルの設定」ダイアログ・ボックスで、「設定」タブをクリックします。
4. [設定] タブで、[コンテンツタイプ] 領域まで半分ほど下にスクロールします。

5. コンテンツタイプエリアで、デフォルトのリクエストまたはレスポンスのコンテンツタイプを設定します。
   - デフォルトのリクエストコンテンツタイプを設定するには、使用するコンテンツタイプの MIME/タイプ定義をデフォルトリクエストテキストボックスに入力します。
   - デフォルトのレスポンスコンテンツタイプを設定するには、使用するコンテンツタイプの MIME/タイプ定義をデフォルトレスポンステキストボックスに入力します。
   - 新しい許可コンテンツタイプを作成するには、「追加」をクリックします。「許可されたコンテンツタイプの追加」ダイアログ・ボックスが表示されます。
   - 既存の許可されているコンテンツタイプを編集するには、そのコンテンツタイプを選択し、「開く」をクリックします。「許可されたコンテンツタイプの変更」ダイアログ・ボックスが表示されます。
6. 許可されるコンテンツタイプを管理するには、「許可されたコンテンツタイプを管理」をクリックします。
7. 新しいコンテンツタイプを追加するか、既存のコンテンツタイプを変更するには、[追加] または [開く] をクリックし、[許可されたコンテンツタイプの追加] または [許可されたコンテンツタイプの変更] ダイアログボックスで、次の手順を実行します。
   a) 「有効」チェック・ボックスを選択または選択解除して、コンテンツ・タイプを許可するコンテンツ・タイプのリストに含めたり、リストから除外したりします。
   b) コンテンツタイプテキストボックスに、追加するコンテンツタイプを説明する正規表現を入力するか、既存のコンテンツタイプ正規表現を変更します。
      コンテンツタイプは、MIME タイプの説明とまったく同じようにフォーマットされます。
      > 注:
      > 許可されるコンテンツタイプリストには、任意の有効な MIME タイプを含めることができます。多くの種類のドキュメントにはアクティブなコンテンツが含まれている可能性があり、したがって悪意のあるコンテンツが含まれている可能性があるため、このリストに MIME タイプを追加する場合は注意が必要です。
   c) この特定の MIME タイプを許可コンテンツタイプリストに追加した理由を説明する簡単な説明を入力してください。
   d) 「作成」または「**OK**」をクリックして変更を保存します。
8. 「閉じる」をクリックして「許可されたコンテンツタイプの管理」ダイアログ・ボックスを閉じ、「設定」タブに戻ります。
9. [**OK**] をクリックして変更を保存します。

NetScaler GUI を使用して **URL** エンコードおよびマルチパートフォームのコンテンツタイプを管理するには

1. セキュリティ > **Web App Firewall** > プロファイルに移動します。
2. 詳細ウィンドウで、構成するプロファイルを選択し、[編集] をクリックします。

3.「**Web App Firewall** プロファイルの設定」ページで、「****** 詳細設定」セクションの「プロファイル設定 ****** 」を選択します。

4.「検査済みコンテンツタイプ」セクションで、次のパラメータを設定します。

   a) アプリケーション/x-www-フォーム-URL エンコード。チェックボックスを選択して、URL エンコードされたコンテンツタイプを検査します。
   b) マルチパート/フォームデータ。チェックを選択すると、マルチパートフォームのコンテンツタイプが検査されます。

5.［**OK**］をクリックします。

プロファイル

August 15, 2023

プロファイルは、特定の種類の Web コンテンツまたは Web サイトの特定の部分を保護するために使用されるセキュリティ設定の集まりです。プロファイルでは、Web App Firewall が各フィルタ (またはチェック) を Web サイトへのリクエストとそれらからの応答にどのように適用するかを決定します。Web App Firewall は、2 種類のプロファイルをサポートします。4 つの組み込み (デフォルト) プロファイルは、追加の構成を必要としない 4 つの組み込み (デフォルト) プロファイルで、もう 1 つは追加構成が必要なユーザー定義プロファイルです。

組み込みプロファイル

4 つの Web App Firewall ビルトインプロファイルは、保護を必要としない、またはユーザーが直接アクセスしてはならないアプリケーションやウェブサイトを簡単に保護します。これらのプロファイルタイプは次のとおりです。

- **APPFW_BYPASS.** Web App Firewall のフィルタリングをすべてスキップし、変更されていないトラフィックを保護されたアプリケーション、Web サイト、またはクライアントに送信します。
- **APPFW_RESET**。接続をリセットします。クライアントは、指定された開始ページにアクセスしてセッションを再確立する必要があります。
- **APPFW_DROP**。保護されたアプリケーションまたは Web サイトとの間のトラフィックをすべてドロップし、クライアントにはいかなる種類の応答も送信しません。
- **APPFW_BLOCK**。保護されたアプリケーションまたは Web サイトとの間のトラフィックをブロックします。

組み込みプロファイルは、ユーザー定義プロファイルとまったく同じように使用します。プロファイルを適用するトラフィックを選択するポリシーを設定し、そのプロファイルをポリシーに関連付ける必要があります。組み込みポリシーを設定する必要がないため、特定のタイプのトラフィックや特定のアプリケーションや Web サイトに送信されるトラフィックをすばやく許可またはブロックできます。

ユーザー定義プロファイル

ユーザー定義プロファイルは、ユーザーによって作成および設定されるプロファイルです。デフォルトプロファイルとは異なり、保護対象アプリケーションとの間のトラフィックをフィルタリングするには、ユーザー定義プロファイルを設定する必要があります。

ユーザー定義プロファイルには次の 3 種類があります。

- **HTML**。HTML ベースの Web ページを保護します。
- **XML**。XML ベースの Web サービスおよび Web サイトを保護します。
- **Web 2.0**。ATOM フィード、ブログ、RSS フィードなど、HTML コンテンツと XML コンテンツを組み合わせた Web 2.0 コンテンツを保護します。

Web App Firewall には多数のセキュリティチェックがあり、これらはすべて有効または無効にでき、各プロファイルでさまざまな方法で構成できます。各プロファイルには、さまざまなタイプのコンテンツの処理方法を制御する多数の設定もあります。最後に、すべてのセキュリティチェックを手動で設定するのではなく、学習機能を有効にして設定できます。この機能は、保護されている Web サイトへの通常のトラフィックを一定期間監視し、その観測結果に基づいて、一部のセキュリティチェックの推奨例外 (緩和) と、他のセキュリティチェックの追加ルールを、カスタマイズしたリストを提供します。

初期構成時には、Web App Firewall Wizard を使用するか手動を使用するかにかかわらず、通常、1 つの汎用プロファイルを作成して、より具体的なプロファイルの対象とならない Web サイト上のすべてのコンテンツを保護します。その後、特定のプロファイルをいくつでも作成して、より専門的なコンテンツを保護できます。

プロファイルペインは、以下の要素を含むテーブルで構成されています。

[名前**]**。アプライアンスに設定されているすべての Web App Firewall プロファイルを表示します。

バインドされた署名。前の列のプロファイルにバインドされているシグニチャオブジェクトが表示されます（存在する場合）。

ポリシー。その行の左端の列にプロファイルを呼び出す Web App Firewall ポリシーが表示されます（存在する場合）。

[コメント**]**。プロファイルに関連するコメントがあれば、その行の左端の列に表示します。

プロファイルタイプ。プロファイルのタイプが表示されます。タイプには、ビルトイン、HTML、XML、および Web 2.0 があります。

表の上には、プロファイルの作成、設定、削除、および情報の表示を行うためのボタンとドロップダウンリストが並んでいます。

- **Add** リストに新しいプロファイルを追加します。
- **[**編集**]**。選択したプロファイルを編集します。
- **[**削除**]**。選択したプロファイルをリストから削除します。
- 統計情報。選択したプロファイルの統計を表示します。

- アクション。追加のコマンドを含むドロップダウンリスト。現在、別の Web App Firewall 設定からエクスポートされたプロファイルをインポートできます。

Web App Firewall プロファイルの作成

March 20, 2024

Web App Firewall プロファイルは、コマンドラインを使用する方法と GUI を使用する方法の 2 つの方法のいずれかで作成できます。コマンドラインを使用してプロファイルを作成するには、コマンドラインでオプションを指定する必要があります。このプロセスは、プロファイルの設定と同様であり、いくつかの例外を除いて、2 つのコマンドが同じパラメータを取ります。

> 注
>
> : コアプロファイル: このプロファイルはビルド 33.x 以降で使用できます。制限はあるが基本的なセキュリティチェックがデフォルトで有効になっているのに対し、基本プロファイルと詳細プロファイルでは他にも多くのセキュリティチェックがデフォルトで有効になっています。コアプロファイルには次のセキュリティチェックが含まれています。
>
> - 文法ベースの SQL インジェクション
> - 文法ベースの CMD インジェクション
> - クロスサイトスクリプティング
> - バッファオーバーフロー
> - ブロックキーワード
>
> **CVE** プロファイル: このプロファイルは、ビルド 42.x 以降で使用できます。このプロファイルは、署名の追加とバインドにのみ使用してください。これにより、CVE チェックを除く Citrix Web App Firewall からのすべてのチェックが無効になります。

プロファイルを作成するときは、ベーシック、アドバンス、コア、CVE のいずれかのオプションを指定します。そのプロファイルに含まれるさまざまなセキュリティチェックと設定のデフォルト構成が適用されます。オプションで、コメントを追加することもできます。プロファイルを作成したら、データペインでプロファイルを選択し、[編集] をクリックしてプロファイルを構成する必要があります。

学習機能を使用するか、多くの高度な保護を有効にして構成する場合は、高度なデフォルトを選択する必要があります。特に、SQL インジェクションチェック、クロスサイトスクリプティングチェック、Web フォーム攻撃に対する保護を提供するチェック、または Cookie の一貫性チェックのいずれかを構成する場合は、学習機能の使用を計画する必要があります。これらのチェックを設定するときに、保護された Web サイトの適切な例外を含めない限り、正当なトラフィックをブロックする可能性があります。あまりにも広範囲な例外を作成せずにすべての例外を予測することは困難です。学習機能を使用すると、このタスクがはるかに簡単になります。それ以外の場合、基本的なデフォルト設定は迅速で、Web アプリケーションが必要とする保護を提供する必要があります。

次の 3 つのプロファイルタイプがあります。

- **HTML**。標準的な HTML ベースの Web サイトを保護します。
- **XML**。XML ベースの Web サービスおよび Web サイトを保護します。
- **Web 2.0 (HTML XML)**。ATOM フィード、ブログ、RSS フィードなど、HTML 要素と XML 要素の両方を含む Web サイトを保護します。

また、プロフィールに付けることができる名前にはいくつかの制限があります。プロファイル名は、NetScaler アプライアンスのどの機能でも他のプロファイルまたはアクションに割り当てられた名前と同じにすることはできません。特定のアクションまたはプロファイル名は、組み込みのアクションまたはプロファイルに割り当てられ、ユーザープロファイルには使用できません。許可されていない名前の完全なリストは、Web App Firewall プロファイルの補足情報を参照してください。アクションまたはプロファイルに既に使用されている名前のプロファイルを作成しようとすると、エラーメッセージが表示され、プロファイルは作成されません。

コマンドラインインターフェイスを使用して **Web App Firewall** プロファイルを作成するには

コマンドプロンプトで、次のコマンドを入力します:

- `add appfw profile <name> [-defaults ( basic | advanced | core | cve)]`
- `set appfw profile <name> -type ( HTML | XML | HTML XML )`
- `set appfw profile <name> -comment "<comment>"`
- `save ns config`

例

次の例では、基本デフォルトで、pr-basicという名前のプロファイルを追加し、プロファイルタイプとして HTML を割り当てます。これは、HTML Web サイトを保護するためのプロファイルの適切な初期設定です。

```
1 add appfw profile pr-basic -defaults basic -comment "Simple profile for
      websites."
2 set appfw profile pr-basic -type HTML
3 save ns config
4 <!--NeedCopy-->
```

GUI を使用して **Web App Firewall** プロファイルを作成するには

Web App Firewall プロファイルを作成するには、以下の手順を実行します。

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ペインで、[追加] をクリックします。

3. **[Web App Firewall** プロファイルの作成] ページで、次の基本パラメータを設定します。

   a) 名前
   b) プロファイルの種類
   c) コメント
   d) デフォルト
   e) 説明

4. [**OK**] をクリックします。
5. 作成したプロファイルを選択し、[編集] をクリックします。
6. [詳細設定**]** セクションで、次の設定を行います。

   a) セキュリティチェック
   b) プロファイル設定
   c) 動的プロファイリング
   d) リラクゼーションルール
   e) ルールを拒否する
   f) 学習ルール
   g) 拡張ロギング

7. 「セキュリティチェック」セクションで、セキュリティ保護を選択し、「アクション設定」をクリックします。
8. [セキュリティチェック] ページで、パラメータを設定します。

   > 注:
   >
   > [**Active Rule**] 設定は、**SQL** インジェクションチェックのリラクゼーションルールまたは拒否ルールをアクティブにする **HTML** SQL インジェクションチェックでのみ使用できます。詳細については、「緩和ルールと拒否ルール 」トピックを参照してください。

9. 「**OK**」をクリックして「閉じる」をクリックします。
10. [プロファイル設定**]** セクションで、プロファイルパラメータを設定します。詳細については、「Web App Firewall プロファイル設定の構成 」トピックを参照してください。
11. [動的プロファイリング] セクションで、セキュリティチェックを選択して動的プロファイル設定を追加します。詳細については、「動的プロファイル 」トピックを参照してください。
12. [緩和ルール] セクションで、[編集] をクリックして、セキュリティチェックの緩和ルールを追加します。詳細については、緩和ルールを参照してください 。
13. [拒否ルール] セクションで、HTML SQL インジェクションチェックの拒否ルールを追加します。詳細については、「HTML 拒否ルール 」トピックを参照してください。
14. [学習ルール] セクションで、学習設定を設定します。詳細については、「Web App Firewall 学習 」トピックを参照してください。

15. [拡張ログ] セクションで、[追加] をクリックして機密データをマスキングします。詳細については、「拡張ロギングのトピック」を参照してください。
16. [完了] をクリックし、[閉じる] をクリックします。

フェイクアカウント検出ルールを設定する

フェイクアカウントの作成は、実在の人物に関連付けられていない多数のユーザーアカウントを作成したり、本人の同意なしに実在の人物の詳細でユーザーアカウントを作成したりする自動化されたプロセスです。非正規ユーザーが作成した偽のアカウントは、個人の本人情報に対応しない登録情報を使用します。これらのアカウントは、フィッシング攻撃、フェイクニュースの拡散、スキャルピングなどの非合法的な目的でウェブアプリケーションが提供するサービスを悪用するために作成されます。ほとんどの場合、これらのアカウントは悪意のあるユーザーによって実行されるボットによって作成されます。

NetScaler アプライアンスは、偽のアカウント検出ルールを Web App Firewall プロファイルにバインドすることで、偽のアカウントを検出するように拡張されています。このルールは、フォーム URL と各 URL のフォームパラメータで構成されます。受信リクエストが、フェイクアカウント検出ルールに設定された式またはフォーム URL（サインアップページ）に一致する場合、疑わしいサインアップ試行に対して評価は true になり、リクエストデータは ADM サーバーに送信され、さらに検査されます。

コマンドインターフェイスを使用して偽アカウント検出を設定するには、次の手順を実行します。

1. フェイクアカウント検出機能を有効にする
2. フェイクアカウントルールをバインドする

フェイクアカウント検出機能を有効にする

コマンドプロンプトで入力します:

```
add/set appfw profile <name> -FakeAccountDetection ( ON | OFF )
```

例:

```
add appfw profile profile1 -FakeAccountDetection ON
```

フェイクアカウントルールをバインドする

コマンドプロンプトで入力します:

```
bind appfw profile <name> -FakeAccount (string|expression)isFieldNameRegex
 (ON|OFF)-tag <TagExpression> ([-formUrl <FormURL>]| [-formExpression
 <FormExpression>)])-state (ENABLED|DISABLED)
```

各項目の意味は次のとおりです。

- formUrl: HTTP フォームアクション URL。
  formExpression: 評価されるフォーム式。
- fakeaccount: フェイクアカウントの名前。
  tag: タグ式。
- isFieldNameRegex: fieldName が正規表現かどうかを指定します。デフォルト値はオフです。

例:

```
bind appfw profile profile1 -FakeAccount john -formURL  “/signup.php"
-tag "smith"
bind appfw profile profile2 -FakeAccount Will -formExpression "HTTP
.REQ.HEADER(\"Authorization\").CONTAINS(\"/test_accounts\").NOT &&
HTTP.REQ.URL.CONTAINS(\"/login.php\")"-fieldName -tag "smith"
```

example.com サインアップページの HTTP POST リクエストの入力例。

S.no	入力	例
1	サインアップ HTTP POST リクエストエンドポイント URL	https://webapi.example.com/account/api/v1.0/contacts/
2	HTTP POST リクエストの E メールフィールド名	メールアドレス
3	ファーストネーム HTTP POST リクエストのフィールド名	名
4	苗字 HTTP POST リクエストのフィールド名	苗字

GUI を使用して **Web App Firewall** のフェイクアカウント検出ルールを構成する

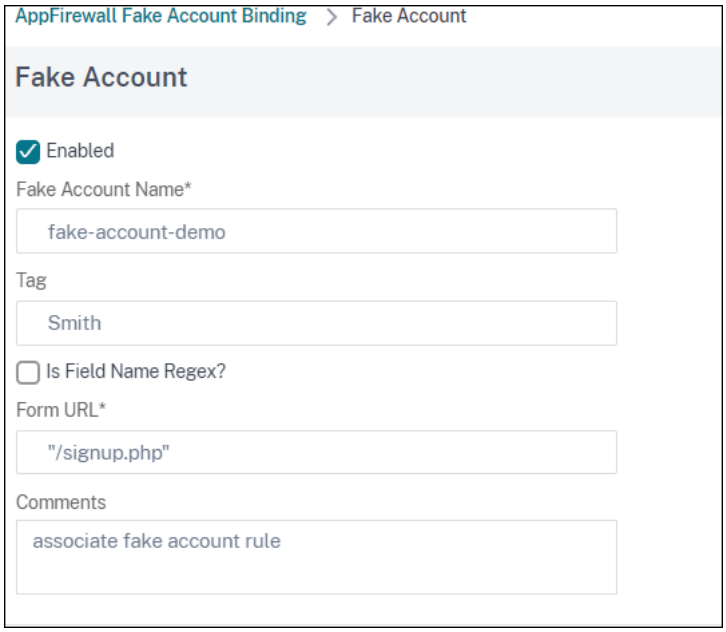
GUI を使用して偽アカウント検出ルールを設定するには、次の手順を実行します。

1. 構成 > セキュリティ > **NetScaler Web App Firewall** > プロファイルの順に移動します。
2. プロファイルを選択し、[編集] をクリックします。
3. [**NetScaler Web App Firewall** プロファイル] ページで、[詳細設定] の [セキュリティチェック] をクリックします。
4. **Citrix Cloud** と統合されたチェック] セクションで、偽のアカウントルールを選択し、[編集] をクリックします。
5. [**AppFirewall Fake Account Binding**] スライダーで、編集するルールを選択するか、[追加] をクリックします。

6. [**Fake Account rule**] ページで、次のパラメータを設定します。

   a) **Enabled**。フェイクアカウントルールを有効にする場合に選択します。
   b) 偽のアカウント名。フェイクアカウントルールの名前。
   c) タグ。偽のアカウント登録フォームの名前。
   d) フィールド名は正規表現ですか? フォームフィールドが正規表現である場合に選択します。
   e) フォーム表現。フェイクアカウントを定義する正規表現。
   f) フォーム **URL**。フェイクアカウント検出 URL を入力します。
   g) [コメント**]**。フェイクアカウント検出ルールに関する簡単な説明。

7. [**Create**] をクリックします。



HTTP RFC コンプライアンスを強制する

August 15, 2023

NetScaler Web App Firewall は、HTTP RFC 準拠の着信トラフィックを検査し、デフォルトで RFC 違反のある要求をドロップします。ただし、特定のシナリオがあり、アプライアンスが非 RFC コンプライアンス要求をバイパスまたはブロックする必要がある場合があります。このような場合、グローバルレベルまたはプロファイルレベルでこのような要求をバイパスまたはブロックするようにアプライアンスを設定できます。

グローバルレベルで非 **RFC** 準拠要求をブロックまたはバイパスする

HTTP モジュールは、リクエストが不完全で、WAF で処理できないリクエストが無効であると識別します。たとえば、受信 HTTP リクエストにホストヘッダーがないとします。このような無効な要求をブロックまたはバイパスする

には、アプリケーションファイアウォールのグローバル設定でmalformedReqActionオプションを構成する必要があります。

‘malformedReqAction’ パラメーターは、受信したリクエストのコンテンツ長が無効か、チャンクされたリクエストが無効か、HTTP バージョンがないか、ヘッダーが不完全かどうかを検証します。

> 注:
>
> malformedReqAction パラメータでブロックオプションを無効にすると、アプライアンスはすべての非 RFC コンプライアンス要求に対してアプリケーションファイアウォール処理全体をバイパスし、要求を次のモジュールに転送します。

コマンドラインインターフェイスを使用して無効な非 **RFC** 苦情 **HTTP** 要求をブロックまたはバイパスするには

無効な要求をブロックまたはバイパスするには、次のコマンドを入力します。

```
set appfw settings -malformedreqaction <action>
```

例**:**

```
set appfw settings -malformedReqAction block
```

不正なリクエストアクション設定を表示するには

不正なリクエストアクション設定を表示するには、次のコマンドを入力します。

```
show appfw settings
```

出力:

```
1 DefaultProfile:  APPFW_BYPASS UndefAction:  APPFW_BLOCK SessionTimeout:
      900     LearnRateLimit:  400     SessionLifetime:  0
    SessionCookieName:  citrix_ns_id ImportSizeLimit:  134217728
    SignatureAutoUpdate:  OFF SignatureUrl:"https://s3.amazonaws.com/
    NSAppFwSignatures/SignaturesMapping.xml" CookiePostEncryptPrefix:
    ENC GeoLocationLogging:  OFF CEFLogging:  OFF       EntityDecoding:
     OFF     UseConfigurableSecretKey:  OFF SessionLimit:  100000
    MalformedReqAction:  block log stats
2  Done
3 <!--NeedCopy-->
```

NetScaler GUI を使用して無効な非 **RFC** 苦情 **HTTP** リクエストをブロックまたはバイパスするには

1. [セキュリティ] **>** [**NetScaler Web App Firewall**] に移動します。
2. [**NetScaler Web App Firewall**] ページで、[設定] の下の [エンジン設定の変更] をクリックします。
3. [**NetScaler Web App Firewall** 設定の構成] ページで、[不正な形式の要求をログに記録する] オプションを選択します。[ブロック]、[ログ]、または [統計] を選択します。

4.「**OK**」をクリックして「閉じる」をクリックします。

> 注:
>
> ブロックアクションの選択を解除するか、不正な形式のリクエストアクションを選択しなかった場合、アプライアンスはユーザーを脅かすことなくリクエストをバイパスします。

プロファイルレベルで非 **RFC** 準拠要求をブロックまたはバイパスする

他の非 RFC 準拠要求は、プロファイルレベルでブロックまたはバイパスするように設定できます。RFC プロファイルは、ブロックモードまたはバイパスモードのいずれかで設定する必要があります。この設定を行うと、Web App Firewall プロファイルに一致する無効なトラフィックはバイパスされるか、それに応じてブロックされます。RFC プロファイルは、次のセキュリティチェックを検証します。

- 無効な GWT-RPC リクエスト
- 無効なコンテンツタイプヘッダー
- 無効なマルチパートリクエスト
- 無効な JSON リクエスト
- 重複する Cookie 名と値のペアチェック

> 注:
>
> RFC プロファイルを「バイパス」モードで設定する場合は、「**HTML** クロスサイトスクリプティング設定」および「HTML SQL インジェクション設定 **」セクションで変換オプションを無効にする必要があります。バイパスモードでRFCプロファイルを有効にして設定すると、アプライアンスには「クロスサイトスクリプトの変換」と「SQL 特殊文字の変換」の両方が現在オンになっているという警告メッセージが表示されます。APPFW_RFC_BYPASSと一緒に使用する場合は、オフにすることをお勧めします。
>
> 重要:
>
> また、アプライアンスには、「Appfw セキュリティチェックは、このプロファイルが設定されている場合、RFC チェックに違反する要求には適用されない可能性があります。RFC 違反を含むリクエストが部分的に変換される可能性があるため、変換設定を有効にすることは推奨されません。」

コマンドラインインターフェイスを使用して **Web App Firewall** プロファイルで **RFC** プロファイルを構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
set appfw profile <profile_name> -rfcprofile <rfcprofile_name
```

例

```
set appfw profile P1 -rfcprofile APPFW_RFC_BLOCK
```


> 注:
>
> デフォルトでは、RFC プロファイルはブロックモードの Web App Firewall プロファイルにバインドされます。

GUI を使用して **Web App Firewall** プロファイルで **RFC** プロファイルを構成するには

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. [プロファイル] ページでプロファイルを選択し、[編集] をクリックします。
3. **Web App Firewall** プロファイルページで、[詳細設定] セクションの **[**プロファイル設定**]** をクリックします。
4. [**HTML** 設定] セクションで、RFC プロファイルを APPFW_RFC_BYPASS mode に設定します。
   警告メッセージが表示されます。「Appfw Security Check enabled は、このプロファイルが設定されている場合、RFC チェックに違反する要求には適用できない可能性があります。RFC 違反を含むリクエストが部分的に変換される可能性があるため、変換設定を有効にすることはお勧めしません。」

Web App Firewall プロファイルの構成

March 20, 2024

ユーザー定義の Web App Firewall プロファイルを構成するには、まずセキュリティチェックを構成します。セキュリティチェックは、Web App Firewall ウィザードでは「ディーププロテクション **」または「高度な保護」と呼ばれます。一部のチェックでは、使用するためには設定が必要です。安全ではあるが範囲が限定された既定の設定があるものもあります。Web サイトでは、特定のセキュリティチェックのより多くの機能を利用する別の設定が必要か、その恩恵を受ける可能性があります。

セキュリティチェックを構成したら、1 つのセキュリティチェックではなく、Web App Firewall 機能の動作を制御するその他の設定も構成できます。ほとんどの Web サイトを保護するには既定の設定で十分ですが、保護された Web サイトに適しているかどうかを確認するには、これらを確認する必要があります。

> 注:
>
> プロファイル名の長さとすべてのインポートオブジェクト名の長さは、最大 127 文字まで設定できます。

Web App Firewall のセキュリティチェックの詳細については、「高度な保護」を参照してください。

コマンドラインを使用して **Web App Firewall** プロファイルを構成するには

コマンドプロンプトで、次のコマンドを入力します:

- `set appfw profile <name> <arg1> [<arg2> ...]`

  各項目の意味は次のとおりです:

  - <arg1> = パラメータと関連するオプション。
  - <arg2> = 2 番目のパラメータと関連するオプション。
  - …= 追加のパラメータとオプション。

  特定のセキュリティチェックを設定するときに使用するパラメータの詳細については、「高度な保護」を参照してください。

- `save ns config`

例

次の例は、pr-basicという名前のプロファイルで HTML SQL インジェクションと HTML クロスサイトスクリプティングチェックのブロックを有効にする方法を示しています。このコマンドは、プロファイルに他の変更を加えずにこれらのアクションをブロックできるようにします。

```
1 set appfw profile pr-basic -crossSiteScriptingAction block -
     SQLInjectionAction block
2 <!--NeedCopy-->
```

緩和ルールを **Web App Firewall** プロファイルにバインドする

Web App Firewall が違反を検出すると、ユーザーは緩和ルールによって適用されたアクションをバイパスできます。緩和ルールは、検出されたセキュリティ違反に適用される例外です。たとえば、開始 URL 緩和ルールは、強制的なブラウジングから保護します。ハッカーによって悪用される既知の Web サーバーの脆弱性は、デフォルトの URL の拒否ルールのセットを有効にすることで検出およびブロックできます。バッファオーバーフロー、SQL、クロスサイトスクリプティングなど、一般的に起動される攻撃も簡単に検出できます。

CLI を使用してセキュリティ免除ルールまたは緩和ルールをバインドするには

コマンドプロンプトで入力します:

```
1 bind appfw profile <name> ((-startURL <expression> [-resourceId  <
     string>]) | -denyURL <expression> | (-fieldConsistency <string>   <
     formActionURL> [-isRegex ( REGEX | NOTREGEX )]) | (-
     cookieConsistency <string> [-isRegex ( REGEX | NOTREGEX )]) | (-
     SQLInjection <string> <formActionURL> [-isRegex ( REGEX | NOTREGEX )
     ] [-location <location>] [-valueType <valueType> <valueExpression
     >....
2 <!--NeedCopy-->
```


GUI を使用してセキュリティ免除ルールまたは緩和ルールをバインドするには

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ペインで、プロファイルを選択し、[編集] をクリックします。
3. [**NetScaler Web App Firewall** プロファイル］ページで、［詳細設定］セクションの［緩和ルール］をクリックします。
4. [緩和ルール] セクションで、[**startURL**] をクリックし、[編集] をクリックします。
5. [開始 **URL** 緩和ルール] ページで、[追加] をクリックします。
6. [**URL** 緩和ルールの開始] ページで、次のパラメータを設定します。
   a) Enabled。緩和ルールを有効にするには、このチェックボックスをオンにします。
   b) 開始 URL。正規表現の値を入力します。
   c) [コメント]。緩和ルールについて簡単に説明してください。
7. **[**作成**]** して **[**閉じる**]** をクリックします。

GUI を使用して **Web App Firewall** プロファイルを構成するには

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ウィンドウで、構成するプロファイルを選択し、[編集] をクリックします。
3. [**Web App Firewall** プロファイルの構成] ダイアログボックスの [セキュリティチェック] タブで、セキュリティチェックを構成します。
   - チェックに対するアクションを有効または無効にするには、一覧でアクションのチェックボックスをオンまたはオフにします。
   - リスト内のセキュリティー検査のパラメータを構成するには、チェックボックスをオンにして [**Active Settings**] をクリックします。
   - 選択したセキュリティー検査のログエントリを確認するには、チェックボックスをオンにして [ログ] をクリックします。この情報を使用して、攻撃と一致するセキュリティチェックを決定し、セキュリティチェックのトラフィックをブロックできます。また、この情報を使用して、正規のトラフィックと一致するチェックを決定し、正当な接続を許可するように適切な除外を設定することもできます。ログの詳細については、ログ、統計、およびレポートを参照してください。
   - チェックを完全に無効にするには、リストで、そのチェックの右側にあるすべてのチェックボックスをオフにします。
4. [設定] タブで、プロファイル設定を構成します。
   - 以前に作成して設定したシグニチャのセットにプロファイルを関連付けるには、「共通設定」で、「シグネチャ」ドロップダウンリストからそのシグニチャセットを選択します。

> 注:
>
> [共通設定] セクションを表示するには、ダイアログボックスの右側にあるスクロールバーを使用して下にスクロールする必要があります。

- HTML または XML エラーオブジェクトを設定するには、該当するドロップダウンリストからオブジェクトを選択します。

  > 注:
  >
  > まず、[インポート] ペインで使用するエラーオブジェクトをアップロードする必要があります。エラーオブジェクトのインポートの詳細については、「インポート」を参照してください。

- デフォルトの XML コンテンツタイプを構成するには、コンテンツタイプの文字列を [既定の要求] および [既定の応答] テキストボックスに直接入力するか、[許可されたコンテンツタイプの管理] をクリックして許可されたコンテンツタイプのリストを管理します。» もっと…。

5. 学習機能を使用する場合は、「ラーニング」をクリックし、「ラーニング機能の構成と使用」の説明に従って、プロファイルの学習設定を構成します。
6. [**OK]** をクリックして変更を保存し、[プロファイル**]** ペインに戻ります。

WAF プロファイルの機密フィールド

> 注**:**
>
> この機能は、リリース 13.1 ビルド 27.x 以降で使用できます。

WAF プロファイルに機密項目を追加できるようになりました。これらのフィールドはマスクされ、違反が発生しても ADC ログに記録されません。以前は、設定のみを使用してこれらのフィールドを追加できました。設定を使用して機密フィールドを追加する方法の詳細については、「機密フィールド」を参照してください。

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. プロファイルを選択し、[編集] をクリックします。
3. [詳細設定**]** で、[機密フィールド] をクリックします。
4. [追加] をクリックします。
5. 次のパラメーターの値を入力します:

   - フォームフィールド名 *
   - アクション URL*
   - コメント

   A * は必須フィールドを示します

6. [**Create**] をクリックします。
7. [完了] をクリックします。

Web アプリケーションファイアウォールのプロファイル設定

August 15, 2023

次に、アプライアンス上で設定する必要があるプロファイル設定を示します。

コマンドプロンプトで入力します。

```
add appfw profile <name> [-invalidPercentHandling <invalidPercentHandling
>] [-checkRequestHeaders ( ON | OFF )] [-URLDecodeRequestCookies
 ( ON | OFF )] [-optimizePartialReqs ( ON | OFF )] [-errorURL <
expression>] [-logEveryPolicyHit ( ON | OFF )] [-stripHtmlComments <
stripHtmlComments>] [-stripXmlComments ( none | all )] [-postBodyLimitSignatur
 <positive_integer>][-fileUploadMaxNum <positive_integer>][-canonicalizeHTMLRe
 ( ON | OFF )][-percentDecodeRecursively ( ON | OFF )] [-multipleHeaderAction
 <multipleHeaderAction> ...][-inspectContentTypes <inspectContentTypes
> ...][-semicolonFieldSeparator ( ON | OFF )]
```

例**:**

```
add appfw profile profile1 [-invalidPercentHandling secure_mode] [-
checkRequestHeaders ON] [-URLDecodeRequestCookies OFF] [-optimizePartialReqs
 OFF]
```

各項目の意味は次のとおりです。

InvalidPercentHandling -パーセントエンコードされた名前と値を処理する方法を設定します。

使用可能な設定は次のように機能します。

asp_mode-解析のために無効なパーセントを取り除き、解析します。

例 `curl -v “http://<vip>/forms/login.html?field=sel%zzect -> Invalid percent encoded char(%zz)`が削除され、残りのコンテンツが検査され、SQLInjection チェックのアクションが実行されます。

secure_mode-無効なパーセントコード値を検出して無視します。

例: - `curl -v “http://<vip>/forms/login.html?field=sel%zzect -> Invalid percent encoded char(%zz)`が検出され、カウンタが増加し、コンテンツはそのままサーバに渡されます。

apache_mode-このモードはセキュアモードと同様に動作します。

> 注:
>
> リリース 13.1 ビルド 45.x 以降、`apache_mode` この関数は廃止されました。

指定できる値: アパッチモード、ASP モード、セキュアモード

デフォルト値: セキュアモード

OptimizePartialReqs -オフ/オン（セーフオブジェクトなし）の場合、NetScaler アプライアンスは部分的な要求をバックエンドサーバーに送信します。この部分的な応答はクライアントに送り返されます。OptimizePartialReqs は、セーフオブジェクトが構成されている場合に意味があります。アプライアンスは、オフのときはサーバーから完全な応答の要求を送信し、オンの場合は部分的な応答のみを要求します。

使用可能な設定は次のとおりです。

ON -クライアントによる部分的な要求は、バックエンドサーバーへの部分的な要求になります。
OFF-クライアントからの部分的な要求は、バックエンドサーバーへの完全な要求に変更されます。

設定可能な値:ON、OFF
デフォルト値:ON

URLDecodeRequestCookie。URL SQL およびクロスサイトスクリプティングチェックの対象にする前に、リクエスト Cookie をデコードします。

可能な値：ON、OFF
デフォルト値：OFF

署名投稿本文制限 **(**バイト**)**。場所が‘HTTP_POST_BODY’として指定された署名について検査されるリクエストペイロード (バイト単位) を制限します。

デフォルト値:8096
最小値:0
最大値:4294967295

ポストボディ制限 **(**バイト**)**。Web アプリケーションファイアウォールによって検査される要求ペイロード (バイト単位) を制限します。

デフォルト値：20000000
最小値：0
最大値：10 ギガバイト

セキュリティ設定とその GUI 手順の詳細については、「Web App Firewall プロファイルの構成 」トピックを参照してください。

ポストボディ制限アクション PostBodyLimit は、許可される HTTP 本文の最大サイズを指定した場合、エラー設定を使用します。エラー設定を適用するには、1 つ以上の Post Body Limit アクションを設定する必要があります。この設定は、転送エンコーディングヘッダーがチャンクされている要求にも適用されます。

```
set appfw profile <profile_name> -PostBodyLimitAction block log stats
```

Where,
Block-このアクションは、セキュリティチェックに違反する接続をブロックします。これは、設定された HTTP 本文の最大サイズ（本文後制限）に基づいています。このオプションは常に有効にする必要があります。

Log - このセキュリティチェックの違反を記録します。

Stats -このセキュリティー検査の統計を生成します。

> 注:
>
> ポストボディ制限アクションのログ形式は、
> 標準の監査ログ形式に従うように変更されました。次に例を示します。

```
ns.log.4.gz:Jun 25 1.1.1.1. <local0.info> 10.101.10.100 06/25/2020:10:10:28
 GMT 0-PPE-0 : default APPFW APPFW_POSTBODYLIMIT 1506 0 : <Netscaler
 IP> 4234-PPE0 - testprof ><URL> Request post body length(<Post
Body Length>)exceeds post body limit.
```

inspectQueryContentTypes 次のコンテンツタイプの注入された SQL スクリプトとクロスサイトスクリプトのリクエストクエリと Web フォームを検査します。

```
set appfw profile p1 -inspectQueryContentTypes HTML XML JSON OTHER
```

可能な値:HTML、XML、JSON、その他

デフォルトでは、このパラメーターは「inspectQueryContentTypes: HTML JSON その他」として設定され、基本アプリケーションプロファイルと高度なアプリ fw プロファイルの両方に対して設定されます。

XML としてクエリコンテンツタイプを検査する場合の例**:**

```
1 > set appfw profile p1 -type XML
2 Warning: HTML, JSON checks except “InspectQueryContentTypes” & “
     Infer Content-Type XML Payload Action” will not be applicable when
     profile type is not HTML or JSON respectively.
3 <!--NeedCopy-->
```

HTML としてインスペクションクエリコンテンツタイプの例**:**

```
1 > set appfw profile p1 -type HTML
2 Warning: XML, JSON checks except “InspectQueryContentTypes” & “Infer
     Content-Type XML Payload Action” will not be applicable when
     profile type is not XML or JSON respectively
3 Done
4 <!--NeedCopy-->
```

JSON としてクエリコンテンツタイプを検査する例**:**

```
1 > set appfw profile p1 -type JSON
2 Warning: HTML, XML checks except “InspectQueryContentTypes” & “Infer
     Content-Type XML Payload Action will not be applicable when profile
     type is not HTML or XML respectively
3 Done
4 <!--NeedCopy-->
```

ErrorURL 式です。NetScaler Web App Firewall がエラー URL として使用する URL。最大長:2047

> 注:
>
> 要求された URL のブロック違反の場合、エラー URL がシグニチャ URL に似ている場合、アプライアンスは接続をリセットします。

logeVeryPolicyHit -セキュリティチェックの結果に関係なく、すべてのプロファイルの一致をログに記録します。
指定可能な値: オン、オフ。
デフォルト値: オフ。

stripXmlComments -ユーザーのリクエストに応答して、保護された Web サイトから送信されたウェブページを転送する前に、XML コメントを削除します。
可能な値: なし、すべて、exclude_script_tag。
デフォルト値: なし

postbodyLimitSignature-シグニチャ内のロケーション HTTP_POST_BODY のシグニチャインスペクションで許可される HTTP ポストボディの最大サイズ（バイト単位）。
値の変更は、CPU および遅延プロファイルに影響を与える可能性があります。
デフォルト値:2048。
最小値:0
最大値:4294967295

fileUploadMaxNum -フォーム送信リクエストごとに許可されるファイルアップロードの最大数。最大設定（65535）では、アップロードの数に制限はありません。
デフォルト値:65535
最小値:0
最大値:65535

CanonicalizeHtmlResponse -保護されたウェブサイトから送信されたレスポンス内の特殊文字に対して HTML エンティティエンコーディングを実行します。
設定可能な値:ON、OFF
デフォルト値:ON

PercentDecodeRecursively -アプリケーションファイアウォールがパーセンテージ再帰的デコードを使用するかどうかを構成します。
設定可能な値:ON、OFF
デフォルト値:ON

MultipleHeaderAction -1 つ以上の複数のヘッダーアクション。使用可能な設定は次のように機能します。

- ブロック。複数のヘッダーを持つ接続をブロックします。
- ログ。複数のヘッダーを持つ接続をログに記録します。
- KeepLast。複数のヘッダーが存在する場合は、最後のヘッダーのみを保持します。

InspectContentTypes —1 つ以上の InspectContentType リスト。

- アプリケーション/x-www-フォーム-URL エンコード

- multipart/form-data
- テキスト/x-gwt-rpc

可能な値：なし、アプリケーション/x-www-form-urlencoded、マルチパート/フォームデータ、テキスト/x-gwt-rpc

semicolonFieldSeparator -URL クエリおよび POST フォーム本文でフォームフィールドの区切り文字として ';' を許可します。
設定可能な値:ON、OFF
デフォルト値:OFF

Web App Firewall プロファイルタイプの変更

August 15, 2023

Web App Firewall プロファイルに間違ったプロファイルタイプを選択した場合や、保護された Web サイトのコンテンツタイプが変更された場合は、プロファイルタイプを変更できます。

> **注:** プロファイルタイプを変更すると、新しいプロファイルタイプがサポートしていない機能に関するすべての構成設定と学習済みの緩和またはルールが失われます。たとえば、プロファイルタイプを Web 2.0 から XML に変更すると、開始 URL、フォームフィールドの一貫性チェック、その他の HTML 固有のセキュリティチェックの設定オプションがすべて失われます。古いプロファイルタイプと新しいプロファイルタイプの両方でサポートされているオプションの設定は変わりません。

コマンドラインインターフェイスを使用して **Web App Firewall** のプロファイルタイプを変更するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw profile <name> -type ( **HTML** | **XML** | **HTML XML** )`
- `save ns config`

例

次の例では、pr-basic という名前のプロファイルのタイプを HTML から HTML XML に変更します。これは GUI の Web 2.0 タイプと同等です。

```
set appfw profile pr-basic -type HTML XML
save ns config
<!--NeedCopy-->
```


GUI を使用して **Web App Firewall** のプロファイルタイプを変更するには

1. [** セキュリティ] > [NetScaler Web App Firewall] **[ポリシー] に移動します。
2. 詳細ペインで、「アクション」をクリックし、「プロファイルタイプの変更」をクリックします。
3. 「**Web App Firewall** プロファイルタイプの変更」ダイアログボックスのプロファイルタイプドロップダウンリストで、新しいプロファイルタイプを選択します。
4. [**OK]** をクリックして変更を保存し、[プロファイル**]** ペインに戻ります。

Web App Firewall プロファイルのエクスポートとインポート

August 15, 2023

Web App Firewall プロファイルの構成全体（HTML エラーオブジェクト、XML エラーオブジェクト、WSDL または XML スキーマ、署名などのバインドされたすべてのオブジェクトを含む）を複数のアプライアンスに複製できます。ターゲットプロファイルを選択して構成をエクスポートし、コンピューターのローカルファイルシステムに保存することも、アーカイブされた構成を転送してサーバーに保存することもできます。同様に、コンピューターのローカルファイルシステムを参照するか、サーバーからアーカイブをインポートして、以前にエクスポートしたプロファイルを選択して NetScaler アプライアンスにインポートできます。

プロファイル構成全体をエクスポートしてから別のアプライアンスにインポートするオプションは、さまざまな用途で役立ちます。たとえば、テストベッドセットアップで Web App Firewall プロファイルを構成して、期待どおりに動作していることをテストおよび検証したい場合があります。問題がなければ、プロファイルをエクスポートして、本番環境の NetScaler アプライアンスにプロファイル構成をインポートできます。この機能は構成のバックアップにも役立ちます。変更を加える前にプロファイルをエクスポートできるので、必要に応じて設定を既知の状態に簡単にロールバックできます。

> 注
>
> あるビルドからエクスポートおよびアーカイブされた Web App Firewall プロファイルは、新しいリリースで導入された変更によって互換性の問題が発生する可能性があるため、別のビルドを実行しているシステムに復元することはできません。アーカイブされたプロファイルを、エクスポート元のビルドとは異なるビルドに復元しようとすると、ns.log にエラーメッセージが記録されます。

プロファイルのエクスポートおよびインポート機能は、GUI (GUI) とコマンドラインインターフェイス (CLI) の両方で使用できます。GUI を使用することをおすすめします。使いやすいアクションオプションが用意されているからです。ボタンをクリックするだけで、プロファイルの構成全体を エクスポート または インポート できます。

CLI Web App Firewall プロファイルのエクスポート

CLI を使用してプロファイルを エクスポート する場合は、構成を アーカイブ してから エクスポート する必要があります。プロファイルをインポートするには、アーカイブを **NetScaler** アプライアンスにインポートしてから、復元

コマンドを実行して構成を抽出する必要があります。次の CLI コマンドセットは、プロファイル構成のエクスポート、インポート、管理に使用できます。

アーカイブをエクスポートする **CLI** コマンド**:**

- `archive appfw profile <name> <archivename> [-comment <string>]`
- `export appfw archive <name> <target>`

アーカイブをインポートする **CLI** コマンド**:**

- `import appfw archive <src> <name> [-comment <string>]`
- `restore appfw profile <archivename>`

アーカイブを管理する **CLI** コマンド**:**

- `show appfw archive`
- `rm appfw archive <name>`

あるアプライアンスからプロファイルをエクスポートして別のアプライアンスへインポートするには、CLI で 5 つのステップが必要です。最初の 3 つのステップは、プロファイル構成が最初に作成されたソースアプライアンスで実行され、次の 2 つのステップは、プロファイル構成が複製されるターゲットアプライアンスで実行されます。

ソース **NetScaler** アプライアンスからプロファイルをエクスポートします。

ステップ **1:** 設定したプロファイルのアーカイブを作成します。

手順 **2**: アーカイブを NetScaler ファイルシステムにエクスポートします。

手順 **3**: scp などのファイル転送ユーティリティを使用して、エクスポートしたアーカイブファイルを NetScaler アプライアンス A からターゲットの NetScaler アプライアンスに転送します。

ターゲットの **NetScaler** アプライアンスにプロファイルをインポートします。

ステップ **4:** インポートコマンドを実行して、アーカイブされたファイルをインポートします。NetScaler のローカルファイルシステムからアーカイブをインポートすることも、HTTP または HTTPS プロトコルを使用して URL を使用してサーバーからアーカイブをインポートすることもできます。

ステップ **5:** 復元コマンドを実行して、インポートしたアーカイブからプロファイル構成を復元する

コマンドラインインターフェイスを使用して **Web App Firewall** プロファイルをエクスポートするには:

まず、プロファイルの設定をアーカイブし、そのアーカイブを目的の場所にエクスポートします。コマンドプロンプトで、次のコマンドを入力します。

```
archive appfw profile <profileName> <archiveName>
```

各項目の意味は次のとおりです:

- <profileName> は、アーカイブするプロファイルの名前です。
- <archiveName> は、作成するアーカイブファイルの名前です。

上記のコマンドを実行すると、アーカイブファイルの 2 つのインスタンスが作成されます。1 つは /var/tmp フォルダーにあり、もう 1 つは /var/archive/appfw フォルダーにあります。

```
export appfw archive <archiveName> <target>
```

各項目の意味は次のとおりです:

- <archiveName> は、エクスポートするアーカイブの名前です。(前のコマンドと同じ名前。
- <target> は、接頭辞として local: で始まるファイルパスで、その後に<archiveName>が続きます。

エクスポートコマンドを実行すると、エクスポートされたアーカイブファイルが NetScaler アプライアンスのファイルシステムの/var/tmp フォルダーに保存されます。

例:

```
> archive appfw profile test_pr archived_test_pr
> export appfw archive archived_test_pr local:dutA_test_pr
```

上記の 2 つのコマンドを実行すると、/var/tmp フォルダーには archived_test_pr ファイルとエクスポートされたコピー Duta_Test_PR が格納されます。これらはサイズが同じです。CLI からシェルにドロップしてフォルダに移動し、これらのファイルがそこにあることを確認できます。

アーカイブファイルをエクスポートしたら、**scp** などのファイル転送ユーティリティを使用して、アーカイブファイルのコピーを、ファイルが作成された NetScaler アプライアンスからターゲットの NetScaler アプライアンスに転送できます。

CLI Web App Firewall プロファイルのインポート

ソースアプライアンスからターゲットアプライアンスにアーカイブファイルを正常に scp できたら、プロファイルのアーカイブをインポートし、復元コマンドを実行してプロファイルの構成をターゲットアプライアンスに複製する準備が整います。

ターゲットアプライアンスにログインします。シェルにドロップし、cd を /var/tmp フォルダにドロップして、このアプライアンスの scp ファイルのサイズがソースアプライアンス上の元のアーカイブファイルのサイズと一致することを確認します。シェルを終了してコマンドラインに戻ります。

CLI を使用してプロファイルをインポートするには:

コマンドプロンプトで、次のコマンドを入力します。

```
import appfw archive <src> <name> [-comment <string>]
```

各項目の意味は次のとおりです。

- <src> は、アーカイブファイルが作成されたソースアプライアンスから転送された後のアーカイブファイルの場所です。ローカルファイルシステムとファイル名を使用できます。アーカイブをサーバーに置いた場合は、URL を使用してアーカイブファイルをインポートできます。パスまたはファイル名にスペースが含まれている場合は、URL を二重引用符で囲みます。

- <name> は、インポートするアーカイブファイルの名前です。
- <string> アーカイブの目的に関するオプションの説明です。

```
restore appfw profile <archiveName>
```

例:

A。ローカルファイルからのインポートと復元:

```
> import appfw archive local:dutA_test_pr dut2_test_pr
> restore appfw profile dut2_test_pr
```

B。**URL** からインポートし、次にリストアします。

```
import appfw archive http://10.217.30.16/FFC/Profile_ImportExport
/dutA_test_pr.tgz my_archive
restore appfw profile my_archive
```

この例では、test_pr プロファイルとすべてのバインドされたオブジェクト（署名、HTML エラーページ、緩和ルールなど）をターゲットの NetScaler アプライアンスに復元します。

次の CLI コマンドを使用してマニュアルページにアクセスすると、詳細を確認できます。

- メインアーカイブ appfw プロファイル
- man エクスポート appfw アーカイブ
- man import appfw archive
- appfw プロファイルのリストア
- man show appfw archive
- man rm appfw archive

GUI による Web App Firewall プロファイルのエクスポートとインポート

GUI は CLI よりも使いやすいです。[エクスポート] をクリックすると、ユーティリティはアーカイブ操作とエクスポート操作の両方を実行します。同様に、**[インポート]** をクリックすると、インポートと復元の両方が実行されます。GUI は、ユーティリティにアクセスするコンピュータのローカルファイルシステムにアクセスできます。アーカイブのコピーをエクスポートして、ローカルコンピューターに保存できます。その後、このコピーをターゲットアプライアンスに直接インポートできます。アーカイブファイルをあるアプライアンスから別のアプライアンスに手動で転送する必要はありません。

GUI を使用して **Web App Firewall** プロファイルをエクスポートするには:

1. [設定] > [セキュリティ] > [**Web App Firewall**] > [プロファイル] に移動します。
2. 詳細ペインで、エクスポートするプロファイルを選択します。[アクション] をクリックし、[エクスポート] を選択して、コピーをダウンロードしてコンピューターのローカルファイルシステムに保存します。

GUI を使用して **Web App Firewall** プロファイルをインポートするには:

1. [設定] > [セキュリティ] > [**Web App Firewall**] > [プロファイル] に移動します。
2. 詳細ペインで、[操作] をクリックし、[インポート] を選択します。[Web App Firewall プロファイルのインポート] ペインの [インポート元 *] 選択ボックスには、次の 2 つのオプションがあります。

URL: URL を指定してアーカイブをインポートできます。このオプションを選択した場合は、**URL** 入力ボックスにアーカイブファイルの絶対パスを指定する必要があります。

ファイル**:** ローカルファイルからアーカイブをインポートすることを選択できます。このオプションを選択すると、ローカルファイル選択フィールドが表示されます。コンピューターのローカルファイルをブラウズして、対象のアーカイブファイルを選択できます。

「作成」をクリックして、指定したアーカイブをインポートします。インポート操作が正常に完了すると、ターゲットアプライアンスにプロファイル構成が作成されます。

ハイライト

- プロファイルのエクスポートとインポートの機能を使用すると、構成手順を繰り返すことなく、構成全体（すべてのインポートオブジェクトとプロファイルに設定された緩和ルールを含む）を複数のアプライアンスに複製できます。
- シグネチャ、WSDL、スキーマ、エラーページなどのインポートされたオブジェクトは、アーカイブされた tar ファイルに含まれ、ターゲットアプライアンスに複製されます。
- カスタマイズされたフィールドタイプはアーカイブされた tar ファイルに含まれ、ターゲットアプライアンスに複製されます。
- 構成を復元しても、アーカイブされたプロファイルのポリシーバインディングは複製されません。アプライアンスにプロファイルをインポートしたら、ポリシーを設定してプロファイルにバインドする必要があります。
- アーカイブファイルの名前は、最大 31 文字です。プロファイル名と同様に、アーカイブ名は英数字またはアンダースコアで始まり、英数字とアンダースコア (_)、数字 (#)、ピリオド (.)、スペース ()、コロン (:)、アット (@)、等号 (=)、またはハイフン (-) のみを含む必要があります。
- アーカイブに関連するコメントは、アーカイブされた設定の目的を伝えるのに十分な説明が必要です。コメントの最大許容長は 255 文字です。
- `clear config -force basic` このコマンドでは、アーカイブされたプロファイルは削除されません。
- プロファイルのインポートおよびエクスポート機能は、高可用性 (HA) 展開でサポートされています。

デバッグに関するヒント

- コマンド実行中に /var/log/ns.log を監視して、エラーメッセージが表示されるかどうかを確認します。
- その他のログ (_restore.log、remove.log、import.log) は /var/tmp/フォルダーに生成されます。対応する操作中に問題をデバッグするのに役立ちます。これらのログのサイズが 1 MB に達すると、ログメッセージは消去され、ログファイルが元のサイズの 4 分の 1 に縮小されます。

- ローカルファイルシステムの代わりに URL オプションを使用しているときにインポートコマンドが失敗した場合は、DNS ネームサーバーとルートの設定が正確に設定されていることを確認してください。
- HTTPS プロトコルを使用してアーカイブをインポートする場合、HTTPS サーバーがクライアント証明書認証を要求すると、コマンドが失敗することがあります。

Web アプリケーションファイアウォールのログによるトラブルシューティングの容易さ

March 20, 2024

セキュリティ攻撃が発生した場合、アプライアンスで詳細な WAF ロギングをキャプチャすることが重要です。そのためには、アプリケーションファイアウォールプロファイルで「VerboseLogLevel」パラメータを設定できます。

Web トラフィックにセキュリティ攻撃があるとします。アプライアンスがトラフィックを受信すると、HTTP ヘッダーの詳細、ログパターン、パターンペイロード情報などの違反の詳細がログに記録され、ADM サーバに送信されます。ADM サーバーは詳細なログを監視し、監視と追跡の目的で [Security Insight] ページに表示します。

コマンドインターフェイスを使用した詳細ログレベルの設定

詳細な WAF ログをキャプチャするには、次のコマンドを設定します。
コマンドインターフェイスで、次のように入力します。

```
set appfw profile <profile_name> -VerboseLogLevel (pattern|patternPayload
|patternPayloadHeader)
```

例

```
set appfw profile profile1 -VerboseLogLevel patternPayloadHeader
```

使用可能なログレベルは次のとおりです。

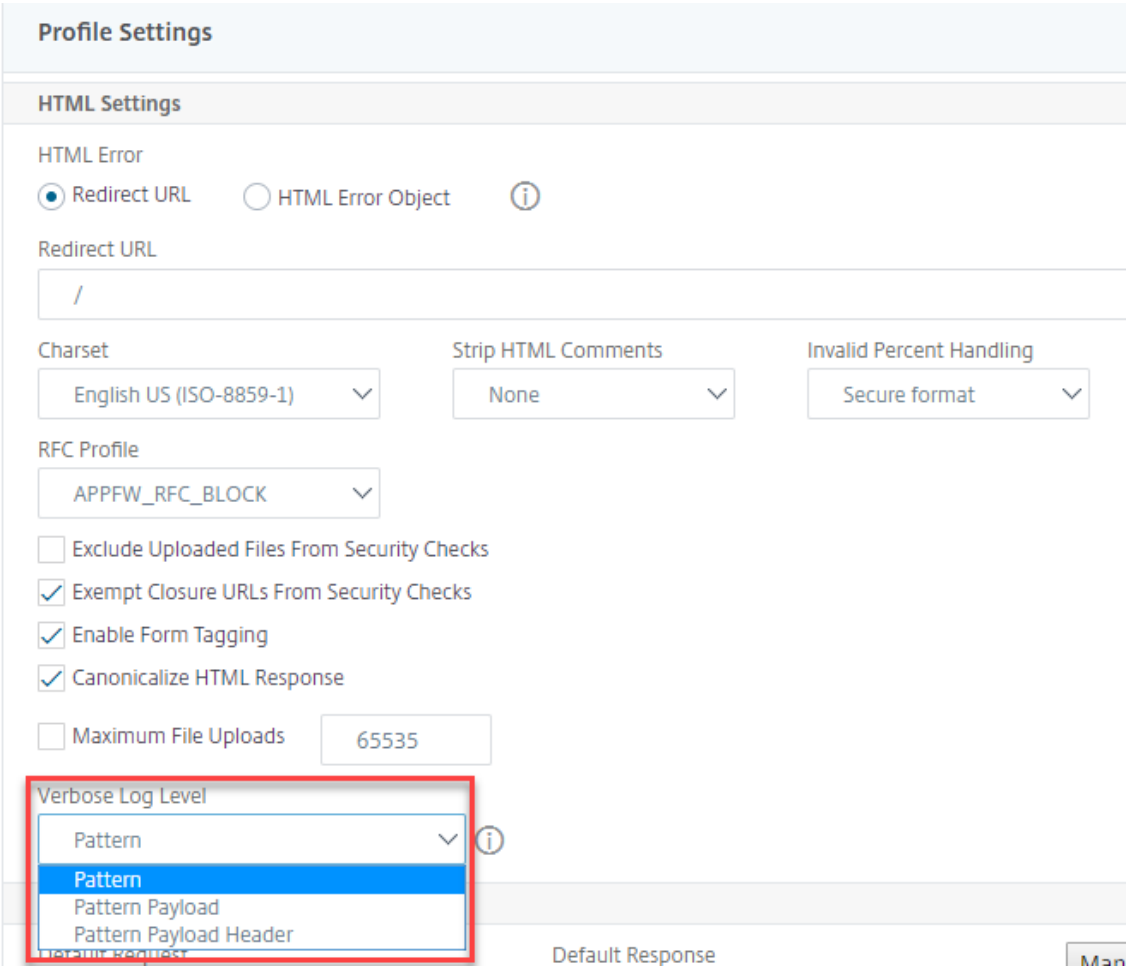
1. [パターン]。違反パターンのみをログに記録します。
2. パターンペイロード。違反パターンと 150 バイトの追加フィールド要素ペイロードをログに記録します。
3. パターン・ペイロード・ヘッダー。違反パターン、150 バイトの追加フィールド要素ペイロード、および HTTP ヘッダー情報をログに記録します。

NetScaler GUI を使用して詳細ログレベルを構成する

WAF プロファイルで詳細ログレベルを設定するには、次の手順を実行します。

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。

2. 「プロファイル」ページで、「追加」をクリックします。
3. **NetScaler Web App Firewall**［プロファイル］ページで、［詳細設定］の［プロファイル設定］をクリックします。
4. [プロファイル設定] セクションの [詳細ログレベル] フィールドで、詳細な WAF ログレベルを選択します。
5. [**OK**] をクリックし、[完了] をクリックします。



JSON セキュリティーチェック **(SQL**、**CMD**、クロスサイトスクリプティング**)** のための冗長ロギング

受信リクエストタイプが JSON の場合、verbose log level パラメータを設定して、パターン、パターンペイロード、HTTP ヘッダー情報などの詳細な違反ログをキャプチャできます。その後、ログの詳細が NetScaler Console サーバーに送信され、JSON 違反の監視とトラブルシューティングが行われます。詳細ログメッセージは ns.log ファイルには保存されません。
JSON コンテンツタイプのセキュリティ保護に関する詳細ロギングは、次の違反タイプに対して構成できます。

- SQL インジェクション
- クロスサイトスクリプティング

- コマンドインジェクション

CLI を使用して **JSON** セキュリティ保護のための詳細ロギングを設定する

詳細な HTTP ヘッダー情報をログとしてキャプチャするために、Web App Firewall プロファイルで詳細ログパラメーターを構成できます。コマンドプロンプトで入力します:

```
set appfw profile <profile_name> -VerboseLogLevel ( pattern |
   patternPayload | patternPayloadHeader )
<!--NeedCopy-->
```

例:

```
set appfw profile profile1 -VerboseLogLevel patternPayloadHeader
```

使用可能なログレベルは次のとおりです。

[パターン]。違反パターンのみをログに記録します。

パターンペイロード。違反パターンと 150 バイトの余分な JSON ペイロードをログに記録します。

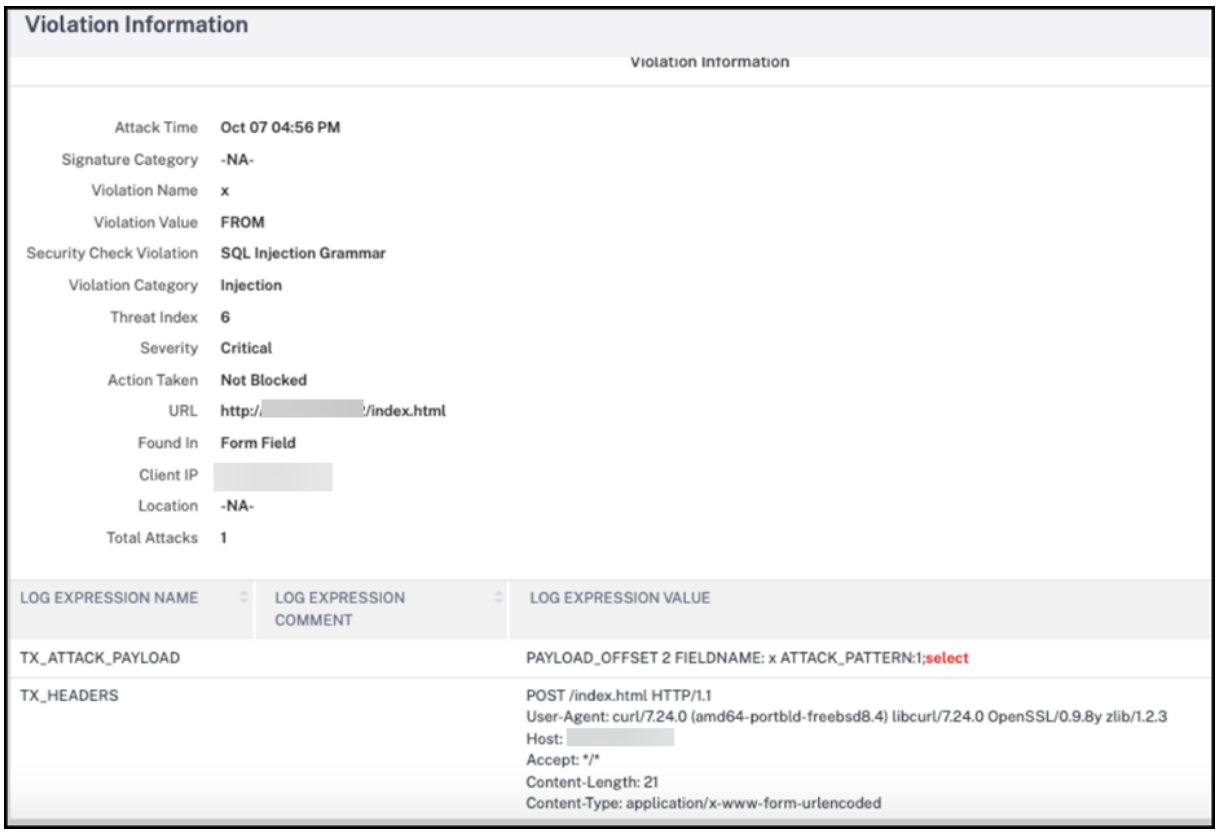
パターン・ペイロード・ヘッダー。違反パターン、150 バイトの余分な JSON ペイロード、および HTTP ヘッダー情報をログに記録します。

NetScaler GUI を使用して詳細ログレベルを構成する

以下の手順に従って、JSON セキュリティ保護用の詳細ログレベルを設定します。

1. ナビゲーションペインで、[セキュリティ**] > [**プロファイル] に移動します。
2.「プロファイル」ページで、「追加」をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、[詳細設定] の [セキュリティチェック] をクリックします。
4. [セキュリティチェック] セクションで [**JSON**] を選択し、[アクション設定] をクリックします。
5. [**JSON** セキュリティ設定**]** ページで、[**Verbose]** ログレベルパラメータを設定します。
6. [**OK**] をクリックし、[完了] をクリックします。

NetScaler WAF JSON 詳細ロギングによってキャプチャされた詳細に基づいて、NetScaler コンソールサーバーで次の違反の詳細を確認できます。



ファイルアップロード保護

March 20, 2024

多くの攻撃者は、マルチフォーム送信中に悪意のあるコード、ウイルス、またはマルウェアを添付ファイルとしてアップロードしようとします。ネットワークを保護し、そのような脅威を克服することが重要です。このような悪意のあるファイルのアップロードを防ぐために、NetScaler 管理者は WAF プロファイルで許容されるファイルアップロード形式のセットを構成します。これにより、ファイルのアップロードを特定の形式に制限し、悪意のあるファイルのアップロードからアプライアンスを保護します。保護は、WAF プロファイルで`ExcludeFileUploadFormChecks`オプションを無効にした場合にのみ機能します。

ファイルアップロードの仕組み

許容されるファイルアップロード形式を設定すると、コンポーネントのインタラクションは次のようになります。

- クライアント要求には、ファイルアップロードタイプ（PDF など）のフォーム送信があります。
- セキュリティチェックの一環として、WAF はリクエストペイロードを検査し、（マジックシグネチャ番号に基づいて）ファイルタイプを検証します。

- ファイルタイプがサポートされていない形式の場合、ファイルタイプバインディングに基づく対応するアクションが適用されます。
- ファイルタイプを検証するために、アプライアンスはペイロードを検査し、既知のオフセットで既知のマジックナンバーをチェックします。各ファイルタイプには、ファイルタイプを検証するマジックナンバーのシーケンスがあります。

NetScaler CLI を使用してファイルタイプのアップロードを構成する

許容されるファイル形式を設定するために、アプライアンスはファイルアップロードパラメータにバインドされた WAF プロファイルを使用します。

1. Web アプリケーションファイアウォールプロファイルの構成

コマンドプロンプトで入力します:

```
set appfw profile <profile_name> [-fileUploadTypesAction <fileUploadTypesActio
>]<fileUploadTypesAction> = ( none | block | log | stats )
```

例

```
set appfw profile profile1 -fileUploadTypesAction block
```

1. ファイルアップロードパラメータを使用して Web アプリケーションファイアウォールプロファイルをバインドします。このコマンドは、指定された除外（緩和）またはルールを指定されたアプリケーションファイアウォールプロファイルにバインドします。

コマンドプロンプトで入力します:

```
bind appfw profile <profile_name> - fileUploadType <form_field >
<form_action_url> [-isNameRegex ( REGEX | NOTREGEX )] -fileType <
fileType> ( pdf | msdoc | text | image | any)
```

> 注意:
>
> フォームフィールド名は正規表現タイプです。デフォルトの値はNOTREGEXです。

例

```
> bind appfw profile test -fileuploadType thefile "http://10.10.10.10/
fileupload_sample/upload.php"-isNameRegex NOTREGEX -filetype image
```

->

NetScaler GUI を使用してファイルアップロードのセキュリティ保護を構成する

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2.「プロファイル」ページで、「追加」をクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、[詳細設定] の [セキュリティチェック] をクリックします。
4.「セキュリティチェック」セクションで、「ファイルのアップロードタイプ」を選択し、「アクション設定」をクリックします。
5. [ファイルアップロードの種類**]** 設定ページで、ファイルのアップロードアクションを設定します。
6. [**OK**] をクリックします。
7. [**NetScaler Web App Firewall** プロファイル] ページで、[**OK**] をクリックし、[完了] をクリックします。

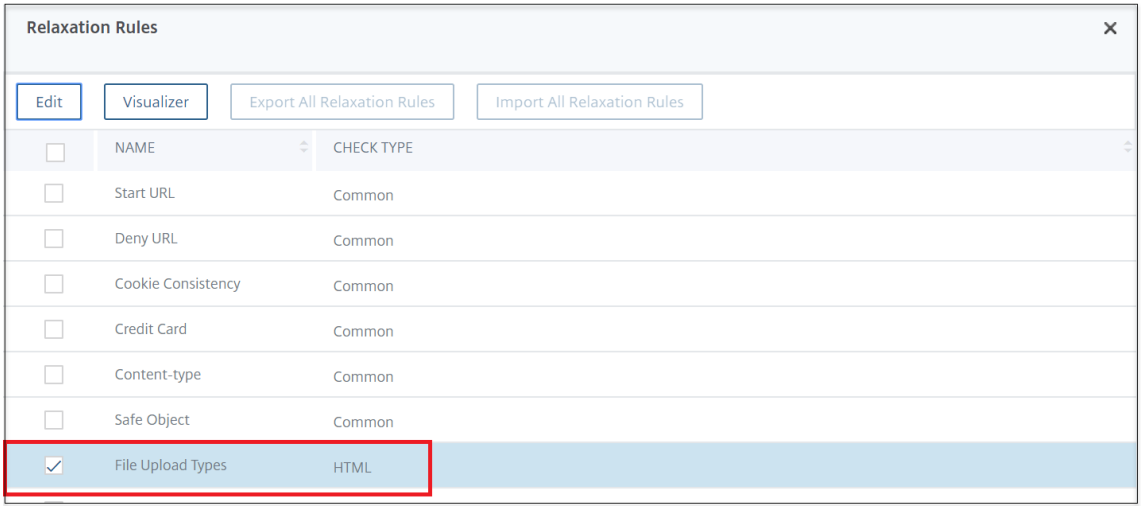
NetScaler GUI を使用してファイルアップロード緩和ルールを構成する

ファイルアップロードのセキュリティ保護を緩和して、誤検出を回避できます。たとえば、アプライアンスがファイルのアップロードをブロックする場合がありますが、緩和ルールを追加して、特定の Web サイトからのファイルアップロードを許可できます。これにより、アプライアンスは指定されたフォームフィールドのセキュリティ検査をバイパスし、アクション URL に記載されている Web サイトからファイルをアップロードすることをユーザに許可します。

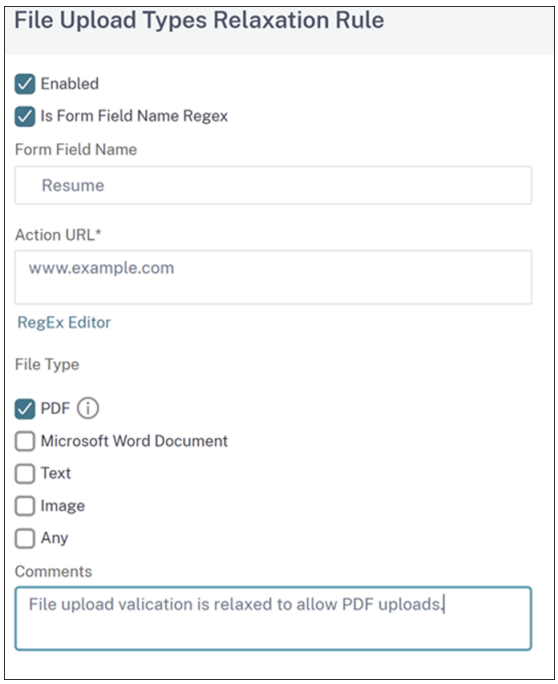
> 注**:**
>
> ファイルアップロードタイプの再評価ルールが有効になっていない場合、ファイルアップロードの検証は失敗します。

以下の手順を実行して緩和ルールを作成します。

1. ナビゲーションペインで、[セキュリティ] **>** [NetScaler Web App Firewall] **>** [プロファイル] に移動します。
2.「プロファイル」ページで、「追加」をクリックします。
3. [**NetScaler Web App Firewall** プロファイル] ページで、[詳細設定] の [緩和ルール] をクリックします。
4. [緩和ルール] セクションで、[ファイルアップロードの種類] を選択し、[編集] をクリックします。



5. [ファイルアップロードタイプの再分類規則] ページで、[追加] をクリックします。
6. [ファイルアップロードタイプの緩和規則] ページで、次のパラメータを設定します。
   a) 有効-選択すると緩和ルールが有効になります。
   b) Is Form Field Name Regex-選択すると、フォームフィールド名の正規表現パターンが更新されます。
   c) フォームフィールド名-セキュリティチェックを必要としないファイル名を入力します。
   d) アクション URL-セキュリティチェックから除外する必要があるフォーム送信 URL。
   e) ファイルタイプ-アップロード可能なサポートされているファイル形式。
   f) コメント-ファイルのアップロードに関する簡単な説明。
7. [**Create**] をクリックします。



8. [**NetScaler Web App Firewall** プロファイル] ページで、[**OK**] をクリックし、[完了] をクリックします。

ラーニング機能の設定と使用

March 20, 2024

学習機能は、Web App Firewall で保護された Web サイトまたはアプリケーションのアクティビティを監視して、その Web サイトまたはアプリケーションでの通常のアクティビティを構成するものを特定する反復パターンフィルターです。次に、学習機能のサポートを含むセキュリティチェックごとに、最大 2,000 の推奨ルールまたは例外 (緩和) のリストを生成します。ユーザーは通常、必要な緩和を手動で入力するよりも、学習機能を使用する方が、緩和を設定するほうが簡単です。

ラーニング機能をサポートするセキュリティー検査は、次のとおりです。

- URL チェックを開始
- クッキーの整合性チェック
- フォームフィールドの一貫性チェック
- [フィールド形式] チェック
- CSRF フォームタグ付けチェック
- HTML SQL インジェクションチェック
- HTML クロスサイトスクリプティングチェック
- XML サービス拒否チェック
- XML 添付ファイルのチェック
- Web サービスの相互運用性チェック

学習機能の使用時には、2 つの異なるタイプのアクティビティを実行します。まず、この機能を使用できるように機能を有効にして設定します。保護された Web アプリケーションへのすべてのトラフィックを学習することも、学習機能で推奨を生成できる IP アドレスのリスト（*[*信頼できる学習クライアントの追加（*Add Trusted* Learning Clients）] リスト）を設定することもできます。次に、この機能を有効にして、保護された Web サイトへの一定量のトラフィックを処理したら、推奨されるルールと緩和 (学習済みルール) の一覧を確認し、それぞれに次のいずれかの指定をマークします。

- 編集とデプロイ。ルールは [編集] ダイアログボックスに取り込まれ、変更可能になり、変更されたフォームが展開されます。
- デプロイ。変更されていない学習済みルールは、このセキュリティチェックのルールまたは緩和のリストに配置されます。
- スキップ。学習したルールは、展開されていないルールまたは緩和のリストに配置されます。学習したルールは、スキップすると削除されます。ただし、それらはリラクゼーションに追加されないため、再び学習される可能性があります。

学習は、フィールド形式のルールを除き、緩和が設定されている場合にのみ実行されるわけではありません。ルールをスキップすると、学習したデータベースからのみ削除されます。リラクゼーションは追加されないので、再び学習される可能性があります。ルールが展開されると、学習済みデータベースから削除され、緩和もルールに追加されま

す。リラクゼーションが追加されると、再び学習されることはありません。フィールド形式の保護では、緩和に関係なく学習が実行されます。

学習機能の基本的な構成にはコマンドラインインターフェイスを使用できますが、この機能は主に Web App Firewall ウィザードまたは GUI を使用した構成のために設計されています。コマンドラインを使用すると、限定されたラーニング機能の設定しか実行できません。

このウィザードは、学習機能の構成と Web App Firewall 全体の構成を統合するため、新しい NetScaler アプライアンスでこの機能を構成したり、単純な Web App Firewall 構成を管理したりする場合に最も簡単な方法です。GUI ビジュアライザーと手動インターフェイスはいずれも、すべてのセキュリティチェックの学習済みルールに直接アクセスできるため、多くのセキュリティチェックで学習済みルールを確認する必要がある場合に適しています。

ラーニングデータベースのサイズは 20 MB に制限されており、ラーニングが有効になっているセキュリティチェックごとに、学習済みルールまたは緩和が約 2,000 個生成された後に到達します。学習済みルールを定期的に確認せず、承認または無視しても、この制限に達すると、NetScaler ログにエラーが記録され、既存の学習済みルールと緩和を確認するまで学習済みルールは生成されません。

データベースのサイズ制限に達したために学習が停止した場合は、既存の学習済みルールと緩和を確認するか、学習データをリセットすることで、学習を再開できます。学習したルールまたは緩和が承認または無視されると、データベースから削除されます。学習データをリセットすると、既存の学習データはすべてデータベースから削除され、最小サイズにリセットされます。データベースのサイズが 20 MB 未満になると、学習が自動的に再開されます。

コマンドラインインターフェイスを使用して学習設定を構成するには

構成する Web App Firewall プロファイルを指定し、そのプロファイルに含めるセキュリティチェックごとに、最小しきい値またはパーセントしきい値を指定します。最小しきい値は、Web App Firewall がルールまたは緩和を学習する前に処理しなければならないユーザーセッションの最小数を表す整数です (デフォルト:1)。パーセントしきい値は、Web App Firewall がルールまたは緩和を学習する前に特定のパターン（URL、Cookie、フィールド、添付ファイル、またはルール違反）を観察しなければならないユーザーセッションの割合を表す整数です（デフォルト: 0)。次のコマンドを使用します。

- `set appfw learningsettings <profileName> [-startURLMinThreshold <positive_integer>] [-startURLPercentThreshold <positive_integer>] [-cookieConsistencyMinThreshold <positive_integer>] [-cookieConsistencyPe <positive_integer>] [-CSRFtagMinThreshold <positive_integer>] [-CSRFtagPercentThreshold <positive_integer>] [-fieldConsistencyMinThreshold <positive_integer>] [-fieldConsistencyPercentThreshold <positive_integer >] [-crossSiteScriptingMinThreshold <positive_integer>] [-crossSiteScripti <positive_integer>] [-SQLInjectionMinThreshold <positive_integer >] [-SQLInjectionPercentThreshold <positive_integer>] [-fieldFormatMinThre <positive_integer>] [-fieldFormatPercentThreshold <positive_integer >] [-XMLWSIMinThreshold <positive_integer>] [-XMLWSIPercentThreshold`


```
 <positive_integer>] [-XMLAttachmentMinThreshold <positive_integer
>] [-XMLAttachmentPercentThreshold <positive_integer>]
```

- `save ns config`

例

次の例では、HTML SQL Injection セキュリティチェックのプロファイルで学習設定を有効にし、設定します。これは、Web App Firewall に送信されるトラフィックを完全に制御できる、適切な初期テストベッド学習設定です。

```
set appfw learningsettings pr-basic -SQLInjectionMinThreshold 10
set appfw learningsettings pr-basic -SQLInjectionPercentThreshold 70
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して学習設定を既定値にリセットするには

指定したプロファイルとセキュリティチェックの学習設定のカスタム設定を削除し、学習設定をデフォルトに戻すには、コマンドプロンプトで次のコマンドを入力します。

- `unset appfw learningsettings <profileName> [-startURLMinThreshold] [-startURLPercentThreshold] [-cookieConsistencyMinThreshold] [-cookieConsistencyPercentThreshold] [-CSRFtagMinThreshold] [-CSRFtagPercentThreshold] [-fieldConsistencyMinThreshold] [-fieldConsistencyPercentThreshold] [-crossSiteScriptingMinThreshold] [-crossSiteScriptingPercentThreshold] [-SQLInjectionMinThreshold] [-SQLInjectionPercentThreshold] [-fieldFormatMinThreshold] [-fieldFormatPercentThreshold] [-XMLWSIMinThreshold] [-XMLWSIPercentThreshold] [-XMLAttachmentMinThreshold] [-XMLAttachmentPerc]`
- `save ns config`

コマンドラインインターフェイスを使用してプロファイルの学習設定を表示するには

コマンドプロンプトで、次のコマンドを入力します:

```
show appfw learningsettings <profileName>
```

コマンドラインインタフェースを使用して、プロファイルの未確認の学習済みルールまたは緩和を表示するには

コマンドプロンプトで、次のコマンドを入力します:


```
show appfw learningdata <profileName> <securityCheck>
```

コマンドラインインターフェイスを使用して、未確認の学習済みルールまたは緩和を学習データベースから削除するには

コマンドプロンプトで、次のコマンドを入力します:

```
rm appfw learningdata <profileName> (-startURL <expression> | -
cookieConsistency <string> | (-fieldConsistency <string> <formActionURL
>)| (-crossSiteScripting <string> <formActionURL>)| (-SQLInjection <
string> <formActionURL>)| (-fieldFormat <string><formActionURL>)| (-
CSRFTag <expression> <CSRFFormOriginURL>)| -XMLDoSCheck <expression
> | -XMLWSICheck <expression> | -XMLAttachmentCheck <expression>)[-
TotalXMLRequests]
```

例

次の例では、**LastName** フォームフィールドに適用される、プロファイルの HTML SQL インジェクションセキュリティチェックの未確認学習緩和をすべて削除します。

```
1 rm appfw learningdata pr-basic -SQLInjection LastName
2 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、未確認の学習済みデータをすべて削除するには

コマンドプロンプトで、次のコマンドを入力します:

```
reset appfw learningdata
```

コマンドラインインターフェイスを使用して学習データをエクスポートするには

コマンドプロンプトで、次のコマンドを入力します:

```
export appfw learningdata <profileName> <securitycheck>[-target <
string>]
```

例

次の例では、プロファイルと HTML SQL インジェクションセキュリティチェックの学習済み緩和を、-target パラメーターで指定されたファイル名の /var/learnt_data/ ディレクトリにあるカンマ区切り値 (CSV) 形式のファイルにエクスポートします。


```
export appfw learningdata pr-basic SQLInjection -target sqli_ld
<!--NeedCopy-->
```

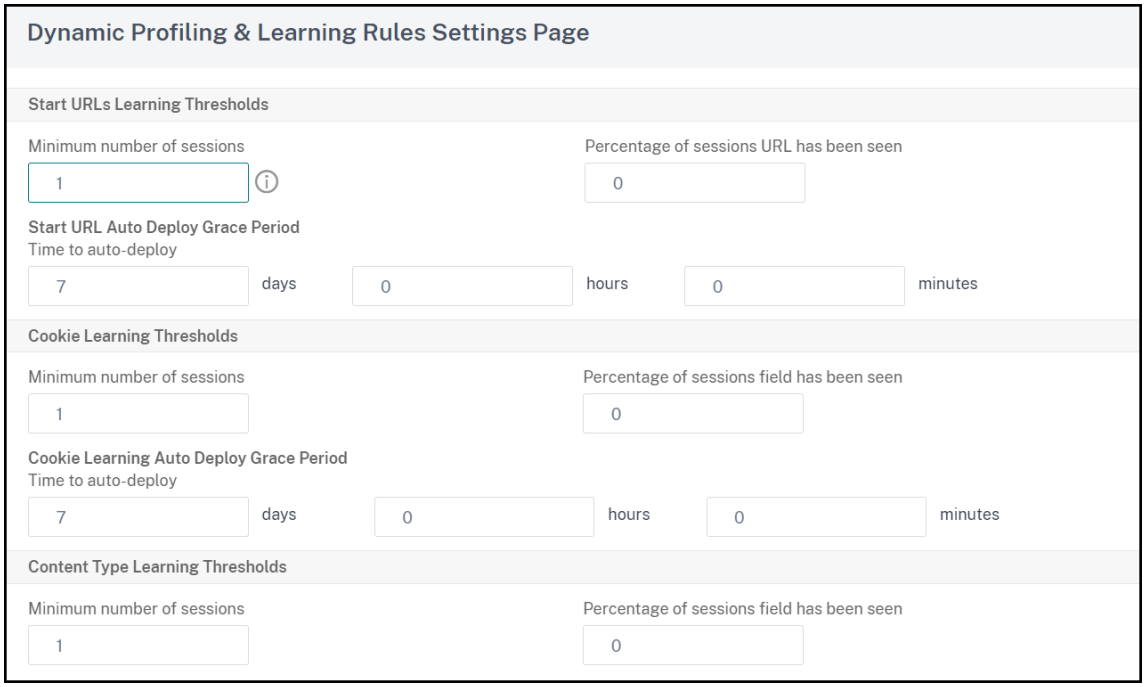
GUI を使用してラーニング機能を設定するには

1. セキュリティ > **Web App Firewall** > プロファイルに移動します。
2. [プロファイル] ペインでプロファイルを選択し、[編集] をクリックします。
3. [詳細設定] セクションの **[**学習ルール] をクリックします。
4. [学習したルール] セクションで、セキュリティチェックを選択し、[設定] をクリックします。
5. [セキュリティチェックの設定] ページで、次のパラメータを設定します。

   a) 最小数のしきい値。設定しているセキュリティチェックの学習設定に応じて、最小数のしきい値は、監視する必要のあるユーザーセッションの合計の最小数、遵守する必要のあるリクエストの最小数、または特定のフォームフィールドを監視する必要がある最小回数を指す場合があります。学習したリラクゼーションが生成される前に。デフォルト:1

   b) 回数のしきい値のパーセンテージ。設定しているセキュリティチェックの学習設定に応じて、しきい値の割合は、セキュリティチェックに違反した観察されたユーザーセッションの合計の割合、リクエストの割合、またはフォームフィールドが特定のフィールドタイプと一致した回数の割合を指す場合があります。学習したリラクゼーションが生成されます。デフォルト:0

6. 「**OK**」をクリックして「閉じる」をクリックします。



7. 学習済みデータをすべて削除して学習機能をリセットするには、[すべての学習済みデータを削除] をクリックします。これにより、最初から観測を再開する必要があります。

   > 注:
   >
   > このボタンは、レビューされておらず、承認またはスキップされた学習済みの推奨事項のみを削除します。受け入れられて展開された学習済み緩和は削除されません。

8. 学習エンジンを特定の IP セットからのトラフィックに制限するには、[信頼できる学習クライアント] をクリックし、使用する IP アドレスをリストに追加します。

   a) [信頼できる学習クライアント] リストに IP アドレスまたは IP アドレス範囲を追加するには、[追加] をクリックします。
   b) [信頼できるラーニングクライアントの追加] ダイアログボックスの [信頼できるクライアント IP] リストボックスに、IP アドレスまたは IP アドレスの範囲を CIDR 形式で入力します。
   c)「コメント」テキスト領域に、この IP アドレスまたは IP アドレス範囲を説明するコメントを入力します。
   d) [**Create**] をクリックして、新しい IP アドレスまたは範囲をリストに追加します。
   e) 既存の IP アドレスまたは範囲を変更するには、IP アドレスまたは範囲をクリックし、[開く] をクリックします。名前以外は、表示されるダイアログボックスは「信頼できる学習クライアントの追加」ダイアログボックスと同じです。
   f) IP アドレスまたは範囲を無効または有効にして、一覧に表示したままにするには、IP アドレスまたは範囲をクリックし、必要に応じて [無効**]** または **[**有効化] をクリックします。
   g) IP アドレスまたは IP 範囲を完全に削除するには、IP アドレスまたは範囲をクリックし、[削除] をクリックします。

9. [閉じる] をクリックして [Web App Firewall プロファイルの構成] ページに戻ります。
10. ［完了］ をクリックします。

GUI を使用して、学習したルールまたは緩和を確認するには

1. セキュリティ > **Web App Firewall** > プロファイルに移動します。
2. [プロファイル] ペインでプロファイルを選択し、[編集] をクリックします。
3. [詳細設定**]** セクションの **[**学習ルール] をクリックします。
4. [学習したルール**]** セクションで、セキュリティチェックを選択し、[設定] をクリックします。
5. 学習したデータを分岐ツリーとして階層的に確認し、学習したパターンの多くに一致する一般的なパターンを選択できるようにするには、[**Visualizer**] をクリックします。
6. 実際に学習したパターンを確認する場合は、次の手順を実行します。
7. 最初に学習したリラクゼーションを選択し、その処理方法を選択します。

a) 緩和を変更して承諾するには、「**Edit & Deploy**」をクリックし、緩和の正規表現を編集して「**OK**」をクリックします。
b) 変更せずに緩和を適用するには、[**Deploy**] をクリックします。
c) 展開せずに緩和をリストから削除するには、[スキップ] をクリックします。
d) 前のステップを繰り返して、学習したリラクゼーションを追加するたびに確認します。

8. [閉じる] をクリックして [学習済みルールの管理] ダイアログボックスに戻ります。
9. [完了] をクリックします。

動的プロファイリング

March 20, 2024

学習機能は、バックエンドサーバー上のアクティビティを監視および学習するパターンフィルタです。観察に基づいて、学習エンジンは、セキュリティチェックごとに最大 2000 のルールまたは例外 (緩和) を生成します。プロセスを自動化し、緩和ルールを自動展開するために、NetScaler アプライアンスは動的プロファイリングを使用します。

動的プロファイリングでは、アプライアンスは事前に定義されたしきい値について学習したデータを記録し、SNMP アラートをユーザーに送信します。猶予期間内にユーザーがデータをスキップしない場合、アプライアンスはそれを緩和ルールとして自動的にデプロイします。以前は、ユーザーは緩和ルールを手動で展開する必要がありました。現在、動的プロファイリングは、次のセキュリティチェックでのみ使用できます。

1. HTML SQL インジェクション
2. HTML クロスサイトスクリプティング
3. フィールド形式
4. 開始 URL
5. コンテンツタイプ
6. フィールド形式
7. CSRF フォームのタグ付け
8. Cookie の整合性
9. URL を拒否する
10. バッファオーバーフロー
11. クレジットカード
12. コンテンツタイプの保護
13. JSON コマンドインジェクション保護

たとえば、動的プロファイリングで有効化された HTML SQL Injection セキュリティチェックを考えます。学習機能でレコメンデーションを生成する必要がある IP のリスト（信頼済みラーニングクライアントリスト）については、ラーニングを使用できます。信頼できるクライアントの一覧を構成するには、信頼されたクライアントの学習トピックを参照してください。着信トラフィックに違反がある場合は、学習データとして記録されます。学習したデータが

ラーニングエンジンに記録されている場合、アプライアンスはユーザに SNMP アラートを送信します。ユーザーが誤検知を認識せず、猶予期間内に学習したデータをスキップしない場合、アプライアンスはそのデータを緩和ルールとして自動的に展開します。

> 注:
> ダイナミックプロファイルを設定したら、緩和ルールの自動展開についてアプライアンス設定を定期的に確認し、アプライアンスに保存する必要があります。

NetScaler コマンドインターフェイスを使用して動的プロファイリングを構成する

動的プロファイリングは、開始 URL、HTML クロスサイトスクリプティング、フィールド形式、または HTML SQL インジェクションのセキュリティチェックで使用できます。動的プロファイリングを設定するには、次の手順を完了する必要があります。

1. ダイナミックラーニングの設定
2. 自動展開の猶予期間を構成する

ダイナミックラーニングの設定

最初のステップとして、アプライアンスでダイナミックラーニングを設定する必要があります。コマンドプロンプトで入力します:

```
set appfw profile <profile_name> dynamicLearning <security_checks>
```

例 `set appfw profile test1 dynamicLearning SQLInjection CrossSiteScripting fieldFormat startURL`

自動展開の猶予期間を構成する

特定のセキュリティチェックでこの機能を有効にしたら、自動展開の猶予期間を設定する必要があります。

```
set appfw learningsettings <profile name> -crossSiteScriptingAutoDeployGracePe
 <seconds>
set appfw learningsettings <profile name> fieldFormatAutoDeploymentGracePeriod
 <seconds>
set appfw learningsettings <profile name> SQLInjectionAutoDeploymentGracePerio
 <seconds>
set appfw learningsettings <profile name> -startURLAutoDeployGracePeriod
 <seconds>
```


例 `set appfw learningsettings test1 -crossSiteScriptingAutoDeployGracePeriod 30`

`set appfw learningsettings test1 -startURLAutoDeployGracePeriod 7`

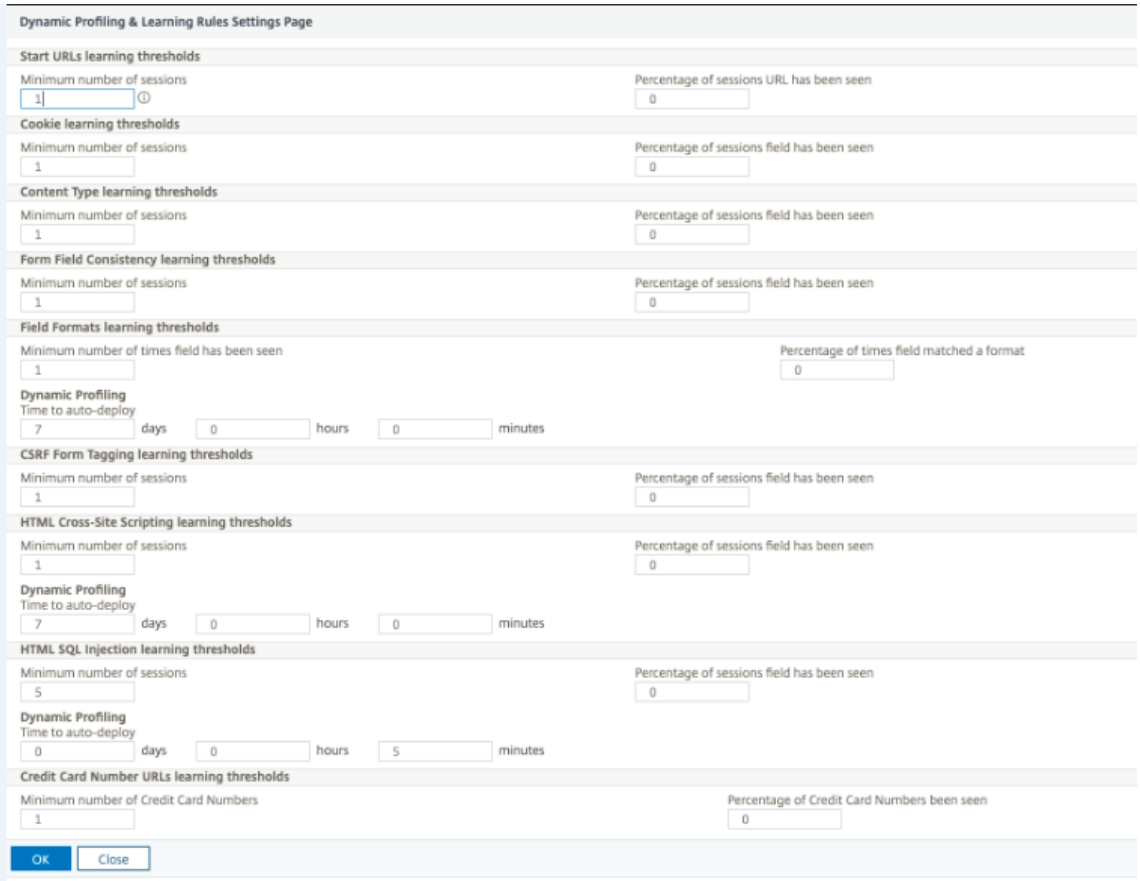
`set appfw learningsettings test1 -fieldFormatAutoDeploymentGracePeriod 10`

`set appfw learning settings test1 -SQLInjectionAutoDeploymentGracePeriod 12`

> 注:
>
> ここでは、自動展開の猶予期間は分単位です。

NetScaler GUI を使用した動的プロファイリングの構成

1. [セキュリティ] **> [NetScaler Web App Firewall] > [**プロファイル**]** に移動します。
2. 詳細ペインで、プロファイルを選択し、[編集] をクリックします。
3. **NetScaler Web App Firewall** ページで、［詳細設定］の［動的プロファイリング］をクリックします。
4. [動的プロファイリング] セクションで、セキュリティチェックを選択し、[編集] をクリックします。
5. [動的プロファイリングとラーニングの設定**]** ページで、セキュリティチェックの猶予期間を設定します。



6. [**OK**] をクリックし、[完了] をクリックします。

緩和ルールのエクスポートとインポート

動的プロファイリングを有効にすると、学習したデータは緩和ルールとして自動的に展開されます。これに加えて、アプライアンスでは、動的プロファイリングベースの緩和ルールと通常の緩和ルールをエクスポートすることもできます。ステージング環境からルールをエクスポートし、実稼働環境にインポートできます。

> 注:
>
> ルールを実稼働環境にインポートするときは、プロセスが追加され、既存の構成が上書きされないようにする必要があります。

緩和ルールをエクスポートおよびインポートする方法

緩和規則をエクスポートおよびインポートするには、次の手順を完了する必要があります。

1. まず、動的プロファイリングベースのデータをエクスポートする必要があります。このため、WAF プロファイルの緩和ルールでエクスポートオプションを使用できます。このオプションを選択すると、動的プロファイリ

ング緩和規則と通常緩和規則がエクスポートされます。エクスポートオプションを使用して、アプライアンスに圧縮バンドルとして設定をダウンロードできます。

2. ステージング環境からデータをエクスポートしたら、別の NetScaler アプライアンスにインポートする必要があります。このためには、WAF プロファイルの緩和ルールで使用可能なインポートオプションを使用する必要があります。このオプションを選択すると、アプライアンスは指定された緩和ルールをバンドルしてインポートし、選択したアプライアンスの WAF プロファイルに復元します。

> 注:
>
> 緩和ルールを WAF プロファイルにインポートする場合、2 種類のアクションがあります。
> Augment —このアクションにより、インポートが追加され、既存の設定が上書きされなくなります。
> 上書き—このアクションは、圧縮されたエクスポートバンドルに存在する設定で既存の構成を上書きします。

CLI を使用してアーカイブされた緩和ルール・ファイルをインポートする

緩和ルールをインポートするには、アーカイブを NetScaler アプライアンスにインポートし、復元コマンドを実行して構成を抽出する必要があります。次の CLI コマンドのセットは、設定のエクスポート、インポート、および管理に使用できます。

特定の場所からアーカイブファイルをインポートして復元するには、コマンドプロンプトで次のように入力します。

```
import appfw archive <src> <name> [-comment <string>]
```

ここで、
「src」: 形式で tar アーカイブファイルのソースを示し、<protocol>://<host>[:<port>][/<path>]
「name」: アーカイブ名を示します。
「comment」: このアーカイブに関連するコメント。

```
restore appfw profile <archivename> [-relaxationRules] [-importProfileName
 <string>] [-matchUrlString <string>] [-replaceUrlString <string>] [-
overwrite] [-augment]
```

ここで、
archivename: は tar アーカイブのソースを示します。これは必須の議論です。
「relaxationRules」: すべての appfw 緩和ルールをインポートするオプション。

importProfileName: リストア操作中に緩和ルールを関連付けるために作成または更新されたプロファイル名を示します。
「MatchurlString」: アーカイブされた緩和ルールで照合するアクション URL 文字列を示します。

replaceUrlString: 緩和ルールの復元中にアクション URL で置き換える文字列を示します。

overwrite: 既存の緩和ルールをパージし、インポート中に置き換えるための既存のルールアクション。

augment: インポート中に緩和ルールを強化する既存のルールアクション。

例:

```
import appfw archive local: dutA_test_pr.tgz demo
restore appfw profile dutA_test_pr
```

CLI を使用して、アーカイブされたファイルを選択したアプライアンスにエクスポートします

CLI を使用して appfw 緩和ルールをエクスポートする場合は、設定をアーカイブしてからエクスポートする必要があります。
アーカイブされたファイルをアーカイブおよびエクスポートするには、コマンドプロンプトで次のように入力します。

```
archive appfw profile <name> <archivename> [-comment <string>]
```

ここで、
archive name: は tar アーカイブのソースを示します。これは必須の議論です。
name: エクスポートする緩和ルールを含む appfw プロファイル名を示します。

```
export appfw archive <name> <target>
```

どこで、
名前。tar アーカイブの名前。これは必須の議論です。最大長:31
ターゲット。エクスポートするファイルへのパス。これは必須の議論です。最大長: 2047

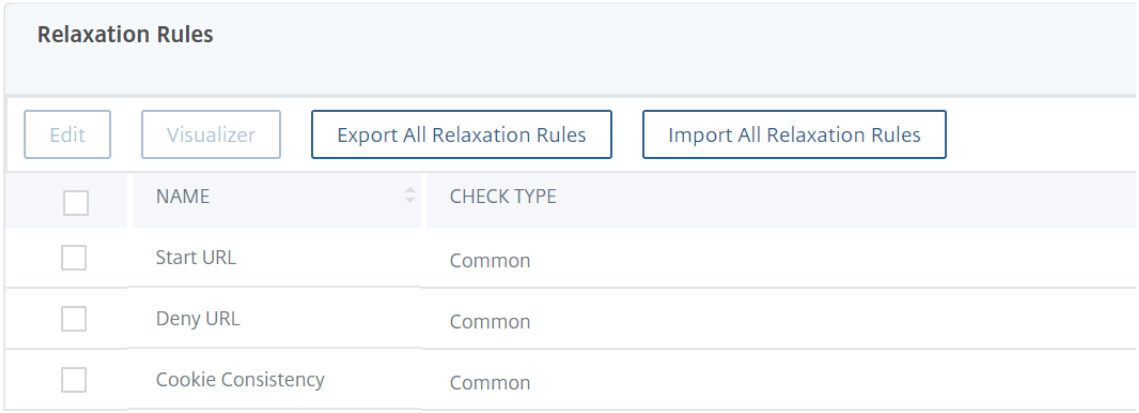
例:

```
> archive appfw profile test_pr archived_test_pr
> export appfw archive archived_test_pr local:dutA_test_pr
```

NetScaler GUI を使用して緩和ルールをエクスポートするには

緩和ルールをエクスポートするには、以下の手順に従います。

1. [セキュリティ] **>** [**NetScaler Web App Firewall**] に移動します。
2. 詳細ページで、[構成の概要] セクションの [**NetScaler Web App Firewall** プロファイル] リンクをクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、「詳細設定」セクションの「緩和ルール」リンクをクリックします。
4. [緩和規則] セクションで、[すべての緩和規則をエクスポート] をクリックします。アクションは、すべてのセキュリティチェックと、そのプロファイルで動的学習が有効になっているセキュリティチェックに適用されます。



NetScaler GUI を使用して緩和ルールをインポートするには

緩和ルールをインポートする手順を完了します。

1. ［セキュリティ］**>**［**NetScaler Web App Firewall**］に移動します。
2. 詳細ページで、［構成の概要］セクションの［**NetScaler Web App Firewall** プロファイル］リンクをクリックします。
3. **NetScaler Web App Firewall** プロファイルページで、「詳細設定」セクションの「緩和ルール」リンクをクリックします。
4. [緩和規則] セクションで、[すべての緩和規則をインポート] をクリックします。
5. ［**NetScaler Web App Firewall** プロファイルの構成］ページで、次のパラメーターを設定します。

    a) ローカルファイル。緩和ルールを含む圧縮アーカイブファイルの名前。
    b) プロファイル名。緩和ルールがバインドされているプロファイルの名前。
    c) 一致する URL 文字列。一致する URL の一部。
    d) URL 文字列を置き換えます。URL 文字列を置き換える URL の一部。
    e) 既存のルールアクション。ルールで既存のルールを上書きするか、既存のルールを補強するかを選択します。

6. ［**OK**］をクリックします。

プロファイルに関する補足情報

August 15, 2023

Web App Firewall プロファイルの特定の側面に関する補足情報を次に示します。ここでは、セキュリティチェックルールまたは緩和に特殊文字を含める方法と、プロファイルを構成するときに変数を使用する方法について説明します。

設定変数のサポート

静的な値を使用する代わりに、Web App Firewall のセキュリティチェックと設定を構成するために、標準の NetScaler 名前付き変数を使用できるようになりました。変数を作成することで、新しい NetScaler ADC アプライアンスに構成をエクスポートしてインポートしたり、単一の構成ファイルから既存の NetScaler ADC アプライアンスを更新したりすることがより簡単になります。これにより、テストベッドセットアップを使用してローカルネットワークとサーバー用に調整された複雑な Web App Firewall 構成を開発し、その構成を本番環境の NetScaler ADC アプライアンスに転送する際の更新が簡単になります。

Web App Firewall 構成変数は、NetScaler の標準的な規則に従って、ほかの NetScaler ADC の名前付き変数と同じ方法で作成します。NetScaler のコマンドラインまたは GUI を使用して、名前付き式変数を作成できます。

次の URL と式は、静的な値の代わりに変数を使用して設定できます。

- 開始 **URL** (-starurl)
- 拒否 **URL** (-denyurl)
- フォームフィールドの一貫性チェック用のフォームアクション **URL** (-fieldconsistency)
- *XML SQL* インジェクションチェックのアクション **URL** (-xmlSQlInjection)
- X*ML* クロスサイトスクリプティングチェックのアクション **URL** (-xmlcross-site スクリプティング)
- *HTML SQL* インジェクションチェック用のフォームアクション **URL** (-sqlInjection)
- フィールドフォーマットチェック用のフォームアクション **URL** (-FieldFormat)
- クロスサイトリクエストフォージェリ *(CSRF)* チェック用のフォームオリジン **URL** とフォームアクション **URL** (-csrfTag)
- *HTML* クロスサイトスクリプティングチェック用のフォームアクション **URL** (-CrossSiteScripting)
- セーフオブジェクト (-safeObject)
- *XML* サービス拒否 *(xDoS)* チェック *(-xmlDos*) のアクション **URL**
- *Web* サービスの相互運用性チェックの URL (-XMLWSIURL)
- < *XML* 検証チェックの **URL** (-xmlValidationURL)
- **XML** 添付ファイルチェック 用の *URL* (-XML 添付ファイル URL)

詳細については、「ポリシーと式」を参照してください。

構成で変数を使用するには、変数名を 2 つのアット (@) 記号で囲み、置き換える静的な値と同じように使用します。たとえば、GUI を使用して URL の拒否チェックを構成し、名前付き式変数 myDenyURL を構成に追加する場合は、[拒否 URL の追加] ダイアログボックスの [拒否 URL の拒否] テキスト領域に @myDenyURL @ と入力します。NetScaler コマンドラインを使用して同じタスクを実行するには、appfw プロファイルの追加 <name> -denyURLAction @myDenyURL @ と入力します。

PCRE 文字エンコード形式

NetScaler オペレーティングシステムでは、印刷可能な ASCII 文字セット、つまり 16 進コードが HEX 20（ASCII 32）から HEX 7E（ASCII 127）の間の文字のみの直接入力をサポートしています。Web App Firewall 構成にその範囲外のコードを持つ文字を含めるには、UTF-8 16 進コードを PCRE 正規表現として入力する必要があります。

Web App Firewall 構成に URL、フォームフィールド名、またはセーフオブジェクト式として含める場合、多くの文字タイプで PCRE 正規表現を使用したエンコードが必要です。それらには以下が含まれます。

- 上部 **ASCII** 文字。16 進数 7F (ASCII 128) から 16 進数 FF (ASCII 255) までのエンコーディングを持つ文字。使用される文字コードによっては、制御コード、アクセントまたはその他の変更を加えた ASCII 文字、非ラテンアルファベット文字、基本 ASCII セットに含まれていない記号がこれらのエンコーディングで参照される場合があります。これらの文字は、URL、フォームフィールド名、およびセーフオブジェクト式で使用できます。
- **2** バイト文字。8 バイトの単語を 2 つ使用するエンコーディングの文字。2 バイト文字は、主に中国語、日本語、韓国語のテキストを電子形式で表すために使用されます。これらの文字は、URL、フォームフィールド名、およびセーフオブジェクト式で使用できます。
- **ASCII** 制御文字。プリンタにコマンドを送信するときに使用される、印刷不能な文字です。16 進数コードが HEX 20 (ASCII 32) 未満の ASCII 文字はすべて、このカテゴリに分類されます。ただし、これらの文字は URL やフォームフィールド名には絶対に使用しないでください。セーフオブジェクト式にはほとんど使用されません。

NetScaler ADC アプライアンスは、UTF-8 文字セット全体をサポートするのではなく、次の 8 つの文字セットに含まれる文字のみをサポートします。

- 英語 **(**米国**) (ISO-8859-1)**。ラベルは「英語 US」と表示されますが、Web App Firewall は ISO-8859-1 文字セット（Latin-1 文字セットとも呼ばれる）のすべての文字をサポートしています。この文字セットは、ほとんどの現代西ヨーロッパ言語を完全に表し、残りのいくつかの珍しい文字を除いてすべてを表します。
- 繁体字中国語 **(Big5)**。Web App Firewall は BIG5 文字セットのすべての文字をサポートします。これには、香港、マカオ、台湾、および中国本土以外に住む多くの中国民族遺産の人々によって話され、書き込まれる現代中国語で一般的に使用されているすべての繁体字中国語（表意文字）が含まれます。
- 簡体字中国語 **(GB2312)**。Web App Firewall は GB2312 文字セットのすべての文字をサポートします。これには、中国本土で話され、書き込まれる現代中国語で一般的に使用されるすべての簡体字中国語（表意文字）が含まれます。
- 日本語 **(SJIS)**。Web App Firewall は Shift-JIS (SJIS) 文字セットのすべての文字をサポートします。この文字セットには、現代日本語で一般的に使用されるほとんどの文字 (表意文字) が含まれます。
- 日本語 **(EUC-JP)**。Web App Firewall は EUC-JP 文字セットのすべての文字をサポートします。これには、現代日本語で一般的に使用されるすべての文字（表意文字）が含まれます。
- 韓国語 **(EUC-KR)**。Web App Firewall は EUC-KR 文字セットのすべての文字をサポートします。これには、現代韓国語で一般的に使用されるすべての文字（表意文字）が含まれます。
- トルコ語 **(ISO-8859-9)**。Web App Firewall は、ISO-8859-9 文字セットのすべての文字をサポートします。これには、現代トルコ語で使用されるすべての文字が含まれます。
- ユニコード **(UTF-8)**。Web App Firewall は、現代ロシア語で使用されている文字を含む、UTF-8 文字セットの特定の追加文字をサポートしています。

Web App Firewall を構成するときは、UTF-8 仕様でその文字に割り当てられた 16 進コードを使用して、ASCII 以外のすべての文字を PCRE 形式の正規表現として入力します。通常の ASCII 文字セット内の記号と文字には、その文字セットで 1 桁の 2 桁のコードが割り当てられ、UTF-8 文字セットでも同じコードが割り当てられます。たとえば、感嘆符 (!)ASCII 文字セットでは 16 進コード 21 が割り当てられ、UTF-8 文字セットでは 16 進数 21 にもなります。サポートされている別の文字セットのシンボルと文字には、UTF-8 文字セットの 16 進コードのペアセットが割り当てられます。たとえば、鋭アクセント（á）の付いた文字 a には、UTF-8 コード C3 A1 が割り当てられます。

Web App Firewall 構成でこれらの UTF-8 コードを表すために使用する構文は、ASCII 文字の場合は「xnN」、英語、ロシア語、トルコ語で使用される非 ASCII 文字の場合は「\ xnN\ xNN」、中国語、日本語、韓国語で使用される文字の場合は「\ xnN\ xnN」です。たとえば、!Web App Firewall 正規表現で UTF-8 文字として入力する場合は、「\ x21」と入力します。á を含める場合は、\ xC3\ xA1 と入力します。

> 注:
>
> 通常、ASCII 文字を UTF-8 形式で表現する必要はありませんが、これらの文字が Web ブラウザや基盤となるオペレーティングシステムを混乱させる可能性がある場合は、文字の UTF-8 表現を使用してこの混乱を避けることができます。たとえば、URL にスペースが含まれている場合、特定のブラウザや Web サーバソフトウェアの混乱を避けるために、スペースを x20 としてエンコードできます。

Web App Firewall 構成に含めるには、PCRE 形式の正規表現として入力する必要がある ASCII 以外の文字を含む URL、フォームフィールド名、およびセーフオブジェクト式の例を次に示します。各例は、実際の URL、フィールド名、または式文字列を最初に示し、その後に PCRE 形式の正規表現が続きます。

- 拡張 ASCII 文字を含む URL。

  実際の URL: `http://www.josénuñez.com`

  エンコードされた URL: `^http://www\[.\]jos\xC3\xA9nu\xC3\xB1ez\[.\]com$`

- 拡張 ASCII 文字を含む別の URL。

  実際の URL: `http://www.example.de/trömso.html`

  エンコードされた URL: `^http://www[.]example\[.]de/tr\xC3\xB6mso[.]html$`

- 拡張 ASCII 文字を含むフォームフィールド名。

  `Actual Name: nome_do_usuário`

  エンコードされた名前: `^nome_do_usu\xC3\xA1rio$`

- 拡張 ASCII 文字を含むセーフオブジェクト式。

  エンコードされていない式 [A-Z] {3,6} ¥[1-9][0-9]{6,6}

  エンコードされた式:[A-Z] {3,6}\ xC2\ xA5[1-9][0-9]{6,6}

Unicode 文字セット全体と一致する UTF-8 エンコーディングを含む多くのテーブルがインターネット上で見つけることができます。この情報を含む便利な Web サイトは、次の URL にあります。

http://www.utf8-chartable.de/unicode-utf8-table.pl

この Web サイトの表に記載されている文字を正しく表示するには、適切な Unicode フォントがコンピュータにインストールされている必要があります。そうしないと、キャラクターの視覚的な表示に誤りがある可能性があります。ただし、文字を表示するための適切なフォントがインストールされていない場合でも、この Web ページの説明と UTF-8 および UTF-16 コードは正しいものになります。

反転 **PCRE** 式

パターンを含むコンテンツの照合に加えて、反転 PCRE 式を使用して、パターンを含まないコンテンツも照合できます。式を反転するには、感嘆符 (!) を含めるだけです。式の最初の文字に空白文字が続きます。

> 注**:** 式が感嘆符だけで構成され、その後に何もない場合、感嘆符は反転した式を示す構文ではなく、リテラル文字として扱われます。

次の Web App Firewall コマンドは、反転 PCRE 式をサポートしています。

- 開始 URL (URL)
- 拒否 URL (URL)
- フォームフィールドの一貫性 (フォームアクション URL)
- Cookie の一貫性 (フォームアクション URL)
- クロスサイトリクエストフォージェリ (CSRF) (フォームアクション URL)
- HTML クロスサイトスクリプティング (フォームアクション URL)
- フィールド形式 (フォームアクション URL)
- フィールドタイプ (タイプ)
- 機密フィールド (URL)

注: セキュリティチェックに isRegex フラグまたはチェックボックスが含まれている場合は、[YES] に設定するか、オンにしてフィールドで正規表現を有効にする必要があります。それ以外の場合、そのフィールドの内容はリテラルとして扱われ、正規表現 (反転または非反転) は解析されません。

Web App Firewall プロファイルで許可されていない名前

次の名前は、NetScaler ADC アプライアンスの組み込みアクションとプロファイルに割り当てられ、ユーザーが作成した Web App Firewall プロファイルの名前として使用することはできません。

- AGRESSIVE
- ALLOW
- BASIC
- CLIENTAUTH
- COMPRESS
- CSSMINIFY

- DEFLATE
- DENY
- DNS-NOP
- DROP
- GZIP
- HTMLMINIFY
- IMGOPTIMIZE
- JSMINIFY
- MODERATE
- NOCLIENTAUTH
- NOCOMPRESS
- NONE
- NOOP
- NOREWRITE
- RESET
- SETASLEARNNSLOG_ACT
- SETNSLOGPARAMS_ACT
- SETSYSLOGPARAMS_ACT
- SETTMSESSSPARAMS_ACT
- SETVPNPARAMS_ACT
- SET_PREAUTHPARAMS_ACT
- default_dns64_Action
- dns_default_act_Cachebypass
- dns_default_act_drop
- nshttp_default_profile
- nshttp_default_strict_validation
- nstcp_default_mobile_profile
- nstcp_default_XA_XD_profile
- nstcp_default_profile
- nstcp_default_tcp_interactive_stream
- nstcp_default_tcp_lan
- nstcp_default_tcp_lan_thin_stream
- nstcp_default_tcp_lfp
- nstcp_default_tcp_lfp_thin_stream
- nstcp_default_tcp_lnp
- nstcp_default_tcp_lnp_thin_stream
- nstcp_internal_apps

HTML、XML、および JSON エラーオブジェクトのカスタムエラーステータスとメッセージ

August 15, 2023

NetScaler Web App Firewall が違反を検出すると、アプライアンスはリダイレクト URL またはエラーオブジェクト（プロファイルにインポートされ、有効化）を使用してエラーシナリオを処理します。シナリオがエラーオブジェクト設定を使用して処理される場合、WAF プロファイルはカスタム応答ステータスコードとメッセージを提供します。WAF プロファイルの HTML、XML、または JSON エラーオブジェクトの応答エラーの詳細をカスタマイズできます。

> 注:
>
> デフォルトでは、エラーオブジェクトの設定が構成されている場合、エラーコードとエラーメッセージは「200」および「OK」に設定されます。

エラーシナリオを処理する場合、問題を解決するために、アプライアンスが適切な HTTP 応答ステータスコードとメッセージで応答することが重要です。カスタムのエラーステータスメッセージとカスタムエラーステータスコードを提供することにより、アプライアンスは、違反が発生したときに問題を解決するためのユーザー介入を向上させることができます。たとえば、応答エラーコードを「404」に設定し、ステータスメッセージを「Not Found」に設定すると、ユーザーは応答ステータスコードとメッセージを検査して、違反が発生したかどうかを確認できます。これにより、エラーオブジェクトを含む応答をフィルタリングできます。

CLI を使用して、WAF プロファイル内の HTML エラーオブジェクトのカスタムステータスコードとメッセージを設定します

コマンドプロンプトで入力します。

```
set appfw profile <profile-name> -HTMLErrorStatusCode <value> -
    HTMLErrorStatusMessage <value> -useHTMLErrorObject ON
<!--NeedCopy-->
```

例:

```
set appfw profile profile_1 -HTMLErrorStatusCode 404 -HTMLErrorStatusMessage
  “Not Found”  -useHTMLErrorObject ON
```

CLI を使用して、WAF プロファイルの XML エラーオブジェクトのカスタムステータスコードとメッセージを設定する

コマンドプロンプトで入力します。


```
set appfw profile <profile-name> -XMLErrorStatusCode <value> -
    XMLErrorStatusMessage <value>
<!--NeedCopy-->
```

例:

```
set appfw profile profile_1 -XMLErrorStatusCode 406 - XMLErrorStatusMessage
  “Not Acceptable”
```

CLI を使用して **WAF** プロファイルの **JSON** エラーオブジェクトのカスタムステータスコードとメッセージを構成する

コマンドプロンプトで入力します。

```
set appfw profile <profile-name> -JSONErrorStatusCode <value> -
    JSONErrorStatusMessage <value>
<!--NeedCopy-->
```

例:

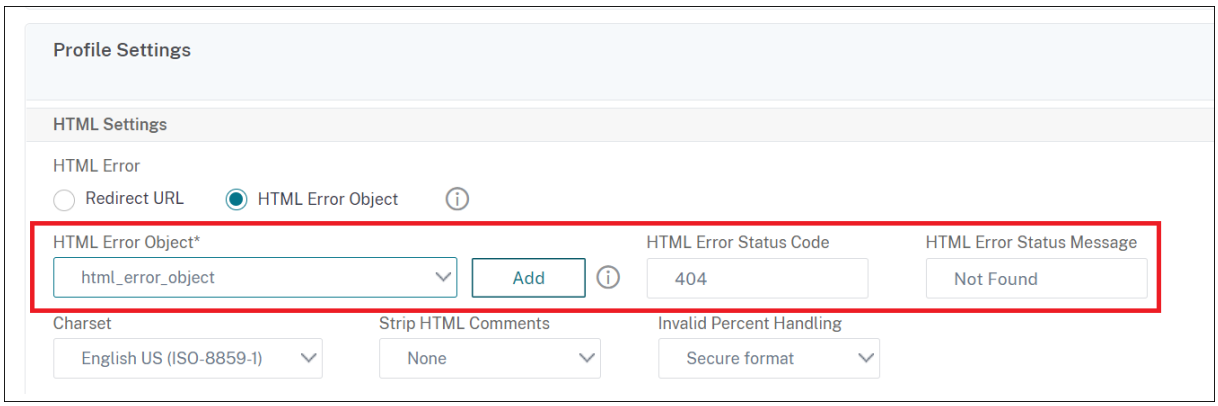
```
set appfw profile profile_1 -JSONErrorStatusCode 500 - JSONErrorStatusMessage
  “Internal Server Error”
```

GUI を使用して **WAF** プロファイルの **HTML**、**JSON**、または **XML** エラーオブジェクトのカスタムステータスコードとメッセージを構成します

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ウィンドウで、[編集] をクリックします。
3. [**Web App Firewall** プロファイルの作成] ページで、[詳細設定] セクションの **[プロファイル設定]** をクリックします。
4. [プロファイル設定] セクションで、次のパラメータを設定します。

   a. HTML エラーオブジェクト。HTML エラーオブジェクトを使用してエラーシナリオを処理するオプションを選択します。URL、ファイル、またはテキストからエラーオブジェクトをインポートします。
   b. HTML エラーステータスコード。カスタムエラーステータスコードを指定します。
   c. HTML エラーステータスメッセージ。カスタマーのエラーメッセージを入力します。
5. [**OK**] をクリックし、[完了] をクリックします。

> 注:
>
> JSON および XML カスタムエラーオブジェクト設定にも同じ手順が適用されます。



ポリシーラベル

August 15, 2023

ポリシーラベルは、一連のポリシー、その他のポリシーラベル、および仮想サーバ固有のポリシーバンクで構成されます。Web App Firewall は、ポリシーラベルにバインドされた各ポリシーを優先順位の順に評価します。ポリシーが一致すると、関連付けられたプロファイルで指定された接続をフィルタリングします。次に、Goto パラメータが指定するものは何でも実行されます。これは、ポリシー評価を終了したり、次のポリシーに移動したり、指定された優先度を持つポリシーに移動したりすることができます。Invoke パラメーターが設定されている場合、現在のポリシーラベルの処理を終了し、指定されたポリシーラベルまたは仮想サーバーの処理を開始します。

コマンドラインを使用して **Web App Firewall** ポリシーラベルを作成するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw policylabel <labelName> http_req`
- `save ns config`

例

次の例では、policylbl1 という名前のポリシーラベルを作成します。

```
1 add appfw policylabel policylbl1 http_req
2 save ns config
3 <!--NeedCopy-->
```

コマンドラインを使用してポリシーをポリシーラベルにバインドするには

コマンドプロンプトで、次のコマンドを入力します。

- `bind appfw policylabel <labelName> <policyName> <priority> [<gotoPriorityExpression>] [-invoke (<labelType> <labelName>)]`
- `save ns config`

例

次に、policy1 をプライオリティ 1 のポリシーラベル policylbl1 にバインドする例を示します。

```
bind appfw policylabel policylbl1 policy1 1
save ns config
<!--NeedCopy-->
```

GUI を使用して Web App Firewall ポリシーラベルを構成するには

1. [セキュリティ] > [NetScaler Web App Firewall] > [ポリシーラベル] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいポリシーラベルを追加するには、[**Add**] をクリックします。
   - 既存のポリシーラベルを設定するには、ポリシーラベルを選択し、[**Open**] をクリックします。

   [**Web App Firewall** ポリシーラベルの作成] または [**Web App Firewall** ポリシーラベルの構成] ダイアログボックスが開きます。ダイアログボックスはほぼ同じです。
3. 新しいポリシーラベルを作成する場合は、[Web App Firewall ポリシーラベルの作成] ダイアログボックスで、新しいポリシーラベルの名前を入力します。

   名前は、文字、数字、またはアンダースコア記号で始まり、1 ～127 の文字、数字、およびハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) 記号で構成できます。
4. [ポリシーの挿入] を 選択して新しい行を挿入し、既存のすべての Web AppFirewall ポリシーを含むドロップダウンリストを表示します。
5. ポリシーラベルにバインドするポリシーを選択するか、[New Policy] を選択して新しいポリシーを作成し、GUI を使用してポリシーを作成および構成するにはの手順に従います。選択または作成したポリシーが、グローバルにバインドされた Web App Firewall ポリシーのリストに挿入されます。
6. 追加の調整を行います。
   - ポリシーの優先度を変更するには、フィールドをクリックして有効にし、新しい優先度を入力します。「優先順位を再作成」を選択して、優先順位を均等に再設定することもできます。
   - ポリシー表現を変更するには、そのフィールドをダブルクリックして Web App Firewall ポリシーの設定ダイアログボックスを開き、ポリシー表現を編集できます。

- Goto Expression を設定するには、Goto Expression 列の見出しにあるフィールドをダブルクリックしてドロップダウンリストを表示し、そこで式を選択できます。
- [呼び出し] オプションを設定するには、[呼び出し] 列見出しのフィールドをダブルクリックしてドロップダウンリストを表示し、式を選択できます。

7. 手順 5 ～7 を繰り返して、追加の Web App Firewall ポリシーをポリシーラベルにバインドします。
8. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。ポリシーラベルが正常に作成または変更されたことを示すメッセージがステータスバーに表示されます。

ポリシー

August 15, 2023

Web App Firewall は、ファイアウォールポリシーと監査ポリシーの 2 種類のポリシーを使用します。ファイアウォールポリシーは、どのトラフィックを Web App Firewall に送信するかを制御します。監査ポリシーは、Web App Firewall のログが送信されるログサーバーを制御します。

ファイアウォールポリシーは、フィルタリングする接続を非常に正確に定義できる本格的なオブジェクト指向プログラミング言語である NetScaler 式言語では、ポリシールールが複数の式で構成される場合があるため、複雑になる可能性があります。ファイアウォールポリシーは Web App Firewall のコンテキスト内で動作するため、Web App Firewall の機能と、Web App Firewall によって適切にフィルタリングされるトラフィックに関連する特定の基準を満たす必要があります。ただし、これらの基準を念頭に置いておけば、ファイアウォールポリシーは他の NetScaler 機能のポリシーと似ています。ここで説明する手順は、ファイアウォールポリシーの作成に関するすべての側面を網羅しているわけではなく、ポリシーの概要と Web App Firewall に固有の基準を網羅しているにすぎません。

ポリシー・ルールは常に ns_true なので、監査ポリシーは非常にシンプルです。必要なのは、ログの送信先となるログサーバー、使用するログレベル、および詳細に説明するその他の条件だけです。

Web App Firewall ポリシー

August 16, 2023

ファイアウォールポリシーは、プロファイルに関連付けられたルールです。ルールは、Web App Firewall がプロファイルを適用してフィルタリングする要求/応答ペアのタイプを定義する式または式のグループです。ファイアウォールポリシー式は、特定の NetScaler ADC 機能をサポートする特別な機能を備えたオブジェクト指向プログラミング言語である NetScaler ADC 式言語で記述されます。プロファイルは、ルールに一致する要求/応答のペアをフィルタリングするために Web App Firewall が使用する一連のアクションです。

ファイアウォールポリシーを使用すると、さまざまな種類の Web コンテンツに異なるフィルタリングルールを割り当てることができます。すべてのウェブコンテンツが似ているわけではありません。複雑なスクリプトを使用せず、プライベートデータにアクセスして処理しない単純な Web サイトは、基本的なデフォルトで作成されたプロファイルによって提供される保護レベルのみを必要とする場合があります。JavaScript で強化された Web フォームを含む Web コンテンツや SQL データベースにアクセスする Web コンテンツには、おそらくよりカスタマイズされた保護が必要です。別のプロファイルを作成してそのコンテンツをフィルタリングし、そのコンテンツへのアクセスを試みている要求を決定できる別のファイアウォールポリシーを作成できます。次に、作成したプロファイルとポリシー式を関連付けて、ポリシーをグローバルにバインドして有効にします。

Web App Firewall は HTTP 接続のみを処理するため、NetScaler ADC 式言語全体のサブセットを使用します。ここで説明する情報は、Web App Firewall の構成時に役立つと思われるトピックと例に限定されています。ファイアウォールポリシーの追加情報と手順へのリンクを次に示します。

- ポリシーを作成して構成する手順については、「Web App Firewall ポリシーの作成と構成」を参照してください。
- ポリシールール (式) の作成方法を詳しく説明する手順については、「Web App Firewall ルール (式) を作成または構成するには」を参照してください。
- [式の追加] ダイアログボックスを使用してポリシールールを作成する方法については、「[式の追加] ダイアログボックスを使用してファイアウォールルール (式) を追加するには、「式の追加」を参照してください。
- ポリシーの現在のバインディングを表示する方法を説明する手順については、ファイアウォールポリシーのバインディングの表示を参照してください。
- Web App Firewall ポリシーをバインドする手順については、「Web App Firewall ポリシーのバインド」を参照してください。
- NetScaler ADC 式の言語の詳細については、「ポリシーと式」を参照してください。

> 注
>
> Web App Firewall は、設定された優先度と goto の式に基づいてポリシーを評価します。ポリシー評価の最後に、true と評価される最後のポリシーが使用され、対応するプロファイルのセキュリティ設定が呼び出されて要求が処理されます。
>
> たとえば、2 つのポリシーがあるシナリオを考えてみましょう。
>
> - Policy_1 は、式 =NS_TRUE を持つ汎用ポリシーで、基本プロファイルである対応する profile_1 があります。プライオリティは 100 に設定されます。
> - Policy_2 は expression=http.req.url.contains (「XYZ」) でより具体的であり、対応する profile_2 が事前プロファイルです。GoTO 式は NEXT に設定され、優先度は 95 に設定されます。これは Policy_1 よりも高い優先度です。
>
> このシナリオでは、処理されたリクエストの URL でターゲット文字列「XYZ」が検出されると、Policy_1 も一致していても、Policy_2 の一致がより高い優先度を持つため、Policy_2 の一致がトリガーされます。ただし、Policy_2 の goTO 式設定に従って、ポリシーの評価は続行され、次の policy_1 も処理されます。ポリシー評価の最後に、Policy_1 は true と評価され、Profile_1 で設定された基本的なセキュリティチェックが呼

> び出されます。
>
> Policy_2 が変更され、goTO 式が **NEXT** から **END** に変更された場合、ターゲット文字列「XYZ」を持つ処理された要求は、優先度の考慮事項により Policy_2 の一致をトリガーし、goTO 式の設定に従って、ポリシーの評価はで終了します。この点。Policy_2 は true と評価され、Profile_2 で設定された高度なセキュリティチェックが呼び出されます。
>
> **NEXT**
> **END**
>
> ポリシー評価は 1 回のパスで完了します。リクエストに対するポリシー評価が完了し、対応するプロファイルアクションが呼び出されると、リクエストはポリシー評価の別のラウンドを通過しません。

Web App Firewall ポリシーの作成と構成

August 15, 2023

ファイアウォールポリシーは、ルールと関連するプロファイルの 2 つの要素で構成されます。ルールは、設定した条件に一致する HTTP トラフィックを選択し、そのトラフィックを Web App Firewall に送信してフィルタリングします。プロファイルには、Web App Firewall が使用するフィルタリング基準が含まれています。

ポリシールールは、NetScaler ADC 式言語の 1 つ以上の式で構成されます。NetScaler ADC 式の構文は、特定のプロファイルで処理するトラフィックを正確に指定できる強力なオブジェクト指向プログラミング言語です。NetScaler ADC 式の言語構文に慣れていないユーザーや、Web ベースのインターフェイスを使用して NetScaler ADC アプライアンスを構成したいユーザーには、****GUI** には接頭辞メニューと式の追加ダイアログボックスの **2** つのツールが用意されています ****** 。どちらも、処理するトラフィックを正確に選択する式を記述するのに役立ちます。構文に精通している経験豊富なユーザーは、NetScaler ADC コマンドラインを使用して NetScaler ADC アプライアンスを構成することをお勧めします。

> 注:
>
> デフォルトの式の構文に加えて、下位互換性のために、NetScaler ADC オペレーティングシステムは、NetScaler Classic、nCore アプライアンスと仮想アプライアンスでの NetScaler ADC クラシック式構文をサポートしています。クラシック式は、NetScaler Cluster アプライアンスおよび仮想アプライアンスではサポートされていません。既存の構成を NetScaler ADC クラスターに移行する現在の NetScaler ADC ユーザーは、従来の式を含むポリシーをデフォルトの式の構文に移行する必要があります。

NetScaler ADC 式の言語の詳細については、「ポリシーと式」を参照してください。

ファイアウォールポリシーは、GUI または NetScaler ADC コマンドラインを使用して作成できます。

コマンドラインインターフェイスを使用してポリシーを作成して構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw policy <name><rule> <profileName>`
- `save ns config`

例

次の例では、ホスト blog.example.com との間で送受信されるすべてのトラフィックをインターセプトするルールを使用して、pl-blog という名前のポリシーを追加し、そのポリシーをプロファイル pr-blog に関連付けます。これは、特定のホスト名でホストされているブログを保護するための適切なポリシーです。

```
1 add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com
     ")" pr-blog
2 <!--NeedCopy-->
```

GUI を使用してポリシーを作成して構成するには

1. [セキュリティ] > [**Web App Firewall**] > [ポリシー] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - ファイアウォールポリシーを作成するには、[追加] をクリックします。**[Web App Firewall** ポリシーの作成] が表示されます。
   - 既存のファイアウォールポリシーを編集するには、ポリシーを選択し、[編集] をクリックします。

   [Web App Firewall ポリシーの作成**]** または **[Web App Firewall** ポリシーの構成] が表示されます。
3. ファイアウォールポリシーを作成する場合は、[**Web App Firewall** ポリシーの作成] ダイアログボックスの [ポリシー名] テキストボックスに、新しいポリシーの名前を入力します。

   名前は、文字、数字、またはアンダースコア記号で始まり、1 ～128 文字の英数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) で構成できます。

   既存のファイアウォールポリシーを構成している場合、このフィールドは読み取り専用です。これは変更できません。
4. [プロファイル（Profile）] ドロップダウンリストから、このポリシーに関連付けるプロファイルを選択します。[新規] をクリックしてポリシーに関連付けるプロファイルを作成し、[変更] をクリックして既存のプロファイルを変更できます。
5. [式] テキスト領域で、ポリシーのルールを作成します。
   - テキスト領域に直接ルールを入力できます。
   - [Prefix] をクリックしてルールの最初の用語を選択し、プロンプトに従って操作できます。

- [追加] をクリックして [式の追加] ダイアログボックスを開き、それを使用してルールを作成できます。

6.「作成」または「**OK**」をクリックし、「閉じる」をクリックします。

Web App Firewall ルール **(**式**)** を作成または構成するには

ポリシールールは、式とも呼ばれ、ポリシーに関連付けられたプロファイルを使用して Web App Firewall がフィルタリングする Web トラフィックを定義します。他の NetScaler ADC ポリシールール（または式）と同様に、Web App Firewall ルールは NetScaler ADC 式の構文を使用します。この構文は強力で柔軟性があり、拡張可能です。この一連の命令で完全に記述するには複雑すぎます。次の手順を使用して単純なファイアウォールポリシールールを作成するか、ポリシー作成プロセスの概要として読むことができます。

1. まだ作成していない場合は、**Web App Firewall** ウィザードまたは **NetScaler** GUI で適切な場所に移動してポリシールールを作成します。

   - **Web App Firewall** ウィザードでポリシーを構成している場合は、ナビゲーションペインで [**Web App Firewall**] をクリックし、詳細ペインで [**Web App Firewall Wizard**] をクリックして、[ルール の指定] 画面に移動します。
   - ポリシーを手動で構成する場合は、ナビゲーションペインで、[**Web App Firewall]**、**[**** ポリシー]、[ファイアウォール**]** の順に展開します。詳細ウィンドウで、ポリシーを作成するには、**[**** 追加] をクリックします。既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。

2. [ルールの指定] 画面、[**Web App Firewall** プロファイルの作成] ダイアログボックス、または [**Web App Firewall** プロファイルの構成] ダイアログボックスで、[接頭辞] をクリックし、ドロップダウンリストから式のプレフィックスを選択します。選択肢は次のとおりです：

   - **HTTP** プロトコルに関連するリクエストの側面を調べる場合は、HTTP プロトコルを選択します。
   - **SYS**。リクエストの受信者に関連するリクエストの側面を調べたい場合は、保護された Web サイトを選択します。
   - **CLIENT**。リクエストを送信したクライアントを選択します。リクエストの送信者の側面を調べる場合は、これを選択します。
   - サーバー。リクエストの送信先クライアントを選択し、リクエストの受信者の側面を調べるかどうかを指定します。

   プレフィックスを選択すると、Web App Firewall に 2 つの部分からなるプロンプトウィンドウが表示され、次に選択可能な選択肢が一番下部に表示されます。

3. 次の用語を選択してください。

   HTTP プロトコルをプレフィクスとして選択した場合は、ReQ しか選択できません。REQ は Request/Response のペアを指定します。(Web App Firewall は、リクエストとレスポンスをそれぞれ別々ではなくユニットとして動作します)。別のプレフィックスを選択した場合、選択肢はより多様になります。特定の選択肢に関するヘルプを表示するには、その選択肢を 1 回クリックすると、その選択に関する情報が下のプロンプトウィンドウに表示されます。

使用する用語が決まったら、ダブルクリックして [式] ウィンドウに挿入します。

4. 選択した期間の後にピリオドを入力します。次に、前の手順で説明したように、次の用語を選択するように求められます。用語で値を入力する必要がある場合は、適切な値を入力します。たとえば、HTTP.REQ.HEADER (「」) を選択した場合、引用符の間にヘッダー名を入力します。
5. 式が終了するまで、プロンプトから用語を選択し、必要な値を入力します。

   特定の目的のための式の例をいくつか挙げます。

   - 特定の **Web** ホスト。特定の Web ホストからのトラフィックを照合するには、次の手順を実行します。

     ```
     1  HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
     2  <!--NeedCopy-->
     ```

     shopping.example.comの場合は、一致させる Web ホストの名前に置き換えます。

   - 特定の **Web** フォルダまたはディレクトリ。Web ホスト上の特定のフォルダまたはディレクトリからのトラフィックを照合するには、次の手順を実行します。

     ```
     1  HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
     2  <!--NeedCopy-->
     ```

     www.example.com の場合は、ウェブホストの名前を代用してください。フォルダの場合は、一致させるコンテンツのフォルダまたはパスを置き換えます。たとえば、ショッピングカートが /solutions/orders というフォルダにある場合、その文字列をフォルダに置き換えます。

   - 特定の種類のコンテンツ**:GIF** 画像。GIF 形式の画像を一致させるには:

     ```
     1  HTTP.REQ.URL.ENDSWITH(".png")
     2  <!--NeedCopy-->
     ```

     他のフォーマットイメージを一致させるには、.png の代わりに別の文字列を置き換えます。

   - 特定の種類のコンテンツ**:** スクリプト。CGI-BIN ディレクトリにあるすべての CGI スクリプトを一致させるには、次の手順を実行します。

     ```
     1  HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
     2  <!--NeedCopy-->
     ```

     すべての JavaScript を.js 拡張子で一致させるには:

     ```
     1  HTTP.REQ.URL.ENDSWITH(".js")
     2  <!--NeedCopy-->
     ```

     ポリシー式の作成の詳細については、「ポリシーと式」を参照してください。

> 注:
>
> コマンドラインを使用してポリシーを構成する場合は、NetScaler ADC 式内の二重引用符をエスケープしてください。たとえば、GUI で入力すると、次の式は正しいです。


```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

ただし、コマンドラインで入力した場合は、代わりに次のコマンドを入力する必要があります。

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

[式の追加] ダイアログボックスを使用してファイアウォールルール (式) を追加するには

[式の追加] ダイアログボックス（式エディタとも呼ばれる）は、NetScaler ADC 式の言語に慣れていないユーザーが、フィルタリングするトラフィックに一致するポリシーを構築するのに役立ちます。

1. まだ行っていない場合は、**Web App Firewall** ウィザードまたは **Citrix** ADC GUI で適切な場所に移動します。
   - **Web App Firewall** ウィザードでポリシーを構成している場合は、ナビゲーションペインで [**Web App Firewall**] をクリックし、詳細ペインで [**Web App Firewall Wizard**] をクリックして、[ルール の指定] 画面に移動します。
   - ポリシーを手動で構成する場合は、ナビゲーションペインで、[**Web App Firewall]**、**[**** ポリシー]、[ファイアウォール**]** の順に展開します。詳細ウィンドウで、ポリシーを作成するには、**[**** 追加] をクリックします。既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。
2. [ルールの指定] 画面の [**Web App Firewall** プロファイルの作成] ダイアログボックス、または [**Web App Firewall** プロファイルの構成] ダイアログボックスで、[追加] をクリックします。
3. **[**式 の追加] ダイアログボックスの [式の構成] 領域の最初のリストボックスで、次のプレフィックスのいずれかを選択します。
   - **HTTP** HTTP プロトコルに関連するリクエストの側面を調べる場合は、[HTTP プロトコル] を選択します。デフォルトの選択肢。
   - **SYS**。リクエストの受信者に関連するリクエストの側面を調べたい場合は、保護された Web サイトを選択します。
   - **CLIENT**。リクエストの送信者の側面を調べる場合は、要求を送信したコンピュータを選択します。
   - サーバー。要求の送信先コンピュータを選択し、要求の受信者の側面を調べます。
4. 2 番目のリストボックスで、次の用語を選択します。ダイアログボックスでは、コンテキストに有効な用語のみが含まれるようにリストが自動的に調整されるため、使用可能な用語は、前の手順で行った選択によって異なります。たとえば、前のリストボックスで [HTTP] を選択した場合、リクエストに対する唯一の選択肢は [REQ] です。Web App Firewall は、リクエストと関連する応答を 1 つのユニットとして扱い、両方をフィルタリングするため、個別にレスポンスを指定する必要はありません。2 番目の用語を選択すると、第 2 項の右に 3 番目のリストボックスが表示されます。[ヘルプ] ウィンドウには 2 番目の項の説明が表示され、[式のプレビュー] ウィンドウにはエクスプレッションが表示されます。

5. 3 番目のリストボックスで、次の用語を選択します。右側に新しいリストボックスが表示され、ヘルプウィンドウが新しい用語の説明が表示されます。エクスプレッションのプレビュー (Preview Expression) ウィンドウが更新され、指定したポイントまでのエクスプレッションが表示されます。
6. 条件の選択を続け、引数の入力を求められたら、式が完成します。既に用語を選択した後に間違えたり、式を変更したい場合は、単に別の用語を選択できます。式が変更され、変更した用語の後に追加した引数またはそれ以上の用語がクリアされます。
7. 式の作成が終了したら、「**OK**」をクリックして「式の追加」ダイアログボックスを閉じます。エクスプレッションがエクスプレッションテキストエリアに挿入されます。

Web App Firewall ポリシーのバインディング

August 15, 2023

Web App Firewall ポリシーを設定したら、それらをグローバルまたはバインドポイントにバインドして有効にします。バインド後、Web App Firewall ポリシーに一致するすべての要求または応答は、そのポリシーに関連付けられたプロファイルによって変換されます。

ポリシーをバインドするときは、そのポリシーにプライオリティを割り当てます。プライオリティによって、定義したポリシーが評価される順序が決まります。優先度は、任意の正の整数に設定できます。NetScaler OS では、ポリシーの優先順位は逆の順序で機能します。数値が大きいほど、優先度は低くなります。

Web App Firewall 機能では、リクエストが一致した最初のポリシーのみが実装され、一致する可能性のある追加のポリシーは実装されないため、意図した結果を得るにはポリシーの優先順位が重要です。最初のポリシーに低い優先度（1000 など）を指定した場合、優先順位の高い他のポリシーが要求と一致しない場合にのみ実行するように Web App Firewall を構成します。最初のポリシーに高い優先度 (1 など) を与える場合は、Web App Firewall が最初に実行するように設定し、一致する可能性のある他のポリシーはすべてスキップします。ポリシーをバインドするときに、各ポリシー間に 50 または 100 の間隔で優先順位を設定することで、優先順位を再割り当てすることなく、他のポリシーを任意の順序で追加できる十分な余裕を残すことができます。

NetScaler ADC アプライアンスでのポリシーのバインドの詳細については、「ポリシーと式」を参照してください。

コマンドラインインターフェイスを使用して **Web App Firewall** ポリシーをバインドするには

コマンドプロンプトで、次のコマンドを入力します。

- `bind appfw global <policyName>`
- `bind appfw profile <profile_name> -crossSiteScripting data`

例

次の例では、pl-blog という名前のポリシーをバインドし、優先度 10 を割り当てます。


```
bind appfw global pl-blog 10
save ns config
<!--NeedCopy-->
```

ログ表現の設定

Web App Firewall をバインディングするためのログ表現のサポートが追加され、違反が発生した場合に HTTP ヘッダー情報をログに記録できます。

ログ式はアプリケーションプロファイルでバインドされ、バインディングには違反が発生した場合に評価してロギングフレームワークに送信する必要がある式が含まれます。

HTTP ヘッダー情報を含む Web App Firewall 違反ログレコードが記録されます。カスタムログ表現を指定でき、現在のフロー（リクエスト/レスポンス）で違反が生成されたときの分析と診断に役立ちます。

設定例

```
bind appfw profile <profile> -logexpression <string> <expression>
add policy expression headers "" HEADERS(100):"+HTTP.REQ.FULL_HEADER"
add policy expression body_100 ""BODY:"+HTTP.REQ.BODY(100)"
bind appfw profile test -logExpression log_body body_100
bind appfw profile test -logExpression log_headers  headers
bind appfw profile test -logExpression ""URL:"+HTTP.REQ.URL+" IP:"+
    CLIENT.IP.SRC"
<!--NeedCopy-->
```

ログの例

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg= HEADERS(100)
    :POST /test/credit.html HTTP/1.1^M User-Agent: curl/7.24.0 (amd64-
    portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Host:
     10.217.222.44^M Accept: /^M Content-Length: 33^M Content-Type:
    application/x-www-form-urlencoded^M ^M  cn1=58 cn2=174 cs1=test cs2=
    PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg=BODY:ata=
    asdadasdasdasddddddddddddddddd cn1=59 cn2=174 cs1=test cs2=PPE1 cs4=
    ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```



```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg=URL:/test/
    credit.html IP:10.217.222.128 cn1=60 cn2=174 cs1=test cs2=PPE1 cs4=
    ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Other violation logs
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_STARTURL|6|src=10.217.222.128 spt=26409 method=POST
    request=http://10.217.222.44/test/credit.html msg=Disallow Illegal
    URL. cn1=61 cn2=174 cs1=test cs2=PPE1 cs4=ALERT cs5=2017 act=not
    blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_SAFECOMMERCE|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg=Maximum
    number of potential credit card numbers seen cn1=62 cn2=174 cs1=test
     cs2=PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

> 注
>
> 1. 監査ログサポートのみ利用可能です。ログストリームのサポートとセキュリティインサイトの可視性は、今後のリリースバージョンで追加される予定です。
> 2. 監査ログが生成される場合、ログメッセージごとに生成できるのは 1024 バイトのデータだけです。
> 3. ログストリーミングを使用する場合、制限はサポートされるログストリームの最大サイズ/IPFIX プロトコルのサイズ制限に基づいています。ログストリームの最大サポートサイズは 1024 バイトを超えています。

GUI を使用して **Web App Firewall** ポリシーをバインドするには

1. 次のいずれかを行います:
   - [セキュリティ] > [**Web App Firewall**] に移動し、詳細ペインで [Web App Firewall ポリシーマネージャー] をクリックします。
   - [セキュリティ] > [**Web App Firewall**] > [ポリシー] > [ファイアウォールポリシー] に移動し、詳細ペインで [ポリシーマネージャー] をクリックします。
2. **Web App Firewall Policy Manager** ダイアログで、ポリシーをバインドするバインドポイントをドロップダウンリストから選択します。選択肢は以下のとおりです。
   - グローバルオーバーライド。このバインドポイントにバインドされたポリシーは、NetScaler アプライアンス上のすべてのインターフェイスからのすべてのトラフィックを処理し、他のポリシーよりも先に適用されます。

- **LB** 仮想サーバー。負荷分散仮想サーバーにバインドされたポリシーは、その負荷分散仮想サーバーによって処理されるトラフィックにのみ適用され、デフォルトのグローバルポリシーの前に適用されます。LB 仮想サーバーを選択したら、このポリシーをバインドする特定の負荷分散仮想サーバーも選択する必要があります。
- **CS** 仮想サーバー。コンテンツスイッチング仮想サーバーにバインドされたポリシーは、そのコンテンツスイッチング仮想サーバーによって処理されるトラフィックにのみ適用され、デフォルトのグローバルポリシーの前に適用されます。CS 仮想サーバーを選択したら、このポリシーをバインドする特定のコンテンツスイッチング仮想サーバーも選択する必要があります。
- デフォルトグローバル。このバインドポイントにバインドされたポリシーは、NetScaler アプライアンス上のすべてのインターフェイスからのすべてのトラフィックを処理します。
- ポリシーラベル。ポリシーラベルにバインドされたポリシーは、ポリシーラベルがルーティングするトラフィックを処理します。ポリシーラベルは、このトラフィックにポリシーが適用される順序を制御します。
- なし。ポリシーをどのバインドポイントにもバインドしないでください。

3. ［続行］をクリックします。既存の Web App Firewall ポリシーのリストが表示されます。
4. バインドするポリシーをクリックして選択します。
5. バインディングをさらに調整します。

   - ポリシーの優先度を変更するには、フィールドをクリックして有効にし、新しい優先度を入力します。「優先順位を再作成」を選択して、優先順位を均等に再設定することもできます。
   - ポリシー表現を変更するには、そのフィールドをダブルクリックして **Web App Firewall** ポリシーの設定ダイアログボックスを開き、ポリシー表現を編集できます。
   - Goto Expression を設定するには、Goto Expression 列の見出しにあるフィールドをダブルクリックしてドロップダウンリストを表示し、そこで式を選択できます。
   - [呼び出し] オプションを設定するには、[呼び出し] 列見出しのフィールドをダブルクリックしてドロップダウンリストを表示し、式を選択できます。

6. ステップ 3 ～6 を繰り返して、グローバルにバインドするその他の Web App Firewall ポリシーを追加します。
7. ［**OK**］をクリックします。ポリシーが正常にバインドされたことを示すメッセージがステータスバーに表示されます。

ポリシー・バインディングの表示

August 15, 2023

GUI でバインディングを表示すると、どのファイアウォールポリシーに対してどのようなバインディングが設定されているかをすばやく確認できます。

Web App Firewall ポリシーのバインディングを表示するには

1. [セキュリティ**] > [NetScaler Web App Firewall] > [**ポリシー**] > [**ファイアウォールポリシー**]** に移動します
2. 詳細ペインで、確認するポリシーを選択し、「Show Bindings」をクリックします。「ポリシーのバインディングの詳細: ポリシー」メッセージボックスに、選択したポリシーのバインディングのリストが表示されます。
3. ［閉じる］をクリックします。

Web App Firewall ポリシーに関する補足情報

August 15, 2023

Web App Firewall ポリシーの特定の側面に関する補足情報を次に示します。Web App Firewall を管理するシステム管理者が知っておく必要があるかもしれません。

正しくても予期しない動作

Web アプリケーションのセキュリティと最新の Web サイトは複雑です。さまざまなシナリオでは、NetScaler ポリシーが原因で、特定の状況で Web App Firewall の動作が、ポリシーに精通しているユーザーが通常予想するものと異なる場合があります。Web App Firewall が予期しない動作をする可能性のある、さまざまなケースを以下に示します。

- **HTTP** ホストヘッダーと絶対 **URL** が欠落しているリクエスト。ユーザーがリクエストを送信すると、ほとんどの場合、リクエスト URL は相対 URL になります。つまり、リファラー URL、つまりリクエストを送信したときにユーザーのブラウザが置かれている URL を出発点とします。リクエストが Host ヘッダーなしで相対 URL で送信された場合、そのリクエストは通常、ブロックされます。これは、HTTP 仕様に違反していることと、ホストを指定していないリクエストが状況によっては攻撃と見なされる可能性があるためです。ただし、リクエストが絶対 URL で送信された場合、Host ヘッダーがなくても、そのリクエストは Web App Firewall をバイパスして Web サーバーに転送されます。このような要求は HTTP 仕様に違反しますが、絶対 URL にはホストが含まれているため、脅威は発生しません。

監査ポリシー

March 20, 2024

監査ポリシーは、Web App Firewall セッション中に生成および記録されるメッセージを決定します。メッセージは SYSLOG 形式でローカルの NSLOG サーバまたは外部ロギングサーバに記録されます。選択したロギングレベルに基づいて、さまざまなタイプのメッセージがログに記録されます。

監査ポリシーを作成するには、まず NSLOG サーバーまたは SYSLOG サーバーを作成する必要があります。次に、ポリシーを作成し、ログタイプとログの送信先サーバーを指定します。

コマンドラインインターフェイスを使用して監査サーバーを作成するには

NSLOG サーバーと SYSLOG サーバーの 2 種類の監査サーバーを作成できます。コマンド名は異なりますが、コマンドのパラメーターは同じです。

監査サーバーを作成するには、コマンドプロンプトで次のコマンドを入力します。

- `add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat (MMDDYYYY | DDMMYYYY)] [-logFacility <logFacility>] [-tcp (NONE | ALL)] [-acl (ENABLED | DISABLED)] [-timeZone (GMT_TIME | LOCAL_TIME)] [-userDefinedAuditlog (YES | NO)] [-appflowExport (ENABLED | DISABLED)]`
- `save ns config`

例

次の例では、IP 10.124.67.91 に syslog1 という名前の syslog サーバを作成します。ログレベルは [緊急]、[緊急]、[警告] で、ログファシリティは LOCAL1 に設定され、すべての TCP 接続を記録します。

```
add audit syslogAction syslog1 10.124.67.91 -logLevel emergency
   critical warning -logFacility
LOCAL1 -tcp ALL
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して監査サーバーを変更または削除するには

- 監査サーバーを変更するには、set audit <type> コマンド、監査サーバーの名前、変更するパラメーターを新しい値とともに入力します。
- 監査サーバーを削除するには、rm audit <type> コマンドと監査サーバーの名前を入力します。

例

次の例では、syslog1 という名前の syslog サーバを変更して、エラーとアラートをログレベルに追加します。

```
set audit syslogAction syslog1 10.124.67.91 -logLevel emergency
   critical warning alert error
-logFacility LOCAL1 -tcp ALL
```



```
save ns config
<!--NeedCopy-->
```

GUI を使用して監査サーバーを作成または設定するには

1. **[**** セキュリティ] > [**NetScaler Web App Firewall**] > [ポリシー] > [監査] > [Nslog] に移動します。**
2. 「Nlog 監査」ページで、「サーバー」タブをクリックします。
3. 次のいずれかを行います:
   - 新しい監査サーバーを追加するには、「追加」をクリックします。
   - 既存の監査サーバーを変更するには、サーバーを選択し、「編集」をクリックします。
4. 「監査サーバーの作成」ページで、次のパラメータを設定します。
   - 名前
   - サーバータイプ
   - IP アドレス
   - ポート
   - ログレベル
   - ログファシリティ
   - 日付フォーマット
   - タイムゾーン
   - TCP ロギング
   - ACL ロギング
   - ユーザー設定可能なログメッセージ
   - AppFlow ロギング
   - 大規模な NAT ロギング
   - ALG メッセージロギング
   - サブスクライバーロギング
   - SSL インターセプション
   - URL フィルタリング
   - コンテンツ検査ロギング
5. **[**作成**]** して **[**閉じる**]** をクリックします。

コマンドラインインターフェイスを使用して監査ポリシーを作成するには

NSLOG ポリシーまたは SYSLOG ポリシーを作成できます。ポリシーのタイプはサーバーのタイプと一致する必要があります。2 種類のポリシーのコマンド名は異なりますが、コマンドのパラメータは同じです。

コマンドプロンプトで、次のコマンドを入力します:

- `add audit syslogPolicy <name> <-rule > <action>`

- save ns config

例

次の例では、SyslogP1 という名前のポリシーを作成して、Web App Firewall トラフィックを syslog1 という名前の syslog サーバに記録します。

```
add audit syslogPolicy syslogP1 rule "ns_true"action syslog1
save ns config
```

コマンドラインインターフェイスを使用して監査ポリシーを設定するには

コマンドプロンプトで、次のコマンドを入力します:

- set audit syslogPolicy <name> [-rule <expression>] [-action <string>]
- save ns config

例

次の例では、SyslogP1 という名前のポリシーを変更して、Web App Firewall トラフィックを syslog2 という名前の syslog サーバに記録します。

```
set audit syslogPolicy syslogP1 rule "ns_true"action syslog2
save ns config
```

GUI を使用して監査ポリシーを設定するには

1. **[**** セキュリティ] > [**NetScaler Web App Firewall**] > [ポリシー] に移動します。**
2. 詳細ペインで、「監査ログポリシー」をクリックします。
3. 「Nslog 監査」ページで、「ポリシー」タブをクリックし、次のいずれかを実行します。
   - 新しいポリシーを追加するには、[追加] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[編集] をクリックします。
4. 「監査ポリシーの作成」ページで、次のパラメータを設定します。
   - 名前
   - 監査タイプ
   - エクスプレッションタイプ
   - サーバー
5. [**Create**] をクリックします。

インポート

August 15, 2023

Web App Firewall の一部の機能は、Web App Firewall の設定時にアップロードした外部ファイルを利用します。GUI を使用して、インポートペインでこれらのファイルを管理します。このペインには、インポートできる 4 種類のファイル (HTML エラーオブジェクト、XML エラーオブジェクト、XML スキーマ、Web サービス記述言語 (WSDL) ファイル) に対応する 4 つのタブがあります。NetScaler コマンドラインを使用すると、これらの種類のファイルをインポートすることはできますが、エクスポートすることはできません。

HTML エラーオブジェクト

HTML または Web 2.0 ページへのユーザーの接続がブロックされたり、ユーザーが存在しない HTML または Web 2.0 ページを要求したりすると、Web App Firewall は HTML ベースのエラー応答をユーザーのブラウザに送信します。Web App Firewall が使用しなければならないエラーレスポンスを設定する場合、次の 2 つの選択肢があります。

- リダイレクト URL を設定できます。リダイレクト URL は、ユーザーがアクセスできる任意の Web サーバーでホストできます。たとえば、Web サーバーに 404.html というカスタムエラーページがある場合、接続がブロックされたときにユーザーをそのページにリダイレクトするように Web App Firewall を構成できます。
- HTML エラーオブジェクトを設定できます。これは Web App Firewall 自体でホストされる HTML ベースの Web ページです。このオプションを選択した場合は、HTML エラーオブジェクトを Web App Firewall にアップロードする必要があります。これは、「インポート」ペインの「HTML エラーオブジェクト」タブで行います。

エラーオブジェクトは、Web App Firewall エラーオブジェクトカスタマイズ変数以外の HTML 以外の構文を含まない標準 HTML ファイルでなければなりません。CGI スクリプト、サーバーで解析されたコード、または PHP コードを含めることはできません。カスタマイズ変数を使用すると、リクエストがブロックされたときにユーザーが受け取るエラーオブジェクトにトラブルシューティング情報を埋め込むことができます。Web App Firewall がブロックするほとんどのリクエストは違法ですが、Web App Firewall が適切に設定されていても、特に最初にデプロイしたときや、保護されているウェブサイトに大幅な変更を加えた後で、正当なリクエストがブロックされることがあります。エラーページに情報を埋め込むことで、ユーザーがテクニカルサポート担当者に提供する必要のある情報をユーザーに提供し、問題を解決できるようになります。

Web App Firewall エラーページのカスタマイズ変数は次のとおりです。

- ${NS_TRANSACTION_ID}. Web App Firewall がこのトランザクションに割り当てたトランザクション ID。
- ${NS_APPFW_SESSION_ID}. Web App Firewall セッション ID。
- ${NS_APPFW_VIOLATION_CATEGORY}. 違反した特定の Web App Firewall セキュリティチェックまたはルール。

- ${NS_APPFW_VIOLATION_LOG}. 違反に関連する詳細なエラーメッセージ。
- $ { COOKIE 指定された Cookie の内容。<CookieName>を、エラーページに表示したい特定の Cookie の名前の代わりに使用してください。トラブルシューティングのために内容を表示したい Cookie が複数ある場合は、このカスタマイズ変数の複数のインスタンスを、それぞれ適切な Cookie 名で使用できます。
  注:Cookie の整合性チェックでブロックを有効にしている場合、Web App Firewall がブロックするため、ブロックされた Cookie はエラーページに表示されません。

これらの変数を使用するには、通常のテキスト文字列であるかのように、エラーページオブジェクトの HTML または XML に埋め込みます。エラーオブジェクトがユーザーに表示されると、Web App Firewall はカスタマイズ変数ごとに、変数が参照する情報を置き換えます。カスタム変数を使用する HTML エラーページの例を以下に示します。

```
<!doctype html public "-//w3c//dtd html 4.0//en"> <html> <head> <
   title>Page Not Accessible</title> </head> <body> <h1>Page Not
   Accessible</h1> <p>The page that you accessed is not available. You
    can:</p> <ul> <li>return to the <b><a href="[homePage]">home page
   </a></b>, re-establish your session, and try again, or,</li> <li>
   report this incident to the help desk via <b><a href="mailto:[
   helpDeskEmailAddress]">email</a></b> or by calling [
   helpDeskPhoneNumber].</li> </ul> <p>If you contact the help desk,
   please provide the following information:</p> <table cellpadding=8
   width=80%> <tr><th align="right" width=30%>Transaction ID:</th><td
   align="left" valign="top" width=70%>${
 NS_TRANSACTION_ID }
 </td></tr> <tr><th align="right" width=30%>Session ID:</th><td align=
    "left" valign="top" width=70%>${
 NS_APPFW_SESSION_ID }
 </td></tr> <tr><th align="right" width=30%>Violation Category:</th><
    td align="left" valign="top" width=70%>${
 NS_APPFW_VIOLATION_CATEGORY }
 </td></tr> <tr><th align="right" width=30%>Violation Log:</th><td
    align="left" valign="top" width=70%>${
 NS_APPFW_VIOLATION_LOG }
 </td></tr> <tr><th align="right" width=30%>Cookie Name:</th><td align
    ="left" valign="top" width=70%>${
 COOKIE("[cookieName]") }
 </td></tr> </table> <body> <html>
<!--NeedCopy-->
```

このエラーページを使用するには、テキストエディターまたは HTML エディターにコピーします。NetScaler 変数と区別できるように角括弧で囲まれた次の変数は、適切なローカル情報に置き換えてください。(これらはそのままにしておきます。):

- [homePage]。ウェブサイトのホームページの URL。
- [helpDeskEmailAddress]。ユーザーにブロックインシデントの報告に使用させたいメールアドレス。
- [helpDeskPhoneNumber]。ブロッキングインシデントを報告するためにユーザーに問い合わせる電話番号。
- [cookieName]。エラーページに内容を表示したい Cookie の名前。

XML エラーオブジェクト

XML ページへのユーザーの接続がブロックされたり、存在しない XML アプリケーションを要求したりすると、Web App Firewall は XML ベースのエラー応答をユーザーのブラウザに送信します。エラーレスポンスを設定するには、XML ベースのエラーページを [インポート] ペインの [XML エラーオブジェクト] タブの Web App Firewall にアップロードします。すべての XML エラー応答は Web App Firewall でホストされます。XML アプリケーションのリダイレクト URL は設定できません。

> **
> 注 **:XML エラーオブジェクトでも HTML エラーオブジェクトと同じカスタマイズ変数を使用できます。

XML スキーマ

Web App Firewall は、XML または Web 2.0 アプリケーションに対するユーザーのリクエストに対して検証チェックを実行すると、そのリクエストをそのアプリケーションの XML スキーマまたはデザインタイプドキュメント（DTD）と照らし合わせて検証し、スキーマまたは DTD に従わないリクエストを拒否できます。XML スキーマと DTD はどちらも、特定タイプの XML ドキュメントの構造を記述する標準 XML 設定ファイルです。

WSDL

Web App Firewall は、XML SOAP ベースの Web サービスに対するユーザーのリクエストに対して検証チェックを実行すると、その Web サービスの Web サービスタイプ定義 (WSDL) ファイルと照らし合わせてリクエストを検証できます。WSDL ファイルは、特定の XML SOAP Web サービスの要素を定義する標準の XML SOAP 設定ファイルです。

ファイルのインポートとエクスポート

August 15, 2023

GUI またはコマンドラインを使用して、HTML または XML エラーオブジェクト、XML スキーマ、DTD、および WSDL を Web App Firewall にインポートできます。これらのファイルは、インポート後に Web ベースのテキスト領域で編集できます。これにより、コンピューターで変更してから再インポートしなくても、NetScaler で直接小さな変更を加えることができます。最後に、GUI を使用して、これらのファイルをコンピューターにエクスポートしたり、これらのファイルを削除したりできます。

> 注:
>
> インポートしたファイルは、コマンドラインを使用して削除またはエクスポートすることはできません。

コマンドラインインターフェイスを使用してファイルをインポートするには

コマンドプロンプトで、次のコマンドを入力します。

- `import appfw htmlerrorpage <src> <name>`
- `<save> ns config`

例

次の例では、error.html という名前のファイルから HTML エラーオブジェクトをインポートし、それに HTMLError という名前を割り当てます。

```
import htmlerrorpage error.html HTMLError
save ns config
<!--NeedCopy-->
```

GUI を使用してファイルをインポートするには

XML スキーマ、DTD、WSDL ファイル、または HTML または XML エラーオブジェクトをネットワーク上の場所からインポートする前に、NetScaler がファイルが置かれているインターネットまたは LAN コンピューターに接続できることを確認してください。そうしないと、ファイルまたはオブジェクトをインポートできません。

1. [** セキュリティ] > [NetScaler Web App Firewall] **[インポート] に移動します。
2. [アプリケーションファイアウォール] > [インポート] に移動します。
3. 「アプリケーションファイアウォールのインポート」ペインで、インポートするファイルの種類のタブを選択し、「追加」をクリックします。

   タブには、HTML エラーページ、XML エラーページ、XML スキーマ、または WSDL があります。ユーザーの観点から見ると、アップロードプロセスは 4 つのタブすべてで同じです。
4. ダイアログのフィールドに入力します。

   - 名前-インポートされたオブジェクトの名前。
   - インポート元-インポートする HTML ファイル、XML ファイル、XML スキーマ、または WSDL の場所をドロップダウンリストから選択します。
     - **URL:** アプライアンスにアクセスできるウェブサイト上のウェブ URL。
     - ファイル**:** ローカルまたはネットワーク上のハードディスクまたはその他のストレージデバイス上のファイル。
     - テキスト**:** カスタムレスポンスのテキストを GUI のテキストフィールドに直接入力または貼り付けます。

   3 番目のテキストボックスが適切な値に変わります。指定できる 3 つの値を以下に示します。

- **URL**—テキストボックスに URL を入力します。
- ファイル-HTML ファイルへのパスとファイル名を直接入力するか、「ブラウズ」をクリックして HTML ファイルをブラウズします。
- テキスト—3 番目のフィールドが削除され、空白のスペースが残ります。

5. [続行] をクリックします。[ファイルの内容] ダイアログが表示されます。[URL] または [ファイル] を選択した場合、[ファイルの内容] テキストボックスには、指定した HTML ファイルが含まれます。「テキスト」を選択した場合、「ファイルの内容」テキストボックスは空です。
6. 「テキスト」を選択した場合は、インポートするカスタムレスポンス HTML を入力するか、コピーして貼り付けます。
7. [完了] をクリックします。
8. オブジェクトを削除するには、オブジェクトを選択し、[削除] をクリックします。

GUI を使用してファイルをエクスポートするには

XML スキーマ、DTD、WSDL ファイル、または HTML または XML エラーオブジェクトをエクスポートする前に、Web App Firewall アプライアンスがファイルを保存するコンピュータにアクセスできることを確認してください。そうしないと、ファイルをエクスポートできません。

1. [セキュリティ] > [**Web App Firewall**] > [インポート] に移動します。
2. **Web App Firewall** のインポートペインで、エクスポートするファイルの種類のタブを選択します。

   ユーザーの観点から見ると、エクスポートプロセスは 4 つのタブすべてで同じです。
3. エクスポートするファイルを選択します。
4. 「アクション」ドロップダウンリストを展開し、「エクスポート」を選択します。
5. ダイアログボックスで、「ファイルを保存」を選択し、「**OK**」をクリックします。
6. 「ブラウズ」ダイアログ・ボックスで、エクスポートしたファイルを保存するローカル・ファイル・システムおよびディレクトリに移動し、「保存」をクリックします。

GUI で **HTML** または **XML** エラーオブジェクトを編集するには

HTML および XML エラーオブジェクトのテキストは、エクスポートしてから再インポートせずに GUI で編集します。

1. [セキュリティ**] > [NetScaler Web App Firewall] > [**インポート] に移動し、変更するファイルの種類のタブを選択します。

2. [アプリケーションファイアウォール] > [インポート] に移動し、変更するファイルの種類のタブを選択します。
3. 変更するファイルを選択し、[編集] をクリックします。

   HTML または XML エラーオブジェクトのテキストは、ブラウザのテキスト領域に表示されます。テキストは、ブラウザの標準ブラウザベースの編集ツールと方法を使用して変更できます。

   注: 編集ウィンドウは、HTML または XML エラーオブジェクトに小さな変更を加えることができるように設計されています。大幅な変更を行うには、エラーオブジェクトをローカルコンピューターにエクスポートし、標準の HTML または XML Web ページ編集ツールを使用することをお勧めします。
4. 「**OK**」をクリックし、「閉じる」をクリックします。

グローバル設定

August 15, 2023

Web App Firewall のグローバル構成は、すべてのプロファイルとポリシーに影響します。グローバル構成項目は次のとおりです。

- エンジン設定。特定の接続のサブセットではなく、Web App Firewall が処理するすべての接続に関連するグローバル設定 (セッション Cookie 名、セッションタイムアウト、最大セッション有効期間、ロギングヘッダー名、未定義プロファイル、デフォルトプロファイル、インポートサイズ制限) のコレクション。
- 機密フィールド。Web App Firewall ログに記録してはならない機密情報を含む Web フォームの一連のフォームフィールド。ログオンページのパスワードフィールドやショッピングカートのチェックアウトフォームのクレジットカード情報などのフォームフィールドは、通常、機密フィールドとして指定されます。
- フィールドタイプ。フィールド形式のセキュリティチェックで使用される Web フォームフィールドタイプのリスト。これらのフィールドタイプはそれぞれ、データのタイプと、そのタイプのフォームフィールドで使用できるデータの最小/最大長を定義する PCRE 準拠の正規表現によって定義されます。
- **XML** コンテンツタイプ。XML として認識され、XML 固有のセキュリティチェックの対象となるコンテンツタイプのリスト。これらのコンテンツタイプは、そのコンテンツに割り当てられる正確な MIME タイプを定義する PCRE 準拠の正規表現によって定義されます。
- **JSON** コンテンツタイプ。JSON として認識され、JSON 固有のセキュリティチェックの対象となるコンテンツタイプのリスト。これらのコンテンツタイプは、そのコンテンツに割り当てられる正確な MIME タイプを定義する PCRE 準拠の正規表現によって定義されます。

エンジン設定

March 20, 2024

エンジン設定は、NetScaler Web App Firewall が処理するすべての要求と応答に影響します。設定は次のとおりです:

- **Cookie** 名—NetScaler セッション ID を保存するクッキーの名前。
- セッションタイムアウト：許可される非アクティブ期間の最大値。この時間にわたってユーザーセッションにアクティビティが表示されない場合、そのセッションは終了し、ユーザーは指定された開始ページにアクセスしてセッションを再確立する必要があります。
- **Cookie** の暗号化後のプレフィックス—暗号化された Cookie の暗号化された部分の前に置く文字列。
- 最大セッション有効期間：セッションをライブ状態に保つことができる最大時間（秒単位）。この期間に達すると、セッションは終了し、ユーザーは指定された開始ページにアクセスしてセッションを再確立する必要があります。この設定はセッションタイムアウトより小さくすることはできません。この設定を無効にして、セッションの有効期間が最大にならないようにするには、値をゼロ (0) に設定します。
- ロギングヘッダー名—ロギング用のクライアント IP を保持する HTTP ヘッダーの名前。
- 未定義プロファイル：対応するポリシーアクションが未定義と評価されたときに適用されるプロファイル。
- デフォルトプロファイル：ポリシーに一致しない接続に適用されるプロファイル。
- インポートサイズ制限：署名、WSDL、スキーマ、HTML、XML エラーページを含む、アプライアンスにインポートされるすべてのファイルの最大バイト数。インポート中に、インポートされたオブジェクトのサイズが原因で、インポートされたすべてのファイルの累積数が設定された制限を超えると、インポート操作は失敗します。また、アプライアンスには次のエラーメッセージが表示されます。「エラー：インポートに失敗しました-インポートされたオブジェクトに設定された合計サイズ制限を超えています」。
- 学習メッセージレート制限—学習エンジンが処理する 1 秒あたりの要求と応答の最大数。この制限を超える追加のリクエストやレスポンスは、ラーニングエンジンに送信されません。
- プロキシサーバー -プロキシサーバーは、ユーザーに代わってインターネットからデータを取得する中間サーバーです。アプライアンスのセキュリティをさらに強化します。プロキシ認証が有効になっている NetScaler アプライアンスは、インターネットからアップデートをダウンロードする前に、プロキシサーバーで認証を行います。これにより、アプライアンスを悪意のあるダウンロードから保護します。次のパラメータを設定します。
  - プロキシサーバー—最新の AWS シグネチャのダウンロード元となるプロキシサーバーの IP アドレス。
  - プロキシポート -最新の AWS シグネチャのダウンロード元となるプロキシサーバーのポート番号。
  - プロキシユーザー名 -最新の AWS シグネチャのダウンロード元となるプロキシサーバーのポート番号。
  - プロキシパスワード -シグネチャアップデートをダウンロードするためにプロキシサーバを認証するためのパスワード。
- エンティティのデコード—Web App Firewall のチェックを実行する際に HTML エンティティをデコードします。
- 不正な形式のリクエストをログに記録する—無効な形式の **HTTP** リクエストのロギングを有効にします。
- 設定可能な秘密鍵を使用—Web App Firewall の操作には、設定可能な秘密鍵を使用します。この秘密鍵は、データの署名と検証に使用されます。「UseConfigurableSecretKey」がオンになっている場合は、「set ns EncryptionParams」パラメーターで有効になっているキーを使用する必要があります。
- 学習データをリセット—学習したすべてのデータを Web App Firewall から削除します。新しいデータを収

集して学習プロセスを再開します。

「学習済みデータのリセット」と「署名の自動更新」の 2 つの設定は、コマンドインターフェイスを使用するか、NetScaler GUI を使用して NetScaler Web App Firewall を構成するかによって、場所が異なります。コマンドインターフェイスを使用する場合は、reset appfw learning data コマンドを使用して学習データのリセットを設定します。これはパラメーターを必要とせず、他の機能もありません。シグニチャ自動更新は set appfw settings コマンドで設定できます。-signatureAutoUpdate パラメータは署名の自動更新を有効または無効にし、-signatureURL は更新された署名ファイルをホストする URL を設定します。

NetScaler GUI を使用する場合は、**[セキュリティ] > [NetScaler Web App Firewall] > [**エンジン設定] で [学習済みデータのリセット] を構成します。[学習データをリセット] オプションはダイアログボックスの下部にあります。**[**** セキュリティ] > [**NetScaler Web App Firewall**] > [署名] で署名セットごとに署名の自動更新を構成するには、署名ファイルを選択してマウスの右ボタンをクリックし、[自動更新設定] を選択します。**

通常、**Web App Firewall** 設定のデフォルト値は正しいです。ただし、デフォルト設定が他のサーバーと競合したり、ユーザーの接続が途中で切断されたりする場合は、設定を変更する必要があります。

Web App Firewall のセッション制限は、次のコマンドを使用して設定できます。

```
> set appfw settings -sessionLimit 500000

Done

Default value:100000   Max value:500000 per PE
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してエンジン設定を行うには

コマンドプロンプトで、次のコマンドを入力します:

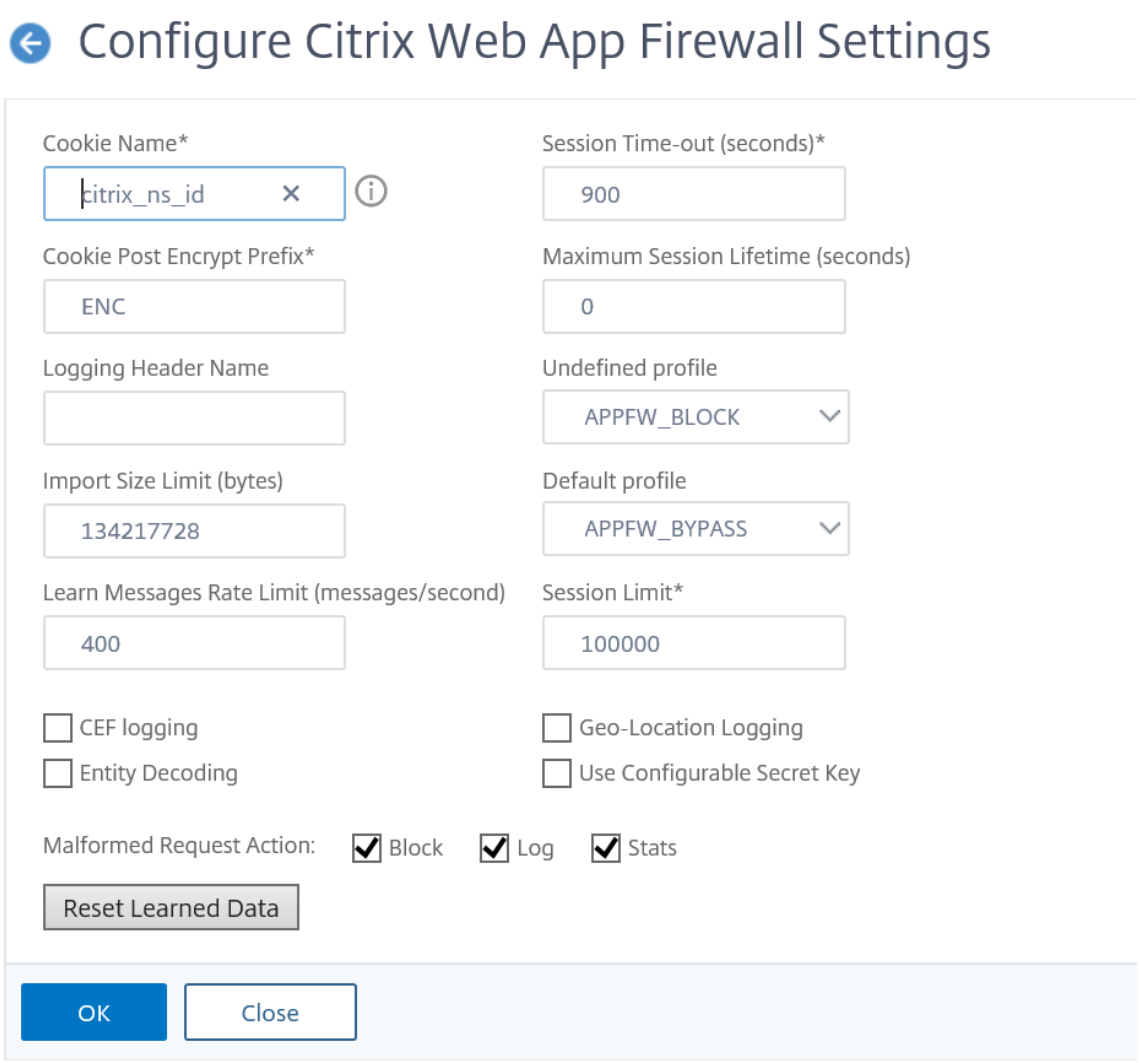
- `set appfw settings [-sessionCookieName <name>] [-sessionTimeout <positiveInteger>] [-sessionLifetime <positiveInteger>][-clientIPLoggingHeader <headerName>] [-undefaction <profileName>] [-defaultProfile <profileName>] [-importSizeLimit <positiveInteger>] [-logMalformedReq (ON | OFF)] [-signatureAutoUpdate (ON | OFF)] [-signatureUrl <expression>] [-cookiePostEncryptPrefix <string>] [-entityDecoding (ON | OFF)] [-useConfigurableSecretKey (ON | OFF)][-learnRateLimit <positiveInteger>] [-proxyServer <proxy server ip>] [-proxyPort <proxy server port>] [-proxyUsername <username>] [-proxyPassword <password>]`
- `save ns config`

例


```
set appfw settings -sessionCookieName citrix-appfw-id -sessionTimeout
    3600
-sessionLifetime 14400 -clientIPLoggingHeader NS-AppFW-Client-IP -
    undefaction APPFW_RESET
-defaultProfile APPFW_RESET -importSizeLimit 4096 -proxyServer
    10.102.30.112 -proxyPort 3128 -proxyUsername defaultusername -
    proxyPassword defaultpassword
save ns config
<!--NeedCopy-->
```

NetScaler GUI を使用してエンジン設定を構成するには

1. [セキュリティ] > [**NetScaler Web**App Firewall] に移動します
2. 詳細ペインの **[**設定**]** の **[** エンジン設定の変更**]** をクリックします。
3. **Web App Firewall** エンジン設定ダイアログボックスで、次のパラメータを設定します。
   - クッキー名
   - セッションタイムアウト
   - クッキーポスト暗号化プレフィックス
   - 最大セッション有効期間
   - ロギングヘッダー名
   - 未定義のプロファイル
   - デフォルトプロファイル
   - インポートサイズ制限
   - ラーニング・メッセージ・レート制限
   - プロキシサーバー
   - プロキシポート
   - プロキシユーザー名
   - プロキシパスワード
   - エンティティデコード
   - 不正な形式のリクエストをログに記録
   - シークレットキーを使う
   - Learn メッセージレート制限
   - シグネチャ自動更新
4. [**OK**] をクリックします。



機密フィールド

August 15, 2023

ユーザーが入力した情報を保護するために、Web フォームフィールドを機密情報として指定できます。通常、保護されている Web サーバーの 1 つでユーザーが Web フォームに入力した情報は、NetScaler ADC ログに記録されます。ただし、機密情報として指定された Web フォームフィールドに入力された情報はログに記録されません。その情報は、ウェブサイトがそのようなデータを保存するように設定されている場合にのみ保存されます。通常は安全なデータベースに保存されます。

機密フィールド指定で保護する必要がある一般的な情報の種類は次のとおりです。

- パスワード

- クレジットカード番号、認証コード、有効期限
- 社会保障番号
- 納税者番号
- 自宅住所
- 個人電話番号

良い習慣であることに加えて、e コマースサーバーでの PCI-DSS コンプライアンス、米国の医療情報を管理するサーバーでの HIPAA コンプライアンス、および他のデータ保護基準への準拠のために、機密フィールド指定の適切な使用が必要になる場合があります。

> 重要:
>
> 次の 2 つのケースでは、機密フィールドの指定が期待どおりに機能しません。

- Web フォームに機密フィールドまたは 256 文字を超えるアクション URL がある場合、そのフィールドまたはアクション URL は NetScaler ADC ログで切り捨てられます。
- 特定の SSL トランザクションでは、機密フィールドまたはアクション URL のいずれかが 127 文字を超えると、ログが切り捨てられます。

いずれの場合も、Web App Firewall は 15 文字の文字列を通常の 8 文字の文字列ではなく、文字「x」でマスクします。機密情報を確実に削除するには、ユーザーは最初の 256 文字または (SSL が使用されている場合) 最初の 127 文字に一致するフォームフィールド名とアクション URL 式を使用する必要があります。

保護された Web サイトの Web フォームフィールドを機密情報として扱うように Web App Firewall を構成するには、そのフィールドを [機密フィールド] リストに追加します。フィールド名を文字列として入力するか、1 つ以上のフィールドを指定する PCRE 互換の正規表現を入力できます。機密フィールドの指定は、フィールドを追加するときに有効にすることも、後で指定を変更することもできます。

> 注:
>
> リリース 13.1 ビルド 27.x から、機密フィールドは WAF プロファイルでもサポートされています。詳細については、WAF プロファイルの機密フィールドを参照してください。

コマンドラインインターフェイスを使用して機密フィールドを追加するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw confidField <fieldName> <url> [-isRegex ( REGEX | NOTREGEX )] [-comment "<string>"] [-state ( ENABLED | DISABLED )]`
- `save ns config`

例

次の使用例は、名前が Password で始まるすべての Web フォームフィールドを機密フィールドリストに追加します。


```
add appfw confidField Password "https?://www[.]example[.]com/[^<>]\*[^a-z]password[0-9a-z._-]\*[.](asp|cgi|htm|html|htp|js|php)" -isRegex REGEX -state ENABLED
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して機密フィールドを変更するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw confidField <fieldName> <url> [-isRegex ( REGEX | NOTREGEX )][-comment "<string>"] [-state ( ENABLED | DISABLED )]`
- `save ns config`

例

次の例では、機密フィールドの指定を変更してコメントを追加します。

```
set appfw confidField Password "https?://www[.]example[.]com/[^<>]\*[^a-z]password[0-9a-z._-]\*[.](asp|cgi|htm|html|htp|js|php)" -comment "Protect password fields." -isRegex REGEX -state ENABLED
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して機密フィールドを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `rm appfw confidField <fieldName> <url>`
- `save ns config`

GUI を使用して機密フィールドを構成するには

1. **Security** > **Application Firewall** に移動します。
2. 詳細ペインの [設定] で、[機密フィールドの管理] をクリックします。
3. [機密フィールドの管理] ダイアログボックスで、次のいずれかの操作を行います。
   - 新しいフォームフィールドをリストに追加するには、[追加] をクリックします。
   - 既存の機密フィールドの指定を変更するには、フィールドを選択し、[編集] をクリックします。[**Web App Firewall** の機密フィールド] ダイアログボックスが表示されます。

> 注記:
>
> 既存の機密フィールドの指定を選択して [追加] をクリックすると、[機密フォームフィールドの作成] ダイアログボックスにその機密フィールドの情報が表示されます。その情報を変更して、新しい機密項目を作成できます。

4. ダイアログボックスで、要素を入力します。これには、次の種類のアカウントがあります。
   - **[**有効**]** チェックボックス。この機密フィールド指定を有効または無効にするにはオンまたはオフにします。
   - フォームフィールド名は正規表現のチェックボックスですか。フォームフィールド名で PCRE 形式の正規表現を有効にする場合は、オンまたはオフにします。
   - フィールド名。特定のフィールド名を表す、またはパターンに従う名前を持つ複数のフィールドに一致する、リテラル文字列または PCRE 形式の正規表現を入力します。
   - アクション **URL**。機密フィールドを含む Web フォームが配置されている Web ページの 1 つ以上の URL を定義するリテラル URL または正規表現を入力します。
   - **[**コメント**]**。コメントを入力します。オプションです。
5. [作成] または [**OK**] をクリックします。
6. 機密フィールドリストから機密フィールドの指定を削除するには、削除する機密フィールドのリストを選択し、「削除」をクリックして削除し、「**OK**」をクリックして選択を確定します。
7. 機密項目指定の追加、変更、および削除が完了したら、[閉じる] をクリックします。

例

以下は、便利なフォームフィールド名を定義する正規表現です。

- `^passwd_ (Applies confidential-field status to all field names that begin with the “passwd_” string.)`
- `^((\[0-9a-zA-Z._-]*||\\x\[0-9A-Fa-f][0-9A-Fa-f])+-)?passwd_ (Applies confidential-field status to all field names that begin with the string passwd_, or that contain the string -passwd_ after another string that might contain non-ASCII special characters.)`

以下に、役に立つと思われる特定の URL タイプを定義する正規表現をいくつか示します。例にあるものを自分のウェブホストとドメインに置き換えてください。

- Web フォームが Web ホスト www.example.com の複数の Web ページに表示されているが、それらの Web ページはすべて logon.pl? では、次の正規表現を使用できます。

```
1  https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)*logon
       [.]pl?
2  <!--NeedCopy-->
```


- Web フォームが Web ホスト www.example-español.com の複数の Web ページに表示され、その中に n-チルダ (ñ) 特殊文字が含まれている場合は、次の正規表現を使用できます。この正規表現は、n チルダの特殊文字を、C3 B1 を含むエンコードされた UTF-8 文字列として表します。UTF-8 文字セットの文字:

```
1 https?://www[.]example-espa\xC3\xB1ol[.]com/([0-9A-Za-z][0-9A-Za-
     z_.-]\*/)\* logon[.]pl?
2 <!--NeedCopy-->
```

- query.pl を含む Web フォームが example.com ドメイン内の異なるホスト上の複数の Web ページに表示されている場合は、次の正規表現を使用できます。

```
1 https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])\*example[.]com/([0-9A-Za
     -z][0-9A-Za-z_-.]\*/)*logon[.]pl?
2 <!--NeedCopy-->
```

- query.pl を含む Web フォームが、異なるドメインの異なるホスト上の複数の Web ページに表示される場合は、次の正規表現を使用できます。

```
1 https?://([0-9A-Za-z][0-9A-Za-z_-.]\*[.])\*[0-9A-Za-z][0-9A-Za-z_
     -.]+[.][a-z]{
2 2,6 }
3 /([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl?
4 <!--NeedCopy-->
```

- Web フォームが Web ホスト www.example.com の複数の Web ページに表示されているが、それらの Web ページはすべて logon.pl? では、次の正規表現を使用できます。

```
1 https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon
     [.]pl?
2 <!--NeedCopy-->
```

フィールドタイプ

August 15, 2023

フィールドタイプは、Web フォームのフォームフィールドの特定のデータ形式と最小/最大データ長を定義する PCRE 形式の正規表現です。フィールドタイプはフィールドフォーマットチェックで使用されます。

Web App Firewall には、次のようないくつかのデフォルトフィールドタイプがあります。

- 整数. 数字のみで構成され、小数点を含まず、オプションで先頭にマイナス記号 (-) を付けた任意の長さの文字列。
- アルファ。文字のみで構成される任意の長さの文字列。
- アルファナム。文字または数字から成る任意の長さの文字列。

- nohtml。句読点やスペースを含む任意の長さの文字列で、HTML シンボルやクエリを含まない。
- 任意。何でもいい

> 重要:
>
> 任意のフィールドタイプをデフォルトのフィールドタイプとして、またはフィールドに割り当てると、アクティブなスクリプト、SQL コマンド、およびその他の危険性のあるコンテンツを、そのフォームフィールド内の保護された Web サイトやアプリケーションに送信できます。any タイプを使用する場合は慎重に使用する必要があります。

フィールドタイプリストに独自のフィールドタイプを追加することもできます。たとえば、お住まいの国の社会保障番号、郵便番号、電話番号のフィールドタイプを追加したい場合があります。また、顧客 ID 番号または店舗のクレジットカード番号のフィールドタイプを追加したい場合もあります。

フィールドタイプのリストにフィールドタイプを追加するには、フィールド名をリテラル文字列または PCRE 形式の正規表現として入力します。

コマンドラインインターフェイスを使用してフィールドタイプを追加するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

例

次の例では、米国の社会保障番号と一致する SSN という名前のフィールドタイプをフィールドタイプリストに追加し、その優先度を 1 に設定します。

```
add appfw fieldType SSN "^[1-9][0-9]{
 2,2 }
 -[0-9 }
 {
 2,2 }
 -[0-9]{
 4,4 }
 $" 1
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してフィールドタイプを変更するには

コマンドプロンプトで、次のコマンドを入力します。

- set appfw fieldType <name> <regex> <priority> [-comment "<string>"]
- save ns config

例

次の例では、フィールドタイプを変更してコメントを追加します。

```
set appfw fieldType SSN "^[1-9][0-9]{
 2,2 }
 -[0-9 }
 {
 2,2 }
 -[0-9]{
 4,4 }
 $" 1 -comment "US Social Security Number"
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してフィールドタイプを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- >rm appfw fieldType <name>
- save ns config

GUI を使用してフィールドタイプを設定するには

1. [セキュリティ] > [アプリケーションファイアウォール] に移動します。
2. 詳細ウィンドウの [設定] で、[フィールドの種類の管理] をクリックします。
3. [フィールド型の管理] ダイアログボックスで、次のいずれかの操作を行います。
   - 新しいフィールドタイプをリストに追加するには、[追加] をクリックします。
   - 既存のフィールドの種類を変更するには、フィールドの種類を選択し、[編集] をクリックします。

   [フィールドの種類の構成] ダイアログボックスが表示されます。

   > 注:
   >
   > 既存のフィールドタイプの指定を選択し、[追加] をクリックすると、そのフィールドタイプの情報がダイアログボックスに表示されます。その情報を変更して、新しいフィールドタイプを作成できます。

4. ダイアログボックスで、要素を入力します。これには、次の種類のアカウントがあります。

- 名前
- 正規表現
- 優先度
- コメント

5. [作成] または [OK] をクリックします。
6. [フィールドの種類] ボックスの一覧からフィールドの種類を削除するには、削除するフィールドの種類の一覧を選択し、[削除] をクリックして削除し、[**OK**] をクリックして選択を確定します。
7. フィールドタイプの追加、変更、および削除が完了したら、[閉じる] をクリックします。

例

以下に、役に立つと思われるフィールドタイプの正規表現をいくつか紹介します。

`^[1-9][0-9]{ 2,2 } -[0-9 } { 2,2 } -[0-9]{ 4,4 } $` 米国社会保障番号

`^\[A-C\]\[0-9\]{ 7,7 } $` カリフォルニア州の運転免許証番号

`^+[0-9]{ 1,3 } [0-9()-]{ 1,40 } $` 国コード付きの国際電話番号

`^[0-9]{ 5,5 } -[0-9]{ 4,4 } $` 郵便番号

`^[0-9A-Za-z][0-9A-Za-z.+_-]{ 0,25 } @([0-9A-Za-z][0-9A-Za-z_-]*[.]){ 1,4 } [A-Za-z]{ 2,6 } $` メール アドレス

XML コンテンツタイプ

August 15, 2023

デフォルトでは、Web App Firewall は特定の命名規則に従うファイルを XML として扱います。Web コンテンツを調べて、それらのファイルが XML ファイルであることを示す追加の文字列やパターンがないかどうかを確認するように Web App Firewall を構成できます。これにより、特定の XML コンテンツが通常の XML 命名規則に従っていない場合でも、Web App Firewall がサイト上のすべての XML コンテンツを認識し、XML コンテンツが XML セキュリティチェックの対象となるようになります。

XML コンテンツタイプを設定するには、適切なパターンを XML コンテンツタイプリストに追加します。コンテンツタイプを文字列として入力することも、1 つ以上の文字列を指定する PCRE 互換の正規表現を入力することもできます。既存の XML コンテンツタイプのパターンを変更することもできます。

コマンドラインインターフェイスを使用して **XML** コンテンツタイプパターンを追加するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw XMLContentType <XMLContenttypevalue> [-isRegex ( REGEX | NOTREGEX )]`
- `save ns config`

例

次の例では、パターンを追加します。*/xml を XML コンテンツタイプリストに追加し、正規表現として指定します。

```
1 add appfw XMLContentType ".*/xml" -isRegex REGEX
2 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **XML** コンテンツタイプパターンを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `rm appfw XMLContentType <XMLContenttypevalue>`
- `save ns config`

GUI を使用して **XML** コンテンツタイプリストを構成するには

1. [セキュリティ] > [**Web App Firewall**] に移動します。
2. 詳細ウィンドウで、[設定] の [**XML** コンテンツタイプの管理] をクリックします。
3. 「**XML** コンテンツタイプの管理」ダイアログボックスで、次のいずれかを実行します。
   - 新しい XML コンテンツタイプを追加するには、「追加」をクリックします。
   - 既存の XML コンテンツタイプを変更するには、そのタイプを選択して [編集] をクリックします。[Web App Firewall の XML コンテンツタイプの設定] ダイアログが表示されます。注: 既存の XML コンテンツタイプパターンを選択して [追加] をクリックすると、ダイアログボックスにその XML コンテンツタイプパターンの情報が表示されます。この情報を変更して、新しい XML コンテンツタイプパターンを作成できます。
4. ダイアログボックスで、要素を入力します。これには、次の種類のアカウントがあります。
   - **IsRegex.** フォームフィールド名で PCRE 形式の正規表現を有効にする場合は、オンまたはオフにします。
   - **XML** コンテンツタイプ追加する XML コンテンツタイプパターンと一致するリテラル文字列または PCRE 形式の正規表現を入力します。
5. [作成] をクリックします。
6. XML コンテンツタイプパターンをリストから削除するには、そのパターンを選択し、[削除] をクリックして削除し、[**OK**] をクリックして選択を確定します。
7. XML コンテンツタイプパターンの追加と削除が完了したら、[閉じる] をクリックします。

JSON コンテンツタイプ

August 15, 2023

デフォルトでは、Web App Firewall はコンテンツタイプ「application/json」のファイルを JSON ファイルとして扱います。デフォルト設定により、Web App Firewall は要求と応答の JSON コンテンツを認識し、そのコンテンツを適切に処理できます。

Web コンテンツを調べて、それらのファイルが JSON ファイルであることを示す追加の文字列やパターンがないかどうかを確認するように Web App Firewall を構成できます。これにより、特定の JSON コンテンツが通常の JSON 命名規則に従っていない場合でも、Web App Firewall がサイト上のすべての JSON コンテンツを認識し、JSON コンテンツが JSON セキュリティチェックの対象となるようになります。

JSON コンテンツタイプを設定するには、JSON コンテンツタイプリストに適切なパターンを追加します。コンテンツタイプを文字列として入力することも、1 つ以上の文字列を指定する PCRE 互換の正規表現を入力することもできます。既存の JSON コンテンツタイプのパターンを変更することもできます。

コマンドラインインターフェイスを使用して JSON コンテンツタイプパターンを追加するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw JSONContentType <JSONContenttypevalue> [-isRegex ( REGEX | NOTREGEX )]`
- `save ns config`

例

次の例では、パターンを追加します。*/json を JSON コンテンツタイプリストに追加し、正規表現として指定します。

```
add appfw JSONContentType ".*/json" -isRegex REGEX
<!--NeedCopy-->
```

GUI を使用して JSON コンテンツタイプリストを設定するには

1. [セキュリティ] > [アプリケーションファイアウォール] に移動します。
2. 詳細ペインの [設定] で、[**JSON** コンテンツタイプの管理] をクリックします。
3. JSON コンテンツタイプの管理ダイアログボックスで、次のいずれかを実行します。
    - 新しい JSON コンテンツタイプを追加するには、「追加」をクリックします。

- 既存の JSON コンテンツタイプを変更するには、そのタイプを選択して [編集] をクリックします。
  [Web App Firewall の JSON コンテンツタイプの設定] ダイアログが表示されます。
  注: 既存の JSON コンテンツタイプパターンを選択して [追加] をクリックすると、ダイアログボックスにその JSON コンテンツタイプパターンの情報が表示されます。この情報を変更して、新しい JSON コンテンツタイプパターンを作成できます。

4. ダイアログボックスで、要素を入力します。これには、次の種類のアカウントがあります。

   - **IsRegex.** フォームフィールド名で PCRE 形式の正規表現を有効にする場合は、オンまたはオフにします。
   - **JSON** コンテンツタイプ追加する JSON コンテンツタイプパターンと一致するリテラル文字列または PCRE 形式の正規表現を入力します。

5. [作成] または [**OK**] をクリックします。
6. JSON コンテンツタイプパターンをリストから削除するには、そのパターンを選択し、「削除」をクリックして削除し、「**OK**」をクリックして選択を確定します。
7. XML コンテンツタイプパターンの追加と削除が完了したら、[閉じる] をクリックします。

統計とレポート

August 15, 2023

ログと統計に保持され、レポートに表示される情報は、Web App Firewall の構成と保守に関する重要なガイダンスを提供します。

Web App Firewall 統計情報

Web App Firewall シグニチャまたはセキュリティチェックの統計アクションを有効にすると、Web App Firewall は、そのシグニチャまたはセキュリティチェックに一致する接続に関する情報を保持します。[グループの選択] リストボックスで次のいずれかのオプションを選択すると、[
監視] タブで累積統計情報を表示できます。

- **Web App Firewall**。Web App Firewall アプライアンスがすべてのプロファイルについて収集したすべての統計情報のサマリー。
- **Web App Firewall (**プロファイルごと**)**。同じ情報ですが、要約ではなくプロファイルごとに表示されます。

この情報を使用して、Web App Firewall の動作を監視し、シグニチャまたはセキュリティチェックに異常なアクティビティまたは異常なヒット数があるかどうかを判断できます。このような異常なアクティビティのパターンが見られる場合は、ログでそのシグニチャまたはセキュリティチェックをチェックして、診断して是正措置を講じることができます。

リラクゼーションヒット統計カウンター

違反したトラフィックに適用された緩和に基づいて、アプライアンスで違反が発生した回数、違反時に適用された緩和ルールの数、最後に適用されたタイムスタンプなどの統計的詳細を表示することもできます。これを実行することにより、集中学習エンジンは未使用または冗長な緩和バインディングを自動的に削除できます。詳細については、「WAF Learn Engine 」トピックを参照してください。

リラクゼーションヒット統計カウンターは、以下のセキュリティチェックでのみ使用できます。

- クロスサイトスクリプティング
- SQL インジェクション
- Cookie の整合性
- JSON SQL
- JSON クロスサイトスクリプティング
- JSON Do
- JSON CMD インジェクション
- クロスサイトリクエストフォージェリ
- フィールド形式
- Starturl
- Denyurl
- コンテンツタイプ保護

CLI を使用して緩和ルールヒットカウンタの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat appfw profile p1
```

例**:**

```
stat appfw profile p1 -fullvalues
```

スタートルール統計

規則	ヒット	レート	最終ヒット時間
87a4…51177	0	0	木…1970 年
5b83…dc12a	0	0	木…1970 年
12345	0	0	木…1970 年

GUI を使用して緩和ルールヒットカウンタの統計情報を表示するには

緩和ルールヒットカウンタの統計情報を表示するには、次の手順を実行します。

1. [** セキュリティ] > [NetScaler Web App Firewall] **[プロファイル] に移動します。
2. 詳細ペインで **Web App Firewall** プロファイルを選択し、「統計」をクリックします。
3. **NetScaler Web App Firewall** の統計情報ページには、統計の詳細が表示されます。
4. 表形式ビューを選択するか、グラフィカルビューに切り替えて、データを表形式またはグラフィカル形式で表示できます。

Web App Firewall レポート

Web App Firewall レポートは、Web App Firewall の設定と、保護された Web サイトのトラフィック処理方法に関する情報を提供します。

PCI DSS レポート

ペイメントカード業界 (PCI) データセキュリティ規格 (DSS) バージョン 1.2 は、ほとんどのクレジットカード会社が、クレジットカードやデビットカードによるオンライン決済を受け入れる企業が満たす必要がある 12 のセキュリティ基準で構成されています。この基準は、個人情報の盗難、ハッキング、その他の詐欺を防止するために設計されています。ISP が PCI DSS 基準を満たさない場合、ISP またはマーチャントは、Web サイトを通じたクレジットカード決済の承認を失う可能性があります。

ISP とオンライン加盟店は、PCI DSS 認定セキュリティ評価者 (QSA) 会社による監査を実施することにより、PCI DSS に準拠していることを証明します。PCI DSS レポートは、監査の前と監査中の両方を支援するように設計されています。監査の前に、PCI DSS に関連するウェブアプリケーションファイアウォールの設定、それらの構成方法、および現在の Web App Firewall 構成が標準を満たしているかどうか（最も重要なこと）が示されます。監査中、レポートを使用して、関連する PCI DSS 基準への準拠を実証できます。

PCI DSS レポートは、Web App Firewall 構成に関連する基準のリストで構成されます。各基準の下には、現在の構成オプション、現在の構成が PCI DSS 基準に準拠しているかどうかを示し、保護された Web サイトが基準に準拠するように Web App Firewall を構成する方法について説明します。

PCI DSS レポートは、[システム**] > [**レポート] の下にあります。レポートを Adobe PDF ファイルとして生成するには、「**PCI DSS** レポートを生成」をクリックします。ブラウザの設定に応じて、レポートがポップアップウィンドウに表示されるか、ハードディスクに保存するように求められます。

> 注:
>
> このレポートやその他のレポートを表示するには、コンピューターに Adobe Reader プログラムをインストールする必要があります。

PCI DSS レポートは、次のセクションで構成されています。

- **[**説明**]**。PCI DSS コンプライアンス概要レポートの説明。
- ファイアウォールのライセンスと機能のステータス。Web App Firewall が NetScaler ADC アプライアンスでライセンスされ、有効になっているかどうかを示します。

- エグゼクティブサマリー。PCI DSS 基準をリストし、これらの基準のうちどれが Web App Firewall に関連しているかを示す表。
- **PCI DSS** 基準の詳細情報。Web App Firewall 構成に関連する各 PCI DSS 基準について、PCI DSS レポートには、構成が準拠しているかどうか、および準拠していない場合はそれを準拠させる方法に関する情報を含むセクションが表示されます。
- 構成。個々のプロファイルのデータ。レポートの上部にある [Web App Firewall 構成] をクリックするか、[レポート] ペインから直接アクセスします。Web App Firewall 構成レポートは PCI DSS レポートと同じで、PCI DSS 固有の概要は省略されています。

Web App Firewall 構成レポート

Web App Firewall 構成レポートは、[システム**] > [**レポート] の下にあります。これを表示するには、[**Web App Firewall** 構成レポートの生成] をクリックします。ブラウザの設定に応じて、レポートがポップアップウィンドウに表示されるか、ハードディスクに保存するように求められます。

Web App Firewall 構成レポートは、次のセクションで構成される [概要] ページから始まります。

- **Web App Firewall** ポリシー。現在の Web App Firewall ポリシーをリストした表。ポリシー名、ポリシーの内容、関連付けられているアクション（またはプロファイル）、およびグローバルバインド情報を示します。
- **Web App Firewall** プロファイル。現在の Web App Firewall プロファイルを一覧表示し、各プロファイルが関連付けられているポリシーを示す表。プロファイルがポリシーに関連付けられていない場合、テーブルには、その場所に INACTIVE と表示されます。

すべてのポリシーのすべてのレポートページをダウンロードするには、[プロファイルの概要] ページの上部にある [すべてのプロファイルのダウンロード] をクリックします。各プロファイルのレポートページを表示するには、画面下部の表でそのプロファイルを選択します。個々のプロファイルの [プロファイル（Profile）] ページには、各チェックに対して各チェックアクションが有効か無効になっているか、およびチェックの他の構成設定が表示されます。

現在のプロファイルの PCI DSS レポートページを含む PDF ファイルをダウンロードするには、ページの上部にある [現在のプロファイルのダウンロード] をクリックします。[プロファイルの概要] ページに戻るには、[**Web App Firewall** プロファイル] をクリックします。メインページに戻るには、[ホーム] をクリックします。PCI DSS レポートは、ブラウザの右上隅にある [**Refresh**] をクリックしていつでも更新できます。

Web App Firewall ログ

March 20, 2024

Web App Firewall は、構成、ポリシー呼び出し、セキュリティチェック違反の詳細を追跡するためのログメッセージを生成します。

セキュリティチェックまたは署名のログアクションを有効にすると、Web App Firewall が Web サイトやアプリケーションを保護する際に監視した要求と応答に関する情報がログメッセージに表示されます。最も重要な情報は、署名またはセキュリティチェック違反が観察されたときに Web App Firewall によって実行されるアクションです。一部のセキュリティチェックでは、ユーザーの場所や違反をトリガーした検出パターンなど、ログメッセージに有用な情報が記録されることがあります。ログ内の違反メッセージの数が過度に増加すると、悪意のある要求が急増している可能性があります。このメッセージは、Web App Firewall 保護によって検出および阻止された特定の脆弱性を悪用するために、アプリケーションが攻撃を受けている可能性があることを警告します。

> 注:
>
> NetScaler Web App Firewall ログをシステムログから分離する場合は、外部の SYSLOG サーバーを使用する必要があります。

NetScaler（ネイティブ）形式のログ

Web App Firewall は、デフォルトで NetScaler 形式のログ（ネイティブ形式のログとも呼ばれます）を使用します。これらのログは、他の NetScaler 機能で生成されるログと同じ形式です。各ログには、次のフィールドが含まれます。

- タイムスタンプ。接続が発生した日時。
- 重大度。ログの重大度レベル。
- モジュール。ログエントリを生成した NetScaler モジュール。
- イベントタイプ。署名違反やセキュリティチェック違反などのイベントのタイプ。
- イベント ID。イベントに割り当てられた ID。
- クライアント IP。接続がログに記録されたユーザーの IP アドレス。
- トランザクション ID。ログの原因となったトランザクションに割り当てられた ID。
- セッション ID。ログの原因となったユーザーセッションに割り当てられた ID。
- メッセージ。ログメッセージ。ログエントリをトリガーした署名またはセキュリティチェックを識別する情報が含まれます。

これらのフィールドのいずれか、または異なるフィールドの情報を組み合わせて検索できます。選択できるのは、ログの表示に使用するツールの機能によってのみ制限されます。NetScaler syslog ビューアにアクセスして GUI で Web App Firewall ログメッセージを確認したり、NetScaler アプライアンスに手動で接続してコマンドラインインターフェイスからログにアクセスしたり、シェルにドロップして/var/log/folderから直接ログを追跡したりできます。

ネイティブ形式のログメッセージの例

```
1 Jun 22 19:14:37 <local0.info> 10.217.31.98 06/22/2015:19:14:37 GMT ns
     0-PPE-1 :
2 default APPFW APPFW_cross-site scripting 60 0 :  10.217.253.62 616-PPE1
      y/3upt2K8ySWWId3Kavbxyni7Rw0000
3 pr_ffc http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
4 12345&drinking_pref=on&text_area=%3Cscript%3E%0D%0A&loginButton=
     ClickToLogin&as_sfid=
```



```
AAAAAAWEXcNQLlSokNmqaYF6dvfqlChNzSMsdyO9JXOJomm2v
BwAMOqZIChv21EcgBc3rexIUcfm0vckKlsgoOeC_BArx1Ic4NLxxkWMtrJe4H7SOfkiv9NL7AG4juPIan

%3D&as_fid=feeec8758b41740eedeeb6b35b85dfd3d5def30c Cross-site script
    check failed for
field text_area="Bad tag: script" <blocked>
<!--NeedCopy-->
```

共通イベントフォーマット **(CEF)** ログ

Web App Firewall は CEF ログもサポートします。CEF は、さまざまなセキュリティ、ネットワークデバイス、アプリケーションからのセキュリティ関連情報の相互運用性を向上させるオープンログ管理標準です。CEF を使用すると、お客様は共通のイベントログ形式を使用できるため、エンタープライズ管理システムによる分析のためにデータを簡単に収集および集約できます。ログメッセージはさまざまなフィールドに分割されるため、メッセージを容易に解析し、重要な情報を識別するためのスクリプトを記述できます。

CEF ログメッセージの分析

Web App Firewall CEF ログメッセージには、日付、タイムスタンプ、クライアント IP、ログ形式、アプライアンス、会社、ビルドバージョン、モジュール、セキュリティチェック情報に加えて、次の詳細が含まれます。

- src —送信元 IP アドレス
- spt —送信元ポート番号
- リクエスト—リクエスト URL
- act —アクション (ブロックされた、変換されたなど)
- msg —message (監視されたセキュリティチェック違反に関するメッセージ)
- オフセット-ファイルの先頭からのバイト数を表します。
- cn1 —イベント ID
- cn2 —HTTP トランザクション ID
- cs1 —プロファイル名
- cs2 —PPE ID (PPE1 など)
- cs3-セッション ID
- cs4：重大度（情報、アラートなど）
- cs5 —イベント年
- cs6-署名違反カテゴリ
- method —メソッド (GET/POST など)

たとえば、開始 URL 違反がトリガーされたときに生成された次の CEF 形式のログメッセージを考えてみましょう。

```
Jun 12 23:37:17 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0
|APPFW|APPFW_STARTURL|6|src=10.217.253.62 spt=47606 method=GET
request=http://aaron.stratum8.net/FFC/login.html msg=Disallow Illegal
    URL. cn1=1340
```



```
cn2=653 cs1=pr_ffc cs2=PPE1 cs3=EsdGd3VD0OaaURLcZnj05Y6DOmE0002 cs4=
    ALERT cs5=2015
act=blocked
<!--NeedCopy-->
```

上記のメッセージは、異なるコンポーネントに分解することができます。CEP ログコンポーネントの表を参照してください 。

CEF ログ形式の要求チェック違反の例：要求がブロックされない

```
Jun 13 00:21:28 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_FIELDCONSISTENCY|6|src=10.217.253.62 spt=761 method=GET request=
http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
123456789234&drinking_pref=on&text_area=&loginButton=ClickToLogin&
    as_sfid
=
    AAAAAAWIahZuYoIFbjBhYMP05mJLTwEfIY0a7AKGMg3jIBaKmwtK4t7M7lNxOgj7Gmd3SZc8KUj6CP

7W5kIWDRHN8PtK1Zc-txHkHNx1WknuG9DzTuM7t1THhluevXu9I4kp8%3D&as_fid=
    feeec8758b4174
0eedeeb6b35b85dfd3d5def30c msg=Field consistency check failed for field
     passwd cn1=1401
cn2=707 cs1=pr_ffc cs2=PPE1 cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=
    ALERT cs5=2015 act=
not blocked
<!--NeedCopy-->
```

CEF 形式の応答チェック違反の例: 応答が変換される

```
Jun 13 00:25:31 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_SAFECOMMERCE|6|src=10.217.253.62 spt=34041 method=GET request=
http://aaron.stratum8.net/FFC/CreditCardMind.html msg=Maximum number of
     potential credit
card numbers seen cn1=1470 cn2=708 cs1=pr_ffc cs2=PPE1
cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

CEF 形式のリクエスト側署名違反の例: リクエストはブロックされています

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method=GET request=
http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=Signature
    violation rule ID 807:
web-cgi /wwwboard/passwd.txt access  cn1=140 cn2=841 cs1=pr_ffc cs2=
    PPE0
cs3=OyTgjbXBqcpBFeENKDlde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=
    blocked
<!--NeedCopy-->
```

オフセットの CEF 形式のレスポンスチェック違反の例:


```
Jan 24 10:00:00 <local0.warn> 10.175.4.47 CEF:0|Citrix|NetScaler|NS13
    .0|APPFW|APPFW_XML_ERR_NOT_WELLFORMED|4|src=5.31.100.129 spt=20644
    method=GET request=https://wifiuae.duwifi.ae/publishApplications/en
    /5dafe3e74fa8015599009bc1/images/fallback_photo.svg msg=XML Format
    check failed: Message is not a well-formed XML.Error string is '
    unclosed token'. Offset:-517597 cn1=547290214 cn2=974226675 cs1=
    WIFI_UAE_AppFw cs2=PPE0 cs4=ERROR cs5=2023 act=blocked
<!--NeedCopy-->
```

この例では、次の理由により XML_ERR_NOT_WELLFORMED 違反が発生しています。`unclosed token` この違反は、ファイルの先頭から 517597 の場所にあります。

Web App Firewall 違反メッセージにジオロケーションを記録する

ログの詳細は、リクエストの発信元を特定し、Web App Firewall を最適なセキュリティレベルに設定するのに役立ちます。クライアントの IP アドレスに依存するレート制限などのセキュリティ実装を回避するために、マルウェアまたは不正なコンピュータは、要求の送信元 IP アドレスを変更し続けることができます。リクエストが送信される特定のリージョンを特定すると、リクエストが有効なユーザーからのものか、サイバー攻撃を開始しようとしているデバイスからのものかを判断するのに役立ちます。たとえば、特定のエリアから過度に多数の要求を受信した場合、それらがユーザーによって送信されているのか、不正なマシンによって送信されているのかを簡単に判断できます。受信したトラフィックのジオロケーション分析は、DoS（DoS; サービス拒否）攻撃などの攻撃を回避するのに役立ちます。

Web App Firewall は、組み込みの NetScaler データベースを使用して、悪意のある要求の発信元の IP アドレスに対応する場所を特定する利便性を提供します。その後、これらの場所からのリクエストに対して、より高いレベルのセキュリティを適用できます。NetScaler のデフォルト構文（PI）式を使用すると、ロケーションベースのポリシーを柔軟に設定できます。これを組み込みの位置データベースで使用してファイアウォールの保護をカスタマイズできるため、特定の地域の不正クライアントから仕掛けられた協調攻撃に対する防御が強化されます。

NetScaler 組み込みデータベースを使用することも、他のデータベースを使用することもできます。データベースに特定のクライアント IP アドレスのロケーション情報がない場合、CEF ログにはジオロケーションが不明なジオロケーションとして表示されます。

> 注:
>
> 位置情報ロギングでは、共通イベント形式 (CEF) が使用されます。デフォルトでは、`CEF logging`および`GeoLocationLogging`はオフになっています。両方のパラメータを明示的に有効にする必要があります。

位置情報を示す CEF ログメッセージの例

```
June 8 00:21:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_STARTURL|6|src=10.217.253.62 geolocation=NorthAmerica.US.Arizona.
    Tucson.*.*
```



```
spt=18655 method=GET request=http://aaron.stratum8.net/FFC/login.html
msg=Disallow Illegal URL. cn1=77 cn2=1547 cs1=test_pr_adv cs2=PPE1
cs3=KDynjg1pbFtfhC/nt0rBU1o/Tyg0001 cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```

ジオロケーション = 不明を示すログメッセージの例

```
June 9 23:50:53 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|
APPFW|APPFW_STARTURL|6|src=10.217.30.251 geolocation=Unknown spt=5086
method=GET request=http://aaron.stratum8.net/FFC/login.html msg=
    Disallow Illegal URL.
cn1=74 cn2=1576 cs1=test_pr_adv cs2=PPE2 cs3=
    PyR0eOEM4gf6GJiTyauiHByL88E0002
cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```

コマンドインターフェイスを使用してログアクションとその他のログパラメータを設定する

コマンドラインを使用してプロファイルのセキュリティー検査のログアクションを構成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します:

- `set appfw profile <name> SecurityCheckAction ([log] | [none])`
- `unset appfw profile <name> SecurityCheckAction`

例

```
set appfw profile pr_ffc StartURLAction log
```

```
unset appfw profile pr_ffc StartURLAction
```

コマンドラインを使用して CEF ロギングを設定するには

CEF ロギングは、デフォルトでは無効になっています。コマンドプロンプトで、次のいずれかのコマンドを入力して、現在の設定を変更または表示します。

- `set appfw settings CEFLogging on`
- `unset appfw settings CEFLogging`
- `sh appfw settings | grep CEFLogging`

コマンドラインを使用してクレジットカード番号のロギングを構成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します:

- `set appfw profile <name> -doSecureCreditCardLogging ([ON] | [OFF])`
- `unset appfw profile <name> -doSecureCreditCardLogging`

コマンドラインを使用して位置情報ログを構成するには

1. set コマンドを使用して GeoLocationLogging を有効にします。CEF ロギングは同時に有効にできます。位置情報ロギングを無効にするには、unset コマンドを使用します。show コマンドは、すべての Web App Firewall パラメータの現在の設定を表示します。ただし、grep コマンドを指定して特定のパラメータの設定を表示する場合を除きます。

   - `set appfw settings GeoLocationLogging ON [CEFLogging ON]`
   - `unset appfw settings GeoLocationLogging`
   - `sh appfw settings | grep GeoLocationLogging`

2. データベースを指定する

   ```
   add locationfile /var/netscaler/inbuilt_db/Citrix_netscaler_InBuilt_GeoIP_
   .csv
   ```

   または

   ```
   add locationfile <path to database file>
   ```

Web App Firewall ログをカスタマイズする

デフォルトフォーマット (PI) 式を使用すると、ログに含まれる情報を柔軟にカスタマイズできます。Web App Firewall で生成されたログメッセージに、キャプチャする特定のデータを含めるオプションがあります。たとえば、Web App Firewall セキュリティチェックとともに AAA-TM 認証を使用していて、セキュリティチェック違反をトリガーしたアクセスされた URL、URL を要求したユーザーの名前、送信元 IP アドレス、およびユーザーが要求を送信した送信元ポートを知りたい場合、次のようになります。では、次のコマンドを使用して、すべてのデータを含むカスタマイズされたログメッセージを指定できます。

```
> sh version
NetScaler NS12.1: Build 50.0013.nc, Date: Aug 28 2018, 10:51:08   (64-
   bit)
 Done
<!--NeedCopy-->
```

```
> add audit messageaction custom1 ALERT 'HTTP.REQ.URL + " " + HTTP.REQ.
   USER.NAME + " " + CLIENT.IP.SRC + ":" + CLIENT.TCP.SRCPORT'
Warning: HTTP.REQ.USER has been deprecated. Use AAA.USER instead.
 Done
<!--NeedCopy-->
```

```
> add appfw profile test_profile
 Done
<!--NeedCopy-->
```

```
> add appfw policy appfw_pol true test_profile -logAction custom1
 Done
<!--NeedCopy-->
```


Web App Firewall ログを分離するように **Syslog** ポリシーを構成する

Web App Firewall には、Web App Firewall セキュリティログメッセージを分離して別のログファイルにリダイレクトするオプションがあります。これは、Web App Firewall が多数のログを生成し、他の NetScaler ログメッセージを表示しにくい場合に適しています。このオプションは、Web App Firewall ログメッセージのみを表示し、他のログメッセージを表示したくない場合にも使用できます。

Web App Firewall ログを別のログファイルにリダイレクトするには、Web App Firewall ログを別のログファシリティに送信するように syslog アクションを構成します。このアクションは、syslog ポリシーを構成するときに使用し、Web App Firewall で使用するためにグローバルにバインドできます。

> 注:
>
> Web App Firewall ポリシーをグローバルにバインドするには、「bind audit syslogGlobal」および「bind audit nsLogGlobal」コマンドでグローバルバインディングパラメータ「APPFW_GLOBAL」を構成できます。グローバルバインドされた監査ログポリシーは、Web App Firewall のログコンテキストでログメッセージを評価できます。

例:

1. シェルに切り替え、vi などのエディタを使用して /etc/syslog.conf ファイルを編集します。次の例に示すように、local2.* を使用してログを別のファイルに送信する新しいエントリを追加します。

   ```
   local2.\* /var/log/ns.log.appfw
   ```

2. syslog プロセスを再起動します。次の例に示すように、grep コマンドを使用して syslog プロセス ID（PID）を識別できます。

   ```
   root@ns\\# **ps -A | grep syslog**

   1063 ?? Ss 0:03.00 /usr/sbin/syslogd -b 127.0.0.1 -n -v -v -8 -C

   root@ns# **kill -HUP** 1063
   ```

3. コマンドラインインターフェイスから、アクションを含む詳細またはクラシック SYSLOG ポリシーを構成し、グローバル Web App Firewall ポリシーとしてバインドします。高度な SYSLOG ポリシーを構成することをお勧めします。

   SYSLOG ポリシーの詳細設定

   ```
   add audit syslogAction sysact1 1.1.1.1 -logLevel ALL -logFacility
    LOCAL2

   add audit syslogPolicy syspol1 true sysact1

   bind audit syslogGlobal -policyName syspol1 -priority 100 -
   globalBindType APPFW_GLOBAL
   ```

   クラシック SYSLOG ポリシー設定


```
add audit syslogAction sysact1 1.1.1.1 -logLevel ALL -logFacility
 LOCAL2

add audit syslogPolicy syspol1 ns_true sysact1

bind appfw global syspol1 100
```

4. Web App Firewall のセキュリティチェック違反はすべて、/var/log/ns.log.appfwファイルにリダイレクトされます。このファイルを末尾に付けて、進行中のトラフィックの処理中にトリガーされる Web App Firewall 違反を表示できます。

   ```
   root@ns# tail -f ns.log.appfw
   ```

警告: ログを別のログファシリティにリダイレクトするように syslog ポリシーを構成した場合、Web App Firewall のログメッセージは/var/log/ns.logファイルに表示されなくなります。

> 注:
>
> ローカルの NetScaler アプライアンス上の別のログファイルにログを送信する場合は、そのローカル NetScaler アプライアンスに syslog サーバーを作成できます。独自の IP にsyslogactionを追加し、外部サーバーの設定と同様に ADC を構成します。ADC はログを保存するサーバーとして機能します。同じ IP とポートで 2 つのアクションを追加することはできません。syslogactionでは、デフォルトでは、IP の値は127.0.0.1に設定され、port の値は514に設定されます。

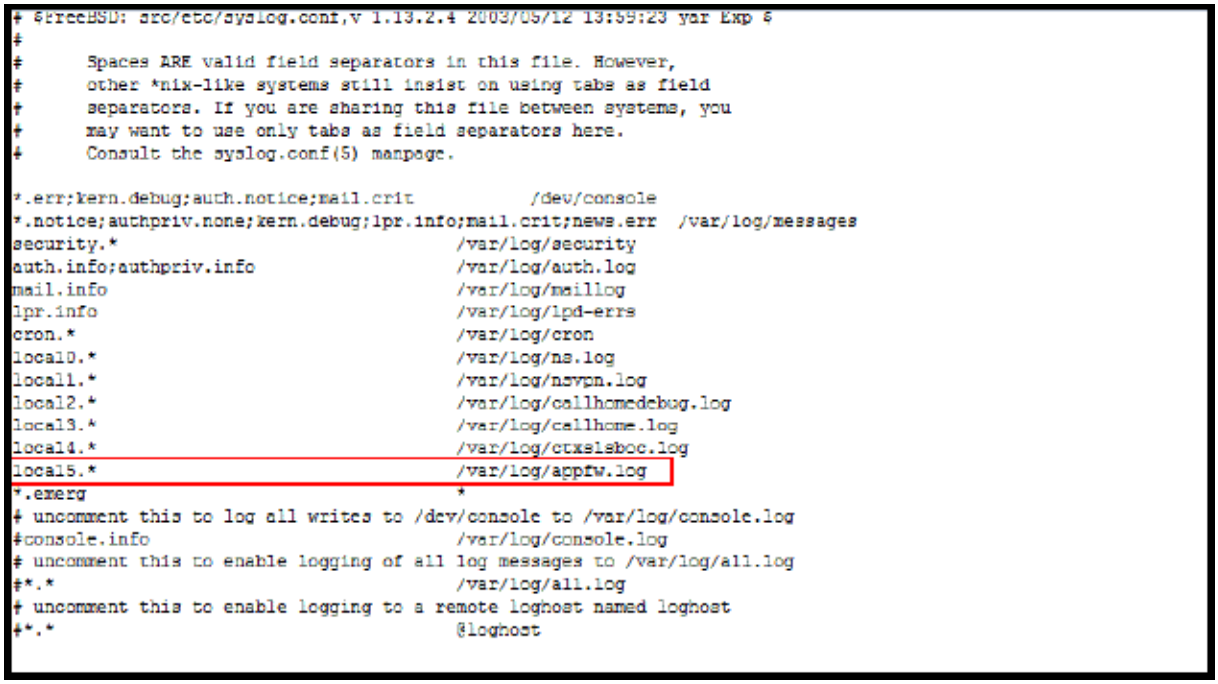
アプリケーションファイアウォールメッセージを別の **SYSLOG** サーバに送信する

アプリケーションファイアウォールメッセージを別の SYSLOG サーバに送信するには、次の手順を実行する必要があります。

- WinSCP などのセキュアなファイル転送ユーティリティ
- PuTTY などのアプライアンスに対して SSH コンソールを開くユーティリティ

アプリケーションファイアウォールメッセージを別の SYSLOG サーバに送信するには、次の手順が必要です。

1. WinSCP 経由で NetScaler アプライアンスにログオンします。
2. /etc/syslog.conf ファイルを更新し、このファイルに次の行を追加します。
   ```
   local5.* /var/log/appfw.log
   ```

```
# $FreeBSD: src/etc/syslog.conf,v 1.13.2.4 2003/05/12 13:59:23 yar Exp $
#
#       Spaces ARE valid field separators in this file. However,
#       other *nix-like systems still insist on using tabs as field
#       separators. If you are sharing this file between systems, you
#       may want to use only tabs as field separators here.
#       Consult the syslog.conf(5) manpage.

*.err;kern.debug;auth.notice;mail.crit           /dev/console
*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err  /var/log/messages
security.*                                      /var/log/security
auth.info;authpriv.info                         /var/log/auth.log
mail.info                                       /var/log/maillog
lpr.info                                        /var/log/lpd-errs
cron.*                                          /var/log/cron
local0.*                                        /var/log/ns.log
local1.*                                        /var/log/nsvpn.log
local2.*                                        /var/log/callhomedebug.log
local3.*                                        /var/log/callhome.log
local4.*                                        /var/log/ctxsleboc.log
local5.*                                        /var/log/appfw.log
*.emerg                                         *
# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                                   /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
#*.*                                            /var/log/all.log
# uncomment this to enable logging to a remote loghost named loghost
#*.*                                            @loghost
```

1. コマンドラインインターフェイスから次のコマンドを実行して、syslog PID を再起動します。

   ```
   kill -HUP <PID>
   ```

2. コマンドラインインターフェイスから次のコマンドを実行して、sysact1 などの syslog アクションを追加します。

   ```
   add audit syslogAction sysact1 127.0.0.1 -logLevel ALL -logFacility
    LOCAL5
   ```

3. sysact1 サーバを使用する syspol1 ポリシーを追加するには、次のコマンドを実行します。

   ```
   add audit syslogPolicy syspol1 ns_true sysact1
   ```

   または、高度な syslog ポリシーを追加します。

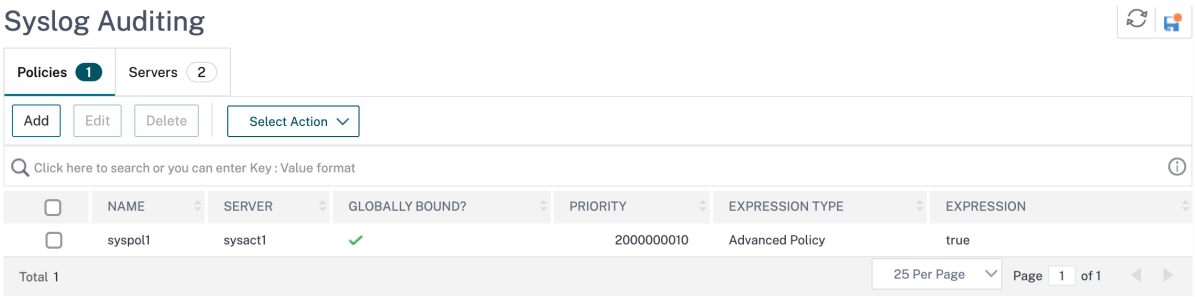
   ```
   add audit syslogPolicy syspol1 true sysact1
   ```

4. 次のコマンドを実行してアプリケーションファイアウォールポリシーをバインドし、ns.conf ファイルに保存されていることを確認します。

   ```
   bind appfw global syspol1 100
   ```

   または、以下のコマンドを実行して Advanced Syslog ポリシーをバインドします。

```
bind audit syslogGlobal -policyName syspol1 -priority 100 -globalBindType
 APPFW_GLOBAL
```

アプリケーションファイアウォールのセキュリティチェック違反はすべて/var/log/appfw.logにリダイレクトされ、ns.logには表示されなくなります。tail コマンドを実行して、/var/log/appfw.logで最新のエントリを表示できるようになりました。

Web App Firewall ログを表示する

syslog ビューアを使用するか、NetScaler アプライアンスにログオンして UNIX シェルを開き、任意の UNIX テキストエディタを使用してログを表示できます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて、**/var/log/** フォルダーの ns.logs を末尾に付けて、**Web App Firewall** のセキュリティチェック違反に関するログメッセージにアクセスします。

- Shell
- tail -f /var/log/ns.log

vi エディタ、または任意の Unix テキストエディタまたはテキスト検索ツールを使用して、特定のエントリのログを表示およびフィルタリングできます。たとえば、grep コマンドを使用して、クレジットカード違反に関するログメッセージにアクセスできます。

- tail -f /var/log/ns.log | grep SAFECOMMERCE

GUI を使用してログメッセージにアクセスするには

GUI には、ログメッセージを分析するための便利なツール (Syslog Viewer) が含まれています。Syslog ビューアにアクセスするには、複数のオプションがあります:

- プロファイルの特定のセキュリティチェックのログメッセージを表示するには、[**Web App Firewall**] > [プロファイル] に移動し、ターゲットプロファイルを選択し、[セキュリティチェック] をクリックします。ターゲットセキュリティチェックの行を強調表示し、[ログ] をクリックします。プロファイルの選択したセキュリティチェックから直接ログにアクセスすると、ログメッセージが除外され、選択したセキュリティチェックの違反に関連するログのみが表示されます。Syslog Viewer では、Web App Firewall ログをネイティブ形式および CEF 形式で表示できます。ただし、Syslog Viewer がターゲットプロファイル固有のログメッセージを除外するには、プロファイルからアクセスしたときに、ログが CEF ログ形式である必要があります。

- 「**NetScaler** 」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。[監査メッセージ] セクションで、[Syslog メッセージ] リンクをクリックして Syslog Viewer を表示します。Syslog Viewer には、すべてのプロファイルのすべての Web App Firewall セキュリティチェック違反ログを含む、すべてのログメッセージが表示されます。このログメッセージは、要求処理中に複数のセキュリティー検査違反がトリガされる可能性がある場合のデバッグに役立ちます。
- [**Web App Firewall**] > [ポリシー] > [監査] に移動します。[Audit Messages] セクションで、[Syslog messages] リンクをクリックして Syslog Viewer を表示します。このビューアには、すべてのプロファイルのすべてのセキュリティチェック違反ログを含む、すべてのログメッセージが表示されます。

HTML ベースの Syslog Viewer には、関心のあるログメッセージのみを選択するための次のフィルタオプションが用意されています。

- ファイル：現在の/var/log/ns.logファイルがデフォルトで選択され、対応するメッセージが Syslog Viewer に表示されます。/var/log ディレクトリにある他のログファイルの一覧が、圧縮された.gz 形式で表示されます。アーカイブされたログファイルをダウンロードして解凍するには、ドロップダウンリストのオプションからログファイルを選択します。選択したファイルに関連するログメッセージが syslog ビューアに表示されます。表示を更新するには、[Refresh] アイコン (2 つの矢印の円) をクリックします。
- [モジュール（**Module**）**]** リストボックス：ログを表示する NetScaler モジュールを選択できます。Web App Firewall ログの APPFW に設定できます。
- [**Event Type]** リストボックス：このボックスには、関心のあるイベントのタイプを選択するための一連のチェックボックスがあります。たとえば、署名違反に関するログメッセージを表示するには、[**APPFW_SIGNATURE_MATCH**] チェックボックスをオンにします。同様に、チェックボックスをオンにして、関心のある特定のセキュリティチェックを有効にすることができます。複数のオプションを選択できます。
- [**Severity**]：特定の重大度レベルを選択して、その重大度のログだけを表示できます。すべてのログを表示する場合は、チェックボックスをすべて空白のままにします。

  特定のセキュリティ検査の Web App Firewall セキュリティ検査違反ログメッセージにアクセスするには、[モジュール] のドロップダウンリストオプションで [**APPFW**] を選択してフィルタリングします。[イベントタイプ] には、選択をさらに絞り込むための豊富なオプションセットが表示されます。たとえば、APPFW_FIELDFORMAT チェックボックスをオンにして [適用] ボタンをクリックすると、フィールド形式のセキュリティチェック違反に関連するログメッセージのみが Syslog ビューアに表示されます。** 同様に、**[**APPFW_SQL**] と [**APPFW_STARTURL**] チェックボックスをオンにして [適用] ボタンをクリックすると、これら 2 つのセキュリティー検査違反に関するログメッセージのみが syslog ビューアに表示されます。

特定のログメッセージの行にカーソルを置くと、[**Module**]、[**EventType]**、**[**EventID]**、**[**** メッセージ] などの複数のオプションがログメッセージの下に表示されます。これらのオプションのいずれかを選択して、ログ内の対応する情報を強調表示できます。

ハイライト

- **CEF** ログ形式のサポート：CEF ログ形式オプションは、Web App Firewall ログメッセージを監視、解析、分析して攻撃を特定し、構成された設定を微調整して誤検出を減らし、統計情報を収集するための便利なオプションを提供します。
- ログメッセージをカスタマイズするオプション-高度な PI 式を使用してログメッセージをカスタマイズし、ログに表示するデータを含めることができます。
- **Web App Firewall** 固有のログを分離する-アプリケーションファイアウォール固有のログをフィルタリングし、別のログファイルにリダイレクトするオプションがあります。
- リモートロギング：ログメッセージをリモート syslog サーバにリダイレクトできます。
- 位置情報ロギング：Web App Firewall を構成して、要求を受信したエリアの地理的位置情報を含めることができます。組み込みの地理位置情報データベースを使用できますが、外部位置情報データベースを使用するオプションもあります。NetScaler アプライアンスは、IPv4 と IPv6 の両方の静的地理位置情報データベースをサポートしています。
- 情報リッチログメッセージ：設定に応じて、ログに含めることができる情報のタイプの例をいくつか示します。
  - Web App Firewall ポリシーがトリガーされました。
  - セキュリティチェック違反がトリガーされました。
  - リクエストは形式に誤りがあると考えられました。
  - リクエストまたはレスポンスがブロックされたか、ブロックされませんでした。
  - リクエストデータ（SQL またはクロスサイトスクリプティングの特殊文字など）またはレスポンスデータ（クレジットカード番号やセーフオブジェクト文字列など）が変換されました。
  - 応答のクレジットカードの数が、設定された制限を超えました。
  - クレジットカード番号とタイプ。
  - 署名ルールで設定されたログ文字列、および署名 ID。
  - リクエストのソースに関する地理位置情報。
  - 保護された機密フィールドに対するユーザ入力のマスク (X アウト)。

正規表現パターンを使用して機密データをマスクする

Web アプリケーションファイアウォール (WAF) プロファイルにバインドされたログ式のREGEX_REPLACE詳細ポリシー (PI) 関数を使用すると、WAF ログ内の機密データをマスクできます。オプションを使用して、正規表現パターンを使用してデータをマスクし、データをマスクする文字または文字列パターンを指定できます。また、PI 関数を設定して、正規表現パターンの最初のオカレンスまたはすべてのオカレンスを置き換えることができます。

デフォルトでは、GUI インターフェイスには次のマスクがあります。

- SSN
- クレジットカード
- パスワード
- ユーザー名

Web アプリケーションファイアウォールのログで機密データをマスクする

WAF プロファイルにバインドされたログ式でREGEX_REPLACE詳細ポリシー式を設定することで、WAF ログ内の機密データをマスクできます。
機密データをマスクするには、次の手順を完了する必要があります。

1. Web アプリケーションファイアウォールプロファイルを追加する
2. ログ式を WAF プロファイルにバインドする

Web アプリケーションファイアウォールプロファイルを追加する　コマンドプロンプトで入力します:

```
add appfw profile <name>
```

例:

```
Add appfw profile testprofile1
```

Web アプリケーションファイアウォールプロファイルでログ式をバインドする　コマンドプロンプトで入力します:

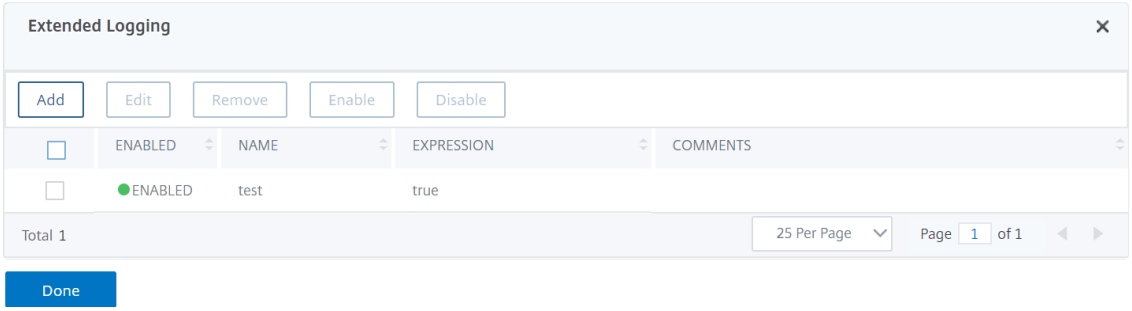
```
bind appfw profile <name> -logExpression <string> <expression> -
comment <string>
```

例:

```
bind appfw profile testProfile -logExpression "MaskSSN""HTTP.REQ.BODY
(10000).REGEX_REPLACE(re!\b\d{ 3 } -\d{ 2 } -\d{ 4 } \b!, “xxx” , ALL
)"-comment "SSN Masked"
```

NetScaler GUI を使用して **Web** アプリケーションファイアウォールログの機密データをマスクする

1. ナビゲーションペインで、[セキュリティ] > [**NetScaler Web App Firewall**] > [プロファイル] の順に展開します。
2. [プロファイル] ページで、[編集] をクリックします。
3. [**NetScaler Web App Firewall** プロファイル] ページで、[詳細設定] セクションに移動し、[拡張ログ] をクリックします。
4. [拡張ログ] セクションで、[追加] をクリックします。



5. [**NetScaler Web App Firewall** 拡張ログバインドの作成] ページで、次のパラメーターを設定します。

   a) 名前。ログ式の名前。
   b) Enabled。機密データをマスクするには、このオプションを選択します。
   c) ログマスク。マスクするデータを選択します。
   d) 式。WAF ログの機密データをマスクできるようにする高度なポリシー式を入力します。
   e) [コメント]。機密データのマスキングに関する簡単な説明。

6. **[作成]** して **[閉じる]** をクリックします。

Appendices

August 15, 2023

次の補足資料には、複雑な Web App Firewall タスクまたは周辺タスクに関する詳細情報が記載されています。

PCRE 文字エンコード形式

August 15, 2023

NetScaler オペレーティングシステムでは、印刷可能な **ASCII** 文字セット、つまり **16** 進コードが **HEX 20**（**ASCII 32**）から **HEX 7E**（**ASCII 127**）の間の文字のみの直接入力をサポートしています。Web App Firewall 構成にその範囲外のコードを持つ文字を含めるには、UTF-8 16 進コードを PCRE 正規表現として入力する必要があります。

Web App Firewall の構成に URL、フォームフィールド名、またはセーフオブジェクト表現として含める場合、多くの文字タイプを PCRE 正規表現を使用してエンコードする必要があります。それらには以下が含まれます。

- 上部 **ASCII** 文字。16 進数 7F (ASCII 128) から 16 進数 FF (ASCII 255) までのエンコーディングを持つ文字。使用される文字コードによっては、制御コード、アクセントまたはその他の変更を加えた ASCII 文字、非ラテンアルファベット文字、基本 ASCII セットに含まれていない記号がこれらのエンコーディングで参照される場合があります。これらの文字は、URL、フォームフィールド名、およびセーフオブジェクト式で使用できます。

- **2** バイト文字。8 バイトの単語を 2 つ使用するエンコーディングの文字。2 バイト文字は、主に中国語、日本語、韓国語のテキストを電子形式で表すために使用されます。これらの文字は、URL、フォームフィールド名、およびセーフオブジェクト式で使用できます。

  ASCII 制御文字。プリンタにコマンドを送信するときに使用される、印刷不能な文字です。16 進数コードが HEX 20 (ASCII 32) 未満の ASCII 文字はすべて、このカテゴリに分類されます。ただし、これらの文字は URL やフォームフィールド名には絶対に使用しないでください。セーフオブジェクト式にはほとんど使用されません。

NetScaler ADC アプライアンスは、UTF-8 文字セット全体をサポートするのではなく、次の 8 つの文字セットに含まれる文字のみをサポートします。

- 英語 **(**米国**) (ISO-8859-1)**。ラベルには「English US」と書かれていますが、Web App Firewall は Latin-1 文字セットとも呼ばれる ISO-8859-1 文字セットのすべての文字をサポートしています。この文字セットは、ほとんどの現代西ヨーロッパ言語を完全に表し、残りのいくつかの珍しい文字を除いてすべてを表します。
- 繁体字中国語 **(Big5)**。Web App Firewall は BIG5 文字セットのすべての文字をサポートします。これには、香港、マカオ、台湾、および中国本土以外に住む多くの中国民族遺産の人々によって話され、書き込まれる現代中国語で一般的に使用されているすべての繁体字中国語（表意文字）が含まれます。
- 簡体字中国語 **(GB2312)**。Web App Firewall は GB2312 文字セットのすべての文字をサポートします。これには、中国本土で話され、書き込まれる現代中国語で一般的に使用されるすべての簡体字中国語（表意文字）が含まれます。
- 日本語 **(SJIS)**。Web App Firewall は Shift-JIS (SJIS) 文字セットのすべての文字をサポートします。この文字セットには、現代日本語で一般的に使用されるほとんどの文字 (表意文字) が含まれます。
- 日本語 **(EUC-JP)**。Web App Firewall は EUC-JP 文字セットのすべての文字をサポートします。これには、現代日本語で一般的に使用されるすべての文字（表意文字）が含まれます。
- 韓国語 **(EUC-KR)**。Web App Firewall は EUC-KR 文字セットのすべての文字をサポートします。これには、現代韓国語で一般的に使用されるすべての文字（表意文字）が含まれます。
- トルコ語 **(ISO-8859-9)**。Web App Firewall は、ISO-8859-9 文字セットのすべての文字をサポートします。これには、現代トルコ語で使用されるすべての文字が含まれます。
- ユニコード **(UTF-8)**。Web App Firewall は、現代のロシア語で使用されている文字を含め、UTF-8 文字セットの文字をさらにサポートしています。

Web App Firewall を構成するときは、UTF-8 仕様でその文字に割り当てられた 16 進コードを使用して、ASCII 以外のすべての文字を PCRE 形式の正規表現として入力します。通常の ASCII 文字セット内の記号や文字（その文字セットに 1 桁の 2 桁のコードが割り当てられている）には、UTF-8 文字セットと同じコードが割り当てられます。たとえば、感嘆符 (!)ASCII 文字セットでは 16 進コード 21 が割り当てられ、UTF-8 文字セットでは 16 進数 21 にもなります。サポートされている別の文字セットのシンボルと文字には、UTF-8 文字セットの 16 進コードのペアセットが割り当てられます。たとえば、鋭アクセント（á）の付いた文字 a には、UTF-8 コード C3 A1 が割り当てられます。

Web App Firewall 構成でこれらの UTF-8 コードを表現するために使用する構文は、ASCII 文字の場合は「\ xNn」、英語、ロシア語、トルコ語で使用される非 ASCII 文字の場合は「\ xNn\ xNn\ xNn」、中国語、日本語、韓国語で使用される文字の場合は「\ xNn\ xNn\ xNn」です。たとえば、！を表現したい場合 Web App Firewall 正規表現の UTF-8 文字では、\ x21 と入力します。á を含める場合は、\ xC3\ xA1 と入力します。

> 注:
>
> 通常、ASCII 文字を UTF-8 形式で表現する必要はありませんが、これらの文字が Web ブラウザや基盤となるオペレーティングシステムを混乱させる可能性がある場合は、文字の UTF-8 表現を使用してこの混乱を避けることができます。たとえば、URL にスペースが含まれている場合、特定のブラウザや Web サーバーソフトウェアを混乱させないように、スペースを\ x20 としてエンコードしたほうがよい場合があります。

Web App Firewall 構成に含めるには、PCRE 形式の正規表現として入力する必要がある ASCII 以外の文字を含む URL、フォームフィールド名、およびセーフオブジェクト式の例を次に示します。各例は、実際の URL、フィールド名、または式文字列を最初に示し、その後に PCRE 形式の正規表現が続きます。

- 拡張 ASCII 文字を含む URL。

  実際の URL: `http://www.josénuñez.com`
  エンコードされた URL: `^http://www\[.\]jos\\xC3\\xA9nu\\xC3\\xB1ez\[.\]com$`

- 拡張 ASCII 文字を含む別の URL。

  実際の URL: `http://www.example.de/trömso.html`
  エンコードされた URL: `^http://www[.]example[.\]de/tr\xC3\xB6mso[.]html$`

  拡張 ASCII 文字を含むフォームフィールド名。

  実際の名前:
  名前エンコードされた名前:^nome_do_usu\ xC3\ xA1Rio$

- 拡張 ASCII 文字を含むセーフオブジェクト式。

  エンコードされていない表現 [A-Z] {3,6} ¥[1-9\][0-9]{6,6}
  エンコードされた表現:[A-Z] {3,6}\ xC2\ xA5 [1-9] [0-9] {6,6}

Unicode 文字セット全体と一致する UTF-8 エンコーディングを含むいくつかのテーブルがあります。これらの情報が掲載されている便利なウェブサイトを以下の表に示します。

`http://www.utf8-chartable.de/unicode-utf8-table.pl`

この Web サイトの表に記載されている文字を正しく表示するには、適切な Unicode フォントがコンピュータにインストールされている必要があります。そうしないと、キャラクターの視覚的な表示に誤りがある可能性があります。ただし、文字を表示するための適切なフォントがインストールされていない場合でも、この Web ページの説明と UTF-8 および UTF-16 コードは正しいです。

WAF 用のホワイトハット WASC 署名タイプ

August 15, 2023

NetScaler Web App Firewall は、Whitehat スキャナーが生成するすべての脆弱性タイプのブロッキングルールを受け入れて生成します。ただし、特定の脆弱性はウェブアプリファイアウォールに最も当てはまります。これらの脆弱性のリストを、WASC 1.0、WASC 2.0、またはベストプラクティスのシグネチャタイプのどれで対処するかによって分類してあります。

WASC 1.0 シグニチャタイプ

- HTTP リクエストスマグリング
- HTTP レスポンス分割
- HTTP レスポンススマグリング
- ヌルバイトインジェクション
- リモートファイルインクルージョン
- URL リダイレクターの不正使用

WASC 2.0 シグニチャタイプ

- 機能の乱用
- ブルートフォース
- コンテンツスプーフィング
- サービス拒否
- ディレクトリインデックス
- 情報漏えい
- アンチオートメーションが不十分
- 認証が不十分
- 権限が不十分
- セッションの有効期限が不十分
- LDAP インジェクション
- セッション固定

ベストプラクティス

- オートコンプリート属性
- クッキーアクセス制御が不十分
- パスワードの強度が不十分

- HTTP メソッドの使い方が無効です
- HTTP 以外のセッションクッキー
- パーシステントセッションクッキー
- 個人を特定できる情報
- セキュリティで保護された、キャッシュ可能な HTTP メッセージ
- セキュリティで保護されていないセッションクッキー

リクエスト処理のストリーミングサポート

August 15, 2023

NetScaler Web App Firewall はリクエストサイドストリーミングをサポートし、パフォーマンスを大幅に向上させます。アプライアンスは、要求をバッファリングする代わりに、SQL、クロスサイトスクリプティング、フィールドの一貫性、フィールド形式などのセキュリティ違反について着信トラフィックを検査します。アプライアンスがフィールドのデータの処理を完了すると、要求はバックエンドサーバに転送され、アプライアンスは他のフィールドを引き続き評価します。このデータ処理により、多くのフィールドを持つフォームを処理する際の処理時間が大幅に短縮されます。

20MB を超えるペイロードコンテンツのストリーミングを有効にすることをお勧めします。また、ストリーミングが有効な場合、バックエンドサーバーはチャンクされた要求を受け入れる必要があります。

> 注:
>
> Post Body Limit アクションは常にブロックに設定され、ストリーミングモードと非ストリーミングモードの両方に適用できます。受信トラフィックが 20MB を超える場合は、`PostBodyLimit` を予想値に構成することをお勧めします。

ストリーミングプロセスはユーザーには透過的ですが、次の変更により、設定を少し調整する必要があります。

RegEx パターンマッチ**:** RegEx パターンマッチは、連続する文字列マッチで 4K に制限されるようになりました。

フィールド名の一致**:** Web App Firewall 学習エンジンは、名前の最初の 128 バイトのみを区別できます。フォームに、最初の 128 バイトで同じ文字列が一致する名前のフィールドが複数ある場合、学習エンジンはそれらを区別しません。同様に、展開された緩和ルールによって、このようなすべてのフィールドが誤って緩和される可能性があります。

ホワイトスペースの除去、パーセントデコード、Unicode デコード、および文字セット変換は、セキュリティー検査のために正規化の間に行われます。128 バイトの制限は、UTF-8 文字フォーマットでのフィールド名の正規表現に適用されます。ASCII 文字の長さは 1 バイトですが、一部の国際言語では、文字の UTF-8 表現は 1 バイトから 4 バイトに及ぶ場合があります。名前の各文字が UTF-8 形式に変換するのに 4 バイトかかる場合、学習したルールでは名前の最初の 32 文字だけが区別されます。

フィールドの一貫性チェック**:** フィールドの一貫性を有効にすると、セッション内のすべてのフォームが Web App Firewall によって挿入された「as_fid」タグに基づいて、「action_url」を考慮せずに保存されます。

- フォームフィールドの一貫性のための必須フォームのタグ付け**:** フィールドの一貫性チェックが有効な場合、フォームタグも有効にする必要があります。フォームのタグ付けがオフになっていると、フィールドの一貫性保護が機能しないことがあります。
- セッションレスフォームフィールドの一貫性**:** セッションレスフィールドの一貫性パラメーターが有効になっている場合、Web App Firewall はフォームの「GET」から「POST」への変換を実行しなくなりました。フォームタグは、セッションレスフィールドの一貫性にも必要です。
- **as_fid** の改ざん**:** as_fid を改ざんした後にフォームが送信されると、改ざんされたフィールドがなくてもフィールドの一貫性違反が発生します。非ストリーミングリクエストでは、セッションに格納されている「action_url」を使用してフォームを検証できるため、これが許可されていました。

Signatures: シグニチャは次の仕様になりました。

- 場所**:** 各パターンに場所を指定することが必須要件になりました。**<Location>** ルール内のすべてのパターンにはタグを付ける必要があります。
- 高速一致**:** すべてのシグニチャルールに高速一致パターンが必要です。ファストマッチパターンがない場合は、可能ならばファーストマッチパターンを選択しようとします。高速一致はリテラル文字列ですが、使用可能なリテラル文字列が含まれている場合は、PCRE 高速一致に使用できます。
- 非推奨の場所**:** 次の場所はシグニチャルールでサポートされなくなりました。
  - HTTP_ANY
  - HTTP_RAW_COOKIE
  - HTTP_RAW_HEADER
  - HTTP_RAW_RESP_HEADER
  - HTTP_RAW_SET_COOKIE

クロスサイトスクリプティング**/SQL** 変換**:** 一重引用符 (‘)、バックスラッシュ ()、セミコロン (;)) などの SQL 特殊文字とクロスサイトスクリプティングタグは同じで、データの正規化を必要としないため、未加工データが変換に使用されます。HTML エンティティエンコーディング、パーセントエンコーディング、ASCII などの特殊文字の表現は、変換操作で評価されます。

Web App Firewall は、クロスサイトスクリプティング変換操作の属性名と値の両方を検査しなくなりました。ストリーミングが有効な場合、クロスサイトスクリプティング属性名のみが変換されるようになりました。

クロスサイトスクリプティングタグの処理： NetScaler 10.5.e ビルド以降のストリーミング変更の一環として、クロスサイトスクリプティングタグの処理が変更されました。以前のリリースでは、開き括弧 (<), or close bracket (>)、または開き括弧と閉じ括弧 (<>) の両方が存在する場合、クロスサイトスクリプティング違反としてフラグが立てられていました。10.5.e ビルド以降では動作が変更されました。開き括弧文字 (<), or only the close bracket character (>) のみが存在しても、攻撃とは見なされなくなりました。これは、開き括弧文字 (<) is followed by a close bracket character (>) で、クロスサイトスクリプティング攻撃のフラグが立てられる場合です。< followed by > クロスサイトスクリプティング違反をトリガするには、両方の文字が正しい順序 () で存在する必要があります。

> 注:
>
> **SQL** 違反ログの変更メッセージ： NetScaler ADC リリース 10.5.e 以降のストリーミング変更の一環として、入力データをブロック単位で処理するようになりました。RegEx パターンマッチングは、連続した文字列マッチングで 4K に制限されるようになりました。この変更により、SQL 違反ログメッセージには、以前のビルドとは異なる情報が含まれる場合があります。入力内のキーワードと特殊文字は数バイトで区切られます。アプライアンスは、入力値全体をバッファリングするのではなく、データを処理するときに SQL キーワードと特殊文字列を追跡します。ログメッセージには、フィールド名に加えて SQL キーワード、SQL 特殊文字、または SQL キーワードと SQL 特殊文字の両方が含まれます。次の例に示すように、残りの入力はログメッセージに含まれなくなります。
>
> 例**:**
>
> 10.5 では、Web App Firewall が SQL 違反を検出すると、入力文字列全体が次のログメッセージに含まれることがあります。
>
> フィールド **text=** “の **SQL** キーワードチェックに失敗しました **testbed1\;\\ (\;\\)**」から名前を選択してください。**\ ***<blocked>
>
> 11.0 では、フィールド名、キーワード、特殊文字 (該当する場合) のみを次のログメッセージに記録します。
>
> フィールド`text="select(;)"`<blocked>
> の SQL キーワードチェックに失敗しました
> この変更は、**application/x-www-form-urlencoded**、マルチパート/フォームデータ、または **text/x-gwt-rpc** コンテンツタイプを含むリクエストに適用されます。**JSON** または **XML** ペイロードの処理中に生成されるログメッセージは、この変更の影響を受けません。

RAW POST 本文**:** セキュリティチェック検査は常に RAW POST 本文に対して行われます。

フォーム **ID**: Web App Firewall が挿入した「as_fid」タグは、フォームの計算されたハッシュであり、ユーザーセッションでは一意ではなくなります。これは、ユーザーやセッションに関係なく、特定のフォームでは同じ値です。

Charset: リクエストに文字セットがない場合、アプリケーションプロファイルで指定されているデフォルトの文字セットがリクエストの処理時に使用されます。

カウンター**:**

プレフィックス「se」と「appfwreq」が付いたカウンターが追加され、ストリーミングエンジンとストリーミングエンジンのリクエストカウンターを追跡します。

```
nsconsmg -d statswt0 -g se_err_
nsconsmg -d statswt0 -g se_tot_
nsconsmg -d statswt0 -g se_cur_
nsconsmg -d statswt0 -g appfwreq_err_
nsconsmg -d statswt0 -g appfwreq_tot_
```


`nsconsmg -d statswt0 -g appfwreq_cur_`

`_err counters`: メモリ割り当ての問題またはその他のリソース不足が原因で、成功したが失敗したはずのまれなイベントを示します。

`_tot counters`: カウンターが増え続ける。

`_cur counters`: 現在のトランザクションの使用状況に基づいて変化し続ける現在の値を示すカウンター。

ヒント:

- Web App Firewall のセキュリティチェックは、以前と同様に機能する必要があります。
- セキュリティー検査の処理の順序は決まっていません。
- レスポンス側の処理は影響を受けず、変更されることはありません。
- クライアントレス VPN が使用されている場合、ストリーミングは実行されません。

> 重要:
>
> **Cookie** の長さの計算： NetScaler ADC リリース 11.0（65.x より前のビルド）に加えて、リリース 10.5.e では、Web App Firewall Cookie ヘッダー処理方法が変更されました。アプライアンスは Cookie を個別に評価し、Cookie ヘッダー内の Cookie の長さが設定された長さを超えると、バッファオーバーフロー違反がトリガーされました。その結果、NetScaler 10.5 以前のリリースでブロックされた要求が許可されることがあります。Cookie ヘッダー全体の長さは、Cookie の長さを決定するために計算されません。場合によっては、Cookie の合計サイズが許容値よりも大きくなり、サーバーが「400 Bad Request」で応答することがあります。
>
> 注**:**
> 変更は元に戻されました。NetScaler バージョン 10.5.e からバージョン 59.13xxe.e 以降のビルドでの動作は、リリース 10.5 の非拡張ビルドと同様です。Cookie の長さを計算するときに、生の Cookie ヘッダー全体が考慮されるようになりました。Cookie の長さの決定には、前後のスペースと名前と値のペアを区切るセミコロン (;) 文字も含まれます。

セキュリティログによる **HTML** リクエストの追跡

August 15, 2023

> 注:
>
> この機能は NetScaler リリース 10.5.e で使用できます。

トラブルシューティングには、クライアントのリクエストで受け取ったデータを分析する必要があり、困難な場合があります。特に、アプライアンスを通過するトラフィックが多い場合。問題を診断すると機能に影響が出たり、アプリケーションのセキュリティに迅速な対応が必要な場合があります。

NetScaler は Web App Firewall プロファイルのトラフィックを分離し、`nstrace` HTML リクエスト用に収集します。`nstrace` appfw モードで収集される情報には、リクエストの詳細とログメッセージが含まれます。トレースで「Follow TCP stream」を使用すると、ヘッダー、ペイロード、対応するログメッセージなど、個々のトランザクションの詳細を同じ画面に表示できます。

これにより、トラフィックに関する包括的な概要がわかります。リクエスト、ペイロード、および関連するログレコードを詳細に把握しておくと、セキュリティチェック違反の分析に役立ちます。違反を引き起こしているパターンを簡単に特定できます。パターンを許可する必要がある場合は、構成を変更するか、緩和ルールを追加するかを決定できます。

長所

1. 特定のプロファイルのトラフィックを分離: この機能拡張は、トラブルシューティングのために 1 つのプロファイルまたはプロファイルの特定のトランザクションのみのトラフィックを分離する場合に便利です。トレースで収集されたデータ全体をざっと調べたり、関心のあるリクエストを分離するために特別なフィルターを使用する必要はもうありません。トラフィックが多いと面倒です。お好みのデータを表示できます。
2. 特定のリクエストのデータ収集: 指定した期間のトレースを収集できます。必要に応じて特定のトランザクションを分離、分析、デバッグするために、2、3 のリクエストのみのトレースを収集できます。
3. リセットまたは中止を識別: 予期せず接続が閉じられたことを簡単に確認することはできません。—appfw モードで収集されたトレースは、Web App Firewall によってトリガーされたリセットまたは中止をキャプチャします。これにより、セキュリティチェック違反メッセージが表示されない場合でも、問題をすばやく切り分けることができます。形式に誤りのあるリクエストや、Web App Firewall によって終了されたその他の RFC 準拠以外のリクエストを識別しやすくなりました。
4. 復号化された **SSL** トラフィックを表示する: HTTPS トラフィックはプレーンテキストでキャプチャされるため、トラブルシューティングが容易になります。
5. 包括的な表示: リクエスト全体をパケットレベルで確認したり、ペイロードを確認したり、ログを確認してどのセキュリティチェック違反がトリガーされているかを確認したり、ペイロード内のマッチパターンを特定したりできます。ペイロードに予期しないデータ、ジャンク文字列、または印刷できない文字 (NULL 文字、\ r、\ n など) が含まれている場合は、トレースで簡単に見つけることができます。
6. 設定の変更: デバッグを行うと、観察された動作が正しい動作なのか、それとも構成を変更する必要があるのかを判断するのに役立つ情報が得られます。
7. 応答時間の短縮: 対象トラフィックのデバッグを高速化すると、応答時間が短縮され、NetScaler のエンジニアリングおよびサポートチームによる説明や根本原因分析が可能になります。

詳細については、「コマンドラインインターフェイスを使用した手動設定 」を参照してください。

コマンドラインインターフェイスを使用してプロファイルのデバッグトレースを構成するには

手順 1. ns トレースを有効にします。

show コマンドを使用して、設定した設定を確認できます。

- `set appfw profile <profile> -trace ON`

手順 2. トレースを収集します。nstrace コマンドに適用可能なすべてのオプションを引き続き使用できます。

- start nstrace -mode APPFW

手順 3. トレースを停止します。

- stop nstrace

トレースの場所:nstraceはタイムスタンプ付きのフォルダに保存されます。このフォルダは /var/nstrace ディレクトリに作成され、wiresharkを使用して表示できます。/var/log/ns.log を末尾に移動すると、新しいトレースの場所に関する詳細が記載されたログメッセージが表示されます。

ヒント:

- appfw モードオプションを使用すると、nstrace は「nstrace」が有効になっている 1 つ以上のプロファイルのデータのみを収集します。
- プロファイルでトレースを有効にしても、「start ns trace」コマンドを明示的に実行してトレースを収集するまで、トレースの収集は自動的に開始されません。
- プロファイルでトレースを有効にしても Web App Firewall のパフォーマンスに悪影響はないかもしれませんが、データを収集する期間のみこの機能を有効にしたい場合があります。トレースを収集したら、—trace フラグをオフにすることをお勧めします。このオプションは、過去にこのフラグを有効にしていたプロファイルからデータを誤って取得するリスクを防ぎます。
- nstraceに含まれるトランザクションレコードのセキュリティチェックを行うには、ブロックアクションまたはログアクションを有効にする必要があります。
- プロファイルのトレースが「オン」の場合、リセットと中止はセキュリティチェックアクションとは別にログに記録されます。
- この機能は、クライアントから受け取ったリクエストのトラブルシューティングにのみ適用されます。—appfw モードのトレースには、サーバーから受信した応答は含まれません。
- nstrace コマンドに適用可能なすべてのオプションを引き続き使用できます。たとえば、次のようにします。

  ```
  start nstrace -tcpdump enabled -size 0 -mode appFW
  ```

- nstrace リクエストが複数の違反をトリガーした場合、そのレコードには対応するすべてのログメッセージが含まれます。
- この機能では、CEF ログメッセージ形式がサポートされています。
- リクエスト側チェックのブロックまたはログアクションをトリガーする署名違反もトレースに含まれます。
- HTML (非 XML) 要求のみがトレースに収集されます。

Web App Firewall によるクラスター構成のサポート

August 15, 2023

> 注:
>
> NetScaler 11.0 バージョンでは、ストライピング構成と部分ストライプ構成用の NetScaler Web App Firewall が導入されました。

クラスターは、単一のシステムとして構成および管理される NetScaler アプライアンスのグループです。クラスタ内の各アプライアンスはノードと呼ばれます。構成がアクティブなノードの数に応じて、クラスター構成はストライピング構成、部分ストライピング構成、またはスポット構成と呼ばれます。Web App Firewall はすべての構成で完全にサポートされています。

クラスタ構成でのストライプ型および部分的ストライプ型の仮想サーバーのサポートには、主に次の 2 つの利点があります。

1. セッションフェイルオーバーのサポート：ストライピングされた仮想サーバー構成と部分的にストライピングされた仮想サーバー構成は、セッションフェイルオーバーをサポートします。開始 URL のクローズやフォームフィールドの一貫性などの高度な Web App Firewall セキュリティ機能により、トランザクション処理中のセッションのチェック、維持、使用が可能になります。高可用性構成またはスポットクラスタ構成では、Web App Firewall トラフィックを処理しているノードに障害が発生すると、すべてのセッション情報が失われ、ユーザーはセッションを再確立する必要があります。ストライピングされた仮想サーバー構成では、ユーザーセッションは複数のノードに複製されます。ノードがダウンすると、レプリカを実行しているノードが所有者になります。セッション情報は、ユーザーに目に見える影響を与えることなく維持されます。
2. スケーラビリティ—クラスタ内のどのノードでもトラフィックを処理できます。クラスターの複数のノードが、ストライピングされた仮想サーバーによって処理される受信要求を処理できます。これにより、Web App Firewall が複数の同時リクエストを処理する能力が向上し、全体的なパフォーマンスが向上します。

セキュリティチェックと署名保護は、クラスター固有の Web App Firewall 構成を追加しなくても導入できます。構成コーディネーター (CCO) ノードで通常の Web App Firewall 構成を実行して、すべてのノードに伝播できます。

> 注:
>
> セッション情報は複数のノードに複製されますが、ストライプ構成のすべてのノードに複製されるわけではありません。したがって、フェイルオーバー・サポートでは、同時に発生する障害の数には限りがあります。複数のノードで同時に障害が発生した場合、セッションが別のノードに複製される前に障害が発生すると、Web App Firewall はセッション情報を失う可能性があります。

ハイライト

- Web App Firewall は、クラスター展開におけるスケーラビリティ、高スループット、およびセッションフェイルオーバーのサポートを提供します。

- Web App Firewall のすべてのセキュリティチェックと署名保護は、すべてのクラスター構成でサポートされています。
- クラスターではまだキャラクターマップはサポートされていません。学習エンジンは、フィールド形式のセキュリティチェックの学習ルールにフィールドタイプを推奨します。
- 統計情報と学習したルールは、クラスター内のすべてのノードから集約されます。
- 分散ハッシュテーブル (DHT) は、セッションをキャッシュし、複数のノードにセッション情報を複製する機能を提供します。仮想サーバーに要求が送信されると、NetScaler ADC アプライアンスは DHT に Web App Firewall セッションを作成し、DHT からセッション情報を取得することもできます。
- クラスタリングは Advanced および Premium ライセンスでライセンスされます。この機能は、Standard ライセンスでは使用できません。

デバッグとトラブルシューティング

August 15, 2023

Web App Firewall の各機能に関連する次のトラブルシューティングおよびデバッグ情報を参照してください。

- アプリケーションファイアウォール-高 CPU
- Memory
- 大きなファイルのアップロードの失敗
- Learning
- Signatures
- トレースログ
- Miscellaneous

高 **CPU**

August 15, 2023

以下は、Web App Firewall を使用する際に発生する機能と CPU 使用率に関するデバッグの問題と、従うべきベストプラクティスです。

ポリシーヒット、バインディング、ネットワーク構成、**Web App Firewall** 構成を確認してください。

- 設定ミスを特定
- 影響を受けるトラフィックを処理している仮想サーバーを特定

次のログファイルのログを調べて、セキュリティ違反や最近の構成変更がないか確認します。

- /var/log/ns.log
- /var/nslog/**import**.log
- /var/nslog/aslearn.log
- tail -f /var/log/ns.log | grep APPFW_SIGNATURE_MATCH

例:

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW| APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method=GET request= http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=Signature violation rule ID 807: web-cgi /wwwboard/passwd.txt access cn1=140 cn2=841 cs1=pr_ffc cs2=PPE0 cs3=OyTgjbXBqcpBFeENKDlde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=not blocked
<!--NeedCopy-->
```

影響を受けるトラフィックを分離してください。

- プロファイルを分離
- セキュリティチェックを分離
- URL、仮想サーバー、トラフィックパラメータを分離

条件付きプロファイルレベルトレースは、トラフィックと違反レコードの識別に役立ちます。

- set appfw profile <profile> -trace ON
- start nstrace -mode APPFW -size 0
- stop nstrace

注: トレースは-size 0 オプションで収集されていることを確認してください。

appfw、**dht**、**IP** レピュテーションアクティビティカウンタを確認してください。

- nsconmsg -g as_ -g appfwreq_ -g iprep -d current

接続中のリセット用のモニターウィンドウサイズ**:**

無効な http メッセージが原因で NetScaler が接続をリセットすると、Appfw はウィンドウサイズを 9845 に設定します。

例:

- 間違った形式のリクエストを受け取りました-接続がリセットされました
- CPU 使用率の高さに関する問題
- システム制限についてはデータシートを確認してください
- CPU 使用率、appfw、DHT、およびメモリ関連のアクティビティを検査します。appfw セッションの監視
- nsconmsg -g cc_cpu_use -g appfwreq -g as -g dht -g mem_AS_OBJ -g mem_AS_COMPONENT -d current

対象期間中に **Web App Firewall** のコンポーネントやオブジェクトに割り当てられたり解放されたりしたメモリを監視します。これにより、CPU 使用率が高くなる保護を分離できます。

- プロファイラー出力
- ログを観察する

CPU 使用率が高くなる原因となる **appfw** チェックを分離**:**

- startURLClosure
- フォームファイルの一貫性
- CSRF
- クッキー保護
- リファラーヘッダーチェック

シグネチャの自動更新が **CPU** 使用率の高さにつながっていないことを確認します **(**確認のため無効化**)**。

メモリ

August 15, 2023

Web App Firewall の使用メモリ関連の問題が発生した場合に従うべきベストプラクティスをいくつか次に示します。

nsconmsg コマンドの使用方法**:**

- 次のコマンドを実行して、グローバルメモリ統計を調べて、システムに十分なメモリがあり、メモリ割り当てに失敗していないことを確認します。

  ```
  * *- nsconmsg -d memstats
  ```

- 次のコマンドを実行して、appsecure、IP レピュテーション、キャッシュ、および圧縮の現在の割り当てメモリ制限と最大メモリ制限を確認してください。

  ```
  nsconmsg -d memstats | egrep -i APPSECURE|IPREP|CACHE|CMP
  ```

- 次のコマンドを実行して、appfw、DHT、IP レピュテーションアクティビティカウンタを確認します。

  ```
  nsconmsg -g as -g appfwreq_ -g iprep -d current
  ```

- 次のコマンドを実行して、すべての Web App Firewall エラーカウンタを確認します。

  ```
  nsconmsg -g as_ -g appfwreq_ -g iprep_ -d stats | grep err
  ```

- 以下のコマンドを実行して、すべてのシステムエラーカウンタを確認します。

  ```
  nsconmsg -g err -d current
  ```


- 次のコマンドを実行して、CPU、APPFWREQ、AS、および DHT のカウンタを確認します。

  ```
  nsconmsg -g cc_cpu_use -g appfwreq -g as -g dht -d current
  ```

- 次のコマンドを実行して、設定されているキャッシュメモリを確認します。
- `show cacheparameter`
- 次のコマンドを実行して、設定したメモリを確認します。

  ```
  nsconmsg -d memstats | egrep -i CACHE
  ```

- Web App Firewall のコンポーネントとオブジェクトのメモリ分布を特定します。

AS_OBJ_ メモリを表示:

```
nsconmsg -K newnslog -d stats | grep AS_OBJ | egrep -v AppFW_cpu0|
total | sort -k3
```

AS_COMPONENT_ メモリを表示:

```
nsconmsg -K newnslog -d stats | grep AS_COMPONENT | egrep -v AppFW_cpu0
|total | sort -k3
```

次のコマンドを実行して、アクティブなセッションの数を確認します。

アクティブセッション数の監視/プロット:

```
nsconmsg -g as_alive_sessions -d current
```

割り当てられたセッション、空いているセッション、更新されたセッションの合計を監視/プロット:

- `nsconmsg -g as_tot_alloc_sessions -g as_tot_free_sessions -d current`
- `nsconmsg -g as_tot_update_sessions -d current`

必要に応じて、次のコマンドを実行してセッションタイムアウトを減らし、セッション制限が使用されないようにします。

```
set appfwsettings -sessionTimeout <300>
```

必要に応じて、次のコマンドを実行してセッションの最大有効期間を設定します。

```
set appfwsettings -sessionLifetime <7200>
```

割り当て済みメモリと使用済みメモリの確認

割り当てられたメモリと使用中のメモリの合計を確認するには:

- **nsconmsg** –**d** memstats コマンドを使用してください。**MEM_APPSECURE** フィールドを確認してください。

- **stat appfw** コマンドを使用して、メモリ消費量情報を取得します。

Web App Firewall は、特定の期間またはサイズを過ぎてもログを自動的に削除しません。

- All AppFw logs are archived in the */var/log/ns.log* ファイル。*ns.log* ファイルはロールオーバータスクを実行します。

詳細については、次のリンクを参照してください。<http://support.citrix.com/article/CTX121898>

Web App Firewall メモリの増加:

- Web App Firewall のメモリを増やすための CLI オプションはありません。Web App Firewall のメモリはプラットフォームによって異なります。
- *nsapimgr* オプションを使用してメモリを増やすこともできますが、お勧めしません。

Web App Firewall の最大許容メモリはプラットフォームによって決まり、IC を無効にしてもメモリ割り当てには影響しません。

大容量ファイルのアップロード失敗

December 8, 2023

大容量ファイルのアップロードに失敗した場合は、次の点を確認してください:

- アプリケーションファイアウォールのポストボディ制限の設定が間違っています。
- ファイルアップロードスキャンを有効にすると、処理時間が長くなります。
- システムの限界に達しています。

19 MB を超えるペイロードでは、アプリケーションファイアウォールプロファイルでストリーミングを有効にすることをお勧めします。ストリーミングを有効にする前に、バックエンドサーバーがチャンクリクエストをサポートしていることを確認してください。

学習

August 15, 2023

学習機能の問題が発生した場合に推奨されるベストプラクティスをいくつか次に示します。

学習プロセス:

- *aslearn* プロセスが実行中であることを確認します。

- トップコマンドの出力をチェック
- 次のコマンドを実行して、ps コマンドの出力を確認します。

  ```
  ps -ax | grep aslearn | grep -v "grep"
  ```

  例:

  ```
  1  root@ns# ps -ax | grep aslearn | grep -v "grep"
  2  1439  ??  Ss     0:03.86 /netscaler/aslearn -start -f /netscaler/
        aslearn.conf
  3  <!--NeedCopy-->
  ```

- *ns.log* ファイルを確認して、問題が発生する前に実行された最近の構成コマンドを特定します。

  ```
  /var/log/ns.log
  ```

- aslearn ログを調べて、aslearn メッセージを確認します。

  ```
  /var/log/aslearn.log
  ```

- 影響を受けるプロファイルとセキュリティチェックを分離する
- 次のコマンドを実行して、失敗している GUI と CLI コマンドを特定します。

  ```
  show appfw learningdata <profileName> <securityCheck>
  ```

  例:

  - `show learningdata test_profile starturl`
  - `show learningdata test_profile crosssiteScripting`
  - `show learningdata test_profile sqLInjection`
  - `show learningdata test_profile csRFtag`
  - `show learningdata test_profile fieldformat`
  - `show learningdata test_profile fieldconsistency`

- bsd シェルプロンプトから sqlite の整合性チェックを実行します。

  ```
  nsshell # sqlite3 /var/nslog/asl/<profile_name_in_lowercase>.db '
  pragma integrity_check;
  ```

  例:

  ```
  1  root@ns# sqlite3 /var/nslog/asl/tsk0247284.db 'pragma
        integrity_check;'
  2  ok
  3  <!--NeedCopy-->
  ```

- ルールをデプロイまたは削除して、学習を再開してください。
  - (保護ごとに) 学習項目が 2000 個に達すると、その保護の学習を開始できなくなります。
  - データベースのサイズが 20 MB に達した場合は、すべての保護機能の学習を停止してください

- 学習プロセスとして再起動

```
*/netscaler/aslearn -start -f/netscaler/aslearn.conf*
```

- 次のコマンドを実行して、/var フォルダー内のスペースを確認します。

```
du -h /var
```

- 以下のコマンドを実行して、学習閾値制限を確認します。

```
show appfwlearningsettings <profile_name> <securityCheck>
```

- 以下のコマンドを実行して学習したデータを収集します。

```
export appfwlearningdata <profile_name> <securityCheck>
```

- 学習したデータがコレクタにアップロードされていることを認識します。

署名

August 15, 2023

シグネチャ入門

署名を追加するには:

1. 「デフォルト-署名」を選択し、「追加」をクリックしてコピーを作成します。
2. わかりやすい名前を付けてください。新しい sig オブジェクトは、ユーザー定義オブジェクトとして追加されます。
3. 特定のニーズに関連するターゲットルールを有効にします。
   - ルールはデフォルトでは無効になっています。
   - ルールが増えると処理も増える
4. アクションを設定します。

   ブロックアクションとログアクションはデフォルトで有効になっています。統計は別の選択肢です
5. プロフィールで使用する署名を設定します。

署名を使用する際のヒント

- アプリケーションの保護に該当するシグネチャのみを有効にすることで、処理のオーバーヘッドを最適化します。

- シグネチャマッチをトリガーするには、ルール内のすべてのパターンが一致する必要があります。
- 独自にカスタマイズしたルールを追加して受信リクエストを検査し、SQL インジェクションやクロスサイトスクリプティング攻撃など、さまざまな種類の攻撃を検出できます。
- また、回答を検査するルールを追加して、クレジットカード番号などの機密情報の漏洩を検出し、ブロックすることもできます。
- 複数のセキュリティチェック条件を追加して、独自のカスタマイズチェックを作成します。

署名を使用する際のベストプラクティス

署名に関連する問題が発生したときに従うことができるベストプラクティスを次に示します。

- インポートコマンドがプライマリとセカンダリで成功したことを確認します。
- CLI と GUI の出力が一致していることを確認します。
- ns.log をチェックして、署名のインポートと自動更新中にエラーがないかどうかを確認してください。
- DNS ネームサーバーが正しく設定されているか確認してください。
- スキーマのバージョン非互換性をチェックしてください。
- デバイスが AWS でホストされている署名更新 URL にアクセスして自動更新できないかどうかを確認します。
- デフォルト署名とユーザーが追加した署名のバージョンの不一致を確認します。
- プライマリノードとセカンダリノードのシグネチャオブジェクト間のバージョンの不一致を確認します。
- 高い CPU 使用率を監視します（シグニチャ更新の問題を除外するには、自動更新を無効にします）。

トレースログ

August 15, 2023

トレースログを記録するには:

1. プロファイルのトレースを有効にします。show コマンドを使用して、設定した設定を確認できます。

```
set appfw profile <profile> -trace ON
```

1. トレースの収集を開始します。nstrace コマンドに適用できるすべてのオプションを引き続き使用できます。

```
start nstrace -mode APPFW
```

1. トレースの収集を停止する

stop nstrace

トレースの場所:nstrace は */var/nstrace* ディレクトリに作成されたタイムスタンプ付きのフォルダに保存され、wireshark を使用して表示できます。*/var/log/ns.log* を末尾に付けて、新しいトレースの場所に関する詳細を提供するログメッセージを表示できます。

トレースログの利点:

- 特定のプロファイルのトラフィックを分離
- 特定の要求に応じてデータを収集
- リセットまたは中止の識別
- 復号化された SSL トラフィックを表示:HTTPS トラフィックはプレーンテキストでキャプチャされるため、トラブルシューティングが容易になります。
- 包括的なビューを提供します。リクエスト全体をパケットレベルで確認し、ペイロードをチェックし、ログを表示してどのセキュリティチェック違反がトリガーされているかを確認し、ペイロード内のマッチパターンを特定できます。ペイロードに予期しないデータ、ジャンクストリング、または印刷できない文字（ヌル文字、\ r、\ n など）が含まれている場合は、トレースで簡単に見つけることができます。
- 応答時間の短縮：ターゲットトラフィックのデバッグを高速化し、根本原因分析を行います。

その他

August 15, 2023

Web App Firewall を使用する際に発生する可能性のあるいくつかの問題の解決策を以下に示します。

- Web App Firewall は、無効な http メッセージの接続をリセットするときに、ウィンドウサイズを 9845 に設定します。

  —不正な形式のリクエストを受信しました-接続リセット [クライアント/サーバが無効なコンテンツ長ヘッダーを送信しました]
  —リクエストヘッダーのコンテンツタイプが不明です

- システム制限：アプリケーションがフリーズしているように見える

  —最大セッション制限に達したときに発生します。(100K)
  —操作に必要なシステムメモリが 少なくなります。

  - IP レピュテーション機能が動作しない
    —レピュテーション機能を有効にしてから iprep プロセスが開始するまでに約 5 分かかります。IP レピュテーション機能はその期間動作しない場合があります。

- 予期しない Web App Firewall 違反がトリガーされました

—セッションタイムアウトのデフォルト値は 900 秒です。セッションタイムアウトが小さい値に設定されている場合、ブラウザはセッション化に依存するチェック（CSRF、FFC など）で誤検出をトリガーする可能性があります。セッションタイムアウトを確認し、セッション ID （CEF ログの cs3）を確認します。SessionID が異なる場合は、セッションタイムアウトが原因である可能性があります。
—フォームが JavaScript によって動的に生成される場合、誤った FFC 違反が発生する可能性があります。

- FFC 違反ログのフィールド名が空です (11.0 リリース以前)

  これは、セッションのフォームにないフォームフィールドに出くわすシナリオで見られることがあります。

  これが発生する可能性のあるシナリオ:

  —フォームがクライアントに送信されてから受信された時点までにセッションがタイムアウトした。
  —フォームは、Java スクリプトを使用してクライアント側で生成されました。

参照ドキュメント

August 15, 2023

Web App Firewall の機能については、次のリソースを参照してください。

- NetScaler Web App Firewall がアプリケーションデータトラフィックをどのように変更するか。
- NetScaler ADC アプライアンスでの Web App Firewall セキュリティ違反による HTML リクエストのトレースとログの仕組み
- トップレベルの保護
- セキュリティ緩和
- アプリケーションの構成とデプロイに関する情報:
  - Application
  - Firewall
  - Logs
- シグネチャ更新記事
- ボット管理

署名アラート記事

August 15, 2023

NetScaler Web App Firewall（WAF）は、ダウンロードしてアプライアンスに適用できる署名の更新を通知します。セキュリティ攻撃を検出すると、新しいシグニチャの更新に関するメール通知が届きます。署名をダウンロードして、アプライアンスに適用できます。

署名アラート通知の受信方法

この記事では、RSS フィードを購読して新しいシグニチャ更新の通知を受け取る方法について説明します。購読すると、新しい署名がダウンロード可能になるたびに、定期的な RSS フィードが届きます。

> 注:
>
> - Web App Firewall シグニチャに関するアップデートを入手するには、シグニチャの自動更新機能を構成する必要があります。詳細については、「署名の自動更新」トピックを参照してください。
> - 新しいボット署名に関する更新を取得するには、ボット署名の自動更新機能を設定する必要があります。詳細については、「ボット署名の自動更新」トピックを参照してください。

新しいシグニチャ更新用の **RSS** フィードを購読するには、以下の手順に従ってください。

1. Web ブラウザーで「署名アラート記事のドキュメント履歴」トピックを開きます。
2. ページの右上にある RSS ボタンをクリックし、RSS フィードの URLをコピーします。
3. コピーした RSS フィード URL を目的の RSS フィードリーダーに追加します。

署名更新バージョン **127**

April 15, 2024

2024-04-04 週に特定された脆弱性について、新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャーバージョン 127 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998512	CVE-2024-25153	5.1.6 より前の WEB-MISC Fortra FileCatalyst ワークフロー-任意のファイルアップロードの脆弱性 (CVE-2024-25153)
998513	CVE-2024-24401	2024R1.01 までの WEB-MISC NagiosXi-SQL インジェクションの脆弱性 (CVE-2024-24401)
998514	CVE-2024-21726	WEB-MISC Joomla! 5.0.3 および 4.4.3 より前の-ForceLanguage パラメーターによる XSS の脆弱性 (CVE-2024-21726)
998515	CVE-2024-21726	WEB-MISC Joomla! 5.0.3 および 4.4.3 より前の-ForcedItemType パラメーターによる XSS の脆弱性 (CVE-2024-21726)
998516	CVE-2024-2055	4.50 より前の WEB-MISC Artica プロキシ-認証バイパスの脆弱性 (CVE-2024-2055)
998517	CVE-2024-2054	4.50 より前の WEB-MISC Artica プロキシ-認証されていない PHP オブジェクトの逆シリアル化の脆弱性 (CVE-2024-2054)
998518	CVE-2024-2053	4.50 より前の WEB-MISC Artica プロキシ-認証されていない任意のファイルインクルージョンの脆弱性 (CVE-2024-2053)
998519	CVE-2023-6063	1.2.2 より前の WEB-WORDPRESS WP 最速キャッシュプラグイン-認証されていない SQL インジェクションの脆弱性 (CVE-2023-6063)
998520	CVE-2023-3550	1.41 より前の WEB-Misc MediaWiki-保存型 XSS の脆弱性 (CVE-2023-3550)

署名更新バージョン **126**

April 15, 2024

2024-03-15 週に特定された脆弱性について、新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャーバージョン 126 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998521	CVE-2024-27198	WEB-MISC 2023.11.4 より前の JetBrains TeamCity-認証バイパスの脆弱性 (CVE-2024-27198)
998522	CVE-2024-25065	WEB-MISC Apache Ofbiz 18.12.12 より前の-認証バイパスの脆弱性 (CVE-2024-25065)
998523	CVE-2024-20738	WEB-MISC Adobe FrameMaker 2022 アップデート 2 より前のパブリッシングサーバー-/doxserver/ による認証バイパス (CVE-2024-20738)
998524	CVE-2024-20738	WEB-MISC Adobe FrameMaker 2022 アップデート 2 より前のパブリッシングサーバー-/server/ による認証バイパス (CVE-2024-20738)

署名ルール	**CVE ID**	説明
998525	CVE-2024-1708	23.9.8 より前の WEB-MISC ConnectWise ScreenConnect-ジップスリップ攻撃によるパストラバーサルの脆弱性（CVE-2024-1708）
998526	CVE-2024-1071	2.1.3 から 2.8.2 までのウェブワードプレスアルティメットメンバー-SQL インジェクションの脆弱性 (CVE-2024-1071)
998527	CVE-2023-5204	4.8.9 までのウェブワードプレス AI チャットボット-SQL インジェクションの脆弱性 (CVE-2023-5204)
998528	CVE-2023-44313	2.2.0 より前の WEB-MISC Apache ServiceComb サービスセンター-サーバー側リクエストフォージェリの脆弱性（CVE-2023-44313）
998529	CVE-2023-41474	WEB-MISC Ivanti Avalanche 最大 6.3.4.153-パストラバーサルの脆弱性 (CVE-2023-41474)
998530	CVE-2023-41474	WEB-MISC Ivanti Avalanche 最大 6.3.4.153-パストラバーサルの脆弱性 (CVE-2023-41474)
998531	CVE-2023-40597	8.2.12 9.0.6 および 9.1.1 より前の WEB-MISC Splunk Enterprise-絶対パストラバーサルの脆弱性（CVE-2023-40597）

署名更新バージョン **125**

March 20, 2024

2024-02-26 週に特定された脆弱性について、新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャーバージョン 125 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998532	CVE-2024-24830	0.8.0 より前の WEB-MISC OpenObserve-権限昇格 (CVE-2024-24830)
998533	CVE-2024-25106	0.8.0 より前の WEB-MISC OpenObserve-不正アクセスの脆弱性 (CVE-2024-25106)
998534	CVE-2024-24747	リリース前の WEB-MISC MiniO。2024-01-31T20-20-33Z-不適切な権限管理の脆弱性 (CVE-2024-24747)
998535	CVE-2024-1709	23.9.8 より前の WEB-MISC ConnectWise ScreenConnect-認証バイパスの脆弱性 (CVE-2024-1709)
998536	CVE-2024-1207	9.9.1 より前の WEB-WORDPRESS WP 予約カレンダープラグイン-認証されていない SQL インジェクションの脆弱性（CVE-2024-1207）
998537	CVE-2024-0221	1.8.19 までのウェブワードプレスフォトギャラリー-ディレクトリトラバーサルの脆弱性 (CVE-2024-0221)
998538	CVE-2024-0221	1.8.19 までのウェブワードプレスフォトギャラリー-ディレクトリトラバーサルの脆弱性 (CVE-2024-0221)

署名ルール	**CVE ID**	説明
998539	CVE-2023-46266	WEB-MISC 6.4.2 より前の Ivanti Avalanche-認証バイパスの脆弱性 (CVE-2023-46266)
998540	CVE-2023-46264	6.4.2 より前の WEB-MISC Ivanti Avalanche-任意のファイルアップロードの脆弱性 (CVE-2023-46264)
998541	CVE-2023-46263	6.4.2 より前の WEB-MISC Ivanti Avalanche-任意のファイルアップロードの脆弱性 (CVE-2023-46263)
998542	CVE-2023-46214	9.0.7 および 9.1.2 より前の WEB-MISC Splunk Enterprise-安全ではない XML 解析の脆弱性 (CVE-2023-46214)
998543	CVE-2021-22962	WEB-MISC 6.4.2 より前の Ivanti Avalanche-認証バイパスの脆弱性 (CVE-2021-22962)

署名更新バージョン **124**

March 20, 2024

2024-02-14 週に特定された脆弱性について、新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャーバージョン 124 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998544	CVE-2024-21893	WEB-MISC Ivanti Connect セキュア複数バージョン-サーバーサイドリクエストフォージェリの脆弱性 (CVE-2024-21893)
998545	CVE-2024-21893	WEB-MISC Ivanti Connect セキュア複数バージョン-サーバーサイドリクエストフォージェリの脆弱性 (CVE-2024-21893)
998546	CVE-2023-7028	WEB-MISC GitLab CE/EE 複数バージョン-不適切なアクセスコントロールの脆弱性（CVE-2023-7028）
998547	CVE-2023-50721	WEB-MISC XWiki プラットフォームの複数のバージョン-コードインジェクションの脆弱性 (CVE-2023-50721)

署名更新バージョン **123**

February 15, 2024

2024-01-25 週に特定された脆弱性について、新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 123 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998548	CVE-2024-21650	WEB-MISC XWiki プラットフォーム以前の複数のバージョン-リモートコード実行の脆弱性 (CVE-2024-21650)
998549	CVE-2023-6875、CVE-2023-7027	2.8.8 より前の WEB-WORDPRESS POST SMTP プラグイン-認証/XSS の脆弱性がありません (CVE-2023-6875、CVE-2023-7027)
998550	CVE-2023-51409	1.9.99 より前のウェブワードプレス AI エンジンプラグイン-認証されていない任意のファイルのアップロード (CVE-2023-51409)
998551	CVE-2023-46805、CVE-2024-21887	WEB-MISC Ivanti Connect セキュア複数バージョン-リモートコード実行の脆弱性 (CVE-2023-46805、CVE-2024-21887)
998552	CVE-2023-46805、CVE-2024-21887	WEB-MISC Ivanti Connect セキュア複数バージョン-リモートコード実行の脆弱性 (CVE-2023-46805、CVE-2024-21887)
998553	CVE-2023-22527	WEB-MISC アトラシアンコンフルエンスサーバーとデータセンターの複数のバージョン-リモートコード実行の脆弱性 (CVE-2023-22527)

署名更新バージョン **122**

February 15, 2024

2024-01-17 週に特定された脆弱性について、新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャーバージョン 122 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998554	CVE-2023-51467	WEB-MISC Apache Ofbiz 複数バージョン-サーバーサイドリクエストフォージェリの脆弱性 (CVE-2023-51467)
998555	CVE-2023-50968	WEB-MISC Apache Ofbiz 複数バージョン-サーバーサイドリクエストフォージェリの脆弱性 (CVE-2023-50968)

署名更新バージョン **121**

January 11, 2024

2024-01-09 週に特定された脆弱性について、新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャーバージョン 121 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998556	CVE-2023-6553	1.3.7 までの WEB-WORDPRESS バックアップ移行プラグイン-リモートコード実行の脆弱性 (CVE-2023-6553)
998557	CVE-2023-48777	3.18.1 より前の WEB-WORDPRESS Elementor プラグイン-ID によるファイルのアップロード/リモートコード実行の脆弱性（CVE-2023-48777）
998558	CVE-2023-44350	WEB-MISC Adobe ColdFusion 複数バージョン-リモートコード実行の脆弱性 (CVE-2023-44350)

署名更新バージョン **120**

January 9, 2024

2023-12-19 週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャーバージョン 120 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998559	CVE-2023-50164	6.3.0.2 より前の WEB-STRUTS Apache Struts-パストラバーサルの脆弱性（CVE-2023-50164）
998560	CVE-2023-49105	10.13.1 より前の WEB-MISC OwnCloud-アクセスコントロールバイパスの脆弱性 (CVE-2023-4105)
998561	CVE-2023-49103	WEB-MISC ownCloud Multiple Versions -情報開示の脆弱性（CVE-2023-49103）
998562	CVE-2023-47246	23.3.36 より前のオンプレミスの WEB-MISC SysAid サーバー-パストラバーサルの脆弱性（CVE-2023-47246）
998563	CVE-2023-46509	WEB-MISC Contec SolarView Compact 6.0 以前-OS コマンドインジェクションの脆弱性 (CVE-2023-46509)
998564	CVE-2023-44450	WEB-MISC 1.7.0.31 より前の NETGEAR ProSafe ネットワーク管理システム- SQL インジェクションの脆弱性 (CVE-2023-44450)
998565	CVE-2023-44449	WEB-MISC 1.7.0.31 より前の NETGEAR ProSafe ネットワーク管理システム- SQL インジェクションの脆弱性 (CVE-2023-44449)
998566	CVE-2023-44351、CVE-2023-44353	WEB-MISC Adobe ColdFusion-信頼できないデータの脆弱性の逆シリアル化 (CVE-2023-44351、CVE-2023-44353)
998567	CVE-2023-43177	10.5.1 より前の WEB-MISC CrushFTP-動的に管理されるコードリソースの不適切な制御の脆弱性（CVE-2023-43177）

署名ルール	**CVE ID**	説明
998568	CVE-2023-40062	WEB-MISC 2023.4.0 より前の SolarWinds Orion-テストアクションによる不適切な入力検証の脆弱性 (CVE-2023-40062)
998569	CVE-2023-40062	2023.4.0 より前の WEB-MISC SolarWinds Orion-/api/WriteToFile/ による不適切な入力検証の脆弱性 (CVE-2023-40062)
998570	CVE-2023-40055	WEB-MISC 2023.4.1 より前の SolarWinds NCM-結果をファイルに保存することによるディレクトリトラバーサルの脆弱性 (CVE-2023-40055)
998571	CVE-2023-40054	2023.4.1 より前の WEB-MISC SolarWinds NCM-txtConfigTemplate によるディレクトリトラバーサルの脆弱性 (CVE-2023-40054)
998572	CVE-2023-40054	WEB-MISC 2023.4.1 より前の SolarWinds NCM-テキストパスによるディレクトリトラバーサルの脆弱性 (CVE-2023-40054)
998573	CVE-2023-39912	7203 より前の Zoho AdManager Plus-ディレクトリトラバーサルの脆弱性 (CVE-2023-39912)
998574	CVE-2023-35150	WEB-MISC XWiki の複数のバージョン-任意のコードインジェクションの脆弱性（CVE-2023-35150）
998575	CVE-2023-32707	WEB-MISC Splunk Enterprise-権限昇格の脆弱性（CVE-2023-32707）
998576	CVE-2023-30943	4.1.3 より前の WEB-MISC Moodle-TinyMCE ローダーには、ローダーを介したクロスサイトスクリプティングの脆弱性が保存されています（CVE-2023-30943）

署名ルール	**CVE ID**	説明
998577	CVE-2023-30943	4.1.3 より前の WEB-MISC Moodle-TinyMCE ローダーには、言語によるクロスサイトスクリプティングの脆弱性が保存されています (CVE-2023-30943)
998578	CVE-2023-2943	7.0.1 より前の WEB-MISC OpenEMR-HTML コードインジェクションの脆弱性 (CVE-2023-2943)

署名更新バージョン **119**

December 8, 2023

2023-11-23 週に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 119 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998579	CVE-2023-40934	5.11.2 より前の WEB-MISC NagiosXi-SQL インジェクションの脆弱性 (CVE-2023-40934)

署名ルール	**CVE ID**	説明
998580	CVE-2023-40932	WEB-MISC NagiosXi 5.11.2 より前の-XSS 脆弱性 (CVE-2023-40932)
998581	CVE-2023-40045	WEB-MISC の進行状況 8.7.4 および 8.8.2 より前の WS_FTP サーバー-アドホッククロスサイトスクリプティングの脆弱性
998582	CVE-2023-37265	0.4.4 より前の WEB-MISC CasaOS-apps_restart によるリモートコード実行の脆弱性 (CVE-2023-37265)
998583	CVE-2023-37265	0.4.4 より前の WEB-MISC CasaOS-/app_management/compose によるリモートコード実行の脆弱性 (CVE-2023-37265)
998584	CVE-2023-3256	2.4.23 より前の WEB-MISC アドバンテック R-Seenet-ハードコードされた認証情報の利用に関する脆弱性 (CVE-2023-3256)
998585	CVE-2023-28323	WEB-MISC Ivanti エンドポイントマネージャー 2022 年まで Su3-信頼できないデータ脆弱性の逆シリアル化 (CVE-2023-28323)
998586	CVE-2023-1669	6.5.0.3 より前の WEB-WORDPRESS WP SeoPress プラグイン-PHP オブジェクトインジェクションの脆弱性 (CVE-2023-1669)
998587	CVE-2022-3214	WEB-MISC デルタエレクトロニクス DiaEnergie-HandlerUploadCalendar による認証されていない任意のファイルのアップロード (CVE-2022-3214)

署名ルール	**CVE ID**	説明
998588	CVE-2022-3214	WEB-MISC デルタエレクトロニクス DiaEnergie-HandlerUploadTag による認証されていない任意のファイルのアップロード (CVE-2022-3214)
998589	CVE-2022-3214	WEB-MISC デルタエレクトロニクス DiaEnergie-HandlerUploadCarbon による認証されていない任意のファイルのアップロード (CVE-2022-3214)
998590	CVE-2022-27665	WEB-MISC Progress WS_FTP Server 8.6.0-クロスサイトスクリプティングの脆弱性 (CVE-2022-27665)

署名更新バージョン **118**

December 8, 2023

2023-11-10 週に特定された脆弱性に対して、新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 118 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998591	CVE-2023-39968	2.7.2 より前の WEB-MISC Jupyter サーバー-オープンリダイレクトの脆弱性 (CVE-2023-39968)
998592	CVE-2023-38743	WEB-MISC Zoho ManageEngine AdManager Plus 7200 以前-リモートコード実行の脆弱性 (CVE-2023-38743)
998593	CVE-2023-22518	WEB-MISC Confluence データセンターとサーバーの複数のバージョン-不適切な認証の脆弱性 (CVE-2023-22518)
998594	CVE-2023-20890	WEB-MISC ネットワーク用 VMware Aria オペレーション-任意のファイル書き込みの脆弱性 (CVE-2023-20890)
998595	CVE-2023-20889	WEB-MISC ネットワーク用 VMware Aria オペレーション-コマンドインジェクションの脆弱性 (CVE-2023-20889)
998596	CVE-2023-20273	WEB-MISC Cisco IOS XE ソフトウェア-コマンドインジェクションの脆弱性 (CVE-2023-20273)

署名更新バージョン **117**

December 8, 2023

2023-11-01 週に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 117 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998597	CVE-2023-20198	WEB-MISC Cisco IOS XE ソフトウェア-認証バイパスの脆弱性 (CVE-2023-20198)

署名更新バージョン **116**

December 8, 2023

2023-10-16 週目に特定された脆弱性に対して、更新された署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 116 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

以下は、更新されたシグニチャルール、CVE ID、およびその説明です。

署名ルール	**CVE ID**	説明
998599	CVE-2023-22515	WEB-MISC Atlassian Confluence Server-/setup/ *.action によるアクセス制御の脆弱性が壊れている (CVE-2023-22515)

署名更新バージョン **115**

December 8, 2023

2023-10-14 週目に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 115 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998598	CVE-2023-24955	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability (CVE-2023-24955)
998599	CVE-2023-22515	WEB-MISC Atlassian Confluence Server-/setup/ *.action によるアクセス制御の脆弱性が壊れている (CVE-2023-22515)

署名ルール	**CVE ID**	説明
998600	CVE-2023-22515	WEB-MISC Atlassian Confluence Server-/server-info.action によるアクセス制御の脆弱性が壊れている (CVE-2023-22515)

署名更新バージョン **114**

October 25, 2023

2023-10-05 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 114 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler ADC CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998601	CVE-2023-42793	2023.05.4 より前の WEB-MISC JetBrains TeamCity-認証バイパスの脆弱性 (CVE-2023-42793)
998602	CVE-2023-40931	5.11.2 より前の WEB-MISC NagiosXi-SQL インジェクションの脆弱性 (CVE-2023-40931)
998603	CVE-2023-40044	WEB-MISC Progress WS_FTP サーバー-信頼できないデータの逆シリアル化の脆弱性 (CVE-2023-40044)

署名ルール	**CVE ID**	説明
998604	CVE-2023-39362	1.2.25 より前の WEB-MISC Cacti-OS コマンドインジェクションの脆弱性（CVE-2023-39362）
998605	CVE-2023-39361	1.2.25 より前の WEB-MISC Cacti-SQL インジェクションの脆弱性（CVE-2023-39361）
998606	CVE-2023-39359	1.2.25 より前の WEB-MISC Cacti-SQL インジェクションの脆弱性（CVE-2023-39359）
998607	CVE-2023-39358	1.2.25 より前の WEB-MISC Cacti-reports_admin による SQL インジェクションの脆弱性（CVE-2023-39358）
998608	CVE-2023-39358	1.2.25 より前の WEB-MISC Cacti-reports_user による SQL インジェクションの脆弱性（CVE-2023-39358）
998609	CVE-2023-35813	10.3 からの WEB-MISC サイトコア-リモートコード実行の脆弱性（CVE-2023-35813）
998610	CVE-2023-20890	ネットワーク用 WEB-MISC VMware Aria オペレーション-インフラ API によるパストラバーサルの脆弱性（CVE-2023-20890）
998611	CVE-2023-20890	ネットワーク用 WEB-MISC VMware Aria オペレーション-データソース API によるパストラバーサルの脆弱性（CVE-2023-20890）
998612	CVE-2022-43719	WEB-MISC アパッチスーパーセットの複数バージョン-CSRF の脆弱性（CVE-2022-43719）
998613	CVE-2022-40881	7.21 より前の WEB-MISC Contec SolarView Compact-OS コマンドインジェクションの脆弱性 (CVE-2022-40881)

署名更新バージョン **113**

October 25, 2023

2023-09-22 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 113 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler ADC CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998614	CVE-2023-38035	9.18.0 までの WEB-MISC Ivanti Sentry-/asproxy/services/ による不正な認証の脆弱性 (CVE-2023-38035)
998615	CVE-2023-38035	9.18.0 までの WEB-MISC Ivanti Sentry-/mics/services/ による不正な認証の脆弱性 (CVE-2023-38035)
998616	CVE-2023-36846	WEB-MISC Juniper JunoS SRX-webauth_operation による重要な機能の脆弱性に対する認証が欠落している（CVE-2023-36846）
998617	CVE-2023-3486	22.1.3 より前の WEB-MISC PaperCut NG-制限のないファイルアップロードの脆弱性 (CVE-2023-3486)

署名ルール	**CVE ID**	説明
998618	CVE-2023-34468、CVE-2023-40037	WEB-MISC Apache NiFI 複数バージョン-コマンドインジェクションの脆弱性 (CVE-2023-34468、CVE-2023-40037)
998619	CVE-2023-33653	ウェブその他のサイトコア-リモートコード実行の脆弱性（CVE-2023-33653）
998620	CVE-2023-33224、CVE-2023-23843	2023.3 以前の WEB-MISC SolarWinds Orion Platform-リモートコード実行の脆弱性 (CVE-2023-33224、CVE-2023-23843)
998621	CVE-2023-32566	WEB-MISC Ivanti Avalanche-セキュアフィルター認証バイパスの脆弱性（CVE-2023-32566）
998622	CVE-2023-32562	6.4.1 より前の WEB-MISC Ivanti Avalanche-無制限のファイルアップロードの脆弱性（CVE-2023-32562）
998623	CVE-2023-32315	WEB-MISC Ignite リアルタイムオープンファイア-パストラバーサルの脆弱性（CVE-2023-32315）
998624	CVE-2023-28128	6.4.0 より前の WEB-MISC Ivanti Avalanche-無制限のアップロードの脆弱性（CVE-2023-28128）
998625	CVE-2023-27066	10.2 までの WEB-MISC サイトコア-パストラバーサルの脆弱性 (CVE-2023-27066)
998626	CVE-2022-23333	7.21 より前の WEB-MISC Contec SolarView Compact-OS コマンドインジェクションの脆弱性 (CVE-2022-23333)
998627	CVE-2022-37044	8.8.15 P33 より前の WEB-MISC Zimbra コラボレーションスイート-オンロードによる XSS の脆弱性 (CVE-2022-37044)

署名ルール	**CVE ID**	説明
998628	CVE-2022-37044	8.8.15 P33 より前の WEB-MISC Zimbra コラボレーションスイート-追加による XSS の脆弱性 (CVE-2022-37044)
998629	CVE-2022-37044	8.8.15 P33 より前の WEB-MISC Zimbra コラボレーションスイート-タイトルによる XSS の脆弱性 (CVE-2022-37044)
998630	CVE-2022-24086	WEB-MISC Adobe Magento-ウィッシュリストによる任意のコード実行の脆弱性（CVE-2022-24086）
998631	CVE-2022-24086	WEB-MISC Adobe Magento-チェックアウトによる任意のコード実行の脆弱性（CVE-2022-24086）

署名更新バージョン **112**

October 25, 2023

2023-08-30 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 112 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler ADC CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998632	CVE-2023-39526	8.0.5、8.1.1、1.7.8.10 より前の WEB-MISC PrestaShop-アウトファイルによる任意のファイル書き込みの脆弱性（CVE-2023-39526）
998633	CVE-2023-39526	8.0.5、8.1.1、1.7.8.10 より前の WEB-MISC PrestaShop-ダンプファイルによる任意のファイル書き込みの脆弱性 (CVE-2023-39526)
998634	CVE-2023-39143	22.1.3 より前の WEB-MISC PaperCut NG/MF-カスタムレポートサンプルサーブレットにおけるパストラバーサルの脆弱性（CVE-2023-39143）
998635	CVE-2023-37979	3.6.25 までの WEB-WORDPRESS 忍者フォームお問い合わせフォームプラグイン-クロスサイトスクリプティングの脆弱性（CVE-2023-37979）
998636	CVE-2023-33652	ウェブその他のサイトコア-リモートコード実行の脆弱性（CVE-2023-33652）
998637	CVE-2023-32563	6.4.1 より前の WEB-MISC Ivanti Avalanche-任意のファイルアップロードの脆弱性（CVE-2023-32563）
998638	CVE-2023-29357	WEB-MISC Microsoft SharePoint Server-アクセストークン/プルーフトークンによる権限昇格の脆弱性 (CVE-2023-29357)
998639	CVE-2023-29357	WEB-MISC Microsoft SharePoint サーバー-認証ヘッダーによる権限昇格の脆弱性（CVE-2023-29357）
998640	CVE-2023-22480	3.16.4 より前の WEB-MISC KubeOperator-不適切な認証の脆弱性（CVE-2023-22480）

署名更新バージョン **111**

October 25, 2023

2023-08-04 の週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 111 は、NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998641	CVE-2023-37580	WEB-MISC Zimbra コラボレーションスイート複数バージョン-XSS の脆弱性 (CVE-2023-37580)
998642	CVE-2023-35082	WEB-MISC MobileIron Core (Ivanti EPMM) 11.2 より前-認証バイパス (CVE-2023-35082)
998643	CVE-2023-35078	WEB-MISC Ivanti エンドポイントマネージャーモバイル-認証バイパス (CVE-2023-35078)
998644	CVE-2023-34192	WEB-MISC Zimbra コラボレーションスイート複数バージョン-XSS の脆弱性 (CVE-2023-34192)
998645	CVE-2023-29382	WEB-MISC Zimbra コラボレーションスイート複数バージョン-RCE Via sfdc_preauth.jsp (CVE-2023-29382)

署名更新バージョン **110**

October 25, 2023

2023-07-25 の週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 110 は NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler ADC CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998646	CVE-2023-35036	WEB-MISC Progress MoveIt 転送-X-Silock-フォルダー ID のスマグリングによる認証済みの SQL インジェクションの脆弱性 (CVE-2023-35036)
998647	CVE-2023-35036	WEB-MISC Progress MoveIt 転送-X-Silock-FolderID による認証済みの SQL インジェクションの脆弱性（CVE-2023-35036）
998648	CVE-2023-3460	WEB-WORDPRESS 2.6.7 より前のアルティメットメンバーワードプレスプラグイン不適切な権限管理 (CVE-2023-3460)
998649	CVE-2023-33651	WEB-MISC サイトコア-MVC デバイスシミュレータによる認証ルールバイパスの脆弱性（CVE-2023-33651）

署名ルール	**CVE ID**	説明
998650	CVE-2023-33157	WEB-MISC Microsoft SharePoint-リモートコード実行の脆弱性 (CVE-2023-33157)
998651	CVE-2023-30777	WEB-WORDPRESS WordPress プラグインの高度なカスタムフィールド 6.1.5 まで-XSS の脆弱性が反映されている (CVE-2023-30777)
998652	CVE-2023-30545	8.0.4 および 1.7.8.9 より前の WEB-MISC PrestaShop-LOAD_FILE による任意のファイル読み取りの脆弱性 (CVE-2023-30545)
998653	CVE-2023-2986	WooCommerce プラグイン用 WEB-WORDPRESS 放棄カートライト 5.14.2 までの認証バイパス (CVE-2023-2986)
998654	CVE-2023-2982	WEB-WORDPRESS Wordpress プラグインの 7.6.4 以前のソーシャルログインと登録-認証バイパス (CVE-2023-2982)
998655	CVE-2023-29489	11.102.0.31 より前の WEB-MISC cPanel-XSS 脆弱性 (CVE-2023-29489)
998656	CVE-2023-29300、CVE-2023-38203、CVE-2023-38204	WEB-MISC Adobe ColdFusion-信頼できないデータの脆弱性の逆シリアル化 (CVE-2023-29300、CVE-2023-38203、CVE-2023-38204)
998657	CVE-2023-29298、CVE-2023-38205	WEB-MISC Adobe ColdFusion 複数バージョン-レストプレイによるアクセスコントロールバイパスの脆弱性 (CVE-2023-29298、CVE-2023-38205)

署名ルール	**CVE ID**	説明
998658	CVE-2023-29298、CVE-2023-38205	WEB-MISC Adobe ColdFusion 複数バージョン-cfide によるアクセス制御バイパスの脆弱性 (CVE-2023-29298、CVE-2023-38205)
998659	CVE-2023-28121	WEB-WORDPRESS WordPress プラグイン WooCommerce 支払い最大 5.6.1-権限昇格の脆弱性 (CVE-2023-28121)
998660	CVE-2023-27372	WEB-MISC SPIP 最大 3.2.17、4.0.0 から 4.0.9、4.1.0 から 4.1.7、4.2.0 までのリモートコード実行 (CVE-2023-27372)
998661	CVE-2023-27372	WEB-MISC SPIP 最大 3.2.17、4.0.0 から 4.0.9、4.1.0 から 4.1.7、4.2.0 までのリモートコード実行 (CVE-2023-27372)
998662	CVE-2023-27350	WEB-MISC PaperCut NG-認証バイパスの脆弱性 (CVE-2023-27350)
998663	CVE-2023-27067	10.2 までの WEB-MISC サイトコア-パストラバーサルの脆弱性 (CVE-2023-27067)
998664	CVE-2023-26360	WEB-MISC Adobe ColdFusion 2018 アップデート 16 以前とアップデート 6 より前の 2021-不適切なアクセスコントロール (CVE-2023-26360)
998665	CVE-2023-26262	WEB-MISC サイトコア-無制限の言語ファイルアップロードの脆弱性 (CVE-2023-26262)
998666	CVE-2023-2611	WEB-MISC Advantech R-Seenet 2.4.23 より前-ハードコードされた認証情報の使用の脆弱性 (CVE-2023-2611)
998667	CVE-2023-25804	6.3.6.0 より前の WEB-MISC Roxy-WI-パストラバーサルの脆弱性 (CVE-2023-25804)

署名ルール	**CVE ID**	説明
998668	CVE-2023-2575	WEB-MISC アドバンテック EKI-15XX-スタックベースのバッファオーバーフローの脆弱性 (CVE-2023-2575)
998669	CVE-2023-2574	WEB-MISC アドバンテック EKI-15XX-OS コマンドインジェクションの脆弱性 (CVE-2023-2574)
998670	CVE-2023-2573	WEB-MISC アドバンテック EKI-15XX-OS コマンドインジェクションの脆弱性 (CVE-2023-2573)
998671	CVE-2023-25690	WEB-MISC Apache HTTP サーバー 2.4.0 から 2.4.55-ラインフィードによるリクエスト密輸の脆弱性 (CVE-2023-25690)
998672	CVE-2023-25690	WEB-MISC Apache HTTP サーバー 2.4.0 から 2.4.55-キャリッジリターンによるリクエスト密輸の脆弱性 (CVE-2023-25690)
998673	CVE-2023-23489	WEB-WORDPRESS Wordpress プラグイン v3.1.0.2 より前の簡単デジタルダウンロード-SQL インジェクションの脆弱性 (CVE-2023-23489)
998674	CVE-2023-20887	WEB-MISC ネットワーク向け VMware Aria オペレーション-コマンドインジェクションの脆弱性 (CVE-2023-20887)
998675	CVE-2023-1671	WEB-MISC 4.3.10.4 より前の Sophos Web アプライアンス-コマンドインジェクション (CVE-2023-1671)
998676	CVE-2023-1196	WEB-WORDPRESS WordPress プラグイン 5.12.5 および 6.1.0 より前の高度なカスタムフィールド-信頼できない逆シリアル化 (CVE-2023-1196)

署名ルール	**CVE ID**	説明
998677	CVE-2023-1138	WEB-MISC Delta Electronics InfraSuite デバイスマスター 1.0.5 より前のバージョン-レポートによる情報開示（CVE-2023-1138）
998678	CVE-2023-1138	WEB-MISC Delta Electronics InfraSuite デバイスマスター 1.0.5 より前のバージョン-モジュール構成による情報開示（CVE-2023-1138）
998679	CVE-2023-1137	WEB-MISC Delta Electronics InfraSuite デバイスマスター 1.0.5 より前-情報開示の脆弱性 (CVE-2023-1137)
998680	CVE-2023-0255	WEB-WORDPRESS Wordpress プラグイン 4.0.2 より前のメディア置換を有効にする-任意のファイルをアップロードする脆弱性 (CVE-2023-0255)
998681	CVE-2022-36963	2023.2 より前の WEB-MISC SolarWinds プラットフォーム-テスト認証情報によるコマンドインジェクションの脆弱性 (CVE-2022-36963)
998682	CVE-2022-29303	WEB-MISC Contec SolarView Compact 7.21 より前-OS コマンドインジェクションの脆弱性 (CVE-2022-29303)
998683	CVE-2022-2185	WEB-MISC GitLab 14.10.5 および 15.1.1 より前の複数のバージョン-リモート実行の脆弱性 (CVE-2022-2185)
998684	CVE-2020-5284	WEB-MISC Next.js 9.3.2 より前のバージョン-パストラバーサルの脆弱性 (CVE-2020-5284)

署名更新バージョン **109**

October 25, 2023

2023-07-14 の週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 109 は NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998685	CVE-2023-36933	WEB-MISC Progress MoveIt Transfer 複数バージョン-サービス拒否の脆弱性（CVE-2023-36933）

署名更新バージョン **108**

October 25, 2023

2023-07-12 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャーバージョン 108 は NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998686	CVE-2023-36934	WEB-MISC Progress MoveIt Transfer 複数バージョン-SQL インジェクションの脆弱性 (CVE-2023-36934)
998687	CVE-2023-36932	WEB-MISC Progress MoveIt 転送 複数バージョン-フォルダリスト再帰による SQL インジェクションの脆弱性（CVE-2023-36932）

署名更新バージョン **107**

October 25, 2023

2023-06-16 週目に特定された脆弱性に対して新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 107 は NetScaler 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998688	CVE-2023-35708	WEB-MISC Progress MoveIt Transfer 複数バージョン-認証されていない SQL インジェクションの脆弱性（CVE-2023-35708）
998689	CVE-2023-35036	WEB-MISC Progress MoveIt Transfer 複数バージョン-認証されていない SQL インジェクションの脆弱性（CVE-2023-35036）

署名更新バージョン **106**

October 25, 2023

2023-06-16 週目に特定された脆弱性に対して新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 106 は、NetScaler 11.1、NetScaler 12.0、Citrix 12.1、Citrix 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998690	CVE-2023-34362	WEB-MISC プログレスの MOVEit による複数バージョンの転送-SQL インジェクションの脆弱性 (CVE-2023-34362)

署名ルール	**CVE ID**	説明
998691	CVE-2023-32243	5.7.1 までの Elementor 用の WEB-WORDPRESS ワードプレスプラグインの必須アドオン-権限昇格の脆弱性（CVE-2023-32243）
998692	CVE-2023-29084	7181 より前の WEB-MISC Zoho ManageEngine AdManager Plus-OS コマンドインジェクションの脆弱性（CVE-2023-29084）
998693	CVE-2023-29004	6.3.9.0 より前の WEB-MISC Roxy-WI-アブソリュートパストラバーサルの脆弱性 (CVE-2023-29004)
998694	CVE-2023-27351	WEB-MISC PaperCut NG-/AutoSetup/SetStatus による認証バイパスの脆弱性 (CVE-2023-27351)
998695	CVE-2023-27351	WEB-MISC PaperCut NG-/register または /RegisterCreate による認証バイパスの脆弱性 (CVE-2023-27351)
998696	CVE-2023-27351	WEB-MISC PaperCut NG-/キープアライブによる認証バイパスの脆弱性（CVE-2023-27351）
998697	CVE-2023-27350	WEB-MISC PaperCut NG-認証バイパスの脆弱性（CVE-2023-27350）
998698	CVE-2023-25812	リリース前の WEB-MISC MinIO。2023-02-17T17-52-43Z-権限の不適切な保存に関する脆弱性 (CVE-2023-25812)
998699	CVE-2023-25812	リリース前の WEB-MISC MinIO。2023-02-17T17-52-43Z-権限の不適切な保存に関する脆弱性 (CVE-2023-25812)
998700	CVE-2023-25803	6.3.6.0 より前の WEB-MISC Roxy-WI-パストラバーサルの脆弱性（CVE-2023-25803）

署名ルール	**CVE ID**	説明
998701	CVE-2023-24031	9.0.0 P30 より前の WEB-MISC Zimbra コラボレーションスイート-clazz による XSS の脆弱性 (CVE-2023-24031)
998702	CVE-2023-24031	9.0.0 P30 より前の WEB-MISC Zimbra コラボレーションスイート-代替キーによる XSS の脆弱性 (CVE-2023-24031)
998703	CVE-2023-24031	9.0.0 P30 より前の WEB-MISC Zimbra コラボレーションスイート-タイトルによる XSS の脆弱性 (CVE-2023-24031)
998704	CVE-2023-24031	9.0.0 P30 より前の WEB-MISC Zimbra コラボレーションスイート-カウンター経由の XSS 脆弱性 (CVE-2023-24031)
998705	CVE-2023-2338	v10.5.21 より前の WEB-MISC ピンコア-SQL インジェクションの脆弱性（CVE-2023-2338）
998706	CVE-2023-2336	v10.5.21 より前の WEB-MISC ピンコア-パストラバーサルの脆弱性 (CVE-2023-2336)
998707	CVE-2023-22973	7.0.0 より前の WEB-MISC OpenEMR-ローカルファイルインクルージョン (LFI) (CVE-2023-22973)
998708	CVE-2023-21742	WEB-MISC Microsoft シェアポイント-リモートコード実行の脆弱性 (CVE-2023-21742)
998709	CVE-2023-20864	ログ 8.10.2 用の WEB-MISC VMware Aria オペレーション-メンバーシップの承認によるデシリアライゼーションの脆弱性 (CVE-2023-20864)
998710	CVE-2023-20864	ログ 8.10.2 用の WEB-MISC VMware Aria オペレーション-セットトークンによるデシリアライゼーションの脆弱性（CVE-2023-20864）

署名ルール	**CVE ID**	説明
998711	CVE-2023-20864	ログ 8.10.2 用の WEB-MISC VMware Aria オペレーション-ApplyMembership によるデシリアライゼーションの脆弱性（CVE-2023-20864）
998712	CVE-2023-1578	v10.5.19 より前の WEB-MISC ピンコア-SQL インジェクションの脆弱性（CVE-2023-1578）
998713	CVE-2023-1406	3.1.3.1 より前のウェブワードプレスジェットエンジンプラグイン-リモートコード実行の脆弱性（CVE-2023-1406）
998714	CVE-2023-0315	WEB-MISC Froxlor Remote Code Execution (CVE-2023-0315)
998715	CVE-2022-45030	WEB-MISC RConfig 3.9.7 およびそれ以前のバージョン-SQL インジェクションの脆弱性（CVE-2022-45030）
998716	CVE-2022-43396	WEB-MISC アパッチキリン-設定の上書きによるコマンドインジェクションの脆弱性（CVE-2022-43396）
998717	CVE-2022-31700	WEB-MISC VMware Workspace ONE Access-マルチパートによるリモートコード実行の脆弱性（CVE-2022-31700）
998718	CVE-2022-31700	WEB-MISC VMware Workspace ONE Access-JSON によるリモートコード実行の脆弱性（CVE-2022-31700）
998719	CVE-2022-2884、CVE-2022-2992、CVE-2022-2865	WEB-MISC GitLab 複数バージョン-リモートコード実行の脆弱性（CVE-2022-2884、CVE-2022-2992、CVE-2022-2865）
998720	CVE-2022-27926	9.0.0 P24 より前の WEB-MISC Zimbra コラボレーションスイート-XSS 脆弱性（CVE-2022-27926）

署名ルール	**CVE ID**	説明
998721	CVE-2022-0824	1.990 より前の WebMin で認証テーマを使用した WEB-CGI によるリモートコード実行への不適切なアクセス制御（CVE-2022-0824）

署名更新バージョン **105**

October 25, 2023

2023-04-18 週目に特定された脆弱性に対して新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 105 は、NetScaler 11.1、NetScaler 12.0、Citrix 12.1、Citrix 13.0、NetScaler 13.1、NetScaler 14.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998722	CVE-2023-28432	リリース前の WEB-MISC MinIO。2023-03-20T20-16-18Z-情報漏えいの脆弱性 (CVE-2023-28432)
998723	CVE-2023-25802	6.3.7.0 より前の WEB-MISC Roxy-WI-パストラバーサルの脆弱性（CVE-2023-25802）

署名ルール	**CVE ID**	説明
998724	CVE-2023-23488	2.9.8 より前のウェブワードプレス有料メンバーシッププロ-REST_ROUTE 認証されていない SQL インジェクションの脆弱性 (CVE-2023-23488)
998725	CVE-2023-23488	2.9.8 より前の WEB-WORDPRESS 有料メンバーシッププロ- REST API による認証されていない SQL インジェクションの脆弱性 (CVE-2023-23488)
998726	CVE-2023-1658	WEB-MISC 3.5.2 より前のコンテック CONPROSYS HMI システム-認証前 SQL インジェクションの脆弱性 (CVE-2023-1658)
998727	CVE-2023-0955	14.0 より前の WEB-WORDPRESS WP 統計プラグイン-ID による REST_ROUTE SQL インジェクションの脆弱性 (CVE-2023-0955)
998728	CVE-2023-0955	14.0 より前の WEB-WORDPRESS WP 統計プラグイン-ID による REST API SQL インジェクションの脆弱性（CVE-2023-0955）
998729	CVE-2023-0955	14.0 より前の WEB-WORDPRESS WP 統計プラグイン-タイプによる REST_ROUTE SQL インジェクションの脆弱性 (CVE-2023-0955)
998730	CVE-2023-0955	14.0 より前の WEB-WORDPRESS WP 統計プラグイン-タイプによる REST API SQL インジェクションの脆弱性 (CVE-2023-0955)
998731	CVE-2023-0669	7.1.2 より前の WEB-MISC Fortra GoAnywhere MFT-認証されていないリモートコード実行の脆弱性 (CVE-2023-0669)
998732	CVE-2022-24697	WEB-MISC アパッチキリン-設定の上書きによるコマンドインジェクションの脆弱性（CVE-2022-24697）

署名ルール	**CVE ID**	説明
998733	CVE-2022-21587	WEB-MISC Oracle Web アプリケーションデスクトップインテグレーター-BNEOfflineLOVService によるパストラバーラルの脆弱性 (CVE-2022-21587)
998734	CVE-2022-21587	WEB-MISC Oracle Web アプリケーションデスクトップインテグレーター-BNEDownloadService によるパストラバーラルの脆弱性 (CVE-2022-21587)
998735	CVE-2022-21587	WEB-MISC Oracle Web アプリケーションデスクトップインテグレーター-BNEviewerXMLService によるパストラバーラルの脆弱性 (CVE-2022-21587)
998736	CVE-2022-21587	WEB-MISC Oracle Web アプリケーションデスクトップインテグレーター-BNEUploaderService によるパストラバーラルの脆弱性 (CVE-2022-21587)

署名更新バージョン **104**

October 25, 2023

2023-03-28 週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 104 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998737	CVE-2023-25135	WEB-MISC vBulletin 複数バージョン-検索設定による PHP オブジェクトインジェクションの脆弱性 (CVE-2023-25135)
998738	CVE-2023-25135	WEB-MISC vBulletin 複数バージョン-pm フォルダーによる PHP オブジェクトインジェクションの脆弱性（CVE-2023-25135）
998739	CVE-2023-25135	WEB-MISC vBulletin 複数バージョン-サブフォルダーによる PHP オブジェクトインジェクションの脆弱性（CVE-2023-25135）
998740	CVE-2023-23752	ウェブその他 Joomla!4.x から 4.2.7 まで-API の不適切なアクセスチェックの脆弱性 (CVE-2023-23752)
998741	CVE-2023-22974	7.0.0 より前の WEB-MISC OpenEMR-情報漏えいの脆弱性 (CVE-2023-22974)
998742	CVE-2023-22952	12.0 より前の WEB-MISC SugarCRM ホットフィックス 91155-メールテンプレート PHP コードインジェクションの脆弱性 (CVE-2023-22952)
998743	CVE-2023-22374	WEB-MISC F5 BIG-IP マルチバージョン-フォーマット文字列の脆弱性 (CVE-2023-22374)
998744	CVE-2023-20858	WEB-MISC VMware カーボンブラックアプリケーションコントロール複数バージョン-SQL インジェクションの脆弱性（CVE-2023-20858）

署名ルール	**CVE ID**	説明
998745	CVE-2022-47002、CVE-2022-47003	WEB-MISC Mura CMS と Masa CMS-認証バイパス脆弱性 (CVE-2022-47002、CVE-2022-47003)
998746	CVE-2022-4506	7.0.0.2 より前の WEB-MISC OpenEMR-任意のファイルアップロードの脆弱性 (CVE-2022-4506)
998747	CVE-2022-44298	7.2.0 より前の WEB-MISC サイトサーバー CMS-SQL インジェクションの脆弱性 (CVE-2022-44298)
998748	CVE-2022-44297	7.2.0 より前の WEB-MISC サイトサーバー CMS-SQL インジェクションの脆弱性 (CVE-2022-44297)
998749	CVE-2022-43709	1.8.32 より前の WEB-MISC MyBB-SQL インジェクションの脆弱性 (CVE-2022-43709)
998750	CVE-2022-40300	WEB-MISC Zoho ManageEngine PasswordManagerPro、PAM360、Access ManagerPlus には SQL インジェクションの脆弱性があります。(CVE-2022-40300)
998751	CVE-2022-36633	WEB-MISC テレポート 9.3.6-コマンドインジェクション (CVE-2022-36633)
998752	CVE-2022-35947	10.0.2 までの WEB-MISC GLPI-JSON による SQL インジェクションの脆弱性 (CVE-2022-35947)
998753	CVE-2022-35947	10.0.2 までの WEB-MISC GLPI-フォームによる SQL インジェクションの脆弱性 (CVE-2022-35947)
998754	CVE-2022-35914	10.0.2 までの WEB-MISC GLPI-HTMLAwedTest における PHP コードインジェクションの脆弱性 (CVE-2022-35914)
998755	CVE-2022-30547	ウェブその他 WBN ビデオパストラバーサル (CVE-2022-30547)

署名ルール	**CVE ID**	説明
998756	CVE-2022-24734	1.8.30 より前の WEB-MISC MyBB-リモートでコードが実行される脆弱性 (CVE-2022-24734)
998757	CVE-2020-17496	WEB-MISC vBulletin 5.5.4 から 5.6.2-ルートストリングを介したリモートコード実行の脆弱性 (CVE-2020-17496)
998758	CVE-2020-17496	WEB-MISC vBulletin 5.5.4 から 5.6.2-リモートでコードが実行される脆弱性 (CVE-2020-17496)
998759	CVE-2019-16759	WEB-MISC vBulletin 5.x から 5.5.4-ルートストリングを介したリモートコード実行の脆弱性 (CVE-2019-16759)
998760	CVE-2019-16759	WEB-MISC vBulletin 5.x から 5.5.4-リモートでコードが実行される脆弱性 (CVE-2019-16759)

署名更新バージョン **103**

October 25, 2023

2023-03-01 週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 103 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998761	CVE-2022-45094	WEB-MISC V1.0 SP2 アップデート 1 より前のシーメンス SINEC INS-BootFileLoc によるリモートコード実行の脆弱性 (CVE-2022-45094)
998762	CVE-2022-40044, CVE-2022-40043	22.04.1 より前の WEB-MISC Centreon-esc_name と esc_alias による XSS の脆弱性 (CVE-2022-40044)
998763	CVE-2022-3361	2.5.1 以前の WEB-MISC WordPress プラグインのアルティメットメンバー-コマンドインジェクションの脆弱性（CVE-2022-3361）
998764	CVE-2022-32573	10.3.1.0 より前の WEB-MISC ランスイーパー-txtdocname によるパストラバーサルの脆弱性 (CVE-2022-32573)
998765	CVE-2022-29517	10.3.1.0 より前の WEB-MISC ランスイーパー-インラインアタッチメント名によるパストラバーサルの脆弱性（CVE-2022-29517）
998766	CVE-2022-29511	10.3.1.0 より前の WEB-MISC ランスイーパー-情報漏えいの脆弱性 (CVE-2022-29511)
998767	CVE-2022-29081	WEB-MISC Zoho ManageEngine の複数の製品-REST API アクセスコントロールバイパスの脆弱性 (CVE-2022-29081)
998768	CVE-2022-25487	2.1 より前の WEB-MISC AtomCMS-無制限のファイルアップロードの脆弱性 (CVE-2022-25487)
998769	CVE-2021-26086	WEB-MISC アトラシアン Jira サーバーとデータセンター-WEB-INF を介した情報開示の脆弱性 (CVE-2021-26086)

署名ルール	**CVE ID**	説明
998770	CVE-2021-26086	WEB-MISC アトラシアン Jira サーバーとデータセンター-META-INF を介した情報開示の脆弱性 (CVE-2021-26086)
998771	CVE-2021-26085、CVE-2021-26086	WEB-MISC Atlassian Confluence サーバー-WEB-INF による情報開示の脆弱性 (CVE-2021-26085、CVE-2021-26086)
998772	CVE-2021-26085、CVE-2021-26086	WEB-MISC Atlassian Confluence サーバー-META-INF による情報開示の脆弱性 (CVE-2021-26085、CVE-2021-26086)
998773	CVE-2020-13818	125144 より前の WEB-MISC Zoho ManageEngine OpManager-ディレクトリトラバーサルの脆弱性 (CVE-2020-13818)

署名更新バージョン **102**

October 25, 2023

2023-02-03 週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 102 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998774	CVE-2022-47966	WEB-MISC Zoho ManageEngine 製品-SAMLResponse サーブレットエンドポイントの XSL 変換による RCE の脆弱性 (CVE-2022-47966)
998775	CVE-2022-47966	WEB-MISC Zoho ManageEngine 製品-SAMLLogin エンドポイントの XSL 変換による RCE の脆弱性 (CVE-2022-47966)
998776	CVE-2022-47615	4.1.7.3.2 までの WEB-WORDPRESS LearnPress プラグイン-REST_ROUTE ローカルファイルインクルードの脆弱性 (CVE-2022-47615)
998777	CVE-2022-47615	4.1.7.3.2 までの WEB-WORDPRESS LearnPress プラグイン-REST API ローカルファイルインクルードの脆弱性 (CVE-2022-47615)
998778	CVE-2022-46169	1.2.23 より前の WEB-MISC サボテンサーバー-コマンドインジェクション (CVE-2022-46169)
998779	CVE-2022-45808	4.2 より前の WEB-WORDPRESS LearnPress プラグイン-order_by による REST_ROUTE SQL インジェクションの脆弱性 (CVE-2022-45808)
998780	CVE-2022-45808	4.2 より前の WEB-WORDPRESS LearnPress プラグイン-order_by による REST API SQL インジェクションの脆弱性 (CVE-2022-45808)
998781	CVE-2022-45808	4.2 より前の WEB-WORDPRESS LearnPress プラグイン-注文による REST_ROUTE SQL インジェクションの脆弱性 (CVE-2022-45808)

署名ルール	**CVE ID**	説明
998782	CVE-2022-45808	4.2 より前の WEB-WORDPRESS LearnPress プラグイン-注文による REST API SQL インジェクションの脆弱性 (CVE-2022-45808)
998783	CVE-2022-44877	0.9.8.1147 より前の WEB-MISC コントロール Web パネル (CWP) 7-OS コマンドインジェクションの脆弱性 (CVE-2022-44877)
998784	CVE-2022-43473	126141 より前の WEB-MISC Zoho ManageEngine OpManager-XML 外部エンティティインジェクションの脆弱性 (CVE-2022-43473)
998785	CVE-2022-43447	WEB-MISC Delta Electronics DIAenergie-TxTPF 経由の AM_EBillAnalysis における SQL インジェクションの脆弱性 (CVE-2022-43447)
998786	CVE-2022-43447	WEB-MISC Delta Electronics DIAenergie-TxtFAV 経由の AM_eBillAnalysis における SQL インジェクションの脆弱性 (CVE-2022-43447)
998787	CVE-2022-4323	6.5.6 より前の WEB-WORDPRESS Google アナリティケータープラグイン-PHP オブジェクトインジェクションの脆弱性 (CVE-2022-4323)
998788	CVE-2022-42904	7160 以前の WEB-MISC Zoho ManageEngine アドマネージャープラス-OS コマンドインジェクションの脆弱性 (CVE-2022-42904)
998789	CVE-2022-34271	2.3.0 より前の WEB-MISC アパッチアトラス-任意のファイルアップロードの脆弱性（CVE-2022-34271）

署名更新バージョン **101**

October 25, 2023

2023-01-24 週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 101 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998790	CVE-2022-43452	WEB-MISC デルタエレクトロニクス DiaEnergie-データ項目行の削除による SQL インジェクションの脆弱性（CVE-2022-43452）
998791	CVE-2022-43452	WEB-MISC デルタエレクトロニクス DiaEnergie-データ型行の削除による SQL インジェクションの脆弱性（CVE-2022-43452）
998792	CVE-2022-41080	WEB-MISC Microsoft Exchange Server-OWA サーバー側リクエスト偽造脆弱性 (CVE-2022-41080)
998793	CVE-2022-40309	2.2.9 より前の WEB-MISC Apache アーカイブ-任意のディレクトリ削除の脆弱性 (CVE-2022-40309)
998794	CVE-2022-40308	2.2.9 より前の WEB-MISC Apache アーカイブ-任意のファイル読み取りの脆弱性 (CVE-2022-40308)

署名ルール	**CVE ID**	説明
998795	CVE-2022-36962	2022.4 より前の WEB-MISC SolarWinds プラットフォーム-ディスクへの保存の作成または更新による RCE の脆弱性 (CVE-2022-36962)
998796	CVE-2022-36962	2022.4 より前の WEB-MISC SolarWinds プラットフォーム-電子メール URL の作成または更新による RCE の脆弱性 (CVE-2022-36962)
998797	CVE-2022-3361	2.5.1 より前の WEB-WORDPRESS WordPress プラグインアルティメットメンバー-ディレクトリトラバーサル (CVE-2022-3361)
998798	CVE-2022-24254	4.0.1 より前の WEB-MISC 拡張ポートフォリオ-バックアップ復元による任意のファイルアップロードの脆弱性（CVE-2022-24254）
998799	CVE-2022-24253	4.0.1 より前の WEB-MISC 拡張ポートフォリオ-ブランディングアップロードによるパストラバーサルの脆弱性（CVE-2022-24253）
998800	CVE-2022-0224	14.0.6 より前の WEB-MISC ドリバール-SQL インジェクションの脆弱性 (CVE-2022-0224)
998801	CVE-2021-35232	12.7.7 ホットフィックス 1 より前の WEB-MISC SolarWinds ウェブヘルプデスク-ハードコードされた認証情報の使用に関する脆弱性 (CVE-2021-35232)

署名更新バージョン **100**

October 25, 2023

2023-01-05 の週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 100 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998802	CVE-2022-47945	WEB-MISC ThinkPHP-ヘッダー経由での言語ファイルインクルージョン RCE の脆弱性 (CVE-2022-47945)
998803	CVE-2022-47945	WEB-MISC ThinkPHP-クッキーによる言語ファイルインクルージョン RCE の脆弱性 (CVE-2022-47945)
998804	CVE-2022-47945	WEB-MISC ThinkPHP-フォームによる言語ファイルインクルージョン RCE の脆弱性 (CVE-2022-47945)
998805	CVE-2022-45462	2.0.6 より前の WEB-MISC Apache DolphinScheduler -OS コマンドインジェクションの脆弱性 (CVE-2022-45462)
998806	CVE-2022-44635	1.8.1 より前の WEB-MISC Apache Fineract -ファイルアップロードパストラバーサルの脆弱性 (CVE-2022-44635)
998807	CVE-2022-43672	ウェブその他 Zoho ManageEngine の複数の製品-SQL インジェクションの脆弱性 (CVE-2022-43672)

署名ルール	**CVE ID**	説明
998808	CVE-2022-43671	ウェブその他 Zoho ManageEngine の複数の製品-SQL インジェクションの脆弱性 (CVE-2022-43671)
998809	CVE-2022-41773	WEB-MISC WEB-MISC Delta DIAEnergie -チェック・ディア・クラウドによる SQL インジェクションの脆弱性 (CVE-2022-41773)
998810	CVE-2022-41351	WEB-MISC Zimbra コラボレーションスイート 8.8.15-XSS の脆弱性 (CVE-2022-41351)
998811	CVE-2022-41350	WEB-MISC Zimbra コラボレーションスイート 8.8.15-XSS の脆弱性 (CVE-2022-41350)
998812	CVE-2022-41133	WEB-MISC Delta Electronics DIAEnergie-getDIAE_LINE_Message_SettingsLIST パラメーターによる SQLi の脆弱性 (CVE-2022-41133)
998813	CVE-2022-40967	WEB-MISC Delta Electronics DIAEnergie -CheckIoT サブネット名による SQL インジェクションの脆弱性が存在しました (CVE-2022-40967)
998814	CVE-2022-34662	2.0.6 より前の WEB-MISC Apache DolphinScheduler -パストラバーサルの脆弱性（CVE-2022-34662）
998815	CVE-2022-3383、CVE-2022-3384	2.5.2 より前の WEB-WORDPRESS WordPress プラグインのアルティメットメンバー-コマンドインジェクションの脆弱性 (CVE-2022-3383、CVE-2022-3384)
998816	CVE-2022-31698	6.5U3U、6.7.0U3S、7.0U3i より前の WEB-MISC VMware vCenter サーバ-サービス拒否の脆弱性 (CVE-2022-31698)

署名ルール	**CVE ID**	説明
998817	CVE-2022-20867	WEB-MISC Cisco E メールセキュリティアプライアンス-SQL インジェクションの脆弱性 (CVE-2022-20867)

署名更新バージョン **99**

October 25, 2023

2022-12-20 週目に特定された脆弱性に対して新しいシグネチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 99 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998818	CVE-2022-1361	WEB-MISC オンプレミス変更ネットワーク CNMaestro-MAC 経由の SQL インジェクションの脆弱性 (CVE-2022-1361)
998819	CVE-2022-1361	WEB-MISC オンプレミス変更ネットワーク CNMaestro-シリアル番号による SQL インジェクションの脆弱性 (CVE-2022-1361)

署名更新バージョン **98**

October 25, 2023

2022-12-06 週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 98 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998820	CVE-2022-43781	WEB-MISC アトラシアンビットバケットサーバーの複数バージョン-OS コマンドインジェクションの脆弱性 (CVE-2022-43781)
998821	CVE-2022-43775	WEB-MISC Dia Energie-HandlerTag_KID による SQL インジェクションの脆弱性 (CVE-2022-43775)
998822	CVE-2022-43774	WEB-MISC Dia Energie-HandlerPagep_KID と KID による SQL インジェクションの脆弱性 (CVE-2022-43774)
998823	CVE-2022-43774	WEB-MISC Dia Energie-HandlerPagep_Kid と HTMLID による SQL インジェクションの脆弱性 (CVE-2022-43774)

署名ルール	**CVE ID**	説明
998824	CVE-2022-42977、CVE-2022-42978	1.3.5 より前の WEB-MISC Netic Confluence ユーザーエクスポートアプリ-情報漏えいの脆弱性 (CVE-2022-42977、CVE-2022-42978)
998825	CVE-2022-40127	2.4.0 より前の WEB その他 Apache のエアフロー-API 経由の DAG コマンドインジェクションの脆弱性の例（CVE-2022-40127）
998826	CVE-2022-40127	2.4.0 より前の WEB-MISC Apache エアフロー-トリガーによる Dags コマンドインジェクションの脆弱性の例（CVE-2022-40127）
998827	CVE-2022-39298	5.0.1 より前の WEB-Misc Melis プラットフォーム-MelisFront の任意の逆シリアル化の脆弱性 (CVE-2022-39298)
998828	CVE-2022-39297	5.0.1 より前の WEB-Misc Melis プラットフォーム-MelisCMS による任意の逆シリアル化の脆弱性 (CVE-2022-39297)
998829	CVE-2022-39296	5.0.1 より前の WEB-Misc Melis プラットフォーム-MelisAssetManager による任意のファイル読み取りの脆弱性 (CVE-2022-39296)
998830	CVE-2022-38772	WEB-MISC Zoho ManageEngine の複数の製品-NMAPScan オプションの設定による OS コマンドインジェクションの脆弱性 (CVE-2022-38772)
998831	CVE-2022-35933	5.0.2 より前の WEB-MISC プレスタショップ製品コメント-クロスサイトスクリプティングの脆弱性 (CVE-2022-35933)
998832	CVE-2022-3214	WEB-その他デルタ・ダイアエナジー-ハードコードされた認証情報の使用の脆弱性（CVE-2022-3214）

署名ルール	**CVE ID**	説明
998833	CVE-2022-24716	WEB-MISC Icina Web 2-icinga-php ライブラリによる任意のファイル読み取りの脆弱性 (CVE-2022-24716)
998834	CVE-2022-24716	WEB-MISC Icina Web 2-icinga-php サードパーティによる任意のファイル読み取りの脆弱性 (CVE-2022-24716)
998835	CVE-2022-24715	WEB-MISC アイシングアウェブ 2-パストラバーサルの脆弱性 (CVE-2022-24715)
998836	CVE-2022-2139	WEB-MISC Advantech iView 5.7.04.6469 より前-NetworkServlet URI とファイル名によるパストラバーサルの脆弱性 (CVE-2022-2139)
998837	CVE-2022-2139	WEB-MISC 5.7.04.6469 より前のアドバンテック iView-CommandServlet URI とファイル名によるパストラバーサルの脆弱性 (CVE-2022-2139)
998838	CVE-2021-39144	WEB-MISC VMware クラウドファンデーション 3.x-XStream 経由でのリモートコード実行の脆弱性 (CVE-2021-39144)
998839	CVE-2021-35220	WEB-MISC 2020.2.6 HF1 以前の SolarWinds Orion-電子メール Web ページおよびテストアクションによる RCE の脆弱性 (CVE-2021-35220)
998840	CVE-2021-35220	WEB-MISC 2020.2.6 HF1 より前の SolarWinds Orion-電子メールによる RCE 脆弱性ウェブページの作成または更新 (CVE-2021-35220)

署名更新バージョン **97**

October 25, 2023

2022-11-15 週に特定された脆弱性に対して新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 97 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998841	CVE-2022-40043	22.04.1 より前の WEB-MISC Centreon-esc_name 経由の SQL インジェクションの脆弱性 (CVE-2022-40043)
998842	CVE-2022-35153	WEB-MISC FusionPBX 5.0.1 およびそれ以前-OS コマンドインジェクションの脆弱性 (CVE-2022-35153)
998843	CVE-2022-3387	2.4.21 より前の WEB-MISC アドバンテック R-SEENet-パストラバーサルの脆弱性 (CVE-2022-3387)
998844	CVE-2022-3385	2.4.21 より前の WEB-MISC アドバンテック R-SEENet-ファイル名によるバッファオーバーフローの脆弱性 (CVE-2022-3385)
998845	CVE-2022-31680	6.5 U3u より前の WEB-MISC VMware vCenter Server-PSC による安全でない逆シリアル化の脆弱性 (CVE-2022-31680)

署名ルール	**CVE ID**	説明
998846	CVE-2022-28732	WEB-MISC Apache JSPWiki 2.11.3 より前 の-Weblog.StartDate 経由の WebLogPlugin XSS の脆弱性 (CVE-2022-28732)
998847	CVE-2022-28732	2.11.3 より前の WEB-MISC Apache JSPWiki-開始日による WebLogPlugin XSS の脆弱性 (CVE-2022-28732)
998848	CVE-2022-28730	2.11.3 より前の WEB-MISC Apache JSPWiki-Denounce プラグインによる AjaxPreview XSS の脆弱性 (CVE-2022-28730)
998849	CVE-2022-23463	WEB-MISC ネプクションディスカバリー-SPeL インジェクションの脆弱性 (CVE-2022-23463)

署名更新バージョン **96**

October 25, 2023

2022-10-23 の週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 96 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998850	CVE-2022-42889	WEB-MISC Apache Commons テキスト-URL 経由でのリモートコード実行の脆弱性 (CVE-2022-42889)
998851	CVE-2022-42889	WEB-MISC Apache Commons テキスト-ヘッダー経由でのリモートコード実行の脆弱性 (CVE-2022-42889)
998852	CVE-2022-42889	WEB-MISC Apache Commons テキスト-BODY によるリモートコード実行の脆弱性 (CVE-2022-42889)
998853	CVE-2022-42889	WEB-MISC Apache Commons テキスト-フォーム経由でのリモートコード実行の脆弱性 (CVE-2022-42889)
998854	CVE-2022-38358	WEB-MISC Eyes of Network -admin_user 経由の XSS 脆弱性 (CVE-2022-38358)
998855	CVE-2022-38358	WEB-MISC Eyes of Network -admin_notifier による XSS の脆弱性 (CVE-2022-38358)
998856	CVE-2022-38358	WEB-MISC Eyes of Network -report_event 経由の XSS 脆弱性 (CVE-2022-38358)
998857	CVE-2022-38257	WEB-MISC Eyes of Network -iFrame インジェクションの脆弱性 (CVE-2022-38257)
998858	CVE-2022-36981	WEB-MISC 6.3.4 より前の Ivanti Avalanche-パストラバーサルの脆弱性によりリモートでコードが実行される（CVE-2022-36981）
998859	CVE-2022-36961	2022.3 より前の WEB-MISC SolarWinds Orion-SQL インジェクションの脆弱性 (CVE-2022-36961)

署名ルール	**CVE ID**	説明
998860	CVE-2022-36804	WEB-MISC Atlassian Bitbucket サーバーとデータセンター-ボディ経由でのリモートコード実行の脆弱性 (CVE-2022-36804)
998861	CVE-2022-36804	WEB-MISC Atlassian Bitbucket サーバーとデータセンター-URL 経由でのリモートコード実行の脆弱性 (CVE-2022-36804)
998862	CVE-2022-3323	WEB-MISC Advantech iView 5.7.04.6469-CommandServlet URI と column_value による SQL インジェクションの脆弱性 (CVE-2022-3323)
998863	CVE-2022-3323	WEB-MISC Advantech iView 5.7.04.6469-CommandServlet URI と column_name による SQL インジェクションの脆弱性 (CVE-2022-3323)
998864	CVE-2022-3323	WEB-MISC Advantech iView 5.7.04.6469-ConfigurationServlet URI と column_value による SQL インジェクションの脆弱性 (CVE-2022-3323)
998865	CVE-2022-3323	WEB-MISC Advantech iView 5.7.04.6469-ConfigurationServlet URI と column_name による SQL インジェクションの脆弱性 (CVE-2022-3323)
998866	CVE-2022-29548	WEB-MISC WSO2 複数製品-不正ログインステータスによる XSS 脆弱性 (CVE-2022-29548)
998867	CVE-2022-29548	WEB-MISC WSO2 複数製品-ログイン失敗ステータスによる XSS 脆弱性 (CVE-2022-29548)

署名ルール	**CVE ID**	説明
998868	CVE-2022-2142	WEB-MISC 5.7.04.6469 より前の Advantech iView-CommandServlet によるセカンドオーダー SQL インジェクションの脆弱性 (CVE-2022-2142)
998869	CVE-2022-2142	WEB-MISC Advantech iView 5.7.04.6469 より前-NetworkServlet 経由のセカンドオーダー SQL インジェクションの脆弱性 (CVE-2022-2142)
998870	CVE-2022-0666	1.2.11 より前の WEB-MISC Microweber -CRLF インジェクションの脆弱性 (CVE-2022-0666)

署名更新バージョン **95**

October 25, 2023

2022-10-07 の週に特定された脆弱性について、変更された署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 95 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
811	CVE-2000-0066	WEB-CGI websitepro パスアクセス
1029	NESSUS-11032	WEB-IIS スクリプト-ブラウズアクセス
1047	CVE-2001-0251	WEB-MISC Netscape Enterprise DOS
1048	CVE-2001-0250	WEB-MISC Netscape Enterprise ディレクトリ一覧表示の試み
1663	NESSUS-11007	WEB-MISC *%20.pl アクセス
1725	CVE-2000-0630、CVE-2001-0004	WEB-IIS +.htr コードフラグメントの試み
16521	CVE-2009-0478	WEB-CLIENT Squid Proxy http バージョン番号オーバーフローの試み

署名更新バージョン **94**

October 25, 2023

2022-10-06 の週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 94 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998871	CVE-2022-41082、 CVE-2022-41040	WEB-MISC Microsoft Exchange Server-RCE 脆弱性 (CVE-2022-41082、 CVE-2022-41040)

署名更新バージョン **93**

October 25, 2023

2022-10-02 週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 93 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998871	CVE-2022-41082、 CVE-2022-41040	WEB-MISC Microsoft Exchange Server-RCE 脆弱性 (CVE-2022-41082、 CVE-2022-41040)
998872	CVE-2022-37299	WEB-MISC Shirne CMS 1.2.0-/static/ueditor/php/controller.php 経由のパストラバーサルの脆弱性 (CVE-2022-37299)

署名ルール	**CVE ID**	説明
998873	CVE-2022-36923	WEB-MISC Zoho ManageEngine 複数製品複数バージョン-認証バイパスの脆弱性 (CVE-2022-36923)
998874	CVE-2022-33891	WEB-MISC Apache Spark UI 複数バージョン-DoAS パラメータによるリモートコード実行の脆弱性 (CVE-2022-33891)
998875	CVE-2022-3184、CVE-2022-3183	1.42.06162022 より前の WEB-MISC DataProbe iBoot-PDU-リモートコード実行の脆弱性 (CVE-2022-3184、CVE-2022-3183)
998876	CVE-2022-31814	2.1.4_26 より前の WEB-MISC pfSense pfBlockerNG-リモートコード実行の脆弱性 (CVE-2022-31814)
998877	CVE-2022-31097	WEB-MISC Apache Grafana-統合アラート保存 XSS 脆弱性 (CVE-2022-31097)
998878	CVE-2022-2903	3.6.13 より前の WEB-WORDPRESS NinjaForms プラグイン-PHP オブジェクトインジェクションの脆弱性 (CVE-2022-2903)
998879	CVE-2022-2552	1.4.7.1 より前の WEB WORDPRESS デュプリケータプラグイン-認証されていない情報漏えいの脆弱性（CVE-2022-2552）
998880	CVE-2022-23854	WEB-MISC AVEVA InTouch どこからでもアクセスできる Secure Gateway-SG URI 経由のパストラバーサルの脆弱性 (CVE-2022-23854)

署名ルール	**CVE ID**	説明
998881	CVE-2022-23854	WEB-MISC AVEVA InTouch どこからでもアクセスできる Secure Gateway-Blaze URI 経由のパストラバーサルの脆弱性 (CVE-2022-23854)
998882	CVE-2022-23854	WEB-MISC AVEVA InTouch どこでもアクセス Secure Gateway-AccessAnywhere URI 経由のパストラバーサルの脆弱性 (CVE-2022-23854)
998883	CVE-2017-9841	WEB-MISC PHPUnit 4.8.28 以前と 5.6.3 より前の 5.x-eval-stdin.php 経由でのリモートコード実行の脆弱性 (CVE-2017-9841)

署名規則の統合と更新

いくつかの冗長なシグニチャルールが削除され、これらのルールの CVE ID が更新されたルールに統合されます。削除した各ルールに対応するシグニチャルールを必ず有効にしてください。

次の表は、統合および更新されたシグニチャルール ID を示しています。

削除されたシグニチャルール	署名規則の更新	CVE ID
1242	1243	CVE-2000-0071
1245	1244	CVE-2000-0071
1589	1221	CVE-2001-0224、NESSUS-10609
1648	832	CVE-1999-0509、NESSUS-10173、www.cert.org/advisories/CA-1996-11.html
1700	821	CVE-1999-0951、NESSUS-10122
2598	2597	CVE-2004-0600
999779	999721	CVE-2019-14994
999861	999859	CVE-2019-12099

削除されたシグニチャルール	署名規則の更新	CVE ID
999862	999857	https://www.wordfence.com/blog/2019/05s-command-injection-vulnerability-patched-in-wp-database-backup-plugin/
999863	999858	https://www.wordfence.com/blog/2019/05/privilege-escalation-flaw-present-in-slick-popup-plugin/

署名更新バージョン **92**

October 25, 2023

2022-09-22 の週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 92 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998884	CVE-2022-38130	2.4.1 より前の WEB-MISC Keysight SMS-任意のファイルアップロードの脆弱性により SQL インジェクションが可能になる (CVE-2022-38130)
998885	CVE-2022-35741	4.16.1.1 より前の WEB-MISC Apache クラウドスタック-SAMLResponse による XML 外部エンティティインジェクションの脆弱性 (CVE-2022-35741)
998886	CVE-2022-35650	WEB-MISC Moodle 複数バージョン-黒板質問によるパストラバーサルの脆弱性 (CVE-2022-35650)
998887	CVE-2022-32551	WEB-MISC Zoho ManageEngine ServiceDesk MSP 10604 以前-/WEB-INF による認証されていない情報開示 (CVE-2022-32551)
998888	CVE-2022-31675	WEB-MISC VMware vRealize Operations Manager-認証バイパスの脆弱性 (CVE-2022-31675)
998889	CVE-2022-31674	WEB-MISC VMware vRealize Operations Manager-情報開示の脆弱性 (CVE-2022-31674)
998890	CVE-2022-31656	WEB-MISC VMware Workspace ONE Access-認証バイパスの脆弱性 (CVE-2022-31656)
998891	CVE-2022-31474	8.7.5 以前の WEB-WORDPRESS BackupBuddy プラグイン-backupbuddy_local_download による情報開示 (CVE-2022-31474)
998892	CVE-2022-31137、CVE-2022-31126	6.1.1.0 より前の WEB-MISC Roxy-WI-複数のコマンドインジェクションの脆弱性 (CVE-2022-31137、CVE-2022-31126)

署名ルール	**CVE ID**	説明
998893	CVE-2022-28731	WEB-MISC 2.11.3 より前の Apache JSPWiki-サーバー側リクエスト偽造の脆弱性 (CVE-2022-28731)
998894	CVE-2022-2551	1.4.7.1 より前の WEB WORDPRESS デュプリケータプラグイン-認証されていないバックアップダウンロードの脆弱性 (CVE-2022-2551)
998895	CVE-2022-2546	7.63 以前の WEB WORDPRESS オールインワン WP 移行プラグイン-ai1wm_export 経由で XSS の脆弱性を反映 (CVE-2022-2546)
998896	CVE-2022-2546	7.63 以前の WEB WORDPRESS オールインワン WP 移行プラグイン-ai1wm_import 経由で XSS の脆弱性を反映 (CVE-2022-2546)
998897	CVE-2022-24948	ウェブその他 2.11.2 より前の Apache jspWiki-XSS 脆弱性 (CVE-2022-24948)
998898	CVE-2022-2139	WEB-MISC 5.7.04.6469 より前のアドバンテック iView-menUservlet URI とページを介したパストラバーサルの脆弱性 (CVE-2022-2139)
998899	CVE-2022-2139	WEB-MISC 5.7.04.6469 より前のアドバンテック iView-CommandServlet URI とページを介したパストラバーサルの脆弱性 (CVE-2022-2139)
998900	CVE-2022-2139	WEB-MISC 5.7.04.6469 より前のアドバンテック iView-CommandServlet URI とファイル名によるパストラバーサルの脆弱性 (CVE-2022-2139)

署名ルール	**CVE ID**	説明
998901	CVE-2022-2139	WEB-MISC Advantech iView 5.7.04.6469 より前-NetworkServlet URI とファイル名によるパストラバーサルの脆弱性 (CVE-2022-2139)
998902	CVE-2022-0817	3.7.1 より前の WEB-WORDPRESS badgeOS プラグイン-獲得アチーブメントと除外による SQLi の脆弱性 (CVE-2022-0817)
998903	CVE-2022-0817	3.7.1 より前の WEB-WORDPRESS badgeOS プラグイン-獲得アチーブメントとインクルードによる SQLi の脆弱性 (CVE-2022-0817)
998904	CVE-2022-0817	3.7.1 以前の WEB-WORDPRESS badgeOS プラグイン-獲得した成果と注文による SQLi の脆弱性 (CVE-2022-0817)
998905	CVE-2022-0817	3.7.1 以前の WEB-WORDPRESS badgeOS プラグイン-獲得実績と注文による SQLi の脆弱性 (CVE-2022-0817)
998906	CVE-2022-0817	3.7.1 より前の WEB-WORDPRESS badgeOS プラグイン-獲得アチーブメントとオフセットによる SQLi の脆弱性 (CVE-2022-0817)
998907	CVE-2022-0817	3.7.1 より前の WEB-WORDPRESS badgeOS プラグイン-獲得実績と制限による SQLi の脆弱性 (CVE-2022-0817)
998908	CVE-2018-20062、CVE-2019-9082	ウェブその他 5.1.32 より前の ThinkPHP 5.x-認証されていないリモートコード実行の脆弱性 (CVE-2018-20062、CVE-2019-9082)

署名更新バージョン **91**

October 25, 2023

2022-08-23 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 91 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998909	CVE-2022-38129	2.4.1 より前の WEB-MISC キーサイト SMS-パストラバーサルの脆弱性により RCE が許される (CVE-2022-38129)
998910	CVE-2022-37042、CVE-2022-27925	WEB-MISC Zimbra コラボレーションスイート-MailboxImportServlet 複数の脆弱性（CVE-2022-37042、CVE-2022-27925）
998911	CVE-2022-36446	WEB-MISC Webmin 複数バージョン-HTML インジェクションとリモートコード実行の脆弱性 (CVE-2022-36446)
998912	CVE-2022-35405	WEB-MISC Zoho ManageEngine パスワードマネージャープロ 12101 より前-Java デシリアライゼーションの脆弱性 (CVE-2022-35405)

署名ルール	**CVE ID**	説明
998913	CVE-2022-34872	WEB-MISC Centreon 21.10.7 より前-vhidden 経由の SQL インジェクションの脆弱性 (CVE-2022-34872)
998914	CVE-2022-34872	21.10.7 より前の WEB-MISC Centreon-rpn_function を介した SQL インジェクションの脆弱性 (CVE-2022-34872)
998915	CVE-2022-34872	21.10.7 より前の WEB-MISC Centreon-unit_name を介した SQL インジェクションの脆弱性 (CVE-2022-34872)
998916	CVE-2022-34872	WEB-MISC Centreon 21.10.7 より前-警告による SQL インジェクションの脆弱性（CVE-2022-34872）
998917	CVE-2022-34872	WEB-MISC Centreon 21.10.7 より前-クリティカル経由の SQL インジェクションの脆弱性 (CVE-2022-34872)
998918	CVE-2022-34872	21.10.7 より前の WEB-MISC Centreon-def_type を介した SQL インジェクションの脆弱性 (CVE-2022-34872)
998919	CVE-2022-31813	WEB-MISC Apache HTTP サーバ最大 2.4.53-mod_proxy X-Forwarded-* ヘッダ削除の脆弱性（CVE-2022-31813）
998920	CVE-2022-31125	WEB-MISC Roxy-WI 6.1.1.0 より前-alert_consumer を介した認証バイパスの脆弱性 (CVE-2022-31125)
998921	CVE-2022-31101	WEB-MISC PrestaShop 2.1.1 より前のブロックウィッシュリスト-SQL インジェクションの脆弱性 (CVE-2022-31101)

署名ルール	**CVE ID**	説明
998922	CVE-2022-26137	WEB-MISC アトラシアン製品の複数バージョン-クロスオリジンリソース共有バイパスの脆弱性 (CVE-2022-26137)
998923	CVE-2022-24299	WEB-MISC pfSense CE 2.6.0 より前-vpn_openvpn_client.php を介したリモートコード実行の脆弱性 (CVE-2022-24299)
998924	CVE-2022-24299	WEB-MISC pfSense CE 2.6.0 より前-vpn_openvpn_server.php を介したリモートコード実行の脆弱性 (CVE-2022-24299)
998925	CVE-2022-0817	3.7.1 より前の WEB-WORDPRESS BadgeOS プラグイン-取得実績とユーザー ID を介した SQL インジェクションの脆弱性（CVE-2022-0817）
998926	CVE-2021-36749	WEB-MISC Apache ドルイド-任意のローカルファイル漏えいの脆弱性 (CVE-2021-36749)
998927	CVE-2021-26919	WEB-MISC Apache Druid 0.20.2 より前-autoDeserialize=True による信頼できないデシリアライゼーションの脆弱性（CVE-2021-26919）
998928	CVE-2021-26919	WEB-MISC Apache ドルイド 0.20.2 より前-DetectCustomCollations=True による信頼できないデシリアライゼーションの脆弱性 (CVE-2021-26919)

署名更新バージョン **90**

October 25, 2023

2022-07-30 週に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 90 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998929	CVE-2022-34871	WEB-MISC Centreon 21.10.6 より前-SQL インジェクションの脆弱性（CVE-2022-34871）
998930	CVE-2022-29846	WEB-MISC 進行中 Ipswitch WhatsUp Gold-情報漏えいの脆弱性（CVE-2022-29846）
998931	CVE-2022-29845	WEB-MISC 進行中 Ipswitch WhatsUp Gold-パストラバーサルの脆弱性（CVE-2022-29845）
998932	CVE-2022-28055	WEB-MISC FusionPBX 5.0.1 より前-リモートでコードが実行される脆弱性（CVE-2022-28055）
998933	CVE-2022-26138	WEB-MISC Confluence アプリに関するアトラシアンの質問-REST API を介したハードコードされた認証情報の脆弱性（CVE-2022-26138）
998934	CVE-2022-26138	WEB-MISC Confluence アプリに関するアトラシアンの質問-ログインフォームによるハードコードされた認証情報の脆弱性（CVE-2022-26138）
998935	CVE-2022-26135	WEB-MISC Jira サーバーとデータセンター-モバイルプラグインのサーバー側リクエストフォージェリの脆弱性（CVE-2022-26135）

署名ルール	**CVE ID**	説明
998936	CVE-2022-21445	WEB-MISC Oracle OBIEE ADF Faces-信頼できないデータのデシリアライズの脆弱性 (CVE-2022-21445)
998937	CVE-2022-2143	WEB-MISC アドバンテック iView 5.7.04.6469 より前-ネットワーク経由の RCE の脆弱性サーブレット URI と fwfilename (CVE-2022-2143)
998938	CVE-2022-2143	WEB-MISC アドバンテック iView 5.7.04.6469 より前-コマンドサーブレット URI および fwfilename による RCE の脆弱性 (CVE-2022-2143)
998939	CVE-2022-2143	WEB-MISC アドバンテック iView 5.7.04.6469 より前-ネットワーク経由の RCE の脆弱性サーブレット URI と backup_filename (CVE-2022-2143)
998940	CVE-2022-2143	WEB-MISC アドバンテック iView 5.7.04.6469 より前-コマンドサーブレット URI と backup_filename による RCE の脆弱性 (CVE-2022-2143)
998941	CVE-2022-2099	6.6.0 より前の WEB-WORDPRESS WooCommerce プラグイン-支払いゲートウェイ HTML インジェクションの脆弱性（CVE-2022-2099）

署名更新バージョン **89**

October 25, 2023

2022-07-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 89 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998942	CVE-2022-32532	WEB-MISC Apache Shiro 1.9.1 より前-RegexRequestMatcher ラインフィードによる脆弱性のバイパス (CVE-2022-32532)
998943	CVE-2022-32532	WEB-MISC Apache Shiro 1.9.1 より前-RegexRequestMatcher キャリッジリターンによる脆弱性のバイパス (CVE-2022-32532)
998944	CVE-2022-30157	WEB-MISC Microsoft SharePoint-信頼できないデータの逆シリアル化による RCE の脆弱性（CVE-2022-30157）
998945	CVE-2022-29847	WEB-MISC 進行中 Ipswitch WhatsUp Gold-認証されていないサーバー側リクエストフォージェリの脆弱性（CVE-2022-29847）
998946	CVE-2022-29535	WEB-MISC Zoho ManageEngine OPManager 複数のバージョン-ビュー経由の SQL インジェクションの脆弱性 (CVE-2022-29535)
998947	CVE-2022-29535	WEB-MISC Zoho ManageEngine OpManager 複数のバージョン-カテゴリを介した SQL インジェクションの脆弱性（CVE-2022-29535）

署名ルール	**CVE ID**	説明
998948	CVE-2022-28219	WEB-MISC Zoho ManageEngine ADAudit Plus 7060 より前-リモートでコードが実行される脆弱性 (CVE-2022-28219)
998949	CVE-2022-28219	WEB-MISC Zoho ManageEngine ADAudit Plus 7060 より前-タスクによる XXE インジェクションの脆弱性新しいコンテンツ (CVE-2022-28219)
998950	CVE-2022-28219	WEB-MISC Zoho ManageEngine ADAudit Plus 7060 より前-タスクコンテンツを介した XXE インジェクションの脆弱性 (CVE-2022-28219)
998951	CVE-2022-23642	WEB-MISC ソースグラフ 3.37 より前-gitserver サービスのリモートコード実行の脆弱性 (CVE-2022-23642)
998952	CVE-2022-23206	WEB-MISC Apache トラフィック制御 5.1.6 および 6.1.0 より前のトラフィック操作-SSRF 脆弱性 (CVE-2022-23206)
998953	CVE-2022-1609	WEB-WORDPRESS Weblizar 9.9.7 より前の学校管理プロプラグイン-リモートでコードが実行される脆弱性 (CVE-2022-1609)
998954	CVE-2022-1209	WEB-WORDPRESS ワードプレスプラグインの最終メンバー 2.3.2 より前-オープンリダイレクトの脆弱性 (CVE-2022-1209)
998955	CVE-2021-46360	WEB-MISC Composr-CMS-リモートでコードが実行される脆弱性 (CVE-2021-46360)
998956	CVE-2021-43350	WEB-MISC Apache トラフィック制御トラフィック操作 5.1.4 および 6.0.1 より前-LDAP インジェクションの脆弱性 (CVE-2021-43350)

署名ルール	**CVE ID**	説明
998957	CVE-2017-9248	WEB-MISC R2 2017 SP1 より前の ASP.NET AJAX 用の Telerik UI-暗号化キー開示の脆弱性 (CVE-2017-9248)

署名更新バージョン **88**

October 25, 2023

2022-06-16 週に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 88 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998958	CVE-2022-28810	WEB-MISC Zoho ManageEngine AdSelfService 6122 より前-ロック解除スクリプトによる OS コマンドインジェクションの脆弱性 (CVE-2022-28810)

署名ルール	**CVE ID**	説明
998959	CVE-2022-28810	WEB-MISC Zoho ManageEngine 6122 より前の AdSelfService-リセットスクリプトによる OS コマンドインジェクションの脆弱性 (CVE-2022-28810)
998960	CVE-2022-25237	WEB-MISC Bonita ウェブ 7.14.0 より前-i18ntranslation/../による認証バイパスの脆弱性 (CVE-2022-25237)
998961	CVE-2022-25237	WEB-MISC Bonita ウェブ 7.14.0 より前-認証バイパスの脆弱性経由; i18ntranslation (CVE-2022-25237)
998962	CVE-2022-0540	WEB-MISC アトラシアン Jira サーバーとデータセンター-Jira Seraph 認証バイパスの脆弱性 (CVE-2022-0540)
998963	CVE-2021-44548	WEB-MISC Apache Solr 8.11.1 より前-データインポートハンドラー SMB 攻撃の脆弱性 (CVE-2021-44548)

署名更新バージョン **87**

October 25, 2023

2022-06-07 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 87 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998964	CVE-2022-30525	WEB-MISC Zyxel ファイアウォール複数バージョン-SetwanPortST における認証されていない OS コマンドインジェクションの脆弱性 (CVE-2022-30525)
998965	CVE-2022-29108	WEB-MISC Microsoft SharePoint-信頼できないデータのデシリアライズによる RCE の脆弱性（CVE-2022-29108）
998966	CVE-2022-26134	WEB-MISC アトラシアン Confluence 複数バージョン-認証されていない OGNL インジェクションの脆弱性（CVE-2022-26134）
998967	CVE-2022-26019	WEB-MISC pfSense CE < 2.6.0-services_ntpd_gps.php および gpsport を介したリモートコード実行の脆弱性 (CVE-2022-26019)
998968	CVE-2022-26019	WEB-MISC pfSense CE < 2.6.0-services_ntpd.php および gpsport を介したリモートコード実行の脆弱性（CVE-2022-26019）
998969	CVE-2022-24288	WEB-MISC Apache エアフロー 2.2.3 まで-DAG 例 my_param を介したリモートコード実行の脆弱性 (CVE-2022-24288)
998970	CVE-2022-24288	WEB-MISC Apache エアフロー 2.2.3 まで-DAG 例 foo または miff を介したリモートコード実行の脆弱性（CVE-2022-24288）

署名ルール	**CVE ID**	説明
998971	CVE-2022-22978	WEB-MISC スプリングセキュリティ 5.5.6 および 5.6.3 まで-RegexRequestMatcher ラインフィードによる脆弱性のバイパス (CVE-2022-22978)
998972	CVE-2022-22978	WEB-MISC スプリングセキュリティ 5.5.6 および 5.6.3 まで-RegexRequestMatcher キャリッジリターンによる脆弱性のバイパス（CVE-2022-22978）
998973	CVE-2022-22957	WEB-MISC VMware 複数製品-リモートでコードが実行される脆弱性 (CVE-2022-22957)
998974	CVE-2021-45232	WEB-MISC 2.10.1 より前の Apache APIIX ダッシュボード-エクスポートによる認証バイパスの脆弱性（CVE-2021-45232）
998975	CVE-2021-45232	WEB-MISC 2.10.1 より前の Apache APIIX ダッシュボード-インポートによる認証バイパスの脆弱性 (CVE-2021-45232)
998976	CVE-2021-41739	WEB-MISC アーティカプロキシ-cyrus.events.php を介した OS コマンドインジェクションの脆弱性 (CVE-2021-41739)
998977	CVE-2021-37927	WEB-MISC 7111 より前の ManageEngine アドマネージャープラス-認証バイパスの脆弱性 (CVE-2021-37927)
998978	CVE-2021-36356	VSM サーバ経由の WEB-MISC クレイマー-writeBrowseFilePathAjax での認証されていないリモートコード実行の脆弱性 (CVE-2021-36356)
998979	CVE-2021-25094	WEB-WORDPRESS プラグイン 3.3.12 より前のタツビルダー-リモートでコードが実行される脆弱性 (CVE-2021-25094)

署名更新バージョン **86**

October 25, 2023

2022-05-20 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 86 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998980	CVE-2022-30525	WEB-MISC Zyxel ファイアウォール複数バージョン-SetwanPortST における認証されていない OS コマンドインジェクションの脆弱性 (CVE-2022-30525)
998981	CVE-2021-25094	WEB-WORDPRESS プラグイン 3.3.12 より前のタツビルダー-リモートでコードが実行される脆弱性 (CVE-2021-25094)

署名更新バージョン **85**

October 25, 2023

2022-05-13 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 85 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998982	CVE-2022-26352	WEB-MISC dotCMS-PUT を介した任意のファイルアップロードの脆弱性（CVE-2022-26352）
998983	CVE-2022-26352	WEB-MISC dotCMS-POST を介した任意のファイルアップロードの脆弱性（CVE-2022-26352）
998984	CVE-2022-1388	WEB-MISC F5 BIG-IP-iControl REST 認証バイパスの脆弱性（CVE-2022-1388）
998985	CVE-2022-1162	WEB-MISC Gitlab CE/EE 複数バージョン-ハードコードされた資格情報の脆弱性（CVE-2022-1162）
998986	CVE-2022-0888	WEB-WORDPRESS プラグイン忍者フォームファイルアップロード 3.3.1 以前-任意のファイルアップロードの脆弱性（CVE-2022-0888）
998987	CVE-2021-35244	WEB-MISC 2020.2.6 より前の SolarWinds Orion HF3-WriteToFile アクションによる任意のファイルアップロードの脆弱性（CVE-2021-35244）

署名更新バージョン **84**

October 25, 2023

2022-05-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 84 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998988	CVE-2022-26986	WEB-MISC 1.4.3 より前の ImpressCMS-mimetypeid を介した SQL インジェクションの脆弱性 (CVE-2022-26986)
998989	CVE-2022-24112	WEB-MISC Apache APIIX バッチリクエストプラグイン-IP 制限バイパスの脆弱性（CVE-2022-24112）
998990	CVE-2021-37558	20.04.14、20.10.8、および 21.04.2 より前の WEB-MISC Centreon-service_description を介した SQL インジェクションの脆弱性（CVE-2021-37558）
998991	CVE-2021-37558	20.04.14、20.10.8、21.04.2 より前の WEB-MISC Centreon-ホスト名を介した SQL インジェクションの脆弱性（CVE-2021-37558）

署名ルール	**CVE ID**	説明
998992	CVE-2021-22056	WEB-MISC VMware Workspace ONE Access および ID マネージャー-サーバー側リクエストフォージェリの脆弱性（CVE-2021-22056）

署名更新バージョン **83**

October 25, 2023

2022-05-04 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 83 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998993	CVE-2022-29464	WEB-MISC WSO2 複数製品-無制限のファイルアップロードの脆弱性（CVE-2022-29464）
998994	CVE-2022-22954	WEB-MISC VMware Workspace ONE Access および ID マネージャー-デバイスタイプを介したリモートコード実行の脆弱性（CVE-2022-22954）

署名ルール	**CVE ID**	説明
998995	CVE-2022-22954	WEB-MISC VMware Workspace ONE アクセスおよびアイデンティティマネージャー-deviceUDID を介したリモートコード実行の脆弱性（CVE-2022-22954）
998996	CVE-2022-1329	WEB-WORDPRESS ワードプレスエレメンター 3.6.3 より前のウェブサイトビルダー-不正な AJAX アクションの脆弱性（CVE-2022-1329）

署名更新バージョン **82**

October 25, 2023

2022-04-23 週に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 82 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998997	CVE-2022-27924	WEB-MISC Zimbra コラボレーションジュール-キャッシュポイズニングの脆弱性（CVE-2022-27924）

署名ルール	**CVE ID**	説明
998998	CVE-2022-21907	WEB-MISC Microsoft HTTP プロトコルスタック-リモートでコードが実行される脆弱性 (CVE-2022-21907)
998999	CVE-2021-37930	WEB-MISC 7111 より前の ManageEngine adManager プラス-SM_DomainName を介した任意のファイルアップロードの脆弱性 (CVE-2021-37930)
999000	CVE-2021-37930	WEB-MISC 7111 より前の ManageEngine adManager プラス-SM_OperationID を介した任意のファイルアップロードの脆弱性 (CVE-2021-37930)

署名更新バージョン **81**

October 25, 2023

2022-04-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 81 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999001	CVE-2022-0479	4.1.1 より前の WEB-WORDPRESS ポップアップビルダープラグイン-SQL インジェクションの脆弱性（CVE-2022-0479）
999002	CVE-2021-36393	WEB-MISC Moodle 3.11.1 より前-SQL インジェクションの脆弱性（CVE-2021-36393）
999003	CVE-2021-26599	WEB-MISC ImpressCMS 1.4.3 より前-SQL インジェクションの脆弱性（CVE-2021-26599）

署名更新バージョン **80**

October 25, 2023

2022-04-04 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 80 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999004	CVE-2022-22965	WEB-MISC Spring4Shell スプリングコアフレームワーク-RCE 脆弱性（CVE-2022-22965）

署名更新バージョン **79**

October 25, 2023

2022-03-29 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。セキュリティに脆弱な攻撃からアプライアンスを保護するために、これらのシグニチャルールをダウンロードして設定できます。

署名バージョン

シグネチャバージョン 79 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
18959 (更新されたルール)	CVE-2022-22965	WEB-MISC VMware Spring4Shell、SpringSource Spring Framework クラス.classloader RCE 試行
999005	CVE-2022-22963	WEB-MISC Spring クラウド関数-コードインジェクションの脆弱性 (CVE-2022-22963)

署名更新バージョン **78**

October 25, 2023

2022-03-29 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。セキュリティに脆弱な攻撃からアプライアンスを保護するために、これらのシグニチャルールをダウンロードして設定できます。

署名バージョン

シグネチャバージョン 78 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999006		WEB-MISC Zabbix 複数バージョン-items.php を介したリモートコード実行の脆弱性
999007	CVE-2022-24266	WEB-MISC Cuppa CMS v1.0-order_orientation 経由の SQL インジェクションの脆弱性 (CVE-2022-24266)
999008	CVE-2022-24266	WEB-MISC Cuppa CMS v1.0-order_by 経由の SQL インジェクションの脆弱性 (CVE-2022-24266)
999009	CVE-2022-22005	WEB-MISC Microsoft SharePoint-信頼できないデータのデシリアライゼーションによる RCE の脆弱性（CVE-2022-22005）
999010	CVE-2022-21705	WEB-MISC OctoberCMS ビルド 474 および v1.1.10 より前-リモートでコードが実行される脆弱性 (CVE-2022-21705)
999011	CVE-2022-0557	1.2.11 より前の WEB-MISC マイクロウェーバー-リモートでコードが実行される脆弱性（CVE-2022-0557）
999012	CVE-2022-0513	13.1.5 より前の WEB-WORDPRESS WP 統計プラグイン-ブラインド SQL インジェクションの脆弱性（CVE-2022-0513）

署名ルール	**CVE ID**	説明
999013	CVE-2022-0332	WEB-MISC ムードル 3.11.0 から 3.11.4-H5P アクティビティ SQL インジェクションの脆弱性 (CVE-2022-0332)
999014	CVE-2021-46088	WEB-MISC Zabbix 複数バージョン-リモートでコードが実行される脆弱性（CVE-2021-46088）
999015	CVE-2021-43789	WEB-MISC PrestaShop 1.7.8.2 より前-並べ替え順序による SQL インジェクションの脆弱性 (CVE-2021-43789)
999016	CVE-2021-43789	WEB-MISC PrestaShop 1.7.8.2 より前-orderBy による SQL インジェクションの脆弱性 (CVE-2021-43789)
999017	CVE-2021-43408	WEB-WORDPRESS 1.1.9 より前の重複投稿プラグイン-SQL インジェクションの脆弱性 (CVE-2021-43408)
999018	CVE-2021-43319	WEB-MISC Zoho ManageEngine NCM 125488 より前-OS コマンドインジェクションの脆弱性 (CVE-2021-43319)
999019	CVE-2021-41282	WEB-MISC pfSense 2.5.2: リモートでコードが実行される脆弱性 (CVE-2021-41282)
999020	CVE-2021-39115、CVE-2021-43947	WEB-MISC アトラシアン Jira サーバーおよびデータセンター-サーバー側テンプレートインジェクションの脆弱性（CVE-2021-39115、CVE-2021-43947）
999021	CVE-2021-38452	WEB-MISC Moxa MxView 3.2.2 より前のネットワーク管理-パストラバーサルの脆弱性 (CVE-2021-38452)

署名ルール	**CVE ID**	説明
999022	CVE-2021-37918	WEB-MISC Zoho ManageEngine AdManager Plus 7111 より前-ドメイン名を介したパストラバーサルの脆弱性（CVE-2021-37918）
999023	CVE-2021-37918	Web-MISC Zoho ManageEngine AdManager Plus 7111 より前-BM_OperationID を介したパストラバーサルの脆弱性（CVE-2021-37918）
999024	CVE-2021-37918	Web-MISC Zoho ManageEngine AdManager Plus 7111 より前-任意のファイルアップロードによる RCE の脆弱性（CVE-2021-37918）
999025	CVE-2021-32649	WEB-MISC OctoberCMS ビルド 473 および v1.1.6 より前-Twig を介したリモートコード実行の脆弱性（CVE-2021-32649）
999026	CVE-2021-32648	WEB-MISC OctoberCMS ビルド 472 および v1.1.5 より前のバージョン-パスワードリセットの脆弱性（CVE-2021-32648）
999027	CVE-2021-32099, CVE-2020-26518	WEB-MISC 記事 743 より前のパンドラ-chart_generator を介した SQL インジェクションの脆弱性（CVE-2021-32099、CVE-2020-26518）
999028	CVE-2021-32098	WEB-MISC 記事 743 より前のパンドラ-progressbubble を介した Phar デシリアライゼーションの脆弱性 (CVE-2021-32098)
999029	CVE-2021-32098	WEB-MISC 記事 743 より前のパンドラ-プログレスバー経由の Phar デシリアライゼーションの脆弱性（CVE-2021-32098）
999030	CVE-2021-30149	WEB-MISC コンポーザ 10.0.36：リモートでコードが実行される脆弱性（CVE-2021-30149）

署名ルール	**CVE ID**	説明
999031	CVE-2021-25114	WEB-WORDPRESS 有料会員プロプラグイン 2.6.7 より前-rest_route と discount_code による SQLi の脆弱性 (CVE-2021-25114)
999032	CVE-2021-25114	WEB-WORDPRESS 有料会員プロプラグイン 2.6.7 より前-wp-json と discount_code 経由の SQLi の脆弱性 (CVE-2021-25114)
999033	CVE-2021-21984	WEB-MISC VMware vRealize ビジネス 7.6.0 より前のクラウド 7.x-リモートでコードが実行される脆弱性（CVE-2021-21984）

署名更新バージョン **77**

October 25, 2023

2022-02-25 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 77 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 のプラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999034		WEB-WORDPRESS WordPress 5.9-Json オブジェクトのページの抜粋を介して格納された XSS の脆弱性
999035		WEB-WORDPRESS WordPress 5.9-フォーム内のページの抜粋を介して保存された XSS の脆弱性
999036		WEB-WORDPRESS WordPress 5.9-post.php 経由で保存された XSS の脆弱性
999037		WEB-WORDPRESS WordPress 5.9-Json オブジェクトにポスト抜粋を介して格納された XSS の脆弱性
999038		WEB-WORDPRESS WordPress 5.9-フォーム内のポストの抜粋を介して保存された XSS の脆弱性
999039		フォームフィールド値による WEB-MISC パストラバーサルの脆弱性
999040		URI 経由の WEB-MISC パストラバーサルの脆弱性
999041	CVE-2022-23221	2.1.210 より前の WEB-MISC H2 コンソール：test.do を介したリモートコード実行の脆弱性 (CVE-2022-23221)
999042	CVE-2022-23221	2.1.210 より前の WEB-MISC H2 コンソール：login.do を介したリモートコード実行の脆弱性 (CVE-2022-23221)
999043	CVE-2022-21662	5.8.3 より前の WEB-WORDPRESS WordPress：ストアドクロスサイトスクリプティングの脆弱性 (CVE-2022-21662)
999044	CVE-2022-0320	WEB-WORDPRESS 5.0.5 より前の Elementor プラグインに不可欠なアドオン-LFI eael_product_gallery (CVE-2022-0320)

署名ルール	**CVE ID**	説明
999045	CVE-2022-0320	WEB-WORDPRESS 5.0.5 より前の Elementor プラグインに必須のアドオン-woo_product_pagination_product 経由の LFI (CVE-2022-0320)
999046	CVE-2022-0320	WEB-WORDPRESS 5.0.5 より前の Elementor プラグインに必須のアドオン-load_more 経由の LFI (CVE-2022-0320)

署名更新バージョン **76**

October 25, 2023

2022-02-20 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 76 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0 プラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999047	CVE-2022-23863	4.5.30 より前の WEB-MISC FusionPBX-fax_page_size を介した OS コマンドインジェクション (CVE-2021-43406)

署名ルール	**CVE ID**	説明
999048	CVE-2021-44515	WEB-MISC JetBrains TeamCity: エージェントプッシュによるリモートコード実行の脆弱性 (CVE-2021-43193)
999049	CVE-2021-43406	5.1.5 より前の WEB-MISC ゴーアヘッド：CGI 環境変数インジェクションの脆弱性（CVE-2021-42342）
999050	CVE-2021-43193	WEB-MISC SonicWall セキュアモバイルアクセス-リモートでコードが実行される脆弱性 (CVE-2021-20045)
999051	CVE-2021-42342	5.1.5 より前の WEB-MISC ゴーアヘッド：CGI 環境変数インジェクションの脆弱性（CVE-2021-42342）
999052	CVE-2021-20045	WEB-MISC SonicWall セキュアモバイルアクセス-リモートでコードが実行される脆弱性 (CVE-2021-20045)
999053	CVE-2021-20044	WEB-MISC SonicWall セキュアモバイルアクセス-コマンドインジェクションの脆弱性 (CVE-2021-20044)
999054		Web-WORDPRESS AdSanity プラグイン-HTML5 ファイルアップロードによるリモートコード実行の脆弱性

署名更新バージョン **75**

October 25, 2023

2022-01-20 の週に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 75 は、NetScaler 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0 プラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999055	CVE-2021-44224	WEB-MISC Apache HTTP サーバー-フォワードプロキシとリバースプロキシを介した不正な形式の UDS の脆弱性（CVE-2021-44224）
999056	CVE-2021-43815	WEB-MISC Apache Grafana-TestData DB データソースパストラバーサルの脆弱性（CVE-2021-43815）
999057	CVE-2021-43813	WEB-MISC Apache Grafana：マークダウンによるパストラバーサルの脆弱性（CVE-2021-43813）
999058	CVE-2021-43405	4.5.30 より前の WEB-MISC FusionPBX-fax_extension を介した OS コマンドインジェクション（CVE-2021-43405）
999059	CVE-2021-42392	2.0.206 より前の WEB-MISC H2 コンソール：リモートでコードが実行される脆弱性（CVE-2021-42392）
999060	CVE-2021-42362	5.3.3 より前の WEB-WORDPRESS ポピュラーな投稿プラグイン：任意のファイルアップロードの脆弱性（CVE-2021-42362）
999061	CVE-2021-42129	6.3.3 より前の WEB-MISC Ivanti Avalanche：TxtuPass を介した OS コマンドインジェクションの脆弱性（CVE-2021-42129）

署名ルール	**CVE ID**	説明
999062	CVE-2021-42129	6.3.3 より前の WEB-MISC Ivanti Avalanche：TxtUname を介した OS コマンドインジェクションの脆弱性（CVE-2021-42129）
999063	CVE-2021-42129	6.3.3 より前の WEB-MISC Ivanti Avalanche：TxTuncPath を介した OS コマンドインジェクションの脆弱性（CVE-2021-42129）
999064	CVE-2021-40345	5.8.6 より前の WEB-MISC Nagios XI：悪意を持って細工された ZIP ファイルを介した OS コマンドインジェクションの脆弱性（CVE-2021-40345）
999065	CVE-2021-37928	WEB-MISC Zoho ManageEngine アドマネージャープラス 7110 より前-無制限のファイルアップロードの脆弱性（CVE-2021-37928）
999066	CVE-2021-25037	4.1.5.3 より前の WEB-WORDPRESS All In One SEO Plugin-オブジェクト REST API と rest_route を介した SQL インジェクションの脆弱性
999067	CVE-2021-25037	4.1.5.3 より前の WEB-WORDPRESS All In One SEO Plugin-オブジェクト REST API を介した SQL インジェクションの脆弱性
999068	CVE-2021-25036	4.1.5.3 より前の WEB-WORDPRESS All In One SEO Plugin-REST API と rest_route を介した権限昇格の脆弱性
999069	CVE-2021-25036	4.1.5.3 より前の WEB-WORDPRESS All In One SEO Plugin-REST API を介した権限昇格の脆弱性

署名ルール	**CVE ID**	説明
999070	CVE-2021-21917	2.4.17 より前の WEB-MISC アドバンテック R-SeeNet-ord 経由の SQL インジェクションの脆弱性 (CVE-2021-21917)
999071	CVE-2021-20040	WEB-MISC SonicWall セキュアモバイルアクセス-任意のファイル書き込みの脆弱性（CVE-2021-20040）
999072	CVE-2021-20039	WEB-MISC SonicWall セキュアモバイルアクセス-コマンドインジェクションの脆弱性（CVE-2021-20039）

署名更新バージョン **74**

October 25, 2023

2021-12-21 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999073	CVE-2021-44077	WEB-MISC Zoho ManageEngine サービスデスクプラス 11306 より前-インポート技術者による事前認証 RCE の脆弱性（CVE-2021-44077）
999074	CVE-2021-43798	WEB-MISC Apache Grafana 8.0.0 8.3.0 まで：パストラバーサルの脆弱性（CVE-2021-43798）
999075	CVE-2021-35216	2020.2.6 より前の WEB-MISC SolarWinds オリオン: EditTopXX.aspx を介したデシリアライゼーションの脆弱性 (CVE-2021-35216)
999076	CVE-2021-34993	WEB-MISC Commvault CommCell-CvSearchService 認証バイパスの脆弱性 (CVE-2021-34993)

署名更新バージョン **73**

October 25, 2023

2021 年 12 月 13 日に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、更新されたシグニチャルール、CVE ID、およびその説明の一覧を示します。

> 注:
>
> 以下のシグニチャルール（999077、999078、999079、999080）は、両方の CVE（CVE-2021-44228 および CVE-2021-45046）に対応しています。

署名ルール	**CVE ID**	説明
999077	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j：フォームを介したリモートコード実行の脆弱性（CVE-2021-44228、CVE-2021-45046）
999078	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j：本文を介してリモートでコードが実行される脆弱性（CVE-2021-44228、CVE-2021-45046）
999079	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j：ヘッダーを介したリモートコード実行の脆弱性（CVE-2021-44228、CVE-2021-45046）
999080	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j：URL を介したリモートコード実行の脆弱性（CVE-2021-44228、CVE-2021-45046）

署名更新バージョン **72**

October 25, 2023

2021 年 12 月 11 日に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999077	CVE-2021-44228	WEB-MISC Apache Log4j：フォーム経由でリモートでコードが実行される脆弱性（CVE-2021-44228）
999078	CVE-2021-44228	WEB-MISC Apache Log4j：本文を介してリモートでコードが実行される脆弱性（CVE-2021-44228）
999079	CVE-2021-44228	WEB-MISC Apache Log4j：ヘッダーを介したリモートコード実行の脆弱性（CVE-2021-44228）
999080	CVE-2021-44228	WEB-MISC Apache Log4j：URL経由でリモートでコードが実行される脆弱性（CVE-2021-44228）
999081	CVE-2021-42847	WEB-MISC Zoho ManageEngine ADAudit Plus 7006 より前：認証されていない任意のファイル書き込みの脆弱性（CVE-2021-42847）
999082	CVE-2021-42321	WEB-MISC Microsoft Exchange Server-リモートでコードが実行される脆弱性（CVE-2021-42321）
999083	CVE-2021-42258	WEB-MISC BQE BillQuick Web スイート 2021-txTID を介した認証されていない SQL インジェクションの脆弱性（CVE-2021-42258）
999084	CVE-2021-42258	WEB-MISC BQE BillQuick Web スイート 2020-txTID を介した認証されていない SQL インジェクションの脆弱性（CVE-2021-42258）

署名ルール	**CVE ID**	説明
999085	CVE-2021-42258	WEB-MISC BQE BillQuick Web スイート 2019-txTID を介した認証されていない SQL インジェクションの脆弱性（CVE-2021-42258）
999086	CVE-2021-42258	WEB-MISC BQE BillQuick Web スイート 2018-txTID を介した認証されていない SQL インジェクションの脆弱性（CVE-2021-42258）
999087	CVE-2021-42237	WEB-MISC サイトコア 7.5.0 から 8.2.7 へ：リモートでコードが実行される脆弱性（CVE-2021-42237）
999088	CVE-2021-41950	リビジョン 18277 より前の WEB-MISC ResourceSpace 9.6: バリアントを介した非認証パストラバーサルの脆弱性 (CVE-2021-41950)
999089	CVE-2021-41950	リビジョン 18277 より前の WEB-MISC ResourceSpace 9.6: プロバイダーを介した非認証パストラバーサルの脆弱性 (CVE-2021-41950)
999090	CVE-2021-41349	WEB-MISC Microsoft Exchange Server-クロスサイトスクリプティングの脆弱性 (CVE-2021-41349)
999091	CVE-2021-35217	2020.2.6 HF1 より前の WEB-MISC SolarWinds オリオン-WSAsyncExecuteTasks.aspx を介したデシリアライゼーションの脆弱性（CVE-2021-35217）
999092	CVE-2021-34416	WEB-MISC Zoom ミーティングコネクタ 4.6.360.20210325：リモートでコードが実行される脆弱性 (CVE-2021-34416)
999093	CVE-2021-22941	WEB-MISC 5.11.20 より前の Citrix ShareFile ストレージ：不適切なアクセス制御の脆弱性 (CVE-2021-22941)

署名ルール	**CVE ID**	説明
999094	CVE-2020-35136	12.0.4 より前の WEB-MISC Dolibarr: zipfilename_template および bz を介したリモートコード実行の脆弱性（CVE-2020-35136）
999095	CVE-2020-35136	12.0.4 より前の WEB-MISC Dolibarr: zipfilename_template および gz を介したリモートコード実行の脆弱性（CVE-2020-35136）
999096	CVE-2020-2950, CVE-2021-2456	ウェブその他 Oracle BI Publisher -任意のファイルアップロードの脆弱性（CVE-2020-2950、CVE-2021-2456）
999097	CVE-2020-2950, CVE-2021-2456	ウェブその他 Oracle BI Publisher -リモートでコードが実行される脆弱性（CVE-2020-2950、CVE-2021-2456）

署名更新バージョン **71**

October 25, 2023

2021-11-18 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999098	CVE-2021-41765	WEB-MISC リビジョン 18274 より前の ResourceSpace 9.5 および 9.6-SQL インジェクションの脆弱性 (CVE-2021-41765)
999099	CVE-2021-41288	WEB-MISC Zoho マネージエンジン運用マネージャー 125467 ビルド前-GetReportData API による SQL インジェクションの脆弱性 (CVE-2021-41288)
999100	CVE-2021-40493	WEB-MISC Zoho マネージエンジン運用マネージャー 125437 ビルド前-デバイス名による SQL インジェクションの脆弱性 (CVE-2021-40493)
999101	CVE-2021-40493	WEB-MISC Zoho ビルド前 125437 の管理エンジン OpManager-ポーリングオブジェクトによる SQL インジェクションの脆弱性 (CVE-2021-40493)
999102	CVE-2021-40438	WEB-MISC Apache HTTP サーバー-mod_proxy リクエスト転送の脆弱性（CVE-2021-40438）
999103	CVE-2021-39341	WEB-WORDPRESS OptinMonster プラグイン 2.6.4 まで-REST_ROUTE 権限バイパスの脆弱性（CVE-2021-39341）
999104	CVE-2021-39341	WEB-WORDPRESS OptinMonster プラグイン 2.6.4 まで-REST API パーミッションバイパスの脆弱性（CVE-2021-39341）

署名ルール	**CVE ID**	説明
999105	CVE-2021-37344	WEB-MISC Nagios XI スイッチウィザード 2.5.7 より前-ip_address パラメータによるリモートコード実行の脆弱性 (CVE-2021-37344)
999106	CVE-2021-35218	WEB-MISC SolarWinds オリオン 2020.2.6 より前-Chart.ashx によるデシリアライゼーションの脆弱性 (CVE-2021-35218)
999107	CVE-2021-35215	WEB-MISC SolarWinds Orion プラットフォーム 2020.2.6 より前-レポートによるリモートコード実行の脆弱性（CVE-2021-35215）
999108	CVE-2021-35215	WEB-MISC SolarWinds Orion プラットフォーム 2020.2.6 より前-アラートによるリモートコード実行の脆弱性（CVE-2021-35215）
999109	CVE-2021-24889	3.6.4 より前の WEB-WORDPRESS 忍者フォームプラグイン-SQL インジェクションの脆弱性 (CVE-2021-24889)
999110	CVE-2021-24381	WEB-WORDPRESS 忍者フォームプラグイン 3.5.8.2 より前-カスタムクラス名が格納されたクロスサイトスクリプティングの脆弱性 (CVE-2021-24381)
999111	CVE-2021-2401	WEB-MISC Oracle ル BI 発行元-DOMParser XXE モバイル X レポートテンプレートサービス経由の脆弱性 (CVE-2021-2401)
999112	CVE-2021-2401	WEB-MISC Oracle ル BI 発行元-DOMParser XXE モバイルレポートテンプレートサービス経由の脆弱性 (CVE-2021-2401)
999113	CVE-2021-2401	WEB-MISC Oracle BI 発行元-xmlpservice X ReportTemplateService 経由の DOMParser XXE 脆弱性 (CVE-2021-2401)

署名ルール	**CVE ID**	説明
999114	CVE-2021-2401	WEB-MISC Oracle BI 発行元-xmlpservice 経由の DOMParser XXE 脆弱性 ReportTemplateService (CVE-2021-2401)
999115	CVE-2021-2392	WEB-MISC Oracle BI 発行元-任意ファイルのアップロードの脆弱性 (CVE-2021-2392)
999116	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase 分析プロバイダ・サービス-Essbase 経由のリモート・コード実行の脆弱性 (CVE-2021-2244)
999117	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase 分析プロバイダ・サービス-管理者経由のリモート・コード実行の脆弱性 (CVE-2021-2244)
999118	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase アナリティック・プロバイダ・サービス-JAPI 経由のリモート・コード実行の脆弱性 (CVE-2021-2244)
999119	CVE-2021-22205	WEB-MISC GitLab CE/EE-悪意を持って作成された JPEG/TIFF ファイルによるリモートコード実行の脆弱性（CVE-2021-22205）
999120	CVE-2021-22017	WEB-MISC VMware vCenter-rhhtproxy を介したパストラバーサルの脆弱性 (CVE-2021-22017)
999121	CVE-2021-20837	WEB-MISC r.5003 より前の可動タイプ-mt.handler_to_coderef を介したリモートコード実行 (CVE-2021-20837)

署名ルール	**CVE ID**	説明
999122	CVE-2021-20131	WEB-MISC Zoho ManageEngine AdManager ビルド 7115 より前-ファイルのアップロードによるリモートでのコード実行の脆弱性 (CVE-2021-20131)
999123	CVE-2021-20130	WEB-MISC Zoho ManageEngine AdManager ビルド 7115 より前-ファイルのアップロードによるリモートでのコード実行の脆弱性 (CVE-2021-20130)
999124	CVE-2021-20034	WEB-MISC SonicWall セキュアモバイルアクセス-パストラバーサルの脆弱性（CVE-2021-20034）
999125		9.1.1 より前の WEB-WORDPRESS のバディプレスプラグイン-サインアップ REST API と rest_route による情報開示の脆弱性
999126		9.1.1 より前の WEB-WORDPRESS BuddyPress プラグイン-サインアップ REST API による情報開示の脆弱性

署名更新バージョン **70**

October 25, 2023

2021-10-26 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999127	CVE-2021-42013	WEB-MISC Apache HTTP サーバー 2.4.49 および 2.4.50-%%32 経由でのパストラバーサルの脆弱性 (CVE-2021-42013)
999128	CVE-2021-42013	WEB-MISC Apache HTTP サーバー 2.4.49 および 2.4.50-% 2% 経由でのパストラバーサルの脆弱性 (CVE-2021-42013)
999129	CVE-2021-41773	WEB-MISC Apache HTTP サーバ 2.4.49-%2e%2e を介したパストラバーサルの脆弱性 (CVE-2021-41773)
999130	CVE-2021-41773	WEB-MISC Apache HTTP サーバー 2.4.49-.% 2e を介したパストラバーサルの脆弱性 (CVE-2021-41773)
999131	CVE-2021-40539	WEB-MISC Zoho ManageEngine AdSelfService Plus 6.1 ビルド 6114 より前-認証バイパスの脆弱性 (CVE-2021-40539)
999132	CVE-2021-34648	WEB-WORDPRESS 忍者フォームプラグイン 3.5.7 まで-送信メールアクションによる REST_ROUTE 脆弱性 (CVE-2021-34648)
999133	CVE-2021-34648	WEB-WORDPRESS 忍者フォームプラグイン最大 3.5.7-送信メールアクションによる REST API 脆弱性 (CVE-2021-34648)

署名ルール	**CVE ID**	説明
999134	CVE-2021-34647	WEB-WORDPRESS 忍者フォームプラグイン 3.5.7 まで-提出物のエクスポートによる REST_ROUTE 脆弱性（CVE-2021-34647）
999135	CVE-2021-34647	WEB-WORDPRESS 忍者フォームプラグイン 3.5.7 まで-提出物のエクスポートによる REST API の脆弱性（CVE-2021-34647）
999136	CVE-2021-34623	3.1.4 より前の WEB-WORDPRESS ProfilePress プラグイン-eup_cover_image を介した任意のファイルアップロードの脆弱性（CVE-2021-34623）
999137	CVE-2021-34623	3.1.4 より前の WEB-WORDPRESS ProfilePress プラグイン-eup_avatar を介した任意のファイルアップロードの脆弱性（CVE-2021-34623）
999138	CVE-2021-2400	WEB-MISC Oracle ル BI 発行元-モバイル X 経由の SAXParser XXE 脆弱性 ReportTemplateService (CVE-2021-2400)
999139	CVE-2021-2400	WEB-MISC Oracle ル BI 発行元-SaxParser XXE モバイルレポートテンプレートサービス経由の脆弱性 (CVE-2021-2400)
999140	CVE-2021-2400	WEB-MISC Oracle BI 発行元-xmlpservice X ReportTemplateService 経由の SAXParser XXE 脆弱性 (CVE-2021-2400)
999141	CVE-2021-2400	WEB-MISC Oracle BI 発行元-xmlpService 経由の SAXParser XXE 脆弱性 ReportTemplateService (CVE-2021-2400)

署名ルール	**CVE ID**	説明
999142	CVE-2021-21985	WEB-MISC VMware vCenter-vSAN ヘルスチェックプラグインのリモートコード実行の脆弱性（CVE-2021-21985）
999143	CVE-2021-20078	WEB-MISC Zoho ManageEngine opManager 12.5 ビルド 125362 より前-パストラバーサルの脆弱性（CVE-2021-20078）
999144	CVE-2020-29448	WEB-MISC アトラシアン Confluence サーバーとデータセンター-WEB INF を介した情報開示の脆弱性 (CVE-2020-29448)
999145	CVE-2020-29448	WEB-MISC アトラシアン Confluence サーバーとデータセンター-META-INF を介した情報開示の脆弱性 (CVE-2020-29448)
999146	CVE-2020-12442	WEB-MISC Ivanti Avalanche 6.3-osupdate エンドポイントを介した認証されていない SQL インジェクションの脆弱性（CVE-2020-12442）
999147	CVE-2020-12442	WEB-MISC Ivanti Avalanche 6.3-壁エンドポイントを介した認証されていない SQL インジェクションの脆弱性（CVE-2020-12442）
999148		9.1.1 より前の WEB-WORDPRESS バディプレスプラグイン-bp-メンバー招待状を介した SQL インジェクションの脆弱性機能

署名更新バージョン **69**

October 25, 2023

2021-10-09 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999149	CVE-2021-38312	WEB-WORDPRESS グーテンベルクテンプレートライブラリと 4.2.12 より前の Redux フレームワークプラグイン-REST_ROUTE の脆弱性（CVE-2021-38312）
999150	CVE-2021-38312	WEB-WORDPRESS グーテンベルクテンプレートライブラリと 4.2.12 より前の Redux フレームワークプラグイン-REST API の脆弱性（CVE-2021-38312）
999151	CVE-2021-34639	3.1.25 より前の WEB-WORDPRESS ダウンロードマネージャープラグイン-二重拡張機能のアップロードの脆弱性（CVE-2021-34639）
999152	CVE-2021-34621	3.1.3 より前の WEB-WORDPRESS ProfilePress プラグイン-wp_capabilities による権限昇格の脆弱性（CVE-2021-34621）
999153	CVE-2021-32682	WEB-MISC elFinder 2.1.59 より前-名前の変更コマンドによるパストラバーサルの脆弱性（CVE-2021-32682）

署名ルール	**CVE ID**	説明
999154	CVE-2021-32682	WEB-MISC elFinder 2.1.59 より前-中止コマンドによるパストラバーサルの脆弱性（CVE-2021-32682）
999155	CVE-2021-26086	WEB-MISC アトラシアン Jira サーバーとデータセンター-WEB-INF を介した情報開示の脆弱性 (CVE-2021-26086)
999156	CVE-2021-26086	WEB-MISC アトラシアン Jira サーバーとデータセンター-META-INF を介した情報開示の脆弱性 (CVE-2021-26086)
999157	CVE-2021-22005	WEB-MISC VMware vCenter-データアプリを介したファイルアップロードの脆弱性（CVE-2021-22005）
999158	CVE-2021-22005	WEB-MISC VMware vCenter-テレメトリステージログによるファイルアップロードの脆弱性 (CVE-2021-22005)
999159	CVE-2021-22005	WEB-MISC VMware vCenter-テレメトリ製品ログによるファイルアップロードの脆弱性 (CVE-2021-22005)
999160	CVE-2021-20081	WEB-MISC 11.2.0.5 より前の Zoho ManageEngine サービスデスク-リモートでコードが実行される脆弱性（CVE-2021-20081）
999161	CVE-2020-29453	WEB-MISC アトラシアン Jira サーバーとデータセンター-WEB INF を介した情報開示の脆弱性 (CVE-2020-29453)
999162	CVE-2020-29453	WEB-MISC アトラシアン Jira サーバーとデータセンター-META-INF を介した情報開示の脆弱性 (CVE-2020-29453)

署名更新バージョン **68**

October 25, 2023

2021-09-11 週に特定された脆弱性について、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999163	CVE-2021-37556	WEB-MISC Centreon 複数バージョン-終了パラメータを介した SQL インジェクションの脆弱性 (CVE-2021-37556)
999164	CVE-2021-37556	WEB-MISC Centreon 複数バージョン-開始パラメータを介した SQL インジェクションの脆弱性 (CVE-2021-37556)
999165	CVE-2021-37353	WEB-MISC Nagios XI Docker ウィザード 1.1.3 より前-URI スキームなしのホストパラメータ経由の SSRF 脆弱性 (CVE-2021-37353)
999166	CVE-2021-37353	WEB-MISC Nagios XI Docker ウィザード 1.1.3 より前-URI スキームによるホストパラメータ経由の SSRF 脆弱性（CVE-2021-37353）

署名ルール	**CVE ID**	説明
999167	CVE-2021-34638	3.1.25 より前の WEB-WORDPRESS のダウンロードマネージャプラグイン-ディレクトリトラバーサルの脆弱性 (CVE-2021-34638)
999168	CVE-2021-33766	WEB-MISC Microsoft Exchange Server-情報漏えいの脆弱性 (CVE-2021-33766)
999169	CVE-2021-32682	2.1.59 より前の WEB-MISC elFinder-アーカイブ経由のコマンドインジェクションの脆弱性 (CVE-2021-32682)
999170	CVE-2021-26084	WEB-MISC Confluence サーバーとデータセンター-doenterpagevariables 経由の OGNL インジェクションの脆弱性 (CVE-2021-26084)
999171	CVE-2021-26084	WEB-MISC Confluence サーバーとデータセンター-作成ページ入力変数経由の OGNL インジェクションの脆弱性（CVE-2021-26084）
999172	CVE-2021-23394	2.1.59 より前の WEB-MISC elFinder-Phar Makefile 経由のリモートコード実行の脆弱性 (CVE-2021-23394)
999173	CVE-2021-23394	2.1.59 より前の WEB-MISC elFinder-Phar の名前変更によるリモートコード実行の脆弱性 (CVE-2021-23394)
999174	CVE-2021-23394	2.1.59 より前の WEB-MISC elFinder-Phar アップロードによるリモートコード実行の脆弱性 (CVE-2021-23394)

署名ルール	**CVE ID**	説明
999175	CVE-2020-36289	WEB-MISC アトラシアン Jira サーバー-QueryComponentRendererValue による情報開示の脆弱性 (CVE-2020-36289)
999176	CVE-2020-16245	WEB-MISC Advantech iView 5.7.03.6112 より前-FindSummaryCFGDeviceListExport によるパストラバーサルの脆弱性 (CVE-2020-16245)
999177	CVE-2020-16245	WEB-MISC Advantech iView 5.7.03.6112 より前-FindUpdateDeviceListExport によるパストラバーサルの脆弱性 (CVE-2020-16245)
999178	CVE-2020-13774	WEB-MISC Ivanti エンドポイントマネージャ複数バージョン-EditLaunchPadDialog.aspx 経由の RCE の脆弱性 (CVE-2020-13774)
999179	CVE-2020-1147	WEB-MISC Microsoft SharePoint サーバー-カスタムページを介したリモートコード実行の脆弱性 (CVE-2020-1147)
999180	CVE-2020-1147	WEB-MISC Microsoft SharePoint Server-quicklinksdialogform.aspx 経由でリモートでコードが実行される脆弱性 (CVE-2020-1147)
999181	CVE-2020-1147	WEB-MISC Microsoft SharePoint Server-quicklinks.aspx 経由でリモートでコードが実行される脆弱性 (CVE-2020-1147)
999182	CVE-2020-11110	WEB-MISC Apache Grafana 6.7.1 まで-XSS 脆弱性 (CVE-2020-11110)

署名ルール	**CVE ID**	説明
999522	CVE-2020-13379	WEB-MISC Grafana 3.0.1 から 7.0.1-DOS の脆弱性につながる CSRF バイパス（CVE-2020-13379）

署名更新バージョン **67**

October 25, 2023

2021-08-29 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999183	CVE-2021-37557	WEB-MISC Centreon 複数のバージョン-SQL インジェクションの脆弱性（CVE-2021-37557）
999184	CVE-2021-35501	ウェブ-MISC Artica Pandora FMS 7.54 まで-ビジュアルコンソールが保存された XSS の脆弱性（CVE-2021-35501）

署名ルール	**CVE ID**	説明
999185	CVE-2021-35464	WEB-MISC ForgeRock アクセス管理と OpenAM-リモートコード実行の脆弱性（CVE-2021-35464）
999186	CVE-2021-34523	WEB-MISC Microsoft Exchange Server-権限昇格の脆弱性 (CVE-2021-34523)
999187	CVE-2021-34473	WEB-MISC Microsoft Exchange Server-クエリによるサーバー側の要求偽造認証バイパスの脆弱性 (CVE-2021-34473)
999188	CVE-2021-34473	WEB-MISC Microsoft Exchange Server-クッキーを介したサーバー側要求偽造認証バイパスの脆弱性 (CVE-2021-34473)
999189	CVE-2021-33203	WEB-MISC Django-絶対パスを介した TemplateDetailView ファイルの存在開示の脆弱性 (CVE-2021-33203)
999190	CVE-2021-33203	WEB-MISC Django-パストラバーサルによる TemplateDetailView ファイルの存在開示の脆弱性 (CVE-2021-33203)
999191	CVE-2021-33203	WEB-MISC Django-バックスラッシュによる TemplateDetailView ファイルの存在開示の脆弱性 (CVE-2021-33203)
999192	CVE-2021-33203	WEB-MISC Django-スラッシュによる TemplateDetailView ファイルの存在開示の脆弱性 (CVE-2021-33203)
999193	CVE-2021-3287, CVE-2020-28653	WEB-MISC Zoho ManageEngine opManager 12.5.329 より前のバージョン-認証されていない RCE の脆弱性（CVE-2021-3287、CVE-2020-28653）

署名ルール	**CVE ID**	説明
999194	CVE-2021-32789	WEB-WORDPRESSWooCommerce プラグイン 5.5.0 まで-SQL インジェクションの脆弱性分類と rest_route（CVE-2021-32789）
999195	CVE-2021-32789	WEB-WORDPRESS WooCommerce プラグイン 5.5.0-タクソノミによる SQL インジェクションの脆弱性 (CVE-2021-32789)
999196	CVE-2021-32604	ウェブ-MISC SolarWinds Serv-U 15.2.3 より前-SenderEmail パラメータを介したクロスサイトスクリプティングの脆弱性 (CVE-2021-32604)
999197	CVE-2021-32093	WEB-MISC 国家安全保障局使者 5.9.0-任意のファイル読み取りの脆弱性 (CVE-2021-32093)
999198	CVE-2021-31760	1.974 より前の WEB-MISC Webmin-CSRF の脆弱性が run.cgi 経由で RCE につながる (CVE-2021-31760)
999199	CVE-2021-31207	WEB-MISC Microsoft Exchange Server-セキュリティ機能のバイパスの脆弱性（CVE-2021-31207）
999200	CVE-2021-31195	WEB-MISC Microsoft Exchange Server-リモートコード実行脆弱性 (CVE-2021-31195)
999201	CVE-2021-28474	WEB-MISC Microsoft SharePoint Server-リモートコード実行の脆弱性 (CVE-2021-28474)
999202	CVE-2021-24385	WEB-WORDPRESS FileBird プラグイン 4.7.3-SelectedFolder パラメータと rest_route を介した SQL インジェクションの脆弱性 (CVE-2021-24385)

署名ルール	**CVE ID**	説明
999203	CVE-2021-24385	WEB-WORDPRESS FileBird プラグイン 4.7.3-SelectedFolder パラメータを介した SQL インジェクションの脆弱性（CVE-2021-24385）
999204	CVE-2021-24385	WEB-WORDPRESS FileBird プラグイン 4.7.3-JSON エンコードされたボディを介した SQL インジェクションの脆弱性（CVE-2021-24385）
999205	CVE-2021-24356	WEB-WORDPRESS Simple 301 が 2.0.4 より前にプラグインをリダイレクトする-任意のプラグインアクティベーションの脆弱性（CVE-2021-24356）
999206	CVE-2021-23024	WEB-MISC F5 BIG-IQ 複数のバージョン-リモートコード実行の脆弱性（CVE-2021-23024）
999207	CVE-2021-22911	WEB-MISC Rocket.Chat Server 3.11、3.12、3.13-ブラインド NOSQL インジェクションの脆弱性（CVE-2021-22911）
999208	CVE-2021-22900	WEB-MISC パルス接続 9.1R11.4 より前のセキュアな接続-smimeCert.cgi を介したリモートコード実行の脆弱性（CVE-2021-22900）
999209	CVE-2021-22900	WEB-MISC パルス接続 9.1R11.4 より前のセキュアな接続-admincert.cgi を介したリモートコード実行の脆弱性（CVE-2021-22900）
999210	CVE-2021-22900	WEB-MISC パルス接続 9.1R11.4 より前のセキュアな接続-clientauthcert.cgi を介したリモートコード実行の脆弱性（CVE-2021-22900）
999211	CVE-2021-22160	WEB-MISC Apache Pulsar-JSON ウェブトークン認証バイパスの脆弱性（CVE-2021-22160）

署名ルール	**CVE ID**	説明
999212	CVE-2021-21809	WEB-MISC Moodle-スペルチェッカプラグインと getSoggends メソッドによるリモートコード実行の脆弱性（CVE-2021-21809）
999213	CVE-2021-21809	WEB-MISC Moodle-スペルチェッカープラグインとチェックワードメソッドによるリモートコード実行の脆弱性（CVE-2021-21809）
999214	CVE-2021-21809	WEB-MISC Moodle-s__aspellpath（CVE-2021-21809）を介したリモートコード実行の脆弱性
999215	CVE-2021-21805	WEB-MISC Advantech R-seenet-認証されていないリモートコード実行の脆弱性（CVE-2021-21805）
999216	CVE-2021-21804	WEB-MISC Advantech R-seenet-sub_opt を介したローカルファイル包含の脆弱性 (CVE-2021-21804)
999217	CVE-2021-21587	3.3 より前の Web-MISC Dell Wyse 管理スイート-/image/os/listfiles によるパストラバーサルの脆弱性（CVE-2021-21587）
999218	CVE-2021-21587	3.3 より前の Web-MISC Dell Wyse 管理スイート-/image/app/rsp/listfiles によるパストラバーサルの脆弱性 (CVE-2021-21587)
999219	CVE-2021-21586	3.3 より前の Web-MISC Dell Wyse 管理スイート-/イメージ/アプリケーションとファイル名によるパストラバーサルの脆弱性（CVE-2021-21586）

署名ルール	**CVE ID**	説明
999220	CVE-2021-21586	3.3 より前の Web-MISC Dell Wyse 管理スイート-/image/os およびファイル名によるパストラバーサルの脆弱性（CVE-2021-21586）
999221	CVE-2021-21586	3.3 より前の Web-MISC Dell Wyse 管理スイート-/image/os および FilePath を介したパストラバーサルの脆弱性（CVE-2021-21586）
999222	CVE-2020-25223	WEB-MISC Sophos SG UTM-SID および /var によるリモートコード実行 (CVE-2020-25223)
999223	CVE-2020-25223	WEB-MISC Sophos SG UTM-SID および /webadmin.plx 経由のリモートコード実行 (CVE-2020-25223)
999224	CVE-2020-21056	WEB-MISC FusionPBX 4.5.7-フォルダを介したパストラバーサルの脆弱性（CVE-2020-21056）
999225	CVE-2020-21055	WEB-MISC FusionPBX 4.5.7-ファイル名の変更機能によるパストラバーサルの脆弱性（CVE-2020-21055）
999226	CVE-2020-16245	WEB-MISC Advantech iView 5.7.03.6112 より前-findSummaryUpdateDeviceListExpo のパストラバーサルの脆弱性 (CVE-2020-16245)
999227	CVE-2020-16245	WEB-MISC Advantech iView 5.7.03.6112 より前のパストラバーサルの脆弱性 findCFGDeviceListExport (CVE-2020-16245)
999228	CVE-2020-14181	ウェブ-MISC アトラシアン Jira サーバー-ViewUserHover.jspa を介した情報開示の脆弱性 (CVE-2020-14181)

署名ルール	**CVE ID**	説明
999229	CVE-2020-14005	WEB-MISC SolarWinds Orion 2020.2.1 HF 2-ExecuteVBScript によるリモートコード実行 Action Type (CVE-2020-14005)
999230	CVE-2020-14005	WEB-MISC SolarWinds Orion 2020.2.1 HF 2-ExecuteExternalProgram Action Type によるリモートコード実行 (CVE-2020-14005)

署名更新バージョン **66**

October 25, 2023

2021-07-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999231	CVE-2021-34074	ウェブ-MISC Artica Pandora FMS 7.54 まで-相対パスを介した任意のファイルアップロードの脆弱性 (CVE-2021-34074)
999232	CVE-2021-32633	WEB-MISC Plone CMS-Zope ページテンプレートアップロードによるリモートコード実行の脆弱性 (CVE-2021-32633)
999233	CVE-2021-32633	WEB-MISC Plone CMS-Zope ページテンプレート新しい経由のリモートコード実行の脆弱性 (CVE-2021-32633)
999234	CVE-2021-31181	WEB-MISC Microsoft SharePoint Server-リモートコード実行の脆弱性 (CVE-2021-31181)
999235	CVE-2021-24370	5.6.9 より前の WEB-WORDPRESS ファンシープロダクトデザイナープラグイン-fpd_custom_uplod_file (CVE-2021-24370) を介した RCE の脆弱性
999236	CVE-2021-24370	5.6.9 より前の WEB-WORDPRESS ファンシープロダクトデザイナープラグイン-custom-image-handler.php (CVE-2021-24370) を介して RCE の脆弱性
999237	CVE-2021-24354	WEB-WORDPRESS Simple 301 が 2.0.4 より前にプラグインをリダイレクトする-任意のプラグインインストールの脆弱性 (CVE-2021-24354)
999238	CVE-2021-24352	WEB-WORDPRESS シンプル 301 は 2.0.4 より前のプラグインをリダイレクトする-エクスポートの脆弱性をリダイレクトする (CVE-2021-24352)

署名ルール	**CVE ID**	説明
999239	CVE-2021-1497, CVE-2021-1498	WEB-MISC 4.0 (2e) より前の Cisco ハイパーフレックス HX-リモートコード実行の脆弱性 (CVE-2021-1497、CVE-2021-1498)
999240	CVE-2020-21057	WEB-MISC FusionPBX 4.5.7-フォルダ削除機能によるパストラバーサルの脆弱性（CVE-2020-21057）
999241	CVE-2020-16245	WEB-MISC Advantech iView 5.7.03.6112 より前のバージョン-BackupDatabase を介したパストラバーサルの脆弱性 (CVE-2020-16245)
999242	CVE-2020-10148	WEB-MISC SolarWinds オリオン複数バージョン-認証バイパスの脆弱性（CVE-2020-10148）

署名更新バージョン **65**

October 25, 2023

2021-06-02 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999243	CVE-2021-31761	1.974 より前の WEB-MISC Webmin-/servers/link.cgi/ (CVE-2021-31761) を介した XSS 脆弱性
999244	CVE-2021-31761	1.974 より前の WEB-MISC Webmin-/tunnel/link.cgi/ (CVE-2021-31761) を介した XSS 脆弱性
999245	CVE-2021-31166	WEB-IIS Microsoft HTTP プロトコルスタック-リモートコード実行の脆弱性 (CVE-2021-31166)
999246	CVE-2021-29447	5.7.1 より前の WEB-WORDPRESS WordPress-メディアライブラリ XXE の脆弱性（CVE-2021-29447）
999247	CVE-2021-28157	2021.1 および 2020.3.18 より前の WEB-MISC デボリューションズサーバ-ユーザ削除による SQL インジェクションの脆弱性 (CVE-2021-28157)
999248	CVE-2021-27905	WEB-MISC Apache Solr 8.2.2 より前のバージョン-leaderURL を介したレプリケーションハンドラー SSRF の脆弱性 (CVE-2021-27905)
999249	CVE-2021-27905	WEB-MISC Apache Solr 8.2.2 より前のバージョン-MasterURL を介したレプリケーションハンドラー SSRF 脆弱性 (CVE-2021-27905)
999250	CVE-2021-27890	1.8.26 より前の WEB-MISC MyBB-テーマプロパティ SQL インジェクションの脆弱性 (CVE-2021-27890)

署名ルール	**CVE ID**	説明
999251	CVE-2021-27850, CVE-2019-0195	WEB-MISC Apache Tapestry-認証されていない情報開示の脆弱性 (CVE-2021-27850 および CVE-2019-0195)
999252	CVE-2021-27183	20.0.4 より前の WEB-MISC MDaemon-任意のファイル書き込みの脆弱性（CVE-2021-27183）
999253	CVE-2021-27181	20.0.4 より前の WEB-MISC MDaemon-アンチ CSRF トークン固定の脆弱性（CVE-2021-27181）
999254	CVE-2021-27180	20.0.4 より前の WEB-MISC MDaemon-反射 XSS 脆弱性 (CVE-2021-27180)
999255	CVE-2021-24340	13.0.8 より前の WEB-WORDPRESSWP 統計-認証されていない SQL インジェクションの脆弱性（CVE-2021-24340）
999256	CVE-2021-24171	WEB-WORDPRESS WooCommerce アップロードファイルプラグイン 59.4-パストラバーサルの脆弱性（CVE-2021-24171）
999257	CVE-2021-24171	WEB-WORDPRESS WooCommerce アップロードファイルプラグイン 59.4-任意のファイルアップロードの脆弱性 (CVE-2021-24171)
999258	CVE-2021-22658	WEB-MISC Advantech iView 5.7.03.6112 より前-UserServlet と user_password による SQLi 脆弱性（CVE-2021-22658）
999259	CVE-2021-22658	WEB-MISC Advantech iView 5.7.03.6112 より前-UserServlet と user_name を介した SQLi 脆弱性（CVE-2021-22658）

署名ルール	**CVE ID**	説明
999260	CVE-2021-22658	WEB-MISC Advantech iView 5.7.03.6112 より前のバージョン-CommandServlet と user_password による SQLi 脆弱性（CVE-2021-22658）
999261	CVE-2021-22658	WEB-MISC Advantech iView 5.7.03.6112 より前-CommandServlet と user_name による SQLi 脆弱性 (CVE-2021-22658)
999262	CVE-2021-21983	WEB-MISC 8.4 より前の VMware vRealize Operations Manager-任意のファイル書き込みの脆弱性 (CVE-2021-21983)
999263	CVE-2020-6754	5.2.4 より前の WEB-MISC dotCMS-アセットを介したディレクトリトラバーサルの脆弱性 (CVE-2020-6754)
999264	CVE-2020-27128	WEB-MISC 20.3.1 より前の Cisco SD-WAN vManage-リモート処理による任意のファイル書き込みの脆弱性（CVE-2020-27128）
999265	CVE-2020-27128	WEB-MISC 20.3.1 より前の Cisco SD-WAN vManage-dr を介した任意のファイル書き込みの脆弱性 (CVE-2020-27128)
999266	CVE-2020-15714	WEB-MISC RConfig 3.9.5 およびそれ以前-SQL インジェクションの脆弱性（CVE-2020-15714）
999267	CVE-2020-15713	3.9.6 より前の WEB-MISC RConfig-SQL インジェクションの脆弱性（CVE-2020-15713）
999268	CVE-2020-14295	1.2.13 より前の WEB-MISC Cacti-SQL インジェクションの脆弱性（CVE-2020-14295）

署名ルール	**CVE ID**	説明
999269	CVE-2020-13778	3.9.5 より前の WEB-MISC RConfig-ajaxEditTemplate.php を介したリモートコード実行の脆弱性（CVE-2020-13778）
999270	CVE-2020-13778	3.9.5 より前の WEB-MISC RConfig-ajaxAddTemplate.php を介したリモートコード実行の脆弱性（CVE-2020-13778）
999271	CVE-2020-13592	WEB-MISC Rukovoditel プロジェクト管理アプリ-selected_fields を介した SQL インジェクションの脆弱性 (CVE-2020-13592)
999272	CVE-2020-13592	WEB-MISC Rukovoditel プロジェクト管理アプリ-lists_id を介した SQL インジェクションの脆弱性 (CVE-2020-13592)
999273	CVE-2020-13591	WEB-MISC Rukovoditel プロジェクト管理アプリ-SQL インジェクションの脆弱性（CVE-2020-13591）
999274	CVE-2020-13550	WEB-MISC Advantech WebAccess/SCADA-ファイル名によるパストラバーサルの脆弱性（CVE-2020-13550）

署名更新バージョン **64**

October 25, 2023

2021-04-22 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999275	CVE-2021-3378	WEB-MISC FortiLogger 4.4.2.2-認証されていない任意のファイルアップロードの脆弱性 (CVE-2021-3378)
999276	CVE-2021-28925	2.4.3 より前の WEB-MISC Nagios ネットワークアナライザ-SQL インジェクションの脆弱性 (CVE-2021-28925)
999277	CVE-2021-28924	2.4.3 より前の WEB-MISC Nagios ネットワークアナライザ-XSS の脆弱性（CVE-2021-28924）
999278	CVE-2021-27927	WEB-MISC Zabbix-CSRF 脆弱性によるアクション =authentication.update (CVE-2021-27927)
999279	CVE-2021-26295	WEB-MISC Apache ofBiz 17.12.06-認証されていない任意の逆シリアル化の脆弱性 (CVE-2021-26295)
999280	CVE-2021-25770	WEB-MISC JetBrains YouTrack 以前の 2020.5.3123-サーバーサイドテンプレートインジェクションの脆弱性（CVE-2021-25770）
999281	CVE-2021-25283	3002.5 より前の WEB-MISC SaltStack-リモートコード実行の脆弱性（CVE-2021-25283）

署名ルール	**CVE ID**	説明
999282	CVE-2021-25283	3002.5 より前の WEB-MISC SaltStack-JSON オブジェクトを介したリモートコード実行の脆弱性 (CVE-2021-25283)
999283	CVE-2021-24218	3.0.4 より前の WordPress プラグインのための WEB-WORDPRESS フェイスブック-保存されたクロスサイトスクリプティングの脆弱性 (CVE-2021-24218)
999284	CVE-2021-24217	3.0.2 より前の WordPress プラグインのための WEB-WORDPRESS フェイスブック-PHP オブジェクトインジェクションの脆弱性 (CVE-2021-24217)
999285	CVE-2021-24209	1.7.2 より前の WEB-WORDPRESS WP スーパーキャッシュプラグイン-wp-cache-config.php のリモートコード実行の脆弱性 (CVE-2021-24209)
999286	CVE-2021-24209	1.7.2 より前の WEB-WORDPRESS WP スーパーキャッシュプラグイン-任意のコードインジェクションの脆弱性（CVE-2021-24209）
999287	CVE-2021-24165	3.4.34 より前の WEB-WORDPRESS 忍者フォームプラグイン-オープンリダイレクトの脆弱性（CVE-2021-24165）
999288	CVE-2021-21975	WEB-MISC vRealize オペレーションマネージャ-認証されていないサーバサイド要求偽造の脆弱性 (CVE-2021-21975)
999289	CVE-2020-35578	WEB-MISC Nagios XI 5.8.0 より前のリモートコード実行の脆弱性 (CVE-2020-35578)
999290	CVE-2020-2766	WEB-MISC Oracle WebLogic Server-認証されていない SSRF の脆弱性（CVE-2020-2766）

署名ルール	**CVE ID**	説明
999291	CVE-2020-17523	WEB-MISC 1.7.1 より前の Apache Shiro-スペースを介した認証バイパスの脆弱性（CVE-2020-17523）
999292	CVE-2020-17523	WEB-MISC 1.7.1 より前の Apache Shiro-ドットによる認証バイパスの脆弱性（CVE-2020-17523）
999293	CVE-2020-15160	1.7.6.8 より前の WEB-MISC PrestaShop-SQL インジェクションの脆弱性（CVE-2020-15160）

署名更新バージョン **63**

October 25, 2023

2021-04-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999294	CVE-2021-3273	WEB-MISC NagiOSXI 5.7 より前-コードインジェクションの脆弱性 (CVE-2021-3273)
999295	CVE-2021-3197	3002.3 より前の WEB-MISC SaltStack-ssh_priv (CVE-2021-3197) を介したリモートコード実行の脆弱性
999296	CVE-2021-3197	3002.3 より前の WEB-MISC SaltStack-ssh_port を介したリモートコード実行の脆弱性 (CVE-2021-3197)
999297	CVE-2021-3197	3002.3 より前の WEB-MISC SaltStack-ssh_options によるリモートコード実行の脆弱性 (CVE-2021-3197)
999298	CVE-2021-3197	3002.3 より前の WEB-MISC SaltStack-JSON オブジェクトの proxyCommand によるリモートコード実行の脆弱性 (CVE-2021-3197)
999299	CVE-2021-25282	3002.3 より前の WEB-MISC SaltStack-pillar_roots.write を介したパストラバーサルの脆弱性 (CVE-2021-25282)
999300	CVE-2021-24166	3.4.34 より前の WEB-WORDPRESS 忍者フォームプラグイン-CSRF の脆弱性 (CVE-2021-24166)
999301	CVE-2021-24085	WEB-MISC Microsoft Exchange Server-なりすましの脆弱性 (CVE-2021-24085)
999302	CVE-2021-22986	WEB-MISC F5 iControl REST API-リモートコード実行の脆弱性 (CVE-2021-22986)
999303	CVE-2021-21978	WEB-MISC VMware View Planner Harness 4.6 より前のセキュリティパッチ 1-リモートコード実行の脆弱性（CVE-2021-21978）

署名ルール	**CVE ID**	説明
999304	CVE-2020-23132	WEB-MISC Joomla! 3.9.25 より前-file_path を介した安全でない com_media アップロードパスの脆弱性 (CVE-2020-23132)
999305	CVE-2020-23132	WEB-MISC Joomla! 3.9.25 より前-安全でない com_media アップロードパスの脆弱性 image_path (CVE-2020-23132)
999306	CVE-2020-22425	20.10.4 より前の WEB-MISC Centreon-SQL インジェクションの脆弱性（CVE-2020-22425）

署名更新バージョン **62**

October 25, 2023

2021-03-11 週目に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999307	CVE-2021-27065	WEB-MISC Microsoft Exchange Server-リモートコード実行の脆弱性（CVE-2021-27065）

署名更新バージョン **61**

October 25, 2023

2021-03-11 週目に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999308	CVE-2021-21302	1.7.7.2 より前の WEB-MISC PrestaShop-CSV インジェクションの脆弱性（CVE-2021-21302）
999309	CVE-2020-35749	WEB-WORDPRESS 2.9.4 より前のシンプルなジョブボード-任意のファイル開示の脆弱性（CVE-2020-35749）

署名ルール	**CVE ID**	説明
999310	CVE-2019-16012	WEB-MISC 19.2.2 より前の Cisco SD-WAN vManage-SQL インジェクションの脆弱性 (CVE-2019-16012)

署名更新バージョン **60**

October 25, 2023

2021-03-09 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999311	CVE-2021-26855	WEB-MISC Microsoft Exchange Server-X-anonResource-Backend (CVE-2021-26855) を介したリモートコード実行の脆弱性

署名ルール	**CVE ID**	説明
999312	CVE-2021-26855	WEB-MISC Microsoft Exchange Server-X-berource を介したリモートコード実行の脆弱性 (CVE-2021-26855)

署名更新バージョン **59**

October 25, 2023

2021-03-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999313	CVE-2021-25299	WEB-MISC NagiosXi 5.7.5 まで-URL 経由の XSS 脆弱性 (CVE-2021-25299)
999314	CVE-2021-25298	WEB-MISC NagiosXi 5.7.5 まで-DigitalOcean Wizard を介したリモートコード実行の脆弱性 (CVE-2021-25298)

署名ルール	**CVE ID**	説明
999315	CVE-2021-25297	WEB-MISC NagiOSXI 5.7.5 まで-スイッチウィザードによるリモートコード実行の脆弱性 (CVE-2021-25297)
999316	CVE-2021-25296	WEB-MISC NagiOSXI 5.7.5 まで-WindowsWMI ウィザードによるリモートコード実行の脆弱性 (CVE-2021-25296)
999317	CVE-2021-24164	3.4.34.1 より前の WEB-WORDPRESS 忍者フォームプラグイン-情報開示の脆弱性 (CVE-2021-24164)
999318	CVE-2021-24163	3.4.34 より前の WEB-WORDPRESS 忍者フォームプラグイン-認可バイパスの脆弱性 (CVE-2021-24163)
999319	CVE-2021-21972	WEB-MISC VMware vCenter Server プラグイン-リモートコード実行の脆弱性（CVE-2021-21972）
999320	CVE-2020-35129	3.2.4 より前の WEB-MISC Mautic-新しいソーシャルモニタリングフォームによる XSS の脆弱性 (CVE-2020-35129)
999321	CVE-2020-35129	3.2.4 より前の WEB-MISC Mautic-ソーシャルモニタリングフォームの編集による XSS の脆弱性 (CVE-2020-35129)
999322	CVE-2020-35128	3.2.4 より前の WEB-MISC Mautic-新しい企業フォームを介した XSS 脆弱性（CVE-2020-35128）
999323	CVE-2020-35128	3.2.4 より前の WEB-MISC Mautic-会社の編集フォームによる XSS 脆弱性（CVE-2020-35128）
999324	CVE-2020-35125	3.2.4 より前の WEB-MISC Mautic-リファラーヘッダーを介した XSS の脆弱性 (CVE-2020-35125)

署名ルール	**CVE ID**	説明
999325	CVE-2020-35125	3.2.4 より前の WEB-MISC Mautic-マルチフォームによる XSS の脆弱性 [リターン] (CVE-2020-35125)
999326	CVE-2020-13933	WEB-MISC 1.6.0 より前の Apache Shiro-セミコロンによる認証バイパスの脆弱性（CVE-2020-13933）
999327	CVE-2020-13921, CVE-2020-9483	WEB-MISC Apache SkyWalking 8.4.0 より前-QueryLogs 機能による SQL インジェクションの脆弱性 (CVE-2020-13921、CVE-2020-9483)

署名更新バージョン **58**

October 25, 2023

2021-02-17 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999328	CVE-2021-3317	WEB-MISC KLog サーバ 2.4.1 およびそれ以前-OS コマンドインジェクションの脆弱性（CVE-2021-3317）
999329	CVE-2021-3110	1.7.7.1 より前の WEB-MISC PrestaShop-id_products（CVE-2021-3110）を介した SQL インジェクションの脆弱性
999330	CVE-2021-3110	1.7.7.1 より前の WEB-MISC PrestaShop-/module/ProductComments/commentGrade（CVE-2021-3110）を介した SQL インジェクションの脆弱性
999331	CVE-2021-25646	WEB-MISC 0.20.1 より前の Apache Druid-リモートコード実行の脆弱性（CVE-2021-25646）
999332	CVE-2020-36171	3.0.14 より前の WEB-WORDPRESS・エレメンターページ・ビルダー・プラグイン-XSS の脆弱性（CVE-2020-36171）
999333	CVE-2020-35765	WEB-MISC Zoho ManageEngine アプリケーションマネージャビルド前の 15000-SQL インジェクションの脆弱性（CVE-2020-35765）
999334	CVE-2020-35589	2.15.2 より前にリロードされた WEB WORDPRESS 制限ログイン試行-クロスサイトスクリプティングの脆弱性の反映（CVE-2020-35589）
999335	CVE-2020-26282	2.1.2 より前の WEB-MISC BrowserUp プロキシ-mostRecentEntry を介した RCE の脆弱性につながるテンプレートインジェクション（CVE-2020-26282）
999336	CVE-2020-26282	2.1.2 より前の WEB-MISC BrowserUp プロキシ-エントリを介した RCE の脆弱性につながるテンプレートインジェクション（CVE-2020-26282）

署名ルール	**CVE ID**	説明
999337	CVE-2020-14815	WEB-MISC オラクル・ビジネス・インテリジェンス・エンタープライズ・エディション-クロスサイト・スクリプティングの脆弱性の反映（CVE-2020-14815）
999338		1.2.5.4 より前の WEB-WORDPRESS のお問い合わせフォーム 7 データベースアドオン-一括削除による SQLi 脆弱性

署名更新バージョン **57**

October 25, 2023

2021-02-03 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999339		WEB-MISC Zoom ミーティングコネクタ 4.6.348.20201217-ProxyPasswd を介したリモートでのコード実行の脆弱性
999340		WEB-MISC Zoom ミーティングコネクタ 4.6.348.20201217-ProxyName 経由でリモートでコードが実行される脆弱性
999341	CVE-2021-3129	2.5.2 より前の WEB-MISC イグニッション-認証されていないリモートコード実行の脆弱性 (CVE-2021-3129)
999342	CVE-2021-3025	4.5.4.2 より前の WEB-MISC インビジョンコミュニティ IPS コミュニティスイート-SortDir を介した SQL インジェクションの脆弱性 (CVE-2021-3025)
999343	CVE-2021-2109	WEB-MISC Oracle WebLogic サーバー-JNDI インジェクションによるリモートコード実行の脆弱性 (CVE-2021-2109)
999344	CVE-2020-7200	WEB-MISC HPE システムインサイトマネージャー 7.6.x-AMF セキュアでないデシリアライゼーションの脆弱性 (CVE-2020-7200)
999345	CVE-2020-7199	WEB-MISC HPE EIM 1.21 より前-/プライベート/EIMApplianceIP に不適切な認証の脆弱性 (CVE-2020-7199)
999346	CVE-2020-7199	WEB-MISC HPE EIM 1.21 より前-/プライベート/adminPassReset に不適切な認証の脆弱性 (CVE-2020-7199)

署名ルール	**CVE ID**	説明
999347	CVE-2020-7199	WEB-MISC HPE EIM 1.21 より前-/プライベート/リセットアプライアンスに不適切な認証の脆弱性 (CVE-2020-7199)
999348	CVE-2020-6136	7.5 より前の Web-MISC OS4ED openSIS: DownloadWindow.php を介した SQLi の脆弱性（CVE-2020-6136）
999349	CVE-2020-35729	WEB-MISC ログサーバー 2.4.1 以前-OS コマンドインジェクションの脆弱性（CVE-2020-35729）
999350	CVE-2020-35701	WEB-MISC Cacti 1.2.16 およびそれ以前-site_id を介した SQL インジェクションの脆弱性 (CVE-2020-35701)
999351	CVE-2020-35489	5.3.2 より前の WEB-WORDPRESS お問い合わせフォーム 7: 無制限のファイルアップロードの脆弱性 (CVE-2020-35489)
999352	CVE-2020-27615	1.6.4 より前の WEB-WORDPRESS Loginizer プラグイン-SQL インジェクションの脆弱性 (CVE-2020-27615)
999353	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/sitevariables/create を介した XSS の脆弱性 (CVE-2020-26046)
999354	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/sitevariables/edit を介した XSS の脆弱性 (CVE-2020-26046)
999355	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/ナビゲーション/作成を介した XSS の脆弱性 (CVE-2020-26046)

署名ルール	**CVE ID**	説明
999356	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/ナビゲーション/編集による XSS の脆弱性（CVE-2020-26046）
999357	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/blocks/create を介した XSS の脆弱性（CVE-2020-26046）
999358	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/blocks/edit を介した XSS の脆弱性（CVE-2020-26046）
999359	CVE-2020-26045	WEB-MISC 燃料 CMS 1.4.11-/fuel/パーミッション/作成による SQLi の脆弱性（CVE-2020-26045）
999360	CVE-2020-17519	1.11.3 より前の WEB-MISC Apache Flink：任意のファイル漏洩の脆弱性（CVE-2020-17519）
999361	CVE-2020-17518	WEB-MISC Apache Flink 1.5.1 から 1.11.2：任意のロケーションのファイルアップロードの脆弱性（CVE-2020-17518）
999362	CVE-2019-16010	19.2.2 より前の WEB-MISC Cisco SD-WAN vManage-格納された XSS の脆弱性（CVE-2019-16010）
999363	CVE-2019-15000	WEB-MISC VMware Bitbucket サーバおよびデータセンター-At 経由の Git コマンドインジェクションの脆弱性（CVE-2019-15000）
999364	CVE-2019-15000	WEB-MISC VMware Bitbucket サーバおよびデータセンター-Until/untilId を介した Git コマンドインジェクションの脆弱性（CVE-2019-15000）

署名ルール	**CVE ID**	説明
999365	CVE-2019-15000	WEB-MISC VMware Bitbucket サーバおよびデータセンター-since/sinceID を介した Git コマンドインジェクションの脆弱性 (CVE-2019-15000)

署名更新バージョン **56**

October 25, 2023

2021-01-18 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999366	CVE-2020-8466	WEB-MISC ビルド 1919 より前の Trend Micro IWSSVA 6.5 SP2: 認証されていない OS コマンドインジェクションの脆弱性 (CVE-2020-8466)

署名ルール	**CVE ID**	説明
999367	CVE-2020-6135	7.5 より前の WEB-MISC OS4ED openSIS: Validator.php を介した SQLi の脆弱性（CVE-2020-6135）
999368	CVE-2020-4001	WEB-MISC VMware SD-WAN Orchestrator-パスザハッシュの脆弱性（CVE-2020-4001）
999369	CVE-2020-4000	WEB-MISC VMware SD-WAN Orchestrator-パストラバーサルの脆弱性（CVE-2020-4000）
999370	CVE-2020-3984	WEB-MISC VMware SD-WAN Orchestrator-モジュラスを介した SQL インジェクションの脆弱性 (CVE-2020-3984)
999371	CVE-2020-35606	1.962 までの WEB-MISC Webmin: リモートでコードが実行される脆弱性（CVE-2020-35606）
999372	CVE-2020-17143	WEB-MISC Microsoft Exchange Server-情報開示の脆弱性 (CVE-2020-17143)
999373	CVE-2020-17141	WEB-MISC Microsoft Exchange Server-ルート苦情によるリモートコード実行の脆弱性 (CVE-2020-17141)
999374	CVE-2020-10816	WEB-MISC ビルド 14790 より前の Zoho ManageEngine アプリケーションマネージャー 14-不適切な認証の脆弱性（CVE-2020-10816）
999375	CVE-2019-5533	WEB-MISC VMware SD-WAN Orchestrator-情報開示の脆弱性 (CVE-2019-5533)
999376	CVE-2018-15961	WEB-MISC アップデート 6 または 14 より前の Adobe ColdFusion 12-任意のファイルアップロードの脆弱性（CVE-2018-15961）

署名更新バージョン **55**

October 25, 2023

2020-12-17 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999377		1.21.11 より前の WEB-WORDPRESS TI WooCommerce ウィッシュリストプラグイン-tinvwl_export_settings を介した情報開示の脆弱性
999378		1.21.11 より前の WEB-WORDPRESS TI WooCommerce ウィッシュリストプラグイン-WP オプションは tinvwl_import_settings 経由で脆弱性を変更します
999379	CVE-2020-6134	7.5 より前の Web-MISC OS4ED openSIS：MassDropModal.php を介した SQLi の脆弱性 (CVE-2020-6134)

署名ルール	**CVE ID**	説明
999380	CVE-2020-6133	7.5 より前の WEB-MISC OS4ED openSIS：CourseMoreInfo.php を介した SQLi の脆弱性 (CVE-2020-6133)
999381	CVE-2020-6132	7.5 より前の WEB-MISC OS4ED openSIS：ChooseCP.php を介した SQLi の脆弱性 (CVE-2020-6132)
999382	CVE-2020-6131	7.5 より前の WEB-MISC OS4ED openSIS-MassScheduleSessionSet.php を介した SQLi の脆弱性 (CVE-2020-6131)
999383	CVE-2020-6130	7.5 より前の Web-MISC OS4ED openSIS：MassDropSessionSet.php を介した SQLi の脆弱性 (CVE-2020-6130)
999384	CVE-2020-6129	7.5 より前の Web-MISC OS4ED openSIS-CpSessionSet.php を介した SQLi の脆弱性 (CVE-2020-6129)
999385	CVE-2020-35234	1.4.4 より前の WEB-WORDPRES Easy WP SMTP プラグイン-情報漏えいの脆弱性 (CVE-2020-35234)
999386	CVE-2020-25042	WEB-MISC Mara CMS 7.5：任意のファイルアップロードの脆弱性 (CVE-2020-25042)
999387	CVE-2020-13526	WEB-MISC ProcessMaker-クライアントセットアップ Ajax を介した SQL インジェクションの脆弱性 (CVE-2020-13526)
999388	CVE-2020-13525	WEB-MISC ProcessMaker-ReportTables_AJAX を介した SQL インジェクションの脆弱性 (CVE-2020-13525)

署名ルール	**CVE ID**	説明
999389	CVE-2020-12147	WEB-MISC Silver Peak Unity Orchestrator -SQLExecution REST API を介した任意の MySQL クエリの脆弱性（CVE-2020-12147）
999390	CVE-2020-12146	WEB-MISC Silver Peak Unity Orchestrator -デバッグファイル REST API を介したパストラバーサルの脆弱性（CVE-2020-12146）
999391	CVE-2020-12145	WEB-MISC Silver Peak Unity Orchestrator -認証バイパスの脆弱性（CVE-2020-12145）
999392	CVE-2019-8394	WEB-MISC 10.0 ビルド 10012 より前の Zoho ManageEngine ServiceDesk Plus-任意のファイルアップロードの脆弱性（CVE-2019-8394）
999393	CVE-2019-11447	WEB-Misc CutePHP CuteNews 2.1.2-リモートでコードが実行される脆弱性（CVE-2019-11447）

署名更新バージョン **54**

October 25, 2023

2020-12-02 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。シグニチャアップデートバージョン 54 の一部として、シグニチャ 999720 のログ文字列が変更され、ASCI 文字だけが含まれるようになりました。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999394	CVE-2020-8255	9.1R9 より前の WEB-MISC パルスコネクトセキュア-情報漏えいの脆弱性（CVE-2020-8255）
999395	CVE-2020-6128	7.5 より前の WEB-MISC OS4ED openSIS-CoursePeriodModal.php を介した SQLi の脆弱性 (CVE-2020-6128)
999396	CVE-2020-6126, CVE-2020-6127	7.5 より前の WEB-MISC OS4ED openSIS-CoursePeriodModal.php を介した SQLi の脆弱性 (CVE-2020-6126、CVE-2020-6127)
999397	CVE-2020-28328	7.11.16 より前の WEB-MISC SuiteCRM：リモートでコードが実行される脆弱性（CVE-2020-28328）
999398	CVE-2020-27995	WEB-MISC Zoho ManageEngine アプリケーションマネージャー 14 ビルド前 14560-SQL インジェクションの脆弱性（CVE-2020-27995）
999399	CVE-2020-26879	1.6.0 より前の WEB-MISC Ruckus vRiot サーバ-/servic/ を介した認可バイパスの脆弱性 (CVE-2020-26879)

署名ルール	**CVE ID**	説明
999400	CVE-2020-26879	1.6.0 より前の WEB-MISC Ruckus vRiot サーバー-/reboot による認証バイパスの脆弱性 (CVE-2020-26879)
999401	CVE-2020-26879	1.6.0 より前の WEB-MISC Ruckus vRiot サーバー-/patch/ を介した認証バイパスの脆弱性 (CVE-2020-26879)
999402	CVE-2020-26879	1.6.0 より前の WEB-MISC Ruckus vRiot サーバー-/upgrade/による認証バイパスの脆弱性 (CVE-2020-26879)
999403	CVE-2020-26879	1.6.0 より前の WEB-MISC Ruckus vRiot サーバ-/module/による認可バイパスの脆弱性 (CVE-2020-26879)
999404	CVE-2020-26878	1.6.0 より前の WEB-MISC Ruckus vRiot サーバ：任意の OS コマンドインジェクションの脆弱性 (CVE-2020-26878)
999405	CVE-2020-25790	WEB-MISC タイプセッター CMS 5.x から 5.1：セキュアでないファイルアップロードの脆弱性 (CVE-2020-25790)
999406	CVE-2020-25540	WEB-MISC ThinkAdmin v6-ディレクトリトラバーサルの脆弱性 (CVE-2020-25540)
999407	CVE-2020-14883	WEB-MISC Oracle WebLogic サーバー-認証済みリモートコード実行の脆弱性（CVE-2020-14883）
999408	CVE-2020-14882, CVE-2020-14750	WEB-MISC Oracle WebLogic サーバー-認証バイパスの脆弱性 (CVE-2020-14882、CVE-2020-14750)
999409	CVE-2020-11975, CVE-2020-13942	1.5.2 より前の WEB-MISC Apache Uonomi-リモートでコードが実行される脆弱性（CVE-2020-11975、CVE-2020-13942）

署名ルール	**CVE ID**	説明
999410	CVE-2020-11803	WEB-MISC Titan SpamTitan 7.08 より前-リモートでコードが実行される脆弱性（CVE-2020-11803）

署名更新バージョン **53**

October 25, 2023

2020-11-10 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999411		WEB-WORDPRESS WordPress プラグイン wpDiscuz 7.0.0 7.0.4 まで-認証されていない任意のファイルアップロードの脆弱性
999412		WEB-WORDPRESS クイズ＆アンケートマスター-クロスサイトスクリプティング問題における脆弱性機能

署名ルール	**CVE ID**	説明
999413		6.9 より前の WEB-WORDPRESS WordPress プラグインファイルマネージャー-認証されていない ElFinder コマンドの実行脆弱性
999414	CVE-2020-11700	WEB-MISC Titan SpamTitan 7.08 より前-情報開示の脆弱性 (CVE-2020-11700)
999415	CVE-2020-9446	WEB-MISC Apache ofBiz 17.12.03-XML-RPC の安全でないデシリアライゼーションの脆弱性 (CVE-2020-9446)
999416	CVE-2020-9446	WEB-MISC Apache ofBiz 17.12.03-XML-RPC クロスサイトスクリプティングの脆弱性 (CVE-2020-9446)
999417	CVE-2020-9047	WEB-MISC exacqVision ウェブサービス 20.06.3.0 まで-OS コマンドインジェクションの脆弱性 (CVE-2020-9047)
999418	CVE-2020-8866	WEB-MISC Horde グループウェアウェブメール版 5.2.22-edit.php を介したファイル脆弱性の無制限アップロード（CVE-2020-8866）
999419	CVE-2020-8866	WEB-MISC Horde グループウェアウェブメール版 5.2.22-add.php を介したファイル脆弱性の無制限アップロード（CVE-2020-8866）
999420	CVE-2020-8865	WEB-MISC Horde グループウェアウェブメール版 5.2.22：edit.php を介した任意のファイルインクルージョンの脆弱性（CVE-2020-8865）
999421	CVE-2020-8816	4.3.2 より前の WEB-MISC パイホール-removeStatic を介したリモートコード実行の脆弱性 (CVE-2020-8816)

署名ルール	**CVE ID**	説明
999422	CVE-2020-8816	4.3.2 より前の WEB-MISC パイホール-AddMac 経由でリモートでコードが実行される脆弱性 (CVE-2020-8816)
999423	CVE-2020-8243	9.1R8.2 より前の WEB-MISC パルス接続セキュア：リモートでコードが実行される脆弱性 (CVE-2020-8243)
999424	CVE-2020-8218	9.1R8 より前の WEB-MISC パルス接続セキュア：リモートでコードが実行される脆弱性 (CVE-2020-8218)
999425	CVE-2020-6143, CVE-2020-6144	ウェブその他 OS4ED OpenSIS-/install/Ins1.php を介したコードインジェクションの脆弱性（CVE-2020-6143、CVE-2020-6144）
999426	CVE-2020-6142	WEB-MISC OS4ED OpenSIS-modname を介したパストラバーサルの脆弱性 (CVE-2020-6142)
999427	CVE-2020-6141	7.4 より前の WEB-MISC OS4ED OpenSIS-ユーザー名を介した認証されていない SQLi の脆弱性 (CVE-2020-6141)
999428	CVE-2020-6140	7.5 より前の WEB-MISC OS4ED OpenSIS-username_stn_id を介した認証されていない SQLi の脆弱性（CVE-2020-6140）
999429	CVE-2020-6139	7.5 より前の WEB-MISC OS4ED OpenSIS-username_stf_email を介した認証されていない SQLi の脆弱性 (CVE-2020-6139)
999430	CVE-2020-6138	7.5 より前の WEB-MISC OS4ED OpenSIS-uname を介した認証されていない SQLi の脆弱性 (CVE-2020-6138)

署名ルール	**CVE ID**	説明
999431	CVE-2020-6137	7.5 より前の WEB-MISC OS4ED OpenSIS-password_stf_email を介した認証されていない SQLi の脆弱性 (CVE-2020-6137)
999432	CVE-2020-6125	7.5 より前の WEB-MISC OS4ED OpenSIS: GetSchool.php および u パラメータを介した SQLi の脆弱性（CVE-2020-6125）
999433	CVE-2020-6124	7.5 より前の WEB-MISC OS4ED openSIS-EmailCheckOthers.php を介した SQLi の脆弱性（CVE-2020-6124）
999434	CVE-2020-6123	7.5 より前の WEB-MISC OS4ED OpenSIS-EmailCheck.php および p_id パラメータを介した SQLi の脆弱性（CVE-2020-6123）
999435	CVE-2020-6123	7.5 より前の WEB-MISC OS4ED OpenSIS-EmailCheck.php および電子メールパラメータを介した SQLi の脆弱性（CVE-2020-6123）
999436	CVE-2020-6122	7.5 より前の WEB-MISC OS4ED OpenSIS: CheckDuplicateStudent.php および mn パラメータを介した SQLi の脆弱性（CVE-2020-6122）
999437	CVE-2020-6121	7.5 より前の WEB-MISC OS4ED OpenSIS-CheckDuplicateStudent.php および ln パラメータを介した SQLi の脆弱性（CVE-2020-6121）
999438	CVE-2020-6120	7.5 より前の WEB-MISC OS4ED OpenSIS: CheckDuplicateStudent.php および fn パラメータを介した SQLi の脆弱性（CVE-2020-6120）

署名ルール	**CVE ID**	説明
999439	CVE-2020-6119	7.5 より前の WEB-MISC OS4ED OpenSIS: CheckDuplicateStudent.php および年ごとのパラメータを介した SQLi の脆弱性（CVE-2020-6119）
999440	CVE-2020-6118	7.5 より前の WEB-MISC OS4ED OpenSIS: CheckDuplicateStudent.php および bmonth パラメータを介した SQLi の脆弱性（CVE-2020-6118）
999441	CVE-2020-6117	7.5 より前の WEB-MISC OS4ED OpenSIS: CheckDuplicateStudent.php および bday パラメータを介した SQLi の脆弱性（CVE-2020-6117）
999442	CVE-2020-5780	WEB-WORDPRESS WordPress プラグイン 4.5.6 より前のメール購読者とニュースレター-メール偽造の脆弱性（CVE-2020-5780）
999443	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4-JSON-RPC を介した安全でない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)
999444	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4：リモートメソッドによるセキュアでない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)
999445	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4-RemoteJavaScript を介した安全でない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)
999446	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4-JSON-RPC を介した安全でない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)

署名ルール	**CVE ID**	説明
999447	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4：リモートメソッドによるセキュアでない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)
999448	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4-RemoteJavaScript を介した安全でない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)
999449	CVE-2020-24786	WEB-MISC Zoho ManageEngine AdManager Plus 7.0 ビルド 55 より前-不適切な認証の脆弱性 (CVE-2020-24786)
999450	CVE-2020-24389	WEB-WORDPRESS 1.3.5.5 より前の複数ファイルアップローダープラグインをドラッグアンドドロップする：セキュリティバイパスの脆弱性 (CVE-2020-24389)
999451	CVE-2020-24046	WEB-MISC TitanHQ SpamTitan Gateway 7.08 - 権限昇格の脆弱性 (CVE-2020-24046)
999452	CVE-2020-17506	WEB-MISC Artica ウェブプロキシ 4.30.000000-Apikey パラメータによる事前認証 SQL インジェクションの脆弱性（CVE-2020-17506）
999453	CVE-2020-17505	WEB-MISC Artica Web プロキシ 4.30.000000-サービス-cmds-peform パラメータを介した OS コマンドインジェクションの脆弱性（CVE-2020-17505）
999454	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/ユーザー/アイテム経由の SQLi 脆弱性 (CVE-2020-17463)
999455	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/サイト変数/アイテムを介した SQLi の脆弱性 (CVE-2020-17463)

署名ルール	**CVE ID**	説明
999456	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/パーミッション/アイテムを介した SQLi の脆弱性 (CVE-2020-17463)
999457	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/ページ/アイテム経由の SQLi 脆弱性 (CVE-2020-17463)
999458	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/ナビゲーション/アイテム経由の SQLi 脆弱性 (CVE-2020-17463)
999459	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/logs/items 経由の SQLi 脆弱性 (CVE-2020-17463)
999460	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/ブロック/アイテム経由の SQLi 脆弱性 (CVE-2020-17463)
999461	CVE-2020-16875	WEB-MISC Microsoft Exchange Server-DLP ポリシーリモートでコードが実行される脆弱性 (CVE-2020-16875)
999462	CVE-2020-16171	WEB-MISC 12.5 ビルド 16342 より前のアクロニスサイバーバックアップ-シャードヘッダー経由の SSRF の脆弱性（CVE-2020-16171）
999463	CVE-2020-14947	2.8 より前の WEB-MISC OCS インベントリ-SNMP_MIB_DIRECTORY を介した OS コマンドインジェクションの脆弱性（CVE-2020-14947）
999464	CVE-2020-14947	2.8 より前の WEB-MISC OCS インベントリ： mib_file を介した OS コマンドインジェクションの脆弱性 (CVE-2020-14947)
999465	CVE-2020-14008	WEB-MISC Zoho ManageEngine アプリケーションマネージャー 14710 まで-リモートでコードが実行される脆弱性（CVE-2020-14008）

署名ルール	**CVE ID**	説明
999466	CVE-2020-13925	WEB-MISC 3.1.0 より前の Apache Kylin-ジョブを介したリモートコード実行の脆弱性（CVE-2020-13925）
999467	CVE-2020-13925	WEB-MISC 3.1.0 より前の Apache Kylin-プロジェクト経由でリモートでコードが実行される脆弱性（CVE-2020-13925）
999468	CVE-2020-13854	WEB-MISC Artica Pandora FMS: 権限昇格の脆弱性（CVE-2020-13854）
999469	CVE-2020-13405	1.1.20 より前の WEB-MISC マイクロウェーバー：認証されていない情報漏えいの脆弱性（CVE-2020-13405）
999470	CVE-2020-13376	WEB-MISC SecureEnvoy SecurMail 9.3.503-SecureEnvoyReply クッキーパストラバーサルの脆弱性（CVE-2020-13376）
999471	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica Web プロキシ-ドメイン経由の OS コマンドインジェクションの脆弱性（CVE-2020-13159）
999472	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica ウェブプロキシ: netbiosname を介した OS コマンドインジェクションの脆弱性（CVE-2020-13159）
999473	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica Web プロキシ-エイリアス経由の OS コマンドインジェクションの脆弱性（CVE-2020-13159）
999474	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica Web プロキシ：ホスト名を介した OS コマンドインジェクションの脆弱性（CVE-2020-13159）

署名ルール	**CVE ID**	説明
999475	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica Web プロキシ-dhclient_server を介した OS コマンドインジェクションの脆弱性 (CVE-2020-13159)
999476	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica Web プロキシ-dhclient_interface を介した OS コマンドインジェクションの脆弱性（CVE-2020-13159）
999477	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica ウェブプロキシ: dhclient_mac を介した OS コマンドインジェクションの脆弱性 (CVE-2020-13159)
999478	CVE-2020-13158	4.30.000000 より前の WEB-MISC Artica Web プロキシ：ポップアップによるパストラバーサルの脆弱性 (CVE-2020-13158)
999479	CVE-2020-12851	2.0.7 より前の WEB-MISC Pydio セル：任意のファイル書き込みの脆弱性（CVE-2020-12851）
999480	CVE-2020-12848	2.0.7 より前の WEB-MISC Pydio セル-一時的な共有ユーザーとしてログインする脆弱性 (CVE-2020-12848)
999481	CVE-2020-11699	WEB-MISC Titan SpamTitan 7.08 より前-リモートでコードが実行される脆弱性（CVE-2020-11699）
999482	CVE-2020-11579	WEB-MISC PHPKBV9：ファイル漏洩の脆弱性（CVE-2020-11579）
999483	CVE-2020-10818	WEB-MISC Artica ウェブプロキシ 4.26-fw.system.info.php を介した OS コマンドインジェクションの脆弱性（CVE-2020-10818）

署名ルール	**CVE ID**	説明
999484	CVE-2020-10228	バージョン 20 より前の WEB-MISC Vtenext CE-危険なタイプの脆弱性を持つファイルの無制限アップロード (CVE-2020-10228)
999485	CVE-2020-10204	3.21.2 より前の WEB-MISC ソナタイプネクサスリポジトリマネージャー-CoreUI_user ロールを介した RCE の脆弱性 (CVE-2020-10204)
999486	CVE-2020-10204	3.21.2 より前の WEB-MISC ソナタイプネクサスリポジトリマネージャー-CoreUI_Role 権限による RCE の脆弱性 (CVE-2020-10204)
999487	CVE-2020-10204	3.21.2 より前の WEB-MISC ソナタイプネクサスリポジトリマネージャー-CoreUI_Role ロールを介した RCE の脆弱性 (CVE-2020-10204)
999488	CVE-2020-10199	3.21.2 より前の WEB-MISC Sonatype ネクサスリポジトリマネージャー-REST エンドポイント/bower/グループを介した RCE の脆弱性 (CVE-2020-10199)
999489	CVE-2020-10199	3.21.2 より前の WEB-MISC Sonatype ネクサスリポジトリマネージャー-REST エンドポイント /go/group を介した RCE の脆弱性 (CVE-2020-10199)
999490	CVE-2020-10199	3.21.2 より前の WEB-MISC Sonatype ネクサスリポジトリマネージャー-REST エンドポイント/docker/グループを介した RCE の脆弱性 (CVE-2020-10199)
999491	CVE-2019-19699	WEB-MISC Centreon 19.10 まで: リモートでコードが実行される脆弱性（CVE-2019-19699）
999492	CVE-2019-19499	WEB-MISC Apache Grafana 6.4.3 まで：任意のファイル読み取りの脆弱性（CVE-2019-19499）

署名ルール	**CVE ID**	説明
999493	CVE-2019-18394	WEB-MISC 4.4.2 までのリアルタイムオープンファイアに火をつける-FaviconServlet サーバー側リクエストフォージェリの脆弱性 (CVE-2019-18394)
999494	CVE-2019-18393	WEB-MISC Ignite リアルタイム Openfire 4.4.2 まで-プラグインサーブレットディレクトリトラバーサルの脆弱性（CVE-2019-18393）
999495	CVE-2019-16759	5.6.2 より前の WEB-MISC vBulletin：ネストされたテンプレートを経由してリモートでコードが実行される脆弱性 (CVE-2019-16759)
999496	CVE-2019-15715	1.3.20 および 2.22.1 より前の WEB-Misc mantisBT: neato_tool を介したリモートでのコード実行の脆弱性 (CVE-2019-15715)
999497	CVE-2019-15715	1.3.20 および 2.22.1 より前の WEB-Misc mantisBT：dot_tool を介したリモートでのコード実行の脆弱性（CVE-2019-15715）
999498	CVE-2019-11043	WEB-MISC PHP-FPM 複数バージョン-領域外書き込み脆弱性により任意のコードが実行される (CVE-2019-11043)
999499		WEB-WORDPRESSWordPress プラグインが最大 2.7.6 まで自動最適化-認証された任意のファイルアップロードの脆弱性

署名更新バージョン **52**

October 25, 2023

2020-10-29 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。また、シグニチャルールログ文字列の一部に、脆弱なバージョンが記載されています。それに応じて有効にする必要があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999500	CVE-2018-14667	WEB-MISC RichFaces フレームワーク 3.X から 3.3.4-ユーザーリソース経由の EL インジェクション (CVE-2018-14667)
999501	CVE-2018-12533	WEB-MISC RichFaces フレームワーク 3.1.0 から 3.3.4-Paint2Resource 経由の EL インジェクション (CVE-2018-12533)
999502	CVE-2015-0279, CVE-2018-12532	WEB-MISC RichFaces フレームワーク 4.X から 4.5.17-メディア出力リソースを介した EL インジェクション (CVE-2015-0279、CVE-2018-12532)
999503	CVE-2013-2165	4.3.3 より前の WEB-MISC RichFaces v4：Java オブジェクトのデシリアライゼーションの脆弱性 (CVE-2013-2165)

署名ルール	**CVE ID**	説明
999504	CVE-2013-2165	3.3.4 より前の WEB-MISCRichFaces v3-Java オブジェクトの逆シリアル化の脆弱性（CVE-2013-2165）

署名更新バージョン **51**

October 25, 2023

2020-10-13 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999505		WEB-WORDPRESS WordPress プラグイン wpDiscuz 7.0.0 7.0.4 まで-認証されていない任意のファイルアップロードの脆弱性
999506		WEB-WORDPRESS クイズ＆アンケートマスター-クロスサイトスクリプティング問題における脆弱性機能

署名ルール	**CVE ID**	説明
999507	CVE-2020-8604	WEB-MISC Trend Micro IWS VA 6.5 SP2 より前のパッチ 4-/log_search と cf パラメータを介したパストラバーサルバルン (CVE-2020-8604)
999508	CVE-2020-8604	WEB-MISC Trend Micro IWS VA 6.5 SP2 より前のパッチ 4-/コレクションと cf パラメータを介したパストラバーサル Vuln (CVE-2020-8604)
999509	CVE-2020-8604	WEB-MISC Trend Micro IWS VA 6.5 SP2 より前のパッチ 4-/log_search とファイルパラメータを介したパストラバーサル Vuln (CVE-2020-8604)
999510	CVE-2020-8604	WEB-MISC Trend Micro IWS VA 6.5 SP2 より前のパッチ 4-/collection とファイルパラメータによるパストラバーサル Vuln (CVE-2020-8604)
999511	CVE-2020-7361	WEB-MISC ZentAo エンタープライズ 8.8.3 以前-リポジトリ編集によるリモートコード実行の脆弱性 (CVE-2020-7361)
999512	CVE-2020-7361	WEB-MISC ZentAo Pro 8.8.3 以前-リポジトリ編集によるリモートコード実行の脆弱性 (CVE-2020-7361)
999513	CVE-2020-7361	WEB-MISC ZentAo エンタープライズ 8.8.3 以前-リポジトリ作成によるリモートコード実行の脆弱性 (CVE-2020-7361)
999514	CVE-2020-7361	WEB-MISC ZentAo Pro 8.8.3 以前-リポジトリ作成によるリモートコード実行の脆弱性 (CVE-2020-7361)

署名ルール	**CVE ID**	説明
999515	CVE-2020-5768	WEB-WORDPRESS Icegram 4.5.1 より前のメール購読者およびニュースレタープラグイン-SQL インジェクションの脆弱性 (CVE-2020-5768)
999516	CVE-2020-5767	WEB-WORDPRESS Icegram 4.5.1 より前のメール購読者およびニュースレタープラグイン-CSRF の脆弱性（CVE-2020-5767）
999517	CVE-2020-15299	WEB-WORDPRESS KingComposer プラグイン 2.9.5 より前-クロスサイトスクリプティングの脆弱性 (CVE-2020-15299)
999518	CVE-2020-13854	WEB-MISC Artica Pandora FMS: 権限昇格の脆弱性 (CVE-2020-13854)
999519	CVE-2020-13852	WEB-MISC Artica Pandora FMS: ファイルマネージャを介した任意のファイルアップロードの脆弱性 (CVE-2020-13852)
999520	CVE-2020-13700	WEB-WORDPRESS WordPress プラグイン acf-to-rest-api 3.3.0 より前-URI を介した情報開示の脆弱性（CVE-2020-13700）
999521	CVE-2020-13700	WEB-WORDPRESS WordPress プラグイン acf-to-rest-api 3.3.0 より前-URL を介した情報開示の脆弱性（CVE-2020-13700）
999522	CVE-2020-13379	WEB-MISC Grafana 3.0.1 から 7.0.1-DOS の脆弱性につながる CSRF バイパス（CVE-2020-13379）
999523	CVE-2020-12851	2.0.7 より前の WEB-MISC Pydio セル：任意のファイル書き込みの脆弱性（CVE-2020-12851）
999524	CVE-2020-12848	2.0.7 より前の WEB-MISC Pydio セル-一時的な共有ユーザーとしてログインする脆弱性 (CVE-2020-12848)

署名ルール	**CVE ID**	説明
999525	CVE-2020-11749	7.47 より前の WEB-Misc Artica Pandora FMS-SNMP ブラウザを介したクロスサイトスクリプティングの脆弱性（CVE-2020-11749）
999526	CVE-2020-11579	WEB-MISC PHPKBV9：ファイル漏洩の脆弱性（CVE-2020-11579）
999527	CVE-2020-10546	3.9.5 より前の WEB-MISC rConfig-SearchColumn を介したコンプライアンスポリシーにおける認証されていない SQLi の脆弱性（CVE-2020-10546）
999528	CVE-2020-10546	3.9.5 より前の WEB-MISC rConfig-SearchField を介したコンプライアンスポリシーにおける認証されていない SQLi の脆弱性（CVE-2020-10546）
999529	CVE-2019-16876	1.22.1 より前の WEB-MISC ポータナー：ディレクトリトラバーサルの脆弱性（CVE-2019-16876）
999530		WEB-WORDPRESS-1.5.6 より前の ADNing プラグイン-認証されていない任意のファイル削除の脆弱性
999531		WEB-WORDPRESS-1.5.6 より前の ADning プラグイン-認証されていない任意のファイルアップロードの脆弱性

署名更新バージョン **50**

October 25, 2023

2020-09-26 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999532	CVE-2020-1956	WEB-MISC Apache Kylin-キューブ dest-config 経由でのリモートコード実行の移行 (CVE-2020-1956)
999533	CVE-2020-1956	WEB-MISC Apache Kylin-キューブは src-config 経由でリモートコード実行を移行します (CVE-2020-1956)
999534	CVE-2020-1956	WEB-MISC Apache Kylin-キューブは ProjectName 経由でリモートコード実行を移行します (CVE-2020-1956)
999535	CVE-2020-3247	WEB-MISC Cisco UCS Director: CopyFileRunnable 任意のシンボリックリンク作成の脆弱性 (CVE-2020-3247)
999536	CVE-2019-16872	1.22.1 より前の WEB-MISC Portainer-アップデートスタックを介した不正なアクセス制御の脆弱性 (CVE-2019-16872)
999537	CVE-2019-16872	1.22.1 より前の WEB-MISC Portainer-スタックの作成による不正なアクセス制御の脆弱性 (CVE-2019-16872)

署名ルール	**CVE ID**	説明
999538	CVE-2020-13855	WEB-MISC Artica Pandora FMS 7.44：ファイルリポジトリマネージャを介した任意のファイルアップロードの脆弱性（CVE-2020-13855）
999539	CVE-2020-5902	WEB-MISC F5 BIG-IP-/hsqldb 経由でのトラフィック管理ユーザーインターフェイス RCE の脆弱性（CVE-2020-5902）
999540	CVE-2020-5902	WEB-MISC F5 BIG-IP-/tmui 経由のトラフィック管理ユーザーインターフェイス RCE の脆弱性（CVE-2020-5902）
999541		WEB-MISC WebERP 4.15.1 以前-認証されていない情報漏えいの脆弱性
999542	CVE-2020-7209	WEB-MISC 6.0-2 より前の HP LinuxKI-timeline.php およびタイムスタンプパラメータによる認証されていない RCE の脆弱性（CVE-2020-7209）
999543	CVE-2020-7209	WEB-MISC 6.0-2 より前の HP LinuxKI-kivis.php および ts パラメータを介した認証されていない RCE の脆弱性（CVE-2020-7209）
999544	CVE-2020-7209	6.0-2 より前の WEB-MISC HP LinuxKI-kivis.php およびエンドパラメータを介した認証されていない RCE の脆弱性（CVE-2020-7209）
999545	CVE-2020-7209	WEB-MISC 6.0-2 より前の HP LinuxKI-kivis.php および起動パラメータを介した認証されていない RCE の脆弱性（CVE-2020-7209）
999546	CVE-2020-7209	WEB-MISC 6.0-2 より前の HP LinuxKI-kivis.php および pid パラメータを介した認証されていない RCE の脆弱性（CVE-2020-7209）

署名ルール	**CVE ID**	説明
999547	CVE-2020-7209	6.0-2 より前の WEB-MISC HP LinuxKI-kidsk_trace_view.php およびエンドパラメータを介した認証されていない RCE の脆弱性 (CVE-2020-7209)
999548	CVE-2020-7209	WEB-MISC 6.0-2 より前の HP LinuxKI-kidsk_trace_view.php および起動パラメータを介した認証されていない RCE の脆弱性 (CVE-2020-7209)
999549		WEB-MISC 9.03.70 より前の PHP フュージョン-PHP オブジェクトインジェクションの脆弱性
999550	CVE-2020-1181	WEB-MISC Microsoft SharePoint サーバー-Web パーツを介したリモートコード実行 (CVE-2020-1181)
999551	CVE-2020-10547	3.9.5 より前の WEB-MISC rConfig-SearchColumn を介したポリシー要素における認証されていない SQLi の脆弱性 (CVE-2020-10547)
999552	CVE-2020-10547	3.9.5 より前の WEB-MISC rConfig: SearchField を介したポリシー要素における認証されていない SQLi の脆弱性 (CVE-2020-10547)
999553	CVE-2020-8605	WEB-MISC Trend Micro InterScan Web Security 仮想アプライアンス 6.5 SP2 パッチ 4 より前-RCE の脆弱性 (CVE-2020-8605)
999554	CVE-2019-10068	WEB-MISC Kentico CMS 複数バージョン-認証されていないリモートコード実行の脆弱性 (CVE-2019-10068)

999555	CVE-2020-11108	4.4 までの WEB-MISCPi-hole-Authenticated RCE Vulnerability（CVE-2020-11108）

署名更新バージョン **49**

October 25, 2023

2020-08-26 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999556	CVE-2020-13241	WEB-MISC Microweber 1.1.18-危険なタイプの脆弱性を持つファイルの無制限アップロード (CVE-2020-13241)
999557	CVE-2020-3250	WEB-MISC Cisco UCS Director-UserapiDownloadFile を介した REST API パストラバーサルの脆弱性（CVE-2020-3250）

署名ルール	**CVE ID**	説明
999558		WEB-WORDPRESS PageBuilder KingComposer プラグイン 2.9.4 より前-アクションによるディレクトリの任意の削除 = 一括削除
999559		WEB-WORDPRESS PageBuilder KingComposer プラグイン 2.9.4 より前-アクション = アップロードによるリモートコード実行の脆弱性
999560	CVE-2018-1999024	WEB-MISC Moodle-MathJax Unicode クロスサイトスクリプティングの脆弱性 (CVE-2018-1999024)
999561	CVE-2020-13693	2.6.5 より前の WEB-WORDPRESS bbPress プラグイン：認証されていない権限昇格の脆弱性 (CVE-2020-13693)
999562	CVE-2020-12847	2.0.7 より前の WEB-MISCPydio セル-リモートコード実行の脆弱性 (CVE-2020-12847)

署名更新バージョン **48**

October 25, 2023

2020-07-01 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999563		WEB-WORDPRESS ページビルダー PageLayer プラグイン 1.1.2 より前-クロスサイトスクリプティング pagelayer_cf_to_email 経由の脆弱性
999564		WEB-WORDPRESS ページビルダー PageLayer プラグイン 1.1.2 より前-ページレイヤ電話を介したクロスサイトスクリプティングの脆弱性
999565		WEB-WORDPRESS ページビルダー PageLayer プラグイン 1.1.2 より前-ページレイヤアドレスを介したクロスサイトスクリプティングの脆弱性
999566	CVE-2020-1961	WEB-MISC Apache Syncope -サーバー側テンプレートインジェクションの脆弱性（CVE-2020-1961）
999567	CVE-2019-18935	ASP.NET AJAX の WEB-MISC 進捗テレリック UI-RadAsyncUpload .NET デシリアライゼーションの脆弱性 (CVE-2019-18935)
999568	CVE-2020-9463	WEB-MISC Centreon 19.10：OS コマンドインジェクションの脆弱性（CVE-2020-9463）
999569		3.7.6 より前の WEB-WORDPRESS サポートレビュープラグイン-認証されていないストアドクロスサイトスクリプティングの脆弱性

署名ルール	**CVE ID**	説明
999570		WEB-WORDPRESS ページビルダー PageLayer プラグイン 1.1.2 より前-pagelayer_save_template 経由での不適切なアクセス制御 Vuln
999571		WEB-WORDPRESS ペスページビルダー PageLayer プラグイン 1.1.2 より前-pagelayer_update_site_title 経由での不適切なアクセス制御 Vuln
999572		WEB-WORDPRESS ペスページビルダー PageLayer プラグイン 1.1.2 より前-pagelayer_save_content 経由での不適切なアクセス制御 Vuln
999573		1.3.3.3 より前のお問い合わせフォーム 7 の WEB-WORDPRESS ドラッグアンドドロップアップロード-任意のファイル拡張子のアップロードの脆弱性
999574	CVE-2020-9314	WEB-MISC Oracle iPlanet ウェブサーバー 7.0.x-イメージインジェクションの脆弱性 (CVE-2020-9314)
999575	CVE-2020-9484	WEB-MISC Apache Tomcat 複数バージョン-信頼できないデータのデシリアライズ (CVE-2020-9484)
999576	CVE-2020-13252	19.04.15 より前の WEB-MISC セントレオン：リモートでコードが実行される脆弱性（CVE-2020-13252）
999577	CVE-2020-11453	WEB-MISC マイクロストラテジーウェブ-SOAP 経由の CSRF 脆弱性 (CVE-2020-11453)
999578	CVE-2020-11453	WEB-MISC マイクロストラテジーウェブ-CSRF 脆弱性 (CVE-2020-11453)
999579	CVE-2020-7237	1.2.8 より前の WEB-MISC Cacti-リモートでコードが実行される脆弱性（CVE-2020-7237）

署名更新バージョン **47**

October 25, 2023

2020-06-12 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999580	CVE-2020-6010	3.2.6.9 より前の WEB-WORDPRESS LearnPress LMS プラグイン-SQL インジェクションの脆弱性 (CVE-2020-6010)
999581		WEB-MISC Nagios XI 5.6.13 まで-サービスコマンド _Test 任意のコマンド実行の脆弱性
999582	CVE-2020-0932	Microsoft SharePoint サーバー-SOAP 1.2 を介した WebPart ソースマークアップのリモートコード実行の脆弱性 (CVE-2020-0932)
999583	CVE-2020-0932	Microsoft SharePoint サーバー-Web パーツソースマークアップ SOAP 1.1 経由でリモートでコードが実行される脆弱性 (CVE-2020-0932)

署名ルール	**CVE ID**	説明
999584	CVE-2020-12642	3.4.24.2 より前の WEB-WORDPRESS Ninja Forms プラグイン-インポートフィールドを介したクロスサイトリクエストフォージェリの脆弱性 (CVE-2020-12642)
999585	CVE-2020-12642	WEB-WORDPRESS 3.4.24.2 より前の Ninja Forms プラグイン-インポートフォームを介したクロスサイトリクエストフォージェリの脆弱性 (CVE-2020-12642)
999586	CVE-2020-11450	WEB-MISC マイクロストラテジー Web 10.4-情報開示の脆弱性 (CVE-2020-11450)
999587	CVE-2020-7935	WEB-MISC Artica Pandora FMS 7.0-危険なタイプの脆弱性を持つファイルを無制限にアップロードすると RCE が許可される (CVE-2020-7935)
999588	CVE-2020-12116	WEB-MISC ビルド 125125 より前の Zoho ManageEngine OpManager-情報漏えいの脆弱性 (CVE-2020-12116)
999589		2.9.6 より前の WEB-WORDPRESS エレメントまたはページビルダー-権限昇格の脆弱性
999590	CVE-2020-11738	WEB-WORDPRESS-1.3.28 より前のスナップクリークデュプリケータープラグイン-パストラバーサルの脆弱性（CVE-2020-11738）
999591	CVE-2020-10389	WEB-MISC Chadha PHPKB 標準多言語 9-リモートでコードが実行される脆弱性（CVE-2020-10389）
999592	CVE-2020-11516	WEB-WORDPRESS お問い合わせフォーム 7 Datepicker プラグイン 2.6.0 まで-ストアドクロスサイトスクリプティングの脆弱性 (CVE-2020-11516)

署名ルール	**CVE ID**	説明
999593		WEB-MISC Nagios XI 5.6.13 まで-ステップ経由でのエクスポート RRD 任意のコマンド実行の脆弱性
999594		WEB-MISC Nagios XI 5.6.13 まで-エンド経由でのエクスポート RRD 任意のコマンド実行の脆弱性
999595		WEB-MISC Nagios XI 5.6.13 まで-エクスポート RRD 任意のコマンドが起動時に実行される脆弱性
999596	CVE-2019-19799	Zoho ManageEngine アプリケーションマネージャー 14600 より前-情報漏えいの脆弱性 (CVE-2019-19799)
999597	CVE-2020-10458	WEB-MISC Chadha PHPKB 標準多言語 9-任意のフォルダを削除する脆弱性（CVE-2020-10458）
999598	CVE-2017-9822	9.1.1 より前の WEB-MISC DNN: DNNPersonalization クッキーを介したリモートコード実行の脆弱性 (CVE-2017-9822)
999599	CVE-2020-7953	WEB-MISC opServices opMon 9.3.2：nmap_options パラメータを介した認証されていない情報漏洩の脆弱性（CVE-2020-7953）
999600	CVE-2020-7953	WEB-MISC OpServices opMon 9.3.2：ホストパラメータを介した認証されていない情報漏洩の脆弱性 (CVE-2020-7953)
999601		WEB-MISC Bolt CMS 3.7.0-newname パラメータによる危険なタイプの脆弱性へのファイル名変更
999602		WEB-MISC Bolt CMS 3.7.0-newname パラメータによるパストラバーサルの脆弱性
999603		WEB-MISC Bolt CMS 3.7.0-oldname パラメータによるパストラバーサルの脆弱性

署名ルール	**CVE ID**	説明
999604		WEB-MISC Bolt CMS 3.7.0-親パラメータによるパストラバーサルの脆弱性
999605		WEB-MISC Bolt CMS 3.7.0-表示名パラメータの不適切なフィールド検証の脆弱性
999606	CVE-2020-9004	WEB-MISC-Wowza ストリーミングエンジン 4.7.8-ビューログに不正な認証の脆弱性（CVE-2020-9004）
999607	CVE-2020-9004	WEB-MISC-Wowza ストリーミングエンジン 4.7.8-メディアキャッシュ設定における不正な認証の脆弱性 (CVE-2020-9004)
999608	CVE-2020-9004	WEB-MISC-Wowza ストリーミングエンジン 4.7.8-アプリケーション設定での不正な認証の脆弱性 (CVE-2020-9004)
999609	CVE-2020-9004	WEB-MISC-Wowza ストリーミングエンジン 4.7.8-サーバー設定での不正な認証の脆弱性 (CVE-2020-9004)
999610		WEB-MISC PrestaShop 1.7.6.5-ファイルマネージャを介した CSRF の脆弱性
999611	CVE-2020-10238	WEB-MISC Joomla! 3.9.16 より前：com_templates を介したセキュリティバイパスの脆弱性 (CVE-2020-10238)
999612	CVE-2020-11510	3.2.6.9 より前の WEB-WORDPRESS LearnPress LMS プラグイン-learnpress_create_page を介した権限昇格 (CVE-2020-11510)
999613	CVE-2020-11510	WEB-WORDPRESS LearnPress LMS プラグイン 3.2.6.9 より前-learnpress_update_order_status による権限昇格 (CVE-2020-11510)

署名ルール	**CVE ID**	説明
999614	CVE-2020-8636	WEB-MISC OpServices opMon 9.3.2-nmap_options パラメーターによる認証されていないリモートコード実行の脆弱性 (CVE-2020-8636)
999615	CVE-2020-8636	WEB-MISC OpServices opMon 9.3.2：ホストパラメータによる認証されていないリモートコード実行の脆弱性（CVE-2020-8636）
999616	CVE-2020-11511	3.2.6.9 より前の WEB-WORDPRESS LearnPress LMS プラグイン-教師になる受け入れによる権限昇格 (CVE-2020-11511)
999617	CVE-2020-11451	WEB-MISC Microstrategy Web-JSP 経由でのセキュアでないファイルタイプのアップロードの脆弱性（CVE-2020-11451）
999618	CVE-2020-11451	WEB-MISC Microstrategy Web-ASP 経由でのセキュアでないファイルタイプのアップロードの脆弱性（CVE-2020-11451）
999619	CVE-2020-11515	1.0.41 より前の WEB-WORDPRESS WP SEO プラグインのランク数学-URL を介した REST API を介したリダイレクトの脆弱性（CVE-2020-11515）
999620	CVE-2020-11515	1.0.41 より前の WEB-WORDPRESS WP SEO プラグインのランク数学-REST API rest_route パラメータを介したリダイレクトの脆弱性 (CVE-2020-11515)
999621	CVE-2020-10457	WEB-MISC Chadha PHPKB 標準多言語 9-imgName を介した任意のファイル名変更の脆弱性 (CVE-2020-10457)

署名ルール	**CVE ID**	説明
999622	CVE-2020-10457	WEB-MISC Chadha PHPKB 標準多言語 9-imgURL を介した任意のファイル名変更の脆弱性 (CVE-2020-10457)
999623	CVE-2019-1821	WEB-MISC Cisco プライムインフラストラクチャ：リモートでコードが実行される脆弱性 (CVE-2019-1821)
999624		2.10.16 より前の WEB-WORDPRESS ページビルダープラグイン-Ajax action_builder_content を介した CSRF の脆弱性
999625		2.10.16 より前の WEB-WORDPRESS ページビルダープラグイン-ライブエディターによる CSRF の脆弱性
999626	CVE-2020-11514	WEB-WORDPRESS WP SEO プラグイン Rank Math 1.0.41 より前-URL 経由の REST API による権限昇格 (CVE-2020-11514)
999627	CVE-2020-11514	WEB-WORDPRESS WP SEO プラグイン Rank Math 1.0.41 より前-REST API を介した権限昇格 rest_route パラメーター (CVE-2020-11514)
999628	CVE-2019-6713	5.0.190312 より前の WEB-MISC ThinkCMF-/route/editpost.html を介したコードインジェクションの脆弱性（CVE-2019-6713）
999629	CVE-2019-6713	5.0.190312 より前の WEB-MISC ThinkCMF-/route/addpost.html を介したコードインジェクションの脆弱性（CVE-2019-6713）
999630		1.8.0 より前の WEB-WORDPRESS Google サイトキットプラグイン-保護されていない検証の脆弱性

署名ルール	**CVE ID**	説明
999631	CVE-2020-9315	WEB-MISC Oracle iPlanet ウェブサーバー 7.0.x-不正なアクセス制御の脆弱性（CVE-2020-9315）
999632	CVE-2020-1947	WEB-MISC Apache ShardingSphere 4.0.0-RC3 および 4.0.0-SnakeYAML リモートでコードが実行される脆弱性（CVE-2020-1947）
999633	CVE-2020-7961	7.2.1 CE GA2 より前の Liferay ポータル-JSON-RPC を介した JSONWS デシリアライゼーション RCE の脆弱性（CVE-2020-7961）
999634	CVE-2020-7961	7.2.1 CE GA2 より前の Liferay ポータル-URL パス経由の JSONWS デシリアライゼーション RCE の脆弱性（CVE-2020-7961）
999635	CVE-2020-7961	7.2.1 CE GA2 より前の Liferay ポータル-フォームおよび URI クエリによる JSONWS デシリアライゼーション RCE の脆弱性（CVE-2020-7961）
999636	CVE-2020-8518	WEB-MISC Horde グループウェアウェブメール版 5.2.22：リモートでコードが実行される脆弱性（CVE-2020-8518）
999637	CVE-2020-7351	WEB-MISC Fonality Trixbox CE 2.8.0.4 およびそれ以前-リモートでコードが実行される脆弱性（CVE-2020-7351）
999638	CVE-2020-12720	5.6.1 より前の WEB-MISC vBulletin パッチレベル 1-認証されていない SQL インジェクションの脆弱性（CVE-2020-12720）
999639	CVE-2019-19800	Zoho ManageEngine アプリケーションマネージャー 14520 より前：パストラバーサルの脆弱性（CVE-2019-19800）

署名ルール	**CVE ID**	説明
999640	CVE-2020-10386	WEB-MISC Chadha PHPKB 標準多言語 9-リモートコード実行 (CVE-2020-10386)
999641	CVE-2020-8497	WEB-MISC Artica Pandora FMS 7.0：認証されていない情報漏洩の脆弱性（CVE-2020-8497）
999642	CVE-2020-6009	WEB-WORDPRESS LearnDash LMS プラグイン 3.1.6 より前-認証されていない SQL インジェクションの脆弱性 (CVE-2020-6009)

署名更新バージョン **46**

October 25, 2023

2020-06-03 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999643		WEB-WORDPRESS 10.0.64 より前の Google マッププラグイン用ウェブマップビルダー-gmwd_setup ページを介した認証されていないクロスサイトスクリプティングの脆弱性
999644		WEB-WORDPRESS 10Google マッププラグイン 10.0.64 およびそれ以前のウェブマップビルダー-options_gmwd ページを介したクロスサイトスクリプティングの脆弱性
999645	CVE-2020-5187	WEB-MISC DNN 9.4.4 まで：URL 経由のパストラバーサルの脆弱性 (CVE-2020-5187)
999646	CVE-2020-5187	WEB-MISC DNN 9.4.4 まで：ローカル経由のパストラバーサルの脆弱性（CVE-2020-5187）
999647	CVE-2020-9335	1.5.46 より前の WEB-WORDPRESS フォトギャラリープラグイン-image_alt_text_ フィールドを介したクロスサイトスクリプティングの脆弱性 (CVE-2020-9335)
999648	CVE-2020-9335	1.5.46 より前の WEB-WORDPRESS フォトギャラリープラグイン-名前フィールドを介したクロスサイトスクリプティングの脆弱性（CVE-2020-9335）
999649	CVE-2020-9335	1.5.46 より前の WEB-WORDPRESS フォトギャラリープラグイン-説明フィールドを介したクロスサイトスクリプティングの脆弱性（CVE-2020-9335）

署名ルール	**CVE ID**	説明
999650	CVE-2020-10189	WEB-MISC 10.0.479 より前の Zoho ManageEngine デスクトップセントラル-認証されていないリモートコード実行 Vuln (CVE-2020-10189)
999651	CVE-2020-10189	WEB-MISC 10.0.479 より前の Zoho ManageEngine デスクトップセントラル-認証されていない任意のファイルアップロードの Vuln (CVE-2020-10189)
999652		WEB-WORDPRESS 2.3.2 より前の WooCommerce プラグインの柔軟なチェックアウトフィールド-認証されていない設定変更 Vuln
999653	CVE-2020-0688	WEB-MISC Microsoft Exchange Server-検証キーのリモートコード実行の脆弱性（CVE-2020-0688）
999654	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0：ip_src パラメータを介したリモートコード実行の脆弱性 (CVE-2020-8947、 CVE-2019-20224)
999655	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0-dst_port パラメータを介したリモートコード実行の脆弱性 (CVE-2020-8947、 CVE-2019-20224)
999656	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0：src_port パラメータを介したリモートコード実行の脆弱性 (CVE-2020-8947、 CVE-2019-20224)
999657	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0：ip_dst パラメータによるリモートコード実行の脆弱性 (CVE-2020-8947、 CVE-2019-20224)

署名ルール	**CVE ID**	説明
999658	CVE-2020-5186	WEB-MISC DNN 9.5.0 まで：ジャーナル XML アップロードによるクロスサイトスクリプティングの脆弱性（CVE-2020-5186）
999659		WEB-WORDPRESS WP サイトマップページプラグイン 1.6.2 以前-クロスサイトスクリプティング wsp_exclude_pages 経由の脆弱性
999660	CVE-2020-5188	WEB-MISC DNN 9.5.0 まで-UploadFromURL を介したセキュアでない権限の脆弱性（CVE-2020-5188）
999661	CVE-2020-5188	WEB-MISC DNN 9.5.0 まで-UploadFromLocal を介したセキュアでない権限の脆弱性（CVE-2020-5188）
999662	CVE-2020-7799	1.11.0 より前の WEB-MISC FusionAuth：API テーマを介したリモートコード実行の脆弱性（CVE-2020-7799）
999663	CVE-2020-7799	1.11.0 より前の WEB-MISC FusionAuth：API 電子メールテンプレートを介したリモートコード実行の脆弱性（CVE-2020-7799）
999664	CVE-2020-7799	1.11.0 より前の WEB-MISC FusionAuth：GUI テーマを介したリモートコード実行の脆弱性（CVE-2020-7799）
999665	CVE-2020-7799	1.11.0 より前の WEB-MISC FusionAuth：GUI 電子メールテンプレートを介したリモートコード実行の脆弱性（CVE-2020-7799）

署名更新バージョン **45**

October 25, 2023

2020-05-26 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。最新の Snort リリースによると、ID が 1258、1306、2520、2661、5695、10996、11817、12056、15471、17049、21634 のシグニチャルールは削除されました。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999666		1.3.28 より前の WEB-WORDPRESS デュプリケータプラグイン-認証されていない任意のファイルダウンロードの脆弱性
999667	CVE-2020-10220	WEB-MISC rConfig から 3.94 まで-SQL インジェクションの脆弱性 (CVE-2020-10220)
999668	CVE-2020-5844	WEB-MISC Artica Pandora FMS 7.0-/attachment/files_repo/ を介した危険なタイプの任意のファイルの実行 (CVE-2020-5844)

署名ルール	**CVE ID**	説明
999669	CVE-2020-8813	1.2.10 より前の WEB-MISC Cacti-graph_realtime.php 経由でリモートでコードが実行される脆弱性（CVE-2020-8813）
999670	CVE-2020-8654	WEB-MISC EyesofNetwork 5.3：リモートでコードが実行される脆弱性（CVE-2020-8654）
999671	CVE-2020-10196	WEB-WORDPRESS Sygnoos ポップアップビルダープラグイン 3.64.1 より前：認証されていないクロスサイトスクリプティングの脆弱性（CVE-2020-10196）
999672	CVE-2019-15949	5.6.6 より前の WEB-MISC Nagios XI：ルートとしてリモートでコードが実行される脆弱性（CVE-2019-15949）
999673	CVE-2020-10879	WEB-MISC rConfig 3.9.5 以前-search.crud.php 経由でリモートでコードが実行される脆弱性（CVE-2020-10879）
999674	CVE-2020-8656	WEB-MISC EyesOfNetwork 5.3 -EyesOfNetwork API 2.4.2 SQL インジェクションの脆弱性 (CVE-2020-8656)
999675	CVE-2020-10195	3.64.1 より前の WEB-WORDPRESS Sygnoos ポップアップビルダープラグイン-認証済みシステム情報開示 (CVE-2020-10195)
999676	CVE-2020-10195	3.64.1 より前の WEB-WORDPRESS Sygnoos ポップアップビルダープラグイン-認証済みサブスクライバー情報開示 (CVE-2020-10195)
999677	CVE-2020-10195	3.64.1 より前の WEB-WORDPRESS Sygnoos ポップアップビルダープラグイン-認証済み設定の変更 (CVE-2020-10195)

署名ルール	**CVE ID**	説明
999678	CVE-2020-0646	Microsoft SharePoint Server-.NET フレームワークワークフローで SOAP 1.2 経由でリモートでコードが実行される脆弱性 (CVE-2020-0646)
999679	CVE-2020-0646	Microsoft SharePoint Server-.NET フレームワークワークフローで SOAP 1.1 経由でリモートでコードが実行される脆弱性 (CVE-2020-0646)
999680	CVE-2020-10221	WEB-MISC rConfig から 3.94 まで-リモートでコードが実行される脆弱性（CVE-2020-10221）
999681	CVE-2019-19134	WEB-WORDPRESS ヒーローマッププレミアム 2.2.3 より前-認証されていないリフレクションクロスサイトスクリプティングの脆弱性（CVE-2019-19134）
999682	CVE-2020-10385	1.5.9 より前の WEB-WORDPRESS WPForms プラグイン-ストアドクロスサイトスクリプティングの脆弱性（CVE-2020-10385）

署名更新バージョン **44**

October 25, 2023

2020-04-27 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999683	CVE-2020-9043	1.5.1 より前の WEB-WORDPRESS WPCentral プラグイン-接続キー漏洩の脆弱性（CVE-2020-9043）
999684		WEB-WORDPRESS Duplicate-Post プラグインバージョン 3.2.3 以前-永続的なクロスサイトスクリプティング
999685		WEB-WORDPRESS Duplicate-Post プラグインバージョン 3.2.3 以前-永続的なクロスサイトスクリプティング
999686	CVE-2020-0618	WEB-MISC Microsoft SQL Server Reporting Services-リモートでコードが実行される脆弱性 (CVE-2020-0618)
999687	CVE-2019-16278	WEB-MISC 1.3.7 より前のノストロモ Nhttpd-strcutl 関数が認証されていないリモートコード実行を許可する (CVE-2019-16278)
999688	CVE-2019-1937	WEB-MISC Cisco UCS Director 6.6.0.0 ～6.6.1.0 および 6.7.0.0 ～6.7.1.0：認証バイパスの脆弱性（CVE-2019-1937）
999689		WEB-WORDPRESS Duplicate-Post プラグインバージョン 3.2.3 以前-永続的なクロスサイトスクリプティング

署名ルール	**CVE ID**	説明
999690	CVE-2020-9006	3.0 より前の WEB-WORDPRESS ポップアップビルダープラグイン-PHP デシリアライゼーションによる SQL インジェクションの脆弱性（CVE-2020-9006）
999691		WEB-WORDPRESS Duplicate-Post プラグインバージョン 3.2.3 以前-永続的なクロスサイトスクリプティング
999692		WEB-MISC はコンテンツ長と転送エンコーディングヘッダーによるリクエストの密輸を防止
999693		1.6.3 より前の WEB-WORDPRESS ThemeGrill デモインポータープラグイン-認証バイパスとデータベースワイプの脆弱性
999694	CVE-2019-17237	WEB-WORDPRESS IgniteUp 近日公開およびメンテナンスモードプラグイン 3.4.1 より前-メッセージを介した CSRF の脆弱性 (CVE-2019-17237)
999695	CVE-2019-17237	WEB-WORDPRESS IgniteUp 近日公開、3.4.1 より前のメンテナンスモードプラグイン-サブジェクト経由の CSRF の脆弱性 (CVE-2019-17237)

署名更新バージョン **43**

October 25, 2023

2020-02-27 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999696	CVE-2019-15983	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：CablePlan 経由の XML 外部エンティティの脆弱性 (CVE-2019-15983)
999697	CVE-2019-20197	WEB-MISC Nagios XI 5.6.9：認証された任意のコマンド実行の脆弱性 (CVE-2019-20197)
999698	CVE-2020-8417	2.14.0 より前の WEB-WORDPRESS コードスニペットプラグイン-CSRF の脆弱性 (CVE-2020-8417)
999699		バージョン 1.4.8 より前の WEB-WORDPRESS WPCentral プラグイン-権限昇格の脆弱性
999700	CVE-2020-8596	1.9.5.6 より前の WEB-WORDPRESS 参加者データベースプラグイン-認証された SQL インジェクションの脆弱性 (CVE-2020-8596)

署名ルール	**CVE ID**	説明
999701	CVE-2020-8426	2.8.5 より前の WEB-WORDPRESS Elementor ページビルダープラグイン-認証済みリフレクションクロスサイトスクリプティングの脆弱性（CVE-2020-8426）
999702	CVE-2019-19509	ウェブその他 rConfig 3.9.3: ajaxArchiveFiles.php 経由でリモートでコードが実行される脆弱性（CVE-2019-19509）
999703	CVE-2019-8449	WEB-MISC 8.4.0 より前の Atlassian Jira サーバー-情報漏えいの脆弱性 (CVE-2019-8449)
999704	CVE-2019-9194	2.1.48 より前の WEB-MISC elFinder-PHP コネクタコマンドインジェクションの脆弱性（CVE-2019-9194）
999705	CVE-2019-15985	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ-GetvmHostData を介した SQL インジェクションの脆弱性（CVE-2019-15985）
999706	CVE-2020-8549	2.40.1 より前の WEB-WORDPRESS 強力なお客様の声プラグイン-ストアドクロスサイトスクリプティングの脆弱性（CVE-2020-8549）

署名更新バージョン **42**

October 25, 2023

2020-02-11 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999707		バージョン 1.4.8 より前の WEB-WORDPRESS WPCentral プラグイン-権限昇格の脆弱性
999708	CVE-2019-15979	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：コマンドインジェクションの脆弱性（CVE-2019-15979）
999709	CVE-2019-15978	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：コマンドインジェクションの脆弱性（CVE-2019-15978）
999710	CVE-2019-15975	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：認証バイパスの脆弱性 (CVE-2019-15975)
999711	CVE-2019-15976	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：認証バイパスの脆弱性 (CVE-2019-15976)
999712	CVE-2019-16405	バージョン 19.10.2 より前の WEB-MISC Centreon：リモートでコードが実行される脆弱性 (CVE-2019-16405)

署名ルール	**CVE ID**	説明
999713	CVE-2020-7048	WEB-WORDPRESS WP データベースリセットプラグイン 3.1 まで-認証されていないデータベーステーブルのリセットの脆弱性（CVE-2020-7048）
999714	CVE-2020-7108	バージョン 3.1.2 より前の WEB-WORDPRESS LearnDash プラグイン-反射型クロスサイトスクリプティングの脆弱性（CVE-2020-7108）
999715	CVE-2019-15977	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：認証バイパスの脆弱性（CVE-2019-15977）
999716	CVE-2020-2096	WEB-MISC Jenkins Gitlab Hook プラグインバージョン 1.4.2 以前-クロスサイトスクリプティングの脆弱性（CVE-2020-2096）

署名更新バージョン **41**

August 15, 2023

2020-02-04 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。シグニチャアップデートには、シグニチャ ID、シグニチャバージョン、およびアドレス指定された CVE のリストが含まれます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> シグニチャアップデートバージョン 41 には、不正なシグニチャルール 1861 に対する修正が含まれています。投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999717		WEB-WORDPRESS WordPress バージョン 5.3.x 以前-xmlrpc.php pingback.ping メソッドによるサービス拒否の脆弱性
999718		1.21.16 より前の WP Time Capsule プラグインによる WEB-WORDPRESS バックアップとステージング-認証バイパスの脆弱性
999719	CVE-2019-19731	WEB-MISC .NET 1.4.5 用ロキシーファイルマン：RENAMEFILE によるパストラバーサルの脆弱性 (CVE-2019-19731)
999720	CVE-2019-19915	WEB-WORDPRESS 301 リダイレクト—2.4.0 までの簡単リダイレクトマネージャープラグイン-複数の脆弱性 (CVE-2019-19915)
999721	CVE-2019-17662	バージョン 1.0b1 より前の WEB-MISC Cybele ソフトウェア ThinVNC：ディレクトリトラバーサルの脆弱性（CVE-2019-17662）
999722	CVE-2020-6168	WEB-WORDPRESS 最小 2.17 より前のメンテナンスモードプラグイン-メンテナンス設定の脆弱性 (CVE-2020-6168)
999723	CVE-2020-6166	WEB-WORDPRESS 最小 2.17 より前のメンテナンスモードプラグイン-テーマ変更の脆弱性 (CVE-2020-6166)

署名ルール	**CVE ID**	説明
999724	CVE-2020-6166	WEB-WORDPRESS 最小 2.17 より前のメンテナンスモードプラグイン-エクスポート設定の脆弱性 (CVE-2020-6166)
999725		1.9.4.5 より前の WEB-WORDPRESS iniFiniteWP クライアントプラグイン-認証バイパスの脆弱性
999726	CVE-2019-16773	5.3.1 より前のバージョンの WEB-WORDPRESS WordPress-JSON オブジェクトを使用した REST API を介したクロスサイトスクリプティングの脆弱性 (CVE-2019-16773)
999727	CVE-2019-16773	5.3.1 より前のバージョンの WEB-WORDPRESS WordPress-フォームフィールドを使用した REST API を介したクロスサイトスクリプティングの脆弱性 (CVE-2019-16773)
999728	CVE-2019-16773	5.3.1 より前の WEB-WORDPRESS WordPress のバージョン: user-edit.php を介したクロスサイトスクリプティングの脆弱性 (CVE-2019-16773)
999729	CVE-2019-16773	5.3.1 より前の WEB-WORDPRESS WordPress のバージョン: profile.php を介したクロスサイトスクリプティングの脆弱性 (CVE-2019-16773)
999730	CVE-2019-16113	WEB-MISC Bludit 3.9.2: uuid を介したリモートでのイメージアップロードによるコードの実行の脆弱性 (CVE-2019-16113)

999731	CVE-2019-16113	WEB-MISC Bludit 3.9.2：ファイル名を介したイメージアップロードのリモートコード実行の脆弱性（CVE-2019-16113）

署名更新バージョン **40**

August 15, 2023

2020-01-14 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。シグニチャアップデートには、シグニチャ ID、シグニチャバージョン、およびアドレス指定された CVE のリストが含まれます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> シグニチャアップデートバージョン 40 には、不正なシグニチャルール 1861 に対する修正が含まれています。投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999732	CVE-2019-1620	WEB-MISC 11.2 (1) より前の Cisco データセンターネットワークマネージャ：任意のファイルアップロードの脆弱性（CVE-2019-1620）

署名ルール	**CVE ID**	説明
999733	CVE-2019-16702	WEB-MISC Integard Pro 2.2.0.9026：NoJS バッファオーバーフローの脆弱性 (CVE-2019-16702)
999734	CVE-2019-1621	WEB-MISC 11.2 (1) より前の Cisco データセンターネットワークマネージャ：任意のファイルダウンロードの脆弱性（CVE-2019-1621）
999735	CVE-2019-8451	WEB-MISC 8.4.0 より前の Atlassian Jira サーバー-サーバー側リクエストフォージェリの脆弱性 (CVE-2019-8451)
999736		4.0.3 より前の WEB-WORDPRESS GDPR クッキーコンプライアンスプラグイン-認証された任意の設定の削除の脆弱性
999737	CVE-2019-11287	3.7.21 より前の WEB-MISC Pivotal RabbitMQ 3.7.x および 3.8.1 より前の 3.8.x-サービス拒否の脆弱性（CVE-2019-11287）
999738		WEB-WORDPRESS 1.20.1 より前の Elementor の究極のアドオン-Facebook ログインによる認証バイパスの脆弱性
999739		WEB-WORDPRESS 1.20.1 より前の Elementor の究極のアドオン-Google ログインによる認証バイパスの脆弱性
999740	CVE-2019-19366	4.4.10 より前の WEB-MISC FusionPBX：リダイレクトパラメータによる xml_cdr_search.php のクロスサイトスクリプティングの脆弱性（CVE-2019-19366）
999741	CVE-2019-16931	バージョン 3.3.1 より前の WEB-WORDPRESS ビジュアライザープラグイン-認証されていないクロスサイトスクリプティングの脆弱性（CVE-2019-16931）

署名ルール	**CVE ID**	説明
999742	CVE-2019-16932	バージョン 3.3.1 より前の WEB-WORDPRESS ビジュアライザープラグイン-非認証 SSRF (CVE-2019-16932)
999743	CVE-2019-1619	WEB-MISC 11.1 (1) より前の Cisco データセンターネットワークマネージャ：認証バイパスの脆弱性 (CVE-2019-1619)
999744	CVE-2019-12562	WEB-MISC 9.4.0 より前の DotnetNuke：ストアドクロスサイトスクリプティングの脆弱性 (CVE-2019-12562)
999745	CVE-2019-8371	5.0.2 より前の WEB-MISC OpenEMR-Form_Filedata フィールドを介したリモートコード実行の脆弱性（CVE-2019-8371）
999746	CVE-2019-8371	5.0.2 より前の WEB-MISC OpenEMR-Form_Image フィールドを介したリモートコード実行の脆弱性（CVE-2019-8371）
999747		WEB-WORDPRESS Beaver Builder Ultimate Addons 1.24.1 より前-Facebook ログインによる認証バイパスの脆弱性
999748		WEB-WORDPRESS Beaver Builder Ultimate Addons 1.24.1 より前-Google ログインによる認証バイパスの脆弱性
999749	CVE-2019-19650	WEB-MISC Zoho ManageEngine AM ビルド前 13640-エージェントサーブレット経由の SQLi (CVE-2019-19650)
999750		WEB-MISC Zoho ManageEngine AM ビルド前 13620-opmRequestHandlerServlet サーブレットを介した API キーの開示

署名ルール	**CVE ID**	説明
999751	CVE-2019-1622	WEB-MISC Cisco データセンターネットワークマネージャ 11.0 (1): 情報開示の脆弱性 (CVE-2019-1622)
999752	CVE-2019-16759	5.5.4 パッチレベル 1 より前の WEB-MISC vBulletin-リモートでコードが実行される脆弱性 (CVE-2019-16759)
999753		2.7.8 より前の URL プラグインからの WEB-WORDPRESS の注目画像-REST API の脆弱性でアクセス制御が行われ
999754	CVE-2019-10098	ウェブその他 Apache HTTP サーバ 2.4.39 まで-mod_rewrite 自己参照リダイレクトの脆弱性 (CVE-2019-10098)
999755	CVE-2019-1936	WEB-MISC Cisco UCS Director 6.0 ～6.6.1.0 および 6.7.0.0 ～ 6.7.1.0: コマンドインジェクションの脆弱性（CVE-2019-1936）
999756	CVE-2019-19649	WEB-MISC ビルド前の Zoho ManageEngine AM 13620-EventID パラメーターによる認証されていない SQLi (CVE-2019-19649)
999757	CVE-2019-19649	WEB-MISC ビルド 13620 より前の Zoho ManageEngine AM-エンティティパラメータによる認証されていない SQLi（CVE-2019-19649）
999758	CVE-2019-15036	WEB-MISC JetBrains 2019.1 より前のチームシティ： OS コマンドインジェクションの脆弱性 (CVE-2019-15036)

999759	CVE-2019-17239	WEB-WORDPRESS ダッシュボードプラグインからプラグインとテーマをダウンロードする最大 1.5-ストアドクロスサイトスクリプティングの脆弱性（CVE-2019-17239）

署名更新バージョン **39**

October 25, 2023

2019-12-19 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999760		4.4.7 および 4.5.5 より前のバージョンの WEB-MISC FusionPBX-/app/exec/exec.php を介したリモートでのコード実行の脆弱性

署名ルール	**CVE ID**	説明
999761	CVE-2019-12747	8.7.27 および 9.5.8 より前の WEB-MISC Typo3-信頼できないデータのデシリアライズ (CVE-2019-12747)
999762	CVE-2019-13608	WEB-MISC Citrix StoreFront サーバー-XML 外部エンティティインジェクションの脆弱性 (CVE-2019-13608)
999763		5.2.4 より前の WEB-WORDPRESS WordPress-フォームを介したプライベートまたはドラフト投稿/ページの脆弱性の認証されていない表示
999764		5.2.4 より前の WEB-WORDPRESS WordPress-URL を介したプライベートまたはドラフト投稿/ページの脆弱性の認証されていない表示
999765	CVE-2019-15954	ウェブその他 Total.js CMS 12.0.0-JSON を介したウィジェット JavaScript コードインジェクションの脆弱性（CVE-2019-15954）
999766	CVE-2019-15954	WEB-MISC Total.js CMS 12.0.0-フォームを介したウィジェット JavaScript コードインジェクションの脆弱性（CVE-2019-15954）
999767		5.3.1 より前の WEB-WORDPRESS SyntaxHighlighter Evolved プラグイン-コメントによる保存型クロスサイトスクリプティング脆弱性
999768		5.3.1 より前の WEB-WORDPRESS SyntaxHighlighter 進化型プラグイン-POST 経由でのストアドクロスサイトスクリプティング
999769		5.3.1 より前の WEB-WORDPRESS SyntaxHighlighter 進化型プラグイン-JSON 経由で保存されたクロスサイトスクリプティング脆弱性

署名ルール	**CVE ID**	説明
999770	CVE-2019-16120	4.10.7.2 より前の WEB-WORDPRESS イベントチケットプラグイン：CSV インジェクションの脆弱性（CVE-2019-16120）
999771	CVE-2019-15029	4.4.8 より前の WEB-MISC FusionPBX：リモートでコードが実行される脆弱性（CVE-2019-15029）
999772		WEB-WORDPRESS Sassy ソーシャル共有プラグイン 3.3.4 より前-認証されていないクロスサイトスクリプティングの脆弱性
999773		WEB-WORDPRESS メール購読者およびニュースレタープラグインバージョン 4.3.1 以前-認証されていないブラインド SQLi の脆弱性
999774	CVE-2019-3398	WEB-MISC Atlassian Confluence またはデータセンター-すべての添付ファイルをダウンロードするパストラバーサルの脆弱性（CVE-2019-3398）
999775	CVE-2019-15952	WEB-MISC Total.js CMS 12.0.0：ページテンプレートパストラバーサルの脆弱性（CVE-2019-15952）
999776	CVE-2019-17236	WEB-WORDPRESS IgniteUp 近日公開およびメンテナンスモードプラグイン 3.4.0 まで-ストアドクロスサイトスクリプティング (CVE-2019-17236)
999777	CVE-2019-10475	WEB-MISC Jenkins ビルドメトリクスプラグイン 1.3-反射型クロスサイトスクリプティングの脆弱性（CVE-2019-10475）
999778	CVE-2019-17132	5.5.4 より前の WEB-MISC vBulletin パッチレベル 2-UpdateAvatar API エンドポイントのリモートコード実行の脆弱性（CVE-2019-17132）

署名ルール	**CVE ID**	説明
999779	CVE-2019-14994	WEB-MISC Atlassian Jira サービスデスク-パストラバーサルの脆弱性 (CVE-2019-14994)
999780	CVE-2019-19367	WEB-MISC FusionPBX 4.4.1 およびそれ以前-クロスサイトスクリプティングの脆弱性 (CVE-2019-19367)
999781	CVE-2019-18668	2.11.2 より前の WEB-WORDPRESS 通貨スイッチャープラグイン-POST による通貨設定バイパスの脆弱性 (CVE-2019-18668)
999782	CVE-2019-18668	2.11.2 より前の WEB-WORDPRESS 通貨スイッチャープラグイン-GET 経由での通貨設定バイパスの脆弱性 (CVE-2019-18668)
999783	CVE-2019-16663	WEB-MISC rConfig 3.9.2 以前-Search.crud.php 経由でリモートでコードが実行される脆弱性 (CVE-2019-16663)
999784		WEB-MISC Apache Solr 8.3.0 まで-VelocityResponseWriter カスタムテンプレートによる認証されていないリモートコードの実行
999785	CVE-2019-17235	WEB-WORDPRESS IgniteUp 近日公開とメンテナンスモードプラグイン 3.4.0 まで-Csv による情報開示 (CVE-2019-17235)
999786	CVE-2019-17235	WEB-WORDPRESS IgniteUp 近日公開とメンテナンスモードプラグイン 3.4.0 まで-BCC を介した情報開示 (CVE-2019-17235)
999787	CVE-2019-12276	WEB-MISC GrandNode 4.40-letsEncryptController パストラバーサルの脆弱性 (CVE-2019-12276)

署名ルール	**CVE ID**	説明
999788		バージョン 4.2.3 より前の WEB-WORDPRESS メール購読者とニュースレタープラグイン-認証されていない情報開示
999789	CVE-2019-4013	WEB-MISC IBM BigFix プラットフォーム 9.5-root 権限による認証済み任意のファイルアップロード (CVE-2019-4013)
999790	CVE-2019-11409	WEB-MISC FusionPBX バージョン 4.4.3 およびそれ以前-/app/basic_operator_panel/exec.php を介したリモートコードの実行 (CVE-2019-11409)
999791	CVE-2019-11409	WEB-MISC FusionPBX バージョン 4.4.3 およびそれ以前-/app/operator_panel/exec.php を介したリモートコードの実行 (CVE-2019-11409)
999792	CVE-2019-16662	WEB-MISC rConfig 3.9.2 およびそれ以前-AjaxServerSettingsChk.php を介した認証されていないリモートコードの実行 (CVE-2019-16662)
999793	CVE-2019-7609	5.6.15 および 6.6.1 より前の WEB-MISC Elastic Kibana: プロトタイプの汚染の脆弱性により認証されていない RCE が許可される (CVE-2019-7609)
999794	CVE-2019-10092	GrandNode Apache HTTP サーバー 2.4.39 まで-mod_proxy リミテッドクロスサイトスクリプティング (CVE-2019-10092)
999795	CVE-2019-16520	WEB-WORDPRESS 3.2.7 より前のオールインワン SEO パックプラグイン-ストアドクロスサイトスクリプティングの脆弱性 (CVE-2019-16520)

署名ルール	**CVE ID**	説明
999796	CVE-2019-17234	WEB-WORDPRESS IgniteUp 近日公開とメンテナンスモードプラグイン 3.4.0 まで-任意のファイルの削除 (CVE-2019-17234)
999797	CVE-2019-16525	バージョン 1.1.9 より前の WEB-WORDPRESS チェックリストプラグイン-クロスサイトスクリプティングの脆弱性 (CVE-2019-16525)
999798		1.9.6 より前の WEB-WORDPRESS Safe SVG プラグイン-クロスサイトスクリプティングの脆弱性
999799		バージョン 4.2.3 より前の WEB-WORDPRESS メール購読者とニュースレターのプラグイン-認証されていない任意のオプションの作成

署名更新バージョン **38**

August 15, 2023

バージョン 38 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999800	CVE-2019-12517	WEB-WORDPRESS SlickQuiz プラグインバージョン 1.3.7.1 以前-クロスサイトスクリプティングの脆弱性（CVE-2019-12517）
999801	CVE-2019-10392	WEB-MISC Jenkins Git クライアントプラグイン 2.8.4 以前-OS コマンドインジェクションの脆弱性（CVE-2019-10392）
999802	CVE-2019-8371	5.0.2 より前の WEB-MISC OpenEMR-Form_Filedata フィールドを介したリモートコード実行の脆弱性（CVE-2019-8371）
999803	CVE-2019-8371	5.0.2 より前の WEB-MISC OpenEMR-Form_Image フィールドを介したリモートコード実行の脆弱性（CVE-2019-8371）
999804	CVE-2019-12516	WEB-WORDPRESS SlickQuiz プラグインバージョン 1.3.7.1 以前-SQL インジェクションの脆弱性（CVE-2019-12516）
999805	CVE-2019-1262	WEB-MISC Microsoft SharePoint サーバー-クロスサイトスクリプティングの脆弱性 (CVE-2019-1262)

署名更新バージョン **37**

August 15, 2023

バージョン 37 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999806	CVE-2019-3394	WEB-MISC Atlassian Confluence またはデータセンター-ローカルファイル開示の脆弱性 (CVE-2019-3394)
999807	CVE-2019-13569	WEB-WORDPRESS Icegram 4.1.8 より前のメール購読者とニュースレタープラグイン-ESFPX_lists パラメータ経由の SQLi (CVE-2019-13569)
999808	CVE-2019-13569	WEB-WORDPRESS Icegram メール購読者＆ニュースレタープラグイン 4.1.8 より前-SQLi Via Order Param (CVE-2019-13569)
999809	CVE-2019-2768	WEB-MISC Oracle BI Publisher -予測可能なセッショントークンの脆弱性（CVE-2019-2768）
999810	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy プラグイン最大 2.61-ジョブ更新によるサンドボックスバイパスの脆弱性（CVE-2019-1003001）
999811	CVE-2019-13575	WEB-WORDPRESS wpEverest Everest Forms プラグイン 1.5.0 より前-SQL インジェクション (CVE-2019-13575)

署名ルール	**CVE ID**	説明
999812	CVE-2019-15896	WEB-WORDPRESS LifterLMS プラグイン 3.34.5 まで-セキュリティバイパスの脆弱性 (CVE-2019-15896)
999813	CVE-2019-3396	WEB-MISC アトラシアン Confluence またはデータセンター-リモートでコードが実行される脆弱性（CVE-2019-3396）
999814	CVE-2019-5475	2.14.14 より前の WEB-MISC Sonatype Nexus リポジトリマネージャー-Createrepo パスを介したリモートコード実行 (CVE-2019-5475)
999815	CVE-2019-5475	WEB-MISC 2.14.14 より前の Sonatype Nexus リポジトリマネージャー-Mergerepo パスを介したリモートコード実行 (CVE-2019-5475)
999816	CVE-2019-15104	WEB-MISC Zoho ManageEngine 12.4 より前のバージョンの OpManager-SQL インジェクションの脆弱性（CVE-2019-15104）

署名更新バージョン **36**

August 15, 2023

バージョン 36 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。シグニチャルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999817		バージョン 2.4.22 より前の WEB-WORDPRESS WordPress 広告インサータープラグイン-リモートコード実行
999818	CVE-2019-7839	WEB-MISC Adobe ColdFusion 複数バージョン-HTTP/SOAP ドットネットから Java へのリモートコード実行の脆弱性（CVE-2019-7839）
999819	CVE-2019-7839	WEB-MISC Adobe ColdFusion 複数バージョン-HTTP/SOAP 経由でのリモートコード実行の脆弱性 Java-to-dotNet (CVE-2019-7839)
999820	CVE-2019-11469	WEB-MISC 14150 ビルドより前の Zoho ManageEngine アプリケーションマネージャーで resourceid パラメーター経由で SQLi を許可する (CVE-2019-11469)
999821	CVE-2019-11448	WEB-MISC Zoho ManageEngine アプリケーションマネージャー 11.0 から 14.0-非認証 SQL インジェクション (CVE-2019-11448)
999822	CVE-2019-1003000	WEB-MISC Jenkins スクリプトセキュリティプラグイン 1.49 まで: サンドボックスバイパスの脆弱性 (CVE-2019-1003000)

署名ルール	**CVE ID**	説明
999823		WEB-WORDPRESS WordPress Cforms2 プラグイン 15.0.1 まで-認証されていない HTML インジェクションの脆弱性
999824	CVE-2019-0193	8.2 より前の WEB-MISC Apache Solr-DataConfig パラメーターを介した DIH リモートコード実行の脆弱性（CVE-2019-0193）
999825	CVE-2019-11580	WEB-MISC Atlassian Crowd Pdkinstall 開発プラグインが有効-認証されていない RCE (CVE-2019-11580)
999826	CVE-2019-0192	WEB-MISC Apache Solr 5.5.5/6.6.5 まで-設定 API リモートでコードが実行される脆弱性 (CVE-2019-0192)
999827		WEB-WORDPRESS WooCommerce バリエーションスウォッチプラグイン 1.0.61 まで-反射型クロスサイトスクリプティングの脆弱性
999828	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy プラグイン最大 2.61-ジョブ作成によるサンドボックスバイパスの脆弱性（CVE-2019-1003001）
999829	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy プラグイン最大 2.61-サンドボックスバイパスの脆弱性 (CVE-2019-1003001)
999830		2.3.2 より前の WEB-WORDPRESS WordPress 太字ページビルダープラグイン-セキュリティバイパスの脆弱
999831	CVE-2019-15107	1.930 より前の WEB-MISC Webmin：認証されていないリモートコード実行の脆弱性 (CVE-2019-15107)

署名ルール	**CVE ID**	説明
999832	CVE-2019-2767	ウェブその他 Oracle BI Publisher 11.1.1.9.0 および 12.2.1.4-XXE 脆弱性（CVE-2019-2767）
999833	CVE-2019-15106	WEB-MISC Zoho ManageEngine OpManager から 12.4x まで-認証バイパスの脆弱性 (CVE-2019-15106)
999948	CVE-2014-0114	Apache Struts 1 から 1.3.10 までは、HTTP_FORM_FIELD 経由で任意のコードを実行できるように ClassLoader の操作が可能です
999949	CVE-2013-4316	2.3.15.2 より前の Apache Struts 2 では、機密性、整合性、可用性に影響を与えることで動的メソッド呼び出しが可能になります
999950	CVE-2013-4316	2.3.15.2 より前の Apache Struts 2 では、機密性、整合性、可用性に影響を与えることで動的メソッド呼び出しが可能になります

> 注:
>
> パフォーマンスの問題により、シグニチャルール 999947 が削除されました。

署名更新バージョン **35**

August 15, 2023

バージョン 35 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999834	CVE-2019-13024	WEB-MISC Centreon バージョン 19.04 以前-コマンドインジェクションの脆弱性
999835	CVE-2019-5420	WEB-MISC Rails 開発モード-シークレットトークン漏洩の脆弱性
999836	CVE-2019-5418	WEB-MISC Rails アクションビュー-ファイルコンテンツ漏洩の脆弱性
999837	CVE-2018-12426, CVE-2019-11185	WEB-WORDPRESS WP ライブチャットサポート Pro プラグイン 8.0.26 より前-任意のファイルのアップロード
999838	CVE-2019-10270	WEB-WORDPRESS WordPress プラグインバージョン 2.0.40 より前のアルティメットメンバー-任意のパスワード
999839	CVE-2019-12826	5.10.2 より前の WEB-WORDPRESS WordPress プラグインウィジェットロジック-CSRF の脆弱性
999840		WEB-WORDPRESS WordPress プラグイン 2.5.39 より前のオールインワンイベントカレンダー-クロスサイトスクリプティングの脆弱性
999841	CVE-2019-11565	WEB-WORDPRESS WordPress プラグイン 1.6.7 より前のブログを印刷する-認証されていない SSRF の脆弱性

署名ルール	CVE ID	説明
999842		WEB-WORDPRESS WordPress のプラグインバージョン 2.0.46 より前の究極のメンバー-複数 `cross-site scripting </LogString`

署名更新バージョン 34

August 15, 2023

バージョン 34 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ (CVE) の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	CVE ID	説明
999843		WEB-WORDPRESS WordPress プラグインバージョン 2.0.46 より前のアルティメットメンバー-任意のファイルを読み込む設定

署名ルール	**CVE ID**	説明
999844		WEB-WORDPRESS WordPress プラグインバージョン 2.0.46 より前のアルティメットメンバー-任意のファイルを読み
999845		WEB-WORDPRESS WordPress プラグインバージョン 2.0.46 より前の究極のメンバー-ファイル置換によるファイル削除
999846		WEB-WORDPRESS WordPress のプラグインバージョン 2.0.46 より前の究極のメンバー-ファイルの削除
999847		2.1.10 より前の WEB-WORDPRESS WordPress プラグインショートリンク-CSV インジェクションの脆弱性
999848		WEB-WORDPRESS WordPress プラグイン 2.1.10 より前のショートリンク-認証されていないストアドクロスサイトスクリプティングの脆弱性
999849		7.3.13.727 より前の WEB-WORDPRESS WordPress プラグイン FV Flowplayer ビデオプレーヤー-認証されていないストアドクロスサイトスクリプティングの脆弱性
999850		WEB-WORDPRESS WordPress プラグイン 2.9.16 より前の簡単なデジタルダウンロード-認証されていないストアドクロスサイトスクリプティングの脆弱性
999851		WEB-WORDPRESS WordPress プラグイン Crelly スライダーバージョン 1.3.5 より前-任意のファイルアップロードの脆弱性
999853	CVE-2019-2615	WEB-MISC Oracle WebLogic サーバ情報漏えいの脆弱性

署名ルール	**CVE ID**	説明
999854	CVE-2019-11872	WordPress プラグイン Hustle 6.0.8.1 より前-CSV インジェクションの脆弱性
999855	CVE-2019-11231	WEB-MISC GetSimple CMS バージョン 3.3.15 以前-任意のファイルアップロードの脆弱性
999856	CVE-2019-11231	WEB-MISC GetSimple CMS バージョン 3.3.15 およびそれ以前-API キー情報の開示
999857		5.2 より前の WEB-WORDPRESS WordPress プラグイン WP データベースバックアップ-コマンドインジェクションの脆弱性
999858		WEB-WORDPRESS WordPress プラグインスリックポップアップ 1.7.1 まで-権限昇格の脆弱性
999859	CVE-2019-12099	WEB-MISC PHP Fusion CMS バージョン 9.03.00 以前でリモートでコードが実行される脆弱性

署名更新バージョン **33**

August 15, 2023

バージョン 33 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

規則	**CVE**	説明	脆弱性リファレンス
999860		WordPress プラグイン Yuzo 関連記事クロスサイトスクリプティング脆弱性	https://www.wordfence.com/blog/2019/04/yuzo-related-posts-zero-day-vulnerability-exploited-in-the-wild
999861	CVE-2019-12099		cve,2019-12099
999862		WordPress プラグインデータベースバックアップ <= 5.2-リモートコード実行	https://www.wordfence.com/blog/2019/05/os-command-injection-vulnerability-patched-in-wp-database-backup-plug-in
999863		WordPress プラグイン Slick Popup-権限昇格	https://www.wordfence.com/blog/2019/05/privilege-escalation-flaw-present-in-slick-popup-plug-in
999864	CVE-2019-10866	WordPress プラグイン Form Maker 1.13.3-SQL インジェクション	cve,2019-10866

規則	CVE	説明	脆弱性リファレンス
999865		WordPress プラグイン Give ―ドナー用に格納されたクロスサイトスクリプティング	https://blog.sucuri.net/2019/05/wordpress-plug-in-give-stored-xss-for-donors.html
999866		WordPress プラグインマイカレンダー <= 3.1.9-認証されていないクロスサイトスクリプティングの脆弱性	https://wpvulndb.com/vulnerabilities/9267
999867		WordPress プラグイン Slimstat <= 4.8-認証されていないストアドクロスサイトスクリプティング	https://blog.sucuri.net/2019/05/slimstat-stored-xss-from-visitors.html
999868	CVE-2019-2618	WebLogic 任意アップロードの脆弱性	cve,2019-2618
999869	CVE-2019-11871	2.5.15 より前の WEB-WORDPRESS WordPress プラグインのカスタムフィールドスイート-クロスサイトスクリプティングの脆弱性	cve,2019-11871
999870		WEB-WORDPRESS WordPress ライブチャットサポートプラグイン永続的なクロスサイトスクリプティング 8.0.27 より前の wplc_custom_js パラメータによる脆弱性	https://blog.sucuri.net/2019/05/persistent-cross-site-scripting-in-wp-live-chat-support-plug-in.html

規則	CVE	説明	脆弱性リファレンス
999871		0.9.7.4 より前の WEB-WORDPRESS WordPress プラグイン W3 合計キャッシュ-PHAR リモートでコードが実行される脆弱性	https://wpvulndb.com/vulnerabilities/9270
999872		0.9.7.4 より前の WEB-WORDPRESS WordPress プラグイン W3 合計キャッシュ-PHAR リモートでコードが実行される脆弱性	https://wpvulndb.com/vulnerabilities/9269
999873	CVE-2019-0604	WEB-MISC Microsoft Windows Sharepoint サーバー-リモートでコードが実行される脆弱性	cve,2019-0604
999874		WEB-WORDPRESS 雄三関連記事 5.12.91 における認証されていないストアドクロスサイトスクリプティングの脆弱性	https://www.wordfence.com/blog/2019/04/yuzo-related-posts-zero-day-vulnerability-exploited-in-the-wild

署名更新バージョン **32**

August 15, 2023

バージョン 32 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999875	CVE-2016-4438, CVE-2016-3087	WEB-STRUTS Apache Struts 2.3.20 から 2.3.28.1 URL 経由でのリモート実行の脆弱性
999876	CVE-2019-10867	5.7.1 より前の WEB-MISC Pimcore-デシリアライズの脆弱性 (CVE-2019-10867)

署名更新バージョン **30**

August 15, 2023

バージョン 30 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999879	<>	WEB-MISC WordPress プラグイン WooCommerce チェックアウトマネージャー-任意のファイルアップロードの脆弱性
999880	<>	WEB-MISC WordPress プラグイン アドバンスお問い合わせフォーム 7 DB 1.6.1 より前-SQL インジェクションの脆弱性
999881	<>	WEB-MISC WordPress プラグイン 1.0.67 より前のお問い合わせフォームビルダー-ローカルファイルインクルードの脆弱性
999882	<>	SQL HTTP URI ブラインドインジェクション試行
999883	<>	WEB-MISC Loco Translate WordPress プラグイン 2.1.1 以前-ローカルファイルインクルージョンの脆弱性
999884	<>	3.4 より前の WEB-MISC WordPress プラグインの重複ページ-SQL インジェクションの脆弱性
999885	CVE-2019-0232	MS Windows で enableCmdLineArguments=true の場合 WEB-MISC Apache Tomcat RCE Via .CMD CGI スクリプト
999886	CVE-2019-0232	WEB-MISC Apache Tomcat RCE 経由.BAT 経由 CGI スクリプト有効にすると CMdlineArguments=True (MS Windows の場合)

署名ルール	**CVE ID**	説明
999887	CVE-2019-10692	WWEB-MISC WordPress プラグイン wp-google-maps 7.11.18 より前-SQL インジェクションの脆弱性。
999888	CVE-2019-10946	WEB-MISC Joomla! 3.9.5 より前-セキュリティバイパスの脆弱性
999889	CVE-2019-10945	WEB-MISC Joomla! 3.9.5 より前: ディレクトリトラバーサルの脆弱性
999890	CVE-2019-9912	WEB-MISC WPGoogleMaps 7.10.41 より前の WordPress プラグイン反射型クロスサイトスクリプティングの脆弱性
999890	CVE-2019-9912	WEB-MISC WPGoogleMaps 7.10.41 より前の WordPress プラグイン反射型クロスサイトスクリプティングの脆弱性
999891	CVE-2019-9911	WEB-MISC WordPress プラグイン ソーシャルネットワーク 4.2.8 より前の自動ポスター-反射型クロスサイトスクリプティングの脆弱性
999892	CVE-2019-9908	WEB-MISC WordPress プラグイン Font_Organizer 2.1.1-反射型クロスサイトスクリプティング
999893	CVE-2019-9787	4.9.7 より前の WEB-MISC WordPress-リモートでコードが実行される脆弱性
999894	CVE-2019-9568	WEB-MISC Forminator お問い合わせフォーム、投票、クイズビルダー WordPress プラグイン 1.6 より前のブラインド SQLi 脆弱性
999895	CVE-2019-9567	WEB-MISC Forminator お問い合わせフォーム、投票、クイズビルダー 1.6 より前の WP プラグイン永続的なクロスサイトスクリプティングの脆弱性
999877	CVE-2018-20062	WEB-MISC noneCMS V1.3-ThinkPHP フィルタ任意の PHP コード実行の脆弱性

署名ルール	**CVE ID**	説明
999878	CVE-2019-9082	5.1.32 より前の ThinkPHP 5.x での WEB-MISC リモートコード実行の脆弱性

署名更新バージョン **29**

August 15, 2023

バージョン 29 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

注:

> 投稿本文と応答本文の署名ルールを有効にすると、NetScaler CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999896	CVE-2019-2725	Weblogic 10.3.6 リモートでのコード実行
999897	CVE-2019-2725	Weblogic 10.3.6 リモートでのコード実行

署名更新バージョン **28**

August 15, 2023

バージョン 28 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。シグニチャアップデートには、シグニチャ ID、シグニチャバージョン、およびアドレス指定された CVE のリストが含まれます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999898	CVE-2018-12895	4.9.7 より前の WEB-MISC WordPress ディレクトリトラバーサルの脆弱性。
999899	CVE-2019-9618	Web-MISC-GraceMedia メディアプレーヤー WordPress プラグイン 1.0 任意のローカルファイルインクルード脆弱性
999900	CVE-2018-20714	WEB-MISC WordPress プラグイン WooCommerce 3.4.6 より前-ファイル削除の脆弱性。
999901	CVE-2018-11868	2.3.7 より前の WEB-MISC FlowPaper FlexPaper では、リモートでのコード実行-設定ファイルのリセットを許可できます。
999902	CVE-2018-11868	2.3.7 より前の WEB-MISC FlowPaper FlexPaper では、リモートコード実行を許可できます

署名ルール	**CVE ID**	説明
999903	CVE-2019-9184	ウェブその他 Joomla! 3.3.7 より前の J2Store プラグイン 3.x は SQL インジェクションを可能にします。
999904	CVE-2019-9168	WEB-MISC WordPress プラグイン WooCommerce Photoswipe キャプションを介して 3.5.5 クロスサイトスクリプティングの前に。
999905		WEB-MISC WordPress プラグイン WooCommerce で保存されたクロスサイトスクリプティング用の 5.1.3 より前の放棄されたカート。
999906	CVE-2019-8942	4.9.9 より前の WEB-MISC WordPress と 5.0.1 のリモートコード実行前の 5.x。
999907	CVE-2019-8942	4.9.9 より前の WEB-MISC WordPress と 5.0.1 のリモートコード実行前の 5.x。
999908	CVE-2019-8942	WEB-MISC 4.9.9 より前の WordPress と 5.0.1 より前の 5.x のリモートコード実行
999909	CVE-2017-16562	web-misc-デラックステーマ UserPro WordPress プラグイン up_auto_log=true パラメータによるセキュリティバイパスの脆弱性
999910	CVE-2018-20782	WEB-MISC WordPress プラグイン GloBee 1.1.2 より前 WooCommerce IPN メッセージスプーフィング
999911	CVE-2019-6340	Drupal-Drupal Core 8 RESTful Web サービスでの任意のリモートコード実行

署名更新バージョン **27**

August 15, 2023

バージョン 27 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。シグニチャアップデートには、シグニチャ ID、シグニチャバージョン、およびアドレス指定された CVE のリストが含まれます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

NetScaler バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999921	cve-2018-1002000	Web-miscWordPress Arigato オートレスポンダーとニュースレター SQL インジェクションの脆弱性。
999920		WEB-MISCWordPress プラグインコーナー広告 1.0.7-ストアドクロスサイトスクリプティング
999919	cve-2018-1002009	WEB-MISCWordPress Arigato 自動応答とニュースレター bft_unsubscribe クロスサイトスクリプティングの脆弱性。
999918	cve-2018-1002002	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999918	cve-2018-1002003	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。

署名ルール	**CVE ID**	説明
999918	cve-2018-1002004	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999918	cve-2018-1002005	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999918	cve-2018-1002006	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999918	cve-2018-1002007	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999917	cve-2018-1002001	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999917	cve-2018-1002008	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999916	cve-2018-8719	WEB-MISCWordPress プラグイン WP セキュリティ監査ログ-wp-content/uploads/wp-security 監査ログ/* 無制限アクセス
999915	cve-2019-7743	WEB-MISC-Joomla phar://ストリームラッパーオブジェクトインジェクションの脆弱性アップロードされた非 phar ファイルの実行
999914		WEB-MISCWordPress プラグイン電子メール購読者とニュースレター 3.4.7 情報漏えいの脆弱性

署名ルール	**CVE ID**	説明
999913		WEB-MISCWordPress プラグイン AD マネージャー WD v1.0.11-wd_ads_admin_class.php 任意のファイルのダウンロード
999912		Web-IISMicrosoft IIS-ショートファイル/フォルダ名の開示

ボット管理

August 15, 2023

着信ウェブトラフィックはボットで構成され、ほとんどの組織はボット攻撃に苦しむことがあります。Web およびモバイルアプリケーションはビジネスの重要な収益ドライバーであり、ほとんどの企業はボットなどの高度なサイバー攻撃の脅威にさらされています。

ボットは、人間よりもはるかに速い速度で特定のアクションを繰り返し自動的に実行するソフトウェアプログラムです。ボットは、Web ページの操作、フォームの送信、アクションの実行、テキストのスキャン、コンテンツのダウンロードを行うことができます。動画にアクセスしたり、コメントを投稿したり、ソーシャルメディアプラットフォームでツイートしたりできます。チャットボットと呼ばれる一部のボットは、人間のユーザーと基本的な会話をすることができます。

カスタマーサービス、自動チャット、検索エンジンのクローラーなど、役立つサービスを実行するボットは優れたボットです。同時に、ウェブサイトからコンテンツをスクラップまたはダウンロードしたり、ユーザーの資格情報、スパムコンテンツを盗んだり、他の種類のサイバー攻撃を実行したりできるボットは、悪いボットです。

悪意のあるタスクを実行する悪意のあるボットが数多くいるため、ボットのトラフィックを管理し、ボット攻撃からウェブアプリケーションを保護することが不可欠です。NetScaler のボット管理を使用すると、受信するボットトラフィックを検出し、ボット攻撃を軽減して Web アプリケーションを保護できます。

NetScaler のボット管理は、悪質なボットを特定し、高度なセキュリティ攻撃からアプライアンスを保護するのに役立ちます。良いボットと悪いボットを検出し、着信トラフィックがボット攻撃であるかどうかを識別します。ボット管理を使用することで、攻撃を軽減し、ウェブアプリケーションを保護できます。

NetScaler ボット管理には、次のような利点があります。

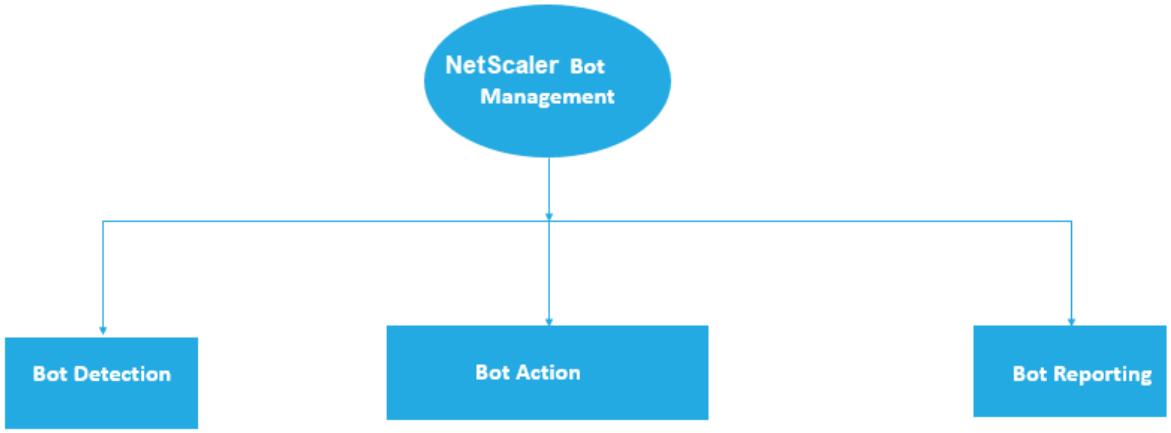
- ボット、スクリプト、ツールキットから身を守りましょう。静的シグネチャベースの防御とデバイスフィンガープリントを使用して、脅威をリアルタイムで軽減します。
- 自動化された基本攻撃と高度な攻撃を無力化します。アプリ層 DDoS、パスワードスプレー、パスワードスタッフィング、価格スクレーパー、コンテンツスクレーパーなどの攻撃を防ぎます。
- **API** と投資を保護しましょう。API を不当な悪用から保護し、自動トラフィックからインフラストラクチャへの投資を保護します。

NetScaler ボット管理システムを使用することでメリットが得られるユースケースには、次のようなものがあります。

- ブルートフォースログイン。政府のウェブポータルは、ブルートフォースユーザーログインを試みるボットによる攻撃を絶えず受けています。この組織は、Web ログを調べ、特定のユーザーが何度も何度も選択され、ログインの試行が速く、パスワードが辞書攻撃の手法で増えていくのを確認したことで、この攻撃を発見しました。法律により、彼らは彼ら自身と彼らのユーザーを保護しなければなりません。NetScaler のボット管理を導入することで、デバイスフィンガープリントやレート制限技術を使ったブルートフォースログインを阻止できます。
- 不良ボットとデバイスフィンガープリント不明ボットをブロックします。Web エンティティには 1 日あたり 100,000 人の訪問者が訪れます。基盤となるフットプリントをアップグレードする必要があり、多額の費用を費やしています。最近の監査で、チームはトラフィックの 40%がボット、コンテンツのスクレイピング、ニュースの選択、ユーザープロフィールの確認などによるものであることを発見しました。このトラフィックをブロックしてユーザーを保護し、ホスティングコストを削減したいと考えています。ボット管理を使用すると、既知の不正ボットをブロックしたり、サイトに攻撃を仕掛けている未知のボットをフィンガープリントしたりできます。これらのボットをブロックすることで、ボットのトラフィックを 90% 削減できます。

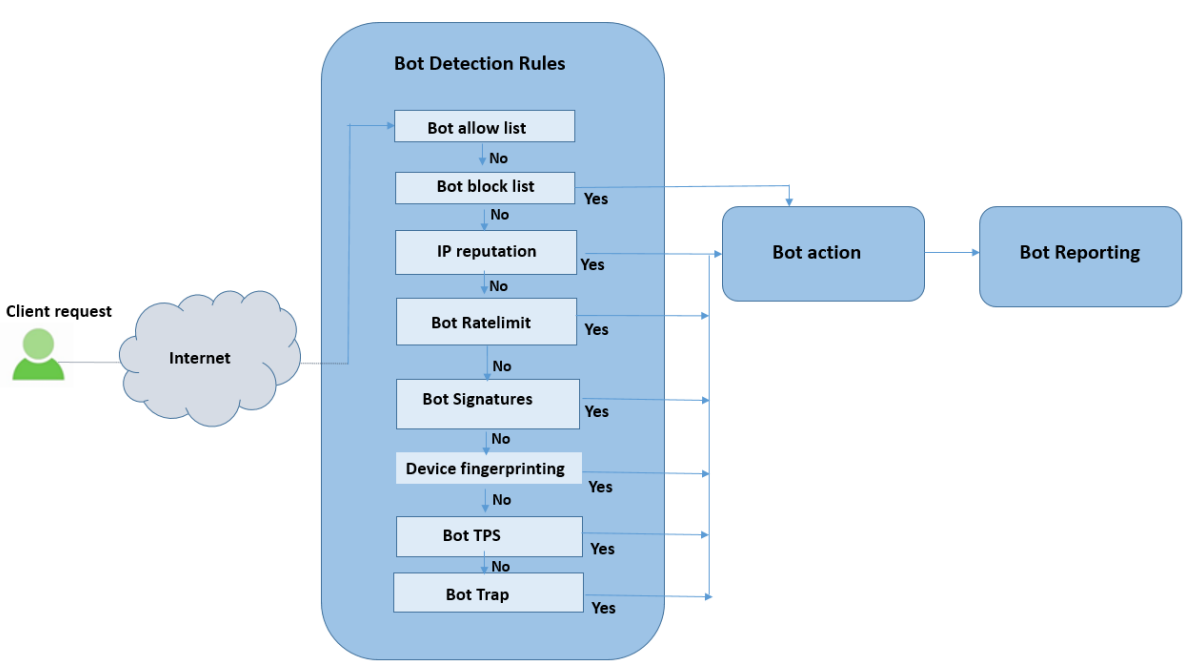
NetScaler のボット管理は何をしますか

NetScaler のボット管理は、組織が高度なセキュリティ攻撃から Web アプリケーションと公共資産を保護するのに役立ちます。着信トラフィックがボットの場合、次の図に示すように、ボット管理システムはボットタイプを検出し、アクションを割り当てて、ボットインサイトを生成します。



NetScaler のボット管理はどのように機能しますか

次の図は、NetScaler ADC ボット管理の仕組みを示しています。このプロセスには、着信トラフィックを良好または悪いボットとして検出するのに役立つ 8 つの検出技術が含まれます。デフォルトでは、シグニチャによって検出された良好なボットは許可され、シグニチャによって検出された不良ボットはドロップされます。



1. このプロセスでは、まずアプライアンスのボット管理機能を有効にします。
2. クライアントがリクエストを送信すると、アプライアンスはボットポリシールールを使用してトラフィックを評価します。受信リクエストがボットとして識別された場合、アプライアンスはボット検出プロファイルを適用します。
3. デフォルトまたはカスタムのボット署名ファイルをボット検出プロファイルにバインドする必要があります。ボット署名ファイルには、入ってくるボットタイプを識別するためのボット署名ルールのリストがあります。
4. ボット検出ルールは、シグニチャファイルの 8 つの検出カテゴリで使用できます。カテゴリは、許可リスト、ブロックリスト、静的シグニチャ、IP レピュテーション、デバイスフィンガープリント、およびレート制限です。ボットトラフィックに基づいて、システムはトラフィックに検出ルールを適用します。
5. 着信ボットトラフィックがボット許可リストのエントリと一致する場合、システムは他の検出手法をバイパスし、関連するアクションがデータをログに記録します。
6. ボット許可リスト以外の検出手法では、着信要求が設定されたルールと一致する場合、対応するアクションが適用されます。可能なアクションは、ドロップ、リダイレクト、リセット、軽減、およびログです。CAPTCHA は、IP レピュテーション、デバイスフィンガープリント、および TPS 検出技術でサポートされている緩和アクションです。

ボットの検出

March 20, 2024

NetScaler ボット管理システムは、さまざまな手法を使用して受信ボットトラフィックを検出します。この手法は、ボットタイプを検出するための検出ルールとして使用されます。テクニックは次のとおりです。

> 注:
>
> ボット管理では、ブロックリスト、許可リスト、およびレート制限の手法について、最大 32 個の構成エンティティがサポートされています。

ボット許可リスト -許可リストとしてバイパスできる IP アドレス (IPv4 と IPv6)、サブネット (IPv4 と IPv6)、およびポリシー表現のカスタマイズされたリスト。

ボットブロックリスト -ウェブアプリケーションへのアクセスをブロックする必要がある IP アドレス (IPv4 と IPv6)、サブネット (IPv4 と IPv6)、およびポリシー表現のカスタマイズされたリスト。

IP レピュテーション -このルールは、受信ボットトラフィックが悪質な IP アドレスからのものかどうかを検出します。

デバイスフィンガープリント -このルールは、受信ボットトラフィックの受信リクエストヘッダーにデバイスフィンガープリント ID と、受信クライアントボットトラフィックのブラウザ属性があるかどうかを検出します。

> 制限事項:
>
> 1. クライアントブラウザで JavaScript を有効にする必要があります。
> 2. XML 応答では機能しません。

ボットログ表現 -検出技術により、追加情報をログメッセージとしてキャプチャできます。データには、URL を要求したユーザーの名前、送信元 IP アドレス、およびユーザーが要求を送信した送信元ポート、または式から生成されたデータを指定できます。

レート制限 -このルールは、同じクライアントからの複数のリクエストをレート制限します。

ボットトラップ -クライアントの応答にトラップ URL をアドバタイズすることで、自動ボットを検出してブロックします。クライアントが人間のユーザーの場合、URL は見えず、アクセスできないように見えます。この検出技術は、自動ボットからの攻撃をブロックするのに効果的です。

TPS -最大リクエスト数とリクエストの増加率が設定された時間間隔を超えると、受信トラフィックをボットとして検出します。

CAPTCHA -このルールはボット攻撃を軽減するために CAPTCHA を使用します。CAPTCHA は、受信トラフィックが人間のユーザーからのものか、自動化されたボットからのものかを判断するためのチャレンジ/レスポンスの検証です。この検証は、Web アプリケーションにセキュリティ違反を引き起こす自動ボットをブロックするのに役立ちます。CAPTCHA は、IP レピュテーションおよびデバイスフィンガープリント検出技術のボットアクションとして設定できます。

それでは、ボットトラフィックを検出して管理するための各手法をどのように構成できるかを見てみましょう。

アプライアンスを **NetScaler CLI** ベースのボット管理構成にアップグレードする方法

アプライアンスを古いバージョン（NetScaler リリース 13.0 ビルド 58.32 以前）からアップグレードする場合は、最初に既存のボット管理構成を NetScaler CLI ベースのボット管理構成に手動で変換する必要があります。ボット管

理設定を手動で変換するには、次の手順を実行します。

1. 最新バージョンにアップグレードした後、次のコマンドを使用してアップグレードツール「upgrade_bot_config.py」に接続します。

   コマンドプロンプトで入力します:

   ```
   shell "/var/python/bin/python /netscaler/upgrade_bot_config.py > /var/bot_upgrade_commands.txt"
   ```

2. 次のコマンドを使用して、構成を実行します。

   コマンドプロンプトで入力します:

   ```
   batch -f /var/bot_upgrade_commands.txt
   ```

3. アップグレードした設定を保存します。

```
save ns config
```

NetScaler CLI ベースのボット管理の設定

ボット管理設定では、1 つ以上のボット検出手法を特定のボットプロファイルにバインドできます。
NetScaler ベースのボット管理を設定するには、次の手順を完了する必要があります。

1. ボット管理を有効にする
2. ボット署名のインポート
3. ボットプロファイルの追加
4. ボットプロファイルのバインド
5. ボットポリシーの追加
6. バインドボットポリシー
7. ボット設定の構成

> 注:
>
> アプライアンスを古いバージョンからアップグレードする場合は、最初に既存のボット管理構成を手動で変換する必要があります。詳しくは、「NetScaler CLI ベースのボット管理構成へのアップグレード方法」セクションを参照してください。

ボット管理を有効にする

開始する前に、アプライアンスでボット管理機能が有効になっていることを確認します。新しい NetScaler または VPX をお持ちの場合は、構成する前にその機能を有効にする必要があります。NetScaler アプライアンスを以前のバージョンから現在のバージョンにアップグレードする場合は、構成する前にこの機能を有効にする必要があります。
コマンドプロンプトで入力します:

```
enable ns feature Bot
```


ボット署名のインポート

デフォルトのシグニチャボットファイルをインポートし、ボットプロファイルにバインドできます。コマンドプロンプトで入力します:

```
import bot signature [<src>] <name> [-comment <string>] [-overwrite]
```

各項目の意味は次のとおりです:

src -ローカルパス名または URL (プロトコル、ホスト、パス、およびファイル名)。最大の長さ:2047 インチ。
> 注:
>
> インポートするオブジェクトが、アクセスにクライアント証明書認証を必要とする HTTPS サーバー上にある場合、インポートは失敗します。

name -ボット署名ファイルオブジェクトの名前。これは必須の議論です。最大長:31

comment -署名ファイルオブジェクトに関する説明。最大長:255

overwrite -既存のファイルを上書きするアクション。
> 注:
>
> overwriteオプションを使用して、署名ファイルの内容を更新します。または、update bot signature <name> コマンドを使用して NetScaler アプライアンス上の署名ファイルを更新します。

例 `import bot signature http://www.example.com/signature.json signaturefile -comment commentsforbot -overwrite`

> 注:
>
> 上書きオプションを使用して、署名ファイル内のコンテンツを更新できます。また、update bot signature <name>コマンドを使用して、NetScaler アプライアンスの署名ファイルを更新することもできます。

ボットプロファイルの追加

ボットプロファイルは、アプライアンスでボット管理を設定するためのプロファイル設定の集まりです。ボット検出を実行するように設定を構成できます。

コマンドプロンプトで入力します:

```
add bot profile <name> [-signature <string>] [-errorURL <string>]
[-trapURL <string>] [-whiteList ( ON | OFF )] [-blackList ( ON |
OFF )] [-rateLimit ( ON | OFF )] [-deviceFingerprint ( ON | OFF )]
```



```
[-deviceFingerprintAction ( none | log | drop | redirect | reset |
mitigation )] [-ipReputation ( ON | OFF )] [-trap ( ON | OFF )]
```

例:

```
add bot profile profile1 -signature signature -errorURL http://www.
example.com/error.html -trapURL /trap.html -whitelist ON -blacklist
ON -ratelimit ON -deviceFingerprint ON -deviceFingerprintAction drop
-ipReputation ON -trap ON
```

ボットプロファイルのバインド

ボットプロファイルを作成したら、ボット検出メカニズムをプロファイルにバインドする必要があります。

コマンドプロンプトで入力します:

```
bind bot profile <name> | (-ipReputation [-category <ipReputationCategory
>] [-enabled ( ON | OFF )] [-action ( none | log | drop | redirect |
reset | mitigation )] [-logMessage <string>]
```

例:

次に、IP レピュテーション検出手法を特定のボットプロファイルにバインドする例を示します。

```
bind bot profile profile5 -ipReputation -category BOTNET -enabled ON
-action drop -logMessage message
```

ボットポリシーの追加

ボットトラフィックを評価するためのボットポリシーを追加する必要があります。

コマンドプロンプトで入力します:

```
add bot policy <name> -rule <expression> -profileName <string> [-
undefAction <string>] [-comment <string>] [-logAction <string>]
```

各項目の意味は次のとおりです。

Name-ボットポリシーの名前。文字、数字、またはアンダースコア文字（_）で始まり、文字、数字、ハイフン（-）、ピリオド（.）ポンド（#）、スペース（）、アットマーク（@）、等号（=）、コロン（:）、およびアンダースコア文字のみを含める必要があります。ボットポリシーの追加後に変更できます。

Rule-指定されたリクエストにボットプロファイルを適用するかどうかをポリシーが決定するために使用する式。これは必須の議論です。最大長：1499

profileName-リクエストがこのボットポリシーに一致した場合に適用するボットプロファイルの名前。これは必須の議論です。最大長：127

undefAction-ポリシー評価の結果が未定義（UNDEF）の場合に実行するアクション。UNDEF イベントは、内部エラー状態を示します。最大長：127

Comment- このボットポリシーに関する説明。最大長：255

logAction -このポリシーに一致するリクエストに使用するログアクションの名前。最大長：127

例：

```
add bot policy pol1 -rule "HTTP.REQ.HEADER(\"header\").CONTAINS(\"
custom\")"- profileName profile1 -undefAction drop -comment commentforbotpolic
 -logAction log1
```

ボットポリシーをグローバルにバインドする

コマンドプロンプトで入力します：

```
bind bot global -policyName <string> -priority <positive_integer
> [-gotoPriorityExpression <expression>][-type ( REQ_OVERRIDE |
 REQ_DEFAULT )] [-invoke (-labelType ( vserver | policylabel )-
labelName <string>)]
```

例：

```
bind bot global -policyName pol1 -priority 100 -gotoPriorityExpression
 NEXT -type REQ_OVERRIDE
```

ボットポリシーを仮想サーバーにバインドする

コマンドプロンプトで入力します：

```
bind lb vserver <name>@ ((<serviceName>@ [-weight <positive_integer
>] )| <serviceGroupName>@ | (-policyName <string>@ [-priority <
positive_integer>] [-gotoPriorityExpression <expression>]
```

例：

```
bind lb vserver lb-server1 -policyName pol1 -priority 100 -gotoPriorityExpressi
 NEXT -type REQ_OVERRIDE
```

ボット設定の構成

必要に応じて、デフォルト設定をカスタマイズできます。
コマンドプロンプトで入力します：


```
set bot settings [-defaultProfile <string>] [-javaScriptName <string>]
    [-sessionTimeout <positive_integer>] [-sessionCookieName <string>]
    [-dfpRequestLimit <positive_integer>] [-signatureAutoUpdate ( ON |
    OFF )] [-signatureUrl <URL>] [-proxyServer <ip_addr|ipv6_addr|*>] [-
    proxyPort <port|*>]
<!--NeedCopy-->
```

各項目の意味は次のとおりです。

`defaultProfile` -接続がどのポリシーにも一致しない場合に使用するプロファイル。デフォルト設定は“”で、一致しない接続はそれ以上フィルタリングされずに NetScaler に送り返されます。最大長：31

`javaScriptName` -ボットネット機能が応答に使用する JavaScript の名前。文字または数字で始まり、1 ～31 の英字、数字、およびハイフン（-）とアンダースコア（_）記号で構成する必要があります。次の要件は、NetScaler CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、「私のクッキー名」または「私のクッキー名」）。最大長：31

`sessionTimeout` -セッションは秒単位でタイムアウトし、その後ユーザーセッションは終了します。

`Minimum value` -1、最大値:65535

`sessionCookieName` -ボットネット機能が追跡に使用するセッション Cookie の名前。文字または数字で始まり、1 ～31 の英字、数字、およびハイフン（-）とアンダースコア（_）記号で構成する必要があります。次の要件は、NetScaler CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、「私のクッキー名」または「私のクッキー名」）。最大長：31

`dfpRequestLimit` -デバイスフィンガープリントが有効になっている場合に、ボットセッション Cookie なしで許可するリクエストの数。最小値:1、最大値:4294967295

`signatureAutoUpdate` -ボットの自動更新シグネチャを有効/無効にするために使用されるフラグ。指定可能な値: オン、オフ。
デフォルト値:OFF

`signatureUrl` -ボット署名マッピングファイルをサーバーからダウンロードするための URL。デフォルト値: `https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json`。最大長：2047

`proxyServer` - AWS から更新された署名を取得するためのプロキシサーバー IP。

`proxyPort` -AWS から更新された署名を取得するためのプロキシサーバーポート。デフォルト値:8080

`proxyUsername` -シグニチャアップデートをダウンロードする際にプロキシサーバを認証するためのユーザー名。

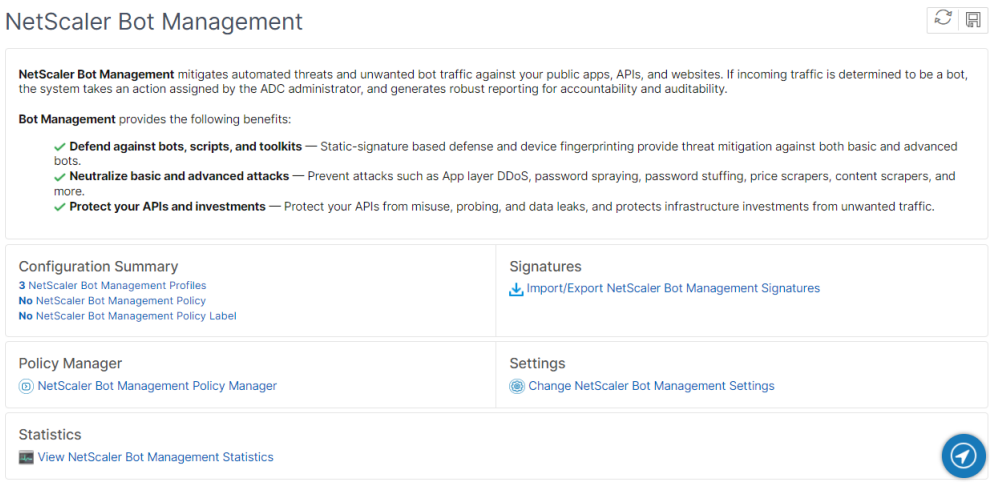
`proxyPassword` —シグニチャアップデートをダウンロードする際にプロキシサーバを認証するためのパスワード。

例:


```
set bot settings -defaultProfile profile1 -javaScriptName json.js -
sessionTimeout 1000 -sessionCookieName session -proxyServer 10.102.30.112
 -proxyPort 3128 -proxyUsername defaultuser -proxyPassword defaultPassword
```

NetScaler GUI を使用してボット管理を構成する

NetScaler のボット管理は、最初にアプライアンスで機能を有効にすることで構成できます。有効にすると、ボットポリシーを作成して、着信トラフィックをボットとして評価し、そのトラフィックをボットプロファイルに送信できます。次に、ボットプロファイルを作成し、そのプロファイルをボット署名にバインドします。別の方法として、デフォルトのボットシグネチャファイルのクローンを作成し、シグネチャファイルを使用して検出手法を設定することもできます。署名ファイルを作成したら、それをボットプロファイルにインポートできます。



1. ボット管理機能を有効にする
2. ボット管理設定を構成する
3. NetScaler ボットのデフォルト署名のクローニング
4. NetScaler ボット署名のインポート
5. ボット署名設定を構成する
6. ボットプロファイルの作成
7. ボットポリシーの作成

ボット管理機能を有効にする

ボット管理を有効にするには、次の手順を実行します。

1. ナビゲーションペインで、[システム] を展開し、[設定] **> [**拡張機能の構成] をクリックします。
2. [拡張機能の構成] ページで、[**NetScaler** ボット管理] チェックボックスを選択します。
3. [**OK**] をクリックします。

デバイスフィンガープリント技術のボット管理設定を構成する

次の手順を実行して、デバイスフィンガープリント技術を設定します:

1. [セキュリティ] > [**NetScaler** ボット管理] に移動します。
2. 詳細ペインの［設定］で、［**NetScaler Bot** 管理設定の変更］をクリックします。
3. **NetScaler** ボット管理設定の構成で、次のパラメーターを設定します。
   a) デフォルトプロファイル。ボットプロファイルを選択します。
   b) JavaScript の名前。ボット管理がクライアントへの応答で使用する JavaScript ファイルの名前です。
   c) セッションタイムアウト。ユーザーセッションが終了するまでのタイムアウト（秒単位）。
   d) セッションクッキー。ボット管理システムが追跡に使用するセッション Cookie の名前。
   e) デバイスフィンガープリント要求制限。デバイスフィンガープリントが有効になっている場合に、ボットセッション Cookie なしで許可するリクエストの数。
   f) プロキシサーバー-最新の署名がアップロードされるプロキシサーバーの IP アドレス。
   g) プロキシポート—最新のシグネチャがアップロードされたマシンのポート番号。
   h) プロキシユーザー名—プロキシサーバーの認証用のユーザー名
   i) プロキシパスワード-プロキシサーバーの認証用パスワード。

   > 注:
   >
   > 「プロキシ・ユーザー名」フィールドと「プロキシ・パスワード」フィールドは、「プロキシ・サーバー」フィールドと「プロキシ・ポート」フィールドが設定されている場合に有効になります。

4. ［**OK**］をクリックします。

ボット署名ファイルの複製

次の手順を実行して、ボット署名ファイルのクローンを作成します。

1. **[**** セキュリティ] > [**NetScaler** ボットの管理と署名] に移動します。**
2. **NetScaler** のボット管理署名ページで、デフォルトのボット署名レコードを選択し、「複製」をクリックします。
3. [ボット署名の複製] ページで、名前を入力し、署名データを編集します。
4. ［**Create**］をクリックします。

ボット署名ファイルをインポートする　独自の署名ファイルがある場合は、ファイル、テキスト、または URL としてインポートできます。ボット署名ファイルをインポートするには、次の手順を実行します。

1. **[**** セキュリティ] > [**NetScaler** ボットの管理と署名] に移動します。*****
2. **NetScaler** ボット管理署名ページで、ファイルを URL、ファイル、またはテキストとしてインポートします。
3. [続行] をクリックします。
4. NetScaler ボット管理署名のインポートページで、次のパラメータを設定します。

    a) 名前-ボット署名ファイルの名前。
    b) コメント-インポートされたファイルに関する簡単な説明。
    c) 上書き-ファイルの更新中にデータを上書きできるようにするには、このチェックボックスを選択します。
    d) 署名データ-署名パラメータの変更

5. [完了] をクリックします。

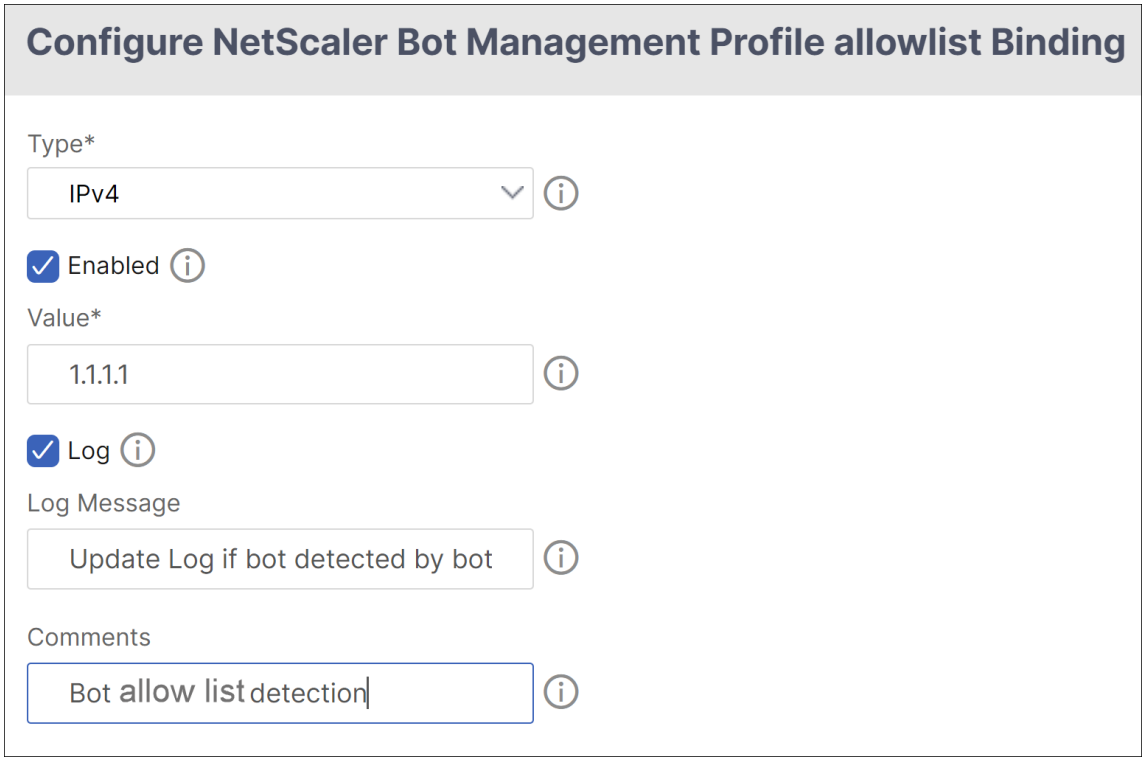
NetScaler GUI を使用してボット許可リストを構成する

この検出技術により、許可されたリストの URL を設定した URL をバイパスできます。許可リストの URL を設定するには、次の手順を実行します:

1. **[** セキュリティ**]>[NetScaler** ボット管理とプロファイル**]** に移動します。
2. **NetScaler Bot** 管理プロファイルページで、ファイルを選択して **[**編集**]** をクリックします。
3. **NetScaler Bot** 管理プロファイルページで、「署名設定」セクションに移動し、「ホワイトリスト」をクリックします。
4. [ホワイトリスト] セクションで、次のパラメータを設定します。

    a) Enabled。このチェックボックスを選択すると、検出プロセスの一環として許可リストの URL が検証されます。
    b) [タイプ] を設定します。許可リスト URL を構成します。URL は、ボットの検出中にバイパスされます。「追加」をクリックして、ボット許可リストに URL を追加します。
    c) **NetScaler Bot** 管理プロファイルのホワイトリストバインディングの設定ページで、次のパラメーターを設定します。

        i. タイプ。URL タイプは、IPv4 アドレス、サブネット IP アドレス、またはポリシー式に一致する IP アドレスです。
        ii. Enabled。チェックボックスを選択して URL を検証します。
        iii. 価値。URL アドレス。
        iv. ログ。ログエントリを保存するには、このチェックボックスを選択します。
        v. メッセージをログに記録します。ログの簡単な説明。
        vi. [コメント]。許可リスト URL に関する簡単な説明。
        vii. [**OK**] をクリックします。



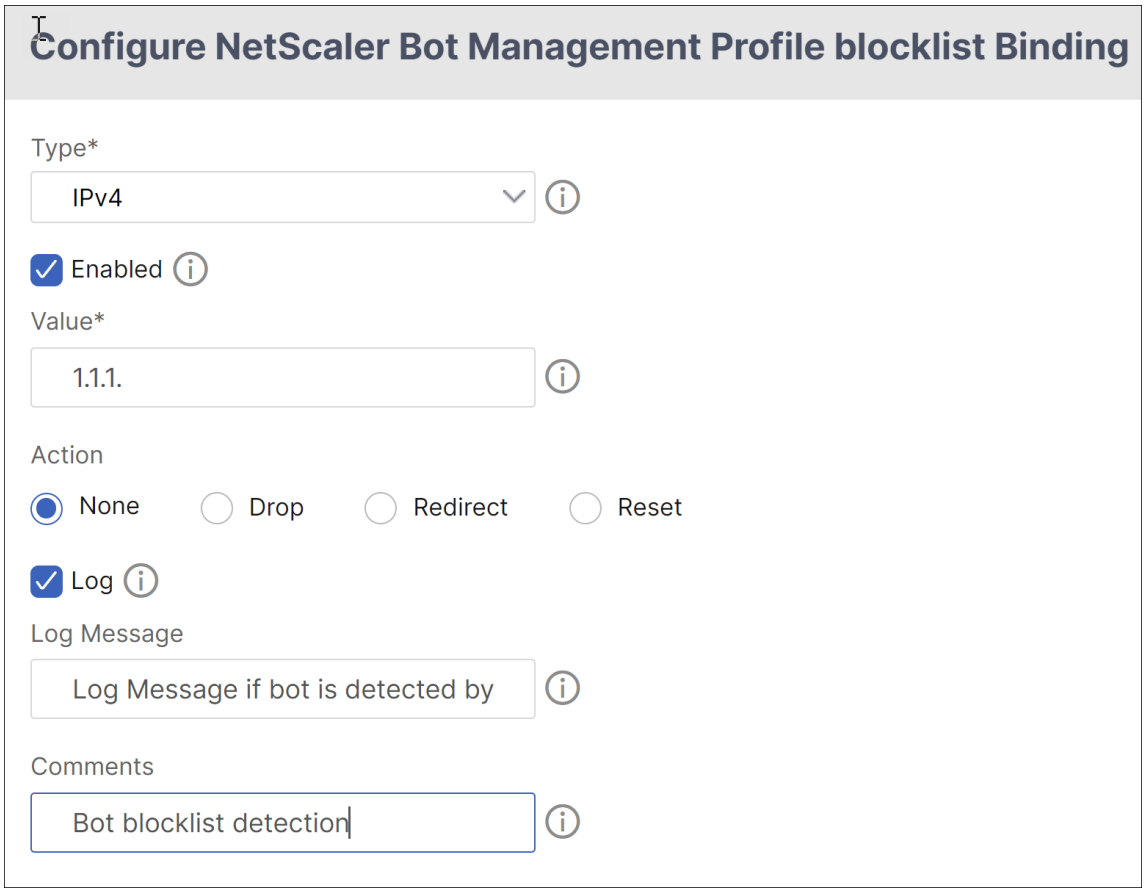
5. [**Update**] をクリックします。
6. [完了] をクリックします。

NetScaler GUI を使用してボットブロックリストを構成する

この検出技術により、ブロックリストとして設定した URL を削除できます。ブロックリスト URL を設定するには、次の手順を実行します。

1. **[** セキュリティ**]>[NetScaler** ボット管理とプロファイル**]** に移動します。
2. **NetScaler Bot** 管理プロファイルページで、署名ファイルを選択して **[**編集**]** をクリックします。
3. **NetScaler Bot** 管理プロファイルページで、「署名設定」セクションに移動し、「ブラックリスト」をクリックします。
4. [ブラックリスト] セクションで、次のパラメータを設定します。
   a) Enabled。このチェックボックスを選択すると、検出プロセスの一環としてブロックリストの URL を検証できます。
   b) [タイプ] を設定します。URL をボットブロックリスト検出プロセスの一部として設定します。これらの URL は、ボット検出中にドロップされます。「追加」をクリックして、ボット禁止リストに URL を追加します
   c) **NetScaler Bot** 管理プロファイルのブラックリストバインディングの設定ページで、次のパラメーターを設定します。

i. タイプ。URL タイプは、IPv4 アドレス、サブネット IP アドレス、または IP アドレスです。
ii. Enabled。チェックボックスを選択して URL を検証します。
iii. 価値。URL アドレス。
iv. ログ。ログエントリを保存するには、このチェックボックスを選択します。
v. メッセージをログに記録します。ログインの簡単な説明。
vi. [コメント]。ブロックリスト URL に関する簡単な説明。
vii. [**OK**] をクリックします。



5. [**Update**] をクリックします。
6. [完了] をクリックします。

NetScaler GUI を使用して **IP** レピュテーションを設定する

IP レピュテーションボット手法では、Webroot の IP レピュテーションデータベースとクラウドサービスプロバイダーデータベースを使用して、クライアントのリクエストが悪意のある IP アドレスかパブリッククラウド IP アドレスかを検証します。ボットカテゴリの一部として設定され、ボットアクションが関連付けられます。Webroot IP レピュテーションとクラウドサービスプロバイダーのデータベースカテゴリを設定するには、次の手順を実行します。

1. [セキュリティ] > [**NetScaler**Bot の管理とプロファイル] に移動します。

2. [**NetScaler bot** 管理プロファイル] ページでプロファイルを選択し、[編集] をクリックします。
3. **NetScaler Bot** 管理プロファイルページで、「プロファイル設定」セクションに移動し、「**IP** レピュテーション」をクリックします。
4. [**IP** レピュテーション] セクションで、次のパラメータを設定します。

    a) Enabled。このチェックボックスを選択すると、検出プロセスの一環として受信ボットトラフィックが検証されます。

    b) カテゴリを設定します。IP レピュテーション技術は、さまざまなカテゴリの着信ボットトラフィックに使用できます。設定されたカテゴリに基づいて、ボットトラフィックをドロップまたはリダイレクトできます。[追加] をクリックして、悪意のあるボットカテゴリを設定します。

    c) **NetScaler** ボット管理プロファイルの **IP** レピュテーションバインディングの設定ページで、次のパラメーターを設定します。

        i. カテゴリ。Webroot IP レピュテーションボットカテゴリを選択して、クライアントリクエストを悪意のある IP アドレスとして検証します。

            A. IP_BASED-このカテゴリは、クライアント IP アドレス (IPv4 および IPv6) が悪意のあるものかどうかをチェックします。
            B. BOTNET-このカテゴリには、ボットネット C&C チャネル、およびボットマスターによって制御される感染したゾンビマシンが含まれます。
            C. SPAM_SOURCES-このカテゴリには、プロキシを介したスパムメッセージのトンネリング、異常な SMTP アクティビティ、フォーラムスパムアクティビティが含まれます。
            D. SCANNERS-このカテゴリには、プローブ、ホストスキャン、ドメインスキャン、パスワード総当たり攻撃など、すべての偵察が含まれます。
            E. DOS：このカテゴリには、DOS、DDOS、異常同期フラッド、異常トラフィック検出が含まれます。
            F. REPUTATION-このカテゴリは、マルウェアに感染していることが現在知られている IP アドレス (IPv4 および IPv6) からのアクセスを拒否します。このカテゴリには、Webroot レピュテーションインデックススコアが平均的に低い IP アドレスも含まれます。このカテゴリを有効にすると、マルウェアの配布ポイントに接触していると特定されたソースからのアクセスが防止されます。
            G. フィッシング-このカテゴリには、フィッシングサイトをホストしている IP アドレス (IPv4 および IPv6) や、広告クリック詐欺やゲーム詐欺などのその他の詐欺行為が含まれます。
            H. PROXY-このカテゴリには、プロキシサービスを提供する IP アドレス (IPv4 および IPv6) が含まれます。
            I. ネットワーク-TOR またはダークネットとも呼ばれるオニオンルーターなど、プロキシおよび匿名化サービスを提供する IP。
            J. MOBILE_THREATS-このカテゴリは、クライアントの IP アドレス (IPv4 および IPv6) と、モバイルデバイスに有害なアドレスのリストをチェックします。

ii. カテゴリ。Webroot パブリッククラウドサービスプロバイダーのカテゴリを選択して、クライアントリクエストがパブリッククラウド IP アドレスであることを検証します。

A. AWS-このカテゴリは、AWS のパブリッククラウドアドレスのリストでクライアント IP アドレスをチェックします。
B. GCP-このカテゴリは、クライアントの IP アドレスを Google Cloud Platform のパブリッククラウドアドレスのリストで確認します。
C. AZURE-このカテゴリは、Azure のパブリッククラウドアドレスのリストでクライアントアドレスをチェックします。
D. ORACLE-このカテゴリは、Oracle のパブリッククラウドアドレスのリストでクライアント IP アドレスをチェックします。
E. IBM-このカテゴリは、クライアントの IP アドレスと IBM のパブリック・クラウド・アドレスのリストをチェックします。
F. SALESFORCE-このカテゴリは、クライアントの IP アドレスと Salesforce のパブリッククラウドアドレスのリストをチェックします。

Webroot IP レピュテーションボットカテゴリに指定可能な値:IP、ボットネット、SPAM_SOURCES、スキャナー、DOS、レピュテーション、フィッシング、プロキシ、ネットワーク、MOBILE_THREATS。

Webroot パブリッククラウドサービスプロバイダーのカテゴリに指定可能な値:AWS、GCP、AZURE、ORACLE、IBM、SALESFORCE。

iii. Enabled。このチェックボックスを選択して、IP レピュテーションシグニチャの検出を検証します。

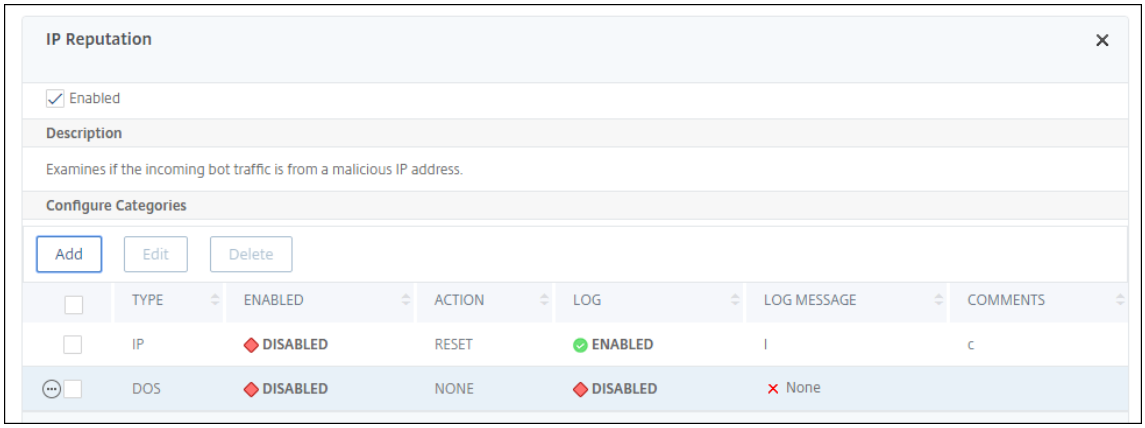
iv. Bot action. 設定したカテゴリに基づいて、アクション、ドロップ、リダイレクト、または軽減アクションを割り当てることはできません。

v. ログ。ログエントリを保存するには、このチェックボックスを選択します。

vi. メッセージをログに記録します。ログの簡単な説明。

vii. [コメント]。ボットカテゴリに関する簡単な説明。

5. [**OK**] をクリックします。
6. [**Update**] をクリックします。
7. [完了] をクリックします。



> 注:
>
> **IP** レピュテーションを無効にする場合は、必ずダウンロードを停止してください。IP レピュテーションのダウンロードを停止するには、次の手順を実行します。
>
> 1. [セキュリティ] **> [NetScaler bot** 管理 **] > [NetScaler bot** 管理設定の変更] に移動します
> 2. デフォルトの非侵入型プロファイルを **BOT_BYPASS** に変更します。

ボットレート制限テクニックの設定

ボットレート制限技術を使用すると、ユーザーの位置情報、クライアント IP アドレス、セッション、Cookie、または構成済みリソース (URL) に基づいて、特定の時間枠内のボットトラフィックを制限できます。
ボットのレート制限手法を設定することで、次のことが保証されます。

- 悪意のあるボットアクティビティをブロックします。
- ウェブサーバーへのトラフィックの負担を軽減。

NetScaler CLI を使用してボットレート制限を構成する

コマンドプロンプトで入力します:

```
1 bind bot profile <name>... -ratelimit  -type <type> Geolocation -
      countryCode <countryName> -rate  <positive_integer> -timeSlice <
      positive_integer> [-action <action> ...] [-limitType ( BURSTY |
      SMOOTH )] [-condition <expression>] [-enabled ( ON | OFF )]
2 <!--NeedCopy-->
```

各項目の意味は次のとおりです。

*SOURCE_IP -クライアント IP アドレスに基づくレート制限。

*SESSION -設定された Cookie 名に基づくレート制限。

*URL -設定された URL に基づくレート制限。

*GEOLOCATION -設定された国名に基づくレート制限。

Possible values -セッション、ソース IP、URL、ジオロケーション

例:

```
bind bot profile geo_prof -ratelimit -type Geolocation -countryCode IN
    -rate 100 -timeSlice 1000 -limitType SMOOTH -condition HTTP.REQ.
   HEADER("User-Agent").contains("anroid") -action log,drop -enabled
   on
<!--NeedCopy-->
```

NetScaler GUI を使用してボットレート制限を構成する

以下の手順を実行して、ボットレート制限検出技術を設定します。

1. **[** セキュリティ**] > [NetScaler** ボット管理とプロファイル**]** に移動します。
2. **NetScaler Bot** 管理プロファイルページで、プロファイルを選択して **[**編集**]** をクリックします。
3. **NetScaler Bot** 管理プロファイルページで、プロファイル設定セクションに移動し、レート制限をクリックします。
4. 「レート制限」セクションで、次のパラメータを設定します。
   a) Enabled。このチェックボックスを選択すると、検出プロセスの一環として受信ボットトラフィックが検証されます。
   b) [追加] をクリックしてレート制限バインディングを設定します。
5. **NetScaler Bot** 管理レート制限の設定ページで、次のパラメーターを設定します。
   a) タイプ-以下のパラメータに基づいてボットのトラフィックをレート制限します。
      i. 位置情報-ユーザーの地理的位置に基づくレート制限。
      ii. Source_IP-クライアント IP アドレスに基づいてトラフィックをレート制限します。
      iii. セッション-セッションまたは Cookie 名に基づいてボットトラフィックをレート制限します。
      iv. URL-設定された URL に基づいてボットトラフィックをレート制限します。
   b) 国-位置情報を国または地域として選択します。
   c) レート制限タイプ—次のタイプに基づいてトラフィックのタイプを制限します。
      - Bursty —設定されたしきい値および指定した期間内にあるすべてのリクエストを転送します。
      - スムーズ—指定した期間にわたってリクエストを均等に転送します。
   d) レート制限接続-1 つの条件に対して複数のルールを作成できます。
   e) 有効-このチェックボックスを選択して、受信ボットトラフィックを検証します。
   f) リクエストしきい値-特定の期間内に許可されるリクエストの最大数。

g) 期間-ミリ秒単位の時間枠。

h) アクション-選択したカテゴリのボットアクションを選択します。

i) ログ-ログエントリを保存するチェックボックスを選択します。

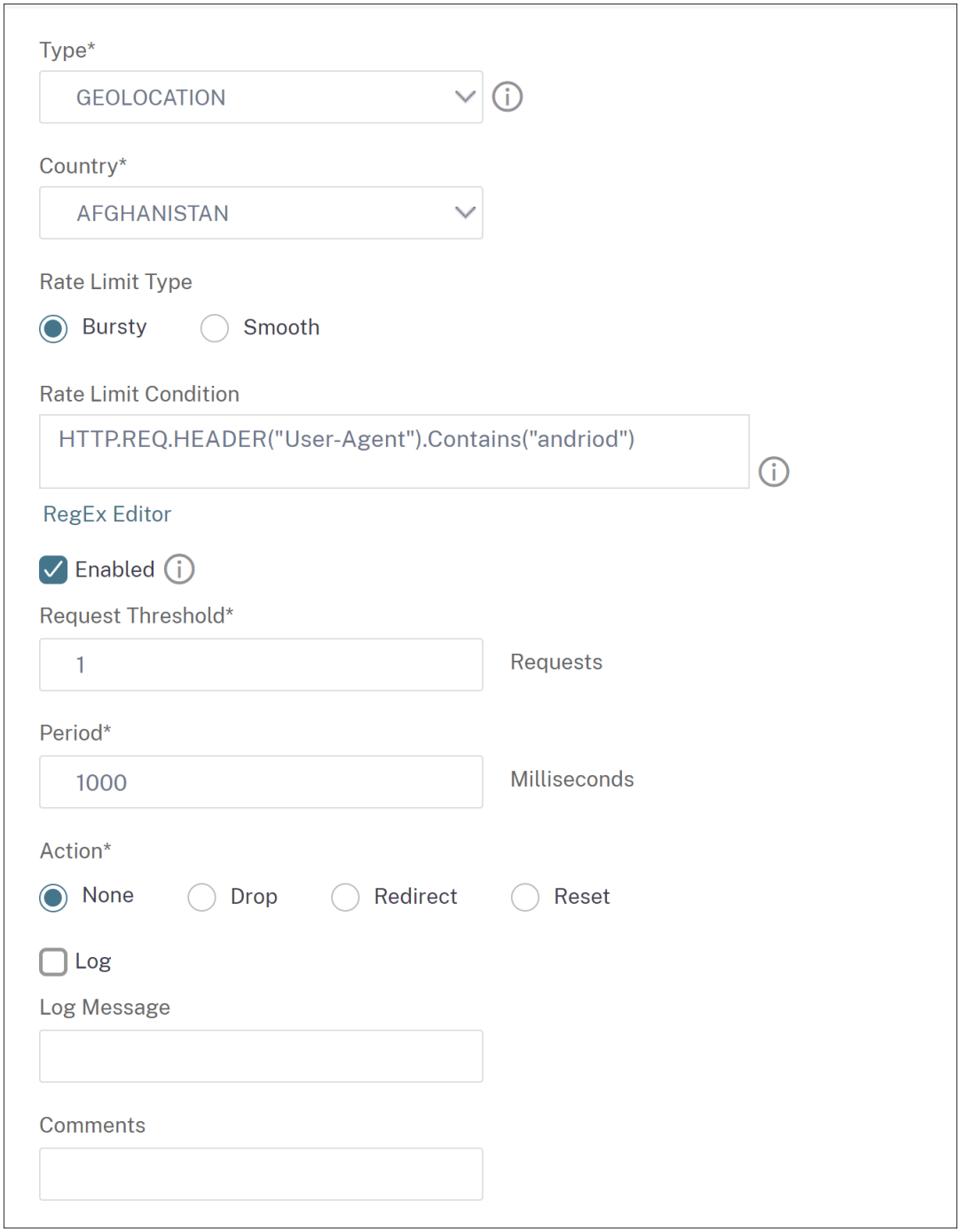
j) ログメッセージ-ログの簡単な説明。

k) コメント-ボットカテゴリに関する簡単な説明。

6. [**OK**] をクリックします。

7. [**Update**] をクリックします。

8. [完了] をクリックします。



NetScaler GUI を使用してデバイスフィンガープリント技術を構成する

この検出技術は、Java スクリプトチャレンジをクライアントに送信し、デバイス情報を抽出します。デバイス情報に基づいて、この技術はボットトラフィックをドロップまたはバイパスします。手順に従って、検出手法を設定します。

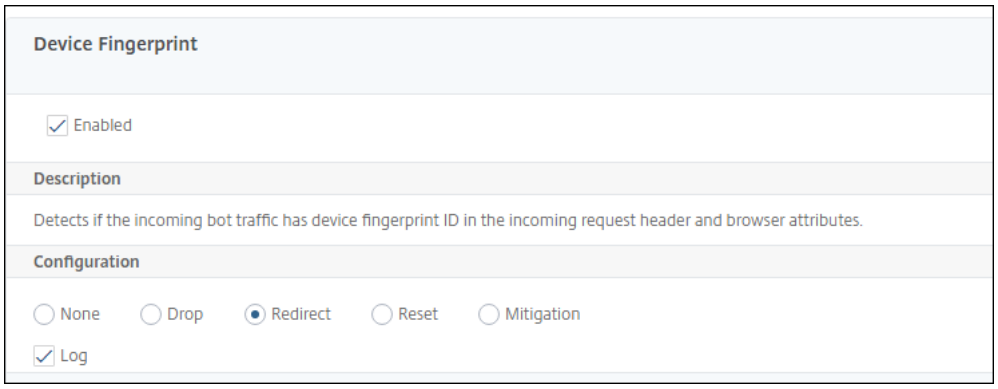
1. **[** セキュリティ**]>[NetScaler** ボット管理とプロファイル**]** に移動します。

2. **NetScaler Bot** 管理プロファイルページで、署名ファイルを選択して **[編集]** をクリックします。
3. **NetScaler Bot** 管理プロファイルページで、「署名設定」セクションに移動し、「デバイスフィンガープリント」をクリックします。
   デバイスフィンガープリントセクションで 、次のパラメータを設定します:

   a) Enabled - Select to enable the rule.
   b) Configuration - Select one of the following options:
      i. None - Allows the traffic.
      ii. Drop - Drops the traffic.
      iii. Redirect - Redirects the traffic to error URL.
      iv. Mitigation, or CAPTCHA - Validates and allows the traffic.

      > **Note:**
      > During session replay attacks using the device fingerprint cookies, requests are dropped even if the device fingerprint configuration is set to **Mitigation**.

4. ［**Update**］をクリックします。
5. ［完了］をクリックします。



モバイル **(Android)** アプリケーション用のデバイスフィンガープリント技術を構成する

デバイスフィンガープリント技術は、クライアントへの HTML 応答に JavaScript スクリプトを挿入することにより、着信トラフィックをボットとして検出します。JavaScript スクリプトは、ブラウザによって呼び出されると、ブラウザとクライアントの属性を収集し、アプライアンスに要求を送信します。これらの属性を調べて、トラフィックが Bot か人間かを判断します。

検出技術はさらに拡張され、モバイル (Android) プラットフォームでボットを検出できます。ウェブアプリケーションとは異なり、モバイル (Android) トラフィックでは、JavaScript スクリプトに基づくボット検出は適用されません。モバイルネットワーク内のボットを検出するために、この手法では、クライアント側のモバイルアプリケーションと統合されたボットモバイル SDK を使用します。SDK は、モバイルトラフィックをインターセプトし、デバイスの詳細を収集し、データをアプライアンスに送信します。アプライアンス側では、検出技術はデータを調べ、接続が Bot または人間からのものかどうかを判断します。

モバイルアプリケーション用のデバイスフィンガープリント手法の仕組み

次の手順では、モバイルデバイスからのリクエストが人間かボットかを検出するボット検出ワークフローについて説明します。

1. ユーザーがモバイルアプリケーションを操作すると、ボットモバイル SDK はデバイスの動作を記録します。
2. クライアントは NetScaler アプライアンスに要求を送信します。
3. 応答を送信すると、アプライアンスは、セッションの詳細とクライアントパラメータを収集するためのパラメータを含むボットセッションクッキーを挿入します。
4. モバイルアプリケーションが応答を受信すると、モバイルアプリケーションと統合された NetScaler ボット SDK が応答を検証し、記録されたデバイスフィンガープリントパラメーターを取得してアプライアンスに送信します。
5. アプライアンス側のデバイスフィンガープリント検出技術は、デバイスの詳細を検証し、ボットセッションクッキーが疑わしいボットであるかどうかにかかわらず、ボットセッションクッキーを更新します。
6. クッキーの有効期限が切れた場合、またはデバイスの指紋保護がデバイスパラメータを定期的に検証して収集することを好む場合は、すべての手順またはチャレンジが繰り返されます。

前提要件

モバイルアプリケーション向けの NetScaler デバイスフィンガープリント検出技術を使い始めるには、モバイルアプリケーションにボットモバイル SDK をダウンロードしてインストールする必要があります。

CLI を使用してモバイル（**Android**）アプリケーションの指紋検出技術を構成する

コマンドプロンプトで入力します:

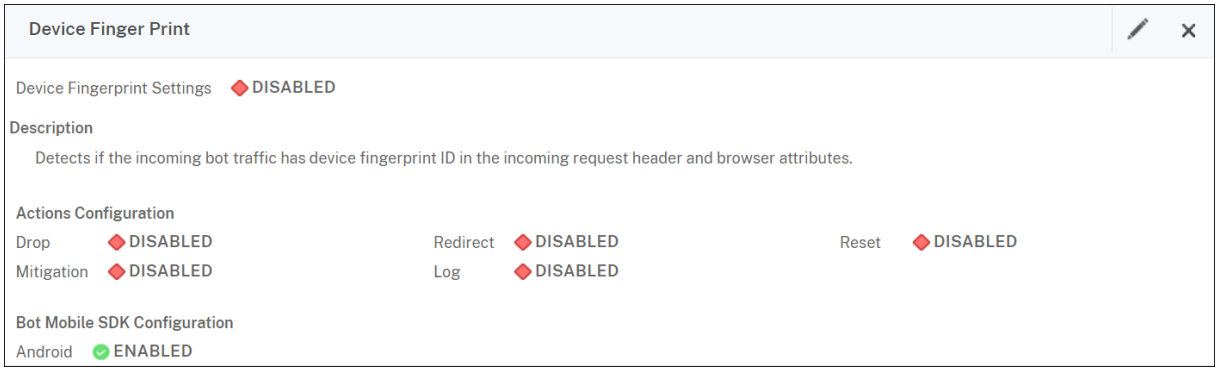
```
set bot profile <profile name> -deviceFingerprintMobile ( NONE |
Android )
```

例:

```
set bot profile profile 1 -deviceFingerprintMobile Android
```

GUI を使用してモバイル（**Android**）アプリケーションのデバイス指紋検出技術を構成する

1. **[** セキュリティ**]>[NetScaler** ボット管理とプロファイル**]** に移動します。
2. **NetScaler Bot** 管理プロファイルページで、ファイルを選択して **[**編集**]** をクリックします。
3. **NetScaler Bot** 管理プロファイルページで、プロファイル設定の **[**** デバイスフィンガープリント] をクリックします。**
4. [**Bot Mobile SDK** の設定] セクションで、モバイルクライアントの種類を選択します。
5. [更新して完了] をクリックします。



ボットログ式の設定

クライアントがボットとして識別された場合、NetScaler のボット管理により追加情報をログメッセージとしてキャプチャできます。データには、URL を要求したユーザーの名前、送信元 IP アドレス、およびユーザーが要求を送信した送信元ポート、または式から生成されたデータを指定できます。カスタムログを実行するには、ボット管理プロファイルでログ式を設定する必要があります。

CLI を使用してボットプロファイルのログ式をバインドします

コマンドプロンプトで入力します:

```
1 bind bot profile <name> (-logExpression -name <string> -expression <
     expression> [-enabled ( ON | OFF )]) -comment <string>
2 <!--NeedCopy-->
```

例:

```
bind bot profile profile1 -logExpression exp1 -expression HTTP.REQ.
URL -enabled ON -comment "testing log expression"
```

GUI を使用してログ式をボットプロファイルにバインドする

1. [セキュリティ **] > [NetScaler** ボット管理**] [**プロファイル] に移動します。
2. **NetScaler** のボット管理プロファイルページで、「プロファイル設定 **** 」セクションから「**** ボットログ表現」を選択します。
3. [ボットログ式の設定 ***] セクションで、[**** 追加] をクリックします。
4. **NetScaler** ボット管理プロファイルのボットログ表現バインディングの設定ページで、次のパラメーターを設定します。

   a) ログ式名。ログ式の名前。
   b) 式。ログ式を入力します。
   c) Enabled。ログ式バインディングを有効または無効にします。

d) [コメント]。ボットログ式バインディングについての簡単な説明。

5.「**OK**」をクリックして「完了」をクリックします。

ボットトラップ手法を構成する

NetScaler のボットトラップ技術は、サーバーの応答にトラップ URL をランダムまたは定期的に挿入します。トラップ URL リストを作成し、その URL を追加することもできます。クライアントが人間のユーザーの場合、URL は見えず、アクセスできないように見えます。ただし、クライアントが自動化されたボットの場合、URL はアクセス可能であり、アクセスされると、攻撃者はボットに分類され、ボットからの後続のリクエストはブロックされます。このトラップ技術は、ボットからの攻撃をブロックするのに効果的です。

トラップ URL は、設定可能な長さの英数字の URL で、設定可能な間隔で自動生成されます。また、この方法では、よくアクセスする Web サイトまたは頻繁にアクセスする Web サイトのトラップ挿入 URL を設定することもできます。これにより、トラップ挿入 URL に一致するリクエストに対してボットトラップ URL を挿入する目的を指示できます。

> 注:
>
> ボットトラップ URL は自動生成されますが、NetScaler ボット管理では、ボットプロファイルでカスタマイズされたトラップ URL を構成できます。これは、
> ボット検出技術を強化し、攻撃者がトラップ URL にアクセスしにくくするために行われます。

ボットトラップの設定を完了するには、次の手順を完了する必要があります。

1. ボットトラップ URL を有効にする
2. ボットプロファイルでのボットトラップ URL の設定
3. ボットトラップ挿入 URL をボットプロファイルにバインドする
4. ボット設定でボットトラップの URL の長さと間隔を構成する

ボットトラップ **URL** 保護を有効にする

開始する前に、アプライアンスでボットトラップの URL 保護が有効になっていることを確認する必要があります。コマンドプロンプトで入力します:

```
enable ns feature Bot
```

ボットプロファイルでのボットトラップ **URL** の設定

ボットトラップ URL を設定し、ボットプロファイルでトラップアクションを指定できます。
コマンドプロンプトで入力します:

```
add bot profile <name> -trapURL <string> -trap ( ON | OFF )-trapAction
 <trapAction>
```


各項目の意味は次のとおりです。

- trapURLはボット保護がトラップ URL として使用する URL です。最大長: 127
- trapはボットトラップ検出を有効にすることです。指定可能な値: オン、オフ。デフォルト値:OFF
- trapActionはボット検出に基づいて実行されるアクションです。可能な値: なし、ログ、ドロップ、リダイレクト、リセット、緩和。デフォルト値: なし

例:

```
add bot profile profile1 -trapURL www.bottrap1.com trap ON -trapAction
 RESET
```

ボットトラップ挿入 **URL** をボットプロファイルにバインドする

ボットトラップ挿入 URL を設定し、ボットプロファイルにバインドできます。
コマンドプロンプトで入力します:

```
bind bot profile <profile_name> trapInsertionURL -url <url> -enabled
ON|OFF -comment <comment>
```

各項目の意味は次のとおりです。

URL -ボットトラップ URL が挿入されるリクエスト URL の正規表現パターン。最大長: 127

例:

```
bind bot profile profile1 trapInsertionURL -url www.example.com -
enabled ON -comment insert a trap URL randomly
```

ボット設定でボットトラップの **URL** の長さと間隔を構成する

ボットトラップの URL の長さを設定し、ボットトラップ URL を自動生成する間隔を設定することもできます。
コマンドプロンプトで入力します:

```
set bot settings -trapURLAutoGenerate ( ON | OFF )-trapURLInterval <
positive_integer> -trapURLLength <positive_integer>
```

各項目の意味は次のとおりです。

- trapURLIntervalはボットトラップ URL が更新されるまでの秒単位の時間です。デフォルト値:3600、最小値:300、最大値:86400
- trapURLLength。自動生成されたボットトラップ URL の長さ。デフォルト値:32、最小値:10、最大値:255

例:

```
set bot settings -trapURLAutoGenerate ON -trapURLInterval 300 -trapURLLength
 60
```

GUI を使用したボットトラップ **URL** の設定

1. [セキュリティ**] > [NetScaler** ボット管理**] [**プロファイル] に移動します。
2. **NetScaler** のボット管理プロファイルページで、「編集」をクリックしてボットトラップ URL の手法を設定します。
3. **NetScaler** ボット管理プロファイルの作成ページで、一般セクションにボットトラップ URL を入力します。
4. **NetScaler Bot** 管理プロファイルページで、「プロファイル設定」から「ボットトラップ」をクリックします。
5. [**Bot Trap**] セクションで、次のパラメータを設定します。

   a) Enabled。チェックボックスを選択してボットトラップ検出を有効にします
   b) 説明。URL に関する簡単な説明。
   c)「アクションを設定」セクションで、次のパラメータを設定します。

```
1.  Action. Action to be taken for bot detected by bot trap access
    .
1.  Log. Enable or disable logging for bot trap binding.
```

6. [トラップ挿入 **URL** の設定] セクションで、[追加] をクリックします。
7. **NetScaler** ボット管理プロファイルのボットトラップバインディングの設定ページで、次のパラメーターを設定します。

   a) トラップ URL。ボットトラップ挿入 URL として確認する URL を入力します。
   b) Enabled。ボットトラップ挿入 URL を有効または無効にします。
   c) [コメント]。トラップ挿入 URL に関する簡単な説明。

8. [更新して完了] をクリックします。

ボットトラップ **URL** 設定の構成

ボットトラップ URL 設定を構成するには、次の手順を実行します。

1. [セキュリティ**] > [NetScaler** ボット管理] に移動します。
2. 詳細ペインの［設定］で、［**NetScaler Bot** 管理設定の変更］をクリックします。
3. **NetScaler** ボット管理設定の構成で、次のパラメーターを設定します。

   a) トラップ URL 間隔。ボットトラップ URL が更新されるまでの時間（秒単位）。

b) トラップ URL の長さ。自動生成されたボットトラップ URL の長さ。

4. [**OK**] をクリックし、[完了] をクリックします。

ボット検出用のクライアント **IP** ポリシー式

NetScaler のボット管理では、高度なポリシー表現を構成して、HTTP リクエストヘッダー、HTTP リクエスト本文、HTTP リクエスト URL からクライアント IP アドレスを抽出したり、高度なポリシー表現を使用したりできるようになりました。抽出された値は、ボット検出メカニズム (TPS、ボットトラップ、レート制限など) によって使用され、受信したリクエストがボットかどうかを検出します。

> 注:
>
> クライアント IP 式を設定していない場合は、デフォルトまたは既存の送信元クライアント IP アドレスがボットの検出に使用されます。式が設定されている場合、評価結果には、ボット検出に使用できるクライアント IP アドレスが提供されます。

受信要求がプロキシサーバを介して送信され、クライアントの IP アドレスがヘッダーに存在する場合は、クライアント IP 式を設定して使用して、実際のクライアント IP アドレスを抽出できます。この設定を追加することで、アプライアンスはボット検出メカニズムを使用して、ソフトウェアクライアントとサーバーにより多くのセキュリティを提供できます。

CLI を使用して、ボットプロファイルでクライアント **IP** ポリシー式を設定します　コマンドプロンプトで入力します:

```
add bot profile <name> [-clientIPExpression <expression>]
<!--NeedCopy-->
```

例:

```
add bot profile profile1 -clientIPExpression 'HTTP.REQ.HEADER("X-Forwarded-For")ALT CLIENT.IP.SRC.TYPECAST_TEXT_T'

add bot profile profile1 -clientIPExpression 'HTTP.REQ.HEADER("X-Forwarded-For")ALT CLIENT.IPv6.SRC.TYPECAST_TEXT_T'
```

GUI を使用してボットプロファイルでクライアント **IP** ポリシー式を構成する

1. [セキュリティ] **> [NetScaler** ボット管理**] [**プロファイル] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **NetScaler Bot** 管理プロファイルの作成ページで、クライアント IP 表現を設定します。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

IP レピュテーションとデバイスフィンガープリント検出のための CAPTCHA の構成

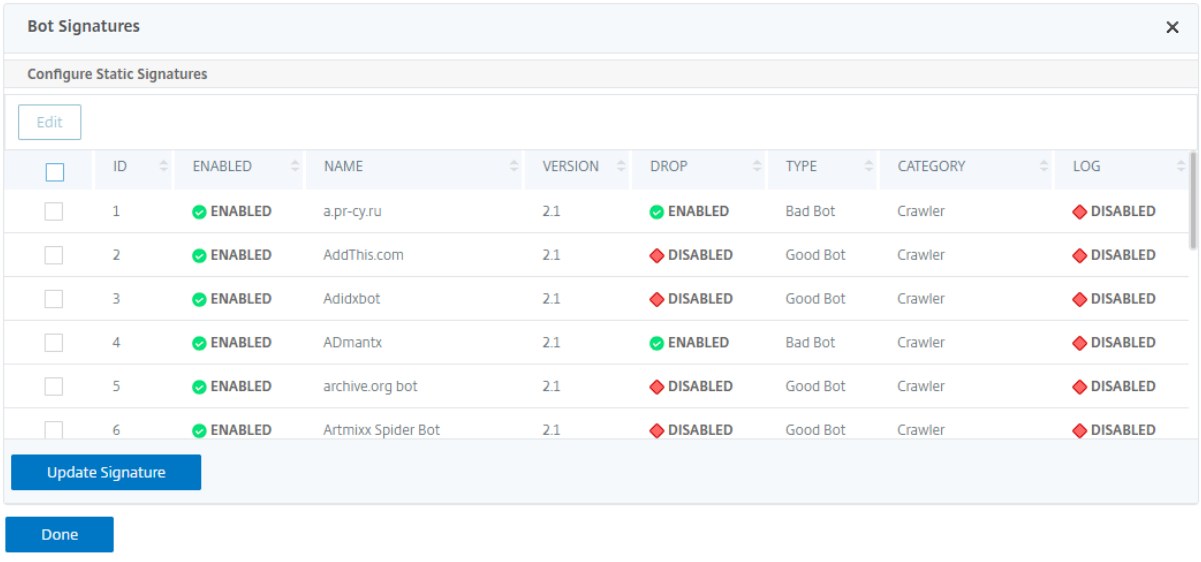
CAPTCHA は、「Computers and Humans Apart を伝える完全に自動化された Public Turing test」の略の頭字語です。CAPTCHA は、着信トラフィックが人間のユーザーまたは自動ボットからのものかどうかをテストするように設計されています。CAPTCHA は、ウェブアプリケーションにセキュリティ違反を引き起こす自動ボットをブロックするのに役立ちます。NetScaler では、CAPTCHA はチャレンジレスポンスモジュールを使用して、受信トラフィックが自動ボットではなく人間のユーザーからのものかどうかを識別します。

ボットの静的シグネチャの設定

この検出手法により、ブラウザの詳細からユーザエージェント情報を識別できます。ユーザーエージェント情報に基づいて、ボットは不良または良好なボットとして識別され、ボットアクションを割り当てます。

スタティックシグニチャ技術を設定するには、次の手順を実行します。

1. ナビゲーションペインで、[セキュリティ] **>** **[NetScaler** ボット管理**] > [**署名**]** の順に展開します。
2. **NetScaler Bot Management** の署名ページで、署名ファイルを選択して「編集」をクリックします。
3. **NetScaler Bot Management** の署名ページで、「署名設定」セクションに移動し、「ボット署名」をクリックします。
4. [ボット署名] セクションで、次のパラメータを設定します。
   a) スタティックシグニチャを設定します。このセクションには、ボットの静的署名レコードのリストがあります。レコードを選択し、[編集] をクリックして、そのレコードにボットアクションを割り当てることができます。
   b) [**OK**] をクリックします。
5. [署名の更新] をクリックします。
6. [完了] をクリックします。



ボットの静的シグニチャの描写

NetScaler のボット管理は、Web アプリケーションをボットから保護します。ボットの静的シグネチャは、着信リクエストのユーザーエージェントなどのリクエストパラメータに基づいて、良いボットと悪いボットを識別するのに役立ちます。
ファイル内のシグネチャのリストは巨大で、新しいルールが追加され、古いルールが定期的に削除されます。管理者は、カテゴリで特定のシグニチャまたはシグニチャのリストを検索したい場合があります。署名を簡単にフィルタリングするために、ボットシグネチャページには拡張された検索機能があります。検索機能を使用すると、シグニチャルールを検索し、アクション、シグニチャ ID、開発者、シグニチャ名などの 1 つ以上のシグニチャパラメータに基づいてそのプロパティを設定できます。

アクション-特定のカテゴリのシグニチャルールに設定したいボットアクションを選択します。使用可能なアクションタイプは次のとおりです。

- 選択項目を有効にする-選択したすべての署名ルールを有効にします。
- 選択項目を無効にする-選択したすべてのシグニチャルールを無効にします。
- 選択したものをドロップ-選択したすべてのシグニチャルールに「ドロップ」アクションを選択します。
- 選択したリダイレクト-選択したすべてのシグニチャルールに「リダイレクト」アクションを適用します。
- 選択項目をリセット-選択したすべてのシグニチャルールに「リセット」アクションを適用します。
- 選択したログ-選択したすべてのシグニチャルールに「ログ」アクションを適用します。
- Drop Selected を削除-選択したすべてのシグニチャルールに対するドロップアクションの設定を解除します。
- 選択したリダイレクトを削除-選択したすべてのシグニチャルールへのリダイレクトアクションの設定を解除します。
- Reset Selected を削除-選択したすべてのシグニチャルールに対するリセットアクションの設定を解除します。
- 選択したログを削除-選択したすべてのシグニチャルールへのログアクションの設定を解除します。

カテゴリ-カテゴリを選択して、それに応じてシグニチャルールをフィルタリングします。次に、シグニチャルールのソートに使用できるカテゴリの一覧を示します。

- アクション-ボットのアクションに基づいてソートします。
- カテゴリ-ボットカテゴリに基づいてソートします。
- 開発者-ホスト企業の発行元に基づいてソートします。
- 有効-有効になっている署名ルールに基づいてソートします。
- Id-署名ルール ID に基づいてソートします。
- ログ-ロギングが有効になっているシグニチャルールに基づいてソートします。
- 名前-シグニチャルール名に基づいてソートします。
- タイプ-署名タイプに基づいてソートします。
- バージョン-シグニチャルールのバージョンに基づいてソートします。

NetScaler GUI を使用して、アクションとカテゴリの種類に基づいてボットの静的署名ルールを検索します

1. [** セキュリティ] > [NetScaler ボット管理] **[署名] に移動します。
2. 詳細ページで、[追加] をクリックします。
3. **NetScaler Bot** 管理署名ページで、静的署名セクションの「編集」をクリックします。
4. [静的署名の構成] セクションで、ドロップダウンリストからシグニチャアクションを選択します。
5. 検索機能を使用してカテゴリを選択し、それに応じてルールをフィルタリングします。
6. [**Update**] をクリックします。

NetScaler GUI を使用してボットの静的署名ルールプロパティを編集します

1. [** セキュリティ] > [NetScaler ボット管理] **[署名] に移動します。
2. 詳細ページで、[追加] をクリックします。
3. **NetScaler Bot** 管理署名ページで、静的署名セクションの「編集」をクリックします。
4. [静的署名の構成] セクションで、ドロップダウンリストからアクションを選択します。
5. 検索機能を使用してカテゴリを選択し、それに応じてルールをフィルタリングします。
6. 静的署名リストから、シグニチャを選択してプロパティを変更します。
7. [**OK**] をクリックして確定します。

NetScaler のボット管理における **CAPTCHA** の仕組み

NetScaler ボット管理では、CAPTCHA 検証は、ボットポリシーが評価された後に実行されるポリシーアクションとして構成されます。CAPTCHA アクションは、IP レピュテーションおよびデバイスフィンガープリント検出技術でのみ使用できます。以下に、CAPTCHA の仕組みを理解する手順を示します。

1. IP レピュテーションまたはデバイスフィンガープリントボットの検出中にセキュリティ違反が観察された場合、ADC アプライアンスは CAPTCHA チャレンジを送信します。
2. クライアントは CAPTCHA レスポンスを送信します。
3. アプライアンスは CAPTCHA 応答を検証し、CAPTCHA が有効な場合、要求は許可され、バックエンドサーバーに転送されます。
4. CATCHA 応答が無効な場合、アプライアンスは最大試行回数に達するまで新しい CAPTCHA チャレンジを送信します。
5. 最大試行回数の後でも CAPTCHA 応答が無効である場合、アプライアンスは要求をドロップするか、設定されたエラー URL にリダイレクトします。
6. ログアクションを設定した場合、アプライアンスは要求の詳細を ns.log ファイルに保存します。

NetScaler GUI を使用して **CAPTCHA** 設定を構成する

ボット管理 CAPTCHA アクションは、IP レピュテーションおよびデバイスフィンガープリント検出技術でのみサポートされます。**CAPTCHA** 設定を構成するには、次の手順を実行します。

1. [セキュリティ**] > [NetScaler** ボット管理とプロファイル**]** に移動します。

2. **NetScaler Bot** 管理プロファイルページで、プロファイルを選択して **[**編集**]** をクリックします。
3. **NetScaler Bot** 管理プロファイルページで、「プロファイル設定」セクションに移動し、「**CAPTCHA**」をクリックします。
4. 「**CAPTCHA** 設定」セクションで、「追加」をクリックして、プロファイルに **CAPTCHA** 設定を構成します。
5. **NetScaler** ボット管理 **CAPTCHA** の構成ページで、次のパラメーターを設定します。

   a) URL。IP レピュテーションおよびデバイスフィンガープリント検出技術中に CAPTCHA アクションが適用されるボット URL。

   b) Enabled。CAPTCHA サポートを有効にするには、このオプションを設定します。

   c) 猶予時間。現在の有効な CAPTCHA 応答を受信した後、新しい CAPTCHA チャレンジが送信されないまでの期間。

   d) 待ち時間。ADC アプライアンスがクライアントが CAPTCHA 応答を送信するまで待機するまでに要した時間。

   e) ミュート期間。不正な CAPTCHA 応答を送信したクライアントが、次の試行を許可されるまで待つ必要がある期間。このミュート期間中、ADC アプライアンスは要求を許可しません。範囲:60 ～900 秒、推奨:300 秒

   f) リクエストの長さ制限。CAPTCHA チャレンジがクライアントに送信される要求の長さ。長さがしきい値より大きい場合、要求はドロップされます。デフォルト値は 10 ～3000 バイトです。

   g) 再試行回数。クライアントが CAPTCHA チャレンジを解決するために再試行できる試行回数。範囲:1—10、推奨:5。

   h) クライアントが CAPTCHA 検証に失敗した場合、アクション/ドロップ/リダイレクトアクションは実行されません。

   i) ログ。このオプションを設定すると、レスポンス CAPTCHA が失敗したときにクライアントからのリクエスト情報を保存できます。データは`ns.log`ファイルに保存されます。

   j) [コメント]。CAPTCHA 設定に関する簡単な説明。

6. [**OK**] をクリックし、[完了] をクリックします。
7. [** セキュリティ] > [NetScaler ボット管理] **[署名] に移動します。
8. **NetScaler Bot Management** の署名ページで、署名ファイルを選択して「編集」をクリックします。
9. **NetScaler Bot Management** の署名ページで、「署名設定」セクションに移動し、「ボット署名」をクリックします。
10. [ボット署名] セクションで、次のパラメータを設定します。
11. スタティックシグニチャを設定します。ボット静的署名レコードを選択し、[編集] をクリックして、ボットアクションをそのレコードに割り当てます。

12. [**OK**] をクリックします。
13. [署名の更新] をクリックします。
14. [完了] をクリックします。

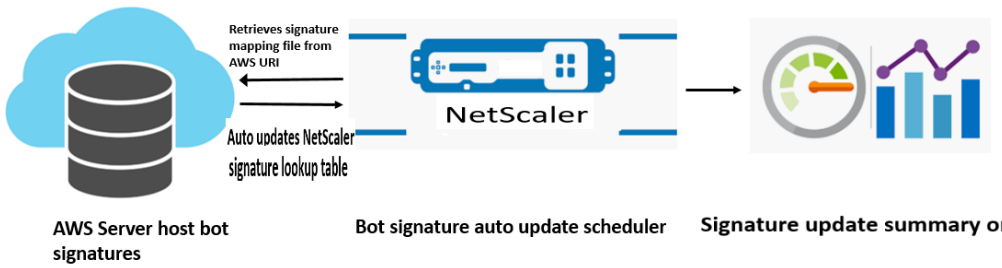
ボット署名の自動更新

ボット静的シグニチャ手法では、シグニチャルックアップテーブルと良いボットと不良ボットのリストを使用します。ボットは、ユーザーエージェント文字列とドメイン名に基づいて分類されます。着信ボットトラフィックのユーザーエージェント文字列とドメイン名がルックアップテーブルの値と一致する場合、設定されたボットアクションが適用されます。
ボット署名の更新は AWS クラウドでホストされ、署名ルックアップテーブルは、署名の更新のために AWS データベースと通信します。自動署名更新スケジューラーは 1 時間ごとに実行され、**AWS** データベースをチェックし、NetScaler アプライアンスの署名テーブルを更新します。

設定するシグニチャ自動更新 URL は、`https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json`

> 注:
>
> また、プロキシサーバーを設定し、プロキシを介して AWS クラウドからアプライアンスにシグネチャを定期的に更新することもできます。プロキシ設定の場合、ボット設定でプロキシ IP アドレスとポートアドレスを設定する必要があります。

ボット署名の自動更新の仕組み　次の図は、ボット署名が AWS クラウドから取得され、NetScaler で更新され、NetScaler Console で署名更新の概要がどのように表示されるかを示しています。



ボットシグネチャ自動更新スケジューラは、次のことを行います。

1. AWS URI からマッピングファイルを取得します。
2. マッピングファイル内の最新のシグニチャを、ADC アプライアンスの既存のシグニチャでチェックします。
3. AWS から新しい署名をダウンロードし、署名の整合性を検証します。
4. 既存のボット署名を、ボット署名ファイル内の新しい署名で更新します。
5. SNMP アラートを生成し、署名更新の概要を NetScaler コンソールに送信します。

ボット署名の自動更新を設定する　ボット署名の自動更新を設定するには、次の手順を実行します。

ボット署名の自動更新を有効にする　ADC アプライアンスのボット設定で自動更新オプションを有効にする必要があります。
コマンドプロンプトで入力します:

```
set bot settings -signatureAutoUpdate ON
```

プロキシサーバーの設定を構成する **(**オプション**)**　プロキシサーバーを介して AWS 署名データベースにアクセスする場合は、プロキシサーバーとポートを設定する必要があります。

```
set bot settings -proxyserver -proxyport
```

例:

```
set bot settings -proxy server 1.1.1.1 -proxyport 1356
```

NetScaler GUI を使用してボット署名の自動更新を構成する　ボット署名の自動更新を設定するには、次の手順を実行します。

1. [セキュリティ**] > [NetScaler** ボット管理] に移動します。
2. 詳細ペインの［設定］で、［**NetScaler Bot** 管理設定の変更］をクリックします。
3.「**NetScaler Bot** 管理設定の構成」で、「署名の自動更新」チェックボックスを選択します。
4.「**OK**」をクリックして「閉じる」をクリックします。

ボット管理プロファイルの作成

ボットプロファイルは、ボットタイプの検出に使用されるボット管理設定の集まりです。プロファイルでは、Web App Firewall が各フィルタ（またはチェック）を Web サイトへのボットトラフィックに適用する方法と、それらからの応答を決定します。

ボットプロファイルを設定するには、次の手順を実行します。

1. **[** セキュリティ**] > [NetScaler** ボット管理**] > [**プロファイル**]** に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **NetScaler Bot** 管理プロファイルの作成ページで、次のパラメーターを設定します。
   a) 名前。ボットプロファイル名。
   b) 署名。ボット署名ファイルの名前。
   c) エラー URL。リダイレクトの URL。
   d) [コメント]。プロファイルに関する簡単な説明。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

ボットポリシーの作成

ボットポリシーは、ボット管理システムへのトラフィックを制御し、監査ログサーバーに送信されるボットログも制御します。手順に従って、ボットポリシーを設定します。

1. **[**** セキュリティ] > [**NetScaler** ボット管理**] > [**ボットポリシー] に移動します。******
2. 詳細ペインで、[追加] をクリックします。
3. **NetScaler** ボット管理ポリシーの作成ページで、次のパラメーターを設定します。

    a) 名前。ボットポリシーの名前。
    b) 式。ポリシー式またはルールをテキスト領域に直接入力します。
    c) ボットプロファイル。ボットポリシーを適用するボットプロファイル。
    d) 未定義のアクション。割り当てるアクションを選択します。
    e) [コメント]。ポリシーに関する簡単な説明。
    f) ログアクション。ボットトラフィックを記録するための監査ログメッセージアクション。監査ログアクションの詳細については、「監査ログ」のトピックを参照してください。

4. **[**作成**]** して **[**閉じる**]** をクリックします。

1 秒あたりのボットトランザクション **(TPS)**

1 秒あたりのトランザクション数 (TPS) ボット技術は、1 秒あたりの要求数 (RPS) と RPS の増加率が設定されたしきい値を超えた場合に、着信トラフィックをボットとして検出します。この検出技術は、ウェブスクレイピングアクティビティ、ブルートフォースログイン、その他の悪意のある攻撃を引き起こす可能性のある自動ボットから Web アプリケーションを保護します。

> 注:
>
> ボット技術は、両方のパラメータが設定されていて、両方の値がしきい値制限を超えて増加した場合にのみ、着信トラフィックをボットとして検出します。
> アプライアンスが特定の URL から多くのリクエストを受け取り、NetScaler のボット管理者にボット攻撃があるかどうかを検出させたいシナリオを考えてみましょう。TPS 検出技術は、1 秒以内に URL から送信された要求の数（設定値）と 30 分以内に受信された要求数の増加率（設定値）を調べます。値がしきい値制限を超えると、トラフィックはボットと見なされ、アプライアンスは設定されたアクションを実行します。

1 秒あたりのボットトランザクション **(TPS)** 手法の設定　TPS を設定するには、次の手順を完了する必要があります:

1. ボットの TPS を有効にする
2. TPS 設定をボット管理プロファイルにバインドする

TPS 設定をボット管理プロファイルにバインドする　ボット TPS 機能を有効にしたら、TPS 設定をボット管理プロファイルにバインドする必要があります。

コマンドプロンプトで入力します:

```
bind bot profile <name>… (-tps [-type ( SourceIP | GeoLocation |
RequestURL | Host )] [-threshold <positive_integer>] [-percentage <
positive_integer>] [-action ( none | log | drop | redirect | reset |
mitigation )] [-logMessage <string>])
```

例:

```
bind bot profile profile1 -tps -type RequestURL -threshold 1 -percentage
 100000 -action drop -logMessage log
```

1 秒あたりのボットトランザクション **(TPS)** を有効にする　開始する前に、アプライアンスでボット TPS 機能が有効になっていることを確認する必要があります。コマンドプロンプトで入力します:

```
set bot profile profile1 -enableTPS ON
```

NetScaler GUI を使用してボットトランザクション/秒（**TPS**）を構成する　1 秒あたりのボットトランザクションを設定するには、次の手順を実行します。

1. [セキュリティ**] > [NetScaler** ボット管理**] [**プロファイル] に移動します。
2. **NetScaler Bot** 管理プロファイルページで、プロファイルを選択して **[**編集**]** をクリックします。
3. **NetScaler Bot** 管理プロファイルの作成ページで、「署名設定」セクションの「**TPS** 」をクリックします。
4. [**TPS**] セクションで、機能を有効にし、[追加] をクリックします。
5. [**NetScaler Bot** 管理プロファイルの **TPS** バインドの構成］ページで、次のパラメーターを設定します。

    a) タイプ-次のタイプのいずれかを選択します:

    SOURCE_IP —クライアント IP アドレスに基づいた TPS。

    GEOLOCATION —クライアントの地理的位置に基づく TPS。

    HOST-特定のバックエンドサーバーの IP アドレスに転送されたクライアント要求に基づく TPS。

    URL —特定の URL からのクライアント要求に基づく TPS。

    b) 固定しきい値-1 秒間隔内に TPS 入力タイプから許可されるリクエストの最大数。

    c) パーセンテージしきい値-30 分間隔内の TPS 入力タイプからのリクエストの最大増加率。

    d) アクション-TPS バインディングによって検出されたボットに対して実行するアクション。

    e) ログ-TPS バインディングのロギングを有効または無効にします。

f) メッセージをログに記録します。TPS バインディングによって検出されたボットのログを記録するメッセージです。最大長:255

g) コメント-TPS 構成に関する簡単な説明。最大長：255

6. **OK**] をクリックし、[閉じる] をクリックします。

マウスとキーボードのダイナミクスに基づくボット検出

NetScaler のボット管理では、ボットを検出し、Web スクレイピングの異常を軽減するために、マウスとキーボードの動作に基づく高度なボット検出技術を使用しています。人間の直接的なインタラクション（CAPTCHA 検証など）を必要とする従来のボット手法とは異なり、強化された手法はマウスとキーボードのダイナミクスを受動的に監視します。次に、NetScaler アプライアンスはリアルタイムのユーザーデータを収集し、人間とボットの間の行動を分析します。

マウスとキーボードのダイナミクスを用いたパッシブボット検出は、既存のボット検出メカニズムに比べて次のような利点があります。

- ユーザー・セッション全体にわたって継続的な監視を提供し、単一のチェックポイントを排除します。
- 人間による操作は不要で、ユーザーには透過的です。

マウスとキーボードのダイナミクスを使用したボット検出の仕組み

キーボードとマウスのダイナミクスを使用したボット検出技術は、ウェブページロガーとボット検出器の 2 つのコンポーネントで構成されています。ウェブページロガーは、ユーザーがウェブページでタスクを実行しているときのキーボードとマウスの動きを記録する JavaScript です（登録フォームへの入力など）。その後、ロガーはデータをバッチで NetScaler アプライアンスに送信します。その後、アプライアンスはデータを KM レコードとして保存し、NetScaler Console サーバー上のボット検出器に送信し、ユーザーが人間かボットかを分析します。

次の手順では、コンポーネントがどのように相互作用するかを説明します:

1. NetScaler 管理者は、ADM StyleBook、CLI、NITRO、またはその他の方法でポリシー表現を構成します。
2. URL は、管理者がアプライアンスで機能を有効にすると、ボットプロファイルで設定されます。
3. クライアントが要求を送信すると、NetScaler アプライアンスはセッションとセッション内のすべての要求を追跡します。
4. リクエストがボットプロファイルで設定された式と一致する場合､アプライアンスはレスポンスに JavaScript（ウェブページロガー）を挿入します。
5. JavaScript はすべてのキーボード、マウスアクティビティを収集し、KM データを POST URL (一時的) に送信します。
6. NetScaler アプライアンスはデータを保存し、セッションの終了時に NetScaler コンソールサーバーに送信します。アプライアンスが POST 要求の完全なデータを受信すると、そのデータは ADM サーバーに送信されます。

7. NetScaler コンソールサービスがデータを分析し、その分析結果に基づいて、NetScaler コンソールサービスの GUI で結果を確認できます。

JavaScript ロガーは、次のマウスとキーボードの動きを記録します:

- キーボードイベント—すべてのイベント
- マウスイベント-マウスの移動、マウスアップ、マウスダウン
- クリップボードイベント-貼り付け
- カスタムイベント-オートフィル、オートフィルキャンセル
- 各イベントのタイムスタンプ

マウスとキーボードのダイナミクスを使用してボット検出を構成する

NetScaler のボット管理構成には、キーボードとマウスによる検出機能の有効化または無効化が含まれ、ボットプロファイルで JavaScript URL を構成します。
マウスとキーボードのダイナミクスを使用してボットの検出を設定するには、次の手順を実行します。

1. キーボードとマウスベースの検出を有効にする
2. HTTP レスポンスに JavaScript を注入できるタイミングを決定する式を設定します。

キーボードマウスベースのボット検出を有効にする　設定を開始する前に、アプライアンスでキーボードおよびマウスベースのボット検出機能が有効になっていることを確認します。

コマンドプロンプトで入力します:

```
1 add bot profile <name> -KMDetection ( ON | OFF )
2 <!--NeedCopy-->
```

例:

```
add bot profile profile1 -KMDetection ON
```

JavaScript 挿入用のボット式を構成する　トラフィックを評価し、JavaScript を挿入するボット式を設定します。JavaScript は、式が true と評価された場合にのみ挿入されます。

コマンドプロンプトで入力します:

```
1 bind bot profile <name> -KMDetectionExpr -name <string> -expression <
     expression> -enabled ( ON | OFF ) - comment <string>
2 <!--NeedCopy-->
```

例:

```
bind bot profile profile1 -KMDetectionExpr -name test -expression
http.req.url.startswith("/testsite")-enabled ON
```


キーボードマウスベースのボット検出のために **HTTP** レスポンスに挿入された **JavaScript** ファイル名を設定するユーザーアクションの詳細を収集するために、アプライアンスは HTTP レスポンスに JavaScript ファイル名を送信します。JavaScript ファイルは KM レコードのすべてのデータを収集し、アプライアンスに送信します。

コマンドプロンプトで入力します:

```
set bot profile profile1 - KMJavaScriptName <string>
<!--NeedCopy-->
```

例:

```
set bot profile profile1 -KMJavaScriptName script1
```

ビヘイビアバイオメトリクスサイズの構成　KM レコードとしてアプライアンスに送信し、ADM サーバーで処理できるマウスとキーボードの動作データの最大サイズを設定できます。

コマンドプロンプトで入力します:

```
set bot profile profile1 -KMEventsPostBodyLimit <positive_integer>
<!--NeedCopy-->
```

例:

```
set bot profile profile1 - KMEventsPostBodyLimit 25
```

JavaScript を構成してキーボードとマウスの動作バイオメトリクスを収集するように NetScaler アプライアンスを構成すると、アプライアンスはデータを NetScaler Console サーバーに送信します。NetScaler Console サーバーが行動バイオメトリクスからボットを検出する方法の詳細については、「ボット違反」トピックを参照してください。

GUI を使用してキーボードとマウスのボットの式設定を構成する

1. [セキュリティ **] > [NetScaler** ボット管理とプロファイル**]** に移動します。
2. **NetScaler Bot** 管理プロファイルページで、プロファイルを選択して **[**編集**]** をクリックします。
3. **NetScaler** ボット管理プロファイルページで、編集アイコンをクリックします。
4. 基本設定の「キーボードとマウスベースのボット検出」セクションで、次のパラメータを設定します:
   a) 検出を有効にします。チェックボックスを選択すると、ボットベースのキーボードとマウスのダイナミクスの動作が検出されます。
   b) イベント投稿本体の制限。ブラウザから送信され、NetScaler アプライアンスで処理されるキーボードとマウスのダイナミクスデータのサイズ。
5. [**OK**] をクリックします。
6. [**NetScaler bot Management** プロファイル] ページで、[プロファイル設定] セクションに移動し、[キーボードおよびマウスベースのボット式の設定] をクリックします。

7. [キーボードとマウスベースのボットの式の設定] セクションで、[追加] をクリックします。
8. [**NetScaler bot Management** プロファイルのボットのキーボードとマウス式のバインドの構成] ページで、次のパラメータを設定します。
   a) エクスプレッション名。検出キーボードとマウスのダイナミクスのボットポリシー式の名前。
   b) 式。ボットポリシー式。
   c) Enabled。チェックボックスを選択すると、キーボードとキーボードとマウスの両方のエクスプレッションバインディングが有効になります。
   d) [コメント]。ボットポリシー式とボットプロファイルへのバインドについての簡単な説明。
   e)「**OK**」をクリックして「閉じる」をクリックします。
9. [キーボードとマウスベースのボット式の設定] セクションで、[更新] をクリックします。

ボットトラフィックの詳細ロギング

着信要求がボットとして識別されると、NetScaler アプライアンスは監視とトラブルシューティングのために HTTP ヘッダーの詳細をさらにログに記録します。ボットの詳細ロギング機能は、Web App Firewall モジュールの詳細ロギングと似ています。

クライアントからの着信トラフィックを考えてみましょう。クライアントがボットとして識別された場合、NetScaler アプライアンスは詳細ログ機能を使用して、ドメインアドレス、URL、ユーザーエージェントヘッダー、Cookie ヘッダーなどの完全な HTTP ヘッダー情報を記録します。ログの詳細は ADM サーバーに送信され、目的の監視とトラブルシューティングを行います。詳細ログメッセージは「ns.log」ファイルに保存されません。

CLI を使用してボットの詳細ロギングを設定する

詳細な HTTP ヘッダー情報をログとしてキャプチャするには、ボットプロファイルの verbose ロギングパラメータを設定します。コマンドプロンプトで入力します:

```
set bot profile <name> [-verboseLogLevel ( NONE | HTTP_FULL_HEADER ) ]
<!--NeedCopy-->
```

例:

```
set bot profile p1 -verboseLogLevel HTTP_FULL_HEADER
```

NetScaler GUI を使用してボットの詳細ログを構成する

手順に従って、ボットプロファイルの詳細ログレベルを設定します。

1. ナビゲーションペインで、[セキュリティ] **> [NetScaler Bot**Management] に移動します。
2. **NetScaler Bot** 管理プロファイルページで、「追加」をクリックします。

3. **NetScaler Bot** 管理プロファイルの作成ページで、詳細なログレベルを **HTTP** フルヘッダーとして選択します。
4. [**OK**] をクリックし、[完了] をクリックします。

なりすましボットリクエストに対するアクションの設定

攻撃者は、優れたボットになりすましてアプリケーションサーバーにリクエストを送信しようとする可能性があります。このようなボットは、ボットシグネチャを使用してなりすましボットとして識別されます。なりすましボットに対して以下のアクションを設定して、アプリケーションサーバーを保護してください。

- DROP
- NONE
- リダイレクト
- RESET

CLI を使用してスプーフィングされたボットリクエストのアクションを設定する

以下のコマンドを実行して、なりすましボットリクエストのアクションを設定します。

```
set bot profile <bot-profile-name> -spoofedReqAction <action> LOG
<!--NeedCopy-->
```

例:

```
set bot profile bot_profile -spoofedReqAction DROP LOG
<!--NeedCopy-->
```

この例では、なりすましボットからの要求はドロップされ、NetScaler アプライアンスに記録されます。

> ヒントスプーフィングされたボットからのイベントをログに記録するには
>
> 、コマンドでLOGを指定します。

GUI を使用してスプーフィングされたボットリクエストのアクションを設定する

以下の手順に従って、スプーフィングされたボットリクエストに対するアクションを設定します。

1. [セキュリティ**] > [NetScaler** ボット管理] に移動します。
2. **NetScaler Bot** 管理プロファイルページで、「追加」をクリックします。
3. スプーフィングリクエストアクションリストからアクションを選択します。
4. 「なりすましリクエストをログに記録する」を選択します。

   このアクションは、なりすましボットからのイベントをログに記録します。
5. [**Create**] をクリックします。

NetScaler bot Management によってドロップされたリクエストヘッダー

キャッシュに関連するリクエストヘッダーの多くは、セッションのコンテキスト内のすべてのリクエストを表示するためにドロップされます。同様に、ウェブサーバーが圧縮された応答を送信できるようにするエンコーディングヘッダーが要求に含まれている場合、ボット管理者はこのヘッダーを削除して、圧縮されていないサーバー応答の内容をボット管理者が検査して JavaScript を挿入するようにします。

ボット管理者は以下のリクエストヘッダーを削除します。

範囲-失敗したファイル転送または部分的なファイル転送からの回復に使用されます。

If-Range-すでにキャッシュにそのオブジェクトの一部が含まれている場合、クライアントはオブジェクトの一部を取得できます (条件付き GET)。

If-Modified-Since-要求されたオブジェクトがこのフィールドで指定された時間以降に変更されていない場合、エンティティはサーバーから返されません。HTTP 304 変更されていないというエラーが表示されます。

If-None-Match-キャッシュされた情報を最小限のオーバーヘッドで効率的に更新できます。

Accept-Encoding-gzip などの特定のオブジェクトにはどのようなエンコーディング方法が許可されていますか。

ボット管理

August 15, 2023

NetScaler のボット管理の対象となるトラブルシューティングシナリオの一部を次に示します。

1. 偽陽性の場合の対処方法は?

   ボットの許可リスト機能を使用して誤検出のケースを管理でき、これらのトランザクションはバイパスできます。

2. 不正なボットトラフィックの詳細を調べるにはどうすればいいですか?

   監査ログ機能を使用して、不正なボットとして分類されたトラフィックの詳細を取得できます。

3. デフォルトの署名名を変更する必要があるのはなぜですか?

   NetScaler アプライアンスが提供するエンドポイントリソースで競合が検出された場合は、デフォルトの署名名を変更できます。

ボット管理

August 15, 2023

1. NetScaler ADC ボット管理とは何ですか?

   NetScaler ADC ボット管理は、良いボット、不良ボット、および人間のクライアントからのトラフィックを検出して区別します。ボット管理機能は、受信リクエストに設定されたアクションを適用することで、不正なボットからウェブアプリケーションを保護します。

2. NetScaler ADC が Web アプリケーションのボットを管理する必要があるのはなぜですか

   悪意のあるボットはあなたのインターネットトラフィックの 30% を構成します。悪意のあるボットは、DoS 攻撃の開始、電子メールアドレスのスパム、ダウンローダープログラムを使用したアプリケーションの速度低下、Web サイトからのコンテンツのダウンロードなど、さまざまな方法で Web アプリケーションに影響を与えます。さらに、ボットは、よく知られた検出メカニズムの一部を簡単に迂回し、データの損失、収益、および組織への評判につながります。

3. 着信ボットを検出するために使用される技術は何ですか?

   アプライアンスは、IP レピュテーション、レート制限、デバイスフィンガープリント、TPS、ボットトラップ検出などの検出技術を使用します。さらに、NetScaler GUI でカスタマイズされたブロックリストを構成して、組織固有の不良ボットを分類できます。

4. ボット署名ファイルとその目的は何ですか?

   ボットシグネチャファイルには、既知の良いボットと不良ボットのフットプリントが含まれています。シグネチャファイルは定期的に更新され、ボット保護を強化するために最新のボットシグネチャが組み込まれます。

5. どのような種類の NetScaler ADC ライセンスを購入する必要がありますか?

   ボット管理は ADC Premium ライセンスで利用できます。

6. トラブルシューティング用のボットログはどこで確認できますか?

   NetScaler ADC 監査ログには、検出されたボットの詳細が表示されます。詳細については、「監査ログ 」トピックを参照してください。

7. ボット署名ファイルの自動更新機能はありますか?

   はい、NetScaler ADC ボット管理は自動更新機能をサポートしています。

8. ボットの IP レピュテーション手法を使用するための前提条件はありますか?

   ボットプロファイルで IP レピュテーションを有効にして設定する前に、IP レピュテーション機能を有効にします。

ボット署名の自動更新

August 15, 2023

Bot シグニチャの自動更新機能により、最新のシグネチャを取得して、善良なボットと悪いボットの両方からより優れた保護とトラフィック管理を実現できます。

署名は 1 時間ごとに自動更新されるため、最新の更新が利用可能かどうかを常に確認する必要がなくなります。署名の自動更新機能を有効にした場合、NetScaler ADC アプライアンスは署名をホストしているサーバーに接続して、新しいバージョンが利用可能かどうかを確認します。

Amazon クラウドでホストされている最新のボット署名は、最新の更新を確認するためのデフォルトの署名 URL として設定されます。自動更新機能を使用するには、DNS サーバーが外部サイトにアクセスするように構成する必要もあります。

署名の更新

ボットのデフォルト署名オブジェクトを使用して作成されたすべてのユーザー定義署名オブジェクトには、0 より大きいバージョンがあります。署名の自動更新を有効にすると、すべての署名が自動的に更新されます。NetScaler ADC ボット管理 GUI の検索機能を使用して署名または署名のグループを選択することで、ボット署名のデフォルトアクションを更新できます。

ボット署名更新 URL: `https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json`

シグニチャ自動更新を構成する

シグニチャ自動更新機能を有効にするには、次のコマンドを実行する必要があります。

コマンドプロンプトで入力します。

```
set bot settings SignatureAutoUpdate ON
<!--NeedCopy-->
```

ボットシグネチャアラート記事

August 15, 2023

NetScaler のボット管理機能により、ダウンロードしてアプライアンスに適用できるシグネチャアップデートが通知されます。ボット攻撃を検出すると、新しい署名の更新に関する電子メール通知が届きます。署名をダウンロードして、アプライアンスに適用できます。

新しいボットシグネチャに関するアップデートを取得するには、シグニチャの自動更新機能を設定する必要があります。詳細については、「ボット署名の自動更新」トピックを参照してください。

2020 年 11 月のボット署名の更新

August 15, 2023

2020-11-11 週に特定されたボットに対して新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 5 は、NetScaler 13.0 プラットフォームに適用されます。

新しいボット署名

以下は、ボットシグネチャルール、カテゴリ、およびそのタイプの一覧です。

カテゴリ	ボットタイプ	署名カウント
スクレーパー	いいボット	3
マーケティング	グッドボット	23
フィードフェッチャー	グッドボット	2
ツール	悪いボット	3
検索エンジン	グッドボット	34
昇降補助具	グッドボット	6
未分類	悪いボット	6
ウイルススキャナー	グッドボット	1
スクリーンショットクリエーター	グッドボット	7
スクレーパー	悪いボット	1
ツール	グッドボット	7

2021 年 1 月のボット署名の更新

August 15, 2023

既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 6 は、13.0 61.x ビルド以降の NetScaler ADC プラットフォームに適用されます。

ボットの署名を更新しました

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
143	クローラー	いいボット
561	スクレーパー	いいボット
857	サイトモニター	いいボット
892	サイトモニター	悪いボット
894	サイトモニター	悪いボット
980	スクレーパー	悪いボット
1025	サイトモニター	悪いボット
1029	フィードフェッチャー	悪いボット
1030	スクリーンショットクリエーター	悪いボット
1034	ツール	悪いボット
1039	マーケティング	悪いボット
1042	サイトモニター	悪いボット
1047	サイトモニター	悪いボット
1053	サイトモニター	悪いボット
1072	検索エンジン	悪いボット
1073	フィードフェッチャー	悪いボット
1074	未分類	悪いボット
1078	スクリーンショットクリエーター	悪いボット
1109	マーケティング	悪いボット
1132	フィードフェッチャー	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1138	マーケティング	悪いボット
1150	検索エンジン	悪いボット
1164	検索エンジン	悪いボット
1167	マーケティング	悪いボット
1173	ツール	悪いボット
1174	マーケティング	悪いボット
1176	検索エンジン	悪いボット
1178	スピードテスター	悪いボット
1185	スクリーンショットクリエーター	悪いボット
1209	未分類	悪いボット
1244	サイトモニター	悪いボット
1251	検索エンジン	悪いボット
1254	サイトモニター	悪いボット
1256	未分類	悪いボット
1259	ツール	悪いボット
1287	検索エンジン	悪いボット
1296	検索エンジン	悪いボット
1312	未分類	悪いボット
1316	マーケティング	悪いボット
1322	サイトモニター	悪いボット
1325	スクリーンショットクリエーター	悪いボット
1328	検索エンジン	悪いボット
1330	マーケティング	悪いボット
1337	ツール	悪いボット
1360	検索エンジン	悪いボット
1367	検索エンジン	悪いボット
1374	ツール	悪いボット
1380	未分類	悪いボット
1388	検索エンジン	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1400	フィードフェッチャー	悪いボット
1413	未分類	悪いボット
1420	フィードフェッチャー	悪いボット
1422	サイトモニター	悪いボット
1442	未分類	悪いボット
1447	検索エンジン	悪いボット
1460	マーケティング	悪いボット
1467	ツール	悪いボット
1469	ツール	悪いボット
1471	検索エンジン	悪いボット
1484	未分類	悪いボット
1493	マーケティング	悪いボット
1502	サイトモニター	悪いボット
1504	未分類	悪いボット
1506	未分類	悪いボット
1518	未分類	悪いボット
1520	検索エンジン	悪いボット
1531	フィードフェッチャー	悪いボット
1533	未分類	悪いボット
1540	検索エンジン	悪いボット
1556	マーケティング	悪いボット
1560	未分類	悪いボット
1564	ツール	悪いボット
1570	サイトモニター	悪いボット
1575	検索エンジン	悪いボット
1586	ウイルススキャナー	悪いボット
1588	未分類	悪いボット
1594	ツール	悪いボット
1619	マーケティング	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1623	ツール	悪いボット
1626	検索エンジン	悪いボット
1632	フィードフェッチャー	悪いボット
1648	検索エンジン	悪いボット
1652	マーケティング	悪いボット
1660	マーケティング	悪いボット
1713	ツール	悪いボット
1719	検索エンジン	悪いボット
1722	未分類	悪いボット
1744	未分類	悪いボット
1754	未分類	悪いボット
1757	未分類	悪いボット
1762	未分類	悪いボット
1769	未分類	悪いボット
1771	マーケティング	悪いボット
1779	ツール	悪いボット
1782	ツール	悪いボット
1785	スピードテスター	悪いボット
1786	ツール	悪いボット
1792	サイトモニター	悪いボット
1869	ツール	悪いボット
1928	マーケティング	悪いボット
1942	サイトモニター	悪いボット
1949	マーケティング	悪いボット
1954	マーケティング	悪いボット
1964	未分類	悪いボット
1969	検索エンジン	悪いボット
2294	検索エンジン	悪いボット
2303	未分類	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2308	スクレーパー	悪いボット
2335	マーケティング	悪いボット
2374	未分類	悪いボット
2377	未分類	悪いボット
2385	ツール	悪いボット
2389	未分類	悪いボット
2414	未分類	悪いボット
2421	未分類	悪いボット
2424	未分類	悪いボット
2427	未分類	悪いボット
2429	検索エンジン	悪いボット
2437	未分類	悪いボット
2440	検索エンジン	悪いボット
2443	未分類	悪いボット
2453	マーケティング	悪いボット
2472	マーケティング	悪いボット
2474	フィードフェッチャー	悪いボット
2482	未分類	悪いボット
2500	スクリーンショットクリエーター	悪いボット
2503	未分類	悪いボット
2507	未分類	悪いボット
2516	ツール	悪いボット
2536	マーケティング	悪いボット
2543	ツール	悪いボット
2548	ツール	悪いボット
2557	マーケティング	悪いボット
2561	未分類	悪いボット
2572	未分類	悪いボット
2578	未分類	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2584	未分類	悪いボット
2588	未分類	悪いボット
2592	検索エンジン	悪いボット
2600	ツール	悪いボット
2606	未分類	悪いボット
2611	未分類	悪いボット
2622	ツール	悪いボット
2625	ツール	悪いボット
2631	ツール	悪いボット
2635	ツール	悪いボット
2637	スクリーンショットクリエーター	悪いボット
2641	検索エンジン	悪いボット
2655	未分類	悪いボット
2657	マーケティング	悪いボット
2663	未分類	悪いボット
2666	ツール	悪いボット
2672	フィードフェッチャー	悪いボット
2674	ツール	悪いボット
2681	検索エンジン	悪いボット
2684	マーケティング	悪いボット
2690	未分類	悪いボット
2704	未分類	悪いボット
2707	未分類	悪いボット
2714	フィードフェッチャー	悪いボット
2722	未分類	悪いボット
2726	フィードフェッチャー	悪いボット
2730	スクリーンショットクリエーター	悪いボット
2736	未分類	悪いボット
2749	未分類	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2753	ツール	悪いボット
2756	ツール	悪いボット
2760	スピードテスター	悪いボット
2780	ツール	悪いボット
2785	サイトモニター	悪いボット
2789	未分類	悪いボット
2797	ツール	悪いボット
2801	ツール	悪いボット
2808	ツール	悪いボット
2810	未分類	悪いボット
2813	未分類	悪いボット
2816	未分類	悪いボット
2820	リンクチェッカー	悪いボット
2824	リンクチェッカー	悪いボット
2831	スクリーンショットクリエーター	悪いボット
2843	ツール	悪いボット
2846	ツール	悪いボット
2849	マーケティング	悪いボット
2851	未分類	悪いボット
2855	未分類	悪いボット
2859	ツール	悪いボット
2873	未分類	悪いボット
2875	スクリーンショットクリエーター	悪いボット
2879	未分類	悪いボット
2881	未分類	悪いボット
2886	サイトモニター	悪いボット
2899	未分類	悪いボット
2916	未分類	悪いボット
2924	ツール	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2932	マーケティング	悪いボット
2935	リンクチェッカー	悪いボット
2939	マーケティング	悪いボット
2942	未分類	悪いボット
2955	検索エンジン	悪いボット
2960	ツール	悪いボット
2964	未分類	悪いボット
2972	マーケティング	悪いボット
2978	脆弱性スキャナ	悪いボット
2980	ツール	悪いボット
2985	マーケティング	悪いボット
2993	未分類	悪いボット
2999	スクリーンショットクリエーター	悪いボット
3003	フィードフェッチャー	悪いボット
3005	未分類	悪いボット
3013	未分類	悪いボット
3016	未分類	悪いボット
3021	検索エンジン	悪いボット
3026	未分類	悪いボット
3030	マーケティング	悪いボット
3065	マーケティング	悪いボット
3068	未分類	悪いボット
3072	マーケティング	悪いボット
3077	マーケティング	悪いボット
3080	未分類	悪いボット
3086	スクレーパー	悪いボット
3092	検索エンジン	悪いボット
3100	未分類	悪いボット
3104	ツール	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3111	未分類	悪いボット
3116	サイトモニター	悪いボット
3118	ツール	悪いボット
3120	マーケティング	悪いボット
3122	検索エンジン	悪いボット
3126	マーケティング	悪いボット
3141	ツール	悪いボット
3143	未分類	悪いボット
3145	スクレーパー	悪いボット
3150	未分類	悪いボット
3173	リンクチェッカー	悪いボット
3176	未分類	悪いボット
3186	スピードテスター	悪いボット
3190	スクレーパー	悪いボット
3203	検索エンジン	悪いボット
3216	未分類	悪いボット
3220	ツール	悪いボット
3223	リンクチェッカー	悪いボット
3241	未分類	悪いボット
3245	サイトモニター	悪いボット
3285	未分類	悪いボット
3304	マーケティング	悪いボット
3307	リンクチェッカー	悪いボット
3316	ツール	悪いボット
3326	マーケティング	悪いボット
3333	検索エンジン	悪いボット
3340	検索エンジン	悪いボット
3344	マーケティング	悪いボット
3350	未分類	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3355	マーケティング	悪いボット
3365	未分類	悪いボット
3378	未分類	悪いボット
3388	ツール	悪いボット
3396	未分類	悪いボット
3400	未分類	悪いボット
3421	未分類	悪いボット
3439	未分類	悪いボット
3447	フィードフェッチャー	悪いボット
3451	ツール	悪いボット
3459	スクリーンショットクリエーター	悪いボット
3469	脆弱性スキャナ	悪いボット
3475	未分類	悪いボット
3485	検索エンジン	悪いボット
3493	ツール	悪いボット
3502	マーケティング	悪いボット
3507	検索エンジン	悪いボット
3523	未分類	悪いボット
3535	スピードテスター	悪いボット
3549	未分類	悪いボット
3556	未分類	悪いボット
3561	未分類	悪いボット
3565	未分類	悪いボット
3572	検索エンジン	悪いボット
3578	未分類	悪いボット
3610	検索エンジン	悪いボット
3617	未分類	悪いボット
3621	マーケティング	悪いボット
3632	ツール	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3635	マーケティング	悪いボット
3653	未分類	悪いボット
3661	検索エンジン	悪いボット
3704	未分類	悪いボット
3707	未分類	悪いボット
3711	未分類	悪いボット
3730	検索エンジン	悪いボット
3740	サイトモニター	悪いボット
3759	検索エンジン	悪いボット
3764	未分類	悪いボット
3770	未分類	悪いボット

2021 年 **3** 月のボットシグネチャの更新

August 15, 2023

既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 7 は、13.0 61.x ビルド以降の NetScaler ADC プラットフォームに適用されます。

ボットの署名を更新しました

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
278	スクレーパー	いいボット
378	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
379	スクレーパー	いいボット
380	スクレーパー	いいボット
381	スクレーパー	いいボット
382	スクレーパー	いいボット
383	スクレーパー	いいボット
384	スクレーパー	いいボット
385	スクレーパー	いいボット
386	スクレーパー	いいボット
387	スクレーパー	いいボット
389	スクレーパー	いいボット
390	スクレーパー	いいボット
391	スクレーパー	いいボット
494	スクレーパー	いいボット
627	検索エンジン	いいボット
660	検索エンジン	いいボット
3840	クローラー	いいボット

2021 年 **8** 月のボット署名の更新

August 15, 2023

新しい署名が追加され、既存のボット署名の一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 8 は、13.0 61.x ビルド以降の NetScaler ADC プラットフォームに適用されます。

ボットの署名を更新しました

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
236	スクレーパー	いいボット
378	スクレーパー	いいボット
381	スクレーパー	いいボット
382	スクレーパー	いいボット
390	スクレーパー	いいボット
544	スクレーパー	いいボット
702	検索エンジン	いいボット
979	スクレーパー	悪いボット
3791	スピードテスター	いいボット
3797	マーケティング	いいボット
3800	マーケティング	いいボット
3824	クローラー	悪いボット
3833	検索エンジン	いいボット
3849	クローラー	いいボット
3871	マーケティング	いいボット
3963	マーケティング	いいボット
4027	検索エンジン	いいボット

新しいボット署名

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4028	マーケティング	いいボット
4029	ツール	いいボット
4030	スクレーパー	いいボット
4031	スクレーパー	いいボット
4032	未分類	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4033	クローラー	いいボット
4034	クローラー	いいボット
4035	マーケティング	いいボット
4036	脆弱性スキャナ	いいボット
4037	脆弱性スキャナ	いいボット
4038	未分類	悪いボット
4039	ツール	いいボット
4040	クローラー	いいボット
4041	ツール	いいボット
4042	クローラー	いいボット
4043	スクリーンショットクリエーター	いいボット
4044	スクレーパー	悪いボット
4045	スクレーパー	悪いボット
4046	スクレーパー	悪いボット
4047	未分類	悪いボット
4048	フィードフェッチャー	いいボット
4049	未分類	悪いボット
4050	クローラー	いいボット
4051	クローラー	いいボット
4052	ツール	いいボット
4053	ツール	いいボット
4054	スクレーパー	悪いボット
4055	未分類	いいボット
4056	マーケティング	いいボット
4057	スクリーンショットクリエーター	いいボット
4058	クローラー	いいボット
4059	未分類	悪いボット
4060	検索エンジン	いいボット
4061	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4062	検索エンジン	いいボット
4063	検索エンジン	いいボット
4064	ツール	いいボット
4065	スクレーパー	いいボット
4066	マーケティング	いいボット
4067	マーケティング	いいボット
4068	未分類	悪いボット
4069	未分類	悪いボット
4070	未分類	悪いボット
4071	ツール	いいボット
4072	ツール	悪いボット
4073	未分類	悪いボット
4074	未分類	悪いボット
4075	ツール	悪いボット
4076	マーケティング	いいボット
4077	スクレーパー	いいボット
4078	クローラー	いいボット
4079	クローラー	いいボット
4080	ツール	悪いボット
4081	検索エンジン	いいボット
4082	ツール	いいボット
4083	未分類	悪いボット
4084	未分類	悪いボット
4085	ツール	いいボット
4086	ツール	いいボット
4087	ツール	悪いボット
4088	検索エンジン	いいボット
4089	マーケティング	いいボット
4090	ツール	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4091	ツール	いいボット
4092	ツール	いいボット
4093	ツール	いいボット
4094	未分類	いいボット
4095	サイトモニター	いいボット
4096	サイトモニター	いいボット
4097	サイトモニター	いいボット
4098	クローラー	いいボット
4099	検索エンジン	いいボット
4100	検索エンジン	いいボット
4101	検索エンジン	いいボット
4102	検索エンジン	いいボット
4103	マーケティング	いいボット
4104	マーケティング	いいボット
4105	マーケティング	いいボット
4106	マーケティング	いいボット
4107	マーケティング	いいボット
4108	マーケティング	いいボット
4109	検索エンジン	いいボット
4110	クローラー	いいボット
4111	クローラー	いいボット
4112	クローラー	いいボット
4113	脆弱性スキャナ	いいボット
4114	クローラー	いいボット
4115	ツール	いいボット
4116	未分類	悪いボット
4117	未分類	悪いボット
4118	未分類	悪いボット
4119	未分類	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4120	マーケティング	いいボット
4121	マーケティング	いいボット
4122	マーケティング	いいボット
4123	マーケティング	いいボット
4124	マーケティング	いいボット
4125	マーケティング	いいボット
4126	マーケティング	いいボット
4127	マーケティング	いいボット
4128	マーケティング	いいボット
4129	マーケティング	いいボット
4130	マーケティング	いいボット
4131	ツール	いいボット
4132	マーケティング	いいボット
4133	マーケティング	いいボット
4134	ツール	いいボット
4135	マーケティング	いいボット
4136	マーケティング	いいボット
4137	マーケティング	いいボット
4138	マーケティング	いいボット
4139	マーケティング	いいボット
4140	マーケティング	いいボット
4141	マーケティング	いいボット
4142	マーケティング	いいボット
4143	マーケティング	いいボット
4144	マーケティング	いいボット
4145	検索エンジン	いいボット
4146	検索エンジン	いいボット
4147	検索エンジン	いいボット
4148	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4149	検索エンジン	いいボット
4150	検索エンジン	いいボット
4151	検索エンジン	いいボット
4152	検索エンジン	いいボット
4153	検索エンジン	いいボット
4154	検索エンジン	いいボット
4155	検索エンジン	いいボット
4156	スクリーンショットクリエーター	いいボット
4157	検索エンジン	いいボット
4158	検索エンジン	いいボット
4159	検索エンジン	いいボット
4160	スクリーンショットクリエーター	いいボット
4161	検索エンジン	いいボット
4162	検索エンジン	いいボット
4163	ツール	いいボット
4164	検索エンジン	いいボット
4165	マーケティング	いいボット
4166	未分類	悪いボット
4167	ツール	悪いボット
4168	スピードテスター	いいボット
4169	スクレーパー	悪いボット
4170	ツール	いいボット
4171	スクレーパー	悪いボット
4172	ウェブクローラ	いいボット
4173	ツール	いいボット
4174	クローラー	いいボット
4175	クローラー	いいボット
4176	ツール	いいボット
4177	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4178	ツール	いいボット
4179	ウェブクローラ	いいボット
4180	ツール	いいボット
4181	サイトモニター	いいボット
4182	サイトモニター	いいボット
4183	サイトモニター	いいボット
4184	サイトモニター	いいボット
4185	検索エンジン	いいボット
4186	ツール	いいボット
4187	ツール	いいボット
4188	スクリーンショットクリエーター	いいボット
4189	マーケティング	いいボット
4190	検索エンジン	いいボット
4191	検索エンジン	いいボット
4192	検索エンジン	いいボット
4193	検索エンジン	いいボット
4194	ツール	いいボット
4195	検索エンジン	悪いボット
4196	ツール	いいボット
4197	ツール	いいボット
4198	マーケティング	いいボット
4199	マーケティング	いいボット
4200	脆弱性スキャナ	いいボット
4201	ツール	いいボット
4202	ツール	いいボット
4203	未分類	悪いボット
4204	未分類	悪いボット
4205	検索エンジン	いいボット
4206	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4207	マーケティング	いいボット
4208	検索エンジン	いいボット
4209	検索エンジン	いいボット
4210	スピードテスター	いいボット
4211	ツール	いいボット
4212	フィードフェッチャー	いいボット
4213	フィードフェッチャー	いいボット
4214	スクレーパー	悪いボット
4215	ツール	いいボット
4216	ツール	いいボット
4217	ツール	悪いボット
4218	スクレーパー	悪いボット
4219	マーケティング	いいボット
4220	ツール	いいボット
4221	ツール	悪いボット
4222	サイトモニター	いいボット
4223	マーケティング	いいボット
4224	検索エンジン	いいボット
4225	検索エンジン	いいボット
4226	検索エンジン	いいボット
4227	マーケティング	いいボット
4228	マーケティング	いいボット
4229	ツール	いいボット
4230	未分類	悪いボット
4231	スクリーンショットクリエーター	いいボット
4232	ツール	いいボット
4233	サイトモニター	いいボット
4234	サイトモニター	いいボット
4235	サイトモニター	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4236	サイトモニター	いいボット
4237	サイトモニター	いいボット
4238	サイトモニター	いいボット
4239	未分類	悪いボット
4240	マーケティング	いいボット
4241	マーケティング	いいボット
4242	マーケティング	いいボット
4243	マーケティング	いいボット
4244	マーケティング	いいボット
4245	マーケティング	いいボット
4246	マーケティング	いいボット
4247	検索エンジン	いいボット
4248	検索エンジン	いいボット
4249	スクリーンショットクリエーター	いいボット
4250	検索エンジン	いいボット
4251	検索エンジン	いいボット
4252	クローラー	いいボット
4253	クローラー	いいボット
4254	クローラー	いいボット
4255	ツール	いいボット
4256	未分類	いいボット
4257	ツール	いいボット
4258	クローラー	いいボット
4259	クローラー	いいボット
4260	ツール	いいボット
4261	ツール	いいボット
4262	ツール	いいボット
4263	マーケティング	いいボット
4264	クローラー	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4265	検索エンジン	いいボット
4266	未分類	いいボット
4267	ツール	いいボット
4268	ツール	いいボット
4269	検索エンジン	いいボット
4270	検索エンジン	いいボット
4271	検索エンジン	いいボット
4272	検索エンジン	いいボット
4273	検索エンジン	いいボット
4274	検索エンジン	いいボット
4275	検索エンジン	いいボット
4276	未分類	悪いボット
4277	未分類	悪いボット
4278	未分類	悪いボット
4279	マーケティング	いいボット
4280	クローラー	いいボット
4281	未分類	悪いボット
4282	マーケティング	いいボット
4283	マーケティング	いいボット
4284	マーケティング	いいボット
4285	マーケティング	いいボット
4286	マーケティング	いいボット
4287	マーケティング	いいボット
4288	マーケティング	いいボット
4289	マーケティング	いいボット
4290	マーケティング	いいボット
4291	マーケティング	いいボット
4292	マーケティング	いいボット
4293	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4294	マーケティング	いいボット
4295	検索エンジン	いいボット
4296	検索エンジン	いいボット
4297	検索エンジン	いいボット
4298	検索エンジン	いいボット
4299	検索エンジン	いいボット
4300	検索エンジン	いいボット
4301	検索エンジン	いいボット
4302	検索エンジン	いいボット
4303	検索エンジン	いいボット
4304	検索エンジン	いいボット
4305	検索エンジン	いいボット
4306	スクリーンショットクリエーター	いいボット
4307	検索エンジン	いいボット
4308	検索エンジン	いいボット
4309	検索エンジン	いいボット
4310	検索エンジン	いいボット
4311	スクリーンショットクリエーター	いいボット
4312	検索エンジン	いいボット
4313	検索エンジン	いいボット
4314	検索エンジン	いいボット
4315	検索エンジン	いいボット
4316	検索エンジン	いいボット
4317	検索エンジン	いいボット
4318	スクリーンショットクリエーター	いいボット
4319	スクリーンショットクリエーター	いいボット
4320	未分類	悪いボット
4321	未分類	いいボット
4322	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4323	ツール	いいボット
4324	ツール	いいボット
4325	ツール	いいボット
4326	スクレーパー	悪いボット
4327	検索エンジン	いいボット
4328	マーケティング	いいボット
4329	未分類	悪いボット
4330	サイトモニター	いいボット
4331	検索エンジン	いいボット
4332	検索エンジン	いいボット
4333	未分類	悪いボット
4334	スクレーパー	いいボット
4335	マーケティング	いいボット
4336	マーケティング	いいボット
4337	ツール	いいボット
4338	ツール	いいボット
4339	ツール	いいボット
4340	クローラー	いいボット
4341	クローラー	いいボット
4342	脆弱性スキャナ	いいボット
4343	脆弱性スキャナ	いいボット
4344	スクレーパー	いいボット
4345	マーケティング	いいボット
4346	マーケティング	いいボット
4347	マーケティング	いいボット
4348	マーケティング	いいボット
4349	マーケティング	いいボット
4350	マーケティング	いいボット
4351	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4352	マーケティング	いいボット
4353	マーケティング	いいボット
4354	マーケティング	いいボット
4355	検索エンジン	いいボット
4356	検索エンジン	いいボット
4357	検索エンジン	いいボット
4358	検索エンジン	いいボット
4359	検索エンジン	いいボット
4360	検索エンジン	いいボット
4361	検索エンジン	いいボット
4362	検索エンジン	いいボット
4363	検索エンジン	いいボット
4364	検索エンジン	いいボット
4365	スクリーンショットクリエーター	いいボット
4366	検索エンジン	いいボット
4367	検索エンジン	いいボット
4368	検索エンジン	いいボット
4369	検索エンジン	いいボット
4370	スクリーンショットクリエーター	いいボット
4371	検索エンジン	いいボット
4372	検索エンジン	いいボット
4373	検索エンジン	いいボット
4374	検索エンジン	いいボット
4375	検索エンジン	いいボット
4376	スクリーンショットクリエーター	いいボット
4377	クローラー	いいボット
4378	クローラー	いいボット
4379	検索エンジン	いいボット
4380	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4381	検索エンジン	いいボット
4382	検索エンジン	いいボット
4383	クローラー	いいボット
4384	検索エンジン	いいボット
4385	ツール	いいボット
4386	未分類	いいボット
4387	クローラー	いいボット
4388	クローラー	いいボット
4389	ツール	いいボット
4390	ツール	いいボット
4391	ツール	いいボット
4392	ツール	いいボット
4393	ツール	いいボット
4394	未分類	いいボット
4395	ツール	いいボット
4396	サイトモニター	いいボット
4397	サイトモニター	いいボット
4398	ツール	悪いボット
4399	ツール	悪いボット
4400	ツール	悪いボット
4401	ツール	悪いボット
4402	ツール	悪いボット
4403	ツール	悪いボット
4404	検索エンジン	いいボット
4405	検索エンジン	いいボット
4406	検索エンジン	いいボット
4407	未分類	いいボット

2021 年 9 月のボットシグネチャアップデート

August 15, 2023

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 9 は、13.0 61.48 以降のビルドを持つ NetScaler ADC プラットフォームに適用されます。

ボットの署名を更新しました

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2	クローラー	いいボット
5	クローラー	いいボット
9	クローラー	いいボット
45	クローラー	いいボット
46	クローラー	いいボット
48	クローラー	いいボット
52	クローラー	いいボット
60	クローラー	いいボット
61	クローラー	いいボット
63	クローラー	いいボット
67	クローラー	いいボット
71	クローラー	いいボット
74	クローラー	いいボット
75	クローラー	いいボット
76	クローラー	いいボット
78	クローラー	いいボット
79	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
80	クローラー	いいボット
81	クローラー	いいボット
82	クローラー	いいボット
83	クローラー	いいボット
84	クローラー	いいボット
87	クローラー	いいボット
90	クローラー	いいボット
95	クローラー	いいボット
96	クローラー	いいボット
97	クローラー	いいボット
100	クローラー	いいボット
101	クローラー	いいボット
102	クローラー	いいボット
103	クローラー	いいボット
104	クローラー	いいボット
107	クローラー	いいボット
108	クローラー	いいボット
110	クローラー	いいボット
111	クローラー	いいボット
114	クローラー	いいボット
115	クローラー	いいボット
123	クローラー	いいボット
135	クローラー	いいボット
136	クローラー	いいボット
137	クローラー	いいボット
140	クローラー	いいボット
141	クローラー	いいボット
143	クローラー	いいボット
144	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
145	クローラー	いいボット
146	クローラー	いいボット
147	クローラー	いいボット
149	クローラー	いいボット
152	クローラー	いいボット
155	クローラー	いいボット
156	クローラー	いいボット
157	クローラー	いいボット
158	クローラー	いいボット
159	クローラー	いいボット
160	クローラー	いいボット
161	クローラー	いいボット
162	クローラー	いいボット
163	クローラー	いいボット
164	クローラー	いいボット
165	クローラー	いいボット
166	クローラー	いいボット
167	クローラー	いいボット
172	クローラー	いいボット
173	クローラー	いいボット
174	クローラー	いいボット
176	クローラー	いいボット
177	クローラー	いいボット
180	クローラー	いいボット
187	クローラー	いいボット
197	クローラー	いいボット
201	クローラー	いいボット
202	クローラー	いいボット
203	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
206	クローラー	いいボット
211	フィードフェッチャー	悪いボット
217	フィードフェッチャー	いいボット
219	フィードフェッチャー	いいボット
229	スクレーパー	いいボット
235	スクレーパー	いいボット
236	スクレーパー	いいボット
237	スクレーパー	いいボット
248	スクレーパー	いいボット
250	スクレーパー	いいボット
260	スクレーパー	いいボット
263	スクレーパー	いいボット
265	スクレーパー	いいボット
267	スクレーパー	いいボット
268	スクレーパー	いいボット
271	スクレーパー	いいボット
272	スクレーパー	いいボット
276	スクレーパー	いいボット
277	スクレーパー	いいボット
278	スクレーパー	いいボット
279	スクレーパー	いいボット
280	スクレーパー	いいボット
281	スクレーパー	いいボット
283	スクレーパー	いいボット
285	スクレーパー	いいボット
286	スクレーパー	いいボット
287	スクレーパー	いいボット
290	スクレーパー	いいボット
292	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
293	スクレーパー	いいボット
342	スクレーパー	いいボット
343	スクレーパー	いいボット
344	スクレーパー	いいボット
355	スクレーパー	いいボット
357	スクレーパー	いいボット
360	スクレーパー	いいボット
362	スクレーパー	いいボット
366	スクレーパー	いいボット
370	スクレーパー	いいボット
371	スクレーパー	いいボット
372	スクレーパー	いいボット
373	スクレーパー	いいボット
374	スクレーパー	いいボット
376	スクレーパー	いいボット
377	スクレーパー	いいボット
380	スクレーパー	いいボット
392	スクレーパー	いいボット
393	スクレーパー	いいボット
394	スクレーパー	いいボット
396	スクレーパー	いいボット
397	スクレーパー	いいボット
414	スクレーパー	いいボット
418	スクレーパー	いいボット
419	スクレーパー	いいボット
421	スクレーパー	いいボット
422	スクレーパー	いいボット
423	スクレーパー	いいボット
424	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
425	スクレーパー	いいボット
426	スクレーパー	いいボット
427	スクレーパー	いいボット
428	スクレーパー	いいボット
430	スクレーパー	いいボット
432	スクレーパー	いいボット
433	スクレーパー	いいボット
434	スクレーパー	いいボット
435	スクレーパー	いいボット
441	スクレーパー	いいボット
445	スクレーパー	いいボット
446	スクレーパー	いいボット
451	スクレーパー	いいボット
452	スクレーパー	いいボット
454	スクレーパー	いいボット
455	スクレーパー	いいボット
456	スクレーパー	いいボット
457	スクレーパー	いいボット
458	スクレーパー	いいボット
461	スクレーパー	いいボット
465	スクレーパー	いいボット
466	スクレーパー	いいボット
469	スクレーパー	いいボット
473	スクレーパー	いいボット
474	スクレーパー	いいボット
476	スクレーパー	いいボット
477	スクレーパー	いいボット
484	スクレーパー	いいボット
485	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
487	スクレーパー	いいボット
488	スクレーパー	いいボット
489	スクレーパー	いいボット
490	スクレーパー	いいボット
493	スクレーパー	いいボット
494	スクレーパー	いいボット
495	スクレーパー	いいボット
497	スクレーパー	いいボット
498	スクレーパー	いいボット
499	スクレーパー	いいボット
500	スクレーパー	いいボット
505	スクレーパー	いいボット
506	スクレーパー	いいボット
507	スクレーパー	いいボット
512	スクレーパー	いいボット
513	スクレーパー	いいボット
514	スクレーパー	いいボット
527	スクレーパー	いいボット
533	スクレーパー	いいボット
539	スクレーパー	いいボット
540	スクレーパー	いいボット
542	スクレーパー	いいボット
544	スクレーパー	いいボット
545	スクレーパー	いいボット
546	スクレーパー	いいボット
547	スクレーパー	いいボット
548	スクレーパー	いいボット
551	スクレーパー	いいボット
552	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
554	スクレーパー	いいボット
556	スクレーパー	いいボット
558	スクレーパー	いいボット
560	スクレーパー	いいボット
561	スクレーパー	いいボット
566	スクレーパー	いいボット
575	スクレーパー	いいボット
578	スクレーパー	いいボット
581	スクレーパー	いいボット
591	スクレーパー	いいボット
593	スクレーパー	いいボット
595	スクレーパー	いいボット
600	スクレーパー	いいボット
601	スクレーパー	いいボット
602	スクレーパー	いいボット
604	スクレーパー	いいボット
605	スクレーパー	いいボット
609	スクレーパー	いいボット
610	スクレーパー	いいボット
611	スクレーパー	いいボット
612	スクレーパー	いいボット
613	スクレーパー	いいボット
615	スクレーパー	いいボット
620	検索エンジン	いいボット
622	検索エンジン	いいボット
623	検索エンジン	いいボット
624	検索エンジン	いいボット
626	検索エンジン	いいボット
627	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
628	検索エンジン	いいボット
629	検索エンジン	いいボット
633	検索エンジン	いいボット
634	検索エンジン	いいボット
636	検索エンジン	いいボット
637	検索エンジン	いいボット
639	検索エンジン	いいボット
640	検索エンジン	いいボット
641	検索エンジン	いいボット
642	検索エンジン	いいボット
643	検索エンジン	いいボット
647	検索エンジン	いいボット
649	検索エンジン	いいボット
650	検索エンジン	いいボット
651	検索エンジン	いいボット
654	検索エンジン	いいボット
656	検索エンジン	いいボット
657	検索エンジン	いいボット
658	検索エンジン	いいボット
659	検索エンジン	いいボット
660	検索エンジン	いいボット
663	検索エンジン	いいボット
664	検索エンジン	いいボット
665	検索エンジン	いいボット
666	検索エンジン	いいボット
667	検索エンジン	いいボット
669	検索エンジン	いいボット
670	検索エンジン	いいボット
671	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
672	検索エンジン	いいボット
673	検索エンジン	いいボット
674	検索エンジン	いいボット
675	検索エンジン	いいボット
676	検索エンジン	いいボット
677	検索エンジン	いいボット
679	検索エンジン	いいボット
680	検索エンジン	いいボット
690	検索エンジン	いいボット
693	検索エンジン	いいボット
694	検索エンジン	いいボット
697	検索エンジン	いいボット
698	検索エンジン	いいボット
703	検索エンジン	いいボット
706	検索エンジン	いいボット
712	検索エンジン	いいボット
714	検索エンジン	いいボット
715	検索エンジン	いいボット
716	検索エンジン	いいボット
721	検索エンジン	いいボット
723	検索エンジン	いいボット
725	検索エンジン	いいボット
727	検索エンジン	いいボット
728	検索エンジン	いいボット
729	検索エンジン	いいボット
730	検索エンジン	いいボット
731	検索エンジン	いいボット
732	検索エンジン	いいボット
735	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
736	検索エンジン	いいボット
740	検索エンジン	いいボット
748	検索エンジン	いいボット
749	検索エンジン	いいボット
750	検索エンジン	いいボット
751	検索エンジン	いいボット
756	検索エンジン	いいボット
757	検索エンジン	いいボット
758	検索エンジン	いいボット
759	検索エンジン	いいボット
760	検索エンジン	いいボット
761	検索エンジン	いいボット
762	検索エンジン	いいボット
763	検索エンジン	いいボット
764	検索エンジン	いいボット
765	検索エンジン	いいボット
766	検索エンジン	いいボット
767	検索エンジン	いいボット
768	検索エンジン	いいボット
769	検索エンジン	いいボット
770	検索エンジン	いいボット
771	検索エンジン	いいボット
772	検索エンジン	いいボット
773	検索エンジン	いいボット
776	検索エンジン	いいボット
777	検索エンジン	いいボット
780	検索エンジン	いいボット
781	検索エンジン	いいボット
784	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
786	検索エンジン	いいボット
787	検索エンジン	いいボット
788	検索エンジン	いいボット
789	検索エンジン	いいボット
790	検索エンジン	いいボット
791	検索エンジン	いいボット
792	検索エンジン	いいボット
795	検索エンジン	いいボット
796	検索エンジン	いいボット
798	検索エンジン	いいボット
800	検索エンジン	いいボット
801	検索エンジン	いいボット
802	検索エンジン	いいボット
803	検索エンジン	いいボット
805	検索エンジン	いいボット
806	検索エンジン	いいボット
807	検索エンジン	いいボット
809	検索エンジン	いいボット
810	検索エンジン	いいボット
811	検索エンジン	いいボット
812	検索エンジン	いいボット
814	検索エンジン	いいボット
815	検索エンジン	いいボット
816	検索エンジン	いいボット
817	検索エンジン	いいボット
818	検索エンジン	いいボット
819	検索エンジン	いいボット
820	検索エンジン	いいボット
821	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
822	検索エンジン	いいボット
823	検索エンジン	いいボット
825	検索エンジン	いいボット
827	検索エンジン	いいボット
830	検索エンジン	いいボット
831	検索エンジン	いいボット
834	検索エンジン	いいボット
837	検索エンジン	いいボット
838	検索エンジン	いいボット
849	サイトモニター	いいボット
850	サイトモニター	いいボット
851	サイトモニター	いいボット
853	サイトモニター	いいボット
857	サイトモニター	いいボット
858	サイトモニター	いいボット
859	サイトモニター	いいボット
860	サイトモニター	いいボット
861	サイトモニター	いいボット
862	サイトモニター	いいボット
863	サイトモニター	いいボット
864	サイトモニター	いいボット
865	サイトモニター	いいボット
866	サイトモニター	いいボット
867	サイトモニター	いいボット
868	サイトモニター	いいボット
869	サイトモニター	いいボット
870	サイトモニター	いいボット
871	サイトモニター	いいボット
872	サイトモニター	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
873	サイトモニター	いいボット
874	サイトモニター	いいボット
875	サイトモニター	いいボット
876	サイトモニター	いいボット
877	サイトモニター	いいボット
880	サイトモニター	いいボット
883	サイトモニター	いいボット
885	サイトモニター	いいボット
886	サイトモニター	いいボット
888	サイトモニター	いいボット
889	サイトモニター	いいボット
895	サイトモニター	いいボット
896	サイトモニター	いいボット
897	サイトモニター	いいボット
898	サイトモニター	いいボット
900	サイトモニター	いいボット
901	サイトモニター	いいボット
904	サイトモニター	いいボット
906	サイトモニター	いいボット
908	サイトモニター	いいボット
909	サイトモニター	いいボット
910	サイトモニター	いいボット
911	サイトモニター	いいボット
912	サイトモニター	いいボット
913	サイトモニター	いいボット
917	サイトモニター	いいボット
918	サイトモニター	いいボット
919	サイトモニター	いいボット
920	サイトモニター	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
921	サイトモニター	いいボット
924	サイトモニター	いいボット
926	サイトモニター	いいボット
927	サイトモニター	いいボット
928	サイトモニター	いいボット
929	サイトモニター	いいボット
930	サイトモニター	いいボット
931	サイトモニター	いいボット
938	サイトモニター	いいボット
939	サイトモニター	いいボット
943	サイトモニター	悪いボット
958	サイトモニター	いいボット
959	サイトモニター	いいボット
960	サイトモニター	いいボット
963	サイトモニター	いいボット
984	スクレーパー	いいボット
996	スクレーパー	いいボット
997	スクレーパー	いいボット
998	スクレーパー	いいボット
1002	スクレーパー	いいボット
1006	スクレーパー	いいボット
1588	未分類	悪いボット
2561	スクレーパー	悪いボット
2810	クローラー	いいボット
3782	マーケティング	いいボット
3783	検索エンジン	いいボット
3788	ツール	いいボット
3789	ツール	いいボット
3790	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3792	ツール	いいボット
3793	ツール	いいボット
3794	クローラー	いいボット
3796	スクレーパー	いいボット
3798	マーケティング	いいボット
3799	マーケティング	いいボット
3801	マーケティング	いいボット
3802	スクリーンショットクリエーター	いいボット
3803	検索エンジン	いいボット
3804	スクリーンショットクリエーター	いいボット
3805	検索エンジン	いいボット
3806	ツール	いいボット
3807	クローラー	いいボット
3808	クローラー	いいボット
3809	ツール	いいボット
3810	スクレーパー	いいボット
3811	ツール	いいボット
3813	ツール	いいボット
3814	クローラー	いいボット
3815	未分類	いいボット
3817	ツール	いいボット
3818	ツール	いいボット
3819	ツール	いいボット
3820	クローラー	いいボット
3821	検索エンジン	いいボット
3822	マーケティング	いいボット
3823	未分類	いいボット
3831	スクレーパー	いいボット
3834	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3835	検索エンジン	いいボット
3836	未分類	いいボット
3837	未分類	いいボット
3838	未分類	いいボット
3839	マーケティング	いいボット
3840	クローラー	いいボット
3842	クローラー	いいボット
3843	クローラー	いいボット
3844	マーケティング	いいボット
3845	マーケティング	いいボット
3846	マーケティング	いいボット
3847	マーケティング	いいボット
3848	未分類	いいボット
3850	ツール	いいボット
3851	未分類	いいボット
3852	ツール	いいボット
3853	脆弱性スキャナ	いいボット
3854	クローラー	いいボット
3855	クローラー	いいボット
3856	ツール	いいボット
3861	マーケティング	いいボット
3862	マーケティング	いいボット
3863	マーケティング	いいボット
3864	マーケティング	いいボット
3865	マーケティング	いいボット
3866	マーケティング	いいボット
3867	マーケティング	いいボット
3868	マーケティング	いいボット
3869	ツール	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3870	マーケティング	いいボット
3872	マーケティング	いいボット
3873	検索エンジン	いいボット
3874	検索エンジン	いいボット
3875	検索エンジン	いいボット
3876	検索エンジン	いいボット
3877	スクリーンショットクリエーター	いいボット
3878	検索エンジン	いいボット
3879	検索エンジン	いいボット
3880	スクリーンショットクリエーター	いいボット
3881	スクリーンショットクリエーター	いいボット
3882	検索エンジン	いいボット
3883	検索エンジン	いいボット
3884	検索エンジン	いいボット
3885	検索エンジン	いいボット
3886	ツール	いいボット
3887	クローラー	いいボット
3888	クローラー	いいボット
3889	未分類	いいボット
3890	マーケティング	いいボット
3893	クローラー	いいボット
3894	ツール	いいボット
3895	ツール	いいボット
3896	検索エンジン	いいボット
3897	ツール	いいボット
3898	ツール	いいボット
3899	未分類	いいボット
3901	クローラー	いいボット
3903	ツール	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3904	検索エンジン	いいボット
3905	検索エンジン	いいボット
3906	検索エンジン	いいボット
3912	クローラー	いいボット
3918	クローラー	いいボット
3919	未分類	いいボット
3920	未分類	いいボット
3921	未分類	いいボット
3922	未分類	いいボット
3923	未分類	いいボット
3924	未分類	いいボット
3925	未分類	いいボット
3926	マーケティング	いいボット
3927	マーケティング	いいボット
3928	マーケティング	いいボット
3929	ツール	いいボット
3930	マーケティング	いいボット
3931	未分類	いいボット
3932	クローラー	いいボット
3933	マーケティング	いいボット
3934	マーケティング	いいボット
3935	スクレーパー	いいボット
3936	マーケティング	いいボット
3937	スクレーパー	いいボット
3938	フィードフェッチャー	いいボット
3940	検索エンジン	いいボット
3941	クローラー	いいボット
3942	スクレーパー	いいボット
3946	フィードフェッチャー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3947	クローラー	いいボット
3950	ウイルススキャナー	いいボット
3951	マーケティング	いいボット
3952	マーケティング	いいボット
3953	マーケティング	いいボット
3954	マーケティング	いいボット
3955	マーケティング	いいボット
3956	マーケティング	いいボット
3957	マーケティング	いいボット
3958	マーケティング	いいボット
3959	マーケティング	いいボット
3960	マーケティング	いいボット
3961	マーケティング	いいボット
3962	マーケティング	いいボット
3964	マーケティング	いいボット
3965	マーケティング	いいボット
3966	マーケティング	いいボット
3967	マーケティング	いいボット
3968	マーケティング	いいボット
3969	マーケティング	いいボット
3970	検索エンジン	いいボット
3971	スクリーンショットクリエーター	いいボット
3972	スクリーンショットクリエーター	いいボット
3973	検索エンジン	いいボット
3974	検索エンジン	いいボット
3975	検索エンジン	いいボット
3976	検索エンジン	いいボット
3977	検索エンジン	いいボット
3978	スクリーンショットクリエーター	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3979	検索エンジン	いいボット
3980	スクリーンショットクリエーター	いいボット
3981	検索エンジン	いいボット
3982	検索エンジン	いいボット
3983	検索エンジン	いいボット
3984	検索エンジン	いいボット
3985	検索エンジン	いいボット
3986	検索エンジン	いいボット
3987	スクリーンショットクリエーター	いいボット
3988	検索エンジン	いいボット
3989	検索エンジン	いいボット
3990	検索エンジン	いいボット
3991	検索エンジン	いいボット
3992	検索エンジン	いいボット
3993	検索エンジン	いいボット
3994	検索エンジン	いいボット
3995	検索エンジン	いいボット
3996	検索エンジン	いいボット
3997	検索エンジン	いいボット
3998	検索エンジン	いいボット
3999	検索エンジン	いいボット
4000	スクリーンショットクリエーター	いいボット
4001	検索エンジン	いいボット
4002	検索エンジン	いいボット
4003	検索エンジン	いいボット
4004	検索エンジン	いいボット
4005	スクリーンショットクリエーター	いいボット
4006	クローラー	いいボット
4007	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4008	マーケティング	いいボット
4011	ツール	いいボット
4012	クローラー	いいボット
4013	検索エンジン	いいボット
4014	ツール	いいボット
4015	クローラー	いいボット
4016	クローラー	いいボット
4017	ツール	いいボット
4018	ツール	いいボット
4019	ツール	いいボット
4020	ツール	いいボット
4021	マーケティング	いいボット
4024	ツール	いいボット
4025	検索エンジン	いいボット
4026	検索エンジン	いいボット
4028	マーケティング	いいボット
4029	ツール	いいボット
4030	スクレーパー	いいボット
4031	スクレーパー	いいボット
4035	マーケティング	いいボット
4037	脆弱性スキャナ	いいボット
4042	クローラー	いいボット
4043	スクリーンショットクリエーター	いいボット
4048	フィードフェッチャー	いいボット
4052	ツール	いいボット
4055	未分類	いいボット
4056	マーケティング	いいボット
4057	スクリーンショットクリエーター	いいボット
4058	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4060	検索エンジン	いいボット
4061	検索エンジン	いいボット
4062	検索エンジン	いいボット
4063	検索エンジン	いいボット
4065	スクレーパー	いいボット
4066	マーケティング	いいボット
4067	マーケティング	いいボット
4071	ツール	いいボット
4076	マーケティング	いいボット
4078	クローラー	いいボット
4079	クローラー	いいボット
4081	検索エンジン	いいボット
4082	ツール	いいボット
4085	ツール	いいボット
4086	ツール	いいボット
4090	ツール	いいボット
4091	ツール	いいボット
4092	ツール	いいボット
4093	ツール	いいボット
4094	未分類	いいボット
4095	サイトモニター	いいボット
4096	サイトモニター	いいボット
4097	サイトモニター	いいボット
4098	クローラー	いいボット
4099	検索エンジン	いいボット
4100	検索エンジン	いいボット
4101	検索エンジン	いいボット
4102	検索エンジン	いいボット
4103	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4104	マーケティング	いいボット
4105	マーケティング	いいボット
4106	マーケティング	いいボット
4107	マーケティング	いいボット
4108	マーケティング	いいボット
4109	検索エンジン	いいボット
4110	クローラー	いいボット
4111	クローラー	いいボット
4112	クローラー	いいボット
4113	脆弱性スキャナ	いいボット
4114	クローラー	いいボット
4115	ツール	いいボット
4120	マーケティング	いいボット
4121	マーケティング	いいボット
4122	マーケティング	いいボット
4123	マーケティング	いいボット
4124	マーケティング	いいボット
4125	マーケティング	いいボット
4126	マーケティング	いいボット
4127	マーケティング	いいボット
4128	マーケティング	いいボット
4129	マーケティング	いいボット
4130	マーケティング	いいボット
4131	ツール	いいボット
4132	マーケティング	いいボット
4133	マーケティング	いいボット
4134	ツール	いいボット
4135	マーケティング	いいボット
4136	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4137	マーケティング	いいボット
4138	マーケティング	いいボット
4139	マーケティング	いいボット
4140	マーケティング	いいボット
4141	マーケティング	いいボット
4142	マーケティング	いいボット
4143	マーケティング	いいボット
4144	マーケティング	いいボット
4147	検索エンジン	いいボット
4148	検索エンジン	いいボット
4149	検索エンジン	いいボット
4150	検索エンジン	いいボット
4151	検索エンジン	いいボット
4152	検索エンジン	いいボット
4153	検索エンジン	いいボット
4154	検索エンジン	いいボット
4155	検索エンジン	いいボット
4156	スクリーンショットクリエーター	いいボット
4157	検索エンジン	いいボット
4158	検索エンジン	いいボット
4159	検索エンジン	いいボット
4160	スクリーンショットクリエーター	いいボット
4161	検索エンジン	いいボット
4162	検索エンジン	いいボット
4163	ツール	いいボット
4164	検索エンジン	いいボット
4168	スピードテスター	いいボット
4170	ツール	いいボット
4172	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4173	ツール	いいボット
4174	クローラー	いいボット
4175	クローラー	いいボット
4176	ツール	いいボット
4177	検索エンジン	いいボット
4178	ツール	いいボット
4179	クローラー	いいボット
4180	ツール	いいボット
4181	サイトモニター	いいボット
4182	サイトモニター	いいボット
4183	サイトモニター	いいボット
4184	サイトモニター	いいボット
4185	検索エンジン	いいボット
4186	ツール	いいボット
4187	ツール	いいボット
4190	検索エンジン	いいボット
4191	検索エンジン	いいボット
4192	検索エンジン	いいボット
4193	検索エンジン	いいボット
4194	ツール	いいボット
4196	ツール	いいボット
4197	ツール	いいボット
4198	マーケティング	いいボット
4199	マーケティング	いいボット
4200	脆弱性スキャナ	いいボット
4201	ツール	いいボット
4202	ツール	いいボット
4205	検索エンジン	いいボット
4206	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4207	マーケティング	いいボット
4208	検索エンジン	いいボット
4209	検索エンジン	いいボット
4210	スピードテスター	いいボット
4211	ツール	いいボット
4212	フィードフェッチャー	いいボット
4213	フィードフェッチャー	いいボット
4215	ツール	いいボット
4216	ツール	いいボット
4219	マーケティング	いいボット
4220	ツール	いいボット
4222	サイトモニター	いいボット
4223	マーケティング	いいボット
4224	検索エンジン	いいボット
4225	検索エンジン	いいボット
4226	検索エンジン	いいボット
4227	マーケティング	いいボット
4228	マーケティング	いいボット
4229	ツール	いいボット
4231	スクリーンショットクリエーター	いいボット
4232	ツール	いいボット
4233	サイトモニター	いいボット
4234	サイトモニター	いいボット
4235	サイトモニター	いいボット
4236	サイトモニター	いいボット
4237	サイトモニター	いいボット
4238	サイトモニター	いいボット
4240	マーケティング	いいボット
4241	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4242	マーケティング	いいボット
4243	マーケティング	いいボット
4244	マーケティング	いいボット
4245	マーケティング	いいボット
4246	マーケティング	いいボット
4247	検索エンジン	いいボット
4248	検索エンジン	いいボット
4249	スクリーンショットクリエーター	いいボット
4250	検索エンジン	いいボット
4251	検索エンジン	いいボット
4252	クローラー	いいボット
4253	クローラー	いいボット
4254	クローラー	いいボット
4255	ツール	いいボット
4256	未分類	いいボット
4257	ツール	いいボット
4258	クローラー	いいボット
4259	クローラー	いいボット
4260	ツール	いいボット
4261	ツール	いいボット
4262	ツール	いいボット
4265	検索エンジン	いいボット
4266	未分類	いいボット
4267	ツール	いいボット
4268	ツール	いいボット
4269	検索エンジン	いいボット
4270	検索エンジン	いいボット
4271	検索エンジン	いいボット
4272	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4273	検索エンジン	いいボット
4274	検索エンジン	いいボット
4275	検索エンジン	いいボット
4279	マーケティング	いいボット
4280	クローラー	いいボット
4282	マーケティング	いいボット
4283	マーケティング	いいボット
4284	マーケティング	いいボット
4285	マーケティング	いいボット
4286	マーケティング	いいボット
4287	マーケティング	いいボット
4288	マーケティング	いいボット
4289	マーケティング	いいボット
4290	マーケティング	いいボット
4291	マーケティング	いいボット
4292	マーケティング	いいボット
4293	マーケティング	いいボット
4294	マーケティング	いいボット
4295	検索エンジン	いいボット
4296	検索エンジン	いいボット
4297	検索エンジン	いいボット
4298	検索エンジン	いいボット
4299	検索エンジン	いいボット
4300	検索エンジン	いいボット
4301	検索エンジン	いいボット
4302	検索エンジン	いいボット
4303	検索エンジン	いいボット
4304	検索エンジン	いいボット
4305	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4306	スクリーンショットクリエーター	いいボット
4307	検索エンジン	いいボット
4308	検索エンジン	いいボット
4309	検索エンジン	いいボット
4310	検索エンジン	いいボット
4311	スクリーンショットクリエーター	いいボット
4312	検索エンジン	いいボット
4313	検索エンジン	いいボット
4314	検索エンジン	いいボット
4315	検索エンジン	いいボット
4316	検索エンジン	いいボット
4317	検索エンジン	いいボット
4318	スクリーンショットクリエーター	いいボット
4319	スクリーンショットクリエーター	いいボット
4321	未分類	いいボット
4322	クローラー	いいボット
4323	ツール	いいボット
4324	ツール	いいボット
4325	ツール	いいボット
4328	マーケティング	いいボット
4330	サイトモニター	いいボット
4331	検索エンジン	いいボット
4332	検索エンジン	いいボット
4335	マーケティング	いいボット
4336	マーケティング	いいボット
4337	ツール	いいボット
4338	ツール	いいボット
4339	ツール	いいボット
4340	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4341	クローラー	いいボット
4342	脆弱性スキャナ	いいボット
4343	脆弱性スキャナ	いいボット
4344	スクレーパー	いいボット
4345	マーケティング	いいボット
4346	マーケティング	いいボット
4347	マーケティング	いいボット
4348	マーケティング	いいボット
4349	マーケティング	いいボット
4350	マーケティング	いいボット
4351	マーケティング	いいボット
4352	マーケティング	いいボット
4353	マーケティング	いいボット
4354	マーケティング	いいボット
4355	検索エンジン	いいボット
4356	検索エンジン	いいボット
4357	検索エンジン	いいボット
4358	検索エンジン	いいボット
4359	検索エンジン	いいボット
4360	検索エンジン	いいボット
4361	検索エンジン	いいボット
4362	検索エンジン	いいボット
4363	検索エンジン	いいボット
4364	検索エンジン	いいボット
4365	スクリーンショットクリエーター	いいボット
4366	検索エンジン	いいボット
4367	検索エンジン	いいボット
4368	検索エンジン	いいボット
4369	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4370	スクリーンショットクリエーター	いいボット
4371	検索エンジン	いいボット
4372	検索エンジン	いいボット
4373	検索エンジン	いいボット
4374	検索エンジン	いいボット
4375	検索エンジン	いいボット
4376	スクリーンショットクリエーター	いいボット
4377	クローラー	いいボット
4378	クローラー	いいボット
4379	検索エンジン	いいボット
4380	検索エンジン	いいボット
4381	検索エンジン	いいボット
4382	検索エンジン	いいボット
4383	クローラー	いいボット
4384	検索エンジン	いいボット
4385	ツール	いいボット
4386	未分類	いいボット
4387	クローラー	いいボット
4388	クローラー	いいボット
4389	ツール	いいボット
4390	ツール	いいボット
4391	ツール	いいボット
4392	ツール	いいボット
4393	ツール	いいボット
4394	未分類	いいボット
4395	ツール	いいボット
4396	サイトモニター	いいボット
4397	サイトモニター	いいボット
4404	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4405	検索エンジン	いいボット
4406	検索エンジン	いいボット
4407	未分類	いいボット

2021 年 **10** 月のボット署名の更新

August 15, 2023

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 10 は、ビルドが 13.0 76.31 以降の NetScaler NetScaler ADC プラットフォームに適用されます。

ボットの署名を更新しました

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
71	クローラー	いいボット
74	クローラー	いいボット
75	クローラー	いいボット
372	スクレーパー	いいボット
373	スクレーパー	いいボット
374	スクレーパー	いいボット
375	スクレーパー	いいボット
376	スクレーパー	いいボット
377	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
378	スクレーパー	いいボット
379	スクレーパー	いいボット
380	スクレーパー	いいボット
381	スクレーパー	いいボット
382	スクレーパー	いいボット
383	スクレーパー	いいボット
384	スクレーパー	いいボット
385	スクレーパー	いいボット
386	スクレーパー	いいボット
387	スクレーパー	いいボット
389	スクレーパー	いいボット
390	スクレーパー	いいボット
391	スクレーパー	いいボット
639	検索エンジン	いいボット
702	検索エンジン	いいボット
703	検索エンジン	いいボット
1173	ツール	いいボット
1174	マーケティング	いいボット
1176	検索エンジン	いいボット
1178	スピードテスター	いいボット
1185	スクリーンショットクリエーター	いいボット
1209	未分類	いいボット
1531	フィードフェッチャー	いいボット
2586	未分類	いいボット
2674	ツール	いいボット
2756	ツール	いいボット
2758	未分類	いいボット
2759	ツール	いいボット
2784	ツール	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2952	ツール	いいボット
3163	ツール	いいボット
3554	ツール	いいボット
3782	マーケティング	いいボット
3788	ツール	いいボット
3789	ツール	いいボット
3797	マーケティング	いいボット
3798	マーケティング	いいボット
3799	マーケティング	いいボット
3800	マーケティング	いいボット
3801	マーケティング	いいボット
3802	スクリーンショットクリエーター	いいボット
3803	検索エンジン	いいボット
3804	スクリーンショットクリエーター	いいボット
3805	検索エンジン	いいボット
3861	マーケティング	いいボット
3862	マーケティング	いいボット
3863	マーケティング	いいボット
3864	マーケティング	いいボット
3865	マーケティング	いいボット
3866	マーケティング	いいボット
3867	マーケティング	いいボット
3868	マーケティング	いいボット
3869	ツール	いいボット
3871	マーケティング	いいボット
3872	マーケティング	いいボット
3873	検索エンジン	いいボット
3874	検索エンジン	いいボット
3875	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3876	検索エンジン	いいボット
3877	スクリーンショットクリエーター	いいボット
3878	検索エンジン	いいボット
3879	検索エンジン	いいボット
3880	スクリーンショットクリエーター	いいボット
3881	スクリーンショットクリエーター	いいボット
3882	検索エンジン	いいボット
3883	検索エンジン	いいボット
3884	検索エンジン	いいボット
3885	検索エンジン	いいボット
3963	マーケティング	いいボット
4040	クローラー	いいボット
4041	ツール	いいボット
4120	マーケティング	いいボット
4122	マーケティング	いいボット
4123	マーケティング	いいボット
4124	マーケティング	いいボット
4125	マーケティング	いいボット
4133	マーケティング	いいボット
4134	ツール	いいボット
4135	マーケティング	いいボット
4136	マーケティング	いいボット
4137	マーケティング	いいボット
4138	マーケティング	いいボット
4139	マーケティング	いいボット
4140	マーケティング	いいボット
4141	マーケティング	いいボット
4142	マーケティング	いいボット
4143	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4144	マーケティング	いいボット
4145	検索エンジン	いいボット
4146	検索エンジン	いいボット
4147	検索エンジン	いいボット
4148	検索エンジン	いいボット
4149	検索エンジン	いいボット
4150	検索エンジン	いいボット
4151	検索エンジン	いいボット
4152	検索エンジン	いいボット
4153	検索エンジン	いいボット
4154	検索エンジン	いいボット
4155	検索エンジン	いいボット
4156	スクリーンショットクリエーター	いいボット
4157	検索エンジン	いいボット
4158	検索エンジン	いいボット
4159	検索エンジン	いいボット
4160	スクリーンショットクリエーター	いいボット
4161	検索エンジン	いいボット
4162	検索エンジン	いいボット
4163	ツール	いいボット
4164	検索エンジン	いいボット
4209	検索エンジン	いいボット
4240	マーケティング	いいボット
4241	マーケティング	いいボット
4248	検索エンジン	いいボット
4249	スクリーンショットクリエーター	いいボット
4250	検索エンジン	いいボット
4251	検索エンジン	いいボット
4282	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4283	マーケティング	いいボット
4284	マーケティング	いいボット
4285	マーケティング	いいボット
4286	マーケティング	いいボット
4287	マーケティング	いいボット
4288	マーケティング	いいボット
4289	マーケティング	いいボット
4290	マーケティング	いいボット
4291	マーケティング	いいボット
4292	マーケティング	いいボット
4293	マーケティング	いいボット
4294	マーケティング	いいボット
4295	検索エンジン	いいボット
4296	検索エンジン	いいボット
4297	検索エンジン	いいボット
4298	検索エンジン	いいボット
4299	検索エンジン	いいボット
4300	検索エンジン	いいボット
4301	検索エンジン	いいボット
4302	検索エンジン	いいボット
4303	検索エンジン	いいボット
4304	検索エンジン	いいボット
4305	検索エンジン	いいボット
4306	スクリーンショットクリエーター	いいボット
4307	検索エンジン	いいボット
4308	検索エンジン	いいボット
4309	検索エンジン	いいボット
4310	検索エンジン	いいボット
4311	スクリーンショットクリエーター	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4312	検索エンジン	いいボット
4313	検索エンジン	いいボット
4314	検索エンジン	いいボット
4315	検索エンジン	いいボット
4316	検索エンジン	いいボット
4317	検索エンジン	いいボット
4318	スクリーンショットクリエーター	いいボット
4319	スクリーンショットクリエーター	いいボット
4337	ツール	いいボット
4338	ツール	いいボット
4345	マーケティング	いいボット
4346	マーケティング	いいボット
4347	マーケティング	いいボット
4348	マーケティング	いいボット
4349	マーケティング	いいボット
4350	マーケティング	いいボット
4351	マーケティング	いいボット
4352	マーケティング	いいボット
4353	マーケティング	いいボット
4354	マーケティング	いいボット
4355	検索エンジン	いいボット
4356	検索エンジン	いいボット
4357	検索エンジン	いいボット
4358	検索エンジン	いいボット
4359	検索エンジン	いいボット
4360	検索エンジン	いいボット
4361	検索エンジン	いいボット
4362	検索エンジン	いいボット
4363	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4364	検索エンジン	いいボット
4365	スクリーンショットクリエーター	いいボット
4366	検索エンジン	いいボット
4367	検索エンジン	いいボット
4368	検索エンジン	いいボット
4369	検索エンジン	いいボット
4370	スクリーンショットクリエーター	いいボット
4371	検索エンジン	いいボット
4372	検索エンジン	いいボット
4373	検索エンジン	いいボット
4374	検索エンジン	いいボット
4375	検索エンジン	いいボット
4376	スクリーンショットクリエーター	いいボット

2021 年 **11** 月のボット署名の更新

August 15, 2023

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 11 は、ビルドが 13.0 76.31 以降の NetScaler NetScaler ADC プラットフォームに適用されます。

新しいボット署名

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4408	スクレーパー	いいボット
4409	クローラー	悪いボット
4411	マーケティング	いいボット
4412	マーケティング	いいボット
4413	マーケティング	いいボット
4421	スクリーンショットクリエーター	いいボット
4422	クローラー	いいボット
4423	ツール	悪いボット
4424	サイトモニター	いいボット
4425	マーケティング	いいボット
4426	クローラー	悪いボット
4427	スクレーパー	いいボット
4428	スクレーパー	いいボット
4429	スクリーンショットクリエーター	いいボット
4430	ウイルススキャナー	いいボット
4431	サイトモニター	いいボット
4432	ツール	いいボット
4433	検索エンジン	いいボット
4434	検索エンジン	いいボット
4435	検索エンジン	いいボット
4436	マーケティング	いいボット
4437	マーケティング	いいボット
4438	スクレーパー	いいボット
4439	スクレーパー	いいボット
4440	スクレーパー	いいボット
4441	フィードフェッチャー	いいボット
4442	マーケティング	いいボット
4443	スクレーパー	いいボット
4445	未分類	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4446	スクレーパー	いいボット
4450	スクリーンショットクリエーター	いいボット
4451	スピードテスター	いいボット
4452	検索エンジン	いいボット
4466	未分類	いいボット
4467	スクリーンショットクリエーター	いいボット
4468	ツール	いいボット
4469	未分類	いいボット
4470	ツール	いいボット
4472	スクレーパー	いいボット
4473	未分類	いいボット
4474	マーケティング	いいボット
4476	クローラー	いいボット
4477	クローラー	いいボット
4478	クローラー	いいボット
4479	クローラー	いいボット
4480	クローラー	いいボット
4481	クローラー	いいボット
4482	クローラー	いいボット
4483	クローラー	いいボット
4484	クローラー	いいボット
4485	クローラー	いいボット
4486	スクレーパー	いいボット
4487	スクレーパー	いいボット
4488	スクレーパー	いいボット
4489	検索エンジン	いいボット
4491	ツール	いいボット
4492	未分類	悪いボット
4493	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4494	ツール	いいボット
4496	ツール	いいボット
4497	クローラー	いいボット
4498	未分類	悪いボット
4499	未分類	悪いボット
4501	マーケティング	いいボット
4502	マーケティング	いいボット
4503	マーケティング	いいボット
4508	未分類	いいボット
4509	未分類	いいボット
4510	未分類	いいボット
4511	未分類	いいボット
4512	ツール	いいボット
4513	ツール	いいボット
4514	ツール	いいボット
4515	ツール	いいボット
4516	未分類	いいボット
4518	スクレーパー	悪いボット
4519	スクリーンショットクリエーター	いいボット
4520	マーケティング	いいボット
4521	未分類	いいボット
4522	ツール	いいボット
4523	未分類	悪いボット
4524	未分類	悪いボット
4525	クローラー	いいボット
4526	クローラー	いいボット
4527	クローラー	いいボット
4528	クローラー	いいボット
4529	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4530	未分類	悪いボット
4531	マーケティング	いいボット
4532	マーケティング	いいボット
4533	マーケティング	いいボット
4534	マーケティング	いいボット
4535	マーケティング	いいボット
4541	マーケティング	いいボット
4552	未分類	いいボット
4553	ツール	悪いボット
4554	ツール	悪いボット
4555	ツール	いいボット
4556	ツール	いいボット
4558	スクレーパー	いいボット
4559	クローラー	いいボット
4560	クローラー	いいボット
4561	サイトモニター	いいボット
4562	検索エンジン	いいボット
4563	検索エンジン	いいボット
1000000	ブラウザー	いいボット
1000001	スクレーパー	悪いボット
1000002	アプリケーション	悪いボット
1000003	ブラウザー	いいボット
1000004	スクレーパー	いいボット
1000005	スクレーパー	いいボット
1000006	クローラー	悪いボット
1000007	ブラウザー	悪いボット
1000008	未分類	悪いボット
1000009	ブラウザー	いいボット
1000010	スクレーパー	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000011	ブラウザー	悪いボット
1000012	ブラウザー	いいボット
1000013	ブラウザー	悪いボット
1000014	スクレーパー	いいボット
1000015	スクレーパー	悪いボット
1000016	スクレーパー	悪いボット
1000017	ブラウザー	いいボット
1000018	ブラウザー	悪いボット
1000019	未分類	悪いボット
1000020	スクレーパー	いいボット
1000021	ブラウザー	悪いボット
1000022	スクレーパー	いいボット
1000023	スクレーパー	いいボット
1000024	クローラー	いいボット
1000025	ブラウザー	悪いボット
1000026	Analyzer	いいボット
1000027	Analyzer	いいボット
1000028	Analyzer	いいボット
1000029	Analyzer	いいボット
1000030	Analyzer	いいボット
1000031	ブラウザー	いいボット
1000032	Analyzer	いいボット
1000033	Analyzer	いいボット
1000034	ブラウザー	悪いボット
1000035	スクレーパー	いいボット
1000036	スクレーパー	いいボット
1000037	Analyzer	いいボット
1000038	Analyzer	いいボット
1000039	Analyzer	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000040	Analyzer	いいボット
1000041	スクレーパー	いいボット
1000042	Analyzer	いいボット
1000043	Analyzer	いいボット
1000044	クローラー	いいボット
1000045	ブラウザー	悪いボット
1000046	ブラウザー	悪いボット
1000047	スクレーパー	いいボット
1000048	ブラウザー	悪いボット
1000049	Analyzer	いいボット
1000050	ブラウザー	悪いボット
1000051	ブラウザー	いいボット
1000052	ブラウザー	悪いボット
1000053	スクレーパー	いいボット
1000054	ブラウザー	いいボット
1000055	ブラウザー	いいボット
1000056	スクレーパー	悪いボット
1000057	クローラー	悪いボット
1000058	スクレーパー	悪いボット
1000059	Analyzer	いいボット
1000060	ブラウザー	悪いボット
1000061	ブラウザー	悪いボット
1000062	ブラウザー	悪いボット
1000063	スクレーパー	悪いボット
1000064	スクレーパー	悪いボット
1000065	スクレーパー	悪いボット
1000066	アプリケーション	悪いボット
1000067	スクレーパー	悪いボット
1000068	ブラウザー	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000069	スクレーパー	悪いボット
1000070	スクレーパー	いいボット
1000071	ブラウザー	いいボット
1000072	ブラウザー	いいボット
1000073	ブラウザー	悪いボット
1000074	ブラウザー	悪いボット
1000075	アプリケーション	悪いボット
1000076	スクレーパー	悪いボット

ボットの署名を更新しました

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2	クローラー	いいボット
5	クローラー	いいボット
9	クローラー	いいボット
30	クローラー	悪いボット
45	クローラー	いいボット
46	クローラー	いいボット
48	クローラー	いいボット
52	クローラー	いいボット
60	クローラー	いいボット
61	クローラー	いいボット
63	クローラー	いいボット
67	クローラー	いいボット
76	クローラー	いいボット
78	クローラー	いいボット
79	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
80	クローラー	いいボット
81	クローラー	いいボット
82	クローラー	いいボット
83	クローラー	いいボット
84	クローラー	いいボット
87	クローラー	いいボット
90	クローラー	いいボット
95	クローラー	いいボット
96	クローラー	いいボット
97	クローラー	いいボット
100	クローラー	いいボット
101	クローラー	いいボット
102	クローラー	いいボット
103	クローラー	いいボット
104	クローラー	いいボット
107	クローラー	いいボット
108	クローラー	いいボット
110	クローラー	いいボット
111	クローラー	いいボット
114	クローラー	いいボット
115	クローラー	いいボット
123	クローラー	いいボット
135	クローラー	いいボット
136	クローラー	いいボット
137	クローラー	いいボット
140	クローラー	いいボット
141	クローラー	いいボット
143	クローラー	いいボット
144	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
145	クローラー	いいボット
146	クローラー	いいボット
147	クローラー	いいボット
149	クローラー	いいボット
152	クローラー	いいボット
155	クローラー	いいボット
156	クローラー	いいボット
157	クローラー	いいボット
158	クローラー	いいボット
159	クローラー	いいボット
160	クローラー	いいボット
161	クローラー	いいボット
162	クローラー	いいボット
163	クローラー	いいボット
164	クローラー	いいボット
165	クローラー	いいボット
166	クローラー	いいボット
167	クローラー	いいボット
172	クローラー	いいボット
173	クローラー	いいボット
174	クローラー	いいボット
176	クローラー	いいボット
177	クローラー	いいボット
180	クローラー	いいボット
182	クローラー	いいボット
187	クローラー	いいボット
197	クローラー	いいボット
201	クローラー	いいボット
202	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
203	クローラー	いいボット
206	クローラー	いいボット
217	フィードフェッチャー	いいボット
219	フィードフェッチャー	いいボット
229	スクレーパー	いいボット
235	スクレーパー	いいボット
236	スクレーパー	いいボット
237	スクレーパー	いいボット
248	スクレーパー	いいボット
250	スクレーパー	いいボット
252	スクレーパー	いいボット
260	スクレーパー	いいボット
263	スクレーパー	いいボット
265	スクレーパー	いいボット
267	スクレーパー	いいボット
268	スクレーパー	いいボット
271	スクレーパー	いいボット
272	スクレーパー	いいボット
276	スクレーパー	いいボット
277	スクレーパー	いいボット
278	スクレーパー	いいボット
279	スクレーパー	いいボット
280	スクレーパー	いいボット
281	スクレーパー	いいボット
283	スクレーパー	いいボット
285	スクレーパー	いいボット
286	スクレーパー	いいボット
287	スクレーパー	いいボット
290	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
292	スクレーパー	いいボット
293	スクレーパー	いいボット
338	スクレーパー	いいボット
342	スクレーパー	いいボット
343	スクレーパー	いいボット
344	スクレーパー	いいボット
351	スクレーパー	いいボット
352	スクレーパー	いいボット
353	スクレーパー	いいボット
355	スクレーパー	いいボット
357	スクレーパー	いいボット
360	スクレーパー	いいボット
362	スクレーパー	いいボット
366	スクレーパー	いいボット
370	スクレーパー	いいボット
371	スクレーパー	いいボット
392	スクレーパー	いいボット
393	スクレーパー	いいボット
394	スクレーパー	いいボット
396	スクレーパー	いいボット
397	スクレーパー	いいボット
414	スクレーパー	いいボット
418	スクレーパー	いいボット
419	スクレーパー	いいボット
421	スクレーパー	いいボット
422	スクレーパー	いいボット
423	スクレーパー	いいボット
424	スクレーパー	いいボット
425	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
426	スクレーパー	いいボット
427	スクレーパー	いいボット
428	スクレーパー	いいボット
430	スクレーパー	いいボット
432	スクレーパー	いいボット
433	スクレーパー	いいボット
434	スクレーパー	いいボット
435	スクレーパー	いいボット
441	スクレーパー	いいボット
445	スクレーパー	いいボット
446	スクレーパー	いいボット
451	スクレーパー	いいボット
452	スクレーパー	いいボット
454	スクレーパー	いいボット
455	スクレーパー	いいボット
456	スクレーパー	いいボット
457	スクレーパー	いいボット
458	スクレーパー	いいボット
461	スクレーパー	いいボット
465	スクレーパー	いいボット
466	スクレーパー	いいボット
469	スクレーパー	いいボット
473	スクレーパー	いいボット
474	スクレーパー	いいボット
476	スクレーパー	いいボット
477	スクレーパー	いいボット
484	スクレーパー	いいボット
485	スクレーパー	いいボット
487	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
488	スクレーパー	いいボット
489	スクレーパー	いいボット
490	スクレーパー	いいボット
493	スクレーパー	いいボット
494	スクレーパー	いいボット
495	スクレーパー	いいボット
497	スクレーパー	いいボット
498	スクレーパー	いいボット
499	スクレーパー	いいボット
500	スクレーパー	いいボット
505	スクレーパー	いいボット
506	スクレーパー	いいボット
507	スクレーパー	いいボット
512	スクレーパー	いいボット
513	スクレーパー	いいボット
514	スクレーパー	いいボット
527	スクレーパー	いいボット
533	スクレーパー	いいボット
539	スクレーパー	いいボット
540	スクレーパー	いいボット
542	スクレーパー	いいボット
544	スクレーパー	いいボット
545	スクレーパー	いいボット
546	スクレーパー	いいボット
547	スクレーパー	いいボット
548	スクレーパー	いいボット
551	スクレーパー	いいボット
552	スクレーパー	いいボット
554	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
556	スクレーパー	いいボット
558	スクレーパー	いいボット
560	スクレーパー	いいボット
561	スクレーパー	いいボット
566	スクレーパー	いいボット
575	スクレーパー	いいボット
578	スクレーパー	いいボット
581	スクレーパー	いいボット
582	スクレーパー	いいボット
591	スクレーパー	いいボット
593	スクレーパー	いいボット
595	スクレーパー	いいボット
600	スクレーパー	いいボット
601	スクレーパー	いいボット
602	スクレーパー	いいボット
604	スクレーパー	いいボット
605	スクレーパー	いいボット
609	スクレーパー	いいボット
610	スクレーパー	いいボット
611	スクレーパー	いいボット
612	スクレーパー	いいボット
613	スクレーパー	いいボット
615	スクレーパー	いいボット
620	検索エンジン	いいボット
622	検索エンジン	いいボット
623	検索エンジン	いいボット
624	検索エンジン	いいボット
626	検索エンジン	いいボット
627	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
628	検索エンジン	いいボット
629	検索エンジン	いいボット
633	検索エンジン	いいボット
634	検索エンジン	いいボット
636	検索エンジン	いいボット
637	検索エンジン	いいボット
640	検索エンジン	いいボット
641	検索エンジン	いいボット
642	検索エンジン	いいボット
643	検索エンジン	いいボット
647	検索エンジン	いいボット
649	検索エンジン	いいボット
650	検索エンジン	いいボット
651	検索エンジン	いいボット
654	検索エンジン	いいボット
656	検索エンジン	いいボット
657	検索エンジン	いいボット
658	検索エンジン	いいボット
659	検索エンジン	いいボット
660	検索エンジン	いいボット
663	検索エンジン	いいボット
664	検索エンジン	いいボット
665	検索エンジン	いいボット
666	検索エンジン	いいボット
667	検索エンジン	いいボット
669	検索エンジン	いいボット
670	検索エンジン	いいボット
671	検索エンジン	いいボット
672	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
673	検索エンジン	いいボット
674	検索エンジン	いいボット
675	検索エンジン	いいボット
676	検索エンジン	いいボット
677	検索エンジン	いいボット
679	検索エンジン	いいボット
680	検索エンジン	いいボット
690	検索エンジン	いいボット
693	検索エンジン	いいボット
694	検索エンジン	いいボット
697	検索エンジン	いいボット
698	検索エンジン	いいボット
702	検索エンジン	いいボット
706	検索エンジン	いいボット
712	検索エンジン	いいボット
713	検索エンジン	いいボット
714	検索エンジン	いいボット
715	検索エンジン	いいボット
716	検索エンジン	いいボット
721	検索エンジン	いいボット
723	検索エンジン	いいボット
725	検索エンジン	いいボット
727	検索エンジン	いいボット
728	検索エンジン	いいボット
729	検索エンジン	いいボット
730	検索エンジン	いいボット
731	検索エンジン	いいボット
732	検索エンジン	いいボット
735	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
736	検索エンジン	いいボット
740	検索エンジン	いいボット
748	検索エンジン	いいボット
749	検索エンジン	いいボット
750	検索エンジン	いいボット
751	検索エンジン	いいボット
756	検索エンジン	いいボット
757	検索エンジン	いいボット
758	検索エンジン	いいボット
759	検索エンジン	いいボット
760	検索エンジン	いいボット
761	検索エンジン	いいボット
762	検索エンジン	いいボット
763	検索エンジン	いいボット
764	検索エンジン	いいボット
765	検索エンジン	いいボット
766	検索エンジン	いいボット
767	検索エンジン	いいボット
768	検索エンジン	いいボット
769	検索エンジン	いいボット
770	検索エンジン	いいボット
771	検索エンジン	いいボット
772	検索エンジン	いいボット
773	検索エンジン	いいボット
776	検索エンジン	いいボット
777	検索エンジン	いいボット
780	検索エンジン	いいボット
781	検索エンジン	いいボット
784	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
786	検索エンジン	いいボット
787	検索エンジン	いいボット
788	検索エンジン	いいボット
789	検索エンジン	いいボット
790	検索エンジン	いいボット
791	検索エンジン	いいボット
792	検索エンジン	いいボット
795	検索エンジン	いいボット
796	検索エンジン	いいボット
798	検索エンジン	いいボット
800	検索エンジン	いいボット
801	検索エンジン	いいボット
802	検索エンジン	いいボット
803	検索エンジン	いいボット
805	検索エンジン	いいボット
806	検索エンジン	いいボット
807	検索エンジン	いいボット
809	検索エンジン	いいボット
810	検索エンジン	いいボット
811	検索エンジン	いいボット
812	検索エンジン	いいボット
814	検索エンジン	いいボット
815	検索エンジン	いいボット
816	検索エンジン	いいボット
817	検索エンジン	いいボット
818	検索エンジン	いいボット
819	検索エンジン	いいボット
820	検索エンジン	いいボット
821	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
822	検索エンジン	いいボット
823	検索エンジン	いいボット
825	検索エンジン	いいボット
827	検索エンジン	いいボット
830	検索エンジン	いいボット
831	検索エンジン	いいボット
834	検索エンジン	いいボット
837	検索エンジン	いいボット
838	検索エンジン	いいボット
849	サイトモニター	いいボット
850	サイトモニター	いいボット
851	サイトモニター	いいボット
853	サイトモニター	いいボット
857	サイトモニター	いいボット
858	サイトモニター	いいボット
859	サイトモニター	いいボット
860	サイトモニター	いいボット
861	サイトモニター	いいボット
862	サイトモニター	いいボット
863	サイトモニター	いいボット
864	サイトモニター	いいボット
865	サイトモニター	いいボット
866	サイトモニター	いいボット
867	サイトモニター	いいボット
868	サイトモニター	いいボット
869	サイトモニター	いいボット
870	サイトモニター	いいボット
871	サイトモニター	いいボット
872	サイトモニター	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
873	サイトモニター	いいボット
874	サイトモニター	いいボット
875	サイトモニター	いいボット
876	サイトモニター	いいボット
877	サイトモニター	いいボット
880	サイトモニター	いいボット
881	サイトモニター	いいボット
883	サイトモニター	いいボット
885	サイトモニター	いいボット
886	サイトモニター	いいボット
888	サイトモニター	いいボット
889	サイトモニター	いいボット
895	サイトモニター	いいボット
896	サイトモニター	いいボット
897	サイトモニター	いいボット
898	サイトモニター	いいボット
900	サイトモニター	いいボット
901	サイトモニター	いいボット
904	サイトモニター	いいボット
906	サイトモニター	いいボット
908	サイトモニター	いいボット
909	サイトモニター	いいボット
910	サイトモニター	いいボット
911	サイトモニター	いいボット
912	サイトモニター	いいボット
913	サイトモニター	いいボット
917	サイトモニター	いいボット
918	サイトモニター	いいボット
919	サイトモニター	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
920	サイトモニター	いいボット
921	サイトモニター	いいボット
924	サイトモニター	いいボット
926	サイトモニター	いいボット
927	サイトモニター	いいボット
928	サイトモニター	いいボット
929	サイトモニター	いいボット
930	サイトモニター	いいボット
931	サイトモニター	いいボット
934	サイトモニター	いいボット
938	サイトモニター	いいボット
939	サイトモニター	いいボット
958	サイトモニター	いいボット
959	サイトモニター	いいボット
960	サイトモニター	いいボット
963	サイトモニター	いいボット
984	スクレーパー	いいボット
991	スクレーパー	悪いボット
996	スクレーパー	いいボット
997	スクレーパー	いいボット
998	スクレーパー	いいボット
1002	スクレーパー	いいボット
1006	スクレーパー	いいボット
1622	スクリーンショットクリエーター	いいボット
2810	クローラー	いいボット
3432	未分類	悪いボット
3783	検索エンジン	いいボット
3784	スクレーパー	悪いボット
3788	ツール	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3790	クローラー	いいボット
3791	スピードテスター	いいボット
3792	ツール	いいボット
3793	ツール	いいボット
3794	クローラー	いいボット
3796	スクレーパー	いいボット
3797	マーケティング	いいボット
3799	マーケティング	いいボット
3800	マーケティング	いいボット
3806	ツール	いいボット
3807	クローラー	いいボット
3808	クローラー	いいボット
3809	ツール	いいボット
3810	スクレーパー	いいボット
3811	ツール	いいボット
3812	クローラー	いいボット
3813	ツール	いいボット
3814	クローラー	いいボット
3815	未分類	いいボット
3817	ツール	いいボット
3818	ツール	いいボット
3819	ツール	いいボット
3820	クローラー	いいボット
3821	検索エンジン	いいボット
3822	マーケティング	いいボット
3823	未分類	いいボット
3831	スクレーパー	いいボット
3833	検索エンジン	いいボット
3834	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3835	検索エンジン	いいボット
3836	未分類	いいボット
3838	未分類	いいボット
3839	マーケティング	いいボット
3840	クローラー	いいボット
3842	クローラー	いいボット
3843	クローラー	いいボット
3844	マーケティング	いいボット
3845	マーケティング	いいボット
3846	マーケティング	いいボット
3847	マーケティング	いいボット
3848	未分類	いいボット
3849	クローラー	いいボット
3850	ツール	いいボット
3851	未分類	いいボット
3852	ツール	いいボット
3853	脆弱性スキャナ	いいボット
3854	クローラー	いいボット
3855	クローラー	いいボット
3856	ツール	いいボット
3871	マーケティング	いいボット
3886	ツール	いいボット
3887	クローラー	いいボット
3888	クローラー	いいボット
3889	未分類	いいボット
3890	マーケティング	いいボット
3893	クローラー	いいボット
3894	ツール	いいボット
3895	ツール	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3896	検索エンジン	いいボット
3897	ツール	いいボット
3898	ツール	いいボット
3899	未分類	いいボット
3901	クローラー	いいボット
3902	ツール	いいボット
3903	ツール	いいボット
3904	検索エンジン	いいボット
3905	検索エンジン	いいボット
3906	検索エンジン	いいボット
3907	検索エンジン	いいボット
3912	クローラー	いいボット
3917	未分類	いいボット
3918	クローラー	いいボット
3919	未分類	いいボット
3920	未分類	いいボット
3921	未分類	いいボット
3922	未分類	いいボット
3923	未分類	いいボット
3924	未分類	いいボット
3925	未分類	いいボット
3926	マーケティング	いいボット
3927	マーケティング	いいボット
3928	マーケティング	いいボット
3929	ツール	いいボット
3930	マーケティング	いいボット
3931	未分類	いいボット
3932	クローラー	いいボット
3933	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3934	マーケティング	いいボット
3935	スクレーパー	いいボット
3936	マーケティング	いいボット
3937	スクレーパー	いいボット
3938	フィードフェッチャー	いいボット
3940	検索エンジン	いいボット
3941	クローラー	いいボット
3942	スクレーパー	いいボット
3946	フィードフェッチャー	いいボット
3947	クローラー	いいボット
3950	ウイルススキャナー	いいボット
3951	マーケティング	いいボット
3952	マーケティング	いいボット
3953	マーケティング	いいボット
3954	マーケティング	いいボット
3955	マーケティング	いいボット
3956	マーケティング	いいボット
3957	マーケティング	いいボット
3958	マーケティング	いいボット
3959	マーケティング	いいボット
3960	マーケティング	いいボット
3961	マーケティング	いいボット
3962	マーケティング	いいボット
3963	マーケティング	いいボット
3964	マーケティング	いいボット
3965	マーケティング	いいボット
3966	マーケティング	いいボット
3967	マーケティング	いいボット
3968	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3969	マーケティング	いいボット
3970	検索エンジン	いいボット
3971	スクリーンショットクリエーター	いいボット
3972	スクリーンショットクリエーター	いいボット
3973	検索エンジン	いいボット
3974	検索エンジン	いいボット
3975	検索エンジン	いいボット
3976	検索エンジン	いいボット
3977	検索エンジン	いいボット
3978	スクリーンショットクリエーター	いいボット
3979	検索エンジン	いいボット
3980	スクリーンショットクリエーター	いいボット
3981	検索エンジン	いいボット
3982	検索エンジン	いいボット
3983	検索エンジン	いいボット
3984	検索エンジン	いいボット
3985	検索エンジン	いいボット
3986	検索エンジン	いいボット
3987	スクリーンショットクリエーター	いいボット
3988	検索エンジン	いいボット
3989	検索エンジン	いいボット
3990	検索エンジン	いいボット
3991	検索エンジン	いいボット
3992	検索エンジン	いいボット
3993	検索エンジン	いいボット
3994	検索エンジン	いいボット
3995	検索エンジン	いいボット
3996	検索エンジン	いいボット
3997	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3998	検索エンジン	いいボット
3999	検索エンジン	いいボット
4000	スクリーンショットクリエーター	いいボット
4001	検索エンジン	いいボット
4002	検索エンジン	いいボット
4003	検索エンジン	いいボット
4004	検索エンジン	いいボット
4005	スクリーンショットクリエーター	いいボット
4006	クローラー	いいボット
4007	マーケティング	いいボット
4008	マーケティング	いいボット
4011	ツール	いいボット
4012	クローラー	いいボット
4013	検索エンジン	いいボット
4014	ツール	いいボット
4015	クローラー	いいボット
4016	クローラー	いいボット
4017	ツール	いいボット
4018	ツール	いいボット
4019	ツール	いいボット
4020	ツール	いいボット
4021	マーケティング	いいボット
4024	ツール	いいボット
4025	検索エンジン	いいボット
4026	検索エンジン	いいボット
4027	検索エンジン	いいボット
4028	マーケティング	いいボット
4029	ツール	いいボット
4030	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4031	スクレーパー	いいボット
4033	クローラー	いいボット
4034	クローラー	いいボット
4035	マーケティング	いいボット
4036	脆弱性スキャナ	いいボット
4037	脆弱性スキャナ	いいボット
4038	未分類	悪いボット
4039	ツール	いいボット
4042	クローラー	いいボット
4043	スクリーンショットクリエーター	いいボット
4048	フィードフェッチャー	いいボット
4050	クローラー	いいボット
4051	クローラー	いいボット
4052	ツール	いいボット
4053	ツール	いいボット
4055	未分類	いいボット
4056	マーケティング	いいボット
4057	スクリーンショットクリエーター	いいボット
4058	クローラー	いいボット
4060	検索エンジン	いいボット
4061	検索エンジン	いいボット
4062	検索エンジン	いいボット
4063	検索エンジン	いいボット
4064	ツール	いいボット
4065	スクレーパー	いいボット
4066	マーケティング	いいボット
4067	マーケティング	いいボット
4071	ツール	いいボット
4076	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4077	スクレーパー	いいボット
4078	クローラー	いいボット
4079	クローラー	いいボット
4081	検索エンジン	いいボット
4082	ツール	いいボット
4085	ツール	いいボット
4086	ツール	いいボット
4087	ツール	悪いボット
4088	検索エンジン	いいボット
4089	マーケティング	いいボット
4090	ツール	いいボット
4091	ツール	いいボット
4092	ツール	いいボット
4093	ツール	いいボット
4094	未分類	いいボット
4095	サイトモニター	いいボット
4096	サイトモニター	いいボット
4097	サイトモニター	いいボット
4098	クローラー	いいボット
4099	検索エンジン	いいボット
4100	検索エンジン	いいボット
4101	検索エンジン	いいボット
4102	検索エンジン	いいボット
4103	マーケティング	いいボット
4104	マーケティング	いいボット
4105	マーケティング	いいボット
4106	マーケティング	いいボット
4109	検索エンジン	いいボット
4110	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4111	クローラー	いいボット
4112	クローラー	いいボット
4113	脆弱性スキャナ	いいボット
4114	クローラー	いいボット
4115	ツール	いいボット
4121	マーケティング	いいボット
4126	マーケティング	いいボット
4127	マーケティング	いいボット
4128	マーケティング	いいボット
4129	マーケティング	いいボット
4130	マーケティング	いいボット
4131	ツール	いいボット
4132	マーケティング	いいボット
4165	マーケティング	いいボット
4168	スピードテスター	いいボット
4170	ツール	いいボット
4172	クローラー	いいボット
4173	ツール	いいボット
4174	クローラー	いいボット
4175	クローラー	いいボット
4176	ツール	いいボット
4177	検索エンジン	いいボット
4178	ツール	いいボット
4179	クローラー	いいボット
4180	ツール	いいボット
4181	サイトモニター	いいボット
4182	サイトモニター	いいボット
4183	サイトモニター	いいボット
4184	サイトモニター	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4185	検索エンジン	いいボット
4186	ツール	いいボット
4187	ツール	いいボット
4188	スクリーンショットクリエーター	いいボット
4189	マーケティング	いいボット
4190	検索エンジン	いいボット
4191	検索エンジン	いいボット
4192	検索エンジン	いいボット
4193	検索エンジン	いいボット
4194	ツール	いいボット
4196	ツール	いいボット
4197	ツール	いいボット
4198	マーケティング	いいボット
4199	マーケティング	いいボット
4200	脆弱性スキャナ	いいボット
4201	ツール	いいボット
4202	ツール	いいボット
4205	検索エンジン	いいボット
4209	検索エンジン	いいボット
4210	スピードテスター	いいボット
4211	ツール	いいボット
4212	フィードフェッチャー	いいボット
4213	フィードフェッチャー	いいボット
4215	ツール	いいボット
4216	ツール	いいボット
4219	マーケティング	いいボット
4220	ツール	いいボット
4222	サイトモニター	いいボット
4223	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4224	検索エンジン	いいボット
4225	検索エンジン	いいボット
4226	検索エンジン	いいボット
4227	マーケティング	いいボット
4228	マーケティング	いいボット
4229	ツール	いいボット
4231	スクリーンショットクリエーター	いいボット
4232	ツール	いいボット
4233	サイトモニター	いいボット
4236	サイトモニター	いいボット
4242	マーケティング	いいボット
4243	マーケティング	いいボット
4244	マーケティング	いいボット
4245	マーケティング	いいボット
4246	マーケティング	いいボット
4247	検索エンジン	いいボット
4252	クローラー	いいボット
4253	クローラー	いいボット
4254	クローラー	いいボット
4255	ツール	いいボット
4256	未分類	いいボット
4257	ツール	いいボット
4258	クローラー	いいボット
4259	クローラー	いいボット
4260	ツール	いいボット
4261	ツール	いいボット
4262	ツール	いいボット
4263	マーケティング	いいボット
4265	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4266	未分類	いいボット
4267	ツール	いいボット
4268	ツール	いいボット
4269	検索エンジン	いいボット
4270	検索エンジン	いいボット
4271	検索エンジン	いいボット
4272	検索エンジン	いいボット
4273	検索エンジン	いいボット
4274	検索エンジン	いいボット
4275	検索エンジン	いいボット
4279	マーケティング	いいボット
4280	クローラー	いいボット
4321	未分類	いいボット
4322	クローラー	いいボット
4323	ツール	いいボット
4324	ツール	いいボット
4325	ツール	いいボット
4327	検索エンジン	いいボット
4328	マーケティング	いいボット
4330	サイトモニター	いいボット
4331	検索エンジン	いいボット
4334	スクレーパー	いいボット
4335	マーケティング	いいボット
4336	マーケティング	いいボット
4339	ツール	いいボット
4340	クローラー	いいボット
4341	クローラー	いいボット
4342	脆弱性スキャナ	いいボット
4343	脆弱性スキャナ	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4344	スクレーパー	いいボット
4377	クローラー	いいボット
4378	クローラー	いいボット
4379	検索エンジン	いいボット
4380	検索エンジン	いいボット
4381	検索エンジン	いいボット
4382	検索エンジン	いいボット
4383	クローラー	いいボット
4384	検索エンジン	いいボット
4385	ツール	いいボット
4386	未分類	いいボット
4387	クローラー	いいボット
4388	クローラー	いいボット
4389	ツール	いいボット
4390	ツール	いいボット
4391	ツール	いいボット
4392	ツール	いいボット
4393	ツール	いいボット
4394	未分類	いいボット
4395	ツール	いいボット
4396	サイトモニター	いいボット
4397	サイトモニター	いいボット
4404	検索エンジン	いいボット
4405	検索エンジン	いいボット
4406	検索エンジン	いいボット
4407	未分類	いいボット

2022 年 3 月のボット署名の更新

August 15, 2023

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 12 は、13.0 76.31 以降のビルドの NetScaler ADC プラットフォームに適用されます。

新しいボット署名

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4564	マーケティング	いいボット
4565	マーケティング	いいボット
4566	マーケティング	いいボット
4567	マーケティング	いいボット
4568	マーケティング	いいボット
4569	未分類	悪いボット
4570	未分類	悪いボット
4571	クローラー	いいボット
4572	クローラー	いいボット
4573	未分類	悪いボット
4574	未分類	悪いボット
4575	マーケティング	いいボット
4576	マーケティング	いいボット
4577	マーケティング	いいボット
4578	マーケティング	いいボット
4579	マーケティング	いいボット
4580	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4581	マーケティング	いいボット
4582	マーケティング	いいボット
4583	スクリーンショットクリエーター	いいボット
4584	検索エンジン	いいボット
4585	検索エンジン	いいボット
4586	スクリーンショットクリエーター	いいボット
4587	未分類	いいボット
4588	スピードテスター	いいボット
4589	クローラー	いいボット
4590	ツール	いいボット
4591	ツール	いいボット
4592	クローラー	悪いボット
4593	検索エンジン	いいボット
4594	検索エンジン	いいボット
4595	検索エンジン	いいボット
4596	マーケティング	いいボット
4597	ツール	いいボット
4598	検索エンジン	いいボット
4599	マーケティング	いいボット
4600	マーケティング	いいボット
4601	マーケティング	いいボット
4602	検索エンジン	いいボット
4603	未分類	いいボット
4604	マーケティング	いいボット
4605	マーケティング	いいボット
4606	未分類	悪いボット
4607	未分類	悪いボット
4608	ツール	いいボット
4609	未分類	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4610	ツール	いいボット
4611	ツール	いいボット
4612	スクレーパー	いいボット
4613	未分類	いいボット
4614	未分類	いいボット
4615	サイトモニター	いいボット
4616	クローラー	いいボット
4617	サイトモニター	いいボット
4618	検索エンジン	いいボット
4619	マーケティング	いいボット
4620	マーケティング	いいボット
4621	検索エンジン	いいボット
4622	クローラー	いいボット
4623	クローラー	いいボット
4624	クローラー	いいボット
4625	スクレーパー	いいボット
4626	クローラー	いいボット
4627	脆弱性スキャナ	いいボット
4628	ツール	いいボット
4629	未分類	悪いボット
4630	未分類	悪いボット
4631	ツール	いいボット
4632	フィードフェッチャー	いいボット
4633	クローラー	悪いボット
4634	未分類	いいボット
4635	フィードフェッチャー	いいボット
4636	未分類	いいボット
4637	ツール	いいボット
4638	ツール	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4639	スクレーパー	悪いボット
4640	未分類	悪いボット
4641	ツール	いいボット
4642	クローラー	悪いボット
4643	サイトモニター	いいボット
4644	サイトモニター	いいボット
4645	検索エンジン	いいボット
4646	検索エンジン	いいボット
4647	検索エンジン	いいボット
4648	検索エンジン	いいボット
4649	検索エンジン	悪いボット
4650	未分類	いいボット

ボットの署名を更新しました

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2554	未分類	悪いボット
3835	検索エンジン	いいボット
4027	検索エンジン	いいボット
4038	未分類	悪いボット
4085	ツール	いいボット
4098	クローラー	いいボット
4100	検索エンジン	いいボット
4220	ツール	いいボット
4224	検索エンジン	いいボット
4281	未分類	悪いボット
4412	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4425	マーケティング	いいボット
4429	スクリーンショットクリエーター	いいボット
4430	ウイルススキャナー	いいボット
4483	クローラー	いいボット
4552	未分類	いいボット
4562	検索エンジン	いいボット
1000000	ブラウザー	いいボット
1000003	ブラウザー	いいボット
1000004	スクレーパー	いいボット
1000005	Google_Crawler	悪いボット
1000006	ブラウザー	悪いボット
1000007	ボット	悪いボット
1000008	ブラウザー	悪いボット
1000009	ブラウザー	いいボット
1000010	ボット	悪いボット
1000011	ブラウザー	悪いボット
1000012	スクレーパー	いいボット
1000013	スクレーパー	悪いボット
1000014	スクレーパー	悪いボット
1000015	ブラウザー	いいボット
1000016	ボット	悪いボット
1000017	ブラウザー	悪いボット
1000018	ブラウザー	いいボット
1000019	スクレーパー	いいボット
1000020	スクレーパー	いいボット
1000021	スクレーパー	いいボット
1000022	Google_Crawler	いいボット
1000023	ブラウザー	悪いボット
1000024	Analyzer	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000025	Analyzer	いいボット
1000026	Analyzer	いいボット
1000027	Analyzer	いいボット
1000028	Analyzer	いいボット
1000029	ブラウザー	いいボット
1000030	Analyzer	いいボット
1000031	Analyzer	いいボット
1000032	ブラウザー	悪いボット
1000033	Analyzer	いいボット
1000034	ブラウザー	悪いボット
1000035	スクレーパー	いいボット
1000036	スクレーパー	いいボット
1000037	ブラウザー	いいボット
1000038	Analyzer	いいボット
1000039	Analyzer	いいボット
1000040	Analyzer	いいボット
1000041	Analyzer	いいボット
1000042	Analyzer	いいボット
1000043	Analyzer	いいボット
1000044	Analyzer	いいボット
1000045	Google_App_Engine_Software	いいボット
1000046	Google_Crawler	いいボット
1000047	ブラウザー	悪いボット
1000048	ブラウザー	悪いボット
1000049	Analyzer	いいボット
1000050	ブラウザー	悪いボット
1000051	ブラウザー	いいボット
1000052	ブラウザー	悪いボット
1000053	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000054	Google_Crawler	悪いボット
1000055	スクレーパー	悪いボット
1000056	Analyzer	いいボット
1000057	ブラウザー	悪いボット
1000058	ブラウザー	悪いボット
1000059	ブラウザー	悪いボット
1000060	スクレーパー	悪いボット
1000061	アプリケーション	悪いボット
1000062	スクレーパー	悪いボット
1000063	スクレーパー	悪いボット
1000064	スクレーパー	いいボット
1000065	スクレーパー	悪いボット
1000066	スクレーパー	悪いボット
1000067	ブラウザー	悪いボット
1000068	スクレーパー	悪いボット
1000069	ブラウザー	悪いボット
1000070	スクレーパー	悪いボット
1000071	アプリケーション	悪いボット

2022 年 **8** 月のボット署名の更新

August 15, 2023

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 13 は、13.0 76.31 以降のビルドを搭載したプラットフォームに適用されます。

新しいボット署名

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4651	マーケティング	いいボット
4652	未分類	悪いボット
4653	検索エンジン	いいボット
4654	ツール	いいボット
4655	クローラー	いいボット
4656	マーケティング	いいボット
4657	スクレーパー	いいボット
4658	フィードフェッチャー	いいボット
4659	未分類	悪いボット
4660	ツール	いいボット
4661	ツール	いいボット
4662	未分類	悪いボット
4663	未分類	悪いボット
4664	マーケティング	いいボット
4665	未分類	いいボット
4666	未分類	いいボット
4667	フィードフェッチャー	いいボット
4668	未分類	いいボット
4669	ツール	いいボット
4670	ツール	いいボット
4671	検索エンジン	いいボット
4672	ツール	いいボット
4673	未分類	いいボット
4674	未分類	いいボット
4675	未分類	いいボット
4676	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4677	スクレーパー	いいボット
4678	マーケティング	いいボット
4679	クローラー	悪いボット
4680	未分類	いいボット
4681	未分類	いいボット
4682	サイトモニター	いいボット
4683	サイトモニター	いいボット
4684	検索エンジン	いいボット
4685	検索エンジン	いいボット
4686	検索エンジン	いいボット
4687	検索エンジン	いいボット
4688	検索エンジン	いいボット
4689	検索エンジン	いいボット
4690	検索エンジン	いいボット
4691	検索エンジン	いいボット
4692	検索エンジン	いいボット
4693	未分類	いいボット
4694	未分類	悪いボット
4695	クローラー	いいボット
4696	クローラー	いいボット
4697	クローラー	いいボット
4698	検索エンジン	いいボット
4699	検索エンジン	いいボット
4700	検索エンジン	いいボット
4701	ツール	悪いボット
4702	未分類	いいボット
4703	ツール	いいボット
4704	ツール	いいボット
4705	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4706	サイトモニター	いいボット
4707	検索エンジン	いいボット
4708	ツール	いいボット
4709	脆弱性スキャナ	いいボット
4710	脆弱性スキャナ	いいボット
4711	クローラー	いいボット
4712	クローラー	いいボット
4713	クローラー	いいボット
4714	スクレーパー	いいボット
4715	ツール	いいボット
4716	ツール	いいボット
4717	検索エンジン	悪いボット
4718	未分類	いいボット
4719	ツール	いいボット
4720	マーケティング	いいボット
4721	マーケティング	いいボット
4722	検索エンジン	いいボット
4723	未分類	悪いボット
4724	ツール	いいボット
4725	検索エンジン	いいボット
4726	検索エンジン	いいボット
4727	ツール	いいボット
4728	未分類	悪いボット
4729	サイトモニター	いいボット
4730	検索エンジン	いいボット
4731	検索エンジン	いいボット
4732	検索エンジン	いいボット
4733	検索エンジン	いいボット
4734	ツール	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4735	ツール	悪いボット
4736	ツール	いいボット
4737	マーケティング	いいボット
4738	ツール	いいボット
4739	フィードフェッチャー	いいボット
4740	検索エンジン	いいボット
4741	未分類	悪いボット
4742	検索エンジン	いいボット
4743	クローラー	いいボット
4744	ツール	いいボット
4745	ツール	いいボット
4746	マーケティング	いいボット
4747	未分類	悪いボット
4748	検索エンジン	いいボット
4749	検索エンジン	いいボット
4750	検索エンジン	いいボット
4751	検索エンジン	いいボット
4752	検索エンジン	いいボット

ボットの署名を更新しました

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3796	スクレーパー	いいボット
3835	検索エンジン	いいボット
3935	スクレーパー	いいボット
4027	検索エンジン	いいボット
4061	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4100	検索エンジン	いいボット
4451	スピードテスター	いいボット
4562	検索エンジン	いいボット
4575	マーケティング	いいボット
4577	マーケティング	いいボット
4578	マーケティング	いいボット
4579	マーケティング	いいボット
4580	マーケティング	いいボット
4583	スクリーンショットクリエーター	いいボット
4584	検索エンジン	いいボット
4585	検索エンジン	いいボット
4597	ツール	いいボット
4599	マーケティング	いいボット
4601	マーケティング	いいボット
4623	クローラー	いいボット
4630	未分類	悪いボット
4647	検索エンジン	いいボット
1000000	ブラウザー	いいボット
1000001	アプリケーション	悪いボット
1000002	ブラウザー	いいボット
1000003	スクレーパー	いいボット
1000004	ブラウザー	いいボット
1000005	ブラウザー	悪いボット
1000006	Google Crawler	悪いボット
1000007	スクレーパー	悪いボット
1000008	スクレーパー	いいボット
1000009	ブラウザー	悪いボット
1000010	ボット	悪いボット
1000011	ボット	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000012	スクレーパー	悪いボット
1000013	スクレーパー	悪いボット
1000014	ブラウザー	悪いボット
1000015	ブラウザー	いいボット
1000016	ブラウザー	悪いボット
1000017	スクレーパー	いいボット
1000018	スクレーパー	悪いボット
1000019	スクレーパー	悪いボット
1000020	スクレーパー	悪いボット
1000021	ブラウザー	いいボット
1000022	スクレーパー	いいボット
1000023	ブラウザー	悪いボット
1000024	ボット	悪いボット
1000025	Analyzer	いいボット
1000026	スクレーパー	いいボット
1000027	ブラウザー	悪いボット
1000028	ブラウザー	悪いボット
1000029	スクレーパー	いいボット
1000030	Google Crawler	いいボット
1000031	ブラウザー	悪いボット
1000032	Analyzer	いいボット
1000033	ボット	悪いボット
1000034	Analyzer	いいボット
1000035	Analyzer	いいボット
1000036	Analyzer	いいボット
1000037	Analyzer	いいボット
1000038	スクレーパー	いいボット
1000039	Analyzer	いいボット
1000040	ブラウザー	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000041	ブラウザー	悪いボット
1000042	スクレーパー	いいボット
1000043	ブラウザー	いいボット
1000044	Analyzer	いいボット
1000045	Analyzer	いいボット
1000046	Analyzer	いいボット
1000047	Analyzer	いいボット
1000048	Analyzer	いいボット
1000049	ブラウザー	悪いボット
1000050	Google Crawler	いいボット
1000051	ブラウザー	悪いボット
1000052	ブラウザー	悪いボット
1000053	Analyzer	いいボット
1000054	ブラウザー	いいボット
1000055	スクレーパー	いいボット
1000056	ブラウザー	いいボット
1000057	Analyzer	いいボット
1000058	Google Crawler	悪いボット
1000059	スクレーパー	悪いボット
1000060	ブラウザー	悪いボット
1000061	ブラウザー	いいボット
1000062	ブラウザー	悪いボット
1000063	ブラウザー	悪いボット
1000064	ブラウザー	悪いボット
1000065	スクレーパー	悪いボット
1000066	アプリケーション	悪いボット
1000067	スクレーパー	悪いボット
1000068	スクレーパー	悪いボット
1000069	ブラウザー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000070	アプリケーション	悪いボット

2023 年 **4** 月のボットシグネチャの更新

August 15, 2023

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

シグネチャーバージョン 14 は、13.0 76.31 以降のビルドを搭載した NetScaler プラットフォームに適用されます。

新しいボット署名

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4753	ツール	悪いボット
4754	未分類	悪いボット
4755	スクレーパー	いいボット
4756	マーケティング	いいボット
4757	マーケティング	いいボット
4758	マーケティング	いいボット
4759	マーケティング	いいボット
4760	マーケティング	いいボット
4761	マーケティング	いいボット
4762	マーケティング	いいボット
4763	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4764	マーケティング	いいボット
4765	スクレーパー	悪いボット
4766	スクレーパー	悪いボット
4767	ツール	いいボット
4768	スクレーパー	悪いボット
4769	ツール	いいボット
4770	スクレーパー	悪いボット
4771	スクレーパー	悪いボット
4772	スクレーパー	悪いボット
4773	スクレーパー	悪いボット
4774	スクレーパー	悪いボット
4775	クローラー	いいボット
4776	マーケティング	いいボット
4777	ツール	いいボット
4778	ツール	いいボット
4779	クローラー	いいボット
4780	サイトモニター	いいボット
4781	サイトモニター	いいボット
4782	サイトモニター	いいボット
4783	未分類	いいボット
4784	ツール	いいボット
4785	ツール	いいボット
4786	脆弱性スキャナ	いいボット
4787	ツール	いいボット
4788	マーケティング	いいボット
4789	マーケティング	いいボット
4790	マーケティング	いいボット
4791	未分類	いいボット
4792	未分類	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4793	未分類	悪いボット
4794	未分類	悪いボット
4795	ツール	いいボット
4796	サイトモニター	いいボット
4797	サイトモニター	いいボット
4798	未分類	いいボット
4799	検索エンジン	いいボット
4800	検索エンジン	いいボット
4801	検索エンジン	いいボット
4802	未分類	いいボット
4803	ツール	悪いボット
4804	スクレーパー	いいボット
4805	マーケティング	いいボット
4806	クローラー	いいボット
4807	クローラー	いいボット
4808	脆弱性スキャナ	悪いボット
4809	脆弱性スキャナ	いいボット
4810	ツール	いいボット
4811	ツール	いいボット
4812	未分類	いいボット
4813	未分類	いいボット
4814	サイトモニター	いいボット
4815	スクレーパー	悪いボット
4816	検索エンジン	いいボット
4817	未分類	いいボット
4818	サイトモニター	いいボット
4819	検索エンジン	いいボット
4820	検索エンジン	いいボット
4821	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4822	検索エンジン	いいボット
4823	検索エンジン	いいボット
4824	未分類	いいボット
4825	マーケティング	いいボット
4826	スクレーパー	いいボット
4827	スクリーンショットクリエーター	いいボット
4828	未分類	悪いボット
4829	未分類	悪いボット
4830	未分類	悪いボット
4831	未分類	いいボット
4832	未分類	悪いボット
4833	検索エンジン	いいボット
4834	検索エンジン	いいボット
4835	未分類	いいボット
4836	検索エンジン	いいボット
4837	ツール	いいボット
4838	マーケティング	いいボット
4839	ツール	いいボット
4840	スクレーパー	いいボット
4841	検索エンジン	いいボット
4842	サイトモニター	いいボット
4843	未分類	悪いボット
4844	検索エンジン	いいボット
4845	検索エンジン	いいボット
4846	クローラー	いいボット
4847	マーケティング	いいボット
4848	ツール	いいボット
4849	クローラー	いいボット
4850	クローラー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4851	未分類	悪いボット
4852	検索エンジン	いいボット
4853	未分類	いいボット
4854	未分類	いいボット
4855	サイトモニター	いいボット
4856	ツール	いいボット
4857	ツール	いいボット
4858	スクレーパー	悪いボット
4859	スクリーンショットクリエーター	いいボット
4860	サイトモニター	いいボット
4861	サイトモニター	いいボット
4862	クローラー	いいボット
4863	検索エンジン	いいボット
4864	検索エンジン	いいボット
4865	検索エンジン	いいボット
4866	検索エンジン	いいボット
4867	検索エンジン	いいボット
4868	マーケティング	いいボット
4869	マーケティング	いいボット
4870	検索エンジン	いいボット
4871	未分類	悪いボット
4872	未分類	悪いボット
4873	未分類	悪いボット
4874	未分類	悪いボット
4875	未分類	悪いボット
4876	未分類	悪いボット
4877	未分類	悪いボット
4878	未分類	悪いボット
4879	未分類	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4880	未分類	いいボット
4881	検索エンジン	いいボット
4882	未分類	いいボット
4883	ツール	いいボット
4884	ツール	いいボット
4885	ツール	いいボット
4886	サイトモニター	いいボット
4887	サイトモニター	いいボット
4888	スクレーパー	悪いボット
4889	マーケティング	いいボット
4890	未分類	悪いボット
4891	検索エンジン	いいボット
4892	検索エンジン	いいボット
4893	マーケティング	いいボット
4894	未分類	悪いボット
4895	未分類	悪いボット
4896	脆弱性スキャナ	いいボット
4897	未分類	悪いボット
4898	未分類	悪いボット
4899	未分類	悪いボット
4900	クローラー	いいボット
4901	クローラー	いいボット
4902	脆弱性スキャナ	いいボット
4903	ツール	いいボット
4904	フィードフェッチャー	いいボット
4905	ツール	いいボット
4906	クローラー	いいボット
4907	未分類	いいボット
4908	未分類	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4909	未分類	いいボット
4910	検索エンジン	いいボット
4911	検索エンジン	いいボット
4912	未分類	いいボット
4913	検索エンジン	いいボット
4914	クローラー	いいボット

ボットの署名を更新しました

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3935	スクレーパー	いいボット
4012	クローラー	いいボット
4013	検索エンジン	いいボット
4027	検索エンジン	いいボット
4038	未分類	悪いボット
4071	ツール	いいボット
4100	検索エンジン	いいボット
4220	ツール	いいボット
4425	マーケティング	いいボット
4441	フィードフェッチャー	いいボット
4451	スピードテスター	いいボット
4563	検索エンジン	いいボット
4575	マーケティング	いいボット
4577	マーケティング	いいボット
4578	マーケティング	いいボット
4579	マーケティング	いいボット
4580	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4583	スクリーンショットクリエーター	いいボット
4584	検索エンジン	いいボット
4585	検索エンジン	いいボット
4586	スクリーンショットクリエーター	いいボット
4593	検索エンジン	いいボット
4597	ツール	いいボット
4599	マーケティング	いいボット
4600	マーケティング	いいボット
4601	マーケティング	いいボット
4618	検索エンジン	いいボット
4633	クローラー	悪いボット
4639	スクレーパー	悪いボット
4647	検索エンジン	いいボット
4651	マーケティング	いいボット
4660	ツール	いいボット
4687	検索エンジン	いいボット
4717	検索エンジン	悪いボット
4730	検索エンジン	いいボット
1000000	ブラウザー	いいボット
1000001	アプリケーション	悪いボット
1000002	ブラウザー	いいボット
1000003	スクレーパー	いいボット
1000004	ブラウザー	いいボット
1000005	ブラウザー	悪いボット
1000006	Google_Crawler	悪いボット
1000007	スクレーパー	悪いボット
1000008	スクレーパー	いいボット
1000009	ブラウザー	悪いボット
1000010	ボット	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000011	ボット	悪いボット
1000012	スクレーパー	悪いボット
1000013	スクレーパー	悪いボット
1000014	ブラウザー	悪いボット
1000015	ブラウザー	いいボット
1000016	ブラウザー	悪いボット
1000017	スクレーパー	いいボット
1000018	スクレーパー	悪いボット
1000019	スクレーパー	悪いボット
1000020	スクレーパー	悪いボット
1000021	ブラウザー	いいボット
1000022	スクレーパー	いいボット
1000023	ブラウザー	悪いボット
1000024	ボット	悪いボット
1000025	Analyzer	いいボット
1000026	スクレーパー	いいボット
1000027	ブラウザー	悪いボット
1000028	ブラウザー	悪いボット
1000029	スクレーパー	いいボット
1000030	Google_Crawler	いいボット
1000031	ブラウザー	悪いボット
1000032	Analyzer	いいボット
1000033	ボット	悪いボット
1000034	Analyzer	いいボット
1000035	Analyzer	いいボット
1000036	Analyzer	いいボット
1000037	Analyzer	いいボット
1000038	スクレーパー	いいボット
1000039	Analyzer	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000040	ブラウザー	悪いボット
1000041	ブラウザー	悪いボット
1000042	スクレーパー	いいボット
1000043	ブラウザー	いいボット
1000044	Analyzer	いいボット
1000045	Analyzer	いいボット
1000046	Analyzer	いいボット
1000047	Analyzer	いいボット
1000048	Analyzer	いいボット
1000049	ブラウザー	悪いボット
1000050	Google_Crawler	いいボット
1000051	ブラウザー	悪いボット
1000052	ブラウザー	悪いボット
1000053	Analyzer	いいボット
1000054	ブラウザー	いいボット
1000055	スクレーパー	いいボット
1000056	ブラウザー	いいボット
1000057	Analyzer	いいボット
1000058	Google_Crawler	悪いボット
1000059	スクレーパー	悪いボット
1000060	ブラウザー	悪いボット
1000061	ブラウザー	いいボット
1000062	ブラウザー	悪いボット
1000063	ブラウザー	悪いボット
1000064	ブラウザー	悪いボット
1000065	スクレーパー	悪いボット
1000066	アプリケーション	悪いボット
1000067	スクレーパー	悪いボット
1000068	スクレーパー	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000069	ブラウザー	いいボット
1000070	アプリケーション	悪いボット

2023 年 **5** 月のボットシグネチャの更新

April 15, 2024

2023-05-15 週に特定されたボットについて、新しい署名ルールが生成されます。
これらのシグニチャルールをダウンロードして設定することで、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

シグネチャーバージョン 15 は、ビルドが 13.0 76.31 以降の Netscaler プラットフォームに適用されます。

このバージョンの新しいボットシグネチャ

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2000002	ブラウザー	いいボット
2000008	スクレーパー	いいボット
2000009	ブラウザー	悪いボット
2000012	スクレーパー	悪いボット
2000017	スクレーパー	いいボット
2000018	スクレーパー	悪いボット
2000019	スクレーパー	悪いボット
2000020	スクレーパー	悪いボット
2000024	ボット	悪いボット
2000027	ブラウザー	悪いボット
2000028	ブラウザー	悪いボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2000031	ブラウザー	悪いボット
2000032	Analyzer	いいボット
2000033	ボット	悪いボット
2000034	Analyzer	いいボット
2000040	ブラウザー	悪いボット
2000041	ブラウザー	悪いボット
2000047	Analyzer	いいボット
2000048	Analyzer	いいボット
2000049	ブラウザー	悪いボット
2000050	Google_Crawler	いいボット
2000051	ブラウザー	悪いボット
2000054	ブラウザー	いいボット
2000055	スクレーパー	いいボット
2000057	Analyzer	いいボット
2000061	ブラウザー	いいボット
2000064	ブラウザー	悪いボット
2000069	ブラウザー	いいボット

このバージョンではボットシグネチャが更新されました

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
71	クローラー	いいボット
74	クローラー	いいボット
75	クローラー	いいボット
372	スクレーパー	いいボット
373	スクレーパー	いいボット
374	スクレーパー	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
375	スクレーパー	いいボット
376	スクレーパー	いいボット
377	スクレーパー	いいボット
378	スクレーパー	いいボット
379	スクレーパー	いいボット
380	スクレーパー	いいボット
381	スクレーパー	いいボット
382	スクレーパー	いいボット
383	スクレーパー	いいボット
384	スクレーパー	いいボット
385	スクレーパー	いいボット
386	スクレーパー	いいボット
387	スクレーパー	いいボット
389	スクレーパー	いいボット
390	スクレーパー	いいボット
391	スクレーパー	いいボット
702	検索エンジン	いいボット
703	検索エンジン	いいボット
1173	ツール	いいボット
1174	マーケティング	いいボット
1176	検索エンジン	いいボット
1178	スピードテスター	いいボット
1185	スクリーンショットクリエーター	いいボット
1209	未分類	いいボット
1531	フィードフェッチャー	いいボット
2586	未分類	いいボット
2674	ツール	いいボット
2756	ツール	いいボット
2758	未分類	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
2759	ツール	いいボット
2784	ツール	いいボット
2952	ツール	いいボット
3163	ツール	いいボット
3554	ツール	いいボット
3782	マーケティング	いいボット
3788	ツール	いいボット
3797	マーケティング	いいボット
3798	マーケティング	いいボット
3799	マーケティング	いいボット
3800	マーケティング	いいボット
3801	マーケティング	いいボット
3802	スクリーンショットクリエーター	いいボット
3803	検索エンジン	いいボット
3804	スクリーンショットクリエーター	いいボット
3805	検索エンジン	いいボット
3861	マーケティング	いいボット
3862	マーケティング	いいボット
3863	マーケティング	いいボット
3864	マーケティング	いいボット
3865	マーケティング	いいボット
3866	マーケティング	いいボット
3867	マーケティング	いいボット
3868	マーケティング	いいボット
3871	マーケティング	いいボット
3872	マーケティング	いいボット
3873	検索エンジン	いいボット
3874	検索エンジン	いいボット
3875	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3876	検索エンジン	いいボット
3877	スクリーンショットクリエーター	いいボット
3878	検索エンジン	いいボット
3879	検索エンジン	いいボット
3880	スクリーンショットクリエーター	いいボット
3881	スクリーンショットクリエーター	いいボット
3882	検索エンジン	いいボット
3883	検索エンジン	いいボット
3884	検索エンジン	いいボット
3885	検索エンジン	いいボット
3963	マーケティング	いいボット
3970	検索エンジン	いいボット
3973	検索エンジン	いいボット
3974	検索エンジン	いいボット
3975	検索エンジン	いいボット
3976	検索エンジン	いいボット
3977	検索エンジン	いいボット
3979	検索エンジン	いいボット
3981	検索エンジン	いいボット
3982	検索エンジン	いいボット
3983	検索エンジン	いいボット
3984	検索エンジン	いいボット
3985	検索エンジン	いいボット
3986	検索エンジン	いいボット
3988	検索エンジン	いいボット
3989	検索エンジン	いいボット
3990	検索エンジン	いいボット
3991	検索エンジン	いいボット
3992	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
3993	検索エンジン	いいボット
3994	検索エンジン	いいボット
3995	検索エンジン	いいボット
3996	検索エンジン	いいボット
3997	検索エンジン	いいボット
3998	検索エンジン	いいボット
3999	検索エンジン	いいボット
4001	検索エンジン	いいボット
4002	検索エンジン	いいボット
4003	検索エンジン	いいボット
4004	検索エンジン	いいボット
4040	クローラー	いいボット
4041	ツール	いいボット
4120	マーケティング	いいボット
4122	マーケティング	いいボット
4123	マーケティング	いいボット
4124	マーケティング	いいボット
4125	マーケティング	いいボット
4133	マーケティング	いいボット
4134	ツール	いいボット
4135	マーケティング	いいボット
4136	マーケティング	いいボット
4137	マーケティング	いいボット
4138	マーケティング	いいボット
4139	マーケティング	いいボット
4140	マーケティング	いいボット
4141	マーケティング	いいボット
4142	マーケティング	いいボット
4143	マーケティング	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4144	マーケティング	いいボット
4145	検索エンジン	いいボット
4146	検索エンジン	いいボット
4147	検索エンジン	いいボット
4148	検索エンジン	いいボット
4149	検索エンジン	いいボット
4150	検索エンジン	いいボット
4151	検索エンジン	いいボット
4152	検索エンジン	いいボット
4153	検索エンジン	いいボット
4154	検索エンジン	いいボット
4155	検索エンジン	いいボット
4156	スクリーンショットクリエーター	いいボット
4157	検索エンジン	いいボット
4158	検索エンジン	いいボット
4159	検索エンジン	いいボット
4160	スクリーンショットクリエーター	いいボット
4161	検索エンジン	いいボット
4162	検索エンジン	いいボット
4163	ツール	いいボット
4164	検索エンジン	いいボット
4209	検索エンジン	いいボット
4240	マーケティング	いいボット
4241	マーケティング	いいボット
4247	検索エンジン	いいボット
4248	検索エンジン	いいボット
4249	スクリーンショットクリエーター	いいボット
4250	検索エンジン	いいボット
4251	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4282	マーケティング	いいボット
4283	マーケティング	いいボット
4284	マーケティング	いいボット
4285	マーケティング	いいボット
4286	マーケティング	いいボット
4287	マーケティング	いいボット
4288	マーケティング	いいボット
4289	マーケティング	いいボット
4290	マーケティング	いいボット
4291	マーケティング	いいボット
4292	マーケティング	いいボット
4293	マーケティング	いいボット
4294	マーケティング	いいボット
4295	検索エンジン	いいボット
4296	検索エンジン	いいボット
4297	検索エンジン	いいボット
4298	検索エンジン	いいボット
4299	検索エンジン	いいボット
4300	検索エンジン	いいボット
4301	検索エンジン	いいボット
4302	検索エンジン	いいボット
4303	検索エンジン	いいボット
4304	検索エンジン	いいボット
4305	検索エンジン	いいボット
4306	スクリーンショットクリエーター	いいボット
4307	検索エンジン	いいボット
4308	検索エンジン	いいボット
4309	検索エンジン	いいボット
4310	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4311	スクリーンショットクリエーター	いいボット
4312	検索エンジン	いいボット
4313	検索エンジン	いいボット
4314	検索エンジン	いいボット
4315	検索エンジン	いいボット
4316	検索エンジン	いいボット
4317	検索エンジン	いいボット
4318	スクリーンショットクリエーター	いいボット
4319	スクリーンショットクリエーター	いいボット
4345	マーケティング	いいボット
4346	マーケティング	いいボット
4347	マーケティング	いいボット
4348	マーケティング	いいボット
4349	マーケティング	いいボット
4350	マーケティング	いいボット
4351	マーケティング	いいボット
4352	マーケティング	いいボット
4353	マーケティング	いいボット
4354	マーケティング	いいボット
4355	検索エンジン	いいボット
4356	検索エンジン	いいボット
4357	検索エンジン	いいボット
4358	検索エンジン	いいボット
4359	検索エンジン	いいボット
4360	検索エンジン	いいボット
4361	検索エンジン	いいボット
4362	検索エンジン	いいボット
4363	検索エンジン	いいボット
4364	検索エンジン	いいボット

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
4365	スクリーンショットクリエーター	いいボット
4366	検索エンジン	いいボット
4367	検索エンジン	いいボット
4368	検索エンジン	いいボット
4369	検索エンジン	いいボット
4370	スクリーンショットクリエーター	いいボット
4371	検索エンジン	いいボット
4372	検索エンジン	いいボット
4373	検索エンジン	いいボット
4374	検索エンジン	いいボット
4375	検索エンジン	いいボット
4376	スクリーンショットクリエーター	いいボット
4379	検索エンジン	いいボット
4380	検索エンジン	いいボット
4381	検索エンジン	いいボット
4412	マーケティング	いいボット
4421	スクリーンショットクリエーター	いいボット
4450	スクリーンショットクリエーター	いいボット
4452	検索エンジン	いいボット
4503	マーケティング	いいボット
4541	マーケティング	いいボット
4584	検索エンジン	いいボット
4585	検索エンジン	いいボット
4602	検索エンジン	いいボット
1000003	スクレーパー	いいボット

このバージョンで削除されたボットシグネチャ

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
1000002	ブラウザー	いいボット
1000008	スクレーパー	いいボット
1000009	ブラウザー	悪いボット
1000012	スクレーパー	悪いボット
1000017	スクレーパー	いいボット
1000018	スクレーパー	悪いボット
1000019	スクレーパー	悪いボット
1000020	スクレーパー	悪いボット
1000024	ボット	悪いボット
1000027	ブラウザー	悪いボット
1000028	ブラウザー	悪いボット
1000031	ブラウザー	悪いボット
1000032	Analyzer	いいボット
1000033	ボット	悪いボット
1000034	Analyzer	いいボット
1000040	ブラウザー	悪いボット
1000041	ブラウザー	悪いボット
1000047	Analyzer	いいボット
1000048	Analyzer	いいボット
1000049	ブラウザー	悪いボット
1000050	Google_Crawler	いいボット
1000051	ブラウザー	悪いボット
1000054	ブラウザー	いいボット
1000055	スクレーパー	いいボット
1000057	Analyzer	いいボット
1000061	ブラウザー	いいボット
1000064	ブラウザー	悪いボット
1000069	ブラウザー	いいボット

2024 年 3 月のボットシグネチャの更新

April 15, 2024

2024-03-06 週に特定されたボットについて、更新された署名ルールが生成されます。
これらのシグニチャルールをダウンロードして設定することで、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

シグネチャーバージョン 16 は、ビルドが 13.0 76.31 以降の Netscaler プラットフォームに適用されます。

このバージョンではボットシグネチャが更新されました

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボット署名 **ID**	ボットカテゴリ	ボットタイプ
382	スクレーパー	いいボット
628	検索エンジン	いいボット

キャッシュリダイレクト

August 15, 2023

一般的な展開では、さまざまなクライアントが Web サーバーに同じコンテンツを繰り返し要求します。元の Web サーバーが各要求を処理する負担を軽減するために、キャッシュリダイレクトが有効な NetScaler アプライアンスは、元のサーバーからではなくキャッシュサーバーからこのコンテンツを提供できます。

NetScaler アプライアンスは、受信したリクエストを分析し、キャッシュ可能なデータのリクエストをキャッシュサーバーに送信し、キャッシュ不可能なリクエストと動的 HTTP リクエストをオリジンサーバーに送信します。

キャッシュリダイレクトはポリシーベースの機能です。デフォルトでは、ポリシーに一致するリクエストはオリジンサーバーに送信され、他のすべてのリクエストはキャッシュサーバーに送信されます。テストやメンテナンスの場合は、ポリシーの評価をスキップして、すべてのリクエストをキャッシュまたはオリジンサーバーに送ることをお勧めします。

コンテンツスイッチングとキャッシュリダイレクトを組み合わせて、選択的なコンテンツをキャッシュし、特定の種類の要求コンテンツに対して特定のキャッシュサーバーからコンテンツを提供できます。

キャッシュリダイレクト用に構成された NetScaler アプライアンスは、ネットワークのエッジ、オリジンサーバーの前面、またはネットワークバックボーンの任意の場所に導入できます。インターネットサービスプロバイダー(ISP)、ケーブル会社、コンテンツ配信配信ネットワーク、エンタープライズネットワークで一般的に使用されているエッジ展開では、NetScaler アプライアンスはクライアントの目の前に直接配置されます。サーバー側の展開では、NetScaler アプライアンスはオリジンサーバーの近くにあります。

キャッシュリダイレクトは HTTP サービスタイプで最も一般的に使用されますが、セキュア HTTPS プロトコルもサポートしています。

キャッシュリダイレクトのポリシー

August 15, 2023

キャッシュリダイレクション仮想サーバーは、受信リクエストごとにキャッシュリダイレクションポリシーを適用します。デフォルトでは、リクエストが設定されたポリシーのいずれかに一致する場合、そのリクエストはキャッシュ不可と見なされ、NetScaler アプライアンスはそれをオリジンサーバーに送信します。他のリクエストはキャッシュサーバーに送信されます。この動作を逆にして、設定されたキャッシュリダイレクトポリシーと一致するリクエストがキャッシュサーバーに送信されるようにすることができます。

アプライアンスは、キャッシュリダイレクションのポリシーセットを提供します。これらの組み込みポリシーが導入環境に適していない場合は、ユーザー定義のキャッシュリダイレクションポリシーを設定できます。

注: 使用する組み込みキャッシュリダイレクトポリシーを決定するか、ユーザー定義のポリシーを作成したら、キャッシュリダイレクトの設定に進みます。この機能を使用するには、少なくとも 1 つのキャッシュリダイレクション仮想サーバーを構成する必要があります。通常の操作では、少なくとも 1 つのキャッシュリダイレクションポリシーをその仮想サーバーにバインドする必要があります。

組み込みキャッシュリダイレクトのポリシー

August 15, 2023

NetScaler ADC アプライアンスには、一般的なキャッシュ要求を処理する組み込みのキャッシュリダイレクトポリシーが用意されています。これらのポリシーは、HTTP メソッド、着信リクエストの URL または URL トークン、HTTP バージョン、またはリクエスト内の HTTP ヘッダーとその値に基づいています。

組み込みのキャッシュリダイレクトポリシーは、仮想サーバーに直接バインドできるため、追加の構成は必要ありません。

キャッシュリダイレクションポリシーは、クラシックポリシーと詳細ポリシーの 2 種類のアプライアンス式言語を使用します。これらの言語の詳細については、「ポリシーと式」を参照してください。

組み込みのクラシックキャッシュリダイレクトポリシー

従来の式に基づく組み込みのキャッシュリダイレクトポリシーは、クラシックキャッシュリダイレクトポリシーと呼ばれます。クラシック式とその設定方法の詳細については、ポリシーと式を参照してください。

従来のキャッシュリダイレクションポリシーは、トラフィックおよびその他のデータの基本的な特性を評価します。たとえば、従来のキャッシュリダイレクションポリシーでは、HTTP 要求または応答に特定の種類のヘッダーまたは URL が含まれているかどうかを判断できます。

NetScaler ADC アプライアンスには、次の組み込みのクラシックキャッシュリダイレクトポリシーが用意されています。

組み込みポリシー名	説明
バイパス-非取得	リクエストが GET 以外の HTTP メソッドを使用する場合は、キャッシュをバイパスします。
バイパスキャッシュ制御	リクエストヘッダーに Cache-Control: no-cache または Cache-Control: no-store ヘッダーが含まれている場合、または HTTP リクエストにプラグマヘッダーが含まれている場合は、キャッシュをバイパスします。
バイパス-ダイナミック URL	URL がコンテンツが動的であることを示唆している場合は、キャッシュをバイパスします。cgi、asp、exe、cfm、ex、shtml、または htx のいずれかの拡張子が存在することが示されます。また、URL が /cgi-bin/、/bin/、または /exec/ のいずれかで始まる場合は、キャッシュをバイパスします。
urltokens をバイパスする	URL の次のトークンのいずれかで示されるように、リクエストは動的であるため、キャッシュをバイパスします。?、!、または =。
バイパスクッキー	Cookie ヘッダーと拡張子が.png または.jpg 以外のすべての URL のキャッシュをバイパスします。

組み込みの高度なポリシーキャッシュリダイレクトポリシー

高度なポリシー式に基づく組み込みキャッシュリダイレクトポリシーは、高度なポリシーキャッシュリダイレクトポリシーと呼ばれます。高度なポリシー式とその設定方法の詳細については、「ポリシーと式」を参照してください。

従来のキャッシュリダイレクションポリシーと同じタイプの評価に加えて、高度なポリシーキャッシュリダイレクトポリシーを使用すると、より多くのデータ（HTTP 要求の本文など）を分析し、ポリシールールでより多くの操作を構成できます（たとえば、要求をキャッシュまたはオリジンサーバー)。

NetScaler ADC アプライアンスは、高度なポリシーキャッシュリダイレクトポリシーに対して次の 2 つの組み込みアクションを提供します。

- キャッシュ
- ORIGIN

名前が示すとおり、リクエストはそれぞれキャッシュサーバーまたはオリジンサーバーに送信されます。

注：組み込みの詳細ポリシーキャッシュリダイレクトポリシーを使用している場合は、アクションを変更できません。

NetScaler ADC アプライアンスには、次の組み込みの詳細ポリシーキャッシュリダイレクトポリシーが用意されています。

組み込みポリシー名	説明
bypass-non-get_adv	リクエストが GET 以外の HTTP メソッドを使用する場合は、キャッシュをバイパスします。
bypass-cache-control_adv	リクエストヘッダーに Cache-Control: no-cache または Cache-Control: no-store ヘッダーが含まれている場合、または HTTP リクエストにプラグマヘッダーが含まれている場合は、キャッシュをバイパスします。
bypass-dynamic-url_adv	URL がコンテンツが動的であることを示唆している場合は、キャッシュをバイパスします。cgi、asp、exe、cfm、ex、shtml、または htx のいずれかの拡張子が存在することが示されます。また、URL が /cgi-bin/、/bin/、または /exec/ のいずれかで始まる場合は、キャッシュをバイパスします。
bypass-urltokens_adv	URL の次のトークンのいずれかで示されるように、リクエストは動的であるため、キャッシュをバイパスします。?、!、または =。
bypass-cookie_adv	Cookie ヘッダーと拡張子が.png または.jpg 以外のすべての URL のキャッシュをバイパスします。

組み込みキャッシュリダイレクトポリシーを表示する

使用可能なキャッシュリダイレクションポリシーを表示するには、コマンドラインインターフェイスまたは構成ユーティリティを使用します。

CLI を使用して組み込みキャッシュリダイレクトポリシーを表示する

コマンドプロンプトで入力します。

```
show cr policy [<policyName>]
```

例:


```
> show cr policy
1)       Cache-By-Pass RULE: NS_NON_GET          Policy:bypass-non-get
2)       Cache-By-Pass RULE: (NS_CACHECONTROL_NOSTORE ||
   NS_CACHECONTROL_NOCACHE || NS_HEADER_PRAGMA)    Policy:bypass-cache-
   control
3)       Cache-By-Pass RULE: (NS_EXT_CGI || NS_EXT_ASP || NS_EXT_EXE ||
    NS_EXT_CFM || NS_EXT_EX || NS_EXT_SHTML || NS_EXT_HTX) || (
   NS_URL_PATH_CGIBIN || NS_URL_PATH_EXEC || NS_URL_PATH_BIN)
   Policy:bypass-dynamic-url
4)       Cache-By-Pass RULE: NS_URL_TOKENS       Policy:bypass-
   urltokens
5)       Cache-By-Pass RULE: (NS_HEADER_COOKIE && NS_EXT_NOT_GIF &&
   NS_EXT_NOT_JPEG)     Policy:bypass-cookie
 Done
<!--NeedCopy-->
```

GUI を使用して組み込みキャッシュリダイレクトポリシーを表示する

1. [トラフィック管理] > [キャッシュリダイレクト] > [ポリシー] に移動します。構成されたキャッシュリダイレクションポリシーが詳細ペインに表示されます。
2. 設定済みのポリシーの 1 つを選択して、詳細を表示します。

キャッシュリダイレクトのポリシーの構成

August 15, 2023

キャッシュリダイレクションポリシーには、式 (ルールとも呼ばれる) が含まれます。式は、クライアント要求がポリシーと比較されるときに評価される条件を表します。

キャッシュリダイレクションポリシーのアクションは明示的に構成しません。

キャッシュリダイレクションポリシーには名前があり、高度なポリシー式、または論理演算子と次の組み込みアクションを使用して組み合わされた一連の高度なポリシー式句が含まれます。

- キャッシュ
- ORIGIN

高度なポリシー式の詳細については、「ポリシーと式」を参照してください。

CLI を使用したキャッシュリダイレクションポリシーの追加

コマンドプロンプトで次のコマンドを入力して、キャッシュリダイレクトポリシーを追加し、構成を確認します。


```
-  add cr policy <policyName> \*\*-rule\*\* <expression> -action<string
   > [-logAction<string>]

-  show cr policy [<policyName>]

<!--NeedCopy-->
```

例:

単純な表現のポリシー:

```
> add cr policy crpol1 -rule !(HTTP.REQ.URL.ENDSWITH(".jpeg")) -action
   origin
 Done
> show cr policy crpoll
    Policy: crpol1 Rule: !(HTTP.REQ.URL.ENDSWITH(".jpeg")) Action:
       ORIGIN
    Hits: 0
 Done

<!--NeedCopy-->
```

複合式を含むポリシー:

```
> add cr policy crpol11 -rule 'http.req.method.eq(post) && (HTTP.REQ.
   URL.ENDSWITH(".png") || HTTP.REQ.URL.ENDSWITH(".cgi"))'  -action
   cache
 Done
> show cr policy crpol11
    Policy: crpol11    Rule: http.req.method.eq(post) && (HTTP.REQ.URL.
       ENDSWITH(".png") || HTTP.REQ.URL.ENDSWITH(".cgi"))  Action:
       CACHE
    Hits: 0
 Done

<!--NeedCopy-->
```

ヘッダーを評価するポリシー:

```
> add cr policy crpol12 -rule http.req.header("If-Modified-Since").
   exists -action origin
 Done
> show cr policy crpol12
    Policy: crpol12    Rule: http.req.header("If-Modified-Since").
       exists   Action: ORIGIN
    Hits: 0
 Done

<!--NeedCopy-->
```


CLI を使用してキャッシュリダイレクトポリシーを変更または削除する

- キャッシュリダイレクションポリシーを変更するには、set cr policy コマンドを使用します。これは add cr policy コマンドとほぼ同じですが、既存のポリシーの名前を入力し、変更するパラメータを指定するだけで済む点が異なります。
- ポリシーを削除するには、<name> 引数のみを受け入れる`rm cr policy`コマンドを使用します。ポリシーが仮想サーバにバインドされている場合は、ポリシーを削除する前に、ポリシーをバインド解除する必要があります。

キャッシュリダイレクトポリシーのバインド解除の詳細については、「キャッシュリダイレクト仮想サーバーからポリシーをバインド解除する」を参照してください。

GUI を使用して単純な式を使用してキャッシュリダイレクションポリシーを構成する

1. トラフィック管理 **>** キャッシュリダイレクト **>** ポリシーに移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [キャッシュリダイレクトポリシーの作成] ダイアログボックスの [名前] テキストボックスに、ポリシーの名前を入力します。
4. 「アクション」ドロップダウンリストから適切なアクション「キャッシュ」または「オリジン」を選択します。
5. [ログアクション] 領域で、[追加] をクリックします。[監査メッセージアクションの作成] ダイアログボックスに名前を入力します。
    - ドロップダウンリストから適切な値を選択して、ログレベルを設定します。
        - 緊急
        - アラート
        - クリティカル
        - エラー
        - 警告
        - 通知
        - 情報提供
        - デバッグ
    - [式] 領域に式を入力します。
        - 式タイプ-一般
        - フロータイプ-REQ
        - プロトコル-HTTP
        - 修飾子-URL
        - オペレーター-!=
        - 値-/.jpeg

- ［作成］をクリックします。

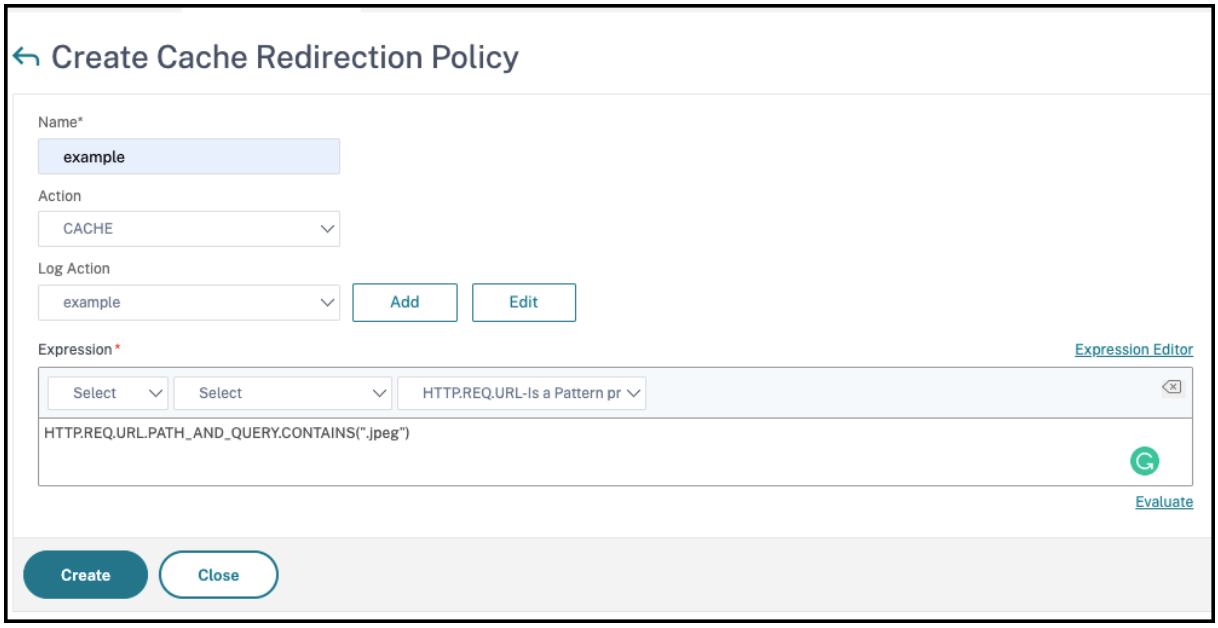
6. 単純式を設定するには、式を入力します。次に示すのは、URL 内の.jpeg拡張子をチェックする式の例です。

   - 式タイプ-一般
   - フロータイプ-REQ
   - プロトコル-HTTP
   - 修飾子-URL
   - オペレーター-!=
   - 値-/.jpeg

   次の例の単純な式は、リクエスト内の IF-Modified-Since ヘッダーをチェックします。

   - 式タイプ-一般
   - フロータイプ-REQ
   - プロトコル-HTTP
   - 修飾子-HEADER
   - 演算子-EXIST
   - ヘッダー名-If-Modified-Snice

7. 式の入力が完了したら、[作成] をクリックします。



GUI を使用した複合式を使用したキャッシュリダイレクションポリシーの設定

1. トラフィック管理 **>** キャッシュリダイレクト **>** ポリシーに移動します。
2. 詳細ペインで、[追加] をクリックします。

3. [**Name**] テキストボックスに、ポリシーの名前を入力します。

   名前は、英字、数字、またはアンダースコア記号で始まり、1 ～127 の文字、数字、およびハイフン（-)、ピリオド（.)、シャープ（#)、スペース（)、アットマーク（@)、等号（=)、およびアンダースコア（_）記号で構成できます。このポリシーが検出するために作成されたコンテンツの種類を他のユーザーがわかりやすいように名前を選択してください。

4.「アクション」ドロップダウンリストから適切なアクション「キャッシュ」または「オリジン」を選択します。

5. [ログアクション] 領域で、[追加] をクリックします。[監査メッセージアクションの作成] ダイアログボックスに名前を入力します。

   - ドロップダウンリストから適切な値を選択して、ログレベルを設定します。
     - 緊急
     - アラート
     - クリティカル
     - エラー
     - 警告
     - 通知
     - 情報提供
     - デバッグ
   - [式] 領域に式を入力します。
     - 式タイプ-一般
     - フロータイプ-REQ
     - プロトコル-HTTP
     - 修飾子-URL
     - オペレーター-!=
     - 値-/.jpeg
   - ［作成］をクリックします。

6. 作成する複合式のタイプを選択します。選択肢は次のとおりです：

   - 任意の式にマッチします。ポリシーは、1 つ以上の個別の式がトラフィックに一致する場合、トラフィックと一致します。
   - [すべての式に一致**]**。ポリシーは、個々の式がすべてトラフィックに一致する場合にのみ、トラフィックを照合します。
   - 表形式の表現。[式] リストを 3 列の表形式に切り替えます。右端の列に、次のいずれかの演算子を配置します。
     - AND [&&] 演算子、つまりポリシーに一致させるには、リクエストが現在の式と次の式の両方と一致する必要があります。

OR [

–

既存の式を選択し、次のいずれかの演算子をクリックして、ネストされたサブグループ内の式をグループ化することもできます。

- BEGIN SUBGROUP [+ (] 演算子。NetScaler アプライアンスに、選択した式でネストされたサブグループを開始するように指示します。(式からこの演算子を削除するには、-(.) をクリックします。
- END SUBGROUP [+)] 演算子。NetScaler アプライアンスに、現在ネストされているサブグループを選択した式で終了するように指示します。(式からこの演算子を削除するには、[-] をクリックします)。

- 高度な自由形式。エクスプレッションエディタ (Expressions Editor) を完全にオフにし、エクスプレッションリストを、複合エクスプレッションを入力できるテキスト領域にします。これは、ポリシー式を作成するための最も強力で難しい方法であり、NetScaler ADC クラシック式言語に精通している方のみに推奨されます。

  アドバンスト・フリー・フォーム・テキスト・エリアでのクラシック式の作成の詳細については、「クラシックポリシーと式の設定」を参照してください。

  注意**:** アドバンスフリーフォーム式編集モードに切り替えると、他のモードに戻すことはできません。このエクスプレッション編集モードは、使用が確実でない限り選択しないでください。

7. [任意の式に一致]、[すべての式に一致]、または [表形式の式] を選択した場合は、[追加] をクリックして [式の追加] ダイアログボックスを表示します。

   キャッシュリダイレクションポリシーの場合は、式の種類を [全般] に設定しておく必要があります。

8. [フロータイプ（Flow Type）] ドロップダウンリストで、式のフロータイプを選択します。

   フロータイプは、ポリシーが着信接続と発信接続のどちらを検査するかを決定します。次の 2 つの選択肢があります。

   - **REQ**。着信接続または要求を検査するように NetScaler ADC アプライアンスを構成します。
   - **RES**。発信接続または応答を検査するようにアプライアンスを設定します。

9. [プロトコル（Protocol）] ドロップダウンリストで、式のプロトコルを選択します。

   プロトコルは、ポリシーが要求または応答で調べる情報のタイプを決定します。前のドロップダウンリストで [REQ] と [RES] のどちらを選択したかに応じて、次の 4 つの選択肢すべて、または 3 つだけを選択できます。

   - **HTTP**。HTTP ヘッダーを検査するようにアプライアンスを設定します。
   - **SSL**。SSL クライアント証明書を検査するようにアプライアンスを設定します。前のドロップダウンリストで [REQ (requests)] を選択した場合にのみ使用できます。

- **TCP**。TCP ヘッダーを検査するようにアプライアンスを設定します。
- **IP**。送信元または宛先 IP アドレスを検査するようにアプライアンスを設定します。

10. [修飾子] ドロップダウンリストから式の修飾子を選択します。

    [Qualifier] ドロップダウンリストの内容は、選択したプロトコルによって異なります。次の表に、各プロトコルで使用できる選択肢を示します。

    表 1. 各プロトコルで使用できるキャッシュリダイレクションポリシー修飾子

プロトコル	修飾子	定義
HTTP	METHOD	リクエストで使用される HTTP メソッド。
-	URL	URL ヘッダーの内容。
-	URLTOKENS	HTTP ヘッダー内の URL トークン。
-	版	接続の HTTP バージョン。
-	HEADER	HTTP リクエストのヘッダー部分。
-	URLLEN	URL ヘッダーのコンテンツの長さ。
-	URLQUERY	URL ヘッダーの内容のクエリ部分。
-	URLQUERYLEN	URL ヘッダーのクエリ部分の長さ。
SSL	CLIENT.CERT	SSL クライアント証明書全体。
-	CLIENT.CERT.SUBJECT	クライアント証明書のサブジェクトフィールドの内容。
-	CLIENT.CERT.ISSUER	クライアント証明書の発行者。
-	CLIENT.CERT.SIGALGO	クライアント証明書で使用される署名アルゴリズム。
-	CLIENT.CERT.VERSION	クライアント証明書のバージョン。
-	CLIENT.CERT.VALIDFROM	クライアント証明書が有効になる日付。(開始日。)
-	CLIENT.CERT.VALIDTO	クライアント証明書が無効になる日付。(終了日。)
-	CLIENT.CERT.SERIALNUMBER	クライアント証明書のシリアル番号。
-	CLIENT.CIPHER.TYPE	クライアント証明書で使用される暗号化方式。
-	CLIENT.CIPHER.BITS	暗号化キーの有効ビット数。
-	CLIENT.SSL.VERSION	クライアント証明書の SSL バージョン。

プロトコル	修飾子	定義
TCP	SOURCEPORT	TCP 接続の送信元ポート。
-	DESTPORT	TCP 接続の宛先ポート。
-	MSS	TCP 接続の最大セグメントサイズ (MSS)。
IP	SOURCEIP	接続の送信元 IP アドレス。
-	DESTIP	接続の宛先 IP アドレス。

11. [演算子] ドロップダウンリストから式の演算子を選択します。

    選択内容は、前の手順で選択した修飾子によって異なります。このドロップダウンリストに表示できる演算子の完全なリストは次のとおりです。

    - == . 次のテキスト文字列と完全に一致します。
    - != . 次のテキスト文字列と一致しません。
      - . 次の整数より大きい。
    - CONTAINS . 次のテキスト文字列が含まれます。
    - CONTENTS . 指定されたヘッダー、URL、または URL クエリの内容。
    - EXISTS . 指定されたヘッダーまたはクエリが存在します。
    - NOTCONTAINS . 次のテキスト文字列は含まれません。
    - NOTEXISTS . 指定されたヘッダーまたはクエリは存在しません。

    このポリシーを特定のホストに送信されたリクエストに対して動作させる場合は、デフォルトの等号 (==) 記号のままにしておきます。

12. [値] テキストボックスが表示されている場合は、適切な文字列または数字をテキストボックスに入力します。

    たとえば、このポリシーでホスト shopping.example.com に送信されるリクエストを選択する場合は、その文字列を [値] テキストボックスに入力します。

13. 修飾子として HEADER を選択した場合は、[ヘッダー名] テキストボックスに目的のヘッダーを入力します。
14. 「**OK**」をクリックして式を「式」リストに追加します。
15. 手順 4 ～11 を繰り返して、さらに式を作成します。
16. [閉じる] をクリックして [式の追加] ダイアログボックスを閉じ、[キャッシュリダイレクトポリシーの作成] ダイアログボックスに戻ります。
17. 式の入力が完了したら、[作成] をクリックします。

キャッシュリダイレクトの構成

August 15, 2023

デプロイメントとネットワークトポロジに応じて、次のいずれかのタイプのキャッシュリダイレクトを構成できます。

- トランスペアレント。トランスペアレントキャッシュは、ネットワークバックボーン上のさまざまなポイントに配置できるため、配信ルート上のトラフィックを軽減できます。トランスペアレントモードでは、キャッシュリダイレクト仮想サーバーが NetScaler アプライアンスに流れるすべてのトラフィックをインターセプトし、キャッシュリダイレクトポリシーを適用して、コンテンツをキャッシュから配信するか、オリジンサーバーから配信するかを決定します。
- フォワードプロキシ。フォワードプロキシキャッシュサーバーは、企業 LAN の端にあり、WAN に面しています。転送プロキシモードでは、キャッシュリダイレクト仮想サーバーは DNS サーバーを使用して受信要求のホスト名を解決し、キャッシュ不可能なコンテンツのリクエストを解決されたオリジンサーバーに転送します。キャッシュ可能なリクエストは、設定されたキャッシュサーバーに送信されます。
- リバースプロキシ。リバースプロキシキャッシュは、特定のオリジンサーバー用に設定されます。リバースプロキシに転送される受信トラフィックは、キャッシュサーバーから処理することも、URL を変更してオリジンサーバーに送信することもできます。

透過的なリダイレクトを構成する

August 15, 2023

トランスペアレントキャッシュリダイレクトを構成すると、NetScaler アプライアンスは受信したすべてのトラフィックを評価して、キャッシュ可能かどうかを判断します。このモードは配信ルート上のトラフィックを軽減し、キャッシュサーバーが ISP または通信事業者のバックボーンにある場合によく使用されます。

デフォルトでは、キャッシュ可能なリクエストはキャッシュサーバーに送信され、キャッシュ不可能なリクエストはオリジンサーバーに送信されます。たとえば、NetScaler アプライアンスは Web サーバー宛のリクエストを受信すると、リクエスト内の HTTP ヘッダーを一連のポリシー表現と比較します。要求がポリシーと一致しない場合、アプライアンスは要求をキャッシュサーバーに転送します。要求がポリシーと一致する場合、アプライアンスは要求をそのまま Web サーバーに転送します。

このデフォルトの動作を変更する方法の詳細については、「オリジンではなくキャッシュへの直接ポリシーヒット」を参照してください。

トランスペアレントリダイレクションを設定するには、まずキャッシュリダイレクションとロードバランシングを有効にし、エッジモードを設定します。次に、ワイルドカード IP アドレス（*）を使用してキャッシュリダイレクト仮想サーバーを作成します。これにより、この仮想サーバーは、アプライアンスが所有する任意の IP アドレスでアプラ

イアンスへのトラフィックを受信できます。この仮想サーバーに、キャッシュしてはいけないリクエストの種類を説明するキャッシュリダイレクトポリシーをバインドします。次に、キャッシュ可能なリクエストのトラフィックをキャッシュリダイレクト仮想サーバーから受信する負荷分散仮想サーバーを作成します。最後に、物理キャッシュサーバーを表すサービスを作成し、それを負荷分散仮想サーバーにバインドします。

キャッシュリダイレクトと負荷分散を有効にする

August 15, 2023

アプライアンスのキャッシュリダイレクトおよび負荷分散機能は、デフォルトでは有効になっていません。キャッシュリダイレクション構成を有効にするには、これらを有効にする必要があります。

CLI を使用してキャッシュのリダイレクトと負荷分散を有効にする

コマンドプロンプトで次のコマンドを入力して、キャッシュのリダイレクトと負荷分散を有効にし、設定を確認します。

```
-  enable ns feature cr lb
-  show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature cr lb
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 3)     Load Balancing                  LB                   ON
 4)     Content Switching               CS                   ON
 5)     Cache Redirection               CR                   ON
            ...
            ...
            ...

 23)    appliance Push                  push                 OFF
 Done
<!--NeedCopy-->
```


GUI を使用したキャッシュのリダイレクトと負荷分散の有効化

1. ナビゲーションペインで、[システム] を展開し、[設定] をクリックします。
2. キャッシュリダイレクトを有効にするには、詳細ウィンドウの [モードと機能] で、[** 高度な機能の構成 **] をクリックします。

   a) ［拡張機能の構成］ダイアログボックスで、［キャッシュリダイレクト］の横にあるチェックボックスをオンにし、［**OK**］をクリックします。
   b) [機能の有効化/無効化] ダイアログボックスで、[はい] をクリックします。

3. 負荷分散を有効にするには、詳細ウィンドウの [モードと機能] で、[基本機能の構成] をクリックします。

   a) ［基本機能の構成］ダイアログボックスで、［負荷分散］の横にあるチェックボックスをオンにし、［**OK**］をクリックします。
   b) [機能の有効化/無効化] ダイアログボックスで、[はい] をクリックします。

エッジモードを構成する

August 15, 2023

NetScaler アプライアンスをネットワークのエッジに展開すると、そのネットワーク上のサーバーについて動的に学習します。エッジモードを使用すると、アプライアンスは最大 40,000 台の HTTP サーバーと、これらのサーバーのプロキシ TCP 接続について動的に学習できます。

このモードは、動的に学習されるサービスの統計情報の収集を有効にします。通常、キャッシュリダイレクト用のトランスペアレント展開で使用されます。

CLI を使用してエッジモードを有効にする

コマンドプロンプトで次のコマンドを入力してエッジモードを有効にし、設定を確認します。

```
-  enable ns mode Edge
-  show ns mode
<!--NeedCopy-->
```

例:

```
> enable ns mode edge
 Done

> show ns mode

        Mode                          Acronym             Status
        -------                       -------             ------
```



```
 8            ...
 9            ...
10            ...
11   6)     MAC-based forwarding           MBF                  ON
12   7)     Edge configuration             Edge                 ON
13   8)     Use Subnet IP                  USNIP                OFF
14            ...
15            ...
16            ...
17   16)    Bridge BPDUs                   BridgeBPDUs          OFF
18   Done
19  <!--NeedCopy-->
```

GUI を使用してエッジモードを有効にする

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. 詳細ペインの [Modes and Features] で [Configure modes] をクリックします。
3. 「モードの設定」ダイアログボックスで、「エッジ構成」の横にあるチェックボックスを選択し、「OK」をクリックします。
4. [機能の有効化/無効化] で? ダイアログボックスで、「はい」をクリックします。

キャッシュリダイレクト仮想サーバーを構成する

August 15, 2023

デフォルトでは、キャッシュリダイレクト仮想サーバーは、キャッシュ可能な要求を負荷分散仮想サーバーに転送してキャッシュし、キャッシュできない要求をオリジンサーバーに転送します（キャッシュできない要求が負荷分散仮想サーバーに送信されるリバースプロキシ構成を除く）。キャッシュリダイレクト仮想サーバーには、トランスペアレント、フォワードプロキシ、リバースプロキシの 3 種類があります。

トランスペアレントキャッシュリダイレクション仮想サーバーは、* の IP アドレスと、アプライアンスが表す任意の IP アドレスに送信される HTTP トラフィックを受け入れることができるポート番号（通常は 80）を使用します。そのため、構成できるトランスペアレントキャッシュリダイレクション仮想サーバーは 1 つだけです。構成する追加のキャッシュリダイレクト仮想サーバーは、フォワードプロキシまたはリバースプロキシリダイレクトサーバーである必要があります。

CLI を使用してキャッシュリダイレクト仮想サーバーをトランスペアレントモードで追加する

コマンドプロンプトで次のコマンドを入力して、キャッシュリダイレクト仮想サーバーを追加し、構成を確認します。


```
-  add cr vserver <name> <serviceType> [<IPAddress> <port> ] [-
    cacheType <cacheType>] [-redirect <redirect>]
-  show cr vserver [<name>]
<!--NeedCopy-->
```

例:

```
add cr vserver Vserver-CRD-1 HTTP * 80 -cacheType TRANSPARENT -redirect
    POLICY
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: RULE        Cache:
           TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
 Done
<!--NeedCopy-->
```

CLI を使用してキャッシュリダイレクト仮想サーバーを変更または削除する

- 仮想サーバーを変更するには、set cr vserver コマンドを使用します。これは add cr vserver コマンドを使用するのと同じですが、既存の仮想サーバーの名前を入力する点が異なります。
- 仮想サーバーを削除するには、\ <name\ > 引数のみを受け入れる`rm cr vserver`コマンドを使用します。

GUI を使用してキャッシュリダイレクト仮想サーバーをトランスペアレントモードで追加する

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. 「仮想サーバーの作成 (キャッシュリダイレクト)」ダイアログ・ボックスで、次のように次のパラメータの値を指定します。

   - 名前 * —名前
   - ポート *-ポート

   * 必須パラメータ
4. 「プロトコル」ドロップダウンリストで、サポートされているプロトコル (たとえば、**HTTP**) を選択します。仮想サーバーが選択したプロトコルの標準ポート以外のポートでトラフィックを受信する場合は、[ポート] フィールドに新しい値を入力します。

5. [詳細設定] タブをクリックします。
6. [キャッシュタイプ] が [トランスパレント] に設定され、[リダイレクト] が [ポリシー] に設定されていることを確認します。
7. [Create] をクリックしてから、[Close] をクリックします。キャッシュリダイレクト仮想サーバーペインには、新しい仮想サーバーが表示されます。
8. 新しいキャッシュリダイレクト仮想サーバーを選択すると、その構成の詳細が表示されます。

ポリシーをキャッシュリダイレクト仮想サーバーにバインドする

August 15, 2023

キャッシュリダイレクションポリシーは、キャッシュリダイレクション仮想サーバーに自動的にバインドされません。ポリシーベースのキャッシュリダイレクト仮想サーバーは、少なくとも 1 つのポリシーをバインドしない限り機能しません。

CLI を使用してポリシーをキャッシュリダイレクト仮想サーバーにバインドする

コマンドプロンプトで入力します。

```
-  bind cr vserver <name> -policyName <string>
-  show cr vserver [<name>]
<!--NeedCopy-->
```

例:

```
> bind cr vserver Vserver-CRD-1 -policyName bypass-cache-control
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-dynamic-url
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-urltokens
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-cookie
 Done

> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: RULE        Cache:
           TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
```



```
19
20 1)      Cache bypass  Policy: bypass-cache-control
21 2)      Cache bypass  Policy: bypass-dynamic-url
22 3)      Cache bypass  Policy: bypass-urltokens
23 4)      Cache bypass  Policy: bypass-cookie
24  Done
25 <!--NeedCopy-->
```

GUI を使用してユーザー定義のポリシーをキャッシュリダイレクト仮想サーバーにバインドする

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 構成する仮想サーバをクリックし、[開く] をクリックします。
3.「ポリシー」タブでポリシーの種類を選択し、「ポリシーの挿入」をクリックします。
4.「ポリシー名」列で、バインドするポリシーを選択します。
5.「OK」をクリックします。

キャッシュリダイレクト仮想サーバーからポリシーをバインド解除する

August 15, 2023

キャッシュリダイレクト仮想サーバーからポリシーをバインド解除すると、NetScaler アプライアンスはクライアント要求を評価するときにポリシーを適用しなくなります。

コマンド **CLI** を使用してキャッシュリダイレクト仮想サーバーからポリシーをバインド解除する

コマンドプロンプトで入力します。

```
1 -  unbind cr vserver <name> -policyName <string>
2 -  show cr vserver [<name>]
3 <!--NeedCopy-->
```

例:

```
1  unbind cr vserver Vserver-CR-1 -policyName bypass-non-get
2  > show cr vserver Vserver-CRD-1
3          Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
4          State: UP  ARP:DISABLED
5          Client Idle Timeout: 180 sec
6          Down state flush: ENABLED
7          Disable Primary Vserver On Down : DISABLED
8          Default:        Content Precedence: RULE        Cache:
               TRANSPARENT
9          On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
10         Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
```



```
11
12 1)      Cache bypass  Policy: bypass-cache-control
13  Done
14 <!--NeedCopy-->
```

GUI を使用してユーザー定義ポリシーをキャッシュリダイレクト仮想サーバーからバインド解除する

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 構成する仮想サーバーをクリックし、[開く] をクリックします。
3. 「ポリシー」タブの「ポリシー名」で、バインド解除するポリシーを選択します。
4. 「ポリシーをバインド解除」をクリックし、「OK」をクリックします。

負荷分散仮想サーバーを作成する

August 15, 2023

NetScaler アプライアンス上のキャッシュリダイレクト仮想サーバーは、リクエストがキャッシュ可能な場合はキャッシュサーバーファームに、リクエストがキャッシュ不可能な場合はオリジンサーバーファームにリクエストを送信できます。

各キャッシュ・サーバーは、サービスによってアプライアンス上で表されます。サービスとは、キャッシュ・リダイレクト仮想サーバーから要求を受け取り、その要求をサーバーに転送する負荷分散仮想サーバーにバインドされます。

負荷分散仮想サーバーおよびその他の構成オプションの設定の詳細については、負荷分散を参照してください。

CLI を使用した負荷分散仮想サーバーの作成

コマンドプロンプトで次のコマンドを入力して負荷分散仮想サーバーを作成し、構成を確認します。

```
1 -  add lb vserver <name> <serviceType> [<IPAddress>] [<port>]
2 -  show lb vserver [<name>]
3 <!--NeedCopy-->
```

例:

```
1 > add lb vserver Vserver-LB-CR HTTP 10.102.20.30 80
2  Done
3 > show lb vserver Vserver-LB-CR
4         Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
5         State: DOWN
6         Last state change was at Fri Jul  2 08:47:52 2010
7         Time since last state change: 0 days, 00:00:08.470
```



```
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
 Done
<!--NeedCopy-->
```

GUI を使用して負荷分散仮想サーバーを作成する

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ペインで、［Add］をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次のように次のパラメータの値を指定します。

   - 名前 *-名前
   - IP アドレス *-IP アドレス
   - ポート *-ポート

   * 必須パラメータ

4. プロトコルリストで、サポートされているプロトコル (**HTTP** など) を選択します。仮想サーバーが、選択したプロトコルの既知のポート以外のポートでトラフィックを受信する場合は、ポートフィールドに新しい値を入力します。
5. ［Create］をクリックしてから、［Close］をクリックします。負荷分散仮想サーバーペインには、新しい仮想サーバーが表示されます。

HTTP サービスを構成する

August 16, 2023

NetScaler アプライアンスでは、サービスはネットワーク上の物理サーバーを表します。トランスペアレントキャッシュリダイレクト設定では、サービスはキャッシュサーバーを表します。キャッシュ可能な要求は、キャッシュリダイレクト仮想サーバーから負荷分散仮想サーバーに送信され、負荷分散仮想サーバーは各要求を適切なサービスに転送し、そこでキャッシュサーバーに渡されます。

CLI を使用して HTTP サービスを設定する

コマンドプロンプトで次のコマンドを入力して HTTP サービスを作成し、構成を確認します。

```
-  add service <name> <IP> <serviceType> <port> -cacheType <cacheType>
-  show service [<name>]
<!--NeedCopy-->
```

例:

```
> add service Service-HTTP-1 10.102.29.40 HTTP 80 -cacheType
   TRANSPARENT
 Done
> show service Service-HTTP-1
        Service-HTTP-1 (10.102.29.40:80) - HTTP
        State: DOWN
        Last state change was at Fri Jul  2 09:14:17 2010
        Time since last state change: 0 days, 00:00:13.820
        Server Name: 10.102.29.40
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): YES
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cache Type: TRANSPARENT Redirect Mode:
        Cacheable: NO
        SC: OFF
        SP: ON
        Down state flush: ENABLED

1)      Monitor Name: tcp-default
                State: DOWN     Weight: 1
                Probes: 3       Failed [Total: 3 Current: 3]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: N/A
 Done
<!--NeedCopy-->
```

CLI を使用してサービスを変更または削除する

- サービスを変更するには、set service コマンドを使用します。これは add service コマンドと同様ですが、既存のサービスの名前を入力する点が異なります。
- サービスを削除するには、<name> 引数のみを受け入れる`rm service`コマンドを使用します。

GUI を使用して HTTP サービスを追加します

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します
2. 詳細ペインで、[Add] をクリックします。
3. [Create Service] ダイアログボックスで、次に示すように次のパラメータの値を指定します。

   - サービス名 *—名前
   - サーバー *—IP
   - ポート *-ポート

   * 必須パラメータ

4. 「プロトコル *」ドロップダウンリストで、サポートされているプロトコル (**HTTP** など) を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。

負荷分散仮想サーバーに対するサービスのバインド/バインド解除

August 15, 2023

サービスを負荷分散仮想サーバーにバインドする必要があります。これにより、ロードバランサーはサービスが表すサーバーにリクエストを転送できます。構成が変更された場合は、負荷分散仮想サーバーからサービスをアンバインドできます。

CLI を使用してサービスを負荷分散仮想サーバーにバインドする

コマンドプロンプトで入力します。

```
-  bind lb vserver <name> <serviceName>
-  show lb vserver [<name>]
<!--NeedCopy-->
```

例:

```
> bind lb vserver vserver-LB-CR service-HTTP-1
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:42:25.610
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
```



```
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
<!--NeedCopy-->
```

CLI を使用して負荷分散仮想サーバーからサービスをバインド解除する

サービスのバインドを解除するには、`bind lb vserver`の代わりに`unbind lb vserver`コマンドを使用します。

GUI を使用して負荷分散仮想サーバーからサービスをバインド/バインド解除する

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します
2. 詳細ウィンドウで、サービスをバインドまたはバインド解除する仮想サーバーを選択し、[開く] をクリックします。
3. [サービス] タブの [アクティブ] 列で、[サービス名] の横にあるチェックボックスをオンまたはオフにします。
4. 「OK」をクリックします。

透過型キャッシュのプロキシポート設定の使用を無効にする

August 15, 2023

NetScaler アプライアンスで構成されたキャッシュサービスでソース IP（USIP）の使用オプションが無効になっている場合、アプライアンスは、アプライアンスが所有するサブネット IP（SNIP）アドレスまたはマップ IP（MIP）アドレスをソース IP アドレスとして使用し、ランダムポートをソースポートとして使用して、クライアント要求をキャッシュサービスに転送します。ランダムに選択されたポートはプロキシポートと呼ばれます。

ただし、完全に透過的なキャッシュ (キャッシュサービスがクライアントの IP アドレスとポート番号を受け取るキャッシュ構成) を設定する場合は、USIP オプションをグローバルまたはキャッシュサービスで有効にするだけでなく、グローバルまたはキャッシュサービスで Use Proxy Port 設定も無効にする必要があります。Use Proxy Port 設定を無効にすると、アプライアンスはキャッシュサービスに接続するときにクライアントのソースポートをソースポートとして使用でき、完全に透過的なキャッシュ構成が保証されます。

グローバルまたはサービスで [プロキシポートを使用] オプションの設定の詳細については、サーバー側接続の送信元ポートの構成を参照してください。

NetScaler アプライアンスにポート範囲を割り当てる

August 15, 2023

クライアント IP アドレスを共有すると競合が発生し、ルーター、キャッシュサーバー、オリジンサーバー、その他の NetScaler アプライアンスなどのネットワークデバイスが、応答の送信先となるアプライアンス、ひいてはクライアントを特定できなくなる可能性があります。

この問題を解決する方法は、NetScaler アプライアンスにソースポート範囲を割り当てることです。この割り当てにより、ネットワークデバイスは要求を送信した NetScaler アプライアンスを明確に識別できます。

CLI を使用してソースポート範囲を NetScaler アプライアンスに割り当てる

コマンドプロンプトで入力します。

```
set ns param -crPortRange <startPortNumber-endPortNumber>
```

アプライアンス GUI を使用してソースポート範囲を NetScaler アプライアンスに割り当てる

1. ナビゲーションペインで、[システム] をクリックし、[設定] をクリックします。
2. 「設定」グループで、「グローバルシステム設定の変更」リンクをクリックします。
3. 「キャッシュリダイレクトポート範囲」グループで、「開始ポート」にポート番号、「終了ポート」にポート番号を入力して、アプライアンスのポート範囲を指定します。
4. 「OK」をクリックします。

負荷分散仮想サーバーを有効にして、要求をキャッシュにリダイレクトする

August 15, 2023

負荷分散仮想サーバーが特定の IP アドレスとポートの組み合わせで受信するように構成されている場合、そのアドレスとポートの組み合わせ宛ての要求については、キャッシュリダイレクト仮想サーバーよりも優先されます。したがって、キャッシュリダイレクト仮想サーバーはこれらの要求を処理しません。

この機能を無効にして、リクエストをキャッシュから処理するかどうかをキャッシュリダイレクト仮想サーバーに決定させたい場合は、特定の負荷分散仮想サーバーをキャッシュ可能に構成します。

このような構成は通常、ISP がネットワークのエッジで NetScaler アプライアンスを使用し、すべてのトラフィックがアプライアンスを経由する場合に使用されます。

CLI を使用して負荷分散仮想サーバーが要求をキャッシュにリダイレクトできるようにします

コマンドプロンプトで入力します。

```
-  set lb vserver <name> [-cacheable ( YES | NO)]
-  show lb vserver [<name>]
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-CR  - cacheable YES
> show lb vserver vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 01:05:51.510
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Cacheable: YES  PQ: OFF SC: OFF
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
<!--NeedCopy-->
```

トランスペアレントキャッシュリダイレクトの場合、アプライアンスはすべてのトラフィックをインターセプトし、すべてのリクエストを評価してキャッシュ可能かどうかを判断します。キャッシュできないリクエストは、変更されずにオリジンサーバーに送信されます。

トランスペアレントキャッシュリダイレクトを使用する場合、常にトラフィックをオリジンサーバーに転送する仮想サーバーの負荷分散のためにキャッシュリダイレクトをオフにしたい場合があります。

CLI を使用して負荷分散仮想サーバーのキャッシュをオフにする

負荷分散仮想のキャッシュをオフにするには、set lb vserver の代わりに unset lb vserver コマンドを使用してください。キャッシュ可能なパラメータには NO の値を指定します。

GUI を使用して要求をキャッシュにリダイレクトする負荷分散仮想サーバーを有効または無効にする

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ペインで、キャッシュを有効または無効にする仮想サーバーを選択し、[開く] をクリックします。
3.「詳細設定」タブで、「キャッシュリダイレクト」チェックボックスをオンまたはオフにします。
4.「OK」をクリックします。

フォワードプロキシリダイレクトを構成する

August 15, 2023

フォワードプロキシは、クライアントまたはクライアントグループの単一窓口です。この構成では、NetScaler アプライアンスはキャッシュ不可能な要求をオリジンサーバーにリダイレクトし、キャッシュ可能な要求をフォワードプロキシキャッシュまたはトランスペアレントキャッシュのいずれかにリダイレクトします。

アプライアンスが転送プロキシとして構成されている場合、ユーザーはブラウザーが送信先サーバーではなく転送プロキシに要求を送信するようにブラウザーを変更する必要があります。

アプライアンス上のフォワードプロキシキャッシュリダイレクト仮想サーバーは、要求をキャッシュのポリシーと比較します。リクエストがキャッシュできない場合、アプライアンスは DNS 負荷分散仮想サーバーに宛先の解決を問い合わせてから、リクエストをオリジンサーバーに送信します。要求がキャッシュ可能な場合、アプライアンスは要求をキャッシュ用の負荷分散仮想サーバーに転送します。

アプライアンスは、リクエストのホストヘッダーにあるホストドメイン名または IP アドレスを使用して、リクエストされた宛先を決定します。リクエストに HOST ヘッダーがない場合、アプライアンスはリクエストの宛先 IP アドレスに基づいて HOST ヘッダーを挿入します。

通常、NetScaler アプライアンスは企業 LAN の転送プロキシとして機能します。このような構成では、アプライアンスは企業 LAN のエッジに配置され、WAN に送られる前にクライアントのリクエストをインターセプトします。アプライアンスをフォワードプロキシモードに設定すると、WAN 上のトラフィックが減少します。

フォワードプロキシキャッシュリダイレクトを設定するには、まずアプライアンスで負荷分散とキャッシュリダイレクトを有効にします。次に、DNS 負荷分散仮想サーバーと関連サービスを構成します。また、負荷分散仮想サーバーを構成し、それにキャッシュ用の適切なサービスをバインドします。フォワードプロキシキャッシュリダイレクト仮想サーバーを構成し、DNS と負荷分散仮想サーバーをそれにバインドします。また、キャッシュポリシーを設定し、キャッシュリダイレクト仮想サーバーにバインドする必要があります。セットアップを完了するには、フォワードプロキシを使用するようにクライアントブラウザを設定します。

アプライアンスでキャッシュリダイレクトと負荷分散を有効にする方法の詳細については、「キャッシュのリダイレクトとロードバランシングの有効化」を参照してください。

負荷分散仮想サーバーの作成方法の詳細については、「負荷分散仮想サーバーを作成する」を参照してください。

キャッシュサーバーを表すサービスを構成する方法の詳細については、「HTTP サービスの構成」を参照してください。

サービスを仮想サーバーにバインドする方法の詳細については、「負荷分散仮想サーバーへのサービスのバインド/バインド解除」を参照してください。

フォワードプロキシキャッシュリダイレクトサーバーの作成方法の詳細については、「キャッシュリダイレクト仮想サーバーの構成」および「TRANSTARENT」または「FORWARD」タイプの仮想サーバーの作成を参照してください。

キャッシュリダイレクトポリシーをキャッシュリダイレクト仮想サーバーにバインドする方法については、「キャッシュリダイレクトポリシーを構成する」を参照してください。

DNS サービスを作成する

August 15, 2023

DNS サービスは、NetScaler アプライアンス上では、ネットワーク内の物理 DNS サーバーを表すものです。DNS 負荷分散仮想サーバーは、このようなサービスを介してネットワーク内の DNS サーバーに DNS 要求を送信します。

CLI を使用して DNS サービスを作成する

コマンドラインで次のコマンドを入力して DNS サービスを作成し、構成を確認します。

```
-  add service <name> <IP> <serviceType> <port>
-  show service [<name>]
<!--NeedCopy-->
```

例:

```
add service Service-DNS-1 10.102.29.41 DNS 53
show service Service-DNS-1
        Service-DNS-1 (10.102.29.41:53) - DNS
        State: DOWN
        Last state change was at Fri Jul  2 10:14:32 2010
        Time since last state change: 0 days, 00:00:13.550
        Server Name: 10.102.29.41
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
```



```
17          Cacheable: NO
18          SC: OFF
19          SP: OFF
20          Down state flush: ENABLED
21
22  1)      Monitor Name: ping-default
23                  State: DOWN     Weight: 1
24                  Probes: 3       Failed [Total: 3 Current: 3]
25                  Last response: Failure - Probe timed out.
26                  Response Time: 2000.0 millisec
27   Done
28  <!--NeedCopy-->
```

GUI を使用して DNS サービスを追加する

1. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
2. 詳細ペインで、［Add］をクリックします。
3. [Create Service] ダイアログボックスで、次に示すように次のパラメータの値を指定します。
   - サービス名 *—名前
   - サーバー *—IP
   - ポート *-ポート

* 必須パラメータ

1. Protocol* ドロップダウンリストで、サポートされているプロトコル (**DNS** など) を選択します。
2. ［Create］をクリックしてから、［Close］をクリックします。

DNS 負荷分散仮想サーバーを作成する

August 15, 2023

DNS 仮想サーバーにより、フォワードプロキシはクライアント要求をオリジンサーバーに転送する前に DNS 解決を実行できます。DNS 負荷分散仮想サーバーは、ネットワーク上の物理 DNS サーバーを表す DNS サービスに関連付けられています。

CLI を使用して DNS 負荷分散仮想サーバーを作成する

コマンドラインで次のコマンドを入力して DNS 負荷分散仮想サーバーを作成し、構成を確認します。


```
-  add lb vserver <name> <serviceType>
-  show lb vserver [<name>]
<!--NeedCopy-->
```

例:

```
> add lb vserver Vserver-DNS-1 DNS
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:00:08.10
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
 Done
<!--NeedCopy-->
```

GUI を使用して DNS 負荷分散仮想サーバーを作成する

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、[Add] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスの [名前] ボックスに、仮想サーバーの名前を入力します。
4. Protocol* ドロップダウンリストで、サポートされているプロトコル (**DNS** など) を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。DNS 仮想サーバーペインには、新しい仮想サーバーが表示されます。

DNS サービスを仮想サーバーにバインドする

August 15, 2023

DNS サーバーが DNS 要求に応答するには、DNS サーバーを表すサービスが DNS 仮想サーバーにバインドされている必要があります。

CLI を使用して DNS サービスを負荷分散仮想サーバーにバインドします

コマンドプロンプトで次のコマンドを入力して DNS サービスを負荷分散仮想サーバーにバインドし、構成を確認します。

```
-  bind lb vserver <name> <serviceName>
-  show lb vserver <name>
<!--NeedCopy-->
```

例:

```
> bind lb vserver Vserver-DNS-1 Service-DNS-1
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:12:16.80
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE

1) Service-DNS-1 (10.102.29.41: 53) - DNS State: DOWN   Weight: 1
 Done
>
<!--NeedCopy-->
```

CLI を使用して DNS サービスを負荷分散仮想サーバーからバインド解除する

`bind lb vserver`の代わりに`unbind lb vserver`コマンドを使用します。

GUI を使用して DNS サービスを負荷分散仮想サーバーにバインド/バインド解除する

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します
2. 詳細ウィンドウで、DNS サービスをバインドまたはバインド解除する仮想サーバーを選択し、[開く] をクリックします。
3. [サービス] タブの [アクティブ] 列で、[サービス名] の横にあるチェックボックスをオンまたはオフにします。
4. 「OK」をクリックします。

フォワードプロキシの使用をクライアントの **Web** ブラウザーで構成する

August 15, 2023

NetScaler アプライアンスをネットワーク内の転送プロキシキャッシュリダイレクト仮想サーバーとして構成する場合、転送プロキシに要求を送信するようにクライアントの Web ブラウザーを構成する必要があります。通常、転送プロキシを使用する場合、ネットワーク内のサーバーへの唯一のルートは転送プロキシを経由することです。

ブラウザがフォワードプロキシを使用するように設定するには、ブラウザのマニュアルを参照してください。この構成のフォワードプロキシキャッシュリダイレクト仮想サーバーの IP アドレスとポート番号を指定します。

リバースプロキシリダイレクトを構成する

August 15, 2023

リバースプロキシは 1 つ以上の Web サーバーの前に配置され、オリジンサーバーをクライアントのリクエストから保護します。多くの場合、リバースプロキシキャッシュは、サーバーへのすべてのクライアントリクエストのフロントエンドです。管理者は、リバース・プロキシ・キャッシュを特定のオリジナル・サーバーに割り当てます。リバースプロキシキャッシュは、任意のオリジンサーバーへのすべてのリクエストに対して頻繁にリクエストされたコンテンツをキャッシュする透過プロキシキャッシュとフォワードプロキシキャッシュとは異なり、サーバーの選択は要求に基づいて行われます。

トランスペアレントプロキシキャッシュとは異なり、リバースプロキシキャッシュは独自の IP アドレスを持っており、キャッシュ不可能な要求内の宛先ドメインと URL を新しい宛先ドメインおよび URL に置き換えることができます。

リバースプロキシキャッシュリダイレクトは、オリジンサーバー側またはネットワークのエッジに展開できます。オリジンサーバーにデプロイすると、リバースプロキシキャッシュリダイレクト仮想サーバーは、オリジンサーバーへのすべてのリクエストのフロントエンドになります。

リバースプロキシモードでは、アプライアンスが要求を受信すると、キャッシュリダイレクト仮想サーバーが要求を評価し、キャッシュ用の負荷分散仮想サーバーまたはオリジンの負荷分散仮想サーバーのいずれかに転送します。受信したリクエストは、バックエンドサーバーに送信される前にホストヘッダーまたはホスト URL を変更することで変換できます。

リバースプロキシキャッシュリダイレクトを構成するには、最初にキャッシュリダイレクトとロードバランシングを有効にします。次に、負荷分散仮想サーバーとサービスを構成して、キャッシュ可能な要求をキャッシュサーバーに送信します。また、オリジンサーバーの負荷分散仮想サーバーと関連サービスを設定します。次に、リバースプロキシキャッシュリダイレクト仮想サーバーを構成し、関連するキャッシュリダイレクトポリシーをその仮想サーバーにバインドします。最後に、マッピングポリシーを構成し、リバースプロキシキャッシュリダイレクト仮想サーバーにバインドします。

マッピングポリシーには、キャッシュリダイレクト仮想サーバーがキャッシュできない要求をオリジンの負荷分散仮想サーバーに転送できるようにするアクションが関連付けられています。

必ずデフォルトのキャッシュサーバーの宛先を作成してください。

アプライアンスでキャッシュリダイレクトと負荷分散を有効にする方法の詳細については、「キャッシュのリダイレクトとロードバランシングの有効化」を参照してください。

負荷分散仮想サーバーの作成方法の詳細については、「負荷分散仮想サーバーを作成する」を参照してください。

キャッシュサーバーを表すサービスを構成する方法の詳細については、「HTTP サービスの構成」を参照してください。

サービスを仮想サーバーにバインドする方法の詳細については、「負荷分散仮想サーバーへのサービスのバインド/バインド解除」を参照してください。

リバースプロキシキャッシュリダイレクトサーバーの作成方法の詳細については、「キャッシュリダイレクト仮想サーバーの構成」および「REVERSE」タイプの仮想サーバーの作成を参照してください。

組み込みのキャッシュリダイレクトポリシーをキャッシュリダイレクト仮想サーバーにバインドする方法については、キャッシュリダイレクト仮想サーバーへのポリシーのバインドを参照してください。

マッピングポリシーの構成

受信リクエストがキャッシュできない場合、リバースプロキシキャッシュリダイレクト仮想サーバーは、リクエスト内のドメインと URL をターゲットオリジンサーバーのドメインと URL に置き換え、リクエストをオリジンの負荷分散仮想サーバーに転送します。

マッピングポリシーにより、リバースプロキシキャッシュリダイレクト仮想サーバーは、宛先ドメインと URL を置き換え、要求をオリジンの負荷分散仮想サーバーに転送できます。

マッピングポリシーは、最初にドメインと URL を変換してから、要求をオリジン負荷分散仮想サーバーに渡す必要があります。

マッピングポリシーでは、ドメイン、URL プレフィックス、URL サフィックスを次のようにマッピングできます。

- ドメインマッピング: プレフィックスまたはサフィックスを付けずにドメインをマップできます。ドメインマッピングは仮想サーバーのデフォルトのマッピングです (たとえば、www.mycompany.com を www.myrealcompany.com にマッピングします)。
- プレフィックスマッピング:URL の一部としてプレフィックスが付いた特定のパターンを置き換えることができます (たとえば、www.mycompany.com/sports/index.html を www.mycompany.com/news/index.html にマッピングするなど)。
- サフィックスマッピング:URL のファイルサフィックスを置き換えることができます (たとえば、www.mycompany.com/sports/index.html を www.mycompany.com/sports/index.asp にマッピングするなど)。

マッピングされるソース文字列と宛先文字列は類似している必要があります。ソースドメインを指定する場合は宛先ドメインを指定し、ソースサフィックスを指定する場合は宛先サフィックスを指定する必要があります。同様に、ソースからの正確な URL を指定する場合、ターゲット URL も正確な URL でなければなりません。

リバースプロキシモードのマッピングポリシーを設定したら、それらをキャッシュリダイレクト仮想サーバーにバインドする必要があります。

ソース URL、ターゲット URL、およびソースドメインとターゲットドメインを組み合わせて、3 種類のドメインマッピングをすべて構成できます。

CLI を使用してリバースプロキシモードのマッピングポリシーを設定する

コマンドプロンプトで、次のコマンドを入力してポリシーマップを追加し、設定を確認します。

```
1 -  add policy map <mapPolicyName> -sd <string> [-su <string>] [-td <
     string>] [-tu <string>]
2 -  show policy map [<mapPolicyName>]
3 <!--NeedCopy-->
```

例**:**

次のコマンドは、クライアントリクエスト内のドメインをターゲットドメインにマップします。

```
1 > add policy map myMappingPolicy -sd www.mycompany.com -td www.
     myrealcompany.com
2  Done
3 > show policy map myMappingPolicy
4 1)      Name: myMappingPolicy
5         Source Domain: www.mycompany.com        Source Url:
6         Target Domain: www.myrealcompany.com    Target Url:
7  Done
8 <!--NeedCopy-->
```

URL サフィックスを別の URL サフィックスにマッピングする例を次に示します。

```
1 > add policy map myOtherMappingPolicy -sd www.mycompany.com -td www.
     myrealcompany.com -su /news.html -tu /realnews.html
2  Done
3 > show policy map myOtherMappingPolicy
4 1)      Name: myOtherMappingPolicy
5         Source Domain: www.mycompany.com        Source Url: /news.html
6         Target Domain: www.myrealcompany.com    Target Url: /realnews.
             html
7  Done
8 <!--NeedCopy-->
```

GUI を使用してリバースプロキシモードのマッピングポリシーを設定する

1. [トラフィック管理] **>** [キャッシュリダイレクト] **>** [マップポリシー] に移動します。

2. 詳細ペインで、[Add] をクリックします。
3. [マップポリシーの作成（Create Map Policy）] ダイアログボックスで、次に示すように次のパラメータの値を指定します。

   - 名前 *-mapPolicyName
   - ソースドメイン *-sd
   - ターゲットドメイン *-td
   - ソース URL-su
   - ターゲット URL-tu

   * 必須パラメータ
4. [Create] をクリックしてから、[Close] をクリックします。マップペインには、新しいマッピングポリシーが表示されます。

CLI を使用してマッピングポリシーをキャッシュリダイレクト仮想サーバーにバインドします

コマンドプロンプトで次のコマンドを入力して、マッピングポリシーをキャッシュリダイレクト仮想サーバーにバインドし、構成を確認します。

```
-  bind cr vserver <name> -policyName <string> [<targetVserver>]
-  show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> bind cr vserver Vserver-CRD-3 -policyName myMappingPolicy Vserver-LB-
   CR
 Done
> show cr vserver Vserver-CRD-3
        Vserver-CRD-3 (10.102.29.50:88) - HTTP  Type: CONTENT
        State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default: Vserver-LB-CR  Content Precedence: RULE        Cache:
           REVERSE
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Policy:         Target: Vserver-LB-CR   Priority: 0     Hits: 0
1)      Map: myMappingPolicy Target: Vserver-LB-CR
 Done
<!--NeedCopy-->
```


GUI を使用してマッピングポリシーをキャッシュリダイレクト仮想サーバーにバインドします

1. [トラフィック管理**] > [**キャッシュリダイレクト**] > [**仮想サーバー] に移動します。
2. 詳細ペインで、マッピングポリシーをバインドする仮想サーバーを選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (キャッシュリダイレクト)] の [ポリシー] タブで、[マップ] を選択し、[ポリシーの挿入] をクリックします。
4. [ポリシー名（Policy Name）] 列で、ドロップダウンリストからポリシーを選択します。
5. [ターゲット] 列で下矢印をクリックし、ドロップダウンリストから仮想サーバーを選択します。
6. [**OK**] をクリックします。

選択的なキャッシュリダイレクト

August 15, 2023

選択的キャッシュリダイレクトは、画像などの特定の種類のコンテンツのリクエストを 1 つのキャッシュサーバーまたはキャッシュサーバーのグループに送信し、他の種類のコンテンツを別のキャッシュサーバーまたはキャッシュサーバーのグループに送信します。トランスペアレントモード、リバースプロキシモード、またはフォワードプロキシモードで高度なキャッシュリダイレクトを設定できます。

選択的キャッシュリダイレクトでは、NetScaler アプライアンスがクライアント要求をインターセプトし、キャッシュ不可能な要求をクライアント要求の元の宛先に転送します。キャッシュ可能なリクエストの場合、アプライアンスは特定のコンテンツタイプのコンテンツを提供できる宛先キャッシュサーバーにリクエストを送信します。

選択的キャッシュリダイレクトでは、キャッシュリダイレクトポリシーに加えてコンテンツスイッチングポリシーを構成する必要があります。アプライアンスはまず、キャッシュリダイレクション仮想サーバーにバインドされているキャッシュリダイレクションポリシーを評価します。リクエストがキャッシュリダイレクションポリシーと一致する場合、キャッシュリダイレクション仮想サーバーはリクエストをオリジンサーバーまたはオリジンの負荷分散仮想サーバーに送信します。要求に一致するキャッシュリダイレクションポリシーがない場合、アプライアンスはキャッシュリダイレクト仮想サーバーにバインドされたコンテンツスイッチングポリシーを評価します。コンテンツスイッチングポリシーが要求と一致する場合、キャッシュリダイレクト仮想サーバーは要求をキャッシュの負荷分散仮想サーバーにリダイレクトします。

選択的キャッシュリダイレクトを構成するには、まず NetScaler アプライアンスでキャッシュリダイレクト、負荷分散、コンテンツスイッチングを有効にします。次に、キャッシュと関連する HTTP サービスの負荷分散仮想サーバーを構成します。その後、キャッシュリダイレクト仮想サーバーを構成し、キャッシュリダイレクトポリシーとコンテンツスイッチングポリシーの両方をそれにバインドします。ポリシーをバインドしたら、ルールベースまたは URL ベースのコンテンツスイッチングポリシーを優先するように仮想サーバーを構成できます。

エッジ展開トポロジでトランスペアレントモードのキャッシュリダイレクトを設定すると、アプライアンスはキャッシュ可能なすべての HTTP トラフィックをトランスペアレントキャッシュファームに送信します。クライアントは、

ポート 80 でトラフィックを受信するレイヤ 4 スイッチとして構成されたアプライアンスを介してインターネットにアクセスします。

アプライアンスは、画像（.png ファイルや.jpg ファイルなど）のリクエストをトランスペアレントキャッシュファーム内の 1 つのサーバーに送信し、静的コンテンツに対する他のすべてのリクエストをファーム内の他のサーバーに送信できます。この構成では、イメージをイメージキャッシュに送信し、その他すべてのキャッシュ可能なコンテンツをデフォルトキャッシュに送信するようにコンテンツスイッチングポリシーを設定します。

注: ここで説明する設定は、透過的な選択的キャッシュリダイレクト用です。したがって、リバースプロキシ構成のように、オリジンに負荷分散仮想サーバーは必要ありません。

このタイプの選択的キャッシュリダイレクトを設定するには、まずキャッシュリダイレクト、負荷分散、およびコンテンツスイッチングを有効にします。次に、キャッシュ用の負荷分散仮想サーバーを構成し、関連する HTTP サービスを設定します。次に、キャッシュリダイレクト仮想サーバーを構成し、キャッシュリダイレクトとコンテンツスイッチングポリシーの両方を作成し、この仮想サーバーにバインドします。

アプライアンスでキャッシュリダイレクトと負荷分散を有効にする方法の詳細については、「キャッシュのリダイレクトとロードバランシングの有効化」を参照してください。

コンテンツスイッチを有効にする

August 15, 2023

選択的キャッシュリダイレクトを構成するには、アプライアンスで負荷分散機能とキャッシュリダイレクト機能の両方を有効にした後に、コンテンツスイッチングを有効にする必要があります。

CLI を使用したコンテンツスイッチングの有効化

コマンドプロンプトで入力します。

```
-  enable ns feature CS

-  show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature cs
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  ON
 2)     Surge Protection                SP                  ON
```



```
 3)     Load Balancing                 LB                   ON
 4)     Content Switching              CS                   ON
 5)     Cache Redirection              CR                   ON
          ...
          ...
          ...
 23)    appliance Push                 push                 OFF
 Done
<!--NeedCopy-->
```

GUI を使用したキャッシュのリダイレクトと負荷分散の有効化

1. ナビゲーションペインで、[システム] を展開し、[設定] をクリックします。
2. 詳細ウィンドウの [モードと機能] で、[基本機能の構成] をクリックします。
3. [基本機能の構成] ダイアログボックスで、[コンテンツの切り替え] の横にあるチェックボックスをオンにし、[**OK**] をクリックします。
4. [機能の有効化/無効化] ダイアログボックスで、[はい] をクリックします。

キャッシュの負荷分散仮想サーバーを構成する

August 15, 2023

使用するキャッシュサーバーのタイプごとに、負荷分散仮想サーバーと HTTP サービスを作成します。たとえば、あるキャッシュサーバーの JPEG ファイルと別のキャッシュサーバーの GIF ファイルを提供し、残りのコンテンツには 3 つ目のキャッシュサーバーを使用する場合は、3 種類のキャッシュサーバーそれぞれに HTTP サービスと仮想サーバーを作成します。次に、各サービスをそれぞれの仮想サーバーにバインドします。

負荷分散仮想サーバーの作成方法の詳細については、「負荷分散仮想サーバーを作成する」を参照してください。

キャッシュサーバーを表すサービスを構成する方法の詳細については、「HTTP サービスの構成」を参照してください。

サービスを仮想サーバーにバインドする方法の詳細については、「負荷分散仮想サーバーへのサービスのバインド/バインド解除」を参照してください。

透過プロキシキャッシュリダイレクトサーバーの作成方法の詳細については、「キャッシュリダイレクト仮想サーバーの構成」および「TRANSTARENT」タイプの仮想サーバーの作成を参照してください。

組み込みのキャッシュリダイレクトポリシーをキャッシュリダイレクト仮想サーバーにバインドする方法については、キャッシュリダイレクト仮想サーバーへのポリシーのバインドを参照してください。

特定の種類のコンテンツに対してキャッシュリダイレクトポリシーを構成する

.png または.jpeg 拡張子を含むリクエストをキャッシュ可能として識別するには、キャッシュリダイレクションポリシーを構成し、それをキャッシュリダイレクト仮想サーバーにバインドします。

注：リクエストがポリシーと一致する場合、NetScaler アプライアンスはリクエストをオリジンサーバーに転送します。その結果、次の手順では、拡張子が「.png」または「.jpeg」
でない 要求を照合するようにポリシーを設定します。

特定のタイプのコンテンツに対してキャッシュリダイレクトを構成するには、「キャッシュリダイレクトポリシーを構成する」の説明に従って、単純な式を使用するポリシーを構成します。

コンテンツスイッチのポリシーを構成する

August 15, 2023

コンテンツスイッチングポリシーを作成して、あるサーバーまたはファームに送信する特定の種類のコンテンツを識別し、別のキャッシュサーバーまたはファームから配信するその他の種類のコンテンツを識別する必要があります。たとえば、拡張子が.png および.jpeg のイメージファイルの場所を決定するポリシーを設定できます。

コンテンツスイッチングポリシーを作成する前に、選択する負荷分散仮想サーバーを記述するコンテンツスイッチングアクションを定義する必要があります。このアクションは、コンテンツスイッチングポリシーで使用されます。

コンテンツスイッチングポリシーを定義したら、コンテンツスイッチング仮想サーバーにバインドし、負荷分散仮想サーバーを指定します。ポリシーに一致する要求は、指定された負荷分散仮想サーバーに転送されます。コンテンツスイッチングポリシーに一致しない要求は、キャッシュの既定の負荷分散仮想サーバーに転送されます。

コンテンツスイッチング機能とコンテンツスイッチングポリシーの設定の詳細については、「コンテンツスイッチング」を参照してください。

コンテンツスイッチングポリシーを作成してから、コンテンツスイッチング仮想サーバーにバインドする必要があります。

コマンド **CLI** を使用してコンテンツスイッチングポリシーを作成する

コマンドラインで、次のように入力します。

```
1 -  add cs action <name> [-targetLBVserver <string> | -targetVserver <
     string> | -targetVserverExpr <expression>]
2 -  add cs policy <policyName> -rule <expression> [-action <string>]
3 -  show cs policy [<policyName>]
4
5 <!--NeedCopy-->
```


例:

```
> add cs action action-CS-JPEG -targetLBVserver lbcachejpeg
Done
> show cs action action-CS-JPEG
    Name: action-CS-JPEG
    Target LB Vserver: lbcachejpeg
    Hits: 0
    Undef Hits: 0
    Action Reference Count: 0
Done

> add cs policy policy-CS-JPEG -rule 'HTTP.REQ.URL.SUFFIX == "jpeg"' -
   action action-CS-JPEG
Done
> show cs policy policy-CS-JPEG
    Policy: policy-CS-JPEG Rule: HTTP.REQ.URL.SUFFIX == "jpeg"
    Action: action-CS-JPEG

    HITS: 0
Done
>

> add cs action action-CS-GIF -targetLBVserver lbcachegif
Done
> show cs action action-CS-GIF
    Name: action-CS-GIF
    Target LB Vserver: lbcachegif
    Hits: 0
    Undef Hits: 0
    Action Reference Count: 0

Done
>
> add cs policy policy-CS-GIF -rule 'HTTP.REQ.URL.SUFFIX == "gif"' -
   action action-CS-GIF
Done
> show cs policy policy-CS-GIF
    Policy: policy-CS-GIF Rule: HTTP.REQ.URL.SUFFIX == "gif"
    Action: action-CS-GIF

    Hits: 0
Done
<!--NeedCopy-->
```

GUI を使用してルールベースのコンテンツスイッチングポリシーを作成する

1. **Traffic Management > Content Switching > Policies** に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [コンテンツスイッチングポリシーの作成] ダイアログボックスの [名前] ボックスに、ポリシーの名前を入力

します。

4. [アクション] タブの [追加] をクリックして、コンテンツスイッチングアクションを作成します。または、ドロップダウンリストから使用可能なアクションを選択します。

   - 「名前」(Name) タブにコンテンツ・スワース・アクションの名前を入力します。
   - ドロップダウンリストから仮想サーバまたは [式] を選択します。
     - 仮想サーバーの負荷分散
     - グローバルサーバ負荷分散仮想サーバ
     - 認証仮想サーバー
     - **NetScalar Gateway** 仮想サーバー
     - 式
   - [追加] または [編集] をクリックして、ターゲット負荷分散仮想サーバーを構成します。

5. [ログアクション] タブの [追加] をクリックして、監査メッセージアクションを作成します。または、ドロップダウンリストから使用可能な監査メッセージアクションを選択します。

6. [式] 領域で、必要な式のタイプを選択します。

7. [式エディタ] ダイアログボックスで、使用する式の構文を選択します。

   [式] 領域で [評価] をクリックして、式エバリュエーターを評価します。エバリュエータは入力した式を評価して有効であることを確認し、[結果] 領域に式の効果の分析を表示します。

8. ポリシー式を入力します。

   高度な構文の使用の詳細については、「高度なポリシー式の構成: はじめに」を参照してください。

9. [作成] をクリックします。作成したポリシーが [コンテンツスイッチングポリシー] ペインに表示されます。

CLI を使用してコンテンツスイッチングポリシーをキャッシュリダイレクト仮想サーバーにバインドする

コマンドプロンプトで次のコマンドを入力して、コンテンツスイッチングポリシーをキャッシュリダイレクト仮想サーバーにバインドし、構成を確認します。

```
1 -  bind cs vserver <name> (-lbvserver <string> | -vServer <string> (-
     policyName <string> [-targetLBVserver <string>] [-priority<
     positive_integer>] [-gotoPriorityExpression <expression>] [-type <
     type>] [-invoke (<labelType> <labelName>) ] )
2
3 -  show cs vserver [<name>]
4 <!--NeedCopy-->
```

例:

```
1 > bind cs vserver Vserver-CR-1 -policyName Policy-CS-JPEG -priority 100
2  Done
3 > bind cs vserver Vserver-CR-1 -policyName Policy-CS-GIF -priority 200
```



```
 Done
> show cs vserver Vserver-CR-1
        Vserver-CR-1 (10.102.29.60:80) - HTTP   Type: CONTENT
        State: UP
        Last state change was at Fri Jul  2 12:53:45 2010
        Time since last state change: 0 days, 00:00:58.920
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Appflow loggig: ENABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Cacheable: YES
        Vserver IP and Port insertion: OFF
        L2Conn: OFF    Case Sensitivity: ON
        Authentication: OFF
        401 Based Authentication: OFF
        Push: DISABLED  Push VServer:
        Push Label Rule: none
        HTTP Redirect Port: 0    Dtls:  OFF
        Persistence: NONE
        Listen Policy: NONE
        IcmpResponse: PASSIVE
        RHIstate: PASSIVE
        Traffic Domain:  0

1)      Content-Switching Policy: Policy-CS-JPEG Priority: 100     Hits
    : 0
2)      Content-Switching Policy: Policy-CS-GIF Priority: 200     Hits:
     0
 Done
>
<!--NeedCopy-->
```

GUI を使用して、コンテンツスイッチングポリシーをキャッシュリダイレクト仮想サーバーにバインドする

1. **[**トラフィック管理**] > [**コンテンツスイッチング**] > [**仮想サーバー**]** に移動します。
2. 詳細ウィンドウで、ポリシーをバインドする仮想サーバ (たとえば、**Vserver-CS-1**) を選択し、[**Edit**] をクリックします。
3. [**Content Switching Virtual Server**] ダイアログボックスの [詳細設定] の [ポリシー] タブで、[追加] アイコンをクリックし、[ポリシーを選択] と [種類の選択**] **** ドロップダウンリストからポリシーの種類を選択します ****** 。
4. [続行] をクリックします。
5. [**Policy Binding**] タブで、リストから使用可能なポリシーを選択し、[**Select**] をクリックするか、[**Add**

] をクリックして新しいポリシーを作成し、[**Create**] をクリックします。

6. [バインド] をクリックして、コンテンツスイッチングポリシーを仮想サーバーにバインドします。
7. 「完了」をクリックします。

ポリシー評価の優先順位を構成する

August 15, 2023

コンテンツスイッチングポリシーは、さまざまなコンテンツタイプに対応するための一般的な設定であるルールか、より具体的で特定のキャッシュサーバーに送信する必要があるコンテンツの種類を正確に定義する URL のいずれかに基づいて設定できます。基本的に、同じコンテンツをルールベースのポリシーまたは URL ベースのポリシーで定義できます。

いずれかのタイプのコンテンツスイッチングポリシーをキャッシュリダイレクト仮想サーバーにバインドすると、ルールベースのポリシーまたは URL ベースのポリシーを優先するように仮想サーバーを構成できます。これにより、特定のリクエストがどのサーバーに送られるかが決まります。

ポリシー評価の優先順位を構成するには、precedence パラメーターを使用します。このパラメーターは、コンテンツリダイレクト仮想サーバーで優先されるポリシーのタイプ (URL または RULE) を指定します。

可能な値:RULE、URL

デフォルト値:RULE

CLI を使用してポリシー評価の優先順位を設定します

コマンドプロンプトで次のコマンドを入力して、ポリシー評価の優先順位を設定し、構成を確認します。

```
-  set cr vserver <name> [-precedence (RULE | URL)]
-  show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> set cr vserver Vserver-CRD-1 -precedence URL
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
```



```
11          Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
12
13  1)      Cache bypass  Policy: bypass-cache-control
14  2)      Cache bypass  Policy: Policy-CRD
15   Done
16  >
17  <!--NeedCopy-->
```

GUI を使用してポリシー評価の優先順位を設定します

1. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動します。
2. 詳細ウィンドウで、優先順位を構成する仮想サーバー (**Vserver-CS-1** など) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (コンテンツの切り替え)] ダイアログボックスの [詳細設定] タブの [優先順位] の横にある [ルールまたは URL] をクリックし、[OK] をクリックします。

キャッシュリダイレクト仮想サーバーを管理する

August 15, 2023

キャッシュリダイレクト仮想サーバーを管理するには、キャッシュリダイレクトの統計情報を表示する必要があります。キャッシュリダイレクションサーバーを有効または無効にしたり、ポリシーヒットをオリジンではなくキャッシュにダイレクトしたりする必要がある場合があります。管理タスクには、キャッシュリダイレクト仮想サーバーのバックアップとクライアント接続の管理も含まれます。

キャッシュリダイレクト仮想サーバーの統計を表示する

August 15, 2023

キャッシュリダイレクション仮想サーバーのプロパティと、キャッシュリダイレクション仮想サーバーを通過したトラフィックの統計を表示できます。また、負荷分散仮想サーバーにバインドしたキャッシュリダイレクト仮想サーバーとポリシーを表示することもできます。

特定のキャッシュリダイレクション仮想サーバーの統計を表示するには、名前パラメーターを使用して、統計情報を表示する仮想サーバーの名前を指定します。それ以外の場合は、すべてのキャッシュリダイレクション仮想サーバーの統計が表示されます。最大長：127

CLI を使用してキャッシュリダイレクト仮想サーバーの統計を表示する

コマンドプロンプトで入力します。

```
stat cr vserver [<name>]
```

例:

```
> stat cr vserver Vserver-CRD-1

Vserver Summary
                            IP  port     Protocol        State
Vser...CRD-1         0.0.0.0    80         HTTP           UP

VServer Stats:
                                        Rate (/s)
                                           Total
Requests                                        0
                        0
Responses                                       0
                        0
Request bytes                                   0
                        0
Response bytes                                  0
                        0

 Done
>
<!--NeedCopy-->
```

GUI を使用してキャッシュリダイレクト仮想サーバーの統計を表示する

1. 「トラフィック管理」>「キャッシュリダイレクト」>「仮想サーバー」に移動します。
2. 詳細ウィンドウで、統計情報を表示する仮想サーバー (たとえば、**Vserver-CRD-1**) を選択し、[統計] をクリックします。

すべてのキャッシュリダイレクション仮想サーバーの基本統計を表示するには、サーバー名を省略します。サーバー名を含めると、仮想サーバーを通過する要求と応答の数とサイズなど、その仮想サーバーの詳細な統計が表示されます。

監視ユーティリティとダッシュボードユーティリティを使用して、キャッシュリダイレクト仮想サーバーの統計を表示します

1. 監視ユーティリティを使用して統計情報を表示するには、監視タブをクリックします。
2. 「グループの選択」ドロップダウンメニューで、「CR 仮想サーバー」を選択します。キャッシュリダイレクト仮想サーバーのリストが表示されます。
3. ダッシュボードユーティリティを使用して統計を表示するには、[ダッシュボード] タブをクリックします。

4. [統計ユーティリティ] の横にある [アプレットクライアント] または [Web Start クライアント] をクリックします。
5. 「グループの選択」ドロップダウンメニューで、「CR 仮想サーバー」を選択します。ダッシュボードには、キャッシュリダイレクション仮想サーバーの要約統計が表示されます。
6. 仮想サーバのアクティビティのチャートを表示するには、[Chart] をクリックします。仮想サーバーの統計がグラフィカルに表示されます。

キャッシュリダイレクト仮想サーバーを有効または無効にする

August 15, 2023

キャッシュリダイレクト仮想サーバーを作成すると、デフォルトで有効になります。キャッシュリダイレクト仮想サーバーを無効にすると、状態が OUT OF SERVICE に変わり、キャッシュ可能なクライアント要求のリダイレクトが停止します。ただし、NetScaler アプライアンスは、この仮想サーバーの IP アドレスに対する ARP および PING 要求に引き続き応答します。

CLI を使用してキャッシュリダイレクト仮想サーバーを有効または無効にする

コマンドラインで、以下のいずれかのコマンドを入力します。

```
-  enable cr vserver <name>
-  show cr vserver <name>
-  disable cr vserver <name>
-  show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> enable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>

```



```
> disable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: OUT OF SERVICE  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```

GUI を使用してキャッシュリダイレクト仮想サーバーを有効または無効にする

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. ナビゲーションペインで、「キャッシュリダイレクト」を展開し、「仮想サーバー」をクリックします。
3. 詳細ペインで、有効または無効にする仮想サーバー (たとえば、**VServer-CRD-1**) を選択し、[統計] をクリックします。
4. 「続行」ダイアログ・ボックスで、「はい」をクリックします。

オリジン **Web** サーバーではなくキャッシュへのポリシーリクエストの直接的な要求

August 15, 2023

デフォルトでは、リクエストがポリシーに一致すると、NetScaler ADC アプライアンスは、キャッシュリダイレクトの構成に応じて、オリジンサーバーに直接送信するか、オリジンの負荷分散仮想サーバーに要求を転送します。

要求がポリシーに一致したときに、要求がキャッシュの負荷分散仮想サーバーに転送されるように、デフォルトの動作を変更できます。

ポリシー要求の送信先をオリジンまたはキャッシュに変更するには、パラメータを使用します。`onPolicyMatch` パラメータを使用して、キャッシュリダイレクトポリシーに一致するリクエストの送信先を指定します。

有効なオプションは次のとおりです。

1. `CACHE` -一致するすべてのリクエストをキャッシュに転送します。
2. `ORIGIN` -一致するすべてのリクエストをオリジンサーバーに送信します。

> 注:
>
> このオプションが機能するには、キャッシュリダイレクトタイプをPOLICYとして選択する必要があります。

可能な値: CACHE, ORIGIN

デフォルト値: ORIGIN

CLI を使用して、ポリシー要求の宛先をオリジンまたはキャッシュに変更する

コマンドプロンプトで次のコマンドを入力して、ポリシーヒットの宛先を変更し、設定を確認します。

```
set cr vserver <name> [-onPolicyMatch (ORIGIN | CACHE)]
<!--NeedCopy-->
```

```
show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> set cr vserver Vserver-CRD-1 -onPolicyMatch CACHE
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

GUI を使用して、ポリシーヒットの宛先をオリジンまたはキャッシュに変更する

1. [トラフィック管理] **>** [キャッシュリダイレクト] **>** [仮想サーバー] に移動します。
2. 詳細ペインで、ポリシー要求の宛先を変更する仮想サーバー (たとえば、**vserver-CRD-1**) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 **(**キャッシュリダイレクト**)**] ダイアログボックスで、[詳細設定] をクリックします。
4. [リダイレクト先] ドロップダウンリストから [キャッシュ] または [オリジン] を選択します。
5. [**OK**] をクリックします。

キャッシュリダイレクト仮想サーバーをバックアップする

August 15, 2023

キャッシュのリダイレクトは、プライマリ仮想サーバーに障害が発生した場合、または過剰なトラフィックを処理できない場合に失敗する可能性があります。プライマリ仮想サーバーに障害が発生したときにトラフィックの処理を引き継ぐバックアップ仮想サーバーを指定できます。

バックアップキャッシュリダイレクト仮想サーバーを指定するには、バックアップ仮想サーバーを指定する BackupVServer パラメーターを使用します。最大長：127

CLI を使用してバックアップキャッシュリダイレクト仮想サーバーを指定します

コマンドプロンプトで次のコマンドを入力して、バックアップキャッシュリダイレクト仮想サーバーを指定し、構成を確認します。

```
-  set cr vserver <name> [-backupVServer <string>]
-  show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> set cr vserver Vserver-CRD-1 -backupVServer Vserver-CRD-2
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

GUI を使用したバックアップキャッシュリダイレクト仮想サーバの指定

1. [トラフィック管理] **>** [キャッシュリダイレクト] **>** [仮想サーバー] に移動します。
2. 詳細ペインで、ポリシー要求の宛先を変更する仮想サーバー (たとえば、**vserver-CRD-1**) を選択し、[開く] をクリックします。

3. [仮想サーバーの構成 (キャッシュリダイレクト)] ダイアログボックスで、[詳細設定] タブを選択します。
4. [仮想サーバーのバックアップ] ドロップダウンリストで、仮想サーバーを選択します。
5.「OK」をクリックします。

仮想サーバーのクライアント接続を管理する

August 15, 2023

キャッシュリダイレクト仮想サーバーでタイムアウトを構成して、クライアント接続が無期限に開いたままにならないようにすることができます。リクエストに Via ヘッダーを挿入することもできます。ネットワークの混雑を緩和するために、開いている TCP 接続を再利用できます。キャッシュリダイレクト仮想サーバー接続の遅延クリーンアップを有効または無効にできます。

設定に従って、PING 要求に ICMP 応答を送信するようにアプライアンスを設定できます。仮想サーバーに対応する IP アドレスで ICMP RESPONSE を VSVR_CNTRLD に設定し、仮想サーバーでは ICMP VSERVER RESPONSE を設定します。

仮想サーバーでは、次の設定を行うことができます。

- すべての仮想サーバーで ICMP VSERVER RESPONSE を PASSIVE に設定すると、アプライアンスは常に応答します。
- すべての仮想サーバーで ICMP VSERVER RESPONSE を ACTIVE に設定すると、1 台の仮想サーバーが稼働していてもアプライアンスは応答します。
- ICMP VSERVER RESPONSE を一部で ACTIVE に設定し、他の仮想サーバーを PASSIVE に設定すると、1 台の仮想サーバーが ACTIVE に設定されていても、アプライアンスは応答します。

このドキュメントでは、次の内容について説明します。

- クライアントタイムアウトの設定
- リクエストに Via ヘッダーを挿入
- TCP 接続を再利用
- 遅延接続クリーンアップの設定

クライアントタイムアウトの設定

キャッシュリダイレクト仮想サーバーのタイムアウト値を設定することで、クライアント要求の有効期限を指定できます。タイムアウト値は、キャッシュリダイレクト仮想サーバーがクライアント要求に対する応答を受信するまで待機する秒数です。

タイムアウト値を設定するには、cltTimeout パラメーターを使用します。このパラメーターでは、NetScaler アプライアンスがアイドル状態のクライアント接続をすべて閉じるまでの時間を秒単位で指定します。デフォルト値は、HTTP/SSL ベースのサービスの場合は 180 秒、TCP ベースのサービスの場合は 9000 秒です。

CLI を使用してクライアントのタイムアウトを設定する

コマンドプロンプトで次のコマンドを入力して、クライアントのタイムアウトを構成し、構成を確認します。

```
-  set cr vserver <name> [-cltTimeout <secs>]
-  show cr vserver <name>
<!--NeedCopy-->
```

例**:**

```
> set cr vserver Vserver-CRD-1 -cltTimeout 6000
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

GUI を使用してクライアントのタイムアウトを設定する

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 詳細ペインで、クライアントタイムアウトを設定する仮想サーバー (たとえば、**VServer-CRD-1**) を選択し、[開く] をクリックします。
3. 仮想サーバーの構成 (キャッシュリダイレクト) ダイアログボックスで、「詳細設定」タブを選択します。
4. 「クライアントタイムアウト (秒)」テキストボックスに、タイムアウト値を秒単位で入力します。
5. 「OK」をクリックします。

リクエストに **Via** ヘッダーを挿入

Via ヘッダーには、要求または応答の開始点と終了点の間のプロトコルと受信者が一覧表示され、要求が送信されたプロキシをサーバーに通知します。各 HTTP 要求に Via ヘッダーを挿入するようにキャッシュリダイレクト仮想サー

バーを構成できます。via パラメータは、キャッシュリダイレクト仮想サーバーを作成するとデフォルトで有効になります。

クライアントリクエストでの Via ヘッダーの挿入を有効または無効にするには、via パラメーターを使用します。このパラメーターは、HTTP リクエストに Via ヘッダーを挿入するときのシステムの状態を指定します。

指定可能な値: オン、オフ

デフォルト値:ON

CLI を使用してクライアントリクエストの **VIA** ヘッダー挿入を有効または無効にする

コマンドプロンプトで入力します。

```
-  set cr vserver <name> [-via (ON|OFF)]
-  show cr vserver <name>
<!--NeedCopy-->
```

例**:**

```
> set cr vserver Vserver-CRD-1 -via ON
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```

GUI を使用してクライアント要求の **VIA** ヘッダー挿入を有効または無効にする

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 詳細ペインで、クライアントタイムアウトを設定する仮想サーバー (たとえば、**VServer-CRD-1**) を選択し、[開く] をクリックします。
3. 仮想サーバーの構成 (キャッシュリダイレクト) ダイアログボックスで、「詳細設定」タブを選択します。
4. 「Via」チェックボックスを選択します。
5. 「OK」をクリックします。

TCP 接続を再利用

NetScaler アプライアンスは、クライアント接続全体でキャッシュサーバーとオリジンサーバーへの TCP 接続を再利用するように構成できます。これにより、サーバーとアプライアンス間のセッションを確立するのに必要な時間を節約できるため、パフォーマンスが向上します。キャッシュリダイレクト仮想サーバーを作成すると、再利用オプションがデフォルトで有効になります。

TCP 接続の再利用を有効または無効にするには、reuse パラメータを使用します。このパラメータは、クライアント接続全体でのキャッシュまたはオリジンサーバーへの TCP 接続の再利用の状態を指定します。

指定可能な値: オン、オフ

デフォルト値:ON

CLI を使用して TCP 接続の再利用を有効または無効にする

コマンドプロンプトで入力します。

```
-  set cr vserver <name> [-reuse (ON|OFF)]
-  show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> set cr vserver Vserver-CRD-1 -reuse ON
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

GUI を使用して TCP 接続の再利用を有効または無効にする

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 詳細ペインで、クライアントタイムアウトを設定する仮想サーバー (たとえば、**VServer-CRD-1**) を選択し、[開く] をクリックします。

3. 仮想サーバーの構成 (キャッシュリダイレクト) ダイアログボックスで、「詳細設定」タブを選択します。
4. 「再利用」チェックボックスを選択します。
5. 「OK」をクリックします。

遅延接続クリーンアップの設定

ダウンステートフラッシュオプションは、キャッシュリダイレクト仮想サーバー上の接続のクリーンアップを遅延して実行します。キャッシュリダイレクト仮想サーバーを作成すると、ダウンステートフラッシュオプションがデフォルトで有効になります。

ダウンステートフラッシュオプションを有効または無効にするには、downStateFlush パラメータを設定します。

設定可能な値: ENABLED, DISABLED

デフォルト値: 有効

CLI を使用してダウンステートフラッシュオプションを有効または無効にする

コマンドプロンプトで次のコマンドを入力して、遅延接続のクリーンアップを設定し、構成を確認します。

```
-  set cr vserver <name> [-downStateFlush (ENABLED | DISABLED)]
-  show cr vserver <name>
<!--NeedCopy-->
```

例**:**

```
> set cr vserver Vserver-CRD-1 -downStateFlush ENABLED
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

GUI を使用して **TCP** 接続の再利用を有効または無効にする

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。

2. 詳細ペインで、クライアントタイムアウトを設定する仮想サーバー (たとえば、**VServer-CRD-1**) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (キャッシュリダイレクト)] ダイアログボックスで、[詳細設定] タブをクリックします。
4.「ダウンステートフラッシュ」チェックボックスを選択します。
5.「OK」をクリックします。

UDP 仮想サーバーの外部 TCP 正常性チェックを有効にする

August 15, 2023

パブリッククラウドでは、ネイティブロードバランサを第 1 層として使用する場合、NetScaler ADC アプライアンスを第 2 層ロードバランサーとして使用できます。ネイティブロードバランサは、アプリケーションロードバランサ（ALB）またはネットワークロードバランサ（NLB）になります。ほとんどのパブリッククラウドは、ネイティブロードバランサーで UDP ヘルスプローブをサポートしていません。UDP アプリケーションの正常性を監視するために、パブリッククラウドでは、サービスに TCP ベースのエンドポイントを追加することをお勧めします。エンドポイントは UDP アプリケーションの正常性を反映します。

NetScaler アプライアンスは、UDP 仮想サーバーの外部 TCP ベースのヘルスチェックをサポートしています。この機能により、キャッシュリダイレクション仮想サーバーの VIP と設定されたポートに TCP リスナーが導入されます。TCP リスナーは、仮想サーバーのステータスを反映します。

CLI を使用して UDP 仮想サーバーの外部 TCP ヘルスチェックを有効にするには

コマンドプロンプトで次のコマンドを入力して、TcpProbeport オプションを指定して外部 TCP ヘルスチェックを有効にします。

```
add cr vserver <name> <serviceType> -tcpProbePort <tcpProbePort>

<!--NeedCopy-->
```

例**:**

```
add cr vserver Vserver-CR-1 HTTP -tcpProbePort 80
<!--NeedCopy-->
```

GUI を使用して UDP 仮想サーバーの外部 TCP ヘルスチェックを有効にするには

1. [トラフィック管理**] > [**キャッシュリダイレクト**] > [**仮想サーバー] に移動し、仮想サーバーを作成します。
2. [追加] をクリックして、仮想サーバーを作成します。
3. [基本設定] ペインの [**TCP** プローブポート**]** フィールドにポート番号を追加します。
4. [**OK**] をクリックします。

N 層キャッシュリダイレクト

August 15, 2023

インターネットサービスプロバイダー (ISP) は、大量のキャッシュデータ (通常は毎秒数ギガバイト) を効率的に処理するために、複数の専用キャッシュサーバーを配備しています。NetScaler アプライアンスのキャッシュリダイレクト機能はキャッシュサーバーの負荷分散に役立ちますが、単一または複数のアプライアンスでは大量のトラフィックを効率的に処理できない場合があります。

この問題は、NetScaler アプライアンスを 2 つの層（レイヤー）に展開することで解決できます。上位層のアプライアンスは下位層のアプライアンスの負荷分散を行い、下層のアプライアンスはキャッシュサーバーの負荷分散を行います。この配置は *n* 層キャッシュリダイレクトと呼ばれます。

監査やセキュリティなどの目的で、ISP は IP アドレス、提供された情報、やり取りの時間などのクライアントの詳細を追跡する必要があります。そのため、NetScaler アプライアンスを介したクライアント接続は完全に透過的でなければなりません。ただし、NetScaler アプライアンスを並行して展開した状態でトランスペアレントキャッシュリダイレクトを構成する場合は、クライアントの IP アドレスをすべてのアプライアンスで共有する必要があります。クライアント IP アドレスを共有すると競合が発生し、ルーター、キャッシュサーバー、オリジンサーバー、その他の NetScaler アプライアンスなどのネットワークデバイスが、応答の送信先となるアプライアンス、ひいてはクライアントを特定できなくなります。

N 層キャッシュリダイレクトの実装方法

この問題を解決するために、アプライアンスの n 層キャッシュリダイレクトは、送信元ポート範囲を下位層のアプライアンス間で分割し、キャッシュサーバーに送信される要求にクライアント IP アドレスを含めます。上位層の NetScaler アプライアンスは、アプライアンスに不必要な負荷がかからないように、セッションレスの負荷分散を行うように構成されています。

下位層の NetScaler アプライアンスがキャッシュサーバーと通信する場合、マップされた IP アドレス（MIP）を使用してソース IP アドレスを表します。そのため、キャッシュサーバーはリクエストを受信したアプライアンスを識別し、同じアプライアンスにレスポンスを送信できます。

下位層の NetScaler アプライアンスは、キャッシュサーバーに送信される要求のヘッダーにクライアント IP アドレスを挿入します。ヘッダーのクライアント IP は、アプライアンスがキャッシュサーバーからの応答を受信したときに、またはキャッシュミスの場合にオリジンサーバーからパケットを転送するクライアントを決定するのに役立ちます。オリジンサーバーは、リクエストヘッダーに挿入されたクライアント IP に従って送信するレスポンスを決定します。

オリジンサーバーは、オリジンサーバーがリクエストを受信したソースポート番号を含む応答を上位層アプライアンスに送信します。ソースポートの全範囲（1024～65535）は、下位層の NetScaler アプライアンスに分散されます。下位層の各アプライアンスには、範囲内のアドレスのグループが排他的に割り当てられます。この割り当てにより、

上位層のアプライアンスは、リクエストを配信元サーバーに送信した下位層の NetScaler アプライアンスを明確に識別できます。そのため、上位層のアプライアンスは適切な下位層のアプライアンスに応答を転送できます。

上位層の NetScaler アプライアンスはポリシーベースのルーティングを行うように構成され、ルーティングポリシーは、送信元ポート範囲から宛先アプライアンスの IP アドレスを決定するように定義されます。

N 階層 CRD の設定に必要なセットアップ

n 層キャッシュリダイレクトを機能させるには、以下の設定が必要です。

上位層の NetScaler アプライアンスごとに、

- レイヤ 3 モードを有効にします。
- トラフィックが宛先ポートの範囲に従って転送されるように、ポリシーベースルート（PBR）のポリシーを定義します。
- 負荷分散仮想サーバーを構成します。
- クライアントからのすべてのトラフィックを受信するように仮想サーバーを構成します。サービスタイプ/プロトコルを ANY に、IP アドレスをアスタリスク (*) に設定します。
- MAC ベースのリダイレクトモードでセッションレスの負荷分散を有効にして、上位層の NetScaler アプライアンスに不必要な負荷がかからないようにします。
- 「プロキシポートを使用」オプションが有効になっていることを確認します。
- 下位層のアプライアンスごとにサービスを作成し、すべてのサービスを仮想サーバーにバインドします。

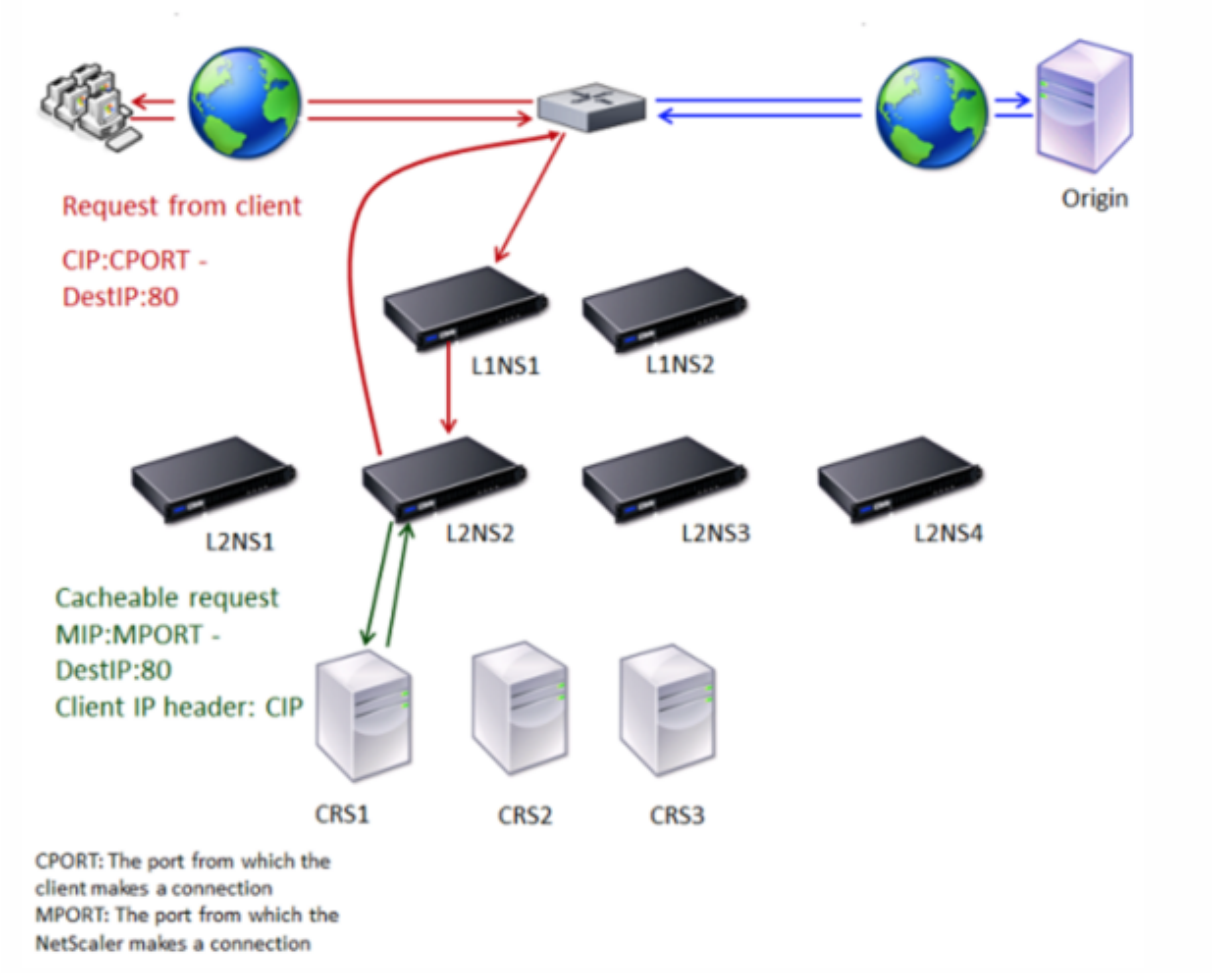
下位階層の NetScaler アプライアンスごとに、

- アプライアンスのキャッシュリダイレクションポート範囲を設定します。下位層の各アプライアンスに排他範囲を割り当てます。
- 負荷分散仮想サーバーを構成し、MAC ベースのリダイレクトを有効にします。
- このアプライアンスによって負荷分散されるキャッシュサーバーごとにサービスを作成します。サービスを作成するときは、ヘッダーにクライアント IP を挿入できるようにします。次に、すべてのサービスを負荷分散仮想サーバーにバインドします。
- トランスペアレントモードのキャッシュリダイレクト仮想サーバーを次の設定で構成します。
  - オリジン USIP オプションを有効にします。
  - ソース IP 表現を追加して、ヘッダーにクライアント IP を含めます。
  - 「ポート範囲を使用」オプションを有効にします。

キャッシュヒット時の N 層キャッシュリダイレクトの仕組み

次の図は、クライアント要求がキャッシュ可能で、応答がキャッシュサーバーから送信される場合のキャッシュリダイレクトの仕組みを示しています。

図 1: キャッシュヒット時にキャッシュリダイレクト



L1NS1 と L1NS2 の 2 つの NetScaler アプライアンスが上位層に展開され、4 つの NetScaler アプライアンス（L2NS1、L2NS2、L2NS3、L2NS4）が下層に展開されます。クライアント A は要求を送信し、その要求はルータによって転送されます。キャッシュサーバー CRS1、CRS2、および CRS3 はキャッシュリクエストを処理します。オリジンサーバー O は、キャッシュされていないリクエストを処理します。

トラフィックフロー

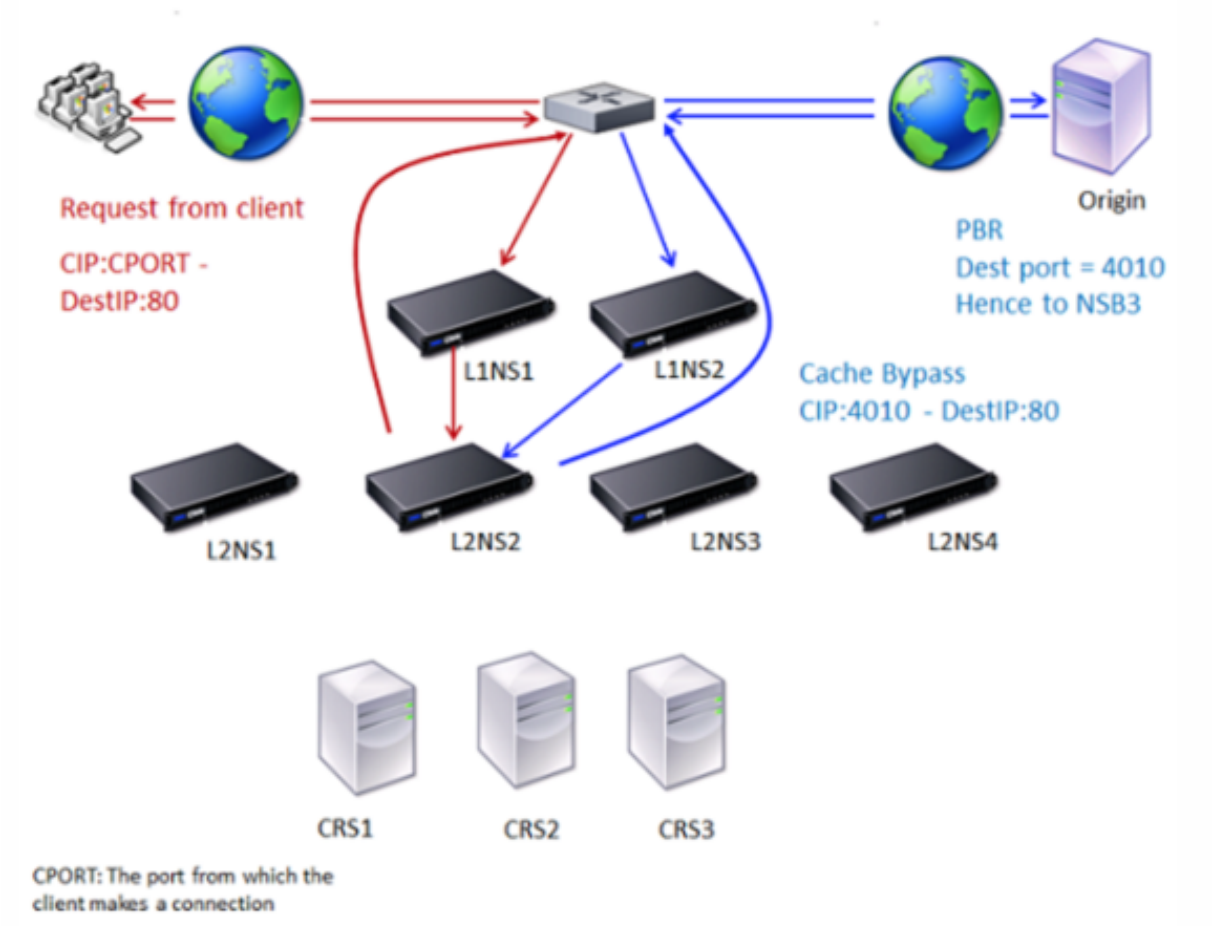
1. クライアントが要求を送信し、ルータはそれを L1NS1 に転送します。
2. L1NS1 は L2NS2 へのリクエストのロードバランシングを行います。
3. L2NS2 は要求をキャッシュサーバー CRS1 にロードバランシングし、要求はキャッシュ可能です。L2NS2 はリクエストヘッダーにクライアント IP を含めます。
4. L2NS2 は CRS1 に接続する際にその MIP を送信元 IP アドレスとして使用したため、CRS1 は L2NS2 に応答を送信します。
5. L2NS2 は、リクエストヘッダーのクライアント IP アドレスを使用して、リクエストの送信元のクライアントを識別します。L2NS2 は応答をルーターに直接送信するため、上位層のアプライアンスに不必要な負荷がかからないようになります。

6. ルータは応答をクライアント A に転送します。

キャッシュバイパス中の **N** 層キャッシュリダイレクトの仕組み

次の図は、クライアント要求が応答を求めてオリジンサーバーに送信されるときのキャッシュリダイレクトの仕組みを示しています。

図 2：キャッシュバイパスの場合、キャッシュリダイレクト



L1NS1 と L1NS2 の 2 つの NetScaler アプライアンスが上位層に展開され、4 つの NetScaler アプライアンス（L2NS1、L2NS2、L2NS3、L2NS4）が下層に展開されます。クライアント A は要求を送信し、その要求はルータによって転送されます。キャッシュサーバー CRS1、CRS2、および CRS3 はキャッシュリクエストを処理します。オリジンサーバー O は、キャッシュされていないリクエストを処理します。

トラフィックフロー

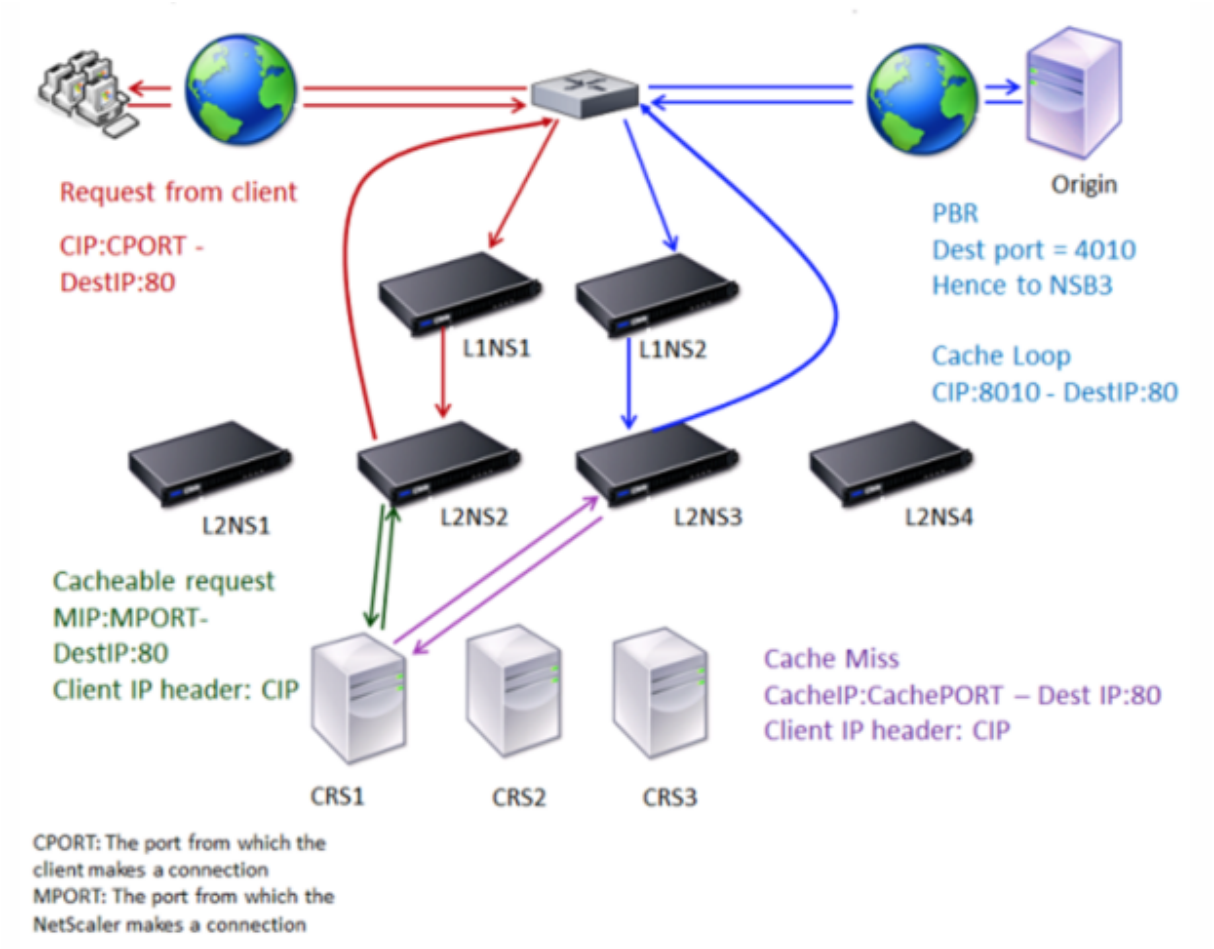
1. クライアントが要求を送信し、ルータはそれを L1NS1 に転送します。
2. L1NS1 は L2NS2 へのリクエストのロードバランシングを行います。

3. リクエストはキャッシュできません (キャッシュバイパス)。そのため、L2NS2 はルーターを介してリクエストをオリジンサーバーに送信します。
4. オリジンサーバーは、応答を上位層のアプライアンス L1NS2 に送信します。
5. PBR ポリシーに従って、L1NS2 は下位階層の適切なアプライアンスである L2NS2 にトラフィックを転送します。
6. L2NS2 は、リクエストヘッダーのクライアント IP アドレスを使用してリクエストの送信元クライアントを識別し、応答をルーターに直接送信します。これにより、上位層のアプライアンスに不必要な負荷がかかります。
7. ルータは応答をクライアント A に転送します。

キャッシュミス時の **N** 層キャッシュリダイレクトの仕組み

次の図は、クライアント要求がキャッシュされていない場合のキャッシュリダイレクトの仕組みを示しています。

図 3: キャッシュミスによるキャッシュリダイレクト



L1NS1 と L1NS2 の 2 つの NetScaler アプライアンスが上位層に展開され、4 つの NetScaler アプライアンス (L2NS1、L2NS2、L2NS3、L2NS4) が下層に展開されます。クライアント A は要求を送信し、その要求はルータによって転送されます。キャッシュサーバー CRS1、CRS2、および CRS3 はキャッシュリクエストを処理します。オ

リジンサーバー O は、キャッシュされていないリクエストを処理します。

トラフィックフロー

1. クライアントが要求を送信し、ルータはそれを L1NS1 に転送します。
2. L1NS1 は L2NS2 へのリクエストのロードバランシングを行います。
3. 要求はキャッシュ可能であるため、L2NS2 は要求をキャッシュサーバー CRS1 にロードバランシングします。
4. CRS1 には応答がありません（キャッシュミス）。CRS1 は、下位層のアプライアンスを介してリクエストをオリジンサーバーに転送します。L2NS3 はトラフィックをインターセプトします。
5. L2NS3 はヘッダーからクライアント IP を取得し、要求をオリジンサーバーに転送します。パケットに含まれる送信元ポートは、要求がオリジンサーバーに送信される L2NS3 ポートです。
6. オリジンサーバーは、応答を上位層のアプライアンス L1NS2 に送信します。
7. PBR ポリシーに従って、L1NS2 は下位階層の適切なアプライアンスである L2NS3 にトラフィックを転送します。
8. L2NS3 は応答をルータに転送します。
9. ルータは応答をクライアント A に転送します。

上位層の **NetScaler** アプライアンスを構成する

December 8, 2023

上位層の NetScaler アプライアンスをそれぞれ次のように構成します。

コマンド **CLI** を使用して **n** 層キャッシュリダイレクト用の上位層アプライアンスを構成します

コマンドプロンプトで、次のコマンドを入力します：

- `add service <name>@ <serviceIP> <serviceType> <port>`

  追加するサービスごとにこのコマンドを実行します。

- `add lb vserver <name>@ ANY * <port> -persistenceType <persistenceMethod> -lbMethod <lbMethod> -m MAC -sessionless ENABLED -cltTimeout <client_Timeout_Value>`

- `bind lb vserver <name>@ <serviceName>`

  バインドするサービスごとにこのコマンドを実行します。

- `enable ns mode l3`

- `add ns pbr <name> <action> -srcPort <sourcePortNumber> -destPort <startPortNumber-endPortNumber> -nexthop <serviceIpAddress> -protocol TCP`
- `apply ns pbrs`

  必要な PBR をすべて追加したら、このコマンドを実行します。

GUI を使用して n 層キャッシュリダイレクト用の上位層アプライアンスを構成する

1. L3 モードを有効にする:

   a) ナビゲーションペインで、[システム] をクリックし、[設定] をクリックします。
   b)「設定」で、「モードを設定」をクリックします。
   c) レイヤー **3** モード **(IP** 転送**)** を選択します。
   d)［**OK**］をクリックします。

2. ポリシーベースルーティング (PBR) の設定:

   a) [システム**] > [**ネットワーク**] > [PBR**] に移動します。
   b) ポリシーベースルーティング **(PBR)** ペインで、「追加」をクリックします。
   c) PBR の名前を入力します。
   d) アクションを [許可] として選択します。
   e)「**Next Hop**」に、下位層のアプライアンスを表すサービスの IP アドレスを入力します。
   f)「プロトコル」で「**TCP**」を選択します。
   g) 追加する下位層のアプライアンスに対応する送信元ポートと宛先ポートの範囲を入力します。
   h)［作成］をクリックします。
   i) 詳細ペインで PBR を選択し、[適用] をクリックします。
   j) 下位層のアプライアンスごとにこれらの手順を繰り返します。

3. 下位層のアプライアンスごとにサービスを作成します:

   a)［**Traffic Management**］**>**［**Load Balancing**］**>**［**Services**］の順に移動します。
   b) 詳細ペインで、[追加] をクリックします。
   c) 名前、プロトコル、IP アドレス、およびポートを指定します。プロトコルは **ANY** でなければなりません。
   d)［作成］をクリックします。

4. 負荷分散仮想サーバーを設定します。

   a) **Traffic Management > Load Balancing > Virtual Servers** に移動します。
   b) 詳細ペインで、[追加] をクリックします。
   c) 名前、プロトコル、IP アドレス、およびポートを指定します。プロトコルは ANY、IP アドレスは * でなければなりません。
   d) [サービス] で、下位層の NetScaler アプライアンスを表すサービスを選択します。

e) [詳細設定] で、[リダイレクションモード] を [**MAC** ベース] として選択し、[セッションレス] を選択します。
f) ［作成］をクリックします。

下位層の **NetScaler ADC** アプライアンスを構成する

August 15, 2023

下位層の NetScaler アプライアンスをそれぞれ次のように構成します。

CLI を使用して **n** 層キャッシュリダイレクト用の下位層アプライアンスを構成する

コマンドプロンプトで、次のコマンドを入力します。

- `add service <name>@ <cacheServiceIP> <serviceType> <port> -cip ENABLED "ClientIP"-cachetype transparent`

  キャッシュサーバーごとに同じ手順を繰り返します。
- `add lb vserver <name>@ <serviceType> -m MAC`
- `bind lb vserver <name>@ <cacheServiceName>`

  キャッシュサーバーごとに同じ手順を繰り返します。
- `add cr vserver <name> <serviceType> * <port> -srcIPExpr "HTTP.REQ .HEADER("ClientIP")"-originusip ON -usePortRange ON`
- `set ns param-crPortRange <startPortNumber-endPortNumber>`

GUI を使用して **n** 層キャッシュリダイレクト用の下位層アプライアンスを構成する

1. キャッシュサーバーごとにサービスを作成します。サービスを作成するには:
   a) ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
   b) 詳細ペインで [追加] をクリックし、名前とプロトコルを指定します。「直接アドレス指定可能」チェックボックスをオフにします。
   c)「詳細設定」タブで、「グローバルオーバーライド」チェックボックスと「クライアント IP」チェックボックスを選択し、「ヘッダー」ボックスに「ClientIP」と入力します。
   d)「キャッシュタイプ」ボックスで、「透明キャッシュ」を選択します。
   e) ［作成］をクリックします。
2. 負荷分散仮想サーバーを設定します。

a) [トラフィック管理] > [負荷分散] > [仮想サービス] に移動します。
b) 詳細ペインで [追加] をクリックし、名前、プロトコル、IP アドレス、およびポートを指定します。IP アドレスはアスタリスク (*) でなければなりません。
c)「サービス」タブで、キャッシュサーバーを表すサービスを選択します。
d)「詳細設定」タブの「リダイレクトモード」で、「MAC ベース」を選択します。
e)［作成］をクリックします。

3. キャッシュリダイレクト仮想サーバーを設定します。

a) [トラフィック管理] > [負荷分散] > [仮想サービス] に移動します。
b) 詳細ペインで [追加] をクリックし、名前、プロトコル、IP アドレス、およびポートを指定します。IP アドレスは * でなければなりません。
c) [キャッシュタイプ] で [透明] を選択します。
d) [詳細設定] タブの [キャッシュサーバー] ボックスで、新しい負荷分散仮想サーバーを選択し、[Origin USIP] と [Use Port Range] チェックボックスをオンにします。「ソース IP 表現」ボックスに「HTTP.REQ.HEADER (「クライアントチップ」) と入力します。
e)［作成］をクリックします。

4. アプライアンスの送信元ポート範囲を割り当てます。

a) ナビゲーションペインで、[システム] をクリックし、[設定] をクリックします。
b)「設定」グループで、「グローバルシステム設定の変更」リンクをクリックします。
c)「キャッシュリダイレクトポート範囲」グループで、「開始ポート」にポート番号、「終了ポート」にポート番号を入力して、アプライアンスのポート範囲を指定します。
d)「OK」をクリックします。

要求の宛先 **IP** アドレスを発信元 **IP** アドレスに変換する

August 15, 2023

NetScaler アプライアンス上のフォワードプロキシキャッシュリダイレクト仮想サーバーを構成して、キャッシュリダイレクト仮想サーバーに着信するリクエストの宛先 IP アドレスをオリジンサーバーの IP アドレスに変換できます。この変換は、リクエストがキャッシュサーバーに送信されるか、オリジンサーバーに送信されるかに関係なく行われます。
以前は、コンテンツスイッチングポリシーを使用するキャッシュリダイレクトの制限により、サービスプロバイダー環境のフォワードプロキシキャッシュリダイレクト仮想サーバーをファイアウォール経由でトラフィックを送信するのに効果的に使用できませんでした。キャッシュリダイレクト仮想サーバーは、パケットがキャッシュに送信されたときに、送信元 IP アドレスを宛先 IP に変換しませんでした。リクエストがキャッシュされたサーバーから処理された場合のみ、宛先 IP アドレスがオリジンサーバーのアドレスになりました。

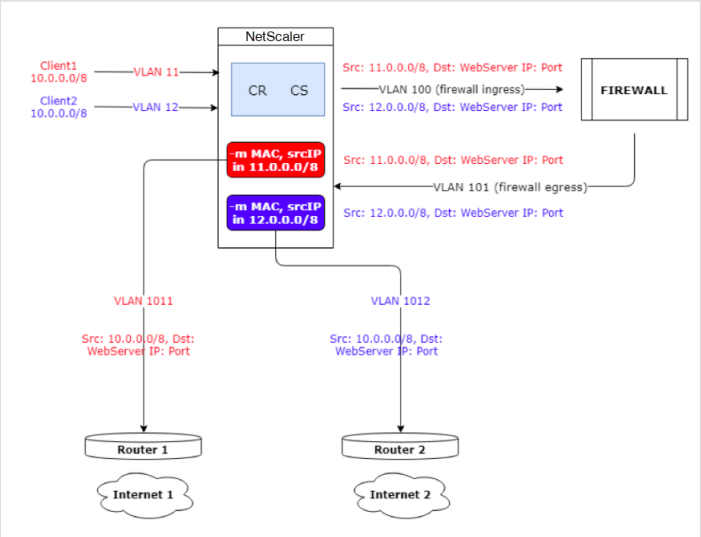
注**:** トランスペアレントキャッシュリダイレクト仮想サーバーでは、リクエストの宛先 IP アドレスから送信元 IP アドレスへの変換はサポートされていません。トランスペアレントキャッシュリダイレクト仮想サーバーの場合、このオプションを OFF に設定する必要があります。

使用例

NetScaler アプライアンスが転送プロキシキャッシュリダイレクト、ファイアウォール、および再利用クライアント IP アドレスで構成されている展開では、ファイアウォールは再利用された IP アドレスを区別したり使用したりできません。そのため、これらの再利用された IP アドレスは別の IP アドレスに変換する必要があります。再利用された IP アドレスを変換するには、NetScaler アプライアンスが以下を実行する必要があります。

1. DNS 負荷分散仮想サーバーに宛先の解決を問い合わせます。
2. 送信元の IP アドレスと宛先のポート番号を更新します。
3. 要求をファイアウォールに送り返します。

転送プロキシキャッシュリダイレクト、ファイアウォール、2 つのルーター（ルーター 1 とルーター 2）用に構成された NetScaler アプライアンスを備えた次の展開を考えてみましょう。ネットワークトラフィックは、それぞれルーター 1 を経由してインターネット 1 に、ルーター 2 を経由してインターネット 2 に流れます。



この例では、クライアントからの入力要求は、VLAN11 または VLAN12 の 2 つの異なる VLAN から送信されます。クライアント IP アドレス (10.0.0.0) は再利用されます。
キャッシュリダイレクトとコンテンツスイッチングポリシーに基づいて、リクエストはオリジンサーバーまたはファイアウォールに直接送信できます。

- 要求がファイアウォールをバイパスしてインターネットに送信する必要がある場合、入力要求 VLAN に基づいてルーター 1 またはルーター 2 のいずれかが選択され、要求はインターネット 1 またはインターネット 2 に送信されます。

- リクエストがファイアウォールを通過する必要がある場合は、リクエストのソース IP を特定の IP アドレスに変換する必要があります。変換された IP アドレスを使用して、リクエストが送信された VLAN を識別できます。たとえば、入力要求が VLAN11 からのものである場合、送信元 IP アドレスは 11.x.x.x に変換されます。要求が VLAN12 からのものである場合、送信元 IP アドレスは 12.x.x.x に変換されます。

ファイアウォールが要求を処理すると、要求はアプライアンスに返送されます。次に、アプライアンスはリッスンポリシーとネットプロファイルの組み合わせを使用して、送信元 IP アドレスを元の IP アドレスに変換し直し、入力された VLAN ID に基づいて要求をルータ 1 またはルータ 2 に送信します。

注**:** キャッシュにバインドされている負荷分散仮想サーバーのモードは、常に MAC モードに設定する必要があります。この機能の IP モードはブロックされていませんが、IP モードに設定すると予期しない動作が発生します。

CLI を使用してリクエストの宛先 **IP** アドレスとポート番号をオリジン **IP** アドレスに変換するには

コマンドプロンプトで次を入力します。

```
set cr vserver <vsname> -useoriginIpPortForCache <YES|NO>
<!--NeedCopy-->
```

例:

```
set cr vserver cvsrv1 -useoriginIpPortForCache YES
<!--NeedCopy-->
```

useOriginIPportForCache が Yes に設定されていて、リクエストをキャッシュされたサーバーから処理する必要がある場合、リクエストの宛先 IP はオリジンサーバーの IP アドレスに変換されます。

注:useOriginIPportForCache が有効な場合は、キャッシュにバインドされている負荷分散仮想サーバーを必ず MAC モードに設定してください。

GUI を使用してリクエストの宛先 **IP** アドレスとポートをオリジン **IP** アドレスに変換するには

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動し、[追加] をクリックします。
2. キャッシュリダイレクト仮想サーバーの詳細を指定します。
3. リクエストの宛先 **IP** アドレスを送信元 **IP** アドレスに変換するには、「キャッシュにオリジン **IP** ポートを使用」を選択します。
4. [**OK**] をクリックします。

クラスタリング

August 15, 2023

> 注
>
> この機能は、NetScaler Advanced エディションまたは Premium エディションのライセンスで利用できます。

NetScaler ADC クラスタは、1 つのシステムイメージとして連携して動作する nCore アプライアンスのグループです。クラスタの各アプライアンスはノードと呼ばれます。クラスターには、1 つのアプライアンスを含めることも、最大 32 の NetScaler nCore ハードウェアまたは仮想アプライアンスをノードとして配置することもできます。

クライアントトラフィックはノード間で分散され、高可用性、高スループット、およびスケーラビリティを提供します。

クラスタを作成するには、次の手順を実行する必要があります:

- アプライアンスをクラスタノードとして追加します。
- ノード間の通信を設定します。
- クライアントおよびサーバーネットワークへのリンクを設定します。
- アプライアンスを構成し、クライアントとサーバーのトラフィックの分散を構成します。

NetScaler クラスターのサポート性マトリックス

August 15, 2023

NetScaler ADC アプライアンスのクラスタリングは、NetScaler ADC 構成の幅広い機能をサポートします。

次の表に、NetScaler の機能と、異なる NetScaler リリースのクラスタセットアップにおけるサポート性のステータスを示します。NetScaler BLX クラスタの一部の NetScaler 機能のサポート状況は、NetScaler 非 BLX（MPX、または VPX、SDX ADC）クラスタとは異なります。

> 重要
>
> 表の「Node-level」エントリは、機能が個々のクラスターノードでのみサポートされていることを示します。

NetScaler の機能	NetScaler 12.1	NetScaler 13.0	NetScaler BLX 13.0	NetScaler 13.1	NetScaler BLX 13.1	NetScaler 14.1	NetScaler BLX 14.1
SSL FIPS	番号	番号	番号	番号	番号	番号	番号
SSL 証明書バンドル	番号	番号	番号	番号	番号	番号	番号
SSL インターセプト	番号	番号	番号	番号	番号	番号	番号

NetScaler の機能	NetScaler 12.1	NetScaler 13.0	NetScaler BLX 13.0	NetScaler 13.1	NetScaler BLX 13.1	NetScaler 14.1	NetScaler BLX 14.1
コンテンツスイッチングアクション	はい	はい	はい	はい	はい	はい	はい
コンテンツスイッチングポリシーのポリシーベースのログ	はい	はい	はい	はい	はい	はい	はい
レート制限	はい	はい	はい	はい	はい	はい	はい
アクション分析	はい	はい	番号	はい	番号	はい	番号
GSLB か	はい	はい	はい	はい	はい	はい	はい
RTSP	はい	はい	はい	はい	はい	はい	はい
DNSSEC	番号	番号	番号	番号	番号	番号	番号
DNS64	番号	番号	番号	番号	番号	番号	番号
FTP	はい	はい	番号	はい	番号	はい	番号
TFTP	はい	はい	はい	はい	はい	はい	はい
接続ミラーリング	番号	番号	番号	番号	番号	番号	番号
統合キャッシング	Node-Level	Node-Level	番号	Node-Level	番号	Node-Level	番号
大容量共有キャッシュ	Node-Level	Node-Level	番号	Node-Level	番号	Node-Level	番号
フロントエンドの最適化	Node-Level	Node-Level	番号	Node-Level	番号	Node-Level	番号
Application firewall	はい	はい	番号	はい	番号	はい	番号
HTTP サービス拒否保護 (HDOSP)	廃止済み	廃止済み	廃止済み	削除	廃止済み	削除	廃止済み

NetScaler の機能	NetScaler 12.1	NetScaler 13.0	NetScaler BLX 13.0	NetScaler 13.1	NetScaler BLX 13.1	NetScaler 14.1	NetScaler BLX 14.1
プライオリティキューイング (PQ)	Node-Level	Node-Level	廃止済み	削除	廃止済み	削除	廃止済み
Sure connect (SC)	Node-Level	Node-Level	廃止済み	削除	廃止済み	削除	廃止済み
AppQoE	はい	はい	番号	はい	番号	はい	番号
サージ保護	Node-Level	Node-Level	はい	Node-Level	はい	Node-Level	はい
MPTCP	はい	はい	番号	はい	番号	はい	番号
ストライピングされた SNIP	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。
MSR	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。
IS-IS (IPv4 および IPv6)	はい	はい	番号	はい	番号	はい	番号

NetScaler の機能	NetScaler 12.1	NetScaler 13.0	NetScaler BLX 13.0	NetScaler 13.1	NetScaler BLX 13.1	NetScaler 14.1	NetScaler BLX 14.1
ジャンボフレーム	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	番号	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	番号	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	番号
IP-IP トンネリング	はい	はい	番号	はい	番号	はい	番号
リンク負荷分散	はい	はい	はい	はい	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。
FIS（フェールオーバーインターフェイスセット）	はい	はい	番号	はい	番号	はい	番号
リンク冗長性（LR）	はい	はい	番号	はい	番号	はい	番号
NAT46	番号	はい	はい	はい	はい	はい	はい
NAT64	番号	はい	はい	はい	はい	はい	はい
RNAT6	はい	はい	はい	はい	はい	はい	はい
LSN/CGNAT	はい	はい	番号	はい	番号	はい	番号
IPv6 レディロゴ	はい	はい	番号	はい	番号	はい	番号

NetScaler の機能	NetScaler 12.1	NetScaler 13.0	NetScaler BLX 13.0	NetScaler 13.1	NetScaler BLX 13.1	NetScaler 14.1	NetScaler BLX 14.1
トラフィックドメイン	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	番号	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	番号	はい。注: L2 クラスターでサポートされています。L3 クラスタではサポートされません。	番号
ルートモニタ	はい	はい	はい	はい	はい	はい	はい
GRE トンネリング (CB)	番号	番号	番号	番号	番号	番号	番号
レイヤ 2 モード	はい	はい	番号	はい	番号	はい	番号
ネットプロファイル	はい	はい	番号	はい	番号	はい	番号
HTTPS コールアウト	はい	はい	はい	はい	はい	はい	はい
AAA-TM	はい	はい	番号	はい	番号	はい	番号
AppFlow	Node-Level	Node-Level	番号	Node-Level	番号	Node-Level	番号
Web Insight	はい	はい	番号	はい	番号	はい	番号
HDX Insight	はい	はい	番号	はい	番号	はい	番号
VMAC/VRRP	はい	はい	番号	はい	番号	はい	番号
NetScaler Push	番号	番号	番号	番号	番号	番号	番号
ステートフル接続フェールオーバー	番号	番号	番号	番号	番号	番号	番号
グレースフルシャットダウン	はい	はい	はい	はい	はい	はい	はい

NetScaler の機能	NetScaler 12.1	NetScaler 13.0	NetScaler BLX 13.0	NetScaler 13.1	NetScaler BLX 13.1	NetScaler 14.1	NetScaler BLX 14.1
DBS Autoscale	番号	はい	はい	はい	はい	はい	はい
TOS 使って DSR	番号	番号	はい	はい	はい	はい	はい
より細かいスタートアップ RR 制御	Node-Level	Node-Level	番号	Node-Level	番号	Node-Level	番号
XML XSM	番号	番号	番号	番号	番号	番号	番号
DHCP RA	番号	番号	番号	はい	番号	はい	番号
ブリッジグループ	はい	はい	番号	はい	番号	はい	番号
ネットワークブリッジ	番号	番号	番号	番号	番号	番号	番号
NetScaler (WIonNS) 上の Web インターフェイス	はい	はい	番号	はい	番号	はい	番号
EdgeSight Monitor-ing	廃止済み	廃止済み	番号	廃止済み	番号	廃止済み	番号
メトリックス表-ローカル	番号	番号	番号	番号	番号	番号	番号
DNS キャッシング	Node-Level	Node-Level	Node-Level	Node-Level	Node-Level	Node-Level	Node-Level
Call Home	Node-Level	Node-Level	番号	Node-Level	番号	Node-Level	番号
NetScaler Gateway ICA プロキシモード	はい	はい	番号	はい	番号	はい	番号

NetScaler の機能	NetScaler 12.1	NetScaler 13.0	NetScaler BLX 13.0	NetScaler 13.1	NetScaler BLX 13.1	NetScaler 14.1	NetScaler BLX 14.1
NetScaler Gateway (SSL VPN /フル VPN およびクライアントレス VPN)	Node-Level	Node-Level	番号	Node-Level	番号	Node-Level	番号
Citrix Cloud-Bridge Connec-tor	はい	はい	番号	はい	番号	はい	番号
ポリシーベースのルーティング (PBR/PBR6)	はい	はい	番号	はい	番号	はい	番号
LLB 仮想サーバをネクストホップとする IPv4 ポリシーベースルーティング（PBR）	番号	はい	番号	はい	番号	はい	番号
LLB 仮想サーバをネクストホップとする IPv6 ポリシーベースルーティング（PBR6）	番号	番号	番号	番号	番号	番号	番号
購読者認識	番号	番号	番号	番号	番号	番号	番号

NetScaler の機能	NetScaler 12.1	NetScaler 13.0	NetScaler BLX 13.0	NetScaler 13.1	NetScaler BLX 13.1	NetScaler 14.1	NetScaler BLX 14.1
動的ルーティング	はい v6 プロトコル (ospfv3、riPNG、ISIS6、BGP6) サポートあり	はい v6 プロトコル (ospfv3、riPNG、ISIS6、BGP6) サポートあり	はい	はい v6 プロトコル (ospfv3、riPNG、ISIS6、BGP6) サポートあり	はい	はい v6 プロトコル (ospfv3、riPNG、ISIS6、BGP6) サポートあり	はい
SYSLOG-TCP、Syslog サーバの負荷分散、SNIP サポート、および Syslog の FQDN サポート	はい	はい	はい	はい	はい	はい	はい
ボット管理	番号	はい	番号	はい	番号	はい	番号
VXLAN	番号	番号	番号	番号	番号	番号	番号
NSVLAN	はい	はい	番号	はい	はい	はい	はい

また、次の NetScaler 構成もサポートされています。

負荷分散、負荷分散の永続性、DNS 負荷分散、SIP、MaxClient、スピルオーバー（接続およびダイナミック）。帯域幅、DataStream、圧縮制御、コンテンツフィルタリング、TCP バッファリング、キャッシュリダイレクト、分散型サービス拒否 (DDoS) に基づくスピルオーバー。クライアント Keep-alive、基本ネットワーク（IPv4 および IPv6）、OSPF（IPv4 および IPv6）、RIP（IPv4 および IPv6）、RIP（IPv4 および IPv6）。VLAN、ICMP、フラグメンテーション、MBF、ACL、簡易 ACL、MSR、パス MTU ディスカバリ、IP-IP、SNMP、ポリシー（クラシックおよびアドバンス）。書き換え、レスポンダー、HTTP コールアウト、Web サーバーログ、監査ログ (NSLOG および syslog)。USIP、ロケーションコマンド、NITRO API、AppExpert、KRPC。

前提条件

August 15, 2023

クラスタに追加する NetScaler ADC アプライアンス（MPX、VPX、SDX ADC、BLX）は、次の前提条件を満たしている必要があります。

- すべてのアプライアンスのソフトウェアバージョンとビルドが同じである必要があります。
- すべてのアプライアンスは、同じプラットフォームタイプである必要があります。つまり、クラスターには、すべてのハードウェアアプライアンス（NetScaler MPX）またはすべての NetScaler VPX アプライアンス、すべての NetScaler BLX アプライアンス、またはすべての NetScaler SDX ADC インスタンスのいずれかが必要です。

  注：

  - ハードウェアアプライアンス (MPX) のクラスタの場合、アプライアンスは同じモデルタイプである必要があります。
  - 異機種混在クラスタを形成するには、すべてのアプライアンスが MPX プラットフォームタイプである必要があります。
  - 仮想アプライアンスのクラスタ（VPX）の場合、アプライアンスは XenServer、Hyper-V、VMware ESX、KVM のハイパーバイザーに展開する必要があります。
  - SDX NetScaler ADC インスタンスのクラスターの設定については、NetScaler ADC インスタンスのクラスターの設定を参照してください。
  - ジャンボフレームは、NetScaler SDX インスタンスで構成される NetScaler ADC クラスターでサポートされます。
  - SDX インスタンスの L3 クラスタを作成できます。
  - NetScaler BLX クラスターの設定の詳細については、NetScaler BLX クラスターを参照してください。
- アプライアンスは異なるネットワークに属することができます。
- 最初に構成され、共通のクライアント側およびサーバー側のネットワークに接続されている。
- 大規模な構成の仮想アプライアンス（NetScaler VPX、NetScaler BLX、または NetScaler SDX ADC インスタンス）のクラスターでは、クラスターの各ノードに 6 GB の RAM を使用することをお勧めします。

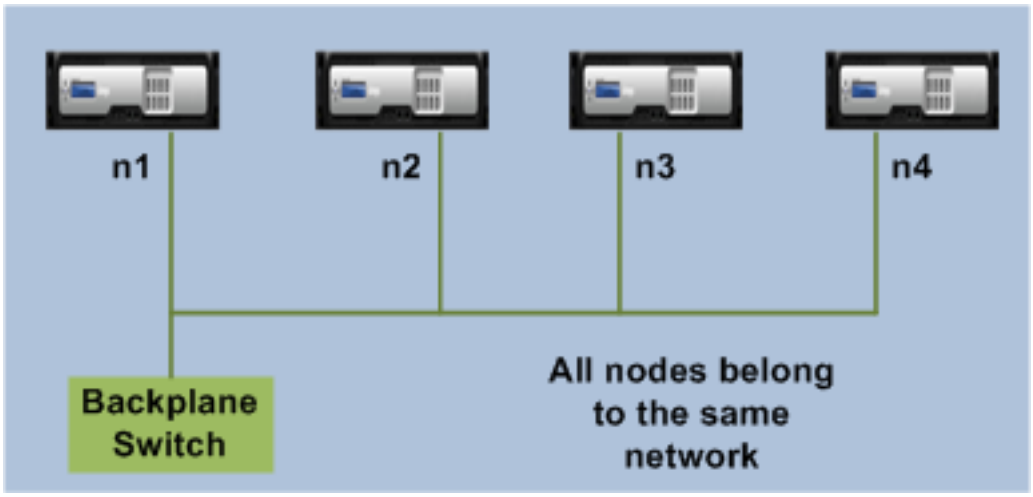
クラスターの概要

August 15, 2023

NetScaler クラスターは、NetScaler アプライアンスをグループ化することによって形成されます。クラスターを追加する NetScaler アプライアンスのネットワークロケーションに応じて、次のクラスター設定に注意する必要があります。

> 注
>
> 特に指定のない限り、クラスターの機能と構成は L2 クラスターと L3 クラスターで同じです。

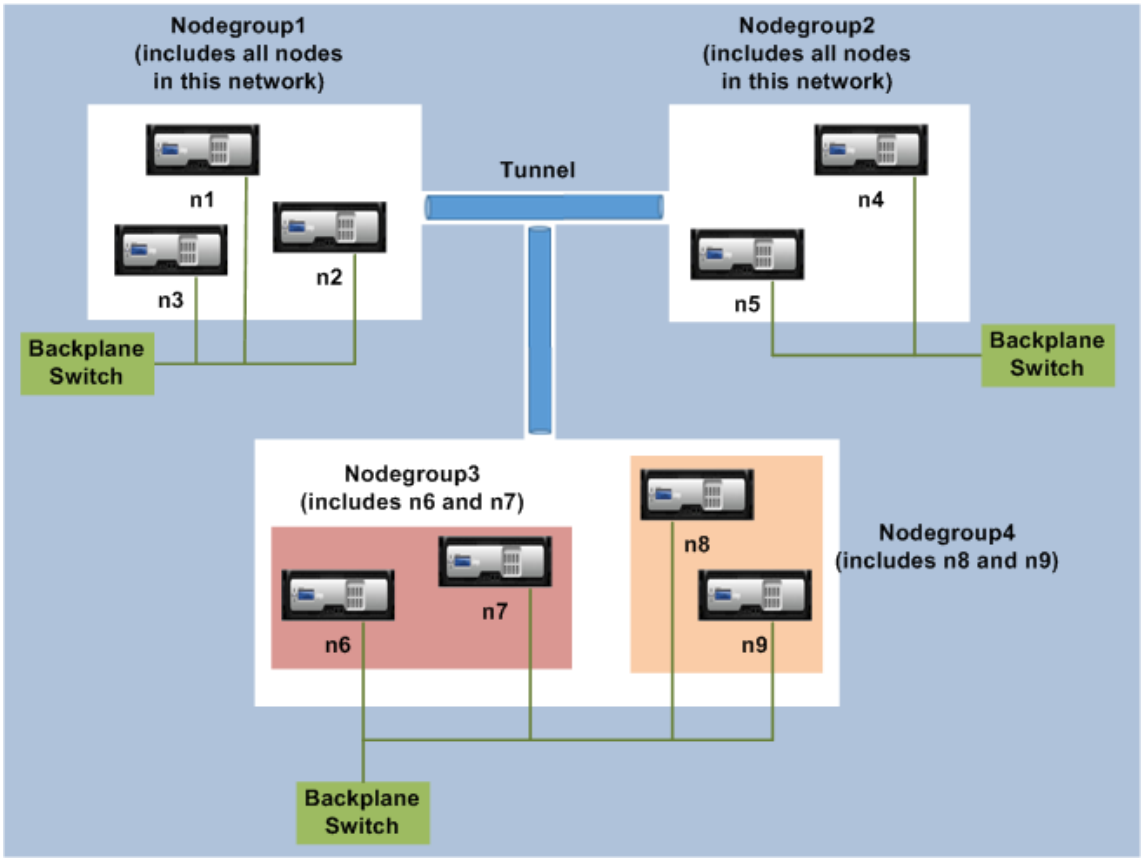
- **L2** クラスタ**:** このクラスタ展開では、すべてのクラスタノードが同じネットワークに属します。



- **L3** クラスタ **(**「**INC** モードのクラスタ」とも呼ばれます**):** このクラスタ展開では、クラスタノードは異なるネットワークに属することができます。特定のネットワークのクラスターノードは、そのネットワークのノードのみを含むノードグループにグループ化する必要があります。次の図から、ノード n1、n2、n3 が同じネットワークにあり、Nodegroup1 にグループ化されていることがわかります。

  同様に、ノードグループ 2 にグループ化されているノード n4 と n5 の場合も同様です。3 番目のネットワークには、2 つのノードグループがあります。ノードグループ 3 には n6 と n7 が含まれ、ノードグループ 4 には n8 と n9 が含まれます。

  > 注
  >
  > NetScaler 11.0 以降でサポートされています。

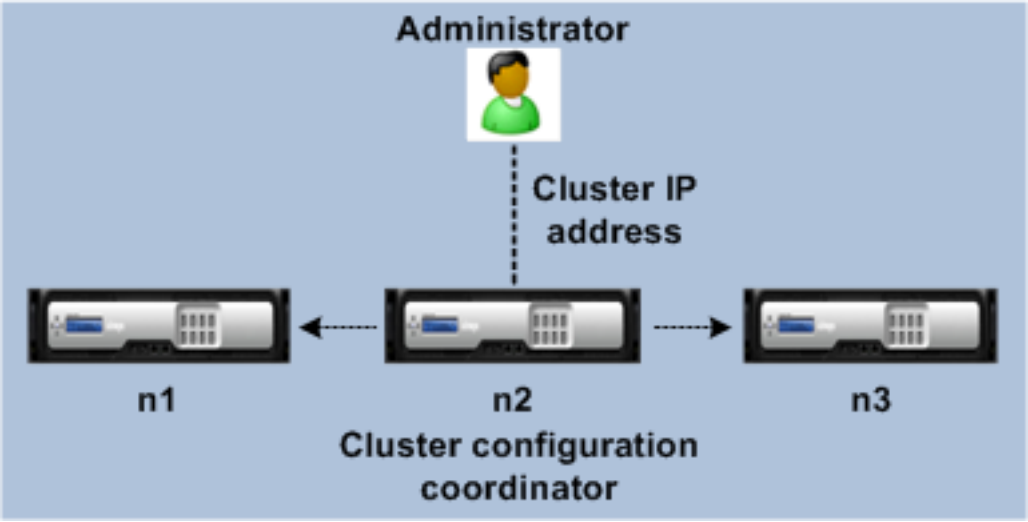


- 同期状態**:show cluster** コマンドは、クラスタノードのステータスを表示します。**show cluster node** コマンドの同期状態は次のとおりです。
  - 有効**:** この状態は、そのノードが他のノード から構成を同期できることを示しています。
  - 進行中**:** これはノードが他のノードの設定を同期しているときに表示される一時的な状態です。
  - 成功**:** この状態は、このノードで最後に発生した同期のステータスを表します。

クラスターノード間の同期

August 15, 2023

NetScaler クラスターのすべての構成は、クラスターの管理アドレスであるクラスター IP アドレスで実行されます。クラスターノードは、次の図に示すように、クラスター構成コーディネーター (CCO) と呼ばれるクラスター IP アドレスを所有しています。



CCO で使用可能な構成は、自動的に他のクラスタノードに伝達されるため、すべてのクラスタノードは同じ構成になります。

- NetScaler では、NSIP アドレスを介して個々のクラスターノードで実行できる構成はごくわずかです。このような場合は、クラスター内のすべてのノードで設定の一貫性を手動で確認する必要があります。これらの構成は、他のクラスタノードに伝播されません。各クラスタノードでサポートされる操作の詳細については、「個々のクラスタノードでサポートされる操作」を参照してください。
- クラスター IP アドレスで実行された場合、次のコマンドは他のクラスターノードに伝達されません。
  - **shutdown.** 構成コーディネーターのみを停止します。
  - **reboot.** 構成コーディネーターのみを再起動します。
  - **rm** クラスタインスタンス。コマンドを実行しているノードからクラスターインスタンスを削除します。
- コマンドを他のクラスターノードに伝播するには:
  - クォーラムはクラスターインスタンスで設定する必要があります。
  - クラスターが動作するためには、(n/2 + 1) のクラスターノードを含むほとんどのクラスタークォーラムがアクティブである必要があります。
  - マジョリティルール (n/2 + 1) が緩和されると、クラスターは最小数のノードで実行できます。

ノードをクラスタに追加すると、CCO で使用可能な構成とファイル (SSL 証明書、ライセンス、DNS など) は、新しく追加されたクラスタノードに同期されます。意図的に無効化された、または障害が発生した既存のクラスターノードが再び追加されると、クラスターはそのノードで使用可能な構成と CCO で使用可能な構成を比較します。構成に不一致がある場合、ノードは次のいずれかを使用して同期されます。

- 完全同期。構成間の差が 255 コマンドを超える場合、CCO のすべての構成がクラスタに再参加するノードに適用されます。同期中、ノードは操作不能のままです。
- インクリメンタル同期。構成間の差が 255 コマンド以下の場合、使用できない構成のみがクラスターに再参加するノードに適用されます。ノードの動作状態は影響を受けません。

> 注
>
> 構成とファイルを手動で同期することもできます。詳細については、「クラスタ構成の同期」および「[クラスタファイルの同期](/ja-jp/citrix-adc/13-1/clustering/cluster-managing/cluster-file-sync.html)」を参照してください。

ストライピング、部分的なストライピング、およびスポット設定

January 9, 2024

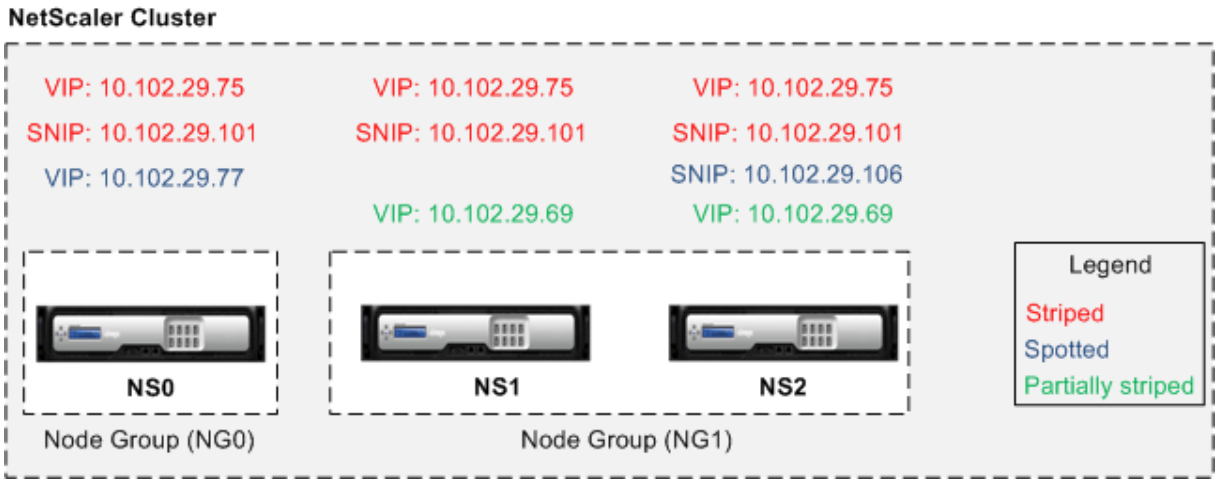
コマンド伝播により、クラスター内のすべてのノードは同じ構成になります。ただし、一部の構成を特定のクラスターノードでのみ使用できるようにしたい場合があります。構成を使用できるノードを制限することはできませんが、構成がアクティブなノードは指定できます。

たとえば、次の作業を行えます。

- 1 つのノードでのみアクティブになるように SNIP アドレスを定義するか、
- すべてのノードでアクティブになる SNIP アドレスを定義するか、
- 1 つのノードでのみアクティブになるように VIP アドレスを定義するか、
- すべてのノードでアクティブになるように VIP アドレスを定義するか、
- 3 ノードクラスタの 2 つのノードでのみ有効な VIP アドレスを定義する

構成がアクティブなノードの数に応じて、クラスター構成はストライピング構成、部分ストライピング構成、またはスポット構成と呼ばれます。

図 1: ストライプ構成、部分ストライプ構成、スポット構成の 3 ノードクラスタ



次の表は、構成の種類に関する詳細を示しています。

構成タイプ	アクティブオン	適用対象	構成
ストライプ構成	すべてのクラスターノード	すべてのエントリー	エンティティをストライプ化するのに特別な設定は必要ありません。デフォルトでは、クラスター IP アドレスに定義されたすべてのエンティティは、すべてのクラスターノードでストライピングされます。
部分的にストライプされた構成	クラスタノードのサブセット	「クラスターノードグループ」を参照	部分的にストライプするエンティティをノードグループにバインドします。構成は、ノードグループに属するクラスターノードでのみアクティブです。

構成タイプ	アクティブオン	適用対象	構成
スポット構成	単一クラスターノード	「スポット設定のリスト」を参照してください。	スポット構成は、2 つの方法のいずれかを使用して定義できます。**SNIP** アドレス SNIP アドレスを作成するときは、SNIP アドレスをアクティブにするノードを所有者ノードとして指定します。例`add ns ip 10.102.29.106 255.255.255.0 -type SNIP -ownerNode 2` (ノード NS2 ID を 2 と仮定)。注: スポッティング SNIP アドレスの所有権は実行時に変更できません。所有権を変更するには、まず SNIP アドレスを削除し、新しい所有者を指定して再度追加する必要があります。ノードグループにバインドできるエンティティ。エンティティを単一メンバーのノードグループにバインドします。

注

- USIP を無効にする場合は、スポッティング SNIP アドレスを使用することをお勧めします。ストライプ SNIP アドレスを使用できるのは、IP アドレスが不足している場合だけです。ストライピング IP アドレスを使用すると、ARP 解決のために同じサブネットにスポッティング IP アドレスが存在しない場合、ARP フラックスの問題が発生する可能性があります。
- USIP を有効にする場合は、ストライプ SNIP アドレスを、サーバーが開始するトラフィックのゲートウェイとして使用することをお勧めします。

スポット設定のリスト

- IPv4 アドレス
- IPv6 アドレス
- ARP エントリー
- インターフェイス設定
- IPv6 スタティックルート (ルート 6)
- IP トンネル
- IPv6 トンネル
- リバースネットワークアドレス変換 (RNAT)
- IPv4 (PBR) のポリシーベースルート
- IPv6 (PBR6) のポリシーベースルート
- クラスターラグ
- 近隣探索プロトコル (ND6) エントリ
- フェイルオーバーインターフェイスセット (FIS)
- リンクセット
- SNMP MIB/エンジン ID
- 直径
- Citrix ADC VPX パラメータ
- ホスト名

ストライプ **IP** の **ARP** 所有者サポート

クラスターセットアップでは、ストライプ IP の ARP 要求に応答するように特定のノードを設定できます。設定されたノードは ARP トラフィックに応答します。

「IP の追加、設定、設定解除」コマンドに新しいパラメータ「ARPowner」が導入されました。

CLI を使用してノードの ARP 所有者を有効にする。

コマンドプロンプトで入力します:

```
add ns ip <ip_address> -arpOwner <node_id>
```

> 注
>
> ARP 所有者パラメータは L2 クラスタでのみサポートされています。

ストライプ **IPv6** アドレスのネイバーディスカバリーオーナーサポート

クラスタセットアップでは、特定のノードをストライピング IPv6 アドレスのネイバー探索 (ND) 所有者として設定し、リンク層アドレスを決定できます。クライアントは、クラスターセットアップ内のすべてのノードにネイバー勧

誘 (NS) メッセージを送信します。ND 所有者は、ストライプされた IPv6 アドレスのリンク層アドレスを含むネイバーアドバタイズメント（NA）メッセージで応答し、トラフィックを処理します。

CLI を使用してノードの **ND** オーナーを有効にするには

コマンドプロンプトで入力します：

```
add ns ip6 <IPv6Address> -ndOwner <node id>

set  ns ip6 <IPv6Address> -ndOwner <node id>
<!--NeedCopy-->
```

例：

```
add ns ip6 2001::21/64 -ndOwner 1

set ns ip6 2001::21/64 -ndOwner 1
<!--NeedCopy-->
```

GUI を使用してノードの **ND** オーナーを有効にするには

1. [システム**] > [**ネットワーク**] > [IP**] に移動します。
2.「**IPs** 」ページで、「**IPv6** 」タブに移動し、「追加」をクリックします。
3.「**IPv6** の作成」ページで、「クラスタ内の nDow ナー」ドロップダウンメニューに表示されているノード ID の **1** つを選択します。

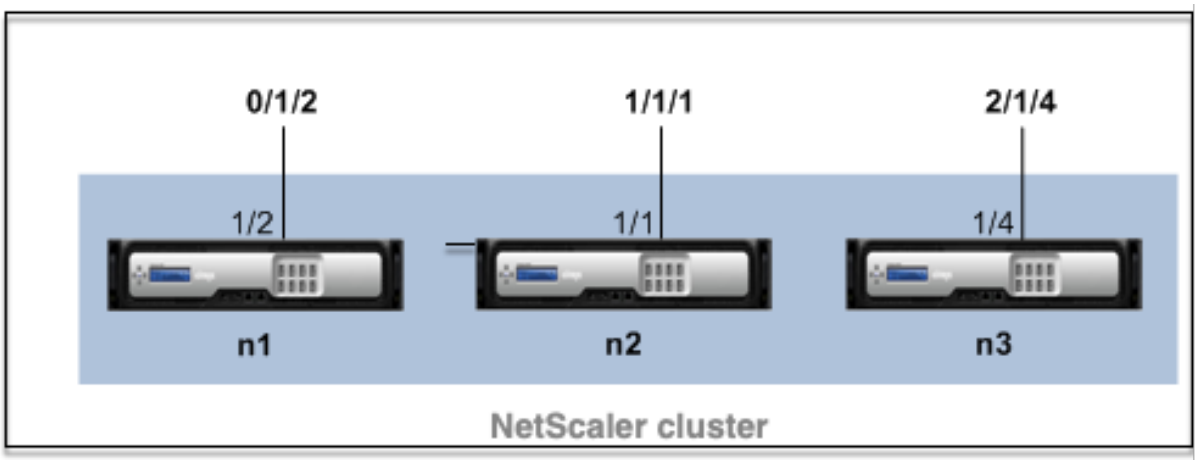
> 注
>
> ND 所有者パラメーターは、L2 クラスターでのみサポートされます。

クラスターセットアップでの通信

August 15, 2023

クラスターに追加される NetScaler アプライアンスのインターフェースには、ノード ID のプレフィックスが付きます。インターフェイスが属するクラスターノードを識別するのに役立ちます。そのため、インタフェース識別子 c/u (c はコントローラ番号、u はユニット番号) は n/c/u (n はノード ID) になりました。たとえば、次の図では、ノード n1 のインターフェイス 1/2 は 0/1/2、ノード n2 のインターフェイス 1/1 は 1/1/1、ノード n3 のインターフェイス 1/4 は 2/1/4 として表されます。

図 1：クラスタ内のインタフェース命名規則



- サーバー通信-
  クラスターは、クラスターノードとサーバー側の接続デバイス間の物理接続を介してサーバーと通信します。これらの物理接続の論理的なグループ化は、サーバーデータプレーンと呼ばれます。
- クライアント通信-クラスターは、クラスターノードとクライアント側の接続デバイス間の物理接続を介してクライアントと通信します。これらの物理接続を論理的にグループ化したものをクライアントデータプレーンと呼びます。
- ノード間通信-クラスターノードは相互に通信することもできます。通信の方法は、ノードが同じネットワーク上に存在するか、ネットワークをまたがっているかによって異なります。
  - 同じネットワーク内のクラスターノードは、クラスターバックプレーンを使用して相互に通信します。バックプレーンは、各ノードの 1 つのインターフェイスがクラスタバックプレーンスイッチと呼ばれる共通のスイッチに接続されているインターフェイスのセットです。ノード間通信で使用されるバックプレーンを通過するトラフィックの種類には、次のようなものがあります。
    * ノード間メッセージング (NNM)
    * ステアリングトラフィック
    * 設定の伝播と同期
  - クラスタの各ノードは、特別な MAC クラスタバックプレーンスイッチアドレスを使用して、バックプレーンを介して他のノードと通信します。クラススペシャル MAC の形式は次のとおりです。ここで `0x02 0x00 0x6F <cluster_id> <node_id> <reserved>`)、`cluster_id` はクラスターインスタンス ID、`node_id` はクラスターに追加される NetScaler アプライアンスのノード番号です。

次の図は、L2 クラスタと L3 クラスタの通信インターフェイスを示しています。

図 2: クラスタ通信インタフェース-L2 クラスタ

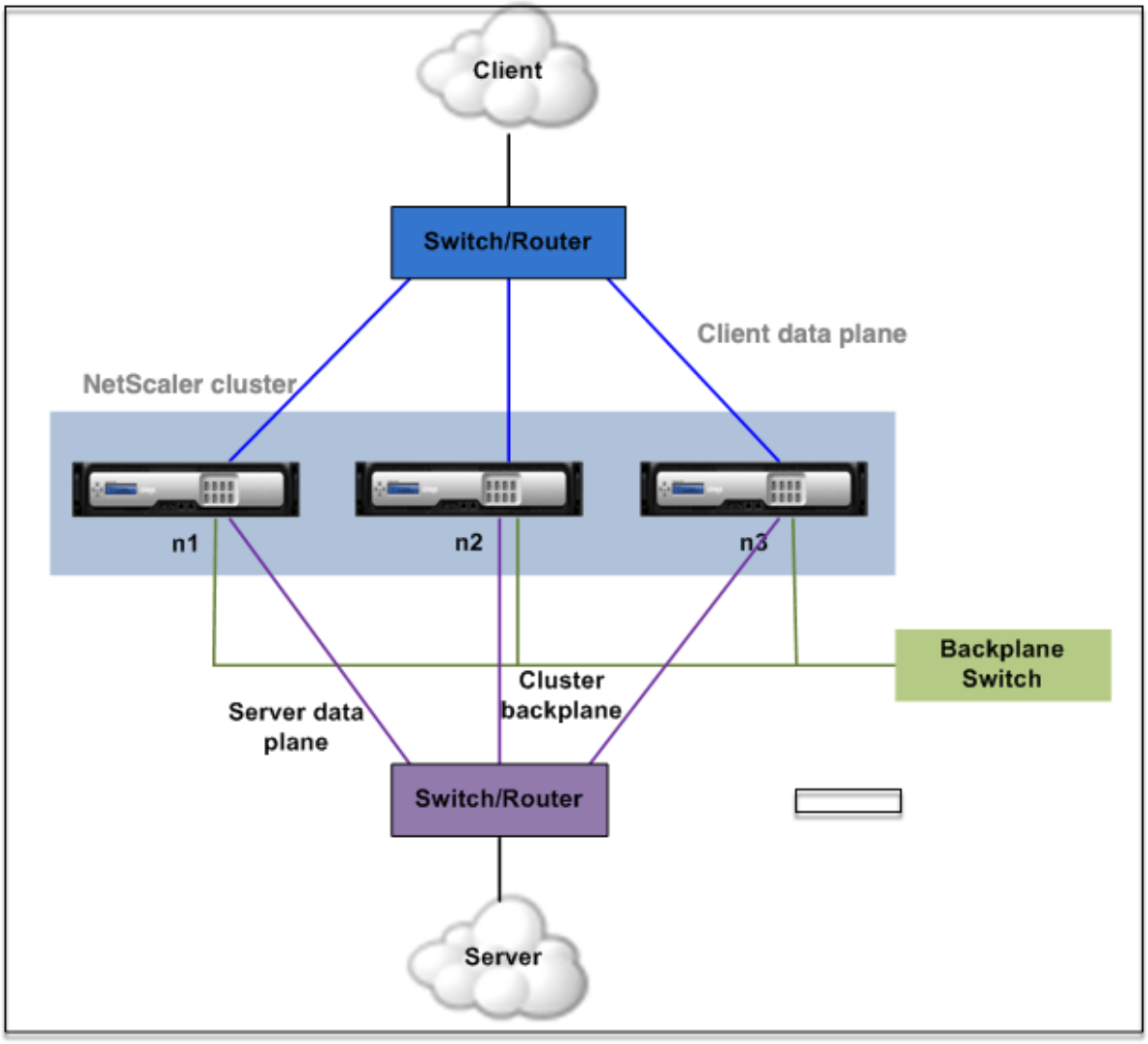
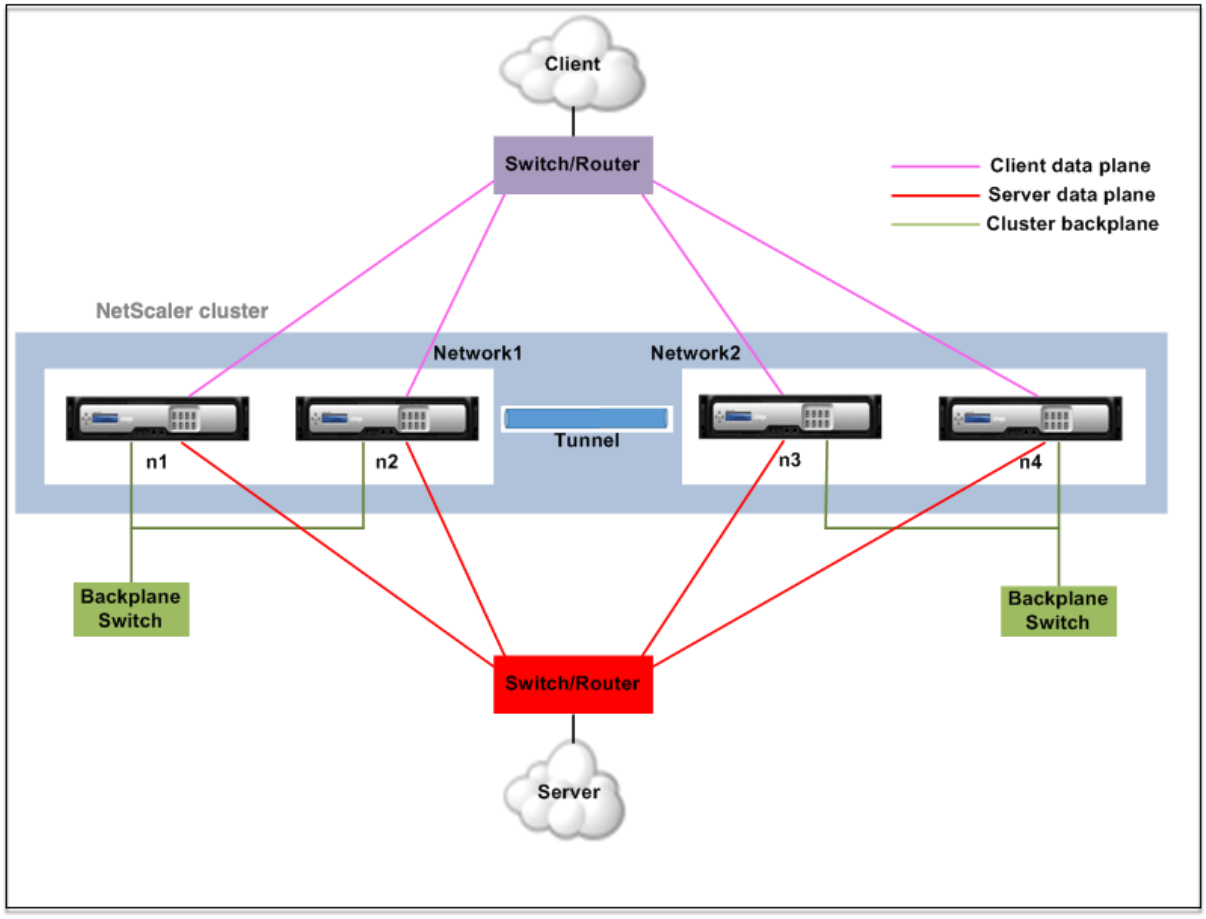


図 3: クラスタ通信インタフェース-L3 クラスタ



クラスターセットアップでのトラフィック分散

August 15, 2023

クラスター設定では、外部ネットワークは NetScaler アプライアンスのコレクションを単一のエンティティと見なします。したがって、クラスターは、トラフィックを受信する必要がある単一のノードを選択する必要があります。クラスタは、Equal Cost Multiple Path (ECMP) またはクラスタリンクアグリゲーションのトラフィック分散メカニズムを使用してこの選択を行います。選択されたノードをフロー受信機と呼びます。

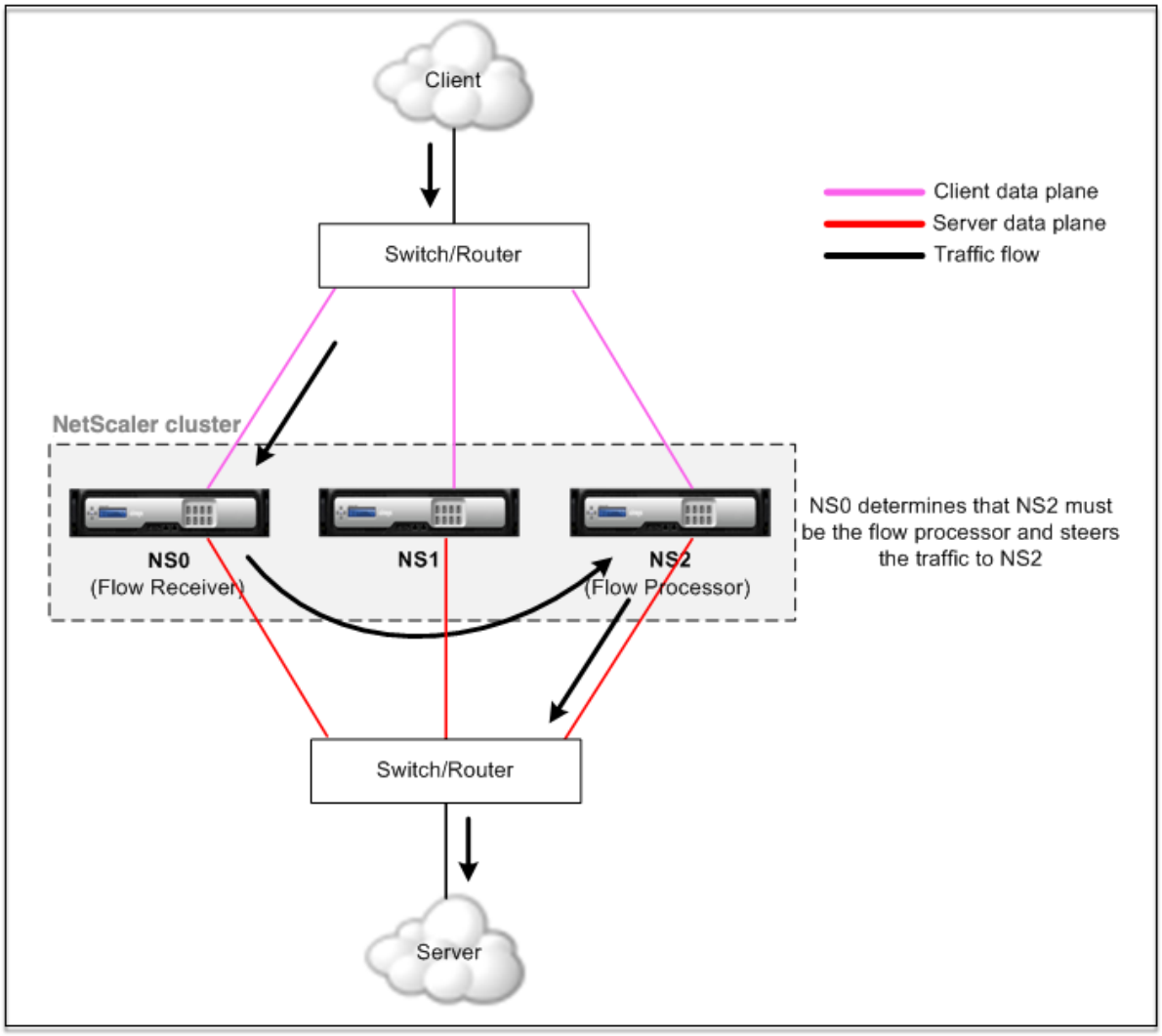
> 注
>
> L3 クラスター (異なるネットワークにまたがるノード) では、ECMP トラフィック分散のみを使用できます。

フローレシーバがトラフィックを取得し、内部クラスタロジックを使用してトラフィックを処理する必要があるノードを決定します。このノードはフロープロセッサと呼ばれます。フローレシーバとフロープロセッサが同じネットワーク上にある場合、フローレシーバはバックプレーンを介してフロープロセッサにトラフィックを誘導します。フローレシーバとフロープロセッサが異なるネットワーク上にある場合、トラフィックはトンネルを通過します。

注

- フローレシーバとフロープロセッサは、トラフィックを処理できるノードでなければなりません。
- NetScaler 11 以降では、クラスタバックプレーンでステアリングを無効にできます。詳細については、クラスタバックプレーンでのステアリングの無効化を参照してください。

図 1: クラスタ内のトラフィック分散



前の図は、クラスターを流れるクライアント要求を示しています。クライアントは、仮想 IP (VIP) アドレスにリクエストを送信します。クライアントデータプレーンに設定されたトラフィック分散メカニズムは、クラスターノードの 1 つをフローレシーバーとして選択します。フローレシーバは、トラフィックを受信し、トラフィックを処理する必要があるノードを決定し、そのノードに要求を操縦します（フローレシーバが自身をフロープロセッサとして選択しない限り）。

フロープロセッサは、サーバとの接続を確立します。サーバは要求を処理し、要求をサーバに送信したサブネット IP（SNIP）アドレスに応答を送信します。

- SNIP アドレスがストライピングされた IP アドレスまたは部分的にストライピングされた IP アドレスの場合、サーバデータプレーンに設定されたトラフィック分散メカニズムは、クラスタノードの 1 つをフローレシーバとして選択します。フローレシーバはトラフィックを受信し、フロープロセッサを決定し、クラスタバックプレーンを介して要求をフロープロセッサに転送します。
- SNIP アドレスがスポット IP アドレスの場合、SNIP アドレスを所有するノードはサーバーから応答を受け取ります。

非対称クラスタトポロジ（すべてのクラスタノードが外部スイッチに接続されていない）では、リンクセットを排他的に使用するか、ECMP またはクラスタリンク集約と組み合わせて使用する必要があります。詳細については、「リンクセットの使用」を参照してください。

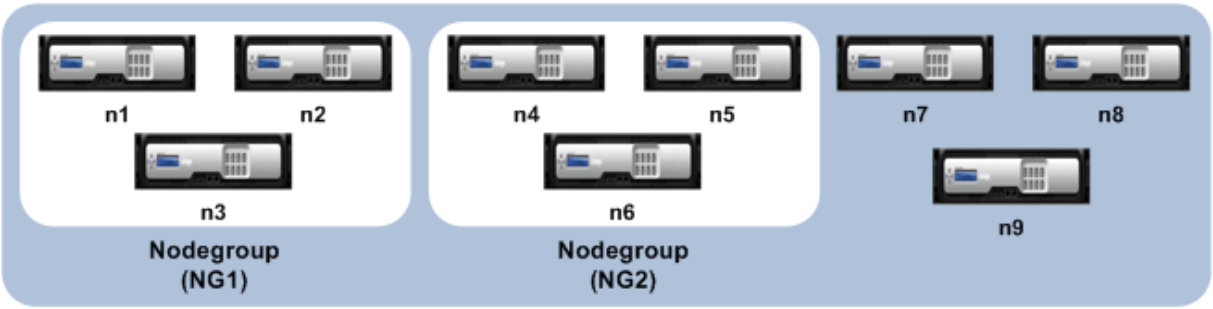
クラスターノードグループ

August 15, 2023

> 注
>
> ノードグループは NetScaler 10.1 以降でサポートされています。

名前が示すように、クラスターノードグループはクラスターノードのグループです。

図 1：ノードグループを含む NetScaler クラスター



上の図は、それぞれ 3 つのクラスターノードを含むノードグループ NG1 と NG2 を含むクラスターを示しています。クラスタには、どのノードグループにも属さない 3 つのノードもあります。

ノードグループは次のように構成できます。

- スポットおよび部分的にストライプされた構成を定義する。詳細については、「スポッティングされた構成と部分ストライプ構成のノードグループ」を参照してください。
- ノードグループの冗長性を構成します。詳細については、「ノードグループの冗長性の設定」を参照してください。
  注：NetScaler 10.5 ビルド 52.1115.e 以降でサポートされています。
- L3 クラスタ (INC モードではクラスタとも呼ばれる) を定義します。L3 クラスターでは、クラスターノードは異なるネットワークに属していてもかまいません。ネットワークに属するノードは 1 つのノードグループにグ

ループ化する必要があります。たとえば、n1、n2、n3 がネットワーク 1 にあり、n4、n5、n6 がネットワーク 2 にある場合、NG1 はネットワーク 1 のノードを含み、NG2 はネットワーク 2 のノードを含める必要があります。L3 クラスターの設定については、NetScaler ADC クラスターの作成を参照してください。

> 注
>
> - NetScaler 11 以降でサポートされています。
> - ノードグループの前述の機能は相互に排他的です。つまり、ノードグループが提供できるのは前述の機能の 1 つだけです。

クラスターとノードの状態

August 15, 2023

クラスターが機能するには、ほとんどのノード (n/2 + 1) が運用上アクティブ (動作状態が ACTIVE) である必要があります。

> 重要
>
> NetScaler リリース 10.5 以降、大多数の基準が満たされない場合でもクラスターが機能するように構成できます。この構成は、クラスターの作成時に実行する必要があります。

クラスタノードの状態の詳細については、「クラスタノードの状態」を参照してください。

クラスター内のルーティング

August 15, 2023

クラスター内のルーティングは、スタンドアロンシステムでのルーティングとほとんど同じように機能します。注意すべき点がいくつかあります。

- すべてのルーティング構成はクラスター IP アドレスから実行する必要があり、構成は他のクラスターノードに伝達されます。
- ルートは、アップストリームルータがサポートする ECMP ルートの最大数に制限されます。
- ノード固有のルーティング設定は、次のように owner-node 引数を使用して実行する必要があります。

```
1  router ospf
2    owner-node 0
3      ospf router-id 97.131.0.1
4    exit-owner-node
```



```
    !
<!--NeedCopy-->
```

次のコマンドは、VTYSH のすべてのノードの統合クラスタ構成を表示します。

```
show cluster-config
```

次のコマンドは、各ノードのクラスターステータスを表示します。

```
show cluser node
```

L2 クラスタでの IPv4 ルーティング

次のセクションには、L2 クラスタで IPv4 OSPF と BGP ルーティングを設定するのに役立つ設定例が含まれています。

スポット SNIP アドレスの追加と動的ルーティングの有効化

次の設定では、OSPF と BGP ルーティングが有効になっています。また、スポッティング SNIP アドレスが追加され、これらの SNIP アドレスで動的ルーティングが有効になります。

```
en ns fea ospf bgp
add vlan 10
add ns ip 10.10.10.1 255.255.255.0 -dynamicrouting enabled -ownernode 1
add ns ip 10.10.10.2 255.255.255.0 -dynamicrouting enabled -ownernode 2
add ns ip 10.10.10.3 255.255.255.0 -dynamicrouting enabled -ownernode 3
bind vlan 10 -ipaddress 10.10.10.1 255.255.255.0
<!--NeedCopy-->
```

VTYSH IPv4 OSPF configuration

L2 クラスタで IPv4 OSPF を設定するには、次のことが必要です。

- 優先度をゼロに設定します。
- ルータ ID をスポット設定として設定します。

> 注
>
> L2 クラスタの OSPF 設定ガイドラインは OSPFv3 にも適用されます。

次の設定例では、IPv4 OSPF が設定されています。

```
       interface vlan10
        IP OSPF PRIORITY 0
        !
       router ospf
           owner-node 1
```



```
6               ospf router-id 97.131.0.1
7             exit-owner-node
8             owner-node 2
9               ospf router-id 97.131.0.2
10            exit-owner-node
11            owner-node 3
12              ospf router-id 97.131.0.3
13            exit-owner-node
14        network 10.10.10.0/24 area 0
15        redistribute kernel
16         !
17 <!--NeedCopy-->
```

VTYSH IPv4 BGP configuration

次の VTYSH サンプル設定では、IPv4 BGP が設定されています。

```
1      router bgp 100
2        neighbor 10.10.10.10 remote-as 200
3       owner-node 1
4        neighbor 10.10.10.10 update-source 10.10.10.1
5       exit-owner-node
6       owner-node 2
7        neighbor 10.10.10.10 update-source 10.10.10.2
8       exit-owner-node
9       owner-node 3
10       neighbor 10.10.10.10 update-source 10.10.10.3
11      exit-owner-node
12     redistribute kernel
13     !
14 <!--NeedCopy-->
```

> 注
>
> 次の設定では owner-node 引数を指定したネイバーごとに update-source コマンドを使用して、適切なソース IP で接続します。

L2 クラスタでの IPv6 ルーティング

次のセクションには、L2 クラスタで IPv6 OSPF と BGP ルーティングを設定するのに役立つ設定例が含まれています。

IPv6 ルーティングを有効にする

L2 クラスタで IPv6 ルーティングを設定する前に、IPv6 機能を有効にする必要があります。

CLI を使用して IPv6 ルーティングを有効にするには、

コマンドプロンプトで入力します。

- `enable ns fea ipv6pt`

スポット付き **SNIP6** アドレスの追加と動的ルーティングの有効化

次の設定では、OSPF と BGP ルーティングが有効になっています。また、スポット付き SNIP6 アドレスが追加され、これらの SNIP6 アドレスで動的ルーティングが有効になります。

```
add ns ip6 3ffa::1/64 -dynamicrouting enabled -ownernode 1
add ns ip6 3ffa::2/64 -dynamicrouting enabled -ownernode 2
add ns ip6 3ffa::3/64 -dynamicrouting enabled -ownernode 3
add vlan 10
bind vlan 10 -ipaddress 3ffa::1/64
<!--NeedCopy-->
```

VTYSH IPv6 BGP configuration

次の VTYSH サンプル設定では、IPv6 BGP が設定されています。

```
  router bgp 100
   neighbor 3ffa::10 remote-as 200
     owner-node 1
      neighbor 3ffa::10 update-source 3ffa::1
     exit-owner-node
     owner-node-2
      neighbor 3ffa::10 update-source 3ffa::2
     exit-owner-node
     owner-node-3
      neighbor 3ffa::10 update-source 3ffa::3
     exit-owner-node
   no neighbor 3ffa::10 activate
   address-family ipv6
    redistribute kernel
    neighbor 3ffa::10 activate
   exit-address-family
  !
<!--NeedCopy-->
```

IPv6 学習ルートのインストール

NetScaler クラスターは、NetScaler クラスタのルーティングテーブルにルートをインストールすると、さまざまなルーティングプロトコルによって学習されたルートを使用できます。

CLI を使用して IPv6 で学習したルートを内部ルーティングテーブルにインストールするには:

コマンドプロンプトで入力します。

- `ns route-install ipv6 bgp`
- `ns route-install ipv6 ospf`
- `ns route-install default`

> 注
>
> - IPv6 ネイバーで IPv4 ルートを交換する必要がある場合は、以前の設定から `no neighbor 3ffa::10 active` VTYSH コマンドを削除する必要があります。
> - 所有者ノードごとに `update-source` VTYSH コマンドを使用して、BGP IPv4 設定で指定されている BGP ピアに接続する際に適切な IPv6 ソース IP を指定する必要があります。

L3 クラスタでのルーティング

L3 クラスターのルーティングは、NetScaler アプライアンスで次の構成が行われた場合にのみ機能します。

- VLAN のダイナミックルーティングを有効にします。

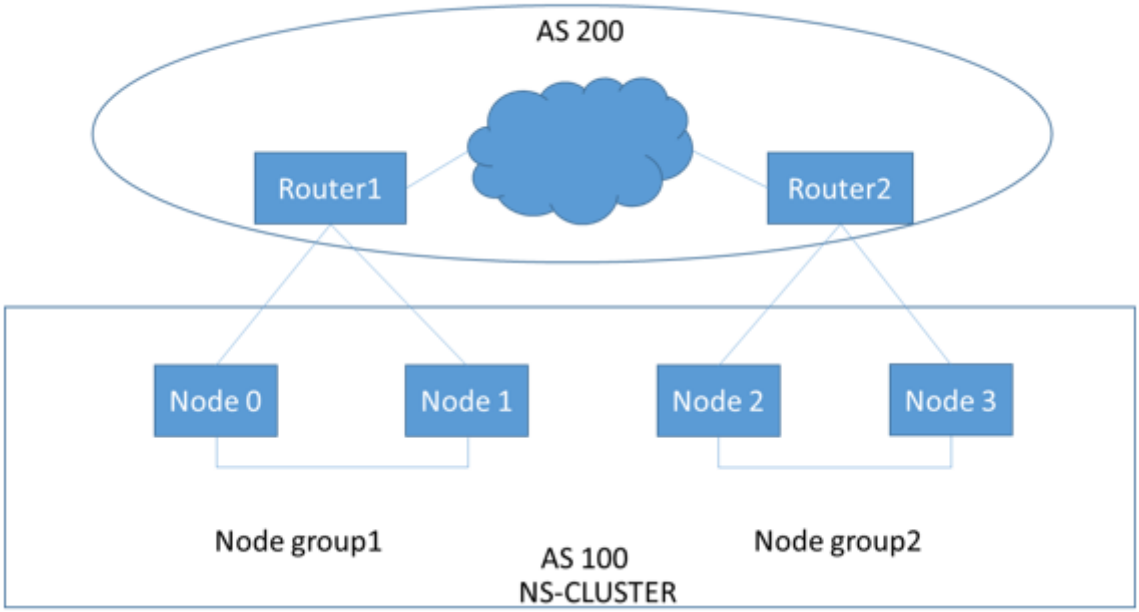
```
set vlan <id> -dynamicrouting enabled
<!--NeedCopy-->
```

- すべてのクラスターノードに到達するには、VIP、CLIP、NetScaler IP（NSIP）をコマンドとともにルーティングプロトコルでアドバタイズする必要があります。`set vlan`

L3 クラスタでの BGP の導入シナリオ

すべてのクラスタノードが AS 100 ネットワークにグループ化され、アップストリームルータが別の AS 200 にある例を考えてみましょう。

次の図は、クラスタセットアップでの AS 100 と AS 200 の導入を示しています。



この展開では、CLIP はアップストリームルータに CCO をアドバタイズします。AS ループが検出されると、一部のクラスターノードはアドバタイズされたトラフィックをドロップします。

この問題を解決するには、ネイバーごとに VTYSH BGP ルータモードで次のコマンドを設定します。

VTYSH コマンドプロンプトで、次のように入力します。

```
neighbor <peer_ip> allowas-in 1
```

ベストプラクティスとして、Citrix では次のいずれかを構成することをお勧めします。

- デフォルトルート、NetScaler IP（NSIP）、クラスターノード上の NSIP サブネットなど、必要なネットワークのみを学習するようにルートマップを構成します。
- クラスター内の CLIP や NetScaler IP（NSIP）など、必要なネットワークのみをアドバタイズするようにアップストリームルートを構成します。

クラスターの **IP** アドレス

August 15, 2023

クラスタ化された NetScaler アプライアンスには、NetScaler NSIP、仮想 IP（VIP）、サブネット IP（SNIP）などの標準タイプの NetScaler 所有 IP アドレスに加えて、クラスター管理 IP（CLIP）アドレスを割り当てることができます。また、IP アドレスをストライプ化したり、スポット化したりすることもできます。

- クリップアドレス。クラスターコーディネーターノード (CCO) が所有する IP アドレス。CLIP アドレスは、クラスタ設定内の異なるノード間でフローティングできます。CLIP をクラスタの別のノードに移動すると、そ

のノードが CCO になります。CCO は、クラスター内の管理タスクを担当する NetScaler アプライアンスです。ネットワーク管理者は CLIP アドレスを使用してクラスタに接続し、統合 GUI へのアクセス、レポート、パケットフローのトレース、ログの収集などの構成および管理タスクを実行します。同じネットワークまたは異なるネットワーク上のクラスタに複数の CLIP アドレスを追加できます。クラスタ IP アドレスを介して CCO で実行された構成のみが、クラスタ内の他のノードに伝達されます。

- ストライプ **IP** アドレス。クラスターのすべてのノードで使用できる論理 IP アドレスで、VIP アドレスでも SNIP アドレスでもかまいません。
- スポッティング **IP** アドレス。論理 IP (できれば SNIP アドレス) は 1 つのノードでのみ使用できます。スポットされた IP アドレスは、そのノードでのみ表示されます。トラフィックステアリングのオーバーヘッドを最小限に抑えるため、Citrix では、サーバーとのバックエンド通信にスポット付き SNIP アドレスを使用することを推奨しています。

次の表は、構成の詳細を示しています。

IP アドレス	NSIP	VIP	SNIP
斑点	はい	はい	はい
縞模様	いいえ	はい	はい

たとえば、4 ノードのクラスターグループでは、各ノードにスポッティング SNIP アドレスを構成する必要があります。スポッティング IP 設定を構成する方法の詳細については、「ストライピング、部分ストライプ、およびスポッティングされた構成」を参照してください。

SNIP アドレスは、1 つのノードでのみアクティブにすることも、すべてのノードでアクティブにすることもできます。仮想 IP アドレスとサブネット IP アドレスが特定のノードでのみ使用可能である場合、そのノードはスポット構成です。サブネット IP アドレスと仮想サーバーの IP アドレスがすべてのノードで使用可能な場合、構成はストライプ構成として定義されます。スポット付き SNIP アドレスは、ステアリングとバックプレーンのトラフィックを減らすのに役立ちます。

ノードをクラスタに参加させる際の **VLAN** バインディングとルート設定のベストプラクティス

VLAN IP バインディング

VLAN をスポット IP アドレスにバインドする場合、NetScaler クラスターはすべてのノード上の同じサブネットのスポット IP アドレスで構成する必要があります。たとえば、Node 0 と Node 1 の 2 ノードクラスタでは、次の構成を使用できます。

```
1 add ns ip 192.254.101.101 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 1
2 add ns ip 192.254.101.102 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 0
```



```
add vlan 100
bind vlan 100 -IPAddress 192.254.101.101 255.255.255.0
<!--NeedCopy-->
```

ルーティング構成

スポッティング IP アドレスをデフォルトゲートウェイとしてルーティング設定が必要な場合は、ADC クラスターをすべてのノードの同じサブネットのスポットされた IP アドレスで構成する必要があります。たとえば、Node 0 と Node 1 の 2 ノードクラスタでは、次の構成を使用できます。

```
add ns ip 192.254.101.101 255.255.255.0 -vServer DISABLED -
    dynamicRouting ENABLED -ownerNode 1
add ns ip 192.254.101.102 255.255.255.0 -vServer DISABLED -
    dynamicRouting ENABLED -ownerNode 0

add route 192.254.102.0 255.255.255.0 192.254.101.103
<!--NeedCopy-->
```

> 注
>
> L3 クラスタ設定では、スポット付き SNIP 構成のみがサポートされます。

レイヤー 3 クラスタリングの構成

March 20, 2024

L3 クラスターについて

高可用性導入を拡張し、さまざまなネットワークにわたるクライアントトラフィックのスケーラビリティを高める必要性から、L3 クラスタの確立につながりました。L3 クラスターでは、NetScaler アプライアンスを個々のサブネット（L2 クラスター）にグループ化できます。

L3 クラスタは「独立ネットワーク構成 (INC) モードのクラスタ」とも呼ばれます。L3 クラスタ導入では、同じネットワーク内のクラスタノードがグループ化されてノードグループを形成します。L3 クラスタは GRE トンネリングを使用してネットワーク間でパケットを誘導します。L3 クラスタ全体のハートビートメッセージはルーティングされます。

アーキテクチャ

L3 クラスタアーキテクチャは、次のコンポーネントで構成されています：

- ノードグループ -次の図に示すように、各ネットワーク (n1, n2) と (n3, n4) のクラスターノードがグループ化されてノードグループを形成します。これらのノードグループは、ネットワークの両側のレイヤ 3 スイッチに終端されます。

  - クラスタは、クラスタノードとクライアント側の接続デバイスとの間の物理接続を介してクライアントと通信します。これらの物理接続を論理的にグループ化したものをクライアントデータプレーンと呼びます。
  - クラスタは、クラスタノードとサーバ側の接続デバイス間の物理接続を介してサーバと通信します。これらの物理接続の論理的なグループ化は、サーバーデータプレーンと呼ばれます。

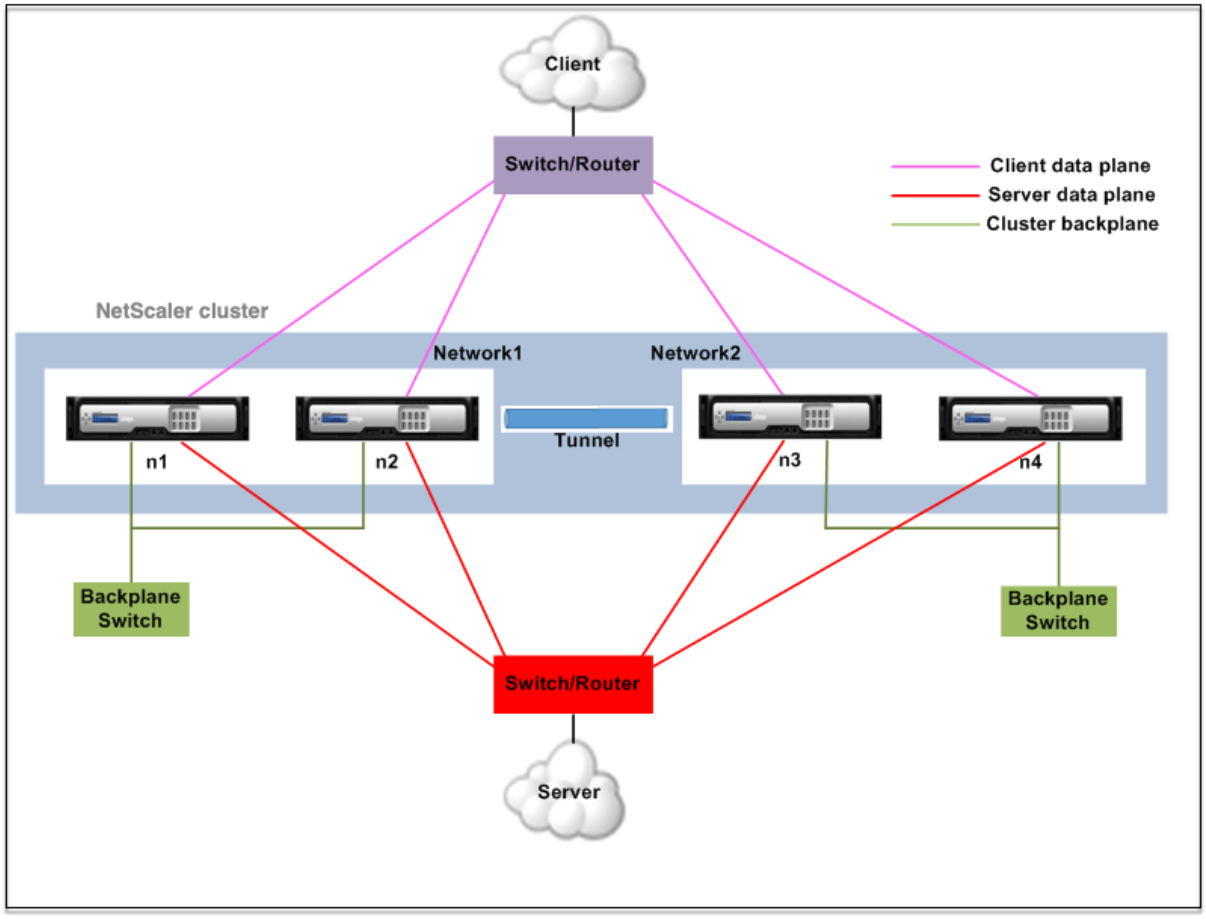
- バックプレーンスイッチ -同じネットワーク内のクラスタノードは、クラスタバックプレーンを使用して相互に通信します。バックプレーンは、各ノードの 1 つのインターフェイスがクラスタバックプレーンスイッチと呼ばれる共通のスイッチに接続されているインターフェイスのセットです。

- トンネル -デフォルトでは、L3 クラスター内のノード間のパケットは、送信元ノードと宛先ノードの NSIP アドレスを使用してルーティングする暗号化されていない GRE トンネルを介して交換されます。異なるネットワークに属するノードでは、ステアリングメカニズムが変わります。パケットは、MAC を書き換える代わりに、GRE トンネルを経由して他のサブネット上のノードに送られます。

  > 注:
  >
  > 新規導入では、UDP トンネルを使用してネットワーク間でパケットを誘導することをお勧めします。UDP トンネルを使用する利点は次のとおりです。
  >
  > - パケットエンジン間のステアリングは 4 タプルに基づいているため、分散がしやすくなります。
  > - CPU 使用率の偏りを回避できます。

次の表を参照して、さまざまなトンネルモードでパケットがどのように操作されるかを理解してください。

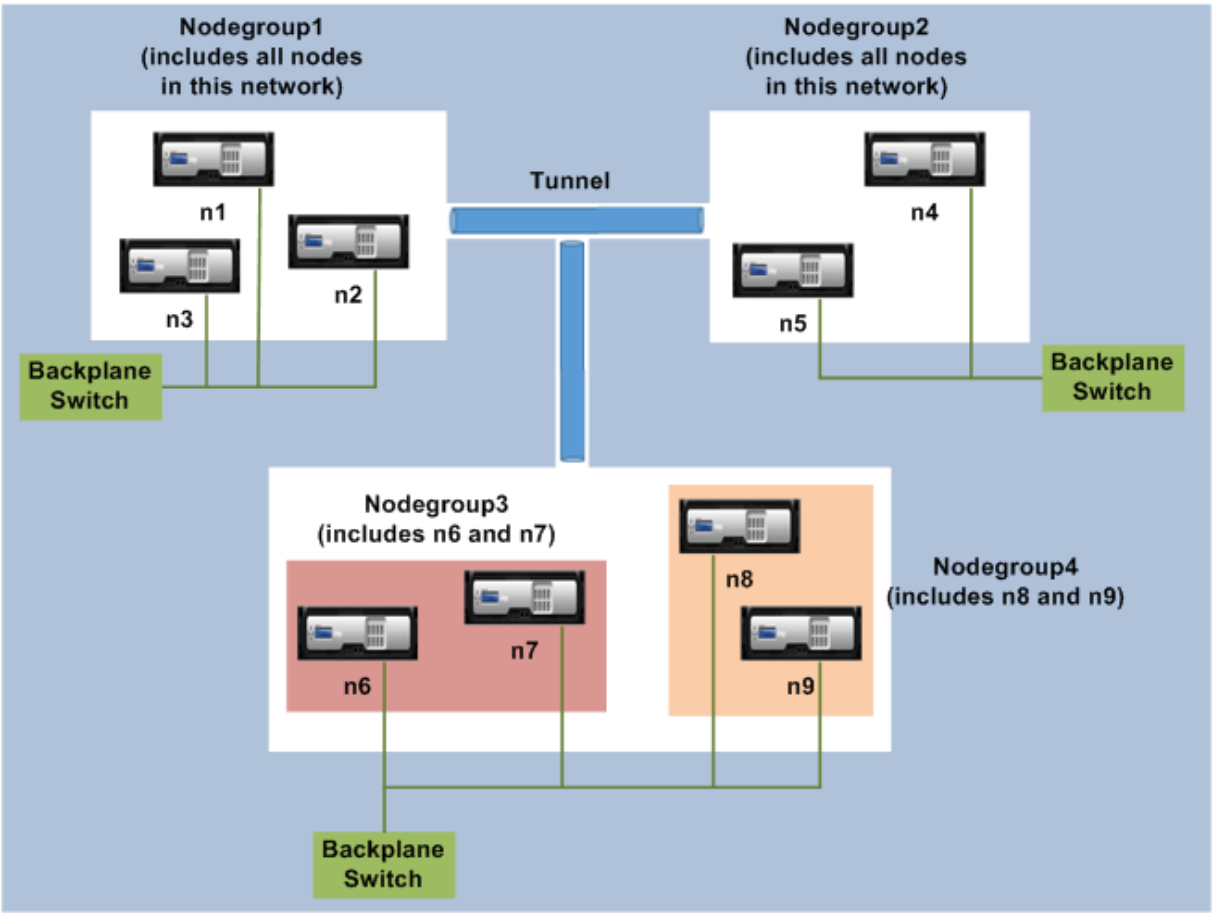
トンネルモード	同じネットワーク	異なるネットワーク
なし	バックプレーンはパケットの操作に使用されます。	GRE トンネルはトラフィックの誘導に使用されます。
GRE	GRE トンネルはパケットの操作に使用されます。	GRE トンネルはパケットの操作に使用されます。
UDP	UDP トンネルはパケットの操作に使用されます。	UDP トンネルはパケットの操作に使用されます。



例

以下で構成される L3 クラスタ展開の例を考えてみましょう。

- 3 つの NetScaler アプライアンス（n1、n2、n3）ノードがノードグループ 1 にグループ化されます。
- 同様に、ノード n4 と n5 はノードグループ 2 にグループ化されます。3 番目のネットワークには、2 つのノードグループがあります。ノードグループ 3 には n6 と n7 が含まれ、ノードグループ 4 には n8 と n9 が含まれます。
- 同じネットワークに属する NetScaler アプライアンスが組み合わされてノードグループを形成します。



L3 クラスタを設定する前に考慮すべきポイント

NetScaler アプライアンスで L3 クラスターを構成する前に、次の点を考慮してください。

- L3 サブネットを設定する場合、バックプレーンは必須ではありません。バックプレーンが指定されていない場合、ノードはバックプレーン障害状態にはなりません。

  > 注:
  >
  > 同じ L2 ネットワークに複数のノードがある場合は、バックプレーンインターフェイスを定義する必要があります。バックプレーンインターフェイスが指定されていない場合、ノードはバックプレーン障害状態になります。

- L2 機能とストライプ SNIP は L3 クラスタではサポートされていません。
- L3 クラスターの外部トラフィック分散は、等価コストマルチパス (ECMP) のみをサポートします。
- L3 クラスタ展開でステアリングが無効になっていると、ICMP エラーとフラグメンテーションは処理されません。
- ネットワークエンティティ (`route`, `route6`, `pbr`、および `pbr6`) は構成ノードグループにバインドする必要があります。

- VLAN、RNAT、および IP トンネルを設定ノードグループにバインドすることはできません。
- 設定ノードグループには必ず STRICT「YES」というプロパティが必要です。
- クラスタノードは、「add cluster node」コマンドを使用して構成ノードグループに追加しないでください。
- `add cluster instance -INC enabled`コマンドは、ネットワークエンティティ (ルート、ルート 6、PBR、pb6、RNAT、IP トンネル、ip6 トンネル) をクリアします。
- `clear config extended`+ コマンドは、L3 クラスター内のエンティティ（route、route6、PBR、pb6、RNAT、IP トンネル、ip6tunnel）をクリアしません。

L3 クラスターの設定

L3 クラスター構成では、cluster コマンドには、ノードおよびノードグループに基づいて構成するさまざまな属性があります。L3 クラスタ構成には、IPv4 プロファイルとは別に IPv6 プロファイルも含まれます。

NetScaler アプライアンスでの L3 クラスターの構成は、次のタスクで構成されます。

- クラスターインスタンスの作成
- L3 クラスタにノードグループを作成する
- NetScaler アプライアンスをクラスターに追加し、ノードグループを使用してグループ化します
- クラスタ IP アドレスをノードに追加します
- クラスターインスタンスを有効にする
- 構成を保存します
- 既存のノードグループにノードを追加
- L3 クラスタにノードグループを作成する
- 新しいノードを新しく作成したノードグループにグループ化します
- ノードをクラスターに参加させる

CLI を使用して以下を設定します

- クラスターインスタンスを作成するには

  ```
  add cluster instance <clid> -inc <ENABLED|DISABLED> -processLocal
   <ENABLED|DISABLED>
  ```

  > 注:
  >
  > L3 クラスターでは「inc」パラメーターが有効になっている必要があります。

- **L3** クラスターでノードグループを作成するには

  ```
  add cluster nodegroup <name>
  ```


- **NetScaler** アプライアンスをクラスターに追加してノードグループに関連付けるには

  ```
  add cluster node <nodeid> <nodeip> -backplane <interface_name>
  nodegroup <ng> -tunnelmode UDP
  ```

- このノードにクラスタ **IP** アドレスを追加するには

  ```
  add ns ip <IPAddress> <netmask> -type clip
  ```

- クラスターインスタンスを有効にする

  ```
  enable cluster instance <clId>
  ```

- 構成を保存します

  ```
  save ns config
  ```

- アプライアンスのウォームリブート

  ```
  reboot -warm
  ```

- 既存のノードグループに新しいノードを追加するには

  ```
  add cluster node <nodeid> <nodeip> -nodegroup <ng> -tunnelmode
  UDP
  ```

- **L3** クラスタで新しいノードグループを作成するには

  ```
  add cluster nodegroup <ng>
  ```

- 新しいノードを新しく作成したノードグループにグループ化するには

  ```
  add cluster node <nodeid> <nodeip> -nodegroup <ng> -tunnelmode
  UDP
  ```

- ノードをクラスターに参加させるには

  ```
  join cluster -clip <ip_addr> -password <password>
  ```

```
Example: The following is an example configuration for nodegroup1 and
   nodegroup2.

    > add cluster instance 1 - inc ENABLED - processLocal ENABLED

       Done

    > add cluster nodegroup ng1

       Done

    > add cluster node 0 1.1.1.1 - state ACTIVE -backplane 0/1/1 -
       nodegroup ng1 - tunnelmode UDP

       Done

```



```
15      > add ns ip 1.1.1.100 255.255.255.255 -type clip
16
17         Done
18
19      > enable cluster instance 1
20
21         Done
22
23      > save ns config
24
25         Done
26
27      > add cluster node 1 1.1.1.2 -state ACTIVE -nodegroup ng1 -
           tunnelmode UDP
28
29         Done
30
31      > add cluster nodegroup ng2
32
33         Done
34
35      > add cluster node 4 2.2.2.1 -state ACTIVE -nodegroup ng2 -
           tunnelmode UDP
36
37         Done
38
39      > add cluster node 5 2.2.2.2 -state ACTIVE -nodegroup ng2 -
           tunnelmode UDP
40
41         Done
42
43      > join cluster -clip 1.1.1.100 -password nsroot
44  <!--NeedCopy-->
```

L3 クラスタのアドバタイジングクラスタ **IP** アドレス

アップストリームルータにアドバタイズするクラスタ IP アドレスを設定して、どのサブネットからでもクラスタ設定にアクセスできるようにします。クラスタ IP アドレスは、ノードに設定された動的ルーティングプロトコルによってカーネルルートとしてアドバタイズされます。

クラスタ IP アドレスのアドバタイズには、次のタスクが含まれます。

- クラスタ **IP** アドレスのホストルートオプションを有効にします。ホストルートオプションはクラスタ IP アドレスを ZeBOS ルーティングテーブルにプッシュし、動的ルーティングプロトコルを介してカーネルルートを再配布します。
- ノードでの動的ルーティングプロトコルの設定。 ダイナミックルーティングプロトコルは、クラスタ IP アドレスをアップストリームルータにアドバタイズします。ダイナミックルーティングプロトコルの設定の詳細については、ダイナミックルートの設定を参照してください。

CLI を使用してクラスター **IP** アドレスのホストルートオプションを有効にするには

コマンドプロンプトで、次のように入力します:

- `add nsip <IPAddress> <netmask> -hostRoute ENABLED`
- `show nsip <IPAddress>`

```
1      > add ns ip 10.102.29.60 255.255.255.255 -hostRoute ENABLED
2
3      > Done
4  <!--NeedCopy-->
```

L3 クラスタのスポット付き、部分的にストライプ化された構成

L3 クラスタのスポット構成と部分的にストライプ構成は、L2 クラスタとは少し異なります。ノードは異なるサブネットにあるため、構成はノードごとに異なる場合があります。ネットワーク構成は L3 クラスタのノード固有にすることができるため、以下のパラメータに基づいてスポット構成または部分的にストライプ構成を構成する必要があります。

L3 クラスター上の NetScaler アプライアンスでスポット構成、部分ストライプ構成を構成するには、次のタスクを実行します。

- クラスター所有者グループを IPv4 静的ルーティングテーブルに追加する
- クラスター所有者グループを IPv6 静的ルーティングテーブルに追加する
- クラスター所有者グループを IPv4 ポリシーベースルーティング (PBR) に追加する
- クラスター所有者グループを IPv6 PBR に追加する
- VLAN を追加してください
- VLAN を特定の所有者グループまたはクラスタノードグループにバインドします

CLI を使用して以下を設定します

- **NetScaler** アプライアンスの **IPv4** 静的ルートテーブルにクラスター所有者グループを追加するには

  `add route <network> <netmask> <gateway> -owner group <ng>`

- **NetScaler** アプライアンスの **IPv6** 静的ルートテーブルにクラスター所有者グループを追加するには

  `add route6 <network> -owner group <ng>`

- クラスター所有者グループを **IPv4 PBR** に追加するには

  `add pbr <name> <action> -owner group <ng>`

- クラスター所有者グループを **IPv6 PBR** に追加するには

  `add pbr6 <name> <action> -owner group <ng>`

- **VLAN** を追加するには

  ```
  add vlan <id>
  ```

- **VLAN** をクラスタノードグループの特定の所有者/グループにバインドするには

  ```
  bind vlan <id> -ifnum - [IPAddress <ipv4_addr | ipv6_addr> [-
  owner group <ng>]
  ```

  次のコマンドは、CLI を使用して設定できるスポット構成と部分ストライプ構成の例です。

```
1      > add route 10.102.29.0 255.255.255.0 10.102.29.2  - ownergroup ng2
2
3          Done
4
5      > add route6 fe80::9404:60ff:fedd:a464/64  - ownergroup ng1
6
7          Done
8
9      > add pbr pbr1 allow  - ownergroup ng1
10
11         Done
12
13     > add pbr6 pbr2 allow  - ownergroup ng2
14
15         Done
16
17     > add vlan 2
18
19         Done
20
21     > bind vlan 2  - ifnum 1/2  - [IPAddress 10.102.29.80 | fe80::9404:60
          ff:fedd:a464/64-ownergroup ng1
22
23         Done
24 <!--NeedCopy-->
```

ノードグループを設定

L3 クラスターでは、同じ構成セットを複数のノードグループに複製するには、次のコマンドを使用します:

CLI を使用して以下を設定します

- **NetScaler** アプライアンスのルーティングテーブルに **IPv4** 静的ルートを追加するには

  ```
  add route <network> <netmask> <gateway> -ownerGroup <ng>
  ```

サンプル構成**:**


```
add route 0 0 10.102.53.1 -ownerGroup ng1

add route 0 0 10.102.53.1 -ownerGroup ng2
<!--NeedCopy-->
```

前述の構成をサポートするために新しいノードグループ「all」を定義し、次のコマンドを設定する必要があります。

CLI を使用して以下を設定します

- 厳密なパラメーターを使用してクラスターに新しいノードグループを追加するには

  ```
  add cluster node group <name> -strict <YES | NO>
  ```

- クラスターノードまたはエンティティを特定のノードグループにバインドするには

  ```
  bind cluster nodegroup <name> -node <nodeid>
  ```

- **IPv4** 静的ルートをすべてのオーナーグループに追加するには

  ```
  add route <network> <netmask> <gateway> -ownerGroup <ng>
  ```

サンプル構成**:**

```
add cluster nodegroup all -strict YES

bind cluster nodegroup all -node 1

bind cluster nodegroup all -node 2

add route 0 0 10.102.53.1 -ownerGroup all
<!--NeedCopy-->
```

L3 クラスタ内のトラフィック分散

クラスター設定では、外部ネットワークは NetScaler アプライアンスのコレクションを単一のエンティティと見なします。したがって、クラスターは、トラフィックを受信する必要がある単一のノードを選択する必要があります。L3 クラスタでは、この選択は ECMP を使用して行われます。選択されたノードをフロー受信機と呼びます。

> 注
>
> L3 クラスター (異なるネットワークにまたがるノード) では、ECMP トラフィック分散のみを使用できます。

フローレシーバがトラフィックを取得し、内部クラスタロジックを使用してトラフィックを処理する必要があるノードを決定します。このノードはフロープロセッサと呼ばれます。フローレシーバとフロープロセッサが同じネットワーク上にある場合、フローレシーバはバックプレーンを介してフロープロセッサにトラフィックを誘導します。フローレシーバとフロープロセッサが異なるネットワーク上にある場合、トラフィックはトンネルを通過します。

> 注
>
> - フローレシーバとフロープロセッサは、トラフィックを処理できるノードでなければなりません。
> - NetScaler 11 以降では、クラスタバックプレーンでステアリングを無効にできます。詳細については、クラスタバックプレーンでのステアリングの無効化を参照してください。

前の図は、クラスターを流れるクライアント要求を示しています。クライアントは、仮想 IP (VIP) アドレスにリクエストを送信します。クライアントデータプレーンに設定されたトラフィック分散メカニズムは、クラスターノードの 1 つをフローレシーバーとして選択します。フローレシーバは、トラフィックを受信し、トラフィックを処理する必要があるノードを決定し、そのノードに要求を操縦します（フローレシーバが自身をフロープロセッサとして選択しない限り）。フロープロセッサとフローレシーバが同じノードグループにある場合、パケットはバックプレーン上でステアリングされます。また、フロープロセッサとフローレシーバが異なるノードグループに属している場合、パケットはルーテッドパスを介してトンネルを通過します。

フロープロセッサは、サーバとの接続を確立します。サーバは要求を処理し、要求をサーバに送信したサブネット IP（SNIP）アドレスに応答を送信します。L3 クラスタでは SNIP は常にスポッティングされた SNIP であるため、SNIP アドレスを所有するノードはサーバからの応答を受信します。

NetScaler ADC クラスターの設定

August 15, 2023

クラスターに追加する NetScaler ADC アプライアンスは、「クラスターノードの前提条件」で指定された基準を満たす必要があります。クラスターを実際にセットアップする前に、クラスターの基本を理解しておく必要があります。詳細については、「クラスターの概要」を参照してください。

クラスターを形成するには、ノード間通信をセットアップし、（最初の NetScaler ADC アプライアンスを追加して）クラスターを作成し、他のクラスターノードを追加する必要があります。これらの各ステップについては、以降のトピックで関連する詳細とともに説明します。

> 注
>
> L2 クラスターと L3 クラスターの設定にはいくつかの違いがありますが、類似点も多数あります。以降のトピックでは、L3 クラスター固有の構成に焦点を当てながら、両方のクラスタータイプの設定について説明します。

ノード間通信の設定

April 15, 2024

クラスタセットアップ内のノードは、次のノード間通信メカニズムを使用して相互に通信します。

- ネットワーク（同じサブネット）にあるノードは、クラスタバックプレーンを介して相互に通信します。バックプレーンは明示的に設定する必要があります。詳細な手順は次のとおりです。
- ネットワーク全体では、パケットのステアリングは GRE トンネルを介して行われ、その他のノード間通信は必要に応じてノード間でルーティングされます。

> 重要
>
> - リリース 11.0 以降、すべてのビルドで、クラスタに異なるネットワークのノードを含めることができます。
> - リリース 13.0 ビルド 58.3 以降、GRE ステアリングは L3 クラスタのフォートビル NIC でサポートされています。

クラスタバックプレーンをセットアップするには、ノードごとに次の操作を行います

1. バックプレーンに使用するネットワークインターフェイスを特定してください。
2. 選択したネットワークインターフェイスからクラスタバックプレーンスイッチに Ethernet ケーブルまたは光ケーブルを接続します。

たとえば、ノード 4 のバックプレーンインターフェイスとしてインターフェイス 1/2 を使用するには、ノード 4 の 1/2 インターフェイスからバックプレーンスイッチにケーブルを接続します。

クラスタバックプレーンをセットアップする際の注意点

- アプライアンスの管理インターフェイス（0/x）をバックプレーンインターフェイスとして使用しないでください。クラスタでは、インターフェイス 0/1/x は次のように読み込まれます：

  0 -> ノード ID 0
  1/x-> NetScaler インターフェイス

- クライアントまたはサーバのデータプレーンには、バックプレーンインターフェイスを使用しないでください。
- クラスタバックプレーンには、リンク集約（LA）チャネルを使用することをお勧めします。

  > 注：
  >
  > LA チャネルからインターフェイスをバインド解除したら、バインドされていないインターフェイスに適切な MTU サイズを設定してください。

- バックプレーンがバックツーバック接続されている 2 ノードクラスタでは、次のいずれかの条件下で、クラスタは動作上 DOWN になります：
  - ノードの 1 つが再起動されます。
  - いずれかのノードのバックプレーンインターフェースが無効になっています。

そのため、Citrix では、他のクラスタノードやトラフィックに影響を与えないように、バックプレーン専用のスイッチを用意することを推奨しています。バックツーバックリンクでクラスターをスケールアウトすることはできません。クラスターノードをスケールアウトすると、本番環境でダウンタイムが発生する可能性があります。

- クラスタのすべてのノードのバックプレーンインターフェイスは、同じスイッチに接続され、同じ L2 VLAN にバインドされている必要があります。
- 同じクラスタインスタンス ID を持つクラスタが複数ある場合は、各クラスタのバックプレーンインターフェイスが異なる VLAN にバインドされていることを確認してください。
- バックプレーンインターフェイスは、そのインターフェイスの HA モニタリング設定に関係なく、常に監視されます。
- さまざまな仮想化プラットフォームでの MAC スプーフィングの状態は、クラスタバックプレーンのステアリングメカニズムに影響を与える可能性があります。そのため、適切な状態が設定されていることを確認してください。
  - XenServer-MAC スプーフィングを無効にする
  - Hyper-V-MAC スプーフィングを有効にする
  - VMware ESX-MAC スプーフィングを有効にする（「偽造送信」が有効になっていることも確認する）
- クラスタバックプレーンの MTU は自動的に更新されます。ただし、クラスタでジャンボフレームを設定する場合は、クラスタバックプレーンの MTU を明示的に設定する必要があります。値は 78 + X に設定する必要があります。X はクライアントとサーバのデータプレーンの最大 MTU です。たとえば、サーバデータプレーンの MTU が 7500 で、クライアントデータプレーンの MTU が 8922 だとします。クラスタバックプレーンの MTU は 78 + 8922 = 9000 に設定する必要があります。この MTU を設定するには、次のコマンドを使用します。

```
> set interface <backplane_interface> -mtu <value>
```

- バックプレーンスイッチのインターフェイスの MTU は、1,578 バイト以上で指定する必要があります。クラスタに MBF、L2 ポリシー、ACL、CLAG 環境でのルーティング、VPath などの機能がある場合に適用できます。

> 注:
>
> バックプレーンインターフェイスのデフォルトの MTU サイズは 1578 です。MTU サイズをデフォルト値にリセットするには、`unset interface <backplane_interface> -mtu` コマンドを使用する必要があります。

L2 および L3 クラスタの UDP ベースのトンネルサポート

NetScaler リリース 13.0 ビルド 36.x 以降、NetScaler L2 および L3 クラスターは UDP ベースのトンネリングを使用してトラフィックを誘導できるようになりました。クラスタ内の 2 つのノードのノード間通信用に定義されてい

ます。「トンネルモード」パラメータを使用すると、クラスタノードを追加および設定するコマンドから GRE または UDP トンネルモードを設定できます。

L3 クラスター展開では、NetScaler ノード間のパケットは、送信元ノードと宛先ノードの NSIP アドレスを使用してルーティングされる暗号化されていない GRE トンネルを介して交換されます。この交換がインターネット上で行われる場合、IPsec トンネルがないと NSIP がインターネット上に公開され、セキュリティ上の問題が発生する可能性があります。

> 重要
>
> Citrix では、L3 クラスタを使用する場合は独自の IPsec ソリューションを確立することをお客様に推奨しています。

次の表は、さまざまな展開に基づいてトンネルサポートを分類するのに役立ちます。

ステアリングタイプ	AWS	Microsoft Azure	オンプレミス
MAC	未サポート	未サポート	サポート対象
GRE トンネル	サポート対象	未サポート	サポート対象
UDP トンネル	サポート対象	サポート対象	サポート対象

> 重要
>
> L3 クラスタでは、トンネルモードはデフォルトで GRE に設定されます。

UDP ベースのトンネルの設定

ノード ID のパラメータを設定して状態を記述することで、クラスタノードを追加できます。インターフェイス名を指定してバックプレーンを設定し、任意のトンネルモード (GRE または UDP) を選択します。

> 注**:**
>
> クラスタ IP アドレスからトンネルモードを設定する必要があります。

CLI のプロシージャ　CLI を使用して UDP トンネルモードを有効にします。

コマンドプロンプトで入力します:

- `add cluster node <nodeId>@ [-state <state>] [-backplane <interface_name>] [-tunnelmode <tunnelmode>]`
- `set cluster node <nodeId>@ [-state <state>] [-tunnelmode <tunnelmode>]`

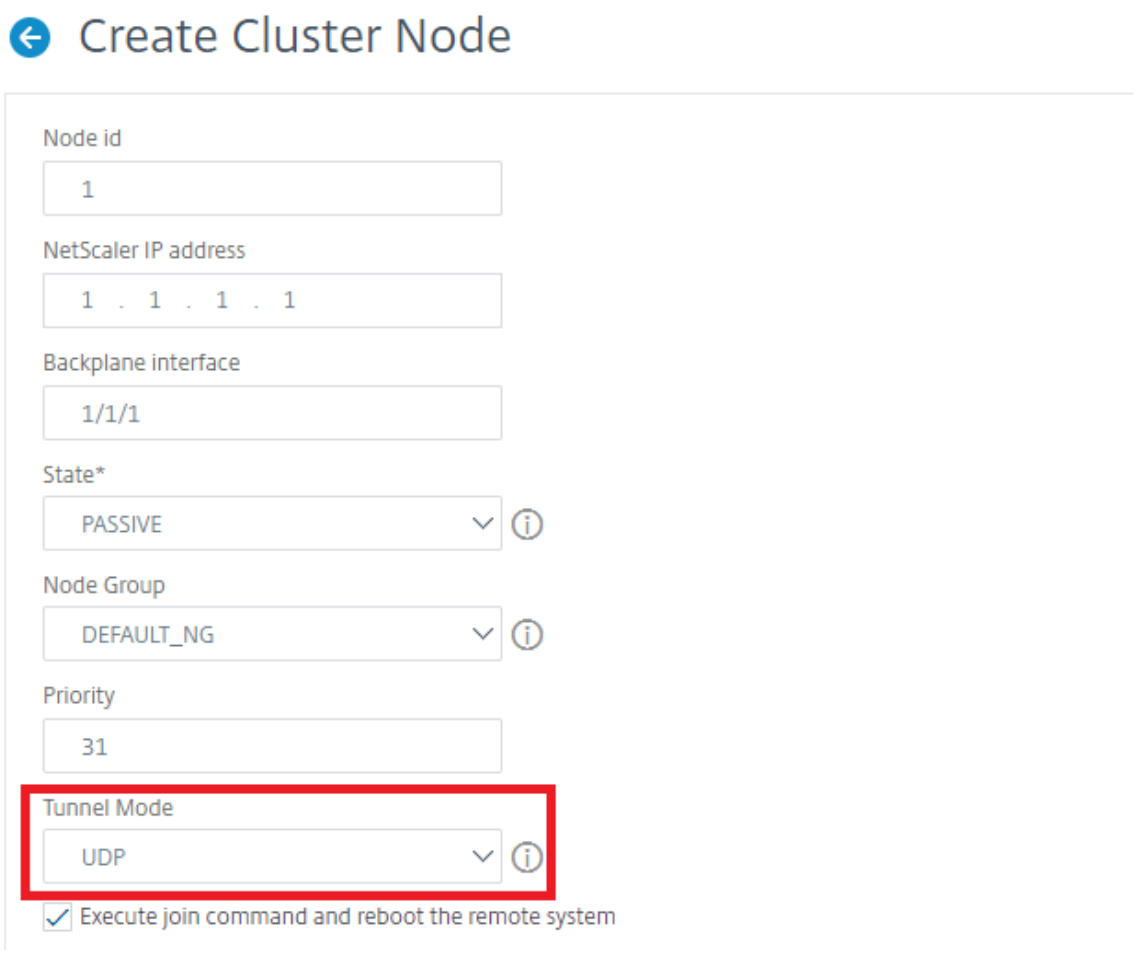
> 注
>
> トンネルモードに設定できる値は、NONE、GRE、UDP です。

例

- `add cluster node 1 -state ACTIVE -backplane 1/1/1 -tunnelmode UDP`
- `set cluster node 1 -state ACTIVE -tunnelmode UDP`

GUI のプロシージャ　GUI を使用して UDP トンネルモードを有効にします。

1. [システム] **>** [クラスタ] **>** [ノード] に移動します。
2. 「クラスターノード」ページで、「追加」をクリックします。
3. 「クラスターノードの作成」で、「トンネルモード」パラメーターを UDP に設定し、「作成」をクリックします。



4. [閉じる] をクリックします。

NetScaler ADC クラスターの作成

August 15, 2023

クラスターを作成するには、まず、クラスターに追加する NetScaler アプライアンスの 1 つを使用します。このノードでは、クラスターインスタンスを作成し、クラスター IP アドレスを定義する必要があります。このノードは最初のクラスタノードで、クラスタ構成コーディネータ (CCO) と呼ばれます。クラスタ IP アドレスで実行されるすべての構成は、このノードに保存され、他のクラスタノードに伝播されます。

クラスタ内の CCO の責任は、特定のノードに定められているわけではありません。次の要因によって、時間の経過とともに変化する可能性があります。

- ノードの優先度。プライオリティが最も高い（プライオリティ番号が最も低い）ノードが CCO になります。そのため、既存の CCO よりも低いプライオリティ番号のノードが追加されると、新しいノードが CCO を引き継ぎます。
- 現在の CCO がダウンすると、プライオリティ番号が次に低いノードが CCO を引き継ぎます。プライオリティが設定されていない場合、またはプライオリティ番号が最も低いノードが複数存在する場合、CCO は使用可能なノードの 1 つから選択されます。

> 注:
>
> アプライアンスの設定（SNIP アドレスと VLAN を含む）は、`clear ns config extended` コマンドを暗黙的に実行することでクリアされます。ただし、デフォルト VLAN と NSVLAN はアプライアンスからクリアされません。したがって、クラスタ上に NSVLAN が必要な場合は、アプライアンスをクラスタに追加する前に NSVLAN が作成されていることを確認してください。L3 クラスタ（異なるネットワーク上のクラスタノード）では、ネットワーク設定はアプライアンスから消去されません。
>
> 重要:
>
> クラスタセットアップの HA Monitor（HANON）は、各ノードのインターフェイスの健全性を監視するために使用されます。インターフェイスの状態を監視するには、各ノードで HAMON パラメータを有効にする必要があります。何らかの理由で HAMON 対応インターフェイスの動作状態がダウンした場合、それぞれのクラスタノードは正常ではない（NOT UP）とマークされ、そのノードはトラフィックを処理できません。

コマンドラインインターフェイスを使用してクラスターを作成する

- クラスターに追加する NetScaler アプライアンス（NSIP アドレスが 10.102.29.60 のアプライアンス）にログオンします。
- クラスターインスタンスを追加します。

```
add cluster instance <clId> -quorumType <NONE | MAJORITY> -inc <
    ENABLED | DISABLED> -backplanebasedview <ENABLED | DISABLED>
<!--NeedCopy-->
```


- -dfdretainl2paramsオプションにより、バックプレーントラフィック用の拡張 L2 ヘッダーを追加できます。

  コマンドプロンプトで入力します。

  ```
  add cluster instance 1 -dfdretainl2params <ENABLED|DISABLED>
  ```

  次のコマンドは、-dfdretainl2paramsのステータスを表示します。

  ```
  show cluster instance <clusterid>
  ```

  次のコマンドを使用して、-dfdretainl2paramsを有効または無効にします。

  ```
  set cluster instance 1 -dfdretainl2params <ENABLED|DISABLED>
  ```

- -proxyarpstatusオプションは、クラスタのプロキシ ARP 機能を有効または無効にします。

  コマンドプロンプトで入力します。

  ```
  add cluster instance 1 -proxyarpstatus <ENABLED|DISABLED>
  ```

  次のコマンドは、proxyarpstatusのステータスを表示します。

  ```
  show cluster instance <clusterid>
  ```

  次のコマンドを使用して、proxyarpstatusを有効または無効にできます。

  ```
  set cluster instance 1 -proxyarpstatus <ENABLED|DISABLED>
  ```

> 注:
>
> - クラスターインスタンス ID は LAN 内で一意である必要があります。
> - 次のシナリオでは、-quorumType パラメーターを [NONE] ではなく [Majority] に設定する必要があります。
>   - Topologies which do not have redundant links between cluster nodes. These topologies might be prone to network partition due to a single point of failure.
>   - During any cluster operations such as node addition or removal.
> - L3 クラスタの場合は、-inc パラメータが ENABLED に設定されていることを確認します。L2 クラスターでは、-incパラメーターは無効にする必要があります。
> - -backplanebasedview パラメータを有効にすると、バックプレーンインターフェイスでのみ受信したハートビートに基づいて、動作ビュー（トラフィックを処理するノードのセット）が決定されます。既定では、このパラメーターは無効になっています。このパラメータが無効の場合、ノードはバックプレーン上のハートビート受信だけに依存しません。

1. [L3 クラスターのみ] ノードグループを作成します。次のステップでは、新しく追加されたクラスタノードをこのノードグループに関連付ける必要があります。

   > 注:
   >
   > このノードグループには、同じネットワークに属する NetScaler アプライアンスのすべてまたはサブセ

> ットが含まれます。

```
add cluster nodegroup <name>
<!--NeedCopy-->
```

2. NetScaler アプライアンスをクラスターに追加します。

```
add cluster node <nodeId> <IPAddress> -state <state> -backplane <
    interface_name> -nodegroup <name>
<!--NeedCopy-->
```

> 注:
>
> L3 クラスタの場合:
>
> - node group パラメーターは、作成するノードグループの名前に設定する必要があります。
> - バックプレーンパラメータは、ネットワーク内のノードが相互に通信できるように、複数のノードを持つノードグループに関連付けられているノードに必須です。

例:

L2 クラスターにノードを追加する (すべてのクラスターノードが同じネットワーク内にある)。

```
  add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1
  <!--NeedCopy-->
```

各ネットワークから 1 つのノードを含む L3 クラスタ用のノードを追加する。ここでは、バックプレーンを設定する必要はありません。

```
  add cluster node 0 10.102.29.60 -state PASSIVE -nodegroup ng1
  <!--NeedCopy-->
```

各ネットワークの複数のノードを含む L3 クラスタにノードを追加する。ここでは、ネットワーク内のノードが相互に通信できるようにバックプレーンを設定する必要があります。

```
  add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1
     -nodegroup ng1
  <!--NeedCopy-->
```

3. このノードにクラスター IP アドレス (10.102.29.61 など) を追加します。

```
add ns ip <IPAddress> <netmask> -type clip
<!--NeedCopy-->
```

例

```
add ns ip 10.102.29.61 255.255.255.255 -type clip
<!--NeedCopy-->
```

4. クラスターインスタンスを有効にします。


```
enable cluster instance <clId>
<!--NeedCopy-->
```

5. 構成を保存します。

```
save ns config
<!--NeedCopy-->
```

6. アプライアンスをウォーム再起動します。

```
reboot -warm
<!--NeedCopy-->
```

show cluster instance コマンドを使用して、クラスタ構成を確認します。コマンドの出力に、クラスタのノードとしてのアプライアンスの NSIP アドレスが表示されることを確認します。

7. ノードが UP になったら、CLIP にログインし、クラスタ IP アドレスとノード IP アドレスの両方の RPC 資格情報を変更します。RPC ノードのパスワードの変更の詳細については、「RPC ノードのパスワードを変更する」を参照してください。

GUI を使用してクラスターを作成するには

1. クラスタに追加するアプライアンス（NSIP アドレスが 10.102.29.60 のアプライアンスなど）にログオンします。
2. [システム**] > [**クラスタ**]** に移動します。
3. 詳細ウィンドウで、[クラスターの管理] リンクをクリックします。
4. [Cluster Configuration] ダイアログボックスで、クラスターの作成に必要なパラメーターを設定します。パラメータの説明を表示するには、対応するテキストボックスにマウスカーソルを合わせます。
5. [作成] をクリックします。
6. [クラスターインスタンスの構成] ダイアログボックスで、[クラスターインスタンスを有効にする] チェックボックスをオンにします。
7. クラスターノードペインでノードを選択し、「開く」をクリックします。
8. [クラスタノードの構成] ダイアログボックスで、[状態] を設定します。
9. 「**OK**」をクリックし、「保存」をクリックします。
10. アプライアンスをウォーム再起動します。
11. ノードが UP になったら、CLIP にログインし、クラスタ IP アドレスとノード IP アドレスの両方の RPC 資格情報を変更します。RPC ノードのパスワードの変更の詳細については、「RPC ノードのパスワードを変更する」を参照してください。

クラスタの同期状態に対する厳密なモードのサポート

設定の適用時にエラーを表示するようにクラスタノードを設定できるようになりました。add と set cluster instance コマンドの両方に、クラスタ内の各ノードのステータスを追跡する新しいパラメータ「syncStatusStrict-

Mode」が導入されました。デフォルトでは、syncStatusStrictModeパラメータは無効になっています。

CLI を使用して **strict** モードを有効にするには

コマンドプロンプトで入力します。

```
set cluster instance <clID> [-syncStatusStrictMode (ENABLED | DISABLED)
    ]
<!--NeedCopy-->
```

例:

```
set cluster instance 1  -syncStatusStrictMode ENABLED
<!--NeedCopy-->
```

CLI を使用して **strict** モードのステータスを表示するには

```
    >show cluster instance
    1) Cluster ID: 1
        Dead Interval: 3 secs
        Hello Interval: 200 msecs
        Preemption: DISABLED
        Propagation: ENABLED
        Quorum Type: MAJORITY
        INC State: DISABLED
        Process Local: DISABLED
        Retain Connections: NO
        Heterogeneous: NO
        Backplane based view: DISABLED
        Cluster sync strict mode: ENABLED
        Cluster Status: ENABLED(admin), ENABLED(operational), UP

      WARNING(s):
        (1) - There are no spotted SNIPs configured on the cluster.
            Spotted SNIPs can help improve cluster performance

        Member Nodes:
        Node ID     Node IP      Health     Admin State  Operational
            State
        -------     -------      ------     -----------
            ------------------
    1)      1           192.0.2.20      UP         ACTIVE       ACTIVE(
        Configuration Coordinator)
    2)      2           192.0.2.21      UP         ACTIVE       ACTIVE
    3)      3           192.0.2.19*     UP         ACTIVE       ACTIVE
<!--NeedCopy-->
```


GUI を使用してクラスタノードの同期失敗の理由を表示するには

1. システム **>** クラスタ **>** クラスタノードに移動します。
2. [**Cluster Nodes**] ページで右端までスクロールして、クラスタノードの同期失敗理由の詳細を表示します。

クラスターへのノードの追加

December 8, 2023

クラスターのサイズを最大 32 ノードまでシームレスにスケーリングできます。NetScaler アプライアンスをクラスターに追加すると、そのアプライアンスの構成は（内部で clear ns config-extended コマンドを実行して）クリアされます。SNIP アドレス、バックプレーンインターフェイスの **MTU** 設定、およびすべての VLAN 設定（デフォルト VLAN と NSVLAN を除く）もアプライアンスから消去されます。

その後、クラスター構成はこのノードで同期されます。同期の進行中は、トラフィックが断続的に減少する可能性があります。

> 重要
>
> NetScaler アプライアンスをクラスターに追加する前に:
>
> - ノードのバックプレーンインターフェイスを設定します。前のトピックを確認してください。
> - アプライアンスで使用可能なライセンスが、構成コーディネーターで利用可能なライセンスと一致しているかどうかを確認してください。アプライアンスは、ライセンスが一致する場合にのみ追加されます。
> - クラスタに NSVLAN を使用する場合は、クラスタに追加する前に、アプライアンスで NSVLAN が作成されていることを確認してください。
> - Citrix では、このノードをパッシブノードとして追加することをお勧めします。次に、ノードをクラスターに参加させた後、クラスター IP アドレスからノード固有の構成を完了します。クラスターにスポットされた IP アドレスしかない場合は、force cluster sync コマンドを実行します。また、L3 VLAN バインディングを備えているか、スタティックルートを備えているか。
> - リンクアグリゲート（LA）チャネルがあらかじめ設定されたアプライアンスをクラスタに追加しても、LA チャネルはクラスタ環境に存在し続けます。LA チャネルの名前は la/X から nodeID/LA/x に変更されます。ここで、la/X は LA チャネル識別子です。

CLI を使用してクラスターにノードを追加するには

> 注
>
> クラスタセットアップにノードを追加するときは、ノードに設定されているデフォルトの静的ルートがクラスタコーディネーターノード (CCO) に存在することを確認してください。ノードに CCO に存在しないスタティックルートがある場合、コマンドは次のエラーで失敗します。

> Node cannot join cluster as **static default** gateway route is not present in CCO

1. クラスタ IP アドレスにログオンし、コマンドプロンプトで次の操作を行います:

   - アプライアンス (たとえば、10.102.29.70) をクラスタに追加します。

   > 注
   >
   > L3 クラスタの場合:
   >
   > - ノードグループパラメータは、同じネットワークのノードを含むノードグループに設定する必要があります。
   > - このノードが、最初に追加されたノードと同じネットワークに属している場合は、そのノードに使用されたノードグループを設定します。
   > - このノードが別のネットワークに属している場合は、ノードグループを作成し、このノードをノードグループにバインドします。
   > - バックプレーンパラメータは、ネットワーク内のノードが相互に通信できるように、複数のノードを持つノードグループに関連付けられているノードに必須です。

   ```
   add cluster node <nodeId> <IPAddress> -state <state> -backplane <
       interface_name> -nodegroup <name>

   Example:

   add cluster node 1 10.102.29.70 -state PASSIVE -backplane 1/1/1
   <!--NeedCopy-->
   ```

   - 構成を保存します。

   ```
   save ns config
   <!--NeedCopy-->
   ```

2. 新しく追加されたノード (たとえば、10.102.29.70) にログオンし、ノードをクラスターに参加させます。

   ```
   join cluster -clip <ip_addr> -password <password>

   Example:

   join cluster -clip 10.102.29.61 -password nsroot
   <!--NeedCopy-->
   ```

3. CLIP で次のコマンドを設定します。

   - VLAN をインターフェイスにバインド

     ```
     bind vlan <id> -ifnum <interface_name>
     <!--NeedCopy-->
     ```


例:

```
bind vlan 1 -ifnum 2/1/2
<!--NeedCopy-->
```

- 新しく追加されたノードにスポッティング IP アドレスを追加

```
add ns ip <IpAddress> <netmask> -ownerNode <positive_interger
    >
<!--NeedCopy-->
```

例:

```
add ns ip 97.131.0.3 255.0.0.0 -ownerNode 2
<!--NeedCopy-->
```

- NSIP の VLAN を検証

```
show vlan <id>
<!--NeedCopy-->
```

例:

```
show vlan 1
<!--NeedCopy-->
```

4. 次の設定を行います。

   - スポット IP のみのクラスターにノードを追加すると、スポット IP アドレスがそのノードに割り当てられる前に構成が同期されます。このような場合、L3 VLAN バインディングが失われる可能性があります。この損失を回避するには、ストライピング IP を追加するか、L3 VLAN バインディングを追加してください。
   - 必要なスポット構成を定義します。
   - バックプレーンインターフェイスの MTU を設定します。

5. 構成を保存します。

```
save ns config
<!--NeedCopy-->
```

6. アプライアンスをウォーム再起動します。

```
reboot -warm
<!--NeedCopy-->
```

7. ノードが UP 状態になり、同期が成功したら、クラスタの IP アドレスからノードの RPC 認証情報を変更します。RPC ノードのパスワードの変更の詳細については、「RPC ノードのパスワードを変更する」を参照してください。

```
set rpcNode <node-NSIP> -password <passwd>
```



```
2
3 Example:
4
5 set rpcNode 192.0.2.4 -password mypassword
6 <!--NeedCopy-->
```

8. クラスタノードを [アクティブ] に設定します。

```
1 set cluster node <nodeID> -state active.
2
3 Example:
4
5 set cluster node 1 -state active
6 <!--NeedCopy-->
```

GUI を使用してクラスターにノードを追加するには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [クラスタ] **>** [ノード] に移動します。
3. 詳細ペインで、[追加] をクリックして新しいノード (たとえば、10.102.29.70) を追加します。
4. クラスターノードの作成ダイアログボックスで、新しいノードを設定します。パラメータの説明を表示するには、対応するテキストボックスにマウスカーソルを合わせます。
5. [作成] をクリックします。ウォームリブートを実行するように求めるメッセージが表示されたら、[はい] をクリックします。
6. ノードが UP 状態になり、同期が成功したら、クラスタの IP アドレスからノードの RPC 認証情報を変更します。RPC ノードのパスワードの変更の詳細については、「RPC ノードのパスワードを変更する」を参照してください。
7. [システム] **>** [クラスタ] **>** [ノード] **>** [編集] に移動します。
8. ステートを **ACTIVE** に変更して確認します。

GUI を使用して以前に追加したノードをクラスターに参加させるには

CLI を使用してクラスターにノードを追加したが、そのノードをクラスターに参加させていない場合は、以下の手順を使用できます。

> 注
>
> ノードがクラスターに参加すると、そのノードがクラスターからトラフィックのシェアを引き継ぐため、既存の接続が終了する可能性があります。

1. クラスターに参加させたいノード (たとえば、10.102.29.70) にログインします。
2. [システム] **>** [クラスタ] に移動します。
3. 詳細ペインの「はじめに」で、「クラスターに参加」リンクをクリックします。

4. 「既存のクラスターに参加」ダイアログボックスで、クラスターの IP アドレスと構成コーディネーターの`nsroot`パスワードを設定します。パラメータの説明を表示するには、対応するテキストボックスにマウスカーソルを合わせます。
5. 「**OK**」をクリックします。

クラスターの詳細の表示

August 15, 2023

クラスター IP アドレスにログオンすると、クラスターインスタンスとクラスターノードの詳細を表示できます。

CLI を使用してクラスターインスタンスの詳細を表示するには

クラスタ IP アドレスにログオンし、コマンドプロンプトで次のように入力します。

```
1 show cluster instance <clId>
```

> 注
>
> 上記のコマンドを非 CCO ノードの NSIP アドレスから実行すると、コマンドはこのノード上のクラスターのステータスを表示します。

CLI を使用してクラスターノードの詳細を表示するには

クラスタ IP アドレスにログオンし、コマンドプロンプトで次のように入力します。

```
1 show cluster node <nodeId>
```

GUI を使用してクラスターインスタンスの詳細を表示するには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [クラスタ] に移動します。
3. 詳細ペインの [は じめに] で、[クラスターの管理] リンクをクリックして、クラスターの詳細を表示します。

GUI を使用してクラスターノードの詳細を表示するには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [クラスタ] **>** [ノード] に移動します。
3. 詳細ペインで、詳細を表示するノードをクリックします。

クラスタノード間でのトラフィックの分散

August 15, 2023

NetScaler クラスターを作成して必要な構成を実行したら、クライアントデータプレーン（クライアントトラフィック用）またはサーバーデータプレーン（サーバートラフィック用）に Equal Cost Multiple Path（ECMP）またはクラスターリンクアグリゲーション（LA）を展開する必要があります。これらのメカニズムは、外部トラフィックをクラスターノード全体に分散します。

ポリシーベースのバックプレーンステアリング

ポリシーベースのバックプレーンステアリング (PBS) はクラスタ展開のメカニズムで、フローに定義されたハッシュ方式に基づいてクラスタノード間でトラフィックを誘導します。フローは、アクセス制御リスト (ACL) と同様の L2 パラメータと L3 パラメータの組み合わせによって定義されます。

PBS は IPv4 と IPv6 の両方のトラフィックをサポートします。IPv6 環境の場合、ステアリングは追加のオプションをサポートします。[dfdprefix <positive_integer>] 同じ IP プレフィックスに対して同じフロープロセッサを柔軟に選択できます。プレフィックスオプションは、送信元 IP または宛先 IP ハッシュ方式でのみサポートされます。

> 注
>
> PBS メカニズムを使用してトラフィックを誘導しない場合、トラフィックはデフォルトの方法で制御されます。

新しい ACL 属性を設定するには、CLI で次のコマンドを入力します。

IPv4 用の **CLI** コマンド

- add ns acl <aclname> <aclaction> [-type (classic | dfd)] [-dfdhash <dfdhash>]
- set ns acl <aclname> <aclaction> [-dfdhash <dfdhash>
- show ns acl [<aclname>][-type (classic | DFD)]
- apply ns acls [-type (classic | DFD)]
- clear ns acls [-type (classic | DFD)]
- renumber ns acls [-type (classic | DFD)]

IPv6 用の **CLI** コマンド

- add ns acl6 <acl6name> <acl6action> [-type (classic | dfd)][-dfdhash <dfdhash>][-dfdprefix <positive_interger>]

- `set ns acl6 <acl6name> <acl6action> [-dfdhash <dfdhash>][-dfdprefix <positive_interger>]`
- `show ns acl6 [<acl6name>][-type (classic | DFD)]`
- `apply ns acls6 [-type (classic | DFD)]`
- `clear ns acls6 [-type (classic | DFD)]`
- `renumber ns acls6 [-type (classic | DFD)]`

パケットをフロープロセッサに転送するために指定できるさまざまなタイプのハッシュメソッドは次のとおりです。

- SIP-SPORT-DIP-DPORT
- SIP
- DIP
- SIP-DIP
- SIP-SPORT

制限事項

1. フロープロセッサは管理者が設定したルールによって決定されるため、クラスタノード全体のトラフィックフローの分散は保証されません。
2. L2 モードはサポートされていません。
3. 展開シナリオがないため、ノードグループとストライピングされた SNIP はサポートされていません。
4. MPTCP はサポートされていません。
5. TCP、UDP、および ICMP トラフィックのみをサポートします。
6. L3 モードのクラスタはサポートされていません。
7. サービスレベルでのローカルプロセスはサポートされていません。

等コスト・マルチパス **(ECMP)** の使用

August 15, 2023

クラスターデプロイメントで等価コストマルチパス (ECMP) メカニズムを使用することで、アクティブなクラスターノードは仮想サーバーの IP アドレスをアドバタイズします。アドバタイズされたトラフィックを受信するクラスターノードは、トラフィックを処理する必要があるノードにトラフィックを誘導します。スポッティングされた仮想サーバーや部分的にストライプされた仮想サーバーでは、冗長なステアリングが可能です。そのため、NetScaler 11 以降では、スポットされた仮想サーバー IP アドレスと部分的にストライプされた仮想サーバーの IP アドレスが所有者ノードをアドバタイズするため、冗長なステアリングが軽減されます。

ECMP を使用するには、ルーティングプロトコルに関する詳細な知識が必要です。詳細については、ダイナミックルートの設定を参照してください。クラスター内のルーティングの詳細については、「クラスター内のルーティング」を参照してください。

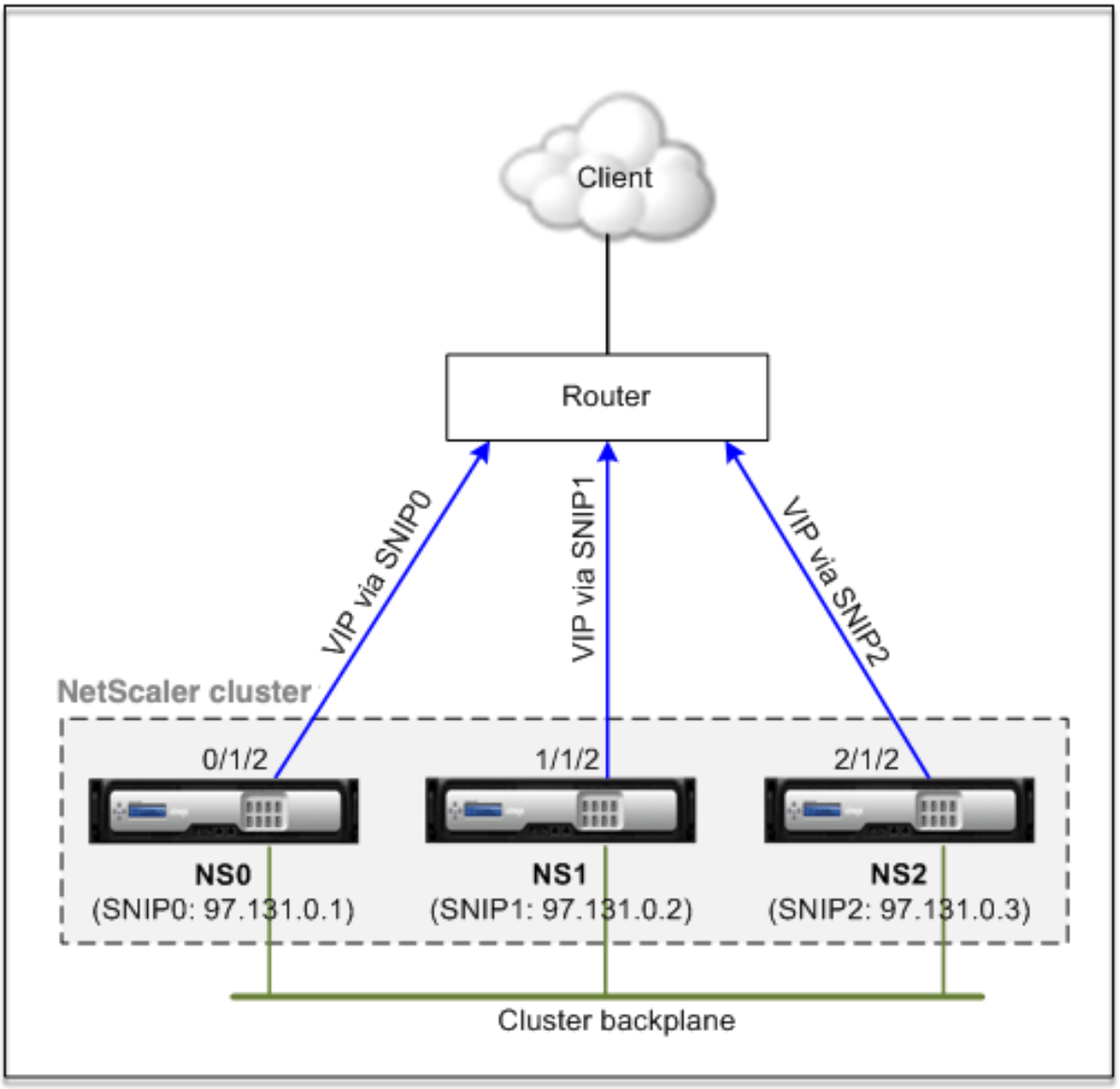
ECMP を使用するには、まず以下を実行する必要があります。

- クラスタ IP アドレスで必要なルーティングプロトコル (OSPF、RIP、BGP、または ISIS) を有効にします。
- インターフェイスとスポッティング IP アドレス (ダイナミックルーティングが有効な場合) を VLAN にバインドします。
- 選択したルーティングプロトコルを設定し、VTYSH シェルを使用して ZeBOS 上のカーネルルートを再配布します。

クラスタ IP アドレスと外部接続デバイスでも同様の設定を行います。

> 注
>
> - クラスターのライセンスが動的ルーティングをサポートしていることを確認してください。そうでない場合、ECMP は機能しません。
> - RHI はルーターとワイルドカード仮想サーバーにアドバタイズするための VIP アドレスを必要とするため、ECMP はワイルドカード仮想サーバーではサポートされていません。VIP アドレスが関連付けられていないためです。

図 1: ECMP トポロジ



クラスターデプロイメントでトラフィック分散に ECMP メカニズムを使用すると、アクティブなクラスターノードは仮想サーバーの IP アドレスをアップストリームルーターにアドバタイズします。ECMP ルーターは SNIP0、SNIP1、または SNIP2 経由で VIP アドレスに到達できます。図 1 のトラフィックフローは次のとおりです。

1. クライアントは、クラスターでホストされている VIP にリクエストを送信します。
2. アップストリームルーターは、学習した VIP のルートに基づいて、パケットをいずれかのノードに転送します。NS1 としましょう。ノード NS1 はフローレシーバーです。
3. フローレシーバ (NS1) は、トラフィックを処理する必要があるノード (フロープロセッサ) を決定します。たとえば、ノード NS2 はフロープロセッサです。
4. SNIP1（97.131.0.2）を備えたフローレシーバー（NS1）は、SNIP2（97.131.0.3）を備えたフロープロセッサ（NS2）にリクエストを誘導します。
5. フロープロセッサ (NS2) はサーバーとの接続を確立します。

6. サーバーは要求を処理し、要求を送信した SNIP アドレスに応答を送信します。

注記:

- アクティブノードのみが VIP ルートをアドバタイズします。
- 非アクティブノードは VIP ルートをアドバタイズしません。
- すべての ACTIVE ノードはストライプ VIP をアドバタイズします。
- アクティブな所有者ノードのみが、スポットされた VIP または部分的にストライプされた VIP をアドバタイズします。

コマンドラインインターフェイスを使用してクラスターで **ECMP** を構成するには

1. クラスタ IP アドレスにログオンします。
2. ルーティングプロトコルを有効にします。

```
enable ns feature <feature>
```

例: OSPF ルーティングプロトコルを有効にするには。

```
enable ns feature ospf
```

3. VLAN を追加します。

```
add vlan <id>
```

例

```
add vlan 97
```

4. クラスタノードのインタフェースを VLAN にバインドします。

```
bind vlan <id> -ifnum <interface_name>
```

例

```
bind vlan 97 -ifnum 0/1/2 1/1/2 2/1/2
```

5. 各ノードにスポッティング SNIP アドレスを追加し、そのノードで動的ルーティングを有効にします。

```
add ns ip <SNIP> <netmask> -ownerNode <positive_integer> -
   dynamicRouting ENABLED
```

例

```
add ns ip 97.131.0.1 255.0.0.0 -ownerNode 0 -dynamicRouting
   ENABLED -type SNIP
add ns ip 97.131.0.2 255.0.0.0 -ownerNode 1 -dynamicRouting
   ENABLED -type SNIP
```



```
3 add ns ip 97.131.0.3 255.0.0.0 -ownerNode 2 -dynamicRouting
    ENABLED -type SNIP
```

6. スポットのある SNIP アドレスの 1 つを VLAN にバインドします。1 つのスポット付き SNIP アドレスを VLAN にバインドすると、そのサブネット内のクラスタに定義されている他のすべてのスポット付き SNIP アドレスが自動的に VLAN にバインドされます。

```
1 bind vlan <id> -IPAddress <SNIP> <netmask>
```

例

```
1 bind vlan 97 -ipAddress 97.131.0.1 255.0.0.0
```

> 注
>
> SNIP アドレスを追加する代わりに、クラスターノードの NSIP アドレスを使用できます。その場合は、手順 3 ～6 を実行する必要はありません。

7. VTYSH シェルを使用して ZebOS でルーティングプロトコルを構成します。

例:

ノード ID 0、1、および 2 に OSPF ルーティングプロトコルを設定します。

```
1 vtysh
2 ! interface vlan97 !
3  router ospf   owner-node 0
4 ospf router-id 97.131.0.1   exit-owner-node
5 owner-node 1    ospf router-id 97.131.0.2
6 exit-owner-node
7 owner-node 2
8 ospf router-id 97.131.0.3   exit-owner-node   redistribute kernel
    network 97.0.0.0/8 area 0  !
```

> 注
>
> VIP アドレスがアドバタイズされるためには、次のように vServerRHIlevel パラメータを使用して RHI 設定を行います。

```
1 add ns ip <IPAddress> <netmask> -type VIP -vserverRHILevel <
    vserverRHILevel>
```

OSPF 固有の RHI 設定では、次のような設定が可能です。

```
1 add ns ip <IPAddress> <netmask> -type VIP -ospfLSAType \( TYPE1 |
    TYPE5 ) -ospfArea <positive\_integer>
```

IPv6 アドレスに対して前述のコマンドを実行するには、add ns ip6 コマンドを使用します。

8. 外部スイッチで ECMP を設定します。Cisco® Nexus 7000 C7010 リリース 5.2 (1) スイッチの設定例を以下に示します。他のスイッチでも同様の設定を行う必要があります。


```
//For OSPF (IPv4 addresses) Global config: Configure terminal
   feature ospf    Interface config:  Configure terminal
   interface Vlan10      no shutdown      ip address 97.131.0.5/8
      Configure terminal  router ospf 1  network 97.0.0.0/8 area
   0.0.0.0  ---------------------------------

//For OSPFv3 (IPv6 addresses) Global config:  Configure terminal
   feature ospfv3    Configure terminal  interface Vlan10      no
   shutdown      ipv6 address use-link-local-only      ipv6 router
    ospfv3 1 area 0.0.0.0    Configure terminal  router ospfv3 1
```

ECMP デプロイメントのルーター監視クラスターノード

クラスター設定で、スポットされた SNIP アドレス構成を持つ所有者ノードで、ownerDownResponse オプションを無効にできるようになりました。デフォルトでは、このオプションは有効になっており、ノードはアップストリームルーターからの ICMP/ARP/ICMP6/ND6 要求に応答できます。このオプションを無効にして、ルータがクラスタノードがアクティブか非アクティブかを監視できるようになりました。ルーターがリクエストを送信するときに、オプションが無効になっていると、所有者ノードが非アクティブでトラフィック分散に使用できないと識別されます。

コマンドラインインターフェイスを使用して **ECMP** をスタティックルートトラフィック分散に設定するには

```
add ns ip <ipddress> <netmask> -ownernode <node-id>  - ownerDownResponse
    disable
```

ユースケース:**ECMP** と **BGP** ルーティング

August 15, 2023

BGP ルーティングプロトコルで ECMP を設定するには、次の手順を実行します。

1. クラスタ IP アドレスにログオンします。
2. BGP ルーティングプロトコルを有効にします。

```
> enable ns feature bgp
```

3. VLAN を追加し、必要なインターフェイスをバインドします。

```
> add vlan 985
> bind vlan 985 -ifnum 0/0/1 1/0/1
```

4. スポッティング IP アドレスを追加し、VLAN にバインドします。


```
1 > add ns ip 10.100.26.14 255.255.255.0 -ownerNode 1 -
    dynamicRouting ENABLED
2 > add ns ip 10.100.26.15 255.255.255.0 -ownerNode 2 -
    dynamicRouting ENABLED
3 > bind vlan 985 -ipAddress 10.100.26.10 255.255.255.0
```

5. VTYSH シェルを使用して ZebOS で BGP ルーティングプロトコルを構成します。

```
1 > vtysh conf t router bgp 65535 neighbor 10.100.26.1 remote-as
    65535
```

6. 外部スイッチで BGP を設定します。Cisco® Nexus 7000 C7010 リリース 5.2 (1) スイッチの設定例を以下に示します。他のスイッチでも同様の設定を行う必要があります。

```
1 > router bgp 65535 no synchronization
2   bgp log-neighbor-changes neighbor 10.100.26.14 remote-as 65535
       neighbor 10.100.26.15 remote-as 65535 no auto-summary
3   dont-capability-negotiate
4   dont-capability-negotiate
5   no dynamic-capability
```

ルーティングプロトコルを備えた **Cisco Nexus 7000** スイッチを使用したクラスタ **ECMP** の設定

August 15, 2023

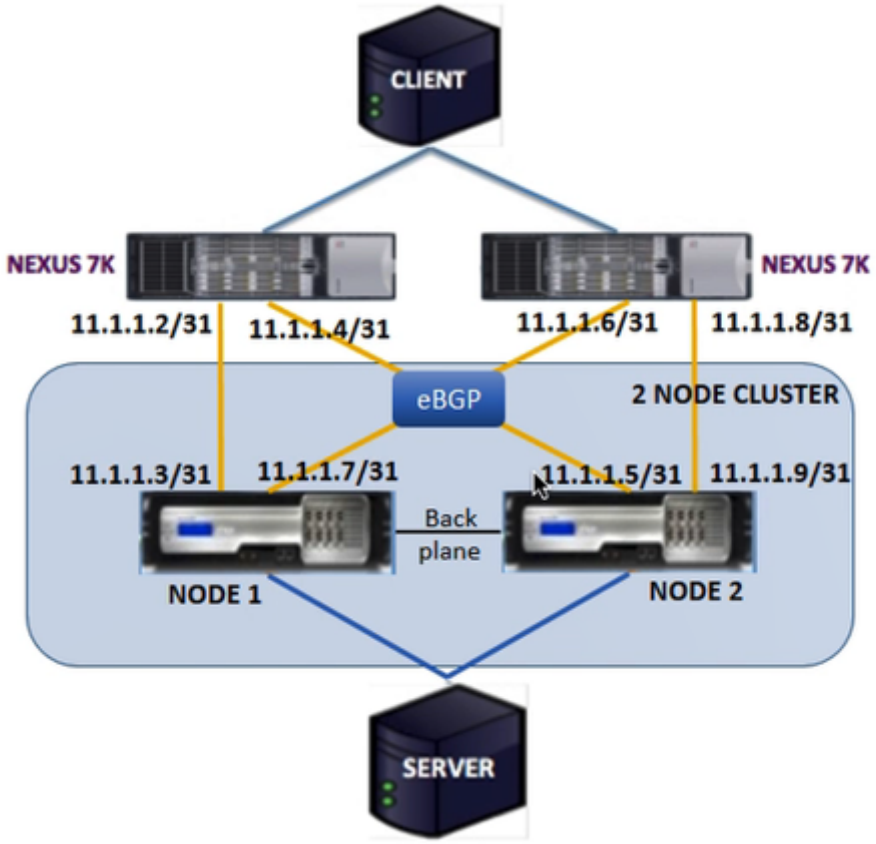
クラスターセットアップ上の ECMP を使用すると、NetScaler アプライアンスはルーティングプロトコルを介してトラフィックを処理できます。ECMP メカニズムは、すべてのアクティブなクラスターノードを通じて仮想サーバーの IP アドレスをアドバタイズするのに役立ちます。

ECMP を使用するには、まずクラスタ IP アドレスで BGP プロトコルを有効にする必要があります。インターフェイスとスポッティング IP アドレス (ダイナミックルーティングが有効な場合) を VLAN にバインドします。選択したルーティングプロトコルを設定し、VTYSH シェルを使用して ZeBOS 上のカーネルルートを再配布します。

使用事例：ルーティングプロトコルを備えた **Cisco Nexus 7000** スイッチを使用したクラスタ **ECMP**

Cisco Nexus 7000 スイッチを使用したクラスタ導入の例を考えてみましょう。

- Nexus スイッチ（アップストリーム）に接続された 2 つの NetScaler アプライアンス（ノード 1 とノード 2)。
- 2 台の Cisco ネクサス 7000 スイッチ。
- クライアントとサーバ（Nexus スイッチ経由の HTTP トラフィックの引き込み）クライアント側でホットスタンシールータプロトコル（HSRP）が有効になっている。



前提条件

NetScaler アプライアンスでクラスターノードを構成する前に、次の点を考慮してください。

1. すべてのアプライアンスは、同じプラットフォームタイプである必要があります。
2. ボーダーゲートウェイプロトコル (BGP) がクラスタノードで有効になっている必要があります。

NetScaler アプライアンスで **CLI** を使用して構成する

1. アプライアンス（たとえば、NSIP アドレスが 1.1.1.1 のアプライアンス）にログオンします
2. クラスターノードを追加します。

```
add cluster node 0 1.1.1.2 -state ACTIVE -backplane 0/10/8
```

3. クラスター IP アドレスを追加するには

```
add ns ip 1.1.1.10 255.255.255.254 -type clip
```

4. 構成を保存します


```
save ns config
```

5. アプライアンスのウォームリブート

```
reboot -warm
```

6. クリップ (CLIP) を使用してノード 1 を追加するには

```
add cluster node 1 2.2.2.2 -state ACTIVE -backplane 1/10/8
```

7. ノードをクラスタに参加するには

```
join cluster -clip 1.1.1.10 -password nsroot
```

8. CLIP で次の設定を行います。

   - `enable ns feature bgp ospf DYNAMICROUTING`
   - `add ns ip 11.1.1.3 255.255.255.254 -dynamicRouting ENABLED -ownerNode 0`
   - `add ns ip 11.1.1.7 255.255.255.254 -dynamicRouting ENABLED -ownerNode 0`
   - `add ns ip 11.1.1.5 255.255.255.254 -dynamicRouting ENABLED -ownerNode 1`
   - `add ns ip 11.1.1.9 255.255.255.254 -dynamicRouting ENABLED -ownerNode 1`

Cisco Nexus ルーター（11.1.1.2/31 および 11.1.1.4/31）で、コマンドラインを使用して次の構成を実行する必要があります:

- `feature ospf`
- `feature bgp`
- `feature interface-vlan`
- `feature hsrp`

```
>  interface vlan100
    no shutdown
    ip address 50.1.1.1/8
     hsrp 50
      ip 50.50.50.50

>  interface Ethernet 4/15
     ip address 11.1.1.2/31
     no shutdown

```



```
> interface Ethernet 4/19
  ip address 11.1.1.4/31
  no shutdown

> interface Ethernet 4/22
  switchport
   switchport access vlan 100
```

Cisco Nexus ルーター（11.1.1.6/31 および 11.1.1.8/31）で、コマンドラインを使用して次の構成を実行する必要があります:

- `feature ospf`
- `feature bgp`
- `feature interface-vlan`
- `feature hsrp`

```
> interface vlan100
  no shutdown
  no ip redirects
  ip address 50.1.1.2/8
   hsrp 50
   ip 50.50.50.50

> interface Ethernet 4/13
  ip address 11.1.1.6/31
  no shutdown

> interface Ethernet 4/15
  ip address 11.1.1.8/31
  no shutdown

> interface Ethernet 4/22
  switchport
  switchport access vlan 100
```

BGP プロトコルの場合、NetScaler ADC アプライアンスの CLIP で次の構成を実行する必要があります。

```
> vtysh
ns# router bgp 1
 redistribute kernel
 owner-node 0
 neighbor 11.1.1.2 remote-as 2
 neighbor 11.1.1.2 as-origination-interval 1
 neighbor 11.1.1.2 advertisement-interval 0
 neighbor 11.1.1.6 remote-as 2
 neighbor 11.1.1.6 as-origination-interval 1
 neighbor 11.1.1.6 advertisement-interval 0
owner-node 1
 neighbor 11.1.1.4 remote-as 2
 neighbor 11.1.1.4 as-origination-interval 1
```



```
 neighbor 11.1.1.4 advertisement-interval 0
 neighbor 11.1.1.8 remote-as 2
 neighbor 11.1.1.8 as-origination-interval 1
 neighbor 11.1.1.8 advertisement-interval 0
exit-owner-node
```

Cisco Nexus ルータ（11.1.1.3 および 11.1.1.5）で次の設定を実行します。

```
> ip access-list acl1
  10 permit ip 50.0.0.0/8 any
  route-map test permit
  match ip address acl1
 router bgp 2
  address-family ipv4 unicast
   redistribute direct route-map test
   maximum-paths 2
 neighbor 11.1.1.3 remote-as 1
  address-family ipv4 unicast
  neighbor 11.1.1.5 remote-as 1
 address-family ipv4 unicast
```

Cisco Nexus ルータ（11.1.1.7 および 11.1.1.9）で次の設定を実行します。

```
> ip access-list acl1
   10 permit ip 50.0.0.0/8 any
  route-map test permit 1
   match ip address acl1
  router bgp 2
   address-family ipv4 unicast
   redistribute direct route-map test
   maximum-paths 2
  neighbor 11.1.1.7 remote-as 1
   address-family ipv4 unicast
  neighbor 11.1.1.9 remote-as 1
   address-family ipv4 unicast
```

OSPF プロトコルの場合、NetScaler ADC アプライアンスの CLIP で次の構成を実行する必要があります。

```
> vtysh
ns# router osfp 1
redistribute kernel
owner-node 0
 network 15.1.1.2/31 area 0
 network 15.1.1.6/31 area 0
exit-owner-node

owner-node 1
 network 15.1.1.4/31 area 0
 network 15.1.1.8/31 area 0
exit-owner-node

route-map map2 permit 1
set metric 10
```


Cisco Nexus ルーター（11.1.1.2/31 および 11.1.1.4/31）で、コマンドラインを使用して次の構成を実行する必要があります:

```
> route-map- map2 permit 1
   set metric 10

   interface Ethernet4/15
     ip address 15.1.1.2/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

   interface Ethernet4/19
     ip address 15.1.1.4/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

   router ospf 1
    router-id 1.1.1.1
    redistribute direct route-map map2
```

Cisco Nexus ルータ（11.1.1.7/31 および 11.1.1.9/31）では、コマンドラインを使用して次の設定を実行する必要があります。

```
> route-map- map2 permit 1
    set metric 10

   interface Ethernet4/13
     ip address 15.1.1.6/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

    interface Ethernet4/15
     ip address 15.1.1.8/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

    router ospf 1
      router-id 1.1.1.2
      redistribute direct route-map map2
```

クラスターリンクアグリゲーションの使用

August 15, 2023

クラスターリンクアグリゲーションは、クラスターノードのインターフェースのグループです。これは NetScaler リンクアグリゲーションの拡張機能です。唯一の違いは、リンクアグリゲーションでは、同じデバイスからのインターフェイスが必要ですが、クラスタリンクアグリゲーションでは、インターフェイスはクラスタの異なるノードからのものであることです。リンクアグリゲーションの詳細については、「リンク集約の設定」を参照してください。

重要

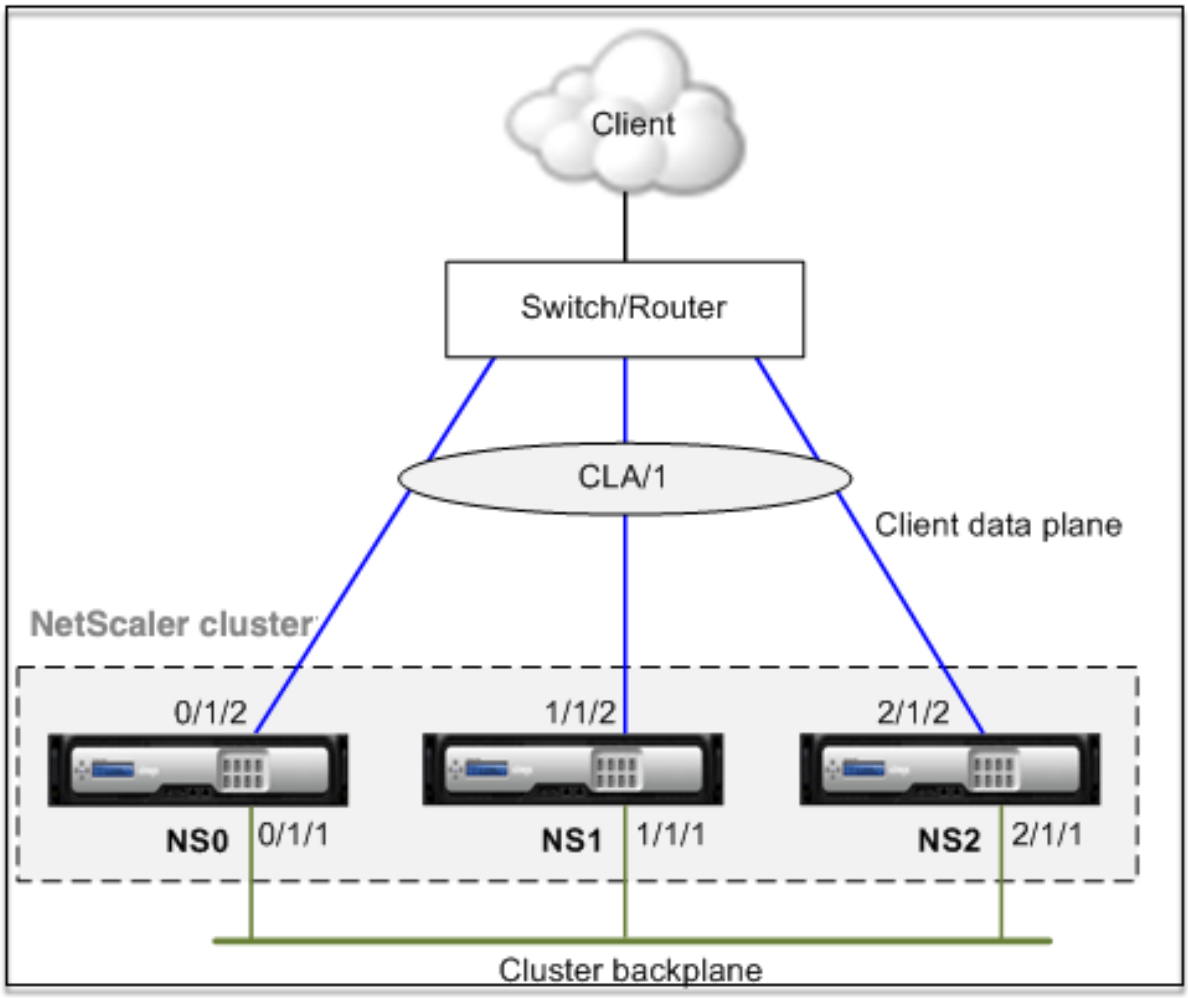
- クラスタリンクアグリゲーションは、クラスタオブハードウェア（MPX）アプライアンスでサポートされています。
- クラスタリンクアグリゲーションは、ESX および KVM ハイパーバイザーに導入された仮想（VPX）アプライアンスのクラスタでサポートされますが、次の制限があります。
- 専用のインターフェースを使用する必要があります。つまり、インターフェースを他の仮想マシンと共有してはいけないということです。
- ノードが非アクティブになると、対応するクラスタ LA インターフェイスがパワーダウンとしてマークされるため、データトラフィックは非アクティブノードに送信されません。
- ノードが ACTIVE になると、対応するクラスタ LA インターフェイスが電源オンとマークされます。
- クラスタリンクアグリゲーションのメンバーインターフェイスを手動で無効にするか、クラスタリンクアグリゲーション自体を手動で無効にした場合、インターフェイスのパワーダウン機能は LACP タイムアウトメカニズムによってのみ実現されます。
- ジャンボ MTU は LACP クラスタリンクアグリゲーションではサポートされていません。

注：クラスターリンクアグリゲーションは、XenServer、AWS、および Hyper-V にデプロイされている VPX アプライアンスではサポートされていません。

- 12.0 リリース以降、NetScaler SDX アプライアンスではクラスターリンクアグリゲーションがサポートされます。
- クラスタ LA にバインドできるインターフェースの数は 16 (各ノードから) です。クラスタ LA のインターフェイスの最大数は (16 * n) です。ここで n はクラスタ内のノード数です。クラスタ LA のインターフェイスの総数は、アップストリームスイッチ上の各ポートチャネルのインターフェイス数によって異なります。
- NetScaler アプライアンスが Intel Fortville インターフェイスを使用している場合、クラスターノードをパッシブモードに切り替えると、CLAG が数秒停止する可能性があります。この問題が発生するのは、CLAG が正常に機能するために LACP が有効になっていて、停止時間は NIC LACP タイマーによって異なるためです。

たとえば、3 つのノードすべてがアップストリームスイッチに接続されている 3 ノードクラスタを考えてみましょう。クラスタ LA チャネル（CLA/1）は、インターフェイス 0/1/2、1/1/2、および 2/1/2 をバインドすることによって形成されます。

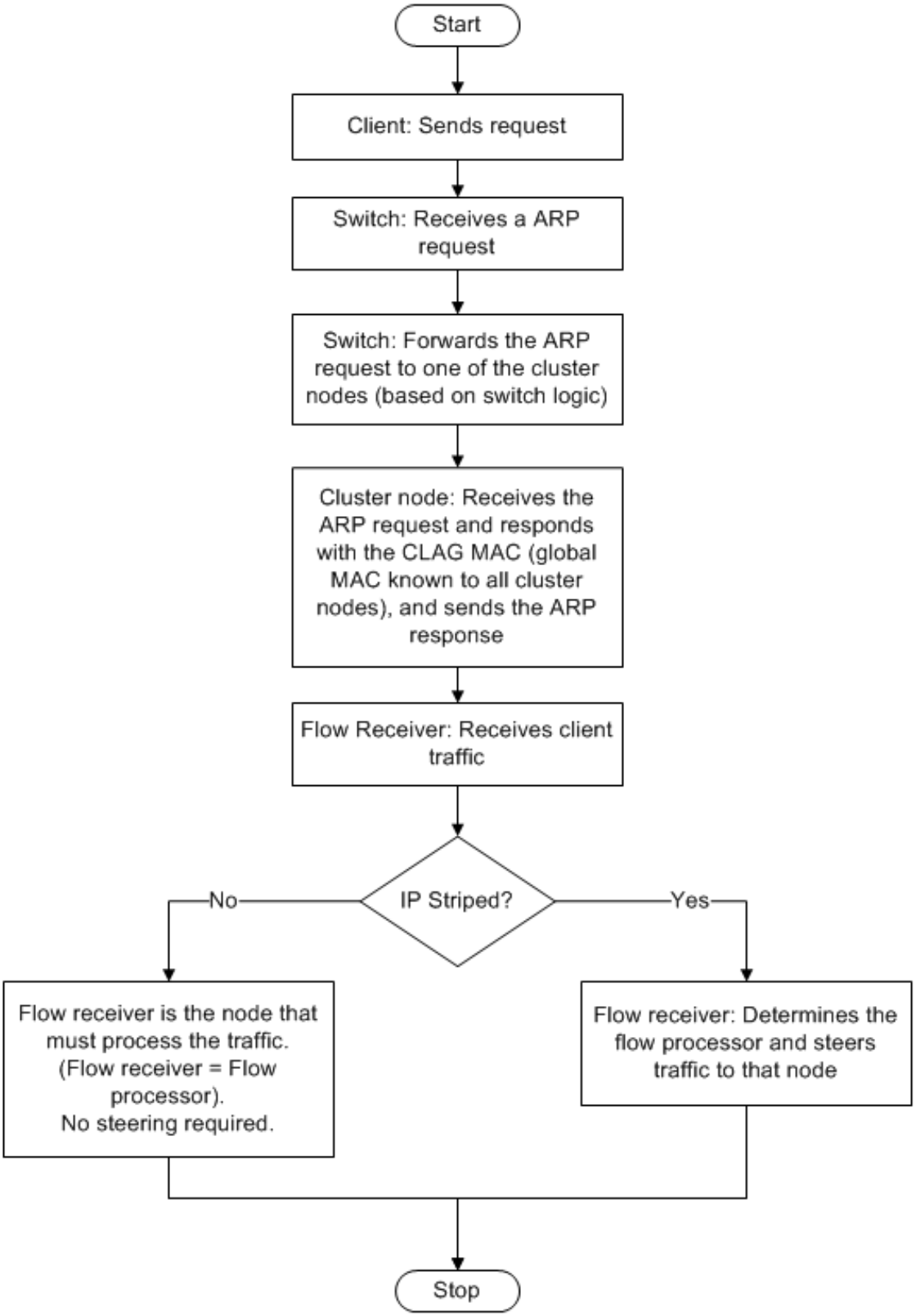
図 1：クラスタリンクアグリゲーショントポロジー



クラスタ LA チャネルには次の属性があります。

- 各チャネルには、クラスタノードによって合意された一意の MAC が割り当てられます。
- このチャネルは、ローカルノードとリモートノードの両方のインターフェースをバインドできます。
- 1 つのクラスタでは、最大 4 つのクラスタ LA チャネルがサポートされます。
- バックプレーンインターフェイスをクラスタ LA チャネルに含めることはできません。
- インターフェイスがクラスタ LA チャネルにバインドされている場合、チャネルパラメータはネットワークインターフェイスパラメータよりも優先されます。ネットワークインターフェースは 1 つのチャネルにのみバインドできます。
- クラスタノードへの管理アクセスは、クラスタ LA チャネル（たとえば、CLA/1）またはそのメンバーインターフェイス上で設定しないでください。これは、ノードが非アクティブになると、対応するクラスタ LA インターフェイスがパワーダウンとマークされ、管理アクセスができなくなるためです。

図 2: クラスタ LA を使用したトラフィック分散フロー



NetScaler MPX でのクラスター LA のバックアップとリストアのサポート

NetScaler MPX 上の LA のクラスターセットアップをバックアップおよび復元できます。クラスタ LA MAC アドレスは、クラスタノードの物理インターフェイス MAC アドレスとは独立しており、バックアップとリストアの処理後に変更される可能性があります。クラスタ LA は、クラスタ復元プロセスが完了した後、トラフィックを処理できます。バックアップと復元の詳細については、「クラスタ設定のバックアップと復元」を参照してください。

静的クラスターリンクアグリゲーション

August 15, 2023

スタティッククラスタ LA チャネルは、クラスタ IP アドレスと外部接続デバイスに設定する必要があります。可能であれば、MAC アドレスではなく IP アドレスまたはポートに基づいてトラフィックを分散するようにアップストリームスイッチを設定します。

CLI を使用して静的クラスタ LA チャネルを構成するには

1. クラスタ IP アドレスにログオンします。

   > 注
   >
   > 外部スイッチでリンクアグリゲーションを設定する前に、必ずクラスタ IP アドレスにクラスタ LA チャネルを設定してください。そうしないと、クラスタ LA チャネルが設定されていなくても、スイッチはトラフィックをクラスタに転送します。トラフィックの損失につながる可能性があります。

2. クラスタ LA チャネルを作成します。

   ```
   add channel <id> -speed <speed>
   ```

   例

   ```
   add channel CLA/1 -speed 1000
   ```

   > 注
   >
   > 速度を AUTO として指定しないでください。むしろ、速度を 10、100、1000、または 10000 として明示的に指定する必要があります。クラスタ LA チャネルの <speed> 属性に一致する速度を持つインターフェイスのみがアクティブな配布リストに追加されます。

3. 必要なインターフェイスをクラスタ LA チャネルにバインドします。これらのインターフェイスがクラスタバックプレーンに使用されていないことを確認します。

   ```
   bind channel <id> <ifnum>
   ```


例

```
bind channel CLA/1 0/1/2 1/1/2 2/1/2
```

4. 設定を確認します。

```
show channel <id>
```

例

```
show channel CLA/1
```

> 注
>
> `bind vlan` コマンドを使用して、クラスタ LA チャネルを VLAN にバインドできます。チャネルのインターフェイスは自動的に VLAN にバインドされます。

5. 外部スイッチにスタティック LA を設定します。Cisco ® Nexus 7000 C7010 リリース 5.2 (1) のサンプル設定は次のとおりです。他のスイッチでも同様の設定を行う必要があります。

```
Global config:
Configure terminal

Interface level config:

interface Ethernet2/47
switchport
switchport access vlan 10
channel-group 7 mode on
no shutdown

interface Ethernet2/48
switchport
switchport access vlan 10
channel-group 7 mode on
no shutdown
```

動的クラスターリンクアグリゲーション

August 15, 2023

動的クラスタ LA チャネルは、Link Aggregation Control Protocol (LACP) を使用します。

クラスタ IP アドレスと外部接続デバイスでも同様の設定を行う必要があります。可能であれば、MAC アドレスではなく IP アドレスまたはポートに基づいてトラフィックを分散するようにアップストリームスイッチを設定します。

> 確認事項
>
> - LACP を有効にします（LACP モードを ACTIVE または PASSIVE に指定します）。

```
>**Note**
>
> Make sure the LACP mode is not set as PASSIVE on both the NetScaler
    cluster and the external connecting device.
```

> - チャネルの一部にする各インターフェイスで、同じ LACP キーを指定します。クラスタ LA チャネルを作成する場合、LACP キーには 5 ～8 の値を指定できます。たとえば、インターフェイス 0/1/2、1/1/2、および 2/1/2 の LACP キーを 5 に設定すると、CLA/1 が作成されます。インターフェイス 0/1/2、1/1/2、および 2/1/2 は自動的に CLA/1 にバインドされます。同様に、LACP キーを 6 に設定すると、CLA/2 チャネルが作成されます。
> - LAG タイプをクラスターとして指定します。

CLI を使用して動的クラスタ LA チャネルを設定するには

クラスタ IP アドレスに、クラスタ LA チャネルに追加する各インターフェイスについて、次のように入力します。

```
set interface <id> -lacpMode <lacpMode> -lacpKey <positive_integer> -
lagType CLUSTER<!--NeedCopy-->
```

例:

3 つのインターフェイスのクラスタ LA チャネル CLA/1 を設定します。

```
> set interface 0/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 1/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 2/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
```

> 注
>
> オプションで、LACP を使用してクラスタのリンク冗長性を有効にできます。

同様に、外部スイッチでダイナミック LA を設定します。Cisco ® Nexus 7000 C7010 リリース 5.2 (1) のサンプル設定は次のとおりです。他のスイッチでも同様の設定を行う必要があります。

```
Global config:
Configure terminal
feature lacp
Interface level config:

interface Ethernet2/47
switchport
switchport access vlan 10
channel-group 7 mode active
no shutdown
```



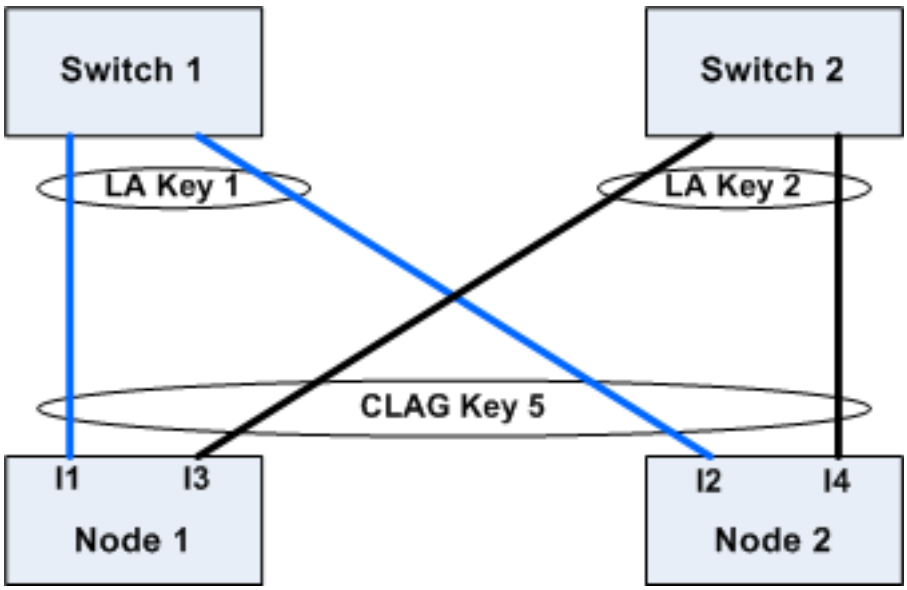
```
11
12 interface Ethernet2/48
13 switchport
14 switchport access vlan 10
15 channel-group 7 mode active
16 no shutdown
```

LACP を使用したクラスターでのリンク冗長性

August 15, 2023

NetScaler クラスターは LACP のリンク冗長性を提供し、すべてのノードが同じパートナーキーを持つようにします。

リンク冗長性の必要性を理解するために、次のクラスタ設定の例とそれに付随するケース（ケース 3 に注目して）を考えてみましょう。



この設定では、インターフェイス I1、I2、I3、および I4 が KEY 5 で LACP チャネルにバインドされます。相手側では、I1 と I2 がスイッチ 1 に接続され、KEY 1 の 1 つの LA チャネルを形成します。同様に、I3 と I4 はスイッチ 2 に接続され、KEY 2 で 1 つの LA チャネルを形成します。

次に、リンク冗長性の必要性を理解するために、次のケースを考えてみましょう。

- ケース **1**: スイッチ **1** はアップで、スイッチ **2** はダウン

  この場合、両方のノードのクラスタ LA は Key2 からの LACPDU の受信を停止し、Key1 からの LACPDU の受信を開始します。両方のノードで、クラスタ LA が KEY 1 と I1 と I2 に接続され、両方のノードのチャネルは UP になります。

- ケース **2**: スイッチ **1** がダウンし、スイッチ **2** がアップ状態になる

  この場合、両方のノードのクラスタ LA は Key1 からの LACPDU の受信を停止し、Key2 からの LACPDU の受信を開始します。両方のノードで、クラスタ LA は Key2 と I3 に接続され、I4 は稼働しており、両方のノードのチャネルは稼働しています。

- ケース **3**: スイッチ **1** とスイッチ **2** の両方が稼働している

  この場合、ノード 1 のクラスタ LA がパートナーとして Key1 を選択し、ノード 2 のクラスタ LA がパートナーとして Key2 を選択する可能性があります。これは、ノード 1 の I1 とノード 2 の I4 が望ましくないトラフィックを受信していることを意味します。これは、LACP ステートマシンがノードレベルで、先着順でパートナーを選択するからです。

  これらの問題を解決するために、動的クラスタ LA のリンク冗長性がサポートされています。チャネルまたはインターフェイスにリンク冗長性を設定するには、リンク冗長性を有効にし、オプションで次のようにスループットのしきい値を指定する必要があります。

  ```
  set channel CLA/1 -linkRedundancy ON -lrMinThroughput <positive_integer
  >
  ```

  パートナーチャネルのスループットは、設定されたスループットのしきい値と照合されます。スループットのしきい値を満たすパートナーチャネルは、先入れ先出し（FIFO）方式で選択されます。どのパートナーチャネルもしきい値を満たさない場合、またはスループットのしきい値が設定されていない場合は、リンク数が最大であるパートナーチャネルが選択されます。

  > 注
  >
  > スループットのしきい値は、NetScaler 11 以降で構成できます。

クラスターでの **USIP** モードの使用

August 15, 2023

ソース IP（USIP）モードでは、クラスタまたは NetScaler ADC アプライアンスが各パケットをクライアント IP アドレスで適切なバックエンドサーバーに転送します。

USIP モードのトラフィック分散

USIP モードの動作は、ECMP および CLAG 展開では、クライアントデータプレーンとサーバデータプレーン間でトラフィックの分散が異なります。次のセクションでは、USIP モードの動作について詳しく説明します。USIP モードでの CLAG の詳細については、「クラスタリンク集約の使用」を参照してください。

USIP モード

クラスターは、クライアント IP を使用してサーバー側接続を開きます。送信元ポートは、useproxyport 設定に基づいて保持される場合とされない場合があります。

USIP useproxyport シナリオ

USIP useproxyport がトラフィックフローで ON の場合、送信元ポートはリバーストラフィックがフロープロセッサにハッシュされるように選択されます。サーバー側でシングルステアリングを確実にします。

USIP useproxyport はトラフィックフローに対してオフになり、送信元ポートは保持されるため、サーバ側にダブルステアリングがあります。

> 重要
>
> - USIP がオンの場合、クライアント IP はバックエンドサーバー接続で使用され、応答のためのトラフィック分散がクラスタノード間で必要になります。ECMP または CLAG 展開は、サーバ側のトラフィック分散に使用できます。サーバー側でトラフィックが分散されていない場合、リターントラフィック全体が単一のクラスタノードに着陸し、輻輳が発生する可能性があります。
> - set rsskeytype -rsskey symmetricコマンドは、useproxyportオフ展開でのトラフィックのダブルステアリングをシングルステアリングに減らすために使用されます。サーバー側とクライアント側で、接続の 4 タプルが同じままです。たとえば、ワイルドカード MAC モード仮想サーバーなどです。

制限事項

プロセスローカルが無効になっている場合、USIP は機能しません。

USIP モードデプロイ

次の図は、クラスタセットアップでの USIP モードの展開を示しています。

CLI を使用して次の設定を行います

1. ルーティングプロトコルを有効にします。

```
enable ns feature <feature>
```

例:

```
enable ns feature ospf
```


2. 各ノードにスポッティング SNIP アドレスを追加し、そのノードで動的ルーティングを有効にします。

```
add ns ip <SNIP> <netmask> -dynamicRouting \( ENABLED | DISABLED ) -ownerNode <positive\_interger> -ownerdownResponse \( YES | NO )
```

例

```
- add ns ip 192.0.2.1 255.255.255.0 -dynamicRouting ENABLED -ownerNode 0 -ownerDownResponse NO
- add ns ip 192.0.2.2 255.255.255.0 -dynamicRouting ENABLED -ownerNode 1 -ownerDownResponse NO
- add ns ip 192.0.2.3 255.255.255.0 -dynamicRouting ENABLED -ownerNode 2 -ownerDownResponse NO
```

3. VLAN を追加します。

```
add vlan <id>
```

例

```
add vlan 300
```

4. クラスタノードのインタフェースを VLAN にバインドします。

```
bind vlan <id> -ifnum <interface_name>
```

例

```
bind vlan 300 -ifnum 0/1/2 1/1/2 2/1/2
```

5. スポットのある SNIP アドレスの 1 つを VLAN にバインドします。1 つのスポット付き SNIP アドレスを VLAN にバインドすると、そのサブネット内のクラスタに定義されている他のすべてのスポット付き SNIP アドレスが自動的に VLAN にバインドされます。

```
bind vlan <id> -IPAddress <ip\_addr | ipv6\_addr> -netmask
```

例

```
bind vlan 300 -IPAddress 192.0.2.1 255.255.255.0
```

6. VTYSH シェルを使用して ZebOS でルーティングプロトコルを構成します。ノード ID 0、1、および 2 に OSPF ルーティングプロトコルを設定します。

```
vtysh
configure terminal
ns block-sec-rtadv
router ospf
owner -node 0
router-id 192.0.2.1
exit-owner-node
owner-node 1
```



```
 9  router-id 192.0.2.2
10  exit-owner-node
11  owner-node 2
12  router-id 192.0.2.3
13  exit-owner-node
14  network 192.0.2.0/24 area 0
15
16  default-information originate always
```

7. CLI を使用して、Cisco 3750 ルータで次の設定を実行します。

```
1  Configure terminal
2  feature ospf
3  interface vlan300
4  no shutdown
5  ip address 192.0.2.100/24
6  Configure terminal
7  router ospf 1
8  router-id 192.0.2.100
9  network 192.0.2.0 0.0.0.255 area 0
```

> メモ
>
> - クライアントとサーバーでのトラフィックの分散は同じである必要はありません。たとえば、クライアント側で ECMP を設定し、サーバ側に CLAG を設定したり、反対の方法で設定したりできます。
> - USIP 展開ではステアリングオーバーヘッドが増えるため、バックプレーンの追加容量を計画してください。
> - CLAG およびモニタスタティックルート（MSR）に関連する設定は、サーバ側で同じままにしておく必要があります。
> - USIP モードの展開では、トラフィックステアリングがさらに高くなります。

NetScaler ADC クラスタの管理

August 15, 2023

クラスタを作成し、必要なトラフィック分散メカニズムを設定すると、クラスタはトラフィックを処理できるようになります。クラスターの存続期間中、次のクラスタータスクを実行できます。

- ノードグループの設定
- クラスターのノードを無効にする
- NetScaler ADC アプライアンスの検出
- 統計を表示する
- クラスター構成とクラスターファイルの同期

- ノード間の時間の同期
- クラスタノードのソフトウェアのアップグレードまたはダウングレード

リンクセットの構成

August 15, 2023

リンクセットは、同じブロードキャストドメインに属するクラスターノードのインターフェースのグループです。リンクセットでは、各ノードには、他のノードのどのインターフェイスが同じブロードキャストドメインに接続されているかに関する情報があります。

> 注
>
> リンクセットは、次のシナリオでは必須の設定です。
>
> - MAC ベースの転送 (MBF) を必要とするデプロイメント用。
> - 仮想サーバーで有効になっている「-m MAC」モードとグローバルに有効な MBF モードの場合。
> - インターフェイスに関する ACL および L2 ポリシーの管理性を向上させるため。インターフェイスのリンクセットを定義し、リンクセットに基づいて ACL ポリシーと L2 ポリシーを追加します。

クラスターセットアップでは、次の機能が内部で MBF を使用します。

- 転送セッション
- L2Conn
- MAC モード仮想サーバー
- 透明モニター
- LLB

リンクセットは、クラスタ IP アドレスからのみ構成する必要があります。

3 ノードクラスターの例を考えてみましょう。次の図では、インターフェイス 0/1/2、1/1/2、および 2/1/2 は同じブロードキャストドメインにあるため、リンクセット（LS/1）として設定できます。

図 1：リンクセットトポロジ

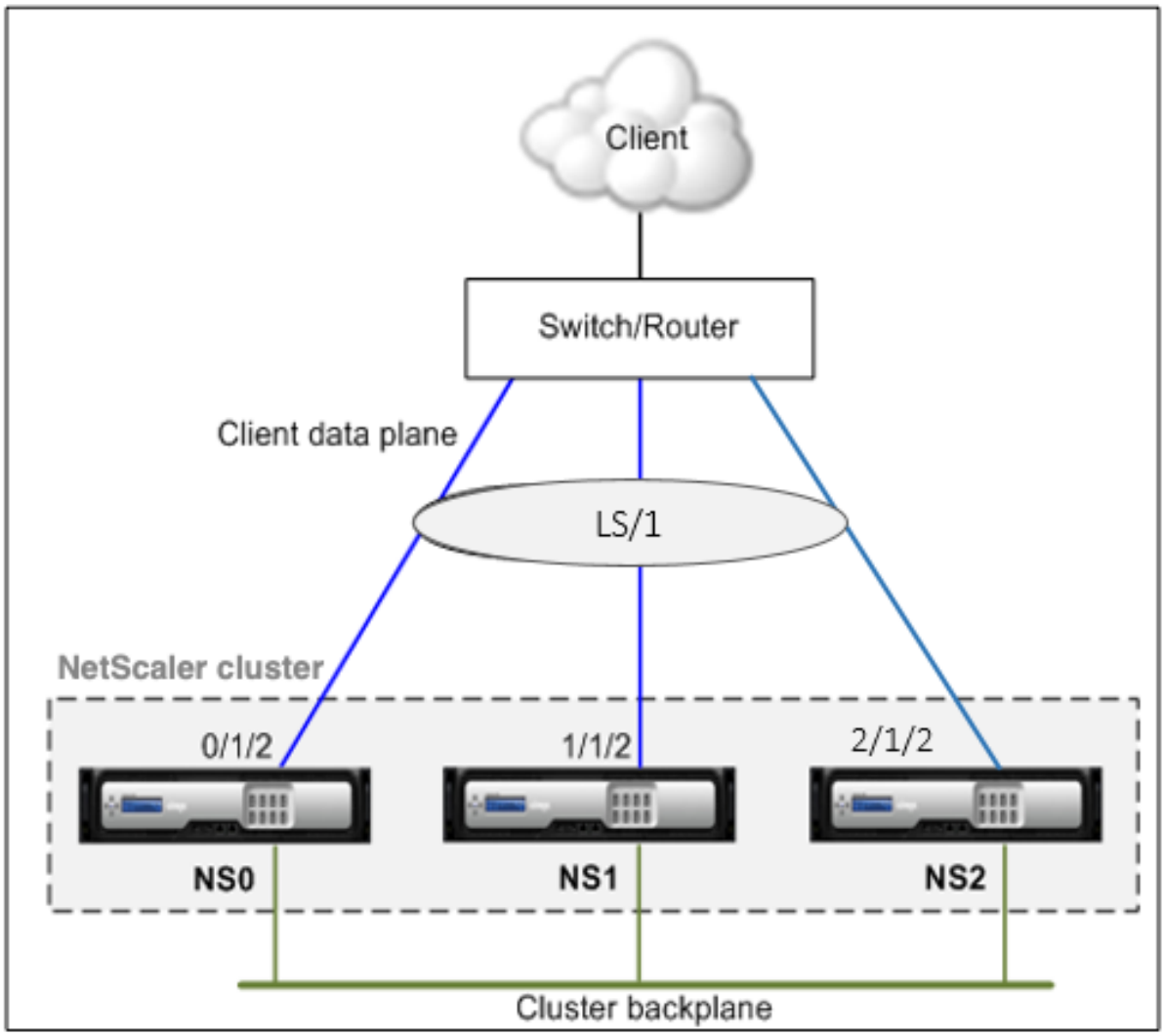
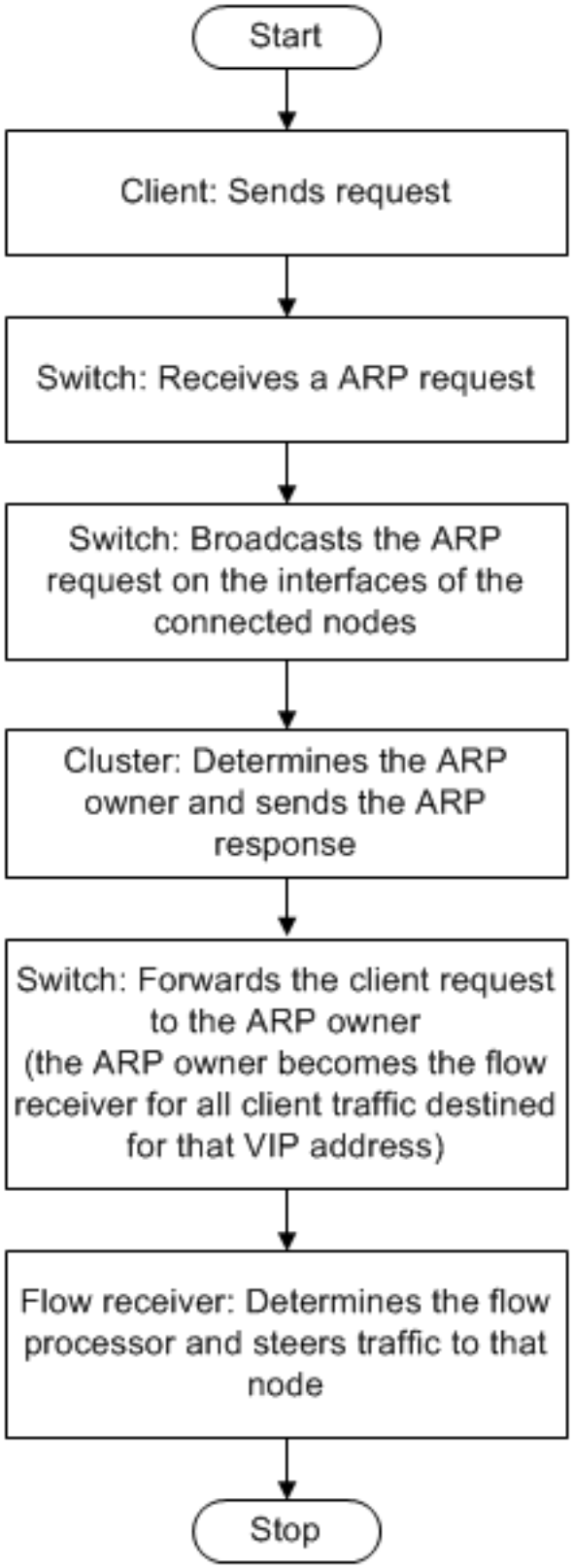


図 2：リンクセットを使用したトラフィック分散フロー



CLI を使用してリンクセットを設定するには

1. クラスタ IP アドレスにログオンします。
2. リンクセットを作成します。

   “‘add linkset

   ```
   ** 例 **

   ```add linkset LS/1<!--NeedCopy-->
   ```

3. 必要なインターフェースをリンクセットにバインドします。インターフェイスがクラスタバックプレーンに使用されていないことを確認してください。

   “‘bind linkset -ifnum ...

   ```
   ** 例 **

   ```bind linkset LS/1 -ifnum 0/1/2 1/1/2 2/1/2<!--NeedCopy-->
   ```

4. リンクセットの設定を確認します。

   “‘show linkset

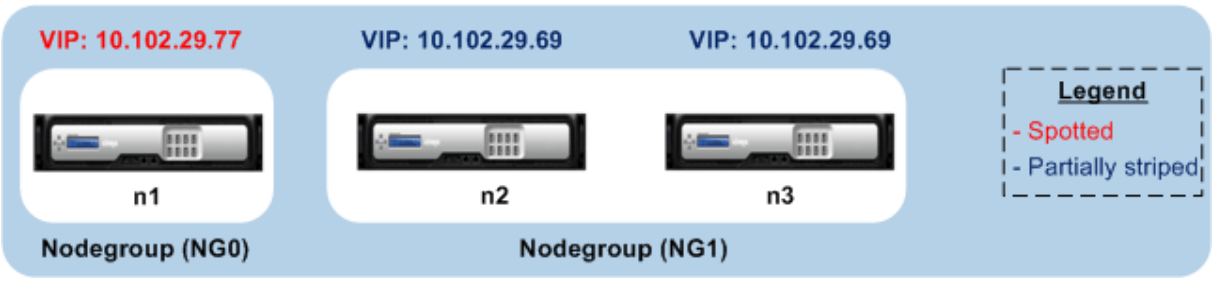
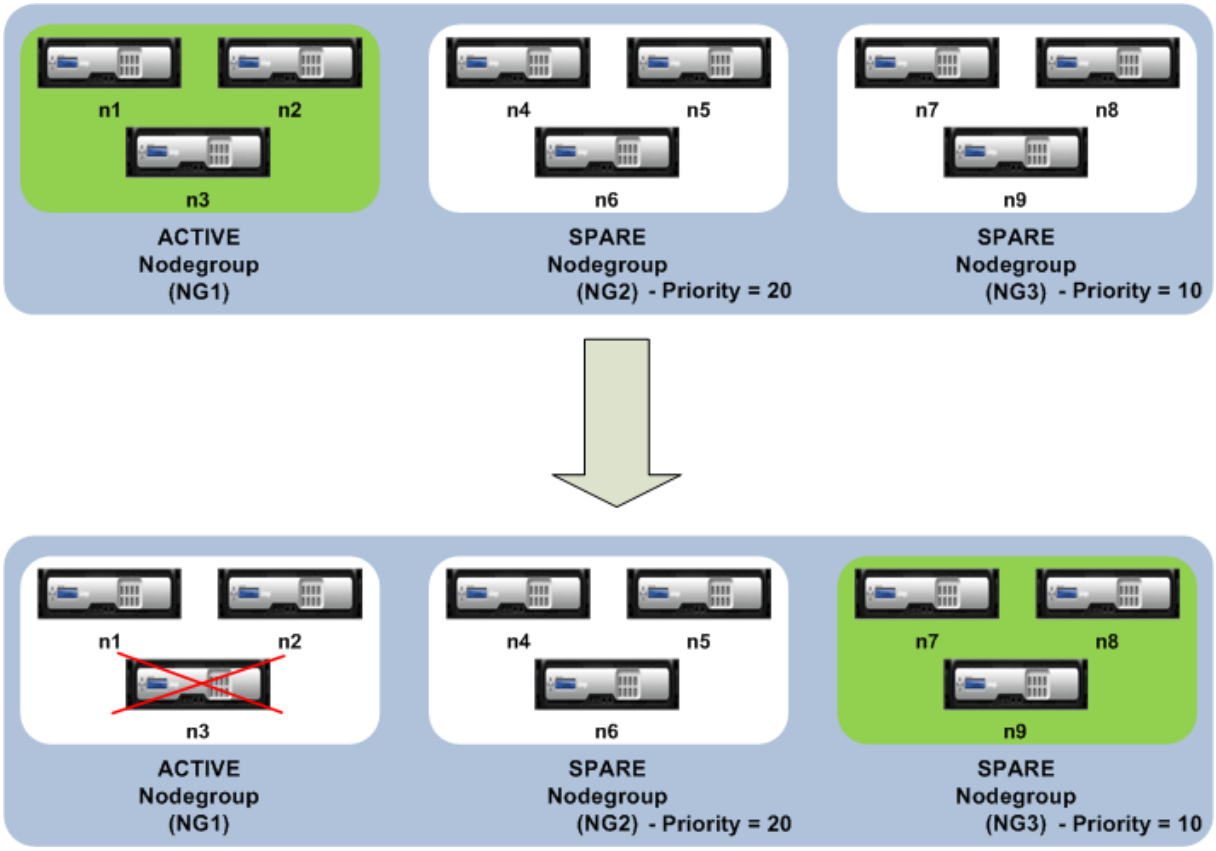
   ```
   ** 例 **

   ```show linkset LS/1<!--NeedCopy-->
   ```

   > 注
   >
   > `bind vlan` コマンドを使用して、リンクセットを VLAN にバインドできます。リンクセットのインターフェイスは、自動的に VLAN にバインドされます。

GUI を使用してリンクセットを構成するには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [ネットワーク] **>** [リンクセット] に移動します。
3. 詳細ペインで、[追加] をクリックします。
4. 「リンクセットの作成」ダイアログ・ボックスで:
   - Link set パラメータを設定して、リンク セットの名前を指定します。
   - リンクセットに追加する インターフェイスを 指定し、[Add] をクリックします。リンクセットに追加するインターフェイスごとに、この手順を繰り返します。
5. [作成] をクリックし、[閉じる] をクリックします。


   ```

# スポット構成と部分ストライプ構成のノードグループ

August 15, 2023

デフォルトのクラスター動作により、クラスター IP アドレスで実行されるすべての構成は、クラスターのすべてのノードで使用できます。ただし、一部の構成を特定のクラスターノードでのみ使用できるようにする必要がある場合があります。

この要件は、特定のクラスターノードを含むノードグループを定義し、そのノードグループに構成をバインドすることで実現できます。これにより、構成はそれらのクラスターノードでのみアクティブになります。これらの構成は、部分的なストライプまたはスポットと呼ばれます（アクティブな場合は単一ノードが 1 つだけ）。詳細については、ストライプ、部分的にストライプ、およびスポッティングされた設定を参照してください。

たとえば、3 つのノードを持つクラスターを考えてみましょう。ノード n1 を含むノードグループ NG0 と、n2 と n3 を含む別のノードグループ NG1 を作成します。負荷分散仮想サーバー 0.77 を NG0 に、負荷分散仮想サーバー 0.69 を NG1 にバインドします。

つまり、仮想サーバー 0.77 は n1 でのみアクティブであるため、0.77 宛のトラフィックを受信するのは n1 だけです。同様に、仮想サーバー 0.69 はノード n2 と n3 でのみアクティブであるため、0.69 宛のトラフィックを受信するのは n2 と n3 だけです。

図 1: スポット構成と部分ストライプ構成用に構成されたノードグループを含む NetScaler クラスター




ノードグループにバインドできるエンティティまたは構成は次のとおりです。

- 負荷分散、コンテンツスイッチング、キャッシュリダイレクト、認証、承認、および監査

  > 注
  >
  > FTP 負荷分散仮想サーバーはノードグループにバインドできません。

- VPN 仮想サーバー（NetScaler 10.5 ビルド 50.10 以降をサポート）
- グローバルサーバー負荷分散（GSLB）サイトおよびその他の GSLB エンティティ（NetScaler 10.5 ビルド 52.11 以降でサポート）
- リミット識別子とストリーム識別子

 

# ノードグループの動作

August 15, 2023

ノードグループは異なる NetScaler の機能やエンティティと相互運用性があるため、注意すべき動作上の側面がいくつかあります。ノードグループ内のノードもバックアップできます。詳細については読んでください。

## クラスターノードグループの一般的な動作

- エンティティがバインドされているノードグループは削除できません。
- エンティティがバインドされたノードグループに属するクラスターノードは削除できません。
- エンティティがバインドされたノードグループを含むクラスターインスタンスは削除できません。
- 別のエンティティに依存しているエンティティは追加できません。ノードグループの一部であってはなりません。必要な場合は、まず依存関係を削除してください。次に、両方のエンティティをノードグループに追加し、エンティティを再関連付けします。

  例:

  - 仮想サーバー VS1 があり、そのバックアップが仮想サーバー VS2 であるとします。VS1 をノードグループに追加するには、まず VS2 が VS1 のバックアップサーバーとして削除されていることを確認します。次に、各サーバーをノードグループに個別にバインドし、VS2 を VS1 のバックアップとして構成します。
  - ターゲット負荷分散仮想サーバーが LBVS1 であるコンテンツスイッチ仮想サーバー CSVS1 があるとします。CSVS1 をノードグループに追加するには、まず LBVS1 をターゲットから削除します。次に、各サーバーをノードグループに個別にバインドし、LBVS1 をターゲットとして構成します。
  - 負荷分散仮想サーバー LBVS1 があり、そのポリシーに別の負荷分散仮想サーバー LBVS2 を呼び出すポリシーが設定されているとします。いずれかの仮想サーバーを追加するには、まず関連付けを削除します。次に、各サーバーをノードグループに個別にバインドし、仮想サーバーを再度関連付けます。

- エンティティをノードグループにバインドすることはできません。ノードがなく、strict オプションが有効になっています。そのため、エンティティがバインドされていて strict オプションが有効になっているノードグループの最後のノードをバインド解除することはできません。
- strict オプションは、ノードがないのにエンティティがバインドされているノードグループでは変更できません。

## ノードグループ内のノードのバックアップ

デフォルトでは、ノードグループはノードグループのメンバーにバックアップノードを提供するように設計されています。ノードグループのメンバーがダウンした場合、そのノードグループのメンバーではないクラスターノードが、

 

障害が発生したノードを動的に置き換えます。このノードは置換ノードと呼ばれます。

> 注
>
> 単一メンバーのノードグループの場合、エンティティがノードグループにバインドされると、バックアップノードが自動的に事前に選択されます。

ノードグループの元のメンバーが起動すると、デフォルトで代替ノードが元のメンバーノードに置き換えられます。

ただし、NetScaler 10.5 ビルド 50.10 以降では、NetScaler ではこの置換動作を変更できるようになりました。スティッキーオプションを有効にすると、元のメンバーノードが起動した後でも、交換用ノードは保持されます。元のノードは、代替ノードがダウンした場合にのみ引き継ぎます。

バックアップ機能を無効にすることもできます。そのためには、strict オプションを有効にする必要があります。このシナリオでは、ノードグループのメンバーがダウンしても、他のクラスターノードはバックアップノードとしてピックアップされません。元のノードは、起動後もノードグループの一部であり続けます。このオプションにより、ノードグループにバインドされたエンティティは、ノードグループのメンバーでのみアクティブになります。

> 注
>
> strict オプションとスティッキーオプションは、ノードグループの作成時にのみ設定できます。

## スポット構成と部分ストライプ構成のノードグループの設定

August 15, 2023

ノードグループをスポット構成と部分ストライプ構成用に構成するには、最初にノードグループを作成してから、必要なノードをノードグループにバインドする必要があります。次に、必要なエンティティをそのノードグループに関連付けます。ノードグループにバインドされているエンティティは次のとおりです。

- **Spoted** -ノードが 1 つしかないノードグループにバインドされている場合。
- 部分ストライプ -複数のノードを持つノードグループにバインドされている場合。

覚えておくべきいくつかのポイント**:**

- GSLB は、GSLB サイトが単一のクラスターノードを持つノードグループにバインドされている場合にのみ、クラスターでサポートされます。詳細については、「クラスタ内の GSLB の設定」を参照してください。
- NetScaler Gateway は、VPN 仮想サーバーが単一のクラスターノードを持つノードグループにバインドされている場合にのみ、クラスターでサポートされます。スティッキーオプションはノードグループで有効になっている必要があります。
- NetScaler 11 より前のバージョンでは、アプリケーションファイアウォールは個々のクラスターノード（スポット構成）でのみサポートされます。アプリケーションファイアウォールプロファイルは、単一のクラスターノードを持つノードグループにバインドされた仮想サーバーにのみ関連付けることができます。つまり、申請者には以下の操作は許可されていないということです。

 

  - アプリケーションファイアウォールプロファイルを、ストライプまたは部分的にストライピングされた仮想サーバーにバインドします。
  - ポリシーをグローバルバインドポイントまたはユーザー定義のポリシーラベルにバインドします。
  - アプリケーションファイアウォールプロファイルを持つ仮想サーバーをノードグループからバインド解除します。

- NetScaler 11 では、ストライプ構成と部分ストライプ構成のアプリケーションファイアウォールサポートが導入されました。詳細については、「クラスタ構成に対するアプリケーションファイアウォールのサポート」を参照してください。

クラスタでサポートされる NetScaler ADC 機能をチェックして、クラスタで GSLB 、NetScaler Gateway、およびアプリケーションファイアウォールがサポートされている NetScaler バージョンを確認します。

コマンドラインインターフェイスを使用してノードグループを構成するには

1. クラスタ IP アドレスにログオンします。
2. ノードグループを作成します。タイプ:

   ```
   add cluster nodegroup <name> -strict (YES | NO)<!--NeedCopy-->
   ```

   例

   ```
   1 add cluster nodegroup NG0 -strict YES
   ```

3. 必要なノードをノードグループにバインドします。ノードグループのメンバーごとに次のコマンドを入力します。

   ```
   bind cluster nodegroup <name> -node <nodeId><!--NeedCopy-->
   ```

   例

   ID が 1、5、および 6 のノードをバインドします。

   ```
   1 > bind cluster nodegroup NG0 -node 1
   2 > bind cluster nodegroup NG0 -node 5
   3 > bind cluster nodegroup NG0 -node 6
   ```

4. エンティティをノードグループにバインドします。バインドするエンティティごとに、次のコマンドを 1 回入力します。

   ```
   bind cluster nodegroup <name> (-vServer <string> | -identifierName
    <string> | -gslbSite <string> -service <string>)<!--NeedCopy-->
   ```

   > 注
   >
   > GSLB サイトとサービスパラメーターは、NetScaler 10.5 以降で使用できます。

例

仮想サーバ VS1 と VS2 をバインドし、識別子 1 という名前のレート制限識別子をバインドします。

```
> bind cluster nodegroup NG0 -vServer VS1
> bind cluster nodegroup NG0 -vServer VS2
> bind cluster nodegroup NG0 -identifierName identifier1
```

5. ノードグループの詳細を表示して、構成を確認します。タイプ:

```
show cluster nodegroup <name><!--NeedCopy-->
```

例

```
> show cluster nodegroup NG0
```

構成ユーティリティを使用してノードグループを構成するには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [クラスタ] **>** [ノードグループ] に移動します。
3. 詳細ペインで、[追加] をクリックします。
4. 「ノードグループの作成」ダイアログで、ノードグループを設定します。

   a) 「クラスターノード」で、「追加」ボタンをクリックします。

   - Available リストにはノードグループにバインドできるノードが表示され、Configured リストにはノードグループにバインドされているノードが表示されます。
   - Available リストの **+** 記号をクリックしてノードをバインドします。同様に、「Configured」リストの「**-**」記号をクリックしてノードをバインド解除します。

   b) 「仮想サーバー」で、ノードグループにバインドする仮想サーバーのタイプに対応するタブを選択します。［追加］をクリックします。

   - Available リストにはノードグループにバインドできる仮想サーバーが表示され、Configured リストにはノードグループにバインドされている仮想サーバーが表示されます。
   - 「使用可能」リストの「**+**」記号をクリックして、仮想サーバーをバインドします。同様に、「構成済み」リストの「-」記号をクリックして仮想サーバーのバインドを解除します。

ノードグループの冗長性の設定

August 15, 2023

> 注
>
> NetScaler 10.5 ビルド 52.1115.e 以降でサポートされています。

ノードグループは、あるノードグループがダウンしたときに、別のノードグループがトラフィックを引き継いで処理できるように構成できます。たとえば、ノードグループ NG1 がダウンすると、NG2 が引き継ぎます。

> 注
>
> この機能を使用して、各ノードグループがデータセンターとして構成されるデータセンターの冗長性を構成できます。

このユースケースを実現するには、クラスタノードを論理的にノードグループにグループ化し、一部のノードグループを ACTIVE として構成し、他のノードグループを SPARE として構成する必要があります。プライオリティが最も高い（つまり、プライオリティ番号が最も低い）アクティブノードグループが運用上アクティブになり、トラフィックを処理します。この運用上アクティブなノードグループのノードがダウンすると、このノードグループのノード数が他のアクティブなノードグループのノード数と優先順位で比較されます。ノードグループのノード数が多いか等しい場合、そのノードグループは動作上アクティブになります。それ以外の場合は、スペアノードグループがチェックされます。

> 注
>
> - 特定の時点でアクティブにできるのは、1 つの州固有のノードグループだけです。
> - クラスターノードはノードグループの状態を継承します。そのため、「SPARE」状態のノードが「ACTIVE」状態のノードグループに追加されると、そのノードは自動的にアクティブノードとして動作します。
> - クラスターインスタンスに定義されているプリエンプションパラメータによって、最初のアクティブノードグループが再び起動したときに制御を引き継ぐかどうかが決まります。
> - アクティブなノードグループがダウンすると、スペアノードグループがノードグループを占有し、アクティブなトラフィックをホストできます。

次の図は、ノードグループの冗長性が定義されているノードグループ設定を示しています。NG1 は最初はアクティブノードグループです。ノードの 1 つが失われると、優先順位が最も高いスペアノードグループ (NG3) がトラフィックの処理を開始します。

図 1: ノードグループの冗長性が構成された NetScaler クラスター。



ノードグループの冗長性の設定

1. クラスタ IP アドレスにログオンします。
2. アクティブノードグループを作成し、必要なクラスターノードをバインドします。

```
1 > add cluster nodegroup NG1 -state ACTIVE
2 > bind cluster nodegroup NG1 -node n1
3 > bind cluster nodegroup NG1 -node n2
4 > bind cluster nodegroup NG1 -node n3
```

3. スペアノードグループを作成し、必要なノードをバインドします。

```
1 > add cluster nodegroup NG2 -state SPARE -priority 20
2 > bind cluster nodegroup NG2 -node n4
3 > bind cluster nodegroup NG2 -node n5
4 > bind cluster nodegroup NG2 -node n6
```

4. 別のスペアノードグループを作成し、必要なノードをバインドします。

```
1 > add cluster nodegroup NG3 -state SPARE -priority 10
2 > bind cluster nodegroup NG3 -node n7
3 > bind cluster nodegroup NG3 -node n8
4 > bind cluster nodegroup NG3 -node n9
```


クラスターバックプレーンでのステアリングの無効化

August 15, 2023

> 注
>
> NetScaler 11 以降でサポートされています。

NetScaler クラスターのデフォルトの動作は、受信するトラフィック（フローレシーバー）を別のノード（フロープロセッサ）に転送することです。その後、フロープロセッサはトラフィックを処理する必要があります。フローレシーバからフロープロセッサにトラフィックを転送するこのプロセスは、クラスタバックプレーンを介して実行され、ステアリングと呼ばれます。

必要な場合は、ステアリングを無効にして、プロセスがフローレシーバーに対してローカルになり、フローレシーバーがフロープロセッサになります。このような構成設定は、リンクの遅延が大きい場合に便利です。

> 注
>
> この構成は、ストライピングされた仮想サーバーにのみ適用されます。

- 部分的にストライピングされた仮想サーバーの場合、フローレシーバーが非所有者ノードの場合、トラフィックは所有者ノードに転送されます。ただし、フローレシーバーがオーナーノードの場合、ステアリングは無効になります。
- スポット仮想サーバーの場合、フローレシーバーはフロープロセッサであるため、ステアリングは不要です。

ステアリング機構を無効にするときに覚えておくべき点がいくつかあります。

- ステアリングが無効になっているため、ストライプ SNIP はサポートされていません。
- MPTCP と FTP は動作しません。
- L2 モードは無効にする必要があります。
- USIP が有効な場合、ステアリングが無効になっているため、トラフィックが同じノードに戻らない場合があります。
- クラスター IP アドレス宛てのトラフィックは、構成コーディネーターに転送されます。
- ノードがクラスターに参加またはクラスターから脱退すると、1/N を超える接続が影響を受ける可能性があります。これは、使用可能なノードが変更されると、ルートが再ハッシュされる可能性があるためです。その結果、トラフィックは別のノードにルーティングされ、ステアリングが使用できないため、トラフィックは処理されません。

ステアリングは、個々の仮想サーバーレベルまたはグローバルレベルで無効にできます。グローバル構成は、仮想サーバー設定よりも優先されます。

- すべてのストライピングされた仮想サーバーのバックプレーンステアリングの無効化

  クラスターインスタンスレベルで設定されます。ストライピングされた仮想サーバ宛てのトラフィックは、クラスタバックプレーンでは処理されません。


```
add cluster instance \<clId\> -processLocal ENABLED<!--NeedCopy
-->
```

- 特定のストライプ仮想サーバーのバックプレーンステアリングの無効化

  ストライピングされた仮想サーバー上で構成されます。仮想サーバ向けのトラフィックは、クラスタバックプレーンでは制御されません。

  ```
  add lb vserver <name> <serviceType> -processLocal ENABLED<!--
  NeedCopy-->
  ```

クラスター構成の同期

August 15, 2023

構成コーディネーターで使用できる NetScaler ADC 構成は、次の場合にクラスターの他のノードと同期されます。

- ノードがクラスタに加わる場合
- ノードがクラスタに再参加する
- クラスター IP アドレスを介して新しいコマンドが実行されます。

また、構成コーディネーターで使用できる構成 (完全同期) を、特定のクラスタノードに強制的に同期することもできます。一度に 1 つのクラスタノードを同期するようにしてください。同期しないと、クラスタに影響が及ぶ可能性があります。

CLI を使用してクラスター設定を同期するには、次の手順を実行します。

設定を同期するアプライアンスのコマンド・プロンプトで、次のように入力します。

```
1 force cluster sync
```

GUI を使用してクラスタ構成を同期するには、次の手順を実行します。

1. 設定を同期するアプライアンスにログオンします。
2. [システム**] > [**クラスタ**]** に移動します。
3. 詳細ペインの「ユーティリティ」で、「強制クラスター同期」をクリックします。
4. 「**OK**」をクリックします。

クラスタ設定の同期中に失敗したコマンドの一覧を表示する

同期ステータス strict モード`syncStatusStrictMode`が有効になっているクラスタセットアップでは、非 CCO ノードでのクラスタ同期中に失敗したコマンドのリストを表示できます。

`show node`操作を実行すると、CCO 以外のノードのクラスタ同期状態を判断できます。`PARTIAL SUCCESS` 同期状態は、クラスタ同期中に非 CCO ノードで一部のコマンドが失敗したことを示します。

CLI を使用してクラスターの同期中にノードで失敗したコマンドの一覧を表示するには、次の手順を実行します。

- `show cluster syncFailures`

設定例

```
> show cluster node

1) Node ID: 1
        IP:                   10.102.201.24
        Backplane:            1/1/1
        Health:               UP
        Admin State:          ACTIVE
        Operational State:    ACTIVE(Configuration Coordinator)
        Sync State:           ENABLED
        Priority:             31
        Tunnel Mode:          NONE
        Node Group:           DEFAULT_NG
2) Node ID: 2
        IP:                   10.102.201.62*
        Backplane:            2/1/1
        Health:               UP
        Admin State:          ACTIVE
        Operational State:    ACTIVE
        Sync State:           PARTIAL SUCCESS
(Refer the files clus_sync_batch_status.log, sync_route_status.log
   and sync_clusdiff_status.log in /var/nssynclog directory for
   list of commands failed)
        Priority:             31
        Tunnel Mode:          NONE
        Node Group:           DEFAULT_NG
3) Node ID: 3
        IP:                   10.102.201.64
        Backplane:            3/1/1
        Health:               UP
        Admin State:          ACTIVE
        Operational State:    ACTIVE
        Sync State:           PARTIAL SUCCESS
(Refer the files clus_sync_batch_status.log, sync_route_status.log
   and sync_clusdiff_status.log in /var/nssynclog directory for
   list of commands failed)
        Priority:             31
        Tunnel Mode:          NONE
        Node Group:           DEFAULT_NG
Done

> show cluster syncFailures

```



```
39      exec: add system user nsroot "********" -encrypted -externalAuth
           ENABLED -timeout 900 -logging ENABLED -maxsession 20 -
           allowedManagementInterface CLI API -devno 32768
40      ERROR: Resource already exists
41      --
42      exec: set interface 2/LO/1 -autoneg ENABLED -haMonitor OFF -
           haHeartbeat OFF -mtu 1500 -ringtype Elastic -tagall OFF -
           trunkmode OFF -state ENABLED -lagtype NODE -lacpPriority 32768 -
           lacpTimeout LONG -throughput 0 -linkRedundancy OFF -
           bandwidthHigh 0 -bandwidthNormal 0 -intftype Loopback -svmCmd 0
           -ifnum 2/LO/1 -lldpmode NONE -lrsetPriority 1024
43      ERROR: Operation not allowed on loopback interface.
```

クラスターノード間で時間の同期

August 15, 2023

クラスターは Precision Time Protocol (PTP) を使用してクラスターノード間で時間を同期します。PTP はマルチキャストパケットを使用して時刻を同期します。時刻同期に問題がある場合は、PTP を無効にし、クラスターでネットワークタイムプロトコル (NTP) を設定する必要があります。

コマンドラインインターフェイスを使用して **PTP** を有効/無効にするには

クラスタ IP アドレスのコマンドプロンプトで、次のように入力します。

```
1  set ptp -state disable
```

構成ユーティリティを使用して **PTP** を有効/無効にするには

1. クラスタ IP アドレスにログオンします。
2. [システム] > [クラスタ] に移動します。
3. 詳細ペインの [ユーティリティ] で、[**PTP** 設定の構成] をクリックします。
4. [**PTP** の有効化/無効化] ダイアログボックスで、PTP を有効にするか無効にするかを選択します。
5. 「**OK**」をクリックします。

クラスターファイルの同期

August 15, 2023

構成コーディネーターで使用できるファイルは、クラスターファイルと呼ばれます。これらのファイルは、ノードがクラスタに追加されたとき、およびクラスタの存続期間中、定期的に他のクラスタノードで自動的に同期されます。また、クラスタファイルを手動で同期することもできます。

重要：クラスタ環境で証明書またはキーファイルを削除すると、ADC アプライアンスでのさらなる構成が制限されます。構成を変更する場合は、ファイルを同じ場所に戻します。

同期される構成コーディネータのディレクトリとファイルは次のとおりです。

- /nsconfig/ssl/
- /var/netscaler/ssl/
- /var/vpn/bookmark/
- /nsconfig/dns/
- /nsconfig/monitors/
- /nsconfig/nstemplates/
- /nsconfig/ssh/
- /nsconfig/rc.netscaler
- /nsconfig/resolv.conf
- /nsconfig/inetd.conf
- /nsconfig/syslog.conf
- /nsconfig/snmpd.conf
- /nsconfig/ntp.conf
- /nsconfig/httpd.conf
- /nsconfig/sshd_config
- /nsconfig/hosts
- /nsconfig/enckey
- /var/nslw.bin/etc/krb5.conf
- /var/nslw.bin/etc/krb5.keytab
- /var/lib/likewise/db/
- /var/download/
- /var/wi/tomcat/webapps/
- /var/wi/tomcat/conf/Catalina/localhost/
- /var/wi/java_home/lib/security/cacerts
- /var/wi/java_home/jre/lib/security/cacerts
- /nsconfig/license/
- /nsconfig/rc.conf

> ヒント
>
> 手動で (またはシェルを使用して) クラスタ構成コーディネータにコピーされたファイル (証明書およびキーファイル) は、他のクラスタノードでは自動的に使用できません。これらのファイルに依存するコマンドを実行する前に、クラスタ IP アドレスから「sync cluster files」コマンドを実行します。

コマンドラインインターフェイスを使用してクラスタファイルを同期するには

クラスタ IP アドレスのコマンドプロンプトで、次のように入力します。

```
sync cluster files <mode>
```

構成ユーティリティを使用してクラスタファイルを同期するには

1. クラスタ IP アドレスにログオンします。
2. [システム**] > [**クラスタ**]** に移動します。
3. 詳細ペインの [ユーティリティ] で、[クラスターファイルの同期] をクリックします。
4. [クラスタファイルの同期] ダイアログボックスで、[モード] ドロップダウンリストで同期するファイルを選択します。
5. 「**OK**」をクリックします。

クラスターの統計の表示

August 15, 2023

クラスターインスタンスとクラスターノードの統計情報を表示して、パフォーマンスを評価したり、クラスターの動作をトラブルシューティングしたりできます。

コマンドラインインターフェイスを使用してクラスターインスタンスの統計を表示するには

クラスタ IP アドレスのコマンドプロンプトで、次のように入力します。

```
stat cluster instance <clId>
```

コマンドラインインターフェイスを使用してクラスターノードの統計情報を表示するには

クラスタ IP アドレスのコマンドプロンプトで、次のように入力します。

```
stat cluster node <nodeid>
```

> 注
>
> `stat cluster node <nodeid>` このコマンドは、クラスター IP アドレスからコマンドを実行すると、クラスターレベルの統計情報を表示します。ただし、クラスターノードの NSIP アドレスから実行すると、コマンドはノードレベルの統計を表示します。

構成ユーティリティを使用してクラスターインスタンスの統計を表示するには

1. クラスタ IP アドレスにログオンします。
2. [システム] > [クラスタ] に移動します。
3. 詳細ウィンドウで、ページの中央にある [統計] をクリックします。

構成ユーティリティを使用してクラスターノードの統計を表示するには

1. クラスタ IP アドレスにログオンします。
2. [システム] > [クラスタ] > [ノード] に移動します。
3. 詳細ペインでノードを選択し、[**Statistics**] をクリックしてノードの統計情報を表示します。すべてのノードの統計情報を表示するには、特定のノードを選択せずに [**Statistics**] をクリックします。

NetScaler ADC アプライアンスの検出

August 15, 2023

現在のノードと同じサブネットにあるアプライアンスを見つけることができます。その後、検出された必要なアプライアンスを選択的にクラスターに追加できます。この操作は、クラスターを作成するか、既存のクラスターにノードを追加するために実行できます。

> 注
>
> - 検出操作は、構成ユーティリティを介してのみ実行できます。
> - この操作では、異なるネットワークから NetScaler アプライアンスを検出することはできません。
> - この操作を実行して既存のクラスタにノードを追加すると、L3 VLAN 構成はノードから消去されます。アプライアンスをクラスタに追加したら、これらの構成を必ず定義してください。

GUI を使用してアプライアンスを検索するには

1. クラスタ IP アドレスにログオンします。
2. [システム] > [クラスタ] > [ノード] に移動します。
3. 詳細ペインのページ下部で、「**Discover NetScalers**」をクリックします。
4. **Discover NetScalers** ダイアログボックスで、次のパラメーターを設定します。
    - **IP** アドレス範囲 -アプライアンスを検出する IP アドレスの範囲を指定します。たとえば、このオプションを 10.102.29.4-15 と指定すると、10.102.29.4 から 10.102.29.15 までのすべての NSIP アドレスを検索できます。

- バックプレーンインターフェイス -バックプレーンインターフェイスとして使用するインターフェイスを指定します。これはオプションのパラメータです。このパラメーターを指定しない場合は、ノードがクラスターに追加された後に更新する必要があります。

5.「**OK**」をクリックします。
6. クラスターに追加するアプライアンスを選択します。
7.「**OK**」をクリックします。

クラスターノードの無効化

August 15, 2023

ノード上のクラスターインスタンスを無効にすることで、そのノードをクラスターから一時的に削除できます。無効になっているノードは、クラスター構成と同期されません。ノードを再度有効にすると、クラスタ構成は自動的に同期されます。詳細については、「クラスタノード間の同期」を参照してください。

無効化されたノードはトラフィックを処理できず、このノード上の既存の接続はすべて終了します。

> 注
>
> 無効にされた非構成コーディネーターノードの構成が（ノードの NSIP アドレスを使用して）変更された場合、構成はそのノードで自動的に同期されません。クラスタ構成の同期の説明に従って、構成を手動で同期できます。

コマンドラインインターフェイスを使用してクラスタノードを無効にするには

無効にするノードのコマンドプロンプトで、次のように入力します。

```
disable cluster instance <clId>
```

> 注
>
> クラスターを無効にするには、クラスター IP
> アドレスでクラスターインスタンスの無効化コマンドを実行します。

構成ユーティリティを使用してクラスターノードを無効にするには

1. 無効にするノードで、[システム] **>** [クラスタ] に移動し、[クラスタの管理] をクリックします。
2. [クラスターインスタンスの 構成] ダイアログボックスで、[クラスターインスタンスの 有効化] チェックボックスの選択を解除します。

> 注
>
> すべてのノードでクラスターインスタンスを無効にするには、クラスター IP アドレスで前述の手順を実行します。

クラスタノードの削除

October 25, 2023

ノードをクラスタから削除すると、そのノードからクラスタ構成がクリアされます (内部で clear ns config-extended コマンドを実行します)。SNIP アドレス、バックプレーンインターフェイスの **MTU** 設定、およびすべての VLAN 設定（デフォルト VLAN と NSVLAN を除く）もアプライアンスから消去されます。

> 注:
>
> - 削除されたノードがクラスタ構成コーディネーター (CCO) だった場合は、別のノードが自動的に CCO として選択され、クラスタ IP アドレスがそのノードに割り当てられます。現在のクラスタ IP アドレスセッションはすべて無効であるため、新しいセッションを開始する必要があります。
> - クラスター全体を削除するには、各ノードを個別に削除する必要があります。最後のノードを削除すると、クラスタ IP アドレスは削除されます。
> - アクティブなノードを削除すると、クラスターのトラフィック処理能力は 1 ノードだけ低下します。このノード上の既存の接続は終了します。

CLI を使用してクラスターノードを削除するには

1. クラスタ IP アドレスにログオンし、コマンドプロンプトで次のように入力します。

```
rm cluster node <nodeId>

save ns config
```

2. 削除されたノード、NSIP アドレスにログオンし、コマンドプロンプトで次のように入力します。

```
save ns config
```

> 注:
>
> クラスタ IP アドレスがノードから到達できない場合は、そのノード自体の NSIP アドレスで rm cluster instance コマンドを実行します。

クラスタ IP アドレスで、[システム] **>** [クラスタ] **>**[ノード] に移動し、削除するノードを選択して [削除] をクリックします。

クラスターリンクアグリゲーションを使用して展開されたクラスターからノードを削除

August 15, 2023

トラフィック分散メカニズムとしてクラスタリンクアグリゲーションを使用するクラスタからノードを削除するには、トラフィックを受信しないようにノードがパッシブになっていることを確認してから、アップストリームスイッチで対応するインターフェイスをチャネルから削除する必要があります。

クラスタリンク集約の詳細については、「クラスタリンク集約の使用」を参照してください。

クラスタリンクアグリゲーションをトラフィック分散メカニズムとして使用するクラスタからノードを削除するには

1. クラスタ IP アドレスにログオンします。
2. 削除するクラスターノードの状態を PASSIVE に設定します。

```
set cluster node <nodeId> -state PASSIVE
```

3. アップストリームスイッチで、スイッチ固有のコマンドを使用して、対応するインターフェイスをチャネルから削除します。

> 注
>
> クラスタリンクアグリゲーションチャネル上のノードインターフェイスを手動で削除する必要はありません。次のステップでノードを削除すると、自動的に削除されます。

4. クラスタからノードを削除します。

```
rm cluster node <nodeId>
```

クラスター上のジャンボプローブの検出

August 15, 2023

クラスタインターフェイスでジャンボフレームが有効になっている場合、バックプレーンインターフェイスはジャンボフレーム内のすべてのパケットをサポートするのに十分な大きさでなければなりません。これは、バックプレーンの最大伝送ユニット (MTU) を次のように設定することで実現されます。

バックプレーン MTU = 最大 (すべてのクラスタインターフェイス MTU) + 78

上記の構成を確認するには、クラスタセットアップのすべてのピアノードに (前述の計算サイズの) ジャンボプローブを送信する必要があります。プローブが成功しない場合、アプライアンスは「show cluster instance」コマンドの出力に警告メッセージを表示します。

コマンドインターフェイスモードで、次のコマンドを入力します。

```
> show cluster instance
  Cluster ID: 1
  Dead Interval: 3 secs
  Hello Interval: 200 msecs
  Preemption: DISABLED
  Propagation: ENABLED
  Quorum Type: MAJORITY
  INC State: DISABLED
  Process Local: DISABLED
  Cluster Status: ENABLED(admin),    ENABLED(operational), UP
```

> 警告
>
> バックプレーンインターフェイスの MTU は、フレーム内のすべてのパケットを処理するのに十分な大きさでなければなりません。これは<MTU_VAL>と等しくなければなりません。推奨値がユーザが設定できない場合は、ジャンボインターフェイスの MTU 値を確認する必要があります。

Sl 番号	メンバーノード	状況	管理ステート	オペレーションステート
1	ノード ID: 1; ノード IP: 10.102.53.167	上へ	Active	アクティブ (構成コーディネーター)
2	ノード ID: 2; ノード IP: 10.102.53.168	上へ	Active	Active

クラスター内の動的ルートのルート監視

August 15, 2023

ルートモニターを使用すると、動的に学習されたルートが含まれているかどうかに関係なく、クラスターノードを内部ルーティングテーブルに依存させることができます。各ノードのルートモニターは、内部ルーティングテーブルをチェックして、特定のネットワークに到達するためのルートエントリが常に存在することを確認します。ルートエントリが存在しない場合、ルートモニタの状態は DOWN に変わります。

クラスターデプロイメントでは、ノードのクライアント側またはサーバー側のリンクがダウンした場合、トラフィックはピアノードを経由してこのノードに転送されて処理されます。トラフィックのステアリングは、動的ルーティングを設定し、すべてのノード上の各ノードの特別な MAC アドレスを指すスタティック ARP エントリを追加すること

によって実装されます。クラスタ展開に多数のノードがある場合、すべてのノードに特別な MAC アドレスを持つ静的 ARP エントリを追加して管理するのは面倒な作業です。現在、ノードはパケットのステアリングに特別な MAC アドレスを暗黙的に使用しています。そのため、特別な MAC アドレスを指す静的 ARP エントリをクラスタノードに追加する必要がなくなりました。

CLI を使用してクラスターノードをバインドするには

コマンドプロンプトで入力します。

```
bind cluster node <nodeId> \(-routeMonitor <ip\_addr|ipv6\_addr|\*> \[<
    netmask>])
unbind cluster node <nodeId> \(-routeMonitor <ip\_addr|ipv6\_addr|\*>
    \[<netmask>])
```

ノード 1 がルートモニタ 1.1.1.0 255.255.255.0 にバインドされているシナリオを考えてみましょう。ダイナミックルートに障害が発生すると、ノード 1 は非アクティブになります。ヘルスステータスは、次のようにノード ID によって `show cluster node` コマンドで確認できます。

```
Node ID: 1
IP:  10.102.169.96
Backplane:  1/1/2
Health: NOT UP
Reason(s): Route Monitor(s) of the node have failed
Route Monitor -  Network: 1.1.1.0   Netmask: 255.255.255.0   State:
    DOWN
```

SNMP MIB と **SNMP** リンクを使用したクラスターセットアップの監視

August 15, 2023

SNMP MIB は、NetScaler アプライアンスを識別するために SNMP エージェント上で構成されるデバイス固有の情報です。アプライアンス名、管理者、場所などの情報を識別できます。クラスタ設定では、set SNMP MIB コマンドに「ownerNode」パラメータを含めることで、どのノードでも SNMP MIB を設定できるようになりました。このパラメータがない場合、set SNMP MIB コマンドはクラスタコーディネーター (CCO) ノードにのみ適用されます。

CCO 以外のクラスタノードの MIB 設定を表示するには、show SNMP MIB コマンドに「ownerNode」パラメータを含めてください。

クリップでの **SNMP MIB** の設定

コマンドラインインターフェイスを使用して CLIP の MIB 設定を設定および表示する。


```
set snmp mib [-contact <string>] [-name <string>] [-location <string>]
      [-customID <string>] [-ownerNode <positive_integer>]
Done
show snmp mib [-ownerNode <positive_integer>]

 > set mib -contact John -name NS59 -location San Jose -customID 123 -
    ownerNode 3
Done
> sh mib -ownerNode 3
        --------------------
         Cluster Node ID: 3
        --------------------
        NetScaler system MIB:
        sysDescr:    NetScaler NS11.1: Build 46.4.a.nc, Date: Jun  7
           2016, 10:27:29
        sysUpTime:   124300
        sysObjectID: .1.3.6.1.4.1.5951.1.1
        sysContact:  John
        sysName:     NS59
        sysLocation: San Jose
        sysServices: 72
        Custom ID: 123
 Done

> unset mib -contact -name -location -customID -ownerNode 3
 Done
> sh mib -ownerNode 3
        --------------------
         Cluster Node ID: 3
        --------------------
        NetScaler system MIB:
        sysDescr:    NetScaler NS11.1: Build 46.4.a.nc, Date: Jun 7
           2016, 10:27:29
        sysUpTime:   146023
        sysObjectID: .1.3.6.1.4.1.5951.1.1
        sysContact:  WebMaster (default)
        sysName:     NetScaler
        sysLocation: POP (default)
        sysServices: 72
        Custom ID: Default
 Done
```

クラスター **SNMP** トラップメッセージ

クラスタ設定では、SNMP トラップアラームの設定は CLIP から行う必要があります。コマンドは各ノードに伝播されます。

SNMP の構成の詳細については、「SNMP トラップを生成するように NetScaler ADC を構成する」を参照してください。

使用可能なクラスター固有のトラップは次のとおりです。

```
>sh snmp alarm | grep cluster
    CLUSTER-BACKPLANE-HB-MISSING N/A N/A 86400   ENABLED - ENABLED
    CLUSTER-CCO-CHANGE N/A N/A N/A ENABLED - ENABLED
    CLUSTER-NODE-HEALTH N/A N/A 86400 ENABLED - ENABLED
    CLUSTER-NODE-QUORUM N/A N/A 86400 ENABLED - ENABLED
    CLUSTER-OVS-CHANGE N/A N/A N/A ENABLED - ENABLED
    CLUSTER-PROP-FAILURE N/A N/A N/A ENABLED - ENABLED
    CLUSTER-SYNC-FAILURE N/A N/A N/A ENABLED - ENABLED
    CLUSTER-SYNC-PARTIAL-SUCCESS N/A N/A N/A ENABLED - ENABLED
    CLUSTER-VERSION-MISMATCH N/A N/A 86400 ENABLED - ENABLED
```

クラスター展開でのコマンド伝達障害の監視

August 15, 2023

クラスター展開では、新しいコマンド「show prop status」を使用して、問題の監視とトラブルシューティングを迅速に行うことができます。CCO 以外のノードでのコマンド伝播の失敗に関連する問題。このコマンドは、CCO 以外のすべてのノードで発生した最新のコマンド伝播障害を最大 20 件表示します。この操作は、NetScaler アプライアンスの CLI または GUI のいずれかを使用して実行できます。CLIP アドレスまたはクラスタ展開内の任意のノードの NSIP アドレスを介してアクセスできます。

ノードの正常なシャットダウン

April 15, 2024

クラスター設定では、クラスターレベルまたは特定の仮想サーバーレベルの既存の接続（1/N 番目の接続、N はクラスターサイズ）の一部が失われます。この動作は、ノードがシステムから離れたりシステムに参加したりした場合に発生します。損失に対処するには、既存の接続を正常に処理する必要があります。CLIP アドレスに「クラスタで接続を保持」オプションを設定し、ノードの NSIP でタイムアウト間隔を指定することで、グレースフルハンドリングが行われます。

接続のグレースフルハンドリングは、次の 2 つのシナリオに適用できます。

1. クラスターアップグレード
2. 新しいノード追加

クラスタアップグレードにおけるノードの適切な処理

クラスターをアップグレードするには、一度に 1 つのノードをアップグレードする必要があります。ノードをアップグレードする前に、ノードをパッシブ状態に設定し、アップグレード後にアクティブ状態に設定する必要があります。ノードのアップグレード時に既存の接続が終了しないように、設定されたタイムアウト間隔で正常にシャットダウンしてください。そうしないと、クラスターの接続の 1/N (N はクラスターサイズ) が終了します。

> 注
>
> - 既存のセッションが設定されたタイムアウト間隔内に完了しない場合、そのセッションは猶予時間の後に終了します。
> - タイムアウト間隔を確認するには、パッシブに設定されているノードの CLIP アドレスまたは NSIP アドレスを使用する必要があります。

クラスターのアップグレードシナリオでノードを正常に処理する手順は次のとおりです:

1. 5 つのノード (n0、n1、n2、n3、n4) のクラスタ構成を考えてみましょう。
2. ノードをシャットダウンする前に、「クラスタ接続を維持」オプションを設定する必要があります。特定の時間間隔で、このノードの既存の接続をすべてクラスターレベルまたは仮想サーバーレベルで維持するのに役立ちます。

   例

   CLIP の場合

   “‘set cluster instance –retainConnectionsOnCluster YES

   ```
   ま た は

   ```set lb vserver <vserver name>  - retainConnectionsOnCluster Yes
      <!--NeedCopy-->
   ```

3. 次に、ノード n3 の NSIP アドレスにログオンし、ノード n3 を内部タイムアウトで PASSIVE に設定します。

   例

   “‘set cluster node n3 –state PASSIVE –delay 60

   ```
   ```saveconfig<!--NeedCopy-->
   ```

4. 猶予期間が終了したら、すべての接続を閉じて n3 をシャットダウンし、NetScaler アプライアンスを再起動します。
5. アプライアンスをアップグレードします。次に、CLI をアプライアンスの NSIP アドレスに接続した状態で、ノードを ACTIVE に設定します。

   例

   “‘set cluster node n3 –state ACTIVE


```
```saveconfig<!--NeedCopy-->
```

6. クラスター内のすべてのノードで手順 4 ～6 を繰り返します。
7. すべてのノードをアップグレードして ACTIVE に設定したら、CLIP アドレスから retainConnectionsOnCluster オプションをリセットします。

   例

   “‘set cluster instance -retainConnectionsOnCluster NO

```
ま た は

```set lb vserver <vserver name>  - retainConnectionsOnCluster NO
   <!--NeedCopy-->
```

> 注
>
> クラスタのアップグレード時にバージョンが一致しない場合、クラスタの伝播は自動的に無効になり、CLIP ではコマンドは許可されません。

新規ノード追加時のノードの正常な処理

ノードの適切な処理とは、既存の NetScaler クラスターに新しいノードを追加する方法を示しています。すでにトラフィックを処理している NetScaler クラスターがあるとします。また、既存の接続を終了せずに、別のアプライアンスをノードとしてクラスターに追加する必要があります。前述のシナリオを実現するには、既存の接続をグローバルレベルまたは特定の仮想サーバーレベルで保持するオプションを設定します。完了したら、設定を保存します。次に、接続を保持するオプションを「いいえ」に設定して、他のノードからの既存の接続を新しいノードに再割り当てできるようにします。

ノードが新しく追加された場合にノードを正常に処理する手順は次のとおりです。

1. 「クラスタ接続を維持」オプションが有効になっている既存の構成を保存します。これにより、このノードの既存の接続をすべてクラスターレベルまたは仮想サーバーレベルで特定の時間間隔で保持できます。

   CLIP の場合

```
set cluster instance x  -  retainConnectionsOnCluster YES
```

   または

```
set lb vserver xxxx  - retainConnectionsOnCluster Yes
```

2. クラスタセットアップにノード ‘n5’ を追加します。
3. 既存の接続を他のノードから新しく追加されたノード n5 に分散するには、「RetainConnectionOnCluster」オプションを「NO」に無効にします。

CLIP の場合

```
set cluster instance x - retainConnectionsOnCluster NO
```

または

```
set lb vserver xxxx - retainConnectionsOnCluster NO
```

> 注
>
> バックプレーンステアリングは、クラスタ設定のトラフィック分散メカニズムのタイプ（ECMP、CLAG、および USIP）によって異なります。バックプレーンステアリングの増加は、トラフィックタイプによって異なります。

クラスター内のノードのグレースフルシャットダウンの設定

クラスター内のノードのグレースフルシャットダウンを設定するには、以下を実行してください。

1. グローバル (クラスタ) レベルで「クラスタ上の接続を維持」オプションを設定します。
2. 仮想サーバーレベルで「クラスターで接続を維持」オプションを設定します。
3. ノードの NSIP アドレスで指定されたグレースフルタイムアウト間隔を使用して、ノード (システムを離れる) をパッシブ状態に設定します。
4. 既存の接続を監視して、すべてのトランザクションが猶予期間内に完了したことを確認します。

CLI を使用して既存の接続をグローバル **(**クラスタ**)** レベルで維持するには

既存の接続は、グローバルレベルまたは特定の仮想サーバーレベルで保持できます。このオプションは、既存の接続をすべてグローバルレベルで保持するように設定されています。デフォルトでは、このオプションは無効になっています。

コマンドプロンプトで次のように入力します:

```
- set cluster instance <clusterID> - retainConnectionsOnCluster YES

- set cluster instance 60 - retainConnectionsOnCluster YES
```

CLI を使用して、クラスター内の特定の仮想サーバーの既存の接続を保持するには

このオプションは、負荷分散仮想サーバー固有の既存の接続を保持するように構成されています。これらの接続を維持するために、仮想サーバーレベルでこのオプションを有効にします。デフォルトでは、このオプションは無効になっています。

コマンドプロンプトで入力します:


```
-  set lb vserver <clusterID>  -retainConnectionsOnCluster Yes

-  set lb vserver v1  -retainConnectionsOnCluster Yes
```

CLI を使用してクラスターノードをパッシブ状態に設定するには

クラスターノードを適切なタイムアウト間隔でパッシブ状態に設定すること。クラスターのアップグレード中は伝播が無効になるため、この設定はノードの NSIP で実行されます。

コマンドプロンプトで入力します:

```
-  set cluster node <clusterID> -state passive
-backplane <interface_name>@
-priority <positive_integer>
-delay <mins>

-  set cluster node 4  -state PASSIVE -delay 60

-  set cluster instance 60  -  retainConnectionsOnCluster YES
-  set lb vserver v1  -retainConnectionsOnCluster Yes
-  set cluster node 4  -state PASSIVE -delay 60
```

> 注
>
> CLIP から遅延オプションを設定してクラスターノードをパッシブに設定すると、クラスターノードで次のような動作が見られる場合があります。
>
> - タイムアウト後、ノードはノードの NSIP からパッシブとして表示されます。
> - CLIP の **show cluster instance** コマンドを実行すると、そのノードが CLIP からアクティブとして表示されます。一方、**CLIP** の **show cluster node** コマンドでは、ノードはパッシブとして表示されます。

GUI を使用してノードのグレースフルシャットダウンを設定するには

1. [構成] **>** [システム] **>** [クラスタ] に移動し、[クラスタの管理] をクリックします。
2. 「クラスタの管理」ページで、「クラスタ接続を維持」オプションを選択します。
3. 「**OK**」をクリックし、「完了」をクリックします。

サービスの正常なシャットダウン

August 15, 2023

NetScaler 12.1 ビルド 49.xx 以降、NetScaler クラスターはサービスのグレースフルシャットダウンをサポートしています。サービスを正常にシャットダウンするには、次のいずれかのタスクを実行できます。

- サービスを明示的に無効にし、
  - 遅延を秒単位で設定します。
  - グレースフルシャットダウンを有効にします。
- TROFS コードまたは文字列をモニタに追加します。

詳細については、「サービスのグレースフルシャットダウン」を参照してください。

CLI を使用してサービスのグレースフルシャットダウンを設定するには

グレースフルオプションのみで無効にする:

コマンドプロンプトで入力します。

```
disable service <name> [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```

例

```
disable service svc1 -graceFul YES
Done
sh service svc1
                svc1 (10.102.225.11:80) - HTTP
                State: GOING OUT OF SERVICE   Graceful (number of
                   active clients: 1)
                Last state change was at Wed Jul 25 10:46:29 2018
                Time since last state change: 0 days, 00:00:02.680
                ... ... ... .
                ... ... ... .
                Traffic Domain: 0

1)            Monitor Name: tcp-default
                                State: UP              Weight: 1
                                                 Passive: 0
                                Probes: 26           Failed [Total: 0
                                   Current: 0]
                                Last response: Success - TCP syn+ack
                                   received.
                                Response Time: 0.0 millisec
<!--NeedCopy-->
```

タイムアウトとグレースフルオプションで無効化:

コマンドプロンプトで入力します。

```
disable service <name> [<delay>] [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```


例

```
disable service svc1 2000 -graceFul YES

Done
> sh service svc1
                svc1 (10.102.225.11:80) - HTTP
                State: GOING OUT OF SERVICE (Graceful (number of active
                    clients: 1), Out Of Service in 1998 seconds)
                Last state change was at Wed Jul 25 10:49:08 2018
                Time since last state change: 0 days, 00:00:01.710
                ... ... ... .
                ... ... ... .
                Traffic Domain: 0

1)            Monitor Name: tcp-default
                                State: UP              Weight: 1
                                                Passive: 0
                                Probes: 57           Failed [Total: 0
                                    Current: 0]
                                Last response: Success - TCP syn+ack
                                    received.
                                Response Time: 0.0 millisec
Done
<!--NeedCopy-->
```

タイムアウトとグレースフルオプションを使用してサービスグループを無効にします。

コマンドプロンプトで入力します。

```
disable serviceGroup <serviceGroupName>@ [<serverName>@ <port>] [-delay
<secs>] [-graceFul ( YES | NO )]
Show service group <serviceGroupName>
<!--NeedCopy-->
```

例:

```
disable servicegroup sg -delay 2000 -graceFul yes
sh servicegroup sg
                sg - HTTP
                State: DISABLED                Effective State: OUT OF
                    SERVICE Monitor Threshold : 0
                Max Conn: 0       Max Req: 0          Max Bandwidth: 0
                    kbits
                Use Source IP: NO
                Client Keepalive(CKA): NO
                ... ... ... .
                ... ... ...


                1)   200.200.10.21:80       Server Name: server3
                    Server ID: None Weight: 1
                                State:    GOING OUT OF SERVICE (learnt
                                    from node:2 )     Graceful (number
```



```
                                        of active clients: 6), Out Of
                                        Service in 1993 seconds
14                                   Last state change was at Mon Aug 13
                                        15:15:11 2018
15                                   … … …
16
17                  2)   200.200.10.22:80       Server Name: server4
                       Server ID: None Weight: 1
18                                   State:    GOING OUT OF SERVICE (learnt
                                        from node:2 )     Graceful (number
                                        of active clients: 7), Out Of
                                        Service in 1993 seconds
19                                   Last state change was at Mon Aug 13
                                        15:15:11 2018
20 <!--NeedCopy-->
```

> 注:
> CLIP には、すべてのクラスタノードからのすべてのアクティブなクライアント接続の集計値が表示されます。

GUI を使用してサービスのグレースフルシャットダウンを設定するには

1. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。
2. サービスを開き、アクションリストから「無効」をクリックします。待機時間を入力し、[Graceful] を選択します。

CLI を使用してモニターの TROFS コードまたは文字列を設定するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
1 add lb monitor <monitor-name> HTTP -trofsCode <respcode>
2 add lb monitor <monitor-name> HTTP-ECV  -trofsString <resp string>
3 add lb monitor <monitor-name> TCP-ECV  -trofsString <resp string>
4 <!--NeedCopy-->
```

GUI を使用してモニタ内の TROFS コードまたは文字列を設定するには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. モニターペインで「追加」をクリックし、以下のいずれかの手順を実行します。
   - 「タイプ」に「HTTP」を選択し、TROFS コードを指定します。
   - [タイプ] に [HTTP-ECV] または [TCP-ECV] を選択し、TROFS 文字列を指定します。

クラスターの **IPv6** 対応ロゴサポート

August 15, 2023

クラスタ化されたアプライアンスの IPv6 Ready Logo 認証をテストできます。ND テストケース、ルーター要請処理、ルートアドバタイズメントやルーターリダイレクションメッセージの送信など、IPv6 コアプロトコルをテストするための変更されたコマンドは、クラスター化されたセットアップで使用できます。IPv6 コアプロトコルのテストに使用できる IPv6 機能は次のとおりです。

IPv6ReadyLogo フェーズ 2 テストスイートにおける ND テストケース、ルーター要求処理、ルートアドバタイズメントの送信、ルーターリダイレクトメッセージングなど、IPv6 コアプロトコルを通過させるために使用できる変更された機能を以下に示します。

- ローカル SNIP をリンク
- アドレス解決とネイバー到達不能
- ルーターとプレフィックスディスカバリー
- ルーターリダイレクト
- DoDAD

これらの変更されたコマンドにより、クラスタ化されたアプライアンスでは次の構成がサポートされます。

IPv6 コアプロトコルをテストするためのサポート可能な構成

IPv6 Ready Logo テストケースに合格するクラスタ化されたセットアップでは、クラスタ管理 IP アドレス (CLIP) で次の構成を実行できます。

- グローバル IP6 設定
- basic IPv6 configuration
- その他の IPv6 構成

グローバル設定

グローバル IPv6 構成では、基本的な IPv6 構成を実行するためのグローバル IPv6 パラメータ (再学習、ルーターリダイレクト、NDBaseRachtime、NReTransmissionTime、td、`natprefix`doodad など) を設定できます。

コマンドプロンプトで、次のように入力します。

```
set ipv6 \[-ralearning \( ENABLED | DISABLED )] \[-routerRedirection \(
    ENABLED | DISABLED )] \[-ndBasereachTime<positive\_integer>]\[-
    ndRetransmissionTime <positive\_integer>] \[-natprefix <ipv6\_addr
    |\*>\[-td<positive\_integer>]] \[-doDAD \( ENABLED | DISABLED )]
```


基本的な **IPv6** 構成

基本的な IPv6 設定により、IPv6 アドレスを作成して VLAN インターフェイスにバインドできます。次の構成を実行して、IPv6 コアプロトコルをテストできます。

CLI を使用して VLAN をクラスタ構成に追加するには

コマンドプロンプトで入力します。

```
1 add vlan <id>
```

CLI を使用してクラスター化されたセットアップに別の VLAN を追加するには

コマンドプロンプトで入力します。

```
1 add vlan <id>
```

CLI を使用してインターフェイスを VLAN にバインドするには

コマンドプロンプトで入力します。

```
1 bind vlan <id> -ifnum <interface_name>
```

CLI を使用してインターフェイスを VLAN にバインドするには

このコマンドは、後続のルータアドバタイズメントのために、グローバルプレフィックスを RA 情報にオンリンクプレフィックスとして追加します。コマンドプロンプトで入力します。

```
1 bind vlan <id> -ifnum <interface_name>
```

CLI を使用して VLAN に IPv6SNIP アドレスを追加するには

コマンドプロンプトで、次のように入力します。

```
1 add ns ip6 <IPv6Address>@ \[-scope \( global | link-local )]\[-type <
      type>
```

CLI を使用して VLAN に IPv6 アドレスを追加するには

コマンドプロンプトで、次のように入力します。

```
1 add ns ip6 <IPv6Address>@ \[-scope \( global | link-local )]\[-type <
      type>
```

CLI を使用して IPv6 アドレスを VLAN にバインドするには

コマンドプロンプトで、次のように入力します。

```
1 bind vlan <id> \[-ifnum <interface\_name> \[-tagged]]\[-IPAddress <ip\
      _addr|ipv6\_addr|
```

CLI を使用して IPv6 アドレスを VLAN にバインドするには

コマンドプロンプトで、次のように入力します。


```
bind vlan <id> \[-ifnum <interface\_name> \[-tagged]]\[-IPAddress <ip\
    _addr|ipv6\_addr|
```

CLI を使用して VLAN に接続されたリンクローカル IPv6 アドレスを表示するには

コマンドプロンプトで、次のように入力します。

```
sh VLAN
```

例 **1**

```
add vlan 2
add vlan 3
bind vlan 2 -ifnum 1/2
bind vlan 3 -ifnum 1/3
add ip6 fe80::9404:60ff:fedd:a464/64 -vlan 2 -scope link-local -type
    SNIP
add ip6 fe80::c0ee:7bff:fede:263f/64 -vlan 3 -scope link-local -type
    SNIP
add ip6 3ffe:501:ffff:100:9404:60ff:fedd:a464/64 -vlan 2
add ip6 3ffe:501:ffff:101:c0ee:7bff:fede:263f/64 -vlan 3
bind vlan 2 -ipAddress 3ffe:501:ffff:100:9404:60ff:fedd:a464/64
bind vlan 3 -ipAddress 3ffe:501:ffff:101:c0ee:7bff:fede:263f/64
```

例 **2**

```
sh vlan
1)      VLAN ID: 2      VLAN Alias Name:
        Interfaces : 1/6
        IPs :
             3ffe:501:ffff:100:2e0:edff:fe15:ea2a/64
3)      VLAN ID: 3      VLAN Alias Name:
        Link-local IPv6 addr: fe80::9404:60ff:fedd:a464/64
        Interfaces : 1/5
        IPs :
             3ffe:501:ffff:101:2e0:edff:fe15:ea2b/64
 Done
```

その他の **IPv6** クラスター構成

IPv6 コアプロトコルをテストするには、次の新しい IPv6 構成または変更された IPv6 構成を使用できます。

CLI を使用して VLAN 固有のルーターアドバタイズメントパラメータを設定するには

コマンドプロンプトで入力します。


```
set nd6RAvariables -vlan <positive\_integer> \[-ceaseRouterAdv \( YES |
     NO)] \[-sendRouterAdv \( YES | NO )] \[-srcLinkLayerAddrOption \(
    YES | NO )]\[-onlyUnicastRtAdvResponse \( YES | NO )] \[-
    managedAddrConfig \( YES | NO)] \[-otherAddrConfig \( YES | NO )]
    \[-currHopLimit <positive\_integer>]\[-maxRtAdvInterval <positive\
    _integer>] \[-minRtAdvInterval<positive\_integer>] \[-linkMTU <
    positive\_integer>] \[-reachableTime<positive\_integer>] \[-
    retransTime <positive\_integer>] \[-defaultLifeTime<integer>]
```

CLI を使用して、リンク上のグローバルプレフィックスの設定可能なパラメータを設定するには

コマンドプロンプトで入力します。

```
set onLinkIPv6Prefix <ipv6Prefix> \[-onlinkPrefix \( YES | NO )]\[-
    autonomusPrefix \( YES | NO )] \[-depricatePrefix \( YES | NO )]\[-
    decrementPrefixLifeTimes \( YES | NO )] \[-prefixValideLifeTime   <
    positive\_integer>] \[-prefixPreferredLifeTime <positive\_integer>]
```

CLI を使用して、構成可能なパラメーターをオンリンクグローバルプレフィックスに追加するには

コマンドプロンプトで入力します。

```
add onLinkIPv6Prefix <ipv6Prefix> \[-onlinkPrefix \( YES | NO )]\[-
    autonomusPrefix \( YES | NO )] \[-depricatePrefix \( YES | NO )]\[-
    decrementPrefixLifeTimes \( YES | NO )]-prefixValideLifeTime <
    positive\_integer>]\[-prefixPreferredLifeTime <positive\_integer>]
```

CLI を使用して、IPv6 プレフィックスの構成可能なパラメーターへのオンリンクリンクを設定するには

コマンドプロンプトで、次のように入力します。

```
help set onLinkIPv6Prefix
```

CLI を使用して、リンク上のリンクを IPv6 プレフィックスの構成可能なパラメーターにバインドするには

コマンドプロンプトで入力します。

```
help bind nd6RAvariables
```

CLI を使用して nd6RA 変数を表示するには

コマンドプロンプトで入力します。

```
help sh nd6RAvariables
```

例

```
> sh nd6RAvariables
  1) Vlan : 1
     SendAdvert     : NO   CeaseAdv       : NO        SourceLLAddress:
        YES
```



```
4       UnicastOnly     : NO    ManagedFlag     : NO        OtherConfigFlag:
        NO
5       CurHopLimit     : 64    MaxRtrAdvInterv: 600        MinRtrAdvInterv:
        198
6       LinkMTU         : 0     ReachableTime   : 0          RetransTimer   :
        0
7       DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay    :
        0
8
9     2) Vlan : 2
10      SendAdvert      : NO    CeaseAdv        : NO         SourceLLAddress:
        YES
11      UnicastOnly     : NO    ManagedFlag     : NO         OtherConfigFlag:
        NO
12      CurHopLimit     : 64    MaxRtrAdvInterv: 600        MinRtrAdvInterv:
        198
13      LinkMTU         : 0     ReachableTime   : 0          RetransTimer   :
        0
14      DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay    :
        0
15  Done
16  >
17  > sh nd6Ravariables  -vlan 2
18    1) Vlan : 2
19      SendAdvert      : NO    CeaseAdv        : NO         SourceLLAddress:
        YES
20      UnicastOnly     : NO    ManagedFlag     : NO         OtherConfigFlag:
        NO
21      CurHopLimit     : 64    MaxRtrAdvInterv: 600        MinRtrAdvInterv:
        198
22      LinkMTU         : 0     ReachableTime   : 0          RetransTimer   :
        0
23      DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay    :
        0
24      Prefix :
25  3ffe:501:ffff:100::/64
26  Done
```

クラスターのハートビートメッセージの管理

August 15, 2023

クラスターでのハートビートメッセージの管理は、高可用性 (HA) 構成での管理と似ています。ノードは、有効になっているすべてのインターフェイス上で相互にハートビートメッセージを送受信できます。ハートビートメッセージによるトラフィックの増加を防ぐため、ノードインターフェースのハートビートオプションを無効にできるようになりました。ただし、バックプレーンインターフェイスのハートビートオプションは、クラスタノード間の接続を維持するために必要であるため、無効にすることはできません。

ハートメッセージの管理の詳細については、「NetScaler アプライアンスでの高可用性ハートビートメッセージの管理」を参照してください。

NetScaler CLI を使用してノードインターフェイスでハートビートメッセージを管理するには

コマンドプロンプトで入力します。

```
set interface <ID> \[-HAHeartBeat \(ON | OFF)]
Show interface <ID>
```

所有者ノードの応答ステータスの構成

August 15, 2023

OwnerDownResponse オプションは、スポット付き SNIP アドレスを持つノードに設定できます。デフォルトでは、このオプションは有効になっています。これにより、ノードが非アクティブのときに、スポットされた IP アドレスが（アップストリームルータからの）PING または ARP 要求に応答できるようになります。このオプションを無効にすると、所有者ノードが非アクティブのとき、IP アドレスはルーター要求に応答できません。

ECMP 展開でスタティックルートのモニタリングにこの機能がどのように使用されるかについては、「等コストマルチパス (ECMP) の使用 」トピックを参照してください。

NetScaler CLI を使用して所有者ノードの応答ステータスを設定するには

コマンドプロンプトで入力します。

```
add ns ip <IPAddress> \[-ownerNode <positive\_integer>] \[-
    ownerDownResponse \(YES | NO )] \[-td <positive\_integer>]
```

例

```
add ns ip 2.2.2.2 255.255.255.0 -ownernode 6  - ownerdownResponse YES
```

NetScaler GUI を使用して所有者ノードの応答ステータスを設定するには

1. [システム] **>** [ネットワーク] **>** [**IP**] に移動し、[追加] をクリックしてスポット付き SNIP アドレスを作成します。
2. 「**IP** アドレスの作成」ページで、「**ownerDownResponse** 」チェックボックスをオンまたはオフにします。

NetScaler GUI を使用して所有者ノードの応答ステータスを編集するには

[システム] **>** [ネットワーク] **>** **[IP]** に移動し、**IP** アドレスを選択し、[編集] をクリックして **OwnerDownResponse** チェックボックスをオンまたはオフにします。

スポットクラスタ構成内の非アクティブノードのスタティックルート **(MSR)** サポートを監視する

August 15, 2023

ルートで MSR オプションを有効にしてセットアップされたクラスタでは、アクティブノードのみがスタティックルートをプローブできます。非アクティブな間はネットワークに到達でき、スペアノードにはルートへのリンクがなく、ルートをプローブできません。PING と ARP プローブを IPv4 ルートに送信し、ping6 と nd6 プローブを IPv6 ルートに送信するように非アクティブノードまたはスペアノードを構成できるようになりました。これを実行できるのは、SNIP アドレスがアクティブで、1 つのノードのみが独占的に所有しているスポットクラスタ構成だけです。

シングルノードアクティブクラスターでの **VRRP** インターフェイスのバインド

August 15, 2023

高可用性 (HA) セットアップをクラスターセットアップに移行する場合、すべての構成に互換性があり、クラスター内でサポートできる必要があります。これを実現するために、ノードインターフェイスに仮想ルータ ID（VRID と VRID6）を設定できるようになりました。

> 重要
>
> 現在、VRID と VRID6 をサポートしているのは単一ノードのアクティブクラスターシステムのみです。

VRID および VRID6 の設定手順については、仮想 MAC アドレスの設定を参照してください。

単一ノードのアクティブクラスタで仮想ルータ ID を設定するには、VRID または VRID6 を追加して、クラスタノードインターフェイスにバインドします。

NetScaler CLI を使用して VRID を追加するには

コマンドプロンプトで入力します。

```
add vrID <ID>
```

NetScaler CLI を使用して VRID をクラスターノードインターフェイスにバインドするには

コマンドプロンプトで入力します。


```
Bind vrid <ID> -ifnum <interface\_name> | -trackifNum <interface\_name

Add vrID 100
Bind vrid 100  - ifnum 1/1  1/2
done
```

NetScaler CLI を使用して VRID6 を追加するには

コマンドプロンプトで入力します。

```
add vrID6 <ID>
```

CLI を使用して VRID6 をクラスターノードインターフェイスにバインドするには

コマンドプロンプトで入力します。

```
bind vrid6 <ID> -ifnum <interface\_name> | -trackifNum <interface\_name
    >

Add vrID6 100
Bind vrid6 100  - ifnum 1/1  1/2
Done
```

クラスターセットアップと使用シナリオ

August 15, 2023

このセクションでは、NetScaler クラスターをさまざまな機能やネットワークトポロジに合わせてセットアップおよび構成できるいくつかのシナリオについて説明します。他のシナリオを文書化したい場合は、フィードバックを提供してください。

2 ノードクラスターの作成

August 15, 2023

2 ノードクラスタは、最小 (n/2 +1) ノード (n はクラスタノード数) がトラフィックを処理できる場合にのみクラスタが機能するという規則の例外です。同じ式を 2 ノードのクラスターに適用した場合、1 つのノードがダウンすると (n/2 +1=2)、クラスターは機能しなくなります。

2 ノードクラスタは、1 つのノードしかトラフィックを処理できない場合でも機能します。

2 ノードクラスターの作成は、他のクラスターの作成と同じです。1 つのノードを構成コーディネーターとして追加し、もう 1 つのノードを他のクラスターノードとして追加します。

> 注
>
> 2 ノードクラスタでは、設定の差分同期はサポートされていません。完全同期のみがサポートされています。

HA セットアップのクラスターセットアップへの移行

August 15, 2023

既存の高可用性（HA）セットアップをクラスターセットアップに移行するには、最初に NetScaler アプライアンスを HA セットアップから削除し、HA 構成ファイルのバックアップを作成する必要があります。その後、2 つのアプライアンスを使用してクラスターを作成し、バックアップした構成ファイルをクラスターにアップロードできます。

> 注
>
> - バックアップした HA 構成ファイルをクラスタにアップロードする前に、クラスタ互換になるように変更する必要があります。手順の関連ステップを参照してください。
> - **batch-f** <backup_filename> コマンドを使用して、バックアップした設定ファイルをアップロードします。

前述のアプローチは基本的な移行ソリューションであり、デプロイされたアプリケーションのダウンタイムが発生します。そのため、アプリケーションの可用性を考慮しないデプロイメントでのみ使用する必要があります。

ただし、ほとんどの導入環境では、アプリケーションの可用性が最も重要です。このような場合は、ダウンタイムを発生させることなく HA セットアップをクラスタセットアップに移行できるアプローチを使用する必要があります。この方法では、最初にセカンダリアプライアンスを削除し、そのアプライアンスを使用して単一ノードクラスタを作成することにより、既存の HA セットアップをクラスタ設定に移行します。クラスタが稼働してトラフィックを処理すると、HA セットアップのプライマリアプライアンスがクラスタに追加されます。

コマンドラインインターフェイスを使用して HA セットアップを (ダウンタイムなしで) クラスターセットアップに変換するには

プライマリアプライアンス（NS1）-10.102.97.131、セカンダリアプライアンス（NS2）-10.102.97.132 の HA セットアップの例を考えてみましょう。

1. HA ペアの構成が安定していることを確認してください。
2. HA アプライアンスのいずれかにログオンし、シェルに移動して、ns.conf ファイル（ns_backup.conf など）のコピーを作成します。
3. セカンダリアプライアンス NS2 にログオンし、設定をクリアします。この操作により、HA セットアップから NS2 が削除され、スタンドアロンアプライアンスになります。

```
1 > clear ns config full
```


> 注
>
> - このステップは、NS2 がスタンドアロンのアプライアンスになった今、VIP アドレスの所有を開始しないようにするために必要です。
> - この段階では、プライマリアプライアンスである NS1 はまだアクティブで、トラフィックの処理を続けています。

4. NS2（セカンダリアプライアンスではなくなった）にクラスタを作成し、PASSIVE ノードとして構成します。

```
> add cluster instance 1

> add cluster node 0 10.102.97.132 -state PASSIVE -backplane
   0/1/1

> add ns ip 10.102.97.133 255.255.255.255 -type CLIP

> enable cluster instance 1

> save ns config

> reboot -warm
```

5. バックアップされた構成ファイルを次のように変更します。

   - クラスターでサポートされていない機能を削除します。サポートされていない機能の一覧については、「クラスターでサポートされている NetScaler ADC 機能」を参照してください。これはオプションのステップです。このステップを実行しないと、サポートされていないコマンドの実行は失敗します。
   - インターフェイスを持つ設定を削除するか、インターフェイス名を c/u 規約から n/c/u 規約に更新します。

     例

     ```
     > add vlan 10 -ifnum 0/1
     ```

     に変更する必要があります

     ```
     > add vlan 10 -ifnum 0/0/1 1/0/1
     ```

   - バックアップ構成ファイルには、SNIP アドレスを持つことができます。これらのアドレスは、すべてのクラスターノードでストライピングされます。ノードごとにスポッティング IP アドレスを追加することをお勧めします。

     例

     ```
     > add ns ip 1.1.1.1 255.255.255.0 -ownerNode 0

     > add ns ip 1.1.1.2 255.255.255.0 -ownerNode 1
     ```

   - ホスト名を更新して、所有者ノードを指定します。

例

```
> set ns hostname ns0 -ownerNode 0

> set ns hostname ns1 -ownerNode 1
```

- 検出された IP に依存する他のすべての関連するネットワーク構成を変更します。たとえば、L3 VLAN、SNIP を NATIP として使用する RNAT 構成、SNIP/MIP を指す INAT ルール)。

6. クラスターで、次の操作を行います。

   - クラスタバックプレーン、クラスタリンクアグリゲーションチャネルなどを接続して、クラスタのトポロジを変更します。
   - バックアップおよび変更された構成ファイルから構成コーディネータに、クラスタ IP アドレスを通じて構成を適用します。

     ```
     > batch -f ns_backup.conf
     ```

   - ECMP やクラスタリンク集約などの外部トラフィック分散メカニズムを設定します。

7. トラフィックを HA セットアップからクラスタに切り替えます。

   - プライマリアプライアンス NS1 にログオンし、その上にあるすべてのインターフェイスを無効にします。

     ```
     > disable interface <interface_id>
     ```

   - クラスタ IP アドレスにログオンし、NS2 を ACTIVE ノードとして構成します。

     ```
     > set cluster node 0 -state ACTIVE
     ```

   > 注
   >
   > インターフェイスを無効にしてからクラスターノードをアクティブにするまでの間に、少量（秒単位）のダウンタイムが発生する可能性があります。

8. プライマリアプライアンス NS1 にログオンし、HA セットアップから削除します。

   - すべての設定をクリアします。この操作により、HA セットアップから NS1 が削除され、スタンドアロンアプライアンスになります。

     ```
     > clear ns config full
     ```

   - すべてのインターフェイスを有効にします。

     ```
     > enable interface <interface_id>
     ```

9. NS1 をクラスタに追加します。

- クラスタの IP アドレスにログオンし、NS1 をクラスタに追加します。

```
> add cluster node 1 10.102.97.131 -state PASSIVE -backplane
   1/1/1
```

- NS1 にログオンし、次のコマンドを順番に実行してクラスターに参加します。

```
> join cluster -clip 10.102.97.133 -password nsroot

> save ns config

> reboot -warm
```

10. NS1 にログオンし、必要なトポロジーと構成の変更を実行します。
11. クラスタ IP アドレスにログオンし、NS1 を ACTIVE ノードとして設定します。

```
    > set cluster node 1 -state ACTIVE
```

L2 クラスタと L3 クラスタ間の移行

August 15, 2023

> 注
>
> NetScaler 11 以降でサポートされています。

L2 クラスターは、すべてのノードが同じネットワークにあるクラスターで、L3 クラスターは異なるネットワークのノードを含むことができるクラスターです。NetScaler にデプロイされているアプリケーションのダウンタイムなしに、あるタイプのクラスターから別のタイプのクラスターにシームレスに移行できます。

L2 から L3 へのクラスターの移行

クラスターに他のネットワークのノードを含めたい場合は、L3 クラスターに移行できます。

クラスタ IP アドレスで、次の操作を行います。

1. ノードグループを作成します。

   例

```
> add cluster nodegroup NG0
```

   このノードグループは、既存の L2 クラスターからすべてのノードをグループ化するために次のステップで使用されます。

2. L2 クラスターを L3 クラスターに移行します。

   例

   ```
   1 > set cluster instance 1 -inc ENABLED -nodegroup NG0
   ```

   このコマンドは、L3 クラスターに移行することと、L2 クラスターのすべてのノードをノードグループに追加することの 2 つの目的を達成します。

3. これで、クラスタへのノードの追加の説明に従って、クラスタにノードを追加できます。

クラスタの **L3** から **L2** への移行

1 つのネットワークに属するノードを維持したい場合は、L2 クラスターに移行できます。

クラスタ IP アドレスで、次の操作を行います。

1. 保持したくないクラスターノードをネットワークから削除します。

   例

   ```
   1 > rm cluster node <nodeId>
   ```

2. L3 クラスタを L2 クラスタに移行。

   例

   ```
   1 > set cluster instance 1 -inc DISABLED
   ```

   クラスターに 1 つのネットワークのノードのみが含まれるようになりました。

クラスターでの **GSLB** の設定

August 15, 2023

> 注
>
> NetScaler 10.5 ビルド 52.11 以降でサポートされています。

クラスターに GSLB を設定するには、異なる GSLB エンティティをノードグループにバインドする必要があります。ノードグループには 1 つのメンバーノードが必要です。

> メモ
>
> - 静的近接 GSLB 方式を設定した場合は、静的近接データベースがすべてのクラスタノードに存在することを確認してください。これは、データベースファイルがデフォルトの場所にある場合にデフォルトで発生します。ただし、データベースファイルが /var/netscaler/locdb/ 以外のディレクトリに保存されて

いる場合は、ファイルをすべてのクラスターノードに手動で同期する必要があります。

- `show gslb domain` このコマンドはクラスター設定ではサポートされていません。

CLI を使用してクラスターに **GSLB** をセットアップするには**:**

クラスタ IP アドレスにログオンし、コマンドプロンプトで次の操作を実行します。

1. 異なる GSLB エンティティを設定します。詳細については、GSLB 構成エンティティを参照してください。

   注

   GSLB サイトを作成するときは、クラスター IP アドレスとパブリッククラスター IP アドレスを必ず指定してください。パブリッククラスタ IP アドレスは、クラスタが NAT デバイスの背後にデプロイされている場合にのみ必要です。GSLB サイトを設定するときは、同じサイトのクラスター IP アドレスを使用する必要があります。これらのパラメータは、GSLB 自動同期機能を利用できるようにするために必要です。

   ```
   add gslb site <siteName> <siteType> <siteIPAddress> -publicIP <
   ip_addr> -clip <ip_addr> <publicCLIP><!--NeedCopy-->
   ```

2. クラスターノードグループを作成します。

   ```
   add cluster nodegroup <name> <name>@ [-strict ( YES | NO )] [-
   sticky ( YES | NO )] [-state <state>] [-priority <positive_integer
   >]<!--NeedCopy-->
   ```

   注

   VPN ユーザー向けに GSLB ベースを設定する場合は、スティッキーオプションを有効にします。

3. 1 つのクラスターノードをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -node <nodeId><!--NeedCopy-->
   ```

4. ローカル GSLB サイトをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -gslbSite <string><!--NeedCopy-->
   ```

   注

   ローカル GSLB サイト IP アドレスの IP アドレスがストライプ化されていることを確認してください (すべてのクラスターノードで使用可能)。

5. ADNS (または ADNS-TCP) サービスまたは DNS (または DNS-TCP) 負荷分散仮想サーバーをノードグループにバインドします。

   ADNS サービスをバインドするには**:**

   “‘bind cluster nodegroup -service


```
**DNS 負荷分散仮想サーバーをバインドするには:**

```bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
```

6. GSLB 仮想サーバーをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
   ```

7. [オプション] VPN ユーザーに基づいて GSLB を設定するには、VPN 仮想サーバーを GSLB ノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
   ```

8. 設定を確認します。

   ```
   show gslb runningConfig<!--NeedCopy-->
   ```

GUI を使用してクラスターに **GSLB** をセットアップするには**:**

クラスタの IP アドレスにログオンし、[構成] タブで次の操作を実行します。

1. GSLB エンティティを設定します。

   [トラフィック管理**] > [GSLB**] に移動し、必要な設定を実行します。

2. ノードグループを作成し、他のノードグループ関連の設定を行います。

   [システム**] > [**クラスタ**] > [**ノードグループ] に移動し、必要な設定を実行します。

実行する詳細な設定については、前述の CLI 手順に記載されている説明を参照してください。

クラスターでの **GSLB** 親子トポロジーのサポート

NetScaler 12.1 ビルド 49.xx 以降、GSLB 親子トポロジはクラスタでサポートされています。

親子トポロジの詳細については、MEP プロトコルを使用した親子トポロジの展開を参照してください。

CLI を使用してクラスターに **GSLB** 親子トポロジを設定するには

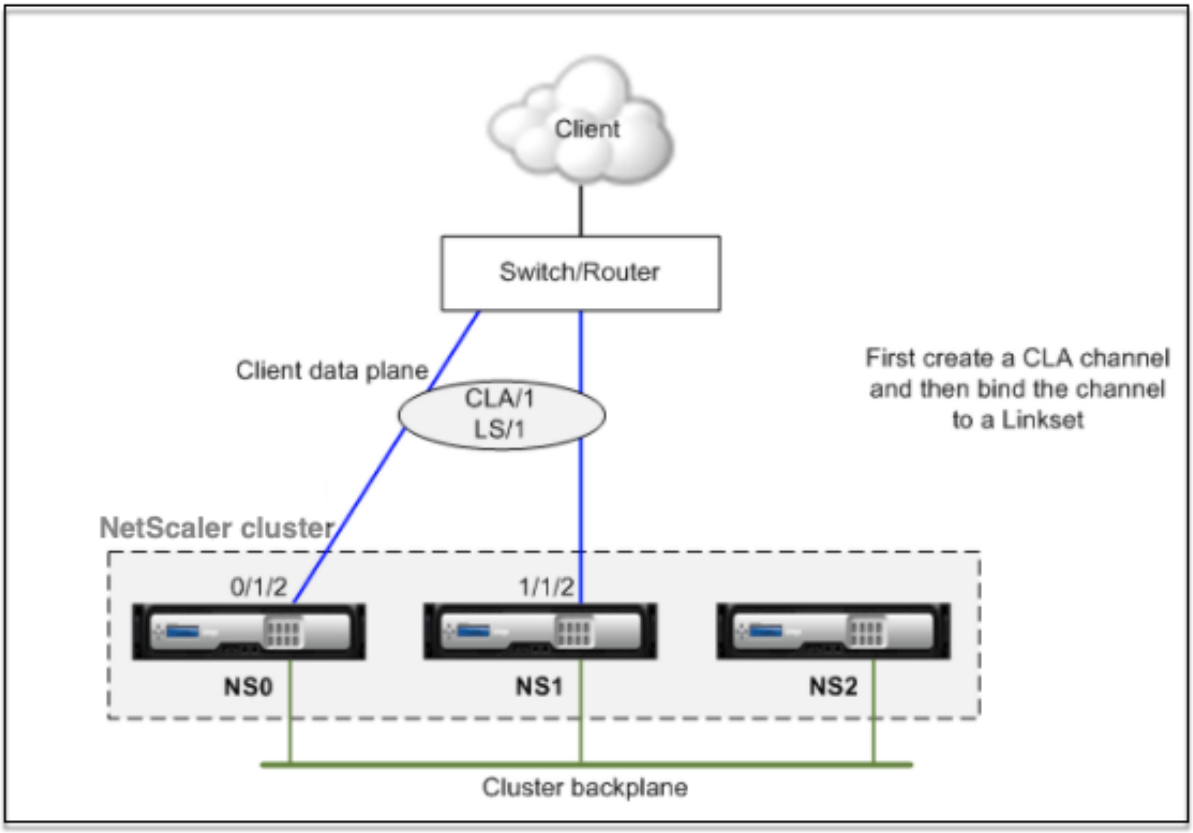
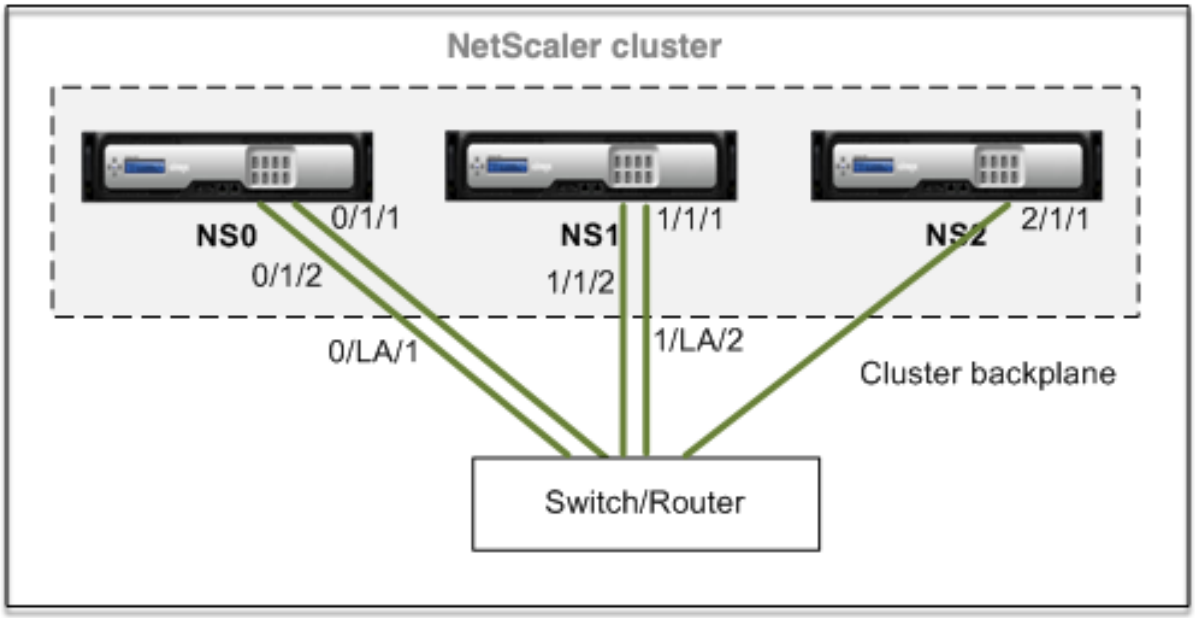
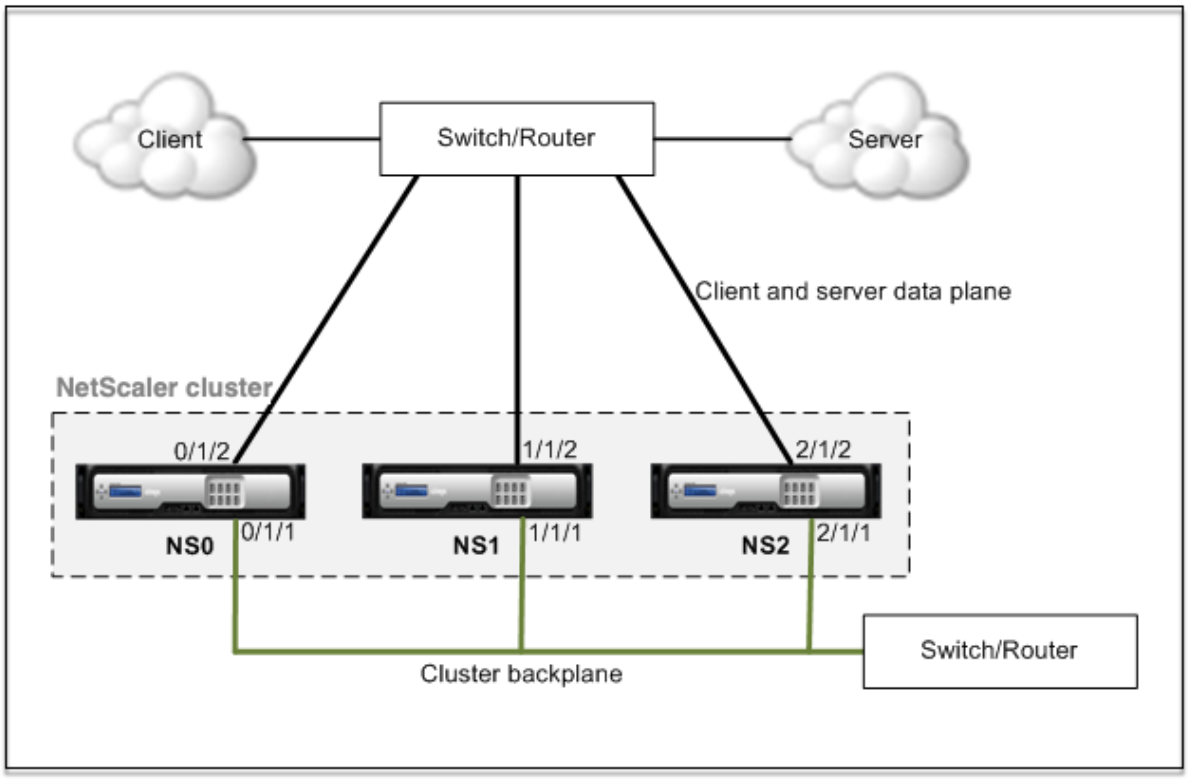
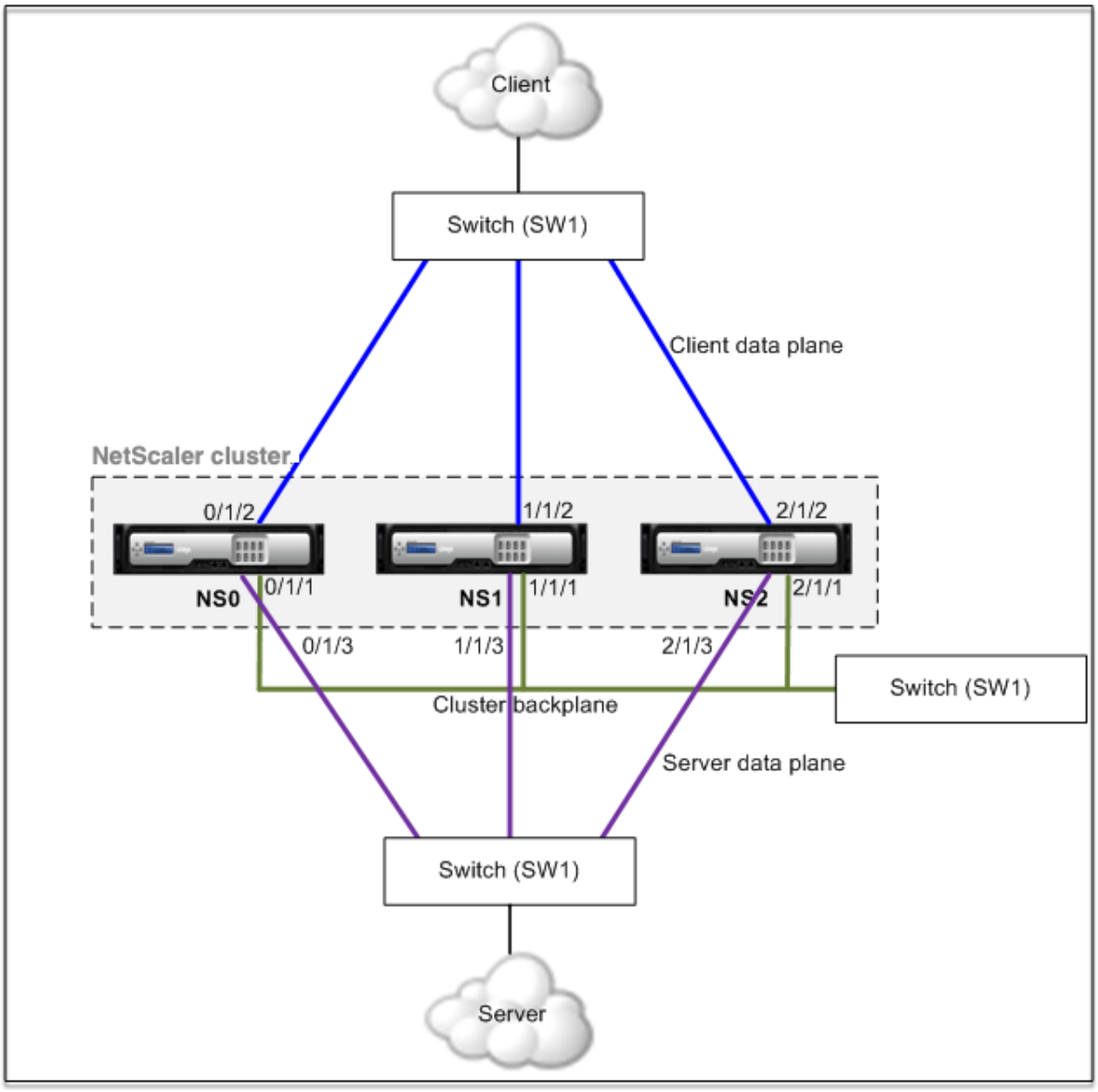
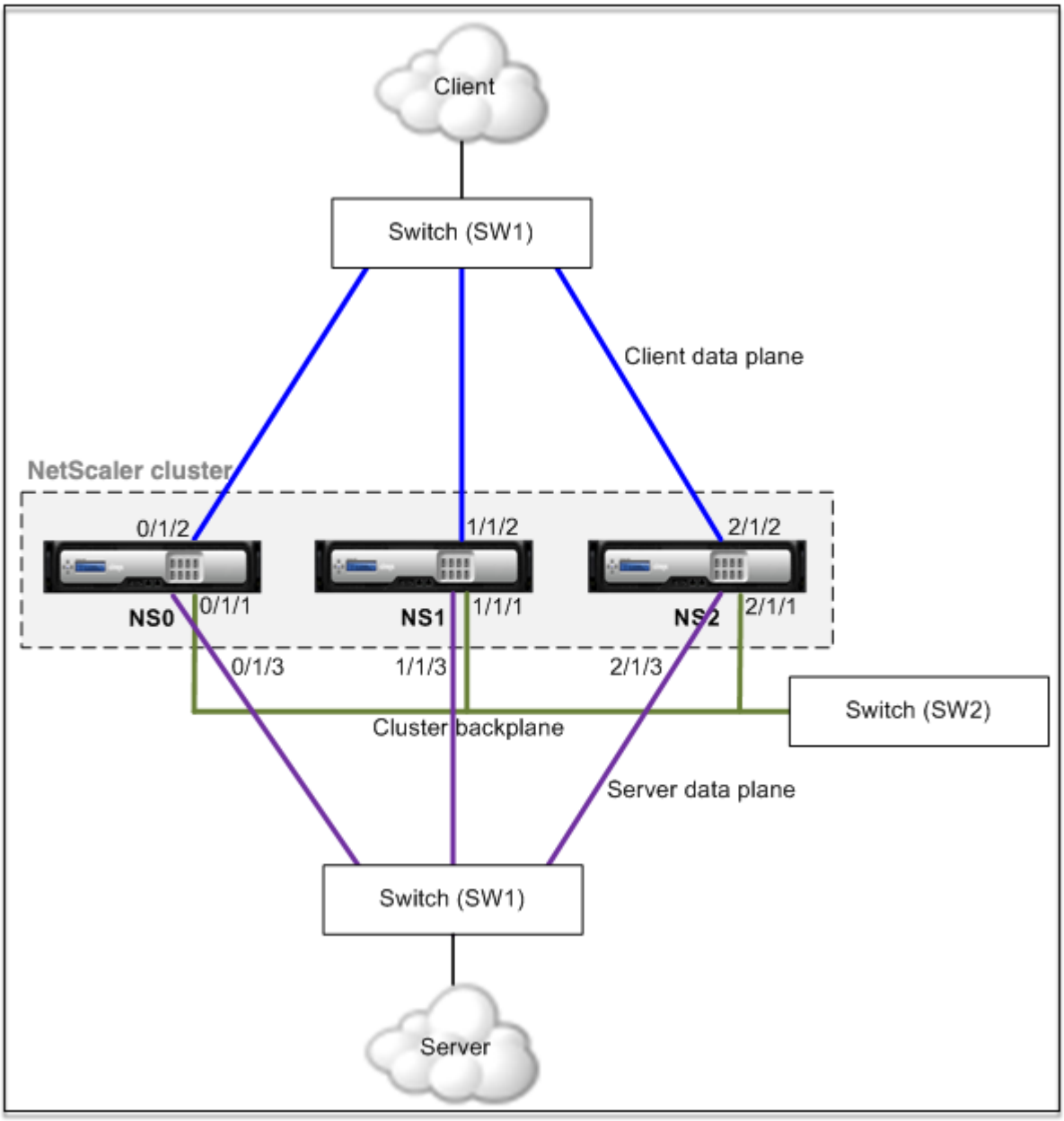
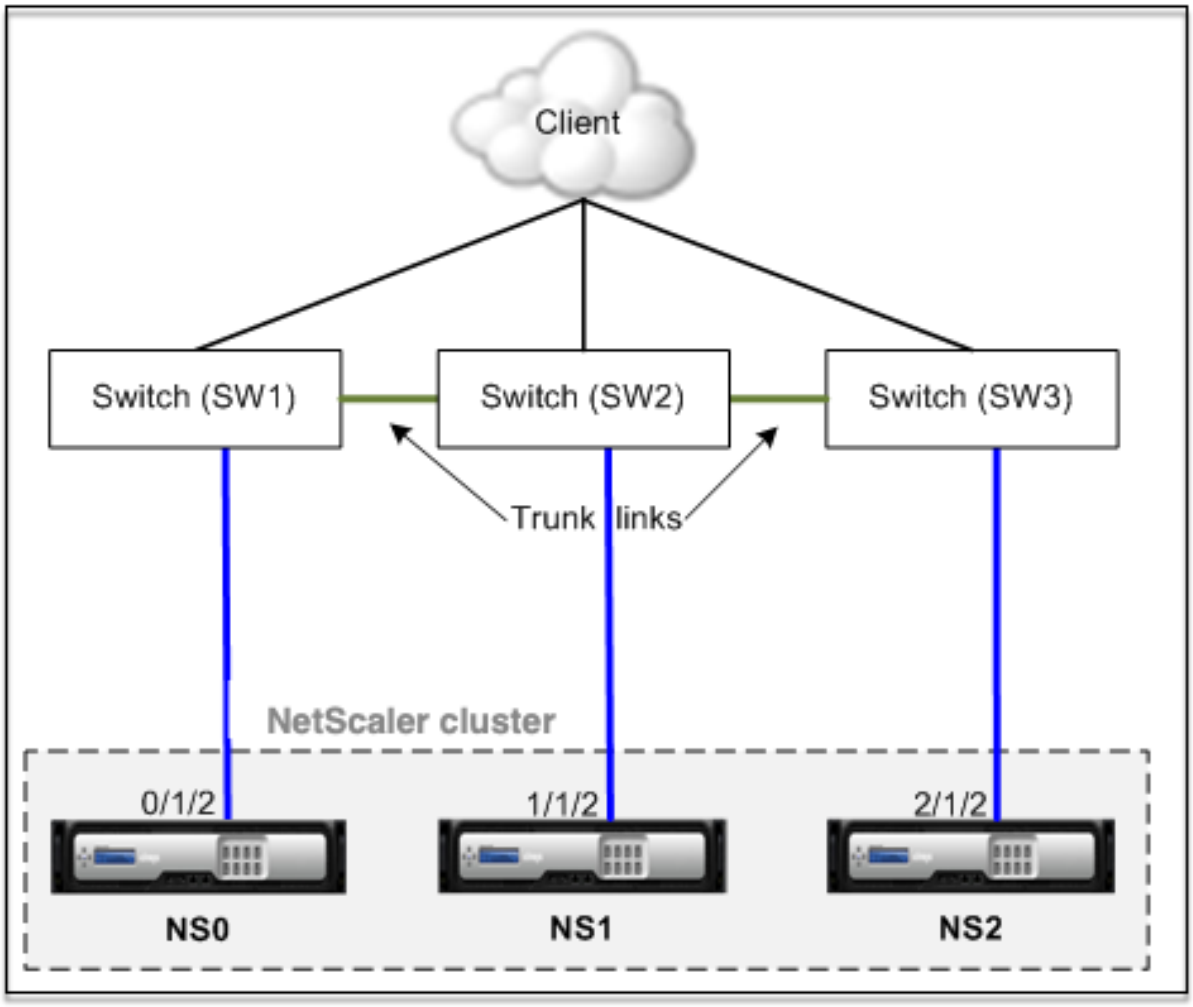
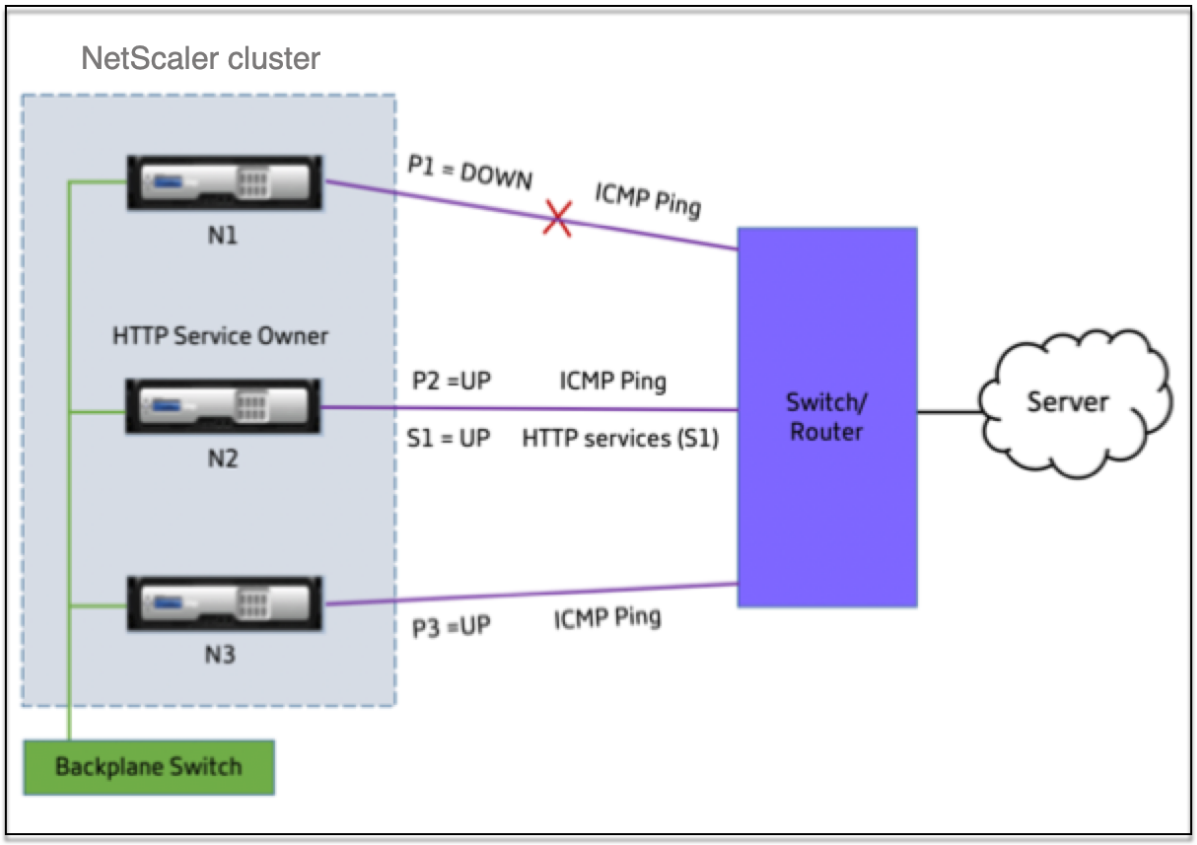
親サイト　次の設定を行います。

1. クラスターノードグループを作成します。

   ```
   add cluster nodegroup <name>
   ```

   例**:**

   ```
   add cluster nodegroup parentng
   ```


   ```

2. 1 つのクラスターノードをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -node <nodeId>
   ```

   例:

   ```
   bind cluster nodegroup parentng -node n2
   ```

3. ローカル GSLB サイトをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -gslbSite <string>
   ```

   例:

   ```
   bind cluster nodegroup parentng -gslbSite site1
   ```

4. ADNS (または ADNS-TCP) サービスまたは DNS (または DNS-TCP) 負荷分散仮想サーバーをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -service <string>
   ```

   例:

   ```
   bind cluster nodegroup parentng - service ADNS
   ```

5. GSLB 仮想サーバーをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -vServer <string>
   ```

   例:

   ```
   bind cluster nodegroup parentng -vServce gslbvs1
   ```

チャイルドサイト　次の設定を行います。

1. クラスターノードグループを作成します。

   ```
   add cluster nodegroup <name>
   ```

   例:

   ```
   add cluster nodegroup childng
   ```

2. 1 つのクラスターノードをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -node <nodeId>
   ```

   例:

   ```
   bind cluster nodegroup childng -node -n3
   ```

3. ローカル GSLB サイトをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -gslbSite <string>
   ```

   例:


```
bind cluster nodegroup childng -gslbSite site1
```

> 注
>
> 親サイトと子サイトがメトリックベースの負荷分散方式で集計された統計を交換するには、子サイトにローカル GSLB サービスを追加する必要があります。メトリックベースの負荷分散方法には、最小接続、最小帯域幅、最小パケットがあります。

GUI を使用してクラスターに **GSLB** 親子トポロジを設定するには

1. GSLB エンティティを設定します。

   [トラフィック管理**]** > **[GSLB]** に移動して、必要な設定を実行します。

2. ノードグループを作成します。

   [システム**]** > **[**クラスタ**]** > **[**ノードグループ**]** に移動して、必要な構成を実行します。

3. 「ノードグループ」ページで、ノードをバインドするノードグループを選択し、「編集」をクリックして、次のタスクを実行します。これらのタスクは、ノードグループを追加するときにも実行できます。

   - ノードをノードグループにバインドします。

     [詳細 設定**]** で、[クラスターノード] をクリックして、次のタスクを実行します。

     - 「クラスターノード」セクションで、「クラスターノードなし」をクリックします。
     - **[**クラスターノードの選択] で、[> そして、ノードグループにバインドするノードを選択します。クラスターノードを追加することもできます。

   - ローカル GSLB サイトをノードグループにバインドします。

     [詳細設定] で、[GSLB サイト] をクリックし、次のタスクを実行します。

     - 「**GSLB** サイト」セクションで、「GSLB サイトなし」をクリックします。
     - 「**GSLB** サイトの選択」で、> をクリックし、ノードグループにバインドする GSLB サイトを選択します。GSLB サイトを追加することもできます。

   - GSLB 仮想サーバーをノードグループにバインドします。

     [詳細 設定**]** で、[仮想サーバー] をクリックして、次のタスクを実行します。

     - 「仮想サーバー」ペインで、「+」をクリックします。
     - 「仮想サーバーの選択」で、ノードグループにバインドするサーバーを選択します。

   - ADNS (または ADNS-TCP) サービスまたは DNS (または DNS-TCP) 負荷分散仮想サーバーをノードグループにバインドします。

     [詳細 設定**]** で、[サービス] をクリックして、次のタスクを実行します。

     - 「サービス」セクションで、「サービスなし」をクリックします。

- 「サービスの選択」で、ノードグループにバインドするサービスを選択します。サービスを追加することもできます。

> 注
>
> 子サイトの場合、クラスターノードとローカル GSLB サイトをノードグループにバインドするだけで済みます。

クラスターでのキャッシュリダイレクトの使用

August 15, 2023

クラスター内のキャッシュリダイレクトは、スタンドアロンの NetScaler アプライアンスと同じように機能します。唯一の違いは、構成がクラスタ IP アドレスで実行されることです。キャッシュリダイレクトの詳細については、「キャッシュリダイレクト」を参照してください。

クラスターでトランスペアレントモードでキャッシュのリダイレクトを使用する際の注意点**:**

- キャッシュリダイレクションを設定する前に、すべてのノードを外部スイッチに接続し、リンクセットを設定していることを確認してください。そうしないと、クライアントリクエストはドロップされます。
- 負荷分散仮想サーバーで MAC モードが有効になっている場合は、enable ns mode MBF コマンドを使用して、クラスターで MBF モードが有効になっていることを確認します。それ以外の場合、リクエストはキャッシュサーバーに送信されるのではなく、オリジンサーバーに直接送信されます。

クラスターセットアップでの **L2** モードの使用

August 15, 2023

> 注
>
> NetScaler 10.5 以降のリリースでサポートされています。

クラスター設定で L2 モードを使用するには、次の点を確認する必要があります。

- スポッティング IP アドレスは、必要に応じてすべてのノードで使用できる必要があります。
- 外部ネットワークとの通信にはリンクセットを使用する必要があります。
- 非対称トポロジまたは非対称クラスタ LA グループはサポートされていません。
- クラスタ LA グループが推奨されます。
- トラフィックは、サービスが存在するデプロイメントに対してのみ、クラスタノード間で分散されます。

リンクセットでクラスター **LA** チャネルを使用

August 15, 2023

非対称クラスタートポロジでは、一部のクラスターノードはアップストリームネットワークに接続されていません。このような場合は、リンクセットを使用する必要があります。パフォーマンスを最適化するには、スイッチに接続されているインターフェイスをクラスタ LA チャネルとしてバインドし、そのチャネルをリンクセットにバインドします。

クラスタ LA チャネルとリンクセットを組み合わせて使用する方法を理解するには、アップストリームスイッチで使用できるポートが 2 つしかない 3 ノードクラスタを考えてみます。2 つのクラスタノードをスイッチに接続し、もう 1 つのノードは接続しないままにしておくことができます。

> 注
>
> 同様に、非対称トポロジでは ECMP とリンクセットを組み合わせて使用することもできます。

図 1: リンクセットとクラスタ LA チャネルトポロジー



CLI を使用してクラスタ **LA** チャネルとリンクセットを設定するには

1. クラスタ IP アドレスにログオンします。
2. 接続されたインターフェイスをクラスタ LA チャネルにバインドします。

```
add channel CLA/1 -ifnum 0/1/2 1/1/2
```

3. クラスタ LA チャネルをリンクセットにバインドします。

```
add linkset LS/1 -ifnum CLA/1
```

LA チャネルのバックプレーン

August 15, 2023

この導入では、LA チャネルがクラスタバックプレーンに使用されます。



- NS0-ノード ID: 0、NSIP: 10.102.29.60
- NS1-ノード ID: 1、NSIP: 10.102.29.70
- NS2-ノード ID: 2、NSIP: 10.102.29.80

バックプレーンインターフェイスを **LA** チャネルとして持つクラスタを導入するには

1. NS0、NS1、NS2 のノードのクラスタを作成します。

a) クラスタに追加する最初のノードにログオンし、次の操作を行います。

```
> create cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE
> enable cluster instance 1
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

b) クラスタの IP アドレスにログオンし、次の操作を行います。

```
> add cluster node 1 10.102.29.70 -state ACTIVE
> add cluster node 2 10.102.29.80 -state ACTIVE
```

c) ノード 10.102.29.70 および 10.102.29.80 にログオンし、ノードをクラスタに参加させます。

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot -warm
```

前のコマンドに見られるように、インターフェース 0/1/1, 1/1/1, そして 2/1/1 3 つのクラスターノードのバックプレーンインターフェイスとして構成されます。

2. クラスタの IP アドレスにログオンし、次の操作を行います。

a) ノード NS0 および NS1 の LA チャネルを作成します。

```
> add channel 0/LA/1 -ifnum 0/1/1 0/1/2
> add channel 1/LA/2 -ifnum 1/1/1 1/1/2
```

b) クラスタノードのバックプレーンを設定します。

```
> set cluster node 0 -backplane 0/LA/1
> set cluster node 1 -backplane 1/LA/2
> set cluster node 2 -backplane 2/1/1
```

クライアントとサーバーの共通インターフェースおよびバックプレーンの専用インターフェース

August 15, 2023

NetScaler クラスターをワンアームで導入できます。この展開では、クライアントとサーバーのネットワークが同じインターフェイスを使用してクラスターと通信します。クラスタバックプレーンは、ノード間通信に専用のインターフェイスを使用します。



- NS0-ノード ID: 0、NSIP: 10.102.29.60
- NS1-ノード ID: 1、NSIP: 10.102.29.70
- NS2-ノード ID: 2、NSIP: 10.102.29.80

  クライアントとサーバ用の共通のインターフェイスと、クラスタバックプレーン用の異なるインターフェイスを持つクラスタを展開するには

  1. NS0、NS1、NS2 のノードのクラスタを作成します。
  2. クラスタに追加する最初のノードにログオンし、次の操作を行います。

```
> create cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE -backplane
    0/1/1
> enable cluster instance 1
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

  3. クラスタの IP アドレスにログオンし、次の操作を行います。

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
    1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
    2/1/1
```


4. ノード 10.102.29.70 および 10.102.29.80 にログオンし、ノードをクラスタに参加させます。

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot -warm
```

前のコマンドに見られるように、インターフェース 0/1/1, 1/1/1, そして 2/1/1 3 つのクラスターノードのバックプレーンインターフェイスとして構成されます。

1. クラスタ IP アドレスに、バックプレーンインターフェイスとクライアントインターフェイスとサーバインターフェイス用の VLAN を作成します。

//バックプレーンインタフェースの場合

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

//クライアントおよびサーバーネットワークに接続されているインターフェイス用。

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

2. スイッチ上で、バックプレーンインターフェイス、およびクライアントおよびサーバインターフェイスに対応するインターフェイスの VLAN を作成します。Cisco® Nexus 7000 C7010 リリース 5.2 (1) スイッチの設定例を以下に示します。他のスイッチでも同様の設定を行う必要があります。

//バックプレーンインターフェース用。Repeat for each interface…

```
> interface Ethernet2/47
  switchport access vlan 100
  switchport mode access
  end
```

//クライアントおよびサーバーネットワークに接続されたインターフェースの場合。Repeat for each interface…

```
> interface Ethernet2/47
  switchport access vlan 200
  switchport mode access
  end
```

クライアント、サーバー、およびバックプレーンの共通スイッチ

August 15, 2023

この展開では、クライアント、サーバー、バックプレーンは同じスイッチ上の専用インターフェイスを使用して NetScaler クラスターと通信します。



- NS0-ノード ID: 0、NSIP: 10.102.29.60
- NS1-ノード ID: 1、NSIP: 10.102.29.70
- NS2-ノード ID: 2、NSIP: 10.102.29.80

クライアント、サーバ、およびバックプレーン用の共通スイッチを使用してクラスタを展開するには

1. NS0、NS1、NS2 のノードのクラスタを作成します。
2. クラスタに追加する最初のノードにログオンし、次の操作を行います。

```
1 > create cluster instance 1
2 > add cluster node 0 10.102.29.60 -state ACTIVE -backplane
     0/1/1
3 > enable cluster instance 1
```



```
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

3. クラスタの IP アドレスにログオンし、次の操作を行います。

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
    1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
    2/1/1
```

4. ノード 10.102.29.70 および 10.102.29.80 にログオンし、ノードをクラスタに参加させます。

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot -warm
```

前のコマンドに見られるように、インターフェース 0/1/1, 1/1/1, そして 2/1/1 3 つのクラスターノードのバックプレーンインターフェイスとして構成されます。

1. クラスタ IP アドレスに、バックプレーン、クライアント、およびサーバインターフェイスの VLAN を作成します。

   //バックプレーンインタフェースの場合

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

   //クライアント側インタフェースの場合

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

   //サーバー側インタフェースの場合

```
> add vlan 30
> bind vlan 30 0/1/3 1/1/3 2/1/3
```

2. スイッチ上で、バックプレーンインターフェイス、およびクライアントおよびサーバインターフェイスに対応するインターフェイスの VLAN を作成します。Cisco® Nexus 7000 C7010 リリース 5.2 (1) スイッチの設定例を以下に示します。他のスイッチでも同様の設定を行う必要があります。

   //バックプレーンインターフェース用。Repeat for each interface…

```
> interface Ethernet2/47
  switchport access vlan 100
  switchport mode access
  end
```

   //クライアントインターフェース用。Repeat for each interface…


```
> interface Ethernet2/48
  switchport access vlan 200
  switchport mode access
  end
```

//サーバーインターフェース用。Repeat for each interface…

```
> interface Ethernet2/49
  switchport access vlan 300
  switchport mode access
  end
```

クライアントとサーバー用の共通スイッチとバックプレーン用の専用スイッチ

August 15, 2023

この展開では、クライアントとサーバーは同じスイッチ上の異なるインターフェイスを使用して NetScaler クラスターと通信します。クラスタバックプレーンは、ノード間通信に専用スイッチを使用します。



- NS0-ノード ID: 0、NSIP: 10.102.29.60
- NS1-ノード ID: 1、NSIP: 10.102.29.70
- NS2-ノード ID: 2、NSIP: 10.102.29.80

  クライアントとサーバには同じスイッチを使用し、クラスタバックプレーンには別のスイッチを使用してクラスタを展開するには

1. NS0、NS1、NS2 のノードのクラスタを作成します。

   - クラスタに追加する最初のノードにログオンし、次の操作を行います。


```
> create cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE -backplane
   0/1/1
> enable cluster instance 1
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

- クラスタの IP アドレスにログオンし、次の操作を行います。

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
   1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
   2/1/1
```

- ノード 10.102.29.70 および 10.102.29.80 にログオンし、ノードをクラスタに参加させます。

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot -warm
```

前のコマンドに見られるように、インターフェース 0/1/1, 1/1/1, そして 2/1/1 3 つのクラスターノードのバックプレーンインターフェイスとして構成されます。

2. クラスタ IP アドレスに、バックプレーン、クライアント、およびサーバインターフェイスの VLAN を作成します。

   //バックプレーンインタフェースの場合

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

   //クライアント側インタフェースの場合

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

   //サーバー側インタフェースの場合

```
> add vlan 30
> bind vlan 30 0/1/3 1/1/3 2/1/3
```

3. スイッチ上で、バックプレーンインターフェイス、およびクライアントおよびサーバインターフェイスに対応するインターフェイスの VLAN を作成します。Cisco® Nexus 7000 C7010 リリース 5.2 (1) スイッチの設定例を以下に示します。他のスイッチでも同様の設定を行う必要があります。

   //バックプレーンインターフェース用。Repeat for each interface…

```
> interface Ethernet2/47
> switchport access vlan 100
> switchport mode access
```



```
4  > end
```

//クライアントインターフェース用。Repeat for each interface…

```
1  > interface Ethernet2/48
2  > switchport access vlan 200
3  > switchport mode access
4  > end
```

//サーバーインターフェース用。Repeat for each interface…

```
1  > interface Ethernet2/49
2  > switchport access vlan 300
3  > switchport mode access
4  > end
```

ノードごとに異なるスイッチ

August 15, 2023

この導入では、各クラスタノードは異なるスイッチに接続され、スイッチ間にトランクリンクが設定されます。



クラスター構成は、他のデプロイメントシナリオと同じです。クライアント側の設定のほとんどは、クライアント側スイッチで行われます。

クラスター構成のサンプル

August 15, 2023

次の例を使用して、ECMP、クラスタ LA、またはリンクセットを使用して 4 ノードクラスタを構成できます。

1. クラスターを作成します。
    - 最初のノードにログインします。
    - クラスターインスタンスを追加します。

```
1   > add cluster instance 1
```


- 最初のノードをクラスターに追加します。

```
1   > add cluster node 0 10.102.33.184 -backplane 0/1/1
```

- クラスターインスタンスを有効にします。

```
1   > enable cluster instance 1
```

- クラスタ IP アドレスを追加します。

```
1   > add ns ip 10.102.33.185 255.255.255.255 -type CLIP
```

- 構成を保存します。

```
1   > save ns config
```

- アプライアンスをウォーム再起動します。

```
1   > reboot -warm
```

2. 他の 3 つのノードをクラスターに追加します。

- クラスタ IP アドレスにログインします。
- 2 番目のノードをクラスターに追加します。

```
1   > add cluster node 1 10.102.33.187 -backplane 1/1/1
```

- 3 番目のノードをクラスターに追加します。

```
1   > add cluster node 2 10.102.33.188 -backplane 2/1/1
```

- 4 番目のノードをクラスターに追加します。

```
1   > add cluster node 3 10.102.33.189 -backplane 3/1/1
```

3. 追加したノードをクラスターに参加させます。このステップは最初のノードには適用されません。

- 新しく追加された各ノードにログオンします。
- ノードをクラスタに参加させます。

```
1   > join cluster -clip 10.102.33.185 -password nsroot
```

- 構成を保存します。

```
1   > save ns config
```

- アプライアンスをウォーム再起動します。

```
1   > reboot -warm
```


4. クラスター IP アドレスを使用して NetScaler ADC クラスターを構成します。

   // 負荷分散機能を有効にする

   ```
   > enable ns feature lb
   ```

   // 負荷分散仮想サーバーを追加します

   ```
   > add lb vserver first_lbvserver http
   ....
   ....
   ```

5. クラスタに対して次のいずれかの（ECMP、クラスタ LA、または Linkset）トラフィック分散メカニズムを設定します。

   ECMP

   - クラスタ IP アドレスにログオンします。
   - OSPF ルーティングプロトコルを有効にします。

     ```
     > enable ns feature ospf
     ```

   - VLAN を追加します。

     ```
     > add vlan 97
     ```

   - クラスタノードのインタフェースを VLAN にバインドします。

     ```
     > bind vlan 97 -ifnum 0/1/4 1/1/4 2/1/4 3/1/4
     ```

   - 各ノードにスポット付き SNIP を追加し、そのノードで動的ルーティングを有効にします。

     ```
     > add ns ip 1.1.1.10 255.255.255.0 -ownerNode 0 -
        dynamicRouting ENABLED
     > add ns ip 1.1.1.11 255.255.255.0 -ownerNode 1 -
        dynamicRouting ENABLED
     > add ns ip 1.1.1.12 255.255.255.0 -ownerNode 2 -
        dynamicRouting ENABLED
     > add ns ip 1.1.1.13 255.255.255.0 -ownerNode 3 -
        dynamicRouting ENABLED
     ```

   - SNIP アドレスのいずれかを VLAN にバインドします。

     ```
     > bind vlan 97 -ipAddress 1.1.1.10 255.255.255.0
     ```

   - VTYSH シェルを使用して、ZebOS でルーティングプロトコルを構成します。

     スタティッククラスタ **LA**

     - クラスタ IP アドレスにログオンします。
     - クラスタ LA チャネルを追加します。


```
1   > add channel CLA/1 -speed 1000
```

- インターフェイスをクラスタ LA チャネルにバインドします。

```
1   > bind channel CLA/1 0/1/5 1/1/5 2/1/5 3/1/5
```

- スイッチで同等の設定を実行します。

  ダイナミッククラスタ **LA**

  * クラスタ IP アドレスにログオンします。
  * クラスタの LA チャネルにインターフェイスを追加します。

```
1   > set interface 0/1/5 -lacpmode active -lacpkey 5 -
      lagtype cluster
2   > set interface 1/1/5 -lacpmode active -lacpkey 5 -
      lagtype cluster
3   > set interface 2/1/5 -lacpmode active -lacpkey 5 -
      lagtype cluster
4   > set interface 3/1/5 -lacpmode active -lacpkey 5 -
      lagtype  cluster
```

  * スイッチで同等の設定を実行します。

  Linksets. NodeID 3 のノードがスイッチに接続されていないと仮定します。接続されていないノードが他のノードインターフェイスを使用してスイッチと通信できるように、リンクセットを設定する必要があります。

a) クラスタ IP アドレスにログオンします。

b) リンクセットを追加します。

```
1  > add linkset LS/1
```

c) 接続されたインターフェイスをリンクセットにバインドします。

```
1  > bind linkset LS/1 -ifnum 0/1/6 1/1/6 2/1/6
```

6. クラスタノードの状態を ACTIVE に更新します。

```
1  > set cluster node 0 -state ACTIVE
2  > set cluster node 1 -state ACTIVE
3  > set cluster node 2 -state ACTIVE
4  > set cluster node 3 -state ACTIVE
```

クラスターセットアップでの **VRRP** の使用

August 15, 2023

仮想ルータ冗長プロトコル (VRRP) は、IPv4 と IPv6 の両方のクラスタ設定でサポートされています。クラスタセットアップでサポートされる 2 つの VRRP 機能は、インターフェイスベースの VRRP と IP ベースの VRRP です。

IP ベースの VRRP

IP ベースの VRRP では、同じ VRID にバインドされたストライプ VIP アドレスがクラスタ設定のすべてのノードに設定されます。これらの VIP アドレスはすべてのノードでアクティブです。

クラスタノードの 1 つが VRID 所有者として機能し、VRRP アドバタイズメントを他のノードに送信します。VRID 所有者ノードに障害が発生すると、クラスタ内の別のノードが VRID の所有権を引き継ぎ、VRRP アドバタイズメントの送信を開始します。特定のクラスタノードを VRID の所有者として割り当てることもできます。

> 注
>
> Citrix では、クラスタでの VRRP 導入には IP ベースの方法を使用することをお勧めします。

IPv4 用の IP ベースの VRRP の設定

IPv4 用の IP ベースの VRRP を設定するには、クラスタセットアップで次のタスクを実行します。

- **VRID** を追加します。VRID は、クラスタセットアップが仮想 MAC アドレスを形成するために使用する整数です。<VRID\ > 汎用 VMAC アドレスは 00:00:5 e: 00:02:\ の形式です。
- **(**オプション**)** 仮想 **MAC** アドレスの所有者としてノードを割り当てます。クラスターノードの ID に所有者ノードパラメーター (VRID6 の追加または変更中) を設定して、そのクラスターノードを仮想 MAC アドレスの所有者として割り当てることができます。割り当てられた所有者ノードに障害が発生すると、UP クラスタノードの 1 つが仮想 MAC アドレスの所有者として動的に選択されます。オーナーノードは、`set vrID <id> -ownerNode <positive_interger>` コマンドを使用して設定できます。
- **VRID** をノードの **VIP** アドレスにバインドします。作成した VRID をストライプ VIP アドレスにバインドします。

CLI を使用して VRID を追加するには

コマンドプロンプトで入力します。

```
1 -  add vrid <ID> [-ownerNode <positive_integer>]
2 -  show vrid <ID>
```

CLI を使用して VRID を VIP アドレスにバインドするには

コマンドプロンプトで入力します。

- `set ns ip <IPv4Address> -vrid <ID><!--NeedCopy-->`
- `show vrid <ID><!--NeedCopy-->`

GUI を使用して **VRID** を追加するには

1. **[**システム**] > [**ネットワーク**] > [VMAC**] に移動し、[**VMAC**] タブで [追加] をクリックします。
2. [**VMAC** の作成] ページで、[仮想ルータ **ID**] フィールドに値を指定し、[作成] をクリックします。

GUI を使用して **VRID** を **VIP** アドレスにバインドするには

1. **[**システム**] > [**ネットワーク**] > [IP**] に移動し、[**IPV4s**] タブで VIP アドレスを選択し、[編集] をクリックします。
2. VIP 設定の編集中に、仮想ルータ **ID** パラメータを設定します。

```
> add vrid 90
 Done
> set ns ip 192.0.2.90  - vrid 90
 Done
```

IPv6 用の **IP** ベースの **VRRP** の設定

IPv6 用の IP ベースの VRRP を設定するには、クラスタセットアップで次のタスクを実行します。

- **VRID6** を追加します。VRID6 は、クラスタセットアップが仮想 MAC6 アドレスを形成するために使用する整数です。<VRID6\ > 汎用 VMAC6 アドレスは 00:00:5 e: 00:02:\ の形式です。
- **(**オプション**)** 仮想 **MAC6** アドレスの所有者としてノードを割り当てます。クラスターノードの ID に所有者ノードパラメーター (VRID6 の追加または変更中) を設定して、そのクラスターノードを仮想 MAC6 アドレスの所有者として割り当てることができます。割り当てられた所有者ノードに障害が発生した場合、UP クラスタノードの 1 つが仮想 MAC6 アドレスの所有者として動的に選択されます。
- **VRID6** をノードの **VIP6** アドレスにバインドします。作成した VRID6 をストライプ VIP6 アドレスにバインドします。

CLI を使用して **VRID6** を追加するには

コマンドプロンプトで入力します。

- `add vrid6 <ID> [-ownerNode <positive_integer>]<!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

CLI を使用して **VRID6** を **VIP6** アドレスにバインドするには

コマンドプロンプトで入力します。

- `set ns ip6 <IPv6Address> -vrid6 <ID><!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

GUI を使用して VRID6 を追加するには

1. **[**システム**] > [**ネットワーク**] > [VMAC**] に移動し、[**VMAC6**] タブで [追加] をクリックします。
2. [仮想 **MAC6** の作成] ページで、[仮想ルータ **ID**] フィールドに値を指定し、[作成] をクリックします。

GUI を使用して VRID6 を VIP6 アドレスにバインドするには

1. **[**システム**] > [**ネットワーク**] > [IP**] に移動し、[**IPV6s**] タブで VIP アドレスを選択し、[編集] をクリックします。
2. VIP6 設定の編集中に、仮想ルータ **ID** パラメータを設定します。

```
> add vrid6 90
 Done
> set ns ip6 2001:db8::5001  - vrid6 90
 Done
```

インターフェイスベースの VRRP

インターフェイスベースの VRRP 機能では、クラスタの両方のノードに同じ仮想 MAC アドレスが設定されます。この仮想 MAC アドレスは、ノードに設定された IP アドレスの GARP アドバタイズメントと ARP 応答に使用されます。この機能は、GARP アドバタイズメントを受け入れない外部デバイス/ルータを含むアクティブスペア 2 ノードクラスタセットアップで役立ちます。

> 注
>
> インターフェイスベースの VRRP 機能は、1 つのノードがアクティブ状態で、もう 1 つのノードがスペアとして機能する 2 ノードクラスタにのみ適用されます。

両方のクラスタノードで同じ仮想 MAC アドレスを使用すると、アクティブノードがダウンしてスペアノードがアクティブに引き継がれても、新しいアクティブノードの IP アドレスの MAC アドレスは変更されず、外部デバイス/ルーターの ARP テーブルを更新する必要はありません。

IPv4 用のインターフェイスベースの VRRP の設定

IPv4 用のインターフェイスベースの VRRP を設定するには、クラスタセットアップで次のタスクを実行します。

- **VRID** を追加します。VRID は、クラスタセットアップが仮想 MAC アドレスを形成するために使用する整数です。
- **VRID** をノードインターフェイスにバインドします。作成した VRID にインターフェイスをバインドします。バインドされたインターフェイス（現在のアクティブノード内）は、IPv4 アドレスの GARP アドバタイズメントと ARP 応答の仮想 MAC アドレスを使用します。VRID をアクティブスペアクラスタセットアップの両方

のノードのインターフェイスに関連付ける必要があります。これは、高可用性セットアップとは異なり、クラスターセットアップではインターフェイス ID が異なるためです。

CLI を使用して **VRID** を追加するには

コマンドプロンプトで入力します。

```
- add vrid <ID>
- show vrid <ID>
```

CLI を使用して **VRID** をインターフェイスにバインドするには

コマンドプロンプトで入力します。

```
- bind vrid <ID> -ifnum <interface_name>
- show vrid <ID>
```

GUI を使用して **VRID** を追加し、インターフェイスにバインドするには

1. **[**システム**] > [**ネットワーク**] > [VMAC**] に移動し、[**VMAC**] タブで [追加] をクリックします。
2. [仮想 **MAC** の作成] ページで、[仮想ルータ **ID***] フィールドに値を指定し、[インターフェイスの関連付け] セクションでインターフェイスをバインドして、[作成] をクリックします。

```
> add vrid 300
 Done
> bind vrid 300 -ifnum 1/1/2 2/1/3
 Done
```

IPv6 用のインターフェイスベースの **VRRP** の設定

IPv6 用のインターフェイスベースの VRRP を設定するには、クラスタセットアップで次のタスクを実行します。

- **VRID6** を追加します。VRID6 は、クラスタセットアップが仮想 MAC6 アドレスを形成するために使用する整数です。<VRID6\ > 汎用 VMAC6 アドレスは 00:00:5 e: 00:01:\ の形式です。
- **VRID6** をノードインターフェイスにバインドします。作成した VRID6 にインターフェイスをバインドします。バインドされたインターフェイス（現在のアクティブノード内）は、IPv6 アドレスの GARP アドバタイズメントと ARP 応答の仮想 MAC6 アドレスを使用します。VRID6 をアクティブスペアクラスタセットアップの両方のノードのインターフェイスに関連付ける必要があります。これは、高可用性セットアップとは異なり、クラスターセットアップではインターフェイス ID が異なるためです。

CLI を使用して **VRID6** を追加するには

コマンドプロンプトで入力します。

```
1 -  add vrid6 <ID>
2 -  show vrid6 <ID>
```

CLI を使用して **VRID6** をインターフェイスにバインドするには

コマンドプロンプトで入力します。

- `bind vrid6 <ID> -ifnum <interface_name><!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

GUI を使用して **VRID6** を追加してインターフェイスにバインドするには

1. **[**システム**] > [**ネットワーク**] > [VMAC**] に移動し、[**VMAC6**] タブで [追加] をクリックします。
2. [仮想 **MAC6** の作成] ページで、[仮想ルータ **ID**] フィールドに値を指定し、[インターフェイスの関連付け] セクションでインターフェイスをバインドして、[作成] をクリックします。

```
1 > add vrid6 100
2  Done
3 > bind vrid6 100 -ifnum 0/1/1 1/1/2 2/1/3
4  Done
```

パス監視を使用したクラスター内のサービスの監視

August 15, 2023

クラスタ構成では、監視サービスの所有権はノード間で分散されます。したがって、さまざまなノードがさまざまなサービスを監視します。サービスを監視するノードはサービスオーナーと呼ばれます。サービス所有者のみがサーバーをプローブして、割り当てられたサービスのステータスを監視します。さらに、サービスのステータスをクラスター内の他のすべてのノードに伝達します。分散監視の欠点は、すべてのノードとサーバー間のネットワーク接続とリンク状態が決定されないことです。この欠点を克服するには、パスモニタリングを使用できます。

> 注
>
> サービスを監視するノードは選択できません。サービスを監視するノードの選択は、内部メカニズムを通じて行われます。`show service <service name>` and `show serviceGroup <service group name>` コマンドを使用すると、サービスを監視する所有者ノードを確認できます。

パス監視は、ノードとサーバーが提供するサービスとの間のネットワーク接続とリンク状態をチェックします。ノードは ICMP ping を送信して、サーバーにアクセスできるかどうかを確認します。

パス・モニタリングの仕組み

N1、N2、N3 の 3 つのノードで構成される NetScaler クラスターの例を考えてみましょう。N2 は HTTP サービス (S1) の状態を監視するサービスオーナーです。クラスター内の他のノードにサービスの状態をアドバタイズします。パス監視は、クラスター内のすべてのノード、すべてのサービスで有効になっています。各ノードは ICMP ping のみをサーバーに送信します。サービス所有者は HTTP サービスリクエストと ICMP ping の両方を送信します。各ノードは、そのパス監視状態をサービス所有者に報告します。



次の 2 つのパラメータは、ノードのサービス状態を決定します。

- S = サービス所有者が通知したサービスの状態
- P = 各ノードのパス監視状態

ノードがサーバーにアクセスできるかどうかによって、そのノードのパス監視状態が決まります。

次の表は、PathMonitorInDV パラメータが有効または無効の場合に、パス監視状態に基づいて設定されたサービス状態を示しています。

パラメーター	経路監視状態	サービス状態
PathMonitorInDV = いいえ。これがデフォルトの構成です。	P1 = ダウン	S1 = ダウン
	P2 = アップ	S1 = ダウン
	P3 = アップ	S1 = ダウン
pathMonitorIndv = YES	P1 = ダウン	S1 = ダウン
	P2 = アップ	S1 = アップ
	P3 = アップ	S1 = アップ

この例では、サービス所有者は、パス監視状態が DOWN に設定されているノードに基づいて、すべてのノードのサービス状態を決定します。いずれかのノードのパス監視状態が DOWN の場合、サービス所有者はすべてのノードのサービス状態を DOWN に設定します。各ノードのパス監視状態が UP の場合のみ、すべてのノードのサービス状態が UP に設定されます。

PathMonitorInDV パラメーターを有効にすることで、個々のノードのパス監視を使用できます。このパラメータにより、サービス所有者は各ノードのパス監視状態に基づいて各ノードのサービス状態を設定できます。

> 注
>
> PathMonitorInDV パラメータを設定すると、パーシスタンスなどの一部の機能が動作しなくなる可能性があります。

パス監視の設定

パスモニタリングは、すべてのサービスとサービスグループに適用されます。パス監視パラメータはデフォルトでは無効になっています。

CLI を使用してサービス/サービスグループのパス監視を有効にするには

コマンドプロンプトで入力します。

```
add service <service name> <IP address> <service type> <port> [-pathMonitor <YES | NO>] [-pathMonitorIndv <YES | NO>]

add servicegroup <servicegroup name> <service type> [-pathMonitor <YES | NO>] [-pathMonitorIndv <YES | NO>]
<!--NeedCopy-->
```

例:


```
add service s1 1.1.1.1 HTTP 80 -pathMonitor YES
add servicegroup sg_1 HTTP -pathMonitor YES

add service s1 1.1.1.1 HTTP 80 -pathMonitor YES -pathMonitorIndv YES
add servicegroup sg_1 HTTP -pathMonitor YES -pathMonitorIndv YES
<!--NeedCopy-->
```

次のように set コマンドからパス監視パラメータを設定することもできます。

```
set service <service name> [-pathMonitor <YES | NO>] [-pathMonitorIndv
    <YES | NO>]
set servicegroup <servicegroup name> [-pathMonitor <YES | NO>] [-
    pathMonitorIndv <YES | NO>]
<!--NeedCopy-->
```

例**:**

```
set service s1 -pathMonitor YES
set servicegroup sg_1 -pathMonitor YES


set service s1 -pathMonitorIndv YES
set servicegroup sg_1 -pathMonitorIndv NO
<!--NeedCopy-->
```

GUI を使用してサービス**/**サービスグループのパス監視を有効にするには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。

   サービスグループの場合は、[トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。

2. サービス**/**サービスグループペインで、リストからサービス/サービスグループを選択し、ダブルクリックして開きます。
3. [サービス設定] タブで、[編集] をクリックします。
4. 「パス・モニタリング」を選択します。
5. 適用する場合は「個別パス監視」を選択し、「**OK**」をクリックします。

> 注
>
> 個別パス監視を有効にできるのは、パス監視を有効にした場合のみです。

クラスターセットアップのバックアップと復元

August 15, 2023

NetScaler クラスターノードの現在の状態をバックアップできます。後で、バックアップしたファイルを使用してノードを同じクラスター状態に復元できます。予防措置として、クラスターノードでアップグレードを実行する前にこの機能を使用する必要があります。

クラスターセットアップをバックアップする

次の条件に応じて、基本バックアップまたは完全バックアップを取ることができます。

- バックアップするデータのタイプ。
- バックアップを作成する頻度。
- 基本的なバックアップ。設定ファイルのみをバックアップします。バックアップするファイルは常に変更するため、頻繁にこのタイプのバックアップの実行が必要になる場合があります。バックアップされたファイルが表に一覧表示されます。

ディレクトリ

サブディレクトリまたはファイル

/nsconfig/

- ns.conf
- ZebOS.conf
- rc.netscaler
- snmpd.conf
- nsbefore.sh
- nsafter.sh
- inetd.conf
- ntp.conf
- syslog.conf
- newsyslog.conf
- crontab
- host.conf
- hosts
- ttys
- sshd_config
- httpd.conf
- monitrc
- rc.conf
- ssh_config
- ローカルタイム
- issue

- issue.net

/var/

- download/*
- log/wicmd.log
- wi/tomcat/webapps/*
- wi/tomcat/logs/*
- wi/tomcat/conf/catalina/localhost/*
- nslw.bin/etc/krb.conf
- nslw.bin/etc/krb.keytab
- netscaler/locdb/*
- lib/likewise/db/*
- vpn/bookmark/*
- netscaler/crl
- nstemplates/*
- learnt_data/*

/netscaler/

- custom.html
- vsr.html
- フルバックアップ。基本バックアップでバックアップされるファイルとは別に、完全バックアップでは更新頻度の低いファイルがバックアップされます。フルバックアップオプションを使用したときにバックアップされるファイルは、表に記載されています。

ディレクトリ

サブディレクトリまたはファイル

/nsconfig/

- ssl/*
- license/*
- fips/*

/var/

- netscaler/ssl/*
- wi/java_home/jre/lib/security/cacerts/*
- wi/java_home/lib/security/cacerts/*

> 重要
>
> SDX クラスタ設定で CLAG が設定されている場合、バックアップと復元は機能しません。

バックアップは、圧縮された TAR ファイルとして /var/ns_sys_backup/ ディレクトリに保存されます。ディスクスペース不足による問題を避けるために、このディレクトリに格納できるバックアップファイルは最大 50 個となっています。rm system backup コマンドを使用して既存のバックアップファイルを削除すると、さらにバックアップを作成できます。

クラスタ設定の CLIP でバックアップ操作を実行すると、各クラスタノードにバックアップファイルが作成されます。

クラスターセットアップのバックアップ方法

NetScaler CLI を使用して CLIP のクラスターセットアップをバックアップすること。

コマンドプロンプトで、次の操作を行います。

- 構成を保存します。

  ```
  save ns config<!--NeedCopy-->
  ```

- バックアップファイル (基本またはフル) を作成します。

  “‘create system backup [][–level (basic | full)][-comment]

  ```
  1  ** 例 **
  2
  3  ```create system backup cluster-backup-1  - level basic<!--
       NeedCopy-->
  ```

上記のコマンドは、指定されたファイル名で各クラスターノードにバックアップ TAR ファイルを作成します。たとえば、Cluster-Backup-1.tgz ファイルは各クラスターノードで作成されます。

> 注
>
> ファイル名が指定されていない場合、バックアップ TAR ファイルは以下の命名規則に従って各クラスターノードに作成されます。
>
> - backup_<level>_<nsip_address of the cluster node 0>_<date-timestamp>.tgz<!--NeedCopy-->
> - backup_<level>_<nsip_address of the cluster node 1>_<date-timestamp>.tgz<!--NeedCopy-->
>
> たとえば、3 ノードのクラスタ構成では、
>
> - backup_<level>_<nsip_address of the cluster node 0>_<date-timestamp>.tgz<!--NeedCopy-->は ノード 0 に作成されます

> - backup _<level>_<nsip_address of the cluster node 1>_<date-timestamp>.tgz<!--NeedCopy-->はノード 1 に作成されます
> - backup_<level>_<nsip_address of the cluster node 2>_<date-timestamp>.tgz<!--NeedCopy-->はノード 2 に作成されます

- CLIP で作成したバックアップファイルを確認します。

  ```
  show system backup<!--NeedCopy-->
  ```

クラスターセットアップの復元

クラスタノードに障害が発生した場合、このノードを新しいノードと交換できます。障害のあるノードのバックアップファイルを使用して、クラスターの新しいノードを設定できます。

たとえば、3 ノードのクラスタ構成で、ノード 1 に障害が発生した場合、この障害のあるノードをノード 1 として新しいノードと交換できます。復元操作を使用すると、障害のあるノードのバックアップファイルの 1 つを新しいノードに復元できます。

> 注
>
> バックアップファイルの名前が変更されたり、ファイルの内容が変更されたりすると、復元操作は成功しません。

クラスターノードを復元する方法

CLI を使用してクラスターノードを復元するには　コマンドプロンプトで、次の操作を行います。

- CLIP で利用できるバックアップファイルのリストを入手してください。

  ```
  show system backup<!--NeedCopy-->
  ```

- バックアップ tar ファイルを、復元するクラスターノードの /var/ns_sys_backup ディレクトリにコピーします。
- クラスターノードで次のコマンドを実行して、バックアップ tar ファイルをクラスターノードのメモリに追加します。

  “‘add system backup

  ```
  1  ** 例 **
  2
  3  ```add system backup CLUSTER-BACKUP-1.tgz<!--NeedCopy-->
  ```

  > 注
  >
  > このコマンドは、復元するクラスタノードで実行する必要があります。

- バックアップファイルを指定してクラスターノードを復元します。

  “‘restore system backup

```
** 例 **

```restore system backup CLUSTER-BACKUP-1.tgz<!--NeedCopy-->
```

> 注
>
> このコマンドは、復元するクラスタノードで実行する必要があります。

- クラスターノードを再起動します。

  リブート

> 注
>
> このコマンドは、復元するクラスタノードで実行する必要があります。

NetScaler クラスターのアップグレードまたはダウングレード

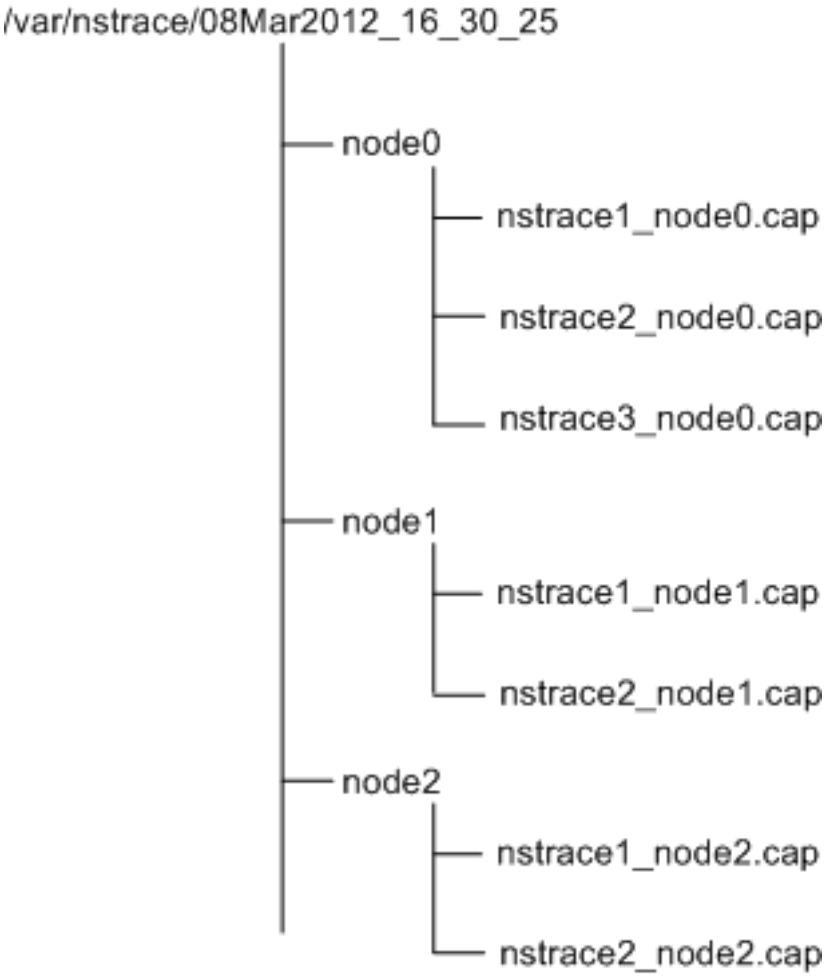
January 9, 2024

NetScaler クラスタのすべてのノードは、同じソフトウェアバージョンを実行している必要があります。したがって、クラスタをアップグレードまたはダウングレードするには、クラスタの各 NetScaler アプライアンスを一度に 1 ノードずつアップグレードまたはダウングレードする必要があります。

アップグレードまたはダウングレード中のノードは、クラスタから削除されません。ノードはクラスタの一部であり続け、アップグレードまたはダウングレード後にノードが再起動するときのダウンタイムを除いて、中断することなくトラフィックを処理します。

最初のクラスターノードをアップグレードまたはダウングレードすると、クラスターノード間でクラスターバージョンの不一致があると、構成の伝達が自動的に無効になります。構成の伝達は、すべてのクラスターノードをアップグレードまたはダウングレードした後にのみ有効になります。構成の伝播が無効になっている場合、クラスタ IP アドレスを使用して構成を実行することはできません。

次の表は、最初のクラスターノードのアップグレードまたはダウングレード中に構成の伝達が無効になる場合を示しています。


```

| アップグレード/ダウングレード | バージョンから | バージョンへ | 説明 |
| --- | --- | --- | --- |
| アップグレード | 13.1 ビルド 21.50 またはそれ以前のバージョン | 13.1 ビルド 24.38 またはそれ以降 | クラスターノードが再起動して起動すると、クラスターの構成伝達は無効になります。注**:** ノードの再起動中は、クラスタ IP アドレスを使用して設定を実行しないでください。 |
| アップグレード | 13.1 ビルド 24.38 またはそれ以降 | 13.1 ビルド 27.59 またはそれ以降 | ノードが再起動する直前に、クラスターの構成伝達は無効になります。 |
| ダウングレード | 13.1 ビルド 24.38 またはそれ以降 | 13.1 ビルド 21.50 またはそれ以前のバージョン | クラスターノードが再起動して起動すると、クラスターの構成伝達は無効になります。注**:** ノードの再起動中は、クラスタ IP アドレスを使用して設定を実行しないでください。 |
| ダウングレード | 13.1 ビルド 27.59 またはそれ以降 | 13.1 ビルド 24.38 またはそれ以降 | ノードが再起動する直前に、クラスターの構成伝達は無効になります。 |

次のコマンドを使用してコマンド伝播のステータスを確認することもできます:

```
show cluster instance
<!--NeedCopy-->
```

注:

- 最大接続 (MaxConn) グローバルパラメータがゼロ以外の値に設定されているクラスタセットアップでは、次の条件のいずれかが満たされると CLIP 接続が失敗することがあります。
  - Upgrading the setup from NetScaler 13.0 76.x build to NetScaler 13.0 79.x build.
  - Restarting the CCO node in a cluster setup running NetScaler 13.0 76.x build.

  回避方法:
  - クラスターセットアップを NetScaler 13.0 76.x ビルドから NetScaler 13.0 79.x ビルドにアップグレードする前に、最大接続（MaxConn）グローバルパラメーターをゼロに設定する必要があります。セットアップをアップグレードした後、maxConn パラメータを目的の値に設定し、構成を保存できます。

- NetScaler 13.0 76.x ビルドはクラスタセットアップには適していません。NetScaler 13.0 76.x ビルドをクラスターセットアップに使用しないことをお勧めします。

- クラスター設定では、NetScaler アプライアンスが次の場合にクラッシュすることがあります。
  - upgrading the setup from NetScaler 13.0 47.x or 13.0 52.x build to a later build, or
  - upgrading the setup to NetScaler 13.0 47.x or 13.0 52.x build

回避策：アップグレードプロセス中に、次の手順を実行します。

- すべてのクラスタノードを無効にし、各クラスタノードをアップグレードします。
- すべてのノードのアップグレード後に、すべてのクラスタノードを有効にします。

クラスタをアップグレードまたはダウングレードする前に注意すべきポイント

- 重要：

  アップグレードの変更とカスタマイズの両方を、アップグレードされた NetScaler ADC アプライアンスに適用することが重要です。そのため、`/etc`ディレクトリにカスタマイズした構成ファイルがある場合は、アップグレードを続行する前に、「カスタマイズされた構成ファイルのアップグレードの考慮事項」を参照してください。

- クラスタソフトウェアバージョンのアップグレード中またはダウングレード中は、クラスタノードを追加できません。
- ノードレベルの構成は、個々のノードの NSIP アドレスを使用して実行できます。同期を維持するために、すべてのノードで同じ構成を実行してください。
- クラスターのアップグレード中は、クラスター IP アドレスから`start nstrace`コマンドを実行できません。ただし、NSIP アドレスを使用して個々のクラスターノードでこの操作を実行することにより、個々のノードのトレースを取得できます。
- NetScaler 13.0 76.x ビルドは、クラスターのセットアップには適していません。NetScaler 13.0 76.x ビルドをクラスターセットアップに使用しないことをお勧めします。
- NetScaler 13.0 47.x および 13.0 52.x ビルドは、クラスターセットアップには適していません。これは、これらのビルドではノード間通信に互換性がないためです。
- クラスタのアップグレード中に、アップグレードされたノードで、まだアップグレードされていないノードでは使用できない追加機能がアクティブになっている可能性があります。その結果、クラスタのアップグレード中にライセンス不一致の警告が表示されます。この警告は、すべてのクラスタノードがアップグレードされると自動的に解決されます。

重要

- 前のノードがアクティブになるのを待ってから、次のノードをアップグレードまたはダウングレードすることをお勧めします。

> - クラスター IP セッションが複数回切断されないように、クラスター構成ノードを最後にアップグレード/ダウングレードすることをお勧めします。

クラスタノードのソフトウェアをアップグレードまたはダウングレードするには

1. クラスターが安定していて、構成がすべてのノードで同期されていることを確認してください。
2. NSIP アドレスを介して各ノードにアクセスし、以下を実行します:
   - クラスタノードをアップグレードまたはダウングレードします。アプライアンスのソフトウェアのアップグレードとダウングレードの詳細については、「NetScaler アプライアンスのアップグレードとダウングレード」を参照してください。
   - 構成を保存します。
   - アプライアンスを再起動します。
3. 他のクラスタノードごとに手順 2 を繰り返します。

個々のクラスターノードでサポートされる操作

August 15, 2023

通常、クラスターの一部である NetScaler アプライアンスは、NSIP アドレスから個別に構成することはできません。ただし、この規則の例外となる操作もあります。これらの操作は、NSIP アドレスから実行された場合、他のクラスターノードには伝播されません。

操作は以下のとおりです。

cluster instance (set	rm	有効にします	無効)

-

cluster node (set rm)

-

ns trace (start	show	stop)

-

interface (set	有効にします	無効)

-

route (add	rm	set	未設定)

-

ARP (add	rm	send -all)

-
- force cluster sync
- sync cluster files
- NTP 同期を無効にする
- save ns config
- リブート
- shutdown

たとえば、クラスタノードの NSIP アドレスからコマンド`disable interface 1/1/1`を実行すると、インターフェイスはそのノードでのみ無効になります。このコマンドは伝播されないため、インターフェイス 1/1/1 は他のすべてのクラスタノードで有効のままです。

異種クラスターのサポート

August 15, 2023

NetScaler アプライアンスは、クラスター展開における異種クラスターをサポートします。異機種クラスタは異なる NetScaler ハードウェアのノードにまたがるため、同じクラスタに異なるプラットフォームを組み合わせることができます。

> 重要
>
> 異種クラスタの形成またはサポートが可能であり、MPX ハードウェアプラットフォームのみに制限されます。

異機種クラスタのサポート性と構成は、特定の NetScaler モデルによって異なります。次の表は、同じ数のパケットエンジンを備えた異種クラスタの形成でサポートされるプラットフォームを示しています。

パケットエンジン数	MPX ハードウェアプラットフォーム	異機種クラスタの形成をサポートする MPX ハードウェアプラットフォーム
5	MPX 11500	MPX 14020
7	MPX 11515	MPX 14040
9	MPX 11530	MPX 14060

次の表は、パケットエンジンの数が異なる異種クラスタの形成でサポートされるプラットフォームを示しています。

ハードウェア・プラットフォーム	異機種混在クラスタを構成するためのサポート対象ハードウェアプラットフォーム
MPX 150XX	MPX 140XX

異なる SSL チップセットにまたがるパケットエンジン数の異なる NetScaler ADC MPX アプライアンスの異種クラスタ展開を形成する方法の詳細については、SSL オフロード構成の 「異機種間クラスタの展開」セクションを参照してください。

> （注）
>
> リリース 13.0 ビルド 47.x より前では、パケットエンジンの数が異なるノードから「joincluster」コマンドを実行すると、「CCO とローカルノード間のアクティブな PPE の数が一致しません」というエラーメッセージが表示されます。

注意事項

1. 追加の管理 CPU 設定は、すべてのクラスターノードで同じでなければなりません。
2. 新しく追加するノードのデータプレーンとバックプレーンの容量は、既存のクラスタノードと同じである必要があります。
3. 異なる暗号をサポートするプラットフォームが混在している場合、クラスタは共通の暗号リストに同意します。

よくある質問

December 8, 2023

クラスタリングに関するよくある質問の一覧です。

1 つの NetScaler クラスターに含めることができる NetScaler アプライアンスはいくつですか?

NetScaler クラスタには、1 つのアプライアンス、または最大 32 台の NetScaler nCore ハードウェアまたは仮想アプライアンスを含めることができます。これらのノードは、クラスタノードの前提条件で指定された基準を満たす必要があります。

NetScaler アプライアンスを複数のクラスターの一部にすることはできますか?

いいえ。NetScaler アプライアンスは 1 つのクラスターにのみ属できます。

クラスタ IP アドレスとは何ですか? そのサブネットマスクとは何ですか?

クラスター IP アドレスは、NetScaler クラスターの管理アドレスです。すべてのクラスター構成は、このアドレスからクラスターにアクセスして実行する必要があります。クラスタ IP アドレスのサブネットマスクは 255.255.255.255 に固定されています。

特定のクラスターノードをクラスター構成コーディネーターにする方法を教えてください

特定のノードをクラスター構成コーディネーターとして手動で設定するには、そのノードの優先度を最小の数値 (最も高い優先度) に設定する必要があります。理解するために、以下の優先順位を持つ 3 つのノードがあるクラスターを考えてみましょう。

n1 - 29, n2 - 30, n3 - 31

ここで、n1 は構成コーディネーターです。n2 を構成コーディネーターにする場合は、その優先順位を n1 より小さい値、たとえば 28 に設定する必要があります。設定を保存すると、n2 が設定コーディネーターになります。

> 注
>
> n1 がダウンすると、元の優先度値が 30 の n2 が設定コーディネーターになります。構成コーディネーターがダウンした場合に備えて、次に優先順位の値が小さいノードが選択されます。

クラスタのネットワークインターフェースが通常の 2 タプル (u/c) 表記ではなく 3 タプル (n/u/c) 表記で表されるのはなぜですか?

NetScaler アプライアンスがクラスターの一部である場合、インターフェイスが属するノードを識別できる必要があります。そのため、クラスターノードのネットワークインターフェース命名規則が u/c から n/u/c に変更されました。ここで n はノード ID を示します。

クラスタノードのホスト名を設定する方法を教えてください

クラスタノードのホスト名は、クラスタ IP アドレスを使用して **set ns hostname** コマンドを実行して指定する必要があります。たとえば、ID 2 のクラスターノードのホスト名を設定するには、コマンドは次のようになります。

ns ホスト名ホスト名を設定します。ホスト名 1-オーナーノード 2

NetScaler アプライアンスを自動的に検出してクラスタに追加できますか**?**

はい。構成ユーティリティを使用すると、構成コーディネータの NSIP アドレスと同じサブネットに存在するアプライアンスを検出できます。詳細については、「NetScaler アプライアンスの検出」を参照してください。

ノードが削除されたり、無効になったり、再起動またはシャットダウンされたり、非アクティブになったりすると、クラスタのトラフィック処理機能が影響を受けますか

はい。これらの操作のいずれかをクラスターのアクティブノードで実行すると、クラスターのトラフィックを処理するノードが 1 つ少なくなります。また、このノード上の既存の接続は終了します。

スタンドアロンアプライアンスが複数あり、それぞれ構成が異なります。これらを **1** つのクラスターに追加できますか**?**

はい。異なる構成のアプライアンスを単一のクラスタに追加できます。ただし、アプライアンスをクラスターに追加すると、既存の構成はクリアされます。個々のアプライアンスで使用可能な構成を使用するには、次のことを行う必要があります。

1. すべての構成用に 1 つの *.conf ファイルを作成します。
2. 構成ファイルを編集して、クラスター環境でサポートされていない機能を削除します。
3. インタフェースの命名規則を 2 タプル (u/c) 形式から 3 タプル (n/u/c) 形式に更新します。
4. batch コマンドを使用して、構成をクラスターの構成コーディネーターノードに適用します。

スタンドアロンの **NetScaler** アプライアンスまたは **HA** 設定の構成をクラスター構成に移行できますか**?**

いいえ。クラスタ化されたセットアップにノードを追加すると、**clear ns config** コマンド (拡張オプション付き) を使用してその構成が暗黙的にクリアされます。さらに、SNIP アドレスとすべての VLAN 設定（デフォルト VLAN と NSVLAN を除く）がクリアされます。そのため、アプライアンスをクラスタに追加する前に構成をバックアップすることをお勧めします。クラスターのバックアップ構成ファイルを使用する前に、次のことを行う必要があります。

1. 構成ファイルを編集して、クラスター環境でサポートされていない機能を削除します。
2. インタフェースの命名規則を 2 タプル (x/y) 形式から 3 タプル (x/y/z) 形式に更新します。
3. **batch** コマンドを使用して、構成をクラスターの構成コーディネーターノードに適用します。

バックプレーンインターフェイスは **L3 VLAN** の一部ですか

はい。デフォルトでは、バックプレーンインターフェイスはクラスタに設定されているすべての L3 VLAN に存在します。

異なるネットワークのノードを含むクラスターを構成する方法を教えてください

> 注
>
> NetScaler 11.0 以降でサポートされています。

異なるネットワークのノードを含むクラスタは、L3 クラスタと呼ばれます (INC モードではクラスタと呼ばれることもあります)。L3 クラスタでは、1 つのネットワークに属するすべてのノードを 1 つのノードグループにグループ化する必要があります。したがって、クラスタにそれぞれ 3 つの異なるネットワークの 2 つのノードが含まれている場合は、3 つのノードグループ (ネットワークごとに 1 つ) を作成し、これらのノードグループのそれぞれをそのネットワークに属するノードに関連付ける必要があります。構成情報については、クラスターをセットアップする手順を参照してください。

クラスタ上の **NSVLAN** を設定/設定解除する方法を教えてください

次のいずれかを実行します。

- NSVLAN をクラスタで使用できるようにするには、クラスタに追加する前に、各アプライアンスに同じ NSVLAN が設定されていることを確認します。
- クラスタノードから NSVLAN を削除するには、まずクラスタからノードを削除し、次に NSVLAN をアプライアンスから削除します。

一部の **NetScaler** ノードが外部ネットワークに接続されていないクラスターを設定しています。クラスターはまだ正常に機能していますか**?**

はい。クラスターはリンクセットと呼ばれるメカニズムをサポートしています。これにより、接続されていないノードは、接続されたノードのインターフェースを使用してトラフィックを処理できます。接続されていないノードは、クラスタバックプレーンを介して接続されたノードと通信します。詳細については、「リンクセットの使用」を参照してください。

MAC ベースフォワーディング **(MBF)** を必要とする展開は、クラスタ化されたセットアップでどのようにサポートできますか**?**

MBF を使用する展開では、リンクセットを使用する必要があります。詳細については、「リンクセットの使用」を参照してください。

クラスタノードの **NSIP** アドレスからコマンドを実行できますか

いいえ。NSIP アドレスによる個々のクラスタノードへのアクセスは読み取り専用です。そのため、クラスターノードの NSIP アドレスにログオンすると、構成と統計しか表示できません。何も設定できません。ただし、クラスターノードの NSIP アドレスから実行できる操作がいくつかあります。詳細については、個々のノードでサポートされる操作を参照してください。

クラスタ・ノード間の構成の伝播を無効にできますか**?**

いいえ。クラスタノード間のクラスタ構成の伝播を明示的に無効にすることはできません。ただし、ソフトウェアのアップグレードまたはダウングレード中に、バージョンの不一致エラーが発生すると、設定の伝達が自動的に無効になることがあります。

NetScaler アプライアンスがクラスタの一部であるときに、**NSIP** アドレスを変更したり、**NSVLAN** を変更したりできますか**?**

いいえ。このような変更を行うには、まずクラスタからアプライアンスを削除し、変更を実行してから、クラスタにアプライアンスを追加する必要があります。

NetScaler クラスターは **L2** および **L3 VLAN** をサポートしていますか**?**

はい。クラスタは、クラスタノード間の VLAN をサポートします。VLAN はクラスタ IP アドレスで設定する必要があります。

- **L2 VLAN**。クラスタのさまざまなノードに属するインターフェイスをバインドすることで、レイヤ 2 VLAN を作成できます。
- **L3 VLAN** クラスタの異なるノードに属する IP アドレスをバインドすることで、レイヤ 3 VLAN を作成できます。IP アドレスは同じサブネットに属している必要があります。次のいずれかの基準が満たされていることを確認してください。そうしないと、L3 VLAN バインディングが失敗する可能性があります。
  - すべてのノードの IP アドレスは、VLAN にバインドされているものと同じサブネットにあります。
  - クラスタにはストライプされた IP アドレスがあり、その IP アドレスのサブネットは VLAN にバインドされています。

スポットされた IP のみを持つクラスターにノードを追加すると、スポットされた IP アドレスがそのノードに割り当てられる前に同期が行われます。このような場合、L3 VLAN バインディングが失われる可能性があります。この損失を避けるには、ストライプ IP を追加するか、新しく追加したノードの NSIP に L3 VLAN バインディングを追加してください。

NetScaler クラスターで **SNMP** を構成する方法を教えてください

SNMP は、スタンドアロンアプライアンスを監視するのと同じ方法で、クラスターとクラスターのすべてのノードを監視します。唯一の違いは、クラスタ上の SNMP はクラスタ IP アドレスを使用して設定する必要があることです。ハードウェア固有のトラップを生成する場合、クラスターのノードを識別するために、ノード ID とノードの NSIP アドレスという 2 つの varbind が追加されます。

クラスタ関連の問題についてテクニカルサポートに連絡する場合、どのような詳細情報を入手する必要がありますか**?**

NetScaler アプライアンスには、すべてのクラスターノードの構成データ、統計情報、およびログを抽出する **show techsupport-scope cluster** コマンドが用意されています。このコマンドをクラスター IP アドレスで実行します。

このコマンドの出力は、<nsip_CCO><date-timestamp> 設定コーディネーターの **/var/tmp/support/cluster/* ディレクトリにある *collector_cluster_ _P_ .tar.gz** という名前のファイルに保存されます。

このアーカイブをテクニカルサポートチームに送って、問題をデバッグしてください。

ストライプされた **IP** アドレスをサーバーのデフォルトゲートウェイとして使用できますか**?**

クラスター展開では、サーバーのデフォルトゲートウェイがストライプ IP アドレスを指していることを確認します (NetScaler が所有する IP アドレスを使用している場合)。たとえば、USIP が有効になっている LB デプロイの場合、デフォルトゲートウェイはストライプされた SNIP アドレスでなければなりません。

クラスタ **IP** アドレスから特定のクラスタノードのルーティング設定を表示できますか**?**

はい。VTYSH シェルに入るときに所有者ノードを指定することで、ノード固有の構成を表示およびクリアできます。

たとえば、ノード 0 と 1 のコマンドの出力を表示するには、コマンドは次のようになります。

```
\> vtysh
ns# owner-node 0 1
ns(node-0 1)\# show cluster state
ns(node-0 1)\# exit-cluster-node
ns\#
```

LACP システムプライオリティを設定するノードはどのように指定できますか**?**

> 注
>
> NetScaler 10.1 以降でサポートされています。

クラスターでは、**set lacp** コマンドを使用してそのノードを所有者ノードとして設定する必要があります。

例:ID が 2 のノードに LACP システム優先度を設定するには:

```
set lacp -sysPriority 5 -ownerNode 2<!--NeedCopy-->
```

IP トンネルはクラスタ設定でどのように設定されますか?

> 注
>
> NetScaler 10.1 以降でサポートされています。

クラスタでの IP トンネルの設定は、スタンドアロンアプライアンスの設定と同じです。唯一の違いは、クラスタ設定では、ローカル IP アドレスはストライプされた SNIP アドレスでなければならないことです。

NetScaler クラスタのノードにフェイルオーバーインターフェイスセット（FIS）を追加する方法を教えてください

> 注
>
> NetScaler 10.5 以降でサポートされています。

クラスタ IP アドレスで、次のコマンドを使用して、FIS を追加する必要があるクラスタノードの ID を指定します。

```
add fis <name> -ownerNode <nodeId>
```

> メモ
>
> - 各クラスタノードの FIS 名は一意でなければなりません。
> - クラスタ LA チャネルを FIS に追加できます。クラスタ LA チャネルに、メンバーインターフェイスとしてローカルインターフェイスがあることを確認してください。

FIS の詳細については、「フェールオーバーインターフェイスセットの設定」を参照してください。

クラスタ設定でのネットプロファイルの構成方法

> 注
>
> NetScaler 10.5 以降でサポートされています。

スポットされた IP アドレスをネットプロファイルにバインドできます。このネットプロファイルは、スポット負荷分散仮想サーバーまたはサービス（ノードグループを使用して定義される）にバインドできます。次の推奨事項に従う必要があります。そうしないと、ネットプロファイル構成が反映されず、USIP/USNIP 設定が使用されます。

> 注
>
> - ノードグループの **strict** パラメータが **Yes** に設定されている場合、ネットプロファイルには各ノードグループメンバーの IP アドレスが少なくとも 1 つ含まれている必要があります。
> - ノードグループの **strict** パラメータが **No** に設定されている場合、ネットプロファイルには各クラスタノードの IP アドレスが少なくとも 1 つ含まれている必要があります。

クラスタ設定で **WiONNS** を設定する方法を教えてください

> 注
>
> NetScaler 11.0 ビルド 62.x 以降でサポートされています。

WiONNS をクラスタで使用するには、以下を実行する必要があります。

1. Java パッケージと WI パッケージがすべてのクラスタノードの同じディレクトリにあることを確認します。
2. 永続性が構成された負荷分散仮想サーバーを作成します。
3. WI トラフィックを処理する各クラスターノードの NSIP アドレスとして IP アドレスを持つサービスを作成します。このステップは、NetScaler CLI を使用してのみ構成できます。
4. サービスを負荷分散仮想サーバーにバインドします。

> 注
>
> VPN 接続で WiONNS を使用している場合は、負荷分散仮想サーバーが WIHOME に設定されていることを確認してください。

クラスタ **LA** チャネルを管理アクセスに使用できますか**?**

いいえ。クラスタノードへの管理アクセスは、クラスタ LA チャネル (CLA/1 など) またはそのメンバーインターフェイスでは設定しないでください。これは、ノードが非アクティブになると、対応するクラスタ LA インターフェイスがパワーダウンとマークされ、管理アクセスできなくなるためです。

クラスタノード同士の通信方法と、バックプレーンを通過するトラフィックの種類を教えてください

バックプレーンは、各ノードの 1 つのインターフェイスがクラスタバックプレーンスイッチと呼ばれる共通のスイッチに接続されているインターフェイスのセットです。ノード間通信で使用されるバックプレーンを通過するトラフィックには、次のような種類があります。

- ノード間メッセージング (NNM)
- ステアリングトラフィック
- 設定の伝播と同期

クラスタの各ノードは、特別な MAC クラスタバックプレーンスイッチアドレスを使用して、バックプレーンを介して他のノードと通信します。クラスタスペシャル MAC の形式は **0x02 0x00 0x6F**<cluster _id> <node_id> です。<reserved> ここで、<cluster_id> はクラスタインスタンス ID です。<node_id> は、クラスターに追加される NetScaler アプライアンスのノード番号です。

> 注
>
> バックプレーンで処理されるトラフィック量の CPU オーバーヘッドはごくわずかです。

レイヤ **3** クラスタの **GRE** トンネルを介してルーティングされるものは何ですか**?**

誘導されたデータトラフィックのみが GRE トンネルを通過します。パケットは GRE トンネルを経由して他のサブネット上のノードに送られます。

ノード間メッセージング **(NNM)** メッセージとハートビートメッセージはどのように交換され、どのようにルーティングされますか**?**

NNM、ハートビートメッセージ、およびクラスタプロトコルはノンステアリングトラフィックです。これらのメッセージはトンネル経由で送信されませんが、直接ルーティングされます。

レイヤー **3** クラスタトンネリングトラフィックの **MTU** の推奨事項を教えてください

GRE トンネル経由のジャンボ MTU のレイヤ 3 クラスタの推奨事項は次のとおりです。

- ジャンボ MTU は、GRE トンネルのオーバーヘッドに対応するため、L3 パス上のクラスタノード間で設定できます。
- フルサイズのパケットでは、フラグメンテーションは発生しません。パケットは処理する必要があります。
- ジャンボフレームが許可されなくてもトラフィックのステアリングは機能し続けますが、フラグメンテーションによりオーバーヘッドが大きくなります。

グローバルハッシュキーはどのように生成され、すべてのノードで共有されるのですか**?**

スタンドアロンアプライアンス用の`rsskey`は、起動時に生成されます。クラスタ設定では、最初のノードがクラスタの`rsskey`を保持します。新しいノードがクラスタに参加するたびに、`rsskey`が同期されます。

***: ***、**USIP** のオン、**useproxyport** オフ、トポロジーの **set rsskeytype -rsskey symmetric** コマンドは何が必要ですか

これはクラスターに固有のものではなく、スタンドアロンアプライアンスにも適用されます。USIP をオンにし、プロキシポートの使用を無効にすると、シンメトリック`rsskey`はコア間（C2C）ステアリングとノード間ステアリングの両方を減らします。

CCO ノードを変更する要因にはどのようなものがありますか?

クラスタセットアップを形成するために追加された最初のノードが、構成コーディネーター (CCO) ノードになります。クラスタ設定の CCO ノードの変更には、次の要因が関係します。

- 現在の CCO ノードがクラスタ設定から削除されたとき
- 現在の CCO ノードがクラッシュしたとき
- CCO 以外のノードの優先度が変更された場合 (優先度が低いほど優先順位が高くなります)
- ノード間のネットワーク接続性などの動的な条件では
- ノードの状態が変化した場合 (アクティブ、スペア、パッシブ)。CCO としてはアクティブノードが推奨されます。
- 構成に変更があり、最新の構成のノードが CCO として優先される場合。

異なるハードウェアプラットフォームのノードでクラスタを形成できますか?

はい。NetScaler アプライアンスは異種クラスターをサポートします。異種クラスタでは、同じクラスタ内に異なるプラットフォームを組み合わせることができます。詳細については、「異種クラスタのサポート」を参照してください。

NetScaler ADC クラスターのトラブルシューティング

August 15, 2023

NetScaler クラスターで障害が発生した場合、トラブルシューティングの最初のステップは、クラスターインスタンスに関する情報を取得することです。この情報は、クラスタノードで`show cluster instance clId`と`show cluster node nodeId`コマンドをそれぞれ実行することで取得できます。

上記の 2 つの方法を使用しても問題が見つからない場合は、次のいずれかを使用できます。

- 障害の原因を特定します。クラスターをバイパスしてサーバーにアクセスしてみてください。試行が成功した場合、問題はおそらくクラスターのセットアップにあります。
- 最近実行されたコマンドを確認してください。 history コマンドを実行して、クラスターで実行された最近の構成を確認します。ns.conf ファイルを確認して、実装された構成を確認することもできます。
- **ns.log** ファイルを確認してください。各ノードの /var/log/ ディレクトリにあるログファイルを使用して、実行されたコマンド、コマンドのステータス、および状態の変化を確認します。
- ニュースログファイルをチェックしてください。各ノードの /var/nslog/ `newnslog` ディレクトリにあるファイルを使用して、クラスターノードで発生したイベントを特定します。ファイルを単一のディレクトリにコピーしてから次のコマンドを実行することにより、複数の `newnslog` ファイルを単一のファイルとして表示できます。


```
nsconmsg -K newnslog-node<id> -K     newnslog.node<id> -d current
```

それでも問題を解決できない場合は、クラスター上のパケットをトレースするか、show techsupport -scope cluster コマンドを使用してみてください。コマンドを使用して、レポートをテクニカルサポートチームに送信できます。

NetScaler ADC クラスターのパケットのトレース

August 15, 2023

NetScaler オペレーティングシステムには、アプライアンスで送受信されるパケットのダンプを取得するための *ns trace* というユーティリティが用意されています。このユーティリティは、パケットをトレースファイルに格納します。これらのファイルを使用して、クラスターノードへのパケットフローの問題をデバッグできます。トレースファイルは Wireshark アプリケーションで表示する必要があります。

ns trace ユーティリティの重要な側面は次のとおりです。

- 従来の式とデフォルトの式を使用して、パケットを選択的にトレースするように設定できます。
- ns トレース形式 (.cap) と TCP ダンプ形式 (.pcap) の複数の形式でトレースをキャプチャできます。
- 構成コーディネーター上のすべてのクラスターノードのトレースファイルを集約できます。
- 複数のトレースファイルを 1 つのトレースファイルにマージできます (.cap ファイルのみ)。

ns トレースユーティリティは、NetScaler コマンドラインまたは NetScaler シェルから使用できます。

スタンドアロンアプライアンスのパケットをトレースするには

アプライアンスで start ns trace コマンドを実行します。<date-timestamp> このコマンドは、/var/nstrace/ ディレクトリにトレースファイルを作成します。<id> トレースファイルの名前は nstrace.cap という形式です。

show ns trace コマンドを実行すると、ステータスを表示できます。stop ns trace コマンドを実行すると、パケットのトレースを停止できます。

> 注
>
> nstrace.sh ファイルを実行して、NetScaler シェルから ns トレースユーティリティを実行することもできます。ただし、nstrace ユーティリティは NetScaler コマンドラインインターフェイスから使用することをお勧めします。

クラスタのパケットをトレースするには

すべてのクラスターノードでパケットをトレースし、構成コーディネーターのすべてのトレースファイルを取得できます。

クラスタ IP アドレスで start ns trace コマンドを実行します。コマンドは伝播され、すべてのクラスターノードで実行されます。トレースファイルは、<date-timestamp> /var/nstrace/ ディレクトリの個々のクラスターノードに保存されます。<id> トレースファイルの名前は nstrace\<id> _node .cap という形式です。

各ノードのトレースファイルを使用して、ノードの操作をデバッグできます。ただし、すべてのクラスターノードのトレースファイルを 1 か所にまとめる場合は、クラスタ IP アドレスで stop ns trace コマンドを実行する必要があります。すべてのノードのトレースファイルが、以下のように<date-timestamp> /var/nstrace/ ディレクトリのクラスター構成コーディネーターにダウンロードされます。



複数のトレースファイルをマージする

トレースファイルから 1 つのファイルを作成できます (のみサポート)。Cap ファイル (クラスタノードから取得したファイル)。シングルトレースファイルから、クラスタパケットのトレースを累積的に表示できます。単一トレースファイル内のトレースエントリは、パケットがクラスターで受信された時間に基づいてソートされます。

トレースファイルをマージするには、NetScaler ADC シェルで次のように入力します。

```
1 > nstracemerge.sh -srcdir \<DIR\> -dstdir \<DIR\> -filename \<name\> -
    filesize \<num\>
```


各項目の意味は次のとおりです。

- srcdir トレースファイルのマージ元のディレクトリです。このディレクトリ内のすべてのトレースファイルが 1 つのファイルにマージされます。
- dstdir は、マージされたトレースファイルが作成されるディレクトリです。
- Filename は、作成されるトレースファイルの名前です。
- Filesize トレースファイルのサイズです。

例

ns trace ユーティリティを使用してパケットをフィルタリングする例をいくつか示します。

- 3 つのノードのバックプレーンインターフェイス上のパケットをトレースするには、次の手順を実行します。

  クラシックな表現を使う**:**

```
1   > start nstrace -filter "INTF == 0/1/1 && INTF == 1/1/1 && INTF
      == 2/1/1"
```

  既定の表現の使用**:**

```
1   > start nstrace -filter "CONNECTION.INTF.EQ("0/1/1") &&
      CONNECTION.INTF.EQ("1/1/1") && CONNECTION.INTF.EQ("2/1/1")"
```

- 送信元 IP アドレス 10.102.34.201、または送信元ポートが 80 より大きく、サービス名が「s1」ではないシステムからのパケットを追跡するには:

  クラシックな表現を使う

```
1   > start nstrace -filter \"SOURCEIP == 10.102.34.201 \|| \(SVCNAME
      != s1 && SOURCEPORT > 80)\"
```

  既定のエクスプレッションの使用

```
1   > start nstrace -filter \"CONNECTION.SRCIP.EQ\(10.102.34.201) \||
      \(CONNECTION.SVCNAME.NE\(\"s1\") && CONNECTION.SRCPORT.GT
    \(80))\"
```

> 注
>
> ns トレースで使用されるフィルタの詳細については、ns トレースを参照してください。

トレース中の **SSL** セッションキーのキャプチャ

「start ns trace」コマンドを実行すると、capsslkeys すべての SSL セッションの SSL マスターキーをキャプチャする新しいパラメータを設定できます。このパラメータを含めると、nstrace.sslkeys という名前のファイルが

パケットトレースとともに生成されます。このファイルを Wireshark にインポートして、対応するトレースファイル内の SSL トラフィックを復号化できます。

この機能は、後で Wireshark にインポートして SSL トラフィックを復号化できるセッションキーをエクスポートする Web ブラウザに似ています。

SSL セッションキーを使用するメリット

SSL セッションキーを使用する利点は次のとおりです。

1. SSLPLAIN モードのキャプチャで作成された余分なパケットを含まない、より小さなトレースファイルを生成します。
2. トレースからプレーンテキスト [SP (1)] を表示し、マスターキーファイルを共有するか、機密データを共有しないことで保護するかを選択できます。

SSL セッションキーを使用する場合の制限事項

SSL セッションキーを使用する場合の制限は次のとおりです。

1. セッションの最初のパケットがキャプチャされない場合、SSL セッションは復号化できません。
2. 連邦情報処理標準 (FIPS) モードが有効になっている場合、SSL セッションはキャプチャできません。

コマンドラインインターフェイス (CLI) を使用して SSL セッションキーをキャプチャするには

コマンドプロンプトで次のコマンドを入力して、トレースファイルの SSL セッションキーを有効または無効にし、トレース操作を確認します。

```
> start nstrace -capsslkeys ENABLED
> show nstrace
Example
> start nstrace -capsslkeys ENABLED
> show nstrace
      State:  RUNNING          Scope:  LOCAL            TraceLocation:
          "/var/nstrace/04May2016_17_51_54/..."
      Nf:  24                  Time:  3600              Size:  164
                        Mode:  TXB NEW_RX
      Traceformat:  NSCAP      PerNIC:  DISABLED        FileName:  04
         May2016_17_51_54 Link:  DISABLED
      Merge:  ONSTOP           Doruntimecleanup:  ENABLED TraceBuffers:
           5000      SkipRPC:  DISABLED
      SkipLocalSSH:  DISABLED  Capsslkeys:  ENABLED     InMemoryTrace:
          DISABLED
 Done
```


NetScaler GUI を使用して **SSL** セッションキーを構成するには

1. [構成**] > [**システム**] > [**診断**] > [**テクニカルサポートツール] に移動し、[新規トレースの開始] をクリックして、アプライアンス上の暗号化されたパケットのトレースを開始します。
2.「トレースの開始」ページで、「**SSL** マスターキーのキャプチャ」チェックボックスを選択します。
3. [**OK**] をクリックし、[完了] をクリックします。

SSL マスターキーを **Wireshark** にインポートするには

Wireshark GUI で、[編集**] > [**設定**] > [**プロトコル**] > [SSL] > (Pre)-Master-Secret** ログファイル名に移動し、アプライアンスから取得したマスターキーファイルを指定します。

よくある問題のトラブルシューティング

August 15, 2023

ノードをクラスターに参加させるときに、次のメッセージが表示されます。「エラー**:** インターフェイスの名前**/**番号が無効です。このエラーを解決するにはどうすればいいですか**?**

このエラーは、add cluster node コマンドを使用してノードを追加する際に、無効または間違ったバックプレーンインターフェイスを指定した場合に発生します。このエラーを解決するには、ノードの追加時に指定したインターフェイスを確認してください。アプライアンスの管理インターフェイスをバックプレーンインターフェイスとして指定していないこと、およびインターフェイスの\ <nodeId\ > ビットがノードの ID と同じであることを確認してください。<c\ > <u\ > たとえば、ノード ID が 3 の場合、バックプレーンインターフェイスは 3/\/\ でなければなりません。

ノードをクラスタに参加させるときに、次のメッセージが表示されます。「エラー**:** ローカルノードはクラスタのメンバーではないため、クラスタリングを有効にできません。このエラーを解決するにはどうすればいいですか**?**

このエラーは、ノードの NSIP をクラスターに追加せずにノードに参加しようとしたときに発生します。このエラーを解決するには、まず add cluster node コマンドを使用してノードの NSIP アドレスをクラスターに追加し、次に **join cluster** コマンドを実行する必要があります。

ノードをクラスターに参加させるときに、次のメッセージが表示されます。「エラー**:** 接続が拒否されました。このエラーを解決するにはどうすればいいですか**?**

このエラーは、以下の理由により発生する可能性があります。

- 接続の問題。ノードはクラスタ IP アドレスに接続できません。参加しようとしているノードからクラスター IP アドレスに ping を実行してみます。
- クラスタ **IP** アドレスが重複しています。クラスタ IP アドレスがクラスタ以外のノードに存在するかどうかを確認します。その場合は、クラスター IP アドレスを作成して、クラスターに再参加してみてください。

ノードをクラスターに参加させるときに、次のメッセージが表示されます。「エラー**:** 構成コーディネーターとローカルノードのライセンスが一致しません。このエラーを解決するにはどうすればいいですか**?**

クラスターに参加させるアプライアンスには、構成コーディネーターと同じライセンスが必要です。このエラーは、参加するノードのライセンスが、構成コーディネーターのライセンスと一致しない場合に発生します。このエラーを解決するには、両方のノードで以下のコマンドを実行し、出力を比較します。

コマンドラインから**:**

- `show ns hardware`
- `show ns license`

シェルから**:**

- `nsconmsg -g feature -d stats`
- `ls /nsconfig/license`
- /var/log/license.log ファイルの内容を表示する

クラスターノードの構成がクラスター構成と同期していない場合はどうすればよいですか

通常、構成はすべてのクラスターノード間で自動的に同期されます。ただし、特定のノードで構成が同期されていないと思われる場合は、同期するノードから force clustersync コマンドを実行して同期を強制する必要があります。詳細については、「クラスター構成の同期」を参照してください。

クラスタノードを構成するときに、「エラー: セッションは読み取り専用です。クラスタの IP アドレスに接続して構成を変更します」

クラスターのすべての構成はクラスター IP アドレスを使用して行う必要があり、構成は他のクラスターノードに伝達されます。個々のノードの NSIP アドレスを介して確立されたセッションはすべて読み取り専用です。

ノードヘルスが「**UP**」と表示されているのに、ノードの状態が「**INACTIVE**」と表示されるのはなぜですか**?**

正常なノードは、さまざまな理由で INACTIVE 状態になることがあります。ns.log またはエラーカウンタをスキャンすると、正確な理由を特定できます。

ノードのヘルスが「**NOT UP**」と表示されている場合に、そのノードのヘルスを解決する方法を教えてください

ノードヘルス「**Not UP**」は、ノードに何らかの問題があることを示します。根本原因を知るには、**show cluster node** コマンドを実行する必要があります。このコマンドは、ノードのプロパティとノード障害の理由を表示します。

ノードの健全性が「**NOT UP**」と表示され、その理由がノード上で構成コマンドが失敗したことを示している場合はどうすればよいですか**?**

この問題は、一部のコマンドがクラスターノードで実行されない場合に発生します。このような場合は、以下のオプションのいずれかを使用して構成が同期されていることを確認する必要があります。

- 一部のクラスターノードがこの状態にある場合は、それらのノードで強制クラスター同期操作を実行する必要があります。詳細については、「クラスター構成の同期」を参照してください。
- すべてのクラスターノードがこの状態にある場合は、すべてのクラスターノードでクラスターインスタンスを無効にしてから有効にする必要があります。

set virtual server コマンドを実行すると、次のメッセージが表示されます。「そのようなリソースはありません。このエラーを解決するにはどうすればいいですか**?**

set vserver コマンドはクラスタリングではサポートされていません。vserver の設定解除、**vserver** の有効化、**vserver** の無効化、および **rm vserver** のコマンドもサポートされません。ただし、**show vserver** コマンドはサポートされています。

Telnet セッションではクラスタを設定できません。どうしたらいいですか**?**

Telnet セッションでは、クラスタ IP アドレスには読み取り専用モードでのみアクセスできます。したがって、Telnet セッションではクラスタを設定できません。

クラスターノード間で大きな時間差があることに気付きました。このエラーを解決するにはどうすればいいですか**?**

バックプレーンのスイッチが原因で PTP パケットがドロップされたり、仮想環境で物理リソースが過負荷になったりすると、時刻が同期されません。

時刻を同期するには、クラスタ IP アドレスで次の操作を行う必要があります。

1. PTP を無効にします。

   ptp ステート無効化を設定

2. クラスタのネットワークタイムプロトコル (NTP) を構成します。詳細については、クロック同期の設定を参照してください。

クラスタ **IP** アドレスとクラスタノードの **NSIP** アドレスに接続できない場合、どうすればよいですか**?**

クラスタ IP アドレスまたはクラスタノードの NSIP にアクセスできない場合は、シリアルコンソールからアプライアンスにアクセスする必要があります。NSIP アドレスに到達できる場合は、シェルプロンプトで次のコマンドを実行して、シェルからクラスタ IP アドレスに SSH 接続できます。

“‘# ssh nsroot@

```
## 接続に問題があるクラスターノードを回復するにはどうすればいいですか?

接続に問題があるノードを回復するには:

1.  そのノードのクラスターインスタンスを無効にします (クラスターノードのNSIPからはコマンドを実行できないため)。

1.  ノードのリカバリに必要なコマンドを実行します。

1.  そのノードでクラスターインスタンスを有効にします。

## クラスターの一部のノードには 2 つのデフォルトルートがあります。クラスターノードから 2 つ目のデフォルトルートを削除する方法を教えてください

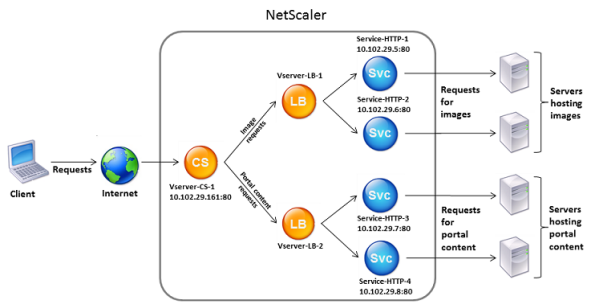
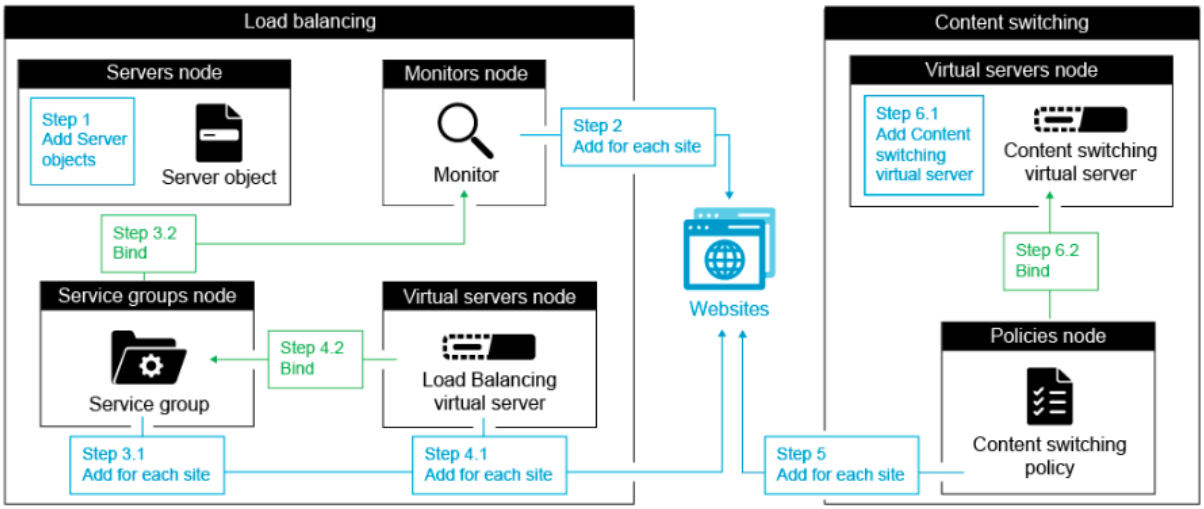
追加のデフォルトルートを削除するには、余分なルートがある各ノードで以下を実行します。

1.  クラスターインスタンスを無効にします。

    ```disable cluster instance <clId><!--NeedCopy-->
```

1. ルートを削除します。

   ```
   rm route <network> <netmask> <gateway><!--NeedCopy-->
   ```


   ```

2. クラスターインスタンスを有効にします。

```
enable cluster instance <clId><!--NeedCopy-->
```

既存のクラスターノードがオンラインになると、クラスター機能が影響を受けます。このエラーを解決するにはどうすればいいですか❓

ノードがクラスター外にあるときにそのノードの RPC パスワードをクラスタ IP アドレスから変更すると、ノードがオンラインになったときに RPC 認証情報が一致せず、クラスターの機能に影響する可能性があります。この問題を解決するには、set ns rpcNode コマンドを使用して、オンラインになったノードの NSIP のパスワードを更新します。

コンテンツスイッチ

August 15, 2023

今日の複雑な Web サイトでは、異なるユーザーに異なるコンテンツを提示したい場合があります。たとえば、顧客またはパートナーの IP 範囲のユーザに、特別な Web ポータルへのアクセスを許可する場合があります。特定の地域に関連するコンテンツを、その地域のユーザーに提示したい場合があります。異なる言語のコンテンツを、それらの言語の話者に提示することができます。スマートフォンなどの特定のデバイスに合わせたコンテンツを、そのデバイスを使用するユーザーに提示することができます。NetScaler ADC コンテンツスイッチング機能を使用すると、アプライアンスは、ユーザーに提示する特定のコンテンツに基づいて、複数のサーバーにクライアント要求を分散できます。

コンテンツスイッチングを構成するには、まず基本的なコンテンツスイッチング設定を作成し、ニーズに合わせてカスタマイズします。これには、コンテンツスイッチング機能の有効化、切り替えされるコンテンツの各バージョンをホストするサーバーの負荷分散の設定、コンテンツスイッチング仮想サーバーの作成、どの要求がどの負荷分散仮想サーバーに送信されるかを選択するポリシーの作成、ポリシーをコンテンツスイッチング仮想サーバーにバインドします。その後、ポリシーの優先順位を設定し、バックアップ仮想サーバーを構成してセットアップを保護し、要求をキャッシュにリダイレクトしてセットアップのパフォーマンスを向上させることで、ニーズに合わせてセットアップをカスタマイズできます。

コンテンツスイッチングの仕組み

コンテンツスイッチングを使用すると、NetScaler ADC アプライアンスは、同じ Web ホストに送信された要求を、異なるコンテンツを持つ異なるサーバーに送信できます。たとえば、動的コンテンツ（サフィックスが.asp、.dll、または.exe の URL など）のリクエストをあるサーバに送信し、静的コンテンツのリクエストを別のサーバに送信するようにアプライアンスを設定できます。TCP/IP ヘッダーとペイロードに基づいてコンテンツスイッチングを実行するようにアプライアンスを設定できます。

また、コンテンツスイッチングを使用して、さまざまなクライアント属性に基づいて異なるコンテンツを持つ異なるサーバーに要求をリダイレクトするようにアプライアンスを構成することもできます。これらのクライアント属性には、次のようなものがあります。

- デバイスタイプ。アプライアンスは、クライアント要求のユーザエージェントまたはカスタム HTTP ヘッダーを調べて、要求の発信元のデバイスのタイプを確認します。デバイスタイプに基づいて、要求を特定の Web サーバーに送信します。たとえば、要求が携帯電話から送信された場合、要求はサーバーに送信され、ユーザーが携帯電話で表示できるコンテンツを提供できます。コンピュータからの要求は、コンピュータ画面用に設計されたコンテンツを提供できる別のサーバに送信されます。
- 言語。アプライアンスは、クライアント要求の Accept-Language HTTP ヘッダーを調べ、クライアントのブラウザで使用される言語を決定します。次に、アプライアンスは、その言語でコンテンツを提供するサーバーに要求を送信します。たとえば、言語に基づくコンテンツスイッチングを使用すると、アプライアンスは、フランス語でコンテンツを要求するようにブラウザが設定されているユーザを、フランス語版の新聞を使用するサーバに送信できます。ブラウザが英語のコンテンツをリクエストするように設定されている他のユーザーを、英語バージョンのサーバーに送信できます。
- クッキー。アプライアンスは、サーバが以前に設定した Cookie の HTTP 要求ヘッダーを調べます。Cookie が見つかると、カスタムコンテンツをホストする適切なサーバーにリクエストを転送します。たとえば、クライアントが顧客ロイヤルティプログラムのメンバーであることを示す Cookie が見つかった場合、リクエストはより高速なサーバーまたは特別なコンテンツを持つサーバーに送信されます。クッキーが見つからない場合、またはクッキーがユーザーがメンバーではないことを示している場合、リクエストは一般向けのサーバーに送信されます。
- **HTTP** メソッド。アプライアンスは、使用されたメソッドの HTTP ヘッダーを調べ、クライアント要求を適切なサーバーに送信します。たとえば、イメージに対する GET リクエストはイメージサーバーに送信され、POST リクエストは動的コンテンツを処理するより高速なサーバーに送信できます。
- レイヤ **3/4** データ。アプライアンスは、送信元または宛先 IP、送信元または宛先ポート、または TCP または UDP ヘッダーに存在するその他の情報に対する要求を調べ、クライアント要求を適切なサーバーに送信します。たとえば、顧客に属する送信元 IP からのリクエストは、より高速なサーバー上のカスタム Web ポータル、または特別なコンテンツを含むカスタム Web ポータルに送信できます。

一般的なコンテンツスイッチング展開は、次の図に示すエンティティで構成されます。

図 1: コンテンツスイッチングアーキテクチャ



コンテンツスイッチの構成には、コンテンツスイッチ仮想サーバー、負荷分散仮想サーバーとサービスから成る負荷

分散セットアップ、およびコンテンツスイッチポリシーが含まれます。コンテンツスイッチを構成するには、コンテンツスイッチ仮想サーバーを構成し、ポリシーおよび負荷分散仮想サーバーに関連付ける必要があります。このプロセスでは、*content group*-は、特定のコンテンツスイッチング構成に関連するすべての仮想サーバーとポリシーのグループを作成します。

コンテンツスイッチングは、HTTP、HTTPS、TCP、および UDP 接続で使用できます。HTTPS の場合は、SSL オフロードを有効にする必要があります。

要求がコンテンツスイッチ仮想サーバーに到達すると、仮想サーバーはその要求に対して関連するコンテンツスイッチポリシーを適用します。ポリシーの優先順位により、コンテンツスイッチ仮想サーバーにバインドされたポリシーを評価する順序を定義します。高度なポリシーポリシーを使用している場合、ポリシーをコンテンツスイッチ仮想サーバーにバインドするときに、そのポリシーに優先順位を割り当てる必要があります。NetScaler ADC クラシックポリシーを使用している場合は、ポリシーに優先順位を割り当てることができますが、必須ではありません。優先順位を割り当てると、設定した順序でポリシーが評価されます。そうしない場合、NetScaler ADC アプライアンスは、ポリシーが作成された順序でポリシーを評価します。

ポリシープライオリティの設定に加えて、Goto 式とポリシーバンクの呼び出しを使用して、ポリシー評価の順序を操作できます。高度なポリシー設定の詳細については、「高度なポリシーポリシーの設定」を参照してください。

ポリシーが評価されると、コンテンツスイッチ仮想サーバーは要求を適切な負荷分散仮想サーバーにルーティングし、適切なサービスに送信します。

コンテンツスイッチ仮想サーバーは、他の仮想サーバーにのみ要求を送信できます。外部ロードバランサを使用している場合は、そのロードバランサ用の負荷分散仮想サーバーを作成し、その仮想サーバーをサービスとしてコンテンツスイッチ仮想サーバーにバインドする必要があります。

基本的なコンテンツ切り替えの設定

December 8, 2023

コンテンツスイッチングを構成する前に、コンテンツスイッチングの設定方法と、サービスと仮想サーバーの接続方法を理解しておく必要があります。

基本的で機能的なコンテンツスイッチング設定を構成するには、まずコンテンツスイッチング機能を有効にします。次に、少なくとも 1 つのコンテンツグループを作成します。コンテンツグループごとに、コンテンツスイッチングを使用する Web サイトのグループへの要求を受け入れるコンテンツスイッチング仮想サーバーを作成します。また、負荷分散セットアップを作成します。これには、コンテンツスイッチング仮想サーバーが要求を転送する負荷分散仮想サーバーのグループが含まれます。どの要求をどの負荷分散仮想サーバーに送信するかを指定するには、リダイレクトされる要求の種類ごとに 1 つずつ、少なくとも 2 つのコンテンツスイッチングポリシーを作成します。仮想サーバーとポリシーを作成したら、ポリシーをコンテンツスイッチ仮想サーバーにバインドします。また、ポリシーを複数のコンテンツスイッチング仮想サーバーにバインドすることもできます。ポリシーをバインドするときは、ポリシーに一致する要求を転送する負荷分散仮想サーバーを指定します。

個々のポリシーをコンテンツスイッチ仮想サーバーにバインドするだけでなく、ポリシーラベルをバインドできます。コンテンツグループをさらに作成する場合は、ポリシーまたはポリシーラベルを複数のコンテンツスイッチ仮想サーバーにバインドできます。

次の図は、コンテンツスイッチングの設定に必要な手順を示しています。



> 注
>
> コンテンツグループの作成後、コンテンツスイッチング仮想サーバーを変更して構成をカスタマイズできます。

コンテンツスイッチを有効にする

コンテンツスイッチング機能を使用するには、コンテンツスイッチングを有効にする必要があります。コンテンツスイッチング機能が無効になっていても、コンテンツスイッチングエンティティを構成できます。ただし、エンティティは機能しません。

CLI を使用したコンテンツスイッチングの有効化

コマンドプロンプトで次のコマンドを入力して、コンテンツスイッチングを有効にし、構成を確認します:

```
enable ns feature CS

show ns feature
<!--NeedCopy-->
```

例:

```
> enable  feature ContentSwitch
    Done
> show feature

    Feature                         Acronym             Status
```



```
    -------                          -------              ------
1)     Web Logging                      WL                   OFF
2)     Surge Protection                 SP                   ON
3)     Load Balancing                   LB                   ON
4)     Content Switching                CS                   ON
.
.
.
22)    Responder                        RESPONDER            ON
23)    HTML Injection                   HTMLInjection        ON
24)    NetScaler Push                   push                 OFF
Done
<!--NeedCopy-->
```

GUI を使用してコンテンツの切り替えを有効にする

[システム] > [設定] に移動し、[モードと機能] グループで [基本機能の構成] を選択し、[コンテンツの切り替え] を選択します。

コンテンツスイッチ仮想サーバーの作成

コンテンツスイッチ仮想サーバーを追加、変更、および削除できます。仮想サーバーの作成時の状態は DOWN です。これは、負荷分散仮想サーバーがまだバインドされていないためです。

CLI を使用して仮想サーバーを作成する

コマンドプロンプトで入力します:

```
add cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

例:

```
add cs vserver Vserver-CS-1 HTTP 10.102.29.161 80
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチ仮想サーバーを追加する

1. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバーを追加します。
2. コンテンツスイッチング仮想サーバーの名前を指定します。

   注

> プロトコルごとに異なるコンテンツスイッチング仮想サーバーがあります。(たとえば、HTTP と SSL)。

3. 関連するフィールドに入力し、「**OK**」をクリックします。

コンテンツスイッチング仮想サーバの統計情報

コンテンツスイッチ仮想サーバの統計情報には、仮想サーバの選択、要求バイト、応答バイト数、受信パケット総数、送信パケット総数、スピルオーバーしきい値、スピルオーバー選択、現在確立されているクライアント接続、仮想サーバのダウンバックアップ選択などの情報が表示されます。

コンテンツスイッチング仮想サーバーの統計情報には、バインドされたデフォルトの負荷分散仮想サーバーの概要の詳細も表示されます。

CLI を使用してコンテンツスイッチング仮想サーバーの統計を表示する

コマンドプロンプトで入力します:

```
stat cs vserver <name>
<!--NeedCopy-->
```

例:

```
stat cs vserver CS_stats
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチング仮想サーバーの統計を表示する

1. **[**トラフィック管理**]** > **[**コンテンツスイッチング**]** > **[**仮想サーバー**]** に移動します。
2. 仮想サーバを選択し、[統計] をクリックします。

コンテンツスイッチングの負荷分散設定を設定する

コンテンツスイッチング仮想サーバーは、すべての要求を負荷分散仮想サーバーにリダイレクトします。切り替えられるコンテンツのバージョンごとに 1 つの負荷分散仮想サーバーを作成する必要があります。これは、コンテンツの各バージョンに対してサーバーが 1 つしかない場合でも当てはまるため、それらのサーバーで負荷分散を行っていません。また、コンテンツの各バージョンをミラーリングする複数の負荷分散サーバーを使用して、実際の負荷分散を構成することもできます。いずれのシナリオでも、コンテンツスイッチング仮想サーバーには、切り替え対象のコンテンツの各バージョンに割り当てられた特定の負荷分散仮想サーバーが必要です。

負荷分散仮想サーバーは、要求をサービスに転送します。バインドされているサービスが 1 つだけの場合は、そのサービスが選択されます。複数のサービスがバインドされている場合は、設定された負荷分散方式を使用して要求のサービスを選択し、その要求を選択したサービスに転送します。

基本的な負荷分散設定を構成するには、次のタスクを実行する必要があります。

- 負荷分散仮想サーバーを作成する
- サービスを作成する
- 負荷分散仮想サーバへのサービスのバインド

負荷分散の詳細については、「負荷分散の仕組み」を参照してください。基本的な負荷分散構成の設定の詳細については、「基本的な負荷分散の設定」を参照してください。

コンテンツ切り替えアクションの設定

コンテンツスイッチングポリシーのターゲット負荷分散仮想サーバーは、ポリシーをコンテンツスイッチ仮想サーバーにバインドするときに指定します。したがって、トラフィックを転送する負荷分散仮想サーバーごとに 1 つのポリシーを構成する必要があります。

ただし、コンテンツスイッチングポリシーでデフォルトの構文ルールが使用されている場合は、ポリシーのアクションを設定できます。アクションでは、ターゲットの負荷分散仮想サーバーの名前を指定するか、実行時に要求を送信する負荷分散仮想サーバーの名前を計算する要求ベースの式を構成できます。アクション式はデフォルトの構文で指定する必要があります。

式オプションを使用すると、コンテンツスイッチング仮想サーバーごとに必要なポリシーが 1 つだけであるため、コンテンツスイッチ構成のサイズを大幅に縮小できます。ターゲット負荷分散仮想サーバーがコンテンツスイッチングポリシーで指定されなくなったため、アクションを使用するコンテンツスイッチングポリシーを複数のコンテンツスイッチング仮想サーバーにバインドすることもできます。単一のポリシーを複数のコンテンツスイッチング仮想サーバーにバインドする機能により、コンテンツスイッチング構成のサイズをさらに縮小できます。

アクションを作成したら、コンテンツスイッチングポリシーを作成し、ポリシーでアクションを指定します。これにより、ポリシーが要求に一致したときにアクションが実行されます。

> 注
>
> また、デフォルトの構文ルールを使用するコンテンツスイッチングポリシーの場合、別のアクションを使用する代わりに、ポリシーをコンテンツスイッチング仮想サーバーにバインドするときにターゲットの負荷分散仮想サーバーを指定することもできます。ドメインベースのポリシー、URL ベースのポリシー、およびクラシック表現を使用するルールベースのポリシーでは、アクションは使用できません。したがって、これらの種類のポリシーでは、ポリシーをコンテンツスイッチ仮想サーバーにバインドするときに、ターゲットの負荷分散仮想サーバーの名前を指定します。

ターゲットの負荷分散仮想サーバーの名前を指定するアクションを設定します

コンテンツスイッチングアクションでターゲットの負荷分散仮想サーバーの名前を指定する場合は、ターゲットの負荷分散仮想サーバーと同じ数のコンテンツスイッチングポリシーが必要です。この場合、コンテンツスイッチングの決定はコンテンツスイッチングポリシーのルールに基づいて行われ、アクションはターゲットの負荷分散仮想サーバーを指定するだけです。要求がポリシーと一致すると、要求は指定された負荷分散仮想サーバーに転送されます。

CLI を使用して、ターゲットの負荷分散仮想サーバーの名前を指定するコンテンツスイッチングアクションを作成して検証します

コマンドプロンプトで入力します:

```
add cs action <name> -targetLBVserver <string> [-comment <string>]

show cs action <name>
<!--NeedCopy-->
```

例:

```
> add cs action mycsaction -targetLBVserver mylbvserver -comment "
    Forwards requests to mylbvserver."
 Done
> show cs action mycsaction
    Name: mycsaction
    Target LB Vserver: mylbvserver
    Hits: 0
    Undef Hits: 0
    Action Reference Count: 0
    Comment: "Forwards requests to mylbvserver."

 Done
>
<!--NeedCopy-->
```

GUI を使用して、ターゲットの負荷分散仮想サーバーの名前を指定するコンテンツスイッチングアクションを構成します

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [アクション] に移動します。
2. コンテンツスイッチングアクションを構成し、ターゲットの負荷分散仮想サーバーの名前を指定します。

実行時にターゲットを選択するための式を指定するアクションを設定します

ターゲットの負荷分散仮想サーバーの名前を動的に計算できる要求ベースの式を構成する場合は、適切な仮想サーバーを選択するためにコンテンツスイッチングポリシーを 1 つだけ構成する必要があります。この場合、コンテンツスイッチングの決定はアクションの式に基づいているため、ポリシーのルールは単純な TRUE (ポリシーはすべての要求に一致します) にすることができます。アクションに式を設定すると、コンテンツスイッチング構成のサイズを大幅に縮小できます。

実行時にターゲットの負荷分散仮想サーバーの名前を計算するための要求ベースの式を構成する場合は、構成内の負荷分散仮想サーバーの名前を慎重に検討する必要があります。アクションでリクエストベースのポリシー式を使用して、名前を導出できる必要があります。

たとえば、URL サフィックス (要求されたリソースの拡張子) に基づいて要求を切り替える場合、負荷分散仮想サーバーの名前を指定するときに、URL サフィックスを事前に定義された文字列に追加する規則に従うことができます (mylb_など) 。たとえば、HTML ページと PDF ファイルの負荷分散仮想サーバーには、それぞれmylb_htmlとmylb_pdfという名前を付けることができます。この場合、コンテンツスイッチングアクションで使用して、適切な負荷分散仮想サーバーを選択できるルールは"mylb_"+HTTP.REQ.URL.SUFFIXです。コンテンツスイッチング仮想サーバーが HTML ページの要求を受信すると、式がmylb_htmlを返し、要求が仮想サーバーmylb_htmlに切り替えられます。

CLI を使用して式を指定するコンテンツスイッチングアクションを作成する

コマンドラインで次のコマンドを入力して、式を指定するコンテンツスイッチングアクションを作成し、構成を確認します:

```
add cs action <name> -targetVserverExpr <expression>) [-comment <string
    >]

show cs action <name>
<!--NeedCopy-->
```

例:

```
> add cs action mycsaction1 -targetvserverExpr '"mylb_" + HTTP.REQ.URL.
    SUFFIX'
 Done
> show cs action mycsaction1
    Name: mycsaction1
    Target Vserver Expression: "mylb_" + HTTP.REQ.URL.SUFFIX
    Target LB Vserver: No_Target
                ...
 Done
>
<!--NeedCopy-->
```

GUI を使用して式を指定するコンテンツ切り替えアクションを設定します

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [アクション] に移動します。
2. コンテンツスイッチングアクションを構成し、ターゲットの負荷分散仮想サーバーの名前を動的に計算する式を指定します。

コンテンツスイッチングポリシーの設定

コンテンツスイッチングポリシーは、負荷分散仮想サーバーに送信される要求の種類を定義します。これらのポリシーは、割り当てられた優先順位の順に適用されます（NetScaler クラシックポリシーを使用し、バインド時に優先順位を割り当てない場合)、ポリシーが作成された順序で適用されます。

ポリシーには次のものがあります:

- ドメインベースのポリシー。NetScaler アプライアンスは、着信 URL のドメインをポリシーで指定されたドメインと比較します。その後、アプライアンスは最も適切なコンテンツを返します。ドメインベースのポリシーは、クラシックポリシーである必要があります。デフォルトの構文ポリシーは、このタイプのコンテンツスイッチングポリシーではサポートされていません。
- **URL** ベースのポリシー。アプライアンスは、着信 URL とポリシーで指定された URL を比較します。次に、アプライアンスは最も適切な URL ベースのコンテンツを返します。通常は、最も長く一致する設定済み URL です。URL ベースのポリシーは、クラシックポリシーである必要があります。デフォルトの構文ポリシーは、このタイプのコンテンツスイッチングポリシーではサポートされていません。
- ルールベースのポリシー。アプライアンスは、受信データをポリシーで指定された式と比較します。ルールベースのポリシーは、従来の式またはデフォルトの構文式を使用して作成します。ルールベースのコンテンツスイッチングポリシーでは、クラシックシンタックスポリシーとデフォルトシンタックスポリシーの両方がサポートされています。

  > 注
  >
  > ルールベースのポリシーは、オプションのアクションで設定できます。アクションを含むポリシーは、複数の仮想サーバまたはポリシーラベルにバインドできます。

  ポリシーをコンテンツスイッチ仮想サーバーにバインドするときに優先度を設定すると、ポリシーは優先度の順に評価されます。ポリシーをバインドするときに特定の優先順位を設定しない場合、ポリシーは作成された順序で評価されます。

NetScaler クラシックポリシーと式の詳細については、「クラシックポリシーと式の構成」を参照してください。デフォルトの構文ポリシーの詳細については、デフォルトの構文式の構成を参照してください。

CLI を使用してコンテンツスイッチングポリシーを作成する

コマンドプロンプトで、次のコマンドのいずれかを入力します:

```
add cs policy <policyName> -domain <domain>

add cs policy <policyName> -url <URLValue>

add cs policy <policyName> -rule <RULEValue>

add cs policy <policyName> -rule <RULEValue> -action <actionName>
<!--NeedCopy-->
```

例:

```
add cs policy Policy-CS-1 -url "http://example.com"

add cs policy Policy-CS-1 -domain "example.com"
```



```
4
5 add cs policy Policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(24).EQ
     (10.217.84.0)"
6
7 add cs policy Policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2009 Nov,GMT 2009
     Dec)"
8
9 add cs policy Policy-CS-3 -rule "http.req.method.eq(GET)" -action act1
10 <!--NeedCopy-->
```

CLI を使用してコンテンツスイッチングポリシーの名前を変更する

コマンドプロンプトで入力します:

```
1 rename cs policy <policyName> <newName>
2 <!--NeedCopy-->
```

例:

```
1 rename cs policy myCSPolicy myCSPolicy1
2 <!--NeedCopy-->
```

GUI を使用してコンテンツスイッチングポリシーの名前を変更する

[トラフィック管理] **>** [コンテンツスイッチング] **>** [ポリシー] に移動し、ポリシーを選択して、[アクション] リストで [名前の変更] を選択します。

GUI を使用してコンテンツスイッチングポリシーを作成する

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [ポリシー] に移動し、[追加] をクリックします。
2. 関連フィールドに入力し、[作成] をクリックします。

コンテンツスイッチングポリシーラベルの設定

ポリシーラベルは、ポリシーがバインドされるユーザー定義のバインドポイントです。ポリシーラベルが呼び出されると、そのラベルにバインドされているすべてのポリシーが、割り当てられた優先順位の順序で評価されます。ポリシーラベルには、1 つ以上のポリシーを含めることができ、各ポリシーに独自の結果を割り当てることができます。ポリシー・ラベル内の 1 つのポリシーが一致すると、次のポリシーに進んだり、別のポリシー・ラベルまたは適切なリソースが呼び出されたり、ポリシー評価が即座に終了し、ポリシー・ラベルを呼び出したポリシーに制御が戻ったりすることがあります。ポリシーラベルは、デフォルトの構文ポリシーにのみ作成できます。

コンテンツスイッチングポリシーラベルは、名前、ラベルタイプ、およびポリシーラベルにバインドされたポリシーのリストで構成されます。ポリシーラベルタイプは、ラベルにバインドされたポリシーに割り当てられたプロトコル

を指定します。これは、ポリシーラベルを呼び出すポリシーがバインドされているコンテンツスイッチング仮想サーバーのサービスタイプと一致する必要があります。たとえば、TCP ペイロードポリシーは、タイプ TCP のみのポリシーラベルにバインドできます。TCP ペイロードポリシーを HTTP タイプのポリシーラベルにバインドすることはサポートされていません。

コンテンツスイッチングポリシーラベル内の各ポリシーは、ターゲット (書き換えポリシーやレスポンダーポリシーなど、他の種類のポリシーに関連付けられているアクションと同等)、または gotoPriorityExpression オプションと呼び出しオプションのいずれかに関連付けられます。つまり、コンテンツスイッチングポリシーラベル内の特定のポリシーに対して、ターゲットを指定するか、gotoPriorityExpression オプションと呼び出しオプションを設定できます。また、複数のポリシーが true と評価された場合、true と評価された最後のポリシーのターゲットだけが考慮されます。

NetScaler CLI または GUI を使用して、コンテンツスイッチングポリシーラベルを構成できます。NetScaler CLI では、最初に add cs ポリシーラベルコマンドを使用してポリシーラベルを作成します。次に、bind cs policy label コマンドを使用して、一度に 1 つのポリシーをポリシーラベルにバインドします。NetScaler GUI では、1 つのダイアログボックスで両方のタスクを実行します。

CLI を使用してコンテンツスイッチングポリシーラベルを作成する

コマンドプロンプトで入力します:

```
add cs policylabel <labelName> <cspolicylabelTypetype>`
<!--NeedCopy-->
```

例:

```
add cs policylabel testpollab http
<!--NeedCopy-->
```

CLI を使用してコンテンツスイッチングポリシーラベル名を変更する

コマンドプロンプトで入力します:

```
rename cs policylabel <labelName> <newName>`
<!--NeedCopy-->
```

例:

```
rename cs policylabel oldPolicyLabelName newPolicyLabelName
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチングポリシーラベルの名前を変更する

[トラフィック管理] **>** [コンテンツスイッチング] **>** [ポリシーラベル] に移動し、ポリシーラベルを選択して、[アク

ション] リストで [名前の変更] を選択します。

CLI を使用してポリシーをコンテンツスイッチングポリシーラベルにバインドする

コマンドプロンプトで次のコマンドを入力して、ポリシーをポリシーラベルにバインドし、構成を確認します:

```
bind cs policylabel <labelName> <policyName> <priority>[-targetVserver
    <string>] | [-gotoPriorityExpression <expression>] | [-invoke <
    labeltype> <labelName>] ]

show cs policylabel <labelName>
<!--NeedCopy-->
```

例:

```
bind cs policylabel testpollab test_Pol 100 -targetVserver LBVIP
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 1
        Number of times invoked: 0
        Policy Name: test_Pol
        Priority: 100
        Target Virtual Server: LBVIP
<!--NeedCopy-->
```

> 注
>
> ポリシーにアクションが設定されている場合、ターゲット仮想サーバ（targetVServer）、優先度の式に移動（gotoPriorityExpression）、および呼び出し（呼び出し）パラメータは必要ありません。ポリシーにアクションが設定されていない場合は、targetVServer、gotoPriorityExpression、および呼び出しのいずれかのパラメータを少なくとも 1 つ設定する必要があります。

CLI を使用してポリシーラベルからポリシーをバインド解除する

コマンドプロンプトで次のコマンドを入力して、ポリシーラベルからポリシーをバインド解除し、構成を確認します:

```
unbind cs policylabel <labelName> <policyName>

show cs policylabel <labelName>
<!--NeedCopy-->
```

例:

```
unbind cs policylabel testpollab test_Pol
show cs policylabel testpollab
        Label Name: testpollab
```



```
        Label Type: HTTP
        Number of bound policies: 0
        Number of times invoked: 0
<!--NeedCopy-->
```

CLI を使用してポリシーラベルを削除する

コマンドプロンプトで入力します:

```
rm cs policylabel <labelName>
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチングポリシーラベルを管理する

[トラフィック管理] > [コンテンツスイッチング] > [ポリシーラベル] に移動し、ポリシーラベルを設定し、ポリシーをラベルにバインドし、オプションで優先度、gotoPriority 式、および呼び出しオプションを指定します。

ポリシーをコンテンツスイッチング仮想サーバーにバインドする

コンテンツスイッチ仮想サーバーとコンテンツスイッチポリシーの作成後、各ポリシーをコンテンツスイッチ仮想サーバーにバインドします。ポリシーをコンテンツスイッチ仮想サーバーにバインドするときには、ターゲットとなる負荷分散仮想サーバーを指定します。

> 注
>
> コンテンツスイッチングポリシーがデフォルトの構文ルールを使用している場合は、ポリシーのコンテンツスイッチングアクションを設定できます。アクションを構成する場合は、ポリシーをコンテンツスイッチ仮想サーバーにバインドするときではなく、アクションの構成時にターゲットの負荷分散仮想サーバーを指定する必要があります。コンテンツスイッチングアクションの構成の詳細については、「コンテンツスイッチングアクションの構成」を参照してください。

ポリシーをコンテンツスイッチ仮想サーバーにバインドし、CLI を使用してターゲット負荷分散仮想サーバーを選択します

コマンドプロンプトで入力します:

```
bind cs vserver <name>[-lbvserver<string> -targetLBVServer<string> -
    policyname <string> -priority <positive_integer>] [-
    gotoPriorityExpression <expression>] [-type ( REQUEST | RESPONSE )]
    [-invoke (<labelType> <labelName>)]
<!--NeedCopy-->
```

例:


```
bind cs vserver csw-vip2 -policyname csw-ape-policy2 -priority 14 -
    gotoPriorityExpression NEXT

bind cs vserver csw-vip3 -policyname rewrite-policy1 -priority 17 -
    gotoPriorityExpression
'q.header("a").count' -flowtype REQUEST -invoke policylabel label1

bind cs vserver Vserver-CS-1 Vserver-LB-1 -policyname Policy-CS-1 -
    priority 20
<!--NeedCopy-->
```

> 注
>
> ポリシーにアクションがある場合、パラメーターであるターゲット負荷分散仮想サーバー (TargetVServer)、移動優先度式 (gotoPriorityExpression)、および呼び出しメソッド (invoke) は使用できません。

ポリシーをコンテンツスイッチ仮想サーバーにバインドし、**GUI** を使用してターゲットの負荷分散仮想サーバーを選択します

[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開き、[コンテンツスイッチングポリシーバインディング] セクションでポリシーを仮想サーバーにバインドし、ターゲット負荷分散仮想サーバーを指定します。

コンテンツスイッチング用のポリシーベースのロギングの設定

コンテンツスイッチングポリシーには、ポリシーベースのロギングを設定できます。ポリシーベースのロギングでは、ログメッセージの形式を指定できます。ログメッセージの内容は、コンテンツスイッチングポリシーのデフォルトの構文式を使用して定義されます。ポリシーで指定されたコンテンツスイッチングアクションが実行されると、NetScaler アプライアンスは式からログメッセージを作成し、メッセージをログファイルに書き込みます。ポリシーベースのロギングは、コンテンツスイッチングアクションが実行時にターゲットの負荷分散仮想サーバーを識別する構成のテストとトラブルシューティングを行う場合に役立ちます。

> 注
>
> 特定の仮想サーバーにバインドされた複数のポリシーが TRUE と評価され、監査メッセージアクションで構成されている場合、NetScaler アプライアンスはすべての監査メッセージアクションを実行するわけではありません。これは、コンテンツスイッチングアクションが実行されるポリシーに対して構成された監査メッセージアクションのみを実行します。

コンテンツスイッチングポリシーのポリシーベースのロギングを設定するには、最初に監査メッセージアクションを設定する必要があります。監査メッセージアクションの構成の詳細については、「監査ログ用の NetScaler アプライアンスの構成」を参照してください。監査メッセージアクションを構成したら、コンテンツスイッチングポリシーでアクションを指定します。

CLI を使用してコンテンツスイッチングポリシーのポリシーベースのロギングを設定する

コマンドラインで次のコマンドを入力して、コンテンツスイッチングポリシーのポリシーベースのロギングを構成し、構成を確認します:

```
set cs policy <policyName> -logAction <string>

show cs policy <policyName>
<!--NeedCopy-->
```

例:

```
> set cs policy cspol1 -logAction csLogAction
 Done
> show cs policy cspol1

    Policy: cspol1  Rule: TRUE  Action: csact1
    LogAction: csLogAction
    Hits: 0

1)  CS Vserver: csvs1
    Priority: 10
 Done
>
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチングポリシーのポリシーベースのロギングを設定する

[** トラフィック管理] > [コンテンツスイッチング] **[ポリシー] に移動し、ポリシーを開き、[ログアクション] リストでポリシーのログアクションを選択します。

構成を確認する

コンテンツスイッチの構成が正しいことを確認するには、コンテンツスイッチングエンティティを表示する必要があります。コンテンツスイッチ構成の展開後に正常に動作することを確認するために、サーバーへのアクセス時に生成される統計情報を表示できます。

コンテンツスイッチ仮想サーバーのプロパティを表示する

NetScaler アプライアンスで構成したコンテンツスイッチング仮想サーバーのプロパティを表示できます。この情報を使用して、仮想サーバーが正しく構成されているかどうかを確認し、必要に応じてトラブルシューティングを行うことができます。名前、IP アドレス、ポートなどの詳細に加えて、仮想サーバーにバインドされたさまざまなポリシーとそのトラフィック管理設定を表示できます。

コンテンツスイッチングポリシーは、優先順位の順序で表示されます。複数のポリシーが同じ優先度を持つ場合、仮想サーバにバインドされている順序で表示されます。

> 注
>
> 負荷分散仮想サーバーにトラフィックを転送するようにコンテンツスイッチング仮想サーバーを構成した場合は、負荷分散仮想サーバーのプロパティを表示して、コンテンツスイッチングポリシーを表示することもできます。

CLI を使用してコンテンツスイッチング仮想サーバーのプロパティを表示する

構成内のすべてのコンテンツスイッチ仮想サーバーの基本プロパティ、または特定のコンテンツスイッチ仮想サーバーの詳細なプロパティを一覧表示するには、コマンドプロンプトで次のコマンドのいずれかを入力します:

```
show cs vserver

show cs vserver <name>
<!--NeedCopy-->
```

例

```
1.
show cs vserver Vserver-CS-1
Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
Last state change was at Thu Jun 30 10:48:59 2011
Time since last state change: 6 days, 20:03:00.760
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
Default: Content Precedence: RULE
Vserver IP and Port insertion: OFF
Case Sensitivity: ON
Push: DISABLED Push VServer:
Push Label Rule: none

...
1) Policy : __ESNS_PREBODY_POLICY Priority:0
2) Policy : __ESNS_POSTBODY_POLICY Priority:0

1) Compression Policy Name: __ESNS_CMP_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

2) Rewrite Policy Name: __ESNS_REWRITE_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

```



```
3) Cache Policy Name: dfbx Priority: 10
GotoPriority Expression: END
Flowtype: REQUEST

4) Responder Policy Name: __ESNS_RESPONDER_POLICY Priority: 2147483647
GotoPriority Expression: END

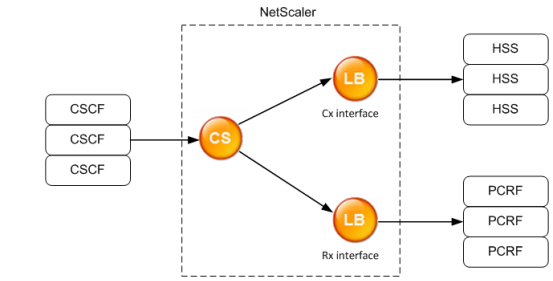
1) Policy: wiki Target: LBVIP2 Priority: 25 Hits: 0
2) Policy: plain Target: LBVIP1 Priority: 90 Hits: 0
3) Policy: DispOrderTest2 Target: KerbAuthLBVS Priority: 91 Hits: 0
4) Policy: test_Pol Target: LBVIP1 Priority: 92 Hits: 0
5) Policy: PolicyNameTesting Target: LBVIP1 Priority: 100 Hits: 0
 Done
>

show cs vserver
1) Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
…
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED

2) apubendpt (10.111.111.1:80) - HTTP Type: CONTENT
State: UP
…
Client Idle Timeout: 180 sec
Down state flush: DISABLED
…

3) apubendpt1 (10.111.111.2:80) - HTTP Type: CONTENT
State: UP
…
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
…
<!--NeedCopy-->
```

コンテンツスイッチングポリシーを表示する

名前、ドメイン、URL または式など、定義したコンテンツスイッチングポリシーのプロパティを表示し、その情報を使用して、構成の誤りを見つけたり、何かが正常に機能しない場合のトラブルシューティングを行うことができます。

CLI を使用してコンテンツスイッチングポリシーのプロパティを表示する

コンテンツスイッチングポリシーの設定の詳細を表示するには、次のコマンドのいずれかを入力します:


```
show cs policy

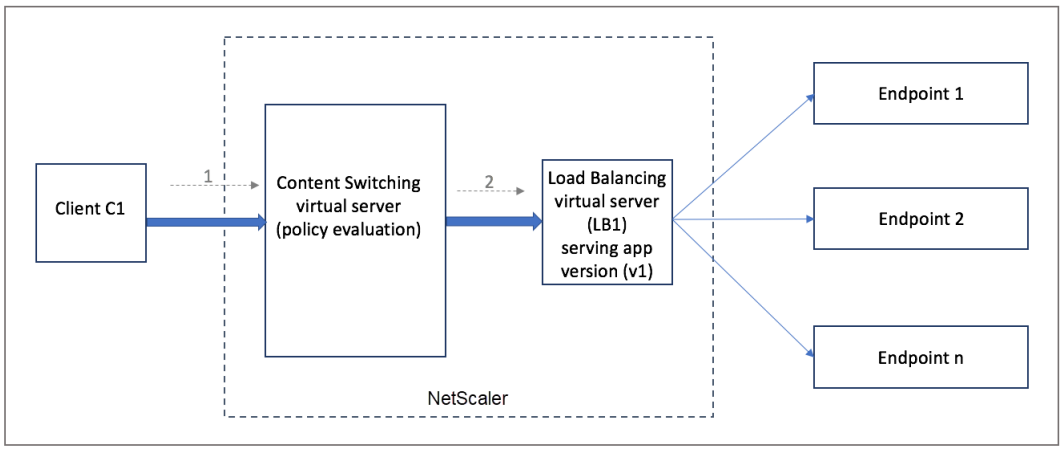
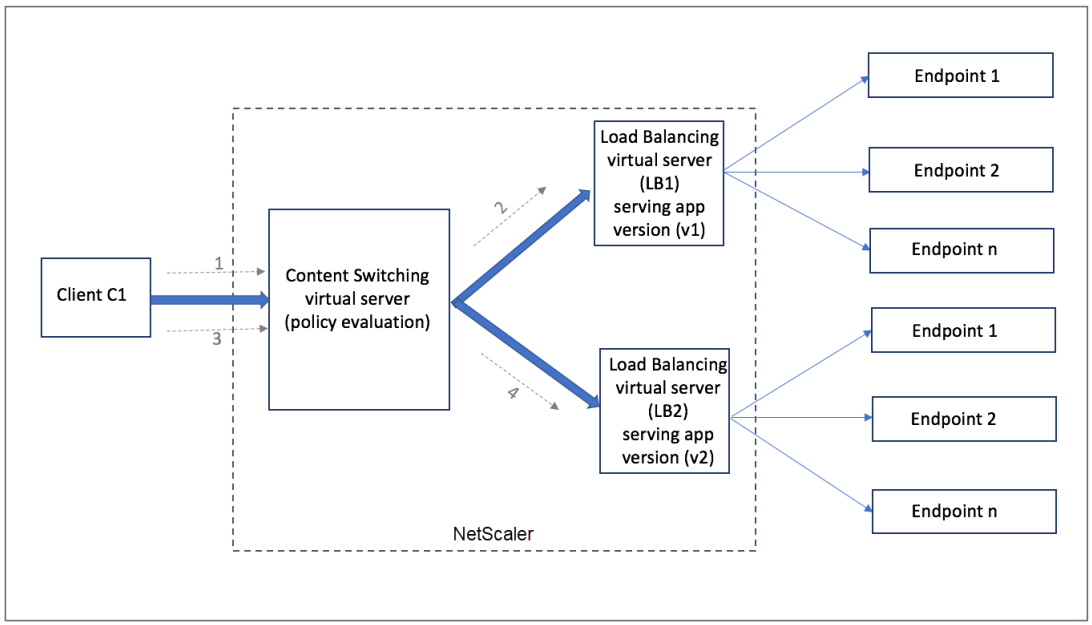
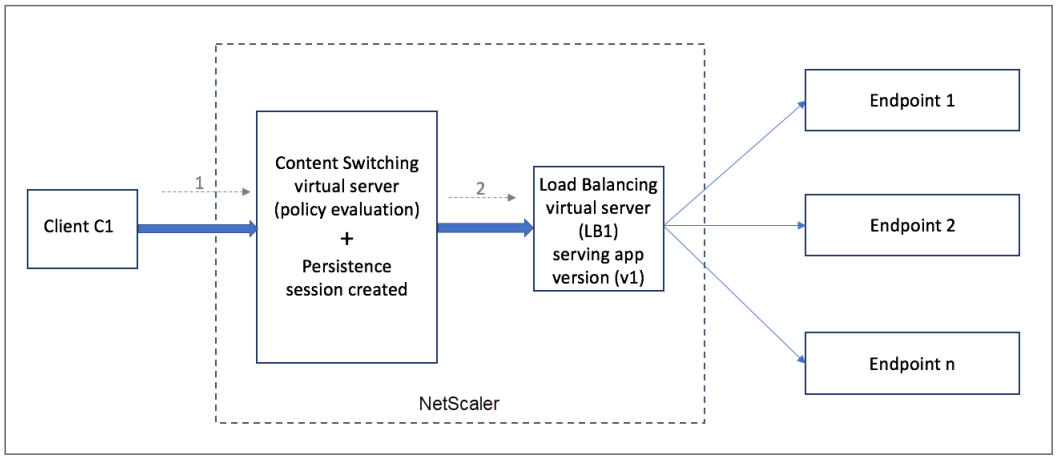
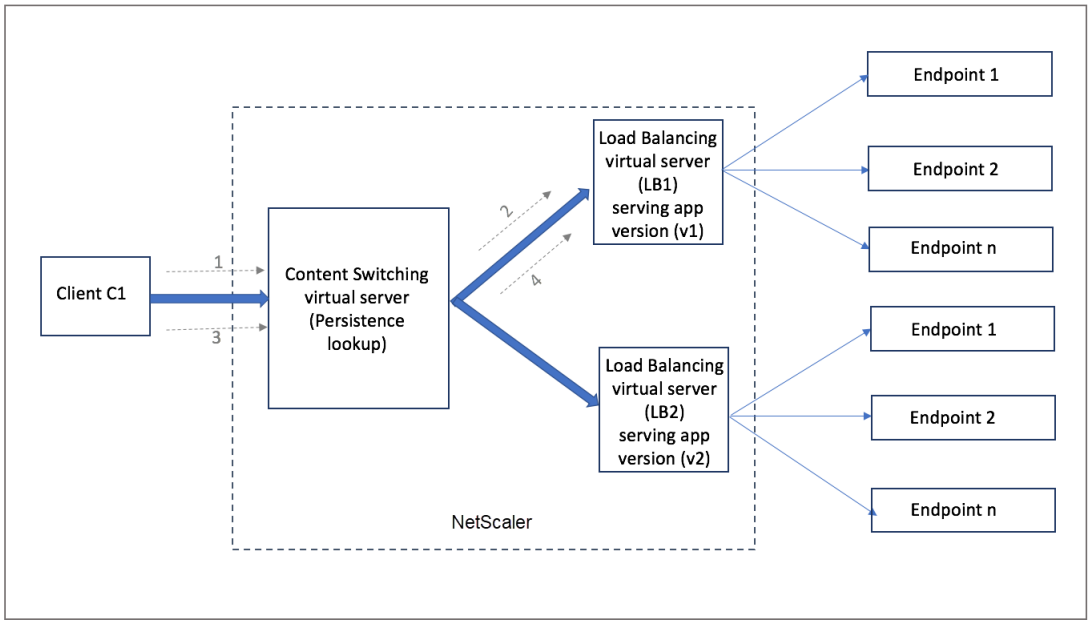
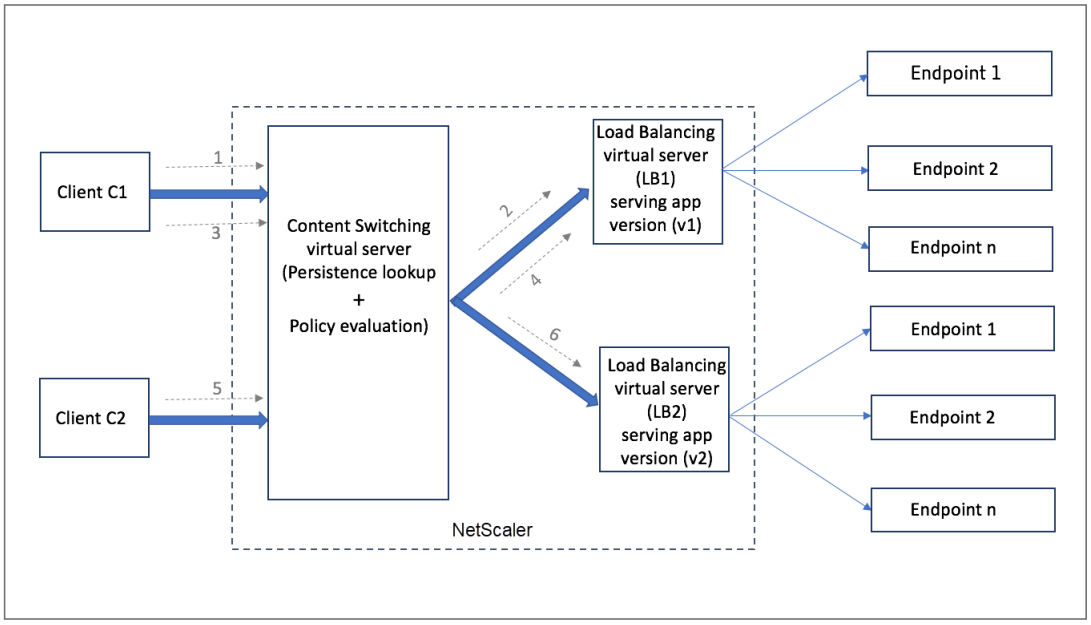
show cs policy <PolicyName>
<!--NeedCopy-->
```

例:

```
show cs policy

show cs policy Policy-CS-1
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチングポリシーのプロパティを表示する

[トラフィック管理] > [コンテンツスイッチング] > [ポリシー] に移動し、ポリシーを選択し、[アクション] リストで [バインドの表示] を選択します。

ビジュアライザーを使用してコンテンツスイッチング仮想サーバー構成を表示する

コンテンツスイッチングビジュアライザーは、コンテンツスイッチング構成をグラフィック形式で表示するために使用できるツールです。ビジュアライザを使用して、次の構成項目を表示できます。

- コンテンツスイッチング仮想サーバーがバインドされている負荷分散仮想サーバーの概要。
- 負荷分散仮想サーバーにバインドされているすべてのサービスおよびサービスグループ、およびサービスにバインドされているすべてのモニター。
- 表示されている要素の構成の詳細。
- コンテンツスイッチ仮想サーバーにバインドされているポリシー。これらのポリシーは、コンテンツスイッチングポリシーである必要はありません。リライトポリシーは、コンテンツスイッチング仮想サーバーにバインドすることもできます。

コンテンツスイッチングと負荷分散の設定が完了したら、構成全体をアプリケーションテンプレートファイルにエクスポートできます。

> 注
>
> ビジュアライザーにはグラフィカルインターフェイスが必要なため、GUI からのみ使用できます。

GUI のビジュアライザーを使用してコンテンツスイッチング構成を表示する

1. [トラフィック管理] **> [**コンテンツスイッチング**] > [**仮想サーバー] に移動します。
2. 詳細ウィンドウで、表示する仮想サーバーを選択し、[ビジュアライザー] をクリックします。
3. [コンテンツスイッチングビジュアライザ] ウィンドウでは、次のように表示可能領域を調整できます。

- ズームイン **(Zoom In**) アイコンとズームアウト (Zoom Out) アイコンをクリックして、表示領域を拡大または縮小します。
- **Save Image** アイコンをクリックして、グラフをイメージファイルとして保存します。
- [検索する文字列] フィールドに、探しているアイテムの名前を入力します。項目を識別するのに十分な文字を入力すると、その場所が強調表示されます。検索を制限するには、ドロップダウンメニューをクリックし、検索する要素のタイプを選択します。

4. この仮想サーバーにバインドされているエンティティの構成の詳細を表示するには、次の操作を行います。

   - 仮想サーバにバインドされているポリシーを表示するには、ダイアログボックスの上部にあるツールバーで、機能固有のポリシーアイコンを 1 つ以上選択します。ポリシーラベルが設定されている場合は、メインビュー領域に表示されます。
   - バインドされたサービスまたはサービスグループの構成の詳細を表示するには、サービスのアイコンをクリックし、[関連タスク] タブ、[メンバーサービスの表示] の順にクリックします。
   - モニタの構成の詳細を表示するには、モニタのアイコンをクリックし、[関連タスク] タブ、[モニタの表示] の順にクリックします。

5. コンテンツスイッチ構成内の仮想サーバーの詳細な統計情報を表示するには、統計情報を表示する仮想サーバーをクリックし、[関連タスク] タブ、[統計] の順にクリックします。

6. 負荷分散仮想サーバーのサービスコンテナ間で値が異なる、または定義されていないパラメータの比較リストを表示するには、コンテナのアイコンをクリックし、[関連タスク] タブをクリックし、[サービス属性の差分] をクリックします。

7. コンテナ内のサービスのモニタバインドの詳細を表示するには、[サービス属性の差分] ダイアログボックスの [グループ] 列で、[詳細] をクリックします。この比較リストは、すべてのサービスコンテナに適用する設定を持つサービスコンテナを判断するのに役立ちます。

8. 構成内の仮想サーバーが特定の時点で受信した 1 秒あたりの要求数、および書き換え、レスポンダー、およびキャッシュポリシーの特定の時点の 1 秒あたりの選択数を表示するには、[統計の表示] をクリックします。統計情報は、ビジュアライザーの各ノードに表示されます。この情報はリアルタイムでは更新されません。これは手動で更新されます。情報を更新するには、[統計情報の更新] をクリックします。

   > 注
   >
   > このオプションは、NetScaler nCore ビルドでのみ使用できます。

9. エレメントの設定詳細をドキュメントまたはスプレッドシートにコピーするには、そのエレメントのアイコンをクリックし、「関連タスク」、「プロパティのコピー」の順にクリックして、情報をドキュメントに貼り付けます。

10. ビジュアライザに表示される構成全体をアプリケーションテンプレートファイルにエクスポートするには、コンテンツスイッチング仮想サーバーのアイコンをクリックし、[関連タスク]、[テンプレートの作成] の順にクリックします。アプリケーションテンプレートを作成するときに、いくつかのポリシー式とアクションで変数を設定できます。詳細については、AppExpertを参照してください。

基本的なコンテンツスイッチング設定のカスタマイズ

December 8, 2023

基本的なコンテンツスイッチング設定を構成した後、要件に合わせてカスタマイズする必要がある場合があります。HTTP および SSL コンテンツスイッチング仮想サーバーは、個別の仮想サーバーを作成する代わりに、複数のポートでリッスンするように構成できます。特定の仮想 LAN にコンテンツスイッチングを設定する場合は、リッスンポリシーを使用してコンテンツスイッチング仮想サーバーを構成できます。

HTTP および **SSL** タイプのコンテンツスイッチング仮想サーバの複数ポートのサポート

HTTP および SSL コンテンツスイッチング仮想サーバーが複数のポートをリッスンするように、NetScaler を構成できます。個別の仮想サーバーを構成する必要はありません。この機能は、URL の一部と他の L7 パラメータに基づいてコンテンツスイッチングを決定する場合に特に便利です。同じ IP アドレスと異なるポートを持つ複数の仮想サーバを設定する代わりに、1 つの IP アドレスを設定し、ポートを * として指定できます。その結果、構成サイズも小さくなります。

コマンドラインを使用して複数のポートをリッスンするように **HTTP** または **SSL** コンテンツスイッチング仮想サーバーを構成するには

コマンドプロンプトで入力します:

```
add cs vserver <name> <serviceType> <IPAddress> Port *
```

例

```
> add cs vserver cs1 HTTP 10.102.92.215 *
 Done
> sh cs vserver cs1
        cs1 (10.102.92.215:*) - HTTP    Type: CONTENT
        State: UP
        Last state change was at Tue May 20 01:15:49 2014
        Time since last state change: 0 days, 00:00:03.270
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Appflow logging: ENABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Vserver IP and Port insertion: OFF
        L2Conn: OFF     Case Sensitivity: ON
        Authentication: OFF
        401 Based Authentication: OFF
        Push: DISABLED  Push VServer:
        Push Label Rule: none
```



```
21          IcmpResponse: PASSIVE
22          RHIstate:  PASSIVE
23          TD: 0
24   Done
25  <!--NeedCopy-->
```

構成ユーティリティを使用して複数のポートをリッスンするように **HTTP** または **SSL** コンテンツスイッチング仮想サーバーを構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、HTTP または SSL タイプの仮想サーバーを作成します。
2. ポートを指定するには、アスタリスク (*) を使用します。

VLAN 単位のワイルドカード仮想サーバの設定

特定の VLAN 上のトラフィックに対してコンテンツスイッチングを設定する場合は、指定された VLAN 上のトラフィックだけを処理するように制限するリッスンポリシーを使用して、ワイルドカード仮想サーバを作成できます。

コマンドラインインターフェイスを使用して特定の **VLAN** をリッスンするワイルドカード仮想サーバを設定するには

コマンドプロンプトで入力します:

```
1  add cs vserver <name> <serviceType> IPAddress `* Port *` -listenpolicy
       <expression> [-listenpriority <positive_integer>]
2  <!--NeedCopy-->
```

例:

```
1  add cs vserver Vserver-CS-vlan1 ANY * *
2  -listenpolicy "CLIENT.VLAN.ID.EQ(2)"  -listenpriority 10
3  <!--NeedCopy-->
```

構成ユーティリティを使用して特定の **VLAN** をリッスンするワイルドカード仮想サーバを構成するには

[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを設定します。指定した VLAN 上のトラフィックだけを処理するように制限するリッスンポリシーを指定します。

この仮想サーバーを作成したら、基本的な負荷分散の設定の説明に従って、仮想サーバーを 1 つ以上のサービスにバインドします。

Microsoft SQL サーバのバージョン設定を構成する

種類が MSSQL のコンテンツスイッチング仮想サーバーの Microsoft® SQL Server® のバージョンを指定できます。一部のクライアントが Microsoft SQL Server 製品と同じバージョンを実行していないことが予想される場合は、バージョン設定をお勧めします。バージョン設定では、すべての通信がサーバーのバージョンに準拠していることを確認することで、クライアント側接続とサーバー側の接続間の互換性を提供します。

コマンドラインインターフェイスを使用して **Microsoft SQL Server** のバージョンパラメータを設定するには

コマンドプロンプトで次のコマンドを入力して、コンテンツスイッチング仮想サーバーの Microsoft SQL Server バージョンパラメータを設定し、構成を確認します。

- `set cs vserver <name> -mssqlServerVersion <mssqlServerVersion>`
- `show cs vserver <name>`

例

```
> set cs vserver myMSSQLcsvip -mssqlServerVersion 2008R2 Done > show cs
    vserver myMSSQLcsvip myMSSQLcsvip (192.0.2.13:1433) - MSSQL Type:
   CONTENT State: UP . . . . . MSsql Server Version: 2008R2 . . . . .
    . Done >
<!--NeedCopy-->
```

構成ユーティリティを使用して **Microsoft SQL Server** のバージョンパラメーターを設定するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを構成し、プロトコルを MSSQL として指定します。
2. **[**詳細設定**]** で、[サーバーのバージョン] を指定します。

UDP 仮想サーバーの外部 TCP 正常性チェックを有効にする

パブリッククラウドでは、ネイティブロードバランサを第 1 層として使用する場合、NetScaler アプライアンスを第 2 層ロードバランサーとして使用できます。ネイティブロードバランサは、アプリケーションロードバランサ（ALB）またはネットワークロードバランサ（NLB）になります。ほとんどのパブリッククラウドは、ネイティブロードバランサーで UDP ヘルスプローブをサポートしていません。UDP アプリケーションの正常性を監視するために、パブリッククラウドでは、サービスに TCP ベースのエンドポイントを追加することをお勧めします。エンドポイントは UDP アプリケーションの正常性を反映します。

NetScaler アプライアンスは、UDP 仮想サーバーの外部 TCP ベースのヘルスチェックをサポートします。この機能により、コンテンツスイッチング仮想サーバの VIP および設定されたポートに TCP リスナーが導入されます。TCP リスナーは、仮想サーバーのステータスを反映します。

CLI を使用して **UDP** 仮想サーバーの外部 **TCP** ヘルスチェックを有効にするには

コマンドプロンプトで次のコマンドを入力して、TcpProbeport オプションを指定して外部 TCP ヘルスチェックを有効にします。

```
add cs vserver <name> <protocol> <IPAddress> <port> -tcpProbePort <
   tcpProbePort>
<!--NeedCopy-->
```

例:

```
add cs vserver Vserver-CS-1 UDP 10.102.29.161 5002 -tcpProbePort 5000
<!--NeedCopy-->
```

GUI を使用して **UDP** 仮想サーバーの外部 **TCP** ヘルスチェックを有効にするには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを作成します。
2. [追加] をクリックして、仮想サーバーを作成します。
3. [基本設定] ペインの [**TCP** プローブポート] フィールドにポート番号を追加します。
4. [**OK**] をクリックします。

Diameter プロトコルのコンテンツスイッチ

August 15, 2023

Diameter プロトコルのトラフィックでは、NetScaler アプライアンス（または仮想アプライアンス）を、メッセージの内容（メッセージ内の AVP 値）に基づいてパケットの負荷分散と適切な宛先に転送するリレーエージェントとして機能するように構成できます。アプライアンスはアプリケーションレベルの処理を一切行わないため、設定されたコンテンツスイッチングポリシーで指定されているとおりに、すべての直径のアプリケーションに中継サービスを提供します。そのため、クライアントが直径接続を確立すると、アプライアンスはケイパビリティエクスチェンジアンサー（CEA）メッセージでリレーアプリケーション ID をアドバタイズします。コンテンツスイッチング仮想サーバー、負荷分散仮想サーバー、および直径ノードを表すサービスを構成する必要があります。要求がコンテンツスイッチング仮想サーバーに到達すると、仮想サーバーはそのタイプのリクエストに関連するコンテンツスイッチングポリシーを適用します。ポリシーを評価すると、コンテンツスイッチ仮想サーバーは要求を適切な負荷分散仮想サーバーにルーティングし、要求は適切なサービスに送信されます。

直径インタフェースは、異なる直径のノード間を接続します。次のサンプルデプロイでは、Cx と Rx のインターフェイスを使用しています。Cx インターフェイスは CSCF と HSS 間の接続を提供します。Rx インターフェイスは CSCF と PCRF 間の接続を提供します。すべてのメッセージは NetScaler アプライアンスに届きます。メッセージが Cx インターフェイス用か Rx インターフェイス用か、および定義されているコンテンツスイッチングポリシーに応じて、NetScaler は適切な負荷分散サーバープールを選択します。



CSCF=Call Session Control Function
HSS=Home Subscriber Server
PCRF=Policy and Charging Rules Function

構成例

1. エンティティごとに、サービスと負荷分散サーバーを作成し、サービスを仮想サーバーにバインドします。

```
add service svc_pcrf[1-3] 1.1.1.1[1-3] DIAMETER 3868
add service svc_hss[1-3] 1.1.1.2[1-3] DIAMETER 3868
add lb vserver vs_rx DIAMETER -persistenceType DIAMETER -
    persistavpno 263
add lb vserver vs_cx DIAMETER -persistenceType DIAMETER -
    persistavpno 263
bind lb vserver vs_rx svc_pcrf[1-3]
bind lb vserver vs_cx svc_hss[1-3]
<!--NeedCopy-->
```

2. コンテンツスイッチング仮想サーバーと 2 つのアクション (負荷分散仮想サーバーごとに 1 つ) を作成します。2 つのコンテンツスイッチングポリシーを作成し、これらのポリシーをコンテンツスイッチ仮想サーバーにバインドし、各ポリシーの優先順位を指定します。

```
add cs vserver cs_diameter DIAMETER 10.1.1.10 3868
add cs action cx_action -targetLBVserver vs_cx
add cs action rx_action - targetLBvserver vs_rx
add cs policy cx_policy -rule "DIAMETER.REQ.AUTH_APPLICATION_ID.EQ
    (16777216)" -action cx_action
add cs policy rx_policy -rule "DIAMETER.REQ.AUTH_APPLICATION_ID.EQ
    (16777236)" -action rx_action
bind cs vserver cs_diameter -policyName rx_policy -priority 100
bind cs vserver cs_diameter -policyName cx_policy -priority 110
<!--NeedCopy-->
```

コンテンツスイッチセットアップの障害からの保護

December 8, 2023

コンテンツスイッチング仮想サーバーがダウンしたり、過剰なトラフィックを処理できなかったり、その他の理由に

より、コンテンツスイッチングが失敗することがあります。失敗する可能性を減らすには、次の対策を講じてコンテンツスイッチングの設定を障害から保護してください:

バックアップ仮想サーバーの設定

プライマリコンテンツスイッチング仮想サーバーが DOWN または DISABLED とマークされている場合、Citrix ADC アプライアンスは要求をバックアップコンテンツスイッチング仮想サーバーに転送できます。また、サイトの停止またはメンテナンスに関する通知メッセージをクライアントに送信することもできます。バックアップコンテンツスイッチング仮想サーバーはプロキシであり、クライアントには透過的です。

バックアップ仮想サーバーを構成するときに、構成パラメーター Disable Primary When Down を指定できます。これにより、プライマリ仮想サーバーが復旧しても、手動で強制的にプライマリとして引き継ぐまでセカンダリのままになります。これは、バックアップのためにサーバー上のデータベースに加えられた更新を確実に保持したい場合に便利です。これにより、プライマリ仮想サーバーを復元する前にデータベースを同期できます。

コンテンツスイッチング仮想サーバーを作成するとき、または既存のコンテンツスイッチング仮想サーバーのオプションパラメーターを変更するときに、バックアップコンテンツスイッチング仮想サーバーを構成できます。既存のバックアップコンテンツスイッチング仮想サーバー用にバックアップコンテンツスイッチング仮想サーバーを構成して、カスケードされたバックアップコンテンツスイッチング仮想サーバーを作成することもできます。カスケードバックアップコンテンツスイッチング仮想サーバーの最大深度は 10 です。アプライアンスは、稼働中のバックアップコンテンツスイッチング仮想サーバーを検索し、そのコンテンツスイッチング仮想サーバーにアクセスしてコンテンツを配信します。

> 注
>
> コンテンツスイッチ仮想サーバーがバックアップコンテンツスイッチング仮想サーバーとリダイレクト URL の両方で構成されている場合、バックアップコンテンツスイッチング仮想サーバーがリダイレクト URL よりも優先されます。リダイレクトは、プライマリ仮想サーバーとバックアップ仮想サーバーがダウンしている場合に使用されます。

コマンドラインインターフェイスを使用してバックアップコンテンツスイッチング仮想サーバーをセットアップするには

コマンドプロンプトで入力します:

```
set cs vserver <name> -backupVserver <string> -disablePrimaryOnDown (ON|OFF)
<!--NeedCopy-->
```

例

```
set cs vserver Vserver-CS-1 -backupVserver Vserver-CS-2 -disablePrimaryOnDown ON
<!--NeedCopy-->
```


構成ユーティリティを使用してバックアップコンテンツスイッチング仮想サーバーをセットアップするには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを構成し、プロトコルを MYSQL として指定します。
2. [詳細設定] で [保護] を選択し、バックアップ仮想サーバーを指定します。

余分なトラフィックをバックアップ仮想サーバーに転送する

スピルオーバーオプションは、コンテンツスイッチング仮想サーバーへの接続数が設定されたしきい値を超えると、コンテンツスイッチング仮想サーバーに到達した新しい接続をバックアップコンテンツスイッチング仮想サーバーに転送します。しきい値は動的に計算されるか、ユーザーが設定できます。仮想サーバーで確立された接続の数 (TCP の場合) がしきい値と比較されます。接続数がしきい値に達すると、新しい接続はバックアップコンテンツスイッチング仮想サーバーに転送されます。

バックアップコンテンツスイッチング仮想サーバーが設定されたしきい値に達して負荷を処理できない場合、プライマリコンテンツスイッチング仮想サーバーはすべての要求をリダイレクト URL に転送します。プライマリコンテンツスイッチング仮想サーバーでリダイレクト URL が設定されていない場合、後続のリクエストはドロップされます。

コマンドラインインターフェイスを使用して、新しい接続をバックアップ仮想サーバーに転送するようにコンテンツスイッチング仮想サーバーを構成するには

コマンドプロンプトで入力します:

```
set cs vserver <name> -soMethod <methodType> -soThreshold <
    thresholdValue> -soPersistence <persistenceValue> -
    soPersistenceTimeout <timeoutValue>
<!--NeedCopy-->
```

例

```
set cs vserver Vserver-CS-1 -soMethod Connection -soThreshold 1000 -
    soPersistence enabled -soPersistenceTimeout 2
<!--NeedCopy-->
```

構成ユーティリティを使用して、新しい接続をバックアップ仮想サーバーに転送するようにコンテンツスイッチ仮想サーバーを設定するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを構成し、プロトコルを MYSQL として指定します。
2. [詳細設定] で [保護] を選択し、スピルオーバーを設定します。

リダイレクト **URL** の設定

HTTP または HTTPS タイプのコンテンツスイッチング仮想サーバーがダウンまたは無効になっている場合に、Citrix ADC アプライアンスのステータスを通知するようにリダイレクト URL を構成できます。この URL はローカルでもリモートでもかまいません。

リダイレクト URL には、絶対 URL と相対 URL があります。構成したリダイレクト URL に絶対 URL が含まれている場合、着信した HTTP 要求で指定された URL に関係なく、その絶対 URL にリダイレクトされます。構成したリダイレクト URL にドメイン名のみが含まれている場合（相対 URL）、そのドメインに着信 URL を追記した場所にリダイレクトされます。

Citrix では絶対 URL を使用することを推奨しています。つまり、たとえば相対 URL の代わりに、`www.example.com//`で終わる URL です。相対 URL リダイレクトを行うと、脆弱性スキャナーが誤検出を報告する可能性があります。

> 注
>
> コンテンツスイッチ仮想サーバーがバックアップ仮想サーバーとリダイレクト URL の両方で構成されている場合、バックアップ仮想サーバーがリダイレクト URL よりも優先されます。リダイレクト URL は、プライマリ仮想サーバーとバックアップ仮想サーバーがダウンしている場合に使用されます。

リダイレクトが構成されていて、コンテンツスイッチング仮想サーバーが利用できない場合、アプライアンスはユーザーのブラウザに HTTP 302 リダイレクトを発行します。

コマンドラインインターフェイスを使用してコンテンツスイッチング仮想サーバーが利用できない場合のリダイレクト **URL** を構成するには

コマンドプロンプトで入力します：

```
set cs vserver <name> -redirectURL <URLValue>
<!--NeedCopy-->
```

例

```
set cs vserver Vserver-CS-1 -redirectURL http://www.newdomain.com/
    mysite/maintenance
<!--NeedCopy-->
```

構成ユーティリティを使用してコンテンツスイッチング仮想サーバーが利用できない場合のリダイレクト **URL** を構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを構成し、プロトコルを MYSQL として指定します。
2. [詳細設定] で [保護] を選択し、リダイレクト URL を指定します。

状態更新オプションの設定

コンテンツスイッチング機能により、ユーザーに表示される特定のコンテンツに基づいて、クライアント要求を複数のサーバーに分散できます。コンテンツスイッチングを効率的に行うために、コンテンツスイッチング仮想サーバーはコンテンツタイプに応じてトラフィックを負荷分散仮想サーバーに分散し、負荷分散仮想サーバーは指定された負荷分散方法に従ってトラフィックを物理サーバーに配信します。

トラフィック管理を円滑に行うためには、コンテンツスイッチング仮想サーバーが負荷分散仮想サーバーのステータスを把握することが重要です。状態更新オプションは、コンテンツスイッチ仮想サーバーにバインドされている負荷分散仮想サーバーがダウンしている場合に、コンテンツスイッチ仮想サーバーを DOWN とマークするのに役立ちます。負荷分散仮想サーバーは、それにバインドされているすべての物理サーバーが DOWN とマークされている場合、DOWN とマークされます。

状態更新が無効になっている場合**:**

コンテンツスイッチ仮想サーバーのステータスは UP とマークされます。稼働中のバウンド負荷分散仮想サーバーがなくても稼働したままです。

状態更新が有効な場合**:**

新しいコンテンツスイッチング仮想サーバーを追加すると、最初はステータスが DOWN と表示されます。ステータスが UP の負荷分散仮想サーバーをバインドすると、コンテンツスイッチング仮想サーバーのステータスは UP になります。

複数の負荷分散仮想サーバーがバインドされており、そのうちの 1 つがデフォルトとして指定されている場合、コンテンツスイッチング仮想サーバーのステータスはデフォルトの負荷分散仮想サーバーのステータスを反映します。

デフォルトとして指定されていない状態で複数の負荷分散仮想サーバーがバインドされている場合、バインドされたすべての負荷分散仮想サーバーが稼働している場合にのみ、コンテンツスイッチング仮想サーバーのステータスは UP とマークされます。

コマンドラインインターフェイスを使用して状態更新オプションを構成するには

コマンドプロンプトで入力します:

```
add cs vserver <name> <protocol> <ipAddress> <port> -stateUpdate
   ENABLED
<!--NeedCopy-->
```

例

```
 add cs vserver csw_vserver HTTP 10.18.250.154 80 -stateupdate ENABLED
    -cltTimeout 180
<!--NeedCopy-->
```


構成ユーティリティを使用して状態更新オプションを構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを構成し、プロトコルを MYSQL として指定します。
2. [詳細設定] で [トラフィック設定] を選択し、次に [状態更新] を選択します。

サージキューのフラッシュ

物理サーバーが要求を大量に受信すると、現在接続しているクライアントへの応答が遅くなり、ユーザーは不満を抱き、不満を抱きます。多くの場合、オーバーロードにより、クライアントはエラーページを受信します。このような過負荷を回避するために、Citrix ADC アプライアンスには、サービスへの新しい接続を確立できる速度を制御するサージ保護などの機能が用意されています。

アプライアンスは、クライアントと物理サーバ間の接続の多重化を行います。サーバー上のサービスにアクセスするクライアント要求を受信すると、アプライアンスはサーバーへの接続がすでに確立されていて、空いているかを探します。空き接続が見つかった場合、その接続を使用してクライアントとサーバー間の仮想リンクを確立します。既存の空き接続が見つからない場合、アプライアンスはサーバーとの新しい接続を確立し、クライアントとサーバー間の仮想リンクを確立します。ただし、アプライアンスがサーバーとの新しい接続を確立できない場合、クライアント要求をサージキューに送信します。負荷分散またはコンテンツスイッチング仮想サーバーにバインドされているすべての物理サーバーがクライアント接続の上限（最大クライアント値、サージ保護しきい値、またはサービスの最大容量）に達すると、アプライアンスはどのサーバーとも接続を確立できなくなります。サージ保護機能は、サージキューを使用して物理サーバーとの接続を開く速度を調整します。アプライアンスは、仮想サーバにバインドされたサービスごとに異なるサージキューを維持します。

サージキューの長さは、アプライアンスが接続を確立できない要求が来るたびに増加し、キュー内の要求がサーバーに送信されるか、要求がタイムアウトしてキューから削除されるたびに長くなります。

サービスまたはサービスグループのサージキューが長すぎる場合は、そのサージキューをフラッシュする必要があります。特定のサービスまたはサービスグループ、または負荷分散仮想サーバーにバインドされているすべてのサービスとサービスグループのサージキューをフラッシュできます。サージキューをフラッシュしても、既存の接続には影響しません。サージキューに存在するリクエストのみが削除されます。これらのリクエストについては、クライアントは新たなリクエストを行う必要があります。

コンテンツスイッチング仮想サーバーのサージキューをフラッシュすることもできます。コンテンツスイッチング仮想サーバーが特定の負荷分散仮想サーバーにいくつかの要求を転送し、負荷分散仮想サーバーも他の要求を受信した場合、コンテンツスイッチング仮想サーバーのサージキューをフラッシュすると、このコンテンツスイッチング仮想サーバーから受信した要求のみがフラッシュされます。負荷分散仮想サーバーのサージキュー内の他の要求はフラッシュされません。

> 注
>
> キャッシュリダイレクト、認証、VPN、または GSLB 仮想サーバーまたは GSLB サービスのサージキューをフラッシュすることはできません。

> [ソース IP の使用（USIP）] が有効になっている場合は、サージ保護機能を使用しないでください。

コマンドラインインターフェイスを使用してサージキューをフラッシュするには

flush ns SurgeQ コマンドは次のように機能します：

- サージキューをフラッシュする必要があるサービス、サービスグループ、または仮想サーバーの名前を指定できます。
- コマンドの実行中に名前を指定すると、指定したエンティティのサージキューがフラッシュされます。複数のエンティティが同じ名前を持つ場合、アプライアンスはこれらすべてのエンティティのサージキューをフラッシュします。
- コマンドの実行中にサービスグループの名前、サーバー名、およびポートを指定すると、アプライアンスは指定されたサービスグループメンバーのみのサージキューをフラッシュします。
- サービスグループの名前 (<name>) を指定せずにサービスグループメンバー (<serverName> および<port >) を直接指定することはできません。また、<serverName>を指定しないで<port>を指定することもできません。特定のサービスグループメンバーのサージキューをフラッシュする場合は、<serverName>および<port>を指定します。
- 名前を指定せずにコマンドを実行すると、アプライアンスはアプライアンスに存在するすべてのエンティティのサージキューをフラッシュします。
- サービスグループメンバーがサーバ名で識別される場合は、このコマンドでサーバ名を指定する必要があります。IP アドレスは指定できません。

コマンドプロンプトで入力します：

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>].
<!--NeedCopy-->
```

例

```
1. flush ns surgeQ  -name SVC1ANZGB  -serverName 10.10.10.1 80
The above command flushes the surge queue of the service or virtual
    server that is named SVC1ANZGB and has IP address as 10.10.10

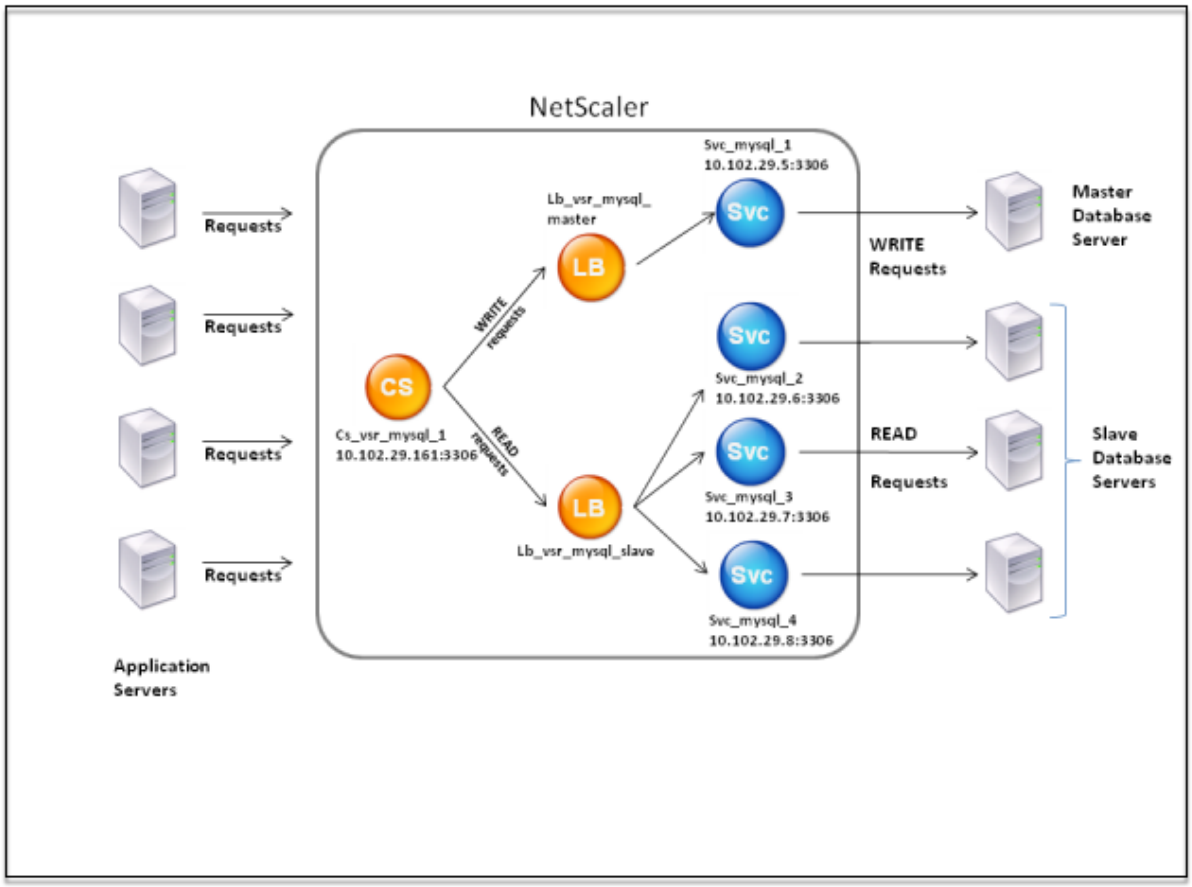
2. flush ns surgeQ
The above command flushes all the surge queues on the appliance.
<!--NeedCopy-->
```

構成ユーティリティを使用してサージキューをフラッシュするには

[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを選択して、[アクション] リストで [フラッシュサージキュー] を選択します。

コンテンツスイッチング設定の管理

August 15, 2023

コンテンツスイッチングの設定後、定期的な変更が必要になる場合があります。オペレーティングシステムまたはソフトウェアが更新された場合、またはハードウェアが消耗して交換された場合は、セットアップを停止する必要がある場合があります。セットアップの負荷が増加し、より多くのリソースが必要になる場合があります。また、パフォーマンスを向上させるために構成を変更することもできます。

これらのタスクでは、コンテンツスイッチ仮想サーバーからのポリシーのバインド解除、またはコンテンツスイッチ仮想サーバーの無効化または削除が必要になる場合があります。設定を変更した後、サーバを再度有効にし、ポリシーを再バインドする必要があります。また、仮想サーバーの名前を変更することもできます。

コンテンツスイッチ仮想サーバーからのポリシーのバインド解除

仮想サーバーからコンテンツスイッチングポリシーをバインド解除すると、仮想サーバーは要求を送信する場所を決定するときにそのポリシーを含まなくなります。

CLI を使用してコンテンツスイッチ仮想サーバーからポリシーをバインド解除するには

コマンドプロンプトで入力します。

```
unbind cs vserver <name> -policyname <string>
```

例:

```
unbind cs vserver Vserver-CS-1 -policyname Policy-CS-1
```

GUI を使用してコンテンツスイッチ仮想サーバーからポリシーをバインド解除するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [ポリシー] セクションをクリックし、ポリシーを選択して、[バインド解除] をクリックします。

コンテンツスイッチング仮想サーバーの削除

通常、コンテンツスイッチ仮想サーバーを削除するのは、仮想サーバーが不要になった場合のみです。コンテンツスイッチング仮想サーバーを削除すると、NetScaler ADC アプライアンスはまずコンテンツスイッチング仮想サーバーからすべてのポリシーをバインド解除し、次にコンテンツスイッチ仮想サーバーを削除します。

CLI を使用してコンテンツスイッチング仮想サーバーを削除するには

コマンドプロンプトで入力します。

```
rm cs vserver <name>
```

例:

```
rm cs vserver Vserver-CS-1
```

GUI を使用してコンテンツスイッチング仮想サーバーを削除するには

[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを選択して [削除] をクリックします。

コンテンツスイッチング仮想サーバーの無効化と再有効化

コンテンツスイッチ仮想サーバーは、作成時にデフォルトで有効になります。メンテナンスのためにコンテンツスイッチング仮想サーバーを無効にすることができます。コンテンツスイッチング仮想サーバーを無効にすると、コンテンツスイッチング仮想サーバーの状態は [サービス外] に変わります。アウトオブサービス中は、コンテンツスイッチング仮想サーバーは要求に応答しません。

CLI を使用して仮想サーバーを無効または再度有効にするには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `disable cs vserver <name>`
- `enable cs vserver <name>`

例:

```
disable cs vserver Vserver-CS-1
enable cs vserver Vserver-CS-1
```

GUI を使用して仮想サーバーを無効または再度有効にするには

[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを選択し、[アクション] リストで [有効化] または [無効化] を選択します。

コンテンツスイッチング仮想サーバーの名前の変更

コンテンツスイッチング仮想サーバーの名前は、バインド解除せずに変更できます。新しい名前は、NetScaler ADC 構成の影響を受けるすべての部分に自動的に反映されます。

CLI を使用して仮想サーバーの名前を変更するには

コマンドプロンプトで入力します。

```
rename cs vserver <name> <newName>
```

例:

```
`rename cs vserver Vserver-CS-1 Vserver-CS-2`
```

GUI を使用して仮想サーバーの名前を変更するには

[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを選択して、[アクション] リストで [名前の変更] を選択します。

コンテンツスイッチングポリシーの管理

既存のポリシーを変更するには、規則を設定するか、ポリシーの URL を変更するか、ポリシーを削除できます。既存の高度なコンテンツスイッチングポリシーの名前を変更することもできます。URL に基づいて異なるポリシーを作成できます。URL ベースのポリシーは、次の表で説明するように、異なるタイプにすることができます。

詳細については、「URL ベースのポリシーの例」を参照してください。

> 注
>
> ルールベースのコンテンツスイッチングは、従来のポリシー式または高度なポリシー式を使用して構成できます。

CLI を使用してポリシーを変更、削除、または名前変更するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `set cs policy <policyName> [-domain <domainValue>] [-rule <ruleValue>] [-url <URLValue>]`
- `rm cs policy <policyName>`
- `rename cs policy <policyName> <newPolicyName>`

例:

```
set cs policy-CS-1 -domain "www.domainxyz.com"

set cs policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(22).EQ(10.100.148.0)"

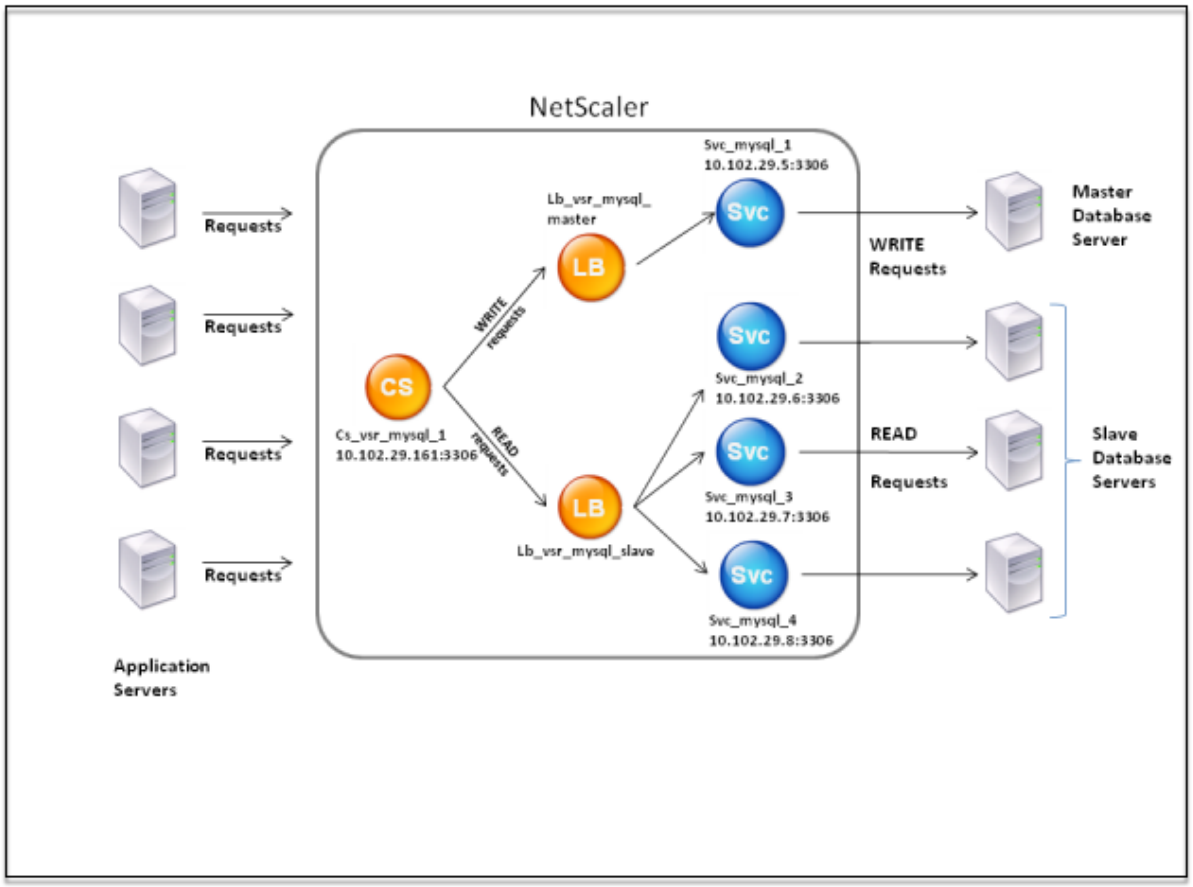
set cs policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2010 Jun,GMT 2010 Jul)"

set cs policy-CS-1 -url /sports/*
```



```
 8
 9  rename cs policy-CS-1 Policy-CS-11
10
11  rm cs policy-CS-1
```

GUI を使用してポリシーを変更、削除、または名前変更するには

1. **Traffic Management > Content Switching > Policies** に移動します。
2. ポリシーを選択し、削除または編集するか、[アクション] リストで [名前の変更] をクリックします。

クライアント接続の管理

December 8, 2023

クライアント接続を効率的に管理するために、NetScaler アプライアンスのコンテンツスイッチ仮想サーバーが次の機能を使用するように構成できます。

- **ICMP** レスポンスの設定。NetScaler アプライアンスは、設定に従って PING リクエストに ICMP 応答を送信するように構成できます。仮想サーバーに対応する IP アドレスで ICMP RESPONSE を VSVR_CNTRLD に設定し、仮想サーバーでは ICMP 仮想サーバーの RESPONSE を設定します。
  仮想サーバーでは、次の設定を行うことができます。
  - すべての仮想サーバーで ICMP 仮想サーバー応答を PASSIVE に設定すると、NetScaler アプライアンスは常に応答します。
  - すべての仮想サーバーで ICMP 仮想サーバーの応答を ACTIVE に設定すると、1 つの仮想サーバーが稼働していても ADC アプライアンスは応答します。
  - ICMP 仮想サーバーの応答を一部で ACTIVE に設定し、他の仮想サーバーを PASSIVE に設定すると、ACTIVE に設定された 1 つの仮想サーバーが稼働していても ADC アプライアンスは応答します。

クライアント要求をキャッシュにリダイレクトする

NetScaler キャッシュリダイレクト機能は、HTTP 要求をキャッシュにリダイレクトします。キャッシュリダイレクト機能を適切に実装することで、HTTP リクエストへの応答の負担を大幅に軽減し、Web サイトのパフォーマンスを向上させることができます。

キャッシュは、頻繁に要求された HTTP コンテンツを格納します。仮想サーバーでキャッシュリダイレクトを構成すると、NetScaler アプライアンスはキャッシュ可能な HTTP リクエストをキャッシュに送信し、キャッシュ不可能な HTTP リクエストを元の Web サーバーに送信します。キャッシュリダイレクトの詳細については、「キャッシュリダイレクト」を参照してください。

CLI を使用して仮想サーバーでキャッシュリダイレクトを構成するには

コマンドプロンプトで入力します:

```
set cs vserver <name> -cacheable <Value>
```

例 `set cs vserver Vserver-CS-1 -cacheable yes`

GUI を使用して仮想サーバーのキャッシュリダイレクトを構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [トラフィック設定] を選択し、[キャッシュ可能] を選択します。

仮想サーバー接続の遅延クリーンアップの有効化

特定の条件下では、サービスまたは仮想サーバーがダウンとマークされたときに既存の接続を終了するようにダウンステートフラッシュ設定を構成できます。既存の接続を終了すると、リソースが解放され、場合によっては過負荷の負荷分散設定の回復が速くなります。

CLI を使用して仮想サーバーでダウン状態のフラッシュ設定を構成するには

コマンドプロンプトで入力します:

```
set cs vserver <name> -downStateFlush <Value>
```

例

```
set cs vserver Vserver-CS-1 -downStateFlush enabled
<!--NeedCopy-->
```

GUI を使用して仮想サーバーのダウンステートフラッシュ設定を構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [トラフィック設定] を選択し、[ダウンステートフラッシュ] を選択します。

リダイレクト用のポートとプロトコルの書き換え

仮想サーバーとそれにバインドされているサービスは、異なるポートを使用する場合があります。サービスが HTTP 接続にリダイレクトで応答する場合、リダイレクトが正常に実行されるように、ポートとプロトコルを変更するように NetScaler アプライアンスを構成する必要がある場合があります。そのためには、RedirectPortRewrite 設定を有効にして構成します。

CLI を使用して仮想サーバーで **HTTP** リダイレクトを構成するには

コマンドプロンプトで入力します:

```
set cs vserver <name> -redirectPortRewrite <Value>
```

例

```
set cs vserver Vserver-CS-1 -redirectPortRewrite enabled
<!--NeedCopy-->
```

GUI を使用して仮想サーバーで **HTTP** リダイレクトを構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [トラフィック設定] を選択し、[書き換え] を選択します。

リクエストヘッダーへの仮想サーバーの **IP** アドレスとポートの挿入

同じサービス上の異なるアプリケーションと通信する複数の仮想サーバーがある場合は、そのサービスに送信される HTTP 要求に適切な仮想サーバーの IP アドレスとポート番号を追加するように NetScaler アプライアンスを構成する必要があります。この設定により、サービスで実行されているアプリケーションが、要求を送信した仮想サーバーを識別できるようになります。

プライマリ仮想サーバーがダウンしていてバックアップ仮想サーバーが稼働している場合、バックアップ仮想サーバーの構成設定がクライアント要求に追加されます。要求がプライマリ仮想サーバーからのものかバックアップ仮想サーバーからのものかに関係なく、同じヘッダータグを追加する場合は、両方の仮想サーバーで必要なヘッダータグを構成する必要があります。

> 注
>
> このオプションは、ワイルドカード仮想サーバーまたはダミー仮想サーバーではサポートされていません。

CLI を使用して仮想サーバーの **IP** アドレスとポートをクライアントリクエストに挿入するには

コマンドプロンプトで入力します:

```
set cs vserver <name> -insertVserverIPPort <vServerIPPORT>
```

例

```
set cs vserver Vserver-CS-1 -insertVserverIPPort 10.201.25.136:80
<!--NeedCopy-->
```


GUI を使用して仮想サーバーの **IP** アドレスとポートをクライアントリクエストに挿入するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [トラフィック設定] を選択し、[仮想サーバー IP ポート挿入] リストで [VIPADDR] または [V6TOV4MAPPING] を選択し、仮想サーバーの IP ポート挿入値にポートヘッダーを指定します。

アイドル状態のクライアント接続のタイムアウト値の設定

設定したタイムアウト期間が経過すると、アイドル状態のクライアント接続をすべて終了するように仮想サーバーを構成できます。この設定を構成すると、NetScaler アプライアンスは指定した時間待機し、その時間以降にクライアントがアイドル状態になると、クライアント接続を閉じます。

CLI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

コマンドプロンプトで入力します:

```
set cs vserver <name> -cltTimeout <Value>
```

例

```
set cs vserver Vserver-CS-1 -cltTimeout 100
<!--NeedCopy-->
```

GUI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2.「詳細設定」で、「トラフィック設定」を選択し、「クライアントアイドルタイムアウト」の値を指定します。

4 タプルとレイヤー **2** の接続パラメーターによる接続の識別

コンテンツスイッチング仮想サーバーの L2Conn オプションを設定できるようになりました。L2Conn オプションを設定すると、コンテンツスイッチング仮想サーバーへの接続は 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) とレイヤー 2 接続パラメーターの組み合わせによって識別されます。レイヤ 2 接続パラメータは、MAC アドレス、VLAN ID、およびチャネル ID です。

CLI を使用してコンテンツスイッチング仮想サーバーの **L2Conn** オプションを設定するには

コマンドラインで次のコマンドを入力して、コンテンツスイッチ仮想サーバーの L2Conn パラメーターを構成し、構成を確認します。


```
    -  set cs vserver <name> -l2Conn (\*\*ON\*\* | \*\*OFF\*\*)
    -  show cs vserver <name>
<!--NeedCopy-->
```

例

```
> set cs vserver mycsvserver -l2Conn ON
Done
> show cs vserver mycsvserver
     mycsvserver (192.0.2.56:80) - HTTP   Type: CONTENT
     State: UP
          . . .
          . . .
     L2Conn: ON Case Sensitivity: ON
          . . .
          . . .
Done
>
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチング仮想サーバーの **L2Conn** オプションを設定するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [トラフィック設定] を選択し、[レイヤ **2** パラメータ] を選択します。

コンテンツスイッチ仮想サーバーのパーシステンスサポート

August 15, 2023

アプリケーションはモノリシックアーキテクチャからマイクロサービスアーキテクチャに移行しています。マイクロサービスアーキテクチャでは、同じアプリケーションの異なるバージョンが共存できます。NetScaler アプライアンスは、アプリケーションの継続的デプロイをサポートする必要があります。これは、Canary デプロイを実行するプラットフォーム (Spinnaker など) によって実現されます。継続的デプロイのセットアップでは、アプリケーションの新しいバージョンが自動的にデプロイされ、アプリケーションが安定してトラフィックを完全に処理できるようになるまで、段階的にクライアントトラフィックにさらされます。また、クライアントへのサービスが中断されない必要があります。

NetScaler のコンテンツスイッチング機能により、NetScaler アプライアンスは、コンテンツスイッチング仮想サーバーにバインドされたポリシーに基づいて、クライアント要求を複数の負荷分散仮想サーバーに分散できます。

継続的なデプロイでは、コンテンツスイッチングを使用して、さまざまなバージョンのアプリケーションを提供する負荷分散仮想サーバーを選択します。

コンテンツスイッチングでは、コンテンツスイッチングポリシーの変更により、特定のアプリケーションバージョン用の負荷分散仮想サーバーの選択が実行時に変わります。この移行中、一部のセッションに古いバージョンのアプリケーションが存在する場合、そのようなトラフィックは引き続き古いバージョンでのみ処理する必要があります。この要件に対応するため、NetScaler アプライアンスはコンテンツスイッチング仮想サーバーの背後にある複数の負荷分散グループにわたって永続性を維持します。コンテンツスイッチング仮想サーバーの永続性により、クライアントをあるバージョンから別のバージョンにシームレスに移行できます。

コンテンツスイッチング仮想サーバーでサポートされているパーシステンスタイプ

コンテンツスイッチ仮想サーバーでは、次のパーシステンスタイプがサポートされています。

持続性タイプ	説明
接続元 IP	ソース **IP**。同じクライアント IP アドレスからの接続は、同じ永続セッションの一部です。詳細については、「送信元 IP アドレスの永続性」を参照してください。
HTTP クッキー	**COOKIEINSERT**. 同じ HTTP Cookie ヘッダーを持つ接続は、同じ持続性セッションの一部です。NetScaler アプライアンスが挿入する Cookie の形式は、**NSC_ =** です。**NSC_XXXX** は仮想サーバー名から派生した仮想サーバー **ID** です。詳細については、「HTTP Cookie パーシステンス」を参照してください。
SSL Session ID	**SSLSESSION**. 同じ SSL セッション ID を持つ接続は、同じ持続性セッションの一部です。詳細については、「SSL セッション ID の永続性」を参照してください。

HTTP Cookie に基づくパーシステンスに対して、タイムアウト値を設定できます。タイムアウト値を 0 に設定すると、使用されている HTTP Cookie バージョンに関係なく、ADC アプライアンスは有効期限を指定しなくなります。その場合、有効期限はクライアントソフトウェアによって異なり、このようなクッキーはソフトウェアが動作している場合にのみ有効です。

構成した永続性のタイプに応じて、仮想サーバーは、NetScaler アプライアンスのメモリ量によって定められた制限まで、250,000 の同時持続接続または任意の数の永続接続のいずれかをサポートできます。次の表は、どのタイプのパーシスタンスが各カテゴリに分類されるかを示しています。

持続性タイプ	サポートされる同時持続接続数
ソース IP、SSL セッション ID	250,000

持続性タイプ	サポートされる同時持続接続数
HTTP クッキー	メモリの上限。CookieInsert では、タイムアウトが 0 でない場合、接続数はメモリによって制限されます。

パーシスタンスには、特定のタイプの仮想サーバーに固有のものもあります。次の表は、各タイプのパーシスタンスを一覧表示し、どのタイプのパーシスタンスがどのタイプの仮想サーバーでサポートされているかを示しています。

持続性タイプ	HTTP	HTTPS	TCP	UDP/IP	SSL_BridgeTCP		RTSP	SIP_UDP
SOURCEIP	はい	はい	はい	はい	はい	はい	いいえ	いいえ
COOKIEINSERT	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
SSLSESSION	いいえ	はい	いいえ	いいえ	はい	はい	いいえ	いいえ

バックアップ永続性サポート

Cookie パーシステンスタイプが失敗した場合に、ソース IP パーシステンスタイプをバックアップパーシステンスタイプとして使用するようにコンテンツスイッチング仮想サーバーを構成できます。マイクロサービスアーキテクチャでの Canary デプロイに役立ちます。

Cookie パーシステンスタイプが失敗した場合、クライアントブラウザがリクエストで Cookie を返さない場合にのみ、アプライアンスはソース IP ベースのパーシスタンスにフォールバックします。ただし、ブラウザが Cookie（パーシステンスクッキーである必要はありません）を返した場合、そのブラウザは Cookie をサポートしていると見なされるため、バックアップパーシスタンスはトリガーされません。

バックアップ永続性のタイムアウト値を設定することもできます。タイムアウトは、永続セッションが有効な期間です。

コンテンツスイッチング仮想サーバーでのパーシスタンスの仕組み

シナリオ **1:** 永続性のないコンテンツスイッチング仮想サーバー

次の例は、コンテンツスイッチング仮想サーバーを永続化せずにアプリケーションの複数のバージョンをデプロイする方法を示しています。



クライアント C1 がアプリケーションに要求を送信すると、その要求は NetScaler アプライアンスのコンテンツスイッチ仮想サーバーに送信されます。コンテンツスイッチング仮想サーバーはポリシーを評価し、アプリケーションのバージョン v1 を提供している負荷分散仮想サーバー (LB1) に要求を転送します。



アプリケーションの新しいバージョン v2 がデプロイされ、一部のユーザーに公開する必要があるとします。v2 バージョンを提供する新しい負荷分散仮想サーバー (LB2) は、適切なコンテンツスイッチングポリシーによってコンテンツスイッチング仮想サーバーにバインドされます。

クライアント C1 が新しい要求を送信すると、ポリシーが再度評価され、要求は負荷分散仮想サーバー LB2 に転送されます。そのため、ステートフルアプリケーションのトランザクションは、複数のバージョンのアプリケーションをデプロイすると失敗します。

シナリオ **2:** 永続性を備えたコンテンツスイッチング仮想サーバー

次の例は、永続性のあるコンテンツスイッチング仮想サーバーを使用して、アプリケーションの複数のバージョンを展開する方法を示しています。



クライアント C1 がアプリケーションに要求を送信すると、その要求は NetScaler アプライアンスのコンテンツスイッチ仮想サーバーに送信されます。コンテンツスイッチング仮想サーバーはポリシーを評価し、永続セッションエントリを作成し、アプリケーションのバージョン v1 を提供している負荷分散仮想サーバー LB1 に要求を転送します。



同じクライアント C1 がアプリケーションを再度要求し、その要求が NetScaler アプライアンスのコンテンツスイッチ仮想サーバーに送信されます。永続セッションの検索が行われ、負荷分散仮想サーバー LB1 が既存の永続セッションから取得され、要求が LB1 に転送されます。このソリューションでは既存のトランザクションが中断されないため、アプリケーションのステートフルな性質が維持されます。



新しいクライアント C2 を考えてみましょう。このクライアントには既存のパーシスタンスセッションがないため、新しいリクエスト C2 はポリシー評価を通じて新しいバージョンのアプリケーションに送信されます。これにより、ステートフルさを損なうことなく、アプリケーションの新しいバージョンを正常にロールアウトできます。

パーシスタンスサポートにより、特にステートフルアプリケーションの場合、お客様は既存のトランザクションに影響を与えることなく、複数のコンテンツや異なるバージョンのアプリケーションをシームレスにデプロイできます。絵にしつこさがなければできません。

CLI を使用してコンテンツスイッチ仮想サーバーのパーシステンスタイプを設定します

コマンドプロンプトで入力します。

```
set cs vserver <name> -PersistenceType <type> [-timeout <integer>]
<!--NeedCopy-->
```

例:

```
set cs vserver Vserver-CS-1 -persistenceType SOURCEIP -timeout 60
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチ仮想サーバーのパーシステンスタイプを設定します

1. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、[追加] をクリックします。
2. 基本設定で、パーシステンスの詳細を設定します。

トラブルシューティング

September 25, 2023

構成後にコンテンツスイッチング機能が期待どおりに機能しない場合は、いくつかの一般的なツールを使用して NetScaler リソースにアクセスし、問題を診断できます。

コンテンツスイッチングのトラブルシューティング用リソース

最良の結果を得るには、次のリソースを使用して NetScaler アプライアンスのコンテンツスイッチングの問題をトラブルシューティングしてください。

- 設定ファイル
- 問題が発生したときに生成された`newnslog`ファイル
- トレースファイル
- お客様のネットワーク設定のためのネットワークトポロジー図
- リリースノート、Knowledge Center 記事、製品ドキュメントなどの NetScaler ドキュメント。

上記のリソースに加えて、次のツールを使用するとトラブルシューティングが容易になります。

- `iehttpheaders` または同様のユーティリティ
- NetScaler トレースファイル用にカスタマイズされた Wireshark アプリケーション
- コマンドラインアクセス用の SSH ユーティリティ
- コンソールにアクセスするためのハイパーターミナルユーティリティ

コンテンツスイッチングに関する問題のトラブルシューティング

最も一般的なコンテンツスイッチングの問題には、コンテンツスイッチング機能がまったく機能しない、または断続的にしか機能しない、および Service Unavailable という応答があります。

- 問題

  コンテンツスイッチング機能が機能していません。

  解像度

  次のように設定を確認します。

  - アプライアンスにコンテンツスイッチングのライセンスが付与されていることを確認します。
  - この機能が有効になっていることを確認します。
  - 構成ファイルから、有効なコンテンツスイッチングポリシーが負荷分散仮想サーバーに正しくバインドされていることを確認します。

- 問題

  クライアントは 503-サービスを利用できないという応答を受け取ります。

  解像度

  - URL とポリシーバインディングを確認してください。設定したポリシーがどれも評価されず、デフォルトの負荷分散仮想サーバーが定義されておらず、コンテンツスイッチング仮想サーバーにバインドされていない場合、クライアントは 503 応答を受け取ります。
  - 設定から、ポリシーと URL がクライアントからアクセスされていることを確認します。
  - すべてのタイプのリクエストについて、それぞれのポリシーが評価されていることを確認してください。ポリシーが評価されない場合は、ポリシー表現を確認し、必要に応じて更新してください。
  - URL と HTTP リクエストおよびレスポンスヘッダーを確認します。そのためには、`HTTPHeader` トレースを記録し、必要に応じてアプライアンスとクライアントのパケットトレースを記録します。

- 問題

  コンテンツスイッチング機能が断続的に期待どおりに動作しない。

  解像度

  - セットアップのネットワークトポロジー図 (可能な場合) を調べて、クライアントとサーバーの間に設置されているさまざまなデバイスを理解してください。
  - 設定とポリシーのバインディングを確認してください。ポリシー表現の URL がクライアントリクエストの URL と一致することを確認してください。
  - ポリシーに適切な優先順位が割り当てられていることを確認します。ポリシーに誤った優先順位や優先順位が割り当てられていると、問題が発生する可能性があります。
  - 次のコマンドを実行して、コマンド出力のポリシー選択カウンタのバインディングと値を確認します。

    ```
    show cs vserver \<CS VServer\>

    show cs policy \<CS Policy\>

    stat cs vserver \<CS VServer\>
    ```

  - `iehttpheaders` または同様のユーティリティを使用して、リクエストまたはレスポンスの HTTP ヘッダーが問題への何らかの指針を提供しているかどうかを判断します。
  - リリースノートと Knowledge Center 記事を確認してください。
  - それでも問題が解決しない場合は、Citrix のテクニカルサポートに連絡して、詳細な調査を依頼してください。

DataStream

August 15, 2023

NetScaler DataStream 機能は、送信される SQL クエリに基づいてリクエストを分散することにより、データベースレイヤーでリクエストを切り替えるインテリジェントなメカニズムを提供します。

NetScaler アプライアンスをデータベースサーバーの前に導入すると、アプリケーションサーバーと Web サーバーからのトラフィックが最適に分散されます。管理者は、SQL クエリ内の情報、データベース名、ユーザー名、文字セット、およびパケットサイズに基づいてトラフィックをセグメント化できます。

負荷分散アルゴリズムに基づいて要求を切り替えるように負荷分散を設定できます。または、SQL クエリパラメータに基づいて決定を行うようにコンテンツスイッチングを設定して、切り替え条件を詳しく説明することもできます。さらに、データベースサーバーの状態を追跡するようにモニターを構成できます。

> 注:
>
> NetScaler DataStream は、MySQL データベースと MS SQL データベースでのみサポートされています。サポートされているプロトコルバージョン、文字セット、特殊クエリ、およびトランザクションについては、「DataStream Reference」を参照してください。

DataStream 仕組み

DataStream では、ADC アプライアンスはアプリケーションまたは Web サーバーとデータベースサーバーの間にインラインで配置されます。アプライアンスでは、データベースサーバーはサービスによって表されます。

一般的な DataStream デプロイメントは、次の図で説明されているエンティティで構成されています。

図 1: DataStream エンティティモデル



この図に示すように、DataStream 構成には次のものを含めることができます。

- オプションのコンテンツスイッチング仮想サーバー (CS)。
- 負荷分散仮想サーバー (LB1 と LB2) で構成される負荷分散設定。
- サービス (Svc1、Svc2、Svc3、および Svc4)。
- コンテンツスイッチングポリシー (オプション)。

クライアント（アプリケーションまたは Web サーバー）は、NetScaler アプライアンスで構成されたコンテンツスイッチ仮想サーバー（CS）の IP アドレスに要求を送信します。次に、アプライアンスは、アプライアンスに設定されているデータベースユーザー認証情報を使用してクライアントを認証します。コンテンツスイッチング仮想サーバー (CS) は、関連するコンテンツスイッチングポリシーを要求に適用します。ポリシーを評価した後、コンテンツスイッチ仮想サーバー (CS) は要求を適切な負荷分散仮想サーバー (LB1 または LB2) にルーティングします。次に、負荷分散仮想サーバーは、負荷分散アルゴリズムに基づいて適切なデータベースサーバー（アプライアンス上のサービスで表される）に要求を分散します。NetScaler アプライアンスは、同じデータベースユーザー資格情報を使用してデータベースサーバーとの接続を認証します。

コンテンツスイッチング仮想サーバーがアプライアンス上に構成されていない場合、クライアント（アプリケーションまたは Web サーバー）は、アプライアンス上に構成された負荷分散仮想サーバーに要求を送信します。NetScaler アプライアンスは、アプライアンスで構成されたデータベースユーザー資格情報を使用してクライアントを認証し、同じ資格情報を使用してデータベースサーバーとの接続を認証します。負荷分散仮想サーバーは、負荷分散アルゴリ

ズムに従って要求をデータベースサーバーに分散します。データベース切り替えに最も効果的な負荷分散アルゴリズムは、最小接続方式です。

DataStream は接続多重化を使用して、同じサーバー側接続で複数のクライアント側リクエストを送信できるようにします。次の接続プロパティが考慮されます。

- ユーザー名
- データベース名
- パケットサイズ
- 文字セット

データベースユーザーを構成する

August 15, 2023

データベースでは、接続は常にステートフルです。つまり、接続が確立されたら認証を受ける必要があります。

NetScaler アプライアンスでデータベースのユーザー名とパスワードを設定します。たとえば、データベースに John というユーザーを設定している場合、ADC にもユーザー John を設定する必要があります。ADC にデータベースのユーザー名とパスワードを追加すると、`nsconfig` それらがファイルに追加されます。

> 注
>
> 名前は大文字と小文字が区別されます。

ADC は、これらのユーザー認証情報を使用してクライアントを認証し、次にデータベースサーバーとのサーバー接続を認証します。

CLI を使用してデータベースユーザーを追加する

コマンドプロンプトで、次のように入力します。

```
add db user <username> - password <password>
```

例:

```
1 add db user nsdbuser -password dd260427edf
2 <!--NeedCopy-->
```

GUI を使用してデータベースユーザーを追加する

[システム] **>** [ユーザー管理] **>** [データベースユーザー] に移動し、データベースユーザーを設定します。

データベースサーバー上のデータベースユーザーのパスワードを変更した場合は、ADC アプライアンスに設定されている対応するユーザーのパスワードをリセットする必要があります。

CLI を使用してデータベースユーザーのパスワードをリセットする

コマンドプロンプトで、次のように入力します。

```
set db user <username> -password <password>
<!--NeedCopy-->
```

例:

```
set db user nsdbuser -password dd260538abs
<!--NeedCopy-->
```

GUI を使用してデータベースユーザーのパスワードをリセットする

[システム] > [ユーザー管理] > [データベースユーザー] に移動し、ユーザーを選択し、パスワードの新しい値を入力します。

データベースユーザーがデータベースサーバーに存在しなくなった場合は、そのユーザーを ADC アプライアンスから削除できます。ただし、ユーザーがデータベースサーバーに残っている場合に ADC アプライアンスからユーザーを削除すると、このユーザー名のクライアントからのリクエストは認証されません。その結果、要求はデータベースサーバーにルーティングされません。

CLI を使用してデータベースユーザーを削除する

コマンドプロンプトで、次のように入力します。

```
rm db user <username>
<!--NeedCopy-->
```

例:

```
rm db user nsdbuser
<!--NeedCopy-->
```

GUI を使用してデータベースユーザーを削除する

[システム] > [ユーザ管理] > [データベースユーザ] に移動し、ユーザを選択して [削除] をクリックします。

データベースプロファイルを構成する

August 15, 2023

データベースプロファイルは、一度設定するだけで、それらの特定のパラメータ設定を必要とする複数の仮想サーバーに適用されるパラメータの名前付きコレクションです。データベースプロファイルを作成したら、それを負荷分散またはコンテンツスイッチング仮想サーバーにバインドします。プロファイルは必要な数だけ作成できます。

CLI を使用してデータベースプロファイルを作成する

コマンドラインで次のコマンドを入力してデータベースプロファイルを作成し、構成を確認します。

```
add db dbProfile <name> [-interpretQuery ( YES | NO )] [-stickiness (
   YES | NO )] [-kcdAccount <string>]

show db dbProfile
<!--NeedCopy-->
```

例:

```
> add dbProfile myDBProfile -interpretQuery YES -stickiness YES -
   kcdAccount mykcdaccnt
 Done
> show dbProfile myDBProfile
    Name: myDBProfile
    Interpret Query: YES
    Stickyness: YES
    KCD Account: mykcdaccnt
    Reference count: 0

 Done
>
<!--NeedCopy-->
```

GUI を使用してデータベースプロファイルを作成する

「システム」>「プロファイル」に移動し、「データベース・プロファイル」タブでデータベース・プロファイルを設定します。

CLI を使用して、データベースプロファイルを負荷分散またはコンテンツスイッチング仮想サーバーにバインドします

コマンドラインで、次のように入力します。


```
set (lb | cs) vserver <name> -dbProfileName <string>
<!--NeedCopy-->
```

GUI を使用して、データベースプロファイルを負荷分散またはコンテンツスイッチング仮想サーバーにバインドします

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] または [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [プロファイル] を選択し、[**DB** プロファイル] リストで、仮想サーバーにバインドするプロファイルを選択します。プロファイルを作成するには、プラスをクリックします (+).

DataStream の負荷分散を構成する

August 15, 2023

負荷分散設定を設定する前に、負荷分散機能を有効にする必要があります。次に、まず、負荷分散グループのデータベースサーバーごとに少なくとも 1 つのサービスを作成します。サービスを構成したら、負荷分散仮想サーバーを作成し、サービスを仮想サーバーにバインドする準備が整いました。

> 注記:
>
> データベースの場合、負荷分散は、同種のデータベースサーバー (まったく同じデータベースを含むデータベースサーバー) でのみ実行できます。異なるサーバー上の一意のデータベースを含む構成の場合は、コンテンツスイッチングを使用する必要があります。一部のデータベースサーバーが同じコンテンツをホストしている場合は、それらのサーバーでのみ負荷分散を使用できます。次に、コンテンツスイッチングポリシーを使用して、それらのデータベースの負荷分散を管理する負荷分散仮想サーバーに要求を送信できます。
>
> NetScaler ADC アプライアンスは、現在、データベースセッション中にデータベース名とログイン情報を格納しています。データベースに対してクエリが行われると、その情報を使用して特定のデータベースサーバーに接続します。

DataStream に固有のパラメーター値

- プロトコル

  仮想サーバーおよびサービスの構成時に、MySQL データベースには MYSQL プロトコルタイプを使用し、MS SQL データベースには MSSQL プロトコルタイプを使用します。MySQL プロトコルと TDS プロトコルは、SQL クエリを使用してそれぞれのデータベースサーバーと通信するためにクライアントによって使用されます。MySQL プロトコルの詳細については、「http://dev.mysql.com/doc/inte

rnals/en/client-server-protocol.html」を参照してください。TDS プロトコルの詳細については、http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspxを参照してください。

- ポート

  仮想サーバーがクライアント接続をリッスンするポート。MySQL データベースサーバーにはポート 3306 を使用します。

- 方法

  負荷分散を向上させ、サーバーの負荷を軽減するために、Least Connection 方式を使用することをお勧めします。ただし、ラウンドロビン、最小応答時間、送信元 IP ハッシュ、送信元 IP 宛先 IP ハッシュ、最小帯域幅、最小パケット、送信元 IP 送信元ポートハッシュなどの他の方法もサポートされています。

  注:URL ハッシュメソッドは DataStream ではサポートされていません。

- MS SQL サーバーバージョン

  Microsoft SQL Server を使用していて、一部のクライアントが Microsoft SQL Server 製品とは異なるバージョンを実行していることが予想される場合は、負荷分散仮想サーバーのサーバーバージョンパラメーターを設定します。バージョン設定では、すべての通信がサーバーのバージョンに準拠していることを確認することで、クライアント側接続とサーバー側の接続間の互換性を提供します。サーバーバージョンパラメーターの設定の詳細については、「MySQL および Microsoft SQL Server のバージョン設定を構成する」を参照してください。

- MySQL サーバーバージョン

  MySQL Server を使用していて、一部のクライアントが MySQL Server 製品とは異なるバージョンを実行していることが予想される場合は、負荷分散仮想サーバーのサーバーバージョンパラメーターを設定します。バージョン設定では、すべての通信がサーバーのバージョンに準拠していることを確認することで、クライアント側接続とサーバー側の接続間の互換性を提供します。サーバーバージョンパラメーターの設定の詳細については、「MySQL および Microsoft SQL Server のバージョン設定を構成する」を参照してください。

DataStream のコンテンツスイッチを構成する

August 15, 2023

データベース名、ユーザー名、文字セット、およびパケットサイズに基づいて、SQL クエリの情報に基づいてトラフィックをセグメント化できます。

高度なポリシー式を使用してコンテンツスイッチングポリシーを構成し、接続プロパティに基づいてコンテンツを切り替えることができます。たとえば、ユーザー名とデータベース名、コマンドパラメータ、サーバーを選択するための SQL クエリなどです。

高度なポリシー式は、MYSQL および MS SQL データベースサーバーに関連付けられたトラフィックを評価します。高度なポリシーポリシーで要求ベースの式を使用して、コンテンツスイッチング仮想サーバーのバインドポイントで

要求の切り替えを決定します。応答ベースの式（MYSQL.RES で始まる式）を使用して、ユーザーが構成したヘルスモニターに対するサーバーの応答を評価します。

高度なポリシー式の詳細については、「高度なポリシー式:DataStream」を参照してください。

> 注:
>
> データベースの場合、負荷分散は、同種のデータベースサーバー (まったく同じデータベースを含むデータベースサーバー) でのみ実行できます。異なるサーバー上の一意のデータベースを含む構成の場合は、コンテンツスイッチングを使用する必要があります。一部のデータベースサーバーが同じコンテンツをホストしている場合は、それらのサーバーでのみ負荷分散を使用できます。次に、コンテンツスイッチングポリシーを使用して、それらのデータベースの負荷分散を管理する負荷分散仮想サーバーに要求を送信できます。
>
> NetScaler ADC アプライアンスは、現在、データベースセッション中にデータベース名とログイン情報を格納しています。データベースに対してクエリが行われると、その情報を使用して特定のデータベースサーバーに接続します。

DataStream に固有のパラメーター値

- プロトコル

  仮想サーバーおよびサービスの構成時に、MySQL データベースには MYSQL プロトコルタイプを使用し、MS SQL データベースには MSSQL プロトコルタイプを使用します。MySQL プロトコルと TDS プロトコルは、SQL クエリを使用してそれぞれのデータベースサーバーと通信するためにクライアントによって使用されます。MySQL プロトコルの詳細については、「http://dev.mysql.com/doc/internals/en/client-server-protocol.html」を参照してください。TDS プロトコルの詳細については、http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspxを参照してください。

- ポート

  仮想サーバーがクライアント接続をリッスンするポート。MySQL データベースサーバーにはポート 3306 を使用します。

- MS SQL Server のバージョン

  Microsoft SQL Server を使用していて、一部のクライアントが Microsoft SQL Server 製品とは異なるバージョンを実行していると予想される場合は、コンテンツスイッチ仮想サーバーの [サーバーのバージョン] パラメーターを設定します。バージョン設定では、すべての通信がサーバーのバージョンに準拠していることを確認することで、クライアント側接続とサーバー側の接続間の互換性を提供します。サーバーバージョンパラメーターの設定の詳細については、「Microsoft SQL Server のバージョン設定の構成」を参照してください。

DataStream のモニターを構成する

August 15, 2023

負荷分散された各データベースサーバーの状態をリアルタイムで追跡するには、モニターを各サービスにバインドする必要があります。モニターは、定期的にプローブをサービスに送信してサービスをテストするように構成されています。これは、ヘルスチェックの実行と呼ばれることもあります。モニターがプローブへのタイムリーな応答を受信すると、サービスを UP としてマークします。指定された数のプローブに対するタイムリーな応答を受信しない場合、サービスは DOWN としてマークされます。

DataStream 場合は、組み込みモニター (MYSQL-ECV と MSSQL-ECV) を使用する必要があります。このモニターを使用すると、SQL 要求を送信し、その応答を解析して文字列を求めることができます。

DataStream のモニターを構成する前に、データベースユーザーの資格情報を NetScaler アプライアンスに追加する必要があります。モニタの構成の詳細については、「負荷分散セットアップでのモニタの設定」を参照してください。

モニターを作成すると、データベースサーバーとの TCP 接続が確立され、モニターの作成時に指定されたユーザー名を使用して接続が認証されます。その後、データベースサーバーに SQL クエリを実行し、サーバーの応答を評価して、構成されたルールと一致するかどうかを確認できます。

以下の例は MYSQL サーバー用です。

例:

次の例では、エラーメッセージの値を評価してサーバーの状態を判断します。

```
add lb monitor lb_mon1 MYSQL-ECV -sqlQuery "select * from
table2;" -evalrule "mysql.res.error.message.contains("Invalid
User")"-database "NS" -userName "user1"
<!--NeedCopy-->
```

次の例では、応答の行数を評価してサーバーの状態を判断します。

```
add lb monitor lb_mon4 MYSQL-ECV -sqlQuery "select * from
table4;" -evalrule "mysql.res.atleast_rows_count(7)" -database "NS" -
    userName "user2"
<!--NeedCopy-->
```

次の例では、特定の列の値を評価してサーバーの状態を判断します。

```
add lb monitor lb_mon3 MYSQL-ECV
-sqlQuery "select * from ABC;" -evalrule "mysql.res.row(1).double_elem
    (2) == 345.12"
-database "NS" -userName "user3"
<!--NeedCopy-->
```

次の例は MSSQL サーバー用です。

例:

次の例では、エラーメッセージの値を評価してサーバーの状態を判断します。

```
add lb monitor lb_mon1 MSSQL-ECV -sqlQuery "select * from
table2;" -evalrule "mssql.res.error.message.contains("Invalid
```



```
User")"-database "NS" -userName "user1"
<!--NeedCopy-->
```

次の例では、応答の行数を評価してサーバーの状態を判断します。

```
add lb monitor lb_mon4 MSSQL-ECV -sqlQuery "select * from
table4;" -evalrule "mssql.res.atleast_rows_count(7)" -database "NS" -
   userName "user2"
<!--NeedCopy-->
```

次の例では、特定の列の値を評価してサーバーの状態を判断します。

```
add lb monitor lb_mon3 MSSQL-ECV
-sqlQuery "select * from ABC;" -evalrule "mssql.res.row(1).double_elem
   (2) == 345.12"
-database "NS" -userName "user3"
<!--NeedCopy-->
```

ユースケース 1：プライマリ/セカンダリデータベースアーキテクチャの DataStream を構成する

August 15, 2023

一般的に使用される導入シナリオは、プライマリデータベースがすべての情報をセカンダリデータベースにレプリケートするプライマリ/セカンダリデータベースアーキテクチャです。

プライマリ/セカンダリデータベースアーキテクチャでは、すべての WRITE 要求をプライマリデータベースに送信し、すべての READ 要求をセカンダリデータベースに送信できます。

次の図は、アプライアンスに設定する必要のあるエンティティとパラメーターの値を示しています。

図 1：プライマリ/セカンダリデータベースセットアップ用の DataStream エンティティモデル



このシナリオ例では、プライマリデータベースを表すサービス (SVC_MySQL_1) が作成され、負荷分散仮想サーバー (lb_vsr_mysql_Primary) にバインドされます。3 つのセカンダリデータベースを表すためにさらに 3 つのサービス (SVC_MySQL_2、svc_mysql_3、および svc_mysql_4) が作成され、それらは別の負荷分散仮想サーバー (lb_vsr_mysql_Secondary) にバインドされます。

コンテンツスイッチ仮想サーバー (cs_vsr_MySQL_1) は、すべての WRITE 要求を負荷分散仮想サーバー lb_vsr_mysql_Primary に送信するように、関連付けられたポリシーを使用して構成されます。すべての READ 要求は、負荷分散仮想サーバー lb_vsr_mysql_secondary に送信されます。

要求がコンテンツスイッチ仮想サーバーに到達すると、仮想サーバーはその要求に対して関連するコンテンツスイッチポリシーを適用します。ポリシーを評価すると、コンテンツスイッチ仮想サーバーは要求を適切な負荷分散仮想サーバーにルーティングし、要求は適切なサービスに送信されます。

次の表に、NetScaler ADC アプライアンスで構成されているエンティティとポリシーの名前と値を示します。

エンティティの種類	名前	IP アドレス	プロトコル	ポート	式
Services	svc_mysql_1	198.51.100.5	MYSQL	3306	-
	svc_mysql_2	198.51.100.6	MYSQL	3306	-

エンティティの種類	名前	IP アドレス	プロトコル	ポート	式
	svc_mysql_3	198.51.100.7	MYSQL	3306	-
	svc_mysql_4	198.51.100.8	MYSQL	3306	-
監視	lb_mon1	-	MYSQL-ECV	-	mysql.res.atleast_rows_co
仮想サーバの負荷分散	lb_vsr_mysql_primary	198.51.100.201	MYSQL	3306	-
	lb_vsr_mysql_secondary	198.51.100.202	MYSQL	3306	-
コンテンツスイッチ仮想サーバー	cs_vsr_mysql_1	198.51.100.161	MYSQL	3306	-
コンテンツスイッチポリシー	Cs_select	-	-	-	MYSQL.REQ.QUERY.COMMAND.contains("select")

表 1. エンティティとポリシーの名前と値

コマンドラインインターフェイスを使用して **DataStream** をプライマリ/セカンダリデータベース設定用に設定するには

コマンドプロンプトで、次のように入力します。

```
add db user user1 -password user1

add service Svc_mysql_1 198.51.100.5 mysql 3306

add service Svc_mysql_2 198.51.100.6 mysql 3306

add service Svc_mysql_3 198.51.100.7 mysql 3306

add service Svc_mysql_4 198.51.100.8 mysql 3306

add lb monitor lb_mon1 MYSQL-ECV -sqlQuery "select * from table1;" -
    evalrule "mysql.res.atleast_rows_count(1)" -database "NS" -userName
     "user1"

add lb vserver Lb_vsr_mysql_primary mysql 198.51.100.201 3306

add lb vserver Lb_vsr_mysql_secondary mysql 198.51.100.202 3306

```



```
bind lb vserver Lb_vsr_mysql_primary svc_mysql_1

bind lb vserver Lb_vsr_mysql_secondary svc_mysql_2

bind lb vserver Lb_vsr_mysql_secondary svc_mysql_3

bind lb vserver Lb_vsr_mysql_secondary svc_mysql_4

add cs vserver Cs_vsr_mysql_1 mysql 198.51.100.161 3306

add cs policy Cs_select  -rule "MYSQL.REQ.QUERY.COMMAND.contains("
   select")"

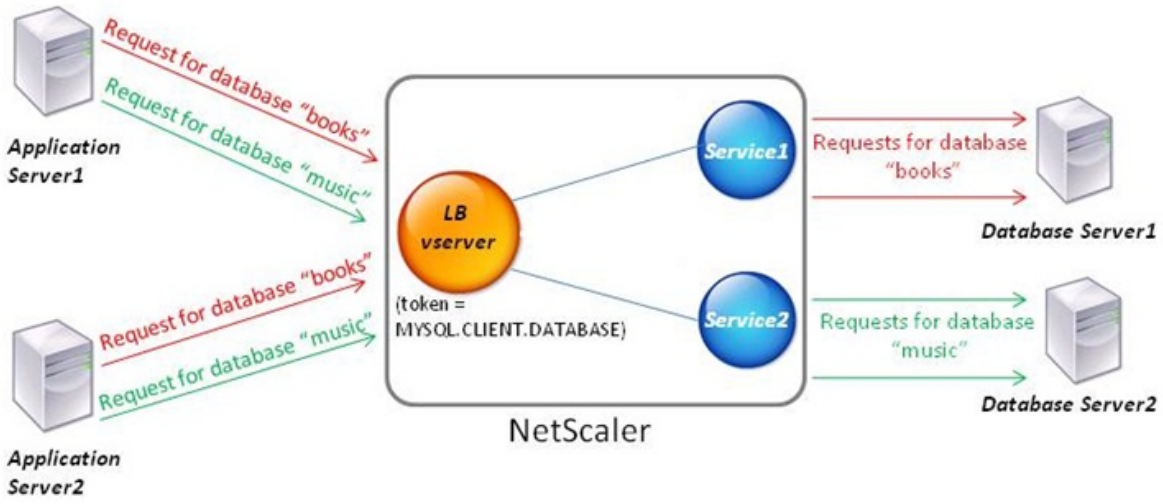
bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_primary

bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_secondary  -policy
   Cs_select  -priority 10

bind service Svc_mysql_1 -monitorName lb_mon1

bind service Svc_mysql_2 -monitorName lb_mon1

bind service Svc_mysql_3 -monitorName lb_mon1

bind service Svc_mysql_4 -monitorName lb_mon1
<!--NeedCopy-->
```

ユースケース 2：**DataStream** の負荷分散のトークン方式を構成する

August 15, 2023

DataStream の負荷分散のトークン方式を設定して、クライアント (アプリケーションまたは Web サーバー) リクエストから抽出されたトークンの値に基づいてデータベースサーバーを選択できます。これらのトークンは SQL 式を使用して定義されます。同じトークンを使用した後続のリクエストでは、NetScaler アプライアンスは最初のリクエストを処理したのと同じデータベースサーバーにリクエストを送信します。同じトークンのリクエストは、最大接続制限に達するか、セッションエントリが期限切れになるまで、同じデータベースサーバーに送信されます。

次のサンプル SQL 式を使用してトークンを定義できます。

MySQL	MS SQL
MYSQL.REQ.QUERY.TEXT	MSSQL.REQ.QUERY.TEXT
MYSQL.REQ.QUERY.TEXT (n)	MSSQL. REQ.QUERY.TEXT(n)
MYSQL.REQ.QUERY.COMMAND	MSSQL.REQ.QUERY.COMMAND

MySQL	MS SQL
MYSQL.CLIENT.USER	MSSQL.CLIENT.USER
MYSQL.CLIENT.DATABASE	MSSQL.CLIENT.DATABASE
MYSQL.CLIENT.CAPABILITIES	

次の例は、負荷分散のトークン方式を構成したときに NetScaler DataStream 機能がどのように機能するかを示しています。

図 1: DataStream とロードバランシングのトークン方式



この例では、トークンはデータベースの名前です。トークンブックを含むリクエストはデータベースサーバー 1 に送信され、トークンミュージックを含むリクエストはデータベースサーバー 2 に送信されます。それ以降のトークンブックのリクエストはすべてデータベースサーバー 1 に送信され、トークンミュージックを含むリクエストはデータベースサーバー 2 に送信されます。この構成では、データベースサーバーとの疑似パーシスタンスを実現します。

CLI を使用してこの例を設定します

コマンドプロンプトで入力します。

```
add service Service1 192.0.2.9 MYSQL 3306

add service Service2 192.0.2.11 MYSQL 3306

add lb vserver token_lb_vserver MYSQL 192.0.2.15 3306 -lbmethod token -
    rule MYSQL.CLIENT.DATABASE

bind lb vserver token_lb_vserver Service1
```



```
8
9  bind lb vserver token_lb_vserver Service2
10 <!--NeedCopy-->
```

GUI を使用してこの例を設定します

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを構成し、プロトコルを **MYSQL** として指定します。
2. サービスセクションをクリックし、プロトコルを MYSQL として指定する 2 つのサービスを設定します。これらのサービスを仮想サーバーにバインドします。
3. 「詳細設定」で「メソッド」をクリックし、「負荷分散方法」リストで「**TOKEN** 」を選択し、式を **MYSQL.CLIENT.DATABASE** として指定します。

ユースケース **3**: 透過モードの **MSSQL** トランザクションをログに記録する

August 15, 2023

NetScaler アプライアンスは、MSSQL クライアントとサーバー間で透過的に動作し、すべてのクライアント/サーバートランザクションの詳細のみをログに記録または分析するように構成できます。トランスペアレントモードは、NetScaler アプライアンスが MSSQL 要求のみをサーバーに転送し、サーバーの応答をクライアントに中継するように設計されています。要求と応答がアプライアンスを通過すると、アプライアンスは、監査ログまたは AppFlow 構成で指定されたとおりに、それらから収集された情報をログに記録するか、アクション分析構成で指定されたとおりに統計を収集します。データベースユーザーをアプライアンスに追加する必要はありません。

トランスペアレントモードで動作している場合、NetScaler アプライアンスは要求に対して負荷分散、コンテンツスイッチング、または接続多重化を実行しません。ただし、サーバーに代わってクライアントのログイン前パケットに応答するため、ログイン前のハンドシェイク中に暗号化が合意されるのを防ぐことができます。ログインパケットとそれ以降のパケットはサーバーに転送されます。

設定タスクの概要

トランスペアレントモードで MSSQL リクエストをロギングまたは分析するには、次の操作を行う必要があります。

- NetScaler アプライアンスをクライアントとサーバーの両方のデフォルトゲートウェイとして構成します。
- NetScaler アプライアンスで次のいずれかを実行します。
  - 送信元 **IP** アドレス **(USIP)** の使用オプションをグローバルに設定**:** ワイルドカード IP アドレスと MSSQL サーバーが要求を受信するポート番号 (ポート固有のワイルドカード仮想サーバー) を使用して

負荷分散仮想サーバーを作成します。次に、USIP オプションをグローバルに有効にします。ポート固有のワイルドカード仮想サーバーを構成する場合、アプライアンスに MSSQL サービスを作成する必要はありません。アプライアンスは、クライアント要求の宛先 IP アドレスに基づいてサービスを検出します。

  - USIP オプションをグローバルに設定したくない場合は、それぞれで **USIP** オプションを有効にして **MSSQL** サービスを作成します。サービスを構成する場合、ポート固有のワイルドカード仮想サーバーを作成する必要はありません。

- 監査ログ、AppFlow、またはアクション分析を設定して、リクエストに関する統計をログに記録または収集します。仮想サーバーを構成する場合、ポリシーを仮想サーバーまたはグローバルバインドポイントにバインドできます。仮想サーバーを設定しない場合は、ポリシーをグローバルバインドポイントにのみバインドできます。

ワイルドカード仮想サーバーを使用してトランスペアレントモードを設定

ポート固有のワイルドカード仮想サーバーを設定し、Use Source IP（USIP）モードをグローバルに有効にすることで、トランスペアレントモードを構成できます。クライアントがデフォルトゲートウェイ（NetScaler アプライアンス）に、宛先 IP アドレスヘッダーに MSSQL サーバーの IP アドレスを含む要求を送信すると、アプライアンスは宛先 IP アドレスが使用可能かどうかを確認します。IP アドレスが使用可能な場合、仮想サーバーは要求をサーバーに転送します。それ以外の場合は、リクエストがドロップされます。

CLI を使用してワイルドカード仮想サーバーを作成する

コマンドプロンプトで次のコマンドを入力してワイルドカード仮想サーバーを作成し、構成を確認します。

```
add lb vserver <name> <serviceType> <IPAddress> <port>

show lb vserver <name>
<!--NeedCopy-->
```

例:

```
> add lb vserver wildcardLbVs MSSQL * 1433
 Done
> show lb vserver wildcardLbVs
    wildcardLbVs (*:1433) - MSSQL   Type: ADDRESS
    State: UP
    . . .

 Done
>
<!--NeedCopy-->
```


GUI を使用してワイルドカード仮想サーバーを作成する

[トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、仮想サーバーを作成します。プロトコルとして MSSQL を指定し、IP アドレスとして * を指定します。

CLI を使用してソース **IP**（**USIP**）モードをグローバルに有効にする

コマンドプロンプトで次のコマンドを入力して USIP モードをグローバルに有効にし、構成を確認します。

```
enable ns mode USIP

show ns mode
<!--NeedCopy-->
```

例**:**

```
> enable ns mode USIP
 Done
> show ns mode

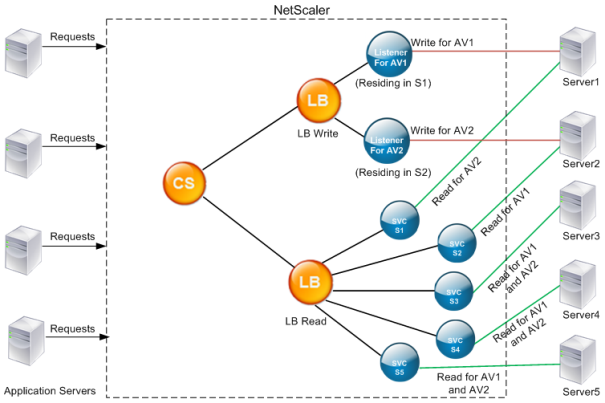
    Mode                                Acronym
        Status
    -------                             -------
        ------
                  . . .
 3) Use Source IP                  USIP                            ON
                  . . .
 Done
>
<!--NeedCopy-->
```

GUI を使用して **USIP** モードをグローバルに有効にする

1. [システム**] > [**設定**]** に移動し、[モードと機能] で [モードの構成] を選択します。
2. 「ソース **IP** を使用」を選択します。

MSSQL サービスを使用してトランスペアレントモードを設定

トランスペアレントモードを設定するには、MSSQL サービスを設定し、各サービスで USIP を有効にします。クライアントがデフォルトゲートウェイ（NetScaler アプライアンス）に、宛先 IP アドレスヘッダーに MSSQL サーバーの IP アドレスを含む要求を送信すると、アプライアンスはその要求を宛先サーバーに転送します。

MSSQL サービスを作成し、**CLI** を使用してサービスの **USIP** モードを有効にします

コマンドプロンプトで次のコマンドを入力して、USIP を有効にした MSSQL サービスを作成し、構成を確認します。

```
add service <name> (<IP> | <serverName>) <serviceType> <port> -usip YES
    `

show service <name>
<!--NeedCopy-->
```

例

```
> add service myDBservice 192.0.2.0 MSSQL 1433 -usip YES
 Done
> show service myDBservice
    myDBservice (192.0.2.0:1433) - MSSQL
    State: UP
                . . .
    Use Source IP: YES      Use Proxy Port: YES
                . . .
 Done
>
<!--NeedCopy-->
```

GUI を使用して **USIP** を有効にした **MSSQL** サービスを作成する

1. [トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、サービスを設定します。
2. プロトコルを **MSSQL** として指定し、**[**設定**]** で [ソース **IP** を使用] を選択します。

ユースケース **4**: データベース固有の負荷分散

August 15, 2023

データベースサーバーファームは、サーバーの状態だけでなく、各サーバーのデータベースの可用性にも基づいて負荷分散する必要があります。サービスが稼働していて、負荷分散デバイスが UP 状態と表示されていても、要求されたデータベースがそのサービスで使用できない場合があります。データベースが使用できないサービスにクエリが転送された場合、要求は処理されません。そのため、負荷分散デバイスは、各サービスのデータベースの可用性を認識する必要があります。また、負荷分散を決定する際には、データベースを利用できるサービスのみを考慮する必要があります。

例として、データベースサーバー server 1、サーバー 2、サーバー 3 が mydatabase1 と mydatabase2 というデータベースをホストしているとします。mydatabase1 が server2 で使用できなくなった場合、負荷分散デバイスはその状態の変化を認識している必要があります。mydatabase1 へのリクエストの負荷をサーバー 1 とサーバ

ー 3 のみに分散する必要があります。mydatabase1 がサーバー 2 で使用できるようになったら、負荷分散デバイスは負荷分散の決定に server2 を含める必要があります。同様に、mydatabase2 がサーバー 3 で使用できなくなった場合、デバイスは mydatabase2 のリクエストをサーバー 1 とサーバー 2 のみに負荷分散する必要があります。mydatabase2 が使用可能になった場合にのみ、ロードバランシングの決定に server3 を含める必要があります。この負荷分散の動作は、サーバーファームでホストされているすべてのデータベースで一貫している必要があります。

NetScaler アプライアンスは、サービス上でアクティブなすべてのデータベースのリストを取得することでこの動作を実装します。アクティブなデータベースのリストを取得するために、アプライアンスは適切な SQL クエリで構成されたモニターを使用します。要求されたデータベースがサービスで使用できない場合、アプライアンスはそのサービスが利用可能になるまで負荷分散の決定から除外します。この動作により、クライアントへのサービスが中断されることはありません。

> 注
>
> データベース固有の負荷分散は、MSSQL と MySQL のサービスタイプでのみサポートされます。このサポートは、Microsoft SQL Server 2012 の高可用性デプロイメントでも利用できます。

データベース固有の負荷分散を設定するには、以下を設定する必要があります。

- 負荷分散機能を有効にし、MSSQL または MySQL タイプの負荷分散仮想サーバーを構成します。
- データベースをホストするサービスを構成し、サービスを仮想サーバーにバインドします。モニターがデータベースサーバーにログオンするには有効なユーザー認証情報が必要なので、各サーバーでデータベースユーザーアカウントを構成し、そのユーザーアカウントを NetScaler アプライアンスに追加する必要があります。
- 次に、MSSQL-ECV または MYSQL-ECV モニターを設定し、モニターを各サービスにバインドします。
- 最後に、構成をテストして、意図したとおりに機能していることを確認する必要があります。これらの構成タスクを実行する前に、データベース固有の負荷分散の仕組みを理解しておいてください。

データベース固有の負荷分散の仕組み

データベース固有の負荷分散では、各データベースサーバーに、そのサーバー上のすべてのアクティブなデータベースの名前を定期的に問い合わせるモニターを構成します。NetScaler アプライアンスは結果を保存し、監視を通じて取得した情報に基づいて定期的にレコードを更新します。クライアントが特定のデータベースにクエリを実行すると、アプライアンスは設定された負荷分散方法を使用してサービスを選択し、そのレコードをチェックして、そのサービスでデータベースが使用可能かどうかを判断します。レコードからデータベースが使用できないことが示された場合、データベースは設定された負荷分散方法を使用して次に使用可能なサービスを選択し、チェックを繰り返します。アプライアンスは、データベースがアクティブになっている最初に使用可能なサービスにクエリを転送します。

負荷分散を有効にする

負荷分散機能が無効になっている場合は、サービスや仮想サーバーなどの負荷分散エンティティを構成できます。この機能を有効にするまで、エンティティは機能しません。

CLI を使用して負荷分散を有効にする

コマンドプロンプトで次のコマンドを入力して負荷分散を有効にし、構成を確認します。

```
enable ns feature LB

show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

GUI を使用して負荷分散を有効にする

[システム] **>** [設定] に移動し、[基本機能の設定] で [負荷分散] を選択します。

データベース固有の負荷分散用の負荷分散仮想サーバーの構成

可用性に基づいてデータベースの負荷分散を行うように仮想サーバーを構成するには、仮想サーバーでデータベース固有の負荷分散パラメーターを有効にします。パラメーターを有効にすると、NetScaler アプライアンスがクエリをそのサービスに転送する前に、選択したサービスに送信された監視プローブの結果を参照するように負荷分散ロジックが変更されます。

CLI を使用してデータベース固有の負荷分散用の負荷分散仮想サーバーを構成する

コマンドプロンプトで次のコマンドを入力して、データベース固有の負荷分散用の負荷分散仮想サーバーを構成し、構成を確認します。

```
add lb vserver <name> <serviceType> <ipAddress> <port> -dbsLb ENABLED

show lb vserver <name>
```



```
<!--NeedCopy-->
```

サービスを構成する

負荷分散機能を有効にしたら、負荷分散設定に含めるアプリケーションサーバーごとに少なくとも 1 つのサービスを作成する必要があります。構成するサービスは、NetScaler アプライアンスと負荷分散サーバー間の接続を提供します。各サービスには名前があり、IP アドレス、ポート、提供されるデータの種類を指定します。

最初にサーバーオブジェクトを作成せずにサービスを作成した場合、サービスの IP アドレスは、サービスをホストするサーバーの名前でもあります。IP アドレスではなく名前でサーバーを識別したい場合は、サーバーオブジェクトを作成してから、サービスの作成時に IP アドレスの代わりにサーバーの名前を指定できます。

データベースユーザーを構成する

データベースでは、接続は常にステートフルです。つまり、接続が確立されたら認証を受ける必要があります。

NetScaler でデータベースのユーザー名とパスワードを設定します。たとえば、データベースに John というユーザーを設定している場合、ADC にもユーザー John を設定する必要があります。ADC `nsconfig` に追加されたデータベースユーザー名とパスワードがファイルに追加されます。

> 注
>
> 名前は大文字と小文字が区別されます。

ADC は、これらのユーザー認証情報を使用してクライアントを認証し、次にデータベースサーバーとのサーバー接続を認証します。

CLI を使用してデータベースユーザーを追加する

コマンドプロンプトで、次のように入力します。

```
add db user <username> - password <password>
<!--NeedCopy-->
```

例:

```
add db user nsdbuser -password dd260427edf
<!--NeedCopy-->
```

GUI を使用してデータベースユーザーを追加する

[システム] **>** [ユーザー管理] **>** [データベースユーザー] に移動し、データベースユーザーを設定します。

データベースサーバー上のデータベースユーザーのパスワードを変更した場合は、NetScaler アプライアンスで構成されている対応するユーザーのパスワードをリセットする必要があります。

CLI を使用してデータベースユーザーのパスワードをリセットする

コマンドプロンプトで、次のように入力します。

```
set db user <username> -password <password>
<!--NeedCopy-->
```

例:

```
set db user nsdbuser -password dd260538abs
<!--NeedCopy-->
```

GUI を使用してデータベースユーザーのパスワードをリセットする

[システム] **>** **[**ユーザー管理] **>** **[**データベースユーザー] に移動し、ユーザーを選択し、パスワードの新しい値を入力します。

データベースサーバーにデータベースユーザーが存在しなくなった場合は、そのユーザーを NetScaler アプライアンスから削除できます。ただし、ユーザーがデータベースサーバーに残っている場合に ADC アプライアンスからユーザーを削除すると、このユーザー名のクライアントからのリクエストは認証されません。そのため、ユーザー名はデータベースサーバーにルーティングされません。

CLI を使用してデータベースユーザーを削除する　コマンドプロンプトで、次のように入力します。

```
rm db user <username>
<!--NeedCopy-->
```

例:

```
rm db user nsdbuser
<!--NeedCopy-->
```

GUI を使用してデータベースユーザーを削除する

[システム] **>** **[**ユーザ管理] **>** **[**データベースユーザ] に移動し、ユーザを選択して [削除] をクリックします。

アクティブなデータベースの名前を取得するようにモニターを設定

モニターを作成して、データベースインスタンス上のすべてのアクティブなデータベースのリストを取得できます。モニターは、有効なユーザー認証情報を使用してデータベースサーバーにログオンし、適切な SQL クエリを実行します。使用する必要がある SQL クエリは、SQL Server のデプロイメントによって異なります。たとえば、MSSQL データベースミラーリング設定では、次のクエリを使用してサーバーインスタンスで使用可能なアクティブなデータベースのリストを取得できます。


```
select name from sys.databases where state=0
<!--NeedCopy-->
```

MySQL データベースの設定では、次のクエリを使用して、サーバーインスタンスで使用可能なアクティブなデータベースのリストを取得できます。

データベースを表示:

また、エラー状態に対する応答を評価し、エラーがない場合は結果を保存するようにモニターを構成します。応答にエラーが含まれている場合、モニターはサービスに DOWN とマークします。アプライアンスは、エラーが返されなくなるまで、サービスを負荷分散の決定から除外します。

> 注
>
> データベース固有の負荷分散機能は、MSSQL と MySQL のサービスタイプでのみサポートされます。したがって、モニタータイプは MSSQL-ECV または MYSQL-ECV でなければなりません。

CLI を使用して、サービスでホストされているすべてのアクティブなデータベースの名前を取得するようにモニターを設定します

コマンドプロンプトで次のコマンドを入力して、サービスでホストされているすべてのアクティブなデータベースの名前を取得し、構成を確認します。

```
add lb monitor <monitorName> <type> -userName <string> -sqlQuery <text>
    -evalRule <expression> -storedb ENABLED

show lb monitor <monitorName>
<!--NeedCopy-->
```

GUI を使用して、サービスでホストされているすべてのアクティブなデータベースの名前を取得するようにモニターを構成します

1. [トラフィック管理] **>** [負荷分散] **>** [モニター] に移動し、MSSQL-ECV または MYSQL-ECV タイプのモニターを設定します。
2. 「特殊パラメータ」で、ユーザー名、クエリ、およびルールを指定します。たとえば、MSSQL-ECV の場合、クエリは「sys.database から名前を選択 (state=0)」で、ルールは MSSQL.RES.TYPE.NE (エラー) でなければなりません。MYSQL-ECV の場合、クエリは「データベースを表示」で、ルールは MYSQL.RES.TYPE.NE (エラー) でなければなりません。

MSSQL の可用性グループ展開サポート

高可用性グループ展開でデータベース固有の負荷分散が設定されている次のシナリオを考えてみましょう。S1 から S5 は ADC アプライアンス上のサービスです。DB1 から DB4 は、サービス S1 から S5 で表されるサーバー上のデ

ータベースです。AV1 と AV2 は可用性グループです。各可用性グループには、最大 1 つのプライマリデータベースサーバーインスタンスと最大 4 つのセカンダリデータベースサーバーインスタンスが含まれます。可用性グループ内のサーバーを表すサービスを、ある可用性グループではプライマリに、別の可用性グループではセカンダリにすることができます。各可用性グループには、異なるデータベースと 1 つのリスナー (サービス) が含まれます。すべてのリクエストは、プライマリデータベースにあるリスナーサービスに到着します。AVI には DB1 と DB2 のデータベースが含まれています。AV2 には DB3 と DB4 のデータベースが含まれています。L1 と L2 はそれぞれ AV1 と AV2 のリスナーです。S1 は AV1 のプライマリサービスで、S2 は AV2 のプライマリサービスです。



サービス	サービス上のアクティブなデータベースのリスト
S1	DB1、DB2、DB3、DB4
S2	DB3、DB4
S3	DB3、DB4
S4	DB1、DB2
S5	DB1、DB2

可用性グループ	データベース	可用性グループのサーバーを表すサービス
AV1	DB1、DB2	S1, S4, S5
AV2	DB3、DB4	S1, S2, S3

クエリの流れは次のとおりです。

1. AV1 の READ クエリは S4 と S5 の間で負荷分散されます。S1 は AV1 のプライマリです。
2. AV1 の書き込みクエリは L1 に送信されます。
3. AV2 の READ クエリは S1 と S3 の間で負荷分散されます。S2 は AV2 のプライマリです。
4. AV1 の書き込みクエリは L2 に送信されます。

構成例

1. 負荷分散とコンテンツスイッチング仮想サーバーを設定します。

   - `add lb vserver lbwrite -dbslb enabled`
   - `add lbvserver lbread MSSQL -dbslb enabled`
   - `add csvserver csv MSSQL 1.1.1.10 1433`

2. 可用性グループごとに 1 つずつ、合計 2 つのリスナーサービスと、データベース DB1 から DB4 を表す 5 つのサービス S1 ～S5 を設定します。

   - `add service L1 1.1.1.11 MSSQL 1433`
   - `add service L2 1.1.1.12 MSSQL 1433`
   - `add service s1 1.1.1.13 MSSQL 1433`
   - `add service s2 1.1.1.14 MSSQL 1433`
   - `add service s3 1.1.1.15 MSSQL 1433`
   - `add service s4 1.1.1.16 MSSQL 1433`
   - `add service s5 1.1.1.17 MSSQL 1433`

3. サービスを負荷分散仮想サーバーにバインドします。

   - `bind lbvserver lbwrite L1`
   - `bind lbvserver lbwrite L2`
   - `bind lbvserver lbread s1`
   - `bind lbvserver lbread s2`
   - `bind lbvserver lbread s3`
   - `bind lbvserver lbread s4`
   - `bind lbvserver lbread s5`

4. データベースユーザーを設定します。

   - `add db user nsdbuser1 -password dd260427edf`
   - `add db user nsdbuser2 -password ccd1234xyzw`

5. リスナーサービスごとに Monitor_L1 と Monitor_L2 の 2 つのモニターを構成して、その可用性グループ内のアクティブなデータベースのリストを取得します。モニター monitor1 を追加して、セカンダリデータベースサーバーインスタンスのデータベースのリストを取得します。

   - ```
     add lb monitor monitor_L1 MSSQL-ECV -userName user1 -sqlQuery
      "SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability
      b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_list
      c on b.group_id = c.group_id INNER JOIN sys.availability_group_listen
      d on c.listener_id = d.listener_id WHERE b.role = 1 and d.
     ip_address like '1.1.1.11'"-evalRule "MSSQL.RES.TYPE.NE(ERROR
     )" -storedb ENABLED
     ```


     ```

- add lb monitor monitor_L2 MSSQL-ECV -userNameuser1 -sqlQuery "SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_ b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_list c on b.group_id = c.group_id INNER JOIN sys.availability_group_listen d on c.listener_id = d.listener_id WHERE b.role = 1 and d. ip_address like '1.1.1.12'"-evalRule "MSSQL.RES.TYPE.NE(ERROR )"-storedb ENABLED
- add lb monitor monitor1 MSSQL-ECV -userNameuser1 -sqlQuery " SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_r b ON a.replica_id=b.replica_id WHERE b.role = 2"-evalRule " MSSQL.RES.TYPE.NE(ERROR)"-storedb ENABLED

6. 読み取りポリシーと書き込みポリシーを設定します。

   - add cs policy pol_write -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS( "insert")"
   - add cs policy pol_read -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS(" select")"

7. ポリシーをコンテンツスイッチング仮想サーバーにバインドします。

   - bind csvserver csv -targetLBVserver lbwrite -policyName pol_write -priority 11
   - bind csvserver csv -targetLBVserver lbread -policyName pol_read -priority 12

8. モニターをサービスにバインドします。モニターをサービス L1 と L2 にバインドして、リスナーとなる可用性グループのアクティブなデータベースのリストを取得します。読み取り専用仮想サーバーにバインドされているすべてのサービスにモニターをバインドします。

   - bind service L1 -monitorName monitor_L1
   - bind service L2 -monitorName monitor_L2
   - bind service s1 -monitorName monitor1
   - bind service s2 -monitorName monitor1
   - bind service s3 -monitorName monitor1
   - bind service s4 -monitorName monitor1
   - bind service s5 -monitorName monitor1

### MSSQL 仮想サーバーの設定例

データベース固有の負荷分散用に負荷分散仮想サーバーを構成するには:

```
1 add lb vserver DBSpecificLB1 MSSQL 192.0.2.10 1433 -dbsLb ENABLED
2
```


```
Done

show lb vserver DBSpecificLB1

DBSpecificLB1 (192.0.2.10:1433) - MSSQL Type: ADDRESS
. . .
DBS_LB: ENABLED

Done
<!--NeedCopy-->
```

サービスを設定するには:

サービス **msservice1** を追加 5.5.5.5 MSQL 143

コマンドラインを使用して、サービスでホストされているすべてのアクティブなデータベースの名前を取得するようにモニターを構成するには、次の手順を実行します。

```
add lb monitor mssql-monitor1 MSSQL-ECV -userName user1 -sqlQuery "
    select name from sys.databases where state=0" -evalRule "MSSQL.RES.
    TYPE.NE(ERROR)" -storedb EN

Done

show lb monitor mssql-monitor1

1)   Name.......: mssql-monitor1    Type......: MSSQL-ECV

...

Special parameters: Database.....:""

User name.....:"user1"

Query..:select name from sys.databases where state=0 EvalRule...:MSSQL.
    RES.TYPE.NE(ERROR)

Version...:70 STORE_DB...:ENABLED

Done
<!--NeedCopy-->
```

MySQL 仮想サーバーの設定例

データベース固有の負荷分散用に負荷分散仮想サーバーを構成するには:

```
add lb vserver DBSpecificLB1 MYSQL 192.0.2.10 3306 -dbsLb ENABLED

Done

show lb vserver DBSpecificLB1
```



```
6
7  DBSpecificLB1 (192.0.2.10:3306) - MYSQL Type: ADDRESS
8
9  . . .
10
11 DBS_LB: ENABLED
12
13 Done
14 <!--NeedCopy-->
```

サービスを設定するには:

```
1  add service msservice1 5.5.5.5 MYSQL 3306
2  <!--NeedCopy-->
```

コマンドラインを使用して、サービスでホストされているすべてのアクティブなデータベースの名前を取得するようにモニターを構成するには、次の手順を実行します。

```
1  add lb monitor mysql-monitor1 MYSQL-ECV -userName user1 -sqlQuery "show
       databases" -evalRule "MYSQL.RES.TYPE.NE(ERROR)" -storedb ENABLED
2
3  Done
4
5  show lb monitor mysql-monitor1
6
7  1)     Name.......: mysql-monitor1  Type......: MYSQL-ECV  State....:
      ENABLED
8
9  ...
10
11 Special parameters: Database.....:""
12
13 User name.....:"user1" Query..:show databases
14
15 EvalRule...:MYSQL.RES.TYPE.NE(ERROR) STORE_DB...:ENABLED
16
17 Done
18 <!--NeedCopy-->
```

DataStream リファレンス

February 15, 2024

このリファレンスでは、MySQL プロトコルと TDS プロトコル、データベースバージョン、認証方法、DataStream 機能でサポートされる文字セットについて説明します。また、接続の状態を変更するトランザクションリクエストや特別なクエリを NetScaler がどのように処理するかについても説明します。

DataStream 機能の監査ログメッセージを生成するように NetScaler アプライアンスを構成することもできま

す。

サポートされているデータベースバージョン、プロトコル、認証方法

	MySQL データベース	MS SQL データベース
データベースバージョン	MySQL データベースバージョン 4.1、5.0、5.1、5.4、5.5、5.6	MS SQL データベースバージョン 70、2000、2000SP1、2005、2008、2008R2、2012、2014 (ケルベロス認証サポート)。NetScaler パラメーター`mssqlServerVersion` `2014`を設定することで、2016 年、2017 年、2019 年、および 2022 年にバックエンド MS SQL サーバーを TDS 7.4 で構成できます。常時接続可用性グループは、MS SQL バージョン 2012 と 2014 でのみサポートされています。
プロトコル	MySQL プロトコルのバージョン 10。MySQL プロトコルの詳細については、「MySQL クライアント/サーバープロトコル」を参照してください	表形式データストリーム (TDS) プロトコルバージョン 7.1 ～7.4。TDS プロトコルの詳細については、表形式データストリームプロトコルを参照してください
認証方法	MySQL ネイティブ認証がサポートされています。	SQL サーバー認証と Windows 認証 (Kerberos/NTLM) がサポートされています。

文字セット

DataStream 機能は UTF-8 文字セットのみをサポートします。

リクエストを送信する際にクライアントが使用する文字セットは、データベースサーバーの応答で使用される文字セットとは異なる場合があります。charset パラメータは接続確立時に設定されますが、SQL クエリを送信することでいつでも変更できます。文字セットは接続に関連付けられているため、ある文字セットの接続に対するリクエストを別の文字セットの接続に多重化することはできません。

NetScaler アプライアンスは、クライアントから送信されたクエリとデータベースサーバーから送信された応答を解析します。

接続に関連する文字セットは、最初のハンドシェイクの後に次の 2 つのクエリを使用して変更できます:


```
SET NAMES <charset> COLLATION <collation>

SET CHARACTER SET <charset>
<!--NeedCopy-->
```

トランザクション

MySQL では、トランザクションは接続パラメータ AUTOCOMMIT または BEGIN: COMMIT クエリを使用して識別されます。AUTOCOMMIT パラメータは、最初のハンドシェイク中に設定することも、接続が確立された後に SET AUTOCOMMIT クエリを使用して設定することもできます。

NetScaler アプライアンスは各クエリを明示的に解析して、トランザクションの開始と終了を決定します。

MySQL プロトコルでは、応答には接続がトランザクションかどうかを示す 2 つのフラグ (TRANSACTION フラグと AUTOCOMMIT フラグ) が含まれます。

接続がトランザクションの場合、TRANSACTION フラグが設定されます。または、オートコミットモードが OFF の場合、AUTOCOMMIT フラグは設定されません。ADC アプライアンスは応答を解析し、TRANSACTION フラグが設定されているか AUTOCOMMIT フラグが設定されていない場合、接続の多重化は行われません。これらの条件が満たされなくなると、ADC アプライアンスは接続の多重化を開始します。

> 注
>
> トランザクションは MS SQL でもサポートされています。

特別なクエリ

SET や PREPARE など、接続の状態を変更してリクエストの切り替えを中断する可能性のある特殊なクエリがあるため、これらのクエリは別の方法で処理する必要があります。

NetScaler アプライアンスは、特別なクエリを含む要求を受信すると、OK 応答をクライアントに送信し、その要求を接続に保存します。

保存されたクエリと一緒に INSERT や SELECT などの非特殊クエリを受信すると、ADC アプライアンスは、保存されたクエリがすでにデータベースサーバーに送信されているサーバー側の接続を探します。そのような接続が存在しない場合、ADC アプライアンスは接続を作成し、保存されたクエリを最初に送信し、次に非特殊クエリを含む要求を送信します。

SET、USE db、INIT_DB の特殊クエリでは、アプライアンスは特殊クエリに対応するサーバー側接続のフィールドを変更します。この変更により、サーバー側の接続をより適切に再利用できます。

各接続に保存されるクエリは 16 個だけです。

以下は、ADC アプライアンスの動作が変更された特殊クエリのリストです。

- SET クエリー

  SET SQL クエリは、接続に関連する変数を定義します。これらのクエリはグローバル変数の定義にも使用されますが、現在のところ、ADC アプライアンスはローカル変数とグローバル変数を区別できません。このクエリでは、ADC アプライアンスは「ストアアンドフォワード」メカニズムを使用します。

- <db> クエリを使う

  このクエリを使用して、ユーザーは接続に関連するデータベースを変更できます。この場合、<db> ADC アプライアンスは送信された値を解析し、使用する新しいデータベースを反映するようにサーバー側接続のフィールドを変更します。

- INIT_DB コマンド

  このクエリを使用して、ユーザーは接続に関連するデータベースを変更できます。この場合、<init_db> ADC アプライアンスは送信された値を解析し、使用する新しいデータベースを反映するようにサーバー側接続のフィールドを変更します。

- COM_PREPARE

  ADC アプライアンスは、このコマンドを受信するとリクエストの切り替えを停止します。

- クエリを準備

  このクエリは、接続に関連するプリペアドステートメントを作成するために使用されます。このクエリでは、ADC アプライアンスは「ストアアンドフォワード」メカニズムを使用します。

監査ログメッセージのサポート

DataStream 機能の監査ログメッセージを生成するように NetScaler アプライアンスを構成できるようになりました。監査ログメッセージは、クライアント側とサーバー側の接続が確立、クローズ、またはドロップされたときに生成されます。記録して表示できるメッセージのカテゴリは ERROR と INFO です。クライアント側接続のエラーメッセージは「CS」で始まり、サーバー側接続のエラーメッセージは「SS」で始まります。必要に応じて追加情報が提供されます。たとえば、閉じた接続 (CS_CONN_CLOSED) のログメッセージには、接続 ID のみが含まれます。ただし、確立された接続 (CS_CONN_ESTD) のログメッセージには、接続 ID に加えて、ユーザー名、データベース名、クライアント IP アドレスなどの情報が含まれます。

ドメインネームシステム

August 15, 2023

注：リリース 13.0 ビルド 41.x 以降、ADNS およびプロキシモードの NetScaler ADC アプライアンスは、2019 年の DNS フラグ日に完全に準拠しています。

NetScaler ADC アプライアンスは、ドメインの権限のあるドメインネームサーバー（ADNS サーバー）として機能するように構成できます。アプライアンスが権限を持つドメインに属する DNS リソースレコードを追加し、リソースレコードパラメータを設定します。また、ネットワーク内またはネットワーク外にある DNS ネームサーバのファームを負荷分散するプロキシ DNS サーバとしてアプライアンスを設定することもできます。アプライアンスをエンドリゾルバおよびフォワーダとして設定します。完全修飾ドメイン名が設定されていない場合に名前解決を有効にする DNS サフィックスを設定できます。アプライアンスは、ドメインに属するすべてのレコードを取得する DNS ANY クエリもサポートしています。

アプライアンスは、あるドメインでは権限のある DNS サーバとして、別のドメインでは DNS プロキシサーバとして同時に機能するように設定できます。アプライアンスをゾーンの権威 DNS サーバーまたは DNS プロキシサーバーとして構成すると、User Datagram Protocol（UDP）に指定されたサイズ制限を超える応答サイズに TCP を使用するようにアプライアンスを有効化できます。

NetScaler ADC での **DNS** の仕組み

NetScaler ADC アプライアンスは、ADNS サーバー、DNS プロキシサーバー、エンドリゾルバー、およびフォワーダーとして機能するように構成できます。NetScaler ADC アプライアンスには、次のレコードを含む DNS リソースレコードを追加できます。

- サービス (SRV) レコード
- IPv6 (AAAA) レコード
- 住所 (A) レコード
- メール交換 (MX) レコード
- 正規名 (CNAME) レコード
- ポインタ (PTR) レコード
- 権限の開始 (SOA) レコード
- テキスト (TXT) レコード
- 名前権限ポインタ (NAPTR) レコード
- DNSKEY レコード
- 認証局認証 (CAA) レコード

また、外部 DNS ネームサーバーの負荷分散を行うように NetScaler ADC を構成することもできます。

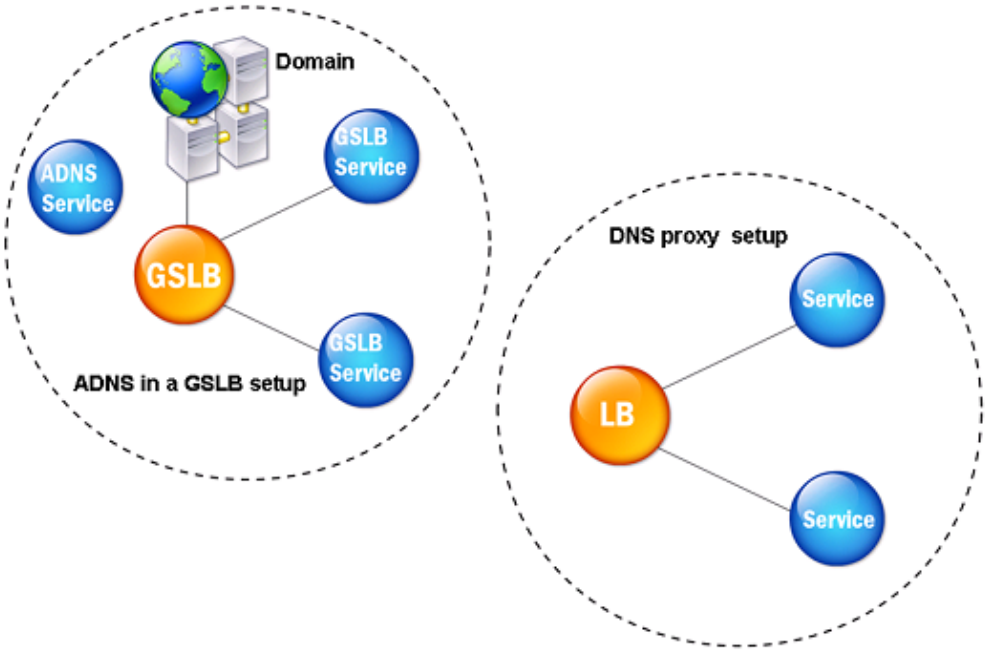
NetScaler ADC アプライアンスは、ドメインの権限として構成できます。ドメインの有効な SOA および NS レコードを追加します。

ADNS サーバーは、ゾーンに関する完全な情報を含む DNS サーバーです。

NetScaler ADC アプライアンスをゾーンの ADNS サーバーとして構成するには、ADNS サービスを追加してからゾーンを構成する必要があります。そのためには、ドメインの有効な SOA および NS レコードを追加します。クライアントが DNS 要求を送信すると、NetScaler ADC アプライアンスは構成されたリソースレコードでドメイン名を検索します。NetScaler グローバルサーバー負荷分散（GSLB）機能と併用するように ADNS サービスを構成できます。

サブドメインの NS レコードを親ドメインのゾーンに追加することで、サブドメインを委任できます。次に、サブドメインネームサーバーごとに「グルーレコード」を追加することで、NetScaler ADC をサブドメインに対して権限のあるものにすることができます。GSLB が構成されている場合、NetScaler ADC はその構成に基づいて GSLB 負荷分散の決定を行い、選択した仮想サーバーの IP アドレスで応答します。次の図は、ADNS GSLB セットアップと DNS プロキシセットアップのエンティティを示しています。

図 1: DNS プロキシエンティティモデル



NetScaler ADC アプライアンスは DNS プロキシとして機能できます。DNS プロキシの重要な機能である DNS レコードのキャッシュは、NetScaler ADC アプライアンスではデフォルトで有効になっています。キャッシュにより、NetScaler ADC アプライアンスは翻訳の繰り返しに対して迅速に対応できます。負荷分散 DNS 仮想サーバーと DNS サービスを作成し、これらのサービスを仮想サーバーにバインドします。

NetScaler ADC には、キャッシュされたデータの存続期間を構成するための最小有効期間（TTL）と最大 TTL の 2 つのオプションがあります。キャッシュされたデータは、これら 2 つのオプションの設定に従ってタイムアウトします。NetScaler ADC は、サーバーから送信される DNS レコードの TTL をチェックします。TTL が設定された最小 TTL より小さい場合、設定されている最小 TTL に置き換えられます。TTL が設定された最大 TTL より大きい場合、設定されている最大 TTL に置き換えられます。

NetScaler ADC では、ドメインに対する否定応答のキャッシュも可能です。否定応答は、要求されたドメインに関する情報が存在しないか、サーバーがクエリに対する応答を提供できないことを示します。この情報の保存はネガテ

ィブキャッシュと呼ばれます。ネガティブキャッシュは、ドメインでのクエリへの応答を高速化するのに役立ち、オプションでレコードタイプを指定することもできます。

否定応答は、次のいずれかになります。

- NXDOMAIN エラーメッセージ-ローカルキャッシュに否定的な応答が存在する場合、NetScaler ADC はエラーメッセージ（NXDOMAIN）を返します。応答がローカルキャッシュにない場合、クエリはサーバーに転送され、サーバーは NXDOMAIN エラーを NetScaler ADC に返します。NetScaler ADC は応答をローカルにキャッシュし、エラーメッセージをクライアントに返します。
- NODATA エラーメッセージ-クエリ内のドメイン名は有効だが、指定されたタイプのレコードが使用できない場合、NetScaler ADC は NODATA エラーメッセージを送信します。

NetScaler ADC は、DNS 要求の再帰的解決をサポートしています。再帰的解決では、リゾルバ (DNS クライアント) はドメイン名の再帰クエリをネームサーバに送信します。照会されたネームサーバがドメインに対して権限を持つ場合、要求されたドメイン名で応答します。それ以外の場合、NetScaler ADC は、要求されたドメイン名が見つかるまで、ネームサーバーに再帰的にクエリを実行します。

再帰クエリオプションを適用するには、まず再帰クエリオプションを有効にする必要があります。DNS ルックアップが失敗した場合に DNS リゾルバーが解決要求 (DNS 再試行) を送信する回数も設定できます。

NetScaler ADC を DNS フォワーダーとして構成できます。フォワーダは DNS 要求を外部のネームサーバに渡します。NetScaler ADC を使用すると、外部ネームサーバーを追加したり、ネットワーク外のドメインに名前解決を提供したりできます。NetScaler ADC では、名前検索の優先順位を DNS または Windows インターネットネームサービス（WINS）に設定することもできます。

ADC アプライアンスが **DNS** を使用してホスト名を対応する **IP** アドレスに解決できるようにします

注：アプライアンスのコマンドラインインターフェイス（CLI）にアクセスするには、SSH ユーティリティが必要です。

デフォルトでは、ADC アプライアンスはホスト名を対応する IP アドレスに解決できません。アプライアンスで名前解決を有効にするには、次のタスクを実行します。

1. ネームサーバを定義する。
2. DNS サフィックスを定義します。

注意事項

CLI から DNS ルックアップを実行します。/etc/resolv.conf ファイルのエントリが 127.0.0.2 の IP アドレスを指しているため、FreeBSD オペレーティングシステムのシェルプロンプトからの DNS ルックアップは失敗します。

次のコマンドは、`drill`コマンドで到達可能なアプライアンスの FreeBSD CLI の`shell`コマンドに置き換えられます。


```
- host
- dig
- getent/MIP
- nslookup
<!--NeedCopy-->
```

たとえば、`dig www.google.com @8.8.8.8`を実行してネームサーバー「8.8.8.8」の「A」レコード「www.google.com」を照会する代わりに、`drill www.google.com @8.8.8.8`コマンドを実行できます。`drill`コマンドは、`dig`コマンドとまったく同じように機能します。

```
root@lab# drill www.google.com @8.8.8.8
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 57980
;; flags: qr rd ra ; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;; www.google.com. IN A

;; ANSWER SECTION:
www.google.com. 300 IN A 142.250.187.196

;; AUTHORITY SECTION:

;; ADDITIONAL SECTION:

;; Query time: 53 msec
;; SERVER: 8.8.8.8
;; WHEN: Thu Jun 9 11:04:55 2022
;; MSG SIZE rcvd: 48
<!--NeedCopy-->
```

アプライアンスが SNIP アドレスで DNS サーバに ping できない場合、サーバステータスは [Down] と表示されます。アプライアンスがファイアウォールの内側にある場合、ping の成功は重要です。

CLI 設定

コマンドプロンプトで入力します。

```
add dns nameServer <Name_Server_IP_Address>
add dns suffix <DNS_Suffix>
<!--NeedCopy-->
```

構成を確認するには、次のように入力します。

```
show dns nameServer
show dns suffix
<!--NeedCopy-->
```

DNS 解決をテストするには、次のように入力します。

```
show dns addrec <Host_Name>
<!--NeedCopy-->
```


GUI 構成

1. トラフィック管理 **>** **DNS >** ネームサーバ **>** 追加に移動します。
2. [ネームサーバの作成] ダイアログボックスで、ネームサーバの IP アドレスを入力し、[作成] をクリックします。
3. トラフィック管理 **>** **DNS > DNS** サフィックス **>** 追加に移動します。
4. [**DNS** サフィックスの作成] ダイアログボックスで、すべてのホストクエリに使用する DNS サフィックス (example.com など) を入力し、[作成] をクリックします。

ラウンドロビン **DNS**

クライアントが DNS リソースレコードを検索する DNS 要求を送信すると、DNS 要求内の名前に解決される IP アドレスのリストを受信します。次に、クライアントはリスト内の IP アドレスの 1 つを使用します。通常は、最初のレコードまたは IP アドレスです。そのため、キャッシュの合計 TTL には 1 つのサーバが使用され、多数の要求が到着すると過負荷になります。

NetScaler ADC は、DNS 要求を受信すると、ラウンドロビン方式で DNS リソースレコードのリストの順序を変更して応答します。この機能はラウンドロビン *DNS* と呼ばれます。ラウンドロビン方式では、データセンター間でトラフィックが均等に分散されます。NetScaler ADC はこの機能を自動的に実行します。この動作を設定する必要はありません。

機能概要

NetScaler ADC が ADNS サーバーとして構成されている場合、レコードが構成された順序で DNS レコードが返されます。NetScaler ADC が DNS プロキシとして構成されている場合、サーバーからレコードを受信した順序で DNS レコードを返します。キャッシュに存在するレコードの順序は、サーバーからレコードを受信する順序と一致します。

NetScaler ADC は、ラウンドロビン方式で DNS 応答でレコードが送信される順序を変更します。最初の応答には最初のレコードが順番に含まれ、2 番目の応答には 2 番目のレコードが順番に含まれ、順序は同じ順序で続きます。したがって、同じ名前を要求するクライアントは、異なる IP アドレスに接続できます。

ラウンドロビン **DNS** の例　ラウンドロビン DNS の例として、次のように追加された DNS レコードを考えてみましょう。

```
1   add dns addRec ns1 1.1.1.1  add dns addRec ns1 1.1.1.2  add dns
       addRec ns1 1.1.1.3  add dns addRec ns1 1.1.1.4
2 <!--NeedCopy-->
```


ドメイン abc.com は、次のように NS レコードにリンクされています。

```
1   add dns nsrec abc.com. ns1
2 <!--NeedCopy-->
```

NetScaler ADC が ns1 の A レコードに対するクエリを受信すると、Address レコードは次のようにラウンドロビン方式で提供されます。最初の DNS 応答では、1.1.1.1 が最初のレコードとして提供されます。

```
1   ns1.                    1H IN A         1.1.1.1  ns1.
                           1H IN A         1.1.1.2  ns1.
                           1H IN A         1.1.1.3  ns1.
                           1H IN A         1.1.1.4
2 <!--NeedCopy-->
```

2 番目の DNS 応答では、2 番目の IP アドレス 1.1.1.2 が最初のレコードとして提供されます。

```
1   ns1.                    1H IN A         1.1.1.2  ns1.
                           1H IN A         1.1.1.3  ns1.
                           1H IN A         1.1.1.4  ns1.
                           1H IN A         1.1.1.1
2 <!--NeedCopy-->
```

3 番目の DNS 応答では、3 番目の IP アドレス 1.1.1.2 が最初のレコードとして提供されます。

```
1   ns1.                    1H IN A         1.1.1.3  ns1.
                           1H IN A         1.1.1.4  ns1.
                           1H IN A         1.1.1.1  ns1.
                           1H IN A         1.1.1.2
2 <!--NeedCopy-->
```

DNS リソースレコードを構成する

August 15, 2023

Citrix® ADC アプライアンスをゾーンの ADNS サーバーとして構成するときに、リソースレコードを構成します。リソースレコードが、アプライアンスが DNS プロキシサーバーであるゾーンに属している場合は、アプライアンスでリソースレコードを設定することもできます。アプライアンスでは、次のレコードタイプを設定できます。

- サービス記録
- AAAA レコード
- 住所レコード
- メール交換レコード
- ネームサーバーレコード
- カノニカルレコード
- ポインターレコード
- NAPTR レコード

- 権限記録の開始
- テキストレコード
- 認証局認証 (CAA) レコード

次の表に、NetScaler ADC アプライアンスのドメイン名レコードに対して構成できるレコードタイプを示します。たとえば、1 つのレコードに対して最大 25 個の IP アドレスを設定できます。

表 1. レコードタイプと番号は設定可能

レコードタイプ	レコード数
住所 (A)	25
IPv6 (AAAA)	5
メール交換 (MX)	12
ネームサーバー (NS)	16
サービス (SRV)	8
ポインター (PTR)	20
正規名 (CNAME)	1
権限の開始 (SOA)	1
テキスト (TXT)	20
命名機関ポインタ (NAPTR)	20
認証局認証 (CAA)	20

> 注:
>
> 特定のホスト名の IP アドレスの最大数は 25 です。ただし、異なる住所レコードの数は 25 を超える場合があります。

サービスの **SRV** レコードを作成する

August 15, 2023

SRV レコードは、NetScaler アプライアンスで利用できるサービスに関する情報を提供します。SRV レコードには次の情報が含まれます。

- サービスとプロトコルの名前
- ドメイン名

- TTL
- DNS クラス
- ターゲットの優先度
- 優先度が同じレコードのウェイト
- サービスのポート
- サービスのホスト名。

NetScaler は、優先度設定が最も低い SRV レコードを最初に選択します。サービスに同じ優先度の SRV レコードが複数ある場合、クライアントは重みフィールドを使用してどのホストを使用するかを決定します。

CLI を使用して SRV レコードを追加する

コマンドプロンプトで次のコマンドを入力して SRV レコードを追加し、構成を確認します。

```
1 -  add dns srvRec <domain> <target> -priority <positive_integer> -
     weight <positive_integer> -port <positive_integer> [-TTL <secs>]
2 -  sh dns srvRec <domain>
3 <!--NeedCopy-->
```

例:

```
1 > add dns srvRec _http._tcp.example.com nameserver1.com -priority 1 -
     weight 1 -port 80
2  Done
3 > show dns srvRec _http._tcp.example.com
4 1)      Domain Name : _http._tcp.example.com
5         Target Host : nameserver1.com
6         Priority : 1    Weight : 1
7         Port : 80       TTL : 3600 secs
8  Done
9 <!--NeedCopy-->
```

CLI を使用して SRV レコードを変更または削除する

- SRV レコードを変更するには、次のように入力します。
  - `set dns srvRec`コマンド
  - SRV レコードが設定されているドメインの名前
  - 関連サービスをホストするターゲットホストの名前
  - 変更するパラメータと新しい値
- SRV レコードを削除するには、次のように入力します。
  - `rm dns srvRec`コマンド
  - SRV レコードが設定されているドメインの名前
  - 関連サービスをホストするターゲットホストの名前

GUI を使用して SRV レコードを設定する

[トラフィック管理] > [DNS] > [レコード] > [SRV レコード] に移動し、SRV レコードを作成します。

ドメイン名の AAAA レコードを作成する

August 15, 2023

AAAA リソースレコードには 1 つの IPv6 アドレスが格納されます。

CLI を使用して AAAA レコードを追加する

コマンドプロンプトで次のコマンドを入力して AAAA レコードを追加し、構成を確認します。

```
-  add dns aaaaRec <hostName> <IPv6Address> ... [-TTL <secs>]
-  show dns aaaaRec <hostName>
<!--NeedCopy-->
```

例:

```
> add dns aaaaRec www.example.com 2001:0db8:0000:0000:0000:0000:1428:57
   ab
 Done
> show dns aaaaRec www.example.com
1)      Host Name : www.example.com
        Record Type : ADNS              TTL : 5 secs
        IPV6 Address : 2001:db8::1428:57ab
 Done
<!--NeedCopy-->
```

AAAA レコードと、そのドメイン名に関連付けられているすべての IPv6 アドレスを削除するには、`rm dns aaaaRec` コマンドと AAAA レコードが設定されているドメイン名を入力します。AAAA レコード内のドメイン名に関連付けられた IPv6 アドレスのサブセットのみを削除するには、次のように入力します。

- `rm dns aaaaRec` コマンド
- AAAA レコードが設定されているドメイン名
- 削除したい IPv6 アドレス

GUI を使用して AAAA レコードを追加する

[トラフィック管理] > [DNS] > [レコード] > [AAAA レコード] に移動し、AAAA レコードを作成します。

ドメイン名のアドレスレコードを作成する

August 15, 2023

アドレス (A) レコードは、ドメイン名を IPv4 アドレスにマッピングする DNS レコードです。

グローバルサーバー負荷分散 (GSLB) に参加しているホストのアドレスレコードは削除できません。ただし、GSLB 仮想サーバーからドメインをバインド解除すると、NetScaler は GSLB ドメインに追加されたアドレスレコードを削除します。手動で削除できるのは、ユーザー設定のレコードだけです。NS、MX、CNAME などのレコードで参照されているホストのレコードは削除できません。

CLI を使用して住所レコードを追加する

コマンドプロンプトで次のコマンドを入力してアドレスレコードを追加し、構成を確認します。

```
-  add dns addRec <hostName> <IPAddress> [-TTL <secs>]
-  show dns addRec <hostName>
<!--NeedCopy-->
```

例:

```
> add dns addRec ns.example.com 192.0.2.0
 Done
> show dns addRec ns.example.com
1)      Host Name : ns.example.com
        Record Type : ADNS                      TTL : 5 secs
        IP Address : 192.0.2.0
 Done
<!--NeedCopy-->
```

アドレスレコードとそのドメイン名に関連付けられているすべての IP アドレスを削除するには、`rm dns addRec` コマンドと、アドレスレコードが設定されているドメイン名を入力します。アドレスレコードのドメイン名に関連付けられた IP アドレスのサブセットのみを削除するには、次のように入力します。

- `rm dns addRec` コマンド
- アドレスレコードが設定されているドメイン名
- 削除する IP アドレス

GUI を使用して住所レコードを追加する

[トラフィック管理] **> [DNS] > [**レコード] **> [**アドレスレコード] に移動し、アドレスレコードを作成します。

メール交換サーバーの **MX** レコードを作成する

August 15, 2023

メールエクスチェンジ (MX) レコードは、インターネット経由でメールメッセージを転送するために使用されます。MX レコードには、使用する MX サーバーを指定する MX 設定が含まれています。MX プリファレンス値の範囲は 0 から 65536 です。MX レコードには固有の MX プリファレンス番号が含まれています。MX レコードの MX プリファレンスと TTL 値を設定できます。

電子メールメッセージがインターネット経由で送信されると、メール転送エージェントはドメイン名の MX レコードを要求する DNS クエリを送信します。このクエリは、ドメインのメール交換サーバーのホスト名のリストと優先番号を返します。MX レコードがない場合、そのドメインのアドレスレコードが要求されます。1 つのドメインに複数のメール交換サーバーを配置できます。

CLI を使用して **MX** レコードを追加する

コマンドプロンプトで次のコマンドを入力して MX レコードを追加し、構成を確認します。

```
1 -  add dns mxRec <domain> -mx <string> -pref <positive_integer> [-TTL <
     secs>]
2 -  show dns mxRec <domain>
3 <!--NeedCopy-->
```

例**:**

```
1 > add dns mxRec example.com -mx mail.example.com -pref 1
2  Done
3 > show dns mxRec example.com
4 1)      Domain : example.com    MX Name : mail.example.com
5         Preference :   1        TTL : 5 secs
6  Done
7 <!--NeedCopy-->
```

CLI を使用して **MX** レコードを変更または削除する

- MX レコードを変更するには、`set dns mxRec` コマンド、MX レコードが設定されているドメインの名前、MX レコードの名前、および変更するパラメータを新しい値とともに入力します。
- TTL パラメータをデフォルト値に設定するには、`unset dns mxRec` コマンド、MX レコードが設定されているドメインの名前、MX レコードの名前、および TTL 値（TTL 値なし）を入力します。`unset dns mxRec` コマンドを使用して設定を解除できるのは TTL パラメータだけです。
- MX レコードを削除するには、`rm dns mxRec` コマンド、MX レコードが設定されているドメインの名前、および MX レコードの名前を入力します。

GUI を使用して MX レコードを追加する

[トラフィック管理**] > [DNS] > [**レコード**] > [**メール交換レコード**]** に移動し、MX レコードを作成します。

権限のあるサーバーの NS レコードの作成

August 15, 2023

ネームサーバー (NS) レコードは、ドメインの権限のあるサーバーを指定します。最大 16 個の NS レコードを設定できます。NS レコードを使用して、サブドメインの制御を DNS サーバーに委任できます。

CLI を使用して NS レコードを作成する

コマンドプロンプトで次のコマンドを入力して NS レコードを作成し、構成を確認します。

```
-  add dns nsRec <domain> <nameServer> [-TTL <secs>]
-  show dns nsRec <domain>
<!--NeedCopy-->
```

例:

```
> add dns nsRec example.com nameserver1.example.com
 Done
> show dns nsRec example.com
1)      Domain : example.com    NameServer : nameserver1.example.com
        TTL : 5 sec
 Done
<!--NeedCopy-->
```

NS レコードを削除するには、`rm dns nsRec` コマンド、NS レコードが属するドメインの名前、およびネームサーバーの名前を入力します。

GUI を使用して NS レコードを作成する

[トラフィック管理**] > [DNS] > [**レコード**] > [**ネームサーバレコード**]** に移動し、NS レコードを作成します。

サブドメインの CNAME レコードを作成する

August 15, 2023

正規名レコード (CNAME レコード) は DNS 名のエイリアスです。これらのレコードは、複数のサービスが DNS サーバーにクエリを実行する場合に役立ちます。アドレス (A) レコードを持つホストは CNAME レコードを持つことはできません。

プロキシモードの NetScaler アプライアンスは、サーバーではなくキャッシュからアドレスレコードを要求することがあります。

CLI を使用して CNAME レコードを追加する

コマンドプロンプトで次のコマンドを入力して CNAME レコードを作成し、構成を確認します。

```
-  add dns cnameRec <aliasName> <canonicalName> [-TTL <secs>]
-  show dns cnameRec <aliasName>
<!--NeedCopy-->
```

例:

```
> add dns cnameRec www.example.com www.examplenw.com
 Done
> show dns cnameRec www.example.com
        Alias Name      Canonical Name  TTL
1)      www.example.com        www.examplenw.com       5 secs
 Done
<!--NeedCopy-->
```

特定のドメインの CNAME レコードを削除するには、`rm dns cnameRec` コマンドとドメイン名のエイリアスを入力します。

GUI を使用して CNAME レコードを追加する

[トラフィック管理] **> [DNS] > [**レコード**] > [**正規レコード**]** に移動し、**CNAME** レコードを作成します。

CNAME レコードをキャッシュする

プロキシモードでデプロイした場合、ADC アプライアンスは必ずしもアドレスレコードのクエリをバックエンドサーバーに送信するとは限りません。この現象は、アドレスレコードのクエリに対する応答として CNAME チェーンの一部がキャッシュに存在している場合に発生します。ADC が CNAME レコードの一部をキャッシュし、キャッシュからクエリを処理する条件はほとんどありません。条件は次のとおりです。

- NetScaler はプロキシモードで展開する必要があります。
- バックエンドサーバーからの応答には CNAME チェーンが必要です。その場合、回答セクションの最後のエントリのレコードタイプは CNAME で、質問タイプは CNAME ではない必要があります。
- バックエンドサーバーからの応答を No-data または NX ドメインにすることはできません。
- バックエンドサーバーからの応答は、信頼できる応答でなければなりません。

通信ドメインの **NAPTR** レコードを作成する

August 15, 2023

NAPTR（ネーミングアドレスポインタ）は、電気通信分野で最も一般的に使用されている DNS レコードの 1 つです。NAPTR レコードは、インターネットテレフォニーアドレス空間をインターネットアドレス空間にマッピングします。そのため、モバイルデバイスが適切なサーバーにリクエストを送信できるようになります。NAPTR レコードとサービスレコード (SRV) を組み合わせると、複数のレコードをチェーンして複雑な書き換えルールを形成し、新しいドメインラベルや統一リソース識別子 (URI) を生成できます。NAPTR の DNS コードは 35 です。

NetScalers は、ADNS モードとプロキシモードの 2 つのモードで NAPTR をサポートします。プロキシモードでは、ADC はサーバーからの応答をキャッシュし、キャッシュされたレコードを使用して今後のクエリをサーバー化します。NetScaler では、特定のドメインに最大 20 個の NAPTR レコードを追加できます。NetScaler は、DNS NAPTR レコードクエリへの応答をキャッシュします。それ以降の NAPTR レコードのリクエストは、キャッシュから処理されます。

CLI を使用して **NAPTR** レコードを作成する

コマンドプロンプトで次のコマンドを入力して NAPTR レコードを追加し、構成を確認します。

通信ドメインの **NAPTR** レコードを作成する

NAPTR（ネーミングアドレスポインタ）は、電気通信分野で最も一般的に使用されている DNS レコードの 1 つです。NAPTR レコードは、インターネットテレフォニーアドレス空間をインターネットアドレス空間にマッピングします。そのため、モバイルデバイスが適切なサーバーにリクエストを送信できるようになります。NAPTR レコードとサービスレコード (SRV) を組み合わせると、複数のレコードをチェーンして複雑な書き換えルールを形成し、新しいドメインラベルや統一リソース識別子 (URI) を生成できます。NAPTR の DNS コードは 35 です。

NetScalers は、ADNS モードとプロキシモードの 2 つのモードで NAPTR をサポートします。プロキシモードでは、ADC はサーバーからの応答をキャッシュし、キャッシュされたレコードを使用して今後のクエリをサーバー化します。NetScaler では、特定のドメインに最大 20 個の NAPTR レコードを追加できます。NetScaler は、DNS NAPTR レコードクエリへの応答をキャッシュします。それ以降の NAPTR レコードのリクエストは、キャッシュから処理されます。

CLI を使用して **NAPTR** レコードを作成する

コマンドプロンプトで次のコマンドを入力して NAPTR レコードを追加し、構成を確認します。

napTrec を追加してください \[flags]\[services]\(regexp|-replacement)\[-TTL]# 通信ドメインの NAPTR レコードを作成する

NAPTR（ネーミングアドレスポインタ）は、電気通信分野で最も一般的に使用されている DNS レコードの 1 つです。NAPTR レコードは、インターネットテレフォニーアドレス空間をインターネットアドレス空間にマッピングします。そのため、モバイルデバイスが適切なサーバーにリクエストを送信できるようになります。NAPTR レコードとサービスレコード (SRV) を組み合わせると、複数のレコードをチェーンして複雑な書き換えルールを形成し、新しいドメインラベルや統一リソース識別子 (URI) を生成できます。NAPTR の DNS コードは 35 です。

NetScalers は、ADNS モードとプロキシモードの 2 つのモードで NAPTR をサポートします。プロキシモードでは、ADC はサーバーからの応答をキャッシュし、キャッシュされたレコードを使用して今後のクエリをサーバー化します。NetScaler では、特定のドメインに最大 20 個の NAPTR レコードを追加できます。NetScaler は、DNS NAPTR レコードクエリへの応答をキャッシュします。それ以降の NAPTR レコードのリクエストは、キャッシュから処理されます。

CLI を使用して **NAPTR** レコードを作成する

コマンドプロンプトで次のコマンドを入力して NAPTR レコードを追加し、構成を確認します。

CLU を使用して **NAPTR** レコードを削除する

```
rm dns naptrRec<domain> (<order> <preference> [-flags <string>] [-
services <string>] (-regexp <expression> | -replacement <string>))| -
recordId <positive_integer>@)
```

GUI を使用して **NAPTR** レコードを設定します

[トラフィック管理**] > [DNS] > [**レコード**] > [NAPTR** レコード**]** に移動し、**NAPTR** レコードを作成します。

IPv4 および **IPv6** アドレスの **PTR** レコードを作成する

August 15, 2023

ポインタ (PTR) レコードは、IP アドレスをそのドメイン名に変換します。IPv4 PTR レコードは、IP アドレスのオクテットを逆の順序で表し、文字列「in-addr.arpa」を付けて表します。「最後に追加されました。たとえば、IP アドレス 1.2.3.4 の PTR レコードは 4.3.2.1.in-addr.arpa です。

IPv6 アドレスは IP6.ARPA というドメインでリバースマップされます。IPv6 リバースマップでは、RFC 3596 で定義されているように、サフィックス「.IP6.ARPA」が付いたドットで区切られた一連のニブルを使用します。たとえば、4321:0:1:2:3:4:567:89 ab というアドレスに対応する逆引きドメイン名は B.A.9.8.7.6.5.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.2.3.4.ip6.ARPA になります。

CLI を使用して PTR レコードを追加する

コマンドプロンプトで次のコマンドを入力して PTR レコードを追加し、構成を確認します。

```
1 -  add dns ptrRec <reverseDomain> <domain> [-TTL <secs>]
2 -  show dns ptrRec <reverseDomain>
3 <!--NeedCopy-->
```

例:

```
1 > add dns ptrRec 0.2.0.192.in-addr.arpa example.com
2  Done
3 > show dns ptrRec 0.2.0.192.in-addr.arpa
4 1)      Reverse Domain Name : 0.2.0.192.in-addr.arpa
5         Domain Name : example.com              TTL : 3600 secs
6  Done
7 <!--NeedCopy-->
```

PTR レコードを削除するには、`rm dns ptrRec` コマンドと PTR レコードに関連付けられているリバースドメイン名を入力します。

GUI を使用して PTR レコードを追加する

[トラフィック管理] > [DNS] > [レコード] > [PTR レコード] に移動し、**PTR** レコードを作成します。

権威情報の SOA レコードを作成する

August 15, 2023

Start of Authority (SOA) レコードはゾーンの頂点でのみ作成され、そのゾーンに関する情報が含まれます。レコードには、他のパラメータの中でも、プライマリネームサーバー、連絡先情報 (電子メール)、およびレコードのデフォルト (最小) 有効期間 (TTL) 値が含まれます。

CLI を使用して SOA レコードを作成する

コマンドプロンプトで次のコマンドを入力して SOA レコードを追加し、構成を確認します。

```
1 -  add dns soaRec <domain> -originServer <originServerName> -contact <
     contactName>
2 -  sh dns soaRec <do main>
3 <!--NeedCopy-->
```

例:


```
> add dns soaRec example.com -originServer nameserver1.example.com -
    contact admin.example.com
 Done
> show dns soaRec example.com
1)      Domain Name : example.com
        Origin Server : nameserver1.example.com
        Contact : admin.example.com
        Serial No. : 100        Refresh : 3600 secs     Retry : 3 secs
        Expire : 3600 secs      Minimum : 5 secs        TTL : 3600 secs
 Done
<!--NeedCopy-->
```

CLI を使用して **SOA** レコードを変更または削除する

- SOA レコードを変更するには、`set dns soaRec` コマンド、レコードが設定されているドメインの名前、変更するパラメータを新しい値とともに入力します。
- SOA レコードを削除するには、`rm dns soaRec` コマンドと、レコードが設定されているドメインの名前を入力します。

GUI を使用して **SOA** レコードを設定する

[トラフィック管理**] > [DNS] > [**レコード**] > [SOA** レコード**]** に移動し、SOA レコードを作成します。

説明テキストを保持するための **TXT** レコードの作成

August 15, 2023

ドメインホストは、情報提供を目的として TXT レコードを保存します。TXT レコードの RDATA コンポーネントは、可変長の 1 つ以上の文字列で構成され、受信者がドメインについて知っておく必要のあるほぼすべての情報を格納できます。また、サービスプロバイダー、連絡先、電子メールアドレス、および関連する詳細に関する情報も含まれる場合があります。SPF（送信者ポリシーフレームワーク）保護は、TXT レコードの最も顕著な使用例でした。

NetScaler ADC アプライアンスのすべての構成タイプ（権限のある DNS、DNS プロキシ、エンドリゾルバー、およびフォワーダー構成）は TXT レコードをサポートします。ドメインには最大 20 の TXT リソースレコードを追加できます。各リソースレコードは、内部で生成された一意のレコード ID で保存されます。TXT リソースレコードには最大 6 つの文字列を含めることができ、各文字列には最大 255 文字を含めることができます。レコードの ID を表示し、それを使用してレコードを削除できます。ただし、TXT リソースレコードは変更できません。

CLI を使用して **TXT** リソースレコードを作成する

コマンドプロンプトで次のコマンドを入力して TXT リソースレコードを作成し、構成を確認します。


```
-  add dns txtRec <domain> <string> ... [-TTL <secs>]
-  show dns txtRec [<domain> | -type <type>]
<!--NeedCopy-->
```

例:

```
> add dns txtRec www.example.com "Contact: Mark" "Email: mark@example.
   com" -TTL 36000
 Done
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 13783       TTL : 36000 secs
     Record Type : ADNS
         "Contact: Mark"
         "Email: mark@example.com"
 Done
<!--NeedCopy-->
```

CLI を使用して TXT リソースレコードの文字列を分割する

255 文字を超える文字列がある場合は、6 つの文字列の制限を考慮して文字列を分割できます。各文字列の長さは 254 バイトです。

```
add dns txtrec domain.com "string1" "string2" string3" "string4"
<!--NeedCopy-->
```

例:

```
add dns txtrec exampledomain.com "Contact: Evan" "Email: evan@example.
   com" "Contact: Mark" "Email: mark1@example.com"
<!--NeedCopy-->
```

CLI を使用して TXT リソースレコードを削除する

コマンドプロンプトで次のコマンドを入力して TXT リソースレコードを削除し、構成を確認します。

```
-  rm dns txtRec <domain> (<string> ... | -recordId <positive_integer>)
-  show dns txtRec [<domain> | -type <type>]
<!--NeedCopy-->
```

例:

`show dns txtRec` 最初にコマンドを使用すると、次に示すように、削除する TXT リソースレコードのレコード ID を表示できます。

```
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 36865       TTL : 36000 secs
     Record Type : ADNS
         "Contact: Evan"
```



```
4           "Email: evan@example.com"
5  2)  Domain : www.example.com    Record id: 14373       TTL : 36000 secs
       Record Type : ADNS
6           "Contact: Mark"
7           "Email: mark1@example.com"
8   Done
9  <!--NeedCopy-->
```

TXT レコードを削除する簡単な方法は、レコード ID を使用することです。文字列を指定する場合は、レコードに保存されている順序で入力してください。次の例では、TXT レコードはそのレコード ID を使用して削除されます。

```
1  >rm dns txtRec www.example.com -recordID 36865
2   Done
3  > show dns txtRec www.example.com
4  1)  Domain : www.example.com    Record id: 14373       TTL : 36000 secs
       Record Type : ADNS
5           "Contact: Mark"
6           "Email: mark1@example.com"
7   Done
8  <!--NeedCopy-->
```

GUI を使用して **TXT** レコードを設定する

[トラフィック管理] **>** **[DNS]** **>** [レコード] **>** **[TXT** レコード] に移動し、**TXT** レコードを作成します。

ドメイン名の **CAA** レコードを作成する

August 15, 2023

認証局認証 (CAA) は、ドメイン所有者がドメインの SSL 証明書を発行できる認証局 (CA) を指定できる DNS レコードの一種です。

サービスへの安全な接続には、ホストの ID を保証し、セキュリティで保護されたチャネルを確立するために SSL/TLS 証明書が必要です。CAA レコードがないと、誰でもドメインの証明書署名要求 (CSR) を生成し、任意の CA によって署名された証明書を取得できるため、セキュリティリスクが発生する可能性があります。

CAA レコードは、ドメイン所有者がドメインの証明書を発行できる認証局を宣言できるようにすることで、Web プレゼンスをさらに保護します。認証されていない CA からの証明書のリクエストがある場合、CAA レコードはドメイン所有者に同じことを通知します。ドメインの CAA レコードが存在しない場合、どの CA もそのドメインの証明書を発行できます。

NetScaler ADC アプライアンスは、次のモードで DNS CAA レコードをサポートします。

- プロキシ：アプライアンスは、バックエンドサーバーからの CAA レコード応答をキャッシュし、キャッシュからの同じタイプのさらなるクエリに応答します。

- **ADNS**: アプライアンスは、構成された DNS レコードからの CAA レコードタイプ DNS クエリに応答します。

注:

> - ドメイン名ごとに最大 20 の CAA レコードを追加できます。
> - 再帰的なリゾルバーモードとフォワーダーモードはサポートされていません。

CLI を使用して CAA レコードを追加する

コマンドプロンプトで、次のコマンドを入力します。

```
add dns caaRec <domain> <issuer-string> -tag <tag-string> -flag [None|Critical] [-TTL <secs>]
<!--NeedCopy-->
```

例:

```
> add dns caaRec newdomain string1 -tag Issue -flag None [-TTL 3600]
<!--NeedCopy-->
```

コマンドの詳細を表示

```
> show dns caaRec

1)  Domain : newdomain  ECS Subnet : None      Record id: 39423 TTL : 3600 secs Record Type : ADNS

Value: string1

Tag: issue

Flag: NONE

2)  Domain : test.com  ECS Subnet : None      Record id: 2572   TTL : 5 secs    Record Type : ADNS

Value: ca1.test.com

Tag: issue

Flag: NONE
<!--NeedCopy-->
```

CAA レコードを削除するには、コマンドプロンプトで次のコマンドを入力します。

```
rm dns caaRec <domain> <issuer-string> -tag <tag-string> | -recordId <positive_integer>@)
<!--NeedCopy-->
```

例:


```
rm dns caaRec newdomain -recordId 39423
<!--NeedCopy-->
```

> 注:
>
> -recordId @ はクラスタではサポートされていません。

GUI を使用して **CAA** レコードを追加する

[トラフィック管理] **> [DNS] > [**レコード**] > [CAA** レコード] に移動し、アドレスレコードを作成します。

DNS 統計の表示

August 15, 2023

NetScaler アプライアンスによって生成された DNS 統計を表示できます。DNS 統計には、ランタイム、構成、およびエラー統計が含まれます。

CLI を使用して **DNS** レコードの統計情報を表示する

コマンドプロンプトで入力します。

```
stat dns
```

例:

```
> stat dns
DNS Statistics

Runtime Statistics
Dns queries                             21
NS queries                               8
SOA queries                             18
.
.
.
Configuration Statistics
AAAA records                            17
A records                               36
MX records                               9
.
.
.
Error Statistics
Nonexistent domain                      17
```



```
No AAAA records                          0
No A records                            13
.
.
.
Done
<!--NeedCopy-->
```

GUI を使用して **DNS** レコードの統計情報を表示する

1. [トラフィック管理] **> [DNS**] に移動します。
2. 詳細ペインで、[統計] をクリックします。

DNS ゾーンを構成する

August 15, 2023

NetScaler アプライアンスの DNS ゾーンエンティティにより、アプライアンス上のドメインの所有が容易になります。アプライアンスのゾーンでは、ゾーンに DNS セキュリティ拡張（DNSSEC）を実装したり、ゾーンの DNSSEC 操作を DNS サーバーからアプライアンスにオフロードしたりすることもできます。DNSSEC 署名操作は、DNS ゾーン内のすべてのリソースレコードに対して実行されます。そのため、ゾーンに署名する場合、またはゾーンの DNSSEC 操作をオフロードする場合は、まず NetScaler アプライアンスでゾーンを作成する必要があります。

次のシナリオでは、アプライアンスに DNS ゾーンを作成します。

- NetScaler アプライアンスはゾーン内のすべてのレコードを所有します。つまり、アプライアンスはゾーンの権限のある DNS サーバーとして動作しています。ゾーンは ProxyMode パラメータを NO に設定して作成する必要があります。
- NetScaler アプライアンスは、ゾーン内のレコードのサブセットのみを所有します。ゾーン内の他のすべてのリソースレコードは、一連のバックエンドネームサーバーでホストされます。アプライアンスは、これらのバックエンドサーバーの DNS プロキシサーバーとして構成されます。NetScaler アプライアンスがゾーン内のリソースレコードのサブセットのみを所有する一般的な構成は、グローバルサーバー負荷分散（GSLB）構成です。NetScaler アプライアンスは GSLB ドメイン名のみを所有し、バックエンドネームサーバーは他のすべてのレコードを所有します。ゾーンは ProxyMode パラメータを YES に設定して作成する必要があります。
- ゾーンの DNSSEC 操作を、権限のある DNS サーバーからアプライアンスにオフロードしたい。ゾーンは ProxyMode パラメータを YES に設定して作成する必要があります。ゾーンにさらに設定する必要がある場合があります。

現在のトピックでは、最初の 2 つのシナリオでゾーンを作成する方法について説明します。アプライアンスに DNSSEC 操作をオフロードするためのゾーンを構成する方法の詳細については、「NetScaler ADC アプライアンスへの DNSSEC 操作のオフロード」を参照してください。

> 注
>
> ADC アプライアンスがゾーンの権限のある DNS サーバーとして動作している場合は、ゾーンを作成する前に、そのゾーンの Start of Authority（SOA）レコードとネームサーバー（NS）レコードを作成する必要があります。NetScaler ADC がゾーンの DNS プロキシサーバーとして動作している場合は、NetScaler ADC アプライアンスで SOA レコードと NS レコードを作成しないでください。SOA レコードと NS レコードの作成の詳細については、「DNS リソースレコードの設定」を参照してください。

ゾーンを作成すると、ゾーン名で終わる既存のドメイン名およびリソースレコードはすべて、自動的にゾーンの一部として処理されます。また、ゾーンの名前と一致するサフィックスで作成された新しいリソースレコードは、暗黙的にゾーンに含まれます。

CLI を使用して NetScaler アプライアンスに DNS ゾーンを作成します

コマンドプロンプトで次のコマンドを入力して NetScaler アプライアンスに DNS ゾーンを追加し、構成を確認します。

```
-  add dns zone <zoneName> -proxyMode ( YES | NO )
-  show dns zone [<zoneName> | -type <type>]
<!--NeedCopy-->
```

例:

```
> add dns zone example.com -proxyMode Yes
 Done
> show dns zone example.com
          Zone Name : example.com
          Proxy Mode : YES
Done
<!--NeedCopy-->
```

CLI を使用して DNS ゾーンを変更または削除する

- DNS ゾーンを変更するには、`set dns zone` コマンド、DNS ゾーンの名前、変更するパラメータを新しい値とともに入力します。
- DNS ゾーンを削除するには、`rm dns zone` コマンドと DNS ゾーンの名前を入力します。

GUI を使用して DNS ゾーンを設定する

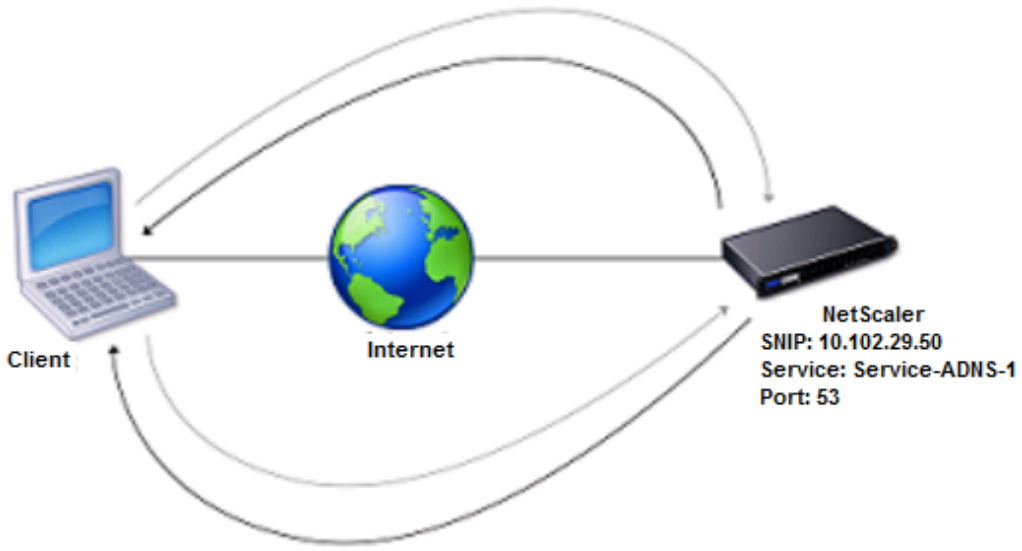
[トラフィック管理] **> [DNS] > [**ゾーン] に移動し、DNS ゾーンを作成します。

NetScaler ADC を ADNS サーバーとして構成する

August 15, 2023

ADC アプライアンスは、ドメインの権限のあるドメインネームサーバー (ADNS) として機能するように構成できます。NetScaler はドメインの ADNS サーバーとして、ドメインに属するあらゆる種類の DNS レコードに対する DNS 要求を解決します。NetScaler をドメインの ADNS サーバーとして機能するように構成するには、ADNS サービスを作成し、NetScaler 上のドメインの NS レコードとアドレスレコードを構成する必要があります。ADNS サービスは、サブネット IP アドレス (SNIP) または別の IP アドレスを使用して設定できます。次のトポロジー図は、構成例と要求と応答の流れを示しています。

図 1: ADNS としての NetScaler



次の表は、前述のトポロジ図に示した ADNS サービスに設定されているパラメータを示しています。

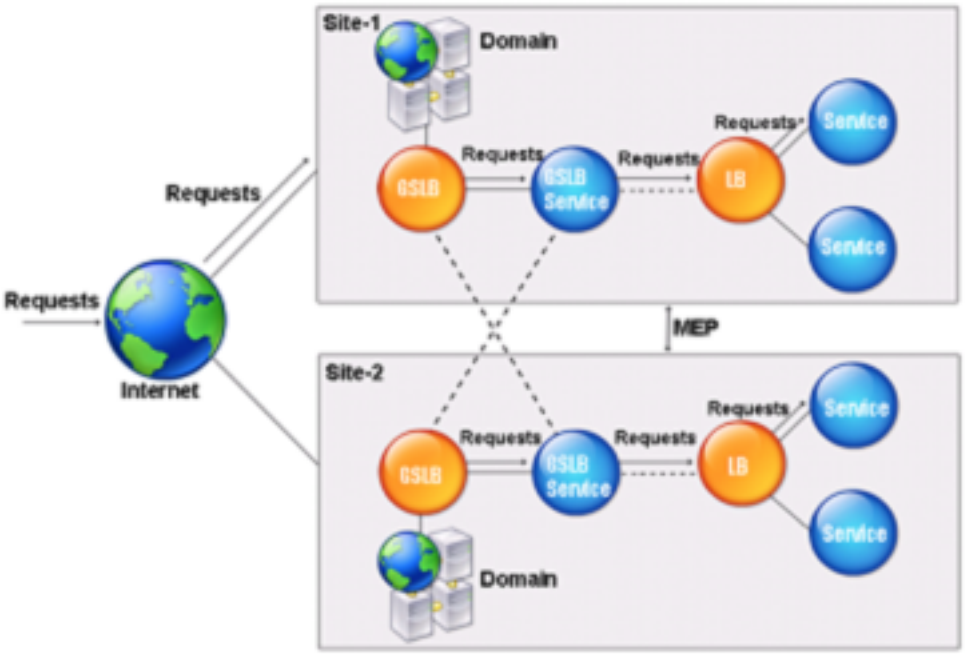
エンティティタイプ	名前	IP アドレス	種類	ポート
ADNS サービス	Service-ADNS-1	10.102.29.51	ADNS	53

表 1. ADNS サービス設定の例

ADNS セットアップを構成するには、ADNS サービスを構成する必要があります。ADNS サービスの構成手順については、「負荷分散」を参照してください。

DNS 解決中、ADNS サーバーは DNS プロキシまたはローカル DNS サーバーに NetScaler ADC にドメインの IP アドレスを問い合わせるよう指示します。NetScaler はドメインに対して権限があるため、IP アドレスを DNS プロキシまたはローカル DNS サーバーに送信します。次の図は、GSLB 構成における ADNS サーバーの配置と役割を示しています。

図 2: GSLB エンティティモデル



注: ADNS モードでは、SOA レコードと ADNS レコードを削除すると、NetScaler ADC がホストするドメインでは機能しません。ANY クエリ（ANY クエリの詳細については、「DNSANY クエリ」を参照）、および NODATA や NXDOMAIN などのネガティブレスポンス。

ADNS サービスを作成する

ADNS サービスは、グローバルサービスの負荷分散に使用されます。GSLB セットアップの作成の詳細については、「グローバルサーバーの負荷分散」を参照してください。ADNS サービスを追加、変更、有効化、無効化、および削除できます。ADNS サービスの作成手順については、「サービスの構成」を参照してください。

注**:** SNIP または任意の新しい IP アドレスを使用するように ADNS サービスを構成できます。

ADNS サービスを作成すると、NetScaler は構成された ADNS サービスの IP とポートで DNS クエリに応答します。

ADNS サービスのプロパティを表示して構成を確認できます。名前、状態、IP アドレス、ポート、プロトコル、最大クライアント接続数などのプロパティを表示できます。

TCP を使用するように ADNS セットアップを設定

デフォルトでは、一部のクライアントは DNS にユーザーデータグラムプロトコル (UDP) を使用します。UDP パケットのペイロード長は 512 バイトに制限されています。サイズが 512 バイトを超えるペイロードを処理するには、クライアントは TCP を使用する必要があります。TCP 経由の DNS 通信を有効にするには、DNS に TCP プロトコルを使用するように NetScaler アプライアンスを構成する必要があります。次に、NetScaler は DNS 応答パケットにトランケーションビットを設定します。切り捨てビットは、応答が UDP には大きすぎるため、クライアントは TCP 接続を介して要求を送信する必要があることを示します。次に、クライアントはポート 53 の TCP プロトコルを使用し、NetScaler への新しい接続を開きます。NetScaler は、ADNS サービスの IP アドレスを使用してポート 53 で受信し、クライアントからの新しい TCP 接続を受け入れます。

TCP プロトコルを使用するように NetScaler ADC を構成するには、ADNS_TCP サービスを構成する必要があります。ADNS_TCP サービスの作成手順については、負荷分散を参照してください。

> 重要
>
> NetScaler が DNS に UDP を使用し、UDP のペイロード長が 512 バイトを超える場合にのみ TCP を使用するように構成するには、ADNS および ADNS_TCP サービスを構成する必要があります。ADNS_TCP サービスの IP アドレスは ADNS サービスの IP アドレスと同じでなければなりません。

DNS リソースレコードの追加

ADNS サービスを作成したら、DNS レコードを追加できます。DNS レコードの追加手順については、「DNS リソースレコードの設定」を参照してください。

ADNS サービスの削除

サービスの削除手順については、「負荷分散」を参照してください。

ドメインの委任を構成する

ドメイン委任とは、ドメインスペースの一部に対する責任を別のネームサーバーに割り当てるプロセスです。そのため、ドメインを委任する際、クエリに応答する責任は別の DNS サーバーに委任されます。委任は NS レコードを使用します。

次の例では、sub1.abc.com は abc.com のサブドメインです。この手順では、サブドメインをネームサーバー ns2.sub1.abc.com に委任し、ns2.sub1.abc.com のアドレスレコードを追加する手順について説明します。

ドメイン委任を設定するには、次のセクションで説明する次のタスクを実行する必要があります。

1. ドメインの SOA レコードを作成します。
2. NS レコードを作成して、ドメインのネームサーバーを追加します。
3. ネームサーバーのアドレスレコードを作成します。
4. NS レコードを作成してサブドメインを委任します。
5. ネームサーバーのグルーレコードを作成します。

SOA レコードの作成

SOA レコードの設定手順については、「信頼できる情報の SOA レコードの作成」を参照してください。

ネーム・サーバの **NS** レコードの作成

NS レコードの構成手順については、「権限のあるサーバーの NS レコードの作成」を参照してください。[ネームサーバー**]** リストで、プライマリ権限のあるネームサーバー（ns1.abc.com など）を選択します。

住所レコードを作成する

アドレスレコードの設定手順については、「ドメイン名のアドレスレコードを作成する」を参照してください。[ホスト名] および [IP アドレス] テキストボックスに、DNS アドレスレコードのドメイン名と IP アドレス（たとえば、ns1.abc.com と 10.102.11.135）を入力します。

ドメイン委任用の **NS** レコードを作成する

NS レコードの構成手順については、「権限のあるサーバーの NS レコードの作成」を参照してください。[ネームサーバー**]** リストで、プライマリ権限のあるネームサーバー（ns2.sub1.abc.com など）を選択します。

グルーレコードの作成

NS レコードは通常 SOA レコードの直後に定義されます (制限はありません)。ドメインには少なくとも 2 つの NS レコードが必要です。NS レコードがドメイン内で定義されている場合、そのレコードには一致するアドレスレコードが必要です。このアドレスレコードはグルーレコードと呼ばれます。グルーレコードは、DNS クエリを高速化します。

サブドメインのグルーレコードを追加する手順については、「アドレス (A) レコードの追加手順」、「DNS リソースレコードの設定」を参照してください。

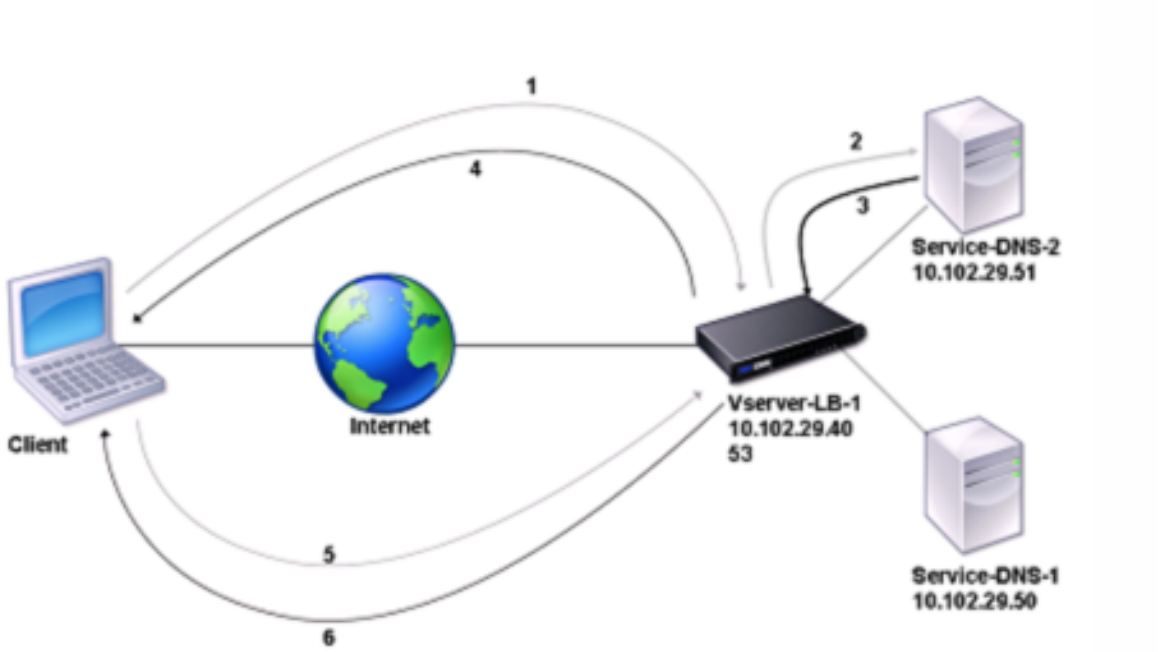
アドレスレコードの設定手順については、「ドメイン名のアドレスレコードを作成する」を参照してください。[ホスト名] および [IP アドレス] テキストボックスに、DNS アドレスレコードのドメイン名と IP アドレスを入力します（例：それぞれ ns2.sub1.abc.com と 10.102.12.135）。

NetScaler アプライアンスを DNS プロキシサーバーとして構成する

August 15, 2023

DNS プロキシサーバーとして、ADC アプライアンスは単一の DNS サーバーまたは DNS サーバーグループのプロキシとして機能できます。リクエストとレスポンスの流れは、次のサンプルトポロジー図に示されています。

図 1：DNS プロキシとしての NetScaler



デフォルトでは、NetScaler アプライアンスは DNS ネームサーバーからの応答をキャッシュします。アプライアンスは DNS クエリを受信すると、クエリされたドメインをキャッシュで確認します。クエリ対象ドメインのアドレスがキャッシュに存在する場合、NetScaler は対応するアドレスをクライアントに返します。それ以外の場合は、クエリを DNS ネームサーバーに転送し、DNS ネームサーバーがアドレスの可用性をチェックして NetScaler に返します。その後、NetScaler はアドレスをクライアントに返します。

以前にキャッシュされたドメインへのリクエストの場合、NetScaler は構成済みの DNS サーバーにクエリを実行せずに、キャッシュからドメインのアドレスレコードを提供します。

アプライアンスは、レコードの存続可能時間（TTL）値が設定値に達すると、キャッシュに保存されているレコードを破棄します。期限切れのレコードをリクエストするクライアントは、NetScaler がサーバーからレコードを取得し

てキャッシュを更新するまで待つ必要があります。この遅延を回避するために、NetScaler はレコードの有効期限が切れる前にサーバーからレコードを取得してキャッシュをプロアクティブに更新します。

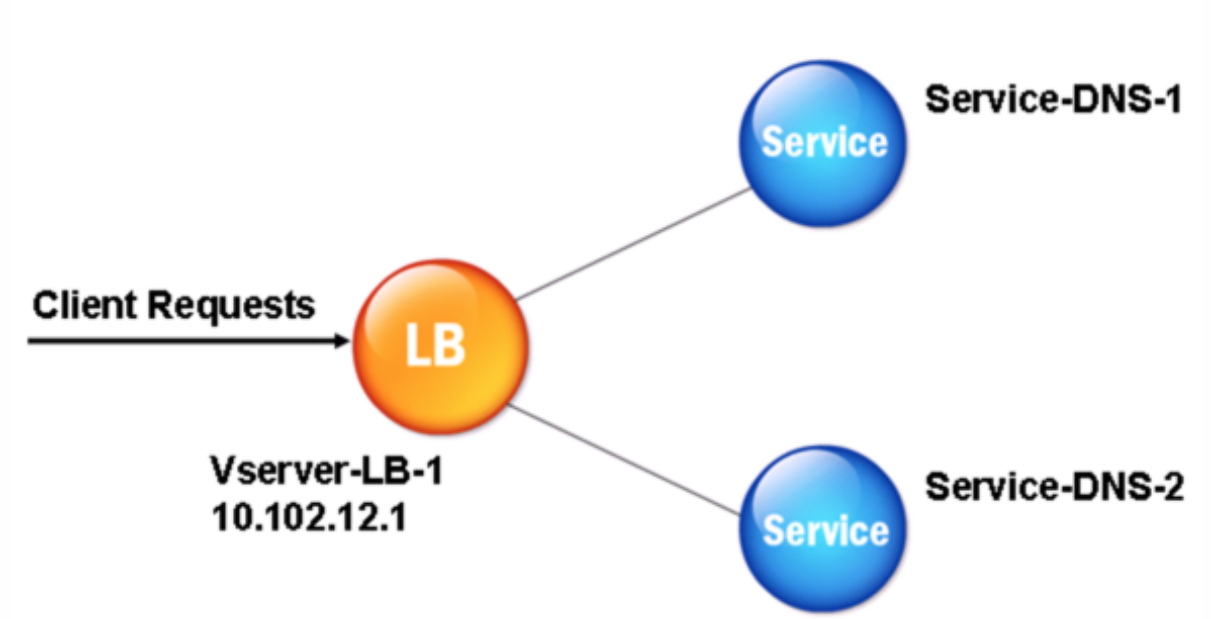
次の表は、NetScaler で構成する必要があるエンティティのサンプル名と値を示しています。

表 1. DNS プロキシエンティティ設定の例

エンティティタイプ	名前	IP アドレス	種類	ポート
LB 仮想サーバー	Vserver-DNS-1	10.102.29.40	DNS	53
サービス	Service-DNS-1	10.102.29.50	DNS	53
Services	Service-DNS-2	10.102.29.51	DNS	53

次の図は、DNS プロキシのエンティティと NetScaler で構成されるパラメーターの値を示しています。

図 2: DNS プロキシエンティティモデル



> 注
>
> DNS プロキシ機能を設定するには、負荷分散サービスと仮想サーバーの設定方法を知っている必要があります。

負荷分散仮想サーバーを作成する

NetScaler で DNS プロキシを構成するには、DNS タイプの負荷分散仮想サーバーを構成します。再帰クエリをサポートする DNS サーバーのセットを負荷分散するように DNS 仮想サーバーを構成するには、Recursion Available オプションを設定する必要があります。このオプションを使用すると、DNS 仮想サーバからの DNS 応答で RA ビットが ON に設定されます。

負荷分散仮想サーバーの作成手順については、負荷分散を参照してください。

DNS サービスの作成

DNS タイプの負荷分散仮想サーバーを作成したら、DNS サービスを作成する必要があります。DNS サービスを追加、変更、有効化、無効化、および削除できます。DNS サービスの作成手順については、「負荷分散」を参照してください。

負荷分散仮想サーバーを DNS サービスにバインドする

DNS プロキシ構成を完了するには、DNS サービスを負荷分散仮想サーバーにバインドする必要があります。サービスを負荷分散仮想サーバーにバインドする手順については、負荷分散を参照してください。

TCP を使用するように DNS プロキシのセットアップを構成する

一部のクライアントは DNS 通信にユーザーデータグラムプロトコル (UDP) を使用します。ただし、UDP では最大パケットサイズを 512 バイトに指定しています。ペイロードの長さが 512 バイトを超える場合、クライアントは TCP を使用する必要があります。クライアントが NetScaler アプライアンスに DNS クエリを送信すると、アプライアンスはクエリをネームサーバーの 1 つに転送します。応答が UDP パケットに対して大きすぎる場合、ネームサーバーは NetScaler への応答で切り捨てビットを設定します。切り捨てビットは、応答が UDP には大きすぎるため、クライアントは TCP 接続を介してクエリを送信する必要があることを示します。ADC アプライアンスは、切り捨てビットをそのままにして応答をクライアントに中継します。クライアントがポート 53 の DNS 負荷分散仮想サーバーの IP アドレスと TCP 接続を開始するのを待ちます。クライアントは TCP 接続を介して要求を送信します。次に、NetScaler アプライアンスは要求をネームサーバーに転送し、その応答をクライアントに中継します。

DNS に TCP プロトコルを使用するように NetScaler を構成するには、DNS_TCP タイプの負荷分散仮想サーバーとサービスを構成する必要があります。DNS_TCP タイプのモニタを構成して、サービスの状態を確認することができます。DNS_TCP 仮想サーバー、サービス、およびモニターの作成手順については、負荷分散を参照してください。

レコードをプロアクティブに更新するために、NetScaler ADC はサーバーへの TCP 接続を使用してレコードを取得します。

> 重要
>
> NetScaler が DNS に UDP を使用し、UDP のペイロード長が 512 バイトを超える場合にのみ TCP を使用するように構成するには、DNS サービスと DNS_TCP サービスの両方を構成する必要があります。DNS_TCP サービスの IP アドレスは DNS サービスの IP アドレスと同じでなければなりません。

DNS エントリの有効期間値の設定

TTL は、同じドメイン名とレコードタイプのすべての DNS レコードで同じです。レコードの 1 つの TTL 値を変更すると、新しい値は同じドメイン名とタイプのすべてのレコードに反映されます。デフォルトの TTL 値は 3600 秒です。最小値は 0 で、最大値は 604800 です。DNS エントリの TTL 値が最小値より小さいか、最大値より大きい場合は、それぞれ最小 TTL 値または最大 TTL 値として保存されます。

CLI を使用して最小 TTL と最大 TTL を指定します

NetScaler のコマンドプロンプトで、次のコマンドを入力して最小 TTL と最大 TTL を指定し、構成を確認します。

```
-  set dns parameter [-minTTL <secs>] [-maxTTL <secs>]
-  show dns parameter
<!--NeedCopy-->
```

例:

```
> set dns parameter -minTTL 1200 -maxTTL 1800
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
        Minimum TTL: 1200               Maximum TTL: 1800
            .
            .
            .
 Done
>
<!--NeedCopy-->
```

GUI を使用して最小 TTL と最大 TTL を指定します

1. [トラフィック管理] **> [DNS**] に移動します。
2. 詳細ペインの [設定] で、[DNS 設定の変更] をクリックします。
3. [DNS パラメータの設定] ダイアログボックスの [TTL] の [最小] テキストボックスと [最大] テキストボックスに、それぞれ最小有効期間と最大有効時間（秒単位）を入力し、[OK] をクリックします。

注**:** TTL の有効期限が切れると、レコードはキャッシュから削除されます。NetScaler は積極的にサーバーに接続し、DNS レコードの有効期限が切れる直前に DNS レコードを取得します。

DNS レコードをフラッシュする

キャッシュに存在するすべての DNS レコードを削除できます。たとえば、変更を加えた後にサーバーが再起動したときに DNS レコードをフラッシュしたい場合があります。

CLI を使用してすべてのプロキシレコードを削除する

NetScaler のコマンドプロンプトで、次のように入力します。

```
flush dns proxyRecords
```

GUI を使用してすべてのプロキシレコードを削除する

1. [トラフィック管理**] > [DNS] > [**レコード] に移動します。
2. 詳細ペインで、「プロキシレコードをフラッシュ」をクリックします。

DNS リソースレコードの追加

NetScaler ADC アプライアンスが DNS プロキシサーバーとして構成されているドメインに DNS レコードを追加できます。DNS レコードの追加の詳細については、「DNS リソースレコードの設定」を参照してください。

負荷分散 **DNS** 仮想サーバーを削除する

負荷分散仮想サーバーの削除については、負荷分散を参照してください。

クライアント接続での同時 **DNS** 要求の数を制限する

<clientip:port>-<vserver ip:port>タプルで識別される 1 つのクライアント接続での同時 DNS リクエストの数を制限できます。同時 DNS リクエストとは、NetScaler アプライアンスがネームサーバーに転送し、アプライアンスが応答を待っているリクエストです。クライアント接続で同時に実行される要求の数を制限することで、敵対的なクライアントが大量の DNS 要求を送信して分散型サービス拒否 (DDoS) 攻撃を試みた場合に、ネームサーバーを保護できます。クライアント接続の制限に達すると、未処理のリクエスト数が制限を下回るまで、その接続に対する後続の DNS リクエストはドロップされます。この制限は、NetScaler アプライアンスがキャッシュから処理するリクエストには適用されません。

このパラメータのデフォルト値は 255 です。ほとんどのシナリオでは、このデフォルト値で十分です。ネームサーバーが通常の動作条件で多数の同時 DNS リクエストを処理する場合は、大きな値またはゼロ (0) の値を指定できます。値が 0 の場合、この機能は無効になり、1 つのクライアント接続で許可される DNS 要求の数に制限がないことを示します。このパラメーターはグローバルパラメーターであり、NetScaler アプライアンスで構成されているすべての DNS 仮想サーバーに適用されます。
このパラメータのデフォルト値は 255 です。ほとんどのシナリオでは、このデフォルト値で十分です。ネームサーバーが通常の動作条件で多数の同時 DNS リクエストを処理する場合は、大きな値またはゼロ (0) の値を指定できます。値が 0 の場合、この機能は無効になり、1 つのクライアント接続で許可される DNS 要求の数に制限がないことを示します。このパラメーターはグローバルパラメーターであり、NetScaler アプライアンスで構成されているすべての DNS 仮想サーバーに適用されます。

このパラメータのデフォルト値は 255 です。ほとんどのシナリオでは、このデフォルト値で十分です。ネームサーバーが通常の動作条件で多数の同時 DNS リクエストを処理する場合は、大きな値またはゼロ (0) の値を指定できます。値が 0 の場合、この機能は無効になり、1 つのクライアント接続で許可される DNS 要求の数に制限がないことを示します。このパラメーターはグローバルパラメーターであり、NetScaler アプライアンスで構成されているすべての DNS 仮想サーバーに適用されます。

CLI を使用して、**1** つのクライアント接続で許可される同時 **DNS** 要求の最大数を指定します

コマンドプロンプトで次のコマンドを入力して、1 つのクライアント接続で許可される同時 DNS 要求の最大数を指定し、構成を確認します。

```
1 -  set dns parameter -maxPipeline <positive_integer>
2 -  show dns parameter
3 <!--NeedCopy-->
```

例**:**

```
1  > set dns parameter -maxPipeline 1000
2   Done
3  > show dns parameter
4          DNS parameters:
5          DNS retries: 5
6          .
7          .
8          .
9          Max DNS Pipeline Requests: 1000
10  Done
11 <!--NeedCopy-->
```

GUI を使用して、**1** つのクライアント接続で許可される同時 **DNS** 要求の最大数を指定します

1. [トラフィック管理**] > [DNS**] に移動します。
2. 詳細ウィンドウで、[DNS 設定の変更] をクリックします。
3.「DNS パラメータの設定」ダイアログで、「DNS パイプラインリクエストの最大数」の値を指定します。
4.「OK」をクリックします。

NetScaler ADC をエンドリゾルバーとして構成する

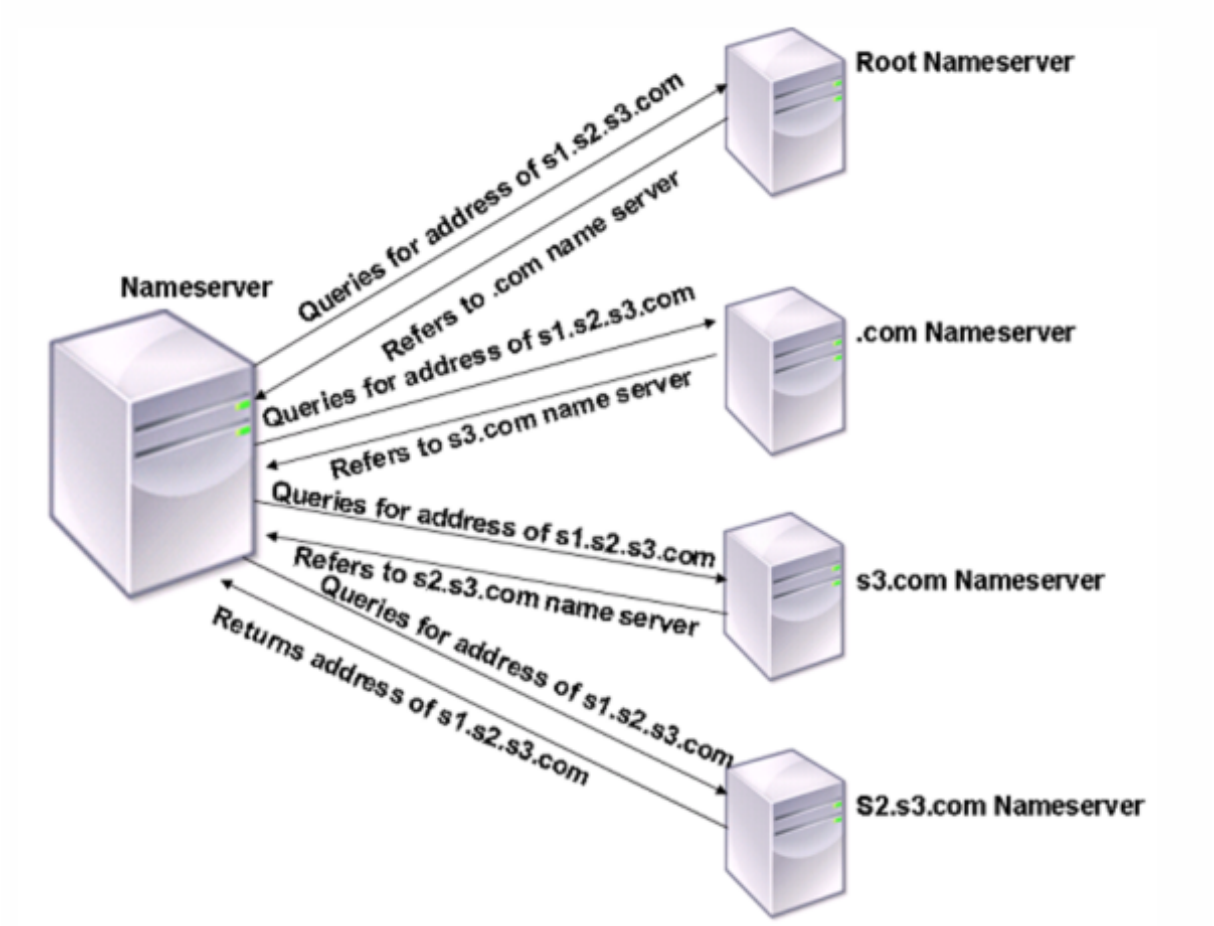
August 15, 2023

リゾルバは、ドメイン/ホスト名をリソースレコードに変換するアプリケーションプログラムによって呼び出されるプロシージャです。リゾルバーは LDNS と通信し、LDNS はドメイン名を検索して IP アドレスを取得します。NetScaler、DNS クエリをエンドツーエンドで解決できます。

再帰的解決では、NetScaler ADC アプライアンスは異なるネームサーバーに再帰的にクエリを実行してドメインの IP アドレスにアクセスします。NetScaler ADC は DNS リクエストを受信すると、キャッシュに DNS レコードがないかチェックします。レコードがキャッシュに存在しない場合は、ns.conf ファイルに設定されているルートサーバに問い合わせます。ルートネームサーバーは、セカンドレベルドメインに関する詳細情報を含む DNS サーバーのアドレスを報告します。この処理は、必要なレコードが見つかるまで繰り返されます。

NetScaler ADC アプライアンスを初めて起動すると、13 のルートネームサーバーが ns.conf ファイルに追加されます。13 台のルートサーバの NS レコードと Address レコードも追加されます。ns.conf ファイルは変更できますが、NetScaler ADC では 13 個のレコードすべてを削除することはできません。アプライアンスが名前解決を実行するには、少なくとも 1 つのネームサーバエントリが必要です。次の図は、名前解決のプロセスを示しています。

図 1：再帰的解決



図に示されているプロセスでは、ネームサーバーは s1.s2.s3.com のアドレスに関するクエリを受信すると、まず s1.s2.s3.com のルートネームサーバーをチェックします。ルートネームサーバーは、.com ネームサーバーのアドレスを報告します。s1.s2.s3.com のアドレスがネームサーバーで見つかると、適切な IP アドレスで応答します。それ以外の場合は、他のネームサーバに s3.com をクエリし、次に s2.s3.com に対してクエリを実行して s1.s2.s3.com のアドレスを取得します。この方法では、解決は常にルートネームサーバーから始まり、ドメインの権限のあるネームサーバーで終了します。

> 注:
>
> 再帰的な解決機能を使用するには、キャッシュを有効にする必要があります。

再帰的解決を有効にする

NetScaler ADC アプライアンスがエンドリゾルバーとして機能するように構成するには、アプライアンスで再帰的解決を有効にする必要があります。また、この機能を動作させるには、ローカルオプションを指定した DNS ネームサーバーを追加する必要があります。

CLI を使用して再帰解決を有効にする

コマンドプロンプトで次のコマンドを入力して再帰解決を有効にし、構成を確認します。

```
-  set dns parameter -recursion ENABLED
-  show dns parameter
<!--NeedCopy-->
```

例**:**

```
> set dns parameter -recursion ENABLED
 Done
> show dns parameter
        DNS parameters:
            .
            .
            .
        Recursive Resolution : ENABLED
            .
            .
            .
 Done
<!--NeedCopy-->
```

GUI を使用して再帰解決を有効にする

1. [トラフィック管理**] > [DNS**] に移動します。
2. 詳細ペインの [設定] で、[DNS 設定の変更] をクリックします。
3.「DNS パラメータの設定」ダイアログ・ボックスで、「再帰を有効にする」チェックボックスを選択し、「OK」をクリックします。

CLI を使用してネームサーバーを追加する（**NetScaler ADC** アプライアンスがリゾルバーとして機能する場合）

コマンドプロンプトで入力します。


```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>)
<!--NeedCopy-->
```

例:

```
add dns nameServer 10.102.9.19 -local
show dns nameServer
1)  10.102.9.19 LOCAL -  State: UP  Protocol: UDP
Done
<!--NeedCopy-->
```

ローカル -IP アドレスを、NetScaler ADC アプライアンス上のローカル再帰 DNS サーバーに属するものとしてマークします。アプライアンスは、ローカルとしてマークされた IP アドレスで受信したクエリを再帰的に解決します。再帰的解決を機能させるには、グローバル DNS パラメータ、recursionも設定する必要があります。
ローカルとしてマークされているネームサーバーがない場合、アプライアンスはスタブリゾルバーとして機能し、ネームサーバーの負荷を分散します。

GUI を使用してネームサーバーを追加する

[トラフィック管理] > [**DNS**] > [ネームサーバー] に移動し、ネームサーバーを作成します。

DNS ルート参照を有効にする

DNS ルートリフェラルはデフォルトでは無効になっています。有効にすると、ADC アプライアンスはルート参照レコードで応答します。

NetScaler ADC アプライアンスで構成/キャッシュされたドメインとは無関係のドメイン名をクライアントが照会した場合は、ルート紹介を送信します。設定が無効になっている場合、アプライアンスはルート紹介の代わりに空白の応答を送信します。アプライアンスが権限を持つドメインに適用されます。関連のないドメインに大量のクエリを送信しているクライアントからアプライアンスが攻撃を受けている場合は、このパラメータを無効にします。

CLI を使用してルート参照を有効にする

コマンドプロンプトで次のコマンドを入力して再帰解決を有効にし、構成を確認します。

```
-  set dns parameter -dnsrootReferral ENABLED
-  show dns parameter
<!--NeedCopy-->
```

例:

```
> set dns parameter -recursion ENABLED
 Done
> show dns parameter
```



```
4          DNS parameters:
5               .
6               .
7               .
8          DNS Root Referral : ENABLED
9               .
10              .
11              .
12  Done
13 <!--NeedCopy-->
```

GUI を使用してルート参照を有効にする

1. [トラフィック管理**] > [DNS**] に移動します。
2. 詳細ペインの [設定] で、[**DNS** 設定の変更] をクリックします。
3.「**DNS** パラメータの設定」ダイアログ・ボックスで、「ルート参照を有効にする」チェック・ボックスを選択し、「**OK**」をクリックします。

リトライ回数の設定

クエリの送信先サーバーから応答がない場合に、事前に設定した回数だけ再試行（DNS 再試行）を行うように ADC アプライアンスを設定します。デフォルトでは、DNS リトライ回数は 5 回に設定されています。

CLI を使用して **DNS** リトライ回数を設定します

コマンドプロンプトで、次のコマンドを入力して再試行回数を設定し、構成を確認します。

```
1 -  set dns parameter -retries <positive_integer>
2 -  show dns parameter
3 <!--NeedCopy-->
```

例**:**

```
1  > set DNS parameter -retries 3
2   Done
3  > show dns parameter
4          DNS parameters:
5          DNS retries: 3
6               .
7               .
8               .
9   Done
10 <!--NeedCopy-->
```


GUI を使用して再試行回数を設定する

1. [トラフィック管理] **> [DNS**] に移動します。
2. 詳細ペインの [設定] で、[DNS 設定の変更] をクリックします。
3.「DNS パラメータの設定」ダイアログ・ボックスの「DNS 再試行」テキスト・ボックスに、DNS リゾルバー要求の再試行回数を入力し、「OK」をクリックします。

NetScaler ADC アプライアンスをフォワーダーとして構成する

August 15, 2023

フォワーダーは、フォワーダーサーバーのネットワーク外にある DNS サーバーに DNS クエリを転送するサーバーです。ローカルで解決できないクエリは、他の DNS サーバーに転送されます。フォワーダーは、DNS クエリを解決するときに外部 DNS 情報をキャッシュに蓄積します。NetScaler ADC アプライアンスをフォワーダーとして構成するには、外部ネームサーバーを追加する必要があります。

NetScaler ADC アプライアンスでは、ローカルで解決できない名前解決クエリを転送できる外部ネームサーバーを追加できます。NetScaler ADC アプライアンスをフォワーダーとして構成するには、名前解決クエリの転送先となるネームサーバーを追加する必要があります。ルックアップの優先順位を指定して、NetScaler ADC アプライアンスが名前解決に使用する必要のあるネームサービスを指定できます。

NetScaler ADC アプライアンスをフォワーダーとして構成する方法については、CLI を使用したネームサーバーの追加（NetScaler ADC アプライアンスがフォワーダーとして機能する場合）を参照してください。

> 注:
>
> フォワーダーモードの NetScaler ADC アプライアンスは、TCP、UDP、および UDP-TCP ネームサーバーをサポートしています。
>
> - TCP ネームサーバーを構成した場合、NetScaler ADC アプライアンスは TCP 経由で DNS 要求を送信します。
> - UDP ネームサーバーを構成した場合、NetScaler ADC アプライアンスは UDP 経由で DNS 要求を送信します。
> - UDP-TCP ネームサーバーを構成した場合、NetScaler ADC アプライアンスは UDP 経由で DNS 要求を送信します。ただし、切り捨てられたビットが DNS 応答に設定されている場合、アプライアンスはそのような DNS 要求を TCP 経由で送信します。

ネームサーバーを追加する

ネームサーバーを作成するには、IP アドレスを指定するか、既存の仮想サーバーをネームサーバーとして構成します。

- **IP** アドレスベースのネームサーバー -ドメイン名解決のために連絡する外部ネームサーバー。アプライアンス上に複数の IP アドレスベースのネームサーバーが設定されていて、いずれにもローカルパラメータが設定されていない場合、受信 DNS クエリはすべてのネームサーバー間でラウンドロビン方式で負荷分散されます。
- 仮想サーバーベースのネームサーバー -NetScaler で構成された DNS 仮想サーバー。外部 DNS ネームサーバーの負荷分散方法をより詳細に制御するには (たとえば、ラウンドロビン以外の負荷分散方法が必要な場合など)、次の操作を行います。
- アプライアンスに DNS 仮想サーバーを設定します。
- 外部ネームサーバーをサービスとしてバインドします。
- このコマンドで仮想サーバーの名前を指定します。

設定を確認するには、`show dns nameServer`コマンドを使用できます。

ネームサーバーを削除するには、NetScaler CLI で`rm dns nameServer`コマンドを入力し、その後にネームサーバーの IP アドレスを入力します。

DNS ネームサーバーの詳細を表示するには、NetScaler CLI で`show dns nameServer`コマンドを入力し、続いてネームサーバーの IP アドレスを入力します。

CLI を使用してネームサーバーを追加する（**NetScaler ADC** アプライアンスがフォワーダーとして機能する場合）

コマンドプロンプトで次を入力します。

```
add dns nameServer ((<IP>) | <dnsVserverName>)
<!--NeedCopy-->
```

または

```
add dns nameServer ((<IP> | <dnsVserverName>) [-type <type>]
<!--NeedCopy-->
```

例:

```
add dns nameServer dnsVirtualNS

add dns nameServer 192.0.2.11 -type TCP

add dns nameServer 192.0.2.12 -type UDP_TCP


add dns nameServer 192.0.2.10
show dns nameServer 192.0.2.10

1) 192.0.2.10  -  State: UP  Protocol: UDP
Done
<!--NeedCopy-->
```


> 注:
>
> ネームサーバータイプが指定されていない場合、デフォルトで UDP ネームサーバーが作成されます。TCP または UDP_TCP タイプのネームサーバーを作成するには、タイプを指定する必要があります。
>
> タイプを UDP_TCP に指定すると、指定した IP アドレスに対して 2 つのネームサーバー (1 つの UDP ネームサーバーと 1 つの TCP ネームサーバー) が作成されます。

CLI を使用してネームサーバーを追加する（**NetScaler ADC** アプライアンスがリゾルバーとして機能する場合）

再帰リゾルバーの`local`パラメータを指定します。`set dns parameter`コマンドを使用して再帰を有効にします。

コマンドプロンプトで入力します。

```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>)
show dns nameServer
set dns parameter -recursion ENABLED
show dns parameter
<!--NeedCopy-->
```

例**:**

```
add dns nameServer 10.102.9.19 -local
show dns nameServer
1)  10.102.9.19 LOCAL -  State: UP  Protocol: UDP
Done
set dns parameter -recursion ENABLED
Done
show dns parameter
        DNS parameters:
            .
            .
            .
        Recursive Resolution : ENABLED
            .
            .
            .
Done
<!--NeedCopy-->
```

ローカル -IP アドレスを、NetScaler ADC アプライアンス上のローカル再帰 DNS サーバーに属するものとしてマークします。アプライアンスは、ローカルとしてマークされた IP アドレスで受信したクエリを再帰的に解決します。
再帰的解決を機能させるには、グローバル DNS パラメータ、`recursion`も設定する必要があります。
ローカルとしてマークされているネームサーバーがない場合、アプライアンスはスタブリゾルバーとして機能し、ネームサーバーの負荷を分散します。

GUI を使用してネームサーバーを追加する

[トラフィック管理] > [**DNS**] > [ネームサーバー] に移動し、ネームサーバーを作成します。

DNS ルックアップの優先度を設定する

検索優先順位は DNS または WINS に設定できます。このオプションは SSL VPN 動作モードで使用されます。

CLI を使用してルックアップ優先度を **DNS** に設定します

コマンドプロンプトで次のコマンドを入力して、検索優先度を DNS に設定し、構成を確認します。

```
-  set dns parameter -nameLookupPriority (DNS | WINS)
-  show dns parameter
<!--NeedCopy-->
```

例**:**

```
> set dns parameter -nameLookupPriority DNS
 Done
> show dns parameter
                .
                .
                .
          Name lookup priority : DNS
                .
                .
                .
 Done
<!--NeedCopy-->
```

GUI を使用してルックアップ優先度を **DNS** に設定する

1. [トラフィック管理**] > [DNS**] に移動します。
2. 詳細ペインの [設定] で、[**DNS** 設定の変更] をクリックします。
3. 「**DNS** パラメータの構成」ダイアログ・ボックスの「名前検索の優先度」で、「DNS」または「WINS」を選択し、「**OK**」をクリックします。

> 注
>
> 構成した DNS 仮想サーバーがダウンしていて、-nameLookupPriorityを DNS に設定した場合、NetScaler ADC は WINS 検索を試みません。したがって、DNS 仮想サーバーが構成されていない、または無効になっている場合は、-nameLookupPriorityを WINS に設定します。

ネームサーバーを無効および有効にする

以下の手順では、既存のネームサーバーを有効または無効にする手順について説明します。

CLI を使用してネームサーバーを有効または無効にする

コマンドプロンプトで次のコマンドを入力してネームサーバーを有効または無効にし、構成を確認します。

```
-  (enable | disable) dns nameServer <IPAddress>
-  show dns nameServer <IPAddress>
<!--NeedCopy-->
```

例**:**

```
> disable dns nameServer 10.102.9.19
 Done
> show dns nameServer 10.102.9.19
1)       10.102.9.19: LOCAL  -  State: OUT OF SERVICE
 Done
<!--NeedCopy-->
```

GUI を使用してネームサーバーを有効または無効にする

1. **Traffic Management > DNS > Name Servers** に移動します。
2. 詳細ペインで、有効または無効にするネームサーバーを選択します。
3. **** 有効または無効をクリックします。ネームサーバーが有効になっている場合は、**** 無効化オプションを使用できます。ネームサーバーが無効になっている場合は、有効化オプションを使用できます。

NetScaler ADC を非検証のセキュリティ対応スタブリゾルバーとして構成する

August 15, 2023

NetScaler 12.1 ビルド 49.xx 以降、NetScaler はセキュリティを検証しないスタブリゾルバーとして機能するようになりました。このサポートを有効にするには、DNS ヘッダーに AD ビットを設定し、OPT ヘッダーに DO ビットを設定解除します。AD ビットが設定されていて DO ビットが設定されていない場合、アップストリームの再帰リゾルバーは DNSSEC 応答を検証します。検証が成功すると、再帰リゾルバーは DNSSEC RR なしに応答します。DNSSEC 検証が失敗した場合、再帰リゾルバーは SERVFAIL 応答を返します。

> 重要:
>
> AD ビットは ADC フォワーダにデフォルトで設定されます。AD ビットは、DBS によって開始されたクエリには設定されません。

サイズの大きな応答を処理するため **DNS** でジャンボフレームをサポート

August 15, 2023

NetScaler 12.1 ビルド 49.xx 以降、DNS は 1,280 バイトを超える UDP 応答を処理するためのジャンボフレームをサポートしています。以前は、NetScaler ADC アプライアンスは最大 1,280 バイトの UDP パケットサイズしかサポートしていませんでした。

最大 UDP パケットサイズパラメータ値を設定することにより、アプライアンスがプロキシ、ADNS、およびフォワーダモードで処理できる最大 UDP パケットサイズを設定できます。たとえば、最大 UDP パケットサイズパラメータ値が 4096 に設定されている場合、アプライアンスはサイズ 4,096 バイトの DNS 応答を処理できます。

> 重要
>
> - プロキシモードでは、クライアントリクエストの OPT ペイロードサイズと最大 UDP パケットサイズ値の間の最小サイズが、バックエンドに DNS クエリを送信する際に考慮されます。たとえば、クライアントリクエストの OPT ペイロードサイズが 3000 で、最大 UDP パケットサイズの値が 4096 の場合、3,000 バイトの DNS クエリがバックエンドに送信されます。
>
>   また、アプライアンスはバックエンドから大きなサイズの応答を受信し、大きなサイズの応答を処理できます。
> - フォワーダーモードでは、アプライアンスは OPT ペイロードサイズを UDP パケットサイズパラメータ値と同じに設定します。
> - DNS レコードがアプライアンスのローカルにある場合、アプライアンスは最大 UDP パケットサイズパラメータ値と同じ大きさの応答サイズを作成できます。この設定は、ADNS、プロキシ、および再帰リゾルバーに適用されます。

CLI を使用して最大 **UDP** パケットサイズを設定するには

コマンドプロンプトで入力します。

```
set dns parameter [-maxUDPPacketSize <positive_integer>]
<!--NeedCopy-->
```

例:

```
set dns parameter -maxUDPPacketSize 10000
<!--NeedCopy-->
```

> 注:
>
> 最大 UDP パケットサイズパラメータに設定できる最小値と最大値は、それぞれ 512 と 16384 です。デフォルト値は 1280 です。

GUI を使用して最大 **UDP** パケットサイズを設定するには

1. **Traffic Management** > **DNS** に移動します。
2. 詳細ウィンドウで、[**DNS** 設定の変更] をクリックします。
3. [最大 UDP パケットサイズ] に、最大 UDP パケットサイズを指定します。
4. [**OK**] をクリックします。

DNS ログを構成する

August 15, 2023

NetScaler ADC アプライアンスは、処理する DNS 要求と応答を記録するように構成できます。アプライアンスは、DNS 要求と応答を SYSLOG 形式で記録します。DNS 要求または DNS 応答、またはその両方をログに記録し、syslog メッセージをリモートログサーバに送信できます。ログメッセージは、次の目的で使用できます。

- クライアントに対する DNS 応答を監査します。
- DNS クライアントを監査する
- DNS 攻撃の検出と防止
- トラブルシューティング

NetScaler ADC アプライアンスは、構成に基づいて、DNS 要求または応答の次のセクションを記録できます。

- ヘッダーセクション
- 質問セクション
- 回答セクション
- 権限セクション
- 追加セクション

DNS プロファイル

DNS プロファイルを使用して、DNS エンドポイントを DNS トラフィックに適用するさまざまな DNS パラメータを設定できます。プロファイルでは、ロギング、キャッシュ、およびネガティブキャッシュを有効にできます。

重要: NetScaler 11.0 リリースから、グローバル DNS パラメーターを使用した DNS キャッシュの有効化は廃止されました。DNS プロファイルを使用して DNS キャッシュを有効または無効にできます。DNS プロファイルで DNS キャッシュを有効にし、DNS プロファイルを個々の仮想サーバーに設定することで、個々の仮想サーバーの DNS キャッシュを有効にできるようになりました。

DNS プロファイルは、次のタイプの DNS ロギングをサポートします。

- DNS クエリロギング

- DNS 応答セクションのロギング
- DNS 拡張ロギング
- DNS エラーロギング

DNS クエリロギング

NetScaler ADC アプライアンスは、アプライアンスの DNS エンドポイントによって受信された DNS クエリのみをログに記録するように構成できます。

注: クエリの処理中にエラーが発生した場合、DNS プロファイルでこのオプションを設定すると、エラーがログに記録されます。

次に、クエリログメッセージの例を示します。

```
DNS DNS_QUERY 143 0 :  U:10.102.27.70#61297:10.102.27.73#53/22142/Q/
(RD)/NO/1/0/0/0#test.com./1#
<!--NeedCopy-->
```

DNS 応答セクションのロギング

NetScaler ADC アプライアンスは、アプライアンスがクライアントに送信する **DNS** 応答のすべての回答セクションを記録するように構成できます。DNS 応答セクションのログは、NetScaler ADC が DNS リゾルバーとして構成されている場合、または GLSB のユースケースで役立ちます。

次に、DNS 応答セクションログの例を示します。

```
DNS DNS_RESPONSE 6678 0 :  U:100.100.100.210#32776:100.100.100.10#
53/61373/Q/(RD,AA,RA,R)/NO/1/1/2/4#n1.citrix.com1./
28#ANS#AAAA/120/1111:2345:6789:ffab:abcd:effa:1234:3212##
<!--NeedCopy-->
```

DNS 拡張ロギング

DNS 応答の権限セクションと追加セクションをログに記録するように **Citrix** ADC アプライアンスを構成するには、応答セクションログを使用して拡張ログを有効にします。

注: クエリまたは応答の処理中にエラーが発生した場合、DNS プロファイルでこのオプションが設定されていれば、エラーがログに記録されます。

次に、キャッシュルックアップが完了し、応答がパケットに埋め込まれたときにログに記録されるメッセージの例を示します。

```
DNS DNS_RESPONSE 2252 0 :  T:100.100.100.118#21411:100.100.100.10
#53/48537/Q/(RD,AA,CD,RA,R)/NO/1/1/2/6#a1.citrix.com1./1#ANS#A/
120/1.1.1.1##AUTH#citrix.com1/NS/120/n2.citrix.com1#n1.citrix.com1##ADD
   #n1.citrix.com1
```



```
/A/120/1.1.1.1#1.1.1.2##n1.citrix.com1/AAAA/120/
1111:2345:6789:ffab:abcd:effa:1234:3212##n2.citrix.com1/A/120/2.1.1.2
##n2.citrix.com1/AAAA/120/2222:faff:3212:8976:123:1241:64:ff9b##OPT
   /0/1280/DO##
<!--NeedCopy-->
```

DNS エラーロギング

NetScaler ADC アプライアンスは、DNS クエリまたは応答を処理するときに発生するエラーまたは障害をログに記録するように構成できます。これらのエラーの場合、アプライアンスは DNS ヘッダー、質問セクション、および OPT レコードを記録します。

次に、DNS 要求または応答の処理中にエラーが発生したときに記録されるメッセージの例を示します。

```
DNS DNS_ERROR 149 0 :  U:10.102.27.70#27832:10.102.27.73#53/61153/Q/
(RD)/NO/1/0/0/0#test.com./1140#Packet Dropped
<!--NeedCopy-->
```

ポリシーベースのロギング

DNS ポリシーの LogAction、書き換え、またはレスポンダーポリシーを構成することにより、DNS 式に基づいてカスタムログを構成できます。特定の DNS ポリシーが true と評価された場合にのみロギングが行われるように指定できます。詳細については、「DNS のポリシーベースのログを構成する」を参照してください。

NetScaler Syslog ログメッセージの形式を理解する

NetScaler ADC アプライアンスは、次の Syslog 形式で DNS 要求と応答を記録します。

```
<transport> :<client IP>#<client ephemeral port>:<DNS endpoint IP>#<
   port>
 : <query id> /opcode/header flags/rcode/question section count/answer
    section count
/ auth section count / additional section count #<queried domain name>
/<queried type>#...
<!--NeedCopy-->
```

- **<transport>:**
  - T = TCP
  - U = UDP
- **\ #\: DNS** <client IP> < client ephemeral port> クライアントの IP アドレスとポート番号
- **\ #\: Citrix** <DNS endpoint IP> <port> ADC DNS エンドポイントの IP アドレスとポート番号

- **\:**
  <query id> クエリ ID
- **\:** <opcode> オペレーションコード。サポートされる値:
  - Q: クエリ
  - I: 逆クエリ
  - S: ステータス
  - X0: 割り当てられていない
  - N: 通知する
  - U: 更新
  - X1-10: 割り当てられていない値
- **\:** <header flags> フラグ。サポートされる値:
  - RD: 再帰が望ましい
  - TC: 切り捨てられた
  - AA: 権威ある応答
  - CD: チェック無効
  - AD: 認証済みデータ
  - Z: 未割り当て
  - RA: 再帰が利用可能
  - R: 応答
- **\:** <rcode> レスポンスコード。サポートされる値:
  - いいえ: エラーなし
  - F フォーマットエラー
  - S: サーバー障害
  - NX: 存在しないドメイン
  - NI: 実装されていません
  - R: クエリが拒否されました
  - YX: 名前が存在してはならないときに存在する
  - YXR: RR セットが存在してはならないときに存在する
  - NXR: 存在しなければならない RR セットは存在しない
  - NAS: サーバーはゾーンに対して権限がありません
  - NA: 承認されていません
  - NZ: ゾーンに名前が含まれていません
  - X1-5: 割り当てられていない
- **/question** セクション数**/**回答セクション数**/**認証セクション数**/**追加セクション数**:** DNS リクエストの質問セクション、権限セクション数、および追加セクション数
- **\/\:** <queried domain name> <queried type> DNS 要求でクエリされたドメインとクエリされたタイプ

- **<ttl>#ANS #\/<record type>\/..<ttl>#AUTH #\/<domain name>\/<record type>..<ttl>#ADD #\/<domain name>\/<record type>…:**

  DNS 応答では、次のようになります。

  応答セクションは、DNS プロファイルで応答セクションのログが有効になっている場合に記録されます。DNS プロファイルで拡張ロギングが有効になっている場合、[権限] セクションと [追加] セクションがログに記録されます。ログ形式は、レコードのタイプによって異なります。詳細については、「レコードログ形式について」を参照してください。

  - ANS: 回答セクション
  - AUTH: 権限
  - 追加: 追加セクション

- **OPT/\<EDNS** バージョン**\>/UDP** 最大ペイロードサイズ**/DO:** DNS ログの OPT レコード形式
- **OPT/\<EDNS** バージョン**\>/\<UDP** ペイロードサイズ**\>/\<DNSSEC OK** ビットが設定されているかどうかに基づいて”**DO**”または空にする**\>/\<RDLEN** の値**\>/ECS/\<Q/R\>/\<** オプションの長さ**\>/\<** ファミリー**\>/\<** ソースプレフィクスの長さ**\>/\<** スコーププレフィクスの長さ**\>/\<ECS** アドレス**\>:**

  DNS クエリまたは応答に EDNS クライアントサブネット (ECS) オプションが含まれている場合、それも DNS ログファイルに OPT レコード形式で記録されます。

  IPv4 または IPv6 アドレスのいずれかを含む ECS オプションを含む DNS クエリが送信されると、ECS オプションは次のいずれかのオプションでログに記録されます。

  - 「ECS/Q」は、ログ内の値がクエリからのものであることを示す
  - 「ECS/R」は、ログ内の値が応答からのものであることを示します。

  スコーププレフィクス-長さの値も適切に設定されます。DNS クエリではゼロに設定され、応答の場合は計算値に設定されます。

  次の表では、さまざまなシナリオでログに記録される詳細について説明します。

シナリオ	DNS クエリの ECS オプションセット	DNS レスポンスの ECS オプションセット	記録された詳細
クエリロギングと拡張ロギングの両方が有効	はい	はい	ECS オプションは文字列「ECS/R/」でログに記録され、スコーププレフィックス-長さは計算された値に設定されます。

シナリオ	DNS クエリの ECS オプションセット	DNS レスポンスの ECS オプションセット	記録された詳細
クエリロギングと拡張ロギングの両方が有効	はい	いいえ	ECS オプションは文字列「ECS/Q」でログに記録され、スコーププレフィクス-長さはゼロに設定されます。
クエリロギングは有効ですが、拡張ロギングは有効になりません	はい	はい	ECS オプションは文字列「ECS/Q/」でログに記録され、スコーププレフィクス-長さはゼロに設定されます。
クエリロギングと拡張ロギングが有効になっていない	はい	はい	ECS オプションはログに記録されません。
クエリロギングは有効ですが、拡張ロギングは有効になりません	はい	いいえ	ECS オプションは文字列「ECS/Q/」でログに記録され、スコーププレフィクス-長さはゼロに設定されます。
クエリロギングは有効ではありませんが、拡張ロギングは有効になっています	はい	はい	ECS オプションは文字列「ECS/R/」でログに記録され、スコーププレフィックス-長さは計算された値に設定されます。
クエリロギングは有効ではありませんが、拡張ロギングは有効になっています	はい	いいえ	ECS オプションはログに記録されません。

レコードロギング形式を理解する

次に、Syslog メッセージのレコードロギング形式の例を示します。

```
<domainname>/<record type>/ <record ttl> / <resource record data>#<
   resource record data>#......##
<!--NeedCopy-->
```

各項目の意味は次のとおりです:

レコードタイプ	サンプルフォーマット	リソースレコードデータ/フォーマット
住所 (A) レコード	A/5/1.1.1.1#1.1.1.2#1.1.1.3##	IPv4 アドレス
AAAA レコード	AAAA/5/1::1#1::2#1::3##	IPv6 アドレス
SOA レコード	SOA/3600/ns1.dnslogging.test./root.dnslogging.test./1200/3600/3/3600/5##	1. 優先順位2. 重み3. その他の詳細。リソースレコードの形式：//////##
NS レコード	NS/5/ns1.dnslogging.test	ネームサーバーのホスト名。
MX レコード	#MX/5/10/host1.dnslogging.test.#10/host2.dnslogging.test.##	1. 優先順位2. メール交換サーバーのホスト名が続く
CNAME レコードロギング	CNAME/5/host1.dnslogging.test.##	標準名
SRV レコード	SRV/5/1/2/3/host1.dnslogging.test.#4/5/6/host2.dnslogging.test.##	1. 優先順位2. 重み3. ポート4. ホスト名
TXT レコード	TXT/5/dns+logging##	データはすべてのテキストで構成されます。
NAPTR レコード	NAPTR/5/10/11////dnslogging#20/21/SIP/sip.dnslogging.test##	1. 順序2. 優先順位3. フラグ4. サービス5. 正規表現6. 置換
DNSKEY レコード	DNSKEY/5/1/3/5/AwEAAanP0K+i5bfv5SU47817601EjDfjaPqI2/Cx6JZgiDBZhSONP	1. フラグ2. プロトコル3. アルゴリズム4. 公開キー
PTR レコード	PTR/3600/test.com.#test4.com.##	ドメイン名

DNS ロギングの制限

DNS ロギングには次の制限があります。

- 応答ログが有効になっている場合、次のレコードタイプのみがログに記録されます。
  - 住所 (A) レコード
  - AAAA レコード
  - SOA レコード
  - NS レコード
  - MX レコード
  - CNAME レコード
  - SRV レコード
  - TXT レコード
  - NAPTR レコード
  - DNSKEY レコード
  - PTR レコード

  他のすべてのレコードタイプでは、L3/L4 パラメータ、DNS ヘッダー、および質問セクションのみがログに記録されます。

- 応答ログが有効になっていても、RRSIG レコードはログに記録されません。
- DNS64 はサポートされていません。
- DNS プロアクティブ更新要求または応答は、デフォルトプロファイルの設定に従ってログに記録されます。
- 仮想サーバでは、セッションレスオプションと応答のロギングが有効になっている場合、応答の代わりに L3/L4 パラメータ、DNS ヘッダー、および DNS 質問セクションがログに記録されます。
- syslog メッセージの最大サイズは 1024 バイトです。
- アクションタイプが応答の書き換えの DNS ポリシーの DNS プロファイルを設定した場合、NetScaler ADC アプライアンスはクエリまたは操作された応答をログに記録しません。必要な情報をログに記録するには、DNS ポリシーで監査メッセージアクションを使用する必要があります。
- DNS モニタリングトラフィックが原因の DNS トランザクションはログに記録されません。

DNS ロギングの設定

DNS ロギングの設定の概要を以下に示します。

1. Syslog アクションを作成し、そのアクションで DNS を有効にします。
2. Syslog ポリシーを作成し、ポリシーで Syslog アクションを指定します。
3. Syslog ポリシーをグローバルにバインドして、すべての NetScaler ADC システムイベントのログ記録を有効にします。または、Syslog ポリシーを特定の負荷分散仮想サーバーにバインドします。
4. DNS プロファイルを作成し、有効にする次のタイプのロギングのいずれかを定義します。
    - DNS クエリロギング
    - DNS 応答セクションのロギング
    - DNS 拡張ロギング
    - DNS エラーロギング
5. 要件に基づいて、次のいずれかを設定します。
    - DNS サービスと DNS 用の仮想サーバー
    - ADNS サービス
    - フォワーダーとしての NetScaler ADC
    - リゾルバーとしての NetScaler ADC
6. 作成した DNS プロファイルを DNS エンティティの 1 つに設定します。

CLI を使用して、**DNS** プロキシとして構成された **NetScaler ADC DNS** ログを構成する

1. syslog アクションを追加し、アクションで DNS を有効にします。コマンドプロンプトで入力します。


```
add audit syslogAction <name> (<serverIP> | -lbVserverName <string>) [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat <dateFormat>] [-logFacility <logFacility>] [-tcp ( NONE | ALL )] [-acl ( ENABLED | DISABLED )] [-timeZone ( GMT_TIME | LOCAL_TIME )] [-userDefinedAuditlog ( YES | NO )] [-appflowExport ( ENABLED |DISABLED )] [-lsn ( ENABLED | DISABLED )] [-alg ( ENABLED | DISABLED )] [-transport ( TCP | UDP )] [-tcpProfileName <string>] [-maxLogDataSizeToHold <positive_integer>] [-dns ( ENABLED | DISABLED)]
<!--NeedCopy-->
```

例:

```
add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone LOCAL_TIME -dns ENABLED
```

2. syslog ポリシーを作成し、作成した syslog アクションをポリシーで指定します。コマンドプロンプトで入力します。

```
add audit syslogPolicy <name> <rule> <action>
```

例:

```
add audit syslogPolicy syslogpol1 ns_true nssyslogact1
```

3. syslog ポリシーをグローバルにバインドします。コマンドプロンプトで入力します。

```
bind system global [<policyName> [-priority <positive_integer>]]
```

例:

```
bind system global syslogpol1
```

4. DNS プロファイルを作成し、設定する次のタイプのログのいずれかを有効にします。

   - DNS クエリロギング
   - DNS 応答セクションのロギング
   - DNS 拡張ロギング
   - DNS エラーロギング

   コマンドプロンプトで入力します。

```
add dns profile <dnsProfileName> [-dnsQueryLogging ( ENABLED | DISABLED )] [-dnsAnswerSecLogging ( ENABLED | DISABLED )] [-dnsExtendedLogging (ENABLED | DISABLED )] [-dnsErrorLogging ( ENABLED | DISABLED )] [-cacheRecords ( ENABLED | DISABLED )] [-cacheNegativeResponses ( ENABLED | DISABLED )]
```

例:

```
add dns profile dnsprofile1 -dnsQueryLogging ENABLED
```


5. DNS タイプのサービスを設定します。コマンドプロンプトで入力します。

   ```
   add service <name> <serverName> <serviceType> <port>
   ```

   例:

   ```
   add service svc1 10.102.84.140 dns 53
   ```

6. サービスタイプが DNS の負荷分散仮想サーバーを構成します。

   ```
   add lb vserver <name> <serviceType> <ip> <port>
   ```

   例:

   ```
   add lb vserver lb1 dns 100.100.100.10 53
   ```

7. サービスを仮想サーバーにバインドします。コマンドプロンプトで入力します。

   ```
   bind lb vserver <name> <serviceName>
   ```

   例:

   ```
   bind lb vserver lb1 svc1
   ```

8. 作成した DNS プロファイルを仮想サーバーに設定します。コマンドプロンプトで入力します。

   ```
   set lb vserver <name> [ - dnsProfileName <string>]
   ```

   例:

   ```
   set lb vserver lb1 -dnsProfileName dnsprofile1
   ```

DNS プロキシとして構成された **NetScaler ADC** アプライアンスの **DNS** ログ構成の例

```
1  > add audit syslogAction nssyslogact1 10.102.151.136 -logLevel
2  CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -
       timeZone
3  LOCAL_TIME -dns ENABLED
4  Done
5  > add audit syslogPolicy syslogpol1 ns_true nssyslogact1
6  Done
7  > bind system global syslogpol1
8  Done
9  > add dns profile dnsprofile1 -dnsqueryLogging ENABLED
10 Done
11 > add lb vserver lb1 dns 100.100.100.10 53  - dnsProfileName dnsprofile1
12 Done
13 > add service svc1 10.102.84.140 dns 53
14 Done
15 > bind lb vserver lb1 svc1
16 Done
17 <!--NeedCopy-->
```


ADNS として構成された **NetScaler ADC** アプライアンスの **DNS** ログ構成の例

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
    LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add lb vserver lb1 dns 100.100.100.10 53  - dnsProfileName dnsprofile1
Done
> add service svc1 10.102.84.140 dns 53
Done
> bind lb vserver lb1 svc1
Done
<!--NeedCopy-->
```

フォワーダーとして構成された **NetScaler ADC** アプライアンスの **DNS** ログ構成の例

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
    LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add dns nameserver 8.8.8.8  - dnsProfileName dnsprofile1
Done
<!--NeedCopy-->
```

リゾルバーとして構成された **NetScaler** アプライアンスの **DNS** ログ構成の例

```
> add audit syslogAction nssyslogact1 10.102.151.136
-logLevel CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -
    logFacility LOCAL4
-timeZone LOCAL_TIME -dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
```



```
 9  > add dns profile dnsprofile1 -dnsqueryLogging ENABLED
10  Done
11  > set dns parameter -recursion enABLED
12  Done
13  > add nameserver 1.1.1.100 -local dnsProfileName dnsprofile1
14  Done
15  <!--NeedCopy-->
```

DNS のポリシーベースのロギングを構成する

ポリシーベースのロギングでは、ログメッセージの形式を指定できます。ログメッセージの内容は、高度なポリシー式を使用して定義されます。ポリシーで指定されたメッセージアクションが実行されると、NetScaler ADC アプライアンスは式からログメッセージを作成し、メッセージをログファイルに書き込みます。特定の DNS ポリシーが True と評価された場合にのみログを記録するようにアプライアンスを設定できます。

> 注
>
> 要求側の DNS プロファイルを使用して DNS ポリシーを設定した場合、NetScaler ADC アプライアンスはクエリのみをログに記録します。

DNS ポリシーのポリシーベースのロギングを設定するには、まず監査メッセージアクションを設定する必要があります。監査メッセージアクションの構成の詳細については、「監査ログ用の NetScaler アプライアンスの構成」を参照してください。監査メッセージアクションを設定したら、DNS ポリシーでメッセージアクションを指定します。

CLI を使用して **DNS** ポリシーのポリシーベースのロギングを設定する

コマンドプロンプトで次のコマンドを入力して、DNS ポリシーのポリシーベースのロギングを構成し、構成を確認します。

```
1  -  add dns action <actionName> <actionType> [-IPAddress <ip_addr|
       ipv6_addr> ... | -viewName <string> | -preferredLocList <string>
       ...] [-TTL <secs>] [-dnsProfileName <string>]
2  -  set dns policy <name> [<rule>] [-actionName <string>] [-logAction <
       string>]
3  -  show dns policy [<name>]
4  <!--NeedCopy-->
```

例 **1**:

GSLB 展開では、一般的な目的で使用される IP アドレス (内部ユーザーの IP アドレスなど) で応答するのではなく、特定のサブネットからのクライアント要求に異なる IP アドレスで応答する場合は、アクションタイプを DNS ビューとして DNS ポリシーを構成できます。この場合、特定の応答をログに記録できるように、指定された DNS アクションで DNS ロギングを構成できます。

```
1  > add dns profile dns_prof1 -dnsqueryLogging enABLED -
       dnsanswerSecLogging enABLED
```



```
 Done
> add dns view dns_view1
 Done
> add dns action dns_act1 viewName -view dns_view1  - dnsprofilename
   dns_prof1
 Done
> add dns policy dns_pol1 "CLIENT.IP.SRC.APPLY_MASK(255.255.255.0).EQ
   (100.100.100.0)”
dns_act1
 Done
> bind dns global dns_pol1 100 -gotoPriorityExpression END -type
   REQ_DEFAULT
 Done
> bind gslb service site_1_svc -viewName dns_view1 123.1.1.1
 Done
> bind gslb service site_5_svc -view dns_view1 132.1.1.1
 Done
<!--NeedCopy-->
```

注**:** 前述の設定では、GSLB 仮想サーバー (*sampletest.com* など) で構成されたドメインを照会すると、サブネット 100.100.100.0/24 のすべての内部ユーザーに DNS ビュー IP アドレスが提供され、応答がログに記録されます。他のサブネットに対するクライアント要求はログに記録されません。

例 **2**:

ドメイン *example.com* のクエリのみをログに記録する場合は、クエリロギングを有効にして DNS プロファイルを作成し、DNS プロファイルをアクションタイプが
NOOP の DNS アクションに設定してから、DNS ポリシーを作成し、DNS アクションを設定します。次に例を示します:

```
>add dns profile query_logging -dnsqueryLogging ENABLED
Done
>add dns action dns_act1 NOOP -dnsprofileName query_logging
Done
>add dns policy dns_pol1 DNS.REQ.QUESTION.DOMAIN.EQ("example.com")
   dns_act1
Done
<!--NeedCopy-->
```

DNS ポリシーのログアクションを構成して、クライアント **IP** アドレスを記録する

ロギングアクションは、次の式を使用して DNS クエリの送信元 IP をログに記録し、DNS ポリシーのログアクションの一部として使用するために使用できます。

```
> add audit messageaction log_act_custom INFORMATIONAL ""ClientIP:"
   CLIENT.IP.SRC" ECS IP:"+((DNS.REQ.OPT.ECS.IP).typecast_text_t ALT "
   NONE")"
Done
<!--NeedCopy-->
```


前述の式では、IP ヘッダー内のソース IP と DNS ECS オプションの ECS IP の両方がキャプチャされ、いずれも必要に応じて除外できます。

NetScaler ADC アプライアンスがクライアント IP アドレスを記録するための DNS ログ構成の例

DNS クエリのロギングをサンプリングする場合は、次の式を使用して実行できます。これにより、10 個のクエリのうち 1 つがログに記録されます。

```
> add audit messageaction log_action_srcip_1of10 INFORMATIONAL ""
    OneOf10: Source IP : "+client.ip.src"
Done
> add responder policy logsrcip_1of10 "sys.random.mul(10).lt(1)" NOOP -
    logAction log_action_srcip_1of10
Done
<!--NeedCopy-->
```

DNS サフィックスの設定

August 15, 2023

名前解決時に NetScaler ADC アプライアンスが非完全修飾ドメイン名を補完できるようにする DNS サフィックスを構成できます。たとえば、完全修飾されていないドメイン名 abc を解決するときに、DNS サフィックス example.com が設定されている場合、アプライアンスはドメイン名にサフィックスを追加します。次に、ドメイン名を解決します。この場合、abc.example.com が解決されます。DNS サフィックスが設定されていない場合、アプライアンスは非完全修飾ドメイン名にピリオドを追加し、ドメイン名を解決します。

DNS サフィックスの作成

DNS サフィックスには意味があり、NetScaler ADC がエンドリゾルバーまたはフォワーダーとして構成されている場合にのみ有効です。最大 127 文字のサフィックスを指定できます。

> 注:
>
> - DNS サフィックスの順序は重要です。ADC アプライアンスは、設定されたサフィックスを順番に試し、サフィックスに対する応答が成功すると停止します。
> - 一度に 1 つのドメイン名のみが処理されます。応答が成功するまで、使用可能なすべてのサフィックスにドメイン名が付加されます。
>
>   例: ドメイン名がwwwで、サフィックスがabc.comおよびabcの場合。NetScaler ADC アプライアンスは最初にwww.abc.comを試行し、それでも応答が成功しない場合、アプライアンスはwww.abcを試行します。www.abc.comの応答が成功した場合、アプライアンスは次のサフィックスを付けよう

> としません。
> - アプライアンスは、応答が成功するまで、すべてのサフィックスを追加順に使用します。

CLI を使用して DNS サフィックスを作成する

コマンドプロンプトで、次のコマンドを入力して DNS サフィックスを作成し、構成を確認します。

```
-  add dns suffix <dnsSuffix>
-  show dns suffix <dnsSuffix>
<!--NeedCopy-->
```

例:

```
> add dns suffix example.com
 Done
> show dns suffix example.com
1)      Suffix: example.com
Done

<!--NeedCopy-->
```

NetScaler コマンドラインを使用して DNS サフィックスを削除するには、`rm dns suffix` コマンドプロンプトでコマンドと DNS サフィックスの名前を入力します。

GUI を使用して DNS サフィックスを作成

[トラフィック管理] **> [DNS] > [DNS** サフィックス] に移動し、DNS サフィックスを作成します。

DNS の ANY クエリ

August 15, 2023

ANY クエリは、ドメイン名で使用可能なすべてのレコードを取得する DNS クエリの一種です。ANY クエリは、ドメインに対して権限のあるネームサーバーに送信する必要があります。

ADNS モードでの動作

ADNS モードでは、NetScaler アプライアンスはローカルキャッシュに保持されているレコードを返します。キャッシュにレコードがない場合、アプライアンスは NXDOMAIN (負) 応答を返します。

NetScaler がドメイン委任レコードと一致する場合は、NS レコードを返します。それ以外の場合は、ルートドメインの NS レコードを返します。

DNS プロキシモードでの動作

プロキシモードでは、NetScaler アプライアンスはローカルキャッシュをチェックします。キャッシュにレコードがない場合、アプライアンスはクエリをサーバーに渡します。

グローバルサーバー負荷分散 (GSLB) ドメインの動作

ADC アプライアンスで GSLB ドメインが設定され、GSLB（サイト）ドメインに対して ANY クエリが送信された場合、アプライアンスは GSLB サービスの IP アドレスを返します。負荷分散の決定によりこのサービスを選択します。マルチ IP レスポンス (MIR) オプションが有効な場合、すべての GSLB サービスの IP アドレスが送信されます。

NetScaler が ANY クエリに応答したときにこれらのレコードを返すには、GSLB ドメインに対応するすべてのレコードが NetScaler 上で構成されている必要があります。

> 注
>
> ドメインのレコードが NetScaler とサーバーの間で分散されている場合、NetScaler 上で構成されたレコードのみが返されます。

NetScaler には、DNS ビューと DNS ポリシーを構成するオプションがあります。これらのビューとポリシーは、グローバルサーバーの負荷分散を実行するために使用されます。詳しくは、「グローバルサーバー負荷分散」を参照してください。

DNS レコードのネガティブキャッシュを構成する

August 15, 2023

NetScaler アプライアンスは、ドメインの否定応答のキャッシュをサポートしています。否定応答は、要求されたドメインに関する情報が存在しないか、サーバーがクエリに対する応答を提供できないことを示します。この情報の保存はネガティブキャッシュと呼ばれます。ネガティブキャッシュは、ドメインに関するクエリへの応答を高速化するのに役立ちます。

> 注:
>
> ネガティブキャッシュは、バックエンドサーバーがクエリ対象ドメインの権限のある DNS (ADNS) サーバーとして構成されている場合にのみサポートされます。

否定応答は、次のいずれかになります。

- NXDOMAIN エラーメッセージ—クエリされたドメイン名にサーバー上にレコードが設定されていない場合、権限のある DNS サーバーは NXDOMAIN エラーメッセージで応答します。このメッセージは、クエリされたドメインが無効または存在しないドメイン名であることを示します。

- NODATA エラーメッセージ—クエリ内のドメイン名は有効でも、指定されたタイプのレコードが利用できない場合、アプライアンスは NODATA エラーメッセージを送信します。

ネガティブキャッシュが有効になっている場合、アプライアンスは DNS サーバーからのネガティブ応答をキャッシュし、キャッシュからの今後のリクエストのみを処理します。このアクションは、クエリへの応答を高速化し、バックエンド DNS トラフィックを減らすのにも役立ちます。ネガティブキャッシュは、すべての展開、つまり NetScaler ADC アプライアンスがプロキシ、エンドリゾルバ、またはフォワーダとして機能している場合に使用できます。

DNS プロファイルを使用してネガティブキャッシュを有効または無効にできます。詳細については、「DNS プロファイル」を参照してください。既定では、ネガティブキャッシュは、既定で DNS 仮想サーバーにバインドされている既定の DNS プロファイル (**default**-dns-profile) または新しく作成された DNS プロファイルに有効になっています。

CLI を使用してネガティブキャッシュを有効または無効にする

コマンドプロンプトで次のコマンドを入力してネガティブキャッシュを有効または無効にし、構成を確認します。

```
1 -  add dns profile <dnsProfileName> [-cacheRecords ( ENABLED | DISABLED
      )] [-cacheNegativeResponses (ENABLED | DISABLED )]
2 -  show dns profile [<dnsProfileName>]
3 <!--NeedCopy-->
```

デフォルトの **DNS** プロファイルの例**:**

```
1 > sh dns profile default-dns-profile
2     1)   default-dns-profile
3         Query logging : DISABLED        Answer section logging :
             DISABLED
4         Extended logging : DISABLED     Error logging : DISABLED
5         Cache Records : ENABLED     Cache Negative Responses: ENABLED
6 Done
7 <!--NeedCopy-->
```

新しく作成された **DNS** プロファイルの例**:**

```
1 > add dnsprofile dns_profile1 -cacheRecords ENABLED -
     cacheNegativeResponses ENABLED
2 Done
3 > show dns profile dns_profile1
4     1)   dns_profile1
5         Query logging : DISABLED        Answer section logging :
             DISABLED
6         Extended logging : DISABLED     Error logging : DISABLED
7         Cache Records : ENABLED     Cache Negative Responses: ENABLED
8 Done
9 <!--NeedCopy-->
```


CLI を使用してサービスレベルまたは仮想サーバーレベルの DNS パラメータを指定します

コマンドプロンプトで、次の操作を実行します。

1. DNS プロファイルを設定します。

   ```
   add dns profile <dnsProfileName> [-cacheRecords ( ENABLED |
   DISABLED )] [-cacheNegativeResponses (ENABLED | DISABLED )]
   ```

2. DNS プロファイルをサービスまたは仮想サーバーにバインドします。

   DNS プロファイルをサービスにバインドするには:

   ```
   set service <name> [-dnsProfileName <string>]
   ```

例:

```
1 >set service service1 -dnsProfileName dns_profile1
2 Done
3 <!--NeedCopy-->
```

DNS プロファイルを仮想サーバーにバインドするには:

```
set lb vserver <name> [-dnsProfileName <string>]
```

例:

```
1 >set lb vserver lbvserver1 -dnsProfileName dns_profile1
2 Done
3 <!--NeedCopy-->
```

GUI を使用してサービスレベルまたは仮想サーバーレベルの DNS パラメータを指定する

1. HTTP プロファイルを設定します。

   [システム] > [プロファイル] > [**DNS** プロファイル] に移動し、DNS プロファイルを作成します。

2. HTTP プロファイルをサービスまたは仮想サーバーにバインドします。

   [トラフィック管理] > [負荷分散] > [サービス/仮想サーバー] に移動し、サービスまたは仮想サーバーにバインドする必要がある DNS プロファイルを作成します。

アプライアンスによるネガティブ・レスポンスのレート制限

NetScaler アプライアンスがキャッシュから返す否定応答のしきい値を設定できます。しきい値を設定すると、アプライアンスはしきい値に達するまでキャッシュからの応答を返します。しきい値に達すると、アプライアンスは NXDOMAIN 応答で応答する代わりに要求をドロップします。

否定的な回答にレート制限を設定することには、次のような利点があります。

- NetScaler アプライアンスのリソースを保存します。
- 存在しないドメイン名に対する悪意のあるクエリを防止します。

> **注:** 否定応答のしきい値を設定できるのは、ADC アプライアンスが権限のあるドメインネームサーバーとして構成されているドメインだけです。権限のあるバックエンドネームサーバーから受信したキャッシュレコードのしきい値を設定することはできません。

CLI を使用してキャッシュが処理するネガティブ・レスポンスのレートを制限する

コマンドプロンプトで、次のように入力します。

```
set dns parameter -NXDOMainRateLimitThreshold <positive-integer>
<!--NeedCopy-->
```

例**:**

```
set dns parameter -NXDOMainRateLimitThreshold 1000
<!--NeedCopy-->
```

nxDomainRateLimitThreshold: このパラメータを正の整数値に設定すると、このしきい値 (秒単位) に達するまでキャッシュから応答が処理されます。しきい値を超えると、リクエストはドロップされます。しきい値はパケットエンジンごとに設定されます。

GUI を使用してキャッシュが処理するネガティブ・レスポンスのレートを制限する

1. [トラフィック管理**] > [DNS**] に移動し、[**DNS** 設定の変更] をクリックします。
2. [**DNS** パラメータの設定］ページの［**NXDOMAIN Rate Limit Threshold** ］フィールドに、応答がキャッシュから処理されるまでのしきい値を入力します。

注**:NXDOMAIN Threshold Crossed** の値には、しきい値に達した後にリクエストがドロップされた回数が表示されます。

NetScaler アプライアンスがプロキシモードのときに EDNS0 クライアントのサブネットデータをキャッシュする

December 8, 2023

NetScaler プロキシモードでは、EDNS0 クライアントサブネット（ECS）をサポートするバックエンドサーバーが ECS オプションを含む応答を送信すると、NetScaler アプライアンスは次の処理を行います。

- 応答をそのままクライアントに転送し、

- 応答をクライアントのサブネット情報とともにキャッシュに保存します。

同じドメインの同じサブネットから送信され、サーバーが同じ応答を送信する DNS リクエストは、キャッシュから処理されます。

注意:

- ECS キャッシュはデフォルトでは無効になっています。関連する DNS プロファイルの EDNS0 クライアントサブネットデータのキャッシュを有効にします。
- ドメインにキャッシュできるサブネットの数は、使用可能なサブネット ID、つまり NetScaler アプライアンスでは 1270 に制限されます。オプションで、制限を小さい数に設定できます (最小値:1 ipv4/ipv6)。

CLI を使用して ECS 応答のキャッシュを有効にする

コマンドプロンプトで入力します:

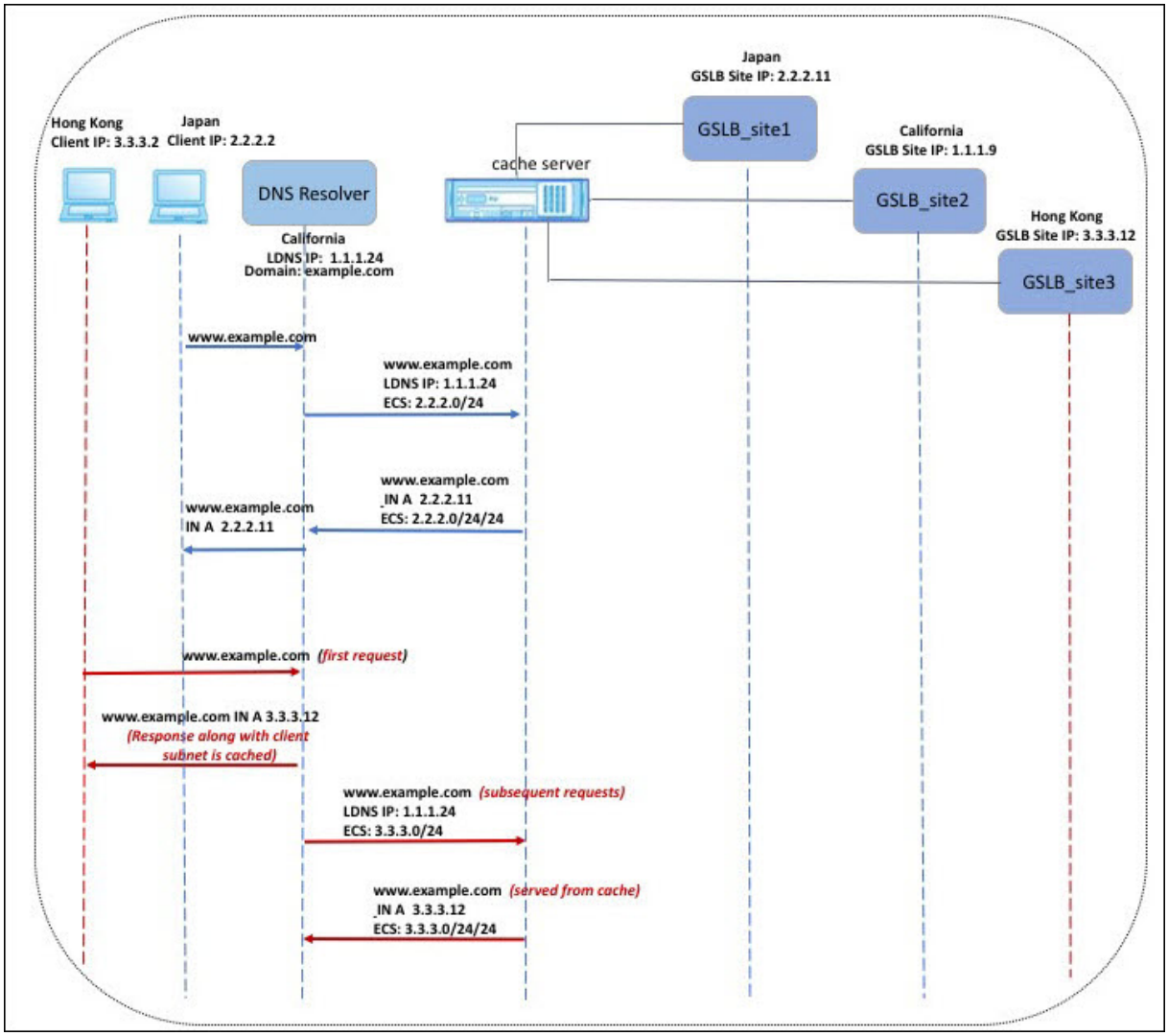
```
set dns profile <dnsProfileName> -cacheECSResponses ( ENABLED |
DISABLED )
```

CLI を使用してドメインごとにキャッシュできるサブネットの数を制限する

コマンドプロンプトで入力します:

```
set dns profile <dnsProfileName> -maxSubnetsPerDomain <positive_integer
>
```

例:



前の図に示した例では、IP アドレス 2.2.2.2 のクライアントが www.example.com のクエリを DNS リゾルバーに送信します。DNS リゾルバーは次の応答を送信します。

www.example.com IN A、IP は 2.2.11、ECS 2.2.0/24/24

この時点で、応答とクライアントサブネット識別子 (2.2.2.0/24) がキャッシュされます。同じサブネットとドメインからのさらなるリクエストは、キャッシュから処理されます。

たとえば、クライアントの IP アドレスが 2.2.2.100 で、クエリが www.example.com の場合、クエリはバックエンドサーバーに送信されるのではなく、キャッシュから提供されます。

DNS Security Extensions（DNSSEC）

August 15, 2023

DNS セキュリティ拡張機能 (DNSSEC) は、インターネット技術特別調査委員会 (IETF) の標準です。これは、UDP 応答を平文で送信しながら、ネームサーバーとクライアント間の通信におけるデータ整合性とデータ送信元認証を実現することを目的としています。DNSSEC は、非対称キー暗号と、その実装に固有の新しいリソースレコードのセットを使用するメカニズムを規定しています。

DNSSEC の仕様については、以下で説明されています。

- RFC 4033「DNS セキュリティの概要と要件」
- RFC 4034「DNS セキュリティ拡張のためのリソースレコード」
- RFC 4035「DNS セキュリティ拡張のプロトコル変更」

DNS 内に DNSSEC を実装する場合の運用上の側面については、RFC 4641「DNSSEC の運用慣行」で説明されています。

DNSSEC は NetScaler 上で構成できます。DNS ゾーンに署名するためのキーを生成およびインポートできます。NetScaler が権限を持つゾーンには DNSSEC を構成できます。ADC は、バックエンドネームサーバーのファームでホストされている署名付きゾーンの DNS プロキシサーバーとして構成できます。ADC が DNS プロキシサーバーとして構成されているゾーンに属するレコードのサブセットに対して権限がある場合は、DNSSEC 実装にレコードのサブセットを含めることができます。

DNSSEC を構成する

August 15, 2023

DNSSEC を設定するには、次の手順を実行します。

1. NetScaler アプライアンスで DNSSEC を有効にします。
2. ゾーン署名キーとゾーンのキー署名キーを作成します。
3. ゾーンに 2 つのキーを追加します。
4. キーでゾーンに署名してください。

NetScaler アプライアンスは DNSSEC リゾルバーとしては機能しません。ADC の DNSSEC は、次の導入シナリオでのみサポートされます。

1. ADN—NetScaler ADNS であり、署名自体を生成します。
2. プロキシ—NetScaler は DNSSEC プロキシとして機能します。NetScaler は、トラステッドモードで ADNS/LDNS サーバーの前に配置されていることを前提としています。ADC はプロキシキャッシュエンティティとしてのみ機能し、シグネチャの検証は行いません。

DNSSEC を有効または無効にする

ADC が DNSSEC 対応クライアントに応答できるように、NetScaler で DNSSEC を有効にします。デフォルトでは、DNSSEC は有効になっています。

NetScaler が DNSSEC 固有の情報でクライアントに応答しないようにするには、DNSSEC 機能を無効にできます。

CLI を使用して DNSSEC を有効または無効にする

コマンドプロンプトで次のコマンドを入力して DNSSEC を有効または無効にし、構成を確認します。

```
-  set dns parameter -dnssec ( ENABLED | DISABLED )
-  show dns parameter
<!--NeedCopy-->
```

例:

```
> set dns parameter -dnssec ENABLED
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
          .
          .
          .
         DNSEC Extension: ENABLED
        Max DNS Pipeline Requests: 255
 Done

<!--NeedCopy-->
```

GUI を使用して DNSSEC を有効または無効にする

1. [トラフィック管理] **> [DNS**] に移動します。
2. 詳細ウィンドウで、[DNS 設定の変更] をクリックします。
3. [**DNS** パラメータの設定] ダイアログボックスで、[**DNSSEC** 拡張を有効にする] チェックボックスをオンまたはオフにします。

ゾーンの DNS キーの作成

署名する DNS ゾーンごとに、2 組の非対称キーを作成する必要があります。ゾーン署名キー (ZSK) と呼ばれる 1 つのペアを使用して、ゾーン内のすべてのリソースレコードセットに署名します。2 番目のペアはキー署名キー (KSK) と呼ばれ、ゾーン内の DNSKEY リソースレコードの署名にのみ使用されます。

ZSK と KSK を作成すると、キーのパブリックコンポーネントの名前に`suffix.key`が追加されます。`suffix.private` プライベートコンポーネントの名前の後ろにが付加されます。追加は自動的に行われます。

NetScaler は委任署名者（DS）レコードも作成し、レコードの名前に接尾辞.ds を追加します。親ゾーンが署名付きゾーンの場合、信頼チェーンを確立するには、親ゾーンに DS レコードを公開する必要があります。

キーを作成すると、`/nsconfig/dns/` キーはディレクトリに保存されますが、ゾーンに自動的に公開されません。コマンドを使用してキーを作成したら、`create dns key` コマンドを使用してキーをゾーンに明示的に公開する必要があります。`add dns key` キーを生成するプロセスはゾーンでキーを公開するプロセスとは別の方法でキーを生成できるようにするためです。たとえば、Secure FTP（SFTP）を使用して、他のキー生成プログラム（`bind-keygen`など）によって生成されたキーをインポートし、ゾーンでキーを発行できます。ゾーンでのキーの公開の詳細については、「ゾーンでの DNS キーの公開」を参照してください。

このトピックで説明されている手順を実行してゾーン署名キーを作成し、その手順を繰り返してキー署名キーを作成します。コマンド構文に従った例では、最初にゾーン example.com のゾーン署名キーペアを作成します。次に、この例ではコマンドを使用してゾーンのキー署名キーペアを作成します。

リリース 13.0 ビルド 61.x 以降、NetScaler アプライアンスは、DNS ゾーンを認証するための RSASHA256 や RSASHA512 などのより強力な暗号化アルゴリズムをサポートするようになりました。以前は、RSASHA1 アルゴリズムのみがサポートされていました。

CLI を使用して **DNS** キーを作成する

コマンドプロンプトで入力します。

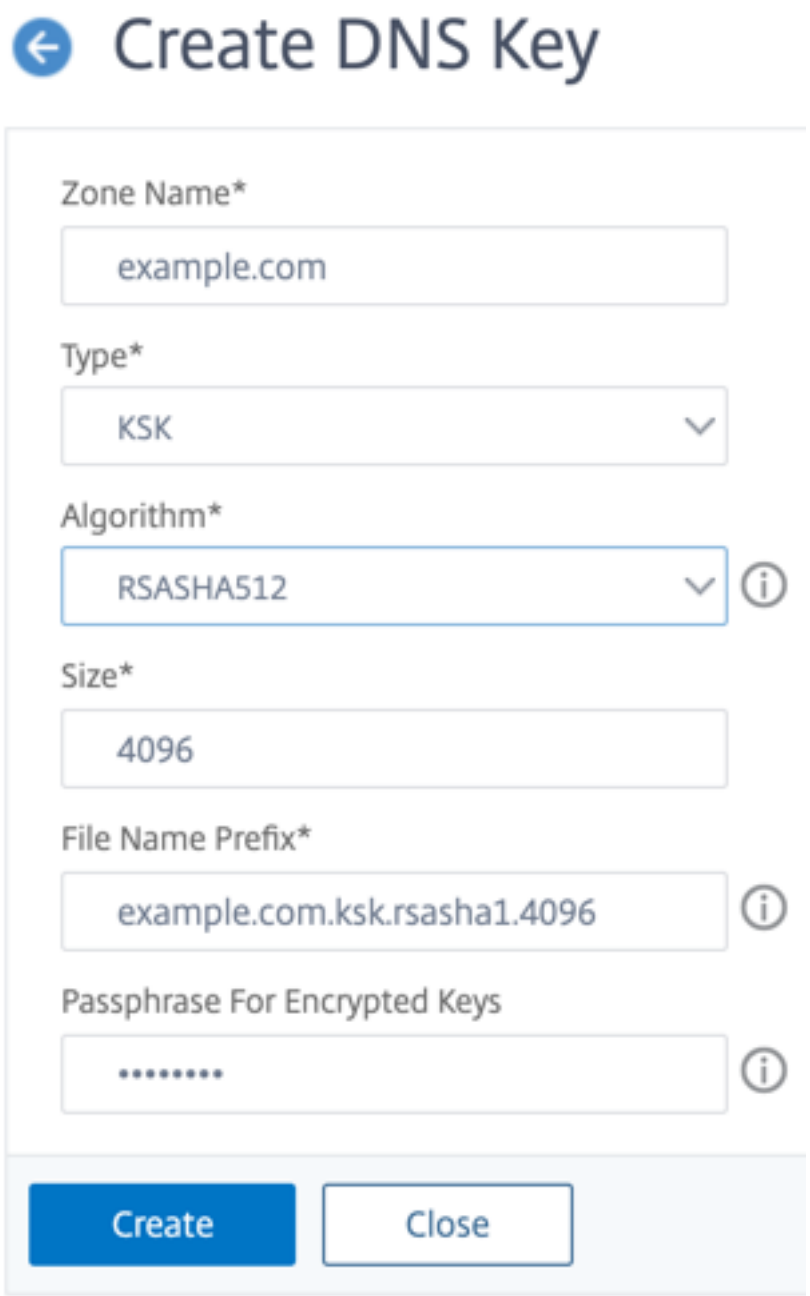
```
create dns key -zoneName <string> -keyType <keyType> -algorithm <
algorithm> -keySize <positive_integer> -fileNamePrefix <string>
```

例**:**

```
1  > create dns key -zoneName example.com -keyType zsk -algorithm
       RSASHA256 -keySize 1024 -fileNamePrefix example.com.zsk.rsasha1.1024
2  File Name: /nsconfig/dns/example.com.zsk.rsasha1.1024.key (public); /
       nsconfig/dns/example.com.zsk.rsasha1.1024.private (private); /
       nsconfig/dns/example.com.zsk.rsasha1.1024.ds (ds)
3  This operation may take some time, Please wait...
4   Done
5  > create dns key -zoneName example.com -keyType ksk -algorithm
       RSASHA512 -keySize 4096 -fileNamePrefix example.com.ksk.rsasha1.4096
6  File Name: /nsconfig/dns/example.com.ksk.rsasha1.4096.key (public); /
       nsconfig/dns/example.com.ksk.rsasha1.4096.private (private); /
       nsconfig/dns/example.com.ksk.rsasha1.4096.ds (ds)
7  This operation may take some time, Please wait...
8   Done
9  <!--NeedCopy-->
```


GUI を使用して **DNS** キーを作成する

1. [トラフィック管理] **> [DNS]** に移動します。
2. 詳細エリアで、「**DNS** キーの作成」をクリックします。
3. さまざまなパラメータの値を入力して、[作成] をクリックします。



注: 既存のキーのファイル名プレフィックスを変更するには:

- 「ブラウズ」ボタンの横にある矢印をクリックします。

- **** ローカルまたはアプライアンスのいずれかをクリックします **** （既存のキーがローカルコンピュータに保存されているか、アプライアンスの/nsconfig/dns/ディレクトリに保存されているかによって異なります）
- キーの場所を参照して、キーをダブルクリックします。
  「ファイル名プリフィックス」ボックスには、既存のキーのプレフィックスだけが入力されます。プレフィックスを適宜変更してください。

ゾーン内の **DNS** キーを発行する

キー（ゾーン署名キーまたはキー署名キー）は、ADC アプライアンスにキーを追加することによってゾーンで公開されます。ゾーンに署名する前に、キーをゾーンで公開する必要があります。

ゾーンでキーを公開する前に、キーが **/nsconfig/dns/** ディレクトリにある必要があります。(bind-keygen プログラムなどを使用して) 別のコンピュータで DNS キーを作成した場合は、そのキーが/nsconfig/dns/ディレクトリに追加されていることを確認してください。次に、キーをゾーンに公開します。ADC GUI /nsconfig/dns/ を使用してキーをディレクトリに追加します。または、Secure FTP (SFTP) など、他のプログラムを使用してキーをディレクトリにインポートします。

add dns key 特定のゾーンで公開する公開鍵と秘密鍵のペアごとにコマンドを使用してください。ゾーン用に ZSK ペアと KSK ペアを作成した場合は、add dns key コマンドを使用して最初にゾーン内のキーペアの 1 つを公開します。コマンドを繰り返して、もう一方のキーペアを公開します。ゾーンで公開するキーごとに、ゾーンに DNSKEY リソースレコードが作成されます。

コマンド構文に従った例では、最初にゾーンの (example.com ゾーン用に作成された) ゾーン署名キーペアを公開します。次に、この例ではコマンドを使用してキー署名キーペアをゾーンに公開します。

CLI を使用してゾーンにキーを公開する

コマンドプロンプトで次のコマンドを入力してゾーンにキーを公開し、構成を確認します。

```
1 -  add dns key <keyName> <publickey> <privatekey> [-expires <
     positive_integer> [<units>]] [-notificationPeriod <positive_integer>
      [<units>]] [-TTL <secs>]
2 -  show dns zone [<zoneName> | -type <type>]
3 <!--NeedCopy-->
```

例**:**

```
1 > add dns key example.com.zsk example.com.zsk.rsasha1.1024.key example.
     com.zsk.rsasha1.1024.private
2  Done
3 > add dns key example.com.ksk example.com.ksk.rsasha1.4096.key example.
     com.ksk.rsasha1.4096.private
4  Done
5 > show dns zone example.com
```



```
6           Zone Name : example.com
7           Proxy Mode : NO
8           Domain Name : example.com
9                   Record Types : NS SOA DNSKEY
10          Domain Name : ns1.example.com
11                  Record Types : A
12          Domain Name : ns2.example.com
13                  Record Types : A
14  Done
15 <!--NeedCopy-->
```

GUI を使用して **DNS** ゾーンにキーを公開する

[トラフィック管理] **> [DNS] > [**キー] に移動します。

注: 公開鍵と秘密鍵の場合、ローカルコンピューターに保存されている鍵を追加するには、「ブラウズ」ボタンの横にある矢印をクリックし、「ローカル」をクリックし、鍵の場所を参照して、鍵をダブルクリックします。

DNS キーの設定

ゾーンで公開されているキーのパラメータを設定できます。キーの有効期限、通知期間、および有効期間 (TTL) パラメータを変更できます。キーの有効期限を変更すると、アプライアンスはゾーン内のすべてのリソースレコードにそのキーで自動的に再署名します。再署名は、ゾーンが特定のキーで署名されている場合に行われます。

CLI を使用してキーを設定する

コマンドプロンプトで次のコマンドを入力してキーを構成し、構成を確認します。

```
1 -  set dns key <keyName> [-expires <positive_integer> [<units>]] [-
     notificationPeriod <positive_integer> [<units>]] [-TTL <secs>]
2 -  show dns key [<keyName>]
3 <!--NeedCopy-->
```

例:

```
1 > set dns key example.com.ksk -expires 30 DAYS -notificationPeriod 3
     DAYS -TTL 3600
2  Done
3 > show dns key example.com.ksk
4 1)      Key Name: example.com.ksk
5         Expires: 30 DAYS        Notification: 3 DAYS    TTL: 3600
6         Public Key File: example.com.ksk.rsasha1.4096.key
7         Private Key File: example.com.ksk.rsasha1.4096.private
8  Done
9 <!--NeedCopy-->
```


GUI を使用してキーを設定する

1. [トラフィック管理] **> [DNS] > [**キー] に移動します。
2. 詳細ウィンドウで、設定するキーをクリックし、[開く] をクリックします。
3. [DNS キーの設定] ダイアログボックスで、次のパラメータの値を次のように変更します。
   - 期限切れ-期限切れ
   - 通知期間—notificationPeriod
   - TTL —TTL
4. 「OK」をクリックします。

DNS ゾーンへの署名と署名解除

DNS ゾーンを保護するには、ゾーンで公開されているキーを使用してゾーンに署名する必要があります。ゾーンに署名すると、NetScaler は所有者名ごとにネクストセキュア（NSEC）リソースレコードを作成します。次に、キー署名キーを使用して DNSKEY リソースレコードセットに署名します。最後に、ZSK を使用して、DNSKEY リソースレコードセットや NSEC リソースレコードセットを含む、ゾーン内のすべてのリソースレコードセットに署名します。署名操作を行うたびに、ゾーン内のリソースレコードセットが署名されます。署名は、RRSIG リソースレコードと呼ばれる新しいリソースレコードに取り込まれます。

ゾーンに署名したら、設定を保存します。

CLI を使用してゾーンに署名する

コマンドプロンプトで次のコマンドを入力してゾーンに署名し、構成を確認します。

```
-  sign dns zone <zoneName> [-keyName <string> ...]
-  show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
-  save config
<!--NeedCopy-->
```

例**:**

```
> sign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
          Zone Name : example.com
          Proxy Mode : NO
          Domain Name : example.com
                  Record Types : NS SOA DNSKEY RRSIG NSEC
          Domain Name : ns1.example.com
                  Record Types : A RRSIG NSEC
          Domain Name : ns2.example.com
                  Record Types : A RRSIG
```



```
12          Domain Name : ns2.example.com
13                  Record Types : RRSIG NSEC
14 Done
15 > save config
16  Done
17 <!--NeedCopy-->
```

CLI を使用してゾーンの署名を解除する

コマンドプロンプトで次のコマンドを入力してゾーンの署名を解除し、構成を確認します。

```
1 -  unsign dns zone <zoneName> [-keyName <string> ...]
2 -  show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
3 <!--NeedCopy-->
```

例**:**

```
1 > unsign dns zone example.com -keyName example.com.zsk example.com.ksk
2  Done
3 > show dns zone example.com
4          Zone Name : example.com
5          Proxy Mode : NO
6          Domain Name : example.com
7                  Record Types : NS SOA DNSKEY
8          Domain Name : ns1.example.com
9                  Record Types : A
10         Domain Name : ns2.example.com
11                 Record Types : A
12  Done
13 <!--NeedCopy-->
```

GUI を使用してゾーンに署名または署名解除する

1. [トラフィック管理**] > [DNS] > [**ゾーン] に移動します。
2. 詳細ウィンドウで、署名するゾーンをクリックし、[署名/署名解除] をクリックします。
3. DNS ゾーンへの署名/署名解除ダイアログボックスで、次のいずれかの操作を行います。
   - ゾーンに署名するには、ゾーンに署名するキー (ゾーン署名キーとキー署名キー) のチェックボックスを選択します。

     複数のゾーン署名キーまたはキー署名キーペアを使用してゾーンに署名できます。
   - ゾーンの署名を解除するには、ゾーンの署名を解除するキー (ゾーン署名キーとキー署名キー) のチェックボックスをオフにします。

     複数のゾーン署名キーまたはキー署名キーペアを使用してゾーンの署名を解除できます。
4. 「OK」をクリックします。

ゾーン内の特定のレコードの **NSEC** レコードを表示する

NetScaler がゾーン内の所有者名ごとに自動的に作成する NSEC レコードを表示できます。

CLI を使用してゾーン内の特定のレコードの **NSEC** レコードを表示する

コマンドプロンプトで次のコマンドを入力すると、ゾーン内の特定のレコードの NSEC レコードが表示されます。

```
show dns nsecRec [<hostName> | -type (ADNS | PROXY | ALL)]
```

例:

```
> show dns nsecRec example.com
1)      Domain Name : example.com
        Next Nsec Name: ns1.example.com
        Record Types : NS SOA DNSKEY RRSIG NSEC
 Done
<!--NeedCopy-->
```

GUI を使用してゾーン内の特定のレコードの **NSEC** レコードを表示する

1. [トラフィック管理] **>** **[DNS]** **>** **[**レコード**]** **>** **[**次のセキュアレコード] に移動します。
2. 詳細ペインで、NSEC レコードを表示するレコードの名前をクリックします。選択したレコードの NSEC レコードが [詳細] 領域に表示されます。

DNS キーを削除する

キーの有効期限が切れた場合、またはキーが不正アクセスされた場合は、公開されているゾーンからキーを削除します。ゾーンからキーを削除すると、ゾーンには自動的にそのキーの署名が解除されます。このコマンドでキーを削除しても、/nsconfig/dns/ ディレクトリにあるキーファイルは削除されません。キーファイルが不要になった場合は、ディレクトリから明示的に削除する必要があります。

CLI を使用して **NetScaler** からキーを削除する

コマンドプロンプトで次のコマンドを入力してキーを削除し、構成を確認します。

```
-  rm dns key <keyName>
-  show dns key <keyName>
<!--NeedCopy-->
```

例:


```
> rm dns key example.com.zsk
 Done
> show dns key example.com.zsk
ERROR: No such resource [keyName, example.com.zsk]

<!--NeedCopy-->
```

GUI を使用して **NetScaler** からキーを削除する

1. [トラフィック管理] **> [DNS] > [**キー] に移動します。
2. 詳細ペインで、ADC から削除するキーの名前をクリックし、[削除] をクリックします。

NetScaler ADC がゾーンに対して権限がある場合に **DNSSEC** を構成する

August 15, 2023

NetScaler が特定のゾーンに対して権限を持っている場合、ゾーン内のすべてのリソースレコードが ADC で構成されます。権限のあるゾーンに署名するには、ゾーンのゾーン署名キーとキー署名キーを作成し、そのキーを ADC に追加してから、ゾーンに署名する必要があります。詳しくは、次のトピックを参照してください:

- ゾーンの DNS キーの作成
- ゾーン内の DNS キーを発行する
- DNS ゾーンに署名し、署名解除します。

ADC で構成された GSLB ドメインが署名されているゾーンに属している場合、GSLB ドメイン名は、ゾーンに属する他のレコードとともに署名されます。

ゾーンに署名すると、DNSSEC 対応クライアントからの要求に対する応答には、要求されたリソースレコードとともに RRSIG リソースレコードが含まれます。DNSSEC は ADC で有効にする必要があります。DNSSEC の有効化の詳細については、DNSSEC の有効化および無効化を参照してください。

最後に、DNSSEC を権限ゾーン用に構成したら、NetScaler ADC 構成を保存する必要があります。

NetScaler ADC が **DNS** プロキシサーバーであるゾーンの **DNSSEC** を構成する

August 15, 2023

NetScaler が DNS プロキシサーバーとして構成されているゾーンに署名する手順は、ADC がバックエンドネームサーバーが所有するゾーン情報のサブセットを所有しているかどうかによって異なります。その場合、その構成は

部分的なゾーン所有権構成と見なされます。ADC がゾーン情報のサブセットを所有していない場合、バックエンドサーバーを管理するための NetScaler 構成はゾーンレスの DNS プロキシサーバー構成と見なされます。どちらの NetScaler 構成でも、基本的な DNSSEC 構成タスクは同じです。ただし、NetScaler の部分ゾーンに署名するには、いくつかの追加構成手順が必要です。

注：ゾーンレスプロキシサーバー構成とパーシャルゾーンという用語は、NetScaler アプライアンスでのみ使用されます。

重要**:** プロキシモードに設定した場合、ADC はキャッシュを更新する前に DNSSEC 応答の署名検証を行いません。

ADC を DNS プロキシとして設定して DNSSEC 対応リゾルバー (サーバー) の負荷を分散する場合は、DNS 仮想サーバーの設定時に Recursion Available オプションを設定する必要があります。DNSSEC クエリが Checking Disabled (CD) ビットを設定して届いた場合、そのクエリは CD ビットを保持したままサーバーに渡されます。サーバーからの応答はキャッシュされません。

ゾーンレス **DNS** プロキシサーバー構成の **DNSSEC** の設定

ゾーンレス DNS プロキシサーバー構成では、バックエンドのネームサーバーでゾーン署名を実行する必要があります。NetScaler では、ADC をゾーンの DNS プロキシサーバーとして構成します。プロトコルタイプ DNS の負荷分散仮想サーバーを作成します。ネームサーバーを表すように ADC 上のサービスを設定します。次に、サービスを負荷分散仮想サーバーにバインドします。これらの構成タスクの詳細については、「NetScaler を DNS プロキシサーバーとして構成する」を参照してください。

クライアントが DNSSEC OK（DO）ビットが設定された DNS 要求を ADC に送信すると、ADC は要求された情報についてキャッシュをチェックします。リソースレコードがキャッシュにない場合、ADC は要求を DNS ネームサーバーの 1 つに転送します。次に、ネームサーバーからの応答をクライアントに中継します。また、ADC は RRSIG リソースレコードをネームサーバーからの応答とともにキャッシュします。DNSSEC 対応クライアントからの後続のリクエストは、有効期間 (TTL) パラメータに従って、キャッシュ (RRSIG リソースレコードを含む) から処理されます。クライアントが DO ビットを設定せずに DNS 要求を送信すると、ADC は要求されたリソースレコードのみで応答します。DNSSEC に固有の RRSIG リソースレコードは含まれていません。

部分ゾーン所有権設定用の **DNSSEC** の設定

ADC 構成によっては、ゾーンの権限がバックエンドのネームサーバーにある場合でも、そのゾーンに属するリソースレコードのサブセットが ADC で設定されることがあります。ADC は、このレコードのサブセットのみを所有 (または権限) しています。このようなレコードのサブセットは、*ADC* の部分的なゾーンを構成していると見なすことができます。ADC は部分ゾーンを所有しています。他のすべてのレコードは、バックエンドのネームサーバーが所有しています。

NetScaler の一般的な部分ゾーン構成は、次の場合に見られます。

- グローバルサーバー負荷分散 (GSLB) ドメインは ADC で設定されます
- GSLB ドメインは、バックエンドのネームサーバーが権限を持つゾーンの一部です。

ADC 上の一部のゾーンのみを含むゾーンに署名するには、次のことが必要です。

- バックエンド・ネーム・サーバー・ゾーン・ファイルへの部分的なゾーン情報の追加
- バックエンドネームサーバー上のゾーンへの署名
- ADC のパーシャルゾーンへの署名

ネームサーバーのゾーンと ADC の部分ゾーンの署名には、同じキーセットを使用する必要があります。

バックエンドのネームサーバー上のゾーンに署名します

1. 部分ゾーンに含まれるリソースレコードをネームサーバーのゾーンファイルに含めます。
2. キーを作成し、そのキーを使用してバックエンドネームサーバー上のゾーンに署名します。

NetScaler の部分ゾーンに署名します

1. バックエンドのネームサーバーが所有するゾーンの名前でゾーンを作成します。部分ゾーンを構成するときは、ProxyMode パラメーターを YES に設定します。このゾーンは、ADC が所有するリソースレコードを含む部分ゾーンです。

   たとえば、バックエンドのネームサーバーで設定されているゾーンの名前が example.com の場合、ADC に example.com という名前のゾーンを作成する必要があります。proxyMode パラメーターを YES に設定します。ゾーンの追加の詳細については、「DNS ゾーンの構成」を参照してください。

   > 注
   >
   > ゾーンに SOA レコードと NS レコードを追加しないでください。これらのレコードは、ADC が権限を持つゾーンの ADC に存在している必要があります。

2. キーを（バックエンドネームサーバーの 1 つから）ADC にインポートしてから、/nsconfig/dns/ ディレクトリ。キーをインポートして ADC に追加する方法の詳細については、「ゾーンでの DNS キーの公開」を参照してください。
3. インポートされたキーで部分ゾーンに署名します。キーを使用して部分ゾーンに署名すると、ADC はリソースレコードセットの RRSIG レコードと NSEC レコードをそれぞれ生成し、部分ゾーン内の個々のリソースレコードを生成します。ゾーンの署名の詳細については、「DNS ゾーンの署名と署名の解除」を参照してください。

グローバルサーバー負荷分散 **(GSLB)** ドメイン名の **DNSSEC** の設定

August 15, 2023

GSLB が NetScaler ADC で構成されており、ADC が GSLB ドメイン名が属するゾーンに対して権限を持っている場合、ゾーンが署名されるときにすべての GLSB ドメイン名が署名されます。ADC が権限のあるゾーンの署名の詳

細については、「NetScaler ADC アプライアンスがゾーンに対して権限を持つ場合に DNSSEC を構成する」を参照してください。

GSLB ドメインが、バックエンドネームサーバーが権限を持つゾーンに属している場合は、次のことを行う必要があります。

- まず、ネームサーバー上のゾーンに署名します。
- 次に、ADC の部分ゾーンに署名して、ゾーンの DNSSEC 構成を完了します。

詳細については、「ゾーンの所有権の一部構成の DNSSEC の構成」を参照してください。

ゾーンのメンテナンス

August 15, 2023

DNSSEC の観点から見ると、ゾーンのメンテナンスでは、キーの有効期限が迫っているときにゾーン署名キーとキー署名キーをロールオーバーする必要があります。これらのゾーンメンテナンスタスクは手動で実行する必要があります。ゾーンは自動的に再署名されるため、手動で操作する必要はありません。

更新したゾーンに再署名する

ゾーンが更新（レコードの追加または既存のレコードの変更）されると、アプライアンスは新しい（または変更された）レコードに自動的に再署名します。ゾーンに複数のゾーン署名キーが含まれている場合、アプライアンスはゾーンの署名に使用されたキーで新しい（または変更された）レコードに再署名します。

DNSSEC キーをロールオーバー

注**:** 有効期限が切れる前に DNSSEC キー (KSK、ZSK) を手動でロールオーバーしてください。

NetScaler では、プリパブリッシュと二重署名の方法を使用して、ゾーン署名キーとキー署名キーのロールオーバーを実行できます。これら 2 つのロールオーバー方法の詳細については、RFC 4641「DNSSEC の運用慣行」を参照してください。

以下のトピックでは、ADC のコマンドを RFC 4641 で説明されているロールオーバー手順のステップにまとめています。

キーの有効期限の通知は、dnskeyExpiry という SNMP トラップを介して送信されます。dnskey 有効期限 SNMP トラップと共に、dnskeyName、有効期限が切れるまでの時間、および期限切れの dnskey 単位の 3 つの MIB 変数が送信されます。詳細については、「*NetScaler 12.0 SNMP OID リファレンス*」の「*NetScaler SNMP OID リファレンス*」を参照してください。

公開前キーのロールオーバー

RFC 4641「DNSSEC の運用慣行」では、プレパブリッシュキーのロールオーバー方法について、初期、新規 DNSKEY、新しい RRSIGs、および DNSKEY の削除という 4 つの段階が定義されています。各ステージには、ADC で実行する必要がある一連のタスクが関連付けられています。次に、各ステージと実行する必要があるタスクの説明を示します。ここで説明するロールオーバー手順は、キー署名キーとゾーン署名キーの両方に使用できます。

- ステージ **1:** 初期。ゾーンには、そのゾーンが現在署名されているキーセットのみが含まれます。初期段階のゾーンの状態は、キーロールオーバープロセスを開始する直前のゾーンの状態です。

  例**:**

  example.com.zsk1 というキーを考えてみましょう。このキーを使用して example.com ゾーンが署名されます。ゾーンには、有効期限が近づいている example.com.zsk1 キーによって生成された RRSIG のみが含まれます。キー署名キーは example.com.ksk1 です。

- ステージ **2**: 新しいダンスキー。新しいキーが作成され、ゾーンで公開されます。つまり、キーは ADC に追加されますが、プレロールフェーズが完了するまでゾーンは新しいキーで署名されません。このステージでは、ゾーンには古いキー、新しいキー、および古いキーによって生成された RRSig が含まれます。プレロールフェーズの全期間にわたって新しいキーを公開すると、新しいキーに対応する DNSKEY リソースレコードがセカンダリネームサーバーに伝達されるまでの時間が与えられます。

  例**:**

  example.com.zsk2 という新しいキーが example.com ゾーンに追加されます。プレロールフェーズが完了するまで、ゾーンは example.com.zsk2 で署名されません。example.com ゾーンには、example.com.zsk1 と example.com.zsk2 の両方の DNSKEY リソースレコードが含まれています。

  NetScaler コマンド**:**

  ADC で次のタスクを実行します。

  - `create dns key`コマンドを使用して DNS キーを作成します。

    例を含む DNS キーの作成の詳細については、「ゾーンの DNS キーを作成する」を参照してください。

  - `add dns key`コマンドを使用して、ゾーン内の新しい DNS キーを発行します。

    例など、ゾーンでのキーの公開の詳細については、「ゾーンでの DNS キーの公開」を参照してください。

- ステージ **3:** 新しい **RRSIGs**。ゾーンは新しい DNS キーで署名され、その後古い DNS キーで署名が解除されます。古い DNS キーはゾーンから削除されず、古いキーによって生成された RRSig の有効期限が切れるまで公開されたままになります。

  例**:**

  ゾーンは example.com.zsk2 で署名され、次に example.com.zsk1 で署名解除されます。ゾーンは example.com.zsk1 によって生成された RRSIG の有効期限が切れるまで、example.com.zsk1 を引き続き公開します。

NetScaler コマンド**:**

ADC で次のタスクを実行します。

- `sign dns zone` コマンドを使用して、新しい DNS キーでゾーンに署名します。
- `unsign dns zone`コマンドを使用して、古い DNS キーを使用してゾーンの署名を解除します。

例を含むゾーンの署名と署名の解除の詳細については、「DNS ゾーンの署名と署名の解除」を参照してください。

- ステージ **4**: **DNSKEY** の取り外し。古い DNS キーによって生成された RRSig の有効期限が切れると、古い DNS キーはゾーンから削除されます。

  例**:**

  古い DNS キー example.com.zsk1 は example.com ゾーンから削除されます。

  NetScaler コマンド

  ADC で、`rm dns key` コマンドを使用して、古い DNS キーを削除します。例など、ゾーンからキーを削除する方法の詳細については、「DNS キーの削除」を参照してください。

二重署名キーのロールオーバー

RFC 4641「DNSSEC 運用慣行」では、二重署名キーのロールオーバーについて、初期、新規 DNSKEY、および DNSKEY 削除の 3 段階が定義されています。各ステージには、ADC で実行する必要がある一連のタスクが関連付けられています。次に、各ステージと実行する必要があるタスクの説明を示します。ここで説明するロールオーバー手順は、キー署名キーとゾーン署名キーの両方に使用できます。

- ステージ **1:** 初期。ゾーンには、そのゾーンが現在署名されているキーセットのみが含まれます。初期段階のゾーンの状態は、キーロールオーバープロセスを開始する直前のゾーンの状態です。

  例**:**

  example.com.zsk1 というキーを考えてみましょう。このキーを使用して example.com ゾーンが署名されます。ゾーンには、有効期限が近づいている example.com.zsk1 キーによって生成された RRSIG のみが含まれます。キー署名キーは example.com.ksk1 です。

- ステージ **2**: 新しいダンスキー。新しい鍵がゾーンで公開され、ゾーンは新しい鍵で署名されます。ゾーンには、古いキーと新しいキーによって生成された RRSig が含まれます。ゾーンに両方の RRSIG セットが含まれている必要がある最小期間は、すべての RRSIG の有効期限が切れるまでに必要な時間です。

  例**:**

  example.com.zsk2 という新しいキーが example.com ゾーンに追加されます。ゾーンは example.com.zsk2 で署名されています。example.com ゾーンには、両方のキーから生成された RRSig が含まれるようになりました。

  NetScaler コマンド

ADC で次のタスクを実行します。

- `create dns key`コマンドを使用して DNS キーを作成します。

  例を含む DNS キーの作成の詳細については、「ゾーンの DNS キーを作成する」を参照してください。

- `add dns key`コマンドを使用して、ゾーンに新しいキーを発行します。

  例など、ゾーンでのキーの公開の詳細については、「ゾーンでの DNS キーの公開」を参照してください。

- `sign dns zone`コマンドを使用して、新しいキーでゾーンに署名します。

  例を含むゾーンの署名の詳細については、「DNS ゾーンの署名と署名の解除」を参照してください。

- ステージ **3**: **DNSKEY** の取り外し。古い DNS キーによって生成された RRSig の有効期限が切れると、古い DNS キーはゾーンから削除されます。

  例**:**

  古い DNS キー example.com.zsk1 は example.com ゾーンから削除されます。

  NetScaler コマンド**:**

  ADC で、`rm dns key` コマンドを使用して、古い DNS キーを削除します。

  例など、ゾーンからキーを削除する方法の詳細については、「DNS キーの削除」を参照してください。

DNSSEC 操作を NetScaler ADC にオフロードする

August 15, 2023

DNS サーバーが権限を持つ DNS ゾーンの場合、DNSSEC 操作を ADC アプライアンスにオフロードできます。DNSSEC オフロード環境では、DNS サーバーは署名されていない応答を送信します。ADC は、応答をクライアントに中継する前に動的に署名します。ADC は署名された応答もキャッシュします。DNS サーバーの負荷を軽減する以外に、DNSSEC 操作を ADC にオフロードすると、次のようなメリットがあります。

- DNS サーバーがプログラムで生成するレコードに署名できます。このようなレコードは、DNS サーバーで実行される日常的なゾーン署名操作では署名できません。
- サーバーに DNSSEC を実装していなくても、署名された応答をクライアントに提供できます。

DNSSEC オフロードを設定するには、DNS 負荷分散仮想サーバーを構成し、DNS サーバーを表すサービスを構成してから、サービスを仮想サーバーにバインドする必要があります。DNS 負荷分散仮想サーバーの構成、サービスの構成、および仮想サーバーへのサービスのバインドの詳細については、「DNS ゾーンの構成」を参照してください。

DNSSEC 操作をオフロードする DNS ゾーンごとに、ADC にゾーンエンティティを作成します。DNS ゾーンごとに、プロキシモードと DNSSEC オフロードパラメータを有効にする必要があります。オプションで、オフロードゾーン

の NSEC レコード生成を設定できます。DNSSEC オフロード用の DNS ゾーンエンティティを作成するには、このトピックの指示に従ってください。

設定を完了するには、ゾーンの DNS キーを生成し、そのキーをゾーンに追加し、そのキーでゾーンに署名する必要があります。このプロセスは、通常の DNSSEC と同じです。キーの作成、ゾーンへのキーの追加、およびゾーンの署名については、「ドメインネームシステムのセキュリティ拡張」を参照してください。

DNS オフロードを構成したら、NetScaler ADC 上の DNS キャッシュをフラッシュする必要があります。DNS キャッシュをフラッシュすると、キャッシュ内の署名されていないレコードがすべて削除され、署名されたレコードに置き換えられます。DNS キャッシュのフラッシュの詳細については、「DNS レコードのフラッシュ」を参照してください。

CLI を使用してゾーンの DNSSEC オフロードを有効にする

コマンドラインで次のコマンドを入力してゾーンの DNSSEC オフロードを有効にし、構成を確認します。

```
1 -  add dns zone <zoneName> -proxyMode YES -dnssecOffload ENABLED [-nsec
      ( ENABLED | DISABLED )
2 -  show dns zone
3 <!--NeedCopy-->
```

例:

```
1 > add dns zone example.com -proxyMode YES -dnssecOffload ENABLED nsec
     ENABLED
2  Done
3 > show dns zone example.com
4       Zone Name : example.com
5       Proxy Mode : YES
6       DNSSEC Offload: ENABLED    NSEC: ENABLED
7  Done
8 <!--NeedCopy-->
```

GUI を使用してゾーンの DNSSEC オフロードを有効にする

1. [トラフィック管理] **> [DNS] > [**ゾーン] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - NetScaler にゾーンを作成するには、「追加」をクリックします。
   - 既存のゾーンの DNSSEC オフロードを設定するには、そのゾーンをダブルクリックします。
3. [DNS ゾーンの作成] または [DNS ゾーンの設定] ダイアログボックスで、[プロキシモード] と [DNSSEC オフロード] チェックボックスを選択します。
4. オプションで、NetScaler にゾーンの NSEC レコードを生成させたい場合は、NSEC チェックボックスを選択します。

DNSSEC の管理パーティションのサポート

August 15, 2023

パーティション化された NetScaler アプライアンスでは、生成された DNS キーは次の場所に保存されます。

- デフォルトパーティション:/nsconfig/dns/
- <partitionname> デフォルト以外のパーティション:/nsconfig/パーティション/ /dns/

これで、DNS キーにパスワードを追加できます。DNS キーにパスワードを追加するには、`create dns key` まずコマンドにパスワードを追加する必要があります。次に、DNS キーを ADC アプライアンスに追加するときに、`add dns key` コマンドに同じパスワードを入力します。次に例を示します:

```
create dns key -zoneName com -keytype ksK -algorithm rsASHA1 -keysize
 4096 - fileNamePrefix com.ksk.rsasha1.4096 -password 1jsfd3Wa

add dns key com.zsk.4096 /nsconfig/dns/com.zsk.rsasha1.4096.private -
password 1jsfd3Wa
```

注:

- デフォルトのパーティション環境では、キーはデフォルトの場所/nsconfig/dns/ から読み込まれます。ただし、キーが別の場所に保存されている場合は、`add dns key -private` コマンドでパス名を指定する必要があります。例: `add dns key -private` <path name>。
- デフォルト以外のパーティション環境の場合、キーはデフォルトの場所 /nsconfig/partitions/<partitionname>/dns/から読み取られます。

ワイルドカード DNS ドメインのサポート

August 15, 2023

ワイルドカード DNS ドメインは、存在しないドメインやサブドメインのリクエストを処理するために使用されます。ゾーンでは、ドメインごとに個別のリソースレコード (RR) を作成するのではなく、ワイルドカードドメインを使用して、存在しないすべてのドメインまたはサブドメインのクエリを特定のサーバーにリダイレクトします。ワイルドカード DNS ドメインの最も一般的な用途は、インターネットから他のメールシステムにメールを転送するために使用できるゾーンを作成することです。

DNS 解決では、ワイルドカード RR はワイルドカードドメインをサポートします。ワイルドカード RR は、存在しないドメイン名のクエリに対する応答を合成するために使用されます。たとえば、`http://image.example.com`クエリを実行したのにサブドメイン「image」が存在しなかった場合、example.com にリダイレクトされる可能性があります。

ワイルドカードレコードには、ドメイン名の左端のラベルとしてアスタリスク (*) 文字があります。例: *.example.com。ドメイン名の他の場所にあるアスタリスクは、ワイルドカード DNS レコードを意味します。たとえば、**new.*.example.com** は有効なワイルドカード DNS レコードではありません。

> 注
>
> - ワイルドカードドメインは、NetScaler アプライアンスがゾーンに対して権限を持ち、ADNS または DNS プロキシサーバーとして構成されている場合にのみサポートされます。
> - ワイルドカードドメインは NS レコードと SOA レコードではサポートされていません。
> - クエリが別のゾーンにある場合、ワイルドカードドメインは適用できません。
> - QNAME またはワイルドカードドメインと QNAME の間の名前が存在することがわかっている場合は、ワイルドカードドメインを適用できません。

設定例

```
add dns soaRec example.com -originServer n1.example.com -contact admin.
   example.com

add dns nsRec example.com n1.example.com

add dns nsRec example.com n2.example.com

add dns zone example.com  -proxyMode no

add dns addrec www.example.com 2.2.2.2

add dns addrec *.example.com 10.10.10.10

add dns addrec *.example.com 10.10.10.11

add dns aaaarec *.example.com 2001::1
<!--NeedCopy-->
```

この例では、A レコードと AAAA レコードにワイルドカードドメイン名が追加されています。

ゾーンに存在するドメイン名に対するクエリを受信すると、NetScaler アプライアンスは対応する応答で応答します。たとえば、この例では www.example.com、アプライアンスは 2.2.2.2 で応答します。

ワイルドカードタイプと一致する存在しないドメイン名については、合成された応答が配信されます。

この例では、NetScaler アプライアンスは、存在しない.example.com または xyz.example.com のドメイン名に対して 10.10.10.10 と 10.10.11 で応答します。

ワイルドカード合成は、ゾーンに存在するドメイン名には適用されません。

たとえば、クエリwww.example.comとタイプが AAAA の場合、NetScaler アプライアンスはタイプ A とwww.example.comが存在するため、ワイルドカードで合成しません。この例では、NetScaler アプライアンスは NODATA 応答で応答します。

abc.example.com と入力して AAAA と入力したクエリの場合、NetScaler アプライアンスは合成された応答で応答します。たとえば、この例では `www.example.com`、アプライアンスは 2001:: 1 で応答します。

DNS DDoS 攻撃を軽減する

August 15, 2023

DNS サーバーはネットワークの最も重要なコンポーネントの 1 つであり、攻撃から身を守る必要があります。DNS 攻撃の最も基本的なタイプの 1 つは DDoS 攻撃です。この種の攻撃は増加傾向にあり、破壊的な被害をもたらす可能性があります。DDoS 攻撃を軽減するには、次の方法があります。

- ネガティブなレコードをフラッシュします。
- ネガティブレコードの存続期間 (TTL) を制限します。
- DNS キャッシュによって消費されるメモリを制限することで、NetScaler のメモリを節約できます。
- DNS レコードをキャッシュに保持します。
- DNS キャッシュバイパスを有効にします。

ネガティブレコードをフラッシュする

DNS 攻撃は、キャッシュをネガティブレコード (NXDOMAIN と NODATA) でいっぱいにします。その結果、正当な要求に対する応答はキャッシュされないため、新しい要求は DNS 解決のためにバックエンドサーバーに送信されます。そのため、応答が遅れます。

NetScaler アプライアンスの DNS キャッシュから負の DNS レコードをフラッシュできるようになりました。

CLI を使用してネガティブキャッシュレコードをフラッシュする

コマンドプロンプトで入力します。

```
flush dns proxyrecords -type (dnsRecordType | negRecType)NXDOMAIN |
NODATA
```

例**:**

```
flush dns proxyrecords -negRecType NODATA
```

GUI によるネガティブ・キャッシュ・レコードのフラッシュ

1. [設定] > [トラフィック管理] > [**DNS**] > [レコード] に移動します。
2. 詳細ペインで、「プロキシレコードをフラッシュ」をクリックします。

3.「フラッシュ・タイプ」ボックスで、「ネガティブ・レコード」を選択します。
4.「ネガティブ・レコード・タイプ」ボックスで、「NXDOMAIN」または「**NODATA**」を選択します。

ランダムなサブドメイン攻撃と **NXDOMAIN** 攻撃からの保護

ランダムなサブドメイン攻撃や NXDOMAIN 攻撃を防ぐために、DNS キャッシュメモリを制限したり、負のレコードの TTL 値を調整したりできます。

DNS キャッシュが消費するメモリ量を制限するには、最大キャッシュサイズ (MB 単位) と、否定応答を保存するためのキャッシュサイズ (MB 単位) を指定します。いずれかの制限に達すると、キャッシュにエントリは追加されません。また、Syslog メッセージがログに記録され、SNMP トラップを設定している場合は、SNMP トラップが生成されます。これらの制限が設定されていない場合、キャッシュはシステムメモリを使い果たすまで続きます。

負のレコードの TTL 値を大きくすると、価値のないレコードが長期間保存される可能性があります。TTL 値が小さいほど、バックエンドサーバーに送信されるリクエストが増えます。

ネガティブレコードの TTL は、TTL 値または SOA レコードの「Expires」値のどちらか小さい方の値に設定されます。

注記:

- この制限はパケットエンジンごとに追加されます。たとえば、MaxCacheSize が 5 MB に設定されていて、アプライアンスにパケットエンジンが 3 つある場合、合計キャッシュサイズは 15 MB になります。
- 負のレコードのキャッシュサイズは、最大キャッシュサイズ以下でなければなりません。
- DNS キャッシュメモリの制限を、すでにキャッシュされているデータ量よりも低い値に減らしても、データが期限切れになるまでキャッシュサイズは制限を超えるままになります。つまり、TTL0 を超えるか、フラッシュされます（`flush dns proxyrecords` コマンド、または NetScaler ADC GUI のプロキシレコードのフラッシュ）。
- SNMP トラップを構成するには、「SNMP トラップを生成するように NetScaler を構成する」を参照してください。

CLI を使用して **DNS** キャッシュによって消費されるメモリを制限する

コマンドプロンプトで入力します。

```
set dns parameter -maxCacheSize <MBytes> -maxNegativeCacheSize <MBytes>
```

例:

```
set dns parameter - maxCacheSize 100 -maxNegativeCacheSize 25
```


GUI を使用して **DNS** キャッシュが消費するメモリを制限する

[構成] > [トラフィック管理] > [**DNS**] に移動し、[**DNS** 設定の変更] をクリックして、次のパラメータを設定します。

- 最大キャッシュサイズ (MB)
- 負の最大キャッシュサイズ (MB)

CLI を使用してネガティブレコードの **TTL** を制限する

コマンドプロンプトで入力します。

```
set dns parameter -maxnegcacheTTL <secs>
```

例**:**

```
set dns parameter -maxnegcacheTTL 360
```

GUI を使用してネガティブレコードの **TTL** を制限する

1. [設定] > [トラフィック管理] > [**DNS**] に移動します。
2.「**DNS** 設定の変更」をクリックし、「最大ネガティブキャッシュ **TTL (**秒**)**」パラメータを設定します。

DNS レコードをキャッシュに保存

攻撃により DNS キャッシュに重要でないエントリが殺到する可能性がありますが、すでにキャッシュされている正当なレコードがフラッシュされ、新しいエントリ用のスペースが確保される可能性があります。攻撃によってキャッシュが無効なデータでいっぱいになるのを防ぐため、TTL 値を超えた後でも正規のレコードを保持できます。

CacheNoExpire パラメーターを有効にすると、パラメーターを無効にするまで、現在キャッシュにあるレコードが保持されます。

注記:

- このオプションは、最大キャッシュサイズ (maxCacheSize パラメーター) が指定されている場合にのみ使用できます。
- MaxNegCachetTL が設定されていて cachenoExpire が有効になっている場合は、CachenoExpire が優先されます。

CLI を使用して **DNS** レコードをキャッシュに保持する

コマンドプロンプトで入力します。

```
set dns parameter -cacheNoExpire ( ENABLED | DISABLED)
```


例:

```
set dns parameter -cacheNoExpire ENABLED
```

GUI を使用して DNS レコードをキャッシュに保持する

1. [構成] > [トラフィック管理] > [**DNS**] に移動し、[**DNS** 設定の変更] をクリックします。
2.「キャッシュ期限なし」を選択します。

DNS キャッシュバイパスを有効にする

DNS リクエストの可視性と制御性を高めるには、CacheHitBypass パラメーターを設定して、すべてのリクエストをバックエンドサーバーに転送し、キャッシュを構築しても使用できないようにします。キャッシュが作成されたら、パラメータを無効にして、リクエストがキャッシュから処理されるようにすることができます。

CLI を使用して DNS キャッシュバイパスを有効にする

コマンドプロンプトで入力します。

```
set dns parameter -cacheHitBypass ( ENABLED | DISABLED )
```

例:

```
set dns parameter -cacheHitBypass ENABLED
```

GUI を使用して DNS キャッシュバイパスを有効にする

1. [構成] > [トラフィック管理] > [**DNS**] に移動し、[**DNS** 設定の変更] をクリックします。
2.「キャッシュヒットバイパス」を選択します。

Slowloris 攻撃を防ぐ

複数のパケットにまたがる DNS クエリは、Slowloris 攻撃の脅威となる可能性があります。NetScaler アプライアンスは、複数のパケットに分割された DNS クエリをサイレントドロップできます。

クエリが複数のパケットに分割されている場合は、splitPktQueryProcessing パラメータを DNS クエリを ALLOW または DROP に設定できます。

注: この設定は DNS TCP にのみ適用されます。

CLI を使用して **DNS** クエリを **1** つのパケットに制限する

コマンドプロンプトで入力します。

```
set dns parameter -splitPktQueryProcessing ( ALLOW | DROP )
```

例**:**

```
set dns parameter -splitPktQueryProcessing DROP
```

GUI を使用して **DNS** クエリを **1** つのパケットに制限する

1. [構成] > [トラフィック管理] > [**DNS**] に移動し、[**DNS** 設定の変更] をクリックします。
2. 「パケット分割クエリ処理」ボックスで、「許可」または「ドロップ」を選択します。

キャッシュから提供された **DNS** 応答の統計を収集します

キャッシュから提供された DNS 応答の統計を収集できます。次に、これらの統計を使用してしきい値を設定し、それを超えるとさらに多くの DNS トラフィックがドロップされ、帯域幅ベースのポリシーでこのしきい値が適用されます。以前は、キャッシュから処理された要求の数が報告されなかったため、DNS 負荷分散仮想サーバーの帯域幅計算は正確ではありませんでした。

プロキシモードでは、要求バイト数、応答バイト数、受信パケット総数、送信パケット総数の統計が継続的に更新されます。以前は、特に DNS 負荷分散仮想サーバーの場合、これらの統計が常に更新されるわけではありませんでした。

プロキシモードでは、キャッシュから提供される DNS レスポンスの数も決定できるようになりました。これらの統計を収集するために、`stat lb vserver <DNSvirtualServerName>` 以下のオプションがコマンドに追加されました。

- リクエスト—DNS または DNS_TCP 仮想サーバーが受信したリクエストの総数。バックエンドに転送されたリクエストとキャッシュから応答されたリクエストが含まれます。
- 仮想サーバーのヒット数—バックエンドに転送されたリクエストの総数。キャッシュから処理される要求の数は、要求の総数と仮想サーバーから処理された要求数の差です。
- レスポンス—この仮想サーバーから送信されたレスポンスの総数。たとえば、DNS LB 仮想サーバーが 5 つの DNS 要求を受信し、そのうちの 3 つをバックエンドに転送し、そのうちの 2 つをキャッシュから処理した場合、これらの各統計の対応する値は次のようになります。
  - 仮想サーバーヒット数:3
  - リクエスト:5
  - レスポンス:5

ファイアウォールの負荷分散

August 15, 2023

ファイアウォールの負荷分散は、トラフィックを複数のファイアウォールに分散し、耐障害性とスループットを向上させます。ファイアウォールの負荷分散は、次の方法でネットワークを保護します。

- ファイアウォール間で負荷を分散することで、単一障害点を排除し、ネットワークを拡張できます。
- 高可用性の向上。

NetScaler アプライアンスのファイアウォール負荷分散の構成は、負荷分散の構成と似ていますが、推奨されるサービスタイプが ANY、推奨モニタータイプが PING、負荷分散仮想サーバーモードが MAC に設定されている点が異なります。

ファイアウォールの負荷分散は、サンドイッチ環境、エンタープライズ環境、または複数のファイアウォール環境構成で設定できます。サンドイッチ環境は、外部からネットワークに入るトラフィックとネットワークから出てインターネットに向かうトラフィックの負荷分散に使用されます。この環境では、一連のファイアウォールの両側に 1 つずつ、合計 2 つの NetScaler アプライアンスを構成する必要があります。ネットワークからインターネットに向かうトラフィックの負荷分散を行うためのエンタープライズ環境を設定します。エンタープライズ環境では、内部ネットワークとインターネットへのアクセスを提供するファイアウォールの間に単一の NetScaler アプライアンスを構成する必要があります。マルチファイアウォール環境は、別のファイアウォールからのトラフィックの負荷分散に使用されます。NetScaler アプライアンスの両側でファイアウォールの負荷分散を有効にすると、出力方向と入力方向の両方のトラフィックフローが改善され、トラフィックの処理が速くなります。マルチファイアウォール環境では、2 つのファイアウォールの間に挟まれた NetScaler アプライアンスを構成する必要があります。

重要: NetScaler アプライアンスで宛先 IP アドレス用の静的ルートを構成し、L3 モードを有効にすると、NetScaler アプライアンスはトラフィックを負荷分散仮想サーバーに送信するのではなく、ルーティングテーブルを使用してトラフィックをルーティングします。

注: FTP が機能するには、NetScaler アプライアンスで IP アドレスとポートをそれぞれ * と 21 に設定し、サービスタイプを FTP として指定する必要があります。この場合、NetScaler アプライアンスは、FTP 制御接続を受け入れ、ペイロードを変更し、データ接続を管理することで、すべて同じファイアウォールを介して FTP プロトコルを管理します。

ファイアウォールの負荷分散は、NetScaler アプライアンスでサポートされている一部の負荷分散方法のみをサポートします。また、設定できるパーシスタンスとモニターは数種類だけです。

ファイアウォールの負荷分散方法

ファイアウォールの負荷分散では、次の負荷分散方法がサポートされています。

- 最小接続数
- ラウンドロビン

- 最小パケット
- 最小帯域幅
- 送信元 IP ハッシュ
- 宛先 IP ハッシュ
- 送信元 IP 宛先 IP ハッシュ
- 送信元 IP 送信元ポートハッシュ
- 最小応答時間法 (LRTM)
- カスタムロード

ファイアウォール・パーシステンス

ファイアウォールのロードバランシングでは、SOURCEIP、DESTIP、および SOURCEIPDESTIP ベースのパーシスタンスのみがサポートされています。

ファイアウォール・サーバー・モニタリング

ファイアウォールのロードバランシングでは、PING とトランスペアレントモニターのみがサポートされます。PING モニター (デフォルト) をファイアウォールを表すバックエンドサービスにバインドできます。ファイアウォールが ping パケットに応答しないように設定されている場合は、個々のファイアウォールを介して信頼できる側のホストを監視するようにトランスペアレントモニターを設定できます。

サンドイッチ環境

August 15, 2023

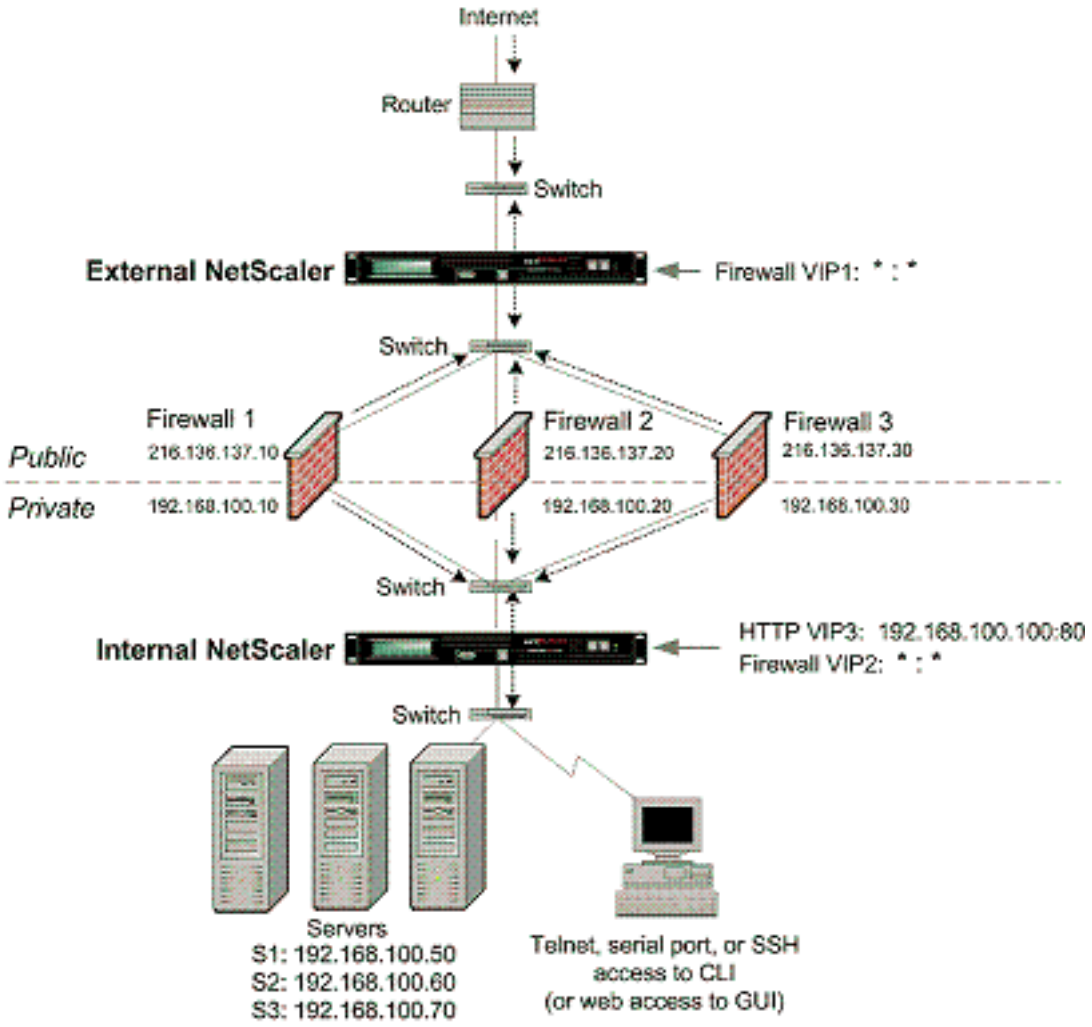
サンドイッチモードの NetScaler 展開では、ファイアウォールを介してネットワークトラフィックの負荷を分散できます。つまり、入力方向（インターネットなどの外部からネットワークに入るトラフィック）と下り（ネットワークから出てインターネットに向かうトラフィック）です。

この設定では、NetScaler は一連のファイアウォールの両側に配置されます。ファイアウォールとインターネットの間に配置された NetScaler は、入力トラフィックを処理する外部 NetScaler と呼ばれ、構成された方法に基づいて最適なファイアウォールを選択します。ファイアウォールとプライベートネットワークの間にある NetScaler は、内部 NetScaler と呼ばれ、セッションの最初のパケットを受信するファイアウォールを追跡します。次に、そのセッションの後続のパケットがすべて同じファイアウォールに送信されるようにします。

内部 NetScaler を通常のトラフィックマネージャーとして構成して、プライベートネットワークサーバー全体のトラフィックの負荷分散を行うことができます。この設定では、プライベートネットワーク（出力）から発信されるトラフィックをファイアウォール全体で負荷分散することもできます。

次の図は、サンドイッチファイアウォールの負荷分散環境を示しています。

図 1: ファイアウォール負荷分散 (サンドイッチ)



サービスタイプ ANY は、すべてのトラフィックを受け入れるように NetScaler ADC を構成します。

HTTP と TCP に関連する利点を利用するには、サービスと仮想サーバーを HTTP または TCP タイプに設定します。FTP が機能するには、FTP タイプでサービスを構成します。

サンドイッチ環境での外部 **NetScaler** の構成

サンドイッチ環境で外部 NetScaler を構成するには、次のタスクを実行します。

- 負荷分散機能を有効にします。
- ファイアウォールごとにワイルドカードサービスを構成します。
- ワイルドカードサービスごとにモニターを設定します。
- インターネットからのトラフィック用にワイルドカード仮想サーバーを設定します。
- 仮想サーバーを MAC 書き換えモードに設定します。
- サービスをワイルドカード仮想サーバーにバインドします。

- 設定を保存して検証します。

負荷分散機能を有効にする

コマンドラインインターフェイスを使用して負荷分散を有効にするには　コマンドプロンプトで次のコマンドを入力して負荷分散を有効にし、構成を確認します。

```
enable ns feature LB
show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   OFF
 2)     Surge Protection                SP                   ON
 3)     Load Balancing                  LB                   ON
 .
 .
 .
 24)    NetScaler Push                  push                 OFF
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して負荷分散を有効にするには　[システム] > [設定] に移動し、[基本機能の設定] で [負荷分散] を選択します。

ファイアウォールごとにワイルドカードサービスを構成する

コマンドラインインターフェイスを使用して各ファイアウォールにワイルドカードサービスを構成するには　コマンドプロンプトで入力します。

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

例:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```


構成ユーティリティを使用してファイアウォールごとにワイルドカードサービスを構成するには　[トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、サービスを追加します。[プロトコル] フィールドに ****ANY**** を指定し、[ポート] フィールドに [*] を指定します。

ワイルドカードサービスごとにモニターを設定する

PING モニターは、デフォルトではサービスにバインドされています。個々のファイアウォールを介して信頼できる側のホストを監視するには、トランスペアレントモニターを設定する必要があります。その後、透過モニターをサービスにバインドできます。デフォルトの PING モニターは、NetScaler ADC アプライアンスと上流デバイス間の接続のみを監視します。透過モニタは、アプライアンスからモニタで指定された宛先 IP アドレスを所有するデバイスまでのパスに存在するすべてのデバイスを監視します。透過モニタが設定されておらず、ファイアウォールのステータスが UP であるが、そのファイアウォールのネクストホップデバイスの 1 つがダウンしている場合、アプライアンスは負荷分散の実行中にファイアウォールを組み込み、パケットをファイアウォールに転送します。ただし、ネクストホップデバイスの 1 つがダウンしているため、パケットは最終的な宛先に配信されません。透過モニターをバインドすると、デバイス（ファイアウォールを含む）のいずれかがダウンしている場合、サービスはダウンとしてマークされ、アプライアンスがファイアウォール負荷分散を実行するときにファイアウォールは含まれません。

トランスペアレントモニターをバインドすると、PING モニターがオーバーライドされます。透過モニタに加えて PING モニタを設定するには、透過モニタを作成してバインドした後、PING モニタをサービスにバインドする必要があります。

コマンドラインインターフェイスを使用して透過モニタを構成するには　コマンドプロンプトで、次のコマンドを入力して透過モニタを構成し、構成を確認します。

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

例**:**

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
To bind a PING monitor, type the following command:
bind monitor PING fw-svc1
<!--NeedCopy-->
```

構成ユーティリティを使用して透過モニターを作成してバインドするには　[トラフィック管理] **>** [負荷分散] **>** [モニター] に移動し、トランスペアレントモニターを作成してバインドします。

インターネットからのトラフィック用にワイルドカード仮想サーバーを設定

コマンドラインインターフェイスを使用してインターネットからのトラフィック用にワイルドカード仮想サーバーを構成するには　コマンドプロンプトで入力します。

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

例**:**

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

構成ユーティリティを使用してインターネットからのトラフィック用にワイルドカード仮想サーバーを構成するには [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、ワイルドカード仮想サーバーを作成します。[プロトコル**]** フィールドに ****ANY**** を指定し、[ポート] フィールドに [*] を指定します。

仮想サーバーを **MAC** 書き換えモードで構成する

コマンドラインインターフェイスを使用して仮想サーバーを **MAC** 書き換えモードに設定するには　コマンドプロンプトで入力します。

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーを **MAC** 書き換えモードで構成するには

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、リダイレクトモードを設定する仮想サーバー (たとえば、VServer-LB-1) を選択します。
2. 「基本設定」セクションを編集し、「詳細」をクリックします。
3. 「リダイレクションモード」ドロップダウンリストから、「**MAC** ベース」を選択します。

サービスをワイルドカード仮想サーバーにバインドする

コマンドラインインターフェイスを使用してサービスをワイルドカード仮想サーバーにバインドするには　コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```


例:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスをワイルドカード仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、サービスをバインドする仮想サーバーを選択します。
2. サービスセクションをクリックして、バインドするサービスを選択します。

設定を保存して検証する

構成タスクが完了したら、必ず構成を保存してください。設定が正しいことを確認してください。

コマンドラインインターフェイスを使用して構成を保存して確認するには　コマンドプロンプトで、次のコマンドを入力して透過モニタを構成し、構成を確認します。

```
save ns config
show vserver
<!--NeedCopy-->
```

例:

```
save config
sh lb vserver FWLBVIP1
FWLBVIP1 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 06:40:14 2010
        Time since last state change: 0 days, 00:00:11.240
        Effective State: UP  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: SRCIPDESTIPHASH
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw_svc_1 (10.102.29.251: *) - ANY State: UP  Weight: 1
2) fw_svc_2 (10.102.29.18: \*) - ANY State: UP   Weight: 1
 Done
show service fw-svc1
        fw-svc1 (10.102.29.251:\*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 10:04:50 2010
        Time since last state change: 0 days, 00:00:38.120
```



```
25          Server Name: 10.102.29.251
26          Server ID : 0   Monitor Threshold : 0
27          Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
28          Use Source IP: NO
29          Client Keepalive(CKA): NO
30          Access Down Service: NO
31          TCP Buffering(TCPB): YES
32          HTTP Compression(CMP): NO
33          Idle timeout: Client: 120 sec   Server: 120 sec
34          Client IP: DISABLED
35          Cacheable: NO
36          SC: OFF
37          SP: OFF
38          Down state flush: ENABLED
39
40  1)      Monitor Name: monitor-HTTP-1
41                  State: DOWN     Weight: 1
42                  Probes: 5       Failed [Total: 5 Current: 5]
43                  Last response: Failure - Time out during TCP connection
                        establishment stage
44                  Response Time: 2000.0 millisec
45  2)      Monitor Name: ping
46                  State: UP       Weight: 1
47                  Probes: 3       Failed [Total: 0 Current: 0]
48                  Last response: Success - ICMP echo reply received.
49                  Response Time: 1.415 millisec
50   Done
51  <!--NeedCopy-->
```

サンドイッチ環境での内部 **NetScaler** の構成

サンドイッチ環境で内部 NetScaler を構成するには、次のタスクを実行します。

サーバーからのトラフィック (下り)

- 負荷分散機能を有効にします。
- ファイアウォールごとにワイルドカードサービスを構成します。
- ワイルドカードサービスごとにモニターを設定します。
- ワイルドカード仮想サーバーを構成して、ファイアウォールに送信されるトラフィックの負荷を分散します。
- 仮想サーバーを MAC 書き換えモードに設定します。
- ファイアウォールサービスをワイルドカード仮想サーバーにバインドします。

プライベートネットワークサーバー間のトラフィック

- 仮想サーバーごとにサービスを構成します。
- サービスごとにモニターを設定します。
- HTTP 仮想サーバーを構成して、サーバーに送信されるトラフィックのバランスを取ります。
- HTTP サービスを HTTP 仮想サーバーにバインドします。
- 設定を保存して検証します。

負荷分散機能を有効にする

負荷分散機能が無効になっている場合は、サービスや仮想サーバーなどの負荷分散エンティティを構成できます。ただし、この機能を有効にするまでは機能しません。

コマンドラインインターフェイスを使用して負荷分散を有効にするには　コマンドプロンプトで次のコマンドを入力して負荷分散を有効にし、構成を確認します。

```
enable ns feature LB
show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して負荷分散を有効にするには　[システム] > [設定] に移動し、[基本機能の設定] で [負荷分散] を選択します。

ファイアウォールごとにワイルドカードサービスを構成する

コマンドラインインターフェイスを使用して各ファイアウォールにワイルドカードサービスを構成するには　コマンドプロンプトで入力します。

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

例:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```


構成ユーティリティを使用してファイアウォールごとにワイルドカードサービスを構成するには　[トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、サービスを追加します。[プロトコル] フィールドに ****ANY**** を指定し、[ポート] フィールドに [*] を指定します。

ワイルドカードサービスごとにモニターを設定する

PING モニターは、デフォルトではサービスにバインドされています。個々のファイアウォールを介して信頼できる側のホストを監視するには、トランスペアレントモニターを設定する必要があります。その後、透過モニターをサービスにバインドできます。デフォルトの PING モニターは、NetScaler ADC アプライアンスと上流デバイス間の接続のみを監視します。透過モニタは、アプライアンスからモニタで指定された宛先 IP アドレスを所有するデバイスまでのパスに存在するすべてのデバイスを監視します。透過モニタが設定されておらず、ファイアウォールのステータスが UP であるが、そのファイアウォールのネクストホップデバイスの 1 つがダウンしている場合、アプライアンスは負荷分散の実行中にファイアウォールを組み込み、パケットをファイアウォールに転送します。ただし、ネクストホップデバイスの 1 つがダウンしているため、パケットは最終的な宛先に配信されません。透過モニターをバインドすると、デバイス（ファイアウォールを含む）のいずれかがダウンしている場合、サービスはダウンとしてマークされ、アプライアンスがファイアウォール負荷分散を実行するときにファイアウォールは含まれません。

トランスペアレントモニターをバインドすると、PING モニターがオーバーライドされます。透過モニタに加えて PING モニタを設定するには、透過モニタを作成してバインドした後、PING モニタをサービスにバインドする必要があります。

コマンドラインインターフェイスを使用して透過モニタを構成するには　コマンドプロンプトで、次のコマンドを入力して透過モニタを構成し、構成を確認します。

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

例**:**

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

構成ユーティリティを使用して透過モニターを作成してバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [モニター] に移動し、モニターを作成します。
2. 「モニターの作成」ダイアログ・ボックスで、必要なパラメーターを入力し、「透明」を選択します。

ワイルドカード仮想サーバーを構成して、ファイアウォールに送信されるトラフィックの負荷を分散します

コマンドラインインターフェイスを使用してファイアウォールに送信されるトラフィックの負荷を分散するようにワイルドカード仮想サーバーを構成するには　コマンドプロンプトで入力します。

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

例**:**

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

構成ユーティリティを使用してインターネットからのトラフィック用にワイルドカード仮想サーバーを構成するには

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、ワイルドカード仮想サーバーを作成します。
2. [プロトコル] フィールドに **ANY** を指定し、[ポート] フィールドに ***** を指定します。

構成ユーティリティを使用してファイアウォールに送信されるトラフィックの負荷を分散するようにワイルドカード仮想サーバーを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次のように次のパラメータの値を指定します。
    - 名前—名前
4. [プロトコル] で [任意] を選択し、[IP アドレスとポート] で [*] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成した仮想サーバーが [負荷分散仮想サーバー] ペインに表示されます。

仮想サーバーを **MAC** 書き換えモードで構成する

コマンドラインインターフェイスを使用して仮想サーバーを **MAC** 書き換えモードに設定するには　コマンドプロンプトで入力します。

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```


構成ユーティリティを使用して仮想サーバーを **MAC** 書き換えモードで構成するには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、リダイレクトモードを設定する仮想サーバー (たとえば、VServer-LB-1) を選択します。
2. 「基本設定」セクションを編集し、「詳細」をクリックします。
3. 「リダイレクションモード」ドロップダウンリストから、「**MAC** ベース」を選択します。

ファイアウォールサービスをワイルドカード仮想サーバーにバインドする

コマンドラインインターフェイスを使用してファイアウォールサービスをワイルドカード仮想サーバーにバインドするには　コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してファイアウォールサービスをワイルドカード仮想サーバーにバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを選択します。
2. サービスセクションをクリックして、バインドするサービスを選択します。

注:1 つのサービスを複数の仮想サーバーにバインドできます。

仮想サーバーごとにサービスを構成する

コマンドラインインターフェイスを使用して各仮想サーバーのサービスを構成するには　コマンドプロンプトで入力します。

```
add service <name> <serverName> HTTP <port>
<!--NeedCopy-->
```

例**:**

```
add service Service-HTTP-1 10.102.29.5 HTTP 80
<!--NeedCopy-->
```

構成ユーティリティを使用して各仮想サーバーのサービスを構成するには

1. [トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、仮想サーバーごとにサービスを設定します。
2. 「プロトコル」フィールドに「**HTTP** 」を指定し、「使用可能なモニター」で「**HTTP**」を選択します。

構成ユーティリティを使用して各仮想サーバーのサービスを構成するには

1. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [Create Service] ダイアログボックスで、次に示すように次のパラメータの値を指定します。
   - サービス名—名前
   - サーバー—サーバー名
   - ポート-ポート
4. [プロトコル] で、HTTP を指定します。[使用可能なモニター] で [HTTP] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成したサービスが [サービス] ペインに表示されます。

サービスごとにモニターを設定する

コマンドラインインターフェイスを使用してモニターをサービスにバインドするには　コマンドプロンプトで入力します。

```
bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

例:

```
bind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してモニターをサービスにバインドするには　[トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスをダブルクリックしてモニターを追加します。

サーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成する

コマンドラインインターフェイスを使用してサーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成するには　コマンドプロンプトで入力します。

```
add lb vserver <name> HTTP <ip> <port>
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```


構成ユーティリティを使用してサーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成するには

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サービス] に移動し、HTTP 仮想サーバーを設定します。
2.「プロトコル」フィールドに「**HTTP** 」を指定します。

構成ユーティリティを使用してサーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次のように次のパラメータの値を指定します。
   - 名前—名前
   - **IP アドレス—IP アドレス注: 仮想サーバーが IPv6 を使用している場合は、IPv6 チェックボックスを選択し、IPv6 形式でアドレスを入力します (たとえば、1000:0000:00:00:0005:0600:700 a: 888b)。**
   - ポート-ポート
4. [プロトコル] で [HTTP] を選択します
5. [Create] をクリックしてから、[Close] をクリックします。作成した仮想サーバーが [負荷分散仮想サーバー] ペインに表示されます。

設定を保存して検証する

構成タスクが完了したら、必ず構成を保存してください。また、設定が正しいことを確認する必要があります。

コマンドラインインターフェイスを使用して構成を保存して確認するには　コマンドプロンプトで、次のコマンドを入力して透過モニタを構成し、構成を確認します。

- `save ns config`
- `show vserver`

例**:**

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
```



```
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: \*) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:\*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して構成を保存して検証するには

1. 詳細ペインで、「保存」をクリックします。
2. 「設定を保存」ダイアログで、「はい」をクリックします。
3. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
4. 詳細ペインで、手順 5 で作成した仮想サーバーを選択します。

5. 詳細ペインに表示されている設定が正しいことを確認します。
6. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。
7. 詳細ペインで、ステップ 5 で作成したサービスを選択します。
8. 詳細ペインに表示されている設定が正しいことを確認します。

サンドイッチ環境におけるファイアウォール・ロード・バランシング・セットアップの監視

構成が起動して実行されたら、各サービスと仮想サーバーの統計情報を表示して、考えられる問題がないかどうかを確認する必要があります。

仮想サーバーの統計情報を表示する

仮想サーバーのパフォーマンスを評価したり、問題をトラブルシューティングしたりするために、NetScaler ADC アプライアンスに構成されている仮想サーバーの詳細を表示できます。すべての仮想サーバーの統計情報の概要を表示することも、仮想サーバーの名前を指定して、その仮想サーバーの統計情報のみを表示することもできます。次の詳細を表示できます。

- 名前
- IP アドレス
- ポート
- プロトコル
- 仮想サーバーの状態
- 受け取ったリクエストの割合
- ヒット率

コマンドラインインターフェイスを使用して仮想サーバーの統計情報を表示するには　NetScaler で現在構成されているすべての仮想サーバー、または 1 つの仮想サーバーの統計の概要を表示するには、コマンドプロンプトで次のように入力します。

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

例**:**

```
>stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
          0/s
Two                          *     0          TCP         DOWN      0/s
          0/s
```



```
Three                          *  2598           TCP          DOWN       0/s
          0/s
dnsVirtualNS     10.102.29.90     53           DNS          DOWN       0/s
          0/s
BRVSERV             10.10.1.1     80          HTTP          DOWN       0/s
          0/s
LBVIP            10.102.29.66     80          HTTP            UP       0/s
          0/s
 Done

<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーの統計情報を表示するには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] **>** [統計] に移動します。
2. 1 つの仮想サーバーの統計のみを表示する場合は、詳細ペインで仮想サーバーを選択し、[統計] をクリックします。

サービスの統計情報の表示

サービス統計を使用して、要求、応答、要求バイト、応答バイト、現在のクライアント接続、サージキュー内の要求、現在のサーバ接続などのレートを表示できます。

コマンドラインインターフェイスを使用してサービスの統計を表示するには　コマンドプロンプトで入力します。

```
stat service <name>
<!--NeedCopy-->
```

例**:**

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスの統計を表示するには

1. [トラフィック管理] **>** [負荷分散] **>** [サービス] **>** [統計] に移動します。
2. 1 つのサービスの統計のみを表示する場合は、サービスを選択して [**Statistics**] をクリックします。

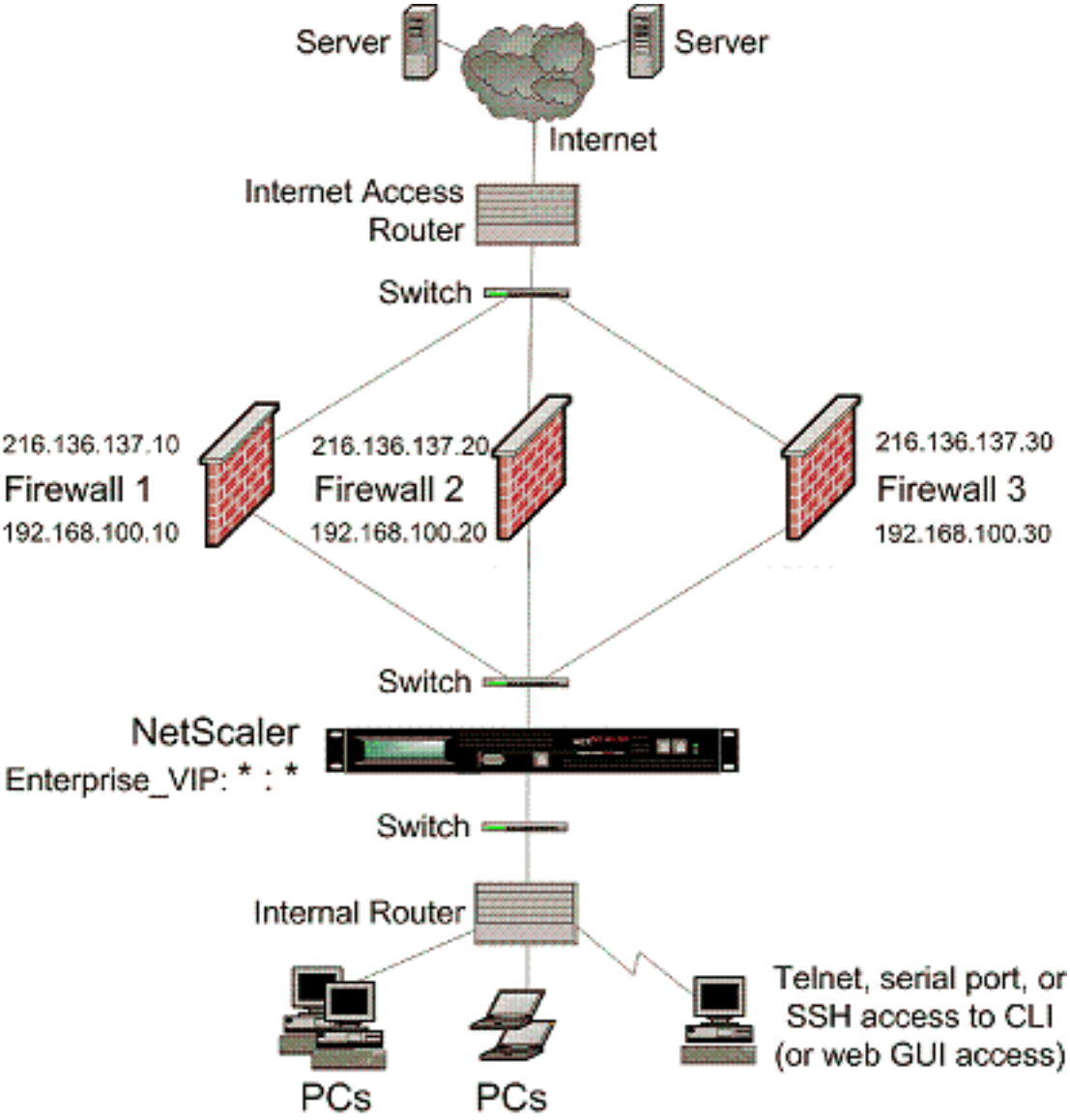
エンタープライズ環境

August 15, 2023

エンタープライズ設定では、NetScaler ADC はパブリックインターネットと内部プライベートネットワークに接続するファイアウォールの間に配置され、下りトラフィックを処理します。NetScaler ADC は、構成された負荷分散ポリシーに基づいて最適なファイアウォールを選択します。

次の図は、エンタープライズファイアウォールの負荷分散環境を示しています。

図 1: ファイアウォール負荷分散 (エンタープライズ)



サービスタイプ ANY は、すべてのトラフィックを受け入れるように NetScaler ADC を構成します。

HTTP と TCP に関連する利点を活用するには、サービスと vserver を HTTP または TCP タイプで構成します。FTP が機能するには、FTP タイプでサービスを構成します。

エンタープライズ環境での **NetScaler ADC** 構成

エンタープライズ環境で NetScaler ADC を構成するには、次のタスクを実行します。

サーバーからのトラフィック (下り)

- 負荷分散機能を有効にします。
- ファイアウォールごとにワイルドカードサービスを構成します。
- ワイルドカードサービスごとにモニターを設定します。
- ワイルドカード仮想サーバーを構成して、ファイアウォールに送信されるトラフィックの負荷を分散します。
- 仮想サーバーを MAC 書き換えモードに設定します。
- ファイアウォールサービスをワイルドカード仮想サーバーにバインドします。

プライベートネットワークサーバー間のトラフィック

- 仮想サーバーごとにサービスを構成します。
- サービスごとにモニターを設定します。
- HTTP 仮想サーバーを構成して、サーバーに送信されるトラフィックのバランスを取ります。
- HTTP サービスを HTTP 仮想サーバーにバインドします。
- 設定を保存して検証します。

以下の構成例では、ファイアウォールサーバーの 1 つがネットワークトポロジ図に示されています（図 1）。

負荷分散機能を有効にする

負荷分散機能が無効になっている場合は、サービスや仮想サーバーなどの負荷分散エンティティを構成できますが、機能を有効にするまで機能しません。

コマンドラインインターフェイスを使用して負荷分散を有効にするには　コマンドプロンプトで次のコマンドを入力して負荷分散を有効にし、構成を確認します。

- ns 機能 LB を有効にする
- show ns feature

例**:**

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
```



```
 .
 .
 .
 24)    NetScaler Push                  push                    OFF
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して負荷分散を有効にするには　[システム] > [設定] に移動し、[基本機能の設定] で [負荷分散] を選択します。

ファイアウォールごとにワイルドカードサービスを構成する

コマンドラインインターフェイスを使用して各ファイアウォールにワイルドカードサービスを構成するには　コマンドプロンプトで入力します。

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

例:

```
add service Service-HTTP-1 192.168.100.10 ANY *
<!--NeedCopy-->
```

構成ユーティリティを使用してファイアウォールごとにワイルドカードサービスを構成するには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [Create Service] ダイアログボックスで、次に示すように次のパラメータの値を指定します。
   - サービス名—名前
   - サーバー—サーバー名
4. [プロトコル] で [任意] を選択し、[ポート] で [*] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成したサービスが [サービス] ペインに表示されます。

ワイルドカードサービスごとにモニターを設定する

PING モニターは、デフォルトではサービスにバインドされています。個々のファイアウォールを介して信頼側のホストを監視するには、透過的なモニターを設定する必要があります。その後、透過モニターをサービスにバインドできます。デフォルトの PING モニターは、NetScaler ADC アプライアンスと上流デバイス間の接続のみを監視します。透過モニタは、アプライアンスからモニタで指定された宛先 IP アドレスを所有するデバイスまでのパスに存在するすべてのデバイスを監視します。透過モニタが設定されておらず、ファイアウォールのステータスが UP であるが、

そのファイアウォールのネクストホップデバイスの 1 つがダウンしている場合、アプライアンスは負荷分散の実行中にファイアウォールを組み込み、パケットをファイアウォールに転送します。ただし、ネクストホップデバイスの 1 つがダウンしているため、パケットは最終的な宛先に配信されません。透過モニターをバインドすると、デバイス（ファイアウォールを含む）のいずれかがダウンしている場合、サービスはダウンとしてマークされ、アプライアンスがファイアウォール負荷分散を実行するときにファイアウォールは含まれません。

透過モニターをバインドすると、PING モニターがオーバーライドされます。透過モニタに加えて PING モニタを設定するには、透過モニタを作成してバインドした後、PING モニタをサービスにバインドする必要があります。

コマンドラインインターフェイスを使用して透過モニタを構成するには　コマンドプロンプトで、次のコマンドを入力して透過モニタを構成し、構成を確認します。

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
   transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

例**:**

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -destport 80 -
   transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

構成ユーティリティを使用して透過モニターを作成してバインドするには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [モニターの作成] ダイアログボックスで、次のように値を指定します。
   - 名前 *
   - type*—type
   - 接続先 IP
   - 透明

   -* 必須パラメータ
4. [Create] をクリックしてから、[Close] をクリックします。[モニター] ペインで、構成したモニターを選択し、画面の下部に表示される設定が正しいことを確認します。

ワイルドカード仮想サーバーを構成して、ファイアウォールに送信されるトラフィックの負荷を分散します

ファイアウォールを通過するトラフィックは、ファイアウォールの背後に配置されたさまざまなプロキシまたはサーバーを対象としています。これらのプロキシまたはサーバーは、異なる IP アドレスとポートを持つことができます。

トラフィックがファイアウォールを透過的に通過するには、ファイアウォールを負荷分散する仮想サーバの IP アドレスとポートを * に設定して、任意の IP アドレスとポートのトラフィックを受け入れる必要があります。

コマンドラインインターフェイスを使用してファイアウォールに送信されるトラフィックの負荷を分散するようにワイルドカード仮想サーバーを構成するには　コマンドプロンプトで入力します。

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

構成ユーティリティを使用してファイアウォールに送信されるトラフィックの負荷を分散するようにワイルドカード仮想サーバーを構成するには

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ペインで、［Add］をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次のように次のパラメータの値を指定します。
    - 名前—名前
4. [プロトコル] で [任意] を選択し、[IP アドレスとポート] で [*] を選択します。
5. ［Create］をクリックしてから、［Close］をクリックします。作成した仮想サーバーが [負荷分散仮想サーバー] ペインに表示されます。

仮想サーバーを **MAC** 書き換えモードで構成する

コマンドラインインターフェイスを使用して仮想サーバーを **MAC** 書き換えモードに設定するには　コマンドプロンプトで入力します。

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーを **MAC** 書き換えモードで構成するには

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。

2. 詳細ウィンドウで、リダイレクトモードを構成する仮想サーバー (Vserver-LB-1 など) を選択し、[開く] をクリックします。
3. [詳細設定] タブの [リダイレクトモード] で、[Mac ベース] をクリックします。
4. 「OK」をクリックします。

ファイアウォールサービスをワイルドカード仮想サーバーにバインドする

コマンドラインインターフェイスを使用してファイアウォールサービスをワイルドカード仮想サーバーにバインドするには　コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してファイアウォールサービスをワイルドカード仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. サービスセクションをクリックして、バインドするサービスを選択します。

注:1 つのサービスを複数の仮想サーバーにバインドできます。

仮想サーバーごとにサービスを構成する

コマンドラインインターフェイスを使用して各仮想サーバーのサービスを構成するには　コマンドプロンプトで入力します。

```
add service <name> <serverName> HTTP <port>
<!--NeedCopy-->
```

例:

```
add service Service-HTTP-1 192.168.100.10 HTTP 80
<!--NeedCopy-->
```

構成ユーティリティを使用して各仮想サーバーのサービスを構成するには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [Create Service] ダイアログボックスで、次に示すように次のパラメータの値を指定します。

- サービス名—名前
- サーバー—サーバー名
- ポート-ポート

4. [プロトコル] で、HTTP を指定します。[使用可能なモニター] で [HTTP] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成したサービスが [サービス] ペインに表示されます。

サービスごとにモニターを設定する

コマンドラインインターフェイスを使用してモニターをサービスにバインドするには　コマンドプロンプトで入力します。

```
bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

例**:**

```
bind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してモニターをサービスにバインドするには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. サービスを開き、モニターを追加します。

サーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成する

コマンドラインインターフェイスを使用してサーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成するには　コマンドプロンプトで入力します。

```
add lb vserver <name> HTTP <ip> <port>
<!--NeedCopy-->
```

例**:**

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

構成ユーティリティを使用してサーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成するには

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、[Add] をクリックします。

3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次のように次のパラメータの値を指定します。

   - 名前—名前
   - IP アドレス-
     IP アドレス注：仮想サーバが IPv6 を使用している場合は、
     IPv6 チェックボックスを選択し、IPv6 形式でアドレスを入力します（たとえば、
     1000:0000:0000:0000:0000:0005:0600:700 a: 888b）。
   - ポート-ポート

4. [プロトコル] で [HTTP] を選択します
5. [Create] をクリックしてから、[Close] をクリックします。作成した仮想サーバーが [負荷分散仮想サーバー] ペインに表示されます。

HTTP サービスを **HTTP** 仮想サーバーにバインドする

コマンドラインインターフェイスを使用して **HTTP** サービスをワイルドカード仮想サーバーにバインドするには
コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **HTTP** サービスをワイルドカード仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. サービスセクションをクリックして、バインドするサービスを選択します。

注:1 つのサービスを複数の仮想サーバーにバインドできます。

設定を保存して検証する

構成タスクが完了したら、必ず構成を保存してください。また、設定が正しいことを確認する必要があります。

コマンドラインインターフェイスを使用して構成を保存して確認するには　コマンドプロンプトで、次のコマンドを入力して透過モニタを構成し、構成を確認します。

- save ns config
- show vserver

例**:**


```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (192.168.100.10: \*) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (192.168.100.10:\*) - ANY
        State: UP
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 192.168.100.10
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: UP     Weight: 1
                Probes: 9       Failed [Total: 0 Current: 0]
                Last response: Success - HTTP response code 200
        received
                Response Time: 100.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```


構成ユーティリティを使用して構成を保存して検証するには

1. 詳細ペインで、[保存] をクリックします。
2. 「設定を保存」ダイアログで、「はい」をクリックします。
3. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
4. 詳細ウィンドウで、手順 5 で作成した仮想サーバーを選択し、[詳細] ウィンドウに表示される設定が正しいことを確認します。
5. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
6. 詳細ウィンドウで、手順 5 で作成したサービスを選択し、[詳細] ウィンドウに表示される設定が正しいことを確認します。

エンタープライズ環境におけるファイアウォール負荷分散設定の監視

構成が起動して実行されたら、各サービスと仮想サーバーの統計情報を表示して、考えられる問題がないかどうかを確認する必要があります。

仮想サーバーの統計情報を表示する

仮想サーバーのパフォーマンスを評価したり、問題をトラブルシューティングしたりするために、NetScaler ADC アプライアンスに構成されている仮想サーバーの詳細を表示できます。すべての仮想サーバーの統計情報の概要を表示することも、仮想サーバーの名前を指定して、その仮想サーバーの統計情報のみを表示することもできます。次の詳細を表示できます。

- 名前
- IP アドレス
- ポート
- プロトコル
- 仮想サーバーの状態
- 受け取ったリクエストの割合
- ヒット率

コマンドラインインターフェイスを使用して仮想サーバーの統計情報を表示するには　NetScaler ADC アプライアンスで現在構成されているすべての仮想サーバー、または単一の仮想サーバーの統計情報の概要を表示するには、コマンドプロンプトで次のように入力します。

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

例**:**

```
>stat lb vserver -detail
Virtual Server(s) Summary
```



```
3                          vsvrIP  port     Protocol        State    Req/s
                                Hits/s
4  One                          *    80         HTTP           UP      5/s
          0/s
5  Two                          *     0          TCP         DOWN      0/s
          0/s
6  Three                        *  2598          TCP         DOWN      0/s
          0/s
7  dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
          0/s
8  BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
          0/s
9  LBVIP           10.102.29.66    80         HTTP           UP      0/s
          0/s
10  Done
11
12
13 <!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーの統計情報を表示するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] > [統計] に移動します。
2. 1 つの仮想サーバーの統計のみを表示する場合は、詳細ペインで仮想サーバーを選択し、[統計] をクリックします。

サービスの統計情報の表示

更新された:2013 年 8 月 28 日

サービス統計を使用して、要求、応答、要求バイト、応答バイト、現在のクライアント接続、サージキュー内の要求、現在のサーバ接続などのレートを表示できます。

コマンドラインインターフェイスを使用してサービスの統計を表示するには　コマンドプロンプトで入力します。

```
1 stat service <name>
2 <!--NeedCopy-->
```

例**:**

```
1 stat service Service-HTTP-1
2 <!--NeedCopy-->
```

構成ユーティリティを使用してサービスの統計を表示するには

1. [トラフィック管理] > [負荷分散] > [サービス] > [統計] に移動します。
2. 1 つのサービスの統計のみを表示する場合は、サービスを選択して [Statistics] をクリックします。

複数ファイアウォール環境

August 15, 2023

マルチファイアウォール環境では、NetScaler アプライアンスは、パブリックインターネットに接続する外部セットと内部のプライベートネットワークに接続する内部セットの 2 つのファイアウォールの間に配置されます。外部セットは通常、出力トラフィックを処理します。これらのファイアウォールは主に、外部リソースへのアクセスを許可または拒否するアクセス制御リストを実装します。通常、内部セットが入力トラフィックを処理します。これらのファイアウォールは、侵入トラフィックの負荷分散とは別に、イントラネットを悪意のある攻撃から保護するセキュリティを実装しています。複数のファイアウォール環境では、別のファイアウォールからのトラフィックを負荷分散できます。デフォルトでは、ファイアウォールからのトラフィックは、NetScaler アプライアンスを介して他のファイアウォールで負荷分散されません。NetScaler の両側でファイアウォールの負荷分散を有効にすると、出力方向と入力方向の両方のトラフィックフローが改善され、トラフィックの処理が速くなります。

次の図は、複数のファイアウォールの負荷分散環境を示しています

図 1：ファイアウォール負荷分散 (マルチファイアウォール)

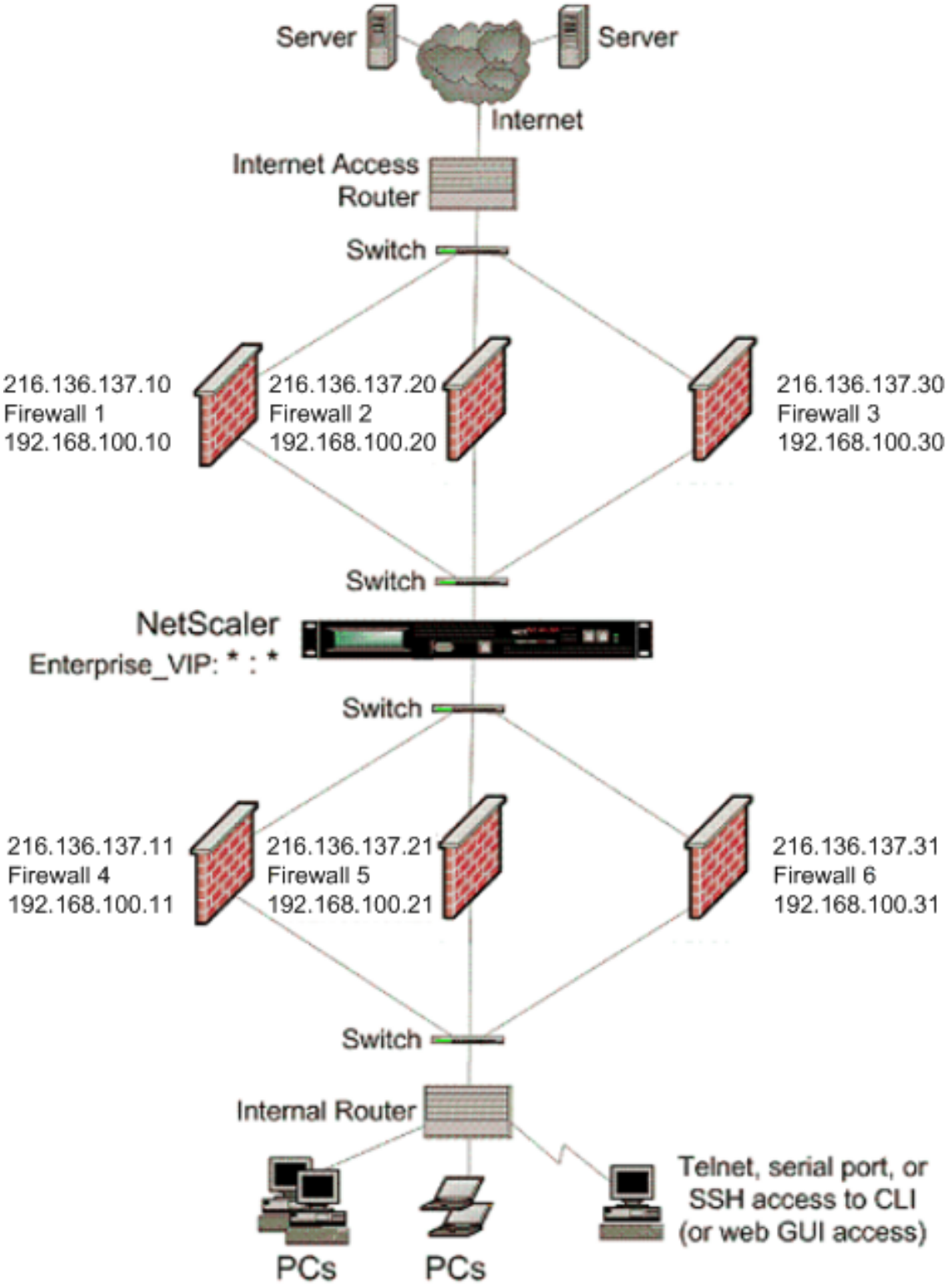


図 1 のような構成では、外部ファイアウォールによって負荷分散されている場合でも、内部ファイアウォールを介してトラフィックの負荷分散を行うように NetScaler を構成できます。たとえば、この機能を設定すると、外部ファイ

アウォール（ファイアウォール 1、2、3）からのトラフィックは、内部ファイアウォール（ファイアウォール 4、5、6）で負荷分散され、その逆も同様です。

ファイアウォールの負荷分散は、MAC モードの LB 仮想サーバーでのみサポートされます。

サービスタイプ ANY は、すべてのトラフィックを受け入れるように NetScaler ADC を構成します。

HTTP と TCP に関連する利点を利用するには、サービスと仮想サーバーを HTTP または TCP タイプに設定します。FTP が機能するには、FTP タイプでサービスを構成します。

マルチファイアウォール環境での **NetScaler** の構成

複数のファイアウォール環境で NetScaler アプライアンスを構成するには、負荷分散機能を有効にし、外部ファイアウォールを介して出力トラフィックの負荷分散を行うように仮想サーバーを構成し、内部ファイアウォールを介して入力トラフィックの負荷分散を行うように仮想サーバーを構成し、NetScaler アプライアンスでファイアウォールの負荷分散を有効にする必要があります。複数のファイアウォール環境でファイアウォールを介してトラフィックの負荷を分散するように仮想サーバーを構成するには、次のことが必要です。

1. ファイアウォールごとにワイルドカードサービスを構成する
2. ワイルドカードサービスごとにモニターを設定する
3. ワイルドカード仮想サーバーを構成して、ファイアウォールに送信されるトラフィックの負荷を分散します。
4. 仮想サーバーを MAC 書き換えモードで構成する
5. ファイアウォールサービスをワイルドカード仮想サーバーにバインドする

負荷分散機能を有効にする

サービスや仮想サーバーなどの負荷分散エンティティを構成して実装するには、NetScaler デバイスで負荷分散機能を有効にする必要があります。

CLI を使用して負荷分散を有効にするには**:**

コマンドプロンプトで次のコマンドを入力して負荷分散を有効にし、構成を確認します。

```
enable ns feature <featureName>
show ns feature
<!--NeedCopy-->
```

例**:**

```
enable ns feature LoadBalancing
Done
show ns feature
Feature Acronym Status
------- ------- ------
1) Web Logging WL OFF
2) Surge Protection SP ON
3) Load Balancing LB ON
```



```
.
.
.
24) NetScaler Push push OFF
Done
<!--NeedCopy-->
```

GUI を使用して負荷分散を有効にするには**:**

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. [設定] ペインの [モードと機能] で、[基本機能の変更] をクリックします。
3. 「基本機能の設定」ダイアログで、「負荷分散」チェックボックスを選択し、「OK」をクリックします。

各ファイアウォールのワイルドカードサービスの設定

すべてのプロトコルからのトラフィックを受け入れるには、すべてのプロトコルとポートのサポートを指定して、ファイアウォールごとにワイルドカードサービスを設定する必要があります。

CLI を使用して各ファイアウォールのワイルドカードサービスを設定するには:

コマンドプロンプトで次のコマンドを入力して、すべてのプロトコルとポートのサポートを設定します。

```
add service <name>@ <serverName> <serviceType> <port_number>
<!--NeedCopy-->
```

例**:**

```
add service fw-svc1 10.102.29.5 ANY *
<!--NeedCopy-->
```

GUI を使用して各ファイアウォールのワイルドカードサービスを設定するには**:**

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. 「サービスの作成」ダイアログ・ボックスで、次に示すように次のパラメータの値を指定します。

   - サービス名—名前
   - サーバー—サーバー名

   -* 必須パラメータ

4. [プロトコル] で [任意] を選択し、[ポート] で [*] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成したサービスが [サービス] ペインに表示されます。

各サービスのモニターの設定

PING モニターは、デフォルトではサービスにバインドされています。個々のファイアウォールを介して信頼側のホストを監視するには、透過的なモニターを設定する必要があります。その後、透過モニターをサービスにバインドできます。デフォルトの PING モニターは、NetScaler ADC アプライアンスと上流デバイス間の接続のみを監視します。透過モニタは、アプライアンスからモニタで指定された宛先 IP アドレスを所有するデバイスまでのパスに存在するすべてのデバイスを監視します。透過モニタが設定されておらず、ファイアウォールのステータスが UP であるが、そのファイアウォールのネクストホップデバイスの 1 つがダウンしている場合、アプライアンスは負荷分散の実行中にファイアウォールを組み込み、パケットをファイアウォールに転送します。ただし、ネクストホップデバイスの 1 つがダウンしているため、パケットは最終的な宛先に配信されません。透過モニターをバインドすると、デバイス（ファイアウォールを含む）のいずれかがダウンしている場合、サービスはダウンとしてマークされ、アプライアンスがファイアウォール負荷分散を実行するときにファイアウォールは含まれません。

透過モニターをバインドすると、PING モニターがオーバーライドされます。透過モニタに加えて PING モニタを設定するには、透過モニタを作成してバインドした後、PING モニタをサービスにバインドする必要があります。

CLI を使用してトランスペアレントモニターを設定するには:

コマンドプロンプトで、次のコマンドを入力して透過モニタを構成し、構成を確認します。

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

例:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

NetScaler アプライアンスは、サービスにバインドされたモニターからサーバー L2 パラメーターを学習します。UDP-ECV モニターの場合は、受信文字列を設定して、アプライアンスがサーバーの L2 パラメーターを学習できるようにします。受信文字列が設定されておらず、サーバーが応答しない場合、アプライアンスは L2 パラメーターを学習しませんが、サービスは UP に設定されます。このサービスのトラフィックはブラックホール化されています。

CLI を使用して受信文字列を設定するには:

コマンドプロンプトで、次のコマンドを入力します。

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )] [-send <string>] [-recv <string>]
<!--NeedCopy-->
```

例:

```
add lb monitor monitor-udp-1 udp-ecv -destip 10.10.10.11 -transparent
    YES  -send "test message"  -recv "site_is_up"
<!--NeedCopy-->
```


GUI を使用してトランスペアレントモニターを作成してバインドするには**:**

1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. 「モニターの作成」ダイアログ・ボックスで、次に示すように次のパラメータの値を指定します。

   - Name*
   - type*—type
   - 接続先 IP
   - 透明

   -* 必須パラメータ

4. [Create] をクリックしてから、[Close] をクリックします。[モニター] ペインで、構成したモニターを選択し、画面の下部に表示される設定が正しいことを確認します。

ファイアウォールに送信されるトラフィックの負荷分散を行うための仮想サーバーの設定

あらゆる種類のトラフィックの負荷を分散するには、プロトコルとポートを任意の値として指定するワイルドカード仮想サーバーを構成する必要があります。

CLI を使用してファイアウォールに送信されるトラフィックの負荷を分散するように仮想サーバーを設定するには:

コマンドプロンプトで、次のコマンドを入力します。

```
add lb vserver <name>@ <serviceType> <IPAddress> <port_number>
<!--NeedCopy-->
```

例**:**

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

GUI を使用してファイアウォールに送信されるトラフィックの負荷を分散するように仮想サーバーを設定するには:

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、[Add] をクリックします。
3. [プロトコル] で [任意] を選択し、[IP アドレスとポート] で [*] を選択します。
4. [Create] をクリックしてから、[Close] をクリックします。作成した仮想サーバーが [負荷分散仮想サーバー] ペインに表示されます。

仮想サーバーを **MAC** リライトモードに設定

着信トラフィックの転送に MAC アドレスを使用するように仮想サーバーを設定するには、MAC 書き換えモードを有効にする必要があります。

CLI を使用して仮想サーバを **MAC** リライトモードに設定するには:

コマンドプロンプトで、次のコマンドを入力します。

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

GUI を使用して仮想サーバを **MAC** リライトモードに設定するには:

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、リダイレクトモードを構成する仮想サーバー (たとえば、VServer-LB1) を選択し、[開く] をクリックします。
3. 「詳細設定」タブの「リダイレクトモード」モードで、「開く」をクリックします。
4. [OK] をクリックします。

ファイアウォールサービスの仮想サーバーへのバインディング

NetScaler アプライアンス上のサービスにアクセスするには、ワイルドカード仮想サーバーにバインドする必要があります。

CLI を使用してファイアウォールサービスを仮想サーバーにバインドするには:

コマンドプロンプトで、次のコマンドを入力します。

```
bind lb vserver <name>@ <serviceName>
<!--NeedCopy-->
```

例**:**

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してファイアウォールサービスを仮想サーバーにバインドするには:

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、リダイレクトモードを構成する仮想サーバー (たとえば、VServer-LB1) を選択し、[開く] をクリックします。

3. [仮想サーバーの構成 (負荷分散)] ダイアログボックスの [サービス] タブで、仮想サーバーにバインドするサービス (たとえば、Service-HTTP-1) の横にある [アクティブ] チェックボックスを選択します。
4. [OK] をクリックします。

NetScaler アプライアンスでのマルチファイアウォールの負荷分散の構成

ファイアウォールの負荷分散を使用して NetScaler の両側でトラフィックの負荷分散を行うには、vServerSpecificMac パラメーターを使用してマルチファイアウォールの負荷分散を有効にする必要があります。

CLI を使用してマルチファイアウォールの負荷分散を設定するには:

コマンドプロンプトで、次のコマンドを入力します。

```
set lb parameter -vServerSpecificMac <status>
<!--NeedCopy-->
```

例**:**

```
set lb parameter -vServerSpecificMac ENABLED
<!--NeedCopy-->
```

GUI を使用してマルチファイアウォールのロードバランシングを設定するには:

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、リダイレクションモードを構成する仮想サーバーを選択します (たとえば、負荷分散パラメーターの構成)。
3. [負荷分散パラメータの設定] ダイアログボックスで、[仮想サーバー固有の MAC] チェックボックスを選択します。
4. [OK] をクリックします。

設定の保存と検証

構成タスクが完了したら、必ず構成を保存してください。また、設定が正しいことを確認する必要があります。

CLI を使用して設定を保存して確認するには:

コマンドプロンプトで、次のコマンドを入力して透過モニタを構成し、構成を確認します。

- save ns config
- show vserver

例**:**

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (\*:\*) - ANY    Type: ADDRESS
        State: UP
```



```
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: \*) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:\*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```

GUI を使用して設定を保存して確認するには:

1. 詳細ペインで、[保存] をクリックします。

2. 「設定を保存」ダイアログで、「はい」をクリックします。
3. ［Traffic Management］> ［Load Balancing］> ［Virtual Servers］の順に選択します。
4. 詳細ウィンドウで、手順 5 で作成した仮想サーバーを選択し、[詳細] ウィンドウに表示される設定が正しいことを確認します。
5. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
6. 詳細ウィンドウで、手順 5 で作成したサービスを選択し、[詳細] ウィンドウに表示される設定が正しいことを確認します。

マルチファイアウォール環境におけるファイアウォールの負荷分散設定の監視

構成が起動して実行されたら、各サービスと仮想サーバーの統計情報を表示して、考えられる問題がないかどうかを確認する必要があります。

仮想サーバーの統計情報を表示する

仮想サーバーのパフォーマンスを評価したり、問題をトラブルシューティングしたりするために、NetScaler ADC アプライアンスに構成されている仮想サーバーの詳細を表示できます。すべての仮想サーバーの統計情報の概要を表示することも、仮想サーバーの名前を指定して、その仮想サーバーの統計情報のみを表示することもできます。次の詳細を表示できます。

- 名前
- IP アドレス
- ポート
- プロトコル
- 仮想サーバーの状態
- 受け取ったリクエストの割合
- ヒット率

コマンドラインインターフェイスを使用して仮想サーバーの統計情報を表示するには　NetScaler ADC アプライアンスで現在構成されているすべての仮想サーバー、または単一の仮想サーバーの統計情報の概要を表示するには、コマンドプロンプトで次のように入力します。

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

例**:**

```
>stat lb vserver -detail
Virtual Server(s) Summary
                      vsvrIP  port     Protocol        State    Req/s
                           Hits/s
```



```
4  One                          *    80          HTTP            UP       5/s
          0/s
5  Two                          *     0           TCP          DOWN       0/s
          0/s
6  Three                        *  2598           TCP          DOWN       0/s
          0/s
7  dnsVirtualNS     10.102.29.90    53           DNS          DOWN       0/s
          0/s
8  BRVSERV             10.10.1.1    80          HTTP          DOWN       0/s
          0/s
9  LBVIP            10.102.29.66    80          HTTP            UP       0/s
          0/s
10  Done
11
12
13 <!--NeedCopy-->
```

GUI を使用して仮想サーバーの統計情報を表示するには:

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] > [統計] に移動します。
2. 1 つの仮想サーバーの統計のみを表示する場合は、詳細ペインで仮想サーバーを選択し、[統計] をクリックします。

サービスの統計情報の表示

サービス統計を使用して、要求、応答、要求バイト、応答バイト、現在のクライアント接続、サージキュー内の要求、現在のサーバ接続などのレートを表示できます。

CLI を使用してサービスの統計情報を表示するには**:**

コマンドプロンプトで入力します。

```
1 stat service <name>
2 <!--NeedCopy-->
```

例**:**

```
1 stat service Service-HTTP-1
2 <!--NeedCopy-->
```

GUI を使用してサービスの統計情報を表示するには**:**

1. [トラフィック管理] > [負荷分散] > [サービス] > [統計] に移動します。
2. 1 つのサービスの統計のみを表示する場合は、サービスを選択して [Statistics] をクリックします。

Global Server Load Balancing

August 15, 2023

> 注:
>
> - リリース 13.0 ビルド 41.x 以降、NetScaler アプライアンスを使用するグローバルサーバー負荷分散（GSLB）展開は、2019 年の DNS フラグデーに完全に準拠しています。
> - GSLB 機能は NetScaler Advance およびプレミアムエディションのライセンスに含まれています。NetScaler オプションライセンスはスタンダードエディションでサポートされています。

GSLB 用に構成された NetScaler アプライアンスは、WAN 内の障害点から保護することで、災害復旧を実現し、アプリケーションの継続的な可用性を確保します。GSLB は、クライアントのリクエストを最も近いまたは最もパフォーマンスの高いデータセンター、または障害が発生した場合は存続しているデータセンターに転送することで、データセンター全体の負荷を分散します。

一般的な構成では、ローカル DNS サーバーは、GSLB サービスにバインドされた GSLB 仮想サーバーにクライアント要求を送信します。GSLB サービスは、ローカルサイトまたはリモートサイトにある負荷分散またはコンテンツスイッチング仮想サーバーを識別します。GSLB 仮想サーバーがリモートサイトで負荷分散またはコンテンツスイッチング仮想サーバーを選択すると、仮想サーバーの IP アドレスが DNS サーバーに送信されます。DNS サーバーはそれをクライアントに送信します。次に、クライアントは要求を新しい IP の新しい仮想サーバーに再送信します。

設定する必要がある GSLB エンティティは、GSLB サイト、GSLB サービス、GSLB 仮想サーバー、負荷分散またはコンテンツスイッチング仮想サーバー、および権限のある DNS（ADNS）サービスです。MEP も設定する必要があります。また、DNS ビューを設定して、さまざまな場所からネットワークにアクセスするクライアントにネットワークのさまざまな部分を公開することもできます。

> 注:
>
> GSLB の機能を最大限に活用するには、各データセンターの負荷分散またはコンテンツスイッチングに ADC アプライアンスを使用してください。そうすれば、GSLB 構成では独自の MEP を使用してサイトメトリックを交換できます。

GSLB の仕組み

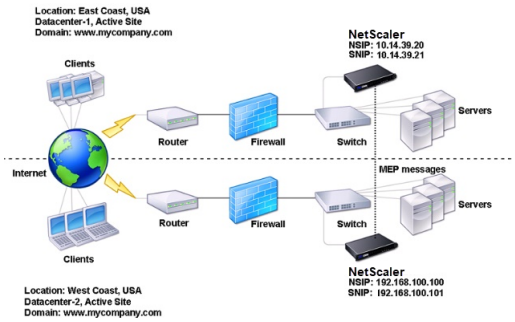
通常の DNS では、クライアントがドメインネームシステム (DNS) 要求を送信すると、ドメインまたはサービスの IP アドレスの一覧を受信します。通常、クライアントはリストの最初の IP アドレスを選択し、そのサーバーとの接続を開始します。DNS サーバーは、DNS ラウンドロビンと呼ばれる手法を使用して、リスト上の IP をローテーションします。最初の IP アドレスをリストの最後に送信し、各 DNS リクエストに応答した後に残りの IP アドレスを昇格させます。この手法では、負荷の均等な分散が保証されますが、障害復旧、サーバーの負荷または近接性に基づく負荷分散、または永続性はサポートされません。

ADC アプライアンスで GSLB を設定し MEP を有効にすると、DNS インフラストラクチャを使用して、設定された基準に最も適合するデータセンターにクライアントを接続します。基準では以下を指定できます。

- 最も負荷の少ないデータセンター
- 最も近いデータセンター
- クライアントの所在地からの要求に最も迅速に応答するデータセンター
- これらのメトリックと SNMP メトリックの組み合わせ。

アプライアンスは、各データセンターの場所、パフォーマンス、負荷、および可用性を追跡します。これらの要因を使用して、クライアント要求を送信するデータセンターを選択します。

次の図は、基本的な GSLB トポロジーを示しています。



GSLB 構成は、構成内の各アプライアンス上の GSLB エンティティのグループで構成されます。これらのエンティティには、GSLB サイト、GSLB サービス、GSLB サービスグループ、GSLB 仮想サーバー、負荷分散サーバー、コンテンツスイッチングサーバー、および ADNS サービスが含まれます。

GSLB 展開タイプ

August 15, 2023

グローバルサーバー負荷分散（GSLB）用に構成された NetScaler アプライアンスは、ワイドエリアネットワーク（WAN）の障害点から保護することにより、災害復旧を実現し、アプリケーションの継続的な可用性を確保します。GSLB は、クライアントのリクエストを最も近いデータセンター、または最もパフォーマンスの高いデータセンター、または停電時に存続しているデータセンターに転送することで、データセンター全体の負荷を分散できます。

以下に、一般的な GSLB デプロイメントの種類の一部を示します。

- アクティブ/アクティブサイト展開
- アクティブ/パッシブサイト展開
- 親子トポロジーの導入

アクティブ/アクティブサイト展開

August 15, 2023

アクティブ-アクティブサイトは、複数のアクティブなデータセンターで構成されます。クライアント要求は、アクティブなデータセンター間で負荷分散されます。この展開の種類は、分散環境でトラフィックをグローバルに分散する必要がある場合に使用できます。

アクティブ/アクティブ展開内のすべてのサイトがアクティブであり、特定のアプリケーション/ドメインのすべてのサービスが同じ GSLB 仮想サーバーにバインドされます。サイトは、メトリック交換プロトコル（MEP）を介してメトリックを交換します。サイト間で交換されるサイトメトリックには、各負荷分散およびコンテンツスイッチング仮想サーバーのステータス、現在の接続数、現在のパケットレート、および現在の帯域幅使用量が含まれます。NetScaler ADC アプライアンスは、サイト間で負荷分散を実行するためにこの情報を必要とします。

MEP は 32 を超えるサイトを同期できないため、アクティブ/アクティブ展開には最大 32 の GSLB サイトを含めることができます。この展開タイプではバックアップサイトは構成されていません。

NetScaler ADC アプライアンスは、GSLB 構成で指定された GSLB 方式によって決定された適切な GSLB サイトにクライアント要求を送信します。

アクティブ/アクティブデプロイでは、次の GSLB メソッドを設定できます。

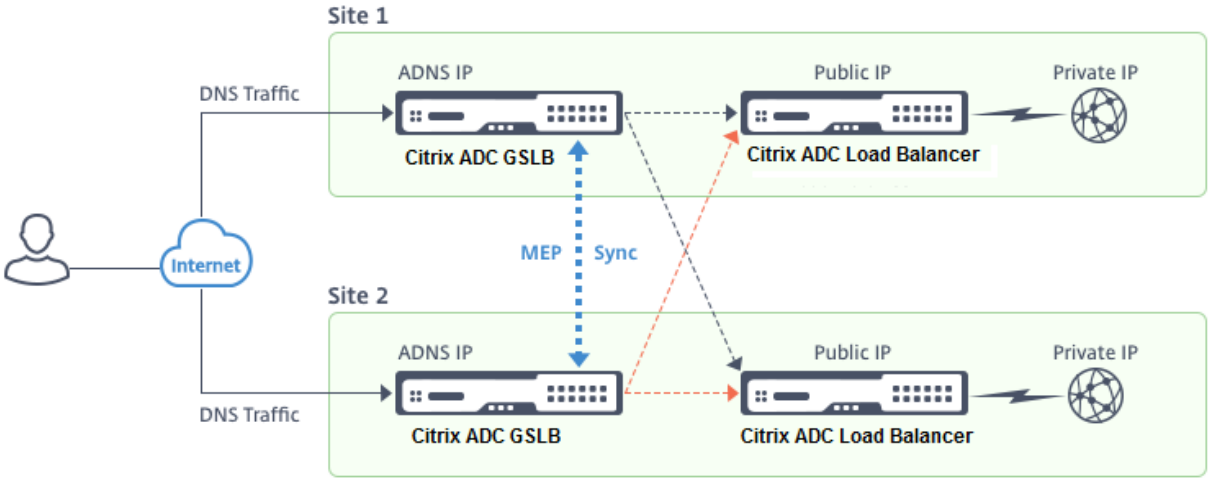
- ラウンドロビン
- 最小接続数
- 最短応答時間
- 最小帯域幅
- 最小パケット
- 送信元 IP ハッシュ
- カスタムロード
- ラウンドトリップ時間（RTT）
- 静的近接

> 注:
>
> - MEP が無効の場合、次の GSLB メソッドはデフォルトでラウンドロビン方式に設定されます。
>   - RTT
>   - 最小接続
>   - 最小帯域幅
>   - 最小パケット
>   - 最小応答時間
> - 静的近接 GLSB 方式では、アプライアンスは近接基準に最もよく一致するサイトの IP アドレスに要求を送信します。
> - ラウンドトリップ時間方式では、動的ラウンドトリップ時間（RTT）の値は、最もパフォーマンスの高い

> サイトの IP アドレスを選択するためのものです。RTT は、クライアントのローカル DNS サーバーとデータリソース間のネットワークの遅延の尺度です。

GSLB アクティブ/アクティブデータセントートポロジ

この図では、サイト 1 とサイト 2 はアクティブな GLSB サイトです。



クライアントは DNS 要求を送信すると、アクティブサイトの 1 つに着陸します。

サイト 1 がクライアント要求を受け取ると、サイト 1 の GSLB 仮想サーバーは負荷分散またはコンテンツスイッチング仮想サーバーを選択し、仮想サーバーの IP アドレスを DNS サーバーに送信し、DNS サーバーはそれをクライアントに送信します。次に、クライアントは新しい IP アドレスの新しい仮想サーバーに要求を再送信します。

両方のサイトがアクティブなので、GSLB アルゴリズムは、設定された GSLB メソッドによって決定された選択を行う際に、両方のサイトのサービスを評価します。

アクティブ/パッシブサイト展開

August 15, 2023

アクティブ/パッシブサイトは、アクティブデータセンターとパッシブデータセンターで構成されます。この展開タイプは、ディザスタリカバリに最適です。

このタイプの展開では、一部のサイト (リモートサイト) は障害復旧専用になります。これらのサイトは、すべてのアクティブサイトがダウンするまで意思決定に参加しません。パッシブサイトは、災害イベントによってフェイルオーバーがトリガーされない限り稼働しません。

プライマリデータセンターを設定したら、バックアップデータセンターの構成を複製し、そのサイトの GSLB 仮想サーバーをバックアップ仮想サーバーとして指定して、それをパッシブ GSLB サイトとして指定します。

MEP では 32 を超えるサイトを同期できないため、アクティブ/パッシブ展開には最大 32 の GSLB サイトを含めることができます。

アクティブ/パッシブ展開では、次の GSLB メソッドを設定できます。

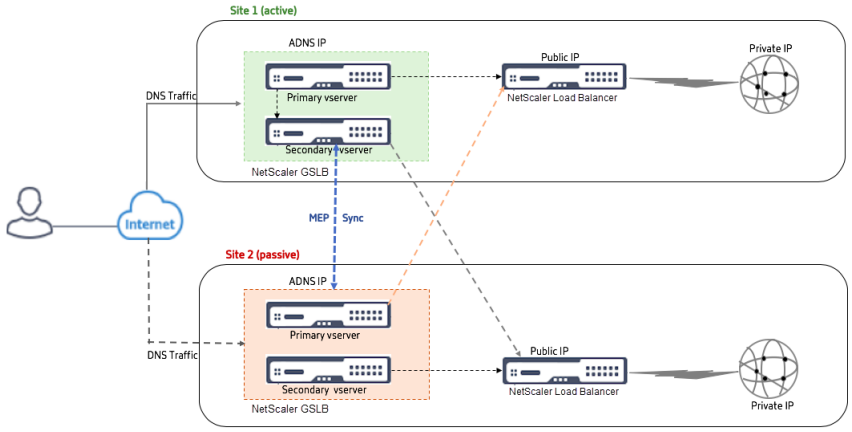
- ラウンドロビン
- 最小接続数
- 最短応答時間
- 最小帯域幅
- 最小パケット
- 送信元 IP ハッシュ
- カスタムロード
- ラウンドトリップ時間（RTT）
- 静的近接

注記:

- MEP が無効になっている場合、次のアルゴリズムメソッドはデフォルトでラウンドロビンになります。
  - RTT
  - 最小接続数
  - 最小帯域幅
  - 最小パケット
  - 最短応答時間
- 静的近接 GLSB 方式では、アプライアンスは近接基準に最もよく一致するサイトの IP アドレスに要求を送信します。
- ラウンドトリップ時間方式では、動的ラウンドトリップ時間（RTT）の値は、最もパフォーマンスの高いサイトの IP アドレスを選択するためのものです。RTT は、クライアントのローカル DNS サーバーとデータリソース間のネットワークの遅延の尺度です。

GSLB アクティブ/パッシブデータセントートポロジー

この図では、サイト 1 はアクティブサイト、サイト 2 はパッシブサイトで、サイト 1 と同じ構成になっています。



サイト 1 が DOWN になると、サイト 2 が動作可能になります。

クライアントが DNS リクエストを送信すると、そのリクエストはどのサイトにも届きます。ただし、サービスが UP の限り、アクティブなサイト（Site1）からのみ選択されます。

パッシブサイト（サイト 2）からのサービスは、アクティブサイト（サイト 1）が DOWN の場合にのみ選択されます。

MEP プロトコルを使用した親子トポロジ展開

August 15, 2023

NetScaler GSLB は、関係するすべてのサイト間にメッシュ接続を作成し、インテリジェントな負荷分散の決定を行うことにより、グローバルなサーバー負荷分散と障害回復を提供します。各サイトは他のサイトと通信し、メトリック交換プロトコル (MEP) を通じて定期的にサーバとネットワークのメトリックを交換します。ただし、ピアサイトの数が増えると、メッシュトポロジが原因で MEP トラフィックの量は指数関数的に増加します。これを解決するには、親子トポロジを使用します。親子トポロジでは、大規模な展開もサポートされます。32 の親サイトに加えて、1024 の子サイトを構成できます。

GSLB 親子トポロジは、次の特徴を持つ 2 レベルの階層設計です。

- 最上位レベルには親サイトがあり、親サイトには他の親とピア関係があります。
- 各親は複数の子サイトを持つことができます。
- 各親サイトは、子サイトや他の親サイトとヘルス情報を交換します。
- 子サイトは親サイトとのみ通信します。
- GSLB の親子関係では、親サイトだけが ADNS クエリに応答します。子サイトは通常の負荷分散サイトとして機能します。
- ADNS サービスまたは DNS 負荷分散仮想サーバーは、親サイトでのみ構成します。

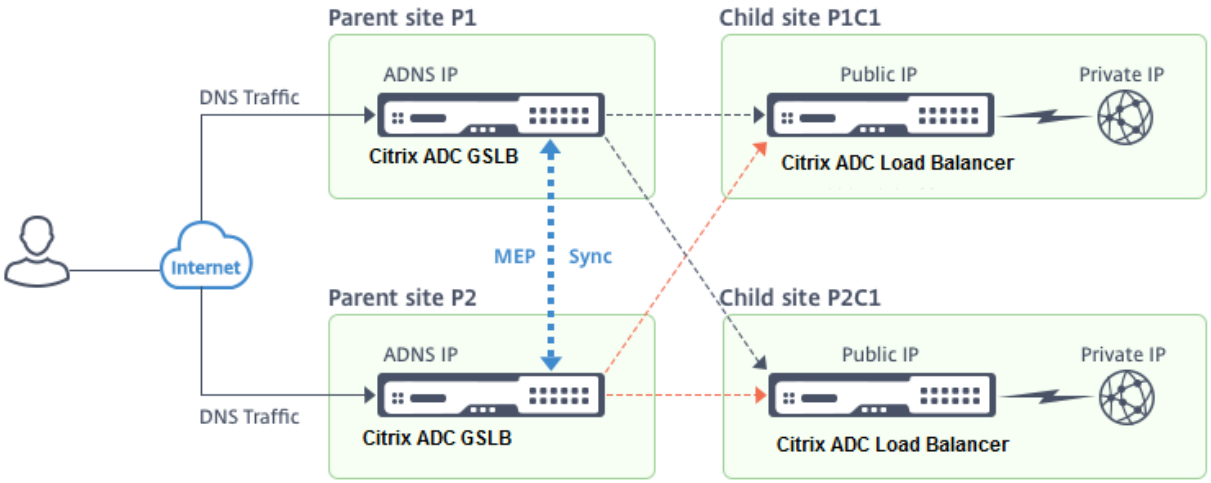
- 親サイトは通常の GSLB 構成、つまりローカルサイトとすべてのリモートサイトからのサービスを持つことができますが、子サイトはローカルサービスのみを持つことができます。また、GSLB 仮想サーバーが構成されているのは親サイトのみです。

> 注
>
> - 親子トポロジでは、2 つの IP アドレスのうち低い方からサイトメトリックの交換が開始されます。ただし、NetScaler ADC リリース 11.1 ビルド 51.x 以降、親サイトは子サイトへの接続を開始し、その逆ではありません。親サイトには、GSLB セットアップのすべての子サイトに関する情報があるためです。

- 親-親接続では、2 つの IP アドレスの下位 IP からサイトメトリックの交換が開始されます。
- 親子トポロジでは、子サイトで GSLB サービスを構成する必要は必ずしもありません。ただし、クライアント認証、クライアント IP アドレスの挿入、その他の SSL 固有の要件など、さらに多くの構成がある場合は、子サイトに明示的な GSLB サービスを追加し、それに応じて構成する必要があります。
- 親子トポロジでは、親サイトと子サイトが異なる NetScaler ソフトウェアバージョンにある可能性があります。ただし、GSLB AutomaticConfigSync オプションを使用して親サイト間で構成を同期するには、すべての親サイトが同じ NetScaler ソフトウェアバージョンである必要があります。AutomaticConfigSync オプションを使用していない場合は、親サイトと子サイトが異なる NetScaler ソフトウェアバージョンを使用している可能性がありますが、最新リリースの新機能を使用していないことを確認してください。これは一般に、GSLB に参加している 2 つの NetScaler ADC ノードにも当てはまります。

基本的な親子トポロジ

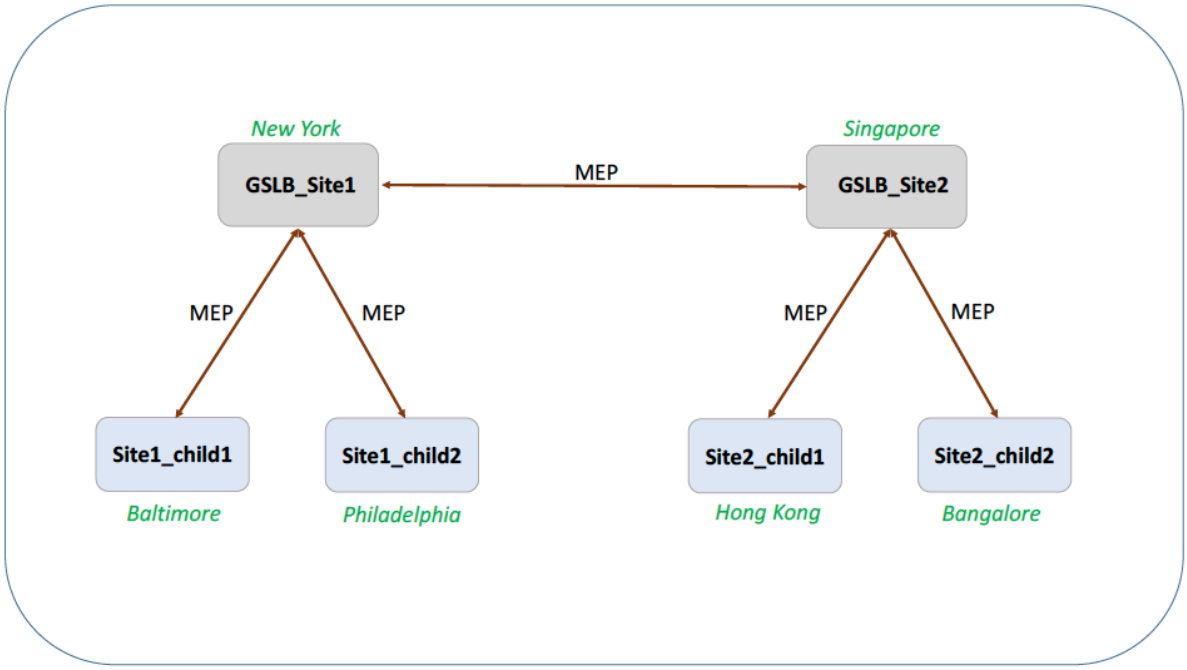
この図では、SiteP1 と SiteP2 はピア関係の親サイトです。サイト P1C1 と P2C1 はそれぞれ P1 と P2 の子サイトです。



GSLB の親子構成を設定する

GSLB サイトでファイアウォールが設定されている場合は、ポート 3011 が開いていることを確認します。

次の図に、親子構成の例を示します。



- 子サイトの構成には子サイトとその親サイトが含まれますが、他の親サイトや子サイトは含まれません。
- RTT やパーシステンスセッション情報などのネットワークメトリックは、親サイト間でのみ同期されます。したがって、NWMetricExchange や SessionExchange などのパラメーターは、すべての子サイトで既定で無効になっています。
- 正しい親子構成を確認するには、親サイトにバインドされているすべての GSLB サービスの状態を確認します。

CLI を使用して **GSLB** の親子構成を設定するには、次の手順を実行します。

1. 各親サイトで、すべての子サイト、ピアの親サイト、およびピアサイトに関連付けられた子サイトを構成します。

   親サイトを追加する際には、以下のコマンドを使用します。

   ```
   add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
       ipv6_addr|*>]
   <!--NeedCopy-->
   ```

   子サイトを追加する際には、以下のコマンドを使用します。

   ```
   add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
       ipv6_addr|*>] [-parentSite <string>]
   <!--NeedCopy-->
   ```

2. 子サイトで、子サイトを構成し、子サイトを親サイトに関連付けます。

> 注:
>
> 親サイトと子サイトの関連付けを正しく構成してください。たとえば、Site1_child1_child1 を GSLB_Site1 で設定する必要があります。Site1_child1 を GSLB_Site2 で設定することはできません。

子サイトが関連付けられている親サイトを構成するには、以下のコマンドを使用します。

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
    ipv6_addr|*>]
<!--NeedCopy-->
```

子サイトを追加して親サイトに関連付けるには、以下のコマンドを使用します。

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
    ipv6_addr|*>] [-parentSite <string>]
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用した親子設定の完全な例については、CLI を使用した完全な親子設定の例を参照してください。

> 注
>
> 負荷分散仮想サーバーの IP アドレスがプライベート IP アドレスで、パブリック IP アドレスがこの IP アドレスと異なる場合は、子サイトのローカル負荷分散仮想サーバーに GSLB サービスを構成する必要があります。これは、親サイトと子サイト間の統計収集に必要です。
>
> 子サイトのコマンドプロンプトで、次のように入力します。
>
> ```
> add gslb service <name> <private IP/lb vserver IP> http 80 -
> sitename <childsite name> -publicip <public IP of LB vserver>
> ```
>
> 例:
>
> ```
> add gslb service Service-GSLB 192.168.1.3 http 80 -GSLB_Site11
> site 11_lb1 172.16.1.1
> ```
>
> 192.168.1.3 は負荷分散仮想サーバーのプライベート IP アドレス、172.16.1.1 は負荷分散仮想サーバーのパブリック IP アドレスです。

親サイトをバックアップする

注：この機能は、NetScaler ADC リリース 11.1 ビルド 51.x で導入されました。バックアップ親サイトトポロジを使用するには、親サイトと子サイトが NetScaler ADC 11.1 build 51.x 以降にあることを確認します。

親サイトのバックアップトポロジは、1 つの親サイトに多数の子サイトが関連付けられている場合に便利です。この親サイトが DOWN になると、その子サイトはすべて使用できなくなります。これを防ぐために、元の親サイトが

DOWN の場合に子サイトが接続できるバックアップ親サイトを構成できるようになりました。親サイトは MEP メッセージを通じてバックアップ親リストを子サイトに送信します。

親サイトが DOWN になると、GSLB 内の他の親サイトは MEP を通じて特定の親サイトが DOWN であることを認識します。これは、その親サイトに対する MEP が DOWN であるためです。GSLB セットアップ内の他の親サイトは、ピアの親のバックアップチェーンを検索します。優先度が最も高い親サイトは、DOWN になった親の子サイトを採用します。その後、新しい親が子サイトとの接続を開始します。子サイトは、既存の接続とバックアップリストの情報を評価した後に、接続を許可または拒否できます。バックアップの親が子サイトを採用するまでに数秒かかります。元の親サイトがバックアップされると、別の親サイトに移行した子サイトとの接続を確立しようとします。接続が成功すると、子サイトは元の親サイトに再割り当てされます。

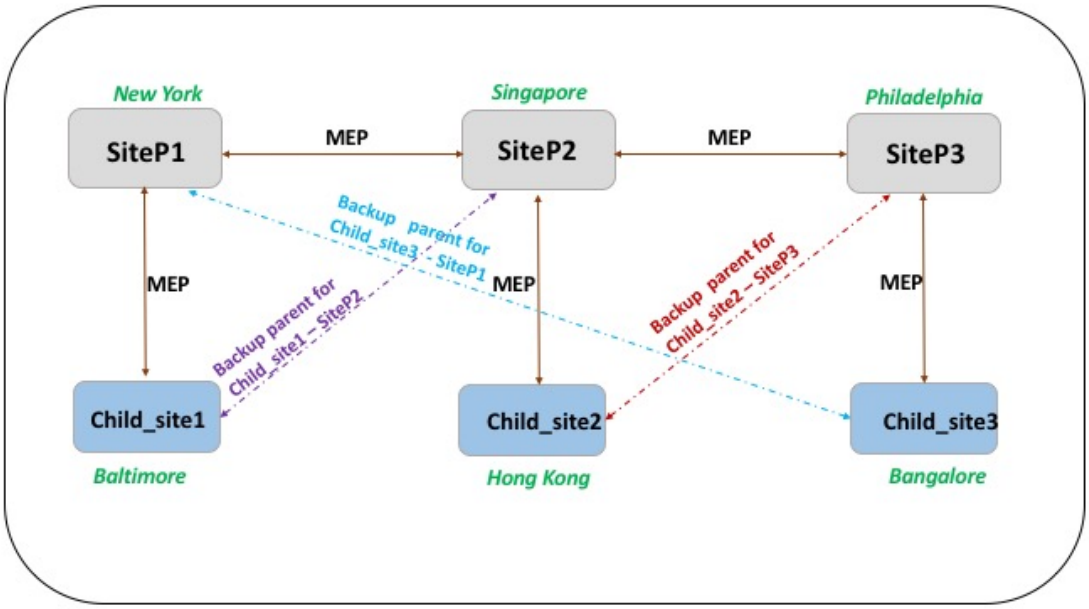
注記:

- バックアップとして構成できるのは親サイトのみで、この構成は親サイトでのみ実行できます。
- すべての子サイトがバックアップの親セットを使用します。
- 同期は親サイトでのみ行われます。GSLB 子サイトの構成は同期の影響を受けません。これは、親サイトと子サイトの構成が同一ではないためです。子サイトの構成は、そのサイトとその親サイトの詳細のみで構成されます。また、GSLB サービスは必ずしも子サイトで設定する必要はありません。

次の図に示す構成について考えてみます。

- SiteP1、siteP2、および siteP3 は親サイトです。
- child_site1、child_site2、および child_site3 は、それぞれ siteP1、siteP2、および siteP3 の子サイトです。
- 親サイトをバックアップする。
  - siteP1 バックアップの親-siteP2 (より高い優先度) と siteP3
  - siteP2 バックアップの親—siteP3 (優先度が高い) と siteP1
  - siteP3 バックアップの親—siteP1 (優先度が高い) と siteP2

注: 説明のため、この図は親サイトごとに 1 つのバックアップ親のみを示しています。



次の一覧は、さまざまなシナリオにおける親サイトと子サイトの動作をまとめたものです。

- シナリオ 1: SiteP1 がダウンする。
  - siteP2 と siteP3 は、siteP1 の MEP 接続がダウンしていることを検出します。siteP2 は siteP1 のバックアップ親のプリファレンスリストの上位にあるため、child_site1 への接続を開始しようとします。siteP3 は、child_site1 が親 siteP2 の子サイトになったと想定しています。
  - siteP2 は、Child_Site1 に SiteP1 のバックアップ親 (siteP2 および siteP3) のリストを child_Site1 に送信します。Child_site1 はこのリストを使用して、SiteP2 からの接続を受け入れるか拒否するかを決定します。接続を受け入れ、siteP2 の子になります。
  - siteP1 がバックアップされると、child_site1 に接続要求が送信されます。新しいリクエストが優先され、child_site 1 は siteP1 に移行されます。
- シナリオ 2: SiteP1 と SiteP2 の間の MEP 接続だけがダウンしました。Child_site1 は siteP2 の接続要求を拒否します。これは、その親 siteP1 がまだ UP であるためです。
- シナリオ 3: siteP3 と Child_Site1 が siteP1 がダウンしていることを検出し、siteP3 と SiteP2 の間の MEP 接続もダウンしています。ただし、siteP2 は siteP1 が UP であることを検出し、siteP1 と SiteP2 の間の MEP 接続は UP です。
  - SiteP2 は何のアクションも実行しません。
  - SiteP3 は SiteP1 のバックアップリストをチェックし、SiteP2 が SiteP3 よりも優先度が高いことを検出します。しかし、siteP2 はダウンしているため、siteP3 は child_site1 との接続を確立しようとします。child_site1 は siteP1 がダウンしていることを検出したため、SiteP3 の接続要求を受け入れます。

- これで、siteP1 と siteP2 の間の接続がダウンします。SiteP2 は SiteP1 のバックアップリストをチェックし、自身を最も優先されるバックアップとして検出し、child_site1 への接続を試みます。child_site1 は siteP1 のリストに基づいて新しい接続要求を評価し、SiteP2 を最も優先されるバックアップとして検出し、siteP3 から siteP2 に移行します。

コマンドラインインターフェイスを使用してバックアップ親サイトを構成するには

コマンドプロンプトで次のように入力します。

```
set gslb site <sitename> -backupParentlist <bkp_site1> <bkp_site2> … <
    bkp_site5>
<!--NeedCopy-->
```

<sitename> は現在の親サイトです。

例**:**

親サイト (SiteP1) では、サイト (SiteP2 と SiteP3) がバックアップ親サイトとして構成されます。

```
set gslb site SiteP1 -backupParentlist SiteP2 SiteP3
<!--NeedCopy-->
```

注記:

- 新しいサイトをバックアップの親として追加することはできません。最初にすべてのサイトを追加してから、そのサイトをバックアップの親として構成する必要があります。
- バックアップの親を削除するには、unset コマンドを使用する必要があります。このコマンドは、以前にバックアップ親サイトとして構成されていたすべてのサイトの設定を解除します。

GUI を使用してバックアップ親サイトを構成するには

1. 構成 > トラフィック管理 > **GSLB** > サイトに移動します。
2. 新しいサイトを追加するか、既存のサイトを選択します。
3. GSLB サイトを作成または構成するときに、**[**親サイトのバックアップ] オプションボックスを選択します。

GSLB 構成エンティティ

August 15, 2023

GSLB 構成は、構成内の各アプライアンス上の GSLB エンティティのグループで構成されます。これらのエンティティには次のものが含まれます。

- GSLB サイト

- GSLB サービス
- GSLB 仮想サーバ
- 負荷分散またはコンテンツスイッチング仮想サーバー
- ADNS サービス
- DNS VIP

GSLB サイト

一般的な GSLB セットアップはデータセンターで構成され、各データセンターには、NetScaler ADC アプライアンスである場合とそうでない場合があるさまざまなネットワークアプライアンスがあります。データセンターは GSLB サイトと呼ばれます。各 GSLB サイトは、そのサイトのローカルにある NetScaler ADC アプライアンスによって管理されます。これらのアプライアンスはそれぞれ、自身のサイトをローカルサイトとして扱い、他のアプライアンスによって管理されている他のすべてのサイトをリモートサイトとして扱います。

サイトを管理するアプライアンスがそのデータセンター内の唯一の NetScaler ADC アプライアンスである場合、そのアプライアンスでホストされている GSLB サイトは、メトリックを収集できないため、監査目的のブックキーピングプレースホルダーとして機能します。通常、これは、アプライアンスが GSLB に対してのみ使用され、データセンター内の他の製品が負荷分散やコンテンツスイッチングに使用される場合に発生します。

GSLB サイト間の関係

サイトの概念は、NetScaler GSLB 実装の中心です。特に指定のない限り、サイトは相互にピア関係を形成します。この関係は、まず健康情報を交換し、次に選択したアルゴリズムで決定された負荷を分散するために使用されます。しかし、多くの状況では、すべての GSLB サイト間のピア関係は望ましくありません。オールピア実装を行わない理由としては、

- GSLB サイトを明確に分離する。たとえば、DNS クエリの解決に関与するサイトをトラフィック管理サイトから分離します。
- ピアサイトの数が増えるにつれて指数関数的に増加するメトリック交換プロトコル（MEP）トラフィックの量を減らすこと。

これらの目標は、親と子の GSLB サイトを使用することによって達成できます。

GSLB サービス

GSLB サービスは通常、負荷分散またはコンテンツスイッチング仮想サーバーを表しますが、どのタイプの仮想サーバーでも表すことができます。GSLB サービスは、仮想サーバーの IP アドレス、ポート番号、およびサービスタイプを識別します。GSLB サービスは、GSLB サイトを管理する NetScaler ADC アプライアンス上の GSLB 仮想サーバーにバインドされます。同じデータセンター内の GSLB 仮想サーバーにバインドされた GSLB サービスは、GSLB 仮想サーバーのローカルサービスです。別のデータセンター内の GSLB 仮想サーバーにバインドされた GSLB サービスは、その GSLB 仮想サーバーからリモートです。

> 注
>
> サイトとサービスは本質的にリンクされており、2 つのサイトとサービスが近接していることを示します。つまり、すべてのサービスはサイトに属している必要があり、近接性の観点から GSLB サイトと同じ場所にあると想定されます。同様に、サービスと仮想サーバーはリンクされているため、ロジックは使用可能なリソースにリンクされます。

GSLB 仮想サーバ

GSLB 仮想サーバーには 1 つ以上の GSLB サービスがバインドされており、それらのサービス間でトラフィックを負荷分散します。設定されている GSLB メソッド (アルゴリズム) を評価して、クライアントリクエストを送信する適切なサービスを選択します。GSLB サービスはローカルサーバーまたはリモートサーバーを表すことができるため、要求に対して最適な GSLB サービスを選択すると、クライアント要求を処理するデータセンターを選択する効果があります。

グローバルサーバーの負荷分散が設定されているドメインは、GSLB 仮想サーバーにバインドする必要があります。これは、仮想サーバーにバインドされた 1 つ以上のサービスが、そのドメインに対して行われたリクエストを処理するためです。

NetScaler ADC アプライアンス上で構成されている他の仮想サーバーとは異なり、GSLB 仮想サーバーには独自の仮想 IP アドレス（VIP）がありません。

負荷分散またはコンテンツスイッチング仮想サーバー

負荷分散またはコンテンツスイッチング仮想サーバーは、ローカルネットワーク上の 1 つまたは複数の物理サーバーを表します。クライアントは負荷分散またはコンテンツスイッチング仮想サーバーの仮想 IP (VIP) アドレスに要求を送信し、仮想サーバーは物理サーバー間で負荷を分散します。GSLB 仮想サーバーがローカルまたはリモートの負荷分散またはコンテンツスイッチング仮想サーバーを表す GSLB サービスを選択した後、クライアントはその仮想サーバーの VIP アドレスに要求を送信します。

負荷分散またはコンテンツスイッチング仮想サーバーおよびサービスの詳細については、「[負荷分散またはコンテンツスイッチング](/ja-jp/citrix-adc/13-1/content-switching.html)」を参照してください。

ADNS サービス

ADNS サービスは、NetScaler ADC アプライアンスが権限を持つドメインの DNS 要求にのみ応答する特別な種類のサービスです。ADNS サービスが設定されると、アプライアンスは ADNS サービスの IP アドレスを所有し、それをアドバタイズします。ADNS サービスによって DNS 要求を受信すると、アプライアンスはそのドメインにバインドされている GSLB 仮想サーバーをチェックします。GSLB 仮想サーバーがドメインにバインドされている場合、DNS 応答の送信先として最適な IP アドレスが照会されます。

DNS VIP

DNS 仮想 IP は、NetScaler ADC アプライアンス上の負荷分散 DNS 仮想サーバーを表す仮想 IP（VIP）アドレスです。NetScaler ADC アプライアンスが権限を持つドメインの DNS 要求は、DNS VIP に送信できます。

GSLB の方式

August 15, 2023

構成されたサーバーの IP アドレスを単純に応答する従来の DNS サーバーとは異なり、GSLB 用に構成された NetScaler ADC アプライアンスは、構成された GSLB 方式によって決定されたサービスの IP アドレスで応答します。デフォルトでは、GSLB 仮想サーバーは最小の接続方法に設定されています。すべての GSLB サービスが停止した場合、アプライアンスは設定済みのすべての GSLB サービスの IP アドレスで応答します。

GSLB メソッドは、GSLB 仮想サーバーが最もパフォーマンスの高い GSLB サービスを選択するために使用するアルゴリズムです。Web アドレスのホスト名が解決されると、クライアントは解決されたサービス IP アドレスにトラフィックを直接送信します。

NetScaler ADC アプライアンスは、次の GSLB 方式を提供します。

- ラウンドロビン
- 最小接続数
- 最短応答時間
- 最小帯域幅
- 最小パケット
- 送信元 IP ハッシュ
- カスタムロード
- ラウンドトリップ時間（RTT）
- 静的近接

GSLB メソッドをリモートサイトで機能させるには、MEP を有効にするか、明示的モニターをリモートサービスにバインドする必要があります。MEP が無効の場合、RTT、最小接続、最小帯域幅、最小パケット、および最小応答時間の方法はデフォルトでラウンドロビンに設定されます。

静的近接法と RTT 負荷分散方式は GSLB に固有のものです。

静的近接またはダイナミック RTT 以外の GSLB 方式の指定

ラウンドロビン、最小接続、最小応答時間、最小帯域幅、最小パケット、送信元 IP ハッシュ、またはカスタムロード方法の詳細については、ロードバランシングを参照してください。

CLI を使用して **GSLB** メソッドを変更するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -lbMethod GSLBMethod
<!--NeedCopy-->
```

例**:**

```
set gslb vserver Vserver-GSLB-1 -lbMethod ROUNDROBIN
<!--NeedCopy-->
```

GUI を使用して **GSLB** メソッドを変更するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. 詳細ペインで **GSLB** 仮想サーバーを選択し、「開く」をクリックします。
3. 「GSLB 仮想サーバーの設定」ダイアログの「メソッドとパーシステンス」タブの「メソッド」で、「メソッドの選択」リストからメソッドを選択します。
4. 「**OK**」をクリックし、選択した方法が画面下部の「詳細」の下に表示されていることを確認します。

GSLB アルゴリズム

August 15, 2023

GSLB では次のアルゴリズムがサポートされています。

- ラウンドロビン:GSLB 仮想サーバーがラウンドロビン方式を使用するように構成されている場合、GSLB 仮想サーバーはそれにバインドされているサービスのリストを継続的にローテーションします。仮想サーバーは、要求を受信すると、リスト内の最初のサービスに接続を割り当て、そのサービスをリストの一番下に移動します。
- 最小応答時間:GSLB 仮想サーバーが最小応答時間方式を使用するように構成されている場合、最も低い値のサービスを選択します。ここで、最小値は、現在のアクティブな接続 X 平均応答時間。

  このメソッドは、HTTP サービスおよび Secure Sockets Layer (SSL) サービスにのみ設定できます。応答時間 (Time to First Byte、または TTFB とも呼ばれる) は、要求パケットをサービスに送信してからサービスから最初の応答パケットを受信するまでの時間間隔です。NetScaler アプライアンスは、応答コード 200 を使用して TTFB を計算します。
- 最小接続:GSLB 仮想サーバーが最小接続 GSLB アルゴリズム（またはメソッド）を使用するように構成されている場合、アクティブな接続が最も少ないサービスを選択します。これは、ほとんどの状況で最高のパフォーマンスを提供するため、デフォルトの方法です。

- 最小帯域幅**:** 最小帯域幅方式を使用するように構成された **GSLB** 仮想サーバーは、メガビット/秒 (Mbps) 単位で測定された、現在最もトラフィック量が少ないサービスを選択します。
- 最小パケット**:** 最小パケット方式を使用するように構成された **GSLB** 仮想サーバーは、過去 14 秒間に受信したパケット数が最も少ないサービスを選択します。
- ソース **IP** ハッシュ**:** ソース **IP** ハッシュ方式を使用するように構成された **GSLB** 仮想サーバーは、クライアントの IPv4 または IPv6 アドレスのハッシュ値を使用してサービスを選択します。特定のネットワークに属する送信元 IP アドレスからのすべての要求を特定の宛先サーバーに転送するには、送信元 IP アドレスをマスクする必要があります。IPv4 アドレスの場合は、NetMask パラメータを使用してください。IPv6 アドレスの場合は、v6NetMaskLength パラメーターを使用します。
- カスタム負荷: カスタム負荷分散は、CPU 使用率、メモリ、応答時間などのサーバーパラメータで実行されます。カスタムロード方式を使用する場合、NetScaler ADC アプライアンスは通常、アクティブなトランザクションを処理していないサービスを選択します。GSLB セットアップのすべてのサービスがアクティブなトランザクションを処理している場合、アプライアンスは負荷が最も小さいサービスを選択します。ロードモニタと呼ばれる特殊なタイプのモニタは、ネットワーク内の各サービスの負荷を計算します。負荷モニターはサービスの状態をマークしませんが、サービスが UP でない場合は GSLB の決定からサービスを取り出します。
- 静的近接性:GSLB は、IP アドレスベースの静的近接データベースを使用して、クライアントのローカル DNS サーバーと GSLB サイト間の近接性を判断します。NetScaler ADC アプライアンスは、近接基準に最も一致するサイトの IP アドレスで応答します。
- ラウンドトリップ時間:RTT は、クライアントのローカル DNS サーバーとデータリソース間のネットワークにおける時間または遅延の尺度です。NetScaler ADC アプライアンスは、クライアントのローカル DNS サーバーを調査し、RTT メトリック情報を収集します。次に、アプライアンスはこのメトリックを使用して負荷分散を決定します。グローバルサーバー負荷分散は、ネットワークのリアルタイムの状態を監視し、RTT 値が最も低いデータセンターにクライアント要求を動的に送信します。
- **API** メソッド:GSLB は REST API を使用して、最もパフォーマンスの高い GSLB サービスを決定します。API メソッドでは、GSLB がクライアントから DNS リクエストを受け取ると、指定されたルールに照らしてリクエストを評価します。

詳細については、「負荷分散」を参照してください。

静的近接度

August 15, 2023

GSLB の静的近接方法では、IP アドレスベースの静的近接データベースを使用して、クライアントのローカル DNS サーバーと GSLB サイト間の近接性を判断します。NetScaler ADC アプライアンスは、近接基準に最も一致するサイトの IP アドレスで応答します。

地理的に異なる場所にある 2 つ以上の GSLB サイトが同じコンテンツを配信している場合、NetScaler アプライアンスは IP アドレス範囲のデータベースを管理し、そのデータベースを使用して受信クライアント要求の送信先となる GSLB サイトを決定します。

静的近接方式が機能するには、ロケーションファイルから入力された既存の静的近接データベースを使用するように NetScaler アプライアンスを構成するか、静的近接データベースにカスタムエントリを追加する必要があります。カスタムエントリを追加したら、そのロケーション修飾子を設定できます。データベースを設定したら、GSLB メソッドとして静的な近接を指定する準備が整いました。

静的近接の設定の詳細については、静的近接の設定を参照してください。

動的往復時間方式

August 15, 2023

動的ラウンドトリップ時間 (RTT) は、クライアントのローカル DNS サーバーとデータリソース間のネットワークの時間または遅延の尺度です。動的 RTT を測定するために、NetScaler アプライアンスはクライアントのローカル DNS サーバーをプローブし、RTT メトリック情報を収集します。次に、アプライアンスはこのメトリックを使用して負荷分散を決定します。グローバルサーバー負荷分散は、ネットワークのリアルタイムの状態を監視し、RTT 値が最も低いデータセンターにクライアント要求を動的に送信します。

クライアントのドメインに対する DNS 要求が、そのドメインの権限のある DNS として構成された NetScaler アプライアンスに届くと、アプライアンスは RTT 値を使用して最もパフォーマンスの高いサイトの IP アドレスを選択し、DNS 要求への応答として送信します。

NetScaler アプライアンスは、ICMP エコー要求または応答（PING）、UDP、TCP などのさまざまなメカニズムを使用して、ローカル DNS サーバーと参加サイト間の接続の RTT メトリックを収集します。アプライアンスはまず Ping プローブを送信して RTT を決定します。ping プローブが失敗すると、DNS UDP プローブが使用されます。そのプローブにも障害が発生した場合、アプライアンスは DNS TCP プローブを使用します。

これらのメカニズムは NetScaler アプライアンスでは負荷分散モニターとして表示され、「ldns」プレフィックスが使用されているため簡単に識別できます。3 つのモニターは、デフォルトの順序で次のとおりです。

- `ldns-ping`
- `ldns-dns`
- `ldns-tcp`

これらのモニターはアプライアンスに組み込まれており、安全なデフォルトに設定されています。ただし、アプライアンス上の他のモニターと同様にカスタマイズできます。

デフォルトの順序は、GSLB パラメータとして明示的に設定することで変更できます。たとえば、DNS UDP クエリ、PING、TCP の順に順序を設定するには、次のコマンドを入力します。


```
set gslb parameter -ldnsprobeOrder DNS PING TCP
<!--NeedCopy-->
```

カスタマイズされていない限り、NetScaler アプライアンスはポート 53 で UDP と TCP のプロービングを実行しますが、通常の負荷分散モニターとは異なり、プローブが成功して有効な RTT 情報を提供する必要はありません。通常、障害と見なされる ICMP ポート使用不可メッセージ、TCP リセット、DNS エラー応答はすべて RTT 値の計算に使用できます。

RTT データがコンパイルされると、アプライアンスは独自のメトリック交換プロトコル (MEP) を使用して参加サイト間で RTT 値を交換します。RTT メトリックを計算した後、アプライアンスは RTT 値をソートして、RTT メトリックが最良（最小）のデータセンターを特定します。

RTT 情報が利用できない場合（たとえば、クライアントのローカル DNS サーバーが初めてサイトにアクセスしたときなど）、NetScaler アプライアンスはラウンドロビン方式を使用してサイトを選択し、クライアントをそのサイトに誘導します。

動的メソッドを設定するには、サイトの GSLB 仮想サーバーをダイナミック RTT 用に構成します。ローカル DNS サーバーがプローブされる間隔をデフォルト以外の値に設定することもできます。

ダイナミック **RTT** 用の **GSLB** 仮想サーバーの設定

GSLB 仮想サーバーをダイナミック RTT 用に設定するには、RTT 負荷分散方式を指定します。

NetScaler アプライアンスは、特定のローカルサーバーのタイミング情報を定期的に検証します。レイテンシーの変化が設定された許容係数を超えると、アプライアンスはデータベースを新しいタイミング情報で更新し、MEP 交換を実行して新しい値を他の GSLB サイトに送信します。デフォルトの許容係数は 5 ミリ秒 (ms) です。

RTT 許容係数は GSLB ドメイン全体で同じでなければなりません。サイトに合わせて変更する場合は、GSLB ドメインに展開されているすべての NetScaler アプライアンスで同じ RTT 許容係数を構成する必要があります。

コマンドラインインターフェイスを使用してダイナミック **RTT** 用の **GSLB** 仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -lbMethod RTT -tolerance <value>
<!--NeedCopy-->
```

例**:**

```
set gslb vserver Vserver-GSLB-1 -lbMethod RTT -tolerance 10
<!--NeedCopy-->
```

構成ユーティリティを使用してダイナミック **RTT** 用の **GSLB** 仮想サーバーを構成するには

トラフィック管理に移動します > **GSLB** > 仮想サーバー をクリックし、仮想サーバーをダブルクリックします。

ローカル **DNS** サーバーのプロービング間隔を設定

NetScaler アプライアンスは、ICMP エコー要求または応答（PING）、TCP、UDP などのさまざまなメカニズムを使用して、ローカル DNS サーバーと参加している GSLB サイト間の接続の RTT メトリックを取得します。デフォルトでは、アプライアンスは ping モニターを使用し、5 秒ごとにローカル DNS サーバーをプローブします。その後、アプライアンスは応答を 2 秒間待ちます。その時間に応答がない場合、TCP DNS モニターを使用してプロービングします。

ただし、ローカル DNS サーバーをプローブする時間間隔は、構成に合わせて変更できます。

コマンドラインインターフェイスを使用してプロービング間隔を変更するには

コマンドプロンプトで入力します。

```
set lb monitor <monitorName> <type> -interval <integer> <units> -
    resptimeout <integer> <units>
<!--NeedCopy-->
```

例**:**

```
set lb monitor ldns-tcp LDNS-TCP -interval 10 sec -resptimeout 5 sec
<!--NeedCopy-->
```

構成ユーティリティを使用してプロービング間隔を変更するには

[トラフィック管理] > [負荷分散] > [モニター] に移動し、変更するモニター (たとえば、ping) をダブルクリックします。

API メソッド

August 15, 2023

API メソッドを使用して、最もパフォーマンスの高い GSLB サービスを決定できます。GSLB の API メソッドは REST API を使用して、最もパフォーマンスの高い GSLB サービスを決定します。

API メソッドでは、GSLB がクライアントから DNS リクエストを受け取ると、指定されたルールに照らしてリクエストを評価します。GSLB は HTTP コールアウト表現 SYS.HTTP_CALLOUT (<name>) を検出すると、HTTP コールアウトエージェントへの REST API リクエストを呼び出します。GSLB は、HTTP コールアウトエージェントからの応答を使用して、最もパフォーマンスの高いサービスを決定します。DNS 応答では、GSLB は最もパフォーマンスの高いサービスの IP アドレスをクライアントに返します。

CLI を使用して GSLB API メソッドを設定するには

以下を実行して GSLB API メソッドを設定します。

1. HTTP コールアウトを設定します。

   詳細については、「HTTP コールアウトの設定」を参照してください。

   コマンドプロンプトで入力します。

```
add policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-
    port <port>] [-vServer <string>] [-returnType <returnType>] [-
    httpMethod (GET | POST)] [-hostExpr <string>] [-urlStemExpr <
    string>] [-headers <name(value)> ...] [-parameters <name(value)
    > ...] [-bodyExpr <string>] [-fullReqExpr <string>] [-scheme (
    http | https)] [-resultExpr <string>] [-cacheForSecs <secs>] [-
    comment <string>]
<!--NeedCopy-->
```

   例:

```
add policy httpCallout GSLB_Method_API -IPAddress 208.111.39.237 -
    port 443 -returnType TEXT -hostExpr  “\” hopx.gslb.com\ “”  -
    urlStemExpr  “\” /zones/1/customers/92395/apps/6/decision\ “”
    -headers Authorization( “Basic 19fbe6db-4332-4e3f-a8bc-
    ee47bdc726f8") -parameters ip(DNS.REQ.OPT.ECS.IP.
    TYPECAST_TEXT_T ALT CLIENT.IP.SRC.TYPECAST_TEXT_T) -scheme
    https -resultExpr  “HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH).
    XPATH_JSON(xp%/providers/Val[1]/provider%)”  -cacheForSecs 30
<!--NeedCopy-->
```

2. 負荷分散の API メソッドを指定します。GSLB は、指定されたルールに照らして DNS リクエストを評価します。

   コマンドプロンプトで入力します。

```
add gslb vserver <name> <serviceType>  [-lbMethod <lbMethod>] [-
    backupLBMethod <backupLBMethod>] -rule <expression>
<!--NeedCopy-->
```

   例:

```
add gslb vserver vs1 HTTP -lbMethod API -backupLBMethod ROUNDROBIN
     -rule  “sys.http_callout(GSLB_Method_API)”
<!--NeedCopy-->
```

LB メソッドとして API を使用して GSLB と ITM を統合するためのサンプル設定

この構成により、GSLB は Citrix のインテリジェントトラフィック管理（ITM）のインターネット可視性の側面を利用して、最もパフォーマンスの高い GSLB サービスを決定できます。


```
/* Enable ns features */

enable ns feature lb gslb cs

/* This is a named expression that is used in the HTTP callout, used
   for result expression. */

add policy expression exp1 "HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH).
   XPATH_JSON(xp%/providers/Val[1]/provider%)"

/* This is a named expression that is used in HTTP callout, used for
   host expression. */

add policy expression exp2 ""hopx.cedexis.com""

/* This is the HTTP callout configured to request the ITM for the GSLB
   decision. */

add policy httpCallout ITM_OpenMix_API -IPAddress 208.111.39.237 -port
   80 -returnType TEXT -hostExpr exp2 -urlStemExpr ""/zones/1/customers
   /61770/apps/3/decision"" -headers Authorization("Basic a310697a-1d69
   -48bf-8f36-55742a8e894e") -parameters ip(DNS.REQ.OPT.ECS.IP.
   TYPECAST_TEXT_T ALT CLIENT.IP.SRC.TYPECAST_TEXT_T) -scheme http -
   resultExpr exp1 -cacheForSecs 30

/* Add service 1 */
add service sg1 98.136.103.24 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
    -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -
   svrTimeout 360 -CKA NO -TCPB NO -CMP NO

/* Add service 2 */
add service sg2 172.217.194.113 HTTP 80 -gslb NONE -maxClient 0 -maxReq
    0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180
   -svrTimeout 360 -CKA NO -TCPB NO -CMP NO

/* Add ADNS service */

add service adns1 10.102.217.106 ADNS 53 -gslb NONE -maxClient 0 -
   maxReq 0 -cip DISABLED -usip NO -useproxyport NO -sp OFF -cltTimeout
    120 -svrTimeout 120 -CKA NO -TCPB NO -CMP NO

/* Add lb vserver 1 for service 1 */
add lb vserver lbvs1 HTTP 10.102.217.116 80 -persistenceType NONE -
   cltTimeout 180

/* Add lb vserver 2 for service 2 */
add lb vserver lbvs2 HTTP 10.102.217.117 80 -persistenceType NONE -
   cltTimeout 180

/* Bind service 1 to lb vserver 1 */

bind lb vserver lbvs1 sg1

```



```
/* Bind service 2 to lb vserver 2 */

bind lb vserver lbvs2 sg2

/* Configure API GSLB method on GSLB virtual server to call the HTTP
   callout. This HTTP callout requests the ITM for the GSLB decision
   and returns GSLB service name, which should serve the request. */

add gslb vserver vs1 HTTP -lbMethod API -backupLBMethod ROUNDROBIN -
   rule "sys.http_callout(ITM_OpenMix_API)" -tolerance 0 -ECS ENABLED

/* Add GSLB site */

add gslb site site1 10.102.217.106 -publicIP 10.102.217.106

/* Add GSLB service 1 */

add gslb service aws_ec2_ap_south_1_asia_pacific_mumbai_1
   10.102.217.116 HTTP 80 -publicIP 10.102.217.116 -publicPort 80 -
   maxClient 0 -siteName site1 -sitePersistence HTTPRedirect -
   sitePrefix gs2. -cltTimeout 180 -svrTimeout 360 -downStateFlush
   ENABLED

/* Add GSLB service 2 */

add gslb service aws_ec2_ap_south_1_asia_pacific_mumbai 10.102.217.117
   HTTP 80 -publicIP 10.102.217.117 -publicPort 80 -maxClient 0 -
   siteName site1 -sitePersistence HTTPRedirect -sitePrefix gs1. -
   cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

/* Bind the GSLB service 1 to GSLB server 1 */
bind gslb vserver vs1 -serviceName
   aws_ec2_ap_south_1_asia_pacific_mumbai_1

/* Bind the GSLB service 2 to GSLB server 2 */
bind gslb vserver vs1 -serviceName
   aws_ec2_ap_south_1_asia_pacific_mumbai

/* Bind a domain name to the GSLB virtual server */
bind gslb vserver vs1 -domainName testruchit104.com -TTL 5

<!--NeedCopy-->
```

静的近接度を構成する

August 15, 2023

静的近接方式が機能するには、ロケーションファイルから入力された既存の静的近接データベースを使用するように NetScaler アプライアンスを構成するか、静的近接データベースにカスタムエントリを追加する必要があります。カ

スタムエントリを追加したら、そのロケーション修飾子を設定できます。データベースを設定したら、GSLB メソッドとして静的な近接を指定する準備が整いました。

このドキュメントでは、次の内容について説明します。

- ロケーションファイルの追加による静的近接データベースの作成
- 静的近接データベースへのカスタムエントリの追加
- ロケーション修飾子の設定
- 近接方法の指定
- GSLB 静的近接データベースの同期

位置情報ファイルを追加して、静的近接データベースを作成する

October 25, 2023

静的近接データベースは UNIX ベースの ASCII ファイルです。ロケーションファイルからこのデータベースに追加されたエントリは、静的エントリと呼ばれます。NetScaler アプライアンスにロードできるロケーションファイルは 1 つだけです。新しい位置情報ファイルを追加すると、既存のファイルが上書きされます。静的近接データベースのエントリ数は、NetScaler アプライアンスに構成されているメモリによって制限されます。

静的近接データベースは、デフォルト形式または市販のサードパーティデータベース (`www.maxmind.com` や`www.ip2location.com`など) から派生した形式で作成できます。

NetScaler アプライアンスには、次の 2 つの IP ジオロケーションデータベースファイルが含まれています。これらは、MaxMind によって公開された GeoLite2 ファイルです。

- Citrix_Netscaler_InBuilt_GeoIP_DB_IPv4
- Citrix_Netscaler_InBuilt_GeoIP_DB_IPv6

これらのデータベースファイルは、NetScaler アプライアンスがサポートする形式の /var/netscaler/inbuilt_db ディレクトリにあります。

これらの IP ジオロケーションデータベースは、静的近接ベースの GSLB 方式のロケーションファイルとして、またはロケーションベースのポリシーで使用できます。

これらのデータベースは、提供する詳細が異なります。デフォルトのファイルには書式タグがある場合を除き、データベースファイル形式の厳密な適用はありません。データベースファイルは、フィールド区切り文字としてカンマを使用する ASCII ファイルです。フィールドの構造と、ロケーションの IP アドレスの表現には違いがあります。

format パラメータは、NetScaler アプライアンスに対するファイルの構造を記述します。format オプションに不正な値を指定すると、内部データが破損する可能性があります。

注:

- アップグレード後、/var/netscaler/inbuilt_db/ ディレクトリに以前の NetScaler ソフトウェアバージョンのデータベースファイル（Citrix_Netscaler_InBuilt_GeoIP_DB.csv）が含まれている場合、そのファイルは保持されます。
- データベースファイルのデフォルトの場所は/var/netscaler/locdb です。高可用性（HA）セットアップでは、ファイルの同一のコピーが両方の NetScaler アプライアンスの同じ場所に存在する必要があります。
- ロケーションファイルが既定の場所以外の場所に保存されている場合は、ロケーションファイルのパスを指定します。
- 管理パーティションのデフォルトパスは:/var/partitions/<partitionName>/netscaler/locdbです。
- 一部のデータベースでは、ISO-3166 に準拠した短い国名や長い国名も提供されています。NetScaler は、修飾子を格納および照合するときに短縮名を使用します。
- 静的近接データベースを作成するには、NetScaler アプライアンスの UNIX シェルにログオンし、エディターを使用して、NetScaler がサポートする形式のいずれかで場所の詳細を含むファイルを作成します。
- NetScaler アプライアンスには GeoLite2 データベース（IPv4 および IPv6）が付属していますが、NetScaler は MaxMind GeoLite2 データベースを定期的に保守または更新しません。必要に応じて、GeoLite2 データベースをhttps://www.maxmind.comから取得し、NetScaler データベース形式に変換できます。詳しくは、MaxMind GeoLite2 データベース形式を NetScaler データベース形式に変換するスクリプトを参照してください。

CLI を使用して静的ロケーションファイルを追加するには

コマンドプロンプトで入力します。

```
add locationFile <locationFile> [-format <format>]
-  show locationFile
<!--NeedCopy-->
```

例:

```
add locationFile /var/netscaler/locdb/nsgeo1.0 -format netscaler
Done

show locationFile
Location File: /var/netscaler/locdb/nsgeo1.0
Format: netscaler
 Done
>
<!--NeedCopy-->
```

例:


```
add locationFile /var/netscaler/inbuilt_db/
    Citrix_Netscaler_InBuilt_GeoIP_DB_IPv4 -format netscaler

add locationFile6 /var/netscaler/inbuilt_db/
    Citrix_Netscaler_InBuilt_GeoIP_DB_IPv6 -format netscaler
<!--NeedCopy-->
```

GUI を使用して静的ロケーションファイルを追加するには、次の手順を実行します。

1. [**AppExpert**] > [場所] に移動し、[静的データベース] タブをクリックします。
2. [追加] をクリックして、静的位置情報ファイルを追加します。

インポートしたロケーションファイルデータベースは、構成ユーティリティの [データベースの表示] ダイアログボックスを使用して表示できます。同等の CLI はありません。

GUI を使用して静的ロケーションファイルを表示するには、次の手順を実行します。

1. [**AppExpert**] > [場所] に移動し、[静的データベース] タブをクリックします。
2. 静的ロケーションファイルを選択し、[アクション] リストから [データベースの表示] をクリックします。

ロケーションファイルを **NetScaler** 形式に変換するには:

デフォルトでは、位置情報ファイルを追加すると、NetScaler 形式で保存されます。他の形式のロケーションファイルを NetScaler 形式に変換できます。

> 注:
>
> nsmapオプションには、コマンドラインインターフェイスからのみアクセスできます。この変換は、NetScaler 形式にのみ可能です。

静的データベース形式を変換するには、**CLI** プロンプトで次のコマンドを入力します。

```
nsmap -f <inputFileFormat> -o <outputFileName> <inputFileName>
<!--NeedCopy-->
```

例**:**

```
 nsmap -f ip-country-region-city -o nsfile.ns ip-country-region-city.
    csv
<!--NeedCopy-->
```

MaxMind GeoLite2 データベース形式を **NetScaler** データベース形式に変換するスクリプト

MaxMind GeoIP データベースは、NetScaler で直接使用することはできません。MaxMind GeoIP データベースは、NetScaler 形式に変換し、GSLB 静的近接方式やポリシーなどの他の機能で IP ロケーションを検出するためにロードする必要があります。
スクリプトを使用して、GeoLite2 データベース形式を NetScaler データベース形式に変換できます。このスクリプトは、IPv4 ファイルと IPv6 ファイルの両方を変換するために使用できます。

スクリプトは次の場所にあります。https://github.com/citrix/MaxMind-GeoIP-Database-Conversion-Citrix-ADC-Format

GeoIP2 データベースを **NetScaler** 形式に変換する手順

1. GeoLite2 City または GeoLite2 国データベースを.csv 形式でhttps://dev.maxmind.com/geoip/geoip2/geolite2/からダウンロードします。
2. NetScaler ディレクトリ（/var など）にファイルをコピーします。次のシェルコマンドを使用してファイルを解凍すると、同じ名前のディレクトリが作成されます。

   ```
   tar -xf <filename>
   ```

3. スクリプト convert_geoipdb_to_netscaler_format.pl をhttps://github.com/citrix/MaxMind-GeoIP-Database-Conversion-Citrix-ADC-Formatからダウンロードし、ステップ #2 で作成したディレクトリにコピーします。
4. スクリプトの実行に使用できるオプションを確認するには、以下のコマンドを実行します。

   ```
   perl Convert_GeoIPDB_To_Netscaler_Format.pl -help
   ```

   以下のようなさまざまなオプションを利用できます。

   - <filename> IPv4 出力ファイル。既定の出力ファイル名:Netscaler_Maxmind_GeoIP_DB_IPv4.csv
   - -p <filename> IPv6 出力ファイル。既定の出力ファイル名:Netscaler_Maxmind_GeoIP_DB_IPv6.csv
   - -logfile <filename> イベント/メッセージのリストを含むファイル
   - -debug すべてのメッセージを STDOUT に出力します。

5. 次のコマンドを実行して、GeoLite2 データベース形式を NetScaler データベース形式に変換します。

   ```
   perl Convert_GeoIPDB_To_Netscaler_Format.pl
   ```

   > 注:
   >
   > 操作には最大 5 分かかることがあります。

   スクリプトで使用される既定のファイル名は、MaxMind GeoLite2 City ベースのデータベースのファイル名です。GeoLite2 Country データベースをダウンロードした場合は、リストされているとおりに入力ファイル名を指定する必要があります。

   - -b <filename> 変換する IPv4 ブロックファイルの名前。既定のファイル名:GeoLite2-City-Blocks-IPv4.csv
   - -i <filename> 変換する IPv6 ブロックファイルの名前。既定のファイル名:GeoLite2-City-Blocks-IPv6.csv
   - -l <filename> 変換するロケーションファイルの名前。既定のファイル名:GeoLite2-City-Locations-en.csv

例:

```
perl Convert_GeoIPDB_To_Netscaler_Format.pl -b GeoLite2-City-Blocks-IPv4.csv -i GeoLite2-City-Blocks-IPv6.csv -l  GeoLite2-City-Locations-en.csv
<!--NeedCopy-->
```

スクリプトの実行後に生成される出力ファイルは次のとおりです。

- Netscaler_Maxmind_GeoIP_DB_IPv4.csv
- Netscaler_Maxmind_GeoIP_DB_IPv6.csv

6. データベースの NetScaler 形式への変換が完了したら、次のコマンドを使用して使用を開始します。

   ```
   add locationFile <locationFile>
   ```

NetScaler アプライアンスにサードパーティの静的データベースファイルを追加する

NetScaler アプライアンスにサードパーティの静的データベースファイルを追加するには、次の手順を実行します。

1. ロケーションデータベースファイルは、www.maxmind.comなどのサードパーティベンダーから入手してください。

   > 注:
   >
   > www.maxmind.comからロケーションデータベースファイルをダウンロードした場合は、すぐに利用できるスクリプトを使用して NetScaler データベース形式に変換できます。スクリプトの使用方法については、「MaxMind GeoLite2 データベース形式を NetScaler データベース形式に変換するスクリプト」を参照してください。
   >
   > 他のサードパーティベンダーからダウンロードしたロケーションデータベースの場合は、NetScaler アプライアンスに追加する前に NetScaler データベース形式に変換する必要があります。

2. 以下のコマンドを実行して、静的な位置情報ファイルを追加します。

   ```
   add location file <locationfile Name>
   <!--NeedCopy-->
   ```

   > 注:
   >
   > - ロケーションデータベースファイルがデフォルトの **/var/netscaler/locdb** ロケーションに配置されていない場合、<locationfile Name> にはファイル名とともにファイルの場所が含まれている必要があります。
   > - add location file <locationfile Name>コマンドを実行する前に、次の操作を行います。
   >   - Make sure that the location database file is present in one of the directories of the

> NetScaler appliance.
> - Run the `sync HA files` command on the high availability setup and the `sync cluster files` command in a cluster setup. These commands ensure that the location database file is copied to the secondary appliance of the high availability pair and peer nodes of the cluster.

3. 次のコマンドを実行して、ロケーションデータベースがロードされていることを確認します。

```
show locationParameter
<!--NeedCopy-->
```

このコマンドは、静的近接ベースのロードバランシングに関連するロケーションパラメータを表示します。最大 3M-1 (300 万から 1 を引いた) エントリをロードできます。データベースのロードが進行中の場合、コマンドは`Loading: In progress`を表示します。ロードが完了すると、コマンドは`Loading: Idle`を表示します。

4. 次のコマンドを実行して、GSLB サイトの場所を表示します。

```
show gslb service
<!--NeedCopy-->
```

> 注
>
> - データベースが正しくロードされると、GSLB サイトの場所がデータベースに自動的に入力されます。
> - アプライアンスの設定には、ロケーションファイルを 1 つだけ指定できます。
> - 着信 IP アドレスに一致するものが見つからない場合、要求はラウンドロビン方式を使用して処理されます。

5. 次のコマンドを実行して、アプライアンスで GSLB メソッドを設定します。

```
set gslb vserver GSLBVserverName -lbMethod MethodType
<!--NeedCopy-->
```

静的近接データベースにカスタムエントリを追加する

August 15, 2023

カスタムエントリは、近接データベースの静的エントリよりも優先されます。最大 3000 のカスタムエントリを追加できます。カスタムエントリの場合、省略された修飾子はすべてアスタリスク (*) で示し、修飾子の名前にピリオドまたはスペースが含まれている場合は、パラメータを二重引用符で囲みます。修飾子ごとに最初の 31 文字が評価されます。静的近接 GSLB 方式のサービスを選択するために、IP アドレス範囲の地理的位置の経度と緯度を指定することもできます。

コマンドラインインターフェイスを使用してカスタムエントリを追加するには

コマンドプロンプトで次のコマンドを入力して、静的近接データベースにカスタムエントリを追加し、構成を確認します。

```
add location < IPfrom> < IPto> <preferredLocation> [-longitude <integer
    >[-latitude <integer>]]
show location
<!--NeedCopy-->
```

例:

```
> add location 192.168.100.1 192.168.100.100 *.us.ca.mycity
Done
<!--NeedCopy-->
```

```
> show location
1) IP from 192.168.100.1    IP to 192.168.100.100
Continent.Country.REgion.City.ISP.Organization =
North America.us.ca.mycity.*.
Coordinated: Not specified
Done
<!--NeedCopy-->
```

カスタムエントリを追加するためのパラメータ

- 送信元 **IP** アドレス: ドット区切りの 10 進表記の範囲内の最初の IP アドレス。

  これは必須の議論です。

- 送信先 **IP** アドレス: ドット区切りの 10 進表記の範囲内の最後の IP アドレス。

  これは必須の議論です。

- ロケーション名: ドット表記の修飾子の文字列は、IP アドレス範囲の地理的位置を表します。各修飾子は、continent.country.region.city.isp.Organization のように、その前の修飾子よりも具体的です。たとえば、「na.us.ca.san Jose.att.Citrix」などです。

  これは必須の議論です。最大長: 197

  > 注:
  >
  > ドット (.) またはスペース () を含む修飾子は二重引用符で囲む必要があります。

- 経度: 度単位の数値は、IP アドレス範囲の地理的位置の経度を指定します。
  最大値:180

- 緯度: 度単位の数値は、IP アドレス範囲の地理的位置の緯度を指定します。
  最大値:180

> 注:
>
> 経度と緯度のパラメータは、静的近接 GSLB 方式のサービスを選択するために使用されます。指定されていない場合、その場所に指定された修飾子に基づいて選択されます。

構成ユーティリティを使用してカスタムエントリを追加するには

AppExpert > **Location** に移動し、「カスタムエントリ」タブをクリックして、カスタムエントリを追加します。

ロケーション修飾子の設定

August 15, 2023

静的近接の実装に使用されるデータベースには、GSLB サイトの場所があります。各ロケーションには IP アドレス範囲と、その範囲に対して最大 6 つの修飾子があります。修飾子はリテラル文字列で、実行時に所定の順序で比較されます。すべての場所には、少なくとも 1 つの修飾子が必要です。修飾子ラベルは、ユーザ定義である修飾子（コンテキスト）の意味を定義します。NetScaler ADC には 2 つの組み込みコンテキストがあります。

地理的コンテキスト。次の修飾子ラベルがあります。

- 予選 1 —「大陸」
- 予選 2 —「国」
- 修飾子 3 —「状態」
- 予選 4 —「都市」
- 予選 5 —「ISP」
- 予選 6 —「組織」

カスタムエントリ。次の修飾子・ラベルがあります。

- 予選 1 —「予選 1」
- 修飾子 2 —「修飾子 2」
- 予選 3 —「予選 3」
- 修飾子 4 —「修飾子 4」
- 修飾子 5 —「修飾子 5」
- 修飾子 6 —「修飾子 6」

地理コンテキストが [大陸] 修飾子なしで設定されている場合、[大陸] は [国] から派生します。組み込みの修飾子ラベルもコンテキストに基づいており、ラベルは変更できます。これらの修飾子ラベルは、静的近接性の決定に使用される IP アドレスでマッピングされる場所を指定します。

静的近接ベースの決定を実行するために、NetScaler ADC アプライアンスは、ローカル DNS サーバーリゾルバーの IP アドレスから派生した場所属性（修飾子）を、参加サイトの場所属性と比較します。一致するサイトが 1 つだけの

場合、アプライアンスはそのサイトの IP アドレスを返します。複数の一致がある場合、選択されたサイトは、一致する GSLB サイトでのラウンドロビンの結果です。一致するものがない場合、選択されたサイトは、すべての構成済みサイトでのラウンドロビンの結果です。修飾子がないサイトは一致と見なされます。

ロケーションベースのポリシー式の GEO 規則を使用すると、ワイルドカードの一致を確認できます。この機能は、ワイルドカード修飾子がワイルドカード以外の修飾子を含む他の修飾子と一致するかどうかをチェックします。ワイルドカード照合は、`set locationParameter`コマンドに追加された`matchWildcardtoany`属性を使用して実行されます。

`matchWildcardtoany`属性は、次の値に設定できます。

- はい**:** ワイルドカード修飾子は他の修飾子と同じです。
- いいえ**:** ワイルドカード修飾子は非ワイルドカード修飾子とは一致しませんが、他のワイルドカード修飾子と一致します。デフォルトのオプションは [いいえ] です。
- 式**:** 式のワイルドカード修飾子は LDNS ロケーションのどの修飾子とも一致しますが、LDNS ロケーションのワイルドカード修飾子は式のワイルドカード以外の修飾子とは一致しません。

例:

```
add dns policy1 "CLIENT.IP.SRC.MATCHES_LOCATION("Continent.country
    \*.\*.\*.\* \ “)”  <action>
<!--NeedCopy-->
```

CLI を使用してロケーションパラメータを設定するには

コマンドプロンプトで入力します。

```
set locationparameter -context <context> -q1label <string> [-q2label <
    string>] [-q3label <string>] [-q4label <string>] [-q5label <string>]
     [-q6label <string>] -matchWildcardtoany [Yes | No | Expression]
<!--NeedCopy-->
```

例**:**

```
set locationparameter -context custom -q1label asia -matchWildcardtoany
    Yes
<!--NeedCopy-->
```

GUI を使用してロケーションパラメータを設定するには

1. [トラフィック管理**] > [GSLB] > [**データベースとエントリ] に移動します。
2. [設定] で、[位置パラメータの変更] をクリックします。
3. **[**ロケーションパラメータの設定**]** ページで、ロケーションパラメータを設定します。

設定例 **(CLI** を使用**)**

次のネットワーク設定について考えてみます。

- GSLB 仮想サーバー名:gv1
- GSLB 仮想サーバー IP アドレス:1.1.1.2
- GSLB サービス:gsvc1 が gv1 にバインドされました
- ロケーションデータベースファイル名:sample.csv
- 位置情報修飾子: 修飾子 1 と 2 が設定されます。残りはワイルドカードと一致するように設定されます。
  - 予選 1 —アジア
  - 修飾子 2 —IR
  - 予選 3-*
  - 予選 4-*
  - 予選 5-*
  - 予選 6-*
- DNS Policy: ポリシーの pol1 は、一致するパケットがあればパケットをドロップするように設定されています。

location パラメータを設定し、DNS ポリシーを次のように設定します。

```
set locationParameter -q2label Country_Code -q3label Subdivision_1_Name
    -q4label Subdivision_2_Name -q5label City

add locationFile "/var/netscaler/inbuilt_db/sample.csv"

add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0

add dns policy pol1 "CLIENT.IP.SRC.MATCHES_LOCATION("Asia.IR
   .\*.\*.\*.\*")||CLIENT.IP.SRC.MATCHES_LOCATION("Asia.SY.\*.\*.\*.\*"
   )||CLIENT.IP.SRC.MATCHES_LOCATION("Asia.SD.\*.\*.\*.\*")||CLIENT.IP.
   SRC.MATCHES_LOCATION("Asia.KP.\*.\*.\*.\*")||CLIENT.IP.SRC.
   MATCHES_LOCATION("North America.CU.\*.\*.\*.\*")||CLIENT.IP.SRC.
   MATCHES_LOCATION("Europe.UA.Crimea.\*.\*.\*.\*")"
   dns_default_act_Drop

bind dns global pol1 1 -gotoPriorityExpression 65535 -type REQ_DEFAULT

add gslb service gsvc1 1.1.1.2 HTTP 80 -publicIP 1.1.1.2 -publicPort 80
    -maxClient 0 -healthMonitor NO -siteName s1 -cltTimeout 180 -
   svrTimeout 360 -downStateFlush ENABLED

bind gslb vserver gv1 -serviceName gsvc1

bind gslb vserver gv1 -domainName www.gslbnew.com -TTL 5
<!--NeedCopy-->
```

ロケーション DB ファイルに次のクライアントエントリを追加します。この例では、ロケーション DB ファイル名は

sample.csv です。

```
10.106.24.170,10.106.24.190,,,,,,,8.0000,47.0000

10.102.82.170,10.102.82.190,Asia,,,,,,-73.9924,40.7553

10.106.24.140,10.106.24.150,,IR,,,,,51.4231,35.6961
<!--NeedCopy-->
```

前述の設定によると、10.106.24.170 ～10.106.24.190 の間のクライアントには、ワイルドカード修飾子が定義されていません。10.106.24.140 と 10.106.24.150 の間のクライアントには、IR として修飾子 2 があります。

match ワイルドカード修飾子を NO に設定します。

```
set locationparameter -matchWildcardtoany no
<!--NeedCopy-->
```

match ワイルドカード修飾子が NO に設定されている場合、ワイルドカード修飾子は定義されたワイルドカード修飾子にのみ一致します。他のワイルドカード以外の修飾子には一致しません。

- 10.106.24.147 の DNS クエリは、定義されているワイルドカード修飾子（修飾子 2 = IR）に一致します。そのため、DNS ポリシーが有効になり、クエリが破棄されます。

  10.106.24.147 クライアントでこの`dig @10.102.82.13 www.gslbnew.com`コマンドを実行すると、サーバに到達できなかったことが出力に表示されます。

  ```
  root@ns#  dig @10.102.82.13 www.gslbnew.com

  ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
  ; (1 server found)
  ;; global options: +cmd
  ;; connection timed out; no servers could be reached
  <!--NeedCopy-->
  ```

- 10.106.24.180 から送信された DNS クエリが、定義された修飾子と一致しません。DNS ポリシーは有効にならず、クエリが処理されます。

  10.106.24.180 クライアントで`dig @10.102.82.13 www.gslbnew.com`コマンドを実行します。出力には、GSLB 仮想サーバーの IP アドレスが表示されます。

  ```
  root@ns# dig @10.102.82.13 www.gslbnew.com

  ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
  ; (1 server found)
  ;; global options: +cmd
  ;; Got answer:
  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64265
  ;; flags: qr aa rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
     ADDITIONAL: 1
  ;; WARNING: recursion requested but not available

  ;; OPT PSEUDOSECTION:
  ```



```
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;www.gslbnew.com. IN    A

;; ANSWER SECTION:
www.gslbnew.com.    5   IN  A   1.1.1.2

;; Query time: 12 msec
;; SERVER: 10.102.82.13#53(10.102.82.13)
;; WHEN: Tue Mar 29 22:46:40 UTC 2022
;; MSG SIZE  rcvd: 60
<!--NeedCopy-->
```

match ワイルドカード修飾子を Yes に設定します。

```
set locationparameter -matchWildcardtoany yes
<!--NeedCopy-->
```

match ワイルドカード修飾子が yes に設定されている場合、ワイルドカード修飾子は任意のワイルドカード修飾子(定義済みおよび非ワイルドカード修飾子) に一致します。

- 10.106.24.147 の DNS クエリは、定義された修飾子（修飾子 2 = IR）に一致します。そのため、DNS ポリシーが有効になり、クエリが破棄されます。

  10.106.24.147 クライアントで`dig @10.102.82.13 www.gslbnew.com`コマンドを実行します。出力には、サーバが到達不能だったことが示されています。

  ```
  root@ns#  dig @10.102.82.13 www.gslbnew.com

  ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
  ; (1 server found)
  ;; global options: +cmd
  ;; connection timed out; no servers could be reached
  <!--NeedCopy-->
  ```

- 10.106.24.180 からのクエリは、ワイルドカード以外の修飾子と一致します。そのため、DNS ポリシーが有効になり、クエリが破棄されます。

  10.106.24.180 クライアントで`dig @10.102.82.13 www.gslbnew.com`コマンドを実行します。出力には、サーバが到達不能だったことが示されています。

  ```
  root@ns# dig @10.102.82.13 www.gslbnew.com

  ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
  ; (1 server found)
  ;; global options: +cmd
  ;; connection timed out; no servers could be reached
  <!--NeedCopy-->
  ```

match ワイルドカード修飾子を [式:] に設定します。


```
set locationparameter -matchWildcardtoany expression
<!--NeedCopy-->
```

match ワイルドカード修飾子が expression に設定されている場合、ワイルドカード修飾子は DNS ポリシーで使用可能な修飾子、またはロケーション DB ファイルで使用可能な修飾子のいずれかに一致します。

- 10.106.24.147 の DNS クエリは、DNS ポリシーで定義されているワイルドカード修飾子と一致します。そのため、DNS ポリシーが有効になり、クエリが破棄されます。

  10.106.24.147 クライアントで`dig @10.102.82.13 www.gslbnew.com`コマンドを実行します。出力には、サーバが到達不能だったことが示されています。

  ```
  root@ns#  dig @10.102.82.13 www.gslbnew.com

  ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
  ; (1 server found)
  ;; global options: +cmd
  ;; connection timed out; no servers could be reached
  <!--NeedCopy-->
  ```

- 10.106.24.180 から送信されたクエリが、DNS ポリシーの修飾子と一致しません。そのため、DNS ポリシーは有効にならず、クエリが処理されます。

  10.106.24.180 クライアントで`dig @10.102.82.13 www.gslbnew.com`コマンドを実行します。出力には、GSLB 仮想サーバーの IP アドレスが表示されます。

  ```
  root@ns# dig @10.102.82.13 www.gslbnew.com

  ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
  ; (1 server found)
  ;; global options: +cmd
  ;; Got answer:
  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64265
  ;; flags: qr aa rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
     ADDITIONAL: 1
  ;; WARNING: recursion requested but not available

  ;; OPT PSEUDOSECTION:
  ; EDNS: version: 0, flags:; udp: 1280
  ;; QUESTION SECTION:
  ;www.gslbnew.com.  IN   A

  ;; ANSWER SECTION:
  www.gslbnew.com.    5   IN  A   1.1.1.2

  ;; Query time: 12 msec
  ;; SERVER: 10.102.82.13#53(10.102.82.13)
  ;; WHEN: Tue Mar 29 22:46:40 UTC 2022
  ;; MSG SIZE  rcvd: 60
  <!--NeedCopy-->
  ```


近接方式を指定する

August 15, 2023

静的近接データベースを設定したら、GLSB 方式として静的近接を指定する準備が整います。

コマンドラインインターフェイスを使用して静的近接性を指定するには

コマンドプロンプトで次のコマンドを入力して静的近接を設定し、構成を確認します。

```
set gslb vserver <name> -lbMethod STATICPROXIMITY
show gslb vserver <name>
<!--NeedCopy-->
```

例**:**

```
set gslb vserver Vserver-GSLB-1 -lbMethod STATICPROXIMITY
show gslb vserver
<!--NeedCopy-->
```

構成ユーティリティを使用して静的近接を指定するには

1. [トラフィック管理] > [GSLB] > [仮想サーバー] に移動し、仮想サーバーをダブルクリックします。
2. 「メソッド」セクションをクリックし、「メソッドの選択」ドロップダウンリストから「**STATICPROXIMITY**」を選択します。

GSLB 静的近接データベースを同期する

August 15, 2023

グローバルサーバー負荷分散 (GSLB) 静的近接データベースを同期するには、いずれかのサイトをマスター GSLB ノードとして識別する必要があります。トポロジ内の任意のサイトをマスターノードとして指定できます。残りの GSLB ノードは自動的にスレーブノードとして指定されます。

GSLB 静的近接データベースを同期すると、/var/netscaler/locdb ディレクトリ内のファイルがスレーブノード間で同期されます。同期プロセス中、マスターノードは各スレーブノードから実行構成を取得し、それをマスターノードの設定と比較します。マスター GSLB ノードは rsync プログラムを使用して、スレーブノード間で静的近接データベースを同期します。同期処理を高速化するために、rsync プログラムは 2 つのファイル間の相違点をなくすだけの変更を行います。同期処理はロールバックできません。

次の例では、スレーブサイトである Site2 をマスターサイト Site1 に同期します。管理者はサイト 1 で **sync gslb config** コマンドを入力します。

```
sync gslb config -nowarn
Sync Time: Feb 24 2014 14:56:16
Retrieving local site info: ok
Retrieving all participating gslb sites info:
0 bytes in 0 blocks
ok
site1[Master]:
        Getting Config: ok
site2[Slave]:
        Syncing gslb static proximity database: ok
        Getting Config: ok
        Comparing config: ok
        Applying changes: ok
 Done
<!--NeedCopy-->
```

サイト間通信を構成する

August 15, 2023

GSLB サイト間通信は、通信サイトに関連付けられているリモートプロシージャコール (RPC) ノード間で行われます。マスター GSLB サイトはスレーブサイトとの接続を確立して、GSLB 構成情報を同期し、サイトメトリックを交換します。

RPC ノードは GSLB サイトの作成時に自動的に作成され、内部で生成されたユーザー名とパスワードが割り当てられます。NetScaler アプライアンスは、接続確立時にこのユーザー名とパスワードを使用してリモート GSLB サイトへの認証を行います。RPC ノードには設定手順は必要ありませんが、任意のパスワードを指定したり、GSLB サイトが交換する情報を暗号化してセキュリティを強化したり、RPC ノードのソース IP アドレスを指定したりできます。

アプライアンスが他の GSLB サイトと通信するときにソース IP アドレスとして使用するには、NetScaler が所有する IP アドレスが必要です。デフォルトでは、RPC ノードはサブネット IP (SNIP) アドレスのいずれかを使用しますが、任意の IP アドレスを指定することもできます。

次のトピックでは、NetScaler アプライアンス上の RPC ノードの動作と構成について説明します。

RPC ノードのパスワードの変更

Citrix では、各 RPC ノードのパスワードを変更して、GSLB 設定内のサイト間の通信を保護することをお勧めします。ローカルサイトの RPC ノードのパスワードを変更したら、その変更を各リモートサイトの RPC ノードに手動で伝達する必要があります。

パスワードは暗号化された形式で保存されます。show RpcNode コマンドを使用して、変更前と変更後の暗号化形式のパスワードを比較することで、パスワードが変更されたことを確認できます。

> **注:** GSLB は内部ユーザーアカウントを使用します。セキュリティを強化するために、Citrix では内部ユーザーアカウントのパスワードも変更することをお勧めします。内部ユーザーアカウントのパスワードは、RPC ノードパスワードによって変更されます。

コマンドラインインターフェイスを使用して **RPC** ノードのパスワードを変更するには

コマンドラインで次のコマンドを入力して、RPC ノードのパスワードを変更します。

```
set ns rpcNode <IPAddress> {
 -password }

show ns rpcNode
<!--NeedCopy-->
```

例:

```
> set rpcNode 192.0.2.4 -password mypassword
 Done
> show rpcNode
.
.
.
2)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
    SrcIP:  *                Secure:  OFF
Done
>

<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **RPC** ノードのパスワードを設定解除するには

CLI を使用して RPC ノードのパスワードを設定解除するには、unset RPCNode コマンド、RPC ノードの IP アドレス、およびパスワードパラメータを値なしで入力します。

構成ユーティリティを使用して **RPC** ノードのパスワードを変更するには

[システム] > [ネットワーク] > [RPC] に移動し、RPC ノードを選択し、パスワードを変更します。

サイトメトリクスの交換を暗号化

GSLB 設定で RPC ノードにセキュアオプションを設定することで、GSLB サイト間で交換される情報を保護できます。Secure オプションを設定すると、NetScaler アプライアンスはノードから他の RPC ノードに送信されるすべ

ての通信を暗号化します。

コマンドラインインターフェイスを使用してサイトメトリックの交換を暗号化するには

コマンドプロンプトで次のコマンドを入力して、サイトメトリックの交換を暗号化し、構成を確認します。

```
set ns rpcNode <IPAddress> [-secure ( YES | NO )]
show rpcNode
<!--NeedCopy-->
```

例:

```
> set rpcNode 192.0.2.4 -secure YES
 Done
>
> show rpcNode
.
.
.
3)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e SrcIP:
    192.0.2.3     Secure:  ON
 Done
>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセキュアパラメータの設定を解除するには

CLI を使用してセキュアパラメータの設定を解除するには、unset RPCNode コマンド、RPC ノードの IP アドレス、およびセキュアパラメータを値なしで入力します。

NetScaler 構成ユーティリティを使用してサイトメトリックの交換を暗号化するには

1. [システム] > [ネットワーク] > [RPC] に移動し、RPC ノードをダブルクリックします。
2. 「セキュア」オプションを選択し、「**OK**」をクリックします。

RPC ノードの送信元 **IP** アドレスの設定

デフォルトでは、NetScaler アプライアンスは RPC ノードのソース IP アドレスとして NetScaler 所有のサブネット IP（SNIP）アドレスを使用しますが、特定の SNIP アドレスを使用するようにアプライアンスを構成できます。SNIP アドレスが利用できない場合、GSLB サイトは他のサイトと通信できません。このようなシナリオでは、NSIP アドレスまたは仮想 IP (VIP) アドレスを RPC ノードの送信元 IP アドレスとして設定する必要があります。RPC ノードがリモートノードである場合にのみ、VIP アドレスを RPC ノードの送信元 IP アドレスとして使用できます。送信元 IP アドレスとして VIP アドレスを設定し、その VIP アドレスを削除すると、アプライアンスは SNIP アドレスを使用します。

> 注
>
> NetScaler 11.0.64.x 以降のリリースでは、GSLB サイト IP アドレスを RPC ノードのソース IP アドレスとして使用するようにアプライアンスを構成できます。

コマンドラインインターフェイスを使用して **RPC** ノードのソース **IP** アドレスを指定するには

コマンドプロンプトで次のコマンドを入力して RPC ノードのソース IP アドレスを変更し、構成を確認します。

```
set ns rpcNode <IPAddress> [-srcIP <ip_addr|ipv6_addr|*>]
show ns rpcNode
<!--NeedCopy-->
```

例**:**

```
set rpcNode 192.0.2.4 -srcIP 192.0.2.3
Done
show rpcNode
<!--NeedCopy-->
```

```
IPAddress: 192.0.2.4 Password: d336004164d4352ce39e SrcIP: 192.0.2.3
    Secure: OFF
Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して送信元 **IP** アドレスパラメータを設定解除するには

CLI を使用してソース IP アドレスパラメータを設定解除するには、unset RPCNodeCommand、RPC ノードの IP アドレス、および SrcIP パラメータを値なしで入力します。

NetScaler 構成ユーティリティを使用して **RPC** ノードのソース **IP** アドレスを指定するには

1. [システム] > [ネットワーク] > [RPC] に移動し、RPC ノードをダブルクリックします。
2. 「送信元 IP アドレス」フィールドに、RPC ノードが送信元 IP アドレスとして使用する IP アドレスを入力し、「OK」をクリックします。

> 重要
>
> RPC ノードのソース IP アドレスは各 NetScaler アプライアンスに固有であるため、ソース IP アドレスを GSLB に参加しているサイト間で同期することはできません。そのため、(sync gslb config –ForceSync コマンドを使用するか、GUI で ForceSync オプションを選択して) 同期を強制した後は、他の NetScaler アプライアンスのソース IP アドレスを手動で変更する必要があります。

メトリック交換プロトコルを構成する

August 15, 2023

GSLB セットアップのデータセンターは、NetScaler アプライアンスの独自のプロトコルであるメトリック交換プロトコル（MEP）を介してメトリックを相互に交換します。メトリック情報の交換は、GSLB サイトの作成時に開始されます。これらのメトリックには、負荷、ネットワーク、および持続性情報が含まれます。

MEP は、データセンターの可用性を確認するためのヘルスチェックに必要です。ネットワークメトリック（ラウンドトリップ時間）を交換するための接続は、交換に関係するどのデータセンターでも開始できますが、サイトメトリックを交換するための接続は、常に IP アドレスの小さいデータセンターによって開始されます。デフォルトでは、データセンターはサブネット IP アドレス (SNIP) を使用して別のデータセンターの IP アドレスへの接続を確立します。ただし、メトリック交換の送信元 IP アドレスとして、特定の SNIP、仮想 IP (VIP) アドレス、または NSIP アドレスを設定できます。GSLB サイト間の通信プロセスでは TCP ポート 3011 または 3009 が使用されるため、このポートは NetScaler ADC アプライアンス間のファイアウォールで開く必要があります。

注:SNIP または GSLB サイトの IP アドレスは、メトリック交換のソース IP アドレスとして設定できます。詳細については、「RPC ノードの送信元 IP アドレスの構成」を参照してください。

送信元サイトとターゲットサイト（MEP 接続を開始するサイト、および接続要求を受信するサイト）にプライベート IP アドレスとパブリック IP アドレスの両方が設定されている場合、サイトはパブリック IP アドレスを使用して MEP 情報を交換します。

「GSLB サービスのモニタリング」の説明に従って、モニターをバインドしてリモートサービスの状態をチェックすることもできます。モニタがバインドされている場合、メトリック交換はリモートサービスの状態を制御しません。モニターがリモートサービスにバインドされていて、メトリック交換が有効になっている場合、モニターはヘルスステータスを制御します。モニターをリモートサービスにバインドすることで、NetScaler アプライアンスは NetScaler 以外の負荷分散デバイスと通信できるようになります。NetScaler アプライアンスは NetScaler 以外のデバイスを監視できますが、モニターがすべての GSLB サービスにバインドされ、静的負荷分散方式（ラウンドロビン、静的近接、ハッシュベースの方法など）のみを使用しない限り、負荷分散を実行できません。

NetScaler リリース 11.1.51.x 以降では、サービスの不必要な中断を避けるために、MEP 接続がダウンしたときに GSLB サービスをダウンとしてマークする時間遅延を設定できます。

高可用性セットアップでの **MEP** の状態

高可用性セットアップでは、プライマリノードはリモートサイトとの接続を確立し、MEP の状態はプライマリノードからセカンダリノードに同期されません。そのため、セカンダリノードの MEP 状態は DOWN のままです。セカンダリノードがプライマリになると、新しい GSLB サイトと MEP 接続を確立し、それに応じて MEP 状態を更新します。

サイトメトリック交換を有効にする

GSLB サイト間で交換されるサイトメトリックには、各負荷分散またはコンテンツスイッチング仮想サーバーのステータス、現在の接続数、現在のパケットレート、および現在の帯域幅使用情報が含まれます。

NetScaler アプライアンスは、サイト間の負荷分散を実行するためにこの情報を必要とします。サイトメトリック交換間隔は 1 秒です。リモートサービスとサイトメトリックを交換できるようにするには、リモート GSLB サービスをローカル GSLB 仮想サーバーにバインドする必要があります。

コマンドラインインターフェイスを使用してサイトメトリックス交換を有効または無効にするには

コマンドプロンプトで次のコマンドを入力してサイトメトリック交換を有効または無効にし、構成を確認します。

```
set gslb site <siteName> -metricExchange (ENABLED|DISABLED)
show gslb site** <siteName>
<!--NeedCopy-->
```

例**:**

```
set gslb site Site-GSLB-East-Coast -metricExchange ENABLED
set gslb site Site-GSLB-East-Coast -metricExchange DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

GUI を使用してサイトメトリック交換を有効または無効にするには

1. [トラフィック管理] > [**GSLB**] > [サイト**]** に移動し、サイトを選択します。
2. 「**GSLB** サイトの設定」ダイアログボックスで、「メトリック交換」オプションを選択します。

ネットワークメトリック交換を有効にする

GSLB サイトがラウンドトリップタイム（RTT）負荷分散方式を使用している場合は、クライアントのローカル DNS サービスに関する RTT 情報の交換を有効または無効にできます。この情報は 5 秒ごとに交換されます。

GSLB メソッドを RTT に基づくメソッドに変更する方法の詳細については、GSLB メソッドを参照してください。

コマンドラインインターフェイスを使用してネットワークメトリック情報の交換を有効または無効にするには

コマンドプロンプトで次のコマンドを入力して、ネットワークメトリック情報の交換を有効または無効にし、構成を確認します。

```
set gslb site <siteName> -nwmetricExchange (ENABLED|DISABLED)
show gslb site <<siteName>
<!--NeedCopy-->
```


例:

```
set gslb site Site-GSLB-East-Coast -nwmetricExchange ENABLED
set gslb site Site-GSLB-East-Coast -nwmetricExchange DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

GUI を使用してネットワークメトリック情報の交換を有効または無効にするには

1. **[**** トラフィック管理] > [**GSLB**] > [サイト] に移動します。**
2. 「**GSLB** サイトの設定」ダイアログボックスで、「ネットワークメトリック交換」オプションを選択します。

MEP 接続がダウンしたときに **GSLB** サービスが **DOWN** とマークされるまでの時間遅延の設定

リモートサイトへの MEP 接続の状態が DOWN に変わると、そのリモートサイト上のすべての GSLB サービスのステータスは DOWN とマークされます。ただし、サイトが実際は DOWN ではない場合もあります。

サイトが DOWN とマークされる前に MEP 接続が再確立されるまでの遅延を設定できるようになりました。遅延の期限が切れる前に MEP 接続が復旧していれば、サービスに影響はありません。

たとえば、遅延を 10 に設定した場合、MEP 接続が 10 秒間停止するまでは GSLB サービスは停止中とマークされます。MEP 接続が 10 秒以内にアップ状態に戻ると、GSLB サービスはアップ状態のままになります。

注: この遅延は、モニターにバインドされていないサービスにのみ適用されます。遅延はトリガーモニターには影響しません。

コマンドラインインターフェイスを使用して時間遅延を設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
set gslb parameter** - GSLBSvcStateDelayTime <sec>
<!--NeedCopy-->
```

例:

gslb パラメータを設定**-GSLBSVC** ステートディレイタイム 10

> 注
>
> 階層型展開 (親子トポロジー) で、親サイトと子サイトの両方で GSLB サービスを設定する場合は、親サイトと子サイトの両方で GSLB パラメータを設定します。子サイトで GSLB サービスを設定しない場合は、親サイトでのみ GSLB パラメータを設定してください。

GUI を使用して時間遅延を設定するには

1. [設定] > [トラフィック管理] > [**GSLB**] > [**GSLB** 設定の変更] に移動します。
2. [**GSLB** サービス状態遅延時間 **(**秒**)**] ボックスに、遅延時間を秒単位で入力します。

GSLB サービスのフラップを回避するために、**MEP** 接続ステータスがアップしたときの **GSLB** サービスの学習時間を設定します

ノードが再起動するとき、または HA フェールオーバー中にシステムが初期化されます。次に、ノードは MEP を介してサービス状態をリモートノードに伝達するために、設定されたローカルおよび子サービスに関する最新情報を学習する必要があります。ノードは正しい情報を学習するのに少し時間がかかります。一方、ピアノードがこのノードに接続して更新を要求すると、ノードが誤ったサービス状態と統計を送信する可能性があります。この誤った情報は、リモートピアノードでサービスフラップやその他の機能に関連する問題を引き起こす可能性があります。このシナリオを回避するために、ローカルおよび子の GSLB サービスの学習時間を設定できるようになりました。

学習タイムアウトを設定すると、GSLB サイトはローカルサービスおよび子サービスに関する正しい統計情報を学習するためのバッファ時間（ラーニングタイムアウト）を取得します。サービスがラーニングフェーズにあるとき、リモート GSLB サイトは MEP アップデートでこの情報を取得し、そのサービスについて MEP を通じて受信したプライマリサイトの状態と統計情報を尊重しません。

GSLB サービスは、次のいずれかのシナリオで学習フェーズに入ります。

- NetScaler ADC アプライアンスが再起動されました
- 高可用性フェイルオーバーが発生しました
- クラスタ GSLB セットアップの所有者ノードが変更されました
- MEP がローカルノードで有効になっている
- GSLB のサイトは島のシナリオから出てくる。GSLB サイトは、他のサイトに接続されていない場合、アイランドになります。

親子展開では、プライマリ親がダウンすると、バックアップの親（構成されている場合）は、採用された子サイトの GSLB サービスを学習フェーズに選択的に移動します。

CLI を使用してサービス状態の学習時間を設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
set gslb parameter - SvcStateLearningTime <sec>
<!--NeedCopy-->
```

“SvcStateLearningTime” は秒単位で設定できます。デフォルト値は 0 で、最大値は 3600 です。このパラメーターは、モニターが GSLB サービスにバインドされていない場合にのみ適用されます。

例**:**


```
set gslb parameter  -  SvcStateLearningTime 10
<!--NeedCopy-->
```

GUI を使用してサービス状態の学習時間を設定するには

1. [構成] **>** [トラフィック管理] **>** [**GSLB**] **>** [ダッシュボード] **>** [**GSLB** 設定の変更] に移動します。

   [**GSLB** パラメータの設定] ページが表示されます。

2. [**GSLB** サービス状態の学習時間 **(**秒**)**] フィールドに、学習時間を秒単位で入力します。

パーシステンス情報交換の有効化

NetScaler アプライアンスを永続的な接続を提供するように構成できます。これにより、グループ内の任意の仮想サーバーへのクライアント転送を、同じクライアントから以前に送信を受信したサーバーに転送できます。

各サイトでのパーシスタンス情報の交換を有効または無効にできます。この情報は、GSLB に参加している NetScaler ADC アプライアンス間で 5 秒に 1 回交換されます。

パーシステンスの設定の詳細については、永続接続の設定を参照してください。

コマンドラインインターフェイスを使用してパーシステンス情報の交換を有効または無効にするには

コマンドプロンプトで次のコマンドを入力して、永続性情報の交換を有効または無効にし、構成を確認します。

```
set gslb site <siteName> -sessionExchange (ENABLED|DISABLED)
show gslb site** <siteName>
<!--NeedCopy-->
```

例**:**

```
set gslb site Site-GSLB-East-Coast -sessionExchange  ENABLED
set gslb site Site-GSLB-East-Coast -sessionExchange  DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

GUI を使用してパーシステンス情報交換を有効または無効にするには

1. [トラフィック管理] > [**GSLB**] > [サイト**]** に移動し、サイトをダブルクリックします。
2. 「**GSLB** サイトの設定」ダイアログで、「持続性セッションエントリ交換」チェックボックスをオンまたはオフにします。

ウィザードを使用して **GSLB** を構成する

August 15, 2023

ウィザードを使用して GSLB 展開タイプ (アクティブ-アクティブ、アクティブ-パッシブ、親子) を構成できるようになりました。

このウィザードは GUI で使用できます。ウィザードにアクセスするには、[設定] > [トラフィック管理] > [**GSLB**] に移動し、[は じめに] をクリックします。

このウィザードには、GSLB ダッシュボードからもアクセスできます。[設定] > [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動し、[**GSLB** の設定] をクリックします。

注:GSLB エンティティを個別に設定することもできます。

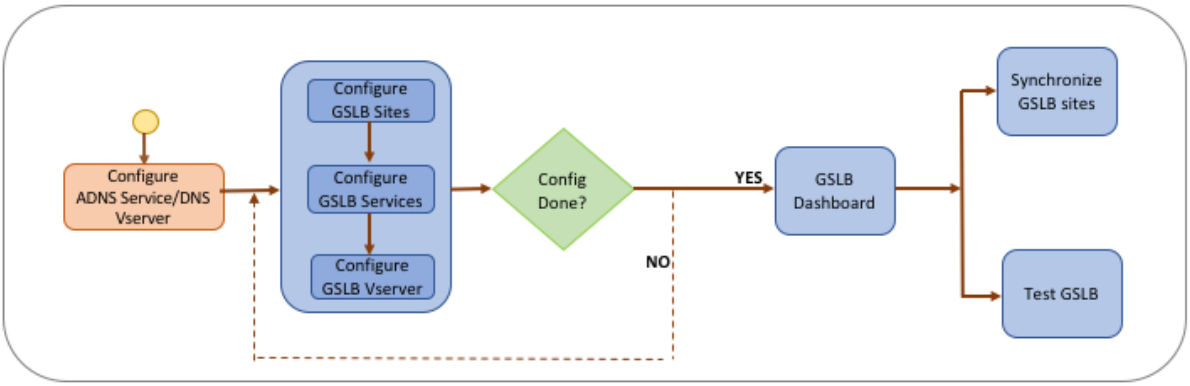
- アクティブ-アクティブサイト構成
- アクティブ/パッシブサイト構成
- 親子トポロジ設定

> 重要
>
> この機能は高可用性デプロイでサポートされており、管理パーティションとクラスターデプロイではサポートされていません。

アクティブ/アクティブサイトを構成する

August 15, 2023

次の図は、GSLB アクティブ-アクティブサイト構成に関連するワークフローを示しています。



アクティブ/アクティブサイトの構成を開始する前に、各サーバーファームまたはデータセンターの標準的な負荷分散設定を構成していることを確認してください。

また、デプロイメント内の GSLB サイト間で GSLB 設定を同期するには、次のことを確認してください。

- ローカル GLSB サイトは、GSLB 構成のすべてのアプライアンスで設定されます。
- 構成内のすべての GSLB サイトで管理アクセスを有効にしました。
- 自動同期と MEP 接続を受け入れるようにファイアウォールを設定しました。
- マスターアプライアンスとスレーブ NetScaler アプライアンスは、同じ NetScaler ソフトウェアバージョンを実行しています。
- サイトとして参加しているすべての NetScaler アプライアンスは、同じ NetScaler ソフトウェアバージョンである必要があります（サイトはマスターとスレーブの関係にはありません）。
- RPC ノードのパスワードは、GSLB 構成内のすべての GSLB サイトで同じです。

ウィザードを使用してアクティブ**-**アクティブサイトを構成するには

「構成」タブで、次の操作を行います。

1. [トラフィック管理] > [**GSLB**] に移動し、[開始] をクリックします。
2. サイトの ADNS サービスまたは DNS 仮想サーバーをまだ設定していない場合は、ここで設定できます。
   a) [表示] をクリックし、[追加] をクリックします。
   b) サービス名と IP アドレスを入力し、サービスとデータを交換するプロトコル（ADNS/ADNS_TCP）を選択します。
3. 「アクティブ**-**アクティブサイト」を選択します。
4. 完全修飾ドメイン名を入力し、レコードを DNS プロキシにキャッシュする必要がある期間を指定します。
5. GSLB サイトを設定します。各サイトはローカル GSLB サイトで構成する必要があり、各サイトの構成には他のすべてのサイトをリモート GSLB サイトとして含める必要があります。ローカルサイトは 1 つだけで、他のサイトはすべてリモートサイトです。
   a) サイト名やサイト IP アドレスなど、サイトの詳細を入力します。
   b) リモートサイトタイプまたはローカルサイトタイプを選択します。
   c) 必要に応じて RPC パスワードを変更し、必要に応じてセキュリティで保護します。
   d) モニターを GSLB サービスにバインドする場合は、モニターがサービスを監視する条件を選択します。これは、モニターがサービスにバインドされた後にのみ有効になります。考えられる条件は以下のとおりです。
      - 常に。GSLB サービスを常に監視してください。
      - **MEP** が失敗します。MEP によるメトリクスの交換が失敗した場合にのみ、GSLB サービスを監視してください。
      - **MEP** に障害が発生し、サービス **ID** がダウンしている。MEP によるメトリックの交換は有効ですが、メトリック交換によって更新されるサービスのステータスは DOWN です。
6. GSLB サービスを設定します。アクティブアクティブサイトを作成するには、少なくとも 2 つの GSLB サービスを追加する必要があります。

a) サービス名、サービスタイプ、ポート番号などのサービスの詳細を入力します。
b) GSLB サービスが属する GSLB サイトを選択して、サービスをサイト (ローカルまたはリモート) に関連付けます。
c) 必要に応じて、MEP に障害が発生した場合にサービスにバインドする必要があるモニターを選択します。サービスは既存のサーバーでも、新しいサーバーまたは仮想サーバーを作成することもできます。
d) 既存のサーバーを関連付けるには、サーバー名を選択します。サービス IP アドレスは自動的に入力されます。

- パブリック IP アドレスがサーバーの IP と異なる場合（NAT 環境で発生する可能性がある）、パブリック IP アドレスとパブリックポートのポート番号を入力します。
- 新しいサーバーを関連付けるには、サーバーの IP 詳細、パブリック IP アドレス、パブリックポート番号を入力してサーバーを作成します。
- 仮想サーバーを関連付けるには、既存の仮想サーバーを選択するか、+ をクリックして新しい仮想サーバーを追加します。この仮想サーバーは、この GSLB サービスが関連付けられる負荷分散仮想サーバーです。

7. GSLB 仮想サーバーを設定します。

a) GSLB 仮想サーバー名の名前を入力し、DNS レコードタイプを選択します。
b)「サービスの選択」ボックスで「**」をクリックし、GSLB 仮想サーバーにバインドする GSLB サービスを選択します。
c) ドメインバインディングボックスの ** をクリックして、この GSLB 仮想サーバーにバインドするドメインを選択します。
d) 最もパフォーマンスの高い GSLB サービスを選択するには、GSLB メソッドを選択してください。GSLB 方式、バックアップ方式、および動的重みのデフォルト値は、デフォルトで自動入力されます。必要に応じて変更できます。

- アルゴリズムベースの方法を選択する場合は、プライマリメソッドとバックアップメソッドを選択し、ダイナミックウェイトオプションも指定してください。
- スタティックプロキシミティ方式を選択する場合は、バックアップ方式と動的ウェイト方式を選択します。また、> アイコンをクリックしてデータベースファイルの場所を指定するか、[場所データベースの選択] ボックスの [**+**] をクリックして新しい場所を追加します。
- **Dynamic Proximity**（**RTT**）方式を選択した場合は、バックアップ方法を選択し、動的加重オプションと往復時間値を指定します。この値に基づいて、最もパフォーマンスの高いサービスを選択します。

8. 設定が完了したら、**[Done**] をクリックします。GSLB ダッシュボードが表示されます。
9. GSLB サイト構成を変更した場合は、ダッシュボードの「**GSLB** の自動同期」をクリックして、**GSLB** 設定の他のサイトと構成を同期します。

- 同期する前に、ローカルサイトの構成にリモートサイトに関する情報が含まれていることを確認してください。また、同期を正常に行うには、他の NetScaler アプライアンスでローカルサイトを構成する必要があります。

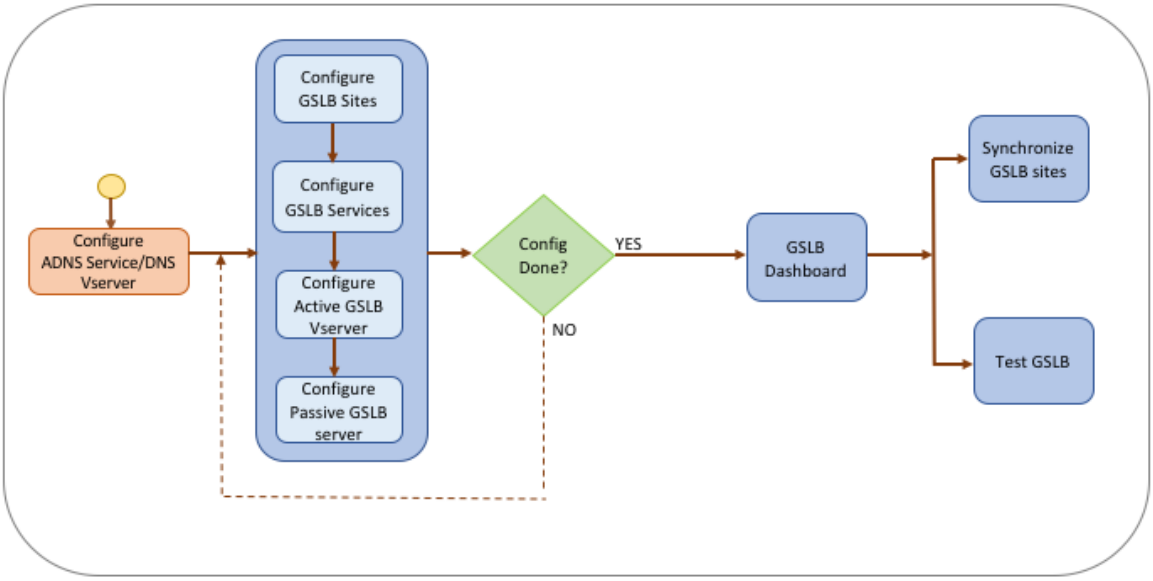
- リアルタイム同期が有効になっている場合は、「**GSLB** の自動同期」をクリックする必要はありません。同期は自動的に行われます。リアルタイム同期を有効にするには、次の操作を行います。
  a) [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動し、[**GSLB** 設定の変更] をクリックします。
  b)「自動構成同期」チェックボックスを選択します。

10.「**Test GSLB Setup**」をクリックして、**ADNS** サービスまたは **DNS** サーバーが **GSLB** セットアップで設定されたドメイン名の正しい IP アドレスで応答していることを確認します。

アクティブ-パッシブサイトを構成する

August 15, 2023

次の図は、アクティブ/パッシブサイト構成に関連するワークフローを示しています。



アクティブ/パッシブサイトの構成を開始する前に、各サーバーファームまたはデータセンターの標準的な負荷分散設定を構成していることを確認してください。

また、デプロイメント内の GSLB サイト間で GSLB 設定を同期するには、次のことを確認してください。

- ローカル GLSB サイトは、GSLB 構成のすべてのアプライアンスで設定されます。
- 構成内のすべての GSLB サイトで管理アクセスを有効にしました。
- 自動同期と MEP 接続を受け入れるようにファイアウォールを設定しました。
- マスターアプライアンスとスレーブ NetScaler アプライアンスは、同じ NetScaler ソフトウェアバージョンを実行しています。

- サイトとして参加しているすべての NetScaler アプライアンスは、同じ NetScaler ソフトウェアバージョンである必要があります（サイトはマスターとスレーブの関係にはありません）。
- RPC ノードのパスワードは、GSLB 構成内のすべての GSLB サイトで同じです。

ウィザードを使用してアクティブ/パッシブサイトを構成するには

「構成」タブで、次の操作を行います。

1. [トラフィック管理] > [**GSLB**] に移動し、[開始] をクリックします。
2. サイトの ADNS サービスまたは DNS 仮想サーバーをまだ設定していない場合は、ここで設定できます。

    a) [表示] をクリックし、[追加] をクリックします。
    b) サービス名と IP アドレスを入力し、サービスとデータを交換するプロトコル（ADNS/ADNS_TCP）を選択します。

3. アクティブ/パッシブサイトを選択します。
4. 完全修飾ドメイン名を入力し、レコードを DNS プロキシにキャッシュする必要がある期間を指定します。
5. GSLB サイトを設定します。各サイトはローカル GSLB サイトで構成する必要があり、各サイトの構成には他のすべてのサイトをリモート GSLB サイトとして含める必要があります。ローカルサイトは 1 つだけで、他のサイトはすべてリモートサイトです。

    a) サイト名やサイト IP アドレスなど、サイトの詳細を入力します。
    b) リモートサイトタイプまたはローカルサイトタイプを選択します。
    c) 必要に応じて RPC パスワードを変更し、必要に応じてセキュリティで保護します。
    d) モニターを GSLB サービスにバインドする場合は、モニターがサービスを監視する条件を選択します。これは、モニターがサービスにバインドされた後にのみ有効になります。考えられる条件は以下のとおりです。

        - 常に。GSLB サービスを常に監視してください。
        - **MEP** が失敗します。MEP によるメトリクスの交換が失敗した場合にのみ、GSLB サービスを監視してください。
        - **MEP** に障害が発生し、サービス **ID** がダウンしている。MEP によるメトリックの交換は有効ですが、メトリック交換によって更新されるサービスのステータスは DOWN です。

6. GSLB サービスを設定します。

    a) サービス名、サービスタイプ、ポート番号などのサービスの詳細を入力します。
    b) GSLB サービスが属する GSLB サイトを選択して、サービスをサイト (ローカルまたはリモート) に関連付けます。
    c) 必要に応じて、MEP に障害が発生した場合にサービスにバインドする必要があるモニターを選択します。サービスは既存のサーバーでも、新しいサーバーまたは仮想サーバーを作成することもできます。
    d) 既存のサーバーを関連付けるには、サーバー名を選択します。サービス IP アドレスは自動的に入力されます。

- パブリック IP アドレスがサーバーの IP と異なる場合（NAT 環境で発生する可能性がある）、パブリック IP アドレスとパブリックポートのポート番号を入力します。
- 新しいサーバーを関連付けるには、サーバーの IP 詳細、パブリック IP アドレス、パブリックポート番号を入力してサーバーを作成します。
- 仮想サーバーを関連付けるには、既存の仮想サーバーを選択するか、**+** をクリックして新しい仮想サーバーを追加します。この仮想サーバーは、この GSLB サービスが関連付けられる負荷分散仮想サーバーです。

7. GSLB バックアップ仮想サーバーを設定します。GSLB バックアップ仮想サーバーは、プライマリ GSLB 仮想サーバーにアクセスできない場合、または何らかの理由でダウンとマークされた場合にのみ動作可能になります。

   a) GSLB 仮想サーバー名の名前を入力し、DNS レコードタイプを選択します。
   b) サービスバインディングの ** をクリックし、GSLB 仮想サーバーにバインドする必要がある GSLB サービスを選択します。
   c) 最もパフォーマンスの高い GSLB サービスを選択するには、GSLB メソッドを選択してください。GSLB 方式、バックアップ方式、および動的重みのデフォルト値は、デフォルトで自動入力されます。必要に応じて変更できます。
      - アルゴリズムベースの方法を選択する場合は、プライマリメソッドとバックアップメソッドを選択します。
      - 静的近接方式を選択した場合は、バックアップ方法を選択し、データベースファイルの場所を指定します。
      - **Dynamic Proximity (RTT)** 方式を選択する場合は、バックアップ方法を選択し、サービスの重みと RTT 値を指定します。これに基づいて最もパフォーマンスの高いサービスを選択します。

8. GSLB 仮想サーバーを設定します。

   a) GSLB 仮想サーバー名の名前を入力し、DNS レコードタイプを選択します。
   b)「サービスの選択」ボックスで「**」をクリックし、GSLB 仮想サーバーにバインドする GSLB サービスを選択します。
   c) ドメインバインディングボックスの ** をクリックして、この GSLB 仮想サーバーにバインドするドメインを選択します。
   d) 最もパフォーマンスの高い GSLB サービスを選択するには、GSLB メソッドを選択してください。GSLB 方式、バックアップ方式、および動的重みのデフォルト値は、デフォルトで自動入力されます。必要に応じて変更できます。
      - アルゴリズムベースの方法を選択する場合は、プライマリメソッドとバックアップメソッドを選択し、ダイナミックウェイトオプションも指定してください。
      - スタティックプロキシミティ方式を選択する場合は、バックアップ方式と動的ウェイト方式を選択します。また、**** アイコンをクリックしてデータベースファイルの場所を指定するか、**[**ロケーションデータベースの選択**]** ボックスの **[****+**] をクリックして新しい場所を追加します。
      - **Dynamic Proximity**（**RTT**）方式を選択した場合は、バックアップ方法を選択し、動的加重オプションと往復時間値を指定します。この値に基づいて、最もパフォーマンスの高いサービスを選

択します。

9. 設定が完了したら、**[Done**] をクリックします。GSLB ダッシュボードが表示されます。
10. GSLB サイト構成を変更した場合は、ダッシュボードの「**GSLB** の自動同期」をクリックして、**GSLB** 設定の他のサイトと構成を同期します。

    - 同期する前に、ローカルサイトの構成にリモートサイトに関する情報が含まれていることを確認してください。また、同期を正常に行うには、他の NetScaler アプライアンスでローカルサイトを構成する必要があります。
    - リアルタイム同期が有効になっている場合は、「**GSLB** の自動同期」をクリックする必要はありません。同期は自動的に行われます。リアルタイム同期を有効にするには、次の操作を行います。

      a) [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動し、[**GSLB** 設定の変更] をクリックします。

      b)「自動構成同期」チェックボックスを選択します。

11.「**Test GSLB Setup**」をクリックして、**ADNS** サービスまたは **DNS** サーバーが **GSLB** セットアップで設定されたドメイン名の正しい IP アドレスで応答していることを確認します。

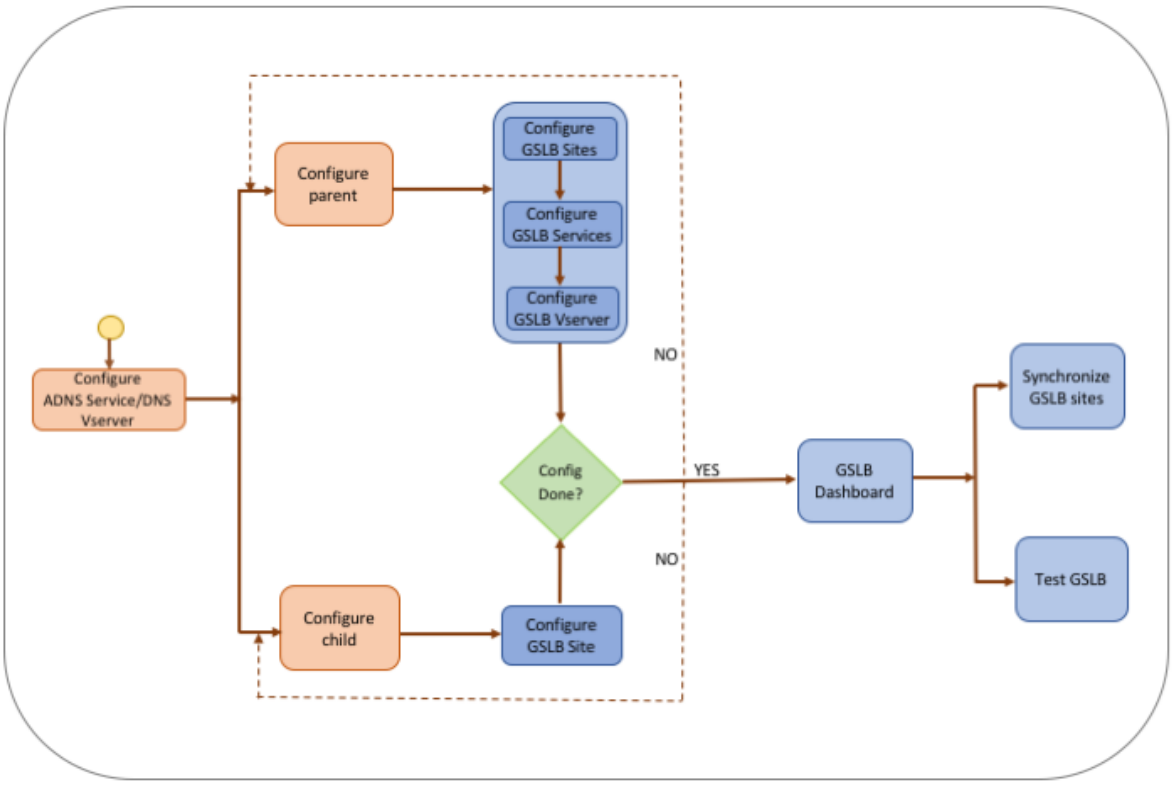
> 注
>
> ディザスタリカバリ用のアクティブ/パッシブ GSLB セットアップの GSLB エンティティの構成の詳細については、「ディザスタリカバリ用の GSLB の設定」を参照してください。

親子トポロジを構成する

August 15, 2023

親子トポロジでは、最上位にあるのが親サイトで、他の親サイトとピア関係にあります。各親サイトは複数の子サイトを持つことができ、各親サイトは子サイトや他の親サイトと健康情報を交換します。ただし、子サイトは親サイトとのみ通信します。

次の図は、GSLB 親子トポロジー構成に関連するワークフローを示しています。



親子トポロジー展開の設定を開始する前に、各サーバーファームまたはデータセンターの標準的な負荷分散設定を構成していることを確認してください。

また、デプロイメント内の GSLB サイト間で GSLB 設定を同期するには、次のことを確認してください。

- ローカル GLSB サイトは、GSLB 構成のすべてのアプライアンスで設定されます。
- 構成内のすべての GSLB サイトで管理アクセスを有効にしました。
- 自動同期と MEP 接続を受け入れるようにファイアウォールを設定しました。
- サイトとして参加しているすべての NetScaler アプライアンスは、同じ NetScaler ソフトウェアバージョンである必要があります（サイトはマスターとスレーブの関係にはありません）。
- RPC ノードのパスワードは、GSLB 構成内のすべての GSLB サイトで同じです。

ウィザードを使用して親子展開を設定するには

「構成」タブで、次の操作を行います。

1. [トラフィック管理] > [**GSLB**] に移動し、[開始] をクリックします。
2. サイトの ADNS サーバーまたは DNS 仮想サーバーをまだ構成していない場合は、ここで設定できます。

   a) [表示] をクリックし、[追加] をクリックします。
   b) サービス名と IP アドレスを入力し、サービスとデータを交換するプロトコル（ADNS/ADNS_TCP）を選択します。

3.「親子トポロジー」を選択します。
4.「サイトタイプを選択」フィールドで、以下を選択します。

- 親—親サイトを設定するときは、関連する子サイトを設定し、GSLB 設定で他の親サイトも設定する必要があります。
- 子サイト—子サイトを構成する場合、子サイトとその親サイトのみを構成する必要があります。

親サイトを設定するには

1. 完全修飾ドメイン名を入力し、レコードを DNS プロキシにキャッシュする必要がある期間を指定します。
2. GSLB サイトを設定します。各サイトはローカル GSLB サイトで構成する必要があり、各サイトの構成には他のすべてのサイトをリモート GSLB サイトとして含める必要があります。ローカルサイトは 1 つしか存在できません。その他のサイトはすべてリモートサイトです。指定されたサイトの IP アドレスがアプライアンスによって所有されている場合（たとえば、MIP アドレスや SNIP アドレス）、そのサイトはローカルサイトです。それ以外の場合は、リモートサイトになります。
3. サイト名やサイト IP アドレスなど、サイトの詳細を入力します。

   a) サイトタイプを選択します。
   b) 必要に応じて RPC パスワードを変更し、必要に応じてセキュリティで保護します。
   c) モニターを GSLB サービスにバインドする場合は、モニターがサービスを監視する条件を選択します。これは、モニターがサービスにバインドされた後にのみ有効になります。考えられる条件は以下のとおりです。

      - **Always**。GSLB サービスを常に監視してください。
      - **MEP** が失敗します。MEP によるメトリクスの交換が失敗した場合にのみ、GSLB サービスを監視してください。
      - **MEP** に障害が発生し、サービスが停止している。MEP によるメトリックの交換は有効ですが、メトリック交換によって更新されるサービスのステータスは DOWN です。

4. GSLB サービスを設定します。

   a) サービス名、サービスタイプ、ポート番号などのサービスの詳細を入力します。
   b) GSLB サービスが属する GSLB サイトを選択して、サービスをサイト (ローカルまたはリモート) に関連付けます。
   c) 必要に応じて、MEP に障害が発生した場合にサービスにバインドする必要があるモニターを選択します。サービスは既存のサーバーでも、新しいサーバーまたは仮想サーバーを作成することもできます。

      - 既存のサーバーを関連付けるには、サーバー名を選択します。サービス IP アドレスは自動入力されます。
      - 新しいサーバーを関連付けるには、サーバーの IP 詳細、パブリック IP アドレス、パブリックポート番号を入力してサーバーを作成します。
      - 仮想サーバーを関連付けるには、既存の仮想サーバーを選択するか、**+** をクリックして新しい仮想サーバーを追加します。この仮想サーバーは、この GSLB サービスが関連付けられる負荷分散仮想

サーバーです。パブリック IP アドレスが NAT 環境で発生する可能性のあるサーバー IP と異なる場合は、パブリック IP アドレスとパブリックポート番号を入力します。

5. GSLB 仮想サーバーを設定します。

   a) GSLB 仮想サーバー名の名前を入力し、DNS レコードタイプを選択します。
   b)「サービスの選択」ボックスで「**」をクリックし、GSLB 仮想サーバーにバインドする GSLB サービスを選択します。
   c) ドメインバインディングボックスの ** をクリックすると、GSLB 仮想サーバーにバインドされているドメイン名が表示されます。
   d) 最もパフォーマンスの高い GSLB サービスを選択するには、GSLB メソッドを選択してください。GSLB 方式、バックアップ方式、および動的重みのデフォルト値は、デフォルトで自動的に入力されます。必要に応じて変更できます。

      - アルゴリズムベースの方法を選択する場合は、プライマリメソッドとバックアップメソッドを選択し、ダイナミックウェイトオプションも指定してください。
      - スタティックプロキシミティ方式を選択する場合は、バックアップ方式と動的ウェイト方式を選択します。また、**** アイコンをクリックしてデータベースファイルの場所を指定するか、**[**ロケーションデータベースの選択**]** ボックスの **[****+**] をクリックして新しい場所を追加します。
      - **Dynamic Proximity (RTT)** 方式を選択する場合は、バックアップ方法を選択し、サービスの重みと RTT 値を指定します。これに基づいて最もパフォーマンスの高いサービスを選択します。

6. 設定が完了したら、**[Done**] をクリックします。GSLB ダッシュボードが表示されます。
7. GSLB 親サイト構成を変更した場合は、「GSLB の自動同期」をクリックして、**GSLB** 設定の他の親サイトに設定を同期します。親子トポロジでは、子サイトの同期はスキップされます。

   - 同期する前に、ローカルサイトの構成にリモートサイトに関する情報が含まれていることを確認してください。
   - リアルタイム同期が有効になっている場合は、「**GSLB** の自動同期」をクリックする必要はありません。同期は自動的に行われます。リアルタイム同期を有効にするには、次の操作を行います。

     a) [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動し、[**GSLB** 設定の変更] をクリックします。
     b)「自動構成同期」チェックボックスを選択します。

8.「**Test GSLB Setup**」をクリックして、**ADNS** サービスまたは **DNS** サーバーが **GSLB** セットアップで設定されたドメイン名の正しい IP アドレスで応答していることを確認します。

子サイトを設定するには

1. GSLB サイトを設定します。

   a) サイト名やサイト IP アドレスなど、サイトの詳細を入力します。
   b) サイトタイプを選択します。

c) 必要に応じて RPC パスワードを変更し、必要に応じてセキュリティで保護します。4。モニターが GSLB サービスにバインドされている場合は、モニターがサービスを監視する条件を選択します。考えられる条件は以下のとおりです。

- **Always**。GSLB サービスを常に監視してください。
- **MEP** が失敗します。MEP によるメトリクスの交換が失敗した場合にのみ、GSLB サービスを監視してください。
- **MEP** に障害が発生し、サービスが停止している。MEP によるメトリックの交換は有効ですが、メトリック交換によって更新されるサービスのステータスは DOWN です。

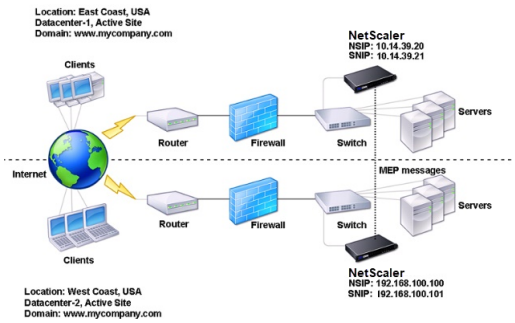
2. 設定が完了したら、**[Done**] をクリックします。GSLB ダッシュボードが表示されます。
3.「**Test GSLB Setup**」をクリックして、**ADNS** サービスまたは **DNS** サーバーが **GSLB** セットアップで設定されたドメイン名の正しい IP アドレスで応答していることを確認します。

GSLB エンティティを個別に構成する

August 15, 2023

グローバルサーバー負荷分散は、理想的には地理的に異なる 2 つのサーバーファームでホストされている Web サイトへのトラフィックフローを管理するために使用されます。たとえば、地理的に離れた 2 つのサーバーファームまたはデータセンターでホストされている Web サイト www.mycompany.com を考えてみましょう。どちらのサーバーファームも NetScaler アプライアンスを使用しています。これらのサーバーファームの NetScaler アプライアンスはワンアームモードでセットアップされ、www.mycompany.com ドメインの権限のある DNS サーバーとして機能します。次の図は、この構成を示しています。

図 1：基本的な GSLB トポロジー



このような GSLB 設定を構成するには、まず各サーバーファームまたはデータセンターの標準の負荷分散設定を構成する必要があります。これにより、各サーバーファームのさまざまなサーバー間で負荷を分散できます。次に、両方の NetScaler アプライアンスを権限のある DNS（ADNS）サーバーとして構成します。次に、サーバーファームごとに GSLB サイトを作成し、サイトごとに GSLB 仮想サーバーを構成し、GLSB サービスを作成して、GSLB サービスを GSLB 仮想サーバーにバインドします。最後に、ドメインを GSLB 仮想サーバーにバインドします。2 つの異な

るサイトにある 2 つのアプライアンスの GSLB 構成は同じですが、各サイトのロードバランシング構成はそのサイトに固有です。

注：NetScaler ADC クラスタ設定で GSLB サイトを構成するには、クラスタでの GSLB の設定を参照してください。

標準ロードバランシング設定の設定

負荷分散仮想サーバーは、データセンター内のさまざまな物理サーバー間で負荷を分散します。これらのサーバーは NetScaler ADC アプライアンス上でサービスとして表され、サービスは負荷分散仮想サーバーにバインドされます。

基本的な負荷分散設定の設定の詳細については、負荷分散を参照してください。

権限を持っている **DNS** サービスを構成する

August 15, 2023

NetScaler アプライアンスを権限のある DNS サーバーとして構成すると、クライアントからの DNS 要求を受け付け、クライアントが要求を送信するデータセンターの IP アドレスで応答します。

注：NetScaler ADC アプライアンスの権限を持つためには、SOA レコードと NS レコードも作成する必要があります。SOA および NS レコードの詳細については、「ドメインネームシステム」を参照してください。

コマンドラインインターフェイスを使用して **ADNS** サービスを作成するには

コマンドプロンプトで次のコマンドを入力して ADNS サービスを作成し、構成を確認します。

```
add service <name> <IP>@ ADNS <port>

show service <name>
<!--NeedCopy-->
```

例:

```
add service Service-ADNS-1 10.14.39.21 ADNS 53

show service Service-ADNS-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **ADNS** サービスを変更するには

コマンドプロンプトで、次のコマンドを入力します。


```
set service <name> <IPAddress> ADNS <port>
<!--NeedCopy-->
```

例:

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **ADNS** サービスを削除するには

コマンドプロンプトで、次のコマンドを入力します。

```
rm service <name>
<!--NeedCopy-->
```

例:

```
rm service Service-ADNS-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **ADNS** サービスを構成するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. 新しい ADNS サービスを追加するか、既存のサービスを選択してその設定を編集します。

基本的な **GSLB** サイトを構成する

August 15, 2023

GSLB サイトはネットワーク内のデータセンターを表し、GSLB 仮想サーバー、サービス、その他のネットワークエンティティを論理的にグループ化したものです。通常、GSLB の設定には、同じコンテンツをクライアントに提供するための機能を備えた GSLB サイトが多数あります。これらは通常、1 つのサイトが完全にダウンした場合でもドメインがアクティブであることを保証するために、地理的に分離されています。GSLB 構成のすべてのサイトは、GSLB サイトをホストするすべての NetScaler アプライアンスで構成する必要があります。つまり、各サイトで、ローカル GSLB サイトと各リモート GSLB サイトを設定します。

ドメインの GSLB サイトが作成されると、NetScaler アプライアンスは、構成された GSLB アルゴリズムによって決定された適切な GSLB サイトにクライアント要求を送信します。

コマンドラインインターフェイスを使用して **GSLB** サイトを作成するには

コマンドプロンプトで次のコマンドを入力して GSLB サイトを作成し、構成を確認します。

```
add gslb site <siteName> <siteIPAddress>
show gslb site <siteName>
<!--NeedCopy-->
```

例**:**

```
add gslb site Site-GSLB-East-Coast 10.14.39.21
show glsb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** サイトを変更または削除するには

- GSLB サイトを変更するには、set gslb site コマンドを使用します。これは add gslb site コマンドを使用するのと同じですが、既存の GSLB サイトの名前を入力する点が異なります。
- サイトパラメータの設定を解除するには、unset gslb site コマンドを使用し、続けて siteName 値とデフォルト値にリセットするパラメータの名前を指定します。
- GSLB サイトを削除するには、引数のみを受け入れる rm gslb site コマンドを使用します。<name>

構成ユーティリティを使用して基本的な **GSLB** サイトを構成するには

1. **[**** トラフィック管理] > [**GSLB**] > [サイト] に移動します。**
2. 新しい GSLB サイトを追加するか、既存の GSLB サイトを選択してその設定を編集します。

コマンドラインインターフェイスを使用して **GSLB** サイトの統計を表示するには

コマンドプロンプトで入力します。

```
stat gslb site <siteName>
<!--NeedCopy-->
```

例**:**

```
stat gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

設定ユーティリティを使用して **GSLB** サイトの統計を表示するには

1. **[**** トラフィック管理] > [**GSLB**] > [サイト] に移動します。**
2. **GSLB** サイトを選択し、**[**統計**]** をクリックします。

GSLB サービスを設定する

August 15, 2023

GSLB サービスは、負荷分散またはコンテンツスイッチング仮想サーバーを表します。ローカル GSLB サービスは、ローカル負荷分散またはコンテンツスイッチング仮想サーバーを表します。リモート GSLB サービスは、GSLB セットアップの他のサイトの 1 つで構成された負荷分散またはコンテンツスイッチング仮想サーバーを表します。GSLB セットアップの各サイトで、1 つのローカル GSLB サービスと任意の数のリモート GSLB サービスを作成できます。

> 重要
>
> 負荷分散仮想サーバーが GSLB ノード自体にあるか、子ノード (親子展開) にあり、モニターが GSLB サービスにバインドされていない場合は、
>
> 次のことを確認します。GLSB サービスの IP アドレス、ポート番号、およびプロトコルが仮想サーバーです。それ以外の場合、サービス状態は DOWN とマークされます。

コマンドラインインターフェイスを使用して GSLB サービスを作成するには

コマンドプロンプトで次のコマンドを入力して GSLB サービスを作成し、構成を確認します。

```
add gslb service <serviceName> <serverName | IP> <serviceType> <port>-
   siteName <string>
show gslb service <serviceName>
<!--NeedCopy-->
```

例:

```
add gslb service Service-GSLB-1 10.14.39.14 HTTP 80  -siteName Site-
   GSLB-East-Coast
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して GSLB サービスを変更または削除するには

- GSLB サービスを変更するには、set gslb サービスコマンドを使用します。<serviceName> このコマンドでは、構成を変更する GSLB サービスの名前を指定します。自分で指定したパラメータまたはデフォルトで設定されているパラメータの既存の値を変更できます。同じコマンドで複数のパラメーターの値を変更できます。パラメータの詳細については、add gslb サービスコマンドを参照してください。例

  ```
  > set gslb service SKP_GSLB_NOTCNAME_SVC2 -maxBandWidth 25 -
     maxClient 8
  Done
  > sh gslb service SKP_GSLB_NOTCNAME_SVC2
  ```



```
SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
...
Max Conn: 8  Max Bandwidth: 25 kbits
<!--NeedCopy-->
```

- パラメータをデフォルト値にリセットするには、unset gslb <serviceName> サービスコマンドと設定解除するパラメータを使用できます。例

```
> unset gslb service SKP_GSLB_NOTCNAME_SVC2 maxBandWidth
Done
> sh gslb service SKP_GSLB_NOTCNAME_SVC2
SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
...
Max Conn: 8  Max Bandwidth: 0 kbits
<!--NeedCopy-->
```

- GSLB サービスを削除するには、rm gslb サービスコマンドを使用します。<serviceName>

構成ユーティリティを使用して **GSLB** サービスを作成するには

1. **[** トラフィック管理] > [**GSLB**] > [サービス] に移動します。**
2. 新しい GSLB サービスを追加するか、既存のサービスを選択してその設定を編集します。

コマンドラインインターフェイスを使用して **GSLB** サービスの統計を表示するには

コマンドプロンプトで入力します。

```
stat gslb service <serviceName>
<!--NeedCopy-->
```

例**:**

```
stat gslb service Service-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** サービスの統計を表示するには

1. **[** トラフィック管理] > [**GSLB**] > [サービス] に移動します。**
2. GSLB サービスを選択し、[統計] をクリックします。

GSLB サービスグループを構成する

August 15, 2023

サービスグループを使用すると、サービスグループを 1 つのサービスと同じくらい簡単に管理できます。サービスグループに対していずれかのオプションを有効または無効にすると、そのサービスグループのすべてのメンバーに対してそのオプションが有効または無効になります。たとえば、この機能は [圧縮]、[ヘルスモニタリング]、[グレースフルシャットダウン] などのオプションに適用できます。

サービスグループを作成したら、次のいずれかを実行できます。

- サービスグループを仮想サーバにバインドします。
- サービスをサービスグループに追加します。
- モニタをサービスグループにバインドします。

> 重要
>
> 負荷分散仮想サーバーが GSLB ノード自体または子ノード (親子展開) にあり、モニターが GSLB サービスにバインドされていない場合は、
>
> 次のことを確認してください。GLSB サービスグループの IP アドレス、ポート番号、およびプロトコルが、サービスの仮想サーバーと一致している表す。それ以外の場合、サービス状態は DOWN とマークされます。

NetScaler は、次の種類の GSLB サービスグループをサポートしています。

- IP アドレスベースのサービスグループ
- ドメイン名ベースのサービスグループ
- ドメイン名ベースの Autoscale サービスグループ

GSLB ドメイン名ベースの **Autoscale** サービスグループ

NetScaler ハイブリッドおよびマルチクラウドのグローバルサーバー負荷分散（GSLB）ソリューションを使用すると、ハイブリッドクラウド、複数のクラウド、オンプレミスの複数のデータセンターにアプリケーショントラフィックを分散できます。NetScaler GSLB ソリューションは、NetScaler ロードバランサー、AWS の弾性負荷分散 (ELB)、その他のサードパーティ製ロードバランサーなど、さまざまな負荷分散ソリューションをサポートします。また、GSLB ソリューションでは、GSLB レイヤと負荷分散レイヤが個別に管理されている場合でも、グローバル負荷分散が実行されます。

クラウド展開では、管理目的で負荷分散ソリューションにアクセスする際に、参照としてドメイン名がユーザーに与えられます。外部エンティティは、これらのドメイン名が解決する IP アドレスを使用しないことをお勧めします。また、負荷分散層は負荷に応じてスケールアップまたはスケールダウンされるため、IP アドレスが静的であることは保証されません。したがって、IP アドレスではなくドメイン名を使用して負荷分散エンドポイントを参照することをお勧めします。これには、IP アドレスではなくドメイン名を使用して GSLB サービスを参照する必要があり、負荷分散レイヤードメイン名に対して返されたすべての IP アドレスを消費し、GSLB で同じ表現を持つ必要があります。

負荷分散エンドポイントを参照するときに IP アドレスの代わりにドメイン名を使用するには、GSLB のドメイン名ベースのサービスグループを使用できます。

GSLB ドメイン名ベースのサービスグループの監視

NetScaler ADC アプライアンスには、TCP ベースのアプリケーションを監視する 2 つの内蔵モニターがあります。`tcp-default`と`ping-default`。`tcp-default` モニターはすべての TCP サービスにバインドされ、`ping-default` モニターはすべての非 TCP サービスにバインドされます。ビルトインモニターは、デフォルトで GSLB サービスグループにバインドされます。ただし、アプリケーション固有のモニターを GSLB サービスグループにバインドすることを推奨します。

トリガモニタオプションを **MEPDOWN** に設定するための推奨事項　[Trigger Monitor] オプションを使用すると、GSLB サイトで常にモニターを使用する必要があるか、メトリック交換プロトコル (MEP) が DOWN のときにモニターを使用する必要があるかを示すことができます。

[トリガーモニター] オプションは、デフォルトで ALWAYS に設定されています。

Trigger Monitor オプションが ALWAYS に設定されている場合、各 GSLB ノードはモニターを個別にトリガーします。各 GSLB ノードが個別にモニターをトリガーする場合、各 GSLB ノードは異なる GSLB サービスセットで動作する可能性があります。これにより、これらの GSLB ノードに到達する DNS 要求に対する DNS 応答に不一致が生じる可能性があります。また、各 GSLB ノードが個別に監視している場合は、ロードバランサーエンティティに到達する監視プローブの数が増加します。永続性エントリも GSLB ノード間で互換性がなくなります。

したがって、GSLB サイトエンティティの [トリガーモニター] オプションを MEPDOWN に設定することをお勧めします。[Trigger Monitor] オプションが MEPDOWN に設定されている場合、負荷分散ドメインの解決と監視の所有権はローカル GSLB ノードにあります。[Trigger Monitor] オプションが MEPDOWN に設定されている場合、負荷分散ドメインの解決とそれ以降の監視は、GSLB サービスグループのローカル GSLB ノードによって実行されます。その結果は、メトリック交換プロトコル (MEP) を使用して GSLB に参加している他のすべてのノードに伝播されます。

また、負荷分散ドメインに関連付けられている IP アドレスのセットが更新されるたびに、MEP を通じて通知されます。

GSLB サービスグループの制限事項

- 負荷分散ドメインの場合、DNS 応答で返される IP アドレスは一般にパブリック IP アドレスです。負荷分散ドメインが解決されると、プライベート IP アドレスを動的に適用することはできません。したがって、GSLB ドメイン名ベースの Autoscale サービスグループの IP ポートバインディングのパブリック IP ポートとプライベート IP ポートは同じです。これらのパラメータは、ドメイン名ベースの Autoscale サービスグループに対して明示的に設定することはできません。
- GSLB サービスグループでは、サイト永続性、DNS ビュー、およびクラスタリングはサポートされていません。

CLI を使用して **GSLB** サービスグループを設定および管理する

GSLB サービスグループを追加するには、次の手順を実行します。


```
add gslb serviceGroup <serviceGroupName>@ <serviceType> [-autoScale (
    DISABLED | DNS )] -siteName <string>
<!--NeedCopy-->
```

例:

```
add gslb serviceGroup Service-Group-1 http -autoScale DNS -siteName
    Site1
<!--NeedCopy-->
```

GSLB サービスグループを仮想サーバーにバインドするには

```
bind gslb serviceGroup <serviceGroupName> ((<IP>@ <port>) | <serverName
    >@ | (-monitorName <string>@))
<!--NeedCopy-->
```

例:

```
bind gslb serviceGroup Service-Group-1 203.0.113.2
bind gslb serviceGroup Service-Group-1 S1 80
bind gslb serviceGroup** Service-Group-1 -monitorName Mon1
<!--NeedCopy-->
```

GSLB サービスグループを仮想サーバにバインド解除するには

```
unbind gslb serviceGroup <serviceGroupName> ((<IP>@ <port>) | <
    serverName>@ | -monitorName <string>@)
<!--NeedCopy-->
```

例:

```
unbind gslb serviceGroup Service-Group-1 -monitorName Mon1
<!--NeedCopy-->
```

GSLB サービスグループのパラメータを設定するには、次の手順を実行します。

```
set gslb serviceGroup <serviceGroupName>@ [(<serverName>@ <port> [-
    weight <positive_integer>] [-hashId <positive_integer>] [-publicIP <
    ip_addr|ipv6_addr|*>] [-publicPort <port>]) | -maxClient <
    positive_integer> | -cip ( ENABLED | DISABLED ) | <cipHeader> | -
    cltTimeout <secs> | -svrTimeout <secs> | -maxBandwidth <
    positive_integer> | -monThreshold <positive_integer> | -
    downStateFlush ( ENABLED | DISABLED )] [-monitorName <string> -
    weight <positive_integer>] [-healthMonitor ( YES | NO )] [-comment <
    string>] [-appflowLog ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

GSLB サービスグループからパラメータの設定を解除するには、次の手順を実行します。

```
unset gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port> [-
    weight] [-hashId] [-publicIP] [-publicPort]] [-maxClient] [-cip] [-
    cltTimeout] [-svrTimeout] [-maxBandwidth] [-monThreshold] [-
```



```
    appflowLog] [-monitorName] [-weight] [-healthMonitor] [-cipHeader]
    [-downStateFlush] [-comment]
<!--NeedCopy-->
```

GSLB サービスグループを有効にするには

```
enable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>]
<!--NeedCopy-->
```

例:

```
enable gslb serviceGroup SG1 S1 80
<!--NeedCopy-->
```

GSLB サービスグループを無効にするには

```
disable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>] [-
    delay <secs>] [-graceFul ( YES /| NO )]
<!--NeedCopy-->
```

例:

```
disable gslb serviceGroup SRG2 S1 80
<!--NeedCopy-->
```

> 注:
>
> 無効にする必要があるサービスグループは、Autoscale サービスグループではなく DBS サービスグループである必要があります。

GSLB サービスグループを削除するには、次の手順を実行します。

```
rm gslb serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

例:

```
rm gslb serviceGroup Service-Group-1
<!--NeedCopy-->
```

GSLB サービスグループの統計情報を表示するには、次の手順を実行します。

```
stat gslb serviceGroup [<serviceGroupName>]
<!--NeedCopy-->
```

例:

```
stat gslb serviceGroup Service-Group-1
<!--NeedCopy-->
```

GSLB サービスグループのプロパティを表示するには、次の手順を実行します。


```
show gslb serviceGroup [<serviceGroupName>  -includeMembers]
<!--NeedCopy-->
```

例:

```
show gslb serviceGroup SG1
show gslb serviceGroup -includeMembers
<!--NeedCopy-->
```

GSLB サービスグループメンバーの有効化または無効化

サービスグループ全体を有効または無効にする代わりに、GSLB (DNS ベース) サービスグループの個々のメンバーを選択的に有効または無効にできます。この機能は、Autoscale サービスグループと Autoscale 以外のサービスグループの両方で使用できます。そのため、GSLB サービスグループの管理が容易になります。

たとえば、GSLB サイト上の特定のサーバーへのトラフィックを避ける必要があるとします。10 個の GSLB サービスまたはサーバー (S1 ～S10) が 1 つのサービスグループ (SG1) にバインドされているとします。サービス 5（S5）のみを無効にして、サーバ 5 へのトラフィックを回避します。この機能を使用しない場合、サービス S1 から S4 とサービス S6 から S10 を個別にバインドする必要があります。このプロセスは、多数のサービスを無効または有効にする必要がある大規模な GSLB サービスグループでは面倒です。この機能を使用すると、サービスグループ内の他のサービスに影響を与えることなく、サービス 5（S5）を直接無効にできます。

CLI を使用して GSLB サービスグループメンバーを有効にするには、次の手順を実行します。

```
enable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>]
<!--NeedCopy-->
```

> 注:
>
> GSLB サービスグループを有効にするには、サービスグループ名だけを指定します。サービスグループのメンバーを有効にするには、GSLB サービスグループ名に加えて、サービスをホストするサーバーの名前とサービスのポート番号を指定します。

例**:**

```
enable gslb serviceGroup http_svc_group 10.102.27.153 80
<!--NeedCopy-->
```

CLI を使用して GSLB サービスグループまたは GSLB サービスグループのメンバーを無効にするには、次の手順を実行します。

```
disable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>]
<!--NeedCopy-->
```

例**:**


```
disable gslb serviceGroup http_svc_group 10.102.27.153 80
<!--NeedCopy-->
```

> 注:
>
> GSLB サービスグループを無効にするには、サービスグループ名だけを指定します。サービスグループのメンバーを無効にするには、GSLB サービスグループ名に加えて、サービスをホストするサーバーの名前とサービスのポート番号を指定します。

既存の **GSLB CLI** コマンドへの変更

GSLB サービスグループの導入後、既存の GSLB コマンドに対して行われた変更は次のとおりです。

- `bind gslb vserver` -bind コマンドにサービスグループ名が追加されます。

  例:

  ```
  bind gslb vserver <name> ((-serviceName <string> [-weight <
     positive_integer>] ) | <serviceGroupName>@ | | (-domainName <
     string> [-TTL <secs>] [-backupIP<ip_addr|ipv6_addr|*>] [-
     cookieDomain <string>] [-cookieTimeout <mins>][-sitedomainTTL
     <secs>]) | (-policyName <string>@ [-priority<positive_integer
     >] [-gotoPriorityExpression <expression>] [-type REQUEST |
     RESPONSE )]))
  <!--NeedCopy-->
  ```

- `unbind gslb vserver` -サービスグループが unbind コマンドに追加されます。

  例:

  ```
  unbind gslb vserver <name> (-serviceName <string> <
     serviceGroupName> @ /(-domainName <string> [-backupIP] [-
     cookieDomain]) | -policyName <string>@)
  <!--NeedCopy-->
  ```

- `show gslb site` -このコマンドを実行すると、GSLB サービスグループも表示されます。
- `show gslb vs` -このコマンドを実行すると、GSLB サービスグループが表示されます。
- `stat gslb vs` -このコマンドを実行すると、GSLB サービスグループの統計情報も表示されます。
- `show lb monitor bindings` -このコマンドを実行すると、GSLB サービスグループバインディングも表示されます。

GUI を使用して **GSLB** サービスグループを構成する

1. [トラフィック管理] > [**GSLB**] > [サービスグループ] に移動します。
2. サービスグループを作成し、AutoScale モードを DNS に設定します。

GSLB サービスグループのサイト永続性を構成する

IP アドレスベースおよびドメイン名ベースのサービスグループに対して、サイト永続性を構成できます。ドメイン名ベースの Autoscale サービスグループでは、サイトの永続性はサポートされていません。

CLI を使用して **HTTP Cookie** に基づいてサイト永続性を設定するには

- 接続プロキシの永続性については、サイトプレフィックスを設定する必要はありません。

  コマンドプロンプトで入力します。

```
set gslb service group <serviceGroupName> [-sitePersistence <
    sitePersistence>]
<!--NeedCopy-->
```

- HTTP リダイレクトの永続性については、まずサービスグループのメンバのサイトプレフィックスを設定し、次にサービスグループの HTTPRedirect persistence パラメータを設定する必要があります。

  コマンドプロンプトで入力します。

```
set gslb servicegroup <serviceGroupName>  <serviceGroup member
    name|Ip> <port>  [-sitePrefix <string>]

set gslb servicegroup <serviceGroupName> [-sitePersistence <
    sitePersistence>]
<!--NeedCopy-->
```

例:

- 接続プロキシの永続性

```
set gslbservicegroup sg1 -sitePersistence connectionProxy
<!--NeedCopy-->
```

- HTTP リダイレクトの永続性

```
set gslb servicegroup sg2 test1 80 -sitePrefix vserver-GSLB-1

set gslb servicegroup sg2 -sitePersistence HTTPRedirect
<!--NeedCopy-->
```

GUI を使用して **Cookie** に基づいてサイトの永続性を設定するには

1. [トラフィック管理] > [**GSLB**] > [サービスグループ] に移動し、サイトの永続性を設定するサービスグループ (ServiceGroup-GSLB-1 など) を選択します。
2. [サイトの永続性] セクションをクリックし、要件を満たすパーシステンスを設定します。

> ヒント
>
> GSLB サービスグループの展開シナリオと設定例については、以下のトピックを参照してください。
>
> - ユースケース: ドメイン名ベースの Autoscale サービスグループの展開
> - ユースケース:IP アドレスベースの Autoscale サービスグループの展開

GSLB 仮想サーバーを構成する

August 15, 2023

GSLB 仮想サーバーは、1 つ以上の GSLB サービスを表し、それらのサービス間でトラフィックのバランスをとるエンティティです。設定済みの GSLB メソッドまたはアルゴリズムを評価して、クライアントリクエストの送信先となる GSLB サービスを選択します。

> 注
>
> GSLB 仮想サーバープロトコル要件は、主に仮想サーバーと仮想サーバーにバインドされているサービスとの関係を構築することです。これにより、他のタイプの仮想サーバーでも CLI/API の一貫性が保たれます。サービスまたは仮想サーバーのサービスタイプパラメータは、DNS 要求の処理中には使用されません。代わりに、サイトの永続化、監視、および MEP 経由の検索時に参照されます。

コマンドラインインターフェイスを使用して GSLB 仮想サーバーを作成するには

コマンドプロンプトで、次のコマンドを入力して GSLB 仮想サーバーを追加し、構成を確認します。

```
1 -  add gslb vserver <name> <serviceType> -ipType (IPv4 | IPv6)
2 -  show gslb vserver <name>
3 <!--NeedCopy-->
```

例:

```
1 add gslb vserver Vserver-GSLB-1 HTTP -ipType IPv4
2 add gslb vserver Vserver-GSLB-2 HTTP -ipType IPv6
3 show gslb vserver Vserver-GSLB-1
4 show gslb vserver Vserver-GSLB-2
5 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して GSLB 仮想サーバーを変更または削除するには

- GSLB 仮想サーバーを変更するには、`set gslb vserver` コマンドを使用します。このコマンドは、既存の GSLB 仮想サーバーの名前を入力する点以外は、`add gslb vserver` コマンドと同様に機能します。

- パラメータをデフォルト値にリセットするには、`unset gslb vserver` コマンドの後に VServer-Name 値と設定解除するパラメーターの名前を入力します。
- GSLB 仮想サーバーを削除するには、name 引数のみを受け入れる`rm gslb vserver`コマンドを使用します。

構成ユーティリティを使用して **GSLB** 仮想サーバーを構成するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. 新しい GSLB 仮想サーバーを追加するか、既存の GSLB 仮想サーバーを選択してその設定を編集します。

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーの統計を表示するには

コマンドプロンプトで入力します。

```
stat gslb vserver <name>
<!--NeedCopy-->
```

例**:**

```
stat gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** 仮想サーバーの統計を表示するには

[トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動し、仮想サーバーを選択して [統計] をクリックします。

GSLB 仮想サーバーの統計情報

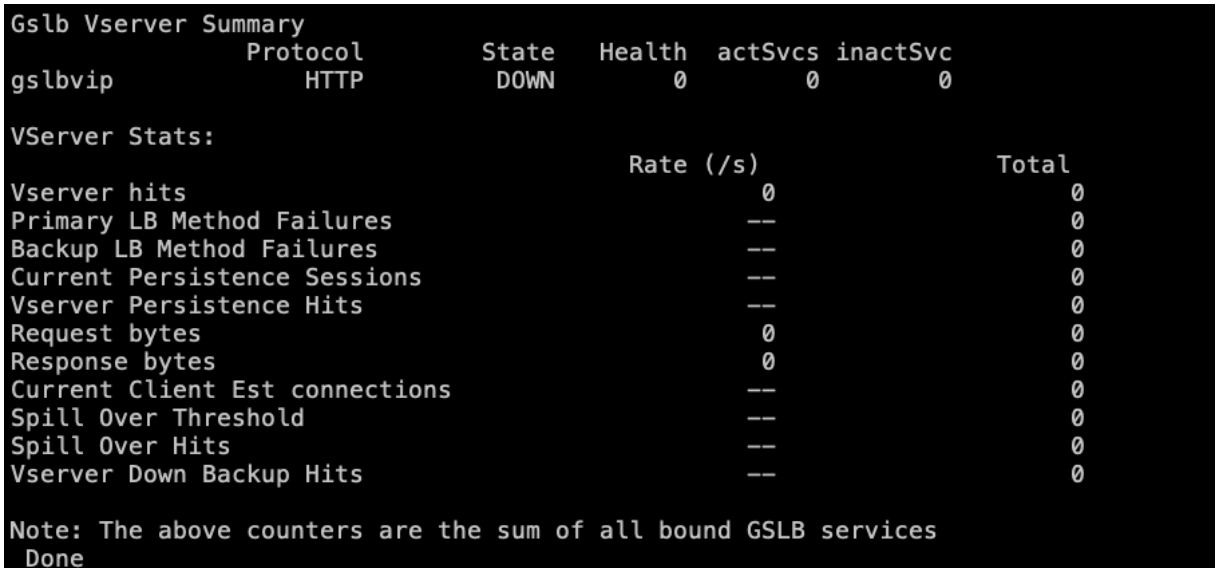
NetScaler バージョン 12.1 ビルド 51.xx 以降では、GSLB 仮想サーバーの統計情報には、仮想サーバーヒット、現在のパーシスタンスセッション、要求バイト、応答バイト、スピルオーバーしきい値、スピルオーバーヒット、現在のクライアント確立接続、仮想サーバーダウンバックアップヒットなどの詳細情報に加えて、次の情報も表示されます。

- プライマリ **LB** メソッドの失敗: プライマリ GSLB メソッドが失敗した回数。
- バックアップ **LB** メソッドの失敗: バックアップ GSLB メソッドが失敗した回数。
- **Vserver** パーシスタンスヒット数: パーシスタンスセッションを通じてリクエストが処理された回数。

GSLB 仮想サーバーの統計には、仮想サーバーにバインドされているサービスグループメンバーの統計も表示されます。

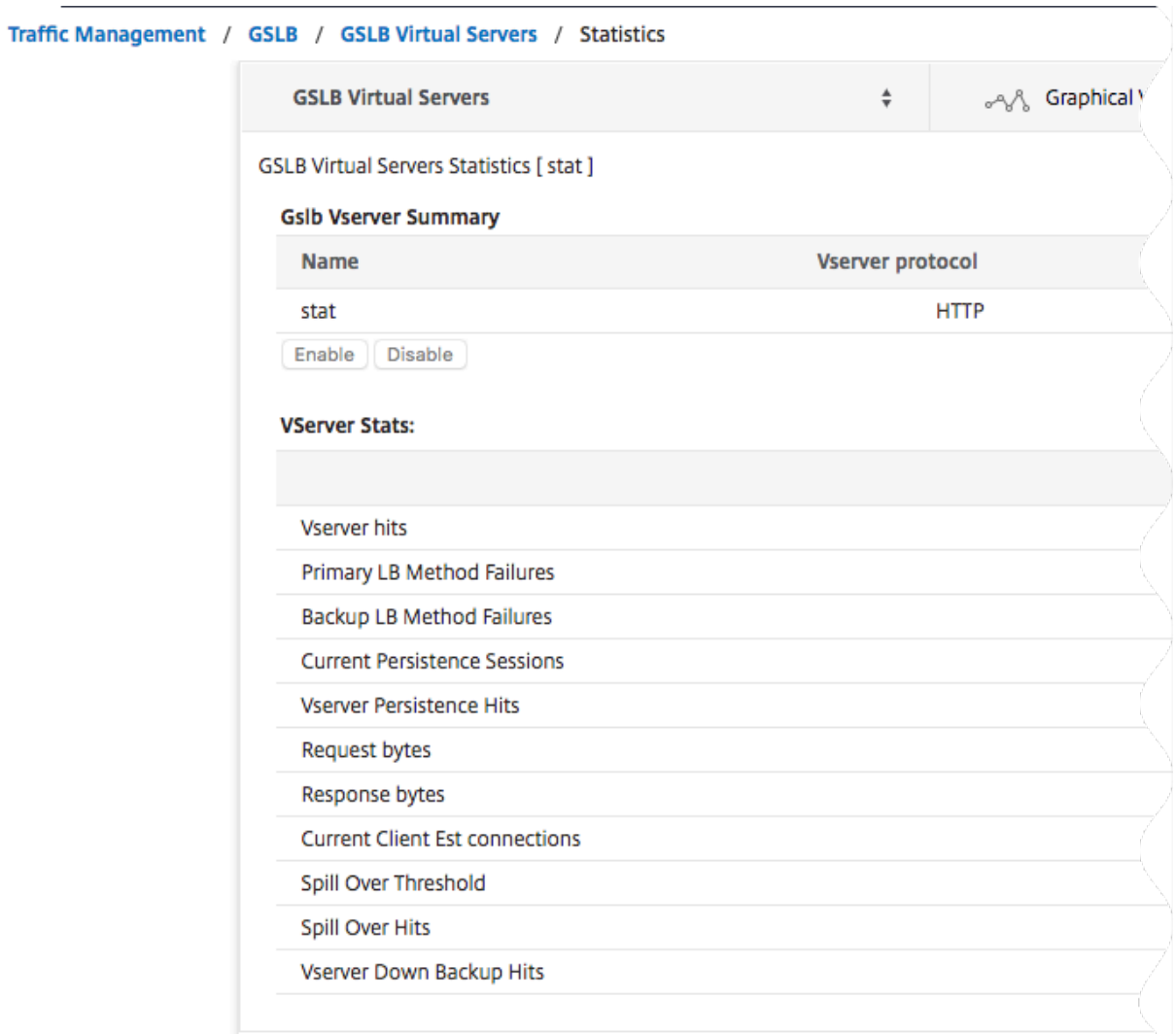
> 注:
> プライマリ方式が静的近接方式で、バックアップ方式が RTT の場合、主方式またはバックアップ方式が失敗する可能性があります。このシナリオでは、LDNS IP に対応するロケーションがない場合、スタティックプロキシミティは失敗し、バックアップ方法が試行されます。統計は以下に基づいて更新されます。
>
> - バックアップ方法が成功すると、プライマリ方式の失敗統計のみが増加します。
> - RTT の計算が成功しなかった場合、バックアップ方法も失敗します。この場合、プライマリメソッドとバックアップメソッドの両方の障害統計が増加します。
>
> バックアップ方法が失敗した場合は、最後の手段であるラウンドロビン方式が使用されます。

次の画像は、CLI からの GSLB 仮想サーバー統計のサンプルです。



```
Gslb Vserver Summary
                Protocol        State   Health  actSvcs inactSvc
gslbvip             HTTP         DOWN        0        0        0

VServer Stats:
                                        Rate (/s)                Total
Vserver hits                                    0                    0
Primary LB Method Failures                     --                    0
Backup LB Method Failures                      --                    0
Current Persistence Sessions                   --                    0
Vserver Persistence Hits                       --                    0
Request bytes                                   0                    0
Response bytes                                  0                    0
Current Client Est connections                 --                    0
Spill Over Threshold                           --                    0
Spill Over Hits                                --                    0
Vserver Down Backup Hits                       --                    0

Note: The above counters are the sum of all bound GSLB services
 Done
```

次の画像は、GUI からの GSLB 仮想サーバー統計のサンプルです。



GSLB サービス統計情報

コマンドラインから`stat gslb service`コマンドを実行するか、構成ユーティリティの **Statistics** リンクをクリックすると、サービスの次の詳細が表示されます。

- リクエストバイト数。このサービスまたは仮想サーバーで受信したリクエストバイトの総数。
- レスポンスバイト。このサービスまたは仮想サーバーが受信した応答バイト数。
- 現在のクライアントは接続を確立しました。確立された状態のクライアント接続の数。
- サービスの現在の負荷。サービスの負荷 (サービスにバインドされた負荷モニターから計算されます)。

要求と応答の数、および現在のクライアントとサーバーの接続数のデータが表示されないか、対応する負荷分散仮想サーバーのデータと同期されない場合があります。

GSLB 仮想サーバーまたはサービスの統計情報の消去

注：この機能は、NetScaler リリース 10.5.e で使用できます。

GSLB 仮想サーバーとサービスの統計を消去できるようになりました。NetScaler ADC には、統計を消去するための次の 2 つのオプションがあります。

- 基本: 仮想サーバー固有の統計情報をクリアしますが、バインドされた GLSB サービスによって提供される統計は保持されます。
- フル: 仮想サーバーとバインドされた GSLB サービスの両方の統計情報をクリアします。

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーの統計を消去するには

コマンドプロンプトで入力します。

```
stat gslb vserver <name> -clearstats <basic | full>
<!--NeedCopy-->
```

例**:**

```
stat gslb vserver Vserver-GSLB-1  - clearstats basic
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** サービスの統計情報を消去するには

コマンドプロンプトで入力します。

```
stat gslb service <name> -clearstats <basic | full>
<!--NeedCopy-->
```

例**:**

```
stat gslb service service-GSLB-1  - clearstats basic
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** 仮想サーバーの統計を消去するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. GSLB 仮想サーバーを選択し、[統計] をクリックし、[クリア] をクリックします。
3. 「クリア」ドロップダウンリストから、「ベーシック」または「フル」を選択し、「**OK**」をクリックします。

構成ユーティリティを使用して **GSLB** サービスの統計情報を消去するには

1. **[**** トラフィック管理] > [**GSLB**] > [サービス] に移動します。*****

2. GSLB サービスを選択し、[統計] をクリックし、[クリア] をクリックします。
3.「クリア」ドロップダウンリストから、「ベーシック」または「フル」を選択し、「**OK**」をクリックします。

GSLB 仮想サーバーの有効化と無効化

GSLB 仮想サーバーを作成すると、デフォルトで有効になります。GSLB 仮想サーバーを無効にすると、DNS 要求を受信しても、NetScaler ADC アプライアンスは構成されている GSLB メソッドに基づいて GSLB を決定しません。代わりに、DNS クエリへの応答には、仮想サーバーにバインドされているすべてのサービスの IP アドレスが含まれます。

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーを有効または無効にするには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
enable gslb vserver <name>@

disable gslb vserver <name>@
<!--NeedCopy-->
```

例**:**

```
enable gslb vserver Vserver-GSLB-1
disable gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** 仮想サーバーを有効または無効にするには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. 仮想サーバーを選択し、アクションリストから [有効化**]** または **[**** 無効化 ****]** を選択します。

ユースケース**-GSLB** 仮想サーバー

GSLB 仮想サーバーを構成できるユースケースの一部を次に示します。

- GSLB セットアップを障害から保護するように GSLB 仮想サーバーを構成する
- GSLB でのパーシステンスの設定
- GSLB API メソッドの設定

GSLB サービスを GSLB 仮想サーバーにバインドする

August 15, 2023

GSLB サービスと仮想サーバーを構成したら、関連する GSLB サービスを GSLB 仮想サーバーにバインドして構成を有効にする必要があります。

コマンドラインインターフェイスを使用して GSLB サービスを GSLB 仮想サーバーにバインドするには

コマンドプロンプトで次のコマンドを入力して、GSLB サービスを GSLB 仮想サーバーにバインドし、構成を確認します。

```
bind gslb vserver <name> -serviceName <string>

show gslb vserver <name>
<!--NeedCopy-->
```

例:

```
bind gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して GSLB サービスを GSLB 仮想サーバーからバインド解除するには

コマンドプロンプトで入力します。

```
unbind gslb vserver <name> -serviceName <string>
<!--NeedCopy-->
```

構成ユーティリティを使用して GSLB サービスをバインドするには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動し、仮想サーバーをダブルクリックします。
2. 「ドメイン」セクションをクリックし、ドメインを構成してドメインをバインドします。

ドメインを GSLB 仮想サーバーにバインドする

August 15, 2023

NetScaler アプライアンスをドメインの信頼できる DNS サーバーにするには、ドメインを GSLB 仮想サーバーにバインドする必要があります。ドメインを GSLB 仮想サーバーにバインドすると、NetScaler アプライアンスは GSLB 仮想サーバーの名前を含むドメインのアドレスレコードを追加します。GSLB ドメインの権限 (SOA) レコードとネームサーバー (NS) レコードを手動で追加する必要があります。

SOA レコードと NS レコードの設定の詳細については、「ドメインネームシステム」を参照してください。

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーにドメインをバインドするには

コマンドプロンプトで次のコマンドを入力して、ドメインを GSLB 仮想サーバーにバインドし、構成を確認します。

```
bind gslb vserver <name> -domainName <string>
show gslb vserver <name>
<!--NeedCopy-->
```

例:

```
bind gslb vserver Vserver-GSLB-1 -domainName www.mycompany.com
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーから **GSLB** ドメインをバインド解除するには

コマンドプロンプトで入力します。

```
unbind gslb vserver <name> -domainName <string>
<!--NeedCopy-->
```

構成ユーティリティを使用してドメインを **GSLB** 仮想サーバーにバインドするには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. GSLB 仮想サーバーペインで、ドメインをバインドする GSLB 仮想サーバー (たとえば、Vserver-GSLB-1) を選択し、「開く」をクリックします。
3. GSLB 仮想サーバーの構成ダイアログボックスの「ドメイン」タブで、次のいずれかを実行します。
   - 新しいドメインを作成するには、[追加] をクリックします。
   - 既存のドメインを変更するには、ドメインを選択し、「開く」をクリックします。
4. 「GSLB ドメインの作成」または「GSLB ドメインの設定」ダイアログボックスで、次に示すように次のパラメータの値を指定します。
   - ドメイン名 * —ドメイン名 (www.mycompany.com など)

* 必須パラメータ

5. ［作成］をクリックします。
6. 「OK」をクリックします。

コマンドラインインターフェイスを使用してドメインの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat gslb domain <name>
<!--NeedCopy-->
```

例:

```
stat gslb domain www.mycompany.com
<!--NeedCopy-->
```

注：特定の GSLB ドメインの統計情報を表示するには、NetScaler アプライアンスに追加されたドメインの名前を正確に入力してください。ドメイン名を指定しない場合、または間違ったドメイン名を指定した場合、設定されているすべての GSLB ドメインの統計が表示されます。

構成ユーティリティを使用してドメインの統計情報を表示するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. GSLB 仮想サーバーペインで、GSLB 仮想サーバー (たとえば、Vserver-GSLB-1) を選択し、「開く」をクリックします。
3. GSLB 仮想サーバーの構成ダイアログボックスの「ドメイン」タブでドメインを選択し、「統計」をクリックします。

コマンドラインを使用して **GSLB** ドメインにバインドされたエンティティの設定の詳細を表示するには

注：この機能は、NetScaler リリース 10.5.e で使用できます。

コマンドプロンプトで入力します。

```
show gslb domain <name>
<!--NeedCopy-->
```

例:

```
show gslb domain gslb1.com
       gslb1.com
       gvs1 - HTTP    state: DOWN
       DNS Record Type: A
       Configured Method: LEASTCONNECTION
```



```
        Backup Method: ROUNDROBIN
        Persistence Type: NONE
        Empty Down Response: DISABLED
        Multi IP Response: DISABLED
        Dynamic Weights: DISABLED

        gsvc1 (10.102.239.165: 80)- HTTP State: DOWN    Weight: 1
               Dynamic Weight: 0    Cumulative Weight: 1
               Effective State: DOWN
               Threshold : BELOW

               Monitor Name : http
                      State: DOWN   Weight: 1
                      Probes: 144   Failed [Total: 144 Current: 144]
                      Last response: Failure - TCP syn sent, reset
                         received.
                      Response Time: 2000 millisec

        gsvc2 (10.102.239.179: 80)- HTTP State: DOWN    Weight: 1
               Dynamic Weight: 0    Cumulative Weight: 1
               Effective State: DOWN
               Threshold : BELOW

               Monitor Name : http-ecv
                      State: DOWN   Weight: 1
                      Probes: 141   Failed [Total: 141 Current: 141]
                      Last response: Failure - TCP syn sent, reset
                         received.
                      Response Time: 2000 millisec
Done
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** ドメインにバインドされたエンティティの設定の詳細を表示するには

注：この機能は、NetScaler リリース 10.5.e で使用できます。

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動し、仮想サーバーをダブルクリックします。
2. ドメインペインの下のフィールドをクリックします。
3. **GSLB** 仮想サーバードメインバインディングダイアログボックスで、ドメインを選択し、「バインディングを表示」をクリックします。

GSLB のセットアップと構成の例

August 15, 2023

組織には地理的に分散したネットワークがあり、米国、メキシコ、コロンビアに 3 つのデータセンターがあります。これらのロケーションに関連する構成では、これらはそれぞれ US、MX、CO と呼ばれます。各拠点には、同じコンテンツを提供するサーバーファームがあり、セットアップは期待どおりに機能しています。各場所の NetScaler アプライアンスは、ポート 80 の HTTP プロトコルを使用する仮想サーバーを介して構成されます。

組織は、各サイトにサイト識別子を追加することで GSLB 設定を実装しました。サイト識別子には、NetScaler アプライアンスが所有し、GSLB 通信に使用されるサイト名と IP アドレスが含まれます。
各サイトには、アプライアンスのローカルサイトがあります。また、各サイトには、ローカルアプライアンスから離れた 2 つのサイトがあります。各サイトに、同じ名前で GSLB 仮想サーバーが作成されます。この仮想サーバーは組織の Web サイトをグローバルに識別し、IP アドレスは関連付けられません。
セットアップには、それぞれの仮想サーバーの IP アドレス、プロトコル、およびポート番号を指定することにより、各 GSLB サイトで構成された負荷分散仮想サーバーを指すように構成された GSLB サービスもあります。これらのサービスは GSLB 仮想サーバーにバインドされます。

注: 以下の手順では、コマンドは GSLB サイトのプライベート IP アドレスを使用します。公開サイトや GSLB サービスの場合、これらのサイトには必ずパブリック IP アドレスを使用してください。

次の表は、この例で使用されている IP アドレスと場所の一覧です。

IP アドレス	位置情報
10.3.1.101	ローカル NetScaler のサイト IP。
172.16.1.101	リモートロケーション Site-MX のサイト IP。
192.168.1.101	リモートロケーション Site-co. のサイト IP
172.16.1.100	リモートロケーション Site-MX のサービス IP。
10.3.1.100	ローカル NetScaler のサービス IP。
192.168.1.100	リモートロケーション Site-co のサービス IP

GSLB サイトを追加する際、サイトがインターネット経由でのみ通信する場合は、「Public IP」フィールドを使用してください。たとえば、GSLB サイト間にサイト間 VPN 接続がない場合です。

CLI コマンドを使用して **NetScaler** アプライアンスの **GSLB** セットアップを構成するには

1. GSLB 機能をまだ有効化していない場合は、有効にします。

```
enable ns feature gslb
<!--NeedCopy-->
```

2. ローカル GSLB サイトを追加するための SNIP を特定してください。
3. ローカルの NetScaler アプライアンスの GSLB サイトを追加します。


```
add gslb site site-US 10.3.1.101
<!--NeedCopy-->
```

4. リモート NetScaler アプライアンスの GSLB サイトを追加します。

```
add gslb site site-MX 172.16.1.101
add gslb site site-CO 192.168.1.101
<!--NeedCopy-->
```

5. GSLB セットアップで使用されているサービスを参照する GSLB 仮想サーバーを追加します。

```
add gslb vserver gslb-lb HTTP
<!--NeedCopy-->
```

6. GSLB セットアップに参加している各サイトに GSLB サービスを追加します。

```
add gslb service gslb_SVC30 172.16.1.100 HTTP 80 -siteName site-MX
add gslb service gslb_SVC10 10.3.1.100 HTTP 80 -siteName site-US
add gslb service gslb_SVC20 192.168.1.100 HTTP 80 -siteName site-
   CO
<!--NeedCopy-->
```

7. GSLB サービスを GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gslb-lb -serviceName gslb_SVC10
bind gslb vserver gslb-lb -serviceName gslb_SVC20
bind gslb vserver gslb-lb -serviceName gslb_SVC30
<!--NeedCopy-->
```

8. ドメインを GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gslb-lb -domainName www.mycompany.com -TTL 30
<!--NeedCopy-->
```

9. DNS クエリをリッスンする ADNS サービスを追加します。

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

GSLB セットアップで構成を同期する

August 15, 2023

通常、GSLB セットアップにはデータセンターがいくつかあり、各データセンターに GSLB サイトが設定されます。GSLB に参加している各 NetScaler で、1 つの GSLB サイトをローカルサイトとして、他の GSLB サイトをリモートサイトとして構成します。後で別の GSLB サイトを追加する場合は、すべての GSLB サイトにわたる構成が同一で

あることを確認する必要があります。NetScaler ADC の GSLB 構成同期オプションを使用して、GSLB サイト間で構成を同期できます。

同期オプションを使用する NetScaler ADC アプライアンスは、「メインサイト」と呼ばれ、構成がコピーされる GSLB サイトは「下位サイト」と呼ばれます。GSLB 設定を同期すると、GSLB セットアップに参加しているすべての GSLB サイトの構成は、メインサイトの構成と同様になります。

同期は親サイトでのみ行われます。同期は GSLB 子サイトの構成には影響しません。これは、親サイトと子サイトの構成が同一ではないためです。子サイトの構成は、そのサイトとその親サイトの詳細のみで構成されます。また、GSLB サービスは必ずしも子サイトで設定する必要はありません。

- メインノードは、メインノードと下位ノードの構成の違いを検出し、下位ノードの構成をメインノードに似せるように変更します。

  同期を強制すると（「強制同期」オプションを使用）、アプライアンスは下位ノードから GSLB 設定を削除し、メインノードと類似するように下位ノードを構成します。

- 同期中にコマンドが失敗した場合、同期は中断されず、エラー・メッセージは **/var/netscaler/gslb** ディレクトリ内の **.err** ファイルに記録されます。

- 同期は親サイトでのみ行われます。同期は GSLB 子サイトの構成には影響しません。これは、親サイトと子サイトの構成が同一ではないためです。子サイトの構成は、そのサイトとその親サイトの詳細のみで構成されます。また、GSLB サービスは必ずしも子サイトで設定する必要はありません。

- 内部ユーザーログインを無効にすると、GSLB 自動同期は SSH キーを使用して設定を同期します。ただし、パーティション環境で GSLB 自動同期を使用するには、内部ユーザーログインを有効にし、ローカルおよびリモート GSLB サイトのパーティションユーザー名が同じであることを確認する必要があります。

> 注
>
> - リモート GSLB サイトの RPC ノードで、リモートサイトの IP（クラスタセットアップ用のクラスタ IP アドレス）とポート（RPC の場合は 3010、セキュア RPC の場合は 3008）を指定して、自動同期接続を受け入れるようにファイアウォールを構成します。ほとんどの場合、リモートサイトに到達するデフォルトルートが管理サブネット内にある場合は、NSIP が送信元 IP アドレスとして使用されます。
>
> 異なる送信元 IP アドレスを設定するには、GSLB サイトの IP アドレスと SNIP が別のサブネットにある必要があります。また、GSLB サイト IP サブネットを介してリモートサイト IP アドレスへの明示的なルートを定義する必要があります。
>
> セキュリティを強化するには、内部ユーザーアカウントと RPC ノードのパスワードを変更することをお勧めします。内部ユーザーアカウントのパスワードは、RPC ノードパスワードによって変更されます。詳細については、「RPC ノードのパスワードを変更する」を参照してください。

saveconfig オプションを使用すると、同期プロセスに参加しているサイトは、次の方法で自動的に構成を保存します。

リモート GSLB サイトの RPC ノードで、リモートサイトの IP（クラスタセットアップ用のクラスタ IP アドレス）とポート（RPC の場合は 3010、セキュア RPC の場合は 3008）を指定して、自動同期接続を受け入れるようにファイアウォールを構成します。ほとんどの場合、リモートサイトに到達するデフォルトルートが管理サブネット内にある場合は、NSIP が送信元 IP アドレスとして使用されます。

異なる送信元 IP アドレスを設定するには、GSLB サイトの IP アドレスと SNIP が別のサブネットにある必要があります。また、GSLB サイト IP サブネットを介してリモートサイト IP アドレスへの明示的なルートを定義する必要があります。RPC ノードの送信元 IP アドレスは各 NetScaler ADC アプライアンスに固有であるため、GSLB に参加しているサイト間でソース IP アドレスを同期できません。そのため、（sync gslb config-ForceSync コマンドを使用するか、GUI で ForceSync オプションを選択して）同期を強制した後は、他の NetScaler アプライアンスのソース IP アドレスを手動で変更する必要があります。ポート 22 は、データベースファイルをリモートサイトに同期させるためにも必要です。

すべての **GSLB** サイトでの構成の同期にかかる時間を改善するには

コマンドプロンプトで TCP プロファイル設定を次のように構成します。

```
set tcpprofile nstcp_internal_apps -bufferSize 4194304 -sendBuffsize
    4194304 -tcpmode ENDPOINT
<!--NeedCopy-->
```

同期の制限

- メインサイトでは、リモート GSLB サイトの名前は、それらのサイトをホストしている NetScaler ADC アプライアンスで構成されたサイトの名前と同じである必要があります。
- 同期中に、トラフィックの中断が発生することがあります。
- NetScaler ADC は、構成の最大 20 万行までの同期がテストされています。
- 同期が失敗することがあります。
  - 流出方法が「接続」から「ダイナミック接続」に変更された場合。
  - メインノード上の GSLB 仮想サーバーにバインドされた GSLB サービスのサイトプレフィックスを交換し、同期を試みる場合。
  - RPC ノードのパスワードが NSIP アドレスとループバック IP アドレスで異なる場合。
  - 同じ NetScaler ADC アプライアンスの異なるパーティションで構成されている GSLB サイトで同期を実行する場合。
- GSLB サイトをハイアベイラビリティ（HA）ペアとして設定した場合、プライマリノードとセカンダリノードの RPC ノードのパスワードは同じである必要があります。
- GSLB 設定の一部である GLSB エンティティの名前を変更する場合（「show gslb runningConfig」コマンドを使用して GSLB 設定を表示します）。設定を他の GSLB サイトに同期するには、強制同期オプションを使用する必要があります。

> 注：
>
> - 増分同期では、設定を他の GSLB サイトに同期するために強制同期オプションを使用する必要はありません。これは、NetScaler ADC リリース 13.0 ビルド 79.x 以降の起動に適用されます。

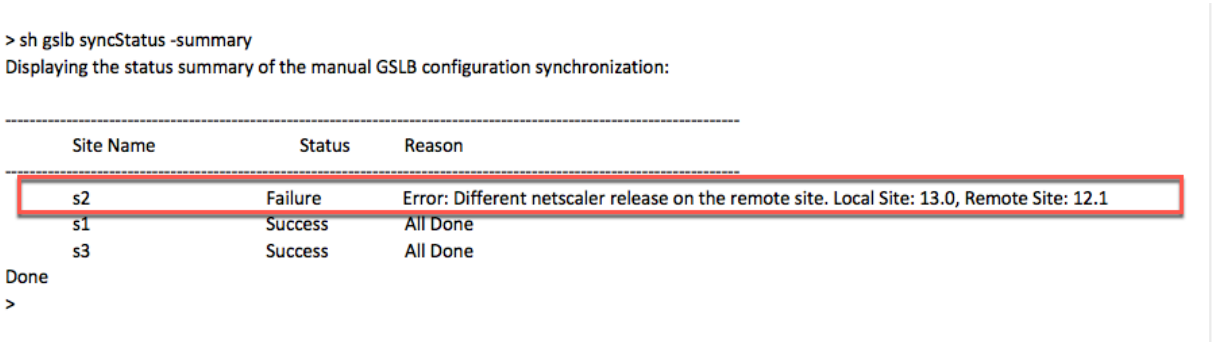
注：GSLB 設定の一部の設定による制限を克服するには、強制同期オプションを使用できます。ただし、強制同期オプションを使用すると、GSLB エンティティは削除されて設定に再追加され、GSLB 統計はゼロにリセットされます。そのため、構成の変更中にトラフィックが中断されます。

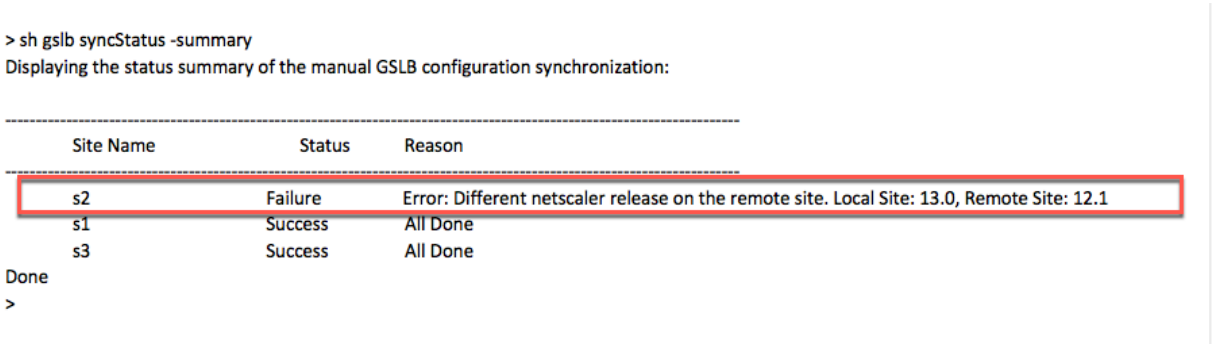
GSLB 設定の同期を開始する前に注意すべき点

GSLB セットアップの同期を開始する前に、次のことを確認してください。

- メインサイトを含むすべての GSLB サイトでは、対応する GSLB サイトの IP アドレスに対して管理アクセスと SSH を有効にする必要があります。GSLB サイトの IP アドレスは、NetScaler ADC アプライアンスが所有する IP アドレスである必要があります。GSLB サイトの IP アドレスの追加と管理アクセスの有効化の詳細については、「基本的な GSLB サイトの構成」を参照してください。
- メインサイトと見なされる NetScaler ADC アプライアンスの GSLB 構成は完全であり、すべてのサイトでコピーするのが適切です。
- GSLB 設定を初めて同期する場合は、GSLB に参加するすべてのサイトに、それぞれのローカルサイトの GSLB サイトエンティティが必要です。
- 設計上、同じ構成を持たないサイトを同期していません。
- メインサイトと下位サイトは同じ NetScaler ADC バージョンを実行します。リリース 12.1、ビルド 50.x 以降、アプライアンスは同期を開始する前にメインサイトと下位サイトのファームウェアバージョンをチェックします。メインサイトと下位サイトが異なるバージョンを実行している場合、バージョン間で互換性のない変更がプッシュされないように、そのリモートサイトの同期は中止されます。また、同期が中止されたサイトの詳細を示すエラーメッセージが表示されます。

  次の図は、CLI と GUI からのエラーメッセージの例を示しています。





> 重要
>
> 次のディレクトリは、GSLB 設定同期の一部として同期されます。
>
> - /var/netscaler/locdb/
> - /var/netscaler/ssl/
> - /var/netscaler/inbuilt_db/

GSLB に参加しているサイト間で手動同期

August 15, 2023

マスターサイトとスレーブサイト間の GSLB 設定の手動同期は、次の方法で実行されます。

- マスターサイトは、自社サイトとスレーブサイトの構成の違いを検出します。
- マスターサイトは、構成の違いをスレーブサイトに適用します。
- マスターサイトは、GSLB セットアップ内のすべてのスレーブサイトと構成の同期を実行し、同期プロセスを完了します。

重要:GSLB 構成を同期した後は、どの GSLB サイトでも構成をロールバックすることはできません。同期処理によってリモートサイトの構成が上書きされないことが確実な場合にのみ、同期を実行してください。ローカルサイトとリモートサイトの構成が設計上異なる場合、サイト同期は望ましくなく、サイトが停止します。一部のコマンドが失敗し、一部のコマンドが成功しても、成功したコマンドはロールバックされません。

注意事項

- 同期を強制すると（「強制同期」オプションを使用）、NetScaler アプライアンスはスレーブサイトから GSLB 構成を削除します。次に、マスターサイトはスレーブサイトを自身のサイトと同様になるように構成します。
- 同期中にコマンドが失敗しても、同期は中止されません。エラーメッセージは /var/netscaler/gslb ディレクトリの.err ファイルに記録されます。

- saveconfigオプションを使用すると、同期プロセスに参加しているサイトは、次の方法で構成を自動的に保存します。
  - マスターサイトは、同期処理を開始する直前に構成を保存します。
  - スレーブサイトは、同期処理が完了した後に設定を保存します。スレーブサイトは、設定の違いが正常に適用された場合にのみ構成を保存します。スレーブサイトで同期が失敗した場合は、障害の原因を手動で調査し、修正措置を講じる必要があります。

CLI を使用して **GSLB** 設定を同期するには**:**

コマンドプロンプトで次のコマンドを入力して GSLB サイトを同期し、構成を確認します。

```
sync gslb config [-preview | -forceSync <string> | -nowarn | -
   saveconfig] [-debug]
show gslb syncStatus
<!--NeedCopy-->
```

例**:**

```
sync gslb config

[WARNING]: Syncing config may cause configuration loss on other site.

Please confirm whether you want to sync-config (Y/N)? [N]:y

Sync Time: Dec 9 2011 10:56:9

Retrieving local site info: ok

Retrieving all participating gslb sites info: ok

Gslb_site1[Master]:

Getting Config: ok

Gslb_site2[Slave]:

Getting Config: ok

Comparing config: ok

Applying changes: ok

Done
<!--NeedCopy-->
```

GUI を使用して **GSLB** 設定を同期するには**:**

1. [** トラフィック管理] > [GSLB] **[ダッシュボード] に移動します。
2. 「自動同期 GSLB」をクリックし、「強制同期」を選択します。
3. 「**GSLB** サイト名」で、マスターノード構成と同期する GSLB サイトを選択します。

GSLB 同期のプレビュー

GSLB 同期操作をプレビューすると、マスターノードと各スレーブノードの違いを確認できます。不一致がある場合は、GSLB 設定を同期する前にトラブルシューティングできます。

CLI を使用して **GSLB** 同期出力をプレビューするには**:**

コマンドプロンプトで、次のコマンドを入力します。

```
sync gslb config -preview
<!--NeedCopy-->
```

GUI を使用して **GSLB** 同期出力をプレビューするには**:**

1. **[**** 設定] > [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動します。**
2. 「自動同期 GSLB」をクリックし、「プレビュー」を選択します。
3. [実行] をクリックします。
   設定に相違がある場合は、進行状況ウィンドウに表示されます。

同期処理中にトリガーされたコマンドのデバッグ

同期処理中にトリガーされた各コマンドのステータス (成功または失敗) を表示し、それに応じてトラブルシューティングを行うことができます。

CLI を使用して **GSLB** 同期コマンドをデバッグするには**:**

コマンドプロンプトで、次のコマンドを入力します。

```
sync gslb config -debug
<!--NeedCopy-->
```

GUI を使用して **GSLB** 同期コマンドをデバッグするには**:**

1. **[**** 設定] > [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動します。**
2. 「自動同期 GSLB」をクリックし、「デバッグ」を選択します。
3. [実行] をクリックします。進行状況ウィンドウには、同期中にトリガーされた各コマンドのステータスが表示されます。

GSLB に参加しているサイト間でリアルタイム同期

October 25, 2023

`AutomaticConfigSync` パラメータを使用すると、メインサイトのリアルタイム GSLB 構成をすべての下位サイトに自動的に同期できます。設定を同期するために AutoSync オプションを手動でトリガーする必要はありません。

増分同期または完全同期を使用して、メインサイトの GSLB 構成をすべての下位サイトに自動的に同期できます。GSLBSyncMode パラメータを使用すると、同期モードを選択できます。

> 注記:
>
> NetScaler ADC リリース 13.0 ビルド 79.x 以降、GSLB 同期の増分同期がサポートされています。デフォルトでは、同期は差分同期を使用して実行されます。増分同期は、IncrementalSyncパラメータを有効にすることで実行できます。詳細については、GSLB 設定の増分同期を参照してください。

リアルタイム同期機能の使用に関するベストプラクティス

- サイトとして参加しているすべての NetScaler アプライアンスの NetScaler ソフトウェアバージョンが同じであることが推奨されます。
- RPC ノードのパスワードを変更するには、まず下位サイトでパスワードを変更し、次にメインサイトでパスワードを変更します。
- GSLB に参加している各サイトでローカル GSLB サイトを構成します。
- 構成が実行されるサイトの 1 つで AutomaticConfigSync を有効にします。このサイトは最終的に他の GSLB サイトと同期されます。
- 新しい設定がある場合、または既存の設定に変更が加えられた場合は、show gslb syncStatusコマンドを使用してステータスを確認し、変更がすべてのサイトで同期されているかどうか、またはエラーが発生したかどうかを確認します。
- RSYNC ポートモニタリングを有効にする必要があります。

CLI を使用してリアルタイム同期を有効にするには

コマンドプロンプトで入力します。

```
1 set gslb parameter [ - automaticConfigSync (ENABLED | DISABLED)] [-
     MEPKeepAliveTimeout <secs>]  [-GSLBSyncMode ( IncrementalSync |
     FullSync )] [-GSLBSyncLocFiles ( ENABLED | DISABLED)] [-
     GslbConfigSyncMonitor ( ENABLED | DISABLED )] [-GSLBSyncInterval <
     secs>] [-GSLBSyncSaveConfigCommand ( ENABLED | DISABLED )]
2 <!--NeedCopy-->
```

例**:**

```
1 set gslb parameter  - automaticConfigSync ENABLED
2 <!--NeedCopy-->
```

リアルタイム同期には、次の設定可能なパラメータがあります。

- **gslbSyncMode**-メインサイトからリモートサイトに構成が同期されるモード。
  - 可能な値: インクリメンタル同期、FullSync
  - デフォルト値:IncrementalSync

- **gslbSyncLocFiles**-GSLB 設定の同期中に、デフォルトでは、場所 DB ファイルの変更が検出され、自動的に同期されます。場所 DB ディレクトリは頻繁に変更されないため、管理者は場所 DB ファイルの自動同期を無効にできます。代わりに、管理者は場所 DB ファイルを GSLB 下位サイトに手動でコピーする必要があります。場所 DB ファイルの同期には長い時間がかかります。したがって、これを避けると、全体の同期時間が短縮されます。

  場所 **DB** ファイルの自動同期を無効にする例**:**

```
set gslb parameter -GSLBSyncMode IncrementalSync -
   GSLBSyncLocFiles DISABLED
<!--NeedCopy-->
```

- **gslbConfigSyncMonitor**: GSLB 構成同期モニタパラメータを有効にして、下位サイトの RSYNC ポート（リモート GSLB サイト IP アドレス上の SSH ポート 22）の状態を監視します。モニタの下位サイトの状態が DOWN と表示されている場合、そのサイトに対する RSYNC 操作はスキップされます。これにより、ダウンしているリモートサイトに接続しようとしたことによる同期の遅延が軽減されます。

  CLI で **RSYNC** ポートモニタリングを有効にする例を示します。

```
set gslb parameter -GSLBSyncMode IncrementalSync -
   GslbConfigSyncMonitor ENABLED
<!--NeedCopy-->
```

- **gslbSyncInterval**: GSLB 設定の同期が行われる時間間隔（秒単位）を設定します。デフォルトでは、GSLB 設定の自動同期機能は、10 秒ごとに自動的に GSLB 設定を同期します。時間間隔は任意の値に変更できます。たとえば、5 秒未満など、この値を低い値に設定しないでください。同期を頻繁に実行すると、管理 CPU の消費が増加する可能性があるためです。

  > 注記:
  >
  > 管理パーティションのセットアップでは、時間間隔はグローバルパラメータであるため、デフォルトパーティションでのみ設定できます。

  同期間隔を設定する例**:**

```
set gslb parameter -AutomaticConfigSync ENABLED -GSLBSyncMode
   IncrementalSync -GSLBSyncInterval 7
<!--NeedCopy-->
```

- **gslbSyncSaveConfigCommand**- この`AutomaticConfigSync`オプションが有効になっている場合、このパラメーターを有効にして、`save ns config`コマンドを従属サイトに同期します。

  「**Save Config**」コマンドの同期を有効にする例**:**

```
set gslb parameter -AutomaticConfigSync ENABLED -
   GSLBSyncSaveConfigCommand ENABLED
<!--NeedCopy-->
```

  次のような特定のシナリオでは、`save ns config`コマンドは下位サイトと同期されません。

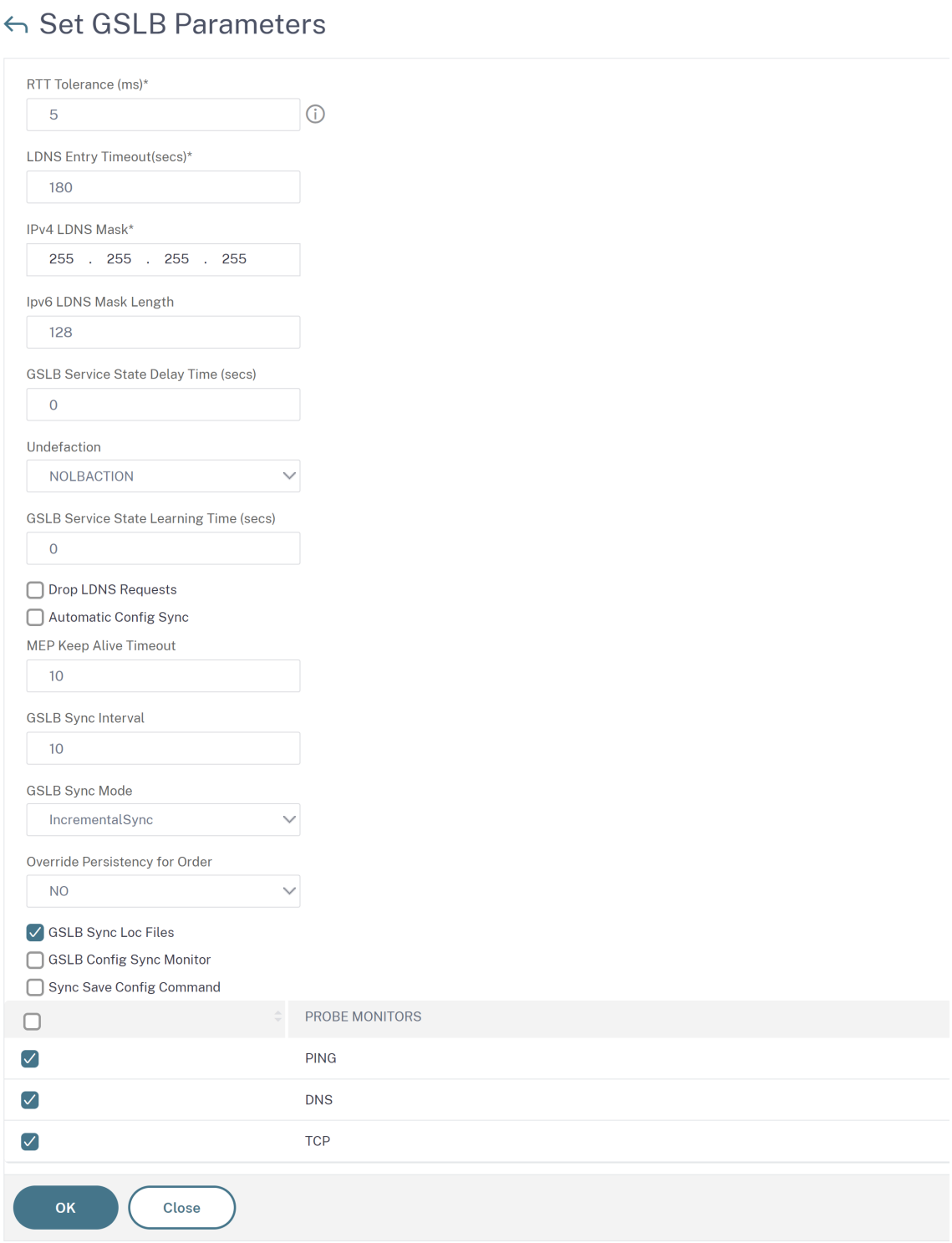
- コンフィグをメインサイトに保存すると、下位サイトがダウンしているか、アクセスできません。
- 下位サイトで構成が失敗しました。

GUI を使用してリアルタイム同期を有効にするには

1. [設定] > [トラフィック管理] > [**GSLB**] > [**GSLB** 設定の変更] に移動します。
2. **[GSLB** パラメータの設定] ページで、次の操作を実行できます。
   - リアルタイム GSLB 設定を自動的に同期するには、自動 **ConfigSync** を選択します。

   注: このオプションは、構成が実行されるサイトでのみ有効にする必要があります。

   - 自動 GSLB 設定の同期間隔を設定するには、[**GSLB** 同期間隔] フィールドに時間を秒単位で入力します。
   - RSYNC ポートの監視を有効にするには、[**GSLB** 構成同期モニタ] チェックボックスをオンにします。
   - 場所 DB ファイルの自動同期を無効にするには、[**GSLB** 同期ロックファイル] チェックボックスをオフにします。
   - 下位サイトへの`save ns config`コマンドの同期を有効にするには、[Save Config Command の同期（**Sync Save Config Command** ）] チェックボックスをオンにします。



次のトピックの詳細については、GSLB に参加しているサイト間の手動同期を参照してください。

- GSLB 同期のプレビュー
- 同期プロセス中にトリガーされたコマンドのデバッグ

注意事項

- リアルタイム同期に関連する統合ログファイルは /var/netscaler/gslb/periodic_sync.log ディレクトリに保存されます。
- デフォルトの設定ファイルは/var/netscaler/gslb_sync/ ディレクトリに保存されます。
- メインサイトでは、次のディレクトリ構造を使用しています。
  - メインサイトは、すべてのファイルを /var/netscaler/gslb_sync/master ディレクトリに保存します。
  - メインサイトは、下位サイトと同期する必要がある構成ファイルを/var/netscaler/gslb_sync/master/gslbconf/ディレクトリに保存します。
  - すべての下位サイトから取得されたステータスファイルは、/var/netscaler/gslb_sync/master/slavestatus/ディレクトリに保存されます。
- 下位サイトは、次のディレクトリ構造を使用します。
  - 下位サイトは、/var/netscaler/gslb_sync/slave/gslbconf ディレクトリから適用する最新の構成ファイルを選択します。
  - 下位サイトは、ステータスファイルを /var/netscaler/gslb_sync/slave/gslbstatus ディレクトリに保存します。
- 管理パーティションのセットアップでは、/var/partitions/`partition name`/netscaler/gslb_sync という場所に同じディレクトリ構造が維持されます。
- すべてのサイトの時計は、協定世界時 (UTC) などのリアルタイム標準に正確に設定する必要があります。

GSLB 設定の増分同期

GSLB 設定の自動同期機能は、メインサイトの構成の変更を 10 秒間隔でチェックし、同期を実行します。この同期間隔値は設定可能です。

差分同期では、最後の同期と後続の同期間隔（10 秒）の間にメインサイトで変更された構成のみが、すべての下位サイト間で同期されます。増分同期がデフォルトの動作です。インクリメンタル構成のみをプッシュすると、構成ファイルのサイズが大幅に減少し、同期時間が短縮されます。差分同期が失敗すると、システムは完全な構成同期を自動的に実行します。

差分同期は、次の方法で実行されます。

- メインサイトは、最新の変更のみで構成された構成ファイルをすべての下位サイトにプッシュします。最新の変更とは、最後の同期とそれ以降の同期間隔（10 秒）の間に変更された設定を指します。
- 各下位サイトは、最新の変更を独自のサイトに適用します。
- ダウン状態の下位サイトでは、差分同期は試行されません。サイトが復帰すると、再び同期が実行されます。
- 下位サイトは、各ステップでステータスログを生成し、特定の場所のファイルにコピーします。
- メインサイトは、指定された場所からステータスログファイルをプルします。

- メインサイトは、すべての下位サイトからのログを組み合わせたログファイルを準備します。
- この結合されたログファイルは、”/var/netscaler/gslb/periodic_sync.log”ファイルに格納されます。

設定ファイルが保存されているディレクトリの詳細については、「注意事項」セクションを参照してください。

CLI を使用して **GSLB** 設定の増分同期を有効にするには

```
set gslb parameter -AutomaticConfigSync (ENABLED | DISABLED) -
    GSLBSyncMode (IncrementalSync | FullSync) -GslbConfigSyncMonitor (
    ENABLED | DISABLED) -GSLBSyncInterval <secs> -GSLBSyncLocFiles (
    ENABLED | DISABLED)
<!--NeedCopy-->
```

例**:**

```
set gslb parameter -AutomaticConfigSync ENABLED -GSLBSyncMode
    IncrementalSync
<!--NeedCopy-->
```

GUI を使用して **GSLB** 増分同期を有効にするには

1. [トラフィック管理**] > [GSLB] > [**ダッシュボード**] > [GSLB** 設定の変更] に移動します。
2. [**GSLB** パラメータの設定] ページで、[**GSLB **** 同期モード**]** ドロップダウンメニューから **[Incremental-Sync****] を選択します。

GSLB 設定の完全同期

メインサイトで設定が変更されると、メインサイトの完全な GSLB 実行構成がすべての下位サイトにプッシュされます。
差分同期が構成されている場合でも、メインサイトが下位サイトの構成ステータスを認識しない場合、完全同期が実行されます。このようなシナリオのいくつかは次のとおりです。

- GSLB 設定の自動同期機能を初めて有効にします。
- NetScaler ADC アプライアンスを再起動します。
- GSLB の展開には複数のメインサイトがあり、別のメインサイトがアクティブなメインサイトになります。
- GSLB デプロイメントに新しい下位サイトを追加します。

GSLB 設定の完全同期は、次の方法で実行されます。

- メインサイトは、最新の構成ファイルをすべての下位サイトにプッシュします。
- 各下位サイトは、メインサイトから送信された最新の構成ファイルと、独自の構成を比較します。下位サイトは、構成の違いを特定し、独自のサイトに対してデルタ構成を適用します。
- 下位サイトは、各ステップでステータスログを生成し、特定の場所のファイルにコピーします。

- メインサイトは、指定された場所からステータスログファイルをプルします。
- メインサイトは、すべての下位サイトからのログを組み合わせたログファイルを準備します。
- この結合されたログファイルは、”/var/netscaler/gslb/periodic_sync.log”ファイルに格納されます。

自動同期中にサイトを手動で（`sync gslb config` コマンドで）同期しようとすると、「同期中」というエラーメッセージが表示されます。自動同期は、手動で同期しているサイトでは起動できません。

> 注:
>
> NetScaler 12.1 ビルド 49.37 以降、GSLB 構成を同期すると SNMP トラップが生成されます。リアルタイム同期では、最初の SNMP トラップの同期ステータスが失敗としてキャプチャされます。2 番目の SNMP トラップは、実際の同期ステータスの最初のトラップの直後に自動的に生成されるため、このステータスは無視できます。ただし、2 回目の試行で同期が失敗した場合、同期ステータスが以前の同期ステータスから変更されていないため、SNMP トラップは生成されません。

トラップを生成するための NetScaler ADC アプライアンスの構成の詳細については、「SNMP トラップを生成するように NetScaler ADC を構成する」を参照してください。

CLI を使用して **GSLB** 完全同期を有効にするには

```
set gslb parameter -GSLBSyncMode (IncrementalSync | FullSync)
<!--NeedCopy-->
```

例**:**

```
set gslb parameter -GSLBSyncMode FullSync
<!--NeedCopy-->
```

GUI を使用して GSLB インクリメンタル同期を有効にするには、次の手順を実行します。

1. [トラフィック管理**] > [GSLB] > [**ダッシュボード**] > [GSLB** 設定の変更] に移動します。
2. [**GSLB** パラメータの設定] ページで、[**GSLB** 同期モード**]** ドロップダウンメニューから **[FullSync**] を選択します。

GSLB 展開内の複数のメインサイト

NetScaler ADC アプライアンスは、アクティブ/パッシブ展開で複数のメインサイトをサポートします。GSLB メインサイトの障害に対処するために、GSLB 展開に 2 つのメインサイトを用意することをお勧めします。メインサイトが 2 つあると、GSLB 設定の同期の単一点障害を回避できます。いつでも、ユーザーから GSLB 設定をアクティブに処理できるメインサイトは 1 つだけです。構成の変更が複数のメインサイトで同時に実行されると、構成の不一致または構成が失われる可能性があります。したがって、一度に 1 つのメインサイトからの構成変更のみを実行し、アクティブなメインサイトに障害が発生した場合は、もう一方のメインサイトをバックアップとして使用することをお勧めします。

> ** 注:
>
> GSLB 展開で複数のメインサイトを使用する場合は、RSYNC モニタリングを有効にする必要があります。

GSLB 構成同期のメインサイトの 1 つとして GSLB ノードを作成するには、次のコマンドを実行します。

```
set gslb parameter -automaticConfigSync Enabled
<!--NeedCopy-->
```

GSLB 同期のステータスとサマリーを表示する

August 15, 2023

GSLB 構成が GSLB サイト間で同期されると、前回の GSLB 同期操作の詳細なステータスと概要を表示できます。これは、手動とリアルタイムの GSLB 同期の両方に適用されます。

CLI を使用して GSLB 同期のステータスまたは概要を表示するには

コマンドプロンプトで入力します。

```
show gslb sync status
<!--NeedCopy-->
```

または

```
show gslb syncStatus -summary
<!--NeedCopy-->
```

GSLB 手動同期の設定出力例

次の出力は、手動の GSLB 設定同期のステータスを表示します。


```
> sh gslb syncStatus
Displaying the status of the manual GSLB configuration synchronization:

gslb_site1[Master]:
        Getting Config:  ok
gslb_site2[Slave]:
        Syncing gslb static proximity database:  ok
        Syncing inbuilt gslb static proximity database : ok
        Getting Config:  ok
        Comparing config:  ok
        Applying changes:  ok
gslb_natsite1[Slave]:
        Syncing gslb static proximity database:  ok
        Syncing inbuilt gslb static proximity database : ok
        Getting Config:  ok
        Comparing config:  ok
        Applying changes:  ok
 Done
>
```

次の出力は、手動の GSLB 設定同期のステータス概要を示しています。

```
> sh gslb syncStatus -summary
Displaying the status summary of the manual GSLB configuration synchronization:

-------------------------------------------------------------------------------------------------------------
        Site Name                Status          Reason
-------------------------------------------------------------------------------------------------------------
        gslb_site1               Success         All Done
        gslb_site2               Failure         Error executing command on gslb site...ERROR: Connection failed
        gslb_natsite1            Success         All Done
Done
>
```

GSLB リアルタイム同期の設定出力例

次の出力は、マスターサイトのリアルタイム GSLB 構成同期のステータスを表示します。

```
1 > sh gslb syncStatus
2 Displaying the status of the real time GSLB configuration
      synchronization as master node:
3
```



```
site2[Master]:
                New GSLB configuration detected at Fri Jan 23 20:54:24
                    2020
                Fetching current configuration: Done
                Updating default.conf file: Done
site1[Slave]:
                Syncing gslb static proximity database to node site1:
                    Done
                Syncing inbuilt GSLB static proximity database to node
                    site1: Done
                Syncing ssl certificates, keys and CRLS to node site1:
                    Done
                Syncing current configuration to site1: Done
                Pulling status files from site1: Status file not
                    available yet(Sync in progress)
                Pulling status files from site1: Done
                site1 received new configuration from 10.102.217.205 in
                     file 2JNSzClRHK5+pdek6szQ3g-default-10.102.217.210.
                    conf
                Firing set gslb parameter -startConfigSync ENABLED
                    command: Done
                Fetching running GSLB Config: Done
                Comparing config: Done
                Applying changes: Done
                Firing set gslb parameter -startConfigSync DISABLED
                    command: Done
                Updating default.conf file: Done
Done
<!--NeedCopy-->
```

次の出力は、スレーブサイトのリアルタイム GSLB 構成同期のステータスを表示します。

```
> sh gslb syncStatus
Displaying the status of the real time GSLB configuration
    synchronization as slave node:

                site1 received new configuration from 10.102.217.205 in
                     file 2JNSzClRHK5+pdek6szQ3g-default-10.102.217.210.
                    conf
                Firing set gslb parameter -startConfigSync ENABLED
                    command: Done
                Fetching running GSLB Config: Done
                Comparing config: Done
                Applying changes: Done
                Firing set gslb parameter -startConfigSync DISABLED
                    command: Done
                Updating default.conf file: Done
Done
<!--NeedCopy-->
```

次の出力は、マスターサイトのリアルタイム GSLB 構成同期のステータス概要を示しています。

```
> sh gslb syncStatus -summary
```



```
Displaying the status summary of the real time GSLB configuration
    synchronization as master node:

-------------------------------------------------------------------------------

                Site Name                                       Status
                                  Reason
-------------------------------------------------------------------------------

                site2                                           Success
                                 All Done
                site1                                           Success
                                 All Done

Done
<!--NeedCopy-->
```

次の出力は、スレーブサイトのリアルタイム GSLB 構成同期のステータス概要を示しています。

```
> sh gslb syncStatus  -summary
Displaying the status summary of the real time GSLB configuration
    synchronization as slave node:

-------------------------------------------------------------------------------

                Site Name                                       Status
                                  Reason
-------------------------------------------------------------------------------

                site1                                           Success
                                 All Done

Done
<!--NeedCopy-->
```

GUI を使用して **GSLB** 同期のステータスまたは概要を表示するには

1. **[**** 設定] > [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動します。**
2. 必要に応じて、[同期サマリーの表示**]** または **[**同期ステータスの 表示] をクリックします。

GSLB 構成同期用の **SNMP** トラップ

August 15, 2023

NetScaler 12.1 ビルド 49.xx 以降、GSLB 構成を同期すると、NetScaler アプライアンスはローカルサイトとリモートサイトの両方に対して SNMP トラップを生成します。SNMP トラップは、手動同期とリアルタイム同期の両方で生成されます。

GSLB 構成を初めて同期すると、SNMP トラップが生成されます。その後の同期試行では、同期ステータスが以前の同期ステータスから変化した場合にのみ SNMP トラップが生成されます。また、SNMP トラップは、同期ステータスが以前の状態から変化したサイトに対してのみ生成されます。

たとえば、最初の GSLB 設定の同期が成功したとします。2 回目に設定を同期し、同期が再び成功した場合、ステータスは変更されないため、SNMP トラップは生成されません。ただし、3 回目の試行では、いずれかのサイトで同期が失敗すると、そのサイトだけで SNMP トラップが生成されます。

高可用性とクラスター設定では、以前の同期ステータスに関係なく、新しいノードから GSLB 構成を同期すると、アプライアンスが SNMP トラップを生成します。また、SNMP トラップオプションが以前に無効になってから有効になっている場合、以前の同期ステータスに関係なく、その時点から SNMP トラップが生成されます。

GSLB 設定同期の SNMP トラップは、次の詳細を提供します。

- SNMP トラップが送信される GSLB サイトの名前。
- GSLB 設定の同期ステータス: 成功または失敗。
- GSLB 構成同期モード： インクリメンタル同期または完全同期。
- （任意）SNMP トラップに関する詳細情報。

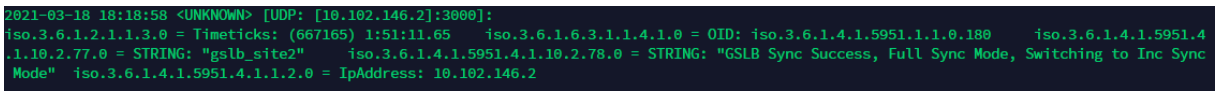
SNMP トラップは、次のシナリオで生成されます。

- GSLB サイトの GSLB 同期ステータスは、[成功] から [失敗]、逆に反転します。
- GSLB 同期モードは、増分同期から完全同期に変更され、逆に変化します。

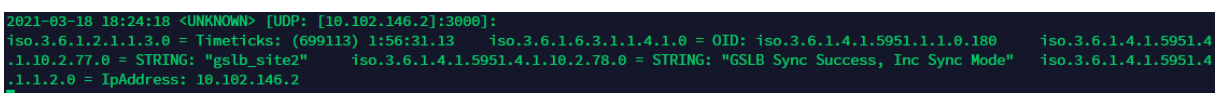
> 注:
>
> 差分同期が有効になっている場合でも、何らかの理由で GSLB サイトで完全同期が実行された場合、完全同期の理由は、トラップメッセージの「詳細情報」セクションに記載されています。たとえば、新しい GSLB サイトが GSLB 構成に追加された場合などです。

SNMP トラップメッセージの例

次の図に、gslb_site2 の SNMP トラップの例を示します。このトラップでは、完全同期モードを使用して GSLB 構成の同期が成功します。



次の図に、gslb_site2 の SNMP トラップの例を示します。この場合、インクリメンタル同期モードを使用して GSLB 構成の同期が正常になります。



次の図に、差分同期モードを使用した GSLB 設定の同期が失敗する gslb_site2 の SNMP トラップの例を示します。エラーメッセージは、同期を完了するために手動でエラーを修正する必要があることを示します。


```
2021-03-18 18:17:34 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (658753) 1:49:47.53    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180    iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"    iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Failure, Inc Sync Mode, Site is not in sync, I
ncremental config application has failed, Switching to Full Sync Mode." iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
2021-03-18 18:17:49 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (660256) 1:50:02.56    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180    iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"    iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Failure,Full Sync Mode, Site is not in sync, F
ull sync config application has failed, Please fix the errors." iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
```

次の図に、差分同期モードを使用した GSLB 設定の同期が失敗する gslb_site2 の SNMP トラップの例を示します。また、同期が失敗した理由、つまりサイトモニターがダウンしている理由も示します。

```
2021-03-18 18:21:39 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (683289) 1:53:52.89    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180    iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"    iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Failure, Inc Sync Mode, Syncing current config
uration to gslb_site2: Skipped, Site Monitor is down"    iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
```

GSLB ダッシュボード

August 15, 2023

GSLB に参加している GSLB サイトの全体的なステータスは、GSLB ダッシュボードで確認できます。

ダッシュボードから GSLB 設定にアクセスできます。ダッシュボードから GSLB 設定ウィザードを起動することもできます。さらに、ダッシュボードから同期を実行して GSLB の設定をテストできます。

GSLB ダッシュボードにアクセスするには、[設定] > [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動します。

GSLB サービスを監視する

August 15, 2023

リモートサービスを GSLB 仮想サーバーにバインドすると、GSLB サイトは、ラウンドトリップ時間と永続性情報であるネットワークメトリック情報を含むメトリック情報を交換します。

参加しているサイト間でメトリック交換接続が一時的に切断された場合、リモートサイトはダウンとマークされ、稼働している残りのサイトで負荷分散が実行されます。サイトのメトリック交換が DOWN の場合、そのサイトに属するリモートサービスも DOWN とマークされます。

NetScaler ADC アプライアンスは、MEP またはリモートサービスに明示的にバインドされているモニターを使用して、リモート GSLB サービスの状態を定期的に評価します。ローカル GSLB サービスの状態はデフォルトで MEP を使用して更新されるため、明示的なモニターをローカルサービスにバインドする必要はありません。ただし、明示的なモニターをリモートサービスにバインドすることはできます。モニタが明示的にバインドされている場合、リモートサービスの状態はメトリック交換によって制御されません。

デフォルトでは、モニターをリモート GSLB サービスにバインドすると、NetScaler アプライアンスはモニターによって報告されたサービスの状態を使用します。ただし、次の状況ではモニターを使用してサービスを評価するように NetScaler アプライアンスを構成できます。

- 常にモニターを使用してください (デフォルト設定)。
- MEP がダウンしているときはモニタを使用してください。
- リモートサービスと MEP がダウンしているときはモニターを使用してください。

上記の設定のうちの 2 番目と 3 番目の設定により、MEP が稼働しているときにアプライアンスは監視を停止できます。たとえば、階層型の GSLB 設定では、GSLB サイトは子サイトの MEP 情報を親サイトに提供します。このような中間サイトでは、サイトの実際の状態は稼働していても、ネットワークの問題が原因で子サイトの状態が DOWN と評価されることがあります。この場合、モニターを親サイトのサービスにバインドし、MEP を無効にして、リモートサービスの実際の状態を確認できます。このオプションにより、リモートサービスの状態を決定する方法を制御できます。

モニターを使用するには、まずモニターを作成してから、GSLB サービスにバインドします。

モニタートリガーの設定

常にモニター（デフォルト）を使用するように、MEP がダウンしているときにモニターを使用するように、またはリモートサービスと MEP の両方がダウンしているときにモニターを使用するように GSLB サイトを構成できます。後者の 2 つのケースでは、MEP が稼働状態に戻ると、NetScaler アプライアンスは監視を停止します。

コマンドラインインターフェイスを使用してモニタートリガーを設定するには

コマンドプロンプトで入力します。

```
set gslb site <siteName>  -triggerMonitor (ALWAYS | MEPDOWN |
   MEPDOWN_SVCDOWN)
<!--NeedCopy-->
```

例:

```
set gslb site Site-GSLB-North-America  -triggerMonitor Always
<!--NeedCopy-->
```

構成ユーティリティを使用してモニタートリガーを構成するには

1. [トラフィック管理] > [**GSLB**] > [サイト**]** に移動し、サイトをダブルクリックします。
2.「トリガーモニター」ドロップダウンリストで、モニタリングをトリガーするタイミングのオプションを選択します。

モニターの追加または削除

モニターを追加するには、タイプとポートを指定します。サービスにバインドされているモニターは削除できません。まず、モニターをサービスからバインド解除する必要があります。

コマンドラインインターフェイスを使用してモニターを追加するには

コマンドプロンプトで次のコマンドを入力してモニターを作成し、構成を確認します。

```
add lb monitor <monitorName> -type <monitorType> -destPort <portNumber>

show lb monitor <monitorName>
<!--NeedCopy-->
```

例:

```
add lb monitor monitor-HTTP-1 -type HTTP -destPort 80
show lb monitor monitor-HTTP-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してモニターを削除するには

コマンドプロンプトで入力します。

```
rm lb monitor <monitorName>
<!--NeedCopy-->
```

構成ユーティリティを使用してモニターを追加するには

[
トラフィック管理] > [負荷分散] > [モニター] に移動し、モニターを追加または削除します。

モニターを **GSLB** サービスにバインドする

モニターを作成したら、それを GSLB サービスにバインドする必要があります。モニターをサービスにバインドするときに、モニターの重みを指定できます。1 つ以上の加重モニターをバインドしたら、サービスのモニターしきい値を設定できます。このしきい値により、バインドされたモニターの重みの合計がしきい値を下回ると、サービスが停止します。

注: 構成ユーティリティでは、モニターをバインドすると同時に重みと監視しきい値の両方を設定できます。コマンドラインを使用する場合は、別のコマンドを実行してサービスの監視しきい値を設定する必要があります。

コマンドラインインターフェイスを使用してモニターを **GSLB** サービスにバインドするには

コマンドプロンプトで入力します。

```
bind monitor <name> <serviceName> [ -state (Enabled | Disabled) ] -
    weight <positiveInteger>
<!--NeedCopy-->
```

例**:**

```
bind monitor monitor-HTTP-1 service-GSLB-1 -state enabled -weight 2
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** サービスの監視しきい値を設定するには

コマンドプロンプトで入力します。

```
set gslb service <ServiceName> -monThreshold <PositiveInteger>
<!--NeedCopy-->
```

例**:**

```
set gslb service service-GSLB-1 -monThreshold 9
<!--NeedCopy-->
```

設定ユーティリティを使用してモニターを **GSLB** サービスにバインドするには

1. [トラフィック管理] > [GSLB] > [サービス] に移動します。
2. **Monitor** セクションをクリックし、モニターを GSLB サービスにバインドします。

構成ユーティリティを使用して **GSLB** サービスの監視しきい値を設定するには

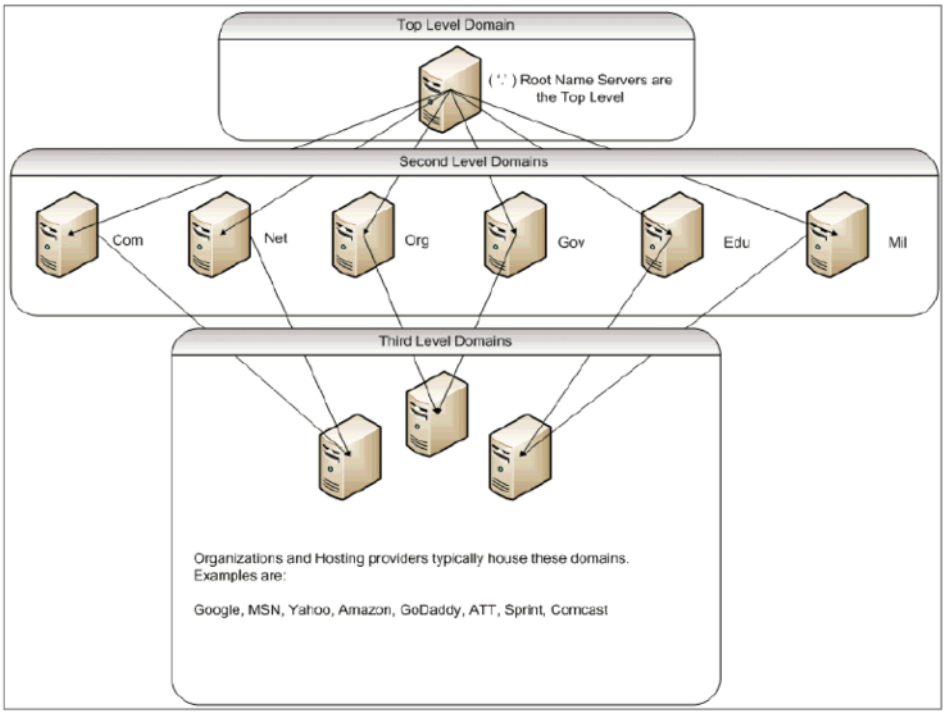
1. [トラフィック管理] > [GSLB] > [サービス] に移動します。
2.「監視しきい値」セクションをクリックし、しきい値を入力します。

ドメインネームシステムが **GSLB** をサポートする方法

August 15, 2023

ドメインネームシステム (DNS) は、クライアント/サーバーアーキテクチャを使用する分散データベースと見なされます。ネームサーバーはアーキテクチャ内のサーバーであり、リゾルバは、ネットワークを介してクエリを作成して送信するオペレーティングシステムにインストールされているライブラリルーチンであるクライアントです。

DNS の論理階層を次の図に示します。



> 注:
>
> 第 2 レベルのルートサーバーは、.com、.net、.org、.gov ドメインなどのネームサーバー委任のネームサーバーからアドレスへのマッピングを維持する責任があります。第 2 レベルドメイン内の各ドメインは、下位レベルの組織ドメインのネームサーバからアドレスへのマッピングを維持する責任があります。組織レベルでは、WWW、FTP、およびその他のサービスを提供するホストに対して、個々のホストアドレスが解決されます。

委任

現在の DNS トポロジの主な目的は、1 つの機関ですべてのアドレスレコードを維持する負担を軽減することです。これにより、組織の名前空間を特定の組織に委任できます。その後、組織はそのスペースを組織内のサブドメインにさらに委任できます。たとえば、citrix.com では、`sales.citrix.com`、`education.citrix.com`、`support.citrix.com`というサブドメインを作成できます。対応する部門は、サブドメインに対して権限のある独自のネームサーバーセットを保持し、ホスト名とアドレスマッピングの独自のセットを維持できます。すべての Citrix アドレスレコードを管理する責任は、単一の部門ではありません。各部門は、アドレスを変更したり、トポロジを変更したり、上位レベルのドメインや組織でより多くの作業を課すことはありません。

階層トポロジの利点

階層トポロジの利点には、次のようなものがあります。

- スケーラビリティ

- キャッシュ機能を各レベルでネームサーバーに追加する。DNS 要求は、特定のドメインに対して権限がないがクエリへの応答を提供できるホストによって処理され、輻輳と応答時間が短縮されます。
- キャッシングは、サーバー障害に対する冗長性と回復力ももたらします。1 つのネームサーバーに障害が発生しても、同じレコードの最近キャッシュされたコピーを持つ他のサーバーからレコードを提供できる可能性があります。

リゾルバ

リゾルバは DNS システム内のクライアントコンポーネントです。ドメイン・ネーム・スペースからの情報を必要とするホスト上で実行されているプログラムは、リゾルバを使用します。リゾルバは次の処理を行います。

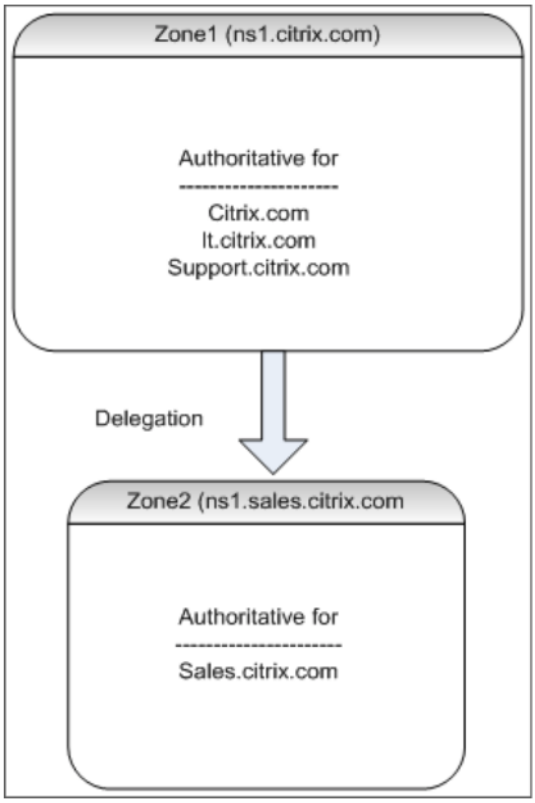
- ネームサーバを照会する。
- レスポンスの解釈（リソースレコードまたはエラーなど）。
- 情報を要求したプログラムに情報を返します。

リゾルバは、telnet、FTP、ping などのプログラムにコンパイルされるライブラリルーチンのセットです。それらは別々のプロセスではありません。リゾルバはクエリをまとめ、送信し、回答を待つことができます。また、特定の時間内に返答されない場合は、もう一度送信してください (おそらくセカンダリネームサーバに)。これらのタイプのリゾルバは、スタブ・リゾルバと呼ばれます。一部のリゾルバーには、レコードをキャッシュし、存続時間 (TTL) を尊重する機能が追加されています。Windows では、この機能は DNS クライアントサービスを通じて使用できます。「services.msc」コンソールから表示できます。

ネームサーバー

ネームサーバーは通常、ドメインネームスペース (ゾーンと呼ばれる) の特定の部分に関する完全な情報を格納します。ネームサーバーには、そのゾーンに対する権限があると言われます。また、複数のゾーンに対して権限を持つこともできます。

ドメインとゾーンの違いは微妙です。ドメインとは、サブドメインを含むエンティティの完全なセットであり、ゾーンはドメイン内の情報であり、別のネームサーバーに委任されません。ゾーンの例としては `citrix.com`、サブドメイン内の別のネームサーバーにゾーンが委任されている場合、`sales.citrix.com`は別のゾーンです。この場合、プライマリ Citrix ゾーンには `citrix.com`、`it.citrix.com`、および`support.citrix.com`を含めることができます。`sales.citrix.com`が委任されるため、`citrix.com`ネームサーバが権限を持つゾーンの一部ではありません。次の図は、2 つのゾーンを示しています。

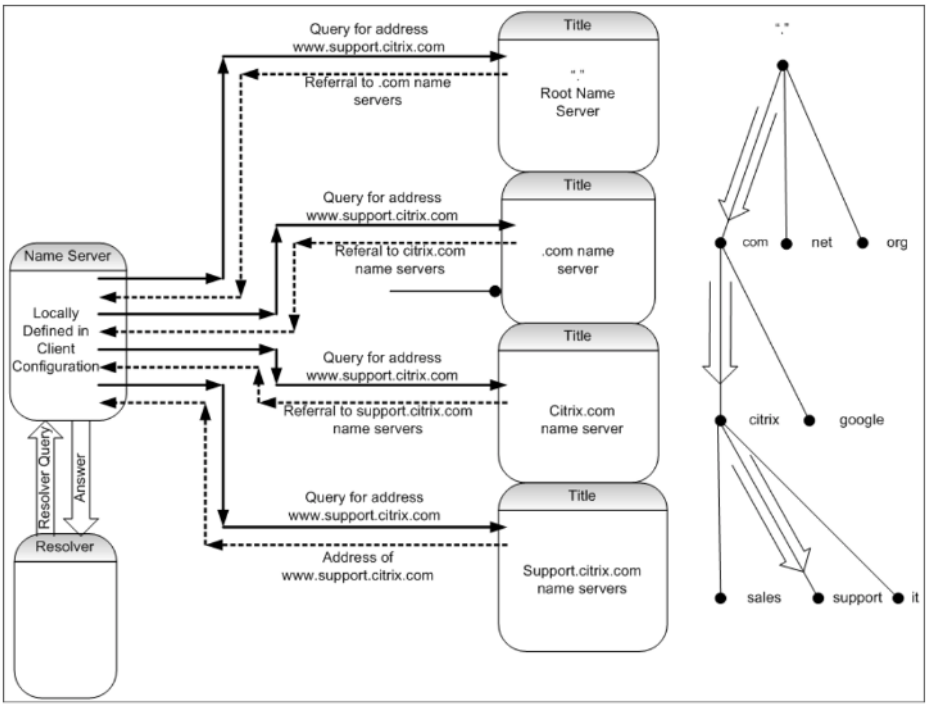


サブドメインを適切に委任するには、サブドメインの権限を別のネームサーバーに割り当てる必要があります。前の例では、ns1.citrix.comにはsales.citrix.com サブドメインに関する情報は含まれていません。代わりに、ns1.sales.citrix.com サブドメインに対して権限のあるネームサーバーへのポインタが含まれます。

ルートネームサーバーとクエリ解決

ルートネームサーバは、第 2 レベルドメインに対して権限を持つすべてのネームサーバの IP アドレスを認識します。ネームサーバーが独自のデータファイルに特定のドメインに関する情報を持っていない場合、最終的に特定のドメインに到達するために、**DNS** ツリー構造の適切なブランチを走査するためにルートサーバーに連絡するだけで済みます。これには、ツリートラバーサルで次の権限のあるネームサーバーを見つけるための複数のネームサーバーへの一連の要求が含まれます。このリクエストは、さらなる解決のために連絡する必要があります。

次の図は、トラバーサル中に要求された名前のキャッシュレコードがないと仮定した、典型的な DNS 要求を示しています。次の例では、Citrix ドメインのモックアップを使用します。



再帰クエリと非再帰クエリ

上記の例では、発生する 2 つのタイプのクエリを示します。

- 再帰クエリ: リゾルバとローカルに設定されたネームサーバー間のクエリは再帰的です。つまり、ネームサーバーはクエリを受信し、クエリが完全に応答されるか、エラーが返されるまでリゾルバに応答しません。ネームサーバがクエリへの参照を受信すると、ネームサーバが最終的に返された回答（IP アドレス）を受け取るまで、ネームサーバは参照に従います。
- 非再帰クエリ: ローカルで構成されたネームサーバーが、後続の権限のあるドメインレベルのネームサーバーに対して行うクエリは、非再帰 (または反復) です。各リクエストは、下位レベルの権限のあるサーバーへの参照、またはクエリに対する応答のいずれかで即座に応答されます（クエリされたネームサーバーがそのデータファイルまたはそのキャッシュに回答が含まれている場合）。

キャッシュ

解決プロセスには関与しており、複数のホストへの小さなリクエストが必要になる可能性がありますが、高速です。DNS 解決の速度を上げる要因の 1 つがキャッシュです。ネームサーバーは、再帰クエリを受信するたびに、最終的に特定の要求に対して適切な権限のあるサーバーに到達するために、他のサーバーと通信しなければならない場合があります。これは、将来の参照のために受け取ったすべての情報を格納します。次のクライアントが、別のホストで同じドメイン内で同様の要求を行う場合、そのドメインに対して権限のあるネームサーバーをすでに認識しており、ルートネームサーバーで起動するのではなく、そこで直接リクエストを送信できます。

キャッシュは、存在しないホストのクエリなど、否定的な応答に対しても発生する可能性があります。この場合、サーバは、ホストが存在しないことを把握するために、要求されたドメインを権限のあるネームサーバに照会してはなりません。時間を節約するために、ネームサーバーはキャッシュをチェックし、ネガティブレコードで応答します。

ネームサーバーはレコードを無期限にキャッシュしないか、IP アドレスを更新することはできません。同期の問題を回避するために、DNS 応答には存続時間 (TTL) が含まれています。このフィールドは、キャッシュがレコードを破棄し、更新されたレコードがあるかどうかを権限のあるネームサーバーで確認する前に、キャッシュがレコードを保存できる時間間隔を示します。レコードが変更されていない場合、TTL を使用すると、GSLB を実行するデバイスからの迅速な動的応答も可能になります。

リソースレコードタイプ

さまざまな RFC は、DNS リソースレコードタイプとその説明の包括的なリストを提供します。次の表に、一般的なリソースレコードタイプを示します。

リソースレコードタイプ	説明	RFC
A	ホストアドレス	RFC 1035
NS	権威あるネームサーバー	RFC 1035
MD	メールの送信先 (廃止-MX を使用)	RFC 1035
MF	メールフォワーダー (廃止-MX を使用)	RFC 1035
CNAME	エイリアスの正規名	RFC 1035
SOA	権限ゾーンの開始をマークします。	RFC 1035
WKS	よく知られているサービスの説明	RFC 1035
PTR	ドメイン名ポインタ	RFC 1035
ヒント	ホスト情報	RFC 1035
MINFO	メールボックスまたはメールリストの情報	RFC 1035
MX	メール交換	RFC 1035
TXT	テキスト文字列	RFC 1035
AAAA	IP6 アドレス	RFC 3596
SRV	サーバー選択	RFC 2782]

GSLB が DNS をサポートする方法

GSLB は、DNS クエリで送信する必要がある IP アドレスを決定するアルゴリズムとプロトコルを使用します。GSLB サイトは地理的に分散されており、NetScaler ADC アプライアンスでサービスとして実行されている各サイトには DNS 権限のあるネームサーバーがあります。関係するさまざまなサイトのすべてのネームサーバーは、同じドメインに対して権限を持ちます。各 GSLB ドメインは、委任が設定されるサブドメインです。したがって、GSLB ネームサーバーは権限があり、さまざまな負荷分散アルゴリズムのいずれかを使用して、どの IP アドレスを返すかを決定できます。

委任は、親ドメインのデータベースファイルに GSLB ドメインのネームサーバーレコードを追加し、委任に使用されるネームサーバーのアドレスレコードを追加することによって作成されます。たとえば、www.citrix.comに GSLB を使用する場合は、次のバインド SOA ファイルを使用して、リクエストをwww.citrix.comからネームサーバーに委任できます。Netscaler1 と Netscaler2。

```
################################################################################

@ IN SOA citrix.com. hostmaster.citrix.com. (
1 ; serial
3h ; refresh
1h ; retry
1w ; expire
1h ) ; negative caching TTL
IN NS ns1
IN NS ns2
IN MX 10 mail

ns1   IN A 10.10.10.10
ns2   IN A 10.10.10.20
mail  IN A 10.20.20.50

### Old Configuration if www was not delegated to a GSLB name server
www IN A 10.20.20.50

### Updated Configuration
Netscaler1 IN A xxx.xxx.xxx.xxx
Netscaler2 IN A yyy.yyy.yyy.yyy
www IN NS Netscaler1.citrix.com.
www IN NS Netscaler2.citrix.com.
###
IN MX 20 mail2
mail2 IN A 10.50.50.20
################################################################################


<!--NeedCopy-->
```

BIND を理解することは、DNS を設定するための要件ではありません。準拠する DNS サーバーの実装には、同等の委任を作成する方法があります。Microsoft DNS サーバーを委任用に構成するには、「[ゾーン委任を作成する](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-

2003/cc785881(v=ws.10)%E3%80%8D%E3%81%AE%E6%89%8B%E9%A0%86%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%97%E3%81%BE%E3%81%99%E3%80%82redirectedfrom =MSDN)。

NetScaler ADC アプライアンスの GSLB が標準の DNS サービスを使用してトラフィックを分散するのとは異なるのは、NetScaler GSLB サイトがメトリック交換プロトコル（MEP）と呼ばれる独自のプロトコルを使用してデータを交換することです。MEP を使用すると、GSLB サイトは他のすべてのサイトに関する情報を維持できます。DNS 要求を受信すると、MEP は GSLB メトリックスを考慮して、次のような情報を決定します。

- 現在の接続数が最も少ないサイト
- ラウンドトリップ時間 (RTT) に基づいてリクエストを送信した LDNS サーバーに最も近いサイト。

使用できる負荷分散アルゴリズムはいくつかありますが、GSLB は、参加サイトのメトリックに基づいてどのアドレスを送信する必要があるかをネームサーバー（NetScaler ADC アプライアンス上でホストされている）に知らせる DNS です。

GSLB が提供するその他の利点は、永続性（またはサイトアフィニティ）を維持できることです。着信 DNS クエリに対する応答をソース IP アドレスと比較して、そのアドレスが最近特定のサイトに誘導されたかどうかを判断できます。その場合は、クライアントセッションが確実に維持されるように、同じアドレスが DNS 応答に送信されます。

別の形式の永続性は、HTTP リダイレクト、または HTTP プロキシを使用してサイトレベルで取得されます。これらの形式の永続性は、DNS 応答が発生した後に発生します。したがって、リクエストを別の参加サイトに誘導する Cookie を含むサイトで HTTP リクエストを受け取った場合、リダイレクトで応答するか、リクエストを適切なサイトにプロキシすることができます。

メトリック交換プロトコル

メトリック交換プロトコル (MEP) は、GSLB 計算で使用されるデータをサイト間で共有するために使用されます。MEP 接続を使用して、3 種類のデータを交換します。これらの接続は、TCP ポート 3011 を介してセキュアである必要はなく、TCP ポート 3009 経由の SSL を使用してセキュアにすることもできます。

以下の 3 種類のデータが交換され、独自の間隔と交換方法がある。

- サイトメトリック交換**:** これはポーリング交換モデルです。たとえば、site1 に site2 サービスの構成がある場合、第 2 サイト 1 は site2 に GSLB サービスのステータスを要求します。Site2 は、状態およびその他のロードの詳細で応答します。
- ネットワークメトリック交換：これは、動的近接負荷分散アルゴリズムで使用される LDNS RTT 情報交換です。これはプッシュ交換モデルです。5 秒ごとに、各サイトはそのデータを他の参加サイトにプッシュします。
- 永続性交換**:** これは SOURCEIP 永続性交換用です。これはプッシュ交換モデルでもあります。5 秒ごとに、各サイトはそのデータを他の参加サイトにプッシュします。

デフォルトでは、サイトサービスはポーリング情報のみに基づいて MEP を介して監視されます。モニタ間隔に基づいてモニタをバインドすると、状態は更新され、それに応じてモニタリング間隔を設定することで更新の頻度を制御できます。

GSLB サービスの優先順位

October 25, 2023

サービスの優先順位機能を使用すると、負荷分散の選択プリファレンスに基づいて、サービスまたはサービスグループの順序に優先順位を付けることができます。次の操作を行うと、優先順位を設定できます。

- サービスを GSLB 仮想サーバーにバインドします。
- サービスグループを GSLB 仮想サーバーにバインドします。
- サービスグループメンバーを GSLB サービスグループにバインドします。

現在、サービスの優先順位は次の方法で構成できます。ただし、これらのアプローチには次の制限があります。

- バックアップ仮想サーバーチェーンの構成: 構成行の数が多いため、各仮想サーバーのすべての GSLB サービスの状態を確認するには、showコマンドを複数回実行する必要があります。
- 優先ロケーションの設定: すべてのアプリケーションエンドポイントに対してロケーションエントリを作成する必要があります。

サービスの優先順位付け機能は、設定コマンドの数を減らして前述の制限に対処し、すべての GSLB サービスの IP アドレスのロケーション表現を必要とせずに、優先ロケーション設定を実現するのに役立ちます。

GSLB サービスの優先順位を設定する

GSLB サービスの優先順位を設定するために、-order <number>パラメータが bind コマンドに追加されます。

> 注:
>
> 順序番号が小さいほど優先度が高くなります。

コマンド:

```
bind gslb vserver <vservername> -servicename/servicegroupname <
servicename/servicegroupname> -order <number>
```

たとえば、GSLB 仮想サーバー (gv1) にバインドされた一連のサービスを考えてみます。
- order <number>パラメータを使用すると、次のように、サービスの選択順序に優先順位を付けることができます。

- gv1 にバインドされた 1 (s1, s2) をセット—オーダ 1
- gv1 にバインドされたセット 2 (s3, s4) —オーダー 2
- gv1 にバインドされた 3 (s5, s6) をセット—オーダ 3

サービスを gv1 にバインドし、gv1 がクライアントトラフィックを受信すると、サービスの選択順序は次のようになります。

- 仮想サーバ (gv1) は、セット 1 (s1 と s2) の順序番号が 1 のサービスを選択します。このセットには最小の順序番号が割り当てられているためです。既定では、最小の順序番号が優先されます。
- セット 1 のすべてのサービスが DOWN の場合、gv1 は順序番号 2 のセット 2 (s3 と s4) を選択します。
- セット 1 とセット 2 のすべてのサービスがダウンしている場合、gv1 は順序番号 3 のセット 3 (s5 と s6) を選択します。

CLI を使用して **GSLB** サービスの優先順位を設定する

GSLB サービスの優先順位を構成するには、コマンドプロンプトで次のコマンドを入力します。

1. GSLB サイトを追加します。

```
add gslb site site1 1.1.1.1
add gslb site site2 1.1.1.2
```

2. GSLB 仮想サーバーを追加します。

```
add gslb vserver gv1 HTTP
```

3. GSLB サービスを追加します。

```
add gslb service gsvc1 1.1.1.3 http 80 -sitename site1
add gslb service gsvc2 1.1.1.4 http 80 -sitename site2
add gslb service gsvc3 1.1.1.5 http 80 -sitename site1
add gslb service gsvc4 1.1.1.6 http 80 -sitename site2
add gslb service gsvc5 1.1.1.7 http 80 -sitename site1
add gslb service gsvc6 1.1.1.8 http 80 -sitename site2
```

4. 注文番号を設定し、サービスを GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gv1 gsvc1 -order 1
bind gslb vserver gv1 gsvc2 -order 1
bind gslb vserver gv1 gsvc3 -order 2
bind gslb vserver gv1 gsvc4 -order 2
bind gslb vserver gv1 gsvc5 -order 3
bind gslb vserver gv1 gsvc6 -order 3
```

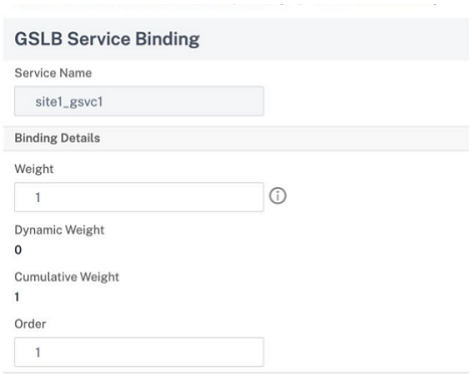

GUI を使用した GSLB サービスの優先順位の設定

前提条件:

- GSLB サイトが作成されました。
- GSLB 仮想サーバーが作成されました。
- GSLB サービスを作成していること。

GSLB サービスの優先順位を設定し、GSLB 仮想サーバーにバインドするには、次の手順を実行します。

1. [トラフィック管理] **> [GSLB] > [**仮想サーバー] に移動し、GSLB 仮想サーバーをダブルクリックします。
2. [**GSLB** 仮想サーバー] の [**GSLB** サービスと **GSLB** サービスグループバインディング] セクションで、[**GSLB** 仮想サーバーから **GSLB** サービスへのバインディング] をクリックします。
3. [**GSLB** サービスと **GSLB** サービスグループバインディング**]** ダイアログで、**[**バインドの追加 **] をクリックします。
4. [**GSLB** サービスバインディング] ダイアログボックスで、サービスを選択します。
5. [順序] フィールドに番号を入力して、サービスの優先順位を設定します。



6. 「バインド」をクリックします。
7. ステップ 1 ～6 を繰り返して、サービスごとに異なる優先順位番号を設定します。

LB ポリシーコマンドを使用して GSLB サービスの優先順位を設定する

既定では、最小の順序番号が優先されます。ただし、新しい LB action コマンドと policy コマンドを使用すると、このデフォルトの動作を延期できます。着信クライアントトラフィックまたはクライアントデータに基づいて、サービスの選択順序を設定できます。

たとえば、GSLB 仮想サーバー (gv1) にバインドされた一連のサービスを考えてみます。- `order <number>`パラメータを使用して、サービスの優先順位を次のように設定しました。

- gv1 にバインドされた 1 (s1, s2) をセット—オーダ 1
- gv1 にバインドされたセット 2 (s3, s4) —オーダー 2

- gv1 にバインドされた 3 (s5, s6) をセット—オーダ 3

既定では、最小の順序番号が優先されます。したがって、set 1、set2、set3 のサービスのデフォルトの優先順位は、それぞれ 1、2、3 です。ただし、特定のクライアントトラフィックについては、優先順位を 3、1、2 に変更する必要があります。これを実現するには、LB ポリシーを追加して gv1 にバインドします。

LB ポリシーコマンドは、ルールとアクションの 2 つの要素で構成されます。ルールはアクションに関連付けられ、リクエストがルールに一致した場合に実行されるアクションです。

> 注:
>
> LB ポリシーコマンドは、LB と GSLB の両方の構成に共通であり、NetScaler ADC アプライアンスによって処理される要求に適用されます。

LB アクション

** 表現:**

```
add lb action <name> <type> <string>
```

** 例:**

```
add lb action act1 -type SELECTIONORDER -value 3 2 1
```

パラメーター:

- name: アクションの名前。
- type: アクションのタイプ。
- string: 指定したアクションの値。

LB ポリシー

** 表現:**

```
add lb policy <name> <rule> <action> <undefaction>
```

** 例:**

```
add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1
```

パラメーター:

- name: ポリシーの名前。
- rule: ルールは 1 つ以上の式で構成されます。ルールはアクションに関連付けられ、リクエストがルールに一致した場合に実行されるアクションです。
- action: DROP、NOLBACTION、およびリセットがサポートされています。

- undefaction: NetScaler ADC アプライアンスは、要求がポリシーと一致しない場合、未定義イベント（UNDEF イベント）を生成します。
  set lb param -undefAction <action>コマンドを使用して、未定義のアクションを設定できます。これらのアクションは、DROP、NOLBACTION、RESET といった未定義のイベントに割り当てることができます。

次のように LB アクション、LB ポリシーを追加し、そのポリシーを GSLB 仮想サーバー (gv1) にバインドする例を考えてみましょう。

```
add lb action act1 -type SELECTIONORDER -value 3 1 2

add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1

bind gslb vserver gv1 -policyName pol1 -priority 20 - gotoPriorityExpression
 END -type REQUEST
```

このルールは、IP アドレス、8.8.8.8と一致するクライアントトラフィックを選択し、そのトラフィックを gv1 に送信します。LB アクションタイプ (SELECTIONORDER) は、サービスの選択順序を定義します。LB ポリシーを gv1 にバインドした後、gv1 が IP アドレス8.8.8.8からクライアントトラフィックを受信すると、サービスは次の順序で選択されます。

1. 仮想サーバ (gv1) は、セット 3 (s5 と s6) のサービスを優先順位 3 で選択します。
2. セット 3 のすべてのサービスが DOWN の場合、gv1 は優先順位が 2 のセット 1 (s1 と s2) を選択します。
3. セット 3 とセット 2 のすべてのサービスがダウンしている場合、gv1 は順序 1 のセット 1 (s1 と s2) を選択します。

CLI を使用して LB ポリシーコマンドを使用して GSLB サービスの優先順位を設定する

LB ポリシーコマンドを使用して GSLB サービスの優先順位を構成するには、コマンドプロンプトで次のコマンドを入力します。

1. LB アクションを追加します。

   ```
   add lb action act1 -type SELECTIONORDER -value 3 1 2
   ```

2. LB ポリシーを追加します。

   ```
   add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1
   ```

3. GSLB サイトを追加します。

   ```
   add gslb site site1 1.1.1.1

   add gslb site site2 1.1.1.2
   ```

4. GSLB 仮想サーバーを追加します。

   ```
   add gslb vserver gv1 HTTP
   ```


5. LB ポリシーを GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gv1 -policyName pol1 -priority 20 - gotoPriorityExpressi
 END -type REQUEST
```

6. GSLB サービスを追加します。

```
add gslb service gsvc1 1.1.1.3 http 80 -sitename site1
add gslb service gsvc2 1.1.1.4 http 80 -sitename site2
add gslb service gsvc3 1.1.1.5 http 80 -sitename site1
add gslb service gsvc4 1.1.1.6 http 80 -sitename site2
add gslb service gsvc5 1.1.1.7 http 80 -sitename site1
add gslb service gsvc6 1.1.1.8 http 80 -sitename site2
```

7. 順序を設定し、サービスを GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gv1 gsvc1 -order 1
bind gslb vserver gv1 gsvc2 -order 1
bind gslb vserver gv1 gsvc3 -order 2
bind gslb vserver gv1 gsvc4 -order 2
bind gslb vserver gv1 gsvc5 -order 3
bind gslb vserver gv1 gsvc6 -order 3
```

GUI を使用して **LB** ポリシーコマンドを使用して **GSLB** サービスの優先順位を設定する

前提条件:

- GSLB サイトが作成されました。
- GSLB 仮想サーバーが作成されました。
- サービスを作成しました。

ステップ **1-LB** アクションを作成します。

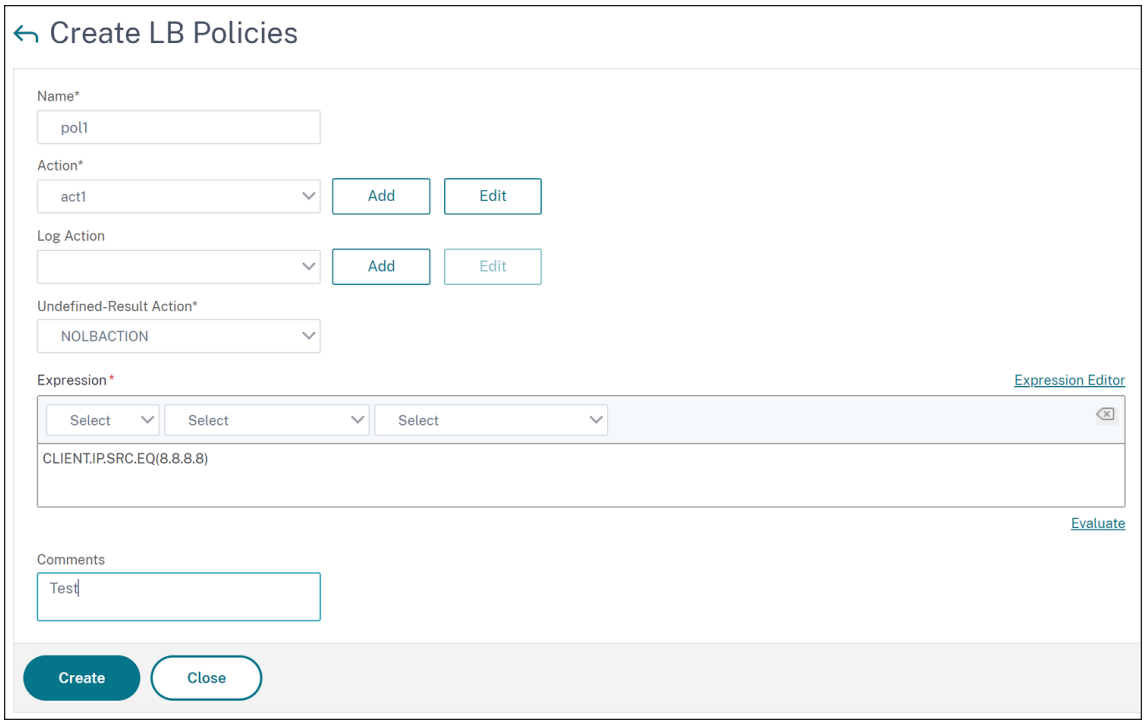
1. **AppExpert > LB >** アクションに移動します。
2. [**LB** アクション] で、[追加] をクリックします。
3. [**Create LB Actions**] ダイアログボックスで、次のパラメータの値を指定します。
    - **[**アクション名**]**
    - 種類
    - 値

> 注:
>
> **Value** フィールドの数字はスペースで区切られます。

4. [作成] をクリックします。

ステップ **2-LB** ポリシーを作成します。

1. **AppExpert > LB >** ポリシーに移動します。
2. [**LB** ポリシー] で [追加] をクリックします。
3. [**Create LB Policies**] ダイアログボックスで、次のパラメータの値を指定します。
   - 名前:pol1
   - アクション:act1
   - 未定義の結果アクション:NOLBACTION
   - 式:CLIENT.IP.SRC.EQ(8.8.8.8)



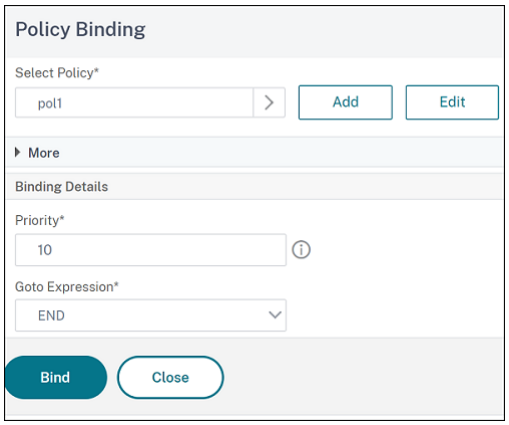
4. 「作成」をクリックします。

ステップ **3-LB** ポリシーを **GSLB** 仮想サーバーにバインドします。

1. [トラフィック管理] **> [GSLB] > [**仮想サーバー] に移動し、GSLB 仮想サーバーをダブルクリックします。
2. **GSLB** 仮想サーバーの [詳細設定] セクションで、[ポリシー] をクリックします。
3. [ポリシー] セクションで、[**GSLB** 仮想サーバー **LB** ポリシーバインド] をクリックします。

4. [ポリシーバインド] ダイアログボックスで、次のパラメータの値を指定します。

   - ポリシーを選択:pol 1
   - 優先度:10
   - **Goto** 表現: 終了



5. [**Bind**] をクリックします。

ステップ **4-GSLB** サービスの優先順位を設定します。

GSLB の優先順位を設定するには、「**GUI** を使用した **GSLB** サービスの優先順位の設定」の手順を参照してください。

サービスのパーシスタンス設定

サービスに永続性が設定されている場合、デフォルトでは常に永続性が優先されます。

たとえば、パーシステンスが設定され、優先順位が 1 のサービスを考えてみます。優先順位が 0 のサービスが UP の場合、優先順位が 1 のサービスが常に優先されます。

ただし、次の CLI コマンドを使用すると、このデフォルトの動作を上書きできます。

```
set gslb param -overridePersistencyforOrder<YES/NO>
```

次の例を考えてみましょう。

一連のサービスが、次の優先順位で GSLB 仮想サーバー (gv1) にバインドされます。

- gv1 にバインドされた 1 (s1, s2) をセット—オーダ 1
- gv1 にバインドされたセット 2 (s3, s4) —オーダー 2

永続性を上書きするには、コマンドプロンプトで次のコマンドを入力します。

```
set gslb parameter -overridePersistencyforOrder YES
```

set 1 (永続性を持つサービスが構成されている) が DOWN の場合、set 2 のサービスが set 1 のサービスが UP になるまで、すべての要求を処理します。優先度 2 の持続性エントリが作成されます。

しばらくすると、set 1 のサービスが起動したと仮定します。これで、set 1 と set 2 の両方のサービスがリクエストを処理するために UP になりました。このシナリオでは、高次のサービスが稼働すると、新しい負荷分散の決定が下されます。パーシステンスエントリは、新しい負荷分散エントリで上書きされます。

優先度切り替え

優先度切り替え機能を使用すると、優先順位の高いサービスのバージョンアップグレード中に、すべてのトラフィックを優先度の低いサービスに切り替えることができます。以下のコマンドを使用して、優先度を切り替えることができます。

- `set gslb vserver -toggleorder <Ascending/Descending>`
- `set gslb vserver v1 -orderthreshold 80`

たとえば、次の優先順位を持つ 2 つのサービスがあるとします。

- Service 1- order 0
- サービス 2 —注文 1

デフォルトでは、サービス 1 がすべてのトラフィックを処理します。サービス 1 をアップグレードする必要がある場合は、トラフィックをサービス 2 に再ルーティングする必要があります。

コマンドプロンプトで次のコマンドを入力して、優先度を切り替えます。

```
set gslb vserver -toggleorder Descending
```

既定では、0 の方が優先度が高くなります。ただし、優先度を切り替えた後は、1 の方が優先度が高いと見なされます。サービスにパーシステンスエントリが存在する場合、パーシステンスプリファレンスの動作は、「サービスのパーシスタンス設定」セクションで説明されているとおりになります。

GSLB 展開のアップグレードに関する推奨事項

March 20, 2024

このセクションでは、さまざまな GSLB セットアップで GSLB ノードをアップグレードする必要がある順序に関する推奨事項について説明します。また、いくつかの FAQ にも対処しています。

注： GSLB の同期が開始される NetScaler アプライアンスは、「メインサイト」と呼ばれ、構成がコピーされる GSLB サイトは「下位サイト」と呼ばれます。

アップグレードプロセスを開始する前に、次のトピックで説明されている前提条件をお読みください。

- はじめに
- 高可用性ペアをアップグレードします。
- クラスターをアップグレードします。

GSLB セットアップをアップグレードするときの注意点

- HA セットアップでは、まず下位サイトをアップグレードし、次にメインサイトをアップグレードします。
- HA セットアップでは、古いビルドプライマリノードから新しいビルドセカンダリノードにサービス状態が伝播しないことがあります。ただし、ビルドのバージョンが異なるが HA バージョンが同じ場合、サービスの状態は引き続き伝播する可能性があります。
- GSLB がクラスタ内で構成されている場合は、まず所有者以外のノードをアップグレードしてから、所有者ノードをアップグレードします。クラスタ内に 1 つのサイトまたは複数のサイトがある場合は、各サイトで同じアップグレード手順に従います。
- 新しい GSLB 機能は、すべてのノードを新しいビルドにアップグレードした後にのみ有効にします。
- すべての GSLB ノードを最新のビルドにアップグレードします。GSLB ノードの一部が古いバージョンを使用し、一部の GSLB ノードが新しいバージョンにアップグレードされている場合、使用可能な機能に機能的影響はありません。

よくある質問

- インスタンスが異なるソフトウェアバージョンを実行する場合、**GSLB** サービスの状態は伝播されますか。

  GSLB MEP は、インスタンスが異なるバージョンで実行され、GSLB サービスの状態が GSLB サイト間で伝播される場合に機能します。アップグレード後にインスタンスが異なるバージョンを実行する場合、MEP 通信には影響しません。

- アップグレード中に構成の変更を行うことを推奨しますか。

  GSLB セットアップでは、メインサイトをアップグレードするときに、他の GSLB ノードで設定を変更することはお勧めしません。

関連情報

以下のリソースは、NetScaler コンソールを使用して NetScaler インスタンスをアップグレードする方法に関する情報を提供します:

- NetScaler コンソールサービスが NetScaler のアップグレードを容易にする 10 の方法
- NetScaler コンソールサービスを使用して NetScaler インスタンスをアップグレードする
- NetScaler コンソールソフトウェアを使用して NetScaler インスタンスをアップグレードする

ユースケース：ドメイン名ベースの自動スケールサービスグループの展開

August 15, 2023

> ヒント
>
> GSLB サービスグループの詳細については、「GSLB サービスグループの設定」を参照してください。

導入シナリオ

2 つのデータセンターが 2 つの AWS リージョン (1 つはシドニー、もう 1 つはノースバージニア) にデプロイされています。別のデータセンターが Azure にデプロイされています。各 AWS リージョンの AWS ELB は、アプリケーションサーバーの負荷分散に使用されます。ALB は Azure でアプリケーションサーバーの負荷分散に使用されます。NetScaler アプライアンスは、GSLB ドメイン名ベースの自動スケールサービスグループを使用して、ELB および ALB 用の GSLB 用に構成されています。

> 重要
>
> AWS で必要なセキュリティグループを設定し、GSLB インスタンスにアタッチする必要があります。セキュリティグループのインバウンドルールとアウトバウンドルールでポート 53 が許可されている必要があります。また、MEP 通信用のポート（セキュア MEP 設定によっては 3009 または 3011）が開いている必要があります。アプリケーションを監視するには、対応するポートをセキュリティグループのアウトバウンドルールで許可する必要があります。

上記のデプロイシナリオの設定手順と対応する CLI コマンドは次のとおりです。

1. データセンター (GSLB サイトに代表される) を作成します。

   ```
   add gslb site aws-sydney 192.0.2.2

   add gslb site aws-nvirginia 198.51.100.111

   add gslb site alb-southindia 203.0.113.6
   ```

2. GSLB ノードが追加される DNS ゲートウェイ IP アドレスを含むネームサーバーを追加します。これはすべてのデータセンターで行う必要があります。

   ```
   add dns nameServer 8.8.8.8
   ```

3. ELB と ALB 用のサーバーを追加します。

   ```
   add server aws-sydney_server lb-sydney-1052691850.ap-southeast-2.
   elb.amazonaws.com

   add server aws-nvirginia_server LB-nvirginia-860559595.us-east-1.
   elb.amazonaws.com
   ```



```
add server alb-southindia_server alb.southindia.cloudapp.azure.
com
```

4. ELB と ALB ごとに GSLB autoscale サービスグループを追加し、各サーバーをそれぞれのサービスグループにバインドします。

```
add gslb serviceGroup aws-nvirginia_sg HTTP -autoScale DNS -
siteName aws-nvirginia

add gslb serviceGroup aws-sydney_sg HTTP -autoScale DNS -siteName
 aws-sydney

add gslb serviceGroup alb-southindia_sg HTTP -autoScale DNS -
siteName alb-southindia

bind gslb serviceGroup aws-nvirginia_sg aws-nvirginia_server 80

bind gslb serviceGroup aws-sydney_sg aws-sydney_server 80

bind gslb serviceGroup alb-southindia_sg alb-southindia_server 80
```

5. GSLB 仮想サーバーを追加し、アプリケーションドメインとサービスグループをこの仮想サーバーにバインドします。

```
add gslb vserver gv1 HTTP

bind gslb vserver gv1 -serviceGroupName aws-nvirginia_sg

bind gslb vserver gv1 -serviceGroupName aws-sydney_sg

bind gslb vserver gv1 -serviceGroupName alb-southindia_sg
```

ユースケース:IP アドレスベースの GSLB サービスグループの展開

August 15, 2023

> ヒント
>
> GSLB サービスグループの詳細については、GSLB サービスグループの設定を参照してください。

導入シナリオ

同じアプリケーションサーバー上で複数のアプリケーションがホストされている場合、GSLB はこれらのアプリケーションを調べて、アプリケーションが応答しているかどうかを確認する必要があります。アプリケーションが応答しない場合は、アプリケーションが UP のサーバーにユーザーを誘導する必要があります。また、アプリケーションの 1 つが DOWN の場合、他のアプリケーションが UP であるため、サーバーに DOWN とマークしないでください。

次の例では、複数のアプリケーション（HTTPS）が各 GSLB サイトの 1 つのサーバーでホストされているため、これらのアプリケーションはすべてそれぞれのサイトの 1 つの IP アドレスに変換されます。

GSLB サービスグループを使用すると、IP アドレスとポートを持つ同じサーバーを複数のサービスグループにバインドできます。各サービスグループは異なるアプリケーションを表します。

アプリケーション固有のモニターはサービスグループにバインドされ、アプリケーションがダウンしている場合はサービスグループが DOWN とマークされます。したがって、アプリケーションが DOWN のたびに、そのアプリケーションだけがセットアップから取り出され、サーバーからは取り出されません。

```
```
add gslb serviceGroup app1_site1 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s1

add gslb serviceGroup app2_site1 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s1

add gslb serviceGroup app1_site2 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s2

add gslb serviceGroup app2_site2 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s2

add lb monitor http_app2 HTTP -respCode 200 -httpRequest "GET /testsite
    /app2.html"

add lb monitor http_app1 HTTP -respCode 200 -httpRequest "GET /testsite
    /app1.html"

bind gslb serviceGroup app1_site1 192.0.2.140 80

bind gslb serviceGroup app1_site1 -monitorName http_app1

bind gslb serviceGroup app2_site1 192.0.2.140 80

bind gslb serviceGroup app2_site1 -monitorName http_app2

bind gslb serviceGroup app1_site2 192.0.2.142 80

bind gslb serviceGroup app1_site2 -monitorName http_app1

bind gslb serviceGroup app2_site2 192.0.2.142 80

bind gslb serviceGroup app2_site2 -monitorName http_app2
<!--NeedCopy--> ```
```

 
```

ハウツー記事

August 15, 2023

GSLB のハウツー記事には、GSLB 構成のカスタマイズ、永続的な接続の構成、災害復旧など、重要な GSLB 構成の一部に関する情報が含まれています。

GSLB 設定のカスタマイズ

持続的接続の構成

クライアント接続の管理

GSLB の近接度の設定

GSLB セットアップの障害からの保護

災害復旧のための GSLB の設定

優先ロケーションの設定による静的近接動作の無効化

コンテンツスイッチングを使用した GSLB サービス選択の設定

NAPTR レコードを使用した DNS クエリのグローバルサーバー負荷分散の設定

グローバルサーバー負荷分散のための EDNS0 クライアントサブネットオプションの使用

メトリック交換プロトコルを使用した完全な親子設定の例

GSLB 構成をカスタマイズする

August 15, 2023

基本的な GSLB 設定が動作可能になったら、GSLB サービスの帯域幅を変更したり、CNAME ベースの GSLB サービス、静的近接性、動的 RTT、永続接続、またはサービスの動的重みを設定したり、GSLB メソッドを変更したりして、カスタマイズできます。

GSLB サービスのモニタリングを設定してその状態を判断することもできます。

これらの設定は、ネットワークの展開と、サーバーに接続する予定のクライアントの種類によって異なります。

GSLB サービスの最大接続数または最大帯域幅の変更

仮想サーバーを表す GSLB サービスの最大クライアント数または最大帯域幅を設定することで、負荷分散またはコンテンツスイッチング仮想サーバーに同時に接続できる新しいクライアントの数を制限できます。

コマンドラインインターフェイスを使用して **GSLB** サービスの最大クライアント数または帯域幅を変更するには

コマンドプロンプトで次のコマンドを入力して、GSLB サービスの最大クライアント接続数または最大帯域幅を変更し、構成を確認します。

```
set gslb service <serviceName> [-maxClients <positive_integer>] [-
    maxBandwidth <positive_integer>]
show gslb service <serviceName>
<!--NeedCopy-->
```

例**:**

```
set glsb service Service-GSLB-1  -maxBandwidth 100  -maxClients 100
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

設定ユーティリティを使用して **GSLB** サービスの最大クライアント数または帯域幅を変更するには

1. [トラフィック管理] > [**GSLB**] > [サービス] に移動し、サービスをダブルクリックします。
2. 「その他の設定」セクションをクリックして、次のパラメータを設定します。
   - 最大クライアント数—最大クライアント数
   - 最大帯域幅—最大帯域幅

CNAME ベースの **GSLB** サービスの作成

GSLB サービスを設定するには、サーバーの IP アドレスまたはサーバーの正規名を使用できます。1 つの IP アドレスから複数のサービス (FTP サーバーや Web サーバーなど) を 1 つの IP アドレスから実行したり、同じ物理ホスト上の同じポートで名前が異なる複数の HTTP サービスを実行したりする場合は、サービスに正規名 (CNAME) を使用できます。

たとえば、DNS に ftp.example.com と www.example.com という 2 つのエントリを設定して、同じドメイン (example.com) 上の FTP サービスと HTTP サービスのエントリを設定できます。CNAME ベースの GSLB サービスは、マルチレベルのドメインリゾルバー設定やマルチレベルのドメイン負荷分散に役立ちます。CNAME ベースの GSLB サービスを設定することも、物理サーバーの IP アドレスが変更される可能性がある場合に役立ちます。

GSLB ドメインに CNAME ベースの GSLB サービスを構成すると、GSLB ドメインにクエリが送信されると、NetScaler アプライアンスは IP アドレスの代わりに CNAME を提供します。この CNAME レコードの A レコードが設定されていない場合、クライアントは CNAME ドメインに IP アドレスを問い合わせる必要があります。この CNAME レコードの A レコードが構成されている場合、NetScaler アプライアンスは対応する A レコード（IP アドレス）を CNAME に提供します。NetScaler アプライアンスは、GSLB 方式で決定された DNS クエリの最終解決を処理します。CNAME レコードは、別の NetScaler アプライアンスまたはサードパーティシステムで管理できます。

IP アドレスベースの GSLB サービスでは、サービスの状態は、それが表すサーバーの状態によって決まります。ただし、CNAME ベースの GSLB サービスの状態はデフォルトで UP に設定されています。仮想サーバーの IP (VIP) アドレスやメトリック交換プロトコル (MEP) は状態の判断には使用されません。デスクトップベースのモニターが CNAME ベースの GSLB サービスにバインドされている場合、サービスの状態はモニターのプローブの結果に基づいて決定されます。

CNAME ベースの GSLB サービスは、DNS レコードタイプが CNAME の GSLB 仮想サーバーにのみバインドできます。また、NetScaler アプライアンスには、特定の CNAME エントリを持つ GSLB サービスを 1 つまで含めることができます。

CNAME ベースの GSLB サービスでサポートされている機能の一部を以下に示します。

- GSLB ポリシーベースのサイトアフィニティがサポートされており、CNAME を推奨ロケーションとしています。
- ソース IP パーシステンスはサポートされています。持続性エントリには、選択したサービスの IP アドレスとポートの代わりに CNAME 情報が含まれます。

CNAME ベースの GSLB サービスの制限は次のとおりです。

- CNAME が参照するサービスはサードパーティのどの場所にも存在する可能性があるため、サイト永続性はサポートされていません。
- 1 つのドメインに複数の CNAME エントリを設定できないため、複数の IP アドレス応答はサポートされていません。
- サポートされている負荷分散方法は、ソース IP ハッシュとラウンドロビンだけです。静的近接方式はサポートされていません。CNAME は IP アドレスに関連付けられておらず、静的近接性は IP アドレスに基づいてのみ維持できるためです。

注:CNAME ベースの GSLB サービスをバインドする GSLB 仮想サーバーで、エンプティ・ダウン・レスポンス機能を有効にする必要があります。Empty-Down-Response 機能を有効にすると、GSLB 仮想サーバーがダウンまたは無効になっているときに、この仮想サーバーにバインドされたドメインの DNS クエリへの応答には、エラーコードの代わりに、IP アドレスのない空のレコードが含まれます。

コマンドラインインターフェイスを使用して **CNAME** ベースの **GSLB** サービスを作成するには

コマンドプロンプトで入力します。

```
add gslb service <serviceName> -cnameEntry <string> -siteName <string>
<!--NeedCopy-->
```

例**:**

```
add gslb service Service-GSLB-1 -cnameEntry transport.mycompany.com -
    siteName Site-GSLB-East-Coast
add gslb service Service-GSLB-2 -cnameEntry finance.mycompany.com -
    siteName Site-GSLB-West-Coast
<!--NeedCopy-->
```


設定ユーティリティを使用して **CNAME** ベースの **GSLB** サービスを作成するには

1. **[**** トラフィック管理] > [**GSLB**] > [サービス] に移動します。**
2. サービスを作成し、「タイプ」を「正規名ベース」に設定します。

GSLB での移行アウトオブサービス状態 **(TROFS)** の設定

サービスがバインドされている GSLB 仮想サーバーで永続性を設定すると、サービスは無効になった後もクライアントからのリクエストを処理し続け、永続性を尊重するためだけに新しいリクエストや接続を受け入れます。グレースフルシャットダウン期間と呼ばれる設定期間が経過すると、新しい要求や接続はサービスに送信されず、既存の接続はすべて閉じられます。

サービスを無効にする場合、delay 引数を使用してグレースフルシャットダウン期間を秒単位で指定できます。正常なシャットダウン期間中、サービスが仮想サーバーにバインドされている場合、その状態は Out of Service と表示されます。

サービスの動的ウェイトの設定

一般的なネットワークには、他のサーバーよりもトラフィック容量が大きいサーバーがあります。ただし、通常の負荷分散構成では、サービスが異なればサーバーの容量が異なる場合でも、負荷はすべてのサービスに均等に分散されます。

GSLB リソースを最適化するために、GSLB 仮想サーバーで動的ウェイトを設定できます。動的重み付けは、仮想サーバーにバインドされたサービスの総数、または仮想サーバーにバインドされた個々のサービスの重みの合計に基づいて設定できます。次に、サービスに設定された重みに基づいてトラフィックが分散されます。

GSLB 仮想サーバーで動的ウェイトが設定されている場合、リクエストは負荷分散方法、GSLB サービスのウェイト、および動的ウェイトに従って分散されます。GSLB サービスの重みと動的重みの積を累積重量と呼びます。そのため、GSLB 仮想サーバーで動的ウェイトが設定されている場合、リクエストは負荷分散方式と累積ウェイトに基づいて分散されます。

仮想サーバーの動的ウェイトが無効になっている場合、数値は 1 に設定されます。これにより、累積ウェイトは常にゼロ以外の整数になります。

動的加重は、負荷分散仮想サーバーにバインドされているアクティブなサービスの総数、またはサービスに割り当てられた重みに基づいて設定できます。

ドメイン用に 2 つの GSLB サイトが構成され、各サイトにクライアントにサービスを提供できる 2 つのサービスがある構成を考えてみましょう。どちらかのサイトのサービスがダウンした場合、そのサイトのもう一方のサーバーは、もう一方のサイトのサービスの 2 倍のトラフィックを処理する必要があります。動的重みがアクティブなサービスの数に基づいている場合、両方のサービスがアクティブなサイトは、1 つのサービスが停止しているサイトの重みが 2 倍になるため、受信するトラフィックは 2 倍になります。

別の方法として、1 番目のサイトのサービスが 2 番目のサイトのサーバーの 2 倍の処理能力を持つような構成を検討してください。動的重みが、サービスに割り当てられた重みに基づいている場合、最初のサイトに 2 倍の数のトラフィックを送信できます。

注: 負荷分散サービスへの重みの割り当ての詳細については、サービスへの重みの割り当てを参照してください。

動的加重の計算方法を示す図として、GSLB サービスがバインドされている GSLB 仮想サーバーについて考えます。GSLB サービスは負荷分散仮想サーバーで、2 つのサービスがバインドされています。GSLB サービスに割り当てられる重みは 3 です。2 つのサービスに割り当てられる重みは、それぞれ 1 と 2 です。この例では、ダイナミックウェイトを次のように設定すると、

- 無効**:**GSLB 仮想サーバーの累積重は、動的加重 (無効 = 1) と GSLB サービスの重み (3) の積なので、累積加重は 3 になります。
- **SERVICECOUNT:** カウントは、GSLB サービスに対応する負荷分散仮想サーバーにバインドされたサービスの数 (2) の合計であり、累積加重は動的ウェイト (2) と GSLB サービスのウェイト (3) の積で、6 です。
- **SERVICEWEIGHT**: 動的重みは、GSLB サービスに対応する負荷分散仮想サーバーにバインドされたサービスの加重（3）の合計であり、累積加重は動的重み（3）と GSLB サービスの重み（3）の積であり、9 です。

注: コンテンツスイッチング仮想サーバーが設定されている場合、動的ウェイトは適用されません。

コマンドラインインターフェイスを使用して動的ウェイトを使用するように **GSLB** 仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -dynamicWeight SERVICECOUNT | SERVICEWEIGHT
<!--NeedCopy-->
```

例**:**

```
set gslb vserver vserver-GSLB-1 -dynamicWeight SERVICECOUNT
<!--NeedCopy-->
```

構成ユーティリティを使用して動的ウェイトを使用するように **GSLB** 仮想サーバーを設定するには

1. [トラフィック管理] > [GSLB] > [仮想サーバー] に移動し、メソッドを変更する GSLB 仮想サーバー（たとえば、Vserver-GSLB-1）をダブルクリックします。
2. 「メソッド」セクションをクリックし、「ダイナミックウェイト」ドロップダウンリストから「サービスカウント」または「サービスウェイト」を選択します。

GSLB でパーシスタンスを設定する方法

August 15, 2023

永続性により、特定のドメイン名に対する一連のクライアント要求は、負荷分散されずに同じデータセンターに送信されます。特定のドメインに永続性が設定されている場合は、設定されている GSLB メソッドよりも優先されます。パーシスタンスは、クライアントトランザクションに関連する情報が、最初のリクエストを処理したインスタンスにローカルに保存されるデプロイに使用できます。たとえば、ショッピングカートを使用する電子商取引の展開では、サーバーが接続の状態を維持してトランザクションを追跡する必要があります。NetScaler アプライアンスは、クライアント要求を処理するデータセンターを選択します。パーシスタンスを有効にすると、それ以降のすべてのドメインネームシステム (DNS) 要求に対して、選択したデータセンターの同じ IP アドレスを転送します。永続セッションがダウンしているデータセンターを指している場合、NetScaler アプライアンスは構成済みの GSLB 方式を使用して新しいデータセンターを選択します。その後、クライアントからの後続のリクエストでも永続的になります。
GSLB で永続化するには、すべてのデータセンターの GSLB 仮想サーバーで同じ永続性識別子（PersistID）のセットを設定する必要があります。GSLB モジュールは、パーシスタンス識別子を使用して GSLB 仮想サーバーを一意に識別します。GSLB 仮想サーバーでソース IP パーシステンスが有効になっている場合、パーシステンスセッションもメトリック交換の一部として交換されます。NetScaler アプライアンスがサイト間のパーシステンスをサポートするには、参加しているすべての GSLB サイトでパーシステンス関連の構成を行う必要があります。Citrix では、ステートフルアプリケーションには GSLB での永続化を推奨しています。この場合、クライアントは後続のリクエストに対して同じアプリケーションインスタンスに再接続する必要があります。

GSLB の永続性は次の方法で実現できます。

- GSLB 仮想サーバーでの永続性
- GSLB サービスのサイト永続性

GSLB 仮想サーバーでの永続性

DNS リクエスト中は、GSLB 仮想サーバー上のパーシスタンスが使用されます。DNS リクエストのソース IP アドレスは、クライアントとデータセンター間のパーシスタンスセッションを作成するために使用されます。DNS クライアントは通常、（ISP 内の）背後にある一連のクライアントをプロキシするローカル DNS（LDNS）または DNS ゲートウェイです。GSLB 仮想サーバー上のパーシスタンスは、アプリケーションプロトコルに依存しません。
一般に、クライアントネットワークには複数の DNS ゲートウェイまたはローカルドメインネームサーバー (LDNS) が設定されます。Citrix では、適切なパーシステンスマスクを構成することをお勧めします。これは、ADC アプライアンスへの接続に使用されるアップストリーム LDNS デバイスに関係なく、クライアントは以前のリクエストを処理したのと同じデータセンターに永続化できるためです。LDNS IP アドレスのパーシスタンスセッションが作成されると、その LDNS を使用して接続しているすべてのエンドクライアントに同じデータセンター IP アドレスが与えられます。

GSLB サービスのサイト永続性

サイトパーシスタンスは、アプリケーションリクエストの処理中に有効になります。サイト永続性は HTTP Cookie を使用して実現されるため、HTTP トラフィックと HTTPS トラフィックに対してのみ機能します。Cookie は HTTP クライアント (ブラウザ) 上で管理されるため、DNS ゲートウェイの背後にいるクライアントを可視化できま

す。Cookie を使用してクライアントの永続性を実現すると、受信クライアントごとに ADC アプライアンスでリソースが消費されることはありません。遅延時間を設定して GSLB サービスを停止させると、サービスはアウトオブサービス (TROFS) 状態に移行します。永続性は、サービスが UP または TROFS 状態である限りサポートされます。つまり、サービスが TROFS とマークされた後、指定された遅延時間内に同じクライアントが同じサービスに対するリクエストを送信した場合、同じ GSLB サイト (データセンター) がそのリクエストを処理します。

エイリアスを使用してアプリケーションにアクセスする場合は、CNAME レコードが NetScaler アプライアンスでも構成されていることを確認してください。親子トポロジでは、エイリアスを介してアプリケーションにアクセスすると、サイトパーシスタンスは機能しません。

> 注
>
> 接続プロキシがサイト永続化方法として指定されていて、LB 仮想サーバーにも永続性を設定する必要がある場合は、ソース IP 永続化はお勧めしません。接続がプロキシされる場合、クライアントの実際の IP アドレスではなく、ADC アプライアンスが所有する IP アドレスが使用されます。
> Cookie パーシステンスやルールベースのパーシスタンスなど、HTTP (S) リクエストのソース IP を使用してクライアントを識別しない適切なパーシステンスを設定します。

送信元 **IP** アドレスに基づいてパーシステンスを設定

ソース IP パーシスタンスが GSLB 仮想サーバーで設定されている場合、DNS リクエストのソース IP アドレスに対してパーシステンスセッションが作成されます。拡張クライアントサブネット (ECS) 機能に応じて、DNS リクエストの送信元 IP アドレスは次のいずれかから取得されます。

- 着信 DNS 要求パケットの IP ヘッダーの送信元 IP
- DNS 要求の ECS オプション ECS の詳細については、「グローバルサーバー負荷分散に EDNS0 クライアントサブネットオプションを使用する」を参照してください。

クライアントの永続セッションは、永続タイムアウトまで続きます。タイムアウト期間が終了すると、既存の永続セッションはクリアされます。後続のリクエストでは、新しい GSLB の決定が行われ、別の GSLB サービス IP アドレスが選択される可能性があります。
GSLB 仮想サーバー上のソース IP パーシスタンスと GSLB サービスのサイトパーシスタンスは互いに補完し合っています。GSLB 仮想サーバーでソース IP パーシスタンスが無効になっている場合、GSLB 仮想サーバーは DNS が解決を試みるたびに異なる GSLB サービスを選択します。また、クライアントは別の GSLB サービスに接続し、アプリケーションリクエストを受信するデータセンターは、最初にクライアントにサービスを提供したデータセンターへの接続をプロキシします。これにより、レイテンシーが発生する可能性があります。そのため、GSLB 仮想サーバーでソース IP パーシスタンスを有効にすることで、アプリケーションリクエストのこのような頻繁なマルチホップを回避できます。ソース IP パーシステンスセッションの有効期限が切れてからクライアントが再接続した場合、サイトパーシスタンスにより、クライアントは最初にクライアントにサービスを提供していたデータセンターに戻されます。また、クライアントが DNS ゲートウェイを経由して接続し直しても、設定されているパーシスタンスマスクの範囲に含まれない場合は、サイトパーシスタンスと同様に、クライアントが最初のリクエストを処理したデータセンターに留まるのにも役立ちます。

CLI を使用して送信元 **IP** アドレスに基づいてパーシステンスを設定するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -persistenceType (SOURCEIP|NONE) -persistenceId
    <positive_integer> [-persistMask <netmask>]  - [timeout <mins>]
<!--NeedCopy-->
```

例**:**

```
set gslb vserver vserver-GSLB-1 -persistenceType SOURCEIP -
   persistenceId 23 -persistMask 255.255.255.255  - timeout 2
<!--NeedCopy-->
```

GUI を使用して送信元 **IP** アドレスに基づいてパーシステンスを設定するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動し、メソッドを変更する GSLB 仮想サーバー（たとえば、Vserver-GSLB-1）をダブルクリックします。
2. 「持続性」セクションをクリックし、「持続性」ドロップダウンリストから「**SOURCEIP** 」を選択し、次のパラメータを設定します。
    - パーシスタンス ID —パーシスタンス ID
    - タイムアウト-タイムアウト
    - IPv4 ネットマスクまたは IPv6 マスク長-パーシステントマスク

HTTP クッキーに基づくサイトパーシステンスの設定

サイトの永続性は、HTTP Cookie (「サイト Cookie」と呼ばれる) を使用してクライアントを同じサーバーに再接続することで実現されます。GSLB アプライアンスが選択した GSLB サイトの IP アドレスを送信してクライアントの DNS 要求に応答すると、クライアントはその GSLB サイトに HTTP 要求を送信します。その GSLB サイトのアプリケーションエンドポイントは HTTP ヘッダーにサイト Cookie を追加し、サイト永続性が有効になります。
クライアントキャッシュの有効期限が切れた後にクライアントが DNS クエリを送信した場合、DNS 要求は別の GSLB サイトに送信される可能性があります。新しい GSLB サイトは、クライアントリクエストヘッダーにあるサイト Cookie を使用して永続性を実装します。サイト永続化機能は次の条件で有効になります。

- ホストヘッダーのドメイン名がいずれかの GSLB ドメインと一致する場合
- アプリケーショントラフィックを受信する仮想サーバーを表す GSLB サービスでサイト永続性が有効になっている場合。

サイト Cookie には、クライアントが常時接続している選択した GSLB サービスに関する情報が含まれています。Cookie が指す GSLB サービスがダウンしているか、GLSB 構成から削除されている場合、トラフィックを受信する仮想サーバーは引き続きトラフィックを処理します。Cookie の有効期限は、NetScaler アプライアンスで設定され

ている Cookie ータイムアウトに基づいています。仮想サーバー名がすべてのサイトで同じでない場合は、パーシスタンス識別子を使用する必要があります。挿入されたクッキーは RFC 2109 に準拠しています。

NetScaler は、次の 2 種類のサイト永続性をサポートしています。

- 接続プロキシ
- HTTP リダイレクト

接続プロキシ

サイトパーシステンスの接続プロキシモードでは、後続のアプリケーション要求を受信するデータセンターが次のタスクを実行して接続を確立します。

1. サイト Cookie を挿入した GSLB サイトへの接続を作成します。
2. クライアントのリクエストを元のサイトにプロキシします。

   > 注:
   >
   > プロキシサーバーは、次の情報を使用して元のサイトとの接続を確立します。
   >
   > - 新しいサイトの SNIP が送信元 IP アドレスです。
   > - 元のサイトの GSLB サービスのパブリック IP アドレスが宛先 IP アドレスです。
   > - エフェメラルポートは送信元ポートで、GSLB サービスポートは宛先ポートです。
   > - GSLB サービスの種類に応じて、HTTP または HTTPS プロトコルのいずれかを使用します。

3. 元の GSLB サイトから応答を受け取ります。
4. その応答をクライアントに中継します。
5. 接続を閉じます。

HTTP リダイレクト

GSLB 設定で HTTP リダイレクトパーシスタンスが使用されている場合、新しいサイトは Cookie を最初に挿入したサイトにリクエストをリダイレクトします。リダイレクト URL のドメイン名はサイトドメインです。Cookie と SSL 証明書の両方が GSLB ドメインとサイトドメインの両方に適用できることを確認してください。GSLB とサイトドメインの両方に Cookie を適用するには、クッキードメインがサイト間 GSLB ドメインでなければなりません。SSL 証明書を GSLB とサイトドメインの両方に適用するには、SSL 仮想サーバーにバインドされた証明書がワイルドカード証明書である必要があります。

接続プロキシは、次の条件が満たされたときに発生します。

- GSLB に参加しているドメインにリクエストが送信されます。ドメインは URL/Host ヘッダーから取得されます。
- ローカルの GSLB サービスでは、接続プロキシが有効になっています。

- リクエストには、アクティブなリモート GSLB サービスの IP アドレスを含む有効な Cookie が含まれています。

> 注
>
> GSLB の親子構成では、GSLB サービスが子サイトで構成されていない場合でも、接続プロキシは意図したとおりに機能します。ただし、クライアント認証、クライアント IP アドレスの挿入、またはその他の SSL 固有の要件などの追加の構成がある場合は、サイトに明示的な GSLB サービスを追加し、それに応じて構成する必要があります。
>
> 親子トポロジの詳細については、「MEP プロトコルを使用した親子トポロジの展開」を参照してください。

CLI を使用して **HTTPCookie** に基づいて永続性を設定するには

コマンドプロンプトで入力します。

```
set gslb service <serviceName> -sitePersistence (ConnectionProxy [-
    sitePrefix <prefix>] | HTTPredirect -sitePrefix <prefix>)
<!--NeedCopy-->
```

例**:**

```
set gslb service service-GSLB-1 -sitePersistence ConnectionProxy
set gslb service service-GSLB-1 -sitePersistence HTTPRedirect -
    sitePrefix vserver-GSLB-1
<!--NeedCopy-->
```

GUI を使用して **Cookie** に基づいてパーシステンスを設定するには

1. [トラフィック管理] > [**GSLB**] > [サービス] に移動し、サイト永続性を設定するサービス (たとえば、Service-GSLB-1) を選択します。
2. 「サイトパーシステンス」セクションをクリックし、クッキーに基づいてパーシステンスを設定します。

クライアント接続を管理する

August 15, 2023

クライアント接続の管理を容易にするために、仮想サーバーへの接続の遅延クリーンアップを有効にできます。その後、DNS ポリシーを設定して、ローカル DNS トラフィックを管理できます。

仮想サーバー接続の遅延クリーンアップを有効にする

仮想サーバーの状態は、それにバインドされているサービスの状態に依存し、各サービスの状態は、それにバインドされているモニターによって異なります。サーバが低速またはダウンしている場合、モニタリングプローブはタイムアウトし、サーバを表すサービスは DOWN としてマークされます。仮想サーバは、そのサーバにバインドされているすべてのサービスが DOWN としてマークされている場合にのみ、DOWN としてマークされます。サービスおよび仮想サーバーは、ダウンしたときにすべての接続を終了するか、接続を許可するように構成できます。後者の設定は、サーバーが遅いためにサービスが DOWN とマークされる状況用です。

ダウン状態のフラッシュオプションを構成すると、NetScaler ADC アプライアンスは、ダウンしている GSLB サービスへの接続の遅延クリーンアップを実行します。

コマンドラインインターフェイスを使用して仮想サーバー接続の遅延クリーンアップを有効にするには

コマンドプロンプトで次のコマンドを入力して、遅延接続のクリーンアップを構成し、構成を確認します。

```
set gslb service <name> -downStateFlush (ENABLED | DISABLED)
show gslb service <name>
<!--NeedCopy-->
```

例**:**

```
set gslb service Service-GSLB-1 -downStateFlush ENABLED
Done

show gslb service Service-GSLB-1
Done
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバー接続の遅延クリーンアップを有効にするには

1. [トラフィック管理] > [**GSLB**] > [サービス] に移動し、サービスをダブルクリックします。
2. [その他の設定] セクションをクリックし、[ダウン状態のフラッシュ] オプションを選択します。

DNS ポリシーを使用してローカル **DNS** トラフィックを管理する

DNS ポリシーを使用して、ローカル DNS リゾルバーまたはネットワークの IP アドレスから事前定義されたターゲット GSLB サイトにトラフィックを誘導することで、サイトアフィニティを実装できます。これは、DNS 式を使用して DNS ポリシーを作成し、NetScaler ADC アプライアンスでポリシーをグローバルにバインドすることによって構成されます。

DNS 式

NetScaler ADC アプライアンスは、ドメイン固有のアクションを構成するために使用できる特定の定義済み DNS 式を提供します。このようなアクションは、たとえば、特定のリクエストを削除したり、特定のドメインの特定のビューを選択したり、特定のリクエストを特定の場所にリダイレクトしたりすることができます。

これらの DNS 式（ルールとも呼ばれます）を組み合わせて、NetScaler ADC アプライアンスでグローバルにバインドされる DNS ポリシーを作成します。

NetScaler ADC アプライアンスで使用可能な定義済みの DNS 修飾子のリストを以下に示します。

- CLIENT.UDP.DNS.DOMAIN.EQ(“domainname”)
- CLIENT.UDP.DNS.IS_AREC
- CLIENT.UDP.DNS.IS_AAAAREC
- CLIENT.UDP.DNS.IS_SRVREC
- CLIENT.UDP.DNS.IS_MXREC
- CLIENT.UDP.DNS.IS_SOAREC
- CLIENT.UDP.DNS.IS_PTRREC
- CLIENT.UDP.DNS.IS_CNAME
- CLIENT.UDP.DNS.IS_NSREC
- CLIENT.UDP.DNS.IS_ANYREC

CLIENT.UDP.DNS.DOMAIN DNS 式は、文字列式とともに使用できます。式の一部としてドメイン名を使用する場合は、ピリオド (.) で終わる必要があります。たとえば、CLIENT.UDP.DNS.DOMAIN.ENDSWITH (「abc.com.」)

構成ユーティリティを使用して式を作成するには

1. [式] テキストボックスの横にあるアイコンをクリックします。[追加] をクリックします。([フロータイプ] および [プロトコル] ドロップダウンリストボックスは空のままにします)。ルールを作成するには、次の手順に従います。
2.「修飾子」ボックスで、修飾子（LOCATION など）を選択します。
3. [演算子] ボックスで、演算子 (== など) を選択します。
4. [値] ボックスに値を入力します (例: アジア、日本…)。
5.「OK」をクリックします。[作成] をクリックし、[閉じる] をクリックします。ルールが作成されます。
6.「OK」をクリックします。

DNS アクションを構成する

DNS ポリシーには、ポリシールールが TRUE と評価されたときに実行される DNS アクションの名前が含まれます。DNS アクションでは、次のいずれかを実行できます。

- DNS ビューを構成した IP アドレスをクライアントに送信します。DNS ビューの詳細については、「DNS ビューの追加」を参照してください。
- 静的近接動作をオーバーライドする優先ロケーションのリストを参照した後、クライアントに GSLB サービスの IP アドレスを送信します。優先ロケーションの詳細については、「優先ロケーションの設定による静的近接動作のオーバーライド」を参照してください。
- DNS クエリーまたは応答 (DNS 応答書き換え) の評価によって決定された特定の IP アドレスをクライアントに送信します。
- アプライアンスの DNS キャッシュで検索を実行せずに、ネームサーバーに要求を転送します。
- リクエストをドロップします。

DNS 要求をドロップしたり、アプライアンスの DNS キャッシュをバイパスしたりするための DNS アクションを作成することはできません。DNS 要求をドロップする場合は、組み込みアクション dns_default_act_drop を使用します。DNS キャッシュをバイパスする場合は、組み込みアクション dns_default_act_cacheBypass を使用します。どちらのアクションも、[DNS ポリシーの作成] ダイアログボックスと [DNS ポリシーの構成] ダイアログボックスでカスタムアクションとともに使用できます。これらのビルトインアクションは、変更または削除できません。

コマンドラインインターフェイスを使用して **DNS** アクションを構成するには　コマンドプロンプトで次のコマンドを入力して、DNS アクションを構成し、構成を確認します。

```
add dns action <actionName> <actionType> (-IPAddress <ip_addr |
    ipv6_addr> ... | -viewName <string> | -preferredLocList <string>
    ...) [-TTL <secs>]

show dns action [<actionName>]
<!--NeedCopy-->
```

例　例 **1: DNS** 応答書き換えの設定。次の DNS アクションは、アクションがバインドされているポリシーが true と評価されたときに、クライアントに事前設定された IP アドレスを送信します。

```
add dns action dns_act_response_rewrite Rewrite_Response -IPAddress
    192.0.2.20 192.0.2.56 198.51.100.10
 Done

show dns action dns_act_response_rewrite
1)  ActionName:  dns_act_response_rewrite ActionType:  Rewrite_Response
     TTL:  3600               IPAddress:  192.0.2.20 192.0.2.56
    198.51.100.10
 Done
<!--NeedCopy-->
```

例 **2: DNS** ビューベースの応答の設定。次の DNS アクションは、DNS ビューを設定した IP アドレスをクライアントに送信します。

```
add dns action send_ip_from_view_internal_ip ViewName -viewName
    view_internal_ip
```



```
Done

show dns action send_ip_from_view_internal_ip
1)  ActionName:  send_ip_from_view_internal_ip ActionType:  ViewName
      ViewName:  view_internal_ip
 Done
<!--NeedCopy-->
```

例 **3:** 優先ロケーションリストに基づく応答の設定。次の DNS アクションは、指定されたロケーションのリストから選択する優先ロケーションに対応する IP アドレスをクライアントに送信します。

```
add dns action send_preferred_location GslbPrefLoc -preferredLocList NA
   .tx.ns1.\*.\*.* NA.tx.ns2.\*.\*.* NA.tx.ns3.\*.\*.*
Done

show dns action  send_preferred_location
1)  ActionName:  send_preferred_location ActionType:  GslbPrefLoc
   PreferredLocList:  "NA.tx.ns1.\*.\*.\*" "NA.tx.ns2.\*.\*.\*" "NA.tx.
   ns3.\*.\*.*"
 Done
<!--NeedCopy-->
```

NetScaler ADC 構成ユーティリティを使用して **DNS** アクションを構成するには

1. [トラフィック管理] > [DNS] > [アクション] に移動し、DNS アクションを作成または編集します。
2. [DNS アクションの作成] または [DNS アクションの構成] ダイアログボックスで、次のパラメータを設定します。

   - アクション名 (既存の DNS アクションでは変更できません)
   - Type (既存の DNS アクションでは変更できません)
     Type
     パラメーターを設定するには、次のいずれかの操作を行います。
     - DNS ビューに関連付けられた DNS アクションを作成するには、[名前の表示] を選択します。次に、[View Name] ボックスの一覧から、アクションで使用する DNS ビューを選択します。
     - 優先ロケーションリストを使用して DNS アクションを作成するには、[優先ロケーションリスト] を選択します。[優先する場所] に場所を入力し、[追加] をクリックします。DNS の場所を必要な数だけ追加します。
     - ポリシー評価に基づいて DNS 応答を書き換える DNS アクションを構成するには、[Rewrite Response] を選択します。[IP アドレス] に IP アドレスを入力し、[追加] をクリックします。IP アドレスを必要な数だけ追加します。
   - TTL ([レスポンスの書き換え] アクションタイプにのみ適用)

DNS ポリシーを構成する

DNS ポリシーは、静的 IP アドレスとカスタム IP アドレスを使用するロケーションデータベースで動作します。着信ローカル DNS 要求の属性は式の一部として定義され、ターゲットサイトは DNS ポリシーの一部として定義されます。アクションと式を定義するときに、一重引用符 (‘) のペアをワイルドカード修飾子として使用して、複数の場所を指定できます。DNS ポリシーが設定され、GSLB 要求を受信すると、まず、カスタム IP アドレスデータベースに対して、送信元のロケーション属性を定義するエントリが照会されます。

- DNS クエリーが LDNS から送信されると、LDNS の特性は設定されたポリシーに対して評価されます。一致すると、適切なアクション (サイトアフィニティ) が実行されます。LDNS の特性が複数のサイトと一致する場合、要求は LDNS の特性に一致するサイト間で負荷分散されます。
- カスタムデータベースにエントリが見つからない場合、スタティック IP アドレスデータベースにエントリが照会され、一致するものがあれば、上記のポリシー評価が繰り返されます。
- カスタムデータベースまたは静的データベースのいずれにもエントリが見つからない場合、最適なサイトが選択され、構成された負荷分散方式に基づいて DNS 応答で送信されます。

NetScaler ADC アプライアンスで作成された DNS ポリシーには、次の制限が適用されます。

- 最大 64 のポリシーがサポートされます。
- DNS ポリシーは NetScaler ADC アプライアンスにグローバルであり、特定の仮想サーバーまたはドメインに適用することはできません。
- ドメインまたは仮想サーバ固有のポリシーのバインドはサポートされていません。

DNS ポリシーを使用して、特定の IP アドレス範囲に一致するクライアントを特定のサイトに誘導できます。たとえば、地理的に分離された複数の GSLB サイトを持つ GSLB セットアップがある場合、IP アドレスが特定の範囲内にあるすべてのクライアントを特定のデータセンターに誘導できます。

TCP ベースと UDP ベースの DNS トラフィックの両方を評価できます。ポリシー式は、サーバ上の UDP ベースの DNS トラフィック、およびクライアント側の UDP ベースの DNS トラフィックと TCP ベースの DNS トラフィックの両方で使用できます。さらに、次の DNS 質問タイプ（または QTYPE 値）のみを含むクエリと応答を評価する式を設定できます。

- A
- AAAA
- NS
- SRV
- PTR
- CNAME
- SOA
- MX
- ANY

次のレスポンスコード (RCODE 値) もサポートされています。

- NOERROR-エラーなし
- FORMERR-フォーマットエラー
- SERVFAIL-サーバー障害
- NXDOMAIN-存在しないドメイン
- NOTIMP-クエリータイプが実装されていません
- 拒否-クエリが拒否されました

DNS トラフィックを評価する式を設定できます。DNS 式は、DNS.REQ または DNS.RES プレフィックスで始まります。関数は、クエリーされたドメイン、クエリータイプ、およびキャリアプロトコルを評価するために利用できます。DNS 式の詳細については、「ポリシーの設定およびリファレンス」の「DNS メッセージの評価とそのキャリアプロトコルを識別するための式」を参照してください。

コマンドラインインターフェイスを使用して **DNS** ポリシーを追加するには　コマンドプロンプトで次のコマンドを入力して DNS ポリシーを作成し、構成を確認します。

```
add dns policy <name> <rule> <actionName>
show dns policy <name>
<!--NeedCopy-->
```

例**:**

```
> add dns policy-GSLB-1 'CLIENT.UDP.DNS.DOMAIN.EQ("domainname")'
    my_dns_action
 Done
> show dns policy-GSLB-1
    Name: policy-GSLB-1
    Rule: CLIENT.UDP.DNS.DOMAIN.EQ("domainname")
    Action Name: my_dns_action
    Hits: 0
    Undef Hits: 0

 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して構成済みの **DNS** ポリシーを削除するには　コマンドプロンプトで入力します。

```
rm dns policy <name>
<!--NeedCopy-->
```

NetScaler ADC 構成ユーティリティを使用して **DNS** ポリシーを構成するには

1. [トラフィック管理] > [DNS] > [ポリシー] に移動し、DNS ポリシーを作成します。
2. [DNS ポリシーの作成] または [DNS ポリシーの構成] ダイアログボックスで、次のパラメータを設定します。
    - ポリシー名 (既存のポリシーでは変更できません)

- アクション
- [式]
  式を指定するには、次の操作を行います。
  a) [追加] をクリックし、表示されるドロップダウンボックスで、式の開始に使用する式要素を選択します。2 つ目のリストが表示されます。このリストには、最初の式要素の直後に使用できる一連の式要素が含まれています。
  b) 2 番目のリストで、目的の式要素を選択し、ピリオドを入力します。
  c) 各選択の後に、ピリオドを入力すると、有効な式要素の次のセットがリストに表示されます。式の要素を選択し、目的の式が得られるまで関数の引数を入力します。

3.「作成」または「OK」をクリックし、「閉じる」をクリックします。

DNS ポリシーのバインド

DNS ポリシーは NetScaler ADC アプライアンスでグローバルにバインドされ、構成されているすべての GSLB 仮想サーバーで使用できます。DNS ポリシーがグローバルにバインドされている場合でも、式でドメインを指定することで、ポリシーの実行を特定の GSLB 仮想サーバーに制限できます。

注:bind dns global コマンドは REQ_OVERRIDE および RES_OVERRIDE を有効なバインドポイントとして受け付けますが、DNS ポリシーはグローバルにしかバインドできないため、これらのバインドポイントは冗長です。DNS ポリシーは、REQ_DEFAULT および RES_DEFAULT のバインドポイントにのみバインドします。

コマンドラインインターフェイスを使用して **DNS** ポリシーをグローバルにバインドするには　コマンドプロンプトで次のコマンドを入力して、DNS ポリシーをグローバルにバインドし、構成を確認します。

```
bind dns global <policyName> <priority> [-gotoPriorityExpression <
    string>] [-type <type>]
show dns global -type <type>
<!--NeedCopy-->
```

例**:**

```
bind dns global policy-GSLB-1 10 -gotoPriorityExpression END
 Done
show dns global -type REQ_DEFAULT
1)  Policy Name: policy-GSLB-1
    Priority: 10
    GotoPriorityExpression: END
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して **DNS** ポリシーをグローバルにバインドするには

1. [トラフィック管理] > [DNS] > [ポリシー] に移動します。

2. 詳細ウィンドウで、[グローバルバインド] をクリックします。
3. [DNS ポリシーをグローバルにバインド/バインド解除] ダイアログボックスで、[ポリシーの挿入] をクリックします。
4. [Policy Name] 列で、バインドするポリシーをリストから選択します。または、一覧の [新しいポリシー] をクリックし、[DNS ポリシーの作成] ダイアログボックスでパラメータを設定して DNS ポリシーを作成します。
5. すでにグローバルにバインドされているポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[DNS ポリシーの構成] ダイアログボックスでポリシーを変更し、[OK] をクリックします。
6. ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
7. ポリシーに割り当てられているプライオリティを変更するには、プライオリティ値をダブルクリックし、新しい値を入力します。
8. 割り当てられた優先度を再生成するには、「優先度を再生成する」をクリックします。優先順位の値は、評価の順序に影響を与えることなく、100 から始まり、10 ずつ増分するように変更されます。
9. 「OK」をクリックします。

コマンドラインインターフェイスを使用して **DNS** ポリシーのグローバルバインディングを表示するには　コマンドプロンプトで入力します。

```
show dns global
```

構成ユーティリティを使用して **DNS** ポリシーのグローバルバインディングを表示するには

1. [トラフィック管理] > [**DNS**] > [ポリシー] に移動します。
2. 詳細ウィンドウで、[グローバルバインド] をクリックします。すべての DNS ポリシーのグローバルバインディングがこのダイアログボックスに表示されます。

DNS ビューの追加

DNS ビューを構成して、さまざまなタイプのクライアントを識別し、同じ GSLB ドメインをクエリするクライアントのグループに適切な IP アドレスを提供できます。DNS ビューは、クライアントに送り返される IP アドレスを選択する DNS ポリシーを使用して構成されます。

たとえば、会社のドメイン用に GSLB を設定し、サーバーを会社のネットワークでホストしている場合、会社の内部ネットワーク内からドメインを照会するクライアントに、パブリック IP アドレスの代わりにサーバーの内部 IP アドレスを提供できます。一方、インターネットからドメインの DNS を照会するクライアントには、ドメインのパブリック IP アドレスを指定できます。

DNS ビューを追加するには、最大 31 文字の名前を割り当てます。先頭の文字は数字または文字でなければなりません。次の文字も使用できます:@ _-.(ピリオド): (コロン) # とスペース ()。ビューを追加したら、ポリシーをクライア

ントとネットワークの一部に関連付けるように設定し、ポリシーをグローバルにバインドします。DNS ポリシーを構成してバインドするには、DNS ポリシーを使用したローカル **DNS** トラフィックの管理を参照してください。

コマンドラインインターフェイスを使用して **DNS** ビューを追加するには

コマンドプロンプトで次のコマンドを入力して DNS ビューを作成し、構成を確認します。

```
add dns view <viewName>
show dns view <viewName>
<!--NeedCopy-->
```

例**:**

```
add dns view PrivateSubnet
show dns view PrivateSubnet
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **DNS** ビューを削除するには

コマンドプロンプトで入力します。

```
rm dns view <viewName>
<!--NeedCopy-->
```

構成ユーティリティを使用して **DNS** ビューを追加するには

[トラフィック管理] > [DNS] > [ビュー] に移動し、DNS ビューを追加します。

DNS ポリシーの作成方法と DNS ポリシーをグローバルにバインドする方法の詳細については、「DNS ポリシーを使用したローカル **DNS** トラフィックの管理」を参照してください。

近接データベースの **GSLB** を構成する

August 15, 2023

GSLB を近接設定すると、クライアントのリクエストは最も近いデータセンターに転送されます。近接ベースの GSLB 方式の主な利点は、利用可能な最も近いデータセンターを選択することで応答時間が短縮されることです。このような展開は、大量のデータへの高速アクセスを必要とするアプリケーションにとって重要です。

GSLB は、ラウンドトリップ時間（RTT）、静的近接時間、またはその 2 つの組み合わせに基づいて近接するように設定できます。

動的ラウンドトリップタイム **(RTT)** 方式の設定

動的ラウンドトリップ時間 (RTT) は、クライアントのローカル DNS サーバーとデータリソース間のネットワークの時間または遅延の尺度です。動的 RTT を測定するために、NetScaler アプライアンスはクライアントのローカル DNS サーバーをプローブし、RTT メトリック情報を収集します。次に、アプライアンスはこのメトリックを使用して負荷分散を決定します。グローバルサーバー負荷分散は、ネットワークの状態をリアルタイムで監視し、クライアントの要求を RTT 値が最も低いデータセンターに動的に送信します。

動的方式で近接するように GSLB を設定するには、まず基本的な GSLB 設定を設定し、次に動的 RTT を設定する必要があります。

まず、ローカルとリモートの 2 つの GSLB サイトを作成します。次に、ローカルサイト用に GSLB 仮想サーバーと GSLB サービスを作成し、サービスを仮想サーバーにバインドします。次に、ADNS サービスを作成し、GSLB を構成するドメインをローカルサイトの GSLB 仮想サーバーにバインドします。最後に、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

基本的な GSLB セットアップを設定したら、ダイナミック RTT 方式を設定します。

ロードバランシングにダイナミック RTT 方式を使用するように GSLB 仮想サーバーを構成する方法の詳細については、ダイナミック RTT の設定を参照してください。

静的近接度を構成する

GSLB の静的近接方法では、IP アドレスベースの静的近接データベースを使用して、クライアントのローカル DNS サーバーと GSLB サイト間の近接性を判断します。NetScaler ADC アプライアンスは、近接基準に最も一致するサイトの IP アドレスで応答します。

地理的に異なる場所にある 2 つ以上の GSLB サイトが同じコンテンツを配信している場合、NetScaler アプライアンスは IP アドレス範囲のデータベースを管理し、そのデータベースを使用して受信クライアント要求の送信先となる GSLB サイトを決定します。

静的近接による近接用に GSLB を設定するには、まず基本的な GSLB 設定を設定し、次に静的近接を設定する必要があります。

まず、ローカルとリモートの 2 つの GSLB サイトを作成します。次に、ローカルサイト用に GSLB 仮想サーバーと GSLB サービスを作成し、サービスを仮想サーバーにバインドします。次に、ADNS サービスを作成し、GSLB を構成するドメインをローカルサイトの GSLB 仮想サーバーにバインドします。最後に、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

基本的な GSLB セットアップを設定したら、静的な近接を設定します。

負荷分散に静的近接を使用するように GSLB 仮想サーバーを構成する方法の詳細については、静的近接の設定を参照してください。

スタティック近接およびダイナミック **RTT** の設定

ブランチオフィスなどの内部ネットワークからクライアントが来ている場合は、静的近接性と動的 RTT を組み合わせて使用するように GSLB 仮想サーバーを構成できます。ブランチオフィスやその他の内部ネットワークからのクライアントが、クライアントネットワークに地理的に近い特定の GSLB サイトに誘導されるように GSLB を構成できます。その他のリクエストには、ダイナミック RTT を使用できます。

まず、ローカルとリモートの 2 つの GSLB サイトを作成します。次に、ローカルサイト用に GSLB 仮想サーバーと GSLB サービスを作成し、サービスを仮想サーバーにバインドします。次に、ADNS サービスを作成し、GSLB を構成するドメインをローカルサイトの GSLB 仮想サーバーにバインドします。最後に、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

基本的な GSLB セットアップを設定したら、内部ネットワークから発信されるすべてのトラフィックに静的近接性を使用し、その他すべてのトラフィックにダイナミック RTT を使用するように GSLB 仮想サーバを設定します。

スタティック近接の設定方法の詳細については、スタティック近接の設定を参照してください 。ダイナミック RTT の設定方法の詳細については、ダイナミック RTT の設定を参照してください。

GSLB セットアップを障害から保護する

August 15, 2023

GSLB サイトまたは GSLB 仮想サーバーの障害から GSLB セットアップを保護するには、以下を構成します。

- バックアップ GSLB 仮想サーバー
- 複数の IP アドレスで応答する NetScaler ADC アプライアンス
- GSLB ドメインのバックアップ IP アドレス

また、スピルオーバーを使用して、余分なトラフィックをバックアップ仮想サーバーに転送することもできます。

バックアップ **GSLB** 仮想サーバーの設定

GSLB 仮想サーバーのバックアップエンティティを設定すると、GSLB 仮想サーバーがダウンしてもサイトへの DNS トラフィックが中断されないようになります。バックアップエンティティは、別の GSLB 仮想サーバーでも、バックアップ IP アドレスでもかまいません。バックアップエンティティが設定されている場合、プライマリ GSLB 仮想サーバーがダウンすると、バックアップエンティティは DNS 要求を処理します。プライマリ GSLB 仮想サーバーが再び復帰したときに何が起こるかを指定するには、プライマリ仮想サーバーを手動で引き継ぐまで(DisablePrimaryonDown オプションを使用して)、トラフィックを処理し続けるようにバックアップエンティティを構成できます。

注:1 つのバックアップエンティティを複数の GSLB 仮想サーバーのバックアップとして構成できます。

コマンドラインインターフェイスを使用してバックアップ **GSLB** 仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、GSLB 仮想サーバーをバックアップ仮想サーバーとして設定し、構成を確認します。

```
set gslb vserver <name> -backupVServer <name> [-disablePrimaryOnDown (
    ENABLED | DISABLED)]

show gslb vserver <name>
<!--NeedCopy-->
```

例**:**

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2 -
    disablePrimaryOnDown ENABLED
show gslb vserver vserver-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** 仮想サーバーをバックアップ仮想サーバーとして設定するには

1. 「トラフィック管理」>「**GSLB** 」>「仮想サーバー」に移動し、GSLB 仮想サーバーをダブルクリックします。
2. [バックアップ仮想サーバ] セクションを選択し、バックアップ仮想サーバを選択します。

複数の **IP** アドレスで応答するように **GSLB** セットアップを構成する

一般的な DNS 応答には、最もパフォーマンスに優れる GSLB サービスの IP アドレスが含まれます。ただし、複数の IP 応答（MIR）を有効にすると、NetScaler ADC アプライアンスはレスポンスの最初のレコードとして最適な GSLB サービスを送信し、残りのアクティブなサービスを追加レコードとして追加します。MIR が無効になっている場合（デフォルト）、NetScaler ADC アプライアンスはレスポンスの唯一のレコードとして最適なサービスを送信します。

コマンドラインインターフェイスを使用して複数の **IP** 応答用に **GSLB** 仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、複数の IP レスポンスに対応する GSLB 仮想サーバーを構成し、構成を確認します。

```
set gslb vserver<name> -MIR (ENABLED | DISABLED)
-  show gslb vserver <name>
<!--NeedCopy-->
```

例**:**

```
set gslb vserver vserver-GSLB-1 -MIR ENABLED
 show gslb vserver <vserverName>
<!--NeedCopy-->
```


構成ユーティリティを使用して複数の **IP** 応答に **GSLB** 仮想サーバーを設定するには

1. 「トラフィック管理」>「**GSLB** 」>「仮想サーバー」に移動し、バックアップ仮想サーバーを設定する GSLB 仮想サーバー（vserver-GSLB-1 など）をダブルクリックします。
2. ［詳細設定］タブの［この仮想サーバーが「稼働中」の場合］で、［すべての「アクティブな」サービス IP をレスポンス（MIR）で送信する］チェックボックスをオンにして、［**OK**］を選択します。

DOWN 時に空のアドレスレコードで応答するように **GSLB** 仮想サーバを設定する

DNS 応答には、要求されたドメインの IP アドレス、またはドメインの IP アドレスが DNS サーバーによって認識されないという回答が含まれる場合があります。この場合、クエリは別のネームサーバーに転送されます。DNS クエリに対して可能な応答は、これらだけです。

GSLB 仮想サーバーが無効またはダウン状態の場合、その仮想サーバーにバインドされた GSLB ドメインの DNS クエリへの応答には、仮想サーバーにバインドされたすべてのサービスの IP アドレスが含まれます。ただし、この場合に空のダウンレスポンス（EDR）を送信するように GSLB 仮想サーバーを構成できます。このオプションを設定すると、ダウン状態の GSLB 仮想サーバーからの DNS 応答には IP アドレスレコードが含まれませんが、応答コードは成功します。これにより、クライアントがダウンしている GSLB サイトに接続しようとするのを防ぎます。

注: この設定は、適用する仮想サーバーごとに構成する必要があります。

コマンドラインインターフェイスを使用して空のダウンレスポンス用に **GSLB** 仮想サーバーを設定するには

コマンドプロンプトで入力します。

```
set gslb vserver<name> -EDR (ENABLED | DISABLED)
<!--NeedCopy-->
```

例**:**

```
> set gslb vserver vserver-GSLB-1 -EDR ENABLED
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** 仮想サーバーを空のダウンレスポンスに設定するには

1. 「トラフィック管理」>「**GSLB** 」>「仮想サーバー」に移動し、バックアップ仮想サーバーを設定する GSLB 仮想サーバー（vserver-GSLB-1 など）をダブルクリックします。
2. [詳細設定] タブの [この仮想サーバーが [ダウンしているとき] で、[応答でサービスの IP アドレスを送信しない (EDR)] チェックボックスをオンにします。
3. ［**OK**］をクリックします。

GSLB ドメインのバックアップ IP アドレスの設定

GSLB 設定のバックアップサイトを設定できます。この構成を設定すると、すべてのプライマリサイトがダウンすると、バックアップサイトの IP アドレスが DNS 応答で提供されます。

通常、GSLB 仮想サーバーがアクティブな場合、その仮想サーバーは、構成された GSLB 方式によって選択されたアクティブサイト IP アドレスのいずれかを使用して DNS 応答を送信します。GSLB 仮想サーバーで設定されているすべてのプライマリサイトが非アクティブ（ダウン状態）の場合、オーソリテイティブドメインネームシステム（ADNS）サーバーまたは DNS サーバーは、バックアップサイトの IP アドレスとともに DNS 応答を送信します。

注：バックアップ IP アドレスが送信されると、永続性は考慮されません。

コマンドラインインターフェイスを使用してドメインのバックアップ IP アドレスを設定するには

コマンドプロンプトで次のコマンドを入力してバックアップ IP アドレスを設定し、構成を確認します。

```
set gslb vserver <name> -domainName <string> -backupIP <IPAddress>
show gslb vserver <name>
<!--NeedCopy-->
```

例:

```
set gslb vserver vserver-GSLB-1 -domainName www.abc.com -backupIP
    10.102.29.66
show gslb vserver vserver-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用してドメインのバックアップ IP アドレスを設定するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動し、バックアップドメインをバインドする GSLB 仮想サーバー (たとえば、VServer-GSLB-1) をダブルクリックします。
2. 「ドメイン」セクションをクリックし、GSLB ドメインを設定し、「バックアップ IP」フィールドにバックアップドメインの **IP** アドレスを指定します。

余分なトラフィックをバックアップ仮想サーバに転送する

プライマリ GSLB 仮想サーバーへの接続数が設定されたしきい値を超えたら、スピルオーバーオプションを使用して新しい接続をバックアップ GSLB 仮想サーバーに転送できます。このしきい値は、動的に計算することも、手動で設定することもできます。プライマリ仮想サーバーへの接続数がしきい値を下回ると、プライマリ GSLB 仮想サーバーはクライアントリクエストの処理を再開します。

スピルオーバーを使用して永続性を設定できます。永続性が構成されている場合、そのクライアントがプライマリ仮想サーバーにまだ接続されていない場合、新しいクライアントはバックアップ仮想サーバーに転送されます。永続性

が設定されている場合、プライマリ仮想サーバーへの接続数がしきい値を下回っても、バックアップ仮想サーバーに転送された接続はプライマリ仮想サーバーに戻されません。代わりに、バックアップ仮想サーバーは、ユーザーが接続を終了するまでそれらの接続を処理し続けます。一方、プライマリ仮想サーバーは新しいクライアントを受け入れます。

しきい値は、接続数、帯域幅、サービスの状態によって測定できます。

バックアップ仮想サーバーが設定されたしきい値に達し、追加の負荷に耐えられない場合、プライマリ仮想サーバーはすべての要求を指定されたリダイレクト URL に転送します。プライマリ仮想サーバーでリダイレクト URL が設定されていない場合、後続のリクエストはドロップされます。

スピルオーバー機能により、プライマリ GSLB 仮想サーバーに障害が発生した場合に、リモートバックアップ GSLB サービス（バックアップ GSLB サイト）にクライアント要求が殺到するのを防ぎます。これは、モニターがリモートの GSLB サービスにバインドされていて、サービスに障害が発生して状態が DOWN になった場合に発生します。ただし、スピルオーバー機能により、モニターはリモート GSLB サービスの状態を引き続き稼働させています。

この問題を解決する一環として、GSLB サービスでは、プライマリ状態と有効状態の 2 つの状態が維持されます。プライマリ状態はプライマリ仮想サーバーの状態で、有効状態は仮想サーバー（プライマリとバックアップチェーン）の累積状態です。仮想サーバーチェーン内の仮想サーバーのいずれかが稼働している場合、実効状態は稼働状態に設定されます。プライマリ VIP がしきい値に達したことを示すフラグも表示されます。しきい値は、接続数または帯域幅のいずれかで測定できます。

サービスが GSLB の対象と見なされるのは、そのプライマリの状態が UP の場合に限られます。トラフィックがバックアップ GSLB サービスに送信されるのは、すべてのプライマリ仮想サーバが DOWN の場合だけです。通常、このような展開には、バックアップ GSLB サービスが 1 つしかありません。

GSLB サービスにプライマリ状態と有効状態を追加すると、次の効果があります。

- 送信元 IP パーシステンスが設定されている場合、選択したサイトのプライマリ仮想サーバーが稼働していてしきい値を下回っている場合にのみ、ローカル DNS は以前に選択したサイトに送信されます。ラウンドロビンモードではパーシスタンスは無視できます。
- Cookie ベースの永続性が構成されている場合、クライアント要求は、選択したサイトのプライマリ仮想サーバーが UP している場合にのみリダイレクトされます。
- プライマリ仮想サーバが飽和状態に達し、バックアップ VIP が存在しないかダウンしている場合、有効状態は DOWN に設定されます。
- 外部モニターが HTTP-HTTPS 仮想サーバーにバインドされている場合、モニターはプライマリ状態を決定します。
- プライマリ仮想サーバへのバックアップ仮想サーバが存在せず、プライマリ仮想サーバがそのしきい値に達した場合、有効状態は DOWN に設定されます。

コマンドラインインターフェイスを使用してバックアップ **GSLB** 仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、バックアップ GSLB 仮想サーバーを構成し、構成を確認します。


```
set gslb vserver <name> -soMethod <method> -soThreshold <threshold> -
    soPersistence ( \*\*ENABLED\*\* | \*\*DISABLED\*\* ) -
    soPersistenceTimeout <timeout>
show gslb vserver <name>
<!--NeedCopy-->
```

例**:**

```
set gslb vserver Vserver-GSLB-1 -soMethod CONNECTION -soThreshold 1000
    -soPersistence ENABLED -soPersistenceTimeout 2
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用してバックアップ **GSLB** 仮想サーバーを構成するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動し、バックアップとして構成する仮想サーバー (vserver-LB-1 など) をダブルクリックします。
2. [**SpillOver**] セクションをクリックし、次のパラメータを設定します。
   - メソッド—メソッド
   - しきい値—SO しきい値
   - パーシステンスタイムアウト (分) —SO パーシステンスタイムアウト
3. 「パーシステンス」オプションを選択し、「**OK**」をクリックします。

障害回復用に **GSLB** を構成する

August 15, 2023

ダウンタイムにはコストがかかるため、ディザスタリカバリ機能は非常に重要です。GSLB 用に構成された NetScaler アプライアンスは、負荷が最も低いデータセンターまたは最もパフォーマンスの高いデータセンターにトラフィックを転送します。この構成は、アクティブ-アクティブ設定と呼ばれ、パフォーマンスが向上するだけでなく、セットアップの一部であるデータセンターがダウンした場合に、トラフィックを他のデータセンターにルーティングすることで、ディザスタリカバリを即座に実行できます。または、障害復旧のみを目的としたアクティブ/スタンバイ GSLB 設定を構成することもできます。

アクティブ/スタンバイデータセンターのセットアップでの災害復旧用の **GSLB** の設定

従来のディザスタリカバリのセットアップには、アクティブデータセンターとスタンバイデータセンターが含まれます。スタンバイデータセンターはリモートサイトです。障害イベントの結果としてフェイルオーバーが発生し、プライマリのアクティブなデータセンターが非アクティブになると、スタンバイデータセンターが稼働状態になります。

アクティブ/スタンバイデータセンターのセットアップにおけるディザスタリカバリの設定は、次のタスクで構成されます。

- アクティブなデータセンターを作成します。
  - ローカルの GSLB サイトを追加します。
  - アクティブなデータセンターを表す GSLB 仮想サーバーを追加します。
  - ドメインを GSLB 仮想サーバーにバインドします。
  - gslb サービスを追加し、そのサービスをアクティブな GSLB 仮想サーバーにバインドします。
- スタンバイデータセンターを作成します。
  - リモート gslb サイトを追加します。
  - スタンバイデータセンターを表す gslb vserver を追加します。
  - スタンバイデータセンターを表す gslb サービスを追加し、そのサービスをスタンバイ gslb vserver にバインドします。
  - スタンバイ GSLB 仮想サーバーをアクティブな GSLB 仮想サーバーのバックアップ仮想サーバーとして構成することにより、スタンバイデータセンターを指定します。

プライマリデータセンターを構成したら、バックアップデータセンターの設定を複製し、そのサイトの GSLB 仮想サーバーをバックアップ仮想サーバーとして指定することで、スタンバイ GSLB サイトとして指定します。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

コマンドラインインターフェイスを使用してスタンバイ **GSLB** サイトを指定するには

アクティブサイトとリモートサイトの両方で、コマンドプロンプトで次のように入力します。

```
set gslb vserver <name> -backupVserver <string>
<!--NeedCopy-->
```

例:

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2
<!--NeedCopy-->
```

構成ユーティリティを使用してスタンバイサイトを構成するには

1. [トラフィック管理] > [GSLB] > [仮想サーバー] に移動し、プライマリサイトの GSLB 仮想サーバーをダブルクリックします。
2. 「仮想サーバーのバックアップ」セクションをクリックし、バックアップ仮想サーバーを選択します。

デフォルトでは、プライマリ仮想サーバーがアクティブになると、トラフィックの受信を開始します。ただし、プライマリ仮想サーバーがアクティブになった後でもトラフィックをバックアップ仮想サーバーに転送したい場合は、「ダウン時にプライマリを無効にする」オプションを使用してください。

アクティブ/アクティブデータセンターのセットアップでのディザスタリカバリの設定

両方の GSLB サイトがアクティブなアクティブ-アクティブ GSLB デプロイでは、スタンバイデータセンターの設置時に発生する可能性のあるリスクを排除できます。このような設定では、ウェブまたはアプリケーションのコンテンツを地理的に離れた場所にミラーリングできます。これにより、各分散型データセンターで一貫してデータを使用できるようになります。

アクティブ-アクティブデータセンターのセットアップで災害復旧用に GSLB を設定するには、まず最初のデータセンターで基本的な GSLB 設定を設定し、次に他のすべてのデータセンターを設定する必要があります。

まず、少なくとも 2 つの GSLB サイトを作成します。次に、ローカルサイト用に GSLB 仮想サーバーと GSLB サービスを作成し、サービスを仮想サーバーにバインドします。次に、ADNS サービスを作成し、GSLB を構成するドメインをローカルサイトの GSLB 仮想サーバーにバインドします。最後に、ローカルサイトで、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。

最初のデータセンターを構成したら、セットアップの他のデータセンターの構成を複製します。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

加重ラウンドロビンを使用したディザスタリカバリの構成

加重ラウンドロビン方式を使用するように GSLB を設定すると、重みが GSLB サービスに追加され、設定された割合の受信トラフィックが各 GSLB サイトに送信されます。たとえば、トラフィックの 80% をあるサイトに、トラフィックの 20% を別のサイトに転送するように GSLB 設定を構成できます。これを行うと、NetScaler アプライアンスは、2 番目のサイトに送信するリクエストごとに 4 つのリクエストを最初のサイトに送信します。

加重ラウンドロビン方式を設定するには、まずローカルとリモートの 2 つの GSLB サイトを作成します。次に、ローカルサイト用に GSLB 仮想サーバーと GSLB サービスを作成し、サービスを仮想サーバーにバインドします。GSLB メソッドをラウンドロビンとして設定します。次に、ADNS サービスを作成し、GSLB を設定するドメインを GSLB 仮想サーバーにバインドします。最後に、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。

ネットワーク内の物理サーバーを表す各サービスには、それぞれ重みが関連付けられています。そのため、GSLB サービスには、それにバインドされたすべてのサービスの重みの合計である動的重みが割り当てられます。次に、トラフィックは、特定のサービスの動的重みと全体の重みの比率に基づいて、GSLB サービス間で分割されます。動的重みの代わりに、GSLB サービスごとに個別の重みを設定することもできます。

サービスにウェイトが関連付けられていない場合、GSLB 仮想サーバーを構成して、バインドされたサービス数を使用してウェイトを動的に計算できます。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

基本的な GSLB 設定を設定したら、個々のサービスに設定された重みに従って、設定された GSLB サイト間でトラフィックが分割されるように、重み付けラウンドロビン方式を設定する必要があります。

コマンドラインインターフェイスを使用してサービスに重みを割り当てるように仮想サーバーを構成するには

コマンドプロンプトで、新しい負荷分散仮想サーバーを作成するか、既存の負荷分散仮想サーバーを構成するかに応じて、次のコマンドのいずれかを入力します。

```
add lb vserver <name>@ -weight <WeightValue> <ServiceName>
set lb vserver <name>@ -weight <WeightValue> <ServiceName>
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
set lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してダイナミックウェイトを設定するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -dynamicWeight DynamicWeightType
<!--NeedCopy-->
```

例:

```
set gslb vserver Vserver-GSLB-1 -dynamicWeight ServiceWeight
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** サービスにウェイトを追加するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -serviceName GSLBServiceName -weight
   WeightValue
<!--NeedCopy-->
```

例:

```
set gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1 -weight 1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスに重みを割り当てるように仮想サーバーを構成するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバー (たとえば、VServer-LB-1) をダブルクリックします。
2. 「サービス」セクションをクリックして、サービスの重みを設定します。

設定ユーティリティを使用して **GSLB** サービスにウェイトを追加するには

1. [トラフィック管理] > [GSLB] > [仮想サーバー] に移動し、仮想サーバー (たとえば、Vserver-GSLB-1) をダブルクリックします
2. 「サービス」セクションをクリックし、「重量」フィールドにサービスの重みを設定します。

設定ユーティリティを使用してダイナミックウェイトを設定するには

1. [トラフィック管理] > [GSLB] > [仮想サーバー] に移動し、仮想サーバー (たとえば、Vserver-GSLB-1) をダブルクリックします。
2. 「メソッド」セクションをクリックし、「ダイナミックウェイト」ドロップダウンリストから「**SERVICEWEIGHT**」を選択します。

データセンターパーシスタンスによるディザスタリカバリの設定

リクエストの負荷分散を行うのではなく、同じサーバーとの接続を維持する必要があるウェブアプリケーションには、データセンターの永続性が必要です。たとえば、電子商取引ポータルでは、クライアントと同じサーバー間の接続を維持することが重要です。このようなアプリケーションでは、HTTP リダイレクトパーシステンスをアクティブ-アクティブ設定で設定できます。

データセンターの永続性を備えた災害復旧用に GSLB を設定するには、まず基本的な GSLB 設定を設定し、次に HTTP リダイレクトパーシステンスを設定する必要があります。

まず、ローカルとリモートの 2 つの GSLB サイトを作成します。次に、ローカルサイト用に GSLB 仮想サーバーと GSLB サービスを作成し、サービスを仮想サーバーにバインドします。次に、ADNS サービスを作成し、GSLB を設定するドメインをローカルサイトの GSLB 仮想サーバーにバインドします。次に、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。最後に、リモート構成の前の手順を複製するか、NetScaler ADC アプライアンスを構成して GSLB 構成を自動同期させます。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

基本的な GSLB セットアップを設定したら、HTTP リダイレクトの優先順位を設定して、データセンターの永続性を有効にします。

コマンドラインインターフェイスを使用して **HTTP** リダイレクトを設定するには

コマンドプロンプトで次のコマンドを入力して HTTP リダイレクトを構成し、構成を確認します。

```
set gslb service <serviceName> -sitePersistence <sitePersistence> -
    sitePrefix <string>
show gslb service <serviceName>
<!--NeedCopy-->
```


例:

```
set gslb service Service-GSLB-1 -sitePersistence HTTPRedirect -
    sitePrefix vserver-GSLB-1
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **HTTP** リダイレクトを構成するには

1. [トラフィック管理] > [GSLB] > [サービス] に移動し、設定する GSLB サービスをダブルクリックします。
2. 「サイトパーシスタンス」セクションをクリックし、「**HttpRedirect**」オプションを選択し、「サイトプレフィックス」テキストボックスにサイトプレフィックス（たとえば、vserver-GSLB-1）を入力します。

> 注
>
> サイト永続性が設定されておらず、ローカル GSLB サービスとして構成されている負荷分散仮想サーバーがダウンしている場合、HTTP リクエストは 302 リダイレクトを使用して他の正常な GSLB サイトにリダイレクトされます。

優先位置情報を構成することにより、静的近接度による動作を上書きする

August 15, 2023

ローカルの DNS (LDNS) サーバーまたはネットワークからのトラフィックを、静的近接方式がそのトラフィックに対して選択する GSLB サービス以外の GSLB サービスにトラフィックを転送したい場合があります。つまり、そのトラフィックに適した場所があるということです。静的近接方式を優先的に適用するには、以下の方法があります。

1. 優先ロケーションのリストで構成される DNS アクションを設定します。DNS アクションの構成の詳細については、「DNS アクションの設定」を参照してください。
2. 静的近接を上書きする LDNS サーバまたはネットワークから着信するトラフィックを識別するように DNS ポリシーを設定し、ポリシーでアクションを適用します。
3. ポリシーをグローバルリクエストのバインドポイントにバインドします。

DNS アクションでは、最大 8 つの優先場所のリストを設定できます。位置はドット修飾子表記で指定する必要があります。ドット修飾子表記とは、静的近接データベースにカスタム位置を追加するための表記法です。場所には、省略する修飾子のワイルドカードを含めることができます。ロケーションのドット付き修飾子の表記法については、静的近接データベースへのカスタムエントリの追加を参照してください。優先ロケーションを入力するときは、優先順位の降順で入力する必要があります。

ポリシーが TRUE と評価されると、NetScaler アプライアンスは優先場所を優先順に、GSLB サービスの場所と照合します。マッチには次の 2 種類があります。

- 優先する場所にあるワイルドカード以外の修飾子がすべて GSLB サービスの場所にある対応する修飾子と一致する場合、その一致は完全に一致したと見なされます。たとえば、GSLB サービスの場所が *.UK.*.* または Europe.uk.*.* であれば、希望する場所 *.UK.*.* と完全に一致します。
- ワイルドカード以外の修飾子のサブセットのみが一致する場合、その一致は部分一致と見なされます。たとえば、GSLB サービスの所在地が Europe.eg の場合、優先する場所である Europe.uk と部分的に一致します。

DNS ポリシーが
TRUE と評価されると、次のアルゴリズムを使用して GSLB サービスが選択されます。

1. アプライアンスは、優先度が最も高い優先場所を評価し、優先場所と GSLB サービスの場所との間で完全に一致するものが見つかるまで優先順位を下げます。

   完全に一致するものが見つかると、アプライアンスは対応する GSLB サービスが稼働しているかどうかを確認します。稼働している場合は、DNS レスポンスで GSLB サービスの IP アドレスを返します。完全に一致するものが複数見つかった場合（優先する場所で 1 つ以上のワイルドカードが使用されている場合など）、アプライアンスは対応する各 GSLB サービスの状態をチェックし、稼働中の GSLB サービスの負荷を分散します。

2. 優先する場所のいずれにも完全に一致するものが見つからない場合、アプライアンスは優先順位が最も高い優先場所に戻り、優先場所と GSLB サービスの場所が部分的に一致するまで優先順位を下げます。

   部分的に一致するものが見つかると、アプライアンスは対応する GSLB サービスが稼働しているかどうかを確認します。稼働している場合は、DNS レスポンスで GSLB サービスの IP アドレスを返します。部分一致が複数見つかった場合、アプライアンスは対応する各 GSLB サービスの状態をチェックし、稼働中の GSLB サービスの負荷を分散します。

3. 完全一致と部分一致のいずれも稼働していない場合、アプライアンスは利用可能な他のすべての GSLB サービスのロードバランシングを行います。

   このようにして、アプライアンスは DNS ポリシーに一致するトラフィックに対してある種のサイトアフィニティを実装します。

例

次の 8 つの GSLB サービスで構成される GSLB 構成を考えてみましょう。

- Asia.IN
- Asia.JPN
- Asia.hk
- Europe.UK
- Europe.RU
- Europe.EG
- Africa.SD
- Africa.ZMB

さらに、次の DNS アクションとポリシー設定を検討してください。


```
> add dns action prefLoc11 GslbPrefLoc -preferredLocList "Asia.HK" "
    Europe.UK"
 Done
> add dns policy dnsPolPrefLoc "CLIENT.IP.SRC.MATCHES_LOCATION("\*.ZMB
    .\*.*")" prefLoc11
 Done
<!--NeedCopy-->
```

アプライアンスがロケーション.*ZMB* からリクエストを受け取ったとき。.*、推奨ロケーションは次のように評価されます。

1. アプライアンスは、優先順位が最も高い優先場所である Asia.hk と完全に一致する場所の GSLB サービスを検索しようとします。その結果、Asia.hk の GSLB サービスがぴったり合っていることがわかりました。GSLB サービスが稼働している場合、GSLB サービスの IP アドレスをクライアントに送信します。
2. Asia.hk の GSLB サービスがダウンしている場合、アプライアンスは 2 番目に優先される場所である Europe.uk に完全に一致するものを見つけようとします。その結果、Europe.uk の GSLB サービスがぴったり合っていることがわかりました。GSLB サービスが稼働している場合、クライアントにサービスの IP アドレスを送信します。
3. Europe.uk の GSLB サービスがダウンしている場合は、優先度が最も高い優先場所である Asia.hk に戻り、部分一致するものを探します。Asia.HK については、Asia.in と Asia.JPN が部分的に一致していることが判明しました。対応する GSLB サービスのうち 1 つだけが稼働している場合、クライアントにサービスの IP アドレスを送信します。両方のロケーションが稼働している場合、2 つのサービスの負荷が分散されます。
4. Asia.hk の部分一致がすべてダウンした場合、アプライアンスは Europe.uk の部分一致を検索します。その結果、Europe.ru と Europe.eg は優先ロケーションと部分的に一致することがわかりました。対応する GSLB サービスのうち 1 つだけが稼働している場合、クライアントにサービスの IP アドレスを送信します。両方のロケーションが稼働している場合、2 つのサービスの負荷が分散されます。
5. Europe.uk のパーシャルマッチがすべてダウンした場合、アプライアンスは利用可能な他のすべての GSLB サービスのロードバランシングを行います。この例では、残りの 6 つの GSLB サービスが停止していることが判明したため、アプライアンスは Africa.sd と Africa.ZMB の負荷分散を行っています。

コンテンツスイッチを使用して **GSLB** サービスの選択を構成する

August 15, 2023

一般的な GSLB デプロイメントでは、GSLB 仮想サーバーにバインドされた GSLB サービスのセットの選択に優先順位を付けることはできますが、次のことはできません。

- 特定のドメインの GSLB 仮想サーバーにバインドされた GSLB サービスのサブセットからの GSLB サービスの選択を制限します。
- デプロイメント内の GSLB サービスのさまざまなサブセットにさまざまな負荷分散方法を適用します。

- GSLB サービスのサブセットにスピルオーバーポリシーを適用しても、GSLB サービスのサブセットのバックアップを作成することはできません。
- さまざまなコンテンツを提供するように GSLB サービスのサブセットを設定します。つまり、異なる GSLB サイトのサーバー間でコンテンツを切り替えることはできません。GSLB 構成では、サーバーには同じコンテンツが含まれていることを前提としています。
- 優先順位の異なるサブセット GSLB サービスを定義し、サブセット内のサービスがリクエストに適用される順序を指定します。

コンテンツスイッチング (CS) ポリシーを設定して、GSLB デプロイメントをカスタマイズできるようになりました。まず、GSLB サービスのセットを構成し、それを GSLB 仮想サーバーにバインドします。次に、ターゲットタイプ GSLB の CS 仮想サーバーを構成し、GSLB 仮想サーバーをターゲット仮想サーバーとして CS ポリシーとアクションを定義し、CS ポリシーを CS 仮想サーバーにバインドします。

> 重要
>
> - ターゲットタイプ GSLB の CS 仮想サーバーにバインドできるのは、DNS ベースの式を含む CS ポリシーだけです。
> - GLSB サービスが GSLB 仮想サーバーを介して CS 仮想サーバーにバインドされている場合、同じ GSLB サービスにバインドされた別の GSLB 仮想サーバーを CS 仮想サーバーにバインドすることはできません。

例
2 つの GSLB サイトを含む GLSB デプロイを考えてみましょう。各サイトでは、4 つの GSLB サービス (S-1、S-2、S-3、S-4) が GSLB 仮想サーバー VS-1 にバインドされています。ターゲットタイプ GSLB のコンテンツスイッチング（CS）仮想サーバーを構成し、VS-1 をターゲット仮想サーバーとして CS ポリシーとアクションを定義できます。これにより、英語のコンテンツのリクエストは S-1 と S-2 によってのみ処理され、ローカル言語でのコンテンツのリクエストは S-3 と S-4 によってのみ処理されます。

バックアップ仮想サーバーを VS-1 に設定し、S-2 をバックアップ仮想サーバーにバインドすることで、S-1 を優先させることができます。S-1 はクライアントのリクエストを処理します。S-1 が表すサーバーがダウンした場合、S-2 は要求を処理します。S-1 と S-2 の両方がダウンしている場合、クライアントは空の応答を受け取ります。

コンテンツスイッチングを使用して **GSLB** サービスセレクションを設定するには**:**

1. GSLB を設定します。手順については、「グローバルサーバー負荷分散の設定」を参照してください。
2. ターゲットタイプ GSLB のコンテンツスイッチング (CS) 仮想サーバーを構成します。詳細については、「コンテンツスイッチング仮想サーバーの作成」を参照してください。
3. コンテンツスイッチング（CS）ポリシーを設定します。詳細については、「コンテンツスイッチングポリシーの設定」を参照してください。
4. GSLB 仮想サーバーをターゲット仮想サーバーとして指定する CS アクションを設定します。詳細については、「コンテンツスイッチングアクションの設定」を参照してください。
5. CS ポリシーを CS 仮想サーバにバインドします。詳細については、コンテンツスイッチング仮想サーバへのポリシーのバインドを参照してください。
6. ドメインを GSLB 仮想サーバーではなく CS 仮想サーバーにバインドします。

構成例

次の設定例では、IP アドレスが 5.5.5.5 のクライアントから SERVICE_GSLB1 と SERVICE_GSLB2 にリクエストを送信します。SERVICE_GSLB1 は SERVICE_GSLB2 よりも優先度が高く、SERVICE_GSLB2 は SERVICE_GSLB1 がダウンしている場合にのみクライアント要求を処理します。SERVICE_GSLB1 と SERVICE_GSLB2 の両方がダウンしている場合、SERVICE_GSLB3 と SERVICE_GSLB4 は考慮されず、空白の応答がクライアントに送信されます。

```
add cs vs CSVSERVER_GSLB http  -targettype GSLB
Done
add gslb vs VSERVER_GSLB1 http
Done
add gslb vs VSERVER_GSLB2 http
Done
add gslb vs VSERVER_GSLB_BACKUP1 http
Done
set gslb vs VSERVER_GSLB1 -backupvserver VSERVER_GSLB_BACKUP1
Done
add gslb service SERVICE_GSLB1 1.1.1.1 HTTP 80 -sitename site1
Done
add gslb service SERVICE_GSLB2 1.1.1.2 HTTP 80 -sitename site1
Done
add gslb service SERVICE_GSLB3 1.1.1.3 HTTP 80 -sitename site2
Done
add gslb service SERVICE_GSLB4 1.1.1.4 HTTP 80 -sitename site2
Done
bind gslb vs VSERVER_GSLB1 -servicename SERVICE_GSLB1
Done
bind gslb vs VSERVER_GSLB_BACKUP1 -servicename SERVICE_GSLB2
Done
bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB3
Done
bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB4
Done
add cs action a1 -targetvserver VSERVER_GSLB1
Done
add cs policy p1 -rule "CLIENT.IP.SRC.EQ(5.5.5.5)" -action a1
Done
bind cs vs CSVSERVER_GSLB -domainName www.abc.com
Done
bind cs vs CSVSERVER_GSLB -policyname p1 -priority 1
Done
add cs action a2 -targetvserver VSERVER_GSLB2
Done
add cs policy p2 -rule "CLIENT.IP.SRC.EQ(6.6.6.6)" -action a2
Done
bind cs vs CSVSERVER_GSLB -policyname p2 -priority 2
Done
<!--NeedCopy-->
```


ターゲット仮想サーバーエクスプレッションを **GSLB** コンテンツスイッチングアクションに関連付ける

これで、ターゲット仮想サーバー式を GSLB コンテンツスイッチングアクションに関連付けることができます。これにより、GSLB コンテンツスイッチング仮想サーバーは、DNS リクエストの処理中にポリシー表現を使用してターゲットの GSLB 仮想サーバー名を作成できます。

CLI を使用して式を指定するコンテンツスイッチングアクションを設定するには

コマンドプロンプトで次のコマンドを入力して、HTTP コールアウト応答を取得するようにコンテンツスイッチングアクションを設定します。

```
1 add cs action <name> -targetVserverExpr <expression>
2 <!--NeedCopy-->
```

例:

```
1 add cs action csact_GSLB_VServer -targetVserverExpr "SYS.HTTP_CALLOUT(
     GSLB_Method_API)"
2 <!--NeedCopy-->
```

GUI を使用して式を指定するコンテンツスイッチングアクションを構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [アクション] に移動します。
2. コンテンツスイッチングアクションを設定し、ターゲットの負荷分散仮想サーバーの名前を動的に計算する式を指定します。

NAPTR レコードの **DNS** クエリを **GSLB** で構成する

August 15, 2023

一般的なグローバルサーバー負荷分散（GSLB）展開では、NetScaler アプライアンスは A/AAAA レコードの DNS クエリを受信し、構成された負荷分散方法に従って最も適切な GSLB サービスを選択し、DNS クエリへの応答としてサービスの IP アドレスを返します。これで、NAPTR レコードの DNS クエリを受信し、ドメインに設定されたサービスのリストで応答するようにアプライアンスを設定できます。アプライアンスはサービスの状態も監視し、応答には稼働中のサービスのみのリストが表示されます。

例**:**

通信事業者の環境では、モバイル管理エンティティ（MME）などのクライアントから NAPTR レコードを含む DNS クエリを受信するように NetScaler アプライアンスを構成できます。これらのクライアントは DNS リゾルバーの役割を果たし、ドメイン名で提供されるすべてのサービスを検出します。アプライアンスは、稼働しているすべてのサ

ービスの NAPTR レコードでクエリに応答します。MME はこの NAPTR 応答を使用して S-NAPTR プロシージャを実行し、提供されるサービス、コロケーション、トポロジカルな近さなどに基づいてノードを選択できます。

複数のノードが選択の対象となる場合、MME は NetScaler アプライアンスの NAPTR レコードの優先フィールドを使用してノードを決定できます。

NAPTR レコードフォーマット

NetScaler アプライアンスは、NAPTR レコードを使用して DNS クエリに応答する際、GSLB サービスごとに応答 NAPTR レコードを作成します。

次の表は、NAPTR レコード内のファイルのリストです。

フィールド	
ドメイン	GSLB ドメイン
TTL	NAPTR レコードをキャッシュできる時間。
クラス	レコードのクラス。デフォルトでは、この値は IN に設定されています。
種類	DNS レコードタイプ。
Order	NAPTR レコードを処理する必要がある順序を指定します。GSLB サービスで順序を指定できます。それ以外の場合は 1 に設定されます。
プリファレンス	「順序」値が等しい NAPTR レコードを処理する順序を指定します。小さい数値は大きい数値の前に処理されます。GSLB サービスで順序が指定されていない場合は、1 に設定されます。
フラグ	レコード内のフィールドの書き換えと解釈の側面を制御します。NetScaler アプライアンスはこの値を A に設定します。
サービス	利用可能なサービスを指定します。
正規表現	正規表現はサポートされていないため、この値は NULL に設定されます。
置換	サービスをホストするノードのドメイン名。

設定手順

GSLB の設定手順の詳細については、「グローバルサーバー負荷分散 (GSLB) の設定」を参照してください。次の作業を行ってください。

- GSLB 仮想サーバーを追加する際には、以下のパラメーターを設定します。
  - serviceType: ANY
  - dnsRecordType: NAPTR
  - lbMethod: CUSTOMLOAD

例:

```
add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD
<!--NeedCopy-->
```

- GSLB サイトを追加する際には、*NAPTRReplacementSuffix* パラメーターを NAPTR レコードに埋め込むドメイン名に設定します。

例:

```
add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com
<!--NeedCopy-->
```

- GSLB サービスを追加する際には、次のパラメータを設定します。
  - ナプトルの交換
  - naptrOrder
  - naptrServices
  - naptrDomainTTL
  - naptrPreference

構成例

```
add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD

Done

add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com

Done

add gslb service sgw1 3.3.3.13 ANY *  -siteName site1 -naptrreplacement
    sgw1.site1. -naptrOrder 2 -naptrServices x-3gpp-sgw:x-s5-gtp -
   naptrDomainTTL 20 -naptrPreference   200

Done

add gslb service sgw2 3.3.3.11 ANY *  -siteName site1 -naptrreplacement
    sgw2.site1. -naptrOrder 5 -naptrServices x-3gpp-sgw:x-s5-gtp  -
   naptrDomainTTL 20  naptrPreference  100

Done
```



```
16
17 add gslb service sgw3 3.3.3.12 ANY *  -siteName site2 -naptrreplacement
       sgw3.site1. -naptrOrder 10 -naptrServices x-3gpp-sgw:x-s5-gtp  -
      naptrDomainTTL 20 naptrPreference   300
18
19 bind gslb vserver gslb_vs -serviceName sgw1
20
21 Done
22
23 bind gslb vserver gslb_vs -serviceName sgw2
24
25 Done
26
27 bind gslb vserver gslb_vs -serviceName sgw3
28
29 Done
30
31 bind gslb service sgw1 -monitorName ping
32
33 Done
34
35 bind gslb service sgw2 -monitorName ping
36
37 Done
38
39 bind gslb service sgw3 -monitorName ping
40
41 Done
42
43 bind gslb vserver gslb_vs -domainName gslb.com -TTL 5
44
45 Done
46 <!--NeedCopy-->
```

> 注
>
> NAPTR レコードを含む DNS クエリは、親子構成ではサポートされていません。

ワイルドカードドメインを **GSLB** で構成する

August 15, 2023

ワイルドカード DNS ドメインを GSLB 仮想サーバーにバインドできます。ワイルドカードドメインの背後にあるアプリケーションにアクセスするユーザーは、それらのアプリケーションをホストする最適なデータセンターにルーティングされます。ワイルドカードドメインは、存在しないドメインおよびサブドメインに対する要求を処理します。ワイルドカードドメインの詳細については、「ワイルドカード DNS ドメインのサポート」を参照してください。DNS ゾーンの詳細については、「DNS ゾーンを構成する」を参照してください。

ワイルドカードドメインの GSLB を設定するには、まず基本的な GSLB 設定を設定する必要があります。基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

CLI を使用してワイルドカードドメインの GSLB 設定を設定するには

ワイルドカードドメインの GSLB 設定を構成するには、次の手順を実行します。

1. GSLB サイトを作成します。

```
add gslb site site1 10.0.1.10
add gslb site site2 20.0.1.10
<!--NeedCopy-->
```

2. GSLB セットアップに参加している各サイトの GSLB サービスを追加します。

```
add gslb service svc1 -sitename site1 10.0.1.10  http 80
add gslb service svc2 -sitename site1 10.0.1.10  http 80
add gslb service svc3 -sitename site2 20.0.1.10  http 80
add gslb service svc4 -sitename site2 20.0.1.10  http 80
<!--NeedCopy-->
```

3. GSLB セットアップで使用されているサービスを参照する GSLB 仮想サーバーを追加します。

```
add gslb vserver gslb_vs http
<!--NeedCopy-->
```

4. DNS クエリをリッスンする ADNS サービスを追加します。

```
add service adns_udp 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

5. GSLB サービスを GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gslb_vs -service svc1
bind gslb vserver gslb_vs -service svc2
bind gslb vserver gslb_vs -service svc3
bind gslb vserver gslb_vs -service svc4
<!--NeedCopy-->
```

6. ゾーンを作成します。

```
add dns soaRec test.com -originServer n1.test.com -contact n1.test
   .com
add dns nsrec test.com n1.test.com
add dns nsrec test.com n2.test.com
add dns zone test.com -proxymode no
<!--NeedCopy-->
```

7. ドメイン名を GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gslb_vs -domainName *.test.com
```



```
<!--NeedCopy-->
```

グローバルサーバ負荷分散には **EDNS0** クライアントサブネットオプションを使用する

August 15, 2023

EDNS クライアントサブネット (ECS) は、クライアントサブネットの詳細を提供するドメインネームサーバー (DNS) ヘッダー拡張です。これらの詳細を使用して、DNS リゾルバーの場所ではなくクライアントネットワークの場所を使用してクライアントのトポロジカルな近さを判断することで、NetScaler Global Server 負荷分散（GSLB）の精度を向上させることができます。

> 注
>
> NetScaler EDNS0 のみをサポートします。

重要:

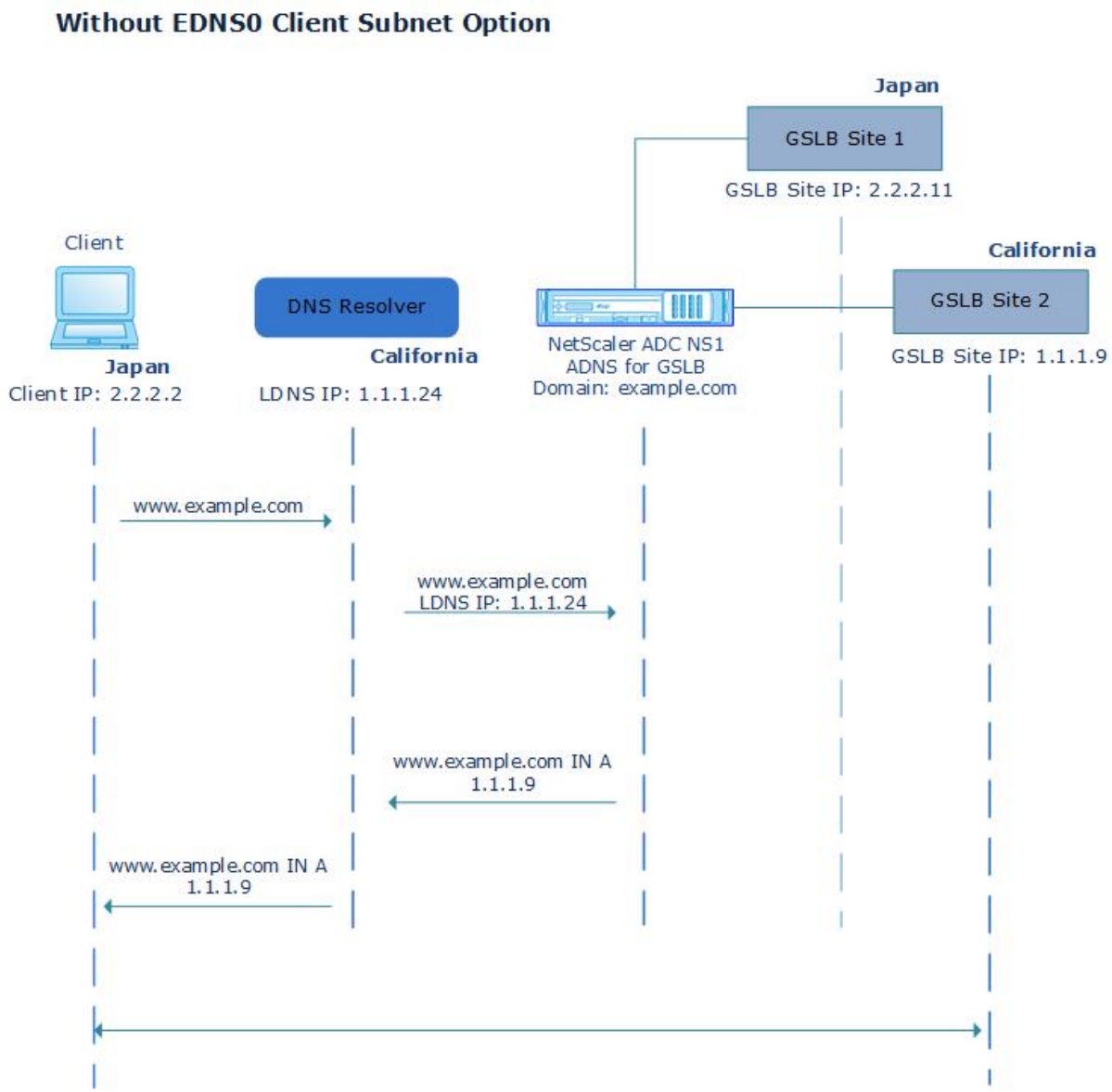
> 導入環境のローカルドメインネームサーバー（LDNS）が EDNS0 クライアントサブネットをサポートしていることを確認してください。これにより、受信 DNS クエリには EDNS0 クライアントサブネットオプションが含まれ、NetScaler アプライアンスは DNS クエリの処理中に ECS アドレスを使用します。

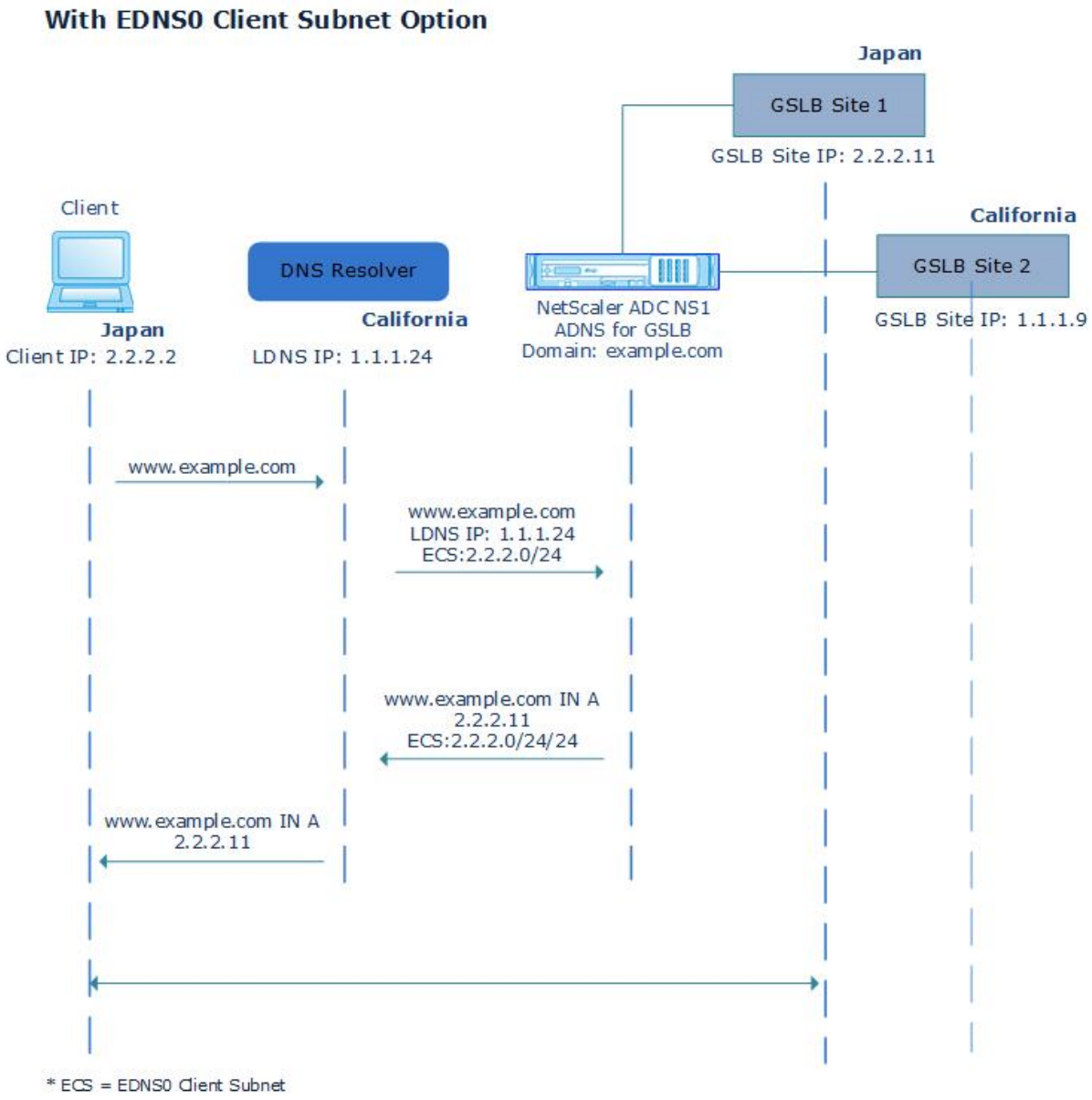
NetScaler アプライアンスは、静的近接性や動的ラウンドトリップ時間（RTT）などの近接ベースの負荷分散方法を使用する場合は、LDNS IP アドレスを使用してクライアントのトポロジカルな近接性を判断し、GSLB を実行します。これは典型的な GSLB デプロイメントで起こります。ただし、Google DNS や OpenDNS などの集中型 DNS リゾルバーが導入に関与している場合、NetScaler アプライアンスは集中型 DNS リゾルバーに近いデータセンターに DNS 要求を送信します。データセンターはクライアントの近くにはない可能性があります。たとえば、静的近接負荷分散方式を使用する一般的な NetScaler GSLB 展開では、日本からのエンドユーザーリクエストは日本のデータセンターに送信され、カリフォルニアからのエンドユーザーリクエストはカリフォルニアのデータセンターに送信されます。ただし、一元化された DNS リゾルバーが関与している場合、NetScaler アプライアンスは日本からカリフォルニアのデータセンターにリクエストを送信する可能性があります。

ECS オプションは、GSLB ドメイン用の権限のある DNS（ADNS）サーバーとして構成された NetScaler アプライアンスを含む展開環境で使用できます。負荷分散方法として静的近接性を使用する場合、EDNS ヘッダーに LDNS IP アドレスの代わりに IP サブネットを使用できます。これは、クライアントの地理的な近さを判断するのに役立ちます。プロキシモードの展開では、NetScaler アプライアンスは ECS 対応の DNS クエリをそのままバックエンドサーバーに転送します。アプライアンスは ECS 対応 DNS 応答をキャッシュしません。

> 注
>
> ECS オプションは、GSLB 以外のドメインの ADNS モード、リゾルバーモード、フォワーダーモードなど、他

> のすべてのデプロイモードには適用されません。前述のモードでは、ECS オプションは NetScaler アプライアンスによって無視されます。また、デフォルトでは、GSLB デプロイの ECS は無効になっています。





コマンドラインインターフェイスを使用して **EDNS0** クライアントサブネットオプションを有効にするには**:**

コマンドプロンプトで入力します。

```
set gslb vserver <vserver_name> **-ECS ENABLED

set gslb vserver vserver-GSLB-1 -ECS ENABLED
<!--NeedCopy-->
```

住所検証

GSLB 仮想サーバーを構成して、DNS クエリの EDNS0 クライアントサブネット（ECS）オプションによって返されるアドレスがプライベートまたはルーティング不可能な IP アドレスではないことを確認できます。アドレス検証を

有効にすると、NetScaler アプライアンスは、次の表に記載されている DNS クエリ内の ECS アドレスを無視し、代わりに LDNS IP アドレスを使用してグローバルサーバーの負荷分散を行います。

> 注
>
> デフォルトでは、住所の検証は無効になっています。

住所の種類	アドレス	説明
IPV4	10.0.0.0/8	私的使用のため
	172.16.0.0/12	私的使用のため
	192.168.0.0/16	私的使用のため
	0.0.0.0/8	ネットワーク上のホストを指します
	100.64.0.0/10	共有アドレス空間
	127.0.0.0/8	ループバックアドレス
	169.254.0.0/16	RFC 3927 で定義されているリンクローカル IPv4 アドレス
	192.0.0.0/24	IETF プロトコルの割り当てに使用されます。プライベートスペース 192.168.0.0/16 も含まれます
	192.0.2.0/24	文書化の目的に使用されます
	192.88.99.0/24	6to4 リレーエニーキャストに使用
	198.18.0.0/15	デバイスベンチマークテストに使用
	198.51.100.0/24	文書化の目的に使用されます
	203.0.113.0/24	文書化の目的に使用されます
	240.0.0.0/4	予約どおりに使用
	255.255.255.255/32	放送に使用
IPv6	::1/128	ループバックアドレス
	::/128	未指定の住所
	::ffff:0:0/96	IPv4 マップアドレス
	100::/64	破棄専用アドレスブロック
	2001::/23	IETF プロトコルの割り当てに使用
	2001::/32	TEREDO

住所の種類	アドレス	説明
	2001:2::/48	ベンチマークに使用
	2001:db8::/32	文書化の目的に使用されます
	2001:10::/28	ORCHID
	2002::/16	6to4 リレーエニーキャストに使用
	fc00::/7	ユニーク・ローカル
	fe80::/10	リンクローカルユニキャストアドレス

コマンドラインインターフェイスを使用して住所検証を有効にするには

コマンドプロンプトで入力します。

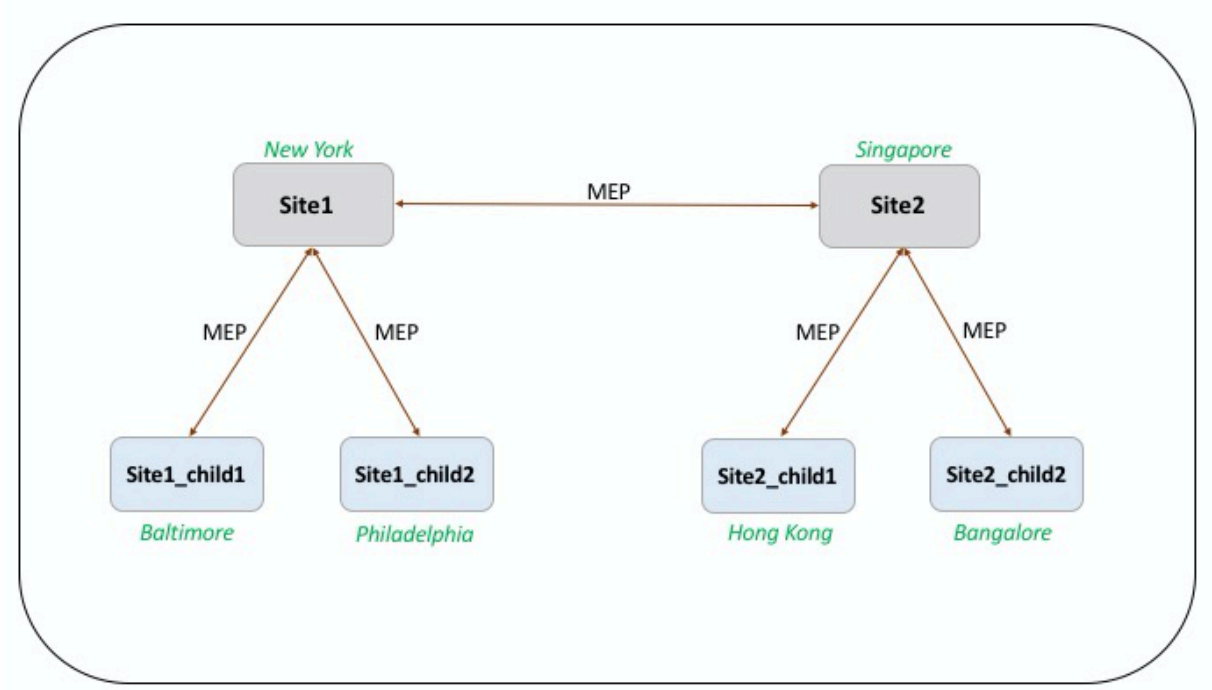
```
set gslb vserver <vserver_name> -ecsAddrValidation ENABLED

set gslb vserver vserver-GSLB-1 -ecsAddrValidation ENABLED
<!--NeedCopy-->
```

メトリック交換プロトコルを使用した完全な親子構成の例

August 15, 2023

GSLB サイトがグローバルに分散している次の親子トポロジを考えてみましょう。

- サイト 1 とサイト 2 は親サイトです。
- サイト 1_ チャイルド 1 とサイト 1_ チャイルド 2 はサイト 1 の子サイトです。
- サイト 2_ チャイルド 1 とサイト 2_ チャイルド 2 はサイト 2 の子サイトです。



次のコマンドは、親子トポロジの完全な構成を示しています。

site1

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

add gslb service site1_child1_http_gsvc1 10.102.82.132 HTTP 80 -
    publicIP 10.102.82.132 -publicPort 80 -maxClient 0 -siteName
    site1_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

```



```
15 add gslb service site1_child2_http_gsvc1 10.102.82.68 HTTP 80 -publicIP
       10.102.82.68 -publicPort 80 -maxClient 0 -siteName site1_child2 -
      cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
16
17 add gslb service site2_child1_http_gsvc1 10.106.24.134 HTTP 80 -
      publicIP 10.106.24.134 -publicPort 80 -maxClient 0 -siteName
      site2_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
18
19 add gslb service site2_child2_http_gsvc1 10.106.24.68 HTTP 80 -publicIP
       10.106.24.68 -publicPort 80 -maxClient 0 -siteName site2_child2 -
      cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
20
21 add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0 -
      appflowLog DISABLED
22
23 bind gslb vserver gv1 -serviceName site1_child1_http_gsvc1
24
25 bind gslb vserver gv1 -serviceName site1_child2_http_gsvc1
26
27 bind gslb vserver gv1 -serviceName site2_child2_http_gsvc1
28
29 bind gslb vserver gv1 -serviceName site2_child1_http_gsvc1
30
31 bind gslb vserver gv1 -domainName www.gslb.com -TTL 5
32 <!--NeedCopy-->
```

site1_child1

```
1 add gslb site site1 10.102.82.164 -publicIP 10.102.82.164
2
3 add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
      nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
      site1
4 <!--NeedCopy-->
```

ロードバランシング設定には、次のコマンドを追加できます。

```
1 add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
       -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
      svrTimeout 360 -CKA NO -TCPB NO -CMP NO
2
3 add lb vserver lb1 HTTP 10.102.82.132 80 -persistenceType NONE -
      cltTimeout 180
4
5 bind lb vserver lb1 svc1
6 <!--NeedCopy-->
```

site1_child2


```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

You can add the following commands for load balancing configuration:

add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.102.82.68 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site2

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

add gslb service site1_child1_http_gsvc1 10.102.82.132 HTTP 80 -
    publicIP 10.102.82.132 -publicPort 80 -maxClient 0 -siteName
    site1_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site1_child2_http_gsvc1 10.102.82.68 HTTP 80 -publicIP
     10.102.82.68 -publicPort 80 -maxClient 0 -siteName site1_child2 -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site2_child1_http_gsvc1 10.106.24.134 HTTP 80 -
    publicIP 10.106.24.134 -publicPort 80 -maxClient 0 -siteName
    site2_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
```



```
18
19 add gslb service site2_child2_http_gsvc1 10.106.24.68 HTTP 80 -publicIP
       10.106.24.68 -publicPort 80 -maxClient 0 -siteName site2_child2 -
       cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
20
21 add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0 -
       appflowLog DISABLED
22
23 bind gslb vserver gv1 -serviceName site1_child1_http_gsvc1
24
25 bind gslb vserver gv1 -serviceName site1_child2_http_gsvc1
26
27 bind gslb vserver gv1 -serviceName site2_child2_http_gsvc1
28
29 bind gslb vserver gv1 -serviceName site2_child1_http_gsvc1
30
31 bind gslb vserver gv1 -domainName www.gslb.com -TTL 5
32 <!--NeedCopy-->
```

site2_child1

```
1 add gslb site site2 10.106.24.164 -publicIP 10.106.24.164
2
3 add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
      nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
      site2
4 <!--NeedCopy-->
```

You can add the following commands for load balancing configuration:

```
1 add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
       -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
      svrTimeout 360 -CKA NO -TCPB NO -CMP NO
2
3 add lb vserver lb1 HTTP 10.106.24.134 80 -persistenceType NONE -
      cltTimeout 180
4
5 bind lb vserver lb1 svc1
6 <!--NeedCopy-->
```

site2_child2

```
1 add gslb site site2 10.106.24.164 -publicIP 10.106.24.164
2
3 add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
      nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
      site2
4 <!--NeedCopy-->
```


You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
    -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
   svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.106.24.68 80 -persistenceType NONE -
   cltTimeout 180

bind lb vserver lb1 svc1
\`\`\`
<!--NeedCopy-->
```

リンク負荷分散

August 15, 2023

リンク負荷分散 (LLB) は、さまざまなサービスプロバイダーが提供する複数のインターネット接続にわたってアウトバウンドトラフィックのバランスをとります。LLB を使用すると、NetScaler アプライアンスはトラフィックを監視および制御できるため、パケットは可能な限り最良のリンクを介してシームレスに送信されます。サービスがサーバーを表し、LLB の場合はサービスがルーターまたはネクストホップを表すサーバーロードバランシングとは異なります。リンクは、NetScaler アプライアンスとルーター間の接続です。

リンクロードバランシングを設定するには、多くのユーザーはまず基本設定をデフォルト設定で構成することから始めます。基本設定には、サービス、仮想サーバー、モニター、ルート、LLB メソッド、およびパーシスタンス (オプション) が含まれます。基本設定が動作可能になったら、環境に合わせてカスタマイズできます。

LLB に適用できるロードバランシング方式には、ラウンドロビン、宛先 IP ハッシュ、最小帯域幅、最小パケットがあります。オプションで、特定のリンクで接続を維持するパーシステンスを設定できます。使用可能なパーシステンスタイプは、送信元 IP アドレスベース、宛先 IP アドレスベース、および送信元 IP アドレスと宛先 IP アドレスベースです。PING がデフォルトモニタですが、トランスペアレントモニタを設定することを推奨します。

リバース NAT（RNAT）とバックアップリンクを設定することで、設定をカスタマイズできます。

基本的な **LLB** セットアップの構成

August 15, 2023

LLB を設定するには、まず、インターネットサービスプロバイダー (ISP) に対して各ルーターに対応するサービスを作成します。PING モニターはデフォルトで各サービスにバインドされます。透明モニターのバインドは任意ですが、推奨されます。次に、仮想サーバーを作成し、サービスを仮想サーバーにバインドし、仮想サーバーのルートを構成します。このルートは、仮想サーバーをサービスによって表される物理ルーターへのゲートウェイとして識別します。

仮想サーバーは、指定した負荷分散方法を使用してルーターを選択します。オプションで、特定のセッションのすべてのトラフィックが特定のリンクを介して送信されるようにパーシステンスを設定できます。

基本的な LLB 設定を設定するには、次の操作を行います。

- サービスを構成する
- LLB 仮想サーバーの設定とサービスのバインディング
- LLB 方式とパーシスタンスの設定
- LLB ルートの設定
- 透明モニターの作成とバインディング

サービスを構成する

デフォルトモニタ（PING）は、サービスの作成時にサービスタイプの ANY に自動的にバインドされますが、トランスペアレントモニタの作成とバインドの説明に従って、デフォルトモニタをトランスペアレントモニタに置き換えることができます。

コマンドラインインターフェイスを使用してサービスを作成するには

コマンドプロンプトで入力します。

```
add service <name> <IP> <serviceType> <port>

show service <name>
<!--NeedCopy-->
```

例:

```
add service ISP1R_svc_any 10.10.10.254 any *
show service ISP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: DOWN
        Last state change was at Tue Aug 31 04:31:13 2010
        Time since last state change: 2 days, 05:34:18.600
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
```



```
19          SP: OFF
20          Down state flush: ENABLED
21
22  1)      Monitor Name: ping
23                  State: UP       Weight: 1
24                  Probes: 244705  Failed [Total: 0 Current: 0]
25                  Last response: Success - ICMP echo reply received.
26                  Response Time: 1.322 millisec
27   Done
28  <!--NeedCopy-->
```

構成ユーティリティを使用してサービスを作成するには

[トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを作成します。

構成ユーティリティを使用してサービスを作成するには

1. ［Traffic Management］ > ［Load Balancing］ > ［Services］の順に移動します。
2. 詳細ペインで、［Add］をクリックします。
3. 「サービスの作成」ダイアログで、次のパラメータの値を指定します。
   - サービス名 *—名前
   - サーバ IP
   - プロトコル *-サービスタイプ (ドロップダウンリストから ANY を選択)
   - ポート *-ポート

必須パラメーター

1. ［作成］をクリックします。
2. 手順 2 ～4 を繰り返して、別のサービスを作成します。
3. ［閉じる］をクリックします。
4. サービスペインで、先ほど設定したサービスを選択し、画面下部に表示される設定が正しいことを確認します。

LLB 仮想サーバーの設定とサービスのバインド

サービスを作成したら、仮想サーバーを作成し、サービスを仮想サーバーにバインドします。LLB では、最小接続のデフォルトの LB 方式はサポートされていません。LB 方式の変更については、LLB 方式と永続性の設定を参照してください。

リンク負荷分散仮想サーバーを作成し、コマンドラインインターフェイスを使用してサービスをバインドするには

コマンドプロンプトで入力します。

```
add lb vserver <name> <serviceType>

bind lb vserver < name> <serviceName>

show lb vserver < name>
<!--NeedCopy-->
```

例:

```
add lb vserver LLB-vip any
bind lb vserver LLB-vip ISP1R_svc_any
sh lb vserver LLB-vip
        LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Thu Sep  2 10:51:32 2010
        Time since last state change: 0 days, 17:51:46.770
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED

1) ISP1R_svc_any (10.10.10.254: *) - ANY State: DOWN    Weight: 1
 Done
<!--NeedCopy-->
```

リンク負荷分散仮想サーバーを作成し、構成ユーティリティを使用してサービスをバインドするには

1. トラフィック管理に移動します > 負荷分散 > 仮想サーバー、およびリンク負荷分散用の仮想サーバーを作成します。「プロトコル」フィールドに「ANY」を指定します。
2. 「**IP** アドレスタイプ」ドロップダウンリストで、目的のオプションを選択します。直接アクセスできない仮想サーバーを作成するには、「アドレス指定不可」を選択します。
3. 「サービス」タブの「アクティブ」列で、仮想サーバーにバインドするサービスのチェックボックスを選択します。

LLB 方式とパーシスタンスの設定

デフォルトでは、NetScaler アプライアンスは最も少ない接続方法を使用して各クライアント要求をリダイレクトするサービスを選択しますが、LLB メソッドをサポートされている方法のいずれかに設定する必要があります。パーシ

ステンスを設定して、同じクライアントからのさまざまな送信が同じサーバーに転送されるようにすることもできます。

コマンドラインインターフェイスを使用して **LLB** 方式やパーシスタンスを設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
set lb vserver <name> -lbMethod <lbMethod> -persistenceType <
   persistenceType>

show lb vserver <name>
<!--NeedCopy-->
```

例:

```
set lb vserver LLB-vip -lbmethod ROUNDROBIN -persistencetype SOURCEIP

show lb vserver LLB-vip
        LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 00:52:21.200
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: SOURCEIP
     Persistence Mask: 255.255.255.255       Persistence v6MaskLength:
         128   Persistence Timeout: 2 min
        Connection Failover: DISABLED
<!--NeedCopy-->
```

設定ユーティリティを使用してリンクロードバランシング方式やパーシステンスを設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、負荷分散方法や永続性設定を構成する仮想サーバーを選択します。
2. 「詳細設定」セクションで、「方法」を選択し、負荷分散方法を設定します。
3. 「詳細設定」セクションで、「持続性」を選択し、持続性パラメータを設定します。

LLB ルートの設定

IPv4 または IPv6 サービス、仮想サーバー、LLB メソッド、およびパーシステンスを設定したら、LLB 仮想サーバーをゲートウェイとして指定するネットワークの IPv4 または IPv6 LLB ルートを構成します。ルートは、負荷分散さ

れたリンクの集まりです。要求は、すべてのアウトバウンドトラフィックのゲートウェイとして機能する LLB 仮想サーバの IP アドレスに送信され、設定された LLB 方式に基づいてルータを選択します。

コマンドラインインターフェイスを使用して **IPv4 LLB** ルートを設定するには

コマンドプロンプトで入力します。

```
add lb route <network> <netmask> <gatewayName>

show lb route [<network> <netmask>]
<!--NeedCopy-->
```

例:

```
add lb route 0.0.0.0 0.0.0.0 LLB-vip
show lb route 0.0.0.0 0.0.0.0
       Network          Netmask        Gateway/VIP             Flags
     -----------      -------------   --------------          --------
1)     0.0.0.0          0.0.0.0        LLB-vip               UP
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **IPv6 LLB** ルートを設定するには

コマンドプロンプトで入力します。

```
add lb route6 <network> <gatewayName>

show lb route6
<!--NeedCopy-->
```

例**:**

```
add lb route6 ::/0 llb6_vs show lb route6 Network VIP Flags -----------
     ---------- -------- 1) ::/0 llb6_vs UP
<!--NeedCopy-->
```

設定ユーティリティを使用して **LLB** ルートを設定するには

[システム] > [ネットワーク] > [ルート] に移動し、**LLB** を選択し、LLB ルートを設定します。

注**:** IPV6 ルートを設定するには LLBV6 を選択してください。

設定ユーティリティを使用して **LLB** ルートを設定するには

1. [システム] > [ネットワーク] > [ルート] に移動します。

2. 詳細ペインで、次のいずれかを選択します。

    - LLB をクリックして IPv4 ルートを設定します。
    - LLBV6 をクリックして IPv4 ルートを設定します。

3.「LB ルートの作成」または「LB IPV6 ルートの作成」ダイアログボックスで、次のパラメータを設定します。

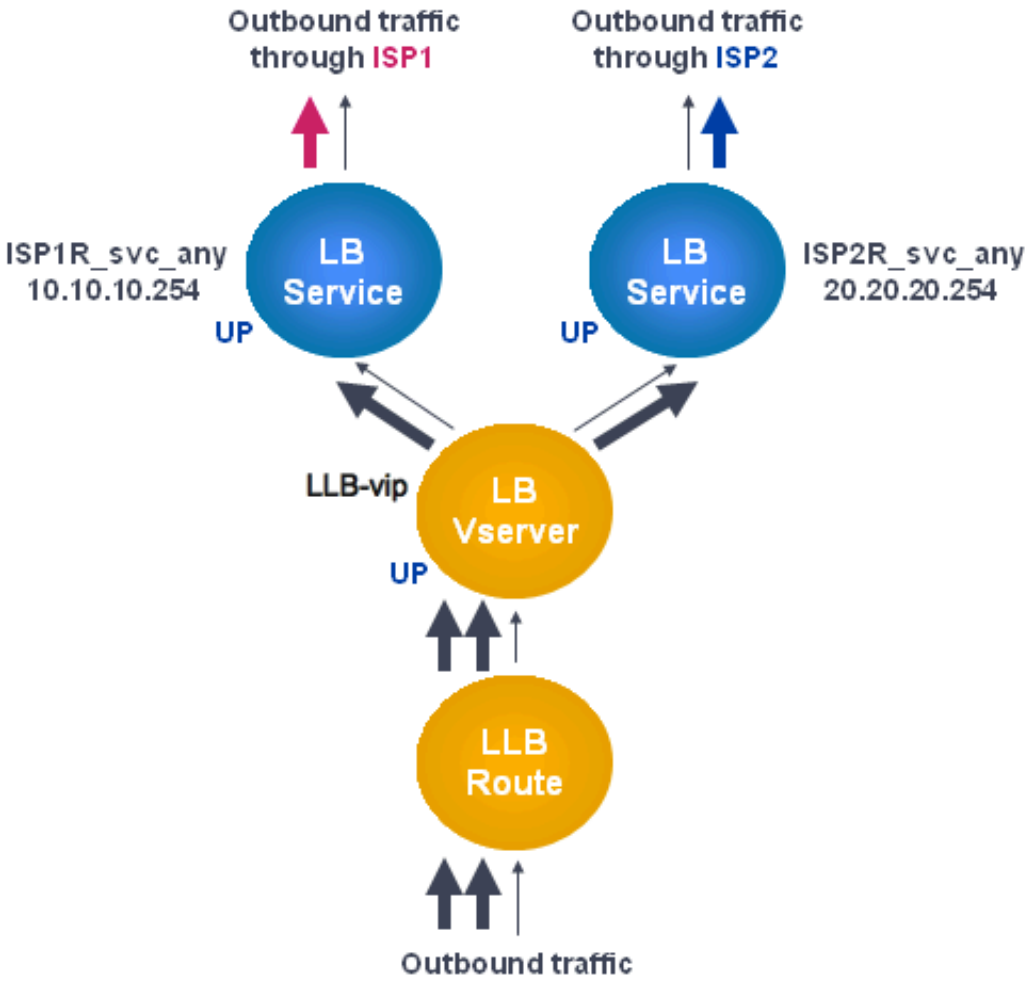
    - ネットワーク *
    - ネットマスク * —IPV4 ルートに必要です。
    - ゲートウェイ名 * —ゲートウェイ名

      * 必須パラメータ

4. [Create] をクリックしてから、[Close] をクリックします。作成したルートは、[ルート] ペインの [LLB] または [LLB6] タブに表示されます。

次の図は、基本的な LLB 設定を示しています。2 つのリンク（ISP）のそれぞれにサービスが設定され、PING モニタはデフォルトでこれらのサービスにバインドされます。リンクは、設定された LLB 方式に基づいて選択されます。

図 1: 基本的な LLB セットアップ



> 注
>
> インターネットサービスプロバイダが IPv6 アドレスを提供している場合は、IPv4 サービスを上の図の IPv6 サービスに置き換えてください。

透明モニターの作成とバインディング

トランスペアレントモニタを作成して、ルータなどのアップストリームデバイスの状態を監視します。その後、透過モニターをサービスにバインドできます。デフォルトの PING モニターは、NetScaler ADC アプライアンスと上流デバイス間の接続のみを監視します。透過モニタは、アプライアンスからモニタで指定された宛先 IP アドレスを所有するデバイスまでのパスに存在するすべてのデバイスを監視します。トランスペアレントモニタが設定されておらず、ルータのステータスが UP でも、そのルータのネクストホップデバイスの 1 つがダウンしている場合、アプライアンスはルータを含めてロードバランシングを実行し、パケットをルータに転送します。ただし、ネクストホップデバイスの 1 つがダウンしているため、パケットは最終的な宛先に配信されません。トランスペアレントモニターをバインドすることで、（ルーターを含む）デバイスのいずれかがダウンした場合、サービスはダウンとしてマークされ、アプ

ライアンスがリンクロードバランシングを実行するときにルーターは含まれません。

コマンドラインインターフェイスを使用して透明モニターを作成するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <type> -destIP <ip_addr|*> -transparent
    YES

show lb monitor [<monitorName>]
<!--NeedCopy-->
```

例:

```
add lb monitor monitor-1 PING -destIP 10.10.10.11 -transparent YES
> show lb monitor monitor-1
1)   Name.......: monitor-1  Type......:      PING   State....:
    ENABLED
Standard parameters:
  Interval.........:            5 sec   Retries..........:
                    3
  Response timeout.:            2 sec   Down time........:
     30 sec
  Reverse..........:               NO   Transparent.......:
                  YES
  Secure...........:               NO   LRTM.............:
     ENABLED
  Action...........:   Not applicable   Deviation........:
     0 sec
  Destination IP...:      10.10.10.11
  Destination port.:    Bound service
  Iptunnel.........:               NO
  TOS..............:               NO   TOS ID...........:
                    0
  SNMP Alert Retries:               0     Success Retries..:
                    1
  Failure Retries..:               0
<!--NeedCopy-->
```

構成ユーティリティを使用して透明モニターを作成するには

[トラフィック管理] > [負荷分散] > [モニター] に移動し、トランスペアレントモニターを設定します。

構成ユーティリティを使用して透明モニターを作成するには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. 「モニター」ペインで、「追加」をクリックします。

3. 「モニターの作成」ダイアログ・ボックスで、次のパラメーターを設定します。

   - 名前 *
   - タイプ *
   - 接続先 IP
   - 透明

   * 必須パラメータ

4. [Create] をクリックしてから、[Close] をクリックします。
5. モニターペインで、先ほど設定したモニターを選択し、詳細ペインに表示されている設定が正しいことを確認します。

構成ユーティリティを使用してモニターをサービスにバインドするには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 「モニター」タブの「使用可能」で、サービスにバインドするモニターを選択し、「追加」をクリックします。

コマンドラインインターフェイスを使用してモニターをサービスにバインドするには

コマンドプロンプトで入力します。

```
bind lb monitor <monitorName> <serviceName>

show service <name>
<!--NeedCopy-->
```

例:

```
bind lb monitor monitor-HTTP-1 ISP1R_svc_any
 Done
> show service ISP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: UP
        Last state change was at Thu Sep  2 10:51:07 2010
        Time since last state change: 0 days, 18:41:55.130
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
```



```
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: UP  Weight: 1
                Probes: 1256       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.322 millisec
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用してモニターをサービスにバインドするには

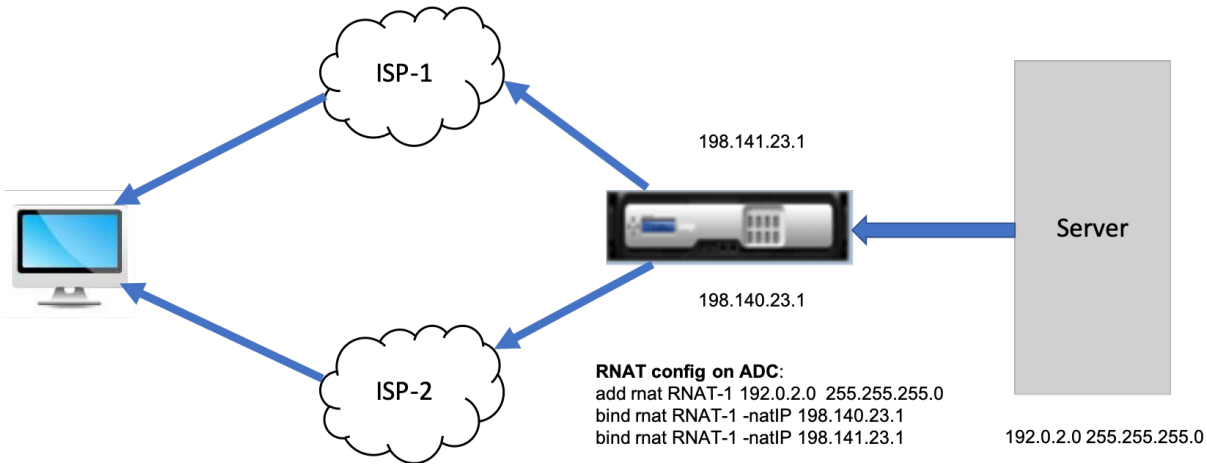
1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 詳細ペインで、モニターをバインドするサービスを選択し、「開く」をクリックします。
3. [サービスの設定] ダイアログボックスの [モニター] タブの [使用可能] で、サービスにバインドするモニターを選択し、[追加] をクリックします。
4. 「OK」をクリックします。
5. サービスペインで、先ほど設定したサービスを選択し、詳細ペインに表示されている設定が正しいことを確認します。

LLB を使用して RNAT を設定

August 15, 2023

アウトバウンドトラフィックのリバースネットワークアドレス変換（RNAT）用の LLB セットアップを設定できます。これにより、特定のフローのリターンネットワークトラフィックが同じパスを経由するようになります。基本的な LLB セットアップの設定の説明に従って基本的な LLBを設定し、次に RNAT の設定の説明に従って RNAT を設定します。次に、「サブネット IP（USNIP）を使用する」モードを有効にします。

次の図では、NetScaler アプライアンスは LLB を使用してアウトバウンドトラフィックをさまざまなリンクにルーティングしています。RNAT の動作中、ADC アプライアンスはアウトバウンドトラフィックの送信元 IP アドレスをパブリック NAT IP アドレス (198.141.23.1) に置き換え、トラフィックを ISP-1 経由でルーティングします。同様に、ADC アプライアンスは送信元 IP アドレスを 198.140.23.1 に置き換え、トラフィックを ISP-2 経由でルーティングします。



CLI を使用して ISP ルーターに SNIP を追加するには

コマンドプロンプトで入力します。

```
add NS IP <subnet of first ISP in the IP router> <subnet mask> -type
    SNIP

add NS IP <subnet of second ISP in the IP router> <subnet mask> -type
    SNIP
<!--NeedCopy-->
```

例:

```
add ns ip 198.140.23.1 255.255.255.0 -type snip

add ns ip 198.141.23.1 255.255.255.0 -type snip
<!--NeedCopy-->
```

CLI を使用して RNAT を設定するには

コマンドプロンプトで入力します。

```
add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))

bind rnat <name> <natIP>@ …

show rnat <name>
<!--NeedCopy-->
```

例:

```
add rnat RNAT-1 192.0.2.0 255.255.255.0
bind rnat RNAT-1 -natIP 198.140.23.1
bind rnat RNAT-1 -natIP 198.141.23.1
```



```
4
5  > show rnat RNAT-1
6      1) RNAT Name: RNAT-1     Network: 192.0.2.0      Netmask:
           255.255.255.0     Traffic Domain: 0
7          UseProxyPort: ENABLED
8
9          NatIP: 198.140.23.1
10         NatIP: 198.141.23.1
11 <!--NeedCopy-->
```

GUI を使用して RNAT を設定するには

1. [システム] > [ネットワーク] > [**NAT]** に移動します。
2. [**RNAT]** タブで、[**RNAT** の設定] をクリックします。
3. RNAT を実行するネットワークを指定します。

> 注
>
> アクセスコントロールリスト（ACL）を使用して RNAT を設定することもできます。詳細については、「RNAT の設定 」を参照してください。

CLI を使用してサブネット IP モードを使用を有効にするには

コマンドプロンプトで入力します。

```
1  enable ns mode USNIP
2
3  show ns mode
4  <!--NeedCopy-->
```

例:

```
1  enable ns mode USNIP
2
3  show ns mode
4          Mode                              Acronym              Status
5          -------                           -------              ------
6   1)     Fast Ramp                         FR                   ON
7   2)     … .
8   8)     Use Subnet IP                     USNIP                ON
9   9)     …
10 <!--NeedCopy-->
```

GUI を使用して [サブネット IP モードを使用] を有効にするには

1. [システム] > [設定] に移動し、[モードと機能] で [モードの設定] をクリックします。
2. 「モードの設定」ダイアログで、「サブネット **IP** を使用」を選択し、「**OK**」をクリックします。

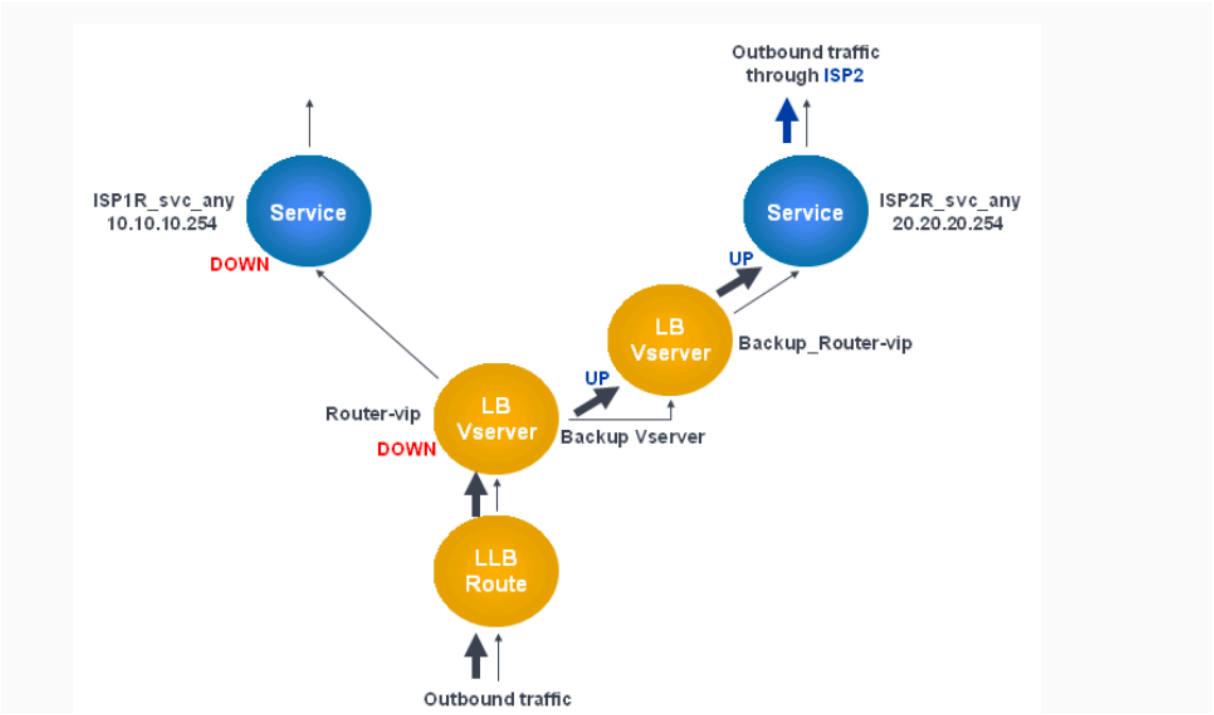
バックアップルートを設定

August 15, 2023

プライマリルートがダウンしたときにサービスが中断されるのを防ぐために、バックアップルートを設定できます。バックアップルートが構成されると、NetScaler ADC アプライアンスはプライマリルートに障害が発生したときにそのルートを自動的に使用します。最初に、LLB 仮想サーバの設定およびサービスのバインドの説明に従って、プライマリ仮想サーバを作成します。バックアップルートを構成するには、プライマリ仮想サーバと同様のセカンダリ仮想サーバを作成し、この仮想サーバをバックアップ仮想サーバ（ルート）として指定します。

以下の図では、ルータ **VIP** がプライマリ仮想サーバで、**Backup_Router-VIP** がバックアップ仮想サーバとして指定されているセカンダリ仮想サーバです。

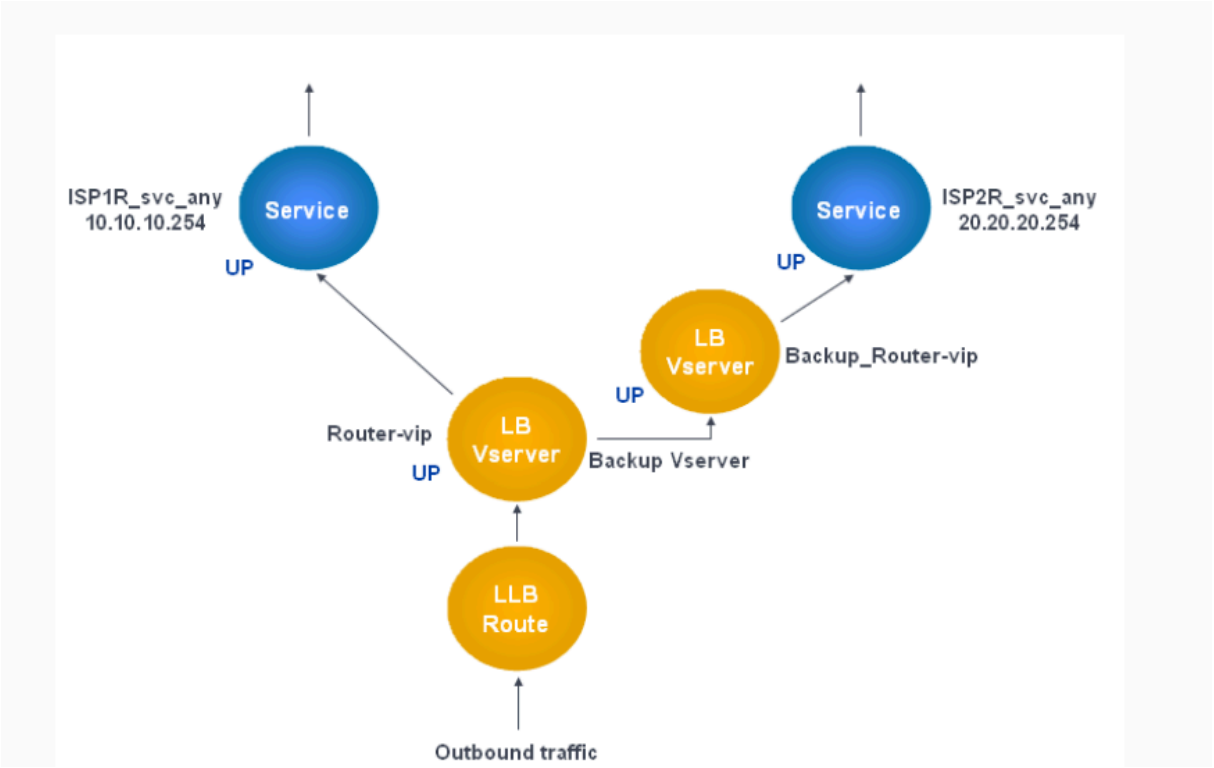
図 1：バックアップルート設定



注**:** ISP が IPv6 アドレスを提供している場合は、IPv4 サービスを前の図の IPv6 サービスに置き換えてください。

デフォルトでは、すべてのトラフィックはプライマリルートを経由して送信されます。ただし、プライマリルートに障害が発生すると、次の図に示すように、すべてのトラフィックがバックアップルートに転送されます。

図 2：運用中のルーティングのバックアップ



注**:** ISP が IPv6 アドレスを提供している場合は、IPv4 サービスを前の図の IPv6 サービスに置き換えてください。

コマンドラインインターフェイスを使用してセカンダリ仮想サーバーをバックアップ仮想サーバーとして設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -backupVserver <string>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Router-vip -backupVServer Backup_Router-vip
> show lb vserver Router-vip
        Router-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 03:09:45.600
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: DESTIP     Persistence Mask: 255.255.255.255
```



```
                    Persistence v6MaskLength: 128   Persistence Timeout: 2
                min
        Backup: Router2-vip
        Connection Failover: DISABLED
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用してセカンダリ仮想サーバーをバックアップ仮想サーバーとして設定するには

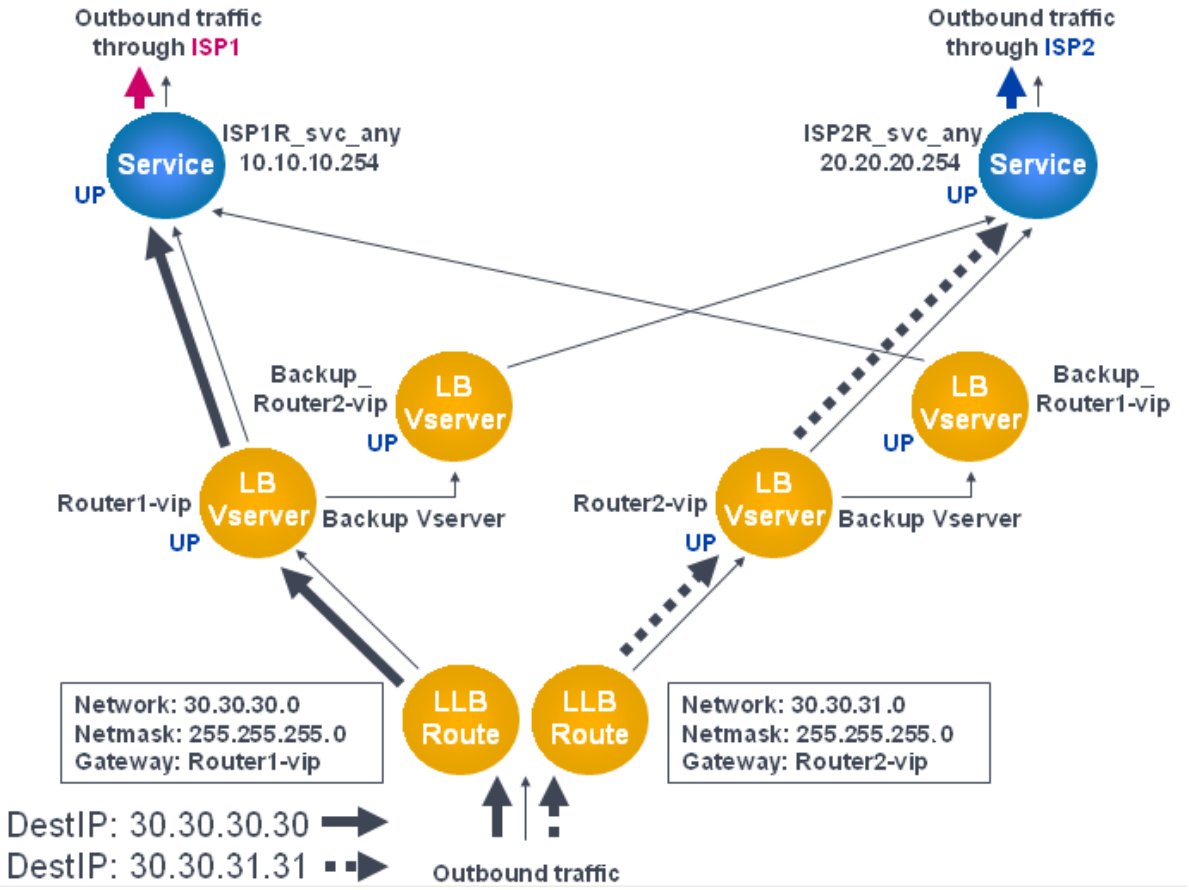
1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、バックアップ仮想サーバーを構成するセカンダリ仮想サーバーを選択します。
2. [仮想サーバーの負荷分散] ダイアログボックスの [詳細設定] で、[保護] を選択します。
3.「バックアップ仮想サーバー」ドロップダウンリストで、セカンダリバックアップ仮想サーバーを選択し、「**OK**」をクリックします。

レジリエントな **LLB** 導入シナリオ

August 15, 2023

次の図には、30.30.30.0 と 30.30.31.0 の 2 つのネットワークがあります。リンク負荷分散は、宛先 IP アドレスに基づいて設定されます。2 つのルートは、それぞれゲートウェイ **Router1-vip** および **Router2-vip** を使用して設定されます。**Router1-vip** は、**Router2-vip** の バックアップとして構成されています。宛先 IP が 30.30.30.30 として指定されたすべてのトラフィックは、**Router1-vip** を介して送信され、宛先 IP が 30.30.31.31 として指定されたトラフィックは、**Router2-vip** を介して送信されます。

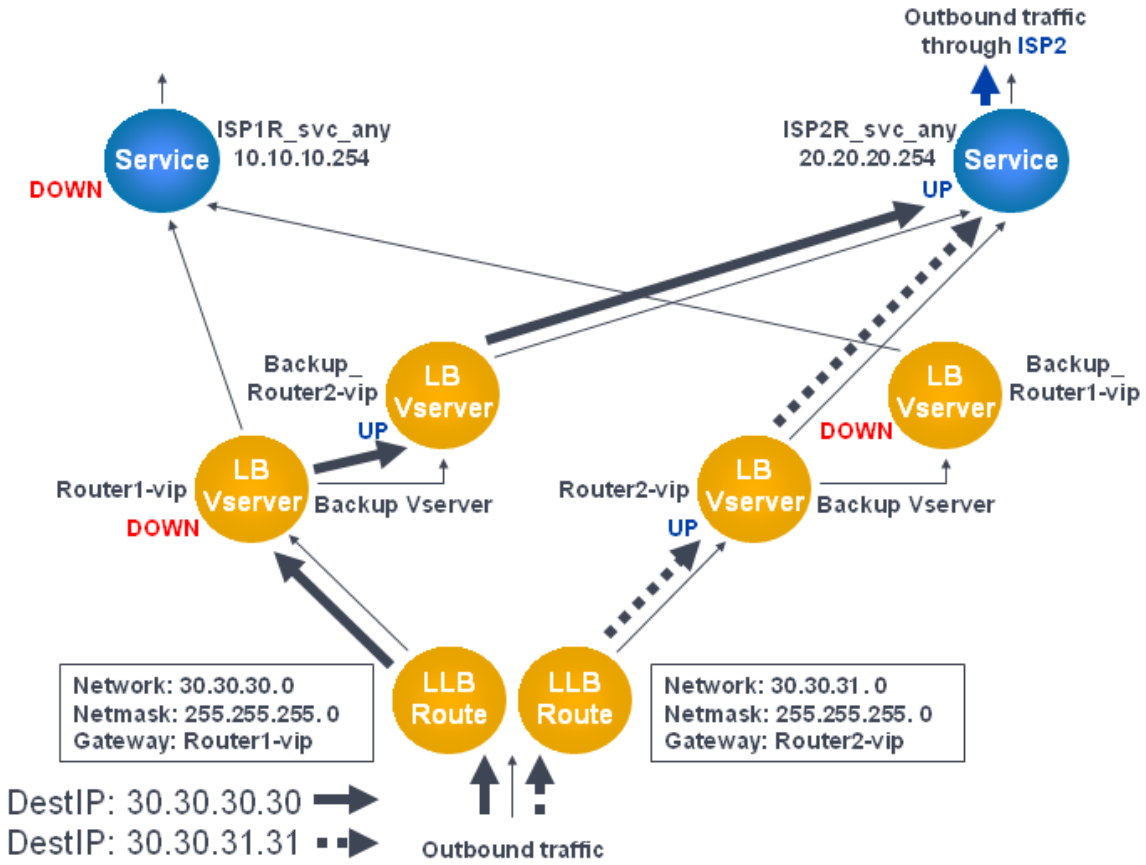
図 1: レジリエントな LLB 導入設定



注: ISP が IPv6 アドレスを提供している場合は、IPv4 サービスを前の図の IPv6 サービスに置き換えてください。

ただし、いずれかのゲートウェイ（**Router1-vip** または **Router2-vip**）が DOWN の場合、トラフィックはバックアップルータを経由してルーティングされます。次の図では、ISP1 の **Router1-vip** が DOWN であるため、宛先 IP が 30.30.30.30 として指定されたすべてのトラフィックも ISP2 経由で送信されます。

図 2: 耐障害性の高い LLB 展開シナリオ



注: ISP が IPv6 アドレスを提供している場合は、IPv4 サービスを前の図の IPv6 サービスに置き換えます。

LLB セットアップを監視する

August 15, 2023

構成が起動して実行されたら、各サービスと仮想サーバーの統計を表示して、発生する可能性のある問題を確認できます。

仮想サーバーの統計情報を表示する

仮想サーバーのパフォーマンスを評価したり、問題をトラブルシューティングしたりするために、NetScaler ADC アプライアンスに構成されている仮想サーバーの詳細を表示できます。すべての仮想サーバーの統計の概要を表示できます。仮想サーバーの名前を指定して、その仮想サーバーの統計のみを表示することもできます。次の詳細を表示できます。

- 名前

- IP アドレス
- ポート
- プロトコル
- 仮想サーバーの状態
- 受け取ったリクエストの割合
- Rate of hits

CLI を使用して仮想サーバーの統計情報を表示する

NetScaler で現在構成されているすべての仮想サーバー、または 1 つの仮想サーバーの統計の概要を表示するには、コマンドプロンプトで次のように入力します。

```
stat lb vserver -detail] [<name>]
<!--NeedCopy-->
```

例:

```
stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80          HTTP           UP      5/s
        0/s
Two                          *     0           TCP         DOWN      0/s
        0/s
Three                        *  2598           TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53           DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80          HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80          HTTP           UP      0/s
        0/s
 Done
<!--NeedCopy-->
```

GUI を使用して仮想サーバーの統計情報を表示する

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] **>** [統計] に移動します。
2. 1 つの仮想サーバーの統計のみを表示する場合は、詳細ペインで仮想サーバーを選択し、[統計] をクリックします。

サービスの統計情報を表示する

サービス統計を使用して、要求、応答、要求バイト、応答バイト、現在のクライアント接続、サージキュー内の要求、現在のサーバー接続などのレートを表示できます。

CLI を使用してサービスの統計情報を表示する

コマンドプロンプトで入力します。

```
stat service <name>
<!--NeedCopy-->
```

例:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスの統計を表示する

1. [トラフィック管理] **>** [負荷分散] **>** [サービス] **>** [統計] に移動します。
2. 1 つのサービスの統計のみを表示する場合は、サービスを選択して [Statistics] をクリックします。

負荷分散

August 15, 2023

負荷分散機能は、Web ページおよびその他の保護されたアプリケーションに対するユーザー要求を、すべて同じコンテンツをホスト（またはミラー）する複数のサーバーに分散します。負荷分散は、主に使用頻度の高いアプリケーションへのユーザー要求を管理するために使用します。これにより、パフォーマンスの低下や停止を防ぎ、保護されたアプリケーションにユーザーがアクセスできるようになります。ロードバランシングは、フォールトトレランスも提供します。保護されたアプリケーションをホストする 1 つのサーバが使用できなくなると、この機能により、同じアプリケーションをホストする他のサーバにユーザ要求が分散されます。

ロードバランシング機能は、次のように設定できます。

- 特定の保護された Web サイト、アプリケーション、またはリソースに対するすべての要求を、同じ構成の 2 つ以上のサーバー間で分散します。
- いくつかの異なるアルゴリズムのいずれかを使用して、現在のユーザー接続が最も少ないサーバーや負荷が最も軽いサーバーなど、さまざまな要因に基づいて、着信するユーザー要求それぞれを受信する必要があるサーバーを決定します。

負荷分散機能は、NetScaler ADC アプライアンスのコア機能です。ほとんどのユーザーは、最初に機能する基本構成をセットアップしてから、接続の永続性など、さまざまな設定をカスタマイズします。さらに、障害からの構成の保護、クライアントトラフィックの管理、サーバーの管理と監視、および大規模な展開の管理のための機能を構成できます。

負荷分散の方法

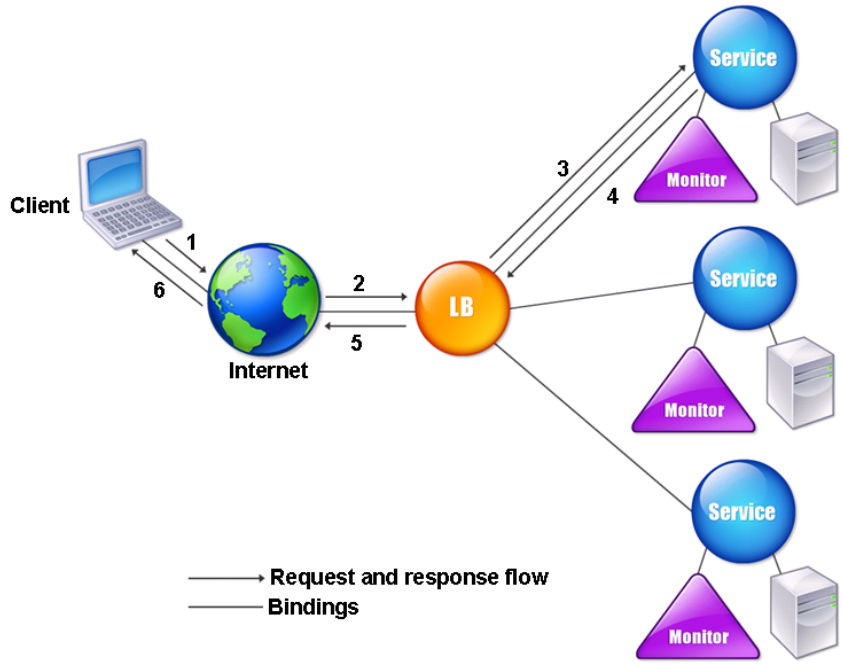
August 15, 2023

基本的な負荷分散設定では、クライアントは NetScaler アプライアンスで構成された仮想サーバーの IP アドレスに要求を送信します。仮想サーバーは、負荷分散アルゴリズムと呼ばれる事前設定されたパターンに従って、負荷分散アプリケーションサーバーに分散します。場合によっては、特定の IP アドレスではなく、負荷分散仮想サーバーにワイルドカードアドレスを割り当てたい場合があります。アプライアンスのグローバル HTTP ポートを指定する手順については、「グローバル **HTTP** ポート」を参照してください。

負荷分散の基本

負荷分散設定には、負荷分散仮想サーバーと複数の負荷分散アプリケーションサーバーが含まれます。仮想サーバーは、受信したクライアント要求を受信し、負荷分散アルゴリズムを使用してアプリケーションサーバーを選択し、選択したアプリケーションサーバーに要求を転送します。次の概念図は、一般的な負荷分散展開を示しています。もう 1 つのバリエーションとして、グローバル HTTP ポートを割り当てる方法があります。

図 1：負荷分散アーキテクチャ



負荷分散仮想サーバーは、複数のアルゴリズム (またはメソッド) を使用して、管理する負荷分散サーバー間で負荷を分散する方法を決定できます。デフォルトの負荷分散方式は、最小接続方式です。この方式では、NetScaler ADC アプライアンスは、現在アクティブなユーザー接続が最も少ない負荷分散アプリケーションサーバーに各着信クライアント接続を転送します。

一般的な NetScaler 負荷分散設定で構成するエンティティは次のとおりです。

- 負荷分散仮想サーバー。クライアントが特定の負荷分散された Web サイトまたはアプリケーションの接続要求を送信する IP アドレス、ポート、およびプロトコルの組み合わせ。アプリケーションがインターネットからアクセス可能な場合、仮想サーバーの IP (VIP) アドレスはパブリック IP アドレスです。アプリケーションが LAN または WAN からのみアクセス可能である場合、VIP は通常、プライベート（ICANN ルーティング不可）IP アドレスです。
- **Service.** 負荷分散された特定のアプリケーションサーバーにリクエストをルーティングするために使用される IP アドレス、ポート、およびプロトコルの組み合わせ。サービスは、アプリケーションサーバー自体を論理的に表現したものでも、複数のアプリケーションをホストするサーバーで実行されているアプリケーションを論理的に表現したものでもかまいません。サービスを作成したら、それを負荷分散仮想サーバーにバインドします。
- サーバーオブジェクト。IP アドレスでサーバーを識別する代わりに、物理サーバーに名前を割り当てることができる仮想エンティティ。サーバーオブジェクトを作成する場合、サービスを作成するときに、サーバーの IP アドレスの代わりにその名前を指定できます。それ以外の場合は、サービスを作成するときにサーバーの IP

アドレスを指定する必要があり、その IP アドレスがサーバーの名前になります。

- モニタ。サービスを追跡し、サービスが正しく動作していることを確認する NetScaler アプライアンス上のエンティティ。モニターは、割り当てられている各サービスを定期的にプローブ（またはヘルスチェックを実行）します。タイムアウトで指定された時間内にサービスが応答せず、指定した回数のヘルスチェックが失敗した場合、そのサービスはダウンとマークされます。NetScaler アプライアンスは、サービスが応答しなくなる原因となった問題が修正されるまで、負荷分散の実行時にそのサービスをスキップします。

負荷分散設定の仮想サーバー、サービス、および負荷分散アプリケーションサーバーは、インターネットプロトコルバージョン 4 (IPv4) またはインターネットプロトコルバージョン 6 (IPv6) の IP アドレスのいずれかを使用できます。1 つの負荷分散設定で IPv4 アドレスと IPv6 アドレスを混在させることができます。

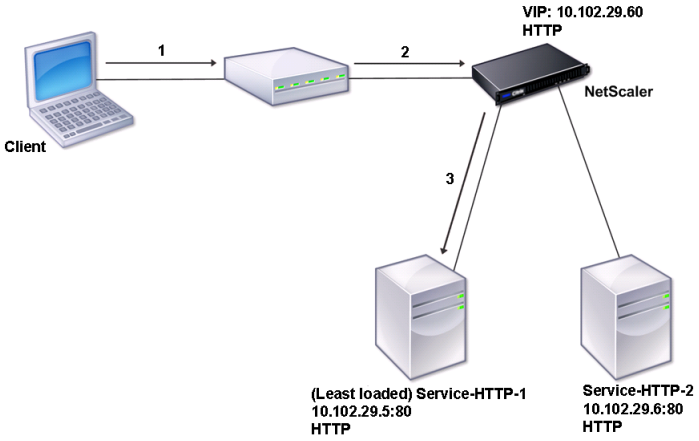
負荷分散設定のバリエーションについては、次のユースケースを参照してください。

- ダイレクトサーバーリターンモードでの負荷分散の設定
- DSR モードでの Linux サーバーの設定
- TOS を使用する場合の DSR モードの設定
- IP over IP を使用した DSR モードでのロードバランシングの設定
- ワンアームモードでのロードバランシングの設定
- インラインモードでの負荷分散の設定
- 侵入検知システムサーバーの負荷分散
- リモートデスクトッププロトコルサーバーの負荷分散

トポロジーの理解

負荷分散設定では、負荷分散サーバーはクライアントとサーバーファームの間に論理的に配置され、サーバーファーム内のサーバーへのトラフィックフローを管理します。NetScaler アプライアンスでは、アプリケーションサーバーはサービスと呼ばれる仮想エンティティによって表されます。次の図は、基本的な負荷分散構成のトポロジを示しています。

図 2：基本的な負荷分散トポロジ

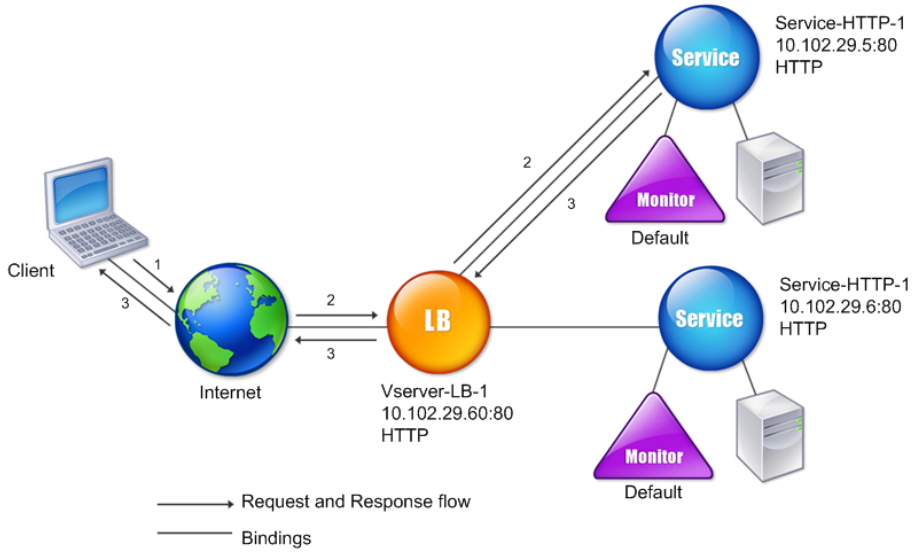


この図では、ロードバランシングを使用してサーバへのトラフィックフローを管理します。仮想サーバーは、クライアントからの要求に対してサービスを選択して割り当てます。サービス HTTP-1 と Service-HTTP-2 が作成され、VServer-LB-1 という名前の仮想サーバーにバインドされるシナリオを考えてみましょう。VServer-LB-1 は、クライアント要求をサービス HTTP-1 またはサービス HTTP-2 のいずれかに転送します。NetScaler アプライアンスは、最小限の接続負荷分散方法を使用して、リクエストごとにサービスを選択します。次の表は、アプライアンスで設定する必要がある基本エンティティの名前と値を示しています。

エンティティ	名前	IP アドレス	ポート	プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.60	80	HTTP
Services	Service-HTTP-1	10.102.29.5	80	HTTP
		Service-HTTP-2	10.102.29.6	80
モニター	デフォルト	なし	なし	なし

次の図は、前述の表で説明した負荷分散のサンプル値と必須パラメータを示しています。

図 3：負荷分散エンティティモデル



IP アドレスとポートの代わりにワイルドカードを使用する

場合によっては、仮想サーバーの IP アドレス、ポート、またはサービスのポートにワイルドカードを使用する必要がある場合があります。次の場合は、ワイルドカードの使用が必要になる場合があります。

- NetScaler ADC アプライアンスが透過パススルーとして構成されている場合、NetScaler ADC アプライアンスは、送信先の IP またはポートに関係なく、送信されるすべてのトラフィックを受け入れる必要があります。
- 1 つ以上のサービスが、あまり知られていないポートでリッスンしている場合。
- 1 つ以上のサービスの場合は、時間が経つにつれて、受信するポートが変更されます。
- 1 つの NetScaler アプライアンスで構成できる IP アドレスとポートの数が上限に達した場合。
- 特定の仮想 LAN 上のすべてのトラフィックをリッスンする仮想サーバーを作成する場合。

ワイルドカードで構成された仮想サーバーまたはサービスがトラフィックを受信すると、NetScaler ADC アプライアンスは実際の IP アドレスまたはポートを特定し、サービスおよび関連する負荷分散アプリケーションサーバーのレコードを作成します。動的に作成されたこれらのレコードは、動的に学習されたサーバおよびサービスレコードと呼ばれます。

たとえば、ファイアウォールの負荷分散設定では、IP アドレスとポートの両方にワイルドカードを使用できます。ワ

イルドカード TCP サービスをこの種類の負荷分散仮想サーバーにバインドすると、仮想サーバーは他のサービスまたは仮想サーバーと一致しないすべての TCP トラフィックを受信して処理します。

次の表に、ワイルドカード構成の種類とその使用時期について説明します。

IP	ポート	プロトコル	説明
*	*	TCP	NetScaler ADC アプライアンス上の任意の IP アドレスおよびポートに送信されるトラフィックを受け入れる一般的なワイルドカード仮想サーバー。ワイルドカード仮想サーバを使用する場合、アプライアンスは各サービスの IP とポートを動的に学習し、トラフィックを処理するときに必要なレコードを作成します。
*	*	TCP	ファイアウォールの負荷分散仮想サーバー。ファイアウォールサービスをこの仮想サーバーにバインドすると、NetScaler アプライアンスはトラフィックをファイアウォール経由で宛先に渡します。
IP アドレス	*	TCP、UDP、および任意	ポートに関係なく、指定された IP アドレスに送信されたすべてのトラフィックを受け入れる仮想サーバ。このタイプの仮想サーバには、トラフィックをリダイレクトするサービスを明示的にバインドする必要があります。ダイナミックに学習するわけではありません。

IP	ポート	プロトコル	説明
*	port	SSL、SSL_TCP	特定のポートの任意の IP アドレスに送信されるすべてのトラフィックを受け入れる仮想サーバー。グローバルトランスペアレント SSL オフロードに使用されます。同じプロトコルタイプのサービスに対して通常実行されるすべての SSL、HTTP、および TCP 処理が、この特定のポートに送信されるトラフィックに適用されます。アプライアンスは、ポートを使用して、使用する必要のあるサービスの IP を動的に学習します。—cleartext が指定されていない場合、NetScaler ADC アプライアンスはエンドツーエンド SSL を使用します。
*	port	該当なし	ポートへのトラフィックを受け入れることができる他のすべての仮想サーバ。これらの仮想サーバーにはサービスをバインドしません。NetScaler ADC アプライアンスはそれらを動的に学習します。

注：NetScaler ADC アプライアンスをグローバル（ワイルドカード）ポートを使用するトランスペアレントパススルーとして構成している場合は、エッジモードをオンにすることができます。
詳細については、「エッジモードの設定」を参照してください。

NetScaler ADC アプライアンスは、最初に完全一致を試行して、仮想サーバーとサービスの検索を試みます。何も見つからない場合は、ワイルドカードに基づいて次の順序で一致するものを探し続けます。

1. 特定の IP アドレスと特定のポート番号

2. 特定の IP アドレスと * (ワイルドカード) ポート
3. • (ワイルドカード) IP アドレスと特定のポート
4. • (ワイルドカード) IP アドレスと * (ワイルドカード) ポート

アプライアンスが IP アドレスまたはポート番号で仮想サーバを選択できない場合、要求で使用されるプロトコルに基づいて、次の順序で仮想サーバを検索します。

1. HTTP
2. TCP
3. ANY

グローバル **HTTP** ポートの設定

グローバル HTTP ポートにはサービスや仮想サーバーを設定しません。代わりに、set ns param コマンドを使用して特定のポートを設定します。このポートを構成すると、NetScaler アプライアンスはポート番号と一致するすべてのトラフィックを受け入れ、HTTP トラフィックとして処理し、そのトラフィックのサービスを動的に学習して作成します。

グローバル HTTP ポートとして複数のポート番号を設定できます。1 回の set ns param コマンドで複数のポート番号を指定する場合は、ポート番号を 1 つの空白で区切ります。1 つ以上のポートがグローバル HTTP ポートとしてすでに指定されていて、現在設定されているポートを削除せずに 1 つ以上のポートを追加する場合は、現在のポート番号と新しいポート番号をすべてコマンドで指定する必要があります。ポート番号を追加する前に、show ns param コマンドを使用して、現在設定されているポートを確認してください。

コマンドラインインターフェイスを使用してグローバル **HTTP** ポートを設定するには

コマンドプロンプトで次のコマンドを入力して、グローバル HTTP ポートを構成し、構成を確認します。

```
set ns param  -httpPort <port>

show ns param
<!--NeedCopy-->
```

例 **1:** ポートをグローバル **HTTP** ポートとして設定する　この例では、ポート 80 はグローバル HTTP ポートとして設定されています。

```
set ns param -httpPort 80
Done
show ns param
         Global configuration settings:
                           HTTP port(s): 80
                       Max connections: 0
           Max requests per connection: 0
                    Client IP insertion: DISABLED
```



```
                          Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
        ...
        ...
<!--NeedCopy-->
```

例 **2:1** つ以上のグローバル **HTTP** ポートがすでに設定されている場合のポートの追加 **** この例では、ポート 8888 がグローバル HTTP ポートリストに追加されます。ポート 80 はすでにグローバル HTTP ポートとして設定されています。

```
> show ns param
        Global configuration settings:
                          HTTP port(s): 80
                       Max connections: 0
           Max requests per connection: 0
                    Client IP insertion: DISABLED
                         Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                          Min Path MTU: 576
        ...
        ...
 Done
> set ns param -httpPort 80 8888
 Done
>  show ns param

        Global configuration settings:
                          HTTP port(s): 80,8888
                       Max connections: 0
           Max requests per connection: 0
                    Client IP insertion: DISABLED
                         Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                          Min Path MTU: 576

        ...
        ...
 Done
>
<!--NeedCopy-->
```

構成ユーティリティを使用してグローバル **HTTP** ポートを構成するには

1. [システム] **>** [設定] **>** [**HTTP** パラメータの変更] に移動し、HTTP ポート番号を追加します。

基本的な負荷分散を設定する

April 15, 2024

初期負荷分散設定を設定する前に、負荷分散機能を有効にします。次に、負荷分散グループのサーバーごとに少なくとも 1 つのサービスを作成します。サービスを構成したら、負荷分散仮想サーバーを作成し、各サービスを仮想サーバーにバインドする準備が整いました。これで初期設定は完了です。さらに構成を進める前に、構成を確認して、各要素が適切に設定され、期待どおりに動作していることを確認してください。

負荷分散を有効にする

負荷分散機能が無効になっている場合は、サービスや仮想サーバーなどの負荷分散エンティティを構成できますが、機能を有効にするまで機能しません。

CLI を使用して負荷分散を有効にするには

コマンドプロンプトで次のコマンドを入力して負荷分散を有効にし、構成を確認します。

- ns 機能 LB を有効にする
- show ns feature

例

```
> enable ns feature LoadBalancing

 Done

> show ns feature


        Feature                         Acronym             Status

        -------                         -------             ------

 1)     Web Logging                     WL                  OFF

 2)     Surge Protection                SP                  ON

 3) Load Balancing LB ON

 .

 .
```



```
22
23       .
24
25       24)    NetScaler Push                  push                  OFF
26
27       Done
28 <!--NeedCopy-->
```

GUI を使用して負荷分散を有効にするには

[システム] > [設定] に移動し、[基本機能の設定] で [負荷分散] を選択します。

サーバーオブジェクトを設定する

NetScaler アプライアンスでサーバーのエントリを作成します。NetScaler アプライアンスは、IP アドレスベースのサーバーとドメインベースのサーバーをサポートします。IP アドレスベースのサーバーを作成する場合、サービスの作成時に IP アドレスの代わりにサーバーの名前を指定できます。ドメインベースのサーバーの DNS の設定については、「ドメインネームシステム」を参照してください。

CLI を使用してサーバオブジェクトを作成するには

コマンドプロンプトで入力します:

```
1 add server `<name>`@ `<IPAddress>`@ | `<domain>`
2 <!--NeedCopy-->
```

IP アドレスベースのネームサーバを追加する例**:**

```
1 add server web_serv 10.102.27.150
2 <!--NeedCopy-->
```

ドメインベースのサーバーを追加する例**:**

```
1 add server web_serv test.com
2 <!--NeedCopy-->
```

GUI を使用してサーバーオブジェクトを作成するには

[トラフィック管理] > [負荷分散] > [サーバー] に移動し、サーバーオブジェクトを追加します。

サービスを構成する

負荷分散機能を有効にしたら、負荷分散設定に含めるアプリケーションサーバーごとに少なくとも 1 つのサービスを作成する必要があります。構成するサービスは、NetScaler アプライアンスと負荷分散サーバー間の接続を提供します。各サービスには名前があり、IP アドレス、ポート、提供されるデータの種類を指定します。

最初にサーバーオブジェクトを作成せずにサービスを作成した場合、サービスの IP アドレスは、サービスをホストするサーバーの名前でもあります。IP アドレスではなく名前でサーバーを識別したい場合は、サーバーオブジェクトを作成してから、サービスの作成時に IP アドレスの代わりにサーバーの名前を指定できます。

UDP をトランスポート層プロトコルとして使用するサービスを作成すると、ping モニターが自動的にサービスにバインドされます。ping モニターは、組み込みモニターの中で最も基本的なものです。トランスポート層プロトコルとして TCP を使用するサービスを作成すると、TCP_DEFAULT モニターが自動的にサービスにバインドされます。負荷分散設定を管理するための戦略を策定する際、別のタイプのモニターまたは複数のモニターをサービスにバインドすることを決定する場合があります。

サービスを作成

サービスを作成する前に、さまざまなサービスの種類とそれぞれの使用方法を理解する必要があります。次のリストは、NetScaler アプライアンスでサポートされているサービスの種類を示しています。

- **HTTP**

  標準の Web サイトや Web アプリケーションなど、HTTP トラフィックを受け入れる負荷分散型サーバーに使用されます。HTTP サービスタイプにより、NetScaler アプライアンスはレイヤー 7 Web サーバーの圧縮、コンテンツフィルタリング、キャッシュ、および Client Keep-Alive サポートを提供できます。このサービスタイプは、仮想サーバーの IP ポート挿入、リダイレクトポート書き換え、Web 2.0 プッシュ、URL リダイレクトのサポートもサポートしています。

  HTTP は TCP ベースのアプリケーションプロトコルなので、Web サーバーには TCP サービスタイプを使用することもできます。ただし、その場合、NetScaler アプライアンスはレイヤー 4 の負荷分散しか実行できません。前述のレイヤ 7 サポートのいずれも提供できません。

- **SSL**

  e コマース Web サイトやショッピングカートアプリケーションなど、HTTPS トラフィックを受け入れるサーバーに使用されます。SSL サービスタイプにより、NetScaler アプライアンスは安全な Web アプリケーションの SSL トラフィックを暗号化および復号化（SSL オフロードを実行）できます。また、HTTP パーシステンス、コンテンツスイッチング、リライト、仮想サーバーの IP ポート挿入、Web 2.0 プッシュ、URL リダイレクトもサポートしています。

  SSL_BRIDGE、SSL_TCP、または TCP サービスタイプを使用することもできます。ただし、その場合、アプライアンスはレイヤー 4 の負荷分散のみを実行します。SSL オフロードや、前述のレイヤ 7 サポートのいずれも提供できません。

- **FTP**

  FTP トラフィックを受け入れるサーバーに使用されます。FTP サービスタイプにより、NetScaler アプライアンスは FTP プロトコルの特定の詳細をサポートできます。

  FTP サーバーには TCP または任意のサービスタイプを使用することもできます。

- **TCP**

  さまざまなタイプの TCP トラフィックを受け入れるサーバー、または特定のタイプのサービスが利用できないタイプの TCP トラフィックを受け入れるサーバーに使用されます。

  これらのサーバーには任意のサービスタイプを使用することもできます。

- **SSL_TCP**

  SSL オフロードをサポートするために、HTTP ベース以外の SSL トラフィックを受け入れるサーバーに使用されます。

  これらのサービスには TCP サービスタイプを使用することもできます。その場合、NetScaler アプライアンスはレイヤー 4 の負荷分散と SSL オフロードの両方を実行します。

- **UDP**

  UDP トラフィックを受け入れるサーバーに使用されます。任意のサービスタイプを使用することもできます。

- **SSL_BRIDGE**

  NetScaler アプライアンスに SSL オフロードを実行させたくない場合に、SSL トラフィックを受け入れるサーバーに使用されます。または、SSL_TCP サービスタイプを使用することもできます。

- **NNTP**

  ネットワークニュース転送プロトコル (NNTP) トラフィックを受け入れるサーバー (通常は Usenet サイト) に使用されます。

- **DNS**

  DNS トラフィックを受け入れるサーバー (通常はネームサーバー) に使用されます。DNS サービスタイプでは、NetScaler アプライアンスは各 DNS 要求と応答のパケット形式を検証します。DNS 応答をキャッシュすることもできます。DNS ポリシーを DNS サービスに適用できます。

  これらのサービスには UDP サービスタイプを使用することもできます。ただし、その場合、NetScaler アプライアンスはレイヤー 4 の負荷分散しか実行できません。DNS 固有の機能はサポートできません。

- **ANY**

  あらゆるタイプの TCP、UDP、または ICMP トラフィックを受け入れるサーバーに使用されます。ANY パラメータは、主にファイアウォールの負荷分散とリンク負荷分散に使用されます。

- **SIP-UDP**

UDP ベースのセッション開始プロトコル (SIP) トラフィックを受け入れるサーバーに使用されます。SIP はマルチメディア通信セッションの開始、管理、終了を行うもので、インターネットテレフォニー (VoIP) の標準として普及しています。

これらのサービスには UDP サービスタイプを使用することもできます。ただし、NetScaler アプライアンスは、レイヤー 4 の負荷分散のみを実行します。SIP 固有の機能はサポートできません。

- **DNS-TCP**

  DNS トラフィックを受け入れるサーバーに使用され、NetScaler アプライアンスは DNS サーバーに送信される TCP トラフィックのプロキシとして機能します。DNS-TCP サービスタイプのサービスでは、NetScaler アプライアンスは各 DNS 要求と応答のパケット形式を検証し、DNS サービスタイプと同様に DNS 応答をキャッシュできます。

  これらのサービスには TCP サービスタイプを使用することもできます。ただし、その場合、NetScaler アプライアンスは外部 DNS ネームサーバーのレイヤー 4 負荷分散のみを実行します。DNS 固有の機能はサポートできません。

- **RTSP**

  リアルタイムストリーミングプロトコル (RTSP) トラフィックを受け入れるサーバーに使用されます。RTSP は、マルチメディアやその他のストリーミングデータを配信します。オーディオ、ビデオ、およびその他のタイプのストリーミングメディアをサポートするには、このタイプを選択してください。

  これらのサービスには TCP サービスタイプを使用することもできます。ただし、NetScaler アプライアンスは、レイヤー 4 の負荷分散のみを実行します。RTSP ストリームを解析したり、RTSPID 永続性または RTSP NAT をサポートしたりすることはできません。

- **DHCPRA**

  DHCP トラフィックを受け入れるサーバーに使用されます。DHCPRA サービスタイプは、VLAN 間の DHCP 要求と応答を中継するために使用できます。

- **DIAMETER**

  複数の Diameter サーバー間の Diameter トラフィックの負荷分散に使用されます。Diameter はメッセージベースのロードバランシングを使用します。

- **SSL_DIAMETER**

  SSL 経由の Diameter トラフィックの負荷分散に使用されます。

  NetScaler アプライアンスが関連する負荷分散サーバーに接続して動作していることを確認するまで、サービスは無効として指定されます。その時点で、サービスは ENABLED と指定されます。その後、NetScaler アプライアンスは定期的にサーバーの状態を監視し、監視プローブ（ヘルスチェックと呼ばれる）に応答しなかったサーバーは、応答するまで無効状態に戻します。

> 注:
>
> 単一の CLI コマンドまたは同じダイアログボックスからさまざまなサービスを作成できます。範囲内の名前は、サフィックス/プレフィックスとして使用される番号によって異なります。たとえば、サービス 1、サービス 2 などです。GUI では、IP アドレスの最後のオクテットでのみ範囲を指定できます。IPv4 アドレスの場合は 4 番目、IPv6 アドレスの場合は 8 番目です。コマンドラインから、IP アドレスの任意のオクテットで範囲を指定できます。

- **QUIC**

  UDP ベースの QUIC ビデオトラフィックを受け入れる負荷分散サーバーで使用されます。このサービスにより、NetScaler アプライアンスは暗号化された ABR ビデオトラフィックを UDP プロトコル上で最適化できます。

CLI を使用してサービスを作成するには

コマンドプロンプトで入力します:

```
add service <name> <serverName> <serviceType> <port>

add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

GUI を使用してサービスを作成するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. 「サービスの作成」ダイアログで、次のパラメータの値を指定します。
   - サービス名—名前
   - サーバー—サーバー名
   - プロトコル—サービスタイプ
   - ポート-ポート
4. [作成] をクリックし、[閉じる] をクリックします。作成したサービスが [サービス] ペインに表示されます。

仮想サーバーの作成

サービスを作成したら、負荷分散された Web サイト、アプリケーション、またはサーバーのトラフィックを受け入れる仮想サーバーを作成する必要があります。負荷分散が設定されると、ユーザーは仮想サーバーの IP アドレスまたは FQDN を介して負荷分散された Web サイト、アプリケーション、またはサーバーに接続します。

> メモ:
>
> - プレフィックスが「app_」の仮想サーバ名は ns.conf ファイルに存在し、show コマンドを実行すると表示されますが、GUI には表示されません。ただし、「app」というプレフィックスが付いた仮想サーバー名は GUI に表示されます。
> - 仮想サーバーは、作成したサービスをバインドするまで、および NetScaler がそれらのサービスに接続して動作可能であることを確認するまで、停止中として指定されます。検証後、仮想サーバーは稼働中と指定されます。
> - 仮想サーバーにすべてのポートをリッスンさせたい場合は、特定のポートの代わりにワイルドカード (*) 文字を使用してください。
>
>   **Example:** `add lb vserver v1 TCP 1.11.1.1 *`.

CLI を使用して仮想サーバーを作成するには

コマンドプロンプトで入力します:

```
add lb vserver <name> <serviceType> <ip> <port>

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

GUI を使用して仮想サーバーを作成するには

[トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを作成します。

サービスを仮想サーバーにバインドする

注:1 つのサービスを最大 500 台の仮想サーバーにバインドできます。

サービスと仮想サーバーを作成したら、サービスを仮想サーバーにバインドする必要があります。通常、サービスは同じタイプの仮想サーバーにバインドされますが、次に示すように、特定のタイプのサービスを特定の異なるタイプの仮想サーバーにバインドできます。

仮想サーバータイプ	サービスの種類	コメント
HTTP	SSL	通常、SSL サービスを HTTP 仮想サーバーにバインドして暗号化します。
SSL	HTTP	通常は、HTTP サービスを SSL 仮想サーバーにバインドして SSL オフロードを行います。

仮想サーバータイプ	サービスの種類	コメント
SSL_TCP	TCP	通常は、TCP サービスを SSL_TCP 仮想サーバーにバインドして、他の TCP の SSL オフロード (コンテンツ認識なしの SSL 復号化) を行います。

仮想サーバーにバインドされたサービスの状態によって、仮想サーバーの状態が決まります。バインドされたすべてのサービスがダウンしている場合、仮想サーバーはダウンとマークされ、バインドされたサービスのいずれかが UP または OUT OF SERVICE の場合、仮想サーバーの状態は UP になります。

CLI を使用してサービスを負荷分散仮想サーバーにバインドするには

コマンドプロンプトで入力します:

```
bind lb vserver <name> <serviceName>

bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスを負荷分散仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. サービスセクションをクリックして、バインドするサービスを選択します。

注:1 つのサービスを複数の仮想サーバーにバインドできます。

構成を確認する

基本構成が完了したら、負荷分散セットアップで各サービスと負荷分散仮想サーバーのプロパティを表示し、それぞれが正しく構成されていることを確認できます。構成が起動して実行されると、各サービスおよび負荷分散仮想サーバーの統計情報を表示して、考えられる問題をチェックできます。

サーバーオブジェクトのプロパティを表示する

NetScaler アプライアンス構成内の任意のサーバーオブジェクトの名前、状態、IP アドレスなどのプロパティを表示できます。

CLI を使用してサーバーオブジェクトのプロパティを表示するには　コマンドプロンプトで入力します:

```
show server <serverName>

show server server-1
<!--NeedCopy-->
```

GUI を使用してサーバーオブジェクトのプロパティを表示するには　[トラフィック管理] > [負荷分散] > [サーバー] に移動します。使用可能なサーバーのパラメーター値が詳細ペインに表示されます。

仮想サーバーのプロパティを表示する

仮想サーバーの名前、状態、有効状態、IP アドレス、ポート、プロトコル、メソッド、バインドされたサービスの数などのプロパティを表示できます。基本的な負荷分散設定以外を構成した場合は、仮想サーバーの永続性設定、それらにバインドされているポリシー、および仮想サーバーにバインドされているキャッシュリダイレクトとコンテンツスイッチング仮想サーバーを表示できます。

CLI を使用して負荷分散仮想サーバーのプロパティを表示するには　コマンドプロンプトで入力します:

```
show lb vserver <name>

show lb vserver Vserver-LB-1
<!--NeedCopy-->
```

GUI を使用して負荷分散仮想サーバーのプロパティを表示するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 詳細ウィンドウで、仮想サーバーをクリックして、詳細ウィンドウの下部にそのプロパティを表示します。
3. この仮想サーバーにバインドされているキャッシュリダイレクトおよびコンテンツスイッチング仮想サーバーを表示するには、[**CS/CR** バインディングの表示] をクリックします。

サービスのプロパティを表示する

設定されているサービスの名前、状態、IP アドレス、ポート、プロトコル、最大クライアント接続、接続あたりの最大リクエスト数、およびサーバータイプを表示し、この情報を使用してサービス設定の間違いをトラブルシューティングできます。

CLI を使用してサービスのプロパティを表示するには　コマンドプロンプトで入力します:


```
show service <name>

show service Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスのプロパティを表示するには　[トラフィック管理] > [負荷分散] > [サービス] に移動します。利用可能なサービスの詳細が [サービス] ペインに表示されます。

サービスのバインディングを表示する

サービスがバインドされている仮想サーバーのリストを表示できます。バインディング情報には、サービスがバインドされている仮想サーバーの名前、IP アドレス、ポート、および状態も示されます。バインディング情報を使用して、サービスを仮想サーバーにバインドする際の問題のトラブルシューティングを行うことができます。

CLI を使用してサービスのバインディングを表示するには　コマンドプロンプトで入力します:

```
show service bindings <name>

show service bindings Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスのバインディングを表示するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. 詳細ペインで、バインディング情報を表示したいサービスを選択します。
3. 「アクション」タブで、「バインディングを表示」をクリックします。

仮想サーバーの統計情報を表示する

仮想サーバーのパフォーマンスを評価したり、問題をトラブルシューティングしたりするために、NetScaler アプライアンスに構成されている仮想サーバーの詳細を表示できます。すべての仮想サーバーの統計情報の概要を表示することも、仮想サーバーの名前を指定して、その仮想サーバーの統計情報のみを表示することもできます。次の詳細を表示できます。

- 名前
- IP アドレス
- ポート
- プロトコル
- 仮想サーバーの状態

- 受け取ったリクエストの割合
- ヒット率

CLI を使用して仮想サーバーの統計情報を表示するには　アプライアンスで現在構成されているすべての仮想サーバー、または単一の仮想サーバーの統計の概要を表示するには、コマンドプロンプトで次のように入力します。

```
stat lb vserver [`<name>`]
<!--NeedCopy-->
```

例:

```
stat lb vserver server-1
<!--NeedCopy-->
```

次の図は、サンプル統計を示しています。


```
> stat lbvserver
[
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
vserver1        10.102.20.200    80          SSL         DOWN      0/s

lb1               203.1.113.5   443         DTLS         DOWN      0/s

vicap                       *     0          TCP         DOWN      0/s

lbicap                2.2.3.4  1344          TCP         DOWN      0/s

app_...stest          0.0.0.0     0         HTTP         DOWN      0/s

app_...ttest          0.0.0.0     0         HTTP         DOWN      0/s

app_...fault          0.0.0.0     0         HTTP         DOWN      0/s

app_...test1          0.0.0.0     0         HTTP         DOWN      0/s

app_...1test          0.0.0.0     0         HTTP         DOWN      0/s

app_...fault          0.0.0.0     0         HTTP         DOWN      0/s

app_...est12          0.0.0.0     0         HTTP         DOWN      0/s

app_...sting          0.0.0.0     0         HTTP         DOWN      0/s

test                  2.2.2.2    80         HTTP         DOWN      0/s

shar...lt-lb          0.0.0.0     0         HTTP         DOWN      0/s

shar...es-lb          0.0.0.0     0         HTTP           UP      0/s

shar...es-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...ts-lb          0.0.0.0     0         HTTP           UP      0/s

shar...ns-lb          0.0.0.0     0         HTTP           UP      0/s

shar...as-lb          0.0.0.0     0         HTTP           UP      0/s

forward-vs            0.0.0.0     0          TCP         DOWN      0/s

tcpcs                 0.0.0.0     0          TCP         DOWN      0/s

test124               0.0.0.0     0          SSL         DOWN      0/s

testssl               0.0.0.0     0          SSL         DOWN      0/s
```


GUI を使用して仮想サーバの統計情報を表示するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 1 つの仮想サーバーの統計のみを表示する場合は、詳細ペインで、統計を表示する仮想サーバーを選択します。
3. 詳細ペインで、[統計] をクリックします。

サービスの統計情報を表示する

サービス統計を使用して、要求、応答、要求バイト、応答バイト、現在のクライアント接続、サージキュー内の要求、現在のサーバー接続などのレートを表示できます。

CLI を使用してサービスの統計情報を表示するには　コマンドプロンプトで入力します:

```
stat service <name>
<!--NeedCopy-->
```

例**:**

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスの統計情報を表示するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. 詳細ペインで、統計情報を表示したいサービス (Service-HTTP-1 など) を選択します。
3. [統計] をクリックします。統計が新しいウィンドウに表示されます。

仮想サーバーの負荷分散とサービスの状態

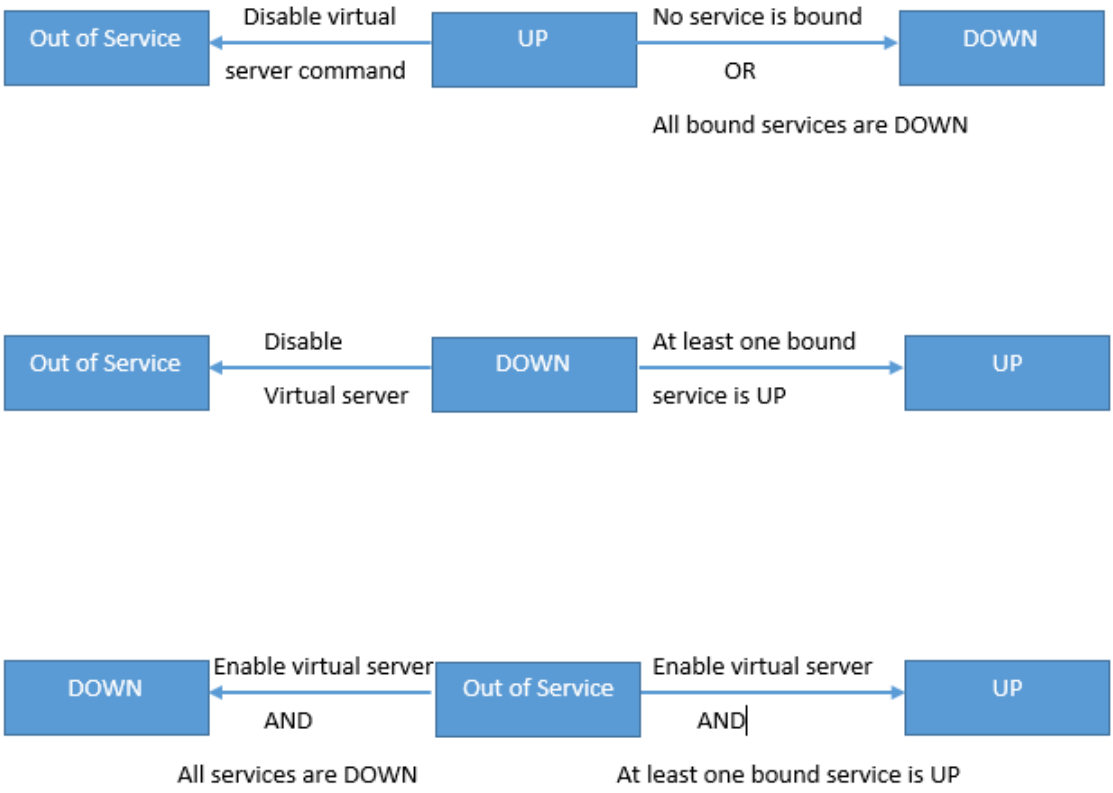
August 15, 2023

バックアップ仮想サーバを持たない負荷分散仮想サーバは、バインドされたサービスの状態および管理上の無効化かどうかに応じて、次の状態になります。

- **UP**: 仮想サーバーにバインドされているサービスの少なくとも 1 つが UP 状態です。
- **DOWN**: 仮想サーバーにバインドされているすべてのサービスが停止しているか、負荷分散機能が有効になっていません。
- アウトオブサービス **(OFS)**: 仮想サーバーを管理上無効にすると、OFS 状態になりますが、有効状態は DOWN です。管理者は、OFS 状態への移行を DOWN または UP 状態から、または OFS 状態から DOWN または UP 状態への移行を制御できます。

バックアップ仮想サーバが構成されていない場合、仮想サーバの状態と有効状態は同じです。ただし、バックアップ仮想サーバまたはバックアップ仮想サーバのチェーンが設定されている場合、有効状態は、プライマリ仮想サーバとバックアップ仮想サーバにバインドされているサービスの状態から得られます。チェーン内のいずれかのバックアップ仮想サーバが UP の場合、プライマリ仮想サーバにバインドされているすべてのサービスが DOWN であっても、プライマリ仮想サーバの有効な状態は UP になります。

次の図は、仮想サーバーが 1 つの状態から別の状態に移行する条件を示しています。



サービスは、次の状態を取ることができます。

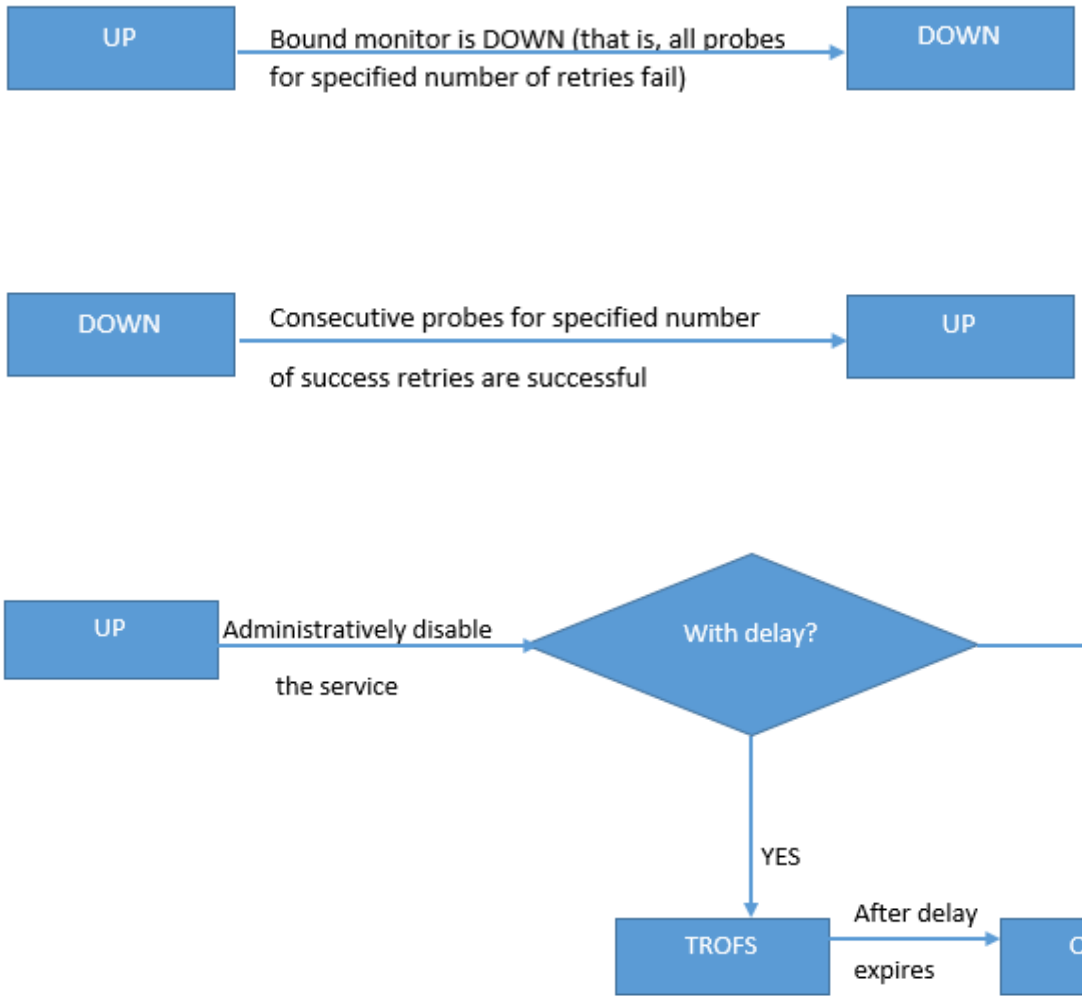
- **UP:** サービスにバインドされたすべてのモニターからのプローブが成功した場合。
- **DOWN**: サービスへの監視プローブが、設定された制限時間内に応答されない場合。
- **OUT OF SERVICE**: 管理上の理由でサービスを無効にした場合、またはサービスを正常にシャットダウンしたのにサービスへのアクティブなトランザクションがない場合
- **GOING OUT OF SERVICE**（**TROFS**）: 遅れてサービスを管理上無効にするか、サービスを正常にシャットダウンし、サービスへのアクティブなトランザクションがある場合。詳細については、「サービスのグレースフルシャットダウン」を参照してください。
- **DOWN WHEN GOING OUT OF SERVICE**（**TROFS_DOWN**）**[]** サービスがサービス停止状態にある間、監視プローブが失敗します。

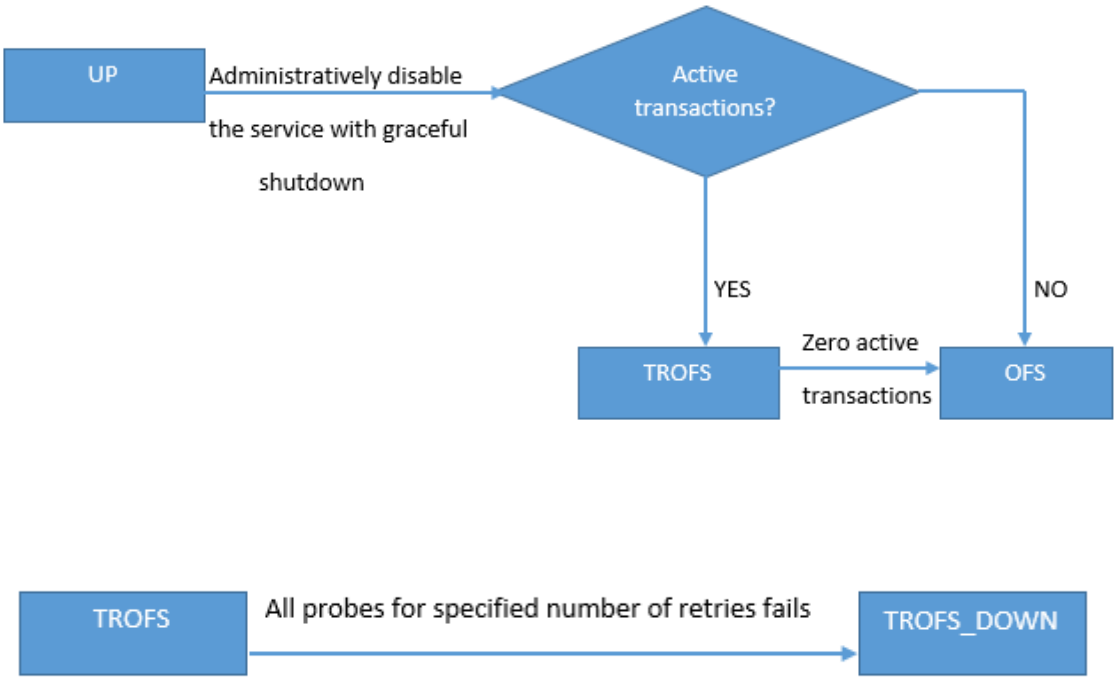
UP から OFS への移行プロセス中のサービスは、サービス終了状態です。ダウンから OFS に移行するサービスは、

DOWN WHEN GOING OUT OF SERVICE 状態になります。たとえば、サービスが DOWN の状態で遅延して無効にした場合、そのサービスは GING OUT OF SERVICE 時に DOWN に移行し、その後 OUT OF SERVICE 状態に移行します。サービスが UP で、遅延を伴って無効化すると、サービスは GOING OUT OF SERVICE に移行します。この間、サーバに対するモニタリングプローブが失敗すると、サービスは DOWN WHEN GOING OUT OF SERVICE に移行し、遅延時間が経過すると OFS 状態になります。

> 注
>
> 「HealthThreshold」パラメーターをゼロ以外の正の値に設定することで、バックアップ仮想サーバーへのスピルオーバーを構成できます。次に、プライマリ仮想サーバにバインドされている単一のサービスが DOWN WHEN GOING OUT OF SERVICE 状態に移行し、ヘルスのしきい値に達していない場合、プライマリ仮想サーバには DOWN とマークされ、新しい接続がバックアップ仮想サーバに送信されます。

次の図は、サービスが 1 つの状態から別の状態に移行する条件を示しています。





負荷分散プロファイルのサポート

August 15, 2023

負荷分散構成には多数のパラメータがあるため、複数の仮想サーバーで同じパラメータを設定すると面倒になることがあります。リリース 11.1 から、ロードバランシング (LB) プロファイルにより、この作業が簡単になりました。各仮想サーバーでこれらのパラメーターを設定する代わりに、プロファイルに負荷分散パラメーターを設定し、このプロファイルを仮想サーバーに関連付けることができるようになりました。

現在、LB プロファイルでは、次のパラメータがサポートされています。

- `HTTPonlyflag`—永続クッキーに HttpOnly 属性を含めます。HttpOnly 属性は、クッキーのスコープを HTTP 要求に制限し、クロスサイトスクリプティング攻撃のリスクを軽減するのに役立ちます。
- usesecuredPersistenceCookie：SHA2 ハッシュアルゴリズムを使用してパーシステンスクッキー値を暗号化します。
- `Cookiepassphrase`：セキュアパーシステンス Cookie 値の生成に使用するパスフレーズを指定します。
- DBS_LB-MySQL および MSSQL サービスタイプのデータベース固有のロードバランシングを有効にします。

- Cl_process_local：クラスタ内の仮想サーバ宛てのパケットは制御されません。単一パケット要求応答モードの場合、またはアップストリームデバイスが接続ベースの配信に適切な RSS を実行している場合に、オプションを有効にします。
- lbHashAlgorithm：次のハッシュベースのロードバランシング方式に使用するハッシュアルゴリズムを指定します。
  - URL ハッシュ方式
  - ドメインハッシュ法
  - 宛先 IP ハッシュ方式
  - ソース IP ハッシュ方式
  - 送信元 IP 宛先 IP ハッシュ方式
  - 送信元 IP 送信元ポートのハッシュ方式
  - コール ID ハッシュメソッド
  - トークン方式

  指定可能な値:DEFAULT、PRAC、JARH
  デフォルト値:DEFAULT
- lbHashFingers-ハッシュベースの LB メソッドの PRAC および JARH アルゴリズムで使用する指の数を指定します。フィンガーの数を増やすと、追加のメモリを犠牲にしてトラフィックの分散が向上します。

  デフォルト値:256
  最小値:1
  最大値:1024
- ProximityFromSelf-Enable を使用すると、クライアントの IP アドレスの代わりに Netscaler のループバック IP アドレスを使用して最も近いサーバーロケーションを取得し、静的近接負荷分散や GSLB 決定を行うことができます。

> 注
>
> DBS_LB パラメータと Cl_process_local パラメータは、仮想サーバーおよびプロファイルで設定できます。仮想サーバでこれらのパラメータを有効にし、この仮想サーバにプロファイルを設定すると、その仮想サーバの"show lb vserver"コマンドの出力にパラメータが無効として表示されます。プロファイルをチェックして、これらのパラメータの実際のステータスを確認します。さらに、プロファイルを仮想サーバーに設定してから設定解除すると、パラメータはその仮想サーバーのデフォルト値で設定されます。

CLI を使用して LB プロファイルを作成するには

コマンドプロンプトで入力します。

```
add lb profile <lbprofilename> -dbsLb ( ENABLED | DISABLED ) -
    processLocal ( ENABLED | DISABLED ) -httpOnlyCookieFlag ( ENABLED |
    DISABLED ) -cookiePassphrase -useSecuredPersistenceCookie ( ENABLED
```



```
      | DISABLED ) -lbHashAlgorithm <lbHashAlgorithm> -lbHashFingers <
      positive_integer>- proximityFromSelf <NO/YES>
<!--NeedCopy-->
```

例**:**

```
> sh lb profile p1
LB Profile name:  p1
DBS LB : DISABLED   Process Local: DISABLED
Persistence Cookie HttpOnly Flag: ENABLED
Use Encrypted Persistence Cookie: DISABLED
Proximity From Self: ENABLED
No of vservers bound: 0
Store MQTT clientid and username in transactional logs: NO
Hash LB algorithm used in LB decision: DEFAULT
Number of fingers for Hash LB algorithm: 256
Done

<!--NeedCopy-->
```

GUI を使用して LB プロファイルを作成するには

[システム] > [プロファイル] > [**LB** プロファイル] に移動し、プロファイルを追加します。

CLI を使用して LB プロファイルを LB 仮想サーバーに関連付けるには

コマンドプロンプトで入力します。

```
set lb vserver <name> -lbprofilename <string>
<!--NeedCopy-->
```

例

```
set lbvserver lbvip1 -lbprofile p1

Done

sh lb vserver lbvip1

lbvip1 (203.0.113.1:80) - HTTP       Type: ADDRESS
State: UP
Last state change was at Wed May 25 12:36:20 2016
Time since last state change: 0 days, 00:01:26.140
Effective State: UP  ARP:DISABLED
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: ENABLED
Port Rewrite : DISABLED
```



```
No. of Bound Services :  2 (Total)       2 (Active)
Configured Method: LEASTCONNECTION      BackupMethod: ROUNDROBIN
Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED  Push VServer:
Push Multi Clients: NO
Push Label Rule: none
L2Conn: OFF
Skip Persistency: None
Listen Policy: NONE
IcmpResponse: PASSIVE
RHIstate: PASSIVE
New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0
Mac mode Retain Vlan: DISABLED
DBS_LB: DISABLED
Process Local: DISABLED
Traffic Domain: 0
LB Profile: p1
Done
<!--NeedCopy-->
```

GUI を使用して LB プロファイルを LB 仮想サーバーに関連付けるには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバーを選択し、[編集] をクリックします。
3. 「詳細設定」で、「プロフィール」をクリックします。
4. **LB** プロファイルリストで、この仮想サーバーに関連付けるプロファイルを選択します。

GUI を使用して負荷分散プロファイルの Proximity from Self パラメータを設定するには

プロファイルがエンティティにアタッチされたときにそのエンティティのパラメータが有効になるように、プロファイルの Proximity from Self パラメータを設定します。

1. [システム] > [プロファイル] > [**LB** プロファイル] に移動します。
2. [追加] をクリックします。
3. 「自分」から「近さ」を選択します。
4. [**OK**] をクリックします。

負荷分散アルゴリズム

February 15, 2024

負荷分散アルゴリズムは、NetScaler アプライアンスが各クライアント要求をリダイレクトするサービスを選択するために使用する基準を定義します。負荷分散アルゴリズムが異なれば、使用する基準も異なります。たとえば、最小接続アルゴリズムはアクティブな接続が最も少ないサービスを選択し、ラウンドロビンアルゴリズムはアクティブなサービスの実行キューを維持し、各接続をキュー内の次のサービスに分散してから、そのサービスをキューの最後に送信します。

負荷分散アルゴリズムには、ウェブサイト上のトラフィックの処理に最適なものもあれば、DNS サーバーへのトラフィックの管理に最適なものもあります。また、電子商取引や会社の LAN や WAN で使用される複雑な Web アプリケーションの処理に最適なものもあります。次の表は、NetScaler アプライアンスがサポートする各負荷分散アルゴリズムと、それぞれの動作方法の簡単な説明を示しています。

名前	以下に基づくサーバー選択
LEASTCONNECTION	現在、クライアント接続が最も少ないサービスはどれか。これはデフォルトの負荷分散アルゴリズムです。
ROUNDROBIN	サービスのリストの一番上にあるのはどのサービスか。そのサービスを接続用に選択すると、リストの一番下に移動します。
LEASTRESPONSETIME	現在、応答時間が最も速い負荷分散サーバーはどれですか。
URLHASH	送信先 URL のハッシュ。
DOMAINHASH	宛先ドメインのハッシュ。
DESTINATIONIPHASH	宛先 IP アドレスのハッシュ。
SOURCEIPHASH	送信元 IP アドレスのハッシュ。
SRCIPDESTIPHASH	送信元と宛先の IP アドレスのハッシュ。
CALLIDHASH	SIP ヘッダーのコール ID のハッシュ。
SRCIPSRCPORTHASH	クライアントの IP アドレスとポートのハッシュ。
LEASTBANDWIDTH	現在、帯域幅の制約が最も少ないサービスはどれか。
LEASTPACKETS	現在、受信しているパケット数が最も少ないサービスはどれですか。
CUSTOMLOAD	負荷モニターからのデータ。
TOKEN	設定されたトークン。
LRTM	アクティブな接続が最も少なく、平均応答時間が最も短い。
静的近接度	近接条件に最も適したサービス。
最小リクエスト	現在、クライアントからのリクエストが最も少ないのはどのサービスか。

名前	以下に基づくサーバー選択

NetScaler アプライアンスは、負荷分散を行うサービスのプロトコルに応じて、クライアントとサーバー間の各接続が異なる時間間隔で持続するように設定します。これは負荷分散の細分性と呼ばれ、リクエストベース、接続ベース、時間ベースの 3 つのタイプがあります。次の表では、各タイプの粒度とその用途について説明します。

粒度	負荷分散サービスのタイプ	指定
リクエストベース	HTTP または HTTPS	TCP 接続とは無関係に、HTTP リクエストごとに新しいサービスが選択されます。すべての HTTP 要求と同様に、Web サーバーが要求を処理すると、接続は閉じられます。
コネクションベース	HTTP 以外の TCP および TCP ベースのプロトコル	新しい TCP 接続ごとにサービスが選択されます。接続は、サービスまたはクライアントによって終了されるまで持続します。
時間ベース	UDP とその他の IP プロトコル	UDP パケットごとに新しいサービスが選択されます。サービスを選択すると、指定した期間、サービスとクライアントの間にセッションが作成されます。この時間が経過すると、セッションは削除され、追加のパケットが同じクライアントから送信された場合でも、新しいサービスが選択されます。

仮想サーバーの起動時、または仮想サーバーの状態が変わるたびに、仮想サーバーは最初にラウンドロビン方式を使用してクライアント要求を物理サーバーに分散できます。このタイプの配布はスタートアップラウンドロビンと呼ばれ、最初の要求が処理されるときに 1 台のサーバーに不必要な負荷がかかることを防ぐのに役立ちます。起動時にラウンドロビン方式を使用した後、仮想サーバーは仮想サーバーで指定された負荷分散方式に切り替わります。

スタートアップ RR ファクターは次のように機能します。

- Startup RR Factor がゼロに設定されている場合、アプライアンスは要求レートに応じて指定された負荷分散方法に切り替わります。
- Startup RR Factor がゼロ以外の数値の場合、アプライアンスは指定された数のリクエストに対してラウンドロビン方式を使用してから、指定された負荷分散方式に切り替えます。

- デフォルトでは、スタートアップ RR 係数はゼロに設定されています。

注: 個々の仮想サーバーの起動 RR 係数は設定できません。指定した値は、NetScaler アプライアンス上のすべての仮想サーバーに適用されます。

CLI を使用して起動時のラウンドロビン係数を設定するには

コマンドプロンプトで入力します:

```
set lb parameter -startupRRFactor <positive_integer>
```

例

```
set lb parameter -startupRRFactor 25000
```

GUI を使用して起動時のラウンドロビン係数を設定するには

1. [トラフィック管理] **>** [ロードバランシング] **>** [ロードバランシングパラメータの設定] に移動し、[スタートアップ RR 係数] を設定します。

最小接続方式

February 15, 2024

仮想サーバーが最も少ない接続負荷分散アルゴリズム (または方法) を使用するように構成されている場合、アクティブな接続が最も少ないサービスを選択します。これは、ほとんどの状況で最高のパフォーマンスを提供するため、デフォルトの方法です。

TCP、HTTP、HTTPS、および SSL_TCP サービスの場合、NetScaler アプライアンスの既存の接続のリストには次の接続タイプが含まれます。

- サービスへのアクティブな接続。クライアントが仮想サーバーに送信し、仮想サーバーがサービスに転送した要求を表す接続。HTTP および HTTPS サービスの場合、アクティブな接続とは、まだ応答を受け取っていない HTTP または HTTPS リクエストのみを指します。
- サージキューで待機中の接続。サージキューで待機していて、まだサービスに転送されていない仮想サーバーへの接続。次のいずれかの理由により、いつでも接続がサージキューに蓄積する可能性があります。
  - サービスには接続制限があり、負荷分散構成のすべてのサービスがその制限に達しています。
  - サージ保護機能が構成され、仮想サーバーへの要求が急増したことで有効化されました。
  - 負荷分散されたサーバーが内部制限に達したため、新しい接続は開かれません。(たとえば、Apache サーバーの接続上限に達したとします。)

仮想サーバーが最も少ない接続方法を使用する場合、待機中の接続は特定のサービスに属していると見なします。したがって、それらのサービスへの新しい接続は開かれません。

UDP サービスの場合、最小接続アルゴリズムで考慮される接続には、クライアントとサービス間のすべてのセッションが含まれます。これらのセッションは論理的で時間ベースのエンティティです。セッションの最初の UDP パケットが到着すると、NetScaler アプライアンスは送信元の IP アドレスとポートと宛先の IP アドレスとポートとの間にセッションを作成します。

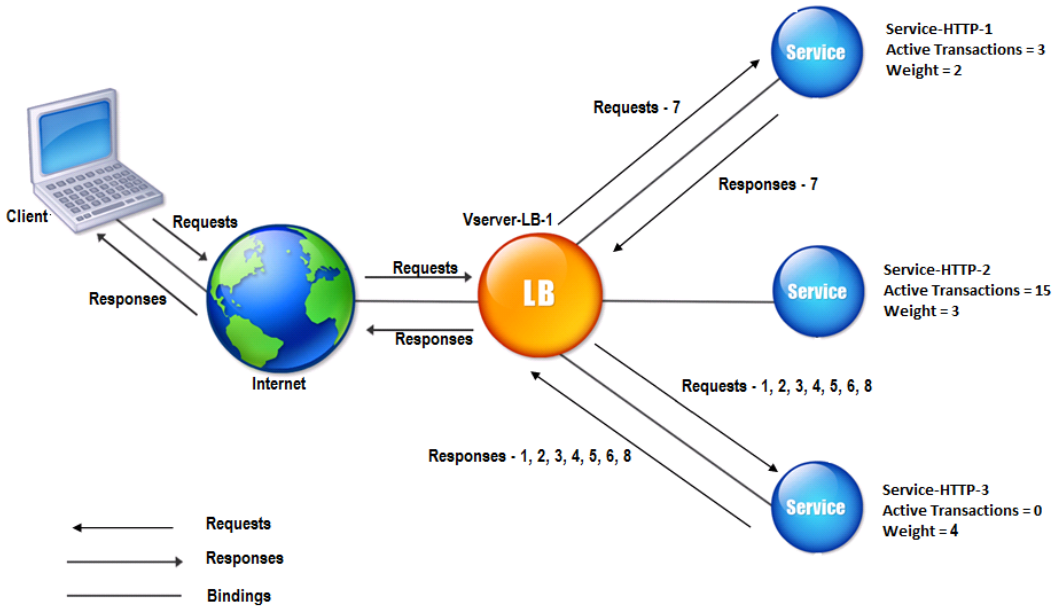
リアルタイムストリーミングプロトコル（RTSP）接続の場合、NetScaler アプライアンスはアクティブな制御接続の数を使用して、RTSP サービスへの最小接続数を決定します。

次の例は、仮想サーバーが最小接続方法を使用して負荷分散用のサービスを選択する方法を示しています。次の 3 つのサービスを検討してください。

- サービス-HTTP-1 は 3 つのアクティブなトランザクションを処理しています。
- サービス HTTP-2 は 15 件のアクティブなトランザクションを処理しています。
- Service-HTTP-3 はアクティブなトランザクションを処理していません。

次の図は、NetScaler アプライアンスが最小接続方法を使用して受信リクエストを転送する方法を示しています。

図 1：最小接続数の負荷分散方式の仕組み



この図では、仮想サーバーは、アクティブなトランザクションが最も少ないサーバーを選択して、受信接続ごとにサービスを選択します。

接続は次のように転送されます。

- Service-HTTP-3 は、アクティブなトランザクションを処理していないため、最初の要求を受信します。

  注: アクティブなトランザクションがないサービスが最初に選択されます。

- Service-HTTP-3 は 2 番目と 3 番目の要求を受信します。これは、サービスのアクティブなトランザクション数が次に最小であるためです。
- Service-HTTP-1 は 4 番目の要求を受信します。Service-HTTP-1 と Service-HTTP-3 のアクティブトランザクション数が同じであるため、仮想サーバはラウンドロビン方式を使用してそれらのいずれかを選択します。
- Service-HTTP-3 は 5 番目の要求を受信します。
- Service-HTTP-1 は、Service-HTTP-1 と Service-HTTP-3 の両方が Service-HTTP-2 と同じ数の要求を処理するまで、6 番目の要求を受信します。次に、NetScaler ADC アプライアンスは、負荷が最も低いサービスであるか、またはラウンドロビンキューでターンが起動したときに、Service-HTTP-2 への要求の転送を開始します。

  メモ:Service-HTTP-2 への接続が終了すると，他の 2 つのサービスに 15 個のアクティブなトランザクションがある前に，新しい接続を取得することがあります。

次の表では、前述の 3 サービス負荷分散設定での接続の分散方法について説明します。

着信接続	サービス選択済み	現在のアクティブ接続数	注釈
Request-1	サービス-HTTP-3; (N = 0)	1	Service-HTTP-3 はアクティブな接続が最も少ないです。
Request-2	サービス-HTTP-3; (N = 1)	2	Service-HTTP-3 はアクティブな接続が最も少ないです。
Request-3	Service-HTTP-3; (N = 2)	3	-
Request-4	Service-HTTP-1; (N = 3)	4	サービス HTTP-1 とサービス HTTP-3 のアクティブな接続数は同じです。
Request-5	Service-HTTP-3; (N = 3)	4	サービス HTTP-1 とサービス HTTP-3 のアクティブな接続数は同じです。
Request-6	サービス-HTTP-1; (N = 4)	5	-
Request-7	Service-HTTP-3; (N = 4)	5	-
Request-8	Service-HTTP-1; (N = 5)	6	-

Service-HTTP-2 は、アクティブなトランザクションが完了して現在の接続が終了したとき、または他のサービス

(Service-HTTP-1 と Service-HTTP-3）の接続がそれぞれ 15 個以上ある場合に、負荷分散の対象として選択されます。

NetScaler アプライアンスは、サービスに重みを割り当てる場合に最小限の接続方法を使用することもできます。次の式の値 (Nw) を使用してサービスを選択します:

Nw = (アクティブなトランザクションの数) * (10000 /重量)

次の例は、NetScaler アプライアンスがサービスに重みを割り当てるときに、最小接続方法を使用して負荷分散用のサービスを選択する方法を示しています。前の例では、サービス HTTP-1 に 2 の重みが割り当てられ、サービス HTTP-2 には 3 の重みが割り当てられ、サービス HTTP-3 には 4 の重みが割り当てられているとします。接続は次のように転送されます。

- Service-HTTP-3 は、サービスがアクティブなトランザクションを処理していないため、最初のを受信します。

> 注:
>
> サービスがアクティブなトランザクションを処理していない場合、NetScaler アプライアンスは各サービスに割り当てられた重みに関係なくラウンドロビン方式を使用します。

- Service-HTTP-3 は、サービスの Nw 値が最も低いため、2 番目、3 番目、4 番目、5 番目、6 番目のリクエストを受け取ります。
- サービス-HTTP-1 は 7 番目のリクエストを受け取ります。Service-HTTP-1 と Service-HTTP-3 は同じ Nw 値を持つため、アプライアンスはラウンドロビン方式でロードバランシングを実行します。したがって、Service-HTTP-3 は 8 番目の要求を受信します。

次の表では、前述の 3 サービス負荷分散設定での接続の分散方法について説明します。

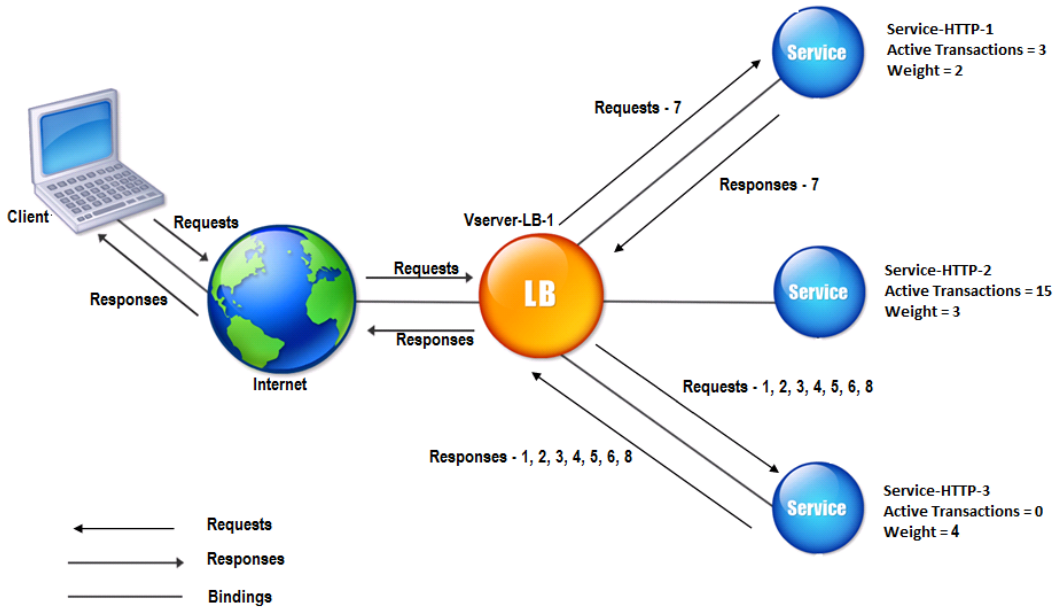
リクエストを受け取りました	サービス選択済み	現在の新規 (アクティブなトランザクションの数) * (10000 /重量) の値	注釈
Request-1	サービス-HTTP-3; (新規 = 0)	Nw = 2500	サービス HTTP-3 の Nw 値は最も低いです。
Request-2	Service-HTTP-3; (New = 2500)	Nw = 5000	
Request-3	Service-HTTP-3; (Nw = 5000)	Nw = 7500	
Request-4	Service-HTTP-3; (New = 7500)	Nw = 10000	
Request-5	Service-HTTP-3; (Nw = 10000)	Nw = 12500	

リクエストを受け取りました	サービス選択済み	現在の新規 (アクティブなトランザクションの数) * (10000 /重量) の値	注釈
Request-6	Service-HTTP-3; (Nw = 12500)	Nw = 15000	
Request-7	Service-HTTP-1; (Nw = 15000)	Nw = 20000	サービス HTTP-1 とサービス HTTP-3 の Nw 値は同じ
Request-8	Service-HTTP-3; (Nw = 15000)	Nw = 17500	

サービス HTTP-2 は、アクティブなトランザクションが完了したとき、または他のサービス (サービス HTTP-1 と Service-HTTP-3) の Nw 値が 50000 になったときに、ロードバランシングの対象として選択されます。

次の図は、サービスに重みを割り当てるときに、NetScaler アプライアンスが最も少ない接続方法を使用する方法を示しています。

図 2: 重み付け時における最小接続負荷分散手法の仕組み



最小接続方法を構成するには、ポリシーを含まない負荷分散方式の構成を参照してください。

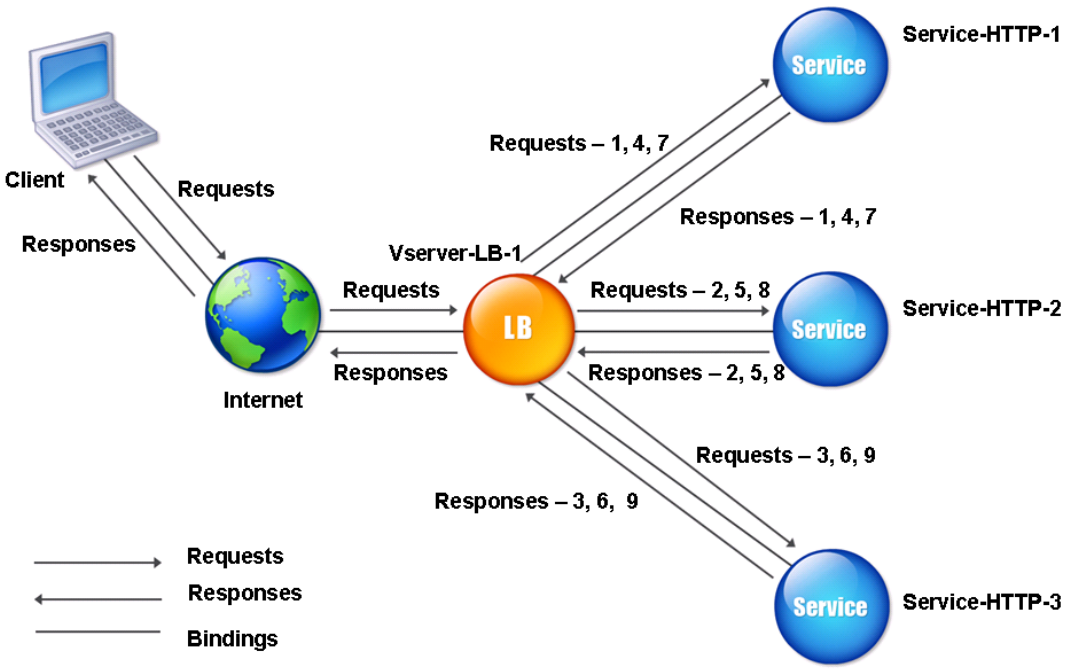
ラウンドロビン方式

August 15, 2023

負荷分散仮想サーバーがラウンドロビン方式を使用するように構成されている場合、それにバインドされているサービスのリストが継続的にローテーションされます。仮想サーバーは、要求を受信すると、リスト内の最初のサービスに接続を割り当て、そのサービスをリストの一番下に移動します。

次の図は、NetScaler ADC アプライアンスが 3 つの負荷分散サーバーとそれに関連するサービスを含む負荷分散設定でラウンドロビン方式を使用する方法を示しています。

図 1：ラウンドロビンロードバランシング方式の動作



各サービスに異なる重みを割り当てると、NetScaler ADC アプライアンスは着信接続の加重ラウンドロビン分散を実行します。これは、より低い重み付けされたサービスを適切な間隔でスキップすることによって行われます。

たとえば、3 つのサービスを含む負荷分散設定があるとします。サービス HTTP-1 をウェイト 2 に、サービス HTTP-2 をウェイト 3 に、サービス HTTP-3 をウェイト 4 に設定します。サービスは vServer-LB-1 にバインドされ、ラウンドロビン方式を使用するように構成されています。この設定では、受信リクエストは次のように配信されます。

- サービス HTTP-1 は最初のリクエストを受け取ります。

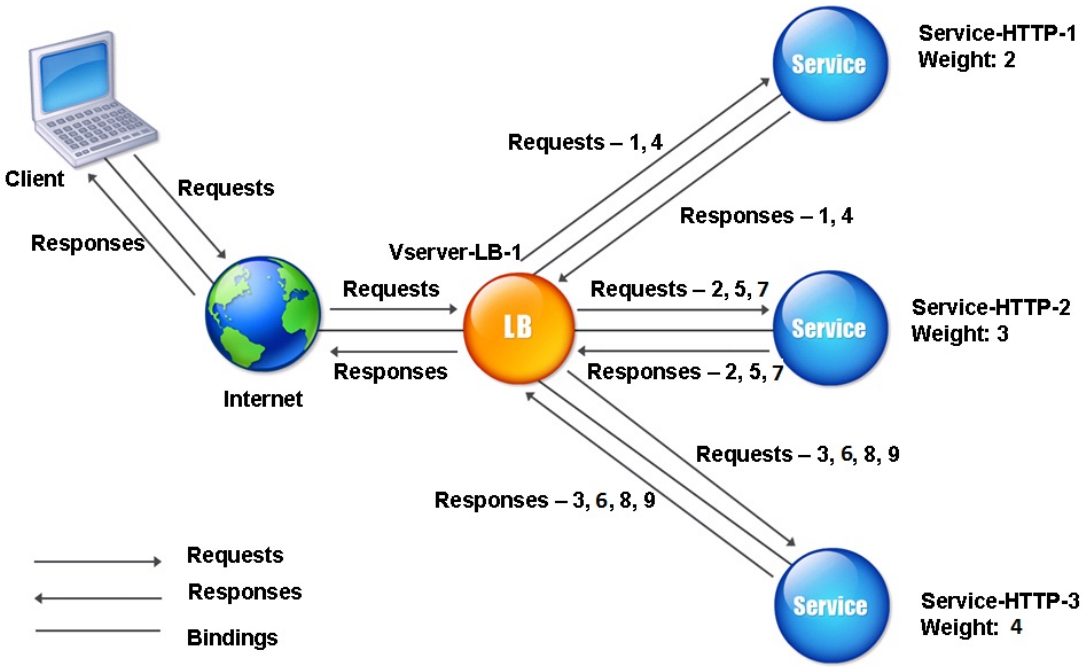
- サービス HTTP-2 は 2 番目のリクエストを受け取ります。
- サービス HTTP-3 は 3 番目のリクエストを受け取ります。
- Service-HTTP-1 は 4 番目の要求を受信します。
- サービス HTTP-2 は 5 番目のリクエストを受け取ります。
- サービス HTTP-3 は 6 番目のリクエストを受け取ります。
- サービス HTTP-2 は 7 番目のリクエストを受け取ります。
- Service-HTTP-3 は、8 番目と 9 番目のリクエストの両方を受け取ります。

> 注記:
>
> また、複数のサービスが同じサーバーを使用してサーバーに過負荷がかかるのを防ぐために、サービスに重みを設定することもできます。

その後、同じパターンを使用して新しいサイクルが開始されます。

次の図は、加重ラウンドロビン方式を示しています。

図 2: ラウンドロビンロードバランシング方式による加重サービスのサポート方法



ラウンドロビン方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

最小応答時間方式

August 15, 2023

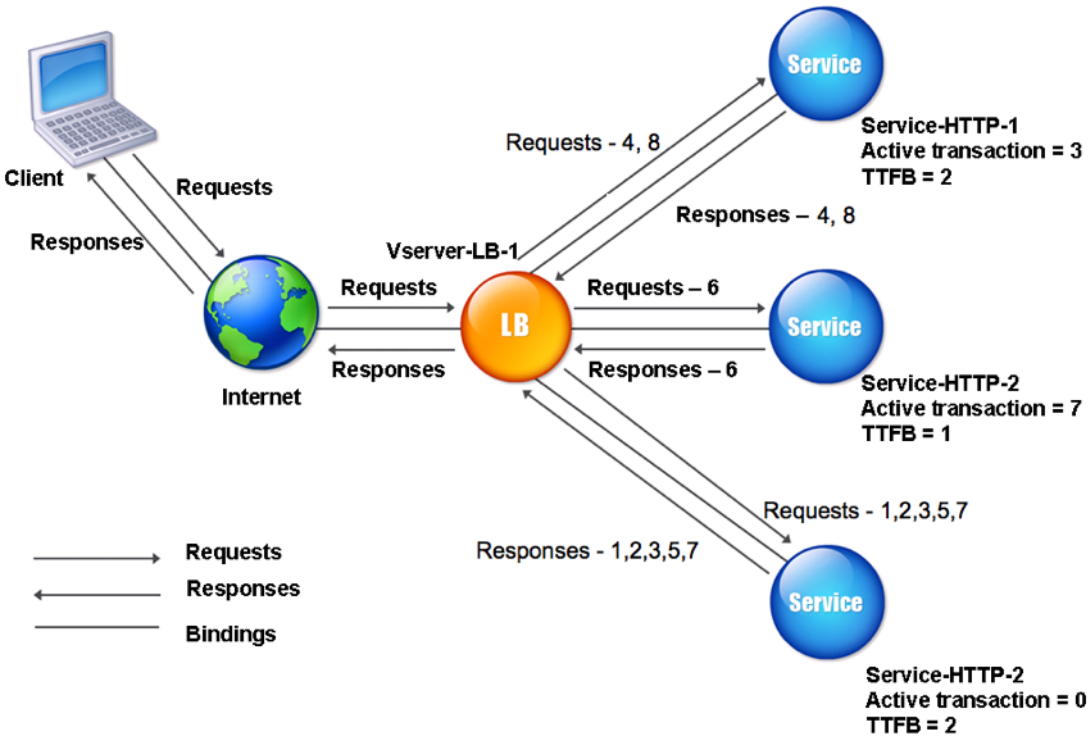
負荷分散仮想サーバーが応答時間が最短の方法を使用するように構成されている場合、アクティブな接続が最も少なく、平均応答時間が最も短いサービスを選択します。この方法は、HTTP および SSL (Secure Sockets Layer) 負荷分散仮想サーバーでのみ構成できます。応答時間 (Time to First Byte、または TTFB とも呼ばれる) は、要求パケットをサービスに送信してからサービスから最初の応答パケットを受信するまでの時間間隔です。NetScaler アプライアンスは、応答コード 200 を使用して TTFB を計算します。

次の例は、仮想サーバーが最小応答時間の方法を使用して負荷分散用のサービスを選択する方法を示しています。次の 3 つのサービスを検討してください。

- Service-HTTP-1 は 3 つのアクティブなトランザクションを処理し、TTFB は 2 秒です。
- Service-HTTP-2 は 7 つのアクティブなトランザクションを処理し、TTFB は 1 秒です。
- Service-HTTP-3 はアクティブなトランザクションを一切処理しておらず、TTFB は 2 秒です。

次の図は、NetScaler アプライアンスが最短応答時間方式を使用して接続を転送する方法を示しています。

図 1：最小応答時間のロードバランシング方式の動作



仮想サーバは、各サービスのアクティブなトランザクション数に TTFB を掛けてから、最も低い結果を持つサービスを選択することによって、サービスを選択します。上記の例では、仮想サーバーは次のように要求を転送します。

- Service-HTTP-3 は、サービスがアクティブなトランザクションを処理していないため、最初の要求を受信します。
- Service-HTTP-3 は、2 番目と 3 番目の要求も受信します。これは、3 つのサービスの中で最も低い結果になるためです。
- Service-HTTP-1 は 4 番目の要求を受信します。サービス HTTP-1 とサービス-HTTP-3 の結果は同じなので、NetScaler アプライアンスはラウンドロビン方式を適用してどちらかを選択します。
- Service-HTTP-3 は 5 番目の要求を受信します。
- Service-HTTP-2 は 6 番目のリクエストを受信します。これは、この時点では最も低い結果が得られるためです。
- この時点では、Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 はすべて同じ結果になるため、アプライアンスはラウンドロビン方式に切り替わり、その方式を使用して接続を配信し続けます。

次の表では、前述の 3 サービス負荷分散設定での接続の分散方法について説明します。

リクエストを受け取りました	サービス選択済み	現在の N 値 (アクティブなトランザクションの数 * TTFB)	注釈
Request-1	サービス-HTTP-3; (N = 0)	N = 2	Service-HTTP-3 は最小 N 値を持ちます。
Request-2	Service-HTTP-3; (N = 2)	N = 4	Service-HTTP-3 は最小 N 値を持ちます。
Request-3	Service-HTTP-3; (N = 4)	N = 6	Service-HTTP-3 は最小 N 値を持ちます。
Request-4	サービス-HTTP-1; (N = 6)	N = 8	サービス HTTP-1 とサービス HTTP-3 の N 値は同じアプライアンスはラウンドロビン方式を使用してリクエストを配信します。
Request-5	サービス-HTTP-3; (N = 6)	N = 8	サービス HTTP-1 とサービス HTTP-3 の N 値は同じ
Request-6	サービス-HTTP-2; (N = 7)	N = 8	Service-HTTP-2 は最小 N 値を持ちます。

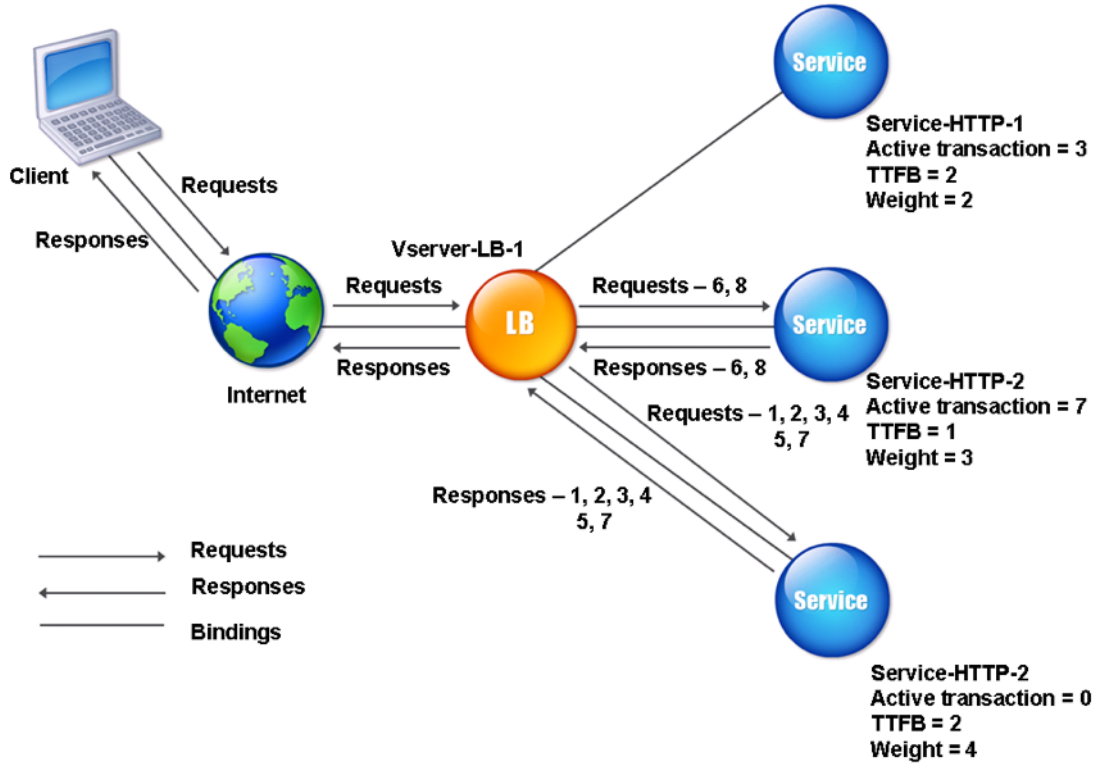
リクエストを受け取りました	サービス選択済み	現在の N 値 (アクティブなトランザクションの数 * TTFB)	注釈
Request-7	サービス-HTTP-3; (N = 8)	N = 10	サービス HTTP-1、サービス HTTP-2、サービス HTTP-3 の N 値は同じ。NetScaler アプライアンスは、ラウンドロビン方式を使用してリクエストを配信します。
Request-8	サービス-HTTP-1; (N = 8)	N = 10	サービス HTTP-1 と Service-HTTP-2 の N 値は同じ。アプライアンスはラウンドロビン方式を使用して要求を配信します。

アクティブなトランザクションが完了したとき、またはその N 値が他のサービス（Service-HTTP-2 と Service-HTTP-3）よりも小さいときに、サービス HTTP-1 がロードバランシングの対象として再び選択されます。

ウェイト割り当て時のサービスの選択

次の図は、重みが割り当てられたときに NetScaler アプライアンスが最小応答時間方式を使用する方法を示しています。

図 2：重みが割り当てられている場合の最小応答時間負荷分散方法の仕組み



仮想サーバーは、次の式の値 (Nw) を使用してサービスを選択します。

Nw = (N) * (10000/weight)。ここで N = (アクティブなトランザクション数 * TTFB)

Service-HTTP-1 に重みが 2、Service-HTTP-2 に重みが 3、Service-HTTP-3 に重みが 4 が割り当てられているとします。

NetScaler アプライアンスは、次のようにリクエストを配信します。

- Service-HTTP-3 は、アクティブなトランザクションを処理していないため、最初の要求を受信します。

  サービスがアクティブなトランザクションを処理していない場合、アプライアンスは割り当てられている重みに関係なくトランザクションを選択します。
- Service-HTTP-3 は、Nw の値が最小であるため、2 番目、3 番目、4 番目、5 番めの要求を受信します。
- Service-HTTP-2 は 6 番目のリクエストを受信します。これは、このサービスの Nw 値が最も低いためです。
- Service-HTTP-3 は 7 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。
- Service-HTTP-2 は 8 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。

Service-HTTP-1 は重みが最も低く、したがって Nw 値が最大であるため、仮想サーバはロードバランシング用に選択しません。

次の表では、前述の 3 サービス負荷分散設定での接続の分散方法について説明します。

リクエストを受け取りました	サービス選択済み	現在の新しい値 = (N) * (10000/重量)	注釈
Request-1	サービス-HTTP-3; (新規 = 0)	Nw = 5000	サービス HTTP-3 の Nw 値は最も低いです。
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 10000	サービス HTTP-3 の Nw 値は最も低いです。
Request-3	Service-HTTP-3; (Nw = 10000)	Nw = 15000	サービス HTTP-3 の Nw 値は最も低いです。
Request-4	Service-HTTP-3; (Nw = 15000)	Nw = 20000	サービス HTTP-3 の Nw 値は最も低いです。
Request-5	Service-HTTP-3; (New = 20000)	Nw = 25000	サービス HTTP-3 の Nw 値は最も低いです。
Request-6	Service-HTTP-2; (New = 23333.34)	Nw = 26666.67	サービス HTTP-2 の Nw 値は最も低いです。
Request-7	Service-HTTP-3; (New = 25000)	Nw = 30000	サービス HTTP-3 の Nw 値は最も低いです。
Request-8	Service-HTTP-2; (Nw = 26666.67)	Nw = 30000	サービス HTTP-2 の Nw 値は最も低いです。

Service-HTTP-1 は、アクティブなトランザクションが完了したとき、または Nw 値が他のサービス（Service-HTTP-2 および Service-HTTP-3）よりも小さい場合に、ロードバランシングの対象として選択されます。

CLI を使用して応答時間を最小限に抑えるロードバランシング方法を設定するには

コマンドプロンプトで、次のように入力します。

```
set lb vserver <name> -lbMethod LEASTRESPONSETIME
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -lbMethod LEASTRESPONSETIME
<!--NeedCopy-->
```

GUI を使用して最小応答時間のロードバランシング方法を設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。

2. [詳細設定] で、[**LEASTRESPONSETIME**] を選択します。

モニタの構成の詳細については、「負荷分散セットアップでのモニタの設定」を参照してください。

LRTM 方式

August 15, 2023

注：LRTM は、モニターを使用する最小応答時間方式（LRTM）の略です。

負荷分散仮想サーバーが LRTM 方式を使用するように構成されている場合、既存の監視インフラストラクチャを使用して最速の応答時間を実現します。次に、負荷分散仮想サーバーは、アクティブなトランザクションの数が最も少なく、応答時間が最も短いサービスを選択します。LRTM 方式を使用する前に、アプリケーション固有のモニターを各サービスにバインドし、これらのモニターで LRTM モードを有効にする必要があります。次に、NetScaler アプライアンスは、監視プローブから計算した応答時間に基づいて負荷分散を決定します。

LRTM メソッドを使用して、HTTP 以外のサービスと非 HTTPS サービスの負荷分散を行うこともできます。この方法は、複数のモニターがサービスにバインドされている場合にも使用できます。各モニターは、バインドされているサービスに対して測定するプロトコルを使用して応答時間を決定します。次に、仮想サーバは結果を平均化して、そのサービスの平均応答時間を計算します。

次の表に、さまざまなモニタの応答時間の計算方法をまとめます。

監視	応答時間の計算
PING	ICMP エコー要求と ICMP エコー応答の時間差。
TCP	SYN リクエストと SYN+ACK レスポンスの時間差。
HTTP	HTTP リクエスト (TCP 接続が確立された後) と HTTP レスポンスの時間差。
TCP-ECV	データ送信文字列が送信されてから、データ受信文字列が返される時間の差。送受信文字列を使用しない TCP-ECV モニタの設定が正しくないと見なされます。
HTTP-ECV	HTTP リクエストと HTTP レスポンスの時間差。
UDP-ECV	UDP の送信文字列と受信文字列の間の時間差。受信文字列がない UDP-ECV モニタは、正しくない構成と見なされます。
DNS	DNS クエリと DNS 応答の時間差。
TCPS	SYN リクエストと SSL ハンドシェイク完了の時間差。
FTP	ユーザー名の送信とユーザー認証の完了との時間差。

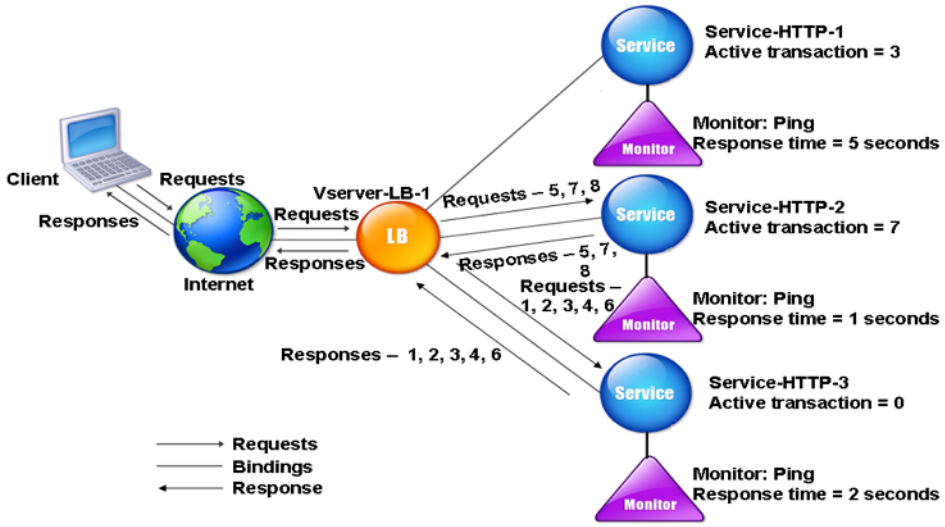
監視	応答時間の計算
HTTPS (HTTPS 要求を監視する)	時差は HTTP モニタの場合と同じです。
HTTPS-ECV（HTTPS 要求の監視）	時差は HTTP-ECV モニタの場合と同じです
USER	要求がディスパッチャーに送信された時刻とディスパッチャー応答が受信された時刻との時間差。

次の例は、NetScaler アプライアンスが LRTM 方式を使用して負荷分散用のサービスを選択する方法を示しています。次の 3 つのサービスを検討してください。

- Service-HTTP-1 は 3 つのアクティブなトランザクションを処理しており、応答時間は 5 秒です。
- Service-HTTP-2 は 7 つのアクティブなトランザクションを処理しており、応答時間は 1 秒です。
- Service-HTTP-3 はアクティブなトランザクションを処理しておらず、応答時間は 2 秒です。

次の図は、NetScaler アプライアンスがリクエストを転送するときに実行するプロセスを示しています。

図 1: LRTM メソッドの仕組み



仮想サーバーは、次の式で値 (N) を使用してサービスを選択します。

N = (アクティブなトランザクションの数 x モニターによって決定される応答時間)

仮想サーバーは、次のようにリクエストを送信します。

- Service-HTTP-3 は、このサービスはアクティブなトランザクションを処理していないため、最初の要求を受信します。

- Service-HTTP-3 は、2 番目、3 番目、4 番目の要求を受信します。これは、このサービスの N 値が最も小さいためです。
- Service-HTTP-2 は、5 番目の要求を受信します。これは、このサービスの N 値が最小であるためです。
- 現在、Service-HTTP-2 と Service-HTTP-3 の両方が同じ N 値を持つため、NetScaler ADC アプライアンスはラウンドロビン方式に切り替わります。したがって、Service-HTTP-3 は 6 番目の要求を受信します。
- Service-HTTP-2 は、7 番目と 8 番目の要求を受信します。これは、このサービスの N 値が最も小さいためです。

Service-HTTP-1 は、他の 2 つのサービスと比べて負荷が大きい（N 値が最も大きい）ため、負荷分散の対象にはなりません。ただし、Service-HTTP-1 がアクティブなトランザクションを完了すると、NetScaler アプライアンスは再びそのサービスを負荷分散の対象と見なします。

次の表は、サービスの N の計算方法をまとめたものです。

リクエストを受け取りました	サービス選択済み	現在の N 値 (アクティブなトランザクションの数 * TTFB)	注釈
Request-1	サービス-HTTP-3; (N = 0)	N = 2	Service-HTTP-3 は最小 N 値を持ちます。
Request-2	Service-HTTP-3; (N = 2)	N = 4	Service-HTTP-3 は最小 N 値を持ちます。
Request-3	Service-HTTP-3; (N = 4)	N = 6	Service-HTTP-3 は最小 N 値を持ちます。
Request-4	サービス-HTTP-3; (N = 6)	N = 8	Service-HTTP-3 は最小 N 値を持ちます。
Request-5	サービス-HTTP-2; (N = 7)	N = 8	Service-HTTP-2 は最小 N 値を持ちます。
Request-6	サービス-HTTP-3; (N = 8)	N = 10	サービス HTTP-2 とサービス HTTP-3 の N 値は同じ。NetScaler アプライアンスがラウンドロビン方式に切り替わり、Service-HTTP-3 を選択する
Request-7	Service-HTTP-2; (N = 8)	N = 9	Service-HTTP-2 は最小 N 値を持ちます。
Request-8	Service-HTTP-2; (N = 9)	N = 10	Service-HTTP-2 は最小 N 値を持ちます。

アクティブなトランザクションが完了したとき、またはその N 値が他のサービス（Service-HTTP-2 と Service-HTTP-3）よりも小さいときに、サービス HTTP-1 がロードバランシングの対象として再び選択されます。

ウェイト割り当て時のサービスの選択

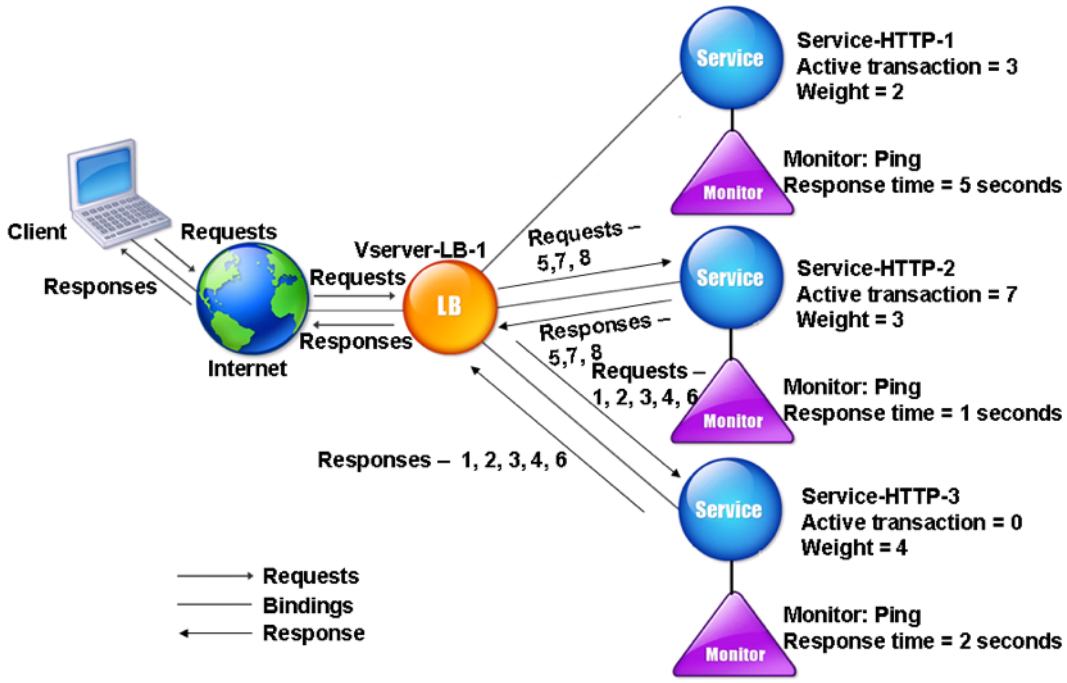
NetScaler アプライアンスは、サービスに異なる重みが割り当てられている場合は、アクティブなトランザクション数、応答時間、および重みを使用して負荷分散も実行します。NetScaler アプライアンスは、次の式の値（Nw）を使用してサービスを選択します。

Nw = (N) * (10000 /重量)

ここで N = (アクティブなトランザクションの数 x モニターによって決定される応答時間)

次の図は、重みが割り当てられるときに仮想サーバーが LRTM 方式を使用する方法を示しています。

図 2：重みが割り当てられている場合の最小応答時間負荷分散方法の仕組み



この例では、Service-HTTP-1 に重みが 2、Service-HTTP-2 に重みが 3、Service-HTTP-3 に重みが 4 が割り当てられているとします。

NetScaler ADC アプライアンスは、次のようにリクエストを送信します。

- Service-HTTP-3 は、アクティブなトランザクションを処理していないため、最初の要求を受信します。
- Service-HTTP-3 は、Nw の値が最小であるため、2 番目、3 番目、4 番目、5 番めの要求を受信します。
- Service-HTTP-2 は 6 番目のリクエストを受信します。これは、このサービスの Nw 値が最も低いためです。
- Service-HTTP-3 は 7 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。

- Service-HTTP-2 は 8 番目のリクエストを受信します。これは、このサービスの Nw 値が最も低いためです。

Service-HTTP-1 は重みが最も低く、Nw 値が最も高いため、NetScaler アプライアンスは負荷分散の対象として選択しません。

次の表は、さまざまなモニターでの Nw の計算方法をまとめたものです。

リクエストを受け取りました	サービス選択済み	Current Nw Value (N) * (10000 / Weight)	注釈
Request-1	サービス-HTTP-3; (新規 = 0)	Nw = 5000	サービス HTTP-3 の Nw 値は最も低いです。
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 10000	サービス HTTP-3 の Nw 値は最も低いです。
Request-3	Service-HTTP-3; (Nw = 10000)	Nw = 15000	サービス HTTP-3 の Nw 値は最も低いです。
Request-4	Service-HTTP-3; (Nw = 15000)	Nw = 20000	サービス HTTP-3 の Nw 値は最も低いです。
Request-5	Service-HTTP-3; (New = 20000)	Nw = 25000	サービス HTTP-3 の Nw 値は最も低いです。
Request-6	Service-HTTP-2; (New = 23333.34)	Nw = 26666.67	サービス HTTP-2 の Nw 値は最も低いです。
Request-7	Service-HTTP-3; (New = 25000)	Nw = 30000	サービス HTTP-3 の Nw 値は最も低いです。
Request-8	Service-HTTP-2; (Nw = 26666.67)	Nw = 30000	サービス HTTP-2 の Nw 値は最も低いです。

Service-HTTP-1 は、アクティブなトランザクションが完了したとき、または Nw 値が他のサービス（Service-HTTP-2 および Service-HTTP-3）よりも小さい場合に、ロードバランシングの対象として選択されます。

CLI を使用して **LRTM** ロードバランシング方式を設定するには

コマンドプロンプトで、次のように入力します。

```
set lb vserver <name> [-lbMethod <lbMethod>]
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -lbMethod LRTM
<!--NeedCopy-->
```


GUI を使用して **LRTM** ロードバランシング方式を設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2.「詳細設定」で「**LRTM**」を選択します。

CLI を使用してモニターの **LRTM** オプションを有効にするには

コマンドプロンプトで、次のように入力します。

```
set lb monitor <monitorName> <type> [-LRTM ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

例**:**

```
set lb monitor monitor-HTTP-1 HTTP -LRTM ENABLED
<!--NeedCopy-->
```

GUI を使用してモニタの **LRTM** オプションを有効にするには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動し、モニターを開きます。
2.「拡張パラメータ」で、「**LRTM (**監視を使用した最小応答時間**)**」を選択します。

モニタの構成の詳細については、「負荷分散セットアップでのモニタの設定」を参照してください。

ハッシュ方式

August 15, 2023

特定の接続情報またはヘッダー情報のハッシュに基づく負荷分散方法は、NetScaler ADC アプライアンスの負荷分散方式のほとんどを構成します。ハッシュは、基づいている情報よりも短く使いやすく、2 つの異なる情報が同じハッシュを生成しないため、互いに混乱しないように十分な情報を保持しています。

ハッシュ負荷分散方法は、キャッシュがインターネットまたは指定したオリジンサーバーからの幅広いコンテンツを提供する環境で使用できます。リクエストをキャッシュすると、リクエストとレスポンスのレイテンシーが削減され、リソース（CPU）の使用率が向上し、頻繁に使用される Web サイトやアプリケーションサーバーでキャッシュが普及します。これらのサイトは負荷分散の恩恵を受けるため、ハッシュ負荷分散方式は広く有用です。

NetScaler アプライアンスには次のハッシュメソッドがあります。

- URL ハッシュ方式
- ドメインハッシュ法
- 宛先 IP ハッシュ方式

- ソース IP ハッシュ方式
- 送信元 IP 宛先 IP ハッシュ方式
- 送信元 IP 送信元ポートのハッシュ方式
- コール ID ハッシュメソッド
- トークン方式

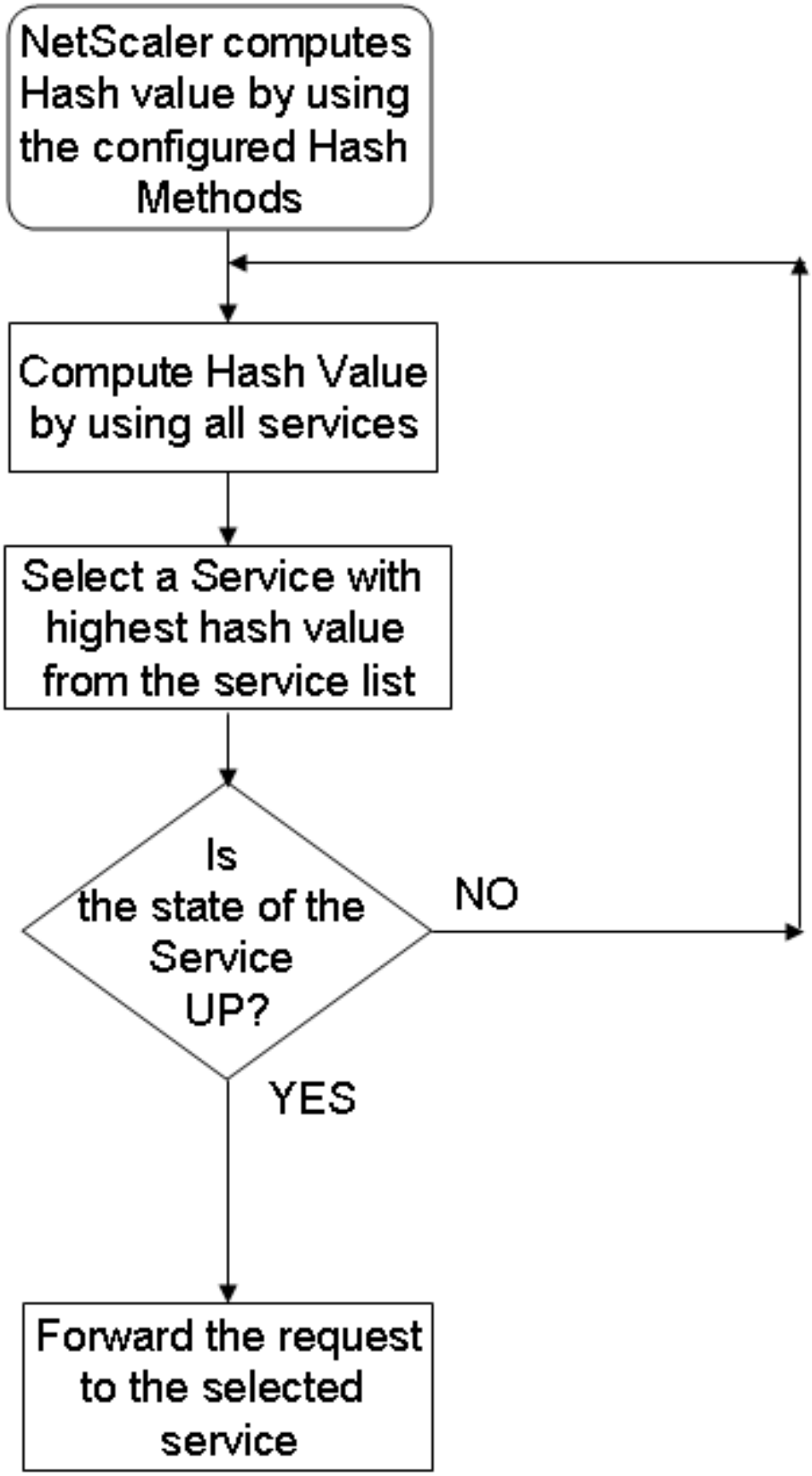
ほとんどのハッシュアルゴリズムは、次の 2 つのハッシュ値を計算します。

- サービスの IP アドレスとポートのハッシュ。
- 設定されているハッシュ方式に応じて、受信 URL、ドメイン名、送信元 IP アドレス、宛先 IP アドレス、または送信元と宛先 IP アドレスのハッシュ。

次に、NetScaler ADC アプライアンスは、これらのハッシュ値の両方を使用して新しいハッシュ値を生成します。最後に、ハッシュ値が最も高いサービスにリクエストを転送します。アプライアンスは各リクエストのハッシュ値を計算し、要求を処理するサービスを選択すると、キャッシュが生成されます。同じハッシュ値を持つ後続の要求は、同じサービスに送信されます。次のフローチャートは、このプロセスを示しています。

> 注
>
> NetScaler ADC リリース 13.0 ビルド 79.x から、プライム再シャッフルアシスト CARP（PRAC）とジャンプテーブルアシストリングハッシュ（JAR）の一貫性のあるハッシュアルゴリズムがサポートされています。一貫性のあるハッシュアルゴリズムにより、ロードバランシングセットアップにサービスが追加または削除されたとき、またはロードバランシングセットアップでサービスフラップイベント中にサービスが中断される最小限に抑えられます。詳細については、「一貫性のあるハッシュアルゴリズム」を参照してください。

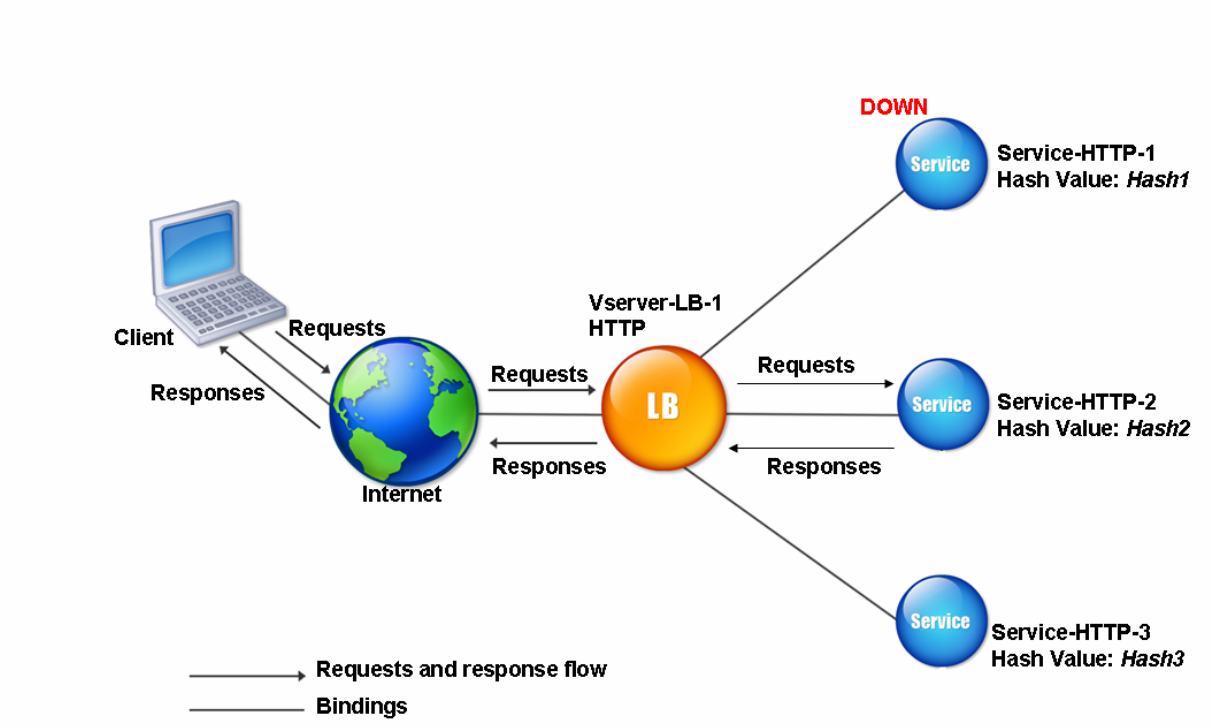
図 1: ハッシュメソッドによるリクエストの配布



ハッシュ方法は、IPv4 アドレスと IPv6 アドレスに適用できます。

3 つのサービス (サービス HTTP-1、サービス HTTP-2、サービス HTTP-3) が仮想サーバーにバインドされ、任意のハッシュ方式が設定され、ハッシュ値が Hash1 であるシナリオを考えてみましょう。設定されたサービスが UP になると、要求は Service-HTTP-1 に送信されます。Service-HTTP-1 がダウンしている場合、NetScaler アプライアンスはサービス数の最後のログのハッシュ値を計算します。次に、アプライアンスは Service-HTTP-2 など、ハッシュ値が最も高いサービスを選択します。次の図は、このプロセスを示しています。

図 2: ハッシュメソッドのエンティティモデル



> 注
>
> NetScaler ADC アプライアンスがハッシュ方式を使用してサービスを選択できない場合、デフォルトでは、着信要求のサービスを選択するための最小接続方法が使用されます。負荷分散設定のパフォーマンスに影響を与えずにキャッシュを再入力できるように、トラフィックが少ない時間帯にサービスを削除してサーバープールを調整します。

一貫性のあるハッシュアルゴリズム

一貫性のあるハッシュアルゴリズムは、ステートレスな永続性を実現するために使用されます。ハッシュベースの LB メソッドは、次の 3 つの一貫性のあるハッシュアルゴリズムのいずれかを使用します。

- キャッシュアレイルーティングプロトコル **(CAP)**

  CARP アルゴリズムは、複数のプロキシキャッシュサーバ間の HTTP 要求のロードバランシングに使用されます。このアルゴリズムはデフォルトで有効になっています。

- プライム再シャッフルアシストカープ **(PRAC)**

  NetScaler ADC アプライアンスは、独自の PRAC アルゴリズムを使用して、均一なトラフィック分散を提供します。

- ジャンプテーブルアシストリングハッシュ **(JARH)**

  NetScaler ADC アプライアンスは、独自の JARH アルゴリズムを使用して、トラフィックの一貫性と均一な分散を提供します。このアルゴリズムはハッシュフィンガーを使用します。指の数が多いほど、トラフィックの分布が向上します。ただし、指の数を増やすと、メモリ使用量も増加します。

CLI を使用して一貫性のあるハッシュアルゴリズムを選択するには

```
set lb parameter [-lbHashAlgorithm [DEFAULT|JARH|PRAC] [-lbHashFingers
    <positive_integer>]
<!--NeedCopy-->
```

例**:**

```
set lb parameter -lbHashAlgorithm JARH -lbHashFingers 10
<!--NeedCopy-->
```

引数**:**

- **lbHashAlgorithm**-次のハッシュベースのロードバランシング方法に使用するハッシュアルゴリズムを指定します。

  - URL ハッシュ方式
  - ドメインハッシュ法
  - 宛先 IP ハッシュ方式
  - ソース IP ハッシュ方式
  - 送信元 IP 宛先 IP ハッシュ方式
  - 送信元 IP 送信元ポートのハッシュ方式
  - コール ID ハッシュメソッド
  - トークン方式

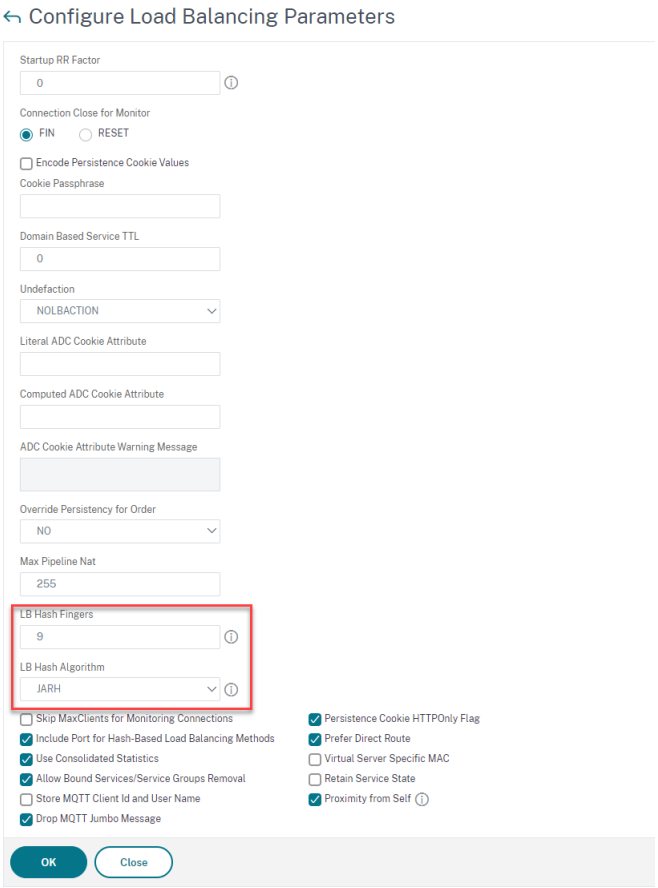
  指定可能な値:DEFAULT、PRAC、JARH
  デフォルト値:DEFAULT

- **lbHashFingers-**ハッシュベースの LB メソッドの PRAC および JARH アルゴリズムで使用する指の数を指定します。指の数を増やすと、余分なメモリを犠牲にしてトラフィックの分散が向上します。

  デフォルト値:256
  最小値:1
  最大値:1024

GUI を使用して一貫性のあるハッシュアルゴリズムを選択するには

1. **[**トラフィック管理**] > [**負荷分散**] > [**負荷分散パラメータの変更**]** に移動します。
2. [負荷分散パラメータの構成] ペインで、要件に基づいて次のフィールドに適切な値を入力します。
   - LB ハッシュフィンガー
   - [**LB Hash Algorithm**] フィールドで、ドロップダウンメニューから一貫性のあるハッシュアルゴリズムを選択します。



URL ハッシュメソッド

サービスの負荷分散やサービスの選択に URL ハッシュ方式を使用するように NetScaler アプライアンスを構成すると、アプライアンスは受信リクエストに含まれる HTTP URL のハッシュ値を生成します。ハッシュ値で選択されたサービスが DOWN の場合、アルゴリズムにはアクティブなサービスのリストから別のサービスを選択する方法があります。アプライアンスは URL のハッシュ値をキャッシュし、同じ URL を使用する後続のリクエストを受信すると、それらを同じサービスに転送します。アプライアンスが受信リクエストを解析できない場合、URL ハッシュ方式の代わりにラウンドロビン方式を使用して負荷分散を行います。

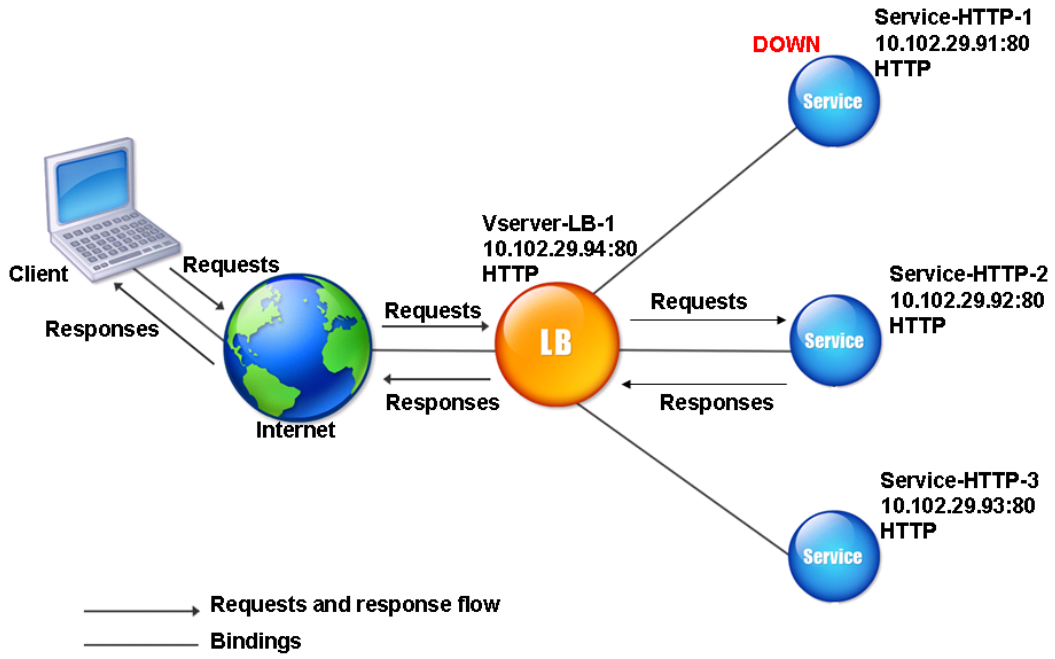
ハッシュ値の生成には、アプライアンスは特定のアルゴリズムを使用し、URL の一部を考慮します。デフォルトで

は、アプライアンスは URL の最初の 80 バイトを考慮します。URL が 80 バイト未満の場合は、完全な URL が使用されます。別の長さを指定できます。ハッシュの長さは 1 バイトから 4096 バイトまでです。一般に、数文字しか異なる長い URL を使用する場合は、より均一な負荷分散を確保するために、ハッシュ長をできるだけ長くすることをお勧めします。

Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 という 3 つのサービスが仮想サーバーにバインドされ、仮想サーバーで構成される負荷分散方式が URL ハッシュ方式であるシナリオを考えてみます。仮想サーバーが要求を受信し、URL のハッシュ値は U1 です。アプライアンスはサービス HTTP-1 を選択します。Service-HTTP-1 が DOWN の場合、アプライアンスは Service-HTTP-2 を選択します。

次の図は、このプロセスを示しています。

図 3: URL ハッシュの仕組み



Service-HTTP-1 と Service-HTTP-2 の両方がダウンしている場合、アプライアンスはハッシュ値 U1 の要求をサービス HTTP-3 に送信します。

Service-HTTP-1 および Service-HTTP-2 がダウンしている場合、ハッシュ URL1 を生成する要求は Service-HTTP-3 に送信されます。これらのサービスが UP の場合、ハッシュ URL1 を生成する要求は次の方法で配布されます。

- Service-HTTP-2 が起動している場合、要求は Service-HTTP-2 に送信されます。
- サービス HTTP-1 が稼働している場合、リクエストはサービス HTTP-1 に送信されます。

- Service-HTTP-1 と Service-HTTP-2 が同時にアップしている場合、要求は Service-HTTP-1 に送信されます。

URL ハッシュ方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。[URL Hash] としてロードバランシング方式を選択し、ハッシュ長をハッシュ値の生成に使用するバイト数に設定します。

ドメインハッシュ法

ドメインハッシュ方式を使用するように構成された負荷分散仮想サーバーは、HTTP リクエスト内のドメイン名のハッシュ値を使用してサービスを選択します。ドメイン名は、HTTP リクエストの受信 URL または Host ヘッダーのいずれかから取得されます。ドメイン名が URL と Host ヘッダーの両方に表示される場合、アプライアンスはその URL を優先します。

ドメイン名ハッシュを構成していて、受信 HTTP リクエストにドメイン名が含まれていない場合、NetScaler アプライアンスはそのリクエストに対してデフォルトでラウンドロビン方式になります。

ハッシュ値の計算では、名前の長さとハッシュ長の値のどちらか小さい方を使用します。デフォルトでは、NetScaler アプライアンスはドメイン名の最初の 80 バイトからハッシュ値を計算します。ハッシュ値を計算するときにドメイン名に異なるバイト数を指定するには、hashLength パラメータ（設定ユーティリティの Hash Length）を 1 ～ 4096（バイト）の値に設定します。

ドメインハッシュ方式を設定するには、ポリシーを含まない負荷分散方式の構成を参照してください。

宛先 **IP** ハッシュ方式

宛先 IP ハッシュ方式を使用するように構成された負荷分散仮想サーバーは、宛先 IP アドレスのハッシュ値を使用してサーバーを選択します。宛先 IP アドレスをマスクして、ハッシュ値の計算にどの部分を使用するかを指定できます。これにより、異なるネットワークから送信され、同じサブネット宛ての要求はすべて同じサーバーに送信されます。このメソッドは、IPv4 および IPv6 ベースの宛先サーバーをサポートします。

この負荷分散方法は、キャッシュリダイレクト機能との併用に適しています。

IPv4 宛先サーバーの宛先 IP ハッシュ方式を設定するには、NetMask パラメータを設定します。IPv6 宛先サーバーに対してこのメソッドを設定するには、v6NetMaskLen パラメータを使用します。構成ユーティリティで、[宛先 **IP** ハッシュ方式] を選択すると、これらのパラメータを設定するためのテキストボックスが表示されます。

宛先 IP ハッシュ方式を設定するには、ポリシーを含まないロードバランシング方式の構成を参照してください。

送信元 **IP** ハッシュ方式

ソース IP ハッシュ方式を使用するように構成された負荷分散仮想サーバーは、クライアントの IPv4 または IPv6 アドレスのハッシュ値を使用してサービスを選択します。特定のネットワークに属する送信元 IP アドレスからのすべ

ての要求を特定の宛先サーバーに転送するには、送信元 IP アドレスをマスクする必要があります。IPv4 アドレスの場合は、NetMask パラメータを使用してください。IPv6 アドレスの場合は、v6NetMaskLength パラメーターを使用します。

送信元 IP ハッシュ方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

送信元 **IP** 宛先 **IP** ハッシュ方式

送信元 IP 宛先 IP ハッシュ方式を使用するように設定されたロードバランシング仮想サーバは、送信元と宛先 IP アドレス（IPv4 または IPv6）のハッシュ値を使用してサービスを選択します。ハッシュは対称です。ハッシュ値は、送信元と宛先 IP の順序に関係なく、同じです。これにより、特定のクライアントから同じ宛先に流れるすべてのパケットが同じサーバに送信されます。

特定のネットワークに属するすべての要求を特定の宛先サーバーに送信するには、送信元 IP アドレスをマスクする必要があります。IPv4 アドレスの場合は、NetMask パラメータを使用してください。IPv6 アドレスの場合は、v6NetMaskLength パラメーターを使用します。

送信元 IP 宛先 IP ハッシュ方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

送信元 **IP** 送信元ポートのハッシュ方式

ソース IP ソースポートハッシュ方式を使用するように構成された負荷分散仮想サーバーは、送信元 IP（IPv4 または IPv6）のハッシュ値および送信元ポートのハッシュ値を使用してサービスを選択します。これにより、特定の接続上のすべてのパケットが同じサービスに送信されます。

この方法は、接続ミラーリングとファイアウォールの負荷分散で使用されます。接続ミラーリングの詳細については、「接続のフェイルオーバー」を参照してください。

特定のネットワークに属するすべての要求を特定の宛先サーバーに送信するには、送信元 IP アドレスをマスクする必要があります。IPv4 アドレスの場合は、NetMask パラメータを使用してください。IPv6 アドレスの場合は、v6NetMaskLength パラメーターを使用します。

送信元 IP 送信元ポートのハッシュ方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

コール **ID** ハッシュメソッド

コール ID ハッシュ方式を使用するように構成された負荷分散仮想サーバーは、SIP ヘッダーのコール ID のハッシュ値を使用してサービスを選択します。そのため、特定の SIP セッションのパケットは常に同じプロキシサーバーに送信されます。

この方法は、SIP ロードバランシングに適用できます。SIP ロードバランシングの詳細については、「SIP サービスのモニタリング」を参照してください。

コール ID ハッシュメソッドを設定するには、ポリシーを含まないロードバランシングメソッドの設定を参照してください。

最小帯域幅方式

August 15, 2023

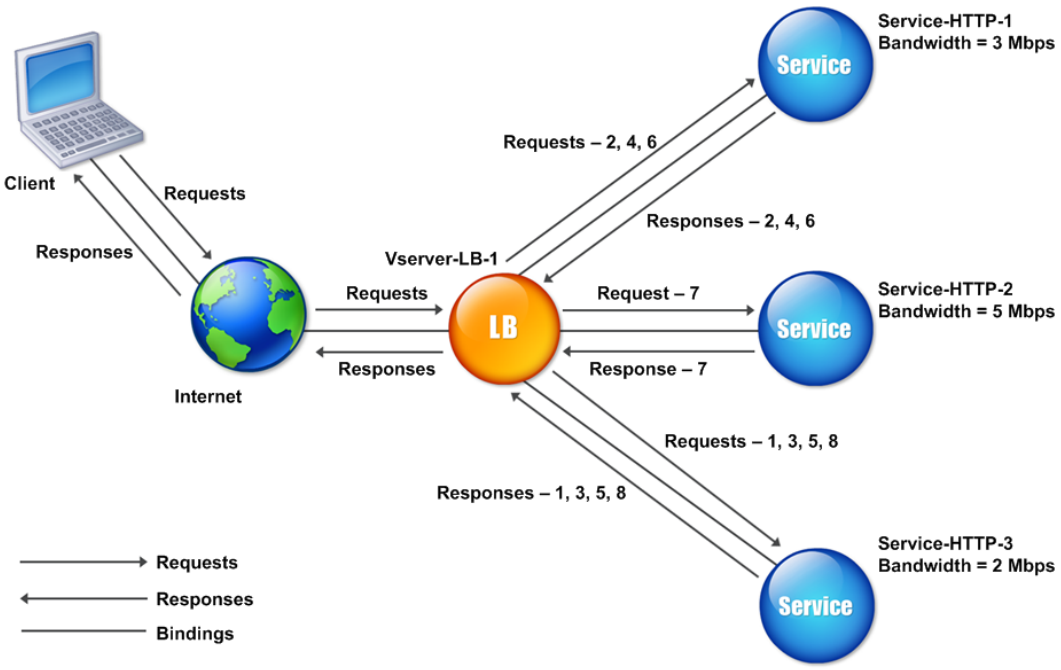
最小帯域幅方式を使用するように構成された負荷分散仮想サーバーは、現在処理しているトラフィックの量（メガビット/秒（Mbps）が最も少ないサービスを選択します。次の例は、仮想サーバーが最小帯域幅方式を使用して負荷分散用のサービスを選択する方法を示しています。

サービス HTTP-1、サービス HTTP-2、サービス HTTP-3 の 3 つのサービスを考えてみましょう。

- サービス HTTP-1 の帯域幅は 3 Mbps です。
- サービス HTTP-2 の帯域幅は 5 Mbps です。
- サービス HTTP-3 の帯域幅は 2 Mbps です。

次の図は、仮想サーバーが最小帯域幅方式を使用して 3 つのサービスに要求を転送する方法を示しています。

図 1：最小帯域幅ロードバランシング方式の動作



仮想サーバは、過去 14 秒間に送受信されたバイト数の合計である帯域幅値（N）を使用してサービスを選択します。各リクエストに 1 Mbps の帯域幅が必要な場合、NetScaler アプライアンスは次のようにリクエストを配信します。

- Service-HTTP-3 は最初の要求を受信します。これは、このサービスの N 値が最小であるためです。
- Service-HTTP-1 と Service-HTTP-3 は同じ N 値を持つため、仮想サーバはこれらのサーバのラウンドロビン方式に切り替わり、それらのサーバ間で交互に切り替えられます。Service-HTTP-1 は 2 番目の要求を受信し、Service-HTTP-3 は 3 番目の要求を受信し、Service-HTTP-1 は 4 番目の要求を受信し、Service-HTTP-3 は 5 番目の要求を受信し、Service-HTTP-1 は 6 番目の要求を受信します。
- Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 はすべて同じ N 値を持つため、仮想サーバは Service-HTTP-2 をラウンドロビンリストに含めます。したがって、Service-HTTP-2 は 7 番目の要求を受信し、Service-HTTP-3 は 8 番目の要求を受信します。

次の表は、N の計算方法をまとめたものです。

リクエストを受け取りました	サービス選択済み	現在の N 値	注釈
Request-1	Service-HTTP-3; (N = 2)	N = 3	Service-HTTP-3 は最小 N 値を持ちます。
Request-2	Service-HTTP-1; (N = 3)	N = 4	サービス HTTP-1 とサービス HTTP-3 の N 値は同じ
Request-3	サービス-HTTP-3; (N = 3)	N = 4	サービス HTTP-1 とサービス HTTP-3 の N 値は同じ
Request-4	Service-HTTP-1; (N = 4)	N = 5	-
Request-5	Service-HTTP-3; (N = 4)	N = 5	-
Request-6	Service-HTTP-1; (N = 5)	N = 6	サービス HTTP-1、サービス HTTP-2、およびサービス HTTP-3 には、同じ N 個の値があります。
Request-7	Service-HTTP-2; (N = 5)	N = 6	サービス HTTP-1、サービス HTTP-2、およびサービス HTTP-3 には、同じ N 個の値があります。
Request-8	Service-HTTP-3; (N = 5)	N = 6	-

注：仮想サーバーで RTSP NAT オプションを有効にすると、NetScaler ADC アプライアンスは交換されたデータ数と制御バイト数を使用して、RTSP サービスの帯域幅使用率を決定します。RTSP NAT オプションの詳細については、「RTSP 接続の管理」を参照してください。

また、NetScaler ADC アプライアンスは、異なるウェイトがサービスに割り当てられている場合、帯域幅と重みを使用して負荷分散を実行します。次の式の値 (Nw) を使用してサービスを選択します。

Nw = (N) * (10000 /重量)

前の例のように、サービス HTTP-1 には 2 の重みが割り当てられ、サービス HTTP-2 には 3、サービス HTTP-3 には 4 の重みが割り当てられているとします。NetScaler ADC アプライアンスは、次のようにリクエストを送信します。

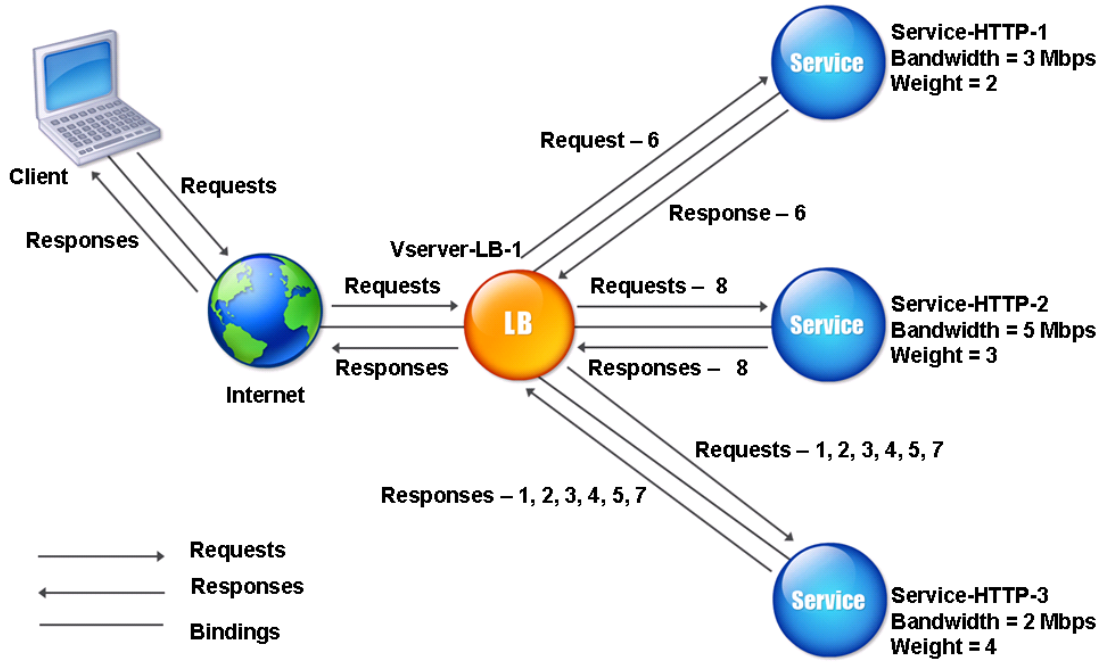
- Service-HTTP-3 は、最初の 2 番目、3 番目、4 番目、5 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。
- Service-HTTP-1 は 6 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。
- Service-HTTP-3 は 7 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。
- Service-HTTP-2 は 8 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。

次の表は、Nw の計算方法をまとめたものです。

リクエストを受け取りました	サービス選択済み	現在の新しい価値 (アクティブなトランザクションの数) * (10000/重量)	注釈
Request-1	Service-HTTP-3; (Nw = 5000)	Nw = 5000	サービス HTTP-3 の Nw 値は最も低いです。
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 7500	-
Request-3	Service-HTTP-3; (New = 7500)	Nw = 10000	-
Request-4	Service-HTTP-3; (Nw = 10000)	Nw = 12500	-
Request-5	Service-HTTP-3; (Nw = 12500)	Nw = 15000	-
Request-6	Service-HTTP-1; (Nw = 15000)	Nw = 20000	サービス HTTP-1 とサービス HTTP-3 の Nw 値は同じです。
Request-7	Service-HTTP-3; (Nw = 15000)	Nw = 17500	サービス HTTP-1 とサービス HTTP-3 の Nw 値は同じです。
Request-8	Service-HTTP-2; (Nw = 16666.67)	Nw = 20000	サービス HTTP-2 の Nw 値は最も低いです。

次の図は、サービスに重みを割り当てる場合に、仮想サーバーが最小帯域幅方式を使用する方法を示しています。

図 2：重みが割り当てられている場合の最小帯域幅ロードバランシング方式の動作



最小帯域幅方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

最小パケット方式

August 15, 2023

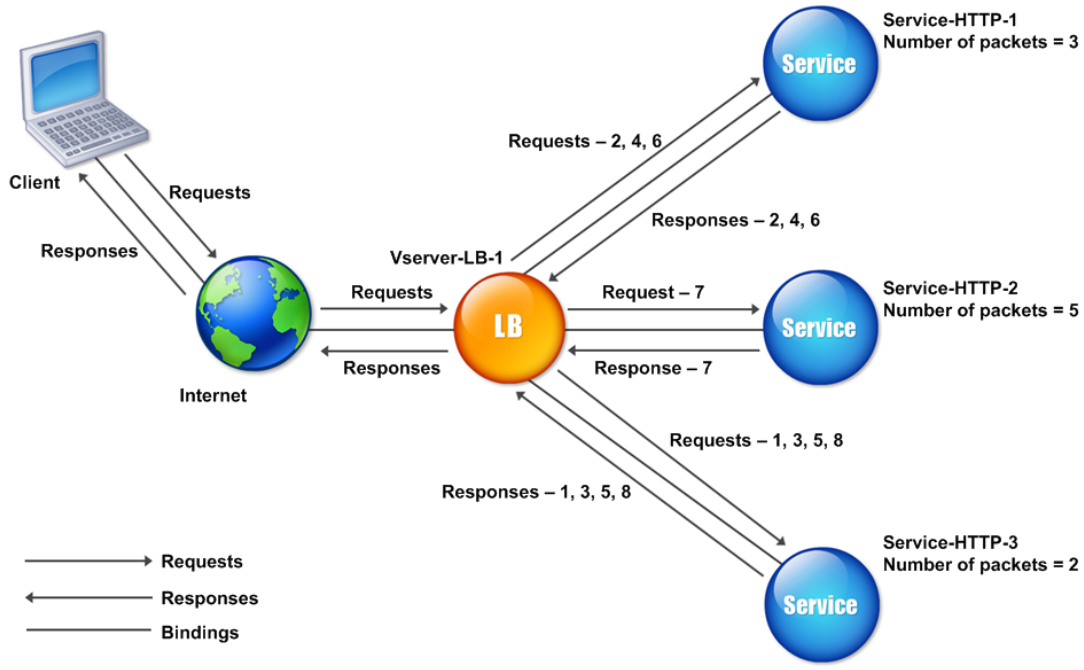
最小パケット方式を使用するように構成された負荷分散仮想サーバーは、過去 14 秒間に受信したパケット数が最も少ないサービスを選択します。

たとえば、Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 の 3 つのサービスを考えてみます。

- Service-HTTP-1 は過去 14 秒で 3 つのパケットを処理しました。
- Service-HTTP-2 は過去 14 秒で 5 つのパケットを処理しました。
- Service-HTTP-3 は過去 14 秒で 2 つのパケットを処理しました。

次の図は、NetScaler ADC アプライアンスが最小パケット方式を使用して、受信する要求ごとにサービスを選択する方法を示しています。

図 1：最小パケット負荷分散方式の仕組み



NetScaler ADC アプライアンスは、過去 14 秒間に各サービスが送受信したパケット数（N）を使用してサービスを選択します。この方法を使用すると、次のようにリクエストが配信されます。

- Service-HTTP-3 は最初の要求を受信します。これは、このサービスの N 値が最小であるためです。
- サービス HTTP-1 とサービス HTTP-3 の N 値が同じになったため、仮想サーバーはラウンドロビン方式に切り替わります。したがって、Service-HTTP-1 は 2 番目の要求を受信し、Service-HTTP-3 は 3 番目の要求を受信し、Service-HTTP-1 は 4 番目の要求を受信し、Service-HTTP-3 は 5 番目の要求を受信し、Service-HTTP-1 は 6 番目の要求を受信します。
- Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 はすべて同じ N 値を持つため、仮想サーバは Service-HTTP-2 のラウンドロビン方式に切り替え、ラウンドロビンリストに含めます。したがって、Service-HTTP-2 は 7 番目の要求を受信し、Service-HTTP-3 は 8 番目の要求を受信します。

次の表は、N の計算方法をまとめたものです。

リクエストを受け取りました	サービス選択済み	現在の N 値	注釈
Request-1	Service-HTTP-3; (N = 2)	N = 3	Service-HTTP-3 は最小 N 値を持ちます。

リクエストを受け取りました	サービス選択済み	現在の N 値	注釈
Request-2	Service-HTTP-1; (N = 3)	N = 4	サービス HTTP-1 とサービス HTTP-3 の N 値は同じ
Request-3	Service-HTTP-3; (N = 3)	N = 4	サービス HTTP-1 とサービス HTTP-3 の N 値は同じ
Request-4	Service-HTTP-1; (N = 4)	N = 5	-
Request-5	Service-HTTP-3; (N = 4)	N = 5	-
Request-6	Service-HTTP-1; (N = 5)	N = 6	サービス HTTP-1、サービス HTTP-2、およびサービス HTTP-3 には、同じ N 個の値があります。
Request-7	Service-HTTP-2; (N = 5)	N = 6	サービス HTTP-1、サービス HTTP-2、およびサービス HTTP-3 には、同じ N 個の値があります。
Request-8	Service-HTTP-3; (N = 5)	N = 6	-

注：仮想サーバで RTSP NAT オプションを有効にすると、アプライアンスはデータおよび制御パケットの数を使用して、RTSP サービスのパケット数を計算します。RTSP NAT オプションの詳細については、「RTSP 接続の管理」を参照してください。

また、NetScaler ADC アプライアンスは、各サービスに異なる重みが割り当てられている場合に、パケット数と重みを使用して負荷分散を実行します。次の式の値 (Nw) を使用してサービスを選択します。

Nw = (N) * (10000 /重量)

前の例のように、サービス HTTP-1 には 2 の重みが割り当てられ、サービス HTTP-2 には 3、サービス HTTP-3 には 4 の重みが割り当てられているとします。NetScaler ADC アプライアンスは、次のようにリクエストを送信します。

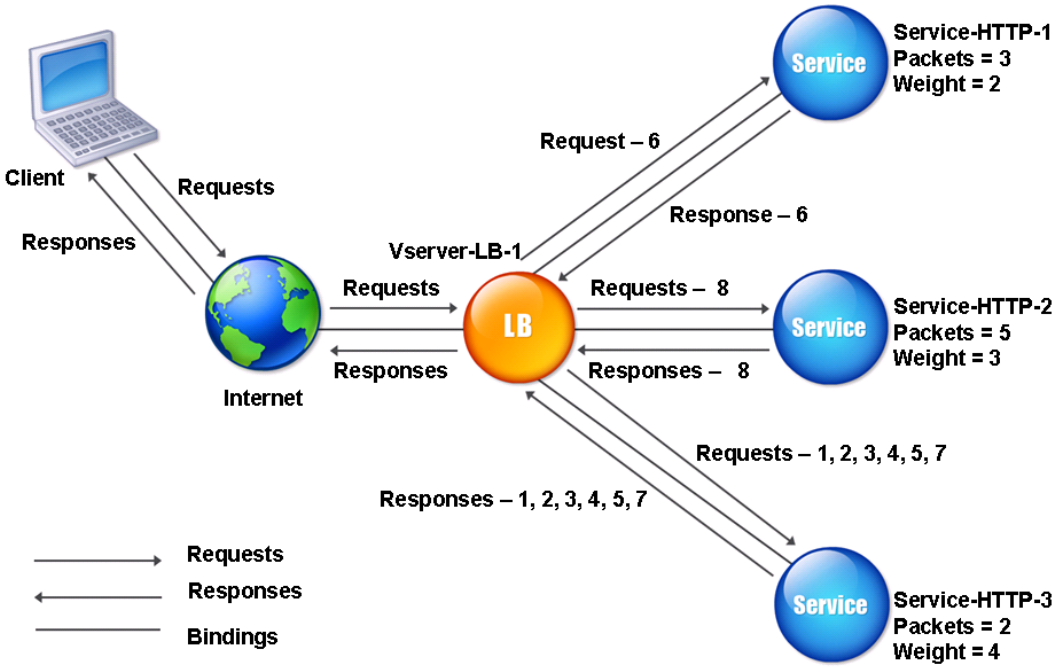
- Service-HTTP-3 は、最初の 2 番目、3 番目、4 番目、5 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。
- Service-HTTP-1 は 6 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。
- Service-HTTP-3 は 7 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。
- Service-HTTP-2 は 8 番目のリクエストを受け取ります。これは、このサービスの Nw 値が最も小さいためです。

次の表は、Nw の計算方法をまとめたものです。

リクエストを受け取りました	サービス選択済み	現在の新価値 (アクティブなトランザクション数) * (10000 /重量)	注釈
Request-1	Service-HTTP-3; (Nw = 5000)	Nw = 5000	サービス HTTP-3 の Nw 値は最も低いです。
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 7500	-
Request-3	Service-HTTP-3;(Nw = 7500)	Nw = 10000	-
Request-4	Service-HTTP-3; (Nw = 10000)	Nw = 12500	-
Request-5	Service-HTTP-3; (Nw = 12500)	Nw = 15000	-
Request-6	Service-HTTP-1; (Nw = 15000)	Nw = 20000	サービス HTTP-1 とサービス HTTP-3 の Nw 値は同じです。
Request-7	Service-HTTP-3; (Nw = 15000)	Nw = 17500	サービス HTTP-1 とサービス HTTP-3 の Nw 値は同じです。
Request-8	Service-HTTP-2; (Nw = 16666.67)	Nw = 20000	サービス HTTP-2 の Nw 値は最も低いです。

次の図は、重みが割り当てられるときに仮想サーバーがどのように最小パケット方式を使用するかを示しています。

図 2：重みが割り当てられている場合の最小パケット方式の仕組み



最小パケット方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

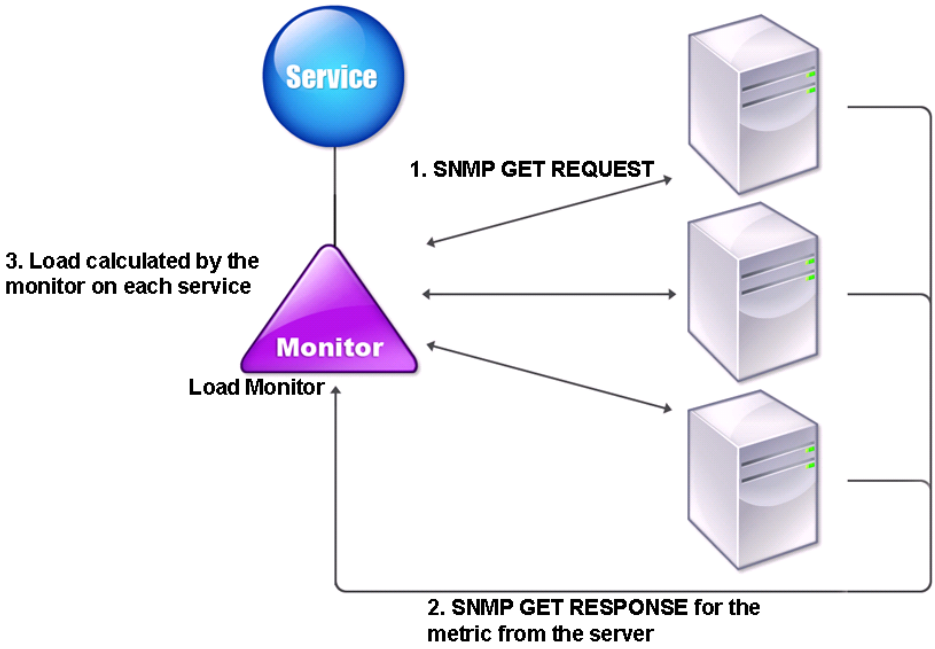
カスタム負荷方式

August 15, 2023

カスタム負荷分散は、CPU 使用率、メモリ、応答時間などのサーバパラメータに対して実行されます。カスタムロード方式を使用する場合、NetScaler ADC アプライアンスは通常、アクティブなトランザクションを処理していないサービスを選択します。ロードバランシング設定のすべてのサービスがアクティブなトランザクションを処理している場合、アプライアンスは負荷が最小のサービスを選択します。ロードモニタと呼ばれる特殊なタイプのモニタは、ネットワーク内の各サービスの負荷を計算します。ロードモニタはサービスの状態をマークしませんが、サービスが UP でない場合、ロードバランシングの決定からサービスを取り出します。

負荷モニタの詳細については、「負荷モニタについて」を参照してください。次の図は、ロードモニタの動作を示しています。

図 1: ロードモニターの仕組み



負荷モニタは SNMP プローブを使用して、SNMP GET 要求をサービスに送信して、各サービスの負荷を計算します。このリクエストには、1 つ以上のオブジェクト ID (OID) が含まれます。サービスは、SNMP OID に対応するメトリックとともに、SNMP GET 応答で応答します。負荷モニターは、レスポンスメトリックを使用してサービスの負荷を計算します。

負荷モニターは、次のパラメーターを使用してサービスの負荷を計算します。

- NetScaler アプライアンスにテーブルとして存在する SNMP プローブを介して取得されたメトリック値。
- 各メトリックに設定されたしきい値。
- 各メトリックに割り当てられた重み。

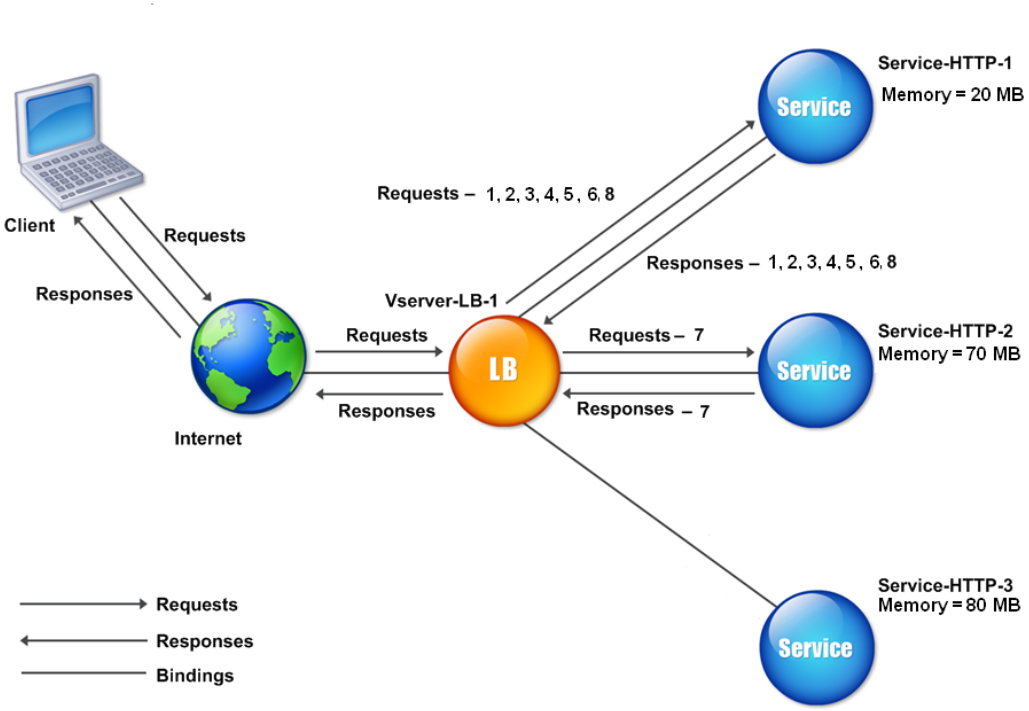
たとえば、Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 の 3 つのサービスを考えてみます。

- Service-HTTP-1 は 20 MB のメモリを使用しています。
- Service-HTTP-2 では、70 MB のメモリを使用しています。
- サービス HTTP-3 は 80 MB のメモリを使用しています。

負荷分散されたサーバーは、CPU やメモリの使用量などのメトリックをサービスにエクスポートし、サービスがそれらを負荷モニターに提供できます。ロードモニターは、OID 1.3.6.1.4.1.5951.4.1.41.1.5、1.3.6.1.4.1.5951.4.1.41.1.4、および 1.3.6.1.4.1.5951.4.1.5951.4.1.41.1.3 を含む SNMP GET リクエストをサービスに送信します。文字列 OID を使用して負荷を計算することはできないため、文字列型の SNMP OID はサポ

ートされていません。負荷は、INT や gauge32 などの他のデータ型を使用して計算できます。3 つのサービスが要求に応答します。NetScaler アプライアンスはエクスポートされたメトリックを比較し、使用可能なメモリの方が多いため Service-HTTP-1 を選択します。次の図は、このプロセスを示しています。

図 2：カスタムロード方法の仕組み



各リクエストが 10 MB のメモリを使用する場合、NetScaler アプライアンスは次のようにリクエストを配信します。

- Service-HTTP-1 は、1 番目、2 番目、3 番目、4 番目、5 番目のリクエストを受信します。これは、このサービスの N 値が最小であるためです。
- Service-HTTP-1 と Service-HTTP-2 の負荷が同じになったため、仮想サーバーはこれらのサーバーのラウンドロビン方式に戻ります。したがって、Service-HTTP-2 は 6 番目の要求を受信し、Service-HTTP-1 は 7 番目の要求を受信します。
- Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 はすべて同じ負荷を持つため、仮想サーバも Service-HTTP-3 のラウンドロビン方式に戻ります。したがって、Service-HTTP-3 は 8 番目の要求を受信します。

次の表は、N の計算方法をまとめたものです。

リクエストを受け取りました	サービスが選択されました	現在の N 値 (アクティブなトランザクションの数)	注釈
Request-1	Service-HTTP-1; (N = 20)	N = 30	Service-HTTP-3 は最小 N 値を持ちます。
Request-2	Service-HTTP-1; (N = 30)	N = 40	-
Request-3	Service-HTTP-1; (N = 40)	N = 50	-
Request-4	Service-HTTP-1; (N = 50)	N = 60	-
Request-5	Service-HTTP-1; (N = 60)	N = 70	-
Request-6	Service-HTTP-1; (N = 70)	N = 80	サービス HTTP-2 とサービス HTTP-3 の N 値は同じ。
Request-7	Service-HTTP-2; (N = 70)	N = 80	サービス HTTP-3 の N 値は同じ。
Request-8	Service-HTTP-1; (N = 80)	N = 90	サービス HTTP-1、サービス HTTP-2、およびサービス HTTP-3 には、同じ N 個の値があります。

サービスに異なる重みが割り当てられている場合、カスタム負荷アルゴリズムは各サービスの負荷と各サービスに割り当てられた重みの両方を考慮します。次の式の値 (Nw) を使用してサービスを選択します。

Nw = (N) * (10000 /重量)

前の例と同様に、サービス HTTP-1 には 4 の重みが割り当てられ、サービス HTTP-2 には 3 の重みが割り当てられ、サービス HTTP-3 には重み 2 が割り当てられているとします。各リクエストが 10 MB のメモリを使用する場合、NetScaler アプライアンスは次のようにリクエストを配信します。

- Service-HTTP-1 は、1 番目、2 番目、3 番目、4 番目、5 番目、6 番目、7 番目、8 番目のリクエストを受信します。これは、このサービスの Nw 値が最も低いためです。
- Service-HTTP-2 は 9 番目のリクエストを受信します。これは、このサービスの Nw 値が最小であるためです。

Service-HTTP-3 は Nw 値が最も高いため、ロードバランシングの対象にはなりません。

次の表は、Nw の計算方法をまとめたものです。

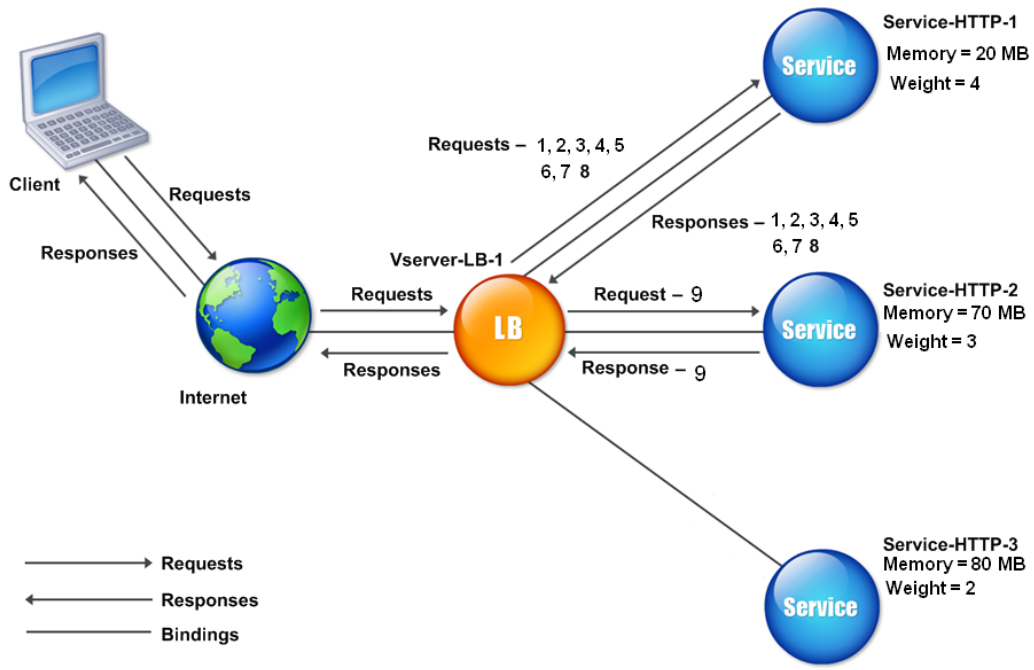
| リクエストを受け取りました | サービスが選択されました | 現在の新しい価値 (アクティブなトランザクションの数) * (10000/重量)| 注釈

|-|-|-|-|
|Request-1|Service-HTTP-1; (Nw = 50000)|Nw = 75000| サービス HTTP-1 の Nw 値は最も低いです。
Request-2	Service-HTTP-1; (Nw = 5000)	Nw = 100000	-
Request-3	Service-HTTP-1; (Nw = 15000)	Nw = 125000	-
Request-4	Service-HTTP-1; (Nw = 20000)	Nw = 150000	-
Request-5	Service-HTTP-1; (New = 23333.34)	Nw = 175000	-
Request-6	Service-HTTP-1; (Nw = 25000)	Nw = 200000	-
Request-7	Service-HTTP-1; (New = 23333.34)	Nw = 225000	-
request-8|service-HTTP-1; (Nw = 25000) |Nw = 250000||-| |request-9|Service-HTTP-2; (Nw = 233333.34) |Nw = 266666.67|Service-HTTP-2 の Nw 値が最も低い。

サービス HTTP-1 は、アクティブなトランザクションが完了したとき、または他のサービス (サービス HTTP-2 と Service-HTTP-3) の Nw 値が 400,000 になったときに、ロードバランシングの対象として選択されます。

次の図は、重みが割り当てられるときに NetScaler アプライアンスがカスタムロード方法を使用する方法を示しています。

図 3: ウェイトが割り当てられている場合のカスタムロード方法の動作



カスタムロード方式を構成するには、ポリシーを含まない負荷分散方式の構成を参照してください。

静的近接度方式

August 15, 2023

仮想サーバーが静的近接方式を使用するように構成されている場合、仮想サーバーは近接基準に最も一致するサービスを選択します。

静的近接方式が機能するには、ロケーションファイルから入力された既存の静的近接データベースを使用するように NetScaler アプライアンスを構成するか、静的近接データベースにカスタムエントリを追加する必要があります。カスタムエントリを追加したら、そのロケーション修飾子を設定できます。データベースを設定したら、負荷分散方法として静的近接性を指定できます。

詳細については、以下のトピックを参照してください。

- ロケーションファイルの追加による静的近接データベースの作成
- 静的近接データベースへのカスタムエントリの追加
- ロケーション修飾子の設定
- スタティックプロキシミティ方式の指定

近接方法の指定

静的近接データベースを設定したら、GLSB 方式として静的近接を指定する準備が整います。

コマンドラインインターフェイスを使用して静的近接性を指定するには

コマンドプロンプトで次のコマンドを入力して静的近接を設定し、構成を確認します。

```
set lb vserver <name> -lbMethod STATICPROXIMITY

show lb vserver <name>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -lbMethod STATICPROXIMITY

show lb vserver
<!--NeedCopy-->
```

GUI を使用して静的近接を指定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2.「編集」をクリックし、「メソッド」セクションを展開します。

3.「負荷分散方法」リストで、「**STATICPROXIMITY**」を選択します。

> 注
>
> ProximityFromSelf パラメーターを有効にすると、クライアントの IP アドレスの代わりに Netscaler のループバック IP アドレスを使用して最も近いサーバーロケーションを取得し、静的近接負荷分散や GSLB 決定を行うことができます。

トークン方式

August 15, 2023

トークン方式を使用するように構成された負荷分散仮想サーバーは、クライアント要求から抽出されたデータセグメントの値に基づいてサービスを選択します。データセグメントはトークンと呼ばれます。トークンの場所とサイズを設定します。同じトークンを持つ後続の要求では、仮想サーバーは最初の要求を処理したのと同じサービスを選択します。

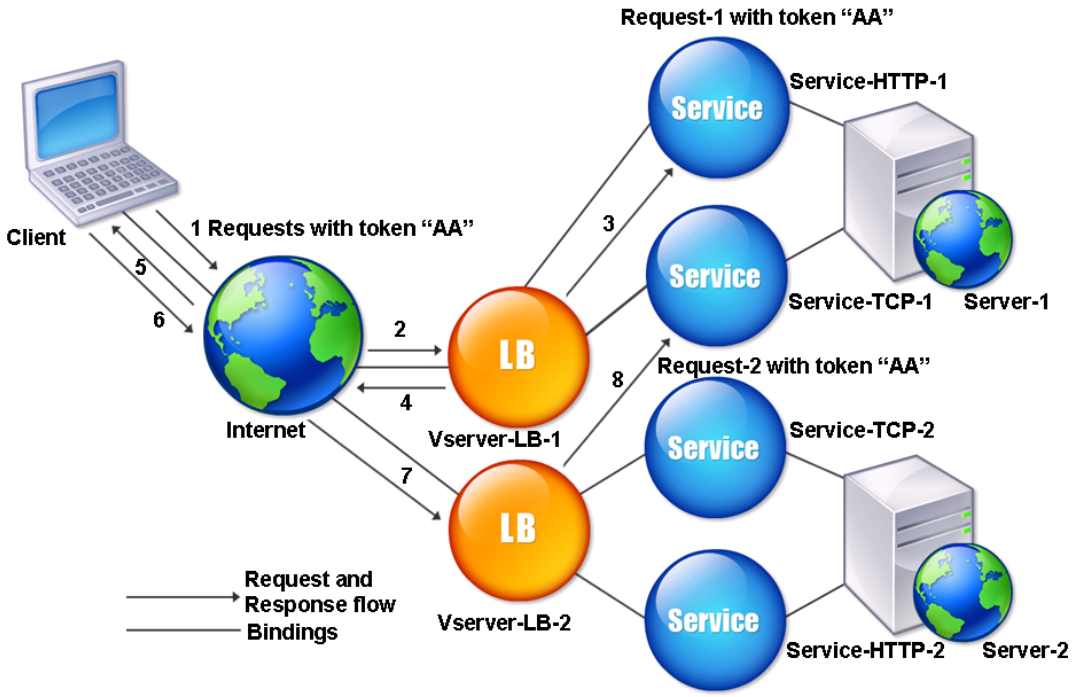
このメソッドはコンテンツ認識です。TCP、HTTP、および HTTPS 接続では、動作が異なります。HTTP サービスまたは HTTPS サービスの場合、トークンは HTTP ヘッダー、URL、または BODY にあります。トークンを検索するには、クラシックエクスプレッションまたはアドバンスエクスプレッションを指定または作成します。クラシック式または高度な式の詳細については、ポリシーの設定とリファレンスを参照してください。

HTTP サービスの場合、仮想サーバは TCP ペイロードの最初の 24 キロバイト（KB）で設定されたトークンを検索します。HTTP（TCP、SSL、SSL_TCP）以外のサービスの場合、仮想サーバーは 16 パケットの合計サイズが 24 KB 未満の場合、最初の 16 パケットで設定されたトークンを検索します。ただし、16 パケットの合計サイズが 24 KB を超える場合、アプライアンスは最初の 24 KB のペイロードでトークンを検索します。さまざまなタイプの仮想サーバーでこの負荷分散方法を使用すると、使用するプロトコルに関係なく、同じトークンを提示するリクエストが適切なサービスに送信されるようになります。

たとえば、Web コンテンツを含むサーバーで構成される負荷分散設定を考えてみましょう。リクエストの URL クエリ部分で特定の文字列（トークン）を検索するように NetScaler アプライアンスを構成する必要があります。サーバー 1 にはサービス HTTP-1 とサービス TCP-1 の 2 つのサービスがあり、サーバー 2 にはサービス HTTP-2 とサービス TCP-2 の 2 つのサービスがあります。TCP サービスは仮想サーバー-LB-2 にバインドされ、HTTP サービスは仮想サーバー-LB-1 にバインドされます。

VServer-LB-1 がトークン AA のリクエストを受け取ると、Service-HTTP-1 (サーバー-1 にバインドされている) サービスを選択してリクエストを処理します。VServer-LB-2 が同じトークン (AA) を使用して別のリクエストを受け取ると、このリクエストをサービスサービス-TCP-1 に転送します。次の図は、このプロセスを示しています。

図 1: トークンメソッドの仕組み



コマンドラインインターフェイスを使用してトークンロードバランシング方式を構成するには

コマンドプロンプトで次のコマンドを入力して、トークンの負荷分散方法を設定し、構成を確認します。

```
set lb vserver <name> -lbMethod TOKEN -rule <rule> -datalength <length>
    -dataoffset <offset>

show lb vserver <name>
<!--NeedCopy-->
```

例:

```
set lb vserver LB-VServer-1 -lbMethod TOKEN -rule 'AA' -datalength 2 -
   dataoffset 25

show lb vserver LB-VServer-1
<!--NeedCopy-->
```

構成ユーティリティを使用してトークンの負荷分散方法を設定するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。

2. [詳細設定] で [メソッド] をクリックします
3.「負荷分散方法」リストで、「トークン」を選択し、式を指定します。

最小リクエスト方式

February 15, 2024

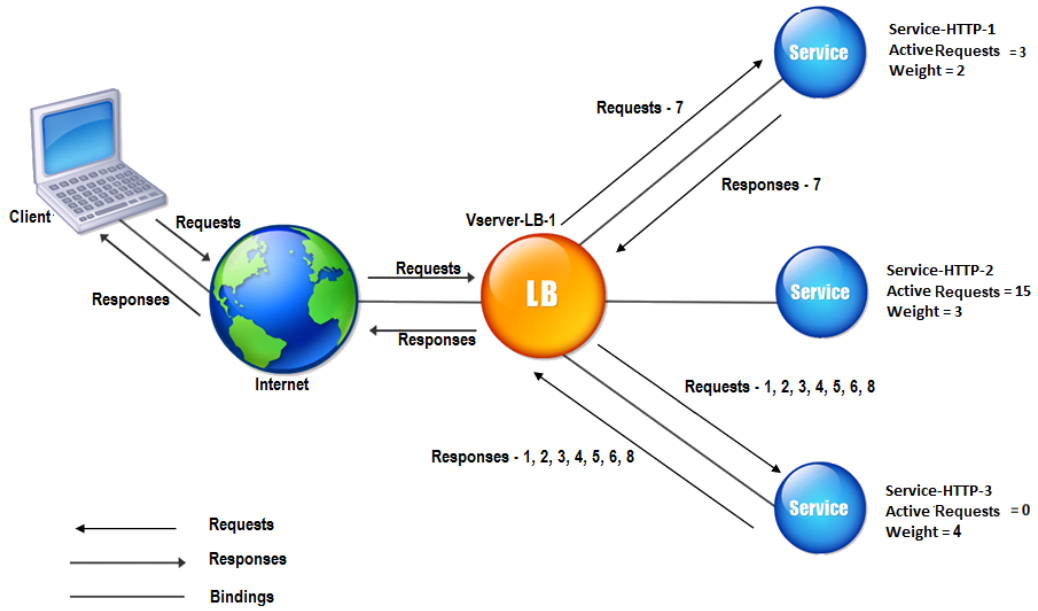
最小リクエスト方式を使用するように構成された負荷分散仮想サーバーは、受信したリクエストが最も少ないサービスを選択します。

たとえば、Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 の 3 つのサービスを考えてみます。

- サービス-HTTP-1 は 3 つのリクエストを処理しています。
- サービス-HTTP-2 は 15 個のリクエストを処理しています。
- サービス-HTTP-3 はリクエストを処理していません。

次の図は、NetScaler が最小リクエスト方式を使用して、受信するリクエストごとにサービスを選択する方法を示しています。

図 1: 最小リクエスト負荷分散方式の仕組み



この例では、仮想サーバーは、リクエストが最も少ないサーバーを選択して、受信リクエストごとにサービスを選択します。

リクエストは次のように転送されます:

- Service-HTTP-3 はリクエストを何も処理していないため、最初のリクエストを受信します。

  注: リクエストのないサービスが最初に選択されます。

- Service-HTTP-3 は、2 番目と 3 番目のリクエストを受信します。これは、サービスのリクエストの数が 2 番目に少ないためです。
- Service-HTTP-1 は、Service-HTTP-1 と Service-HTTP-3 のリクエスト数が同じであるため、4 番目のリクエストを受け取ります。仮想サーバーは、ラウンドロビン方式を使用してそれらのいずれかを選択します。
- Service-HTTP-3 は 5 番目の要求を受信します。
- Service-HTTP-1 は、Service-HTTP-1 と Service-HTTP-3 の両方が Service-HTTP-2 と同じ数の要求を処理するまで、6 番目の要求を受信します。次に、NetScaler は、サービスが最も負荷の少ないサービスになるか、その順番がラウンドロビンキューに入ると、Service-HTTP-2 への要求の転送を開始します。

> 注:
>
> Service-HTTP-2 へのリクエストが完了すると、他の 2 つのサービスのそれぞれが 15 個のリクエストを受け取る前に、新しいリクエストが届く可能性があります。

次の表は、負荷分散設定でリクエストがサービス間でどのように分散されるかを示しています。

受信リクエスト	サービス選択済み	現在のリクエスト数	注釈
Request-1	サービス-HTTP-3; (N = 0)	1	サービス-HTTP-3 のリクエストが最も少ないです。
Request-2	サービス-HTTP-3; (N = 1)	2	サービス-HTTP-3 のリクエストが最も少ないです。
Request-3	Service-HTTP-3; (N = 2)	3	-
Request-4	Service-HTTP-1; (N = 3)	4	サービス-HTTP-1 とサービス-HTTP-3 のリクエスト数は同じです。
Request-5	Service-HTTP-3; (N = 3)	4	サービス-HTTP-1 とサービス-HTTP-3 のリクエスト数は同じです。
Request-6	サービス-HTTP-1; (N = 4)	5	-
Request-7	Service-HTTP-3; (N = 4)	5	-
Request-8	Service-HTTP-1; (N = 5)	6	-

Service-HTTP-2 は、すべてのリクエストが完了したとき、または他のサービス（Service-HTTP-1 と Service-HTTP-3）にそれぞれ 15 個以上のリクエストがある場合に負荷分散のために選択されます。

NetScaler では、ウェイトをサービスに割り当てるときに最小リクエスト方式を使用することもできます。次の式の値 (Nw) を使用してサービスを選択します:

Nw = (アクティブなリクエストの数) * (10000/重量)

次の例は、サービスに重みが割り当てられているときに、NetScaler が最小リクエスト方式を使用して負荷分散するサービスを選択する方法を示しています。前の例では、サービス HTTP-1 に 2 の重みが割り当てられ、サービス HTTP-2 には 3 の重みが割り当てられ、サービス HTTP-3 には 4 の重みが割り当てられているとします。リクエストは次のように転送されます:

- Service-HTTP-3 は、サービスがリクエストを処理していないため、最初のリクエストを受信します。

  注: サービスがリクエストを処理しない場合、NetScaler は各サービスに割り当てられた重みに関係なくラウンドロビン方式を使用します。

- Service-HTTP-3 は、サービスの Nw 値が最も低いため、2 番目、3 番目、4 番目、5 番目、6 番目のリクエストを受け取ります。
- サービス-HTTP-1 は 7 番目のリクエストを受け取ります。Service-HTTP-1 と Service-HTTP-3 は同じ Nw 値を持つため、アプライアンスはラウンドロビン方式でロードバランシングを実行します。したがって、Service-HTTP-3 は 8 番目の要求を受信します。

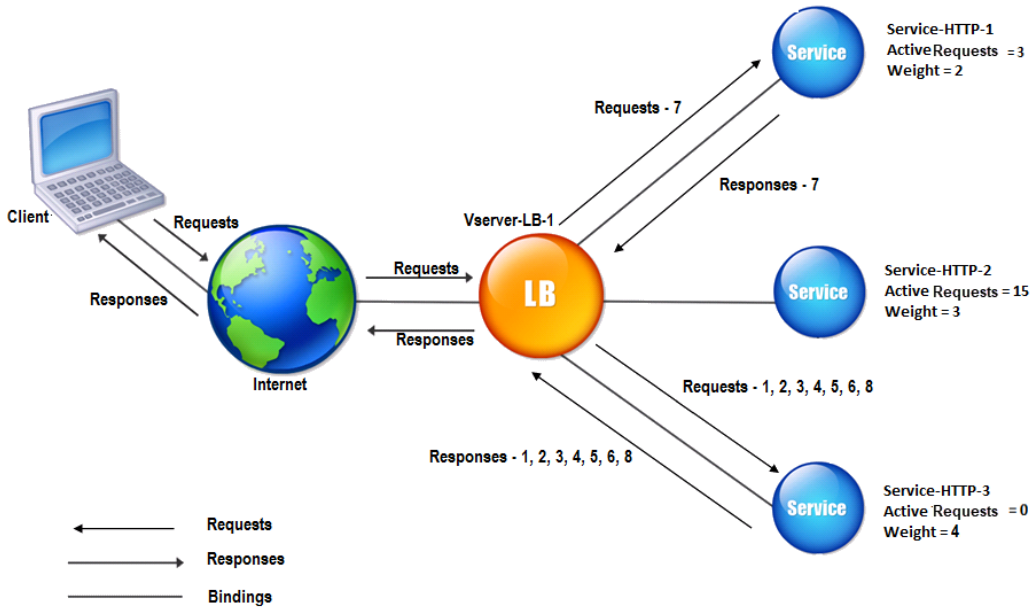
次の表は、前述の 3 つのサービスの負荷分散設定でリクエストがどのように分散されるかを示しています。

リクエストを受け取りました	サービス選択済み	現在の新規 (リクエスト数) * (10000/重量) 値	注釈
Request-1	サービス-HTTP-3; (新規 = 0)	Nw = 2500	サービス HTTP-3 の Nw 値は最も低いです。
Request-2	Service-HTTP-3; (New = 2500)	Nw = 5000	
Request-3	Service-HTTP-3; (Nw = 5000)	Nw = 7500	
Request-4	Service-HTTP-3; (New = 7500)	Nw = 10000	
Request-5	Service-HTTP-3; (Nw = 10000)	Nw = 12500	
Request-6	Service-HTTP-3; (Nw = 12500)	Nw = 15000	
Request-7	Service-HTTP-1; (Nw = 15000)	Nw = 20000	サービス HTTP-1 とサービス HTTP-3 の Nw 値は同じ
Request-8	Service-HTTP-3; (Nw = 15000)	Nw = 17500	

サービス-HTTP-2 は、要求が完了したとき、または他のサービス（サービス-HTTP-1 およびサービス-HTTP-3）の Nw 値が 50000 に等しいときに、負荷分散のために選択されます。

次の図は、サービスに重みが割り当てられているときに NetScaler がどのように最小リクエスト方式を使用するかを示しています。

図 2：重みが割り当てられた場合の最小リクエスト負荷分散方式の仕組み



最小リクエストメソッドを設定するには、「ポリシーを含まない負荷分散メソッドの設定」を参照してください。

ポリシーを含まない負荷分散方式を構成する

August 15, 2023

負荷分散設定用の負荷分散アルゴリズムを選択したら、そのアルゴリズムを使用するように NetScaler アプライアンスを構成する必要があります。CLI または設定ユーティリティを使用して設定できます。

注記:

トークン方式はポリシーベースであり、ここで説明するよりも多くの設定が必要です。トークンメソッドを設定するには、「トークンメソッド」を参照してください。

一部のハッシュベースの方法では、IP アドレスをマスクして、同じサブネットに属する要求を同じサーバーに送信できます。詳細については、「ハッシュメソッド」を参照してください。

コマンドラインインターフェイスを使用して負荷分散方法を設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -lbMethod <method>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -lbMethod LeastConnection
<!--NeedCopy-->
```

構成ユーティリティを使用して負荷分散方法を設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [方法] をクリックし、[負荷分散方法] リストで方法を選択します。

パーシステンスと固定接続

August 15, 2023

HTTP などの負荷分散ステートレスプロトコルは、永続性が構成されていない場合、クライアント接続に関する状態情報の保守を中断します。すべての送信が同じセッションの一部であっても、同じクライアントからの異なる送信が異なるサーバーに送信されることがあります。ショッピングカートアプリケーションなど、特定の種類の Web アプリケーションを処理する負荷分散仮想サーバーで永続性を構成できます。

パーシスタンスを設定する前に、さまざまなタイプのパーシスタンス、その使用方法、およびそれらの意味を理解する必要があります。次に、NetScaler ADC アプライアンスを構成して、それらを必要とする Web サイトおよび Web アプリケーションに永続的な接続を提供する必要があります。

また、バックアップパーシステンスを構成することもできます。これは、負荷分散仮想サーバーに対して構成されたプライマリタイプの永続性に障害が発生した場合に有効になります。グループの任意の仮想サーバへのクライアント転送を、同じクライアントから以前の転送を受信したサーバに転送できるように、永続性グループを設定できます。

RADIUS 負荷分散によるパーシステンスについては、永続性を使用した RADIUS 負荷分散の設定を参照してください。

パーシステンスについて

August 15, 2023

特定の負荷分散仮想サーバーに対して、いくつかのタイプの永続性から選択して、同じユーザーからショッピングカートアプリケーション、Web ベースの電子メール、またはその他のネットワークアプリケーションへのすべての接続を同じサービスにルーティングします。パーシステンスセッションは、指定した時間だけ有効です。

パーシステンスセッションに参加しているサーバが DOWN になると、ロードバランシング仮想サーバは設定されたロードバランシング方式を使用して新しいサービスを選択し、そのサービスによって表されるサーバとの新しいパーシステンスセッションを確立します。サーバーが OUT OF SERVICE になっても、既存の永続セッションの処理は継続しますが、仮想サーバーは新しいトラフィックをそのサーバーに送信しません。シャットダウン期間が経過すると、仮想サーバーは既存のクライアントからのサービスへの接続を停止し、既存の接続を閉じ、必要に応じてそれらのクライアントを新しいサービスにリダイレクトします。

構成する永続性の種類に応じて、NetScaler アプライアンスはソース IP、宛先 IP、SSL セッション ID、ホストまたは URL ヘッダー、またはこれらの組み合わせを調べて、各接続を適切な永続セッションに配置する場合があります。また、Web サーバーによって発行された Cookie、任意に割り当てられたトークン、または論理ルールに基づいてパーシステンスをベースにすることもできます。アプライアンスが適切な永続セッションと接続を一致させることができ、永続性の基礎として使用されるほとんどすべてのもの。

次の表は、NetScaler ADC アプライアンスで使用可能なパーシステンスタイプをまとめたものです。

永続性タイプ	説明
接続元 IP	ソース IP。同じクライアント IP アドレスからの接続は、同じ永続セッションの一部です。
HTTP クッキー	COOKIEINSERT. 同じ HTTP Cookie ヘッダーを持つ接続は、同じ持続性セッションの一部です。
SSL Session ID	SSLSESSION. 同じ SSL セッション ID を持つ接続は、同じ持続性セッションの一部です。
URL Passive	URLPASSIVE. 同じ URL への接続は、同じ永続セッションの一部として扱われます。
Custom Server ID	CUSTOMSERVERID. 同じ HTTP HOST ヘッダーを持つ接続は、同じ永続セッションの一部として扱われます。
接続先 IP	DESTIP. 同じ宛先 IP への接続は、同じ永続セッションの一部として扱われます。
送信元と送信先の IP	SRCIPDESTIP. 同じ送信元 IP からの接続と宛先 IP の両方からの接続は、同じ持続セッションの一部として扱われます。
SIP コール ID	CALLID. SIP ヘッダーに同じコール ID を持つ接続は、同じ永続セッションの一部として扱われます。
RTSP セッション ID	RTSPSID. 同じ RTSP セッション ID を持つ接続は、同じ永続セッションの一部として扱われます。
ユーザー定義ルール	RULE. ユーザー定義のルールに一致する接続は、同じ永続セッションの一部として扱われます。

表 1. パーシスタンスのタイプ

構成した永続性のタイプに応じて、仮想サーバーは、NetScaler アプライアンスの RAM 容量によって定められた制限まで、250,000 の同時持続接続または任意の数の永続接続のいずれかをサポートできます。次の表は、どのタイプのパーシスタンスが各カテゴリに分類されるかを示しています。

永続性タイプ	サポートされる同時持続接続数
送信元 IP、SSL セッション ID、ルール、宛先 IP、送信元 IP/宛先 IP、SIP コール ID、RTSP セッション ID	250 K
クッキー、URL サーバー ID、カスタムサーバー ID	メモリの上限。CookieInsert では、タイムアウトが 0 以外の場合、接続数はメモリによって制限されます。

表 2. パーシステンスの種類とサポートされる同時接続数

パーシスタンスには、特定のタイプの仮想サーバーに固有のものもあります。次の表は、各タイプのパーシスタンスを一覧表示し、どのタイプのパーシスタンスがどのタイプの仮想サーバーでサポートされているかを示しています。

永続性タイプ	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
SOURCEIP	はい	はい	はい	はい	はい	はい	いいえ	いいえ
COOKIEINSERT	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
SSLSESSION	いいえ	はい	いいえ	いいえ	はい	はい	いいえ	いいえ
URLPASSIVE	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
CUSTOMSERVERID	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RULE	はい	はい	はい	いいえ	いいえ		いいえ	いいえ
SRCIPDESTIP	はい	はい	はい	はい	はい	はい	いいえ	いいえ
DESTIP	はい	はい	はい	はい	はい	はい	いいえ	いいえ
CALLID	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	はい
RTSPID	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	はい	いいえ

表 3. パーシステンスタイプと仮想サーバータイプの関係

送信元 **IP** アドレスのパーシステンス

August 15, 2023

ソース IP パーシステンスが構成されている場合、負荷分散仮想サーバーは、設定された負荷分散方法を使用して最初のリクエストのサービスを選択し、次にソース IP アドレス (クライアント IP アドレス) を使用してそのクライアントからの後続のリクエストを識別し、同じサービスに送信します。セッションの最大非アクティブ期間を指定するタイムアウト値を設定できます。タイムアウト値が満了すると、セッションは破棄され、設定された負荷分散アルゴリズムを使用して新しいサーバーが選択されます。

注意: 状況によっては、ソース IP アドレスに基づくパーシステンスを使用すると、サーバーが過負荷になることがあります。単一の Web サイトまたはアプリケーションに対するすべての要求は、複数の場所にリダイレクトされる場合でも、単一のゲートウェイを介して NetScaler ADC アプライアンスにルーティングされます。複数のプロキシ環境では、クライアント要求が同じクライアントから送信される場合でも、クライアント要求が異なる送信元 IP アドレスを持つことがよくあります。その結果、単一のセッションを作成する必要がある永続性セッションが急速に増加します。この問題は、「メガプロキシの問題」と呼ばれます。これを防ぐには、ソース IP ベースの永続性の代わりに HTTP cookie ベースの永続性を使用できます。

送信元 IP アドレスに基づいてパーシステンスを設定するには、ルールを必要としない永続性のタイプの設定を参照してください。

注：すべての着信トラフィックがネットワークアドレス変換（NAT）デバイスまたはプロキシの背後から送られる場合、NetScaler ADC アプライアンスでは単一の送信元 IP アドレスから送られるように見えます。これにより、送信元 IP パーシステンスが正しく機能しなくなります。このような場合は、別のパーシステンスタイプを選択する必要があります。

HTTP Cookie のパーシステンス

August 15, 2023

HTTPCookie パーシステンスを構成すると、NetScaler アプライアンスは最初のクライアントリクエストの HTTP ヘッダーに Cookie を設定します。Cookie には、負荷分散アルゴリズムによって選択されたサービスの IP アドレスとポートが含まれます。他の HTTP 接続と同様に、クライアントはその Cookie を後続のリクエストに含めます。

NetScaler アプライアンスが Cookie を検出すると、接続の持続性を維持したまま、そのリクエストをサービス IP と Cookie 内のポートに転送します。このタイプのパーシスタンスは、HTTP または HTTPS タイプの仮想サーバーで使用できます。このパーシステンスタイプはアプライアンスリソースを消費しないため、永続的なクライアントの数に制限はありません。

注: クライアントの Web ブラウザが Cookie を拒否するように構成されている場合、HTTP Cookie ベースの永続性は機能しません。ウェブサイトでクッキーチェックを設定し、クッキーを適切に保存していないように見えるクライアントに、ウェブサイトで使用する場合は Cookie を有効にする必要があることを警告することをお勧めします。

NetScaler ADC アプライアンスが挿入するクッキーの形式は次のとおりです。

```
NSC_XXXX=<ServiceIP ><ServicePort>
```

各項目の意味は次のとおりです:

- NSC_XXXX は、仮想サーバー名から派生した仮想サーバー ID です。
- ServiceIP と ServicePort は、それぞれサービス IP アドレスとサービスポートをエンコードして表現したものです。IP アドレスとポートは別々にエンコードされます。

このタイプの永続性のタイムアウト値を設定して、セッションの非アクティブ期間を指定できます。指定された期間接続がアクティブでない場合、NetScaler ADC アプライアンスは永続セッションを破棄します。同じクライアントからの後続の接続では、設定された負荷分散方式に基づいて新しいサーバが選択され、新しい永続セッションが確立されます。

注：タイムアウト値を 0 に設定すると、NetScaler アプライアンスは有効期限を指定せず、クライアントのブラウザをシャットダウンしても保存されないセッション Cookie を設定します。

デフォルトでは、NetScaler アプライアンスはクライアントブラウザーとの互換性を最大限に高めるために HTTP バージョン 0 のクッキーを設定します。（バージョン 1 の Cookie を理解できるのは特定の HTTP プロキシのみで、最も一般的に使用されているブラウザでは理解できません）。RFC2109 に準拠するため、HTTP バージョン 1 の Cookie を設定するようにアプライアンスを設定できます。HTTP バージョン 0 Cookie の場合、アプライアンスはクッキーの有効期限の日付と時刻を絶対協定世界時（GMT）として挿入します。この値は、アプライアンスの現在の GMT 時間とタイムアウト値の合計として計算されます。HTTP バージョン 1 Cookie の場合、アプライアンスは HTTP クッキーの「Max-Age」属性を設定することによって相対的な有効期限を挿入します。この場合、クライアントのブラウザは実際の有効期限を計算します。

アプライアンスによって挿入された Cookie に基づいてパーシステンスを設定するには、ルールを必要としない永続性タイプの設定を参照してください。

HTTP Cookie では、アプライアンスはデフォルトで Cookie がスクリプト不能であり、クライアントアプリケーションに公開してはならないことを示すHTTPOnlyフラグを設定します。したがって、クライアント側のスクリプトは Cookie にアクセスできず、クライアントはクロスサイトスクリプティングの影響を受けません。

ただし、一部のブラウザはHTTPOnlyフラグをサポートしていないため、Cookie を返さない場合があります。その結果、永続性が壊れます。このフラグをサポートしていないブラウザでは、永続性 Cookie のHTTPOnlyフラグを省略できます。

CLI を使用して HTTPOnly フラグ設定を変更するには

コマンドプロンプトで入力します。

```
set lb parameter -httpOnlyCookieFlag (ENABLED|DISABLED)
<!--NeedCopy-->
```

例:

```
>  set lb parameter -httpOnlyCookieFlag disabled
   Done
> show lb parameter
  Global LB parameters:
       Persistence Cookie HttpOnly Flag: DISABLED
```



```
        Use port for hash LB: YES
 Done
<!--NeedCopy-->
```

GUI を使用して HTTPOnly フラグの設定を変更するには

1. [トラフィック管理] > [ロードバランシング] > [ロードバランシングパラメータの設定] に移動し、[パーシステンス **Cookie HttpOnly**] フラグを選択またはクリアします。

クッキーの暗号化

リリース 10.5 ビルド 55.8 以降、SSL 暗号化に加えて Cookie を暗号化できるようになりました。

コマンドラインインターフェイスを使用して **Cookie** を暗号化するには、コマンドプロンプトで次のように入力します

```
set lb parameter -UseEncryptedPersistenceCookie ENABLED -
   cookiePassphrase test
<!--NeedCopy-->
```

設定ユーティリティを使用して **Cookie** を暗号化するには

1. **[**** トラフィック管理] > [負荷分散パラメータの変更] に移動し、[パーシステンス **Cookie** 値のエンコード] を選択し、**[Cookie** パスフレーズ] にパスフレーズを入力します。**

SSL セッション ID のパーシステンス

August 15, 2023

SSL セッション ID の永続性が構成されている場合、NetScaler ADC アプライアンスは SSL ハンドシェイクプロセスの一部である SSL セッション ID を使用して、最初の要求がサービスに送信される前に永続セッションを作成します。負荷分散仮想サーバーは、同じ SSL セッション ID を持つ後続の要求を同じサービスに送信します。このタイプの永続性は SSL ブリッジサービスに使用されます。

注記:

このタイプの永続性を選択する前に、ユーザーが考慮する必要のある問題が 2 つあります。まず、このタイプのパーシステンスは、NetScaler ADC アプライアンス上のリソースを消費します。これにより、サポートできる同時パー

シスタンスセッションの数が制限されます。複数のパーシステンス・セッションをサポートする場合は、別のタイプの永続性を選択できます。

次に、クライアントと負荷分散サーバーがトランザクション中にセッション ID を再ネゴシエートする必要がある場合、永続性は維持されず、クライアントの次の要求を受信したときに新しい永続セッションが作成されます。これにより、Web サイトでのクライアントのアクティビティが中断され、クライアントからセッションの再認証または再起動を求められることがあります。また、タイムアウトの値が大きすぎると、多数の放棄セッションが発生する可能性があります。

SSL セッション ID に基づいてパーシステンスを設定するには、ルールを必要としない永続性のタイプの構成を参照してください。

> 注
>
> セッション・チケットでは、SSL セッション ID の永続化はサポートされていません。

SSL セッション ID のパーシステンスのサポートをバックアップする

NetScaler リリース 12.0 ビルド 56.20 から、ソース IP の永続性は、SSL セッション ID の永続性のバックアップパーシステンスタイプとしてサポートされています。クライアントと負荷分散されたサーバーがセッションを再ネゴシエートし、ソース IP パーシスタンスがバックアップパーシスタンスとして設定されている場合、クライアント要求は同じサーバーに転送されます。

SSL セッション ID のバックアップ永続性をサポートするため、NetScaler アプライアンスはクライアント要求を初めて受信したときに、ソース IP と SSL セッション ID の両方のセッションエントリを作成します。同じセッション ID を含む後続のリクエストでは、SSL セッション ID が使用されます。ただし、クライアントと負荷分散されたサーバーがセッションを再ネゴシエートすると、クライアント要求は、ソース IP 永続性を使用して同じサーバーに転送され、新しい SSL セッション ID 永続性エントリが作成されます。

バックアップパーシステンスの構成の詳細については、バックアップ永続性の構成を参照してください。

Diameter の AVP 番号のパーシステンス

August 15, 2023

Diameter メッセージの属性値ペア (AVP) 番号に基づくパーシスタンスを使用して、永続的な Diameter セッションを作成できます。NetScaler アプライアンスが Diameter ameter メッセージで AVP を検出すると、AVP の値に基づいてパーシステンスセッションを作成します。AVP の値と一致する後続のメッセージはすべて、以前に選択したサーバーに転送されます。AVP の値がパーシスタンスセッションと一致しない場合、新しい値に対して新しいセッションが作成されます。

注：AVP 番号が Diameter ベースプロトコル RFC 6733 で定義されておらず、その番号がグループ化された AVP 内にネストされている場合は、AVP 番号のシーケンス（最大 3）を親から子の順に定義する必要があります。たとえば、

持続する AVP 番号 X が Z にネストされている AVP Y 内にネストされている場合、リストを Z Y X として定義します。

コマンドラインインターフェイスを使用して仮想サーバーで **Diameter** ベースの永続性を構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
set lb vserver <name> -PersistenceType <type-> persistAVPno <
   positive_integer>
<!--NeedCopy-->
```

例**:**

```
set lb vserver diameter_vs -persistenceType DIAMETER -persistAVPno 263
<!--NeedCopy-->
```

カスタムサーバー **ID** のパーシステンス

August 15, 2023

カスタムサーバー ID の永続化方法では、クライアントリクエストで指定されたサーバー ID を使用して永続性を維持します。このタイプのパーシスタンスを機能させるには、まずサービスにサーバー ID を設定する必要があります。NetScaler ADC アプライアンスは、クライアント要求の URL をチェックし、指定されたサーバー ID に関連付けられたサーバーに接続します。サービスプロバイダーは、特定のサービスに対するリクエストで提供されるサーバー ID をユーザーが認識していることを確認する必要があります。

たとえば、サイトがイメージ、テキスト、マルチメディアなどの異なるタイプのデータを異なるサーバから提供している場合は、各サーバにサーバ ID を割り当てることができます。NetScaler アプライアンスでは、対応するサービスのサーバー ID を指定し、対応する負荷分散仮想サーバーでカスタムサーバー ID の永続性を構成します。リクエストを送信する際、クライアントは必要なデータタイプを示す URL にサーバー ID を挿入します。

カスタムサーバー ID パーシステンスを設定するには:

- 負荷分散の設定で、永続性を維持するためにユーザー定義のサーバー ID を使用する各サービスにサーバー ID を割り当てます。英数字のサーバー ID を使用できます。
- サーバー ID の URL クエリを調べ、対応するサーバーにトラフィックを転送するルールをデフォルトの構文表現言語で指定します。
- カスタムサーバー ID パーシステンスを設定します。

注: パーシステンスタイムアウト値は、カスタムサーバー ID のパーシステンスタイプには影響しません。この永続性タイプにはクライアント情報が保存されないため、永続クライアントの最大数に制限はありません。

例**:**

2 つのサービスを含む負荷分散設定では、サーバー ID 2345-photo-56789 をサービス 1 に、サーバー ID 2345-drawing-abb123 をサービス 2 に割り当てます。これらのサービスを Web11 という名前の仮想サーバーにバインドします。

```
set service Service-1 10.102.29.5 -CustomServerID 2345-photo-56789

set service Service-2 10.102.29.6 -CustomServerID 2345-drawing-abb123
<!--NeedCopy-->
```

仮想サーバー Web11 で、カスタムサーバー ID の永続性を有効にします。

次の式を作成して、文字列「sid=」を含むすべての URL クエリが検査されるようにします。

HTTP.REQ.URL.AFTER_STR(“sid=”)

例**:**

```
set lb vserver Web11 -persistenceType customserverID -rule "HTTP.REQ.
   URL.AFTER_STR("sid=")"

bind lb vserver Web11 Service-[1-2]
<!--NeedCopy-->
```

クライアントが次の URL を含むリクエストを Web11 の IP アドレスに送信すると、アプライアンスはその要求を Service-2 に転送し、パーシステンスを尊重します。

例**:**

```
http://www.example.com/index.asp?&sid=2345-drawing-abb123
```

デフォルト構文ポリシー式の詳細については、「ポリシーの設定およびリファレンス」を参照してください。

構成ユーティリティを使用してカスタムサーバー **ID** の永続性を構成するには

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
2. サービスを開き、サーバー ID を設定します。
3. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
4. 「詳細設定」で、「持続性」を選択します。
5. 「カスタム」(CUSTOMESERVERID) を選択し、式を指定します。

IP アドレスのパーシステンス

August 15, 2023

パーシステンスは、宛先 IP アドレス、または送信元 IP アドレスと宛先 IP アドレスの両方に基づいて設定できます。

宛先 **IP** アドレスに基づくパーシステンス

宛先 IP アドレスベースの永続性では、NetScaler ADC アプライアンスが新しいクライアントから要求を受信すると、仮想サーバーによって選択されたサービスの IP アドレス（宛先 IP アドレス）に基づいて永続性セッションが作成されます。後で、同じ宛先 IP に要求を同じサービスに送信します。このタイプの永続性は、リンク負荷分散で使用されます。リンク負荷分散の詳細については、リンク負荷分散を参照してください。

宛先 IP パーシステンスのタイムアウト値は、送信元 IP アドレスに基づく永続性で説明されている送信元 IPパーシステンスのタイムアウト値と同じです。

宛先 IP アドレスに基づいてパーシステンスを設定するには、ルールを必要としない永続性のタイプの設定を参照してください。

送信元および宛先 **IP** アドレスに基づくパーシステンス

送信元と宛先の IP アドレスベースのパーシステンスでは、NetScaler ADC アプライアンスが要求を受信すると、クライアントの IP アドレス（送信元 IP アドレス）と仮想サーバーによって選択されたサービスの IP アドレス（宛先 IP アドレス）の両方に基づいてパーシステンスセッションが作成されます。後で、同じ送信元 IP および同じ宛先 IP からの要求を同じサービスに送信します。

宛先 IP パーシステンスのタイムアウト値は、送信元 IP アドレスに基づく永続性で説明されている送信元 IPパーシステンスのタイムアウト値と同じです。

送信元 IP アドレスと宛先 IP アドレスの両方に基づいてパーシステンスを設定するには、ルールを必要としない永続性のタイプの設定を参照してください。

SIP Call ID のパーシステンス

August 15, 2023

SIP コール ID パーシステンスを使用すると、NetScaler アプライアンスは SIP ヘッダーのコール ID に基づいてサービスを選択します。これにより、特定の SIP セッションのパケットを同じサービス、つまり同じ負荷分散サーバーに送信できます。このパーシステンスタイプは、特に SIP ロードバランシングに適用されます。SIP ロードバランシングの詳細については、「SIP サービスのモニタリング」を参照してください。

SIP コール ID に基づいてパーシステンスを設定するには、ルールを必要としない永続性タイプの設定を参照してください。

RTSP セッション ID のパーシステンス

August 15, 2023

RTSP セッション ID 永続性では、NetScaler ADC アプライアンスが新しいクライアントから要求を受信すると、RTSP パケットヘッダーにリアルタイムストリーミングプロトコル（RTSP）セッション ID に基づいて永続セッションが作成され、構成された負荷分散によって選択された RTSP サービスに要求が送信されます。メソッド。これは、同じセッション ID を含む後続の要求を同じサービスに送信します。このパーシステンスタイプは、特に SIP ロードバランシングに適用されます。SIP ロードバランシングの詳細については、「SIP サービスのモニタリング」を参照してください。

注：RTSP セッション ID パーシステンスは、RTSP 仮想サーバ上でデフォルトで設定されており、その設定を変更することはできません。

異なる RTSP サーバが同じセッション ID を発行することがあります。この場合、RTSP セッション ID のみを使用して、クライアントと RTSP サーバ間で一意のセッションを作成できません。同じセッション ID を発行する可能性のある複数の RTSP サーバがある場合は、セッション ID にサーバの IP アドレスとポートを追加するようにアプライアンスを設定して、永続性を確立するために使用できる一意のトークンを作成できます。これをセッション ID マッピングと呼びます。

RTSP セッション ID に基づいてパーシステンスを設定するには、ルールを必要としない永続性タイプの設定を参照してください。

重要: セッション ID マッピングを使用する必要がある場合は、負荷分散セットアップ内で各サービスを構成するときに次のパラメータを設定する必要があります。また、非永続的な接続が RTSP 仮想サーバーを経由しないようにしてください。

URL パッシブパーシステンスを構成する

October 25, 2023

URL パッシブパーシステンスでは、NetScaler アプライアンスがクライアントから要求を受け取ると、クライアントの要求からサーバーの IP アドレスとポート情報（単一の 16 進数で表される）を抽出します。

URL パッシブ永続性では、サーバの IP アドレスポート情報を含むクエリー要素を指定する高度な式を設定する必要があります。クラシックおよび高度なポリシー式の詳細については、「ポリシーと式」を参照してください。

次の式は、文字列「urlp=」を含む URL クエリーの要求を検査し、サーバーの IP アドレスポート情報を抽出し、16 進数の文字列から IP およびポート番号に変換し、要求をこの IP アドレスで構成されたサービスに転送するようにアプライアンスを構成します。ポート番号。

HTTP.REQ.URL.AFTER_STR(“urlp=”)

URL パッシブパーシスタンスが有効で、前の式が設定されている場合、次の URL とサーバの IP アドレスポート文字列を含む要求は 10.102.29.10:80 に送信されます。

```
http://www.example.com/index.asp?&urlp=0A661D0A0050
```

永続性のタイムアウト値は、この永続タイプには影響しません。サーバーの IP アドレス/ポート情報をクライアント要求から抽出できる限り、永続性は維持されます。このパーシステンスタイプはアプライアンスリソースを消費しないため、永続的なクライアントの数に制限はありません。

URL パッシブパーシスタンスを設定するには、ルールを必要としない永続性のタイプの設定で説明されているように、まずパーシステンスを設定します。パーシステンスタイプを URLPASSIVE に設定します。次に、次の手順を実行します。

CLI を使用して URL パッシブパーシステンスを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <vserverName>  [-persistenceType <persistenceType>]  [-rule <expression>]
<!--NeedCopy-->
```

例:

```
set lb vserver LB-VServer-1 -persistenceType URLPASSIVE  -rule HTTP.REQ.URL.AFTER_STR( “urlp=” )
<!--NeedCopy-->
```

GUI を使用して仮想サーバーのパーシステンスを設定するには

1. [** トラフィック管理 **] > [** 負荷分散 **] > [** 仮想サーバー **] に移動し、仮想サーバーを開きます。
2. [永続性] セクションで、要件を満たす永続性タイプを選択します。仮想サーバーに最適なパーシステンスタイプは、オプションボタンとして使用できます。特定の仮想サーバータイプに適用できる他のパーシステンスタイプは、「その他」リストから選択できます。

> 注:
>
> NetScaler リリース 12.0 ビルド 56.20 より前のバージョンでは、オプションボタンがない単一の永続性ドロップダウンリストですべてのパーシステンスタイプを使用できます。

ユーザー定義の規則に基づいてパーシステンスを構成する

October 25, 2023

> 警告:
>
> 負荷分散機能の永続性ルールに対する従来の式の使用は削除され、NetScaler ADC アプライアンスリリース 13.1 以降のフィルタルールでは使用できなくなります。NetScaler ADC コマンドラインインターフェイス、NetScaler GUI、または Nitro オートメーションでこれらのポリシー式を使用しないことをお勧めします。詳細については、「クラシックポリシーの非推奨 FAQ 」ページの表 1 および表 2 を参照してください。

ルールベースのパーシステンスを構成すると、NetScaler ADC アプライアンスは、一致するルールの内容に基づいてパーシステンスセッションを作成してから、設定された負荷分散方式で選択されたサービスにリクエストを送信します。後で、ルールに一致するすべてのリクエストを同じサービスに送信します。HTTP、SSL、RADIUS、ANY、TCP、および SSL_TCP タイプのサービスに対して、規則ベースの永続性を設定できます。

ルールベースの永続性には、クラシックまたは高度なポリシー式が必要です。従来の式を使用してリクエストヘッダーを評価することも、高度なポリシー式を使用してリクエストヘッダー、リクエスト内の Web フォームデータ、レスポンスヘッダー、またはレスポンス本文を評価することもできます。たとえば、従来の式を使用して、HTTP Host ヘッダーの内容に基づいて永続性を設定できます。また、高度なポリシー式を使用して、応答 Cookie またはカスタムヘッダーのアプリケーションセッション情報に基づいて永続性を設定することもできます。クラシックポリシー式と高度なポリシー式の作成と使用の詳細については、「ポリシーと式」を参照してください。

設定できる式は、ルールベースの永続性を構成するサービスのタイプによって異なります。たとえば、RADIUS 以外のプロトコルでは RADIUS 固有の式を使用できず、ANY タイプ以外のサービスタイプでは TCP オプションベースの式を使用できません。TCP および SSL_TCP サービスタイプでは、TCP/IP プロトコルデータ、レイヤ 2 データ、TCP オプション、および TCP ペイロードを評価する式を使用できます。

注意:TCP 経由で送信される財務情報交換（「FIX」）プロトコルデータに基づくルールベースの永続性を構成するユースケースについては、TCP バイトストリーム内の名前と値のペアに基づくルールベースの永続性の構成を参照してください。

ルールベースの永続性は、Citrix SD-WAN アプライアンス、Citrix SD-WCitrix SD-WAN プラグイン、キャッシュサーバー、アプリケーションサーバーなどのエンティティで永続性を維持するために使用できます。

注意: ANY 仮想サーバーでは、応答に対してルールベースの永続性を設定できません。

ユーザー定義のルールに基づいてパーシステンスを設定するには、「ルールを必要としない永続性のタイプの設定」の説明に従ってパーシステンスを設定し、永続性タイプを RULE に設定します。その後、次の手順を実行できます。設定ユーティリティまたは CLI を使用して、ルールベースの永続性を設定できます。

CLI を使用してユーザ定義ルールに基づいてパーシステンスを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <vserverName> [-rule <expression>][-resRule <expression
    >]
<!--NeedCopy-->
```


例:

```
set lb vserver vsvr_name -rule http.req.header("cookie").value(0).
   typecast_nvlist_t('=',';').value("server")

set lb vserver vsvr_name -resrule http.res.header("set-cookie").value
   (0).typecast_nvlist_t('=',';').value("server")

<!--NeedCopy-->
```

GUI を使用してユーザー定義ルールに基づいてパーシステンスを設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [永続性] セクションで、要件を満たす永続性タイプを選択します。仮想サーバーに最適なパーシステンスタイプは、オプションボタンとして使用できます。特定の仮想サーバータイプに適用できる他のパーシステンスタイプは、「その他」リストから選択できます。

> 注
>
> NetScaler リリース 12.0 ビルド 56.20 より前のバージョンでは、オプションボタンがない単一の永続性ドロップダウンリストですべてのパーシステンスタイプを使用できます。

例: リクエストペイロードのクラシック式

次の古典的な式は、文字列「MyBrowser」を含む User-Agent HTTP ヘッダーの存在に基づいて永続セッションを作成し、このヘッダーと文字列を含む後続のクライアント要求を、最初の要求で選択された同じサーバーに送信します。

```
http header User-Agent contains MyBrowser
<!--NeedCopy-->
```

例: リクエストヘッダーの高度なポリシー式

以下の高度なポリシー式は、前のクラシック式と同じ処理を行います。

HTTP.REQ.HEADER(“User-Agent”).CONTAINS (“MyBrowser”)

例: レスポンス **Cookie** の高度なポリシー式

次の式は、「サーバー」Cookie のレスポンスを調べ、その Cookie を含むすべてのリクエストを、最初のリクエストで選択されたサーバーと同じサーバーに送信します。

HTTP.RES.HEADER(“SET-COOKIE”).VALUE(0).TYPECAST_NVLIST_T(‘=’,’;’).VALUE(“server”)

規則を必要としないパーシステンスタイプを構成する

October 25, 2023

パーシステンスを構成するには、基本的な負荷分散の設定の説明に従って、まず負荷分散仮想サーバーをセットアップする必要があります。次に、仮想サーバーで永続性を構成します。

CLI を使用して仮想サーバーのパーシステンスを設定するには

コマンドプロンプトで次のコマンドを入力して永続性を設定し、構成を確認します。

```
set lb vserver <name> -PersistenceType <type> [-timeout <integer>]

show lb vserver
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -persistenceType SOURCEIP -timeout 60

show lb vserver
<!--NeedCopy-->
```

タイムアウトは、永続セッションが有効な期間です。タイムアウトのデフォルト値と最小値 (分単位) は、次の表に示すように、パーシステンスタイプによって異なります。

パーシステンスタイプ	デフォルト値	最小値	最大値
Cookie インサート/グループクッキーインサート	2	0	1440
その他のパーシステンスタイプ	2	2	1440

> 注
>
> - Group Cookie 挿入パーシステンスタイプは、負荷分散グループに設定できます。
> - IP ベースのパーシステンスの場合は、PersistMask パラメーターを設定することもできます。
> - パーシステンスタイプはデフォルトで NONE に設定されています。

GUI を使用して仮想サーバーのパーシステンスを設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。

2. [永続性] セクションで、要件を満たす永続性タイプを選択します。仮想サーバーに最適なパーシステンスタイプは、オプションボタンとして使用できます。特定の仮想サーバータイプに適用できる他のパーシステンスタイプは、「その他」リストから選択できます。

> 注： NetScaler リリース 12.0 ビルド 56.20 より前のバージョンでは、オプションボタンのない 1 つの持続性ドロップダウンリストですべての持続性タイプを使用できました。

バックアップのパーシステンスを構成する

August 15, 2023

プライマリ永続タイプが失敗したときに、ソース IP パーシステンスタイプを使用するように仮想サーバを設定できます。

次の表に、プライマリおよびセカンダリのバックアップパーシステンスタイプの組み合わせ、およびバックアップパーシステンスを使用する場合の条件を示します。

プライマリー・パーシスタンス	バックアップ持続性	プライマリ・パーシスタンス検索が失敗すると…
クッキーインサート	接続元 IP	アプライアンスは、クライアントブラウザがリクエストで Cookie を返さない場合にのみ、ソース IP ベースのパーシステンスにフォールバックします。ただし、ブラウザが Cookie（パーシステンスクッキーである必要はありません）を返した場合、そのブラウザは Cookie をサポートしていると見なされるため、バックアップパーシスタンスはトリガーされません。
規則	接続元 IP	ルールで指定されたパラメータが受信リクエストに含まれていない場合、アプライアンスはソース IP ベースの永続性を使用します。

> 注
>
> - プライマリ永続タイプが HTTP-Cookie ベースの永続性で、バックアップ永続性タイプが Source IP ベースの場合は、バックアップパーシステンスのタイムアウト値を設定できます。手順については、アイド

> ル状態のクライアント接続のタイムアウト値の設定を参照してください。
> - プライマリパーシステンスがルールベースの場合はバックアップパーシステンスのタイムアウト値を設定できません。この場合、セカンダリパーシステンスのタイムアウト値はプライマリパーシステンスのタイムアウト値と同じである必要があります。したがって、プライマリとセカンダリの有効期限は同時に切れます。

コマンドラインインターフェイスを使用して仮想サーバーのバックアップ永続性を設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -persistenceType <PersistenceType> -
    persistenceBackup <BackupPersistenceType>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -persistenceType CookieInsert -
    persistenceBackup SourceIP

set lb vserver Vserver-LB-1 -persistenceType sslsession -
    persistenceBackup SourceIP

set lb vserver Vserver-LB-1  - persistenceType RULE  - rule http.req.
    header("User-Agent").value(0).contains("MyBrowser")  -
    persistenceBackup SOURCEIP

set lb vserver Vserver-LB-1 -persistenceType sslsession -
    persistenceBackup SourceIP
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーのバックアップ永続性を設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[永続性] を選択し、バックアップパーシステンスのタイプを指定します。

注: プライマリパーシスタンスは COOKIEINSERT、RULE、または SSLSESSION に設定する必要があります。

パーシステンスグループを構成する

August 15, 2023

複数の異なる種類の接続を処理する負荷分散サーバー (マルチメディアをホストする Web サーバーなど) がある場合、これらの接続を処理するように仮想サーバーグループを構成できます。仮想サーバーグループを作成するには、負荷

分散サーバーが受け入れる接続の種類ごとに異なる種類の仮想サーバーを 1 つのグループにバインドします。次に、グループ全体のパーシステンスタイプを設定します。

パーシスタンスグループには、ソース IP ベースのパーシステンスまたは HTTP Cookie ベースのパーシステンスのいずれかを設定できます。グループ全体に永続性を設定した後は、グループ内の個々の仮想サーバーの永続性を変更することはできません。グループにパーシステンスを設定してから新しい仮想サーバーをグループに追加すると、新しい仮想サーバーのパーシスタンスはグループのパーシスタンス設定と一致するように変更されます。

仮想サーバーのグループに永続性が構成されている場合、各クライアント要求を受信するグループ内の仮想サーバーに関係なく、最初の要求に対して永続セッションが作成され、その後の要求は、最初の要求と同じサービスに送信されます。

持続性セッションを持つ仮想サーバーを、異なる持続性タイプの負荷分散グループに追加すると、古い持続性タイプに固有の既存の永続セッションは削除されます。永続セッションは、トラフィックを同じ仮想サーバーに送信するか、別のサーバーに送信する必要があるかを決定します。したがって、既存の確立された接続は影響を受けません。

負荷分散グループのパーシステンスタイプは、仮想サーバーのプロトコルタイプに関係なく、そのグループにバインドされているすべての仮想サーバーに適用されます。負荷分散グループは次の持続性タイプをサポートします。

- SourceIP
- CookieInsert
- 規則

一部の仮想サーバーは、特定の持続性タイプのみをサポートします。たとえば、SSL_BRIDGE タイプの仮想サーバーは、LB グループに SourceIP パーシスタンスタイプのみを使用できます。

HTTP クッキーベースのパーシステンスを設定すると、HTTP Cookie のドメイン属性が設定されます。この設定により、異なる仮想サーバーが異なるパブリックホスト名を持つ場合、クライアントソフトウェアはクライアント要求に HTTP cookie を追加します。CookieInsert パーシステンスタイプの詳細については、「HTTP クッキーに基づく永続性」を参照してください。

コマンドラインインターフェイスを使用して仮想サーバー永続性グループを作成するには

コマンドプロンプトで入力します。

```
bind lb group <vServerGroupName> <vServerName> -persistenceType <
    PersistenceType>
<!--NeedCopy-->
```

例:

```
bind lb group Vserver-Group-1 Vserver-LB-1 -persistenceType
    CookieInsert
<!--NeedCopy-->
```


構成ユーティリティを使用して仮想サーバーグループを変更するには

1. **Traffic Management > Load Balancing > Persistency Groups** に移動して永続性グループを作成し、このグループの一部となる仮想サーバーを指定します。

コマンドラインインターフェイスを使用して仮想サーバーグループを変更するには

コマンドプロンプトで入力します。

```
set lb group <vServerGroupName> -PersistenceBackup <
   BackupPersistenceType> -persistMask <SubnetMaskAddress>
<!--NeedCopy-->
```

例**:**

```
set lb group vserver-Group-1 -PersistenceBackup SourceIP -persistMask
   255.255.255.255
<!--NeedCopy-->
```

仮想サーバー間でパーシステンスセッションを共有する

August 15, 2023

一部の顧客環境（通信および ISP）では、単一のサーバーが制御トラフィックとデータトラフィックの両方を処理します。特定のクライアント IP アドレスでは、制御トラフィックとデータトラフィックの両方を同じバックエンドサーバーに送信する必要があります。このため、クライアント認証トラフィックの処理には 1 つの仮想サーバーが必要で、通常はルールベースの永続性がその上に設定されます。たとえば、radius.req.avp (8) .value.typecast_text_t’ などです。データトラフィックを処理する 2 番目の仮想サーバー。通常、SourceIP パーシステンスが設定されています。

以前は、永続性エントリは仮想サーバーに対してローカルでした。複数の仮想サーバーに永続性を適用する必要がある場合は、仮想サーバーを負荷分散グループに追加し、グループに共通の永続性タイプを適用する必要がありました。負荷分散グループにバインドされているすべての仮想サーバが、グループに設定された永続性を継承しているため、この要件は達成できません。

仮想サーバー間の永続性共有機能を使用すると、グループ設定から継承するのではなく、グループ内の仮想サーバーが独自の永続性パラメータを使用できるように、負荷分散グループの新しい`useVserverPersistency`パラメータを設定できます。各仮想サーバーで個別のルールベースの永続性を構成できます。

必要に応じて、グループ内の仮想サーバーの 1 つをメイン仮想サーバーとして指定することもできます。仮想サーバーがメイン仮想サーバーとして指定されている場合、その仮想サーバーだけが永続性エントリを作成します。永続エントリは、グループ内のすべての仮想サーバーによって使用されます。メイン仮想サーバーがダウンしている場合、NetScaler ADC アプライアンスは永続性エントリを作成しません。

注: 仮想サーバー間でのパーシステンス共有は、ルールベースの永続性メソッドでのみサポートされます。メンバー仮想サーバーで互換性のあるルールベースの永続性パラメータを設定します。

例**:**

v1 と v2 がロードバランシンググループにバインドされ、v1 が RADIUS タイプの仮想サーバ、v2 が HTTP タイプの仮想サーバであると仮定します。「Radius.req.avp (8) .value.typecast_text_t」永続性は v1 上で構成され、「client.ip.src」は v2 上で構成されています。

トラフィックが RADIUS 仮想サーバ v1 を通過すると、評価されたルール文字列に基づいて永続的なエントリが作成されます。その後、トラフィックが HTTP タイプの仮想サーバー v2 に到達すると、v2 はロードバランシンググループの永続性エントリをチェックし、同じ永続セッションを使用してトラフィックを同じバックエンドサーバーに送信します。

永続セッションの共有の設定

負荷分散グループ内の仮想サーバー間で永続性パラメータを共有するには、まず usevServerPersistency パラメーターを有効にし、グループ内の仮想サーバーの 1 つをメインサーバーとして指定する必要があります。

コマンドラインインターフェイスを使用して **useVserverPersistency** パラメーターを有効にするには

コマンドプロンプトで入力します。

```
set lb group <name> -useVserverPersistency ( ENABLED)
<!--NeedCopy-->
```

例**:**

```
set lb group lb_grp1 -useVserverPersistency ENABLED
<!--NeedCopy-->
```

GUI を使用して **VServerPersistency** パラメータを有効にするには

1. [設定] > [トラフィック管理] > [ロードバランシング] > [持続性グループ**]** に移動します。
2. [追加] をクリックして新しいグループを追加するか、既存のグループを選択して [編集] をクリックします。
3. 「仮想サーバーの持続性を使用」を選択します。

コマンドラインインターフェイスを使用して仮想サーバーをメイン仮想サーバーとして指定するには

コマンドプロンプトで入力します。

```
set lb group <name> -useVserverPersistency ( ENABLED ) -masterVserver <
    string>
<!--NeedCopy-->
```


例:

```
set lb group lb_grp1 -useVserverPersistency ENABLED  - masterVserver vs1
<!--NeedCopy-->
```

GUI を使用して仮想サーバをメイン仮想サーバとして指定するには

1. [設定] > [トラフィック管理] > [ロードバランシング] > [持続性グループ] に移動します。
2. [追加] をクリックして新しいグループを追加するか、既存のグループを選択して [編集] をクリックします。
3.「仮想サーバーの持続性を使用」を選択します。
4. [仮想サーバー名] ボックスで、[**+**] をクリックして、仮想サーバーをグループに追加します。使用可能な仮想サーバーを選択するか、仮想サーバーを作成できます。
5. 新しいグループを追加する場合は [作成] をクリックし、既存のグループを変更する場合は [閉じる] をクリックします。
6. usevServerPersistency パラメータを有効にしたグループを選択し、[編集] をクリックして、永続性エントリを作成するメインとして仮想サーバーを設定します。
7. [マスター **vServer**] リストから、メイン仮想サーバーとして指定する必要がある仮想サーバーを選択します。

引数

useVserverPersistency

グループ内の仮想サーバーが、グループ設定から永続性設定を継承する代わりに、独自の永続性パラメーターを使用して永続セッションを作成できるようにします。このパラメータを有効にすると、負荷分散グループに永続性を設定できません。

このパラメータを無効にすると、グループの仮想サーバーはグループ設定から永続性パラメータを継承します。

負荷分散グループでこのパラメーターを切り替えると、NetScaler アプライアンスはグループとメンバー仮想サーバーの対応する永続性エントリをすべてフラッシュします。

設定可能な値: ENABLED, DISABLED

既定: 無効

例**:**

```
set lb group lb_grp1 -useVserverPersistency ENABLED
<!--NeedCopy-->
```

masterVserver

負荷分散グループ内のメイン仮想サーバーとして仮想サーバーを指定します。指定すると、メイン仮想サーバーのみが、グループが使用する永続エントリを作成できます。

メモ: このパラメータは，useVserverPersistency パラメータが有効になっている場合にのみ設定できます。

例**:**

```
set lb group lb_grp1 -masterVserver vs1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用した永続セッション共有の設定例

仮想サーバーが作成されます

```
add lb vs vs1 http 10.1.10.11 80 -persistence rule -rule 'client.ip.
    src'

add lb vs vs2 radius 10.2.2.2 1812 -persistenceType rule -rule '
    Radius.req.avp(8).value.typecast_text_t'
<!--NeedCopy-->
```

グループが作成されます。

```
add lb group lb_grp1 -persistenceType NONE -useVserverPersistency
    ENABLED
<!--NeedCopy-->
```

グループ内の仮想サーバは、メイン仮想サーバとして指定されます。

```
set lb group lb_grp1 -masterVserver vs1
<!--NeedCopy-->
```

仮想サーバはグループにバインドされます。

```
bind lb group lb_grp1 vs1
bind lb group lb_grp1 vs2
<!--NeedCopy-->
```

詳細については、[基本的な負荷分散の設定および永続性グループの構成を参照してください](/ja-jp/citrix-adc/13-1/load-balancing/load-balancing-persistence/persistence-groups.html)。

パーシステンスを使用して **RADIUS** 負荷分散を構成する

August 15, 2023

今日の複雑なネットワーク環境では、大容量の負荷分散構成と堅牢な認証と承認の調整が必要になることがよくあります。アプリケーションユーザーは、コンシューマグレードの DSL またはケーブル接続、WiFi、さらにはダイヤルアップノードなどのモバイルアクセスポイントを介して VPN に接続できます。これらの接続は、通常、接続中に変更される可能性がある動的 IP を使用します。

NetScaler アプライアンスで RADIUS 認証サーバーへの永続的なクライアント接続をサポートするように RADIUS 負荷分散を構成すると、アプライアンスはクライアント IP の代わりにユーザーログオンまたは指定された RADIUS 属性をセッション ID として使用し、そのユーザーセッションに関連するすべての接続とレコードを同じ RADIUS サーバーに転送します。そのため、ユーザーは、クライアント IP または WiFi アクセスポイントが変更されても接続が切断されることなく、モバイルアクセスロケーションから VPN にログオンできます。

永続性を持つ RADIUS ロードバランシングを設定するには、まず VPN の RADIUS 認証を設定する必要があります。詳細および手順については、AAA アプリケーショントラフィックの「認証、認可、監査（AAA)」の章を参照してください。また、設定のベースとしてロードバランシング機能またはコンテンツスイッチング機能を選択し、選択した機能が有効になっていることを確認します。どちらの機能でも設定プロセスはほぼ同じです。

次に、2 つのロードバランシングまたは 2 つのコンテンツスイッチング仮想サーバーを構成します。1 台は RADIUS 認証トラフィックを処理し、もう 1 台は RADIUS アカウンティングトラフィックを処理します。次に、負荷分散仮想サーバーごとに 1 つずつ、合計 2 つのサービスを構成し、各負荷分散仮想サーバーをそのサービスにバインドします。最後に、負荷分散永続性グループを作成し、永続性タイプを RULE に設定します。

負荷分散またはコンテンツスイッチング機能の有効化

負荷分散機能またはコンテンツスイッチング機能を使用するには、まずその機能が有効になっていることを確認する必要があります。以前に構成されていない新しい NetScaler ADC アプライアンスを構成する場合は、これらの機能の両方がすでに有効になっているため、次のセクションに進んでください。NetScaler ADC アプライアンスを以前の構成で構成していて、使用する機能が有効になっていることを確認できない場合は、今すぐ実行する必要があります。

- ロードバランシング機能を有効にする手順については、ロードバランシングの有効化を参照してください。
- コンテンツスイッチング機能を有効にする手順については、「コンテンツスイッチングの有効化」を参照してください。

仮想サーバの構成

ロードバランシングまたはコンテンツスイッチング機能を有効にしたら、次に RADIUS 認証をサポートする 2 つの仮想サーバーを構成する必要があります。

- **RADIUS** 認証仮想サーバ。この仮想サーバとその関連サービスは、RADIUS サーバへの認証トラフィックを処理します。認証トラフィックは、保護されたアプリケーションまたは仮想プライベートネットワーク（VPN）にログオンするユーザーに関連付けられた接続で構成されます。
- **RADIUS** アカウンティング仮想サーバ。この仮想サーバとその関連サービスは、RADIUS サーバへのアカウンティング接続を処理します。アカウンティングトラフィックは、保護されたアプリケーションまたは VPN での認証済みユーザーのアクティビティを追跡する接続で構成されます。

重要:RADIUS パーシスタンス構成で使用するには、負荷分散仮想サーバーのペアまたはコンテンツスイッチング仮想サーバーのペアを作成する必要があります。仮想サーバーの種類を混在させることはできません。

コマンドラインインターフェイスを使用して負荷分散仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、負荷分散仮想サーバーを作成し、構成を確認します。

```
add lb vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule
    <rule>

show lb vserver <name>
<!--NeedCopy-->
```

既存の負荷分散仮想サーバーを構成するには、前述の`add lb virtual server`コマンドを同じ引数を取る`set lb vserver`コマンドに置き換えます。

コマンドラインインターフェイスを使用してコンテンツスイッチ仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、コンテンツスイッチング仮想サーバーを作成し、構成を確認します。

```
add cs vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule
    <rule>

show cs vserver <name>
<!--NeedCopy-->
```

既存のコンテンツスイッチング仮想サーバを設定するには、前述の`add cs vserver`コマンドを同じ引数を取る`set cs vserver`コマンドに置き換えます。

例**:**

```
add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

set lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

set lb vserver radius_auth_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
<!--NeedCopy-->
```

構成ユーティリティを使用して負荷分散またはコンテンツスイッチング仮想サーバーを構成するには

[トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動するか、[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを設定します。

サービスの構成

仮想サーバーを構成したら、次に、作成した仮想サーバーごとに 1 つずつ、合計 2 つのサービスを構成する必要があります。

注：これらのサービスは、いったん構成されると、NetScaler ADC アプライアンスが RADIUS サーバーの認証およびアカウンティング IP に接続してステータスを監視できるまで、DISABLED 状態になります。手順については、「サービスの設定」を参照してください。

サービスへの仮想サーバーのバインド

サービスを構成したら、作成した各仮想サーバーを適切なサービスにバインドする必要があります。手順については、仮想サーバーへのサービスのバインドを参照してください。

Radius の持続性グループの設定

負荷分散仮想サーバーを対応するサービスにバインドしたら、永続性をサポートするように RADIUS 負荷分散構成を設定する必要があります。そのためには、RADIUS 負荷分散仮想サーバーとサービスを含む負荷分散永続性グループを設定し、その負荷分散永続性グループがルールベースの永続性を使用するように構成します。認証とアカウンティングの仮想サーバが異なるため、永続性グループが必要です。また、1 人のユーザの認証とアカウンティングメッセージの両方が同じ RADIUS サーバに到達する必要があるためです。永続性グループを使用すると、両方の仮想サーバーで同じセッションを使用できます。手順については、永続性グループの構成を参照してください。

RADIUS 共有シークレットの設定

リリース 12.0 以降、NetScaler アプライアンスは RADIUS 共有シークレットをサポートしています。RADIUS クライアントとサーバは、クライアントとサーバ上で構成された共有シークレットを使用して相互に通信します。RADIUS クライアントとサーバ間のトランザクションは、共有シークレットを使用して認証されます。このシークレットは、RADIUS パケット内の情報の一部を暗号化するためにも使用されます。

RADIUS 共有秘密鍵検証シナリオ

RADIUS 共有秘密キーの検証は、次のシナリオで行われます。

- **RADIUS** 共有秘密鍵は、**RADIUS** クライアントと **RADIUS** サーバーの両方に設定されます。NetScaler アプライアンスは、クライアント側とサーバー側の両方で RADIUS 秘密鍵を使用します。検証が成功すると、アプライアンスは RADIUS メッセージの送信を許可します。それ以外の場合は、RADIUS メッセージはドロップされます。

- **RADIUS** 共有秘密鍵が **RADIUS** クライアントと **RADIUS** サーバーのどちらにも構成されていません。**RADKY** が設定されていないノードでは共有秘密鍵の検証を実行できないため、NetScaler アプライアンスは RADIUS メッセージをドロップします。
- **RADIUS** 共有秘密鍵は **RADIUS** クライアントと **RADIUS** サーバーの両方に設定されていません。**NetScaler** アプライアンスは **RADIUS** 秘密鍵の検証をバイパスし、RADIUS メッセージを通過させます。

デフォルトの RADIUS 共有シークレットを設定することも、クライアント単位またはサブネット単位で設定することもできます。RADIUS ポリシーが設定されたすべての導入環境に RADIUS 共有秘密鍵を追加することを推奨します。アプライアンスは、RADIUS パケットの送信元 IP アドレスを使用して、使用する共有シークレットを決定します。RADIUS クライアントとサーバ、および対応する共有シークレットを次のように構成できます。

CLI プロンプトで、次のように入力します。

```
add radiusNode <clientPrefix/Subnet> -radKey <Shared_secret_key>
<!--NeedCopy-->
```

引数

IPaddress　CIDR 形式の RADIUS クライアントの IP アドレスまたはサブネット。アプライアンスは、着信要求パケットの送信元 IP アドレスを使用してクライアント IP アドレスと一致させます。クライアント IP アドレスを設定する代わりに、クライアントネットワークアドレスを設定できます。一番長いプレフィックスと照合して、受信したクライアントリクエストの共有シークレットを識別します。

Radkey　クライアント、NetScaler アプライアンス、およびサーバー間の共有シークレット。最大長:31。

```
add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

add service radius_auth_service1 192.168.41.68 RADIUS 1812

add service radius_acct_service1 192.168.41.70 RADIUS 1813

bind lb vserver radius_auth_vs1 radius_auth_service1

bind lb vserver radius_acct_vs1 radius_acct_service[1-3]

add radiusNode 192.168.41.0/24 -radKey  serverkey123

add radiusNode 203.0.113.0/24 -radkey clientkey123
<!--NeedCopy-->
```


共有シークレットは、RADIUS クライアントとサーバの両方に設定する必要があります。コマンドは同じです。サブネットは、共有シークレットがクライアント用かサーバー用かを決定します。

たとえば、指定したサブネットがクライアントサブネットの場合、共有シークレットはクライアント用です。指定されたサブネットがサーバー・サブネット（前の例では 192.168.41.0/24）である場合、共有秘密はサーバー用です。

0.0.0.0/0 のサブネットは、すべてのクライアントとサーバーのデフォルトの共有シークレットであることを意味します。

> 注:
>
> RADIUS 共有シークレットでは、PAP と CHAP の認証方法のみがサポートされます。

パーシステンスセッションを表示する

August 15, 2023

グローバルに、または特定の仮想サーバーに対して有効なさまざまな永続セッションを表示できます。

> 注：NetScaler nCore アプライアンスは、パケット処理に複数の CPU コアを使用します。CPU コアは、アプライアンス上のすべてのセッションを所有しています。アプライアンスがセッションが存在しない要求を受信すると、セッションが作成され、コアの 1 つがそのセッションの所有者として指定されます。
>
> そのセッションに属する後続の要求は、常にオーナーコアに到着して処理されるとは限りません。その場合、コア間メッセージングにより、オーナーコアのセッション情報が常に最新であることが保証されます。

ただし、コアが別のコアが所有する永続性セッションに属する要求を受信すると、コア間メッセージングは永続性セッションのタイムアウト値を更新しません。

したがって、オーナコアからのタイムアウト値のみを表示する show lb persistentSessions コマンドの出力では、永続セッションがアクティブのままであっても、永続性セッションのタイムアウト値が 0（ゼロ）に減少することがあります。

コマンドラインインターフェイスを使用して永続セッションを表示するには

コマンドプロンプトで、すべての仮想サーバーに関連する永続セッションを表示するには、次のように入力します。

```
show lb persistentSessions [<vServer>]
<!--NeedCopy-->
```

コマンドプロンプトで、仮想サーバーに関連する永続性セッションを表示するには、次のように入力します。

```
show lb persistentSessions <vServername>
<!--NeedCopy-->
```


例**:**

```
show lb persistentSessions myVserver
<!--NeedCopy-->
```

GUI を使用してパーシステンスセッションを表示するには

[トラフィック管理] > [仮想サーバ永続セッション] に移動します。

パーシステンスセッションをクリアする

August 15, 2023

セッションがタイムアウトしない場合は、NetScaler アプライアンスから永続セッションを消去する必要がある場合があります。次のいずれかの操作を実行できます。

- すべての仮想サーバーのすべてのセッションを一度にクリアします。
- 特定の仮想サーバーのすべてのセッションを一度にクリアします。
- 特定の仮想サーバーに関連付けられている特定のセッションをクリアします。

コマンドラインインターフェイスを使用して永続セッションをクリアするには

コマンドプロンプトで次のコマンドを入力して永続セッションをクリアし、構成を確認します。

```
clear lb persistentSessions [<vServer> [-persistenceParam <string>]]

show persistentSessions <vServer>
<!--NeedCopy-->
```

例:

例 1 は、ロードバランシング仮想サーバー lbvip1 のすべての永続性セッションをクリアします。

例 2 は、まず負荷分散仮想サーバー lbvip1 の永続性セッションを表示し、persistence パラメーター xls を使用してセッションをクリアし、セッションがクリアされたことを検証する永続セッションを表示します。

例 **1**:

```
> clear persistentSessions lbvip1
 Done
> show persistentSessions
 Done
>
<!--NeedCopy-->
```


例 **2**:

```
> show persistentSessions lbvip1
Type         SRC-IP     ...     PERSISTENCE-PARAMETER
RULE         0.0.0.0    ...     xls
RULE         0.0.0.0    ...     txt
RULE         0.0.0.0    ...     html
 Done
> clear persistentSessions lbvip1 -persistenceParam xls
 Done
> show persistentSessions lbvip1
Type         SRC-IP     ...     PERSISTENCE-PARAMETER
RULE         0.0.0.0    ...     txt
RULE         0.0.0.0    ...     html
 Done
>
<!--NeedCopy-->
```

設定ユーティリティを使用して永続セッションをクリアするには

1. [トラフィック管理] **>** [永続セッションのクリア] に移動します。

過負荷のサービスのパーシステンス設定を上書きする

August 15, 2023

サービスがロードされたり、使用できない場合には、クライアントへのサービスが低下します。この場合、オーバーロードされたサービスに関連付けられた永続性セッションに含まれる要求を他のサービスに一時的に転送するように、NetScaler ADC アプライアンスを構成する必要がある場合があります。つまり、負荷分散仮想サーバー用に構成されている永続性設定を上書きする必要があります。この機能を実現するには、skippersistency パラメータを設定します。この skippersistence パラメータが設定され、仮想サーバーがオーバーロードされたサービスの新しい接続を受信すると、次のことが起こります。

- 仮想サーバーは、サービスがリクエストを受け入れることができる状態に戻るまで、そのサービスに関連付けられている既存の永続性セッションを無視します。
- 他のサービスに関連付けられた持続性セッションは影響を受けません。

この機能は、タイプが ANY または UDP の仮想サーバーでのみ使用できます。

Branch Repeater の負荷分散構成では、負荷モニターを構成し、サービスにバインドする必要もあります。モニタは、サービスの負荷が設定されたしきい値を下回るまで、後続の負荷分散決定からサービスを引き出します。仮想サーバーの負荷モニタの構成の詳細については、「負荷モニタについて」を参照してください。

永続性セッションの一部を形成する要求に対して、次のいずれかのアクションを実行するように仮想サーバーを設定できます。

- 各リクエストを他のサービスのいずれかに送信します。仮想サーバーは負荷分散の決定を行い、負荷分散方式に基づいて各要求を他のサービスに送信します。すべてのサービスが過負荷になると、サービスが使用可能になるまで要求がドロップされます。

  ワイルドカードと IP アドレスベースの仮想サーバーの両方がこのオプションをサポートしています。このアクションは、仮想サーバーが Branch Repeater アプライアンスまたはファイアウォールの負荷分散を行っている展開を含め、すべての展開に適しています。

- 仮想サーバーサービス構成をバイパスします。仮想サーバは、負荷分散の決定を行いません。代わりに、リクエスト内の宛先 IP アドレスに基づいて、各リクエストを物理サーバにブリッジするだけです。

  bypass オプションがサポートされるのは、ANY および UDP タイプのワイルドカード仮想サーバだけです。ワイルドカード仮想サーバーには、IP とポートの組み合わせがあります 。このアクションは、仮想サーバーを使用して Branch Repeater アプライアンスまたはファイアウォールの負荷分散を行う展開に適しています。これらの展開では、NetScaler ADC アプライアンスはまずブランチリピーターアプライアンスまたはファイアウォールに要求を転送し、次に処理された応答を物理サーバーに転送します。仮想サーバーは、以下の条件で、宛先 IP アドレスにリクエストを直接送信します。

  - 仮想サーバ（オーバーロードされたサービスのサービス構成）をバイパスするように仮想サーバを構成します。
  - ブランチリピータアプライアンスまたはファイアウォールが過負荷になります。

  仮想サーバは、Branch Repeater アプライアンスまたはファイアウォールが要求を受け入れることができるまで、宛先の IP アドレスに要求を直接送信します。

CLI を使用してオーバーロードされたサービスの永続設定を上書きするには

コマンドプロンプトで次のコマンドを入力して、オーバーロードされたサービスの永続性設定を上書きし、構成を確認します。

```
set lb vserver <name> -skippersistency <skippersistency>

show lb vserver <name>
<!--NeedCopy-->
```

例

```
> set lb vserver mylbvserver -skippersistency ReLb
 Done
> show lb vserver mylbvserver
    mylbvserver (\*:\*) - ANY Type: ADDRESS
       . . .
       . . .
 Skip Persistency: ReLb
       . . .
```



```
 Done
>
<!--NeedCopy-->
```

GUI を使用してオーバーロードされたサービスの永続設定を上書きするには

1. トラフィック管理 **>** 負荷分散 **>** 仮想サーバーに移動し、UDP または ANY タイプの仮想サーバーを選択します。
2.「詳細設定」ペインで「トラフィック設定」を選択し、「スキップパーシステンシー」のタイプを指定します。

トラブルシューティング

August 15, 2023

- **NetScaler VPX** アプライアンスの統計は、アプライアンスがセッション永続性の制限に達したことを示しています。その結果、パーシスタンスセッションは失敗します。セッション持続性の制限を増やすことはできますか**?**

  原因**:** NetScaler アプライアンスのシステム制限は、コアあたり 250,000 パーシステンスセッションです。

  解決策: この問題を解決するには、次のいずれかのタスクを実行できます。

  - 永続性のタイムアウト値を減らす
  - アプライアンスのコア数を増やす

- **NetScaler** アプライアンスで **Cookie Insert** パーシステンスを構成すると、接続はしばらくの間正常に機能したが、その後切断され始めるという報告がユーザーから寄せられました。パーシスタンスを設定する際には、どのようなベストプラクティスに従うべきですか**?**

  原因: デフォルトでは、Cookie Insert パーシステンスのタイムアウト値は 120 秒です。

  解決策: アイドル時間を特定できないアプリケーションのパーシステンスを設定する場合は、Cookie Insert のパーシステンスタイムアウト値を 0 に設定します。この設定では、接続はタイムアウトしません。

- **NetScaler** アプライアンスで **HTTP** 仮想サーバーを構成した後、要求されたコンテンツに対してユーザーが常に同じサーバーに接続するようにする必要があるため、**SourceIP** パーシステンスを構成しました。現在、パーシスタンスのタイムアウト値を増やすとレイテンシーが発生します。パフォーマンスに影響を与えずにタイムアウト値を増やすにはどうすればよいですか**?**

  解決策: タイムアウト値を 0 に設定して Cookie Insert パーシステンスを使用することを検討してください。この設定では、アプライアンスが Cookie の有効期限を指定しないため、長期間のパーシスタンスの設定が可能になります。

- **NetScaler** アプライアンスで **Cookie Insert** パーシステンスを構成すると、同じタイムゾーンのクライアントがコンテンツにアクセスしても期待どおりに機能します。ただし、別のタイムゾーンのクライアントが接続を試みると、接続はすぐにタイムアウトします。

  原因: 同じタイムゾーンのクライアントが接続を行うと、時間ベースの Cookie Insert パーシスタンスは期待どおりに機能します。ただし、クライアントマシンと NetScaler ADC アプライアンスが異なるタイムゾーンにある場合、Cookie は無効です。たとえば、EST タイムゾーンのクライアントが午前 11:00 にクッキーを PST タイムゾーンの NetScaler ADC アプライアンスに送信すると、アプライアンスは太平洋標準時午後 2 時に Cookie を受信します。時間の差異の結果、クッキーは有効ではなく、接続がすぐにタイムアウトします。

  解決策:Cookie Insert パーシステンスのタイムアウト値を 0 に設定します。

- **NetScaler** アプライアンスは、**Oracle Weblogic** サーバーなどのアプリケーションサーバーの負荷分散に使用されます。クライアントがこれらのサーバに常時接続できるようにするため、**SourceIP** パーシステンスが設定されています。コンピューターから接続すると、期待どおりに機能します。ただし、シンクライアントがターミナルサーバー経由で接続を試みた結果、アプライアンスは同じ **IP** アドレス（ターミナルサーバーの **IP** アドレス）から複数のクライアントからの要求を受信します。したがって、すべてのシンクライアントからの接続は、同じアプリケーションサーバーに転送されます。クライアント **IP** アドレスに基づいて個々のシンクライアントからのリクエストに永続性を設定することはできますか**?**

  原因:NetScaler アプライアンスはターミナルサーバーから要求を受信しますが、要求の送信元 IP アドレスは変わりません。その結果、アプライアンスはシンクライアントから受信したリクエストを区別できず、シンクライアントからのリクエストに応じたパーシステンスを提供できません。

  解決策: この問題を回避するには、シンクライアントごとに固有のパラメータ値に基づいてルール永続性を設定できます。

- **NetScaler** アプライアンスは、**Web Interface** サーバーの負荷分散に使用されます。サーバーにアクセスすると、ユーザーは「**State Error**」エラーメッセージを受け取ります。さらに、**Web Interface** サーバーの **1** つがシャットダウンしたり使用できなくなったりすると、一部のユーザーにエラーメッセージが表示されます。

  原因:Web Interface サーバーへの永続性が欠如していると、ユーザーがサーバーに接続しようとしたときにエラーメッセージが表示されることがあります。

  解決策： Web Interface サーバーの負荷分散を行う場合は、NetScaler アプライアンスで Cookie 挿入パーシステンス方式を指定することを Citrix では推奨しています。

ADC で生成された **Cookie** に属性を挿入する

December 8, 2023

Web 管理者は、NetScaler アプライアンスによって生成された Cookie に他の Cookie 属性を挿入できます。これらの追加の Cookie 属性は、アプリケーションのアクセスパターンに基づいて ADC が生成した Cookie に必要なポリシーを適用するのに役立ちます。

以下の機能は、ADC が生成した Cookie を使用して永続性を実現します。

- ロードバランシング Cookie パーシステンス
- 負荷分散グループの Cookie パーシステンス
- GSLB サイトパーシスタンス
- コンテンツスイッチングクッキーの永続性

次のパラメータを使用して、ADC が生成した Cookie に他の Cookie 属性を挿入できます。

- **literaladcCookieAttribute:** ADC が生成したクッキーに、他のクッキー属性を文字列として追加します。
- **ComputedADCCookieAttribute**: ADC ns 変数を使用して、クライアントまたはサーバーの属性（ユーザーエージェントのバージョンなど）に基づいて、ADC で生成された Cookie に Cookie 属性を条件付きで追加します。

> 注
>
> リテラル ADC Cookie 属性と計算された ADC Cookie 属性の両方を、ロードバランシングパラメータまたは単一のロードバランシングプロファイルで同時に設定することはできません。

使用事例**:SameSite Cookie** 属性の設定

すべての Cookie にはドメインが関連付けられています。クッキーのドメインがユーザーのアドレスバーのウェブサイトドメインと一致する場合、同じサイト（またはファーストパーティ）のコンテキストと見なされます。Cookie に関連付けられたドメインが外部サービスと一致し、ユーザーのアドレスバーにあるウェブサイトと一致しない場合、これはクロスサイト（またはサードパーティ）コンテキストと見なされます。

SameSite 属性は、Cookie をクロスサイトコンテキストに使用できるか、同一サイトコンテキストにのみ使用できるかをブラウザに示します。また、アプリケーションにクロスサイトコンテキストでアクセスする場合は、HTTPS 接続を介してのみアクセスすることができます。詳細については、RFC6265を参照してください。

2020 年 2 月まで、**SameSite** プロパティは **Citrix** ADC で明示的に設定されていませんでした。ブラウザはデフォルト値を「なし」に設定し、NetScaler の展開には影響しませんでした。

ただし、Google Chrome 80 などの特定のブラウザをアップグレードすると、Cookie のデフォルトのクロスドメイン動作が変更されます。**SameSite** 属性は、次のいずれかの値に設定できます。Google Chrome のデフォルト値は Lax に設定されています。

- なし**:** 安全な接続でのみクロスサイトコンテキストで Cookie を使用するようブラウザに指示します。
- **Lax:** ブラウザが同じサイトコンテキストでのリクエストに Cookie を使用するように指示します。クロスサイトコンテキストでは、GET リクエストなどの安全な HTTP メソッドのみが Cookie を使用できます。
- **Strict**: 同じサイトのコンテキストでのみ Cookie を使用します。

Cookie に SameSite 属性がない場合、Google Chrome は SameSite=LAX の機能を引き継ぎます。

> 注
>
> 他のブラウザーの特定のバージョンでは、SameSite 属性のデフォルト値が **None** に設定されることがあります。一部のブラウザバージョンでは、「SameSite = 「なし」は別の方法で扱うことができます。たとえば、以下のブラウザは「SameSite = none」の付いた Cookie を拒否します。
>
> - Chrome51 から Chrome66 までの Chrome のバージョン（両端を含む）
> - Android の UC ブラウザのバージョン 12.13.2 より前のバージョン

ADC で生成された **Cookie** を構成する

ADC が生成する Cookie 属性を設定するには、以下を実行する必要があります。

1. 負荷分散仮想サーバーを作成する
2. LB パラメータまたは LB プロファイルを使用して、負荷分散仮想サーバの ADC Cookie 属性を設定します。
3. LB プロファイルを使用する場合は、LB プロファイルを負荷分散仮想サーバーに設定します。
4. Computed ADC Cookie 属性を使用する場合は、関連する書き換えポリシーを設定してください。

> 注
>
> LB プロファイルが LB 仮想サーバーにバインドされている場合、グローバル LB パラメータ設定の代わりにプロファイルパラメータ設定が考慮されます。

ADC が生成する Cookie 属性は、次の方法で設定できます。

- ロードバランシングパラメータの ADC Cookie 属性の設定
- ロードバランシングプロファイルの ADC Cookie 属性の設定

CLI を使用してロードバランシングパラメータで **ADC Cookie** 属性を設定する

NetScaler アプライアンスで構成されたすべてのアプリケーションの ADC 生成 Cookie にポリシーを均一に適用するには、グローバル LB パラメータで ADC Cookie 属性を設定します。

リテラル **ADCCookie** 属性 設定を使用すると、ADC で生成された Cookie に Cookie 属性を無条件に挿入できます。

コマンドプロンプトで入力します：

```
set lb parameter -LiteralADCCookieAttribute <string>
<!--NeedCopy-->
```

例：


```
set lb parameter -LiteralADCCookieAttribute SameSite=None
<!--NeedCopy-->
```

計算された **ADC Cookie** 属性設定では、クライアントまたはサーバーの属性に基づいて、ADC が生成する Cookie に条件付きで Cookie 属性を挿入できます。

コマンドプロンプトで入力します:

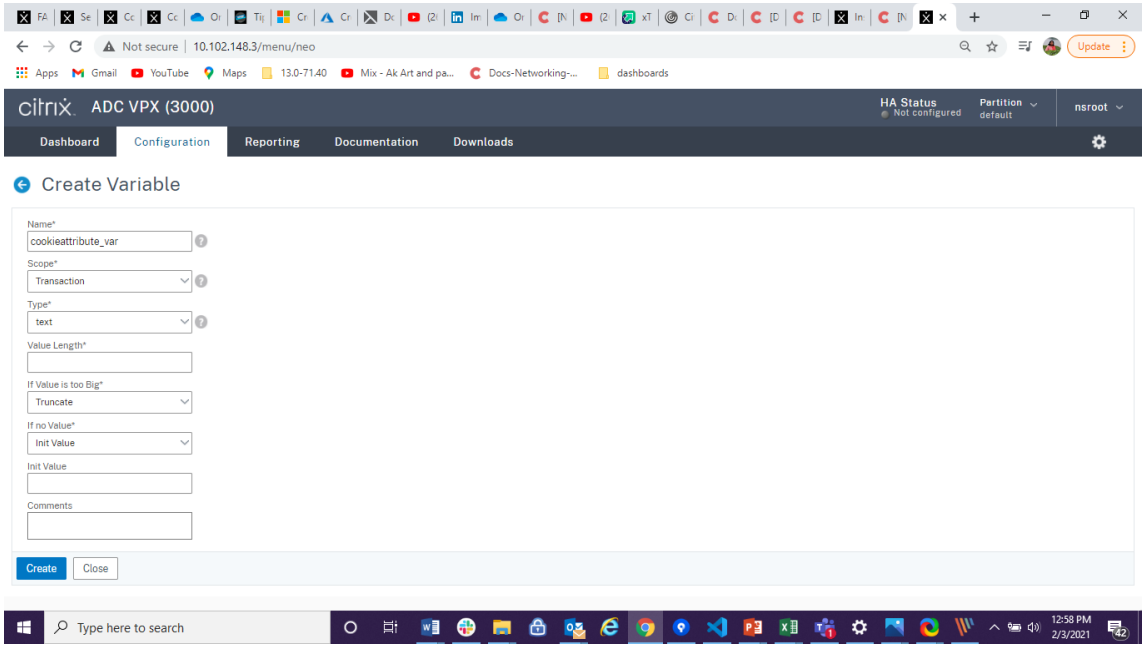
```
set lb parameter -ComputedADCCookieAttribute <ns variable>
<!--NeedCopy-->
```

例:

```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
set lb parameter -ComputedADCCookieAttributE "$cookieattribute_var"
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    ""SameSite=None""

add policy expression pol_iphone "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("iP") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/OS
    \d+\\_/).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT "false").eq("true"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("Chrom") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/
    Chrom.*\d+./).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN
    (51,66).typecast_text_t ALT "false").eq("true"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

bind rewrite global exception_samesite_attribute 90 110 -type
    RES_OVERRIDE
bind rewrite global append_samesite_attribute 100 110 -type
    RES_OVERRIDE
<!--NeedCopy-->
```

GUI を使用して変数を設定する

1. **AppExpert >** 変数に移動し、「追加」をクリックします。
2. 「変数の作成」ページで、ドロップダウンメニューから「スコープ」を「トランザクションとしてスコープ ****** 」を選択し、「テキストとしてタイプ ** 」を選択します。



3. その他の詳細を入力して、[作成] をクリックします。

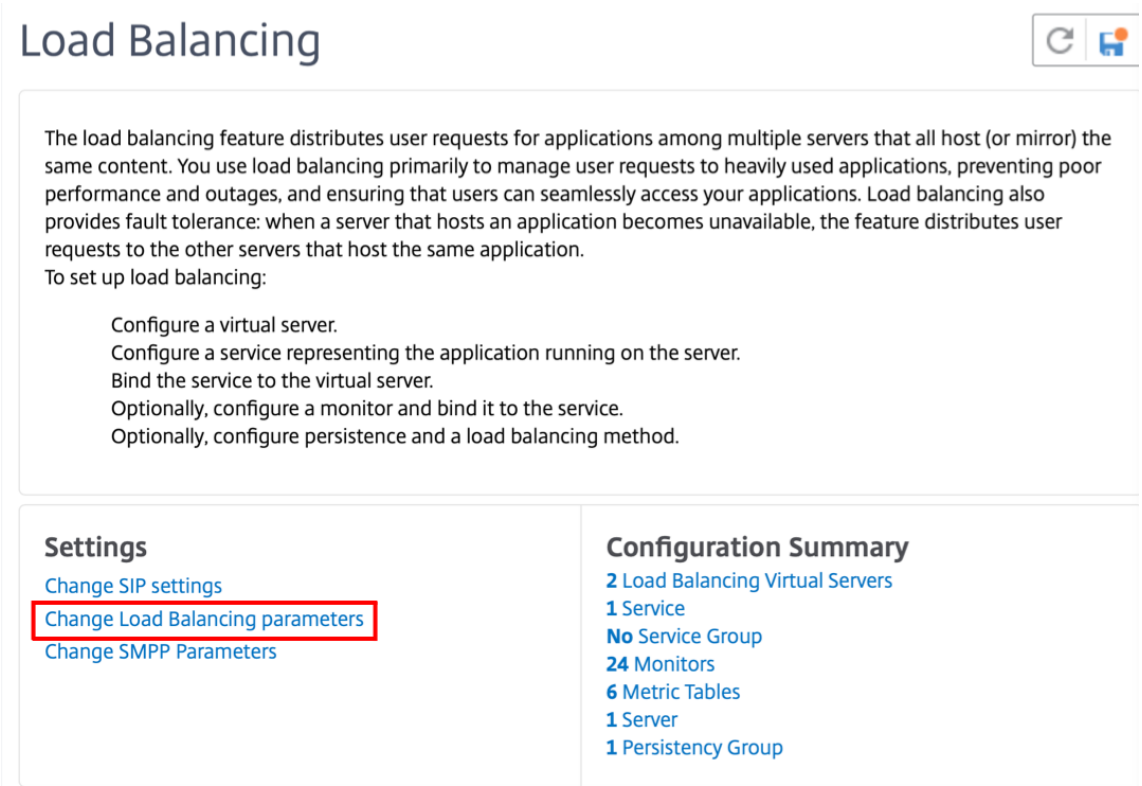
GUI を使用して課題を作成する

変数を設定したら、値を割り当てるか、代入を作成して変数に対して実行する操作を指定できます。

1. **AppExpert > Assignments** に移動して **Add** をクリックします。
2. [割り当ての作成] ページで、詳細を入力し、[作成] をクリックします。

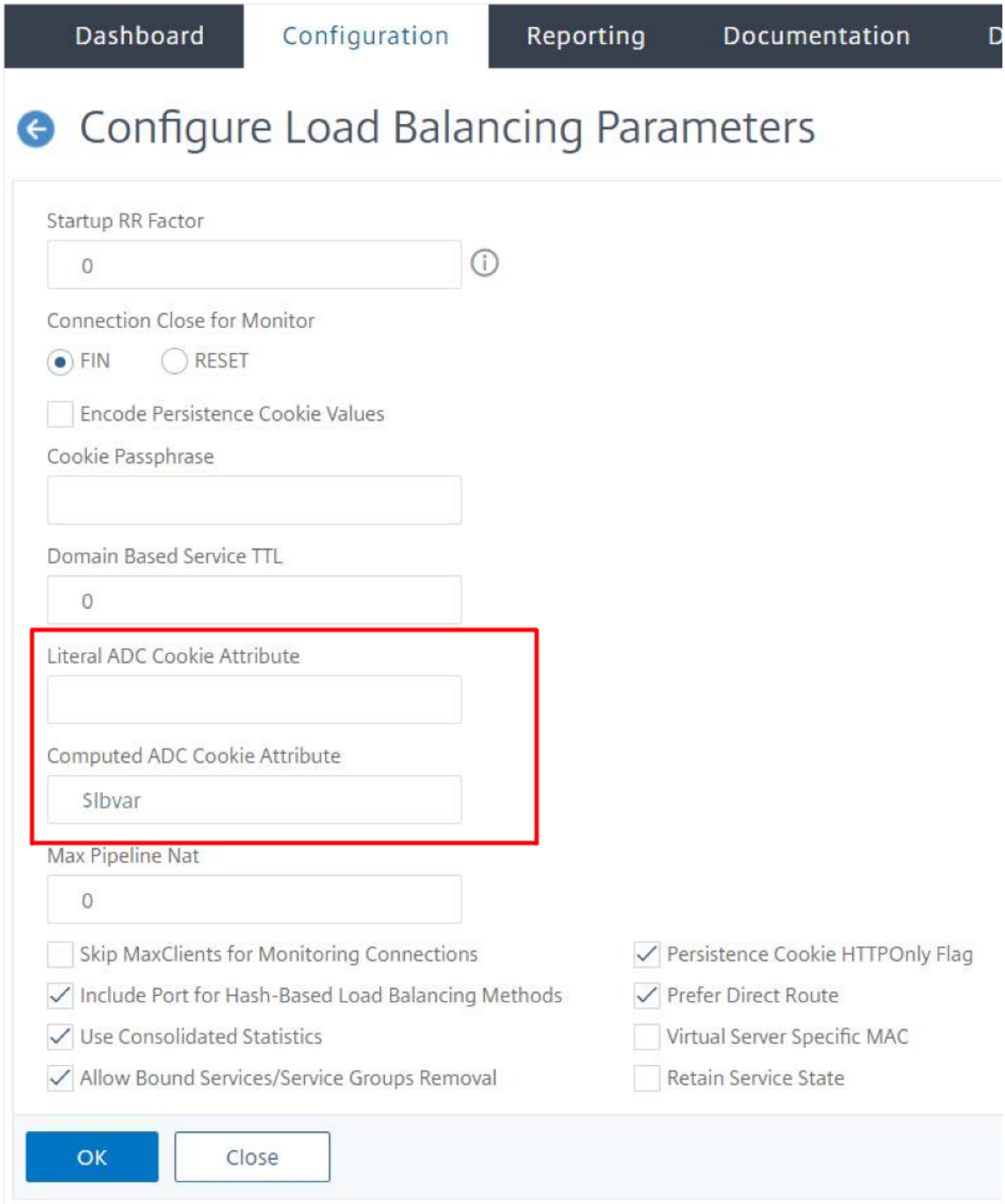
GUI を使用したロードバランシングパラメータでの **ADC Cookie** 属性の設定

1. **[**トラフィック管理**] > [**負荷分散**] > [**負荷分散パラメータの変更**]** に移動します。



2. [負荷分散パラメータの設定] ペインで、要件に基づいていずれかのフィールドに適切な値を入力します。

   - リテラル **ADC** クッキー属性
   - 計算された **ADC** クッキー属性



3. [**OK**] をクリックします。

CLI を使用したロードバランシングプロファイルでの **ADC Cookie** 属性の設定

NetScaler アプライアンスで構成された特定のアプリケーションにポリシーを適用するには、アプリケーション固有の LB 仮想サーバーにバインドされた LB プロファイルの Cookie 属性パラメータを設定できます。

LB プロファイルの リテラル **ADCCookie** 属性 設定を使用すると、仮想サーバーに固有の ADC 生成 Cookie に

Cookie 属性を無条件に挿入できます。

コマンドプロンプトで入力します:

```
add lb profile <profile name> -LiteralADCCookieAttribute <string>
<!--NeedCopy-->
```

例:

```
add lb profile LB-Vserver-Profile-1 -LiteralADCCookieAttribute SameSite
    =None
add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
    COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
<!--NeedCopy-->
```

LB プロファイルの **Computed ADC Cookie** 属性設定を使用すると、クライアントまたはサーバーの属性に基づいて、ADC が生成する Cookie に条件付きで Cookie 属性を挿入できます。次に、この LB プロファイルを LB 仮想サーバーに設定します。

コマンドプロンプトで入力します:

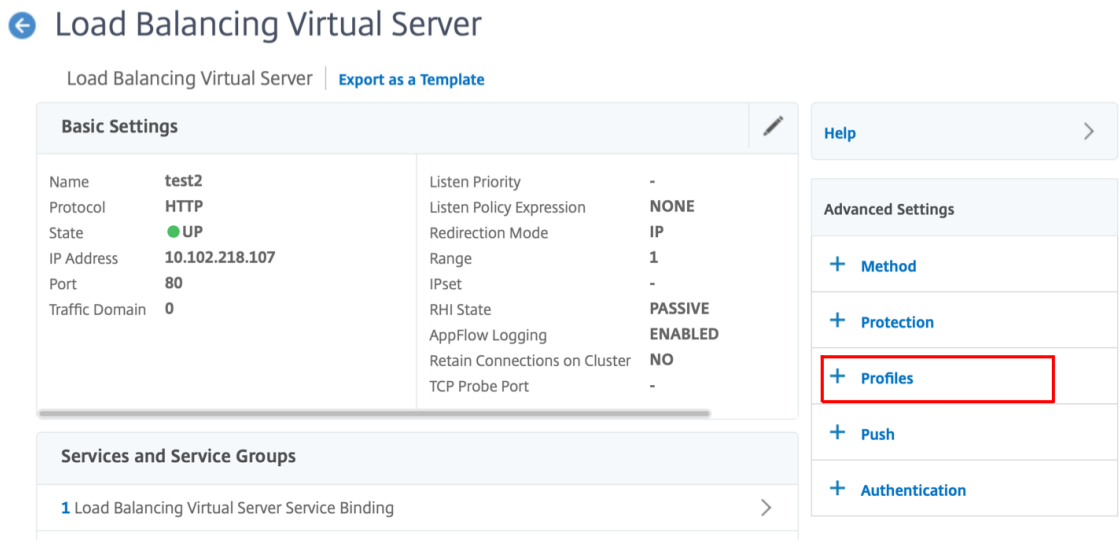
```
add lb profile <profile name> -ComputedADCCookieAttribute <ns variable>
<!--NeedCopy-->
```

例:

```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    ""SameSite=None""
add lb profile LB-Vserver-Profile-1 -ComputedADCCookieAttributE "
    $cookieattribute_var"

add policy expression pol_iphone "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("iP") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/OS
    \d+\\_/).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT "false").eq("true"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("Chrom") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/
    Chrom.*\d+./).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN
    (51,66).typecast_text_t ALT "false").eq("true"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
    COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
bind lb vserver LB-VServer-1 -policyName exception_samesite_attribute -
    priority 90 -gotoPriorityExpression 110 -type RESPONSE
bind lb vserver LB-VServer-1 -policyName append_samesite_attribute -
    priority 100 -gotoPriorityExpression 110 -type RESPONSE
<!--NeedCopy-->
```

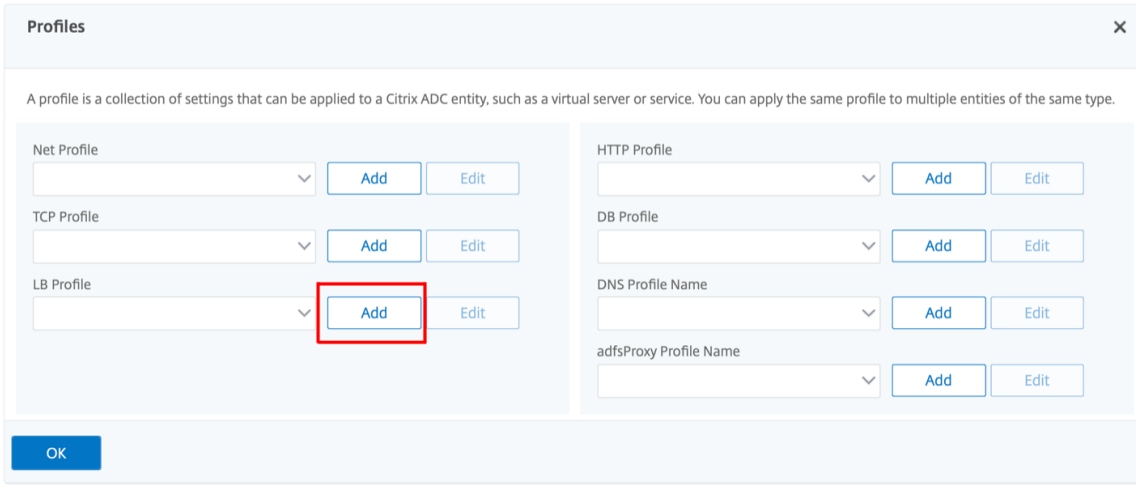

GUI を使用したロードバランシングプロファイルでの **ADC Cookie** 属性の設定

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバを選択し、[編集（Edit）] をクリックします。
3. [詳細設定] セクションの [プロファイルの追加] をクリックします。



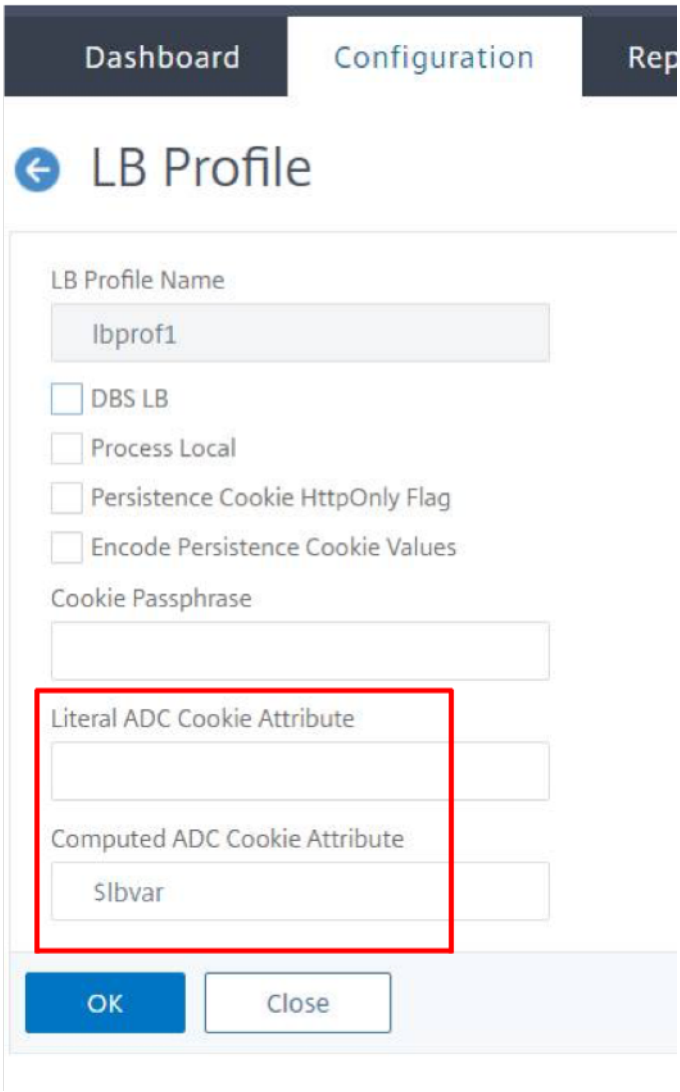
4.「プロファイル」セクションで、「追加」をクリックして LB プロファイルを作成します。

   プロファイルをすでに作成している場合は、[**LB Profile]** ドロップダウンメニューからプロファイルを選択します。



5. [**LB** プロファイル] ペインで、要件に基づいて、いずれかのフィールドに適切な値を入力します。

- リテラル **ADC** クッキー属性
- 計算された **ADC** クッキー属性



1. [**OK**] をクリックします。
2. 作成した LB プロファイルを、手順 **1** で作成した **LB** 仮想サーバーに設定します。

ns 変数設定の検証

ADC ns 変数が LB パラメータまたは LB プロファイルで適切に設定されていることを確認するには、show lb パラメータまたは show lb profile コマンドを使用します。

次の表は、ns 変数が正しく設定されていない場合のさまざまな警告メッセージとその原因を示しています。

警告メッセージ	理由
NS 変数は設定されていません。text () とタイプして変数にスコープトランザクションを設定してください	NS 変数はまだ設定されていません。

警告メッセージ	理由
設定された NS 変数の範囲はトランザクションではありません。 変数のタイプは Text () ではありません。	変数は設定されていますが、スコープが「トランザクション」に設定されていません。 変数は構成されていますが、タイプが「テキスト」に設定されていません。
NS 変数に設定された値の最大サイズが 255 を超えています。	NS 変数に設定された値は 255 文字を超えています。注: ADC で生成された Cookie には、最大 255 文字の長さを追加できます。最大長を超える文字は切り捨てられます。

出力例

次の例では、ns 変数が設定されていない場合に警告メッセージが表示されます。

```
set lb parameter -ComputedADCCookieAttribute "$lbvar"

Warning: NS Variable is not configured. Please configure it with type
    text() and scope transaction
Done
<!--NeedCopy-->
```

警告メッセージは、`show lb parameter` 次のコマンドの出力に表示されます。

```
show lb parameter

Global LB parameters:
Persistence Cookie HttpOnly Flag: ENABLED
Use Encrypted Persistence Cookie: DISABLED
Use Port For Hash LB: YES
Prefer direct route: YES
Retain Service State: OFF
Start RR Factor: 0
Skip Maxclient for Monitoring: DISABLED
Monitor Connection Close: FIN
Use consolidated stats for LeastConnection: YES
Allow mac mode based vserver to pick thereturn traffic from services:
    DISABLED
Allow bound service removal: ENABLED
TTL for Domain Based Server: 0 secs

NetScaler Cookie Variable Name: $lbvar(NS Variable is not configured.
    Please configure it with type text() and scope transaction)

Done
<!--NeedCopy-->
```


GSLB デプロイメントに Cookie 属性を挿入するためのサンプル設定

次の設定例は、LB 仮想サーバーに対応する GSLB サービスで設定されたサイトパーシスタンスに適用されます。GSLB Cookie にいくつかのクッキー属性を追加するには、以下の設定を行います。

- LB プロファイル (LB-VServer-Profile-1) に ADC Cookie 属性を設定します。
- LB プロファイルに「SameSite=None」などのリテラル ADC クッキー属性値を設定します。
- LB プロファイルを GSLB サービスを表す負荷分散仮想サーバー (LB-VServer-1) に設定します。

```
add gslb vserver GSLB-VServer-1 SSL -backupLBMethod ROUNDROBIN -
    tolerance 0 -appflowLog DISABLED
add gslb site site1 10.102.148.4 -publicIP 10.102.148.4
add gslb service site1_gsvc1 10.102.148.35 SSL 443 -publicIP
    10.102.148.35 -publicPort 443 -maxClient 0 -siteName site1 -
    sitePersistence HTTPRedirect -sitePrefix ss1 -cltTimeout 180 -
    svrTimeout 360 -downStateFlush ENABLED

bind gslb vserver GSLB-VServer-1 -serviceName site1_gsvc1
bind gslb vserver GSLB-VServer-1 -domainName www.gslb.com -TTL 5

add service service-1 10.102.84.140 SSL 443

add lb profile LB-Vserver-Profile-1 -LiteralADCCookieAttribute SameSite
    =None
add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
    COOKIEINSERT -lbprofilename LB-Vserver-Profile-1

bind lb vserver LB-VServer-1 service-1
<!--NeedCopy-->
```

> 注
>
> また、計算された ADC Cookie 属性を使用して Cookie 属性を条件付きで挿入することもできます。

コンテンツスイッチングデプロイメントに Cookie 属性を挿入するための設定例

次の設定例は、複数のアプリケーションがコンテンツスイッチング仮想サーバーの背後でホストされている場合に適用されます。すべてのアプリケーションに同じポリシーを適用するには、次のように、書き換えポリシーを LB 仮想サーバーではなくコンテンツスイッチング仮想サーバーにバインドします。

- LB パラメータに ADC Cookie 属性を設定します。

  > 注:
  >
  > ADC Cookie 属性は LB プロファイルでも設定できます。

- タイプを [テキスト] に、[スコープ] を [トランザクション] に設定した ns 変数 (cookieattribute_var) を設定します。

- ns 変数を使用して、計算された ADC Cookie 属性をグローバル LB パラメータに設定します。
- Cookie 属性を挿入するためのリライトポリシー (exception_samesite_attribute と append_samesite_attribute) をコンテンツスイッチング仮想サーバーに設定します。

```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
set lb parameter -ComputedADCCookieAttributE "$cookieattribute_var"
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    ""SameSite=None""

add policy expression pol_iphone "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("iP") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/OS
    \d+\\_/).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT "false").eq("true"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("Chrom") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/
    Chrom.*\d+./).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN
    (51,66).typecast_text_t ALT "false").eq("true"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

add lb vserver LB-VServer-1 SSL 10.102.148.35 443
add lb vserver LB-VServer-2 SSL 10.102.148.36 443

add cs vserver CS-VServer-1 SSL 10.102.148.42 443 -persistenceType
    COOKIEINSERT

add cs action act1 -targetLBVserver v1
add cs action act2 -targetLBVserver v2
add cs policy CS-policy-1 -rule "HTTP.REQ.URL.CONTAINS("file1.html")" -
    action act1
add cs policy CS-policy-2 -rule "HTTP.REQ.URL.CONTAINS("file2.html")" -
    action act2

bind cs vserver CS-VServer-1 -policyName CS-policy-1 -priority 1
bind cs vserver CS-VServer-1 -policyName CS-policy-2 -priority 2

bind cs vserver -policyname exception_samesite_attribute 90 110 -type
    RES_OVERRIDE
bind cs vserver -policyname append_samesite_attribute 100 110 -type
    RES_OVERRIDE
<!--NeedCopy-->
```

負荷分散構成をカスタマイズする

August 15, 2023

基本的なロードバランシング設定を構成した後、必要に応じて負荷を分散するように、いくつかの変更を加えることができます。ロードバランシング機能は複雑です。基本要素を変更するには、次のいずれか 1 つまたは複数を実行します。

- ロードバランシングアルゴリズムの変更
- ロードバランシンググループを構成し、それらを使用してロードバランシング構成を作成する
- 永続的なクライアント/サーバー接続の設定
- リダイレクションモードの設定
- キャパシティが異なる異なるサービスに、異なる重みを割り当てる。

NetScaler ADC アプライアンスのデフォルトの負荷分散アルゴリズムは、最小の接続方法です。最小接続方式では、アプライアンスは現在最も少ない接続を処理しているサービスに、各着信接続を送信します。異なる負荷分散アルゴリズムを指定できます。各アルゴリズムは、異なる条件に適しています。

ショッピングカートなどのアプリケーションに対応するために、同じユーザーからのすべての要求を同じサーバーに送る必要がある場合は、クライアントとサーバー間の永続的な接続を維持するようにアプライアンスを構成できます。仮想サーバーのグループに対して、パーシステンスを指定することもできます。永続性により、グループ内のどの仮想サーバがクライアント要求を受信するかに関係なく、アプライアンスは個々のクライアント要求を同じサービスに送信できます。

ユーザー要求のリダイレクト時にアプライアンスが使用するリダイレクトモードを有効にして構成し、IP ベースと MAC ベースの転送を選択できます。また、各サービスへの着信負荷の割合を指定して、異なるサービスに重みを割り当てることもできます。重みを割り当てると、同じ負荷分散設定に異なる容量のサーバーを含めることができます。

- 低容量サーバの過負荷や
- 大容量のサーバがアイドル状態になるようにします。

仮想サーバー間のパーシステンスのためにハッシュアルゴリズムでカスタマイズする

August 15, 2023

NetScaler アプライアンスは、ハッシュベースのアルゴリズムを使用して仮想サーバー全体の永続性を維持します。デフォルトでは、ハッシュベースの負荷分散方式は、サービスの IP アドレスとポート番号のハッシュ値を使用します。同じサーバー上の異なるポートでサービスを利用できる場合、アルゴリズムは異なるハッシュ値を生成します。そのため、異なる負荷分散仮想サーバーが同じアプリケーションへのリクエストを異なるサービスに送信し、疑似永続性が損なわれる可能性があります。

ポート番号を使用してハッシュ値を生成する代わりに、サービスごとに一意のハッシュ識別子を指定できます。サービスの場合、すべての仮想サーバーで同じハッシュ識別子値を指定する必要があります。物理サーバーが複数のタイプのアプリケーションを提供する場合、各アプリケーションタイプには一意のハッシュ識別子が必要です。

サービスのハッシュ値を計算するアルゴリズムは、次のように機能します。

- デフォルトでは、グローバル設定によりハッシュ計算でのポート番号の使用が指定されます。
- サービスのハッシュ識別子を設定すると、グローバル設定に関係なく、そのハッシュ識別子が使用され、ポート番号は使用されません。
- ハッシュ識別子を設定せずに、ポート番号の使用を指定しないようにグローバル設定のデフォルト値を変更した場合、ハッシュ値はサービスの IP アドレスのみに基づきます。
- ハッシュ識別子を設定しない場合や、ポート番号を使用するようにグローバル設定のデフォルト値を変更した場合、ハッシュ値はサービスの IP アドレスとポート番号に基づきます。

CLI を使用してサービスをサービスグループにバインドするときに、ハッシュ識別子を指定することもできます。構成ユーティリティでは、サービスグループを開き、「メンバー」タブでハッシュ識別子を追加できます。

CLI を使用して use-port-number グローバル設定を変更するには

コマンドプロンプトで入力します。

lb パラメータを設定-HashLB のポートを使用 (はい)	NO)

例:

```
> set lb parameter -usePortForHashLb NO
 Done
>show lb parameter
Global LB parameters:
        Persistence Cookie HttpOnly Flag: DISABLED
        Use port for hash LB: NO
 Done
<!--NeedCopy-->
```

GUI を使用して use-port-number のグローバル設定を変更するには

1. [トラフィック管理] > [負荷分散] > [負荷分散パラメータの設定] に移動します。
2.「ハッシュベースの LB メソッドにポートを使用」を選択または選択解除します。

CLI を使用して新しいサービスを作成し、サービスのハッシュ識別子を指定するには

コマンドプロンプトで次のコマンドを入力してハッシュ ID を設定し、設定を確認します。

add service < name > (< ip >	< serverName >) < serviceType > < port > -hashId < positive_integer >


```
show service <name>
<!--NeedCopy-->
```

例:

```
> add service flbkng 10.101.10.1 http 80 -hashId 12345
 Done
>show service flbkng
        flbkng (10.101.10.1:80) - HTTP
        State: DOWN
        Last state change was at Thu Nov  4 10:14:52 2010
        Time since last state change: 0 days, 00:00:15.990
        Server Name: 10.101.10.1
        Server ID : 0   Monitor Threshold : 0

        Down state flush: ENABLED
        Hash Id: 12345

1)      Monitor Name: tcp-default
                State: DOWN     Weight: 1

 Done
<!--NeedCopy-->
```

CLI を使用して既存のサービスのハッシュ識別子を指定するには

set service コマンド、サービスの名前、**-HashiD** の後に **ID** 値を入力します。

サービスグループメンバーを追加するときにハッシュ識別子を指定するには

グループに追加する各メンバーのハッシュ識別子を指定して設定を確認するには、コマンドプロンプトで次のコマンドを入力します (メンバーごとに一意の HashID を指定してください)。):

```
bind servicegroup <serviceGroupName> <memberName> <port> -hashId <
   positive_integer>

show servicegroup <serviceGroupName>
<!--NeedCopy-->
```

例:

```
 bind servicegroup http_svc_group 10.102.27.153 80 -hashId 2222222

>show servicegroup SRV
        SRV - HTTP
        State: ENABLED  Monitor Threshold : 0
        ...

```



```
        1)          1.1.1.1:80   State: DOWN     Server Name: 1.1.1.1
              Server ID: 123  Weight: 1
        Hash Id: 32211

                Monitor Name: tcp-default       State: DOWN
        ...

        2)          2.2.2.2:80   State: DOWN     Server Name: 2.2.2.2
              Server ID: 123  Weight: 1
        Hash Id: 12345

                Monitor Name: tcp-default       State: DOWN
        ...
Done

<!--NeedCopy-->
```

GUI を使用してサービスのハッシュ識別子を指定するには

1. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
2. 新しいサービスを作成するか、既存のサービスを開いてハッシュ ID を指定します。

GUI を使用して設定済みのサービスグループメンバーのハッシュ識別子を指定するには

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. メンバーを開き、固有のハッシュ ID を入力します。

リダイレクトモードを構成する

August 15, 2023

リダイレクションモードは、着信トラフィックを転送する場所を決定するために仮想サーバが使用する方式を構成します。NetScaler ADC アプライアンスは、次のリダイレクトモードをサポートしています。要求をサーバーに転送する前に、リダイレクトモードは次のように機能します。

- IP ベースの転送 (デフォルト): 宛先 IP アドレスがサーバーの IP アドレスに変更されます。
- MAC ベースの転送: 宛先 MAC アドレスがサーバの MAC アドレスに変更されます。ただし、宛先 IP アドレスは変更されません。MAC ベースのリダイレクションモードは、主にファイアウォール負荷分散展開で使用されます。
- IP トンネルベース: IP-in-IP カプセル化は、クライアント IP パケットに対して実行されます。外部 IP ヘッダーでは、宛先 IP アドレスはサーバの IP アドレスに設定され、送信元 IP アドレスはサブネット IP（SNIP）に

設定されます。クライアント IP パケットは変更されません。これは、IPv4 パケットと IPv6 パケットの両方に適用できます。

- TOS ID ベース：仮想サーバの TOS ID は、IP ヘッダーの TOS フィールドにエンコードされます。

IP TUNNEL または TOS オプションのいずれかを使用して、ダイレクトサーバリターン（DSR）を実装できます。詳細については、以下を参照してください。

- TOS を使用する場合の DSR モードの設定
- TOS フィールドを使用して、IPv6 ネットワークの DSR モードで負荷分散を構成します。
- IP Over IP を使用して DSR モードで負荷分散を構成する

DSR トポロジ、リンク負荷分散、またはファイアウォール負荷分散を使用するネットワーク上で、MAC ベースの転送を構成できます。負荷分散のための MAC ベースの転送の詳細については、負荷分散構成用の MBF の構成を参照してください。

CLI を使用してリダイレクションモードを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -m <RedirectionMode>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

> 注
>
> -m MACオプションが有効になっている仮想サーバーにバインドされているサービスの場合は、非ユーザーモニターをバインドする必要があります。

GUI を使用してリダイレクションモードを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、リダイレクトモードを選択します。

VLAN 単位のワイルドカード仮想サーバーを構成する

August 15, 2023

特定の仮想ローカルエリアネットワーク (VLAN) 上のトラフィックの負荷分散を構成する場合は、指定した VLAN でのみトラフィックを処理するように制限するリッスンポリシーを持つワイルドカード仮想サーバーを作成できます。

CLI を使用して特定の VLAN をリッスンするワイルドカード付きの仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、特定の VLAN を受信するワイルドカード付きの仮想サーバーを構成し、構成を確認します。

```
add lb vserver <name> <serviceType> IPAddress * Port * -listenpolicy <
    expression> [-listenpriority <positive_integer>]

show vserver
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-vlan1 ANY -listenpolicy "CLIENT.VLAN.ID.EQ(2)
    " -listenpriority 10

show vserver Vserver-LB-vlan1
<!--NeedCopy-->
```

GUI を使用して特定の VLAN をリッスンするワイルドカード付きの仮想サーバーを構成するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 新しい仮想サーバーを作成するか、既存の仮想サーバーを開きます。
3. リッスンポリシーの優先順位と式を指定します。

この仮想サーバーを作成したら、「基本負荷分散の設定」の説明に従って、仮想サーバーを 1 つ以上のサービスにバインドします。

サービスに重要度を割り当てる

February 15, 2024

負荷分散構成では、各サービスに送信すべきトラフィックの割合を示す重みをサービスに割り当てます。重みが高いサービスはより多くの要求を処理でき、重みが小さいサービスは処理する要求の数が少なくなります。サービスに重みを割り当てると、NetScaler アプライアンスが各負荷分散サーバーが処理できるトラフィック量を決定できるため、より効果的に負荷を分散できます。

注: サービスの重み付けをサポートする負荷分散方式 (ラウンドロビン方式など) を使用する場合は、サービスに重みを割り当てることができます。

次の表では、重み付けをサポートする負荷分散方法と、各サービスの選択方法に重み付けがどのように影響するかを簡単に説明しています。

負荷分散方法	ウェイト付きサービスセレクション
ラウンドロビン	仮想サーバは、使用可能なサービスのキューに優先順位を付けます。重みが最も高いサービスが、重みが最も小さいサービスよりも頻繁にキューの前面に配置され、比例してトラフィックを受信します。完全な説明については、ラウンドロビン方式を参照してください。
最小接続	仮想サーバは、アクティブなトランザクションが最も少なくて重みが最も高い組み合わせでサービスを選択します。完全な説明については、「最小接続方式」を参照してください。
モニタを用いた最小応答時間・最小応答時間手法	仮想サーバは、最も少ないアクティブなトランザクションと最速の平均応答時間の最適な組み合わせでサービスを選択します。詳細な説明については、「最小応答時間の方法」を参照してください。
最小帯域幅	仮想サーバは、最小トラフィックと最大帯域幅の最適な組み合わせでサービスを選択します。詳細な説明については、「最小帯域幅方式」を参照してください。
最小パケット	仮想サーバは、最も少ないパケットと最も高い重みの最適な組み合わせでサービスを選択します。詳細な説明については、「最小パケット方式」を参照してください。
カスタムロード	仮想サーバは、負荷の最小化と重みの最適な組み合わせでサービスを選択します。詳細な説明については、「カスタムロードメソッド」を参照してください。
ハッシュ法、静的近接法、トークン法	これらの負荷分散方法では重み付けはサポートされていません。
最小リクエスト	仮想サーバーは、アクティブなリクエストが最も少なく、重みが大きいサービスを選択します。詳細な説明については、「最小リクエストメソッド」を参照してください。

CLI を使用してサービスに重みを割り当てるように仮想サーバーを設定するには

コマンドプロンプトで入力します:

```
set lb vserver <name> -weight <Value> <ServiceName>
<!--NeedCopy-->
```

例:


```
set lb vserver Vserver-LB-1 -weight 10 Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスに重みを割り当てるように仮想サーバーを構成するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバーを開き、「サービス」セクションをクリックします。
3. サービスの重量列で、サービスに重みを割り当てます。

MySQL および Microsoft SQL Server のバージョン設定を構成する

August 15, 2023

タイプが MSSQL と MySQL の負荷分散仮想サーバーには、Microsoft® SQL Server® と MySQL サーバーのバージョンをそれぞれ指定できます。一部のクライアントで MySQL または Microsoft SQL Server 製品と同じバージョンが実行されていないことが予想される場合は、バージョン設定をお勧めします。バージョン設定では、すべての通信がサーバーのバージョンに準拠していることを確認することで、クライアント側接続とサーバー側の接続間の互換性を提供します。

CLI を使用して Microsoft SQL サーバーのバージョンパラメーターを設定するには

コマンドプロンプトで次のコマンドを入力して、負荷分散仮想サーバーの Microsoft SQL Server バージョンパラメーターを設定し、構成を確認します。

```
set lb vserver <name> -mssqlServerVersion <mssqlServerVersion>

show lb vserver <name>
<!--NeedCopy-->
```

例

```
> set lb vserver myMSSQLvip -mssqlServerVersion 2008R2
 Done
> show lb vserver myMSSQLvip
    myMSSQLvip (190.0.2.12:1433) - MSSQL    Type: ADDRESS
 . . .
 . . .
    MSsql Server Version: 2008R2
    . . .
    . . .
```



```
 Done
>
<!--NeedCopy-->
```

CLI を使用して MySQL サーバーのバージョンパラメータを設定するには

コマンドプロンプトで次のコマンドを入力して、負荷分散仮想サーバーの MySQL Server バージョンパラメーターを設定し、構成を確認します。

```
set lb vserver <name> -mysqlServerVersion <string>

show lb vserver <name>
<!--NeedCopy-->
```

例

```
> set lb vserver mysqlsvr -mysqlserverversion 5.5.30
 Done
> sh lb vserver mysqlsvr
      mysqlsvr (2.22.2.222:3306) - MYSQL      Type: ADDRESS
 . . .
 . . .
      Mysql Server Version: 5.5.30
     . . .
     . . .
 Done
>
<!--NeedCopy-->
```

GUI を使用して MySQL または Microsoft SQL サーバーのバージョンパラメーターを設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. MySQL または MSSQL タイプの仮想サーバーを開き、サーバーのバージョンを設定します。

マルチ IP 仮想サーバー

August 15, 2023

NetScaler ADC は、VIP タイプの複数の非連続/連続 IPv4 および IPv6 アドレスを持つ単一の負荷分散仮想サーバーの作成をサポートしています。仮想サーバーにバインドされた各 VIP アドレスは、個別の仮想サーバーとして扱わ

れます。これらの仮想サーバーには、同じプロトコルとその他の仮想サーバーレベルの設定があります。複数の VIP アドレスを持つ仮想サーバーは、マルチ IP 仮想サーバーとも呼ばれます。

マルチ IP 仮想サーバーを使用する利点は次のとおりです。

- マルチ IP 仮想サーバーは、同じ設定とサービスバインディングを持つ多数の仮想サーバーを作成する作業を軽減します。
- マルチ IP 仮想サーバーは、仮想サーバーエンティティの上限に達する可能性を効果的に減らします。
- 1 つのマルチ IP 仮想サーバーを異なるサブネットのクライアントに使用して、同じサーバーセットに接続できます。
- IPv6 クライアントと IPv4 クライアントが同じサーバーセットに接続するために使用できるマルチ IP 仮想サーバーは 1 つだけです。

マルチ **IP** 仮想サーバーを構成する

マルチ IP 仮想サーバーの構成は、次のタスクで構成されます。

- IPSet を作成し、複数の IP アドレスをバインドします。
- IPSet を負荷分散仮想サーバーにバインドします。

IPSet の設定に関連する次の点に注意してください。

- IPSet には次のものが含まれます。
  - 非連続/連続 IPv4 アドレスと IPv6 アドレス
  - IPv4 アドレスと IPv6 アドレスの組み合わせ
- IPset を使用して仮想サーバーに関連付けるすべての IPv4/IPv6 アドレスは、VIP タイプである必要があります。
- 1 つの IP セットを複数の仮想サーバーにバインドできます。
- IPv4/IPv6 アドレスは、仮想サーバーへの既存の IPset バインディングに関係なく、IPset にバインド/バインド解除できます。
- 新しい IPSet をバインドする前に、仮想サーバーへの IPSet バインディングの設定を解除する必要があります。

CLI を使用して **IP** セットを追加し、複数の **VIP** アドレスをバインドするには

コマンドプロンプトで入力します。

```
add ipset <name>

bind ipset <name> <IPaddress1 … >

bind ipset <name> <IPaddress2… >

```



```
show ipset <name>
<!--NeedCopy-->
```

CLI を使用して **IPset** を仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
set lb vserver <name> -ipset <ipset name>

show lb vserver <name>
<!--NeedCopy-->
```

GUI を使用して **IPset** を追加し、複数の **VIP** アドレスをバインドするには

[システム] > [ネットワーク] > [**IP** セット**]** に移動し、複数の VIP アドレスを持つ IP セットを作成します。

GUI を使用して **IPset** を仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、作成した IPset をバインドする仮想サーバーを開きます。
2. 「基本設定」で、「**IPset**」パラメータを作成した IPset の名前に設定します。

```
> add ipset IPSET-1


Done

> bind ipset IPSET-1 9.9.9.10


Done

> bind ipset IPSET-1 1000::20


Done

> add lb vserver LBVS-1 HTTP 8.8.8.10 80  - ipset IPSET-1


Done

> add service SVC-1 3.3.3.10 HTTP 80


Done

```



```
> add service SVC-2 3.3.3.100 HTTP 80


Done
> bind lb vserver LBVS-1 SVC-1


Done
> bind lb vserver LBVS-1 SVC-2


Done
```

マルチ **IP** 仮想サーバーの **GSLB** サポート

フローティング IP アドレスは、高可用性展開に必要です。クラウド展開は Floating IP アドレスをサポートしていません。そのため、IP セット機能は、クラウド展開での高可用性をサポートするのに役立ちます。IP セット機能を使用すると、プライベート IP アドレスをプライマリインスタンスとセカンダリインスタンスのそれぞれに関連付けることができます。仮想サーバーの作成時に、プライベート IP アドレスの 1 つが追加されます。もう 1 つの IP アドレスは IP セットにバインドされます。次に、IP セットが仮想サーバーに関連付けられます。通常、パブリック IP アドレスは、どのアプライアンスがトラフィックを受信しているかに基づいて、プライベート IP アドレスの 1 つにマッピングされます。フェールオーバー中、このマッピングは動的に変化し、トラフィックを新しいプライマリにルーティングします。

GSLB 展開では、GSLB サービスは仮想サーバーを表し、仮想サーバーのプライベート IP アドレスとパブリック IP アドレスの両方が必要です。クラウド展開では、IP セットとして表される複数のプライベート IP アドレスがありますが、GSLB サービスは 1 つのプライベート IP アドレスのみを受け入れることができます。そのため、GSLB サービスを構成する際には、仮想サーバーを追加するときに構成された IP アドレス、または IP セット内の IP アドレスの 1 つを指定することをお勧めします。GSLB サービスで IP セット機能を設定する必要はありません。GSLB サービスに関連付けられている負荷分散仮想サーバーで構成されている IP セットで十分です。

GSLB 親子トポロジでは、子サイトの負荷分散仮想サーバーに IP セットを関連付けることができます。このトポロジに対応する GSLB サービスは、パブリック IP アドレスと 1 つのプライベート IP アドレスを保持します。プライベート IP アドレスは、IP セットの IP アドレス、または子サイトに仮想サーバーを追加するときに構成された IP アドレスです。親サイトと子サイト間の通信は、常にパブリック IP アドレスと GSLB サービスのパブリックポートを使用します。

また、IP セットのサポートにより、IPv4 トラフィックと IPv6 トラフィックの両方に対して単一の仮想サーバーエンドポイントを持つことができます。以前は、IPv4 トラフィックと IPv6 トラフィック用に異なる仮想サーバーを構成する必要がありました。IP セットのサポートにより、IPv4 と IPv6 の IP アドレスを同じ IP セットに関連付けることができます。IPv4 エンドポイントと IPv6 エンドポイントを表すさまざまな GSLB サービスを追加できます。

クライアント接続での同時要求の数を制限する

August 15, 2023

1 つのクライアント接続で同時に実行されるリクエストの数を制限できます。同時リクエストの数を制限することで、サーバーをセキュリティの脆弱性から保護できます。クライアント接続が指定された最大制限に達すると、NetScaler アプライアンスは未処理のリクエスト数が制限を下回るまで、その接続での後続のリクエストをドロップします。

MaxPipelineNat パラメータを設定して、1 つのクライアント接続で同時に実行されるリクエストの数を制限できます。このパラメータは、次のサービスタイプと「SvrTimeout」がゼロに設定されている場合にのみ適用されます。

- ANY
- DNS を除くすべての UDP サービスタイプ

MaxPipelineNat パラメーターのデフォルト値は 255 です。値がゼロ (0) の場合、同時リクエストの数に制限はありません。制限が設定されていない場合、NetScaler アプライアンスはすべての要求を実行します。

> 注
>
> MaxPipelineNat を高い値に設定すると、スプーフィング攻撃を受ける可能性が高くなります。そのため、MaxPipelineNat を低い値に設定することをお勧めします。

CLI を使用してクライアントの同時接続数を制限するには

コマンドプロンプトで入力します。

```
set lb parameter -maxPipelineNat <positive_integer>
<!--NeedCopy-->
```

例:

```
set lb parameter -maxPipelineNat 199
<!--NeedCopy-->
```

GUI を使用してクライアントの同時接続数を制限するには

[トラフィック管理] **>** [負荷分散] **>** [負荷分散パラメータの設定] に移動し、Max Pipeline NAT リクエストの値を指定します。

Diameter の負荷分散を構成する

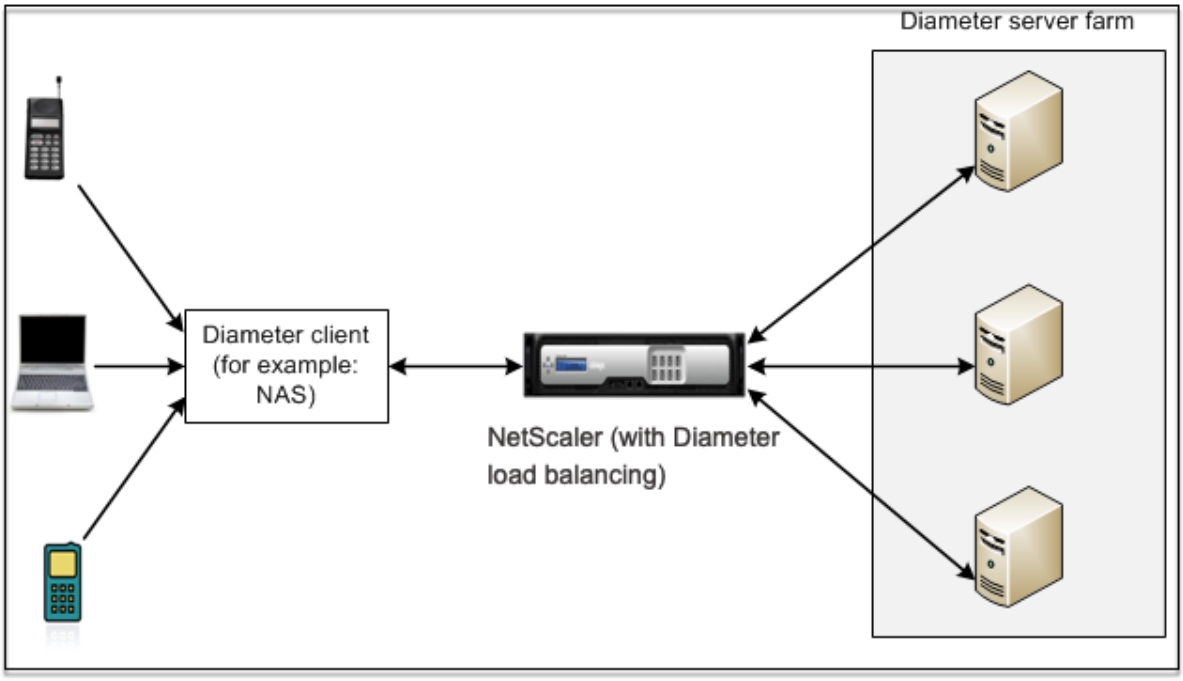
October 25, 2023

Diameter プロトコルは、主にラップトップや携帯電話などのモバイルデバイスで使用される次世代の認証、認可、アカウンティング（AAA）シグナリングプロトコルです。他のほとんどのプロトコルで使用されている従来のクライアント/サーバーモデルとは対照的に、これはピアツーピアプロトコルです。ただし、ほとんどの Diameter ter デプロイメントでは、クライアントがリクエストを送信し、サーバーがリクエストに応答します。

Diameter メッセージが交換されるとき、通常、Diameter サーバーは Diameter クライアントよりもはるかに多くの処理を行います。コントロールプレーンの信号量が増えると、Diameter サーバがボトルネックになります。そのため、Diameter メッセージは複数のサーバに負荷分散する必要があります。Diameter メッセージの負荷分散を実行する仮想サーバーには、次のような利点があります。

- Diameter サーバーの負荷が軽いため、エンドユーザーへの応答時間が短縮されます。
- サーバーのヘルスモニタリングとフェイルオーバー機能の向上。
- クライアント構成を変更せずにサーバーを追加できるため、スケーラビリティが向上します。
- 高可用性。
- SSLDiameter のオフロード。

次の図は、NetScaler 環境における Diameter ameter システムを示しています。



Diameter システムには次のコンポーネントがあります。

- **Diameter** のクライアント。基本プロトコルに加えて Diameter クライアントアプリケーションをサポートします。Diameter クライアントは、多くの場合、ネットワークのエッジにあるデバイスに実装され、そのネ

ットワークにアクセスコントロールサービスを提供します。Diameter クライアントの典型的な例は、ネットワークアクセスサーバ（NAS）とモバイル IP 外部エージェント（FA）です。

- **Diameter** 剤。リレー、プロキシ、リダイレクト、または翻訳サービスを提供します。NetScaler アプライアンス（Diameter ameter 負荷分散仮想サーバーで構成）は、Diameter ameter エージェントの役割を果たします。
- **Diameter** のサーバー。特定のレルムの認証、承認、およびアカウンティング要求を処理します。Diameter サーバーは、基本プロトコルに加えて Diameter サーバーアプリケーションをサポートする必要があります。

一般的な Diameter トポロジでは、エンドユーザーデバイス (携帯電話など) がサービスを必要とするときに、Diameter クライアントにリクエストを送信します。各 Diameter ameter クライアントは、Diameter ベースプロトコル RFC 6733 で規定されているように、Diameter ameter サーバーと 1 つの接続（TCP 接続、SCTP はまだサポートされていません）を確立します。接続は長期間有効で、2 つの Diameter ノード (クライアントとサーバー) 間のメッセージはすべてこの接続を介して交換されます。NetScaler はメッセージベースの負荷分散を使用します。

例**:**

あるモバイルサービスプロバイダーは、請求システムに Diameter を使用しています。加入者がプリペイド番号を使用すると、Diameter クライアントは利用可能な残高を確認するリクエストをサーバーに繰り返し送信します。Diameter プロトコルはクライアントとサーバー間の接続を確立し、すべての要求はその接続を介して交換されます。接続は 1 つしかないため、接続ベースの負荷分散は無意味です。ただし、接続上のメッセージ数が多い場合は、メッセージベースのロードバランシングにより、プリペイドモバイル加入者への請求処理が迅速になります。

直径負荷分散の仕組み

Diameter ter クライアントは NetScaler アプライアンスへの接続を開き、Diameter ameter 機能交換リクエスト (CER) メッセージを送信します。NetScaler は直径サーバーを選択し、サーバーへの接続を開いて、CER メッセージをサーバーに転送します。サーバーはクライアント ID を読み取り、クライアントに直接接続されていると判断します。

Diameter ter サーバーは Diameter ameter ハンドシェイクリプライを準備し、NetScaler アプライアンスに送信します。アプライアンスはハンドシェイクを変更し、独自の ID を挿入します。この時点で、Diameter クライアントは NetScaler（エージェント）に直接接続されていると判断します。

> 注**:**
>
> Diameter ハンドシェイクが完了するまで、クライアントからのすべての Diameter 要求メッセージは、選択したサーバーのキューに格納されます。ハンドシェイクが完了すると、パケットはサーバーに転送されます。

Diameter トラフィックの負荷分散

クライアントが NetScaler ADC アプライアンスに要求を送信すると、アプライアンスは要求を解析し、永続 AVP に基づいてコンテキスト的に Diameter サーバーに負荷分散します。アプライアンスはクライアント ID をサーバにア

ドバタイズしているため、サーバはクライアントからのメッセージを直接受信するため、ルートエントリを追加しません。

サーバーから開始されるリクエストは、クライアントのリクエストほど頻繁ではありません。サーバーから開始される要求は、次の点を除いてクライアントが開始する要求と似ています。

- メッセージは複数のサーバーから受信されるため、アプライアンスは転送された各リクエストメッセージに固有のホップバイホップ（HByH）番号を追加することでトランザクション状態を維持します。メッセージ応答が（同じ HByH 番号で）到着すると、アプライアンスはこの HByH 番号を、要求が到着したときにサーバーで受信した HByH 番号に変換します。
- NetScaler アプライアンスは、クライアントはアプライアンスをリレーエージェントと見なすため、ID を入力してルートエントリを追加します。

注:Diameter メッセージが複数のパケットにまたがる場合、アプライアンスはパケットを不完全なヘッダーキューに蓄積し、メッセージ全体が蓄積されるとサーバーに転送します。同様に、1 つのパケットに複数の Diameter ameter メッセージが含まれている場合、アプライアンスはパケットを分割し、負荷分散仮想サーバーによって決定されたサーバーにメッセージを転送します。

セッションを切断する

Disconnect Peer Request (DPR) は、ピアが接続を閉じる意図と、接続を閉じる理由を示します。ピアは DPA で応答します（TCP は常に DPA が成功します）。

- NetScaler アプライアンスはクライアントから DPR を受信すると、すべてのサーバーに DPR をブロードキャストし、すぐに DPA でクライアントに応答します。サーバーは DPA で応答しますが、アプライアンスはそれを無視します。クライアントは FIN を送信し、アプライアンスはそれをすべてのサーバにブロードキャストします。
- アプライアンスはサーバーから DPR を受信すると、そのサーバーにのみ DPA で応答し、再利用プールからサーバーを削除しません。サーバが FIN を送信すると、アプライアンスは FIN/ACK で応答し、再利用プールから接続を削除します。
- アプライアンスがクライアントから FIN を受信すると、クライアントに FIN/ACK を送信し、FIN をブロードキャストして、すぐに再利用プールからサーバ接続を削除します。
- アプライアンスがサーバーから FIN を受信すると、FIN/ACK を送信して再利用プールから削除します。このサーバーへの新しいメッセージは、すべて新しい接続で送信されます。

直径トラフィックの負荷分散の設定

NetScaler アプライアンスが直径トラフィックの負荷分散を行うように構成するには、まずアプライアンスの Diameter パラメーターを設定し、次に Diameter Monitor を追加し、Diameter サービスを追加し、サービスをモニターにバインドし、Diameter Load Balancing 仮想サーバーを追加して、サービスを仮想サーバーにバインドする必要があります。

コマンドラインインターフェイスを使用して直径トラフィックの負荷分散を設定するには

直径パラメータを設定します。

```
set ns diameter -identity <string> -realm <string> -
   serverClosePropagation <YES|NO>
<!--NeedCopy-->
```

例:

```
set ns diameter -identity mydomain.org -realm org -
   serverClosePropagation YES
<!--NeedCopy-->
```

Diameter モニターを追加します。

```
add lb monitor <monitorName> DIAMETER -originHost <string> -originRealm
    <string>
<!--NeedCopy-->
```

例:

```
add lb monitor diameter_mon DIAMETER -originHost mydomain.org -
   originRealm org
<!--NeedCopy-->
```

Diameter サービスを作成します。

```
add service <name> <IP> DIAMETER <port>
<!--NeedCopy-->
```

例:

```
add service diameter_svc0 10.102.82.86 DIAMETER 3868

add service diameter_svc1 10.102.82.87 DIAMETER 3868

add service diameter_svc2 10.102.82.88 DIAMETER 3868

add service diameter_svc3 10.102.82.89 DIAMETER 3868
<!--NeedCopy-->
```

Diameter サービスを Diameter モニターにバインドします。

```
bind service <name>@ monitorName <monitorName>
<!--NeedCopy-->
```

例:

```
bind service diameter_svc0 -monitorName diameter_mon

bind service diameter_svc1 -monitorName diameter_mon
```



```
4
5 bind service diameter_svc2 -monitorName diameter_mon
6
7 bind service diameter_svc3 -monitorName diameter_mon
8 <!--NeedCopy-->
```

Diameter パーシステンスを備えた Diameter 負荷分散仮想サーバーを追加します。

```
1 add lb vserver <name> DIAMETER <IPAddress> <port> -persistenceType
     DIAMETER -persistAVPno <positive_integer>
2 <!--NeedCopy-->
```

例**:**

```
1 add lb vserver diameter_vs DIAMETER 10.102.112.152 3868 -
     persistenceType DIAMETER -persistAVPno 263
2 <!--NeedCopy-->
```

Diameter サービスを Diameter 負荷分散仮想サーバーにバインドします。

```
1 bind lb vserver <name> <serviceName>
2 <!--NeedCopy-->
```

例**:**

```
1 bind lb vserver diameter_vs diameter_svc0
2
3 bind lb vserver diameter_vs diameter_svc1
4
5 bind lb vserver diameter_vs diameter_svc2
6
7 bind lb vserver diameter_vs diameter_svc3
8 <!--NeedCopy-->
```

構成を保存します。

```
1 save ns config
2 <!--NeedCopy-->
```

注: **SSL_DIAMETER** サービスタイプを使用して SSL 経由の Diameter トラフィックの負荷分散を設定することもできます。

構成ユーティリティを使用して **Diameter** トラフィックの負荷分散を設定するには

1. [システム] > [設定**] > [Diameter** パラメータの変更] に移動し、直径パラメータを設定します。
2. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、Diameter タイプの負荷分散仮想サーバーを作成します。
3. Diameter タイプのサービスを作成します。

4. Diameter タイプのモニターを作成します。特別パラメータで、オリジンホストとオリジンレルムを設定します。
5. モニタをサービスにバインドし、Diameter 仮想サーバにサービスをバインドします。
6. [詳細設定] で、[持続性] をクリックし、直径を指定し、持続性 AVP 番号を入力します。
7. [保存] をクリックし、[完了] をクリックします。

FIX の負荷分散を構成する

August 15, 2023

金融情報交換（FIX）プロトコルは、取引相手国間の証券取引に関連する情報の電子交換のために、金融業界で使用されるオープンメッセージ標準です。FIX/SSL_FIX プロトコルは、買い側と売り側の企業、取引プラットフォーム、および貿易情報を伝達するための規制当局によって広く使用されています。

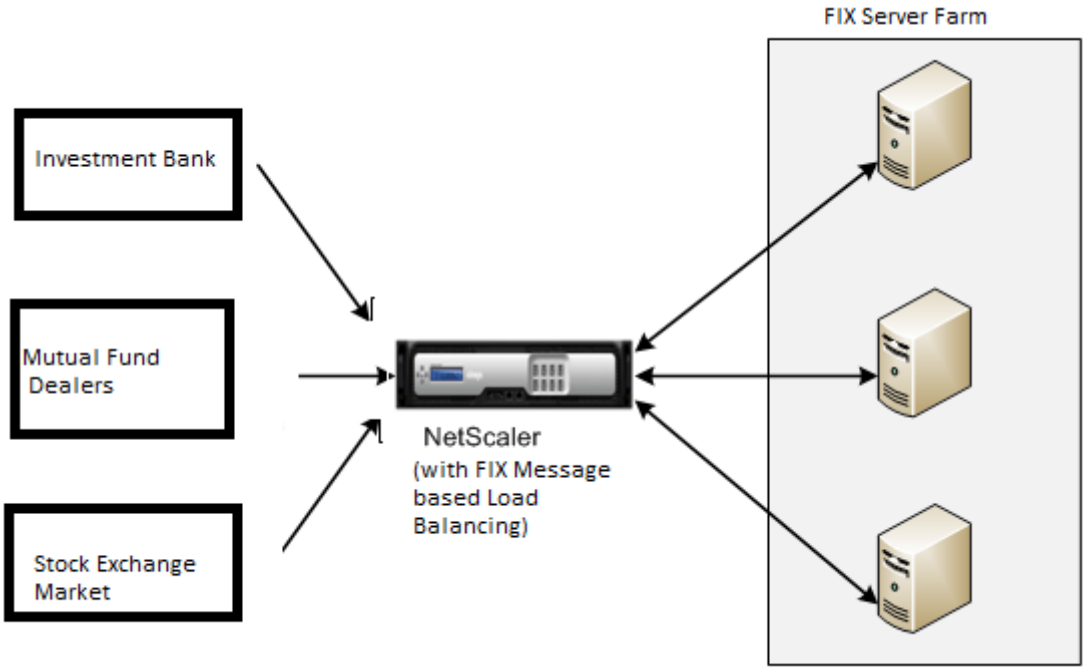
この機能を使用すると、着信 FIX メッセージを配布し、FIX メッセージングでセキュリティを提供するために、FIX または SSL_FIX ロードバランシング仮想サーバーを構成できます。NetScaler ADC は、FIX 4.1、FIX 4.2、FIX 4.3、および FIX 4.4 バージョンの FIX メッセージベースの負荷分散（MBLB）をサポートしています。

NetScaler アプライアンスの FIX MBLB には、次のような利点があります。

1. 優れた HA とヘルスモニタリングにより、FIX または SSL_FIX サーバーを効率的に管理できます。
2. すべての FIX サーバーまたは SSL_FIX サーバーへの同期保護。
3. FIX セッションの永続性。

FIX 負荷分散の仕組み

FIX MBLB セットアップには、FIX ロードバランシング仮想サーバーと複数のロードバランシングされた FIX サーバーが含まれます。FIX 仮想サーバーは、着信クライアントトラフィックを受信し、着信トラフィックを FIX メッセージに解析し、各 FIX メッセージに対して FIX サーバーを選択し、選択した FIX サーバーにメッセージを転送します。次の概念図は、標準的な FIX 負荷分散の設定を示しています。



基本的な FIX MBLB セットアップでは、FIX 仮想サーバーは、ラウンドロビン負荷分散方式を使用して、クライアントからの FIX メッセージを負荷分散された FIX サーバーに配信します。タイプ FIXSESSION の永続性が有効になっていると、FIX 仮想サーバーは、同じ FIX セッションに属する異なる FIX メッセージに対して同じサーバーを選択します。FIX セッションは、**FIX** フィールド senderCompid (タグ 49) と targetCompid (タグ 56) の値に基づいて決定されます。

FIX トラフィックの負荷分散を構成および監視する

FIX メッセージトラフィックの負荷分散を行うには、次の設定を行う必要があります。

1. FIX 負荷分散仮想サーバーの設定
2. SSL_FIX 負荷分散仮想サーバーの設定
3. FIX 負荷分散サービスの設定
4. SSL_FIX 負荷分散サービスの設定
5. FIXSESSION パーシステンスの設定
6. パーシステンスタイムアウトの設定
7. FIX/SSL_FIX 統計情報の表示
8. FIX/SSL_FIX パーシステントセッションのモニタリング

コマンドラインインターフェイスを使用して **FIX** 負荷分散サーバーを構成するには

コマンドプロンプトで入力します。


```
add lb vserver <name> FIX <IP> <PORT>
<!--NeedCopy-->
```

例

```
add lb vserver vs1 FIX 10.102.82.86 3868
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **SSL_FIX** 負荷分散仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> SSL_FIX <IP> <PORT>
<!--NeedCopy-->
```

例

```
add lb vserver vs1 SSL_FIX 10.102.82.86 3868
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **FIX** サービスを構成するには

コマンドプロンプトで入力します。

```
add service <name> <ip-addr> FIX <port>
<!--NeedCopy-->
```

例

```
add service_svc1 10.102.82.86 FIX 3868
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **SSL_FIX** サービスを構成するには

コマンドプロンプトで入力します。

```
add service <name> <ip-addr> SSL_FIX <port>
<!--NeedCopy-->
```

例

```
add service svc1 10.102.82.86 SSL_FIX 3868
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **FIXSESSION** パーシステンスを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -persistenceType FIXSESSION
<!--NeedCopy-->
```

例

```
set lb vserver vs1 -persistenceType FIXSESSION
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してパーシステンスタイムアウトを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -timeout <value>
<!--NeedCopy-->
```

例

```
set lb vserver vs1  - timeout 2
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **FIX** の統計情報を表示するには

コマンドプロンプトで入力します。

```
stat lb vserver <name>
<!--NeedCopy-->
```

例

```
stat lb vserver_svc1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **FIX** サービスを **FIX** 仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> <service name>
<!--NeedCopy-->
```

例

```
bind lb vserver vs1 svc1
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **FIX** 永続セッションを表示するには

コマンドプロンプトで入力します。

```
show lb persistentSessions <name>
<!--NeedCopy-->
```

例

```
show lb persistentSessions vs1
<!--NeedCopy-->
```

> 注
>
> 注:SSL_FIX サービスタイプを使用して SSL 経由の FIX トラフィックの負荷分散を設定できるようになりました。このサービスは、FIX メッセージの安全な通信を提供します。

GUI を使用して **FIX** 負荷分散仮想サーバーを構成するには

1. [構成] > [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動し、[追加] をクリックして FIX 負荷分散仮想サーバーを作成します。
2. 負荷分散仮想サーバーページで、サーバーパラメーターを設定します。
    a) 仮想サーバー名
    b) プロトコルタイプは「FIX」
    c) サーバーの IP アドレスの種類
    d) サーバー IP アドレス
    e) サーバのポート番号
3. 「**OK**」および「続行」をクリックして、他のパラメータを設定します。
4. [サービス] セクションで、新しい FIX 負荷分散仮想サービスを選択または追加し、FIX サーバーにバインドします。
5. **Persistence** セクションで、以下のパラメータを設定します。
    a) パーシステンスタイプは「フィックスセッション」
    b) タイムアウト間隔
6. [**OK]**、[完了**]** の順にクリックします。

GUI を使用して **FIX** 負荷分散仮想サーバーを編集するには

[構成] > [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動し、FIX サーバーを選択して [編集] をクリックします。

GUI を使用して **FIX** 負荷分散仮想サーバーを削除するには

[構成] > [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動し、FIX サーバーを選択して [削除] をクリックします。

GUI を使用して **FIX** 負荷分散仮想サービスを構成するには

1. [設定] > [トラフィック管理] > [負荷分散] > [サービス] ページに移動し、[追加] をクリックして FIX 負荷分散仮想サービスを作成します。
2. [サービス] ページで、次のパラメータを設定します。[詳細] 矢印をクリックして、トラフィックドメイン、ハッシュ ID、サーバ ID、キャッシュタイプ、アクティブ接続数などの他のパラメータを設定できます。
    a) サービス名—FIX 仮想サービス名
    b) 仮想サーバーのタイプを (新規または既存) として選択
    c) プロトコル—プロトコルタイプは「FIX」
    d) サーバ—仮想サーバの IP アドレス
    e) ポート—サーバーのポート番号
3. **OK andContinue** をクリックして、モニター、しきい値とタイムアウト、プロファイル、ポリシーなどの他のパラメータを設定します。
4. [**OK]**、[完了**]** の順にクリックします。

GUI を使用して **FIX** 負荷分散仮想サービスを編集するには

[設定] > [トラフィック管理] > [負荷分散] > [サービス] ページに移動し、**FIX** サービスを選択して [編集] をクリックします。

GUI を使用して **FIX** 負荷分散仮想サービスを削除するには

[構成] > [トラフィック管理] > [負荷分散] > [サービス] ページに移動し、FIX サービスを選択し、[削除] をクリックします。

FIX 負荷分散サーバーの統計情報を表示するには

[設定] > [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動し、[統計] をクリックして FIX サーバーの統計を表示します。

GUI を使用して **FIX** サーバーの永続セッションを表示するには

[設定] > [トラフィック管理] ページに移動し、[監視セッション] で [仮想サーバー永続セッション] をクリックします。

GUI を使用して **FIX** サーバーの永続セッションをクリアするには

1. [設定] > [トラフィック管理] ページに移動し、[監視セッション**]** で **[**** 永続セッションをクリア ******] をクリックします。
2.「永続セッションのクリア」ページで、次のパラメータを設定します。
   a) 仮想サーバー—FIX 仮想サーバーを選択
   b) 持続性パラメータ—FIX 持続性パラメータを選択
3.［**OK**］をクリックします。

MQTT 負荷分散

August 15, 2023

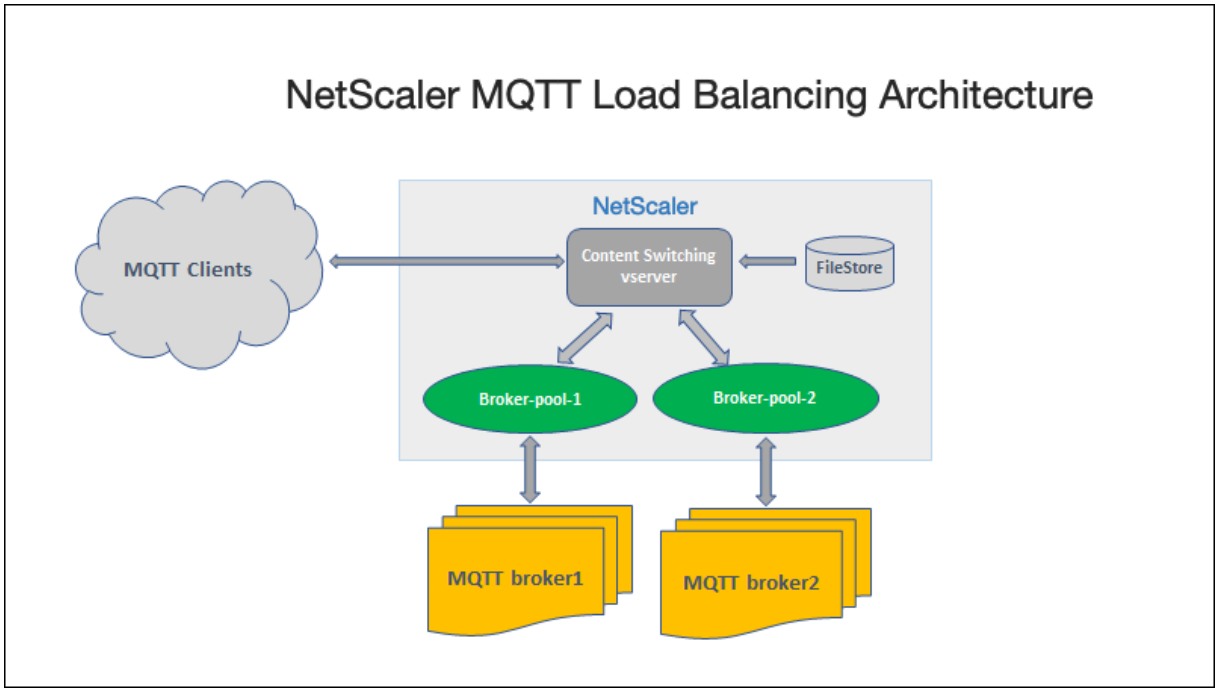
メッセージキューテレメトリトランスポート (MQTT) は、モノのインターネット (IoT) 用の OASIS 標準メッセージングプロトコルです。MQTT は、IoT システム内で効果的な通信を実現する柔軟で使いやすいテクノロジーです。MQTT はブローカーベースのプロトコルで、クライアントとブローカー間のメッセージ交換を容易にするために広く使用されています。

MQTT には次の主な利点があるため、お使いの IoT デバイスに最適なオプションです。

- 信頼性
- 迅速な応答時間
- サポートできるデバイス数は無制限
- 多対多のコミュニケーションに最適なパブリッシュ/サブスクライブメッセージ

IoT は、センサー、ソフトウェア、ネットワーク接続、および必要な電子機器が組み込まれた相互接続されたデバイスのネットワークです。組み込みコンポーネントにより、IoT デバイスはデータを収集して交換できます。IoT デバイスの使用の増加は、ネットワークインフラストラクチャに複数の課題をもたらしますが、その主な課題はスケールです。IoT デバイスを大規模に導入する場合、各 IoT デバイスによって生成されたデータを迅速に分析する必要があります。スケール要件とリソースの効率的な使用を実現するには、ブローカープールの負荷を均等に分散する必要があります。MQTT プロトコルのサポートにより、IoT 展開で NetScaler ADC アプライアンスを使用して、MQTT トラフィックの負荷分散を行うことができます。

次の図は、NetScaler ADC アプライアンスを使用して MQTT トラフィックの負荷を分散する MQTT アーキテクチャを示しています。



MQTT プロトコルを使用した IoT デプロイメントには、次のコンポーネントがあります。

- **MQTT** ブローカー。クライアントからすべてのメッセージを受信し、そのメッセージを適切な宛先クライアントにルーティングするサーバー。ブローカーは、すべてのメッセージの受信、メッセージのフィルタリング、各メッセージの購読者の決定、および購読しているクライアントへのメッセージの送信を行います。ブローカーは、すべてのメッセージが通過する中心的なハブです。
- **MQTT** クライアント。マイクロコントローラーから本格的なサーバーまで、MQTT ライブラリを実行し、ネットワーク経由で MQTT ブローカーに接続するあらゆるデバイス。発行元とサブスクライバーの両方が MQTT クライアントです。発行元とサブスクライバーのラベルは、クライアントがメッセージをパブリッシュしているのか、それともサブスクライブしてメッセージを受信しているのかを示します。
- **MQTT** ロードバランサー。NetScaler ADC アプライアンスは、MQTT トラフィックを負荷分散するために MQTT 負荷分散仮想サーバーで構成されています。

一般的な IoT 展開では、ブローカー（サーバーのクラスタ）が IoT デバイスのグループ（IoT クライアント）を管理します。NetScaler ADC アプライアンスは、クライアント ID、トピック、ユーザー名などのさまざまなパラメーターに基づいて、ブローカーへの MQTT トラフィックの負荷を分散します。

MQTT トラフィックの負荷分散の設定

NetScaler アプライアンスが MQTT トラフィックの負荷分散を行うには、次の構成タスクを実行します。

1. MQTT/MQTT_TLS サービスまたはサービスグループを設定します。
2. MQTT/MQTT_TLS 負荷分散仮想サーバーを設定します。
3. MQTT/MQTT_TLS サービスを MQTT/MQTT_TLS 負荷分散仮想サーバーにバインドします。

4. MQTT/MQTT_TLS コンテンツスイッチング仮想サーバーを設定します。
5. ターゲットの負荷分散仮想サーバーを指定するコンテンツスイッチングアクションを設定します。
6. コンテンツスイッチングポリシーを設定します。
7. コンテンツスイッチングポリシーを、特定の負荷分散仮想サーバーにリダイレクトするようにすでに構成されているコンテンツスイッチング仮想サーバーにバインドします。
8. 構成を保存します。

CLI を使用して **MQTT** トラフィックの負荷分散を設定するには

MQTT/MQTT_TLS サービスまたはサービスグループを設定します。

```
add service <name> <IP> <protocol> <port>
add servicegroup <ServiceGroupName> <Protocol>
bind servicegroup <serviceGroupName> <IP> <port>
<!--NeedCopy-->
```

例:

```
add service srvc1 10.106.163.3 MQTT 1883
add servicegroup srvcg1 MQTT
bind servicegroup srvcg1 10.106.163.3 1883
<!--NeedCopy-->
```

MQTT/MQTT_TLS 負荷分散仮想サーバーを設定します。

```
add lb vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

例:

```
add lb vserver lb1 MQTT 10.106.163.9 1883
<!--NeedCopy-->
```

MQTT/MQTT_TLS サービスまたはサービスグループを MQTT 負荷分散仮想サーバーにバインドします。

```
bind lb vserver <name> <serviceName>
bind lb vserver <name> <servicegroupName>
<!--NeedCopy-->
```

例:

```
bind lb vserver lb1 srvc1
bind lb vserver lb1 srvcg1
<!--NeedCopy-->
```

MQTT/MQTT_TLS コンテンツスイッチング仮想サーバーを設定します。

```
add cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```


例:

```
add cs vserver cs1 MQTT 10.106.163.13 1883
<!--NeedCopy-->
```

ターゲットの負荷分散仮想サーバーを指定するコンテンツスイッチングアクションを設定します。

```
add cs action <name> -targetLBVserver <string> [-comment <string>]
<!--NeedCopy-->
```

例:

```
add cs action act1 -targetlbvserver lbv1
<!--NeedCopy-->
```

コンテンツスイッチングポリシーを設定します。

```
add cs policy <policyName> [-url <string> | -rule <expression>]  -
    action <actName>
<!--NeedCopy-->
```

例:

```
add cs policy cspol1  -rule   “MQTT.COMMAND.EQ(CONNECT) && MQTT.CONNECT
    .FLAGS.QOS.eq(2)”  -action act1
<!--NeedCopy-->
```

コンテンツスイッチングポリシーを、特定の負荷分散仮想サーバーにリダイレクトするようにすでに構成されているコンテンツスイッチング仮想サーバーにバインドします。

```
bind cs vserver <virtualServerName> -policyName <policyName> -priority
    <positiveInteger>
<!--NeedCopy-->
```

例:

```
bind cs vserver cs1  - policyName cspol1 -priority 20
<!--NeedCopy-->
```

構成を保存します。

```
save ns config
<!--NeedCopy-->
```

GUI を使用して **MQTT** トラフィックの負荷分散を設定するには

1. **[**** トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、**MQTT** または **MQTT_TLS** タイプの負荷分散仮想サーバーを作成します。**
2. MQTT タイプのサービスまたはサービスグループを作成します。

3. サービスを MQTT 仮想サーバーにバインドします。
4. ［保存］をクリックします。

MQTT メッセージ長の制限

NetScaler アプライアンスは、メッセージ長が 65536 バイトを超えるメッセージをジャンボパケットとして扱い、デフォルトで破棄します。`dropmqttjumbomessage` lb パラメータは、ジャンボパケットを処理するかどうかを決定します。このパラメータはデフォルトで **YES** に設定されています。つまり、ジャンボ MQTT パケットはデフォルトでドロップされます。このパラメータを **NO** に設定すると、ADC アプライアンスはメッセージ長が 65536 バイトを超えるパケットも処理します。

CLI を使用してジャンボパケットを処理するように ADC アプライアンスを設定するには:

```
Set lb parameter  -dropMqttJumboMessage [YES | NO]
<!--NeedCopy-->
```

例**:**

```
set lb parameter  -dropMqttJumboMessage no
<!--NeedCopy-->
```

負荷分散構成を障害から保護する

August 15, 2023

負荷分散仮想サーバーに障害が発生した場合、または仮想サーバーが過剰なトラフィックを処理できない場合、負荷分散の設定が失敗することがあります。以下の設定により、ロードバランシングの設定を障害から保護できます。

- NetScaler ADC アプライアンスは、余分なトラフィックを代替 URL にリダイレクトし、
- バックアップ負荷分散仮想サーバ、および
- ステートフル接続のフェイルオーバー。

クライアント要求を別の **URL** にリダイレクトする

October 25, 2023

HTTP または HTTPS タイプの負荷分散仮想サーバーがダウンまたは無効になった場合、HTTP 302 リダイレクトを使用してリクエストを代替 URL にリダイレクトできます。代替 URL は、サーバーのステータスに関する情報を提供できます。設定されたリダイレクト URL は、HTTP 応答のロケーションヘッダーで指定されます。レスポンスで指定される正確な URL は、次の設定オプションによって異なります。

- 設定したリダイレクト URL にドメイン名のみが含まれている場合(http://www.sample1.example.comなど)、HTTP 応答で指定されたリダイレクト URL にはユニフォームリソース識別子（URI）が追加されます。設定されたドメイン名への HTTP リクエストで指定されます。たとえば、リクエストに GET http://www.sample2.example.com/images/site_nav.png ヘッダーが含まれている場合、リダイレクトレスポンスの location ヘッダーは location: http://www.sample1.example.com/images/site_nav.png ヘッダーを指定します。

  > 注:
  >
  > 要求と応答のドメイン名は異なる場合があります。このトピックでは、概念を説明するために、この 2 つのドメインを sample1.example.com と sample2.example.com と呼びます。

- 設定済みのリダイレクト URL に完全なパスが含まれている場合、リダイレクトレスポンスでは、リクエストの URI に関係なく、設定済みの完全な URL が指定されます。たとえば、次のような URL があります。
  - リクエストされた URL- http://www.redirect.com/en/index.html
  - リダイレクト URL- http://www.redirect.com/en/site_down.html

次の表は、前述の設定オプションの一覧です。

設定済みリダイレクト URL	HTTP リクエスト内の URL	HTTP レ
http://www.sample1.example.com	http://www.sample2.example.com/en/index.html	http://w
http://www.sample1.example.com/en/error.html	http://www.sample2.example.com/en/index.html	http://w

> 注
>
> - リダイレクト URL を設定する場合、http://example.com URL は http://example.com/ URL と同じではありません。後者には Webroot パス/への完全なパスが含まれているためです。
> - 負荷分散仮想サーバーで、バックアップ仮想サーバーとリダイレクト URL の両方を構成した場合、バックアップ仮想サーバーがリダイレクト URL よりも優先されます。リダイレクトは、プライマリ仮想サーバーとバックアップ仮想サーバーの両方がダウンしている場合にのみ使用されます。

CLI を使用してクライアント要求を **URL** にリダイレクトするように仮想サーバーを構成するには

1. 負荷分散仮想サーバーを作成します。

   ```
   set lb vserver -redirect url
   ```

2. リダイレクト URL オプションが期待どおりに機能していることを確認します。仮想サーバーを無効にします。

   ```
   disable vserver <vserver_name>
   ```


3. Web ブラウザから Web サイトの URL にアクセスして、リクエストが期待どおりにリダイレクトされることを確認します。Web サイトにアクセスする前に、Web ブラウザのキャッシュをクリアして新しい接続を確立する必要がある場合があります。
4. 仮想サーバーを有効にします。

```
enable vserver <vserver_name>
```

GUI を使用してクライアント要求を **URL** にリダイレクトするように仮想サーバーを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ペインで、新しい仮想サーバーを追加するには、[追加] をクリックします。
3. 既存の仮想サーバーを編集するには、リストから仮想サーバーを選択し、[編集] をクリックします。
4. [詳細設定] タブで、[保護] をクリックします。「リダイレクト **URL** 」フィールドに、リダイレクト URL（例:http://www.newdomain.com/mysite/maintenance）を入力します。
5. [**OK**] をクリックします。

バックアップの負荷分散仮想サーバーを構成する

August 15, 2023

プライマリ負荷分散仮想サーバーがダウンまたは使用できないときに、バックアップ仮想サーバーに要求を送信するように NetScaler アプライアンスを構成できます。バックアップ仮想サーバーはプロキシであり、クライアントに対して透過的です。アプライアンスは、サイトの停止に関する通知メッセージをクライアントに送信することもできます。

バックアップ負荷分散仮想サーバーは、主要な方法が使用できない場合の中断を最小限に抑え、負荷分散環境の可用性と信頼性を向上させます。

> 注:
>
> バックアップ仮想サーバーは、プライマリ仮想サーバーが削除または無効化された後でも、既存の接続を引き続き処理します。

バックアップロードバランシング仮想サーバは、作成時に構成することも、既存の仮想サーバのオプションパラメータを変更することもできます。既存のバックアップ仮想サーバー用にバックアップ仮想サーバーを構成して、カスケードバックアップ仮想サーバーを作成することもできます。バックアップ仮想サーバーをカスケードする最大の深さは、10 です。

2 台のサーバーに接続する複数の仮想サーバーがある場合、プライマリ仮想サーバーがダウンしてから稼働を再開した場合にどうなるかを選択できます。デフォルトの動作では、プライマリ仮想サーバの役割をプライマリとして再開

します。ただし、バックアップ仮想サーバが引き継ぐときに制御を維持するように構成することはできます。たとえば、バックアップ仮想サーバ上の更新をプライマリ仮想サーバに同期し、元のプライマリサーバにそのロールを手動で強制的に再開させることができます。この場合、プライマリ仮想サーバがダウンしてから復帰したときに制御を維持するようにバックアップ仮想サーバを指定できます。

プライマリ仮想サーバーとバックアップ仮想サーバーの両方がダウンしているか、要求を処理するしきい値に達したときのフォールバックとして、プライマリ負荷分散仮想サーバーのリダイレクト URL を構成できます。仮想サーバーにバインドされたサービスが OUT OF SERVICE になると、アプライアンスはリダイレクト URL を使用します。

次の負荷分散方法を選択すると、**Backup LB** Method が表示されます。

- 最小接続数
- 最短の応答時間
- ラウンドロビン
- 最小帯域幅
- 最小パケット数
- カスタムロード
- 最小リクエスト
- 静的近接度

> 注
>
> 負荷分散仮想サーバーがバックアップ仮想サーバーとリダイレクト URL の両方で構成されている場合、バックアップ仮想サーバーがリダイレクト URL よりも優先されます。リダイレクトは、プライマリ仮想サーバーとバックアップ仮想サーバーがダウンしている場合にのみ使用されます。

CLI を使用してバックアップ仮想サーバーを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <vServerName> -backupVserver <BackupVServerName> [-
   disablePrimaryOnDown]
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -backupVserver Vserver-LB-2 -
   disablePrimaryOnDown
<!--NeedCopy-->
```

GUI を使用してバックアップ仮想サーバーを設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [保護] をクリックし、バックアップ仮想サーバーを選択します。

3. プライマリ仮想サーバーが復旧しても、プライマリ仮想サーバーを手動で有効にするまでバックアップ仮想サーバーを制御し続ける場合は、「ダウン時にプライマリを無効化」を選択します。

> 注： NetScaler バージョン 12.1 ビルド 51.xx 以降、GUI にはそのサーバーの有効状態が表示され、バックアップがアクティブかどうかが示されます。
>
> 現在のサーバーの有効な状態は、次のいずれかになります。
>
> - **UP** —サーバーが稼働中であることを示します
> - **DOWN** —サーバーがダウンしていることを示します
> - **UP (Backup Active)** —プライマリまたはセカンダリの仮想サーバーのいずれかが稼働していて、トラフィックがバックアップ仮想サーバーに転送されていることを示します。
> - **DOWN (Backup Active)** —プライマリ仮想サーバーとバックアップ仮想サーバーの両方がダウンしていて、トラフィックがバックアップ仮想サーバーにルーティングされていることを示します。
>
> プライマリ仮想サーバーで **Disable When Down** オプションが有効になっていて、プライマリサーバーがダウンして再び稼働状態に戻った場合でも、プライマリ仮想サーバーが明示的に再度有効になるまで、トラフィックはバックアップ仮想サーバーによって処理されます。`enable lb vserver <vserver_name>` コマンドコマンドを使用して、プライマリ仮想サーバーを再度有効にできます。

スピルオーバーを構成する

August 15, 2023

アプライアンスのスピルオーバー構成は、スピルオーバー方式、スピルオーバーしきい値、およびバックアップ仮想サーバーで構成されるプライマリ仮想サーバーで構成されます。バックアップ仮想サーバーをスピルオーバーするように構成して、バックアップ仮想サーバーのチェーンを作成することもできます。

スピルオーバー方式では、スピルオーバー設定の基礎となる動作条件 (確立された接続の数、帯域幅、またはサーバーファームの総合状態など) を指定します。新しい接続が確立されると、アプライアンスはプライマリ仮想サーバーが稼働していることを確認し、動作状態を設定されたスピルオーバーしきい値と比較します。しきい値に達すると、スピルオーバー機能によって新しい接続がバックアップチェーンで最初に使用可能な仮想サーバーに転送されます。バックアップ仮想サーバーは、プライマリの負荷がしきい値を下回るまで、受信した接続を管理します。

スピルオーバーパーシステンスを構成すると、プライマリの負荷がしきい値を下回った後でも、バックアップ仮想サーバーは受信した接続を処理し続けます。スピルオーバーパーシステンスとスピルオーバーパーシステンスタイムアウトを設定すると、バックアップ仮想サーバーは、プライマリの負荷がしきい値を下回った後、指定された期間のみ接続を処理します。

注: 通常、スピルオーバーメソッドに関連する値がしきい値（接続数など）を超えると、スピルオーバーがトリガーされます。ただし、サーバーヘルスのスピルオーバー方式では、サーバーファームの状態がしきい値を下回るとスピルオーバーがトリガーされます。

スピルオーバーは、次のいずれかの方法で設定できます。

- 定義済みのスピルオーバーメソッドを指定します。あらかじめ定義された 4 つのメソッドが用意されており、これらは一般的なスピルオーバー要件を満たしています。
- ポリシーベースのスピルオーバーを設定します。ポリシーベースのスピルオーバーでは、NetScaler ルールを使用してスピルオーバーが発生する条件を指定します。NetScaler のルールにより、さまざまな運用条件に合わせてスピルオーバーを柔軟に設定できます。

定義済みのメソッドが要件を満たさない場合は、ポリシーベースのスピルオーバーを使用してください。プライマリ仮想サーバに両方を設定すると、ポリシーベースのスピルオーバー設定が定義済みの方法よりも優先されます。

まず、プライマリ仮想サーバーとバックアップチェーンに必要な仮想サーバーを作成します。バックアップチェーンを設定するには、1 台の仮想サーバーをプライマリのバックアップとして (つまり、セカンダリ仮想サーバーを作成します)、1 台の仮想サーバーをセカンダリのバックアップとして指定する (つまり、3 次仮想サーバーを作成します) というように指定します。次に、定義済みのスピルオーバー方法を指定するか、スピルオーバーポリシーを作成してバインドすることによって、スピルオーバーを構成します。

仮想サーバーを別の仮想サーバーのバックアップとして割り当てる手順については、「バックアップ負荷分散仮想サーバーの構成」を参照してください。

定義済みのスピルオーバーメソッドの構成

定義済みのスピルオーバーメソッドは、一般的なスピルオーバー要件のいくつかを満たしています。定義済みのスピルオーバー方法のいずれかを使用するには、プライマリ仮想サーバーでスピルオーバーパラメータを設定します。バックアップ仮想サーバーのチェーンを作成するには、バックアップ仮想サーバーのスピルオーバーパラメーターも構成します。

バックアップ仮想サーバーが独自のしきい値に達し、サービスタイプが TCP の場合、NetScaler アプライアンスはクライアントに TCP リセットを送信します。サービスタイプが HTTP、SSL、RTSP の場合、新しい要求をプライマリ仮想サーバーに設定されたリダイレクト URL に転送します。リダイレクト URL は、HTTP、SSL、および RTSP 仮想サーバーにのみ指定できます。リダイレクト URL が設定されていない場合、NetScaler アプライアンスはクライアントに TCP リセット（仮想サーバーのタイプが TCP の場合）または HTTP 503 応答（仮想サーバーのタイプが HTTP または SSL の場合）を送信します。

注：RTSP 仮想サーバーでは、NetScaler アプライアンスはスピルオーバーにデータ接続のみを使用します。バックアップ RTSP 仮想サーバが使用できない場合、要求は RTSP URL にリダイレクトされ、RTSP リダイレクトメッセージがクライアントに送信されます。

コマンドラインインターフェイスを使用して仮想サーバーの事前定義済みのスピルオーバー方式を構成するには

コマンドプロンプトで入力します。


```
set lb vserver <vServerName> -soMethod <spillOverType> -soThreshold <
    positiveInteger> -soPersistence ENABLED -soPersistenceTimeout <
    positiveInteger>
<!--NeedCopy-->
```

例

```
set lb vserver Vserver-LB-1 -soMethod Connection -soThreshold 1000 -
    soPersistence enabled -soPersistenceTimeout 2
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーの事前定義済みのスピルオーバー方式を構成するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [保護] をクリックし、スピルオーバーパラメータを設定します。

ポリシーベースのスピルオーバーの設定

ルール（式）に基づくスピルオーバーポリシーを使用すると、より広範なスピルオーバーシナリオに合わせてアプライアンスを設定できます。たとえば、仮想サーバーの応答時間に基づいて、または仮想サーバーのサージキュー内の接続数に基づいてスピルオーバーを構成できます。

ポリシーベースのスピルオーバーを設定するには、まずスピルオーバーアクションを作成します。次に、スピルオーバーポリシーで使用する表現を選択し、ポリシーを設定して、アクションをそれに関連付けます。最後に、スピルオーバーポリシーを負荷分散、コンテンツスイッチング、またはグローバルサーバー負荷分散仮想サーバーにバインドします。優先度番号を使用して、複数のスピルオーバーポリシーを仮想サーバーにバインドできます。アプライアンスは、スピルオーバーポリシーを優先度番号の昇順で評価し、最後のポリシーに関連するアクションを実行して TRUE と評価します。

仮想サーバーにはバックアップアクションを実行することもできます。バックアップアクションは、仮想サーバーに 1 つ以上のバックアップ仮想サーバーがない場合、またはすべてのバックアップ仮想サーバーがダウンしているか、無効になっているか、独自のスピルオーバー制限に達した場合に実行されます。

スピルオーバー・ポリシーによって UNDEF 条件（ポリシー評価の結果が未定義の場合に発生する例外）が発生すると、UNDEF アクションが実行されます。UNDEF アクションは常に「承認」です。任意の UNDEF アクションは指定できません。

スピルオーバーアクションの設定

スピルオーバーアクションは、関連するスピルオーバーポリシーが TRUE と評価されたときに実行されます。現在、サポートされているスピルオーバーアクションは SPILLOVER のみです。

コマンドラインインターフェイスを使用してポリシーベースのスピルオーバーを構成するには　コマンドプロンプトで次のコマンドを入力してスピルオーバーポリシーを構成し、構成を確認します。

```
add spillover action <name> -action SPILLOVER

show spillover action <name>
<!--NeedCopy-->
```

例

```
add spillover action mySoAction -action SPILLOVER
Done
<!--NeedCopy-->
```

```
show spillover action mySoAction
1) Name: mySoAction Action: SPILLOVER
Done
<!--NeedCopy-->
```

スピルオーバーポリシーの表現の選択

ポリシー表現には、Boolean 値を返す任意の仮想サーバーベースの式を使用できます。たとえば、次のいずれかの式を使用できます。

```
SYS.VSERVER("vserver").RESPTIME.GT(<int>)
SYS.VSERVER("vserver").STATE.EQ( “ <string>” ), and
SYS.VSERVER("vserver").THROUGHPUT.LT    (<int>)
<!--NeedCopy-->
```

RESPTIME、STATE、THROUGHPUT などの既存の機能に加えて、この機能で導入された次の仮想サーバベースの機能を使用できます。

Averagesurgecount　アクティブなサービスのサージキューにあるリクエストの平均数を返します。アクティブなサービスがない場合は 0 (ゼロ) を返します。コンテンツスイッチングまたはグローバルサーバーの負荷分散仮想サーバーで使用すると、UNDEF 条件が発生します。

Activeservices　アクティブなサービスの数を返します。コンテンツスイッチングまたはグローバルサーバーの負荷分散仮想サーバーで使用すると、UNDEF 条件が発生します。

Activetransactions　現在のアクティブなトランザクションの仮想サーバーレベルのカウンタの値を返します。

is_dynamic_limit_reached　仮想サーバーが管理する接続の数が動的に計算されたしきい値と等しい場合は、ブール値 TRUE を返します。動的しきい値は、稼働しているバウンドサービスの最大クライアント (最大クライアント) 設定の合計です。

ポリシー表現を使用して、定義済みの任意のスピルオーバーメソッドを実装できます。次の表は、定義済みのスピルオーバーメソッドを、その実装に使用できる式にまとめたものです。

表 1. 定義済みのスピルオーバーメソッドをポリシー表現に変換

定義済みのスピルオーバー方式	対応する表現
CONNECTION	SYS.VSERVER(“<vserver-name>”).CONNECTIONS, used with the GT(int) arithmetic function.
帯域幅	SYS.VSERVER(“<vserver-name>”).THROUGHPUT, used with the GT(int) arithmetic function.
HEALTH	SYS.VSERVER(“<vserver-name>”).HEALTH, used with the LT(int) arithmetic function.
DYNAMICCONNECTION	SYS.VSERVER(“<vserver-name>”).IS_DYNAMIC_LIMIT_REACHED 注:IS_DYNAMIC_LIMIT_REACHED 関数を使用してポリシーベースのスプルオーバーを実装する場合は、スピルオーバーに必要な統計が機能するように、仮想サーバーに対して事前定義された DYNAMICCONNECTION メソッドも構成する必要があります。が収集されます。

スピルオーバーポリシーの設定

スピルオーバーポリシーでは、ルールとしてブール式を使用して、スピルオーバーが発生するために満たす必要がある条件を指定します。

コマンドラインインターフェイスを使用してスピルオーバーポリシーを設定するには　コマンドプロンプトで次のコマンドを入力してスピルオーバーポリシーを構成し、構成を確認します。

```
1 add spillover policy <name> -rule <expression> -action <string> [-
      comment <string>]
2
3 show spillover policy <name>
4 <!--NeedCopy-->
```


例

```
> add spillover policy mySoPolicy -rule SYS.VSERVER("v1").RESPTIME.GT
    (50) -action mySoAction -comment "Triggers spillover when the
    vserver's response time is greater than 50 ms."
Done

> show spillover policy mySoPolicy

1)  Name: mySoPolicy Rule: "SYS.VSERVER("v1").RESPTIME.GT(50)" Action:
    mySoAction Hits: 0 ActivePolicy: 0
Comment: "Triggers spillover when the vserver's response time is
    greater than 50 ms."
Done
>
<!--NeedCopy-->
```

スピルオーバーポリシーを仮想サーバーにバインドする

スピルオーバーポリシーを負荷分散、コンテンツスイッチング、またはグローバルサーバー負荷分散（仮想サーバー）にバインドできます。Goto 式で評価の流れを制御することで、複数のポリシーを仮想サーバーにバインドできます。

コマンドラインインターフェイスを使用してスピルオーバーポリシーを仮想サーバーにバインドするには　コマンドプロンプトで次のコマンドを入力して、スピルオーバーポリシーを負荷分散、コンテンツスイッチング、またはグローバルサーバー負荷分散仮想サーバーにバインドし、構成を確認します。

```
bind (lb | cs | gslb) vserver <name> -policyName <string> -priority <
    positive_integer> [-gotoPriorityExpression <expression>]

show (lb | cs | gslb) vserver <name>
<!--NeedCopy-->
```

例

```
> bind lb vserver vserver1 -policyName mySoPolicy -priority 5
Done
> show lb vserver vserver1
vserver1 (2.2.2.12:80) - HTTP Type: ADDRESS
. . .

1)  Spillover Policy Name: mySoPolicy Priority: 5
GotoPriority Expression: END
Flowtype: REQUEST
Done
>
<!--NeedCopy-->
```


スピルオーバーイベントのバックアップアクションの設定

バックアップアクションでは、スピルオーバーのしきい値に達したのに、1 つ以上のバックアップ仮想サーバーが設定されていない、ダウンしている、無効になっている、または独自のしきい値に達した場合に何をすべきかを指定します。

注: 仮想サーバー上で (Spilover Method パラメーターの値として) 直接設定されている定義済みのスピルオーバーメソッドの場合、バックアップアクションは設定できません。デフォルトでは、アプライアンスはクライアントに TCP リセット（仮想サーバーのタイプが TCP の場合）または HTTP 503 応答（仮想サーバーのタイプが HTTP または SSL の場合）を送信します。

バックアップアクションは仮想サーバーで設定されます。（ポリシーで指定されたしきい値に達した後に）リクエストを受け付けたり、クライアントを URL にリダイレクトしたり、TCP または SSL 接続を確立する前でもリクエストの数がしきい値を下回るまでリクエストをドロップするように仮想サーバーを構成できます。そのため、データ構造を割り当てる前でも接続がリセットされるため、使用されるメモリリソースは少なくなります。

CLI を使用してスピルオーバー用のバックアップアクションを設定するには　コマンドプロンプトで次のコマンドを入力して、バックアップアクションを構成し、構成を確認します。

```
set lb vserver <name> -soBackupAction <soBackupAction>

show lb vserver <name>
<!--NeedCopy-->
```

例**:**

```
set lb vserver vs1 -soBackupAction REDIRECT -redirectURL `http://www.
   mysite.com/maintenance`
Done
> show lb vserver vs1
vs1 (10.102.29.76:80) - HTTP Type: ADDRESS
State: UP
. . .
Redirect URL: `http://www.mysite.com/maintenance`
. . .
Done
<!--NeedCopy-->
```

GUI を使用してスピルオーバーのバックアップアクションを構成するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [保護] をクリックし、スピルオーバーバックアップアクションを指定します。

接続フェイルオーバー

August 15, 2023

接続フェイルオーバーは、分散環境に展開されているアプリケーションへのアクセスの中断を防ぐのに役立ちます。NetScaler ADC 高可用性（HA）セットアップでは、接続フェイルオーバー（または接続ミラーリング-CM）とは、フェールオーバーが発生したときに、確立された TCP または UDP 接続をアクティブに保つことを指します。新しいプライマリ NetScaler ADC アプライアンスには、フェイルオーバー前に確立された接続に関する情報があり、それらの接続を引き続き提供します。フェイルオーバー後も、クライアントは同じ物理サーバに接続されたままになります。新しいプライマリアプライアンスは、情報を新しいセカンダリアプライアンスと同期します。L2Conn パラメータが設定されている場合、レイヤ 2 接続パラメータもセカンダリと同期されます。

> 注:
>
> HA セットアップについて考えてみます。クライアントはプライマリノードとのセッションを確立し、プライマリノードがバックエンドサーバーとのセッションを確立します。この状態でフェールオーバーがトリガーされると、既存のクライアントおよびサーバノードから新しいプライマリで受信されたパケットは古いパケットとして扱われ、クライアントとサーバの接続がリセットされます。一方、ステートレス接続フェールオーバーが有効になっている場合（USIP がオン）、フェールオーバー後、クライアントまたはサーバーノードからパケットを受信しても、接続がリセットされません。代わりに、クライアント接続とサーバー接続は動的に作成されます。

接続フェールオーバーは、ステートレスモードまたはステートフルモードのいずれかで設定できます。ステートレス接続フェイルオーバーモードでは、HA ノードはフェールオーバーされた接続に関する情報を交換しません。このメソッドには実行時のオーバーヘッドはありません。

ステートフル接続フェールオーバーモードでは、プライマリアプライアンスはフェールオーバー接続のデータを新しいセカンダリアプライアンスと同期します。

接続フェールオーバーは、展開に長期間接続がある場合に役立ちます。たとえば、FTP 経由で大きなファイルをダウンロードしていて、ダウンロード中にフェールオーバーが発生した場合、接続が切断され、ダウンロードが中止されます。ただし、ステートフルモードで接続フェールオーバーを設定すると、フェールオーバー後もダウンロードが継続されます。

NetScaler ADC アプライアンスでの接続フェイルオーバーの仕組み

ステートレス接続フェールオーバーでは、新しいプライマリアプライアンスは、受信したパケットに含まれる情報に基づいてパケットフローを再作成しようとします。

ステートフルフェールオーバーでは、ミラーリングされた接続に関する最新の情報を維持するために、プライマリアプライアンスはセカンダリアプライアンスにメッセージを送信します。セカンダリアプライアンスは、パケットに関連するデータを保持しますが、フェールオーバーの場合にのみ使用します。フェイルオーバーが発生すると、新しいプライマリ（古いセカンダリ）アプライアンスは、ミラーリングされた接続に関する保存されたデータを使用してト

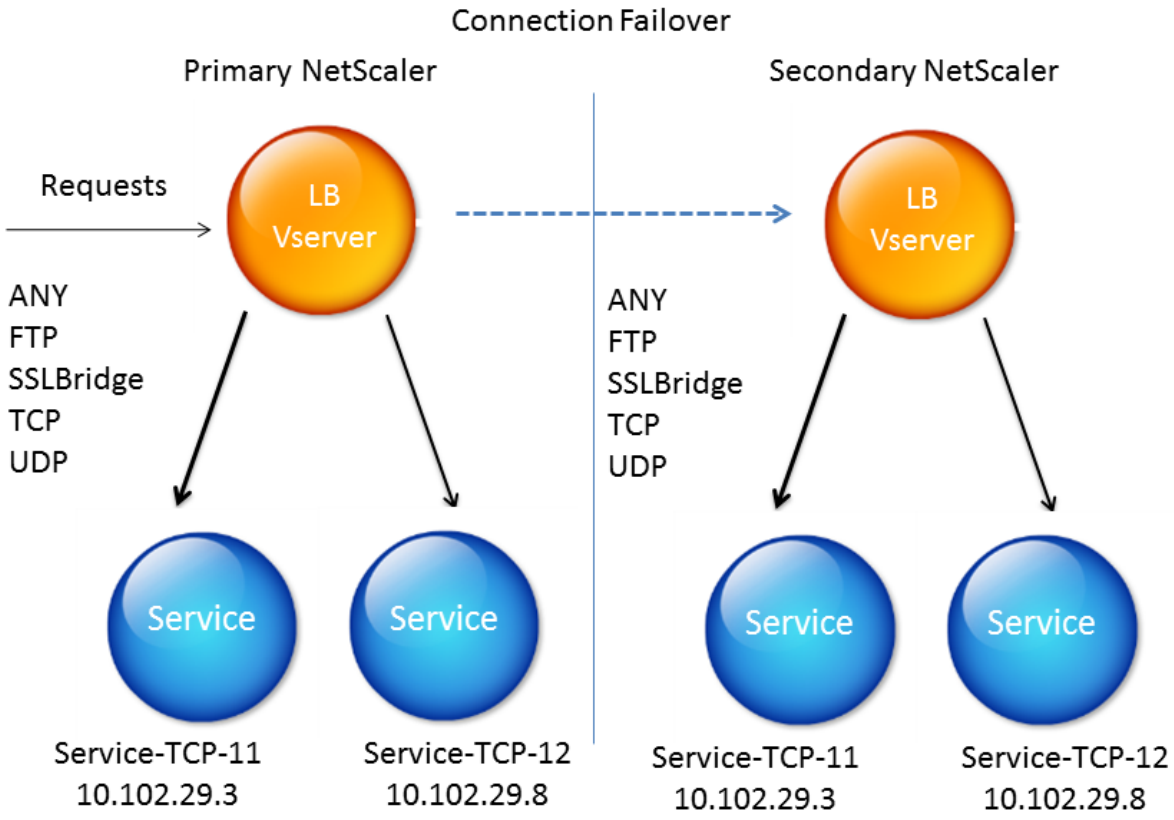
ラフィックを受け入れます。移行期間中、クライアントとサーバーで短時間の中断と再送信が発生する可能性があります。

注記:

プライマリアプライアンスがセカンダリアプライアンスで自分自身を認証できることを確認します。パスワードの正しい設定を確認するには、コマンドラインから show rpcnodeコマンドを使用するか、GUI の [ネットワーク] メニューの [RPC] オプションを使用します。

接続フェイルオーバーを使用する基本的な HA 設定には、次の図に示すエンティティが含まれます。

図 1: 接続フェールオーバーエンティティの図



> 注
>
> 次のいずれかのイベントが発生すると、接続フェールオーバーはサポートされません。
>
> ```
> - An upgrade to a later release.
> - An upgrade to a later build within the same release, if the new
>   build uses a different HA version.
> ```


サポートされているセットアップ

接続フェールオーバーは、負荷分散仮想サーバーでのみ構成できます。コンテンツスイッチング仮想サーバーでは構成できません。コンテンツスイッチング仮想サーバーに接続されている負荷分散仮想サーバーで接続のフェールオーバーを有効にすると、負荷分散仮想サーバーは最初にトラフィックを受け入れないため、接続のフェールオーバーは機能しません。

次の表に、接続フェイルオーバーでサポートされているセットアップを示します。

表 1. 接続フェールオーバー-サポートされている設定

設定	ステートレス	ステートフル
サービスタイプ	どれでも。	任意、UDP、TCP、FTP、SSL_BRIDGE。
負荷分散方法	サービスタイプ ANY でサポートされるすべてのメソッド。ただし、ソース IP 永続性が設定されていない場合は、SRCIPSRCPORTHASH メソッドを使用する必要があります。	サポートされているサービスタイプに適用されるすべての方法。
持続性タイプ	SOURCEIP パーシステンス。	サポートされているサービスタイプに適用されるすべてのタイプがサポートされています。
USIP	オンにする必要があります。	制限なし。オンまたはオフにすることができます。
サービスバインディング	サービスは 1 つの仮想サーバーにのみバインドできます。	サービスは 1 つ以上の仮想サーバーにバインドできます。
インターネットプロトコル (IP) バージョン	IPv4 と IPv6	IPv4 と IPv6
冗長性サポート	クラスタリングと高可用性	高可用性
INC モード	未サポート	仮想サーバーのサービスタイプが ANY で、モードが DSR (MAC、IPTUNNEL、TOS) で、仮想サーバーにバインドされたサービスで USIP が有効になっている場合にサポートされます。

> 注:
>
> ステートフル接続のフェールオーバーは、TCP などの接続ベースのスイッチングサービスでのみサポートされ

> ます。HTTP は要求ベースのスイッチングを使用するため、接続のフェールオーバーはサポートされません。SSL では、フェールオーバー後に既存の接続がリセットされます。

接続フェイルオーバーの影響を受ける機能

次の表に、接続フェールオーバーが構成されている場合に影響を受ける機能を示します。

表 2. 接続フェールオーバーが NetScaler ADC 機能に与える

機能	接続フェールオーバーの影響
SYN プロテクション	どの接続でも、アプライアンスが SYN-ACK を発行した後、最終的な ACK を受信する前にフェールオーバーが発生した場合、接続フェールオーバーによって接続はサポートされません。クライアントは、接続を確立するために要求を再発行する必要があります。
サージ保護	サーバとの接続が確立される前にフェイルオーバーが発生した場合、新しいプライマリアプライアンスはサーバとの接続を確立しようとします。また、サージ保護中に保持されているすべてのパケットを再送信します。
アクセスダウン	有効にすると、アクセスダウン機能は接続のフェールオーバーよりも優先されます。
Application firewall	アプリケーションファイアウォール機能はサポートされていません。
INC	独立ネットワーク構成 (INC) は、仮想サーバーのサービスタイプが ANY、モードが DSR (MAC、IPTUNNEL、TOS) で、仮想サーバーにバインドされたサービスで USIP が有効になっている場合にのみ、高可用性モードでサポートされます。その他のシナリオでは、INC はサポートされていません。
TCP バッファリング	TCP バッファリングは接続ミラーリングと互換性がありません。
応答時に閉じる	フェールオーバー後、応答時に NATPCB が閉じられないことがあります。

GUI を使用して接続のフェールオーバーを構成するには

[トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。仮想サーバーを開き、[詳細設定] で [保護] をクリックし、[ステートフル] として [接続フェイルオーバー] を選択します。

CLI を使用して接続フェールオーバーを構成するには

コマンドプロンプトで、次の操作を行います。

```
set lb vserver <vServerName> -connFailover <Value>
show lb vserver <vServerName>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -connFailover stateful
Done
<!--NeedCopy-->
```

仮想サーバで接続フェイルオーバーを無効にすると、仮想サーバに割り当てられたリソースが解放されます。

CLI を使用して接続フェールオーバーを無効にするには

コマンドプロンプトで、次の操作を行います。

```
set lb vserver <vServerName> -connFailover <Value>
show lb vserver <vServerName>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -connFailover disable
Done
<!--NeedCopy-->
```

GUI を使用して接続のフェールオーバーを無効にするには

トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。仮想サーバーを開き、[保護] で [接続フェイルオーバー] で [無効] を選択します。

サージキューをフラッシュする

August 15, 2023

物理サーバーが要求を大量に受信すると、現在接続しているクライアントへの応答が遅くなり、ユーザーは不満を抱き、不満を抱きます。多くの場合、オーバーロードにより、クライアントはエラーページを受信します。NetScaler ADC アプライアンスは、サービスへの新しい接続を確立できる速度を制御し、過負荷を回避するサージ保護などの機能を提供します。

アプライアンスは、クライアントと物理サーバ間の接続の多重化を行います。サーバー上のサービスにアクセスするクライアント要求を受信すると、アプライアンスはサーバーへの接続がすでに確立されていて、空いているかを探します。空き接続が見つかった場合、その接続を使用してクライアントとサーバー間の仮想リンクを確立します。既存の空き接続が見つからない場合、アプライアンスはサーバーとの新しい接続を確立し、クライアントとサーバー間の仮想リンクを確立します。ただし、アプライアンスがサーバーとの新しい接続を確立できない場合、クライアント要求をサージキューに送信します。負荷分散またはコンテンツスイッチング仮想サーバーにバインドされているすべての物理サーバーがクライアント接続の上限（最大クライアント値、サージ保護しきい値、またはサービスの最大容量）に達すると、アプライアンスはどのサーバーとも接続を確立できなくなります。サージ保護機能は、サージキューを使用して物理サーバーとの接続を開く速度を調整します。アプライアンスは、仮想サーバにバインドされたサービスごとに異なるサージキューを維持します。

アプライアンスが接続を確立できない要求が来るたびに、サージキューの長さが長くなります。サージキューの長さは、次の条件のいずれかで減少します。

- キュー内のリクエストがサーバーに送信されます。
- リクエストがタイムアウトし、キューから削除されます。

サービスまたはサービスグループのサージキューが長くなりすぎる場合は、それをフラッシュすることをお勧めします。特定のサービスまたはサービスグループ、または負荷分散仮想サーバーにバインドされているすべてのサービスとサービスグループのサージキューをフラッシュできます。サージキューをフラッシュしても、既存の接続には影響しません。サージキューに存在するリクエストのみが削除されます。これらのリクエストについては、クライアントは新たなリクエストを行う必要があります。

コンテンツスイッチング仮想サーバーのサージキューをフラッシュすることもできます。コンテンツスイッチング仮想サーバーが特定の負荷分散仮想サーバーにいくつかの要求を転送し、負荷分散仮想サーバーも他の要求を受信した場合、コンテンツスイッチング仮想サーバーのサージキューをフラッシュすると、このコンテンツスイッチング仮想サーバーから受信した要求のみがフラッシュされます。負荷分散仮想サーバーのサージキュー内の他の要求はフラッシュされません。

注: キャッシュリダイレクト、認証、VPN、または GSLB 仮想サーバーまたは GSLB サービスのサージキューをフラッシュすることはできません。

注:[ソース IP (USIP) を使用] が有効になっている場合は、サージ保護機能を使用しないでください。

CLI を使用してサージキューをフラッシュするには

flush ns SurgeQ コマンドは次のように機能します。

- サージキューをフラッシュする必要があるサービス、サービスグループ、または仮想サーバーの名前を指定できます。
- コマンドの実行中に名前を指定すると、指定したエンティティのサージキューがフラッシュされます。複数のエンティティが同じ名前の場合、アプライアンスはそれらすべてのエンティティのサージキューをフラッシュします。

- コマンドの実行中にサービスグループの名前、サーバー名、およびポートを指定すると、アプライアンスは指定されたサービスグループメンバーのみのサージキューをフラッシュします。
- サービスグループの名前（<name>）を指定せずにサービスグループメンバー（<serverName>と<port>）を直接指定することはできません。また、<serverName>なしで<port>を指定することはできません。特定のサービスグループメンバーのサージキューをフラッシュする場合は、<serverName>および<port>を指定します。
- 名前を指定せずにコマンドを実行すると、アプライアンスはアプライアンスに存在するすべてのエンティティのサージキューをフラッシュします。
- サービスグループメンバーがサーバ名で識別される場合は、このコマンドでサーバ名を指定する必要があります。IP アドレスは指定できません。

コマンドプロンプトで入力します。

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
<!--NeedCopy-->
```

例

```
flush ns surgeQ  -name SVC1ANZGB  -serverName 10.10.10.1 80
<!--NeedCopy-->
```

前のコマンドは、SVC1ANZGB という名前で IP アドレスが 10.10.10 のサービスまたは仮想サーバのサージキューをフラッシュします。

```
flush ns surgeQ
<!--NeedCopy-->
```

前のコマンドは、アプライアンスのすべてのサージキューをフラッシュします。

GUI を使用してサージキューをフラッシュするには

[トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバーを選択して、[アクション] リストで [フラッシュサージキュー] を選択します。

負荷分散セットアップを管理する

August 15, 2023

既存の負荷分散設定は、変更されていない限り維持するのにそれほど多くの作業を必要としませんが、ほとんどの場合、長期間変更されないことはありません。負荷を増やすには、新しい負荷分散サーバーと、最終的に新しい NetScaler ADC アプライアンスを構成し、既存のセットアップに追加する必要があります。古いサーバーは消耗し、

交換する必要があります。一部のサーバーを取り外し、他のサーバーを追加する必要があります。ネットワーク機器のアップグレードやトポロジの変更には、ロードバランシングの設定の変更が必要になる場合があります。したがって、サーバーオブジェクト、サービス、および仮想サーバーに対して操作を実行する必要があります。ビジュアライザーは、構成をグラフィカルに表示し、表示内のエンティティに対して操作を実行することができます。また、ロードバランシング設定を通じてトラフィックの管理を容易にする他の機能も利用できます。

サーバーオブジェクトを管理する

August 15, 2023

基本的な負荷分散セットアップ時に、サービスを作成するときに、サービスの IP アドレスを持つサーバーオブジェクトが存在しない場合は作成されます。IP アドレスではなくドメイン名で名前が付けられたサービスオブジェクトを使用する場合は、1 つ以上のサーバーオブジェクトを手動で作成することもできます。どのサーバーオブジェクトも有効化、無効化、削除できます。

サーバーオブジェクトを有効または無効にすると、そのサーバーオブジェクトに関連するすべてのサービスを有効または無効にします。サーバーオブジェクトを無効にした後に NetScaler アプライアンスを更新すると、そのサービスの状態は OUT OF SERVICE と表示されます。サーバーオブジェクトを無効にするときに待機時間を指定すると、サーバーオブジェクトは確立された接続を指定された時間だけ処理し続けますが、新しい接続は拒否されます。サーバーオブジェクトを削除すると、そのオブジェクトにバインドされているサービスも削除されます。

CLI を使用してサーバーを有効にするには

コマンドプロンプトで入力します。

```
enable server <name>
<!--NeedCopy-->
```

例**:**

```
enable server 10.102.29.5
<!--NeedCopy-->
```

GUI を使用してサーバーオブジェクトを有効または無効にするには

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Servers**] の順に移動します。
2. サーバーを選択し、[アクション] リストで [有効] または [無効] を選択します。

CLI を使用してサーバーオブジェクトを無効にするには

コマンドプロンプトで入力します。


```
disable server <name> <delay>
<!--NeedCopy-->
```

例:

```
disable server 10.102.29.5 30
<!--NeedCopy-->
```

CLI を使用してサーバーオブジェクトを削除するには

コマンドプロンプトで入力します。

```
rm server <name>
<!--NeedCopy-->
```

例:

```
rm server 10.102.29.5
<!--NeedCopy-->
```

GUI を使用してサーバーオブジェクトを削除するには

1. [**Traffic Management**] > [**Load Balancing**] > [**Servers**] の順に移動します。
2. サーバーを選択し、[削除] をクリックします。

サービスを管理する

December 8, 2023

サービスは、作成時にデフォルトで有効になっています。各サービスは個別に無効化または有効化できます。サービスを無効にするときには、通常、サービスが確立された接続を引き続き処理するが、新しい接続は拒否するまでの待機時間を指定します。待機時間を指定しない場合、サービスはただちに停止します。待機時間中、サービスの状態は OUT OF SERVICE です。

使用しなくなったサービスは削除できます。サービスを削除すると、そのサービスは仮想サーバーからバインド解除され、NetScaler 構成から削除されます。

CLI を使用してサービスを有効または無効にするには

コマンドプロンプトで入力します:


```
enable service <name>

disable service <name> <DelayInSeconds>
<!--NeedCopy-->
```

例:

```
enable service Service-HTTP-1
disable service Service-HTTP-1 30
<!--NeedCopy-->
```

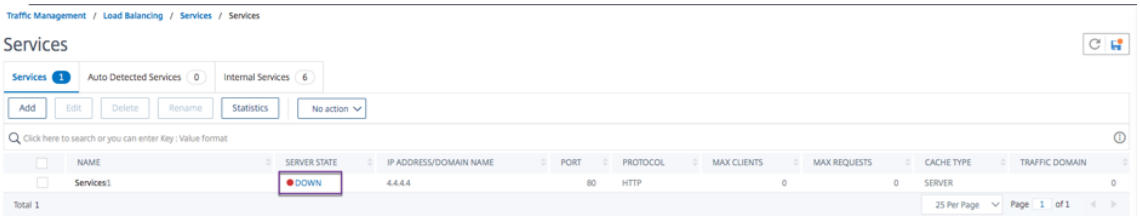
GUI を使用してサービスを有効または無効にするには

1. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。
2. サービスを開き、アクションリストで [有効化] または **[**** 無効化 ****]** を選択します。

GUI を使用してサービス状態が DOWN とマークされた原因を特定します

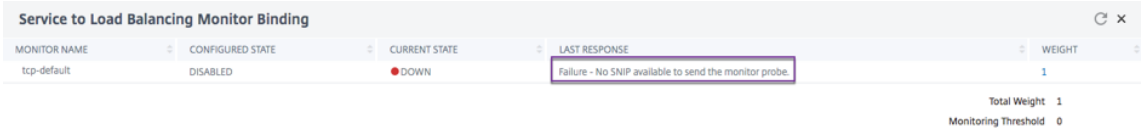
NetScaler バージョン 13.0 ビルド 41.20 以降では、モニターバインディングインターフェイスに移動しなくても、ダウンしているサービスのモニタープローブ情報を GUI で表示できます。「サービス」ページの「サーバー状態」列の値はクリック可能です。[**DOWN**] をクリックすると、サービスが [DOWN] とマークされている根本的な原因を特定できます。

1. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。
2. ダウンしているサービスに対応する [サーバーの状態] 列の [ダウン] をクリックします。



「サービスとロード・バランシング・モニターのバインディング」ページが表示されます。

「最終回答」列には、サービスがダウンとマークされた理由が表示されます。



負荷分散仮想サーバーを管理する

December 8, 2023

仮想サーバーは、作成時にデフォルトで有効になっています。仮想サーバーは手動で無効化および有効化できます。仮想サーバーを無効にすると、仮想サービスの状態は「OUT OF SERVICE」と表示されます。この場合、仮想サーバーは DownStateFlush パラメーターの設定に応じて、直ちに、または既存の接続を完了させた後に、すべての接続を終了します。DownStateFlush が有効 (デフォルト) の場合、すべての接続がフラッシュされます。DISABLED の場合、仮想サーバーは引き続き既存の接続でリクエストを処理します。

仮想サーバーを削除するのは、その仮想サーバーが不要になったときだけです。削除する前に、すべてのサービスをアンバインドする必要があります。

CLI を使用して仮想サーバーを有効または無効にするには

コマンドプロンプトで入力します:

```
enable lb vserver <name>
<!--NeedCopy-->
```

```
disable lb vserver <name>
<!--NeedCopy-->
```

例:

```
enable lb vserver Vserver-LB-1
disable lb vserver Vserver-LB-1
<!--NeedCopy-->
```

GUI を使用して仮想サーバーを有効または無効にするには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを選択し、アクションリストで [有効化] または **[**** 無効化 **] を選択します。

CLI を使用して仮想サーバーからサービスをバインド解除するには

コマンドプロンプトで入力します:

```
unbind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:


```
unbind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

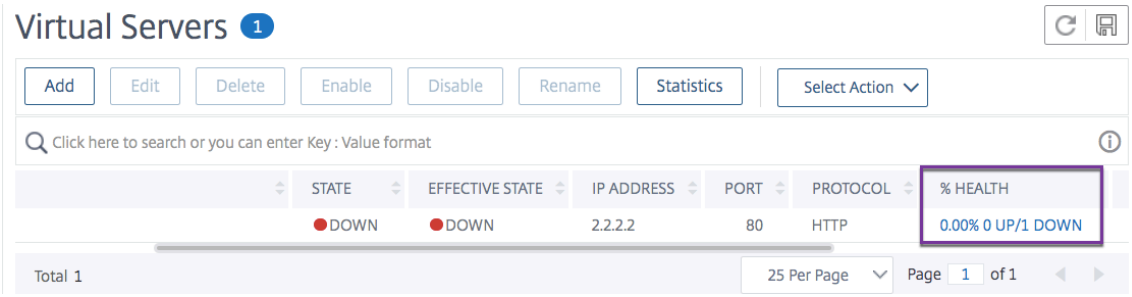
GUI を使用して仮想サーバーからサービスをバインド解除するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、「サービス」セクションをクリックします。
3. サービスを選択して [バインド解除] をクリックします。

GUI を使用して、**DOWN** とマークされた仮想サーバ状態の原因を特定する

NetScaler バージョン 13.0 ビルド 41.20 以降、モニタバインディングインターフェイスに移動することなく、ダウンしている仮想サーバーの GUI でモニタープローブ情報を表示できます。[仮想サーバー] ページの [**% HEALTH**] 列の値は、クリック可能です。**% HEALTH** 列の値をクリックすると、仮想サーバーが DOWN とマークされている根本原因を特定できます。

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. ダウンしている仮想サーバーに対応する **% HEALTH** 列の値をクリックします。

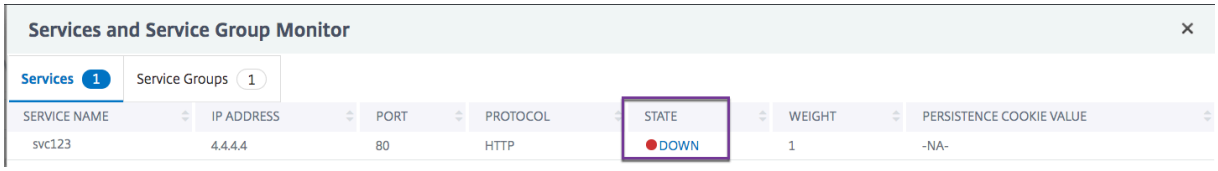


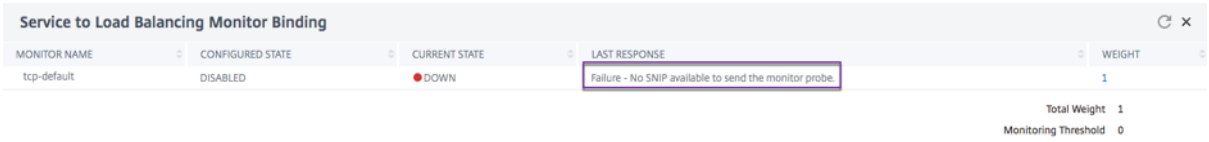
「サービスとサービスグループの監視」ページが表示されます。この仮想サーバーにバインドされているサービスとサービスグループは、それぞれのタブに表示されます。

負荷分散仮想にバインドされたサービスを使用している場合は、以下を実行してください。

「サービス」タブで、停止しているサービスに対応する「下へ」をクリックします。

「Service to Load Balancing Monitor Binding」ページの「**Last Response** 」列には、仮想サーバーがマークダウンされた理由が表示されます。

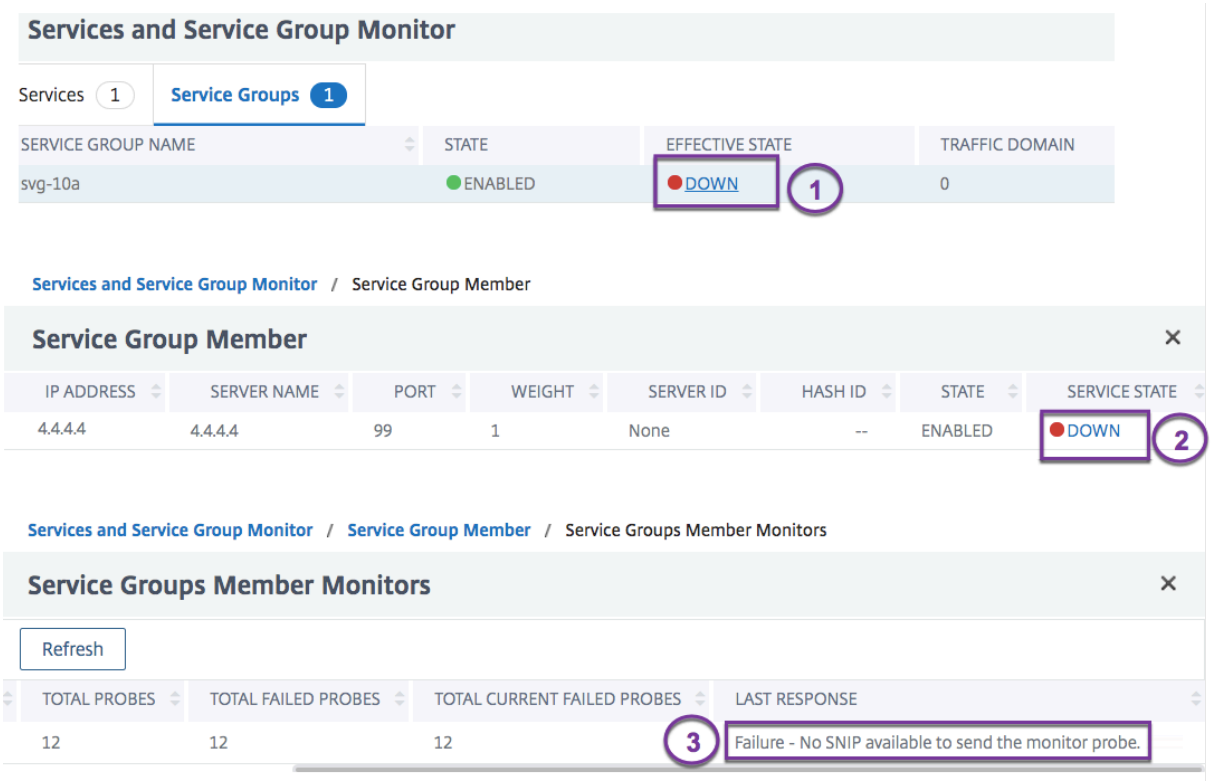




負荷分散仮想にバインドされたサービスグループを使用している場合は、以下を実行してください。

「サービスグループ」タブの「サービスとサービスグループの監視」ページで「**DOWN**」をクリックし、「サービスグループメンバー」ページで「**DOWN**」をクリックします。

「サービスグループのメンバーモニター」ページの「最終回答」列には、仮想サーバーがマークダウンされた理由が表示されます。



負荷分散ビジュアライザー

October 25, 2023

ロード・バランシング・ビジュアライザーは、ロード・バランシング構成をグラフィカルな形式で表示および変更できるツールです。以下は、ビジュアライザーの表示例です。

ビジュアライザーを使用すると、次の内容を表示できます。

- 仮想サーバーにバインドされているサービスとサービスグループ。

- 各サービスにバインドされているモニター。
- 仮想サーバーにバインドされているポリシー。
- ポリシーラベル（設定されている場合）。
- 表示されている要素の設定の詳細。

ビジュアライザーを使用して、新しいオブジェクトの追加とバインド、既存のオブジェクトの変更、オブジェクトの有効化または無効化を行うこともできます。ビジュアライザーに表示されるほとんどの構成要素は、構成ユーティリティの他の部分と同じ名前で表示されます。ただし、他の構成ユーティリティとは異なり、ビジュアライザーは、構成の詳細と監視バインディングが同じサービスをサービスコンテナと呼ばれるエンティティにグループ化します。

サービスコンテナーは、単一の負荷分散仮想サーバーにバインドされた同様のサービスとサービスグループのセットです。コンテナ内のサービスは、名前、IP アドレス、ポートを除いて同じプロパティを持ち、それらのモニタバインディングのウェイトとバインディング状態が同じである必要があります。新しいサービスを仮想サーバーにバインドすると、その構成とモニターのバインディングが他のサービスのものと一致する場合、そのサービスは既存のコンテナに配置されます。それ以外の場合は、独自のコンテナに配置されます。

次の手順では、ビジュアライザーを使用する基本的な手順のみを示します。ビジュアライザーは、負荷分散機能の他の領域の機能を複製するため、ビジュアライザーで構成できるすべての設定を表示または構成するその他の方法は、負荷分散のドキュメント全体で提供されています。

注記: ビジュアライザーにはグラフィックインターフェースが必要なので、設定ユーティリティでのみ使用できます。

Visualizer を使用して負荷分散仮想サーバーのプロパティを表示するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ウィンドウで、表示する仮想サーバーを選択し、[ビジュアライザー] をクリックします。

ビジュアライザーを使用してサービス、サービスグループ、モニターの構成の詳細を表示するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ウィンドウで、表示する仮想サーバーを選択し、[ビジュアライザー] をクリックします。
3. Load Balancing Visualizer ダイアログボックスで、エンティティをダブルクリックすると、この仮想サーバーにバインドされているエンティティの構成の詳細が表示されます。次の操作を実行できます。

構成ユーティリティのビジュアライザーを使用してポリシーとポリシーラベルの構成詳細を表示するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ウィンドウで、表示する仮想サーバーを選択し、[ビジュアライザー] をクリックします。
3. [負荷分散ビジュアライザ] ダイアログボックスで、ポリシーエンティティをダブルクリックして、この仮想サーバーにバインドされているポリシーを表示します。

ビジュアライザーを使用して負荷分散構成内のリソースを変更するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ペインで、構成する仮想サーバーを選択し、「ビジュアライザー」をクリックします。
3. 「ロード・バランシング・ビジュアライザー」ダイアログ・ボックスのビジュアライザー・イメージで、変更するリソースをダブルクリックします。

ビジュアライザーを使用して負荷分散構成を追加するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ペインで、構成する仮想サーバーを選択し、「ビジュアライザー」をクリックします。
3. 「ロード・バランシング・ビジュアライザー」ダイアログ・ボックスで、「**+**」をクリックしてリソースを追加します。

クライアントトラフィックを管理する

August 15, 2023

クライアント接続を適切に管理することで、NetScaler ADC アプライアンスの負荷が高い場合でも、ユーザーがアプリケーションを使用できるようにすることができます。アプライアンスで使用可能なさまざまな負荷分散機能やその他の機能を負荷分散設定に統合して、負荷をより効率的に処理し、必要に応じて負荷を迂回し、アプライアンスが実行する必要があるタスクの優先順位付けを行うことができます。

- セッションレス負荷分散。セッションレス負荷分散仮想サーバーを構成し、DSR または侵入検知システム (IDS) を使用する構成でセッションを作成しなくても負荷分散を実行できます。
- 統合キャッシュ。HTTP リクエストをキャッシュにリダイレクトできます。
- クリーンアップの遅延。仮想サーバー接続の遅延クリーンアップを構成して、NetScaler ADC アプライアンスの負荷が高い期間にクリーンアッププロセスで CPU サイクルが使用されないようにすることができます。
- 書き直し。書き換え機能を使用すると、HTTP リダイレクトの実行時にポートとプロトコルを変更したり、仮想サーバーの IP アドレスとポートをカスタム要求ヘッダーに挿入したりできます。
- **RTSP NAT**。
- レートベースのモニタリング。レートベースのモニタリングを有効にして、過剰なトラフィックを迂回させることができます。
- レイヤ **2** パラメータ。L2 パラメータを使用して接続を識別するように仮想サーバを設定できます。
- **ICMP** 応答。設定に従って、PING 要求に ICMP 応答を送信するようにアプライアンスを設定できます。仮想サーバに対応する IP アドレスで、ICMP 応答を VSVR_CNTRLD に設定し、仮想サーバで`ICMP VSERVER RESPONSE`を設定します。
  仮想サーバーでは、次の設定を行うことができます。

- すべての仮想サーバで`ICMP VSERVER RESPONSE`を PASSIVE に設定すると、アプライアンスは常に応答します。
- すべての仮想サーバで `ICMP VSERVER RESPONSE`を ACTIVE に設定すると、1 つの仮想サーバが稼働していてもアプライアンスは応答します。
- 一部では`ICMP VSERVER RESPONSE`を ACTIVE、他では PASSIVE に設定すると、ACTIVE に設定された 1 つの仮想サーバが稼働していてもアプライアンスは応答します。

セッションレス負荷分散仮想サーバーを構成する

August 15, 2023

NetScaler アプライアンスが負荷分散を実行すると、クライアントとサーバー間のセッションが作成および維持されます。セッション情報の管理はアプライアンスのリソースに大きな負荷をかけ、Direct Server Return（DSR）設定や侵入検知システム（IDS）の負荷分散などのシナリオではセッションが必要ない場合があります。必要のないときにセッションを作成しないようにするには、セッションレスロードバランシング用にアプライアンス上の仮想サーバーを構成できます。セッションレスロードバランシングでは、アプライアンスはパケット単位でロードバランシングを実行します。

セッションレス負荷分散は、MAC ベースの転送モードまたは IP ベースの転送モードで動作できます。

MAC ベースの転送では、トラフィックが転送されるすべての物理サーバーでセッションレス仮想サーバーの IP アドレスを指定する必要があります。

セッションレス負荷分散における IP ベースの転送では、仮想サーバーの IP アドレスとポートは転送されるパケットに含まれるため、物理サーバー上で指定する必要はありません。クライアントから物理サーバーにパケットを転送する場合、アプライアンスは IP アドレスやポートなどのクライアントの詳細を変更せず、宛先の IP アドレスとポートを追加します。

サポートされているセットアップ

NetScaler のセッションレス負荷分散は、次のサービスタイプと負荷分散方法をサポートします。

サービスタイプ

- MAC ベースのリダイレクションの場合は ANY
- IP ベースのリダイレクトには ANY、DNS、UDP

負荷分散方法

- ラウンドロビン

- 最小帯域幅
- LRTM (最小応答時間法)
- 送信元 IP ハッシュ
- 宛先 IP ハッシュ
- 送信元 IP 宛先 IP ハッシュ
- 送信元 IP 送信元ポートハッシュ
- カスタムロード

制限事項

セッションレス負荷分散には次の制限があります。

- アプライアンスはツーアームモードで導入する必要があります。
- サービスは 1 つの仮想サーバーにのみバインドする必要があります。
- セッションレス負荷分散はサービスグループではサポートされていません。
- セッションレス負荷分散は、ドメインベースのサービス (DBS サービス) ではサポートされていません。
- IP モードでのセッションレス負荷分散は、プライマリ仮想サーバーのバックアップとして構成されている仮想サーバーではサポートされません。
- スピルオーバーモードを有効にすることはできません。
- セッションレス負荷分散仮想サーバーにバインドされているすべてのサービスについて、[Use Source IP (USIP)] オプションを有効にする必要があります。
- ワイルドカード仮想サーバーまたはサービスの場合、宛先 IP アドレスは変更されません。

注:

- セッションレス負荷分散用に仮想サーバーを構成する際には、サポートされている負荷分散方法を明示的に指定してください。デフォルトの方法である Least Connection は、セッションレスの負荷分散には使用できません。
- 仮想サーバーで MAC ベースのリダイレクトモードでセッションレスロードバランシングを構成するには、NetScaler アプライアンスで MAC ベースの転送オプションを有効にする必要があります。

CLI を使用してセッションレス仮想サーバーを追加するには

コマンドプロンプトで次のコマンドを入力してセッションレス仮想サーバーを追加し、構成を確認します。

```
add lb vserver <name>@ <serviceType> <IPAddress>@ <port> -m <
    redirectionMode> -sessionless <(ENABLED|DISABLED)> -lbMethod <
    load_balancing_method>

show lb vserver <name>
<!--NeedCopy-->
```

例**:**


```
1  add lb vserver sesslessv1 any 11.11.12.123 54 -sessionless ENABLED -
       lbMethod roundrobin -m ip
2        Done
3  show lb vserver sesslessv1
4        sesslessv1 (11.11.12.123:54) - ANY Type: ADDRESS
5        State: DOWN
6        ...
7        Effective State: DOWN
8        Client Idle Timeout: 120 sec
9        Down state flush: ENABLED
10       ...
11       Persistence: NONE
12       Sessionless LB: ENABLED
13       Connection Failover: DISABLED
14       L2Conn: OFF
15       1) Policy : cmp_text Priority:8680 Inherited
16       2) Policy : cmp_nocmp_ie60 Priority:8690 Inherited
17 <!--NeedCopy-->
```

既存の仮想サーバーにセッションレスロードバランシングを設定するには

コマンドプロンプトで入力します。

```
1 set lb vserver <name>@ -m <redirectionMode> -sessionless <(ENABLED|
      DISABLED)> -lbMethod <load_balancing_method>
2 <!--NeedCopy-->
```

例

```
1 set lb vserver sesslessv1 -m mac -sessionless ENABLED -lbmethod lrtm
2   Done
3 <!--NeedCopy-->
```

> 注
>
> -m MACオプションが有効になっている仮想サーバーにバインドされているサービスの場合は、非ユーザーモニターをバインドする必要があります。

GUI を使用してセッションレス仮想サーバーを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、[詳細設定] で [トラフィック設定] をクリックし、[セッションレス負荷分散] を選択します。

HTTP リクエストをキャッシュにリダイレクトする

August 15, 2023

NetScaler ADC キャッシュリダイレクト機能は、HTTP 要求をキャッシュにリダイレクトします。キャッシュリダイレクト機能を適切に実装することで、HTTP リクエストへの応答の影響を大幅に軽減し、Web サイトのパフォーマンスを向上させることができます。

キャッシュは、頻繁に要求された HTTP コンテンツを格納します。仮想サーバーでキャッシュリダイレクトを構成すると、NetScaler アプライアンスはキャッシュ可能な HTTP リクエストをキャッシュに送信し、キャッシュ不可能な HTTP リクエストを元の Web サーバーに送信します。

CLI を使用して仮想サーバーでキャッシュリダイレクトを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -cacheable <Value>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -cacheable yes
<!--NeedCopy-->
```

GUI を使用して仮想サーバーのキャッシュリダイレクトを構成するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [トラフィック設定] をクリックし、[キャッシュ可能] を選択します。

仮想サーバー接続のクリーンアップを有効にする

August 15, 2023

特定の条件下では、サービスまたは仮想サーバーが DOWN とマークされたときに、既存の接続をただちに終了するように downStateFlush 設定を構成できます。既存の接続を終了すると、リソースが解放され、場合によっては負荷分散セットアップの回復速度が速くなります。

仮想サーバーの状態は、それにバインドされているサービスの状態によって異なります。各サービスの状態は、そのサービスにバインドされたモニターから送信されたプローブとヘルスチェックに対する負荷分散サーバーの応答によって異なります。負荷分散されたサーバーが応答しないことがあります。サーバーが遅いかビジー状態の場合、モニ

タリングプローブがタイムアウトする可能性があります。設定されたタイムアウト期間内に繰り返しモニタリングプローブに応答しない場合、サービスには DOWN とマークされます。

仮想サーバーは、その仮想サーバーにバインドされたすべてのサービスがダウンとマークされている場合にのみダウンとマークされます。仮想サーバーが DOWN になると、その直ちに、または既存の接続の完了を許可した後で、すべての接続が終了します。

トランザクションを完了する必要があるアプリケーションサーバーでは、DownStateFlush 設定を有効にしないでください。この設定は、DOWN とマークされたときに接続を安全に終了できる Web サーバー上で有効にできます。

次の表は、VServer-LB-1 という 1 つのサービスがバインドされた仮想サーバーで構成される構成例でのこの設定の影響をまとめたものです。表内の E と D は DownStateFlush 設定の状態を示しています。E は有効、D は無効を意味します。

Vserver-LB-1	Service-TCP-1	接続の状態
E	E	クライアント接続とサーバー接続の両方が終了します。
E	D	NetScaler アプライアンスが接続の再利用をサポートしていない TCP などの一部のサービスタイプでは、クライアント接続とサーバー接続の両方が終了します。アプライアンスが接続の再利用をサポートする HTTP などのサービスタイプでは、クライアント接続とサーバー接続の両方が、それらの接続でトランザクションがアクティブである場合にのみ終了されます。トランザクションがアクティブでない場合、クライアント接続のみが終了します。

Vserver-LB-1	Service-TCP-1	接続の状態
D	E	NetScaler アプライアンスが接続の再利用をサポートしていない TCP などの一部のサービスタイプでは、クライアント接続とサーバー接続の両方が終了します。アプライアンスが接続の再利用をサポートする HTTP などのサービスタイプでは、クライアント接続とサーバー接続の両方が、それらの接続でトランザクションがアクティブである場合にのみ終了されます。トランザクションがアクティブでない場合、サーバー接続のみが終了します。
D	D	クライアント接続もサーバー接続も終了しません。

サーバーまたはクライアントによって確立されたすべての接続が閉じられた場合にのみサービスを無効にする場合は、正常なシャットダウンオプションを使用できます。サービスのグレースフルシャットダウンの詳細については、「サービスのグレースフルシャットダウン」を参照してください。

CLI を使用して仮想サーバーでダウン状態のフラッシュ設定を構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -downStateFlush <Value>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -downStateFlush enabled
<!--NeedCopy-->
```

GUI を使用して仮想サーバーのダウンステートフラッシュ設定を構成するには

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [トラフィック設定] をクリックし、[ダウンステートフラッシュ] を選択します。

HTTP リダイレクト用のポートとプロトコルを書き換える

August 15, 2023

仮想サーバーとそれにバインドされているサービスは、異なるポートを使用する場合があります。サービスがリダイレクトで HTTP 接続に応答する場合、リダイレクトが正常に行われるようにポートとプロトコルを変更するように NetScaler ADC アプライアンスを構成する必要があります。そのためには、RedirectPortRewrite 設定を有効にして構成します。

この設定は HTTP トラフィックと HTTPS トラフィックにのみ影響します。この設定が仮想サーバーで有効になっている場合、仮想サーバーはリダイレクト時にポートを書き換え、サービスが使用するポートを仮想サーバーが使用するポートに置き換えます。

仮想サーバーまたはサービスのタイプが SSL の場合、仮想サーバーまたはサービスで SSL リダイレクトを有効にする必要があります。仮想サーバーとサービスの両方が SSL タイプの場合は、仮想サーバーで SSL リダイレクトを有効にします。

RedirectPortRewrite 設定は、次のシナリオで使用できます。

- 仮想サーバーのタイプは HTTP で、サービスのタイプは SSL です。
- 仮想サーバーのタイプは SSL で、サービスのタイプは HTTP です。
- 仮想サーバーのタイプは HTTP で、サービスのタイプは HTTP です。
- 仮想サーバーのタイプは SSL で、サービスのタイプは SSL です。

シナリオ 1: 仮想サーバーのタイプは HTTP で、サービスのタイプは SSL です。SSL リダイレクト、およびオプションでポートリライトがサービスで有効になっています。ポート書き換えが有効な場合、HTTPS URL のポートが書き換えられます。サーバーからの HTTP URL はそのままクライアントに送信されます。

SSL リダイレクトのみが有効です。仮想サーバーはどのポートでも構成できます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
`http://domain.com/`	`http://domain.com/`
`http://domain.com:8080/`	`http://domain.com:8080/`
`https://domain.com/`	`https://domain.com/`
`https://domain.com:444/`	`https://domain.com:444/`

SSL リダイレクトとポート書き換えは有効になっています。仮想サーバはポート *80* で設定されます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com/

SSL リダイレクトとポート書き換えは有効になっています。仮想サーバーはポート *8080* で構成されます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	http://domain.com:8080/
https://domain.com:444/	http://domain.com:8080/

シナリオ 2: 仮想サーバーのタイプは SSL で、サービスのタイプは HTTP です。ポート書き換えが有効な場合、HTTP URL のポートのみが書き換えられます。サーバーからの HTTPS URL はそのままクライアントに送信されます。

SSL リダイレクトは仮想サーバーで有効になっています。仮想サーバーはどのポートでも構成できます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com/	https://domain.com/
http://domain.com:8080/	https://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL リダイレクトとポート書き換えは仮想サーバーで有効になっています。仮想サーバーはポート *443* で構成されます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com/	https://domain.com/

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com:8080/	https://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL リダイレクトとポート書き換えは有効になっています。仮想サーバーはポート *444* で構成されます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com/	https://domain.com:444/
http://domain.com:8080/	https://domain.com:444/
https://domain.com/	https://domain.com/
https://domain.com:445/	https://domain.com:445/

シナリオ 3: 仮想サーバーとサービスのタイプは HTTP です。仮想サーバーでポート書き換えが有効になっている必要があります。HTTP URL のポートのみが書き換えられます。サーバーからの HTTPS URL はそのままクライアントに送信されます。

仮想サーバーはポート *80* で構成されます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

仮想サーバーはポート *8080* で構成されます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com:8080/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
https://domain.com:445/	https://domain.com:445/

シナリオ 4: 仮想サーバーとサービスのタイプは SSL です。ポート書き換えが有効な場合、HTTPS URL のポートのみが書き換えられます。サーバーからの HTTP URL はそのままクライアントに送信されます。

SSL リダイレクトは仮想サーバーで有効になっています。仮想サーバーはどのポートでも構成できます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL リダイレクトとポート書き換えは仮想サーバーで有効になっています。仮想サーバーはポート *443* で構成されます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com/

SSL リダイレクトとポート書き換えは仮想サーバーで有効になっています。仮想サーバーはポート *444* で構成されます。次の表を参照してください。

サーバーからの URL をリダイレクト	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com:444/
https://domain.com:445/	https://domain.com:444/

CLI を使用して仮想サーバーで **HTTP** リダイレクトを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -redirectPortRewrite (ENABLED | DISABLED)
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -redirectPortRewrite enabled
<!--NeedCopy-->
```

GUI を使用して仮想サーバーで **HTTP** リダイレクトを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、[詳細設定] ペインで [トラフィック設定] をクリックし、[書き換え] を選択します。

CLI を使用して **SSL** 仮想サーバーまたはサービスに **SSL** リダイレクトを設定するには

コマンドプロンプトで入力します。

```
set ssl vserver <vServerName> - sslRedirect (ENABLED | DISABLED)

set ssl service <serviceName> - sslRedirect (ENABLED | DISABLED)
<!--NeedCopy-->
```

例**:**

```
set ssl vserver Vserver-SSL-1 -sslRedirect enabled

set ssl service service-SSL-1 -sslRedirect enabled
<!--NeedCopy-->
```

GUI を使用して **SSL** 仮想サーバーまたはサービスで **SSL** リダイレクトと **SSL** ポート書き換えを構成するには

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [SSL パラメータ] をクリックし、[SSL リダイレクト] を選択します。

要求ヘッダーに仮想サーバーの **IP** アドレスとポートを挿入する

August 15, 2023

同じサービス上の異なるアプリケーションと通信する複数の仮想サーバーがある場合は、次の操作を実行する必要があります。

NetScaler ADC アプライアンスを構成して、そのサービスに送信される HTTP リクエストに適切な仮想サーバーの IP アドレスとポート番号を追加します。この設定により、サービスで実行されているアプリケーションが、要求を送信した仮想サーバーを識別できるようになります。

プライマリ仮想サーバーがダウンしていてバックアップ仮想サーバーが稼働している場合、バックアップ仮想サーバーの構成設定がクライアント要求に追加されます。要求がプライマリ仮想サーバーからのものかバックアップ仮想サーバーからのものかに関係なく、同じヘッダータグを追加する場合は、両方の仮想サーバーで必要なヘッダータグを構成する必要があります。

注: このオプションは、ワイルドカード仮想サーバーまたはダミー仮想サーバーではサポートされていません。

CLI を使用して仮想サーバーの IP アドレスとポートをクライアントリクエストに挿入するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -insertVserverIPPort <insertVserverIPPort> [<
    vipHeader>]
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -insertVserverIPPort VipAddr
<!--NeedCopy-->
```

GUI を使用して仮想サーバーの IP アドレスとポートをクライアントリクエストに挿入するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、[詳細設定] ウィンドウで、[トラフィックの設定] をクリックし、[仮想サーバー IP ポートの挿入] を選択し、仮想サーバーの IP ポートヘッダーを指定します。

バックエンド通信に指定したソース IP を使用する

February 15, 2024

物理サーバーまたは他のピアデバイスとの通信では、NetScaler ADC アプライアンスはソース IP アドレスとして所有する IP アドレスを使用します。NetScaler ADC アプライアンスは IP アドレスのプールを維持し、サーバーとの接続中に IP アドレスを動的に選択します。物理サーバが配置されているサブネットに応じて、アプライアンスは使用する IP アドレスを決定します。このアドレスプールは、トラフィックおよびモニタプローブの送信に使用されます。

多くの場合、アプライアンスで、バックエンド通信に特定の IP アドレスまたは特定の IP アドレスのセットからの任意の IP アドレスを使用したい場合があります。次に、いくつかの例を示します。

- モニタプローブに使用される送信元 IP アドレスが特定のセットに属している場合、サーバはモニタプローブとトラフィックを区別できます。
- サーバーのセキュリティを向上させるために、特定の一連の IP アドレスからの要求や、場合によっては単一の特定の IP アドレスからの要求に応答するようにサーバーを構成できます。この場合、アプライアンスは、サーバによって受け入れられた IP アドレスのみを送信元 IP アドレスとして使用できます。
- アプライアンスは、IP アドレスを IP セットに分散し、セットのアドレスを特定のサービスへの接続にのみ使用できれば、内部接続を効率的に管理できます。

指定した送信元 IP アドレスを使用するようにアプライアンスを設定するには、ネットプロファイル（ネットワークプロファイル）を作成し、そのプロファイルを使用するようにアプライアンスエンティティを構成します。ネットプロファイルは、負荷分散またはコンテンツスイッチング仮想サーバー、NetScaler Gateway VPN 仮想サーバー、サービス、サービスグループ、またはモニターにバインドできます。ネットプロファイルには、NetScaler ADC が所有する IP アドレス（SNIP および VIP）があり、送信元 IP アドレスとして使用できます。単一の IP アドレスでも、IP セットと呼ばれる IP アドレスのセットでもかまいません。ネットプロファイルに IP セットがある場合、アプライアンスは接続時に IP セットから IP アドレスを動的に選択します。プロファイルに 1 つの IP アドレスがある場合、同じ IP アドレスが送信元 IP として使用されます。

ネットプロファイルが負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにバインドされている場合、そのプロファイルは、バインドされているすべてのサービスにトラフィックを送信するために使用されます。ネットプロファイルがサービスグループにバインドされている場合、アプライアンスはサービスグループのすべてのメンバーに対してそのプロファイルを使用します。ネットプロファイルがモニターにバインドされている場合、アプライアンスはモニターから送信されたすべてのプローブに対してプロファイルを使用します。

> 注:
>
> - NetScaler ADC アプライアンスが VIP アドレスを使用してサーバーと通信する場合、セッションエントリを使用して、VIP アドレス宛てのトラフィックがサーバーからの応答であるか、クライアントからの要求であるかを識別します。
> - ネットプロファイルを NetScaler Gateway VPN 仮想サーバーにバインドできます。ただし、ネットプロファイルをバインドするときは、いくつかの点に注意する必要があります。詳細については、「ネットプロファイルを VPN 仮想サーバーにバインドするときの注意点」を参照してください。
> - サービスまたはサービスグループにバインドされたネットプロファイル IP は、対応するバックエンドサーバーへのトラフィックの送信だけでなく、未解決のバックエンド FQDN によってトリガーされる DNS 要求にも使用されます。

トラフィック送信のためのネットプロファイルの使用

[Use Source IP Address (USIP)] オプションが有効になっている場合、アプライアンスはクライアントの IP アド

レスを使用し、すべてのネットプロファイルを無視します。USIP オプションが有効になっていない場合、アプライアンスは次の方法でソース IP を選択します。

- 仮想サーバまたはサービス/サービスグループにネットプロファイルがない場合、アプライアンスはデフォルトの方式を使用します。
- サービス/サービスグループにのみネットプロファイルがある場合、アプライアンスはそのネットプロファイルを使用します。
- 仮想サーバ上にのみネットプロファイルがある場合、アプライアンスはそのネットプロファイルを使用します。
- 仮想サーバとサービス/サービスグループの両方にネットプロファイルがある場合、アプライアンスはサービス/サービスグループにバインドされたネットプロファイルを使用します。

モニタプローブの送信にネットプロファイルを使用します。

モニタプローブの場合、アプライアンスは次の方法でソース IP を選択します。

- モニターにバインドされたネットプロファイルがある場合、アプライアンスはモニターのネットプロファイルを使用します。仮想サーバまたはサービス/サービスグループにバインドされたネットプロファイルは無視されます。
- モニターにバインドされたネットプロファイルがない場合は、
  - サービス/サービスグループにネットプロファイルがある場合、アプライアンスはサービス/サービスグループのネットプロファイルを使用します。
  - サービス/サービスグループにもネットプロファイルがない場合、アプライアンスはデフォルトの方法でソース IP を選択します。

注: サービスにバインドされたネットプロファイルがない場合、サービスがサービスグループにバインドされている場合、アプライアンスはサービスグループ上のネットプロファイルを検索します。

指定した送信元 IP アドレスを通信に使用するには、次の手順に従います。

1. NetScaler ADC アプライアンスが所有する SNiP および VIP のプールから IP セットを作成します。IP セットは、SNIP アドレスと VIP アドレスの両方で構成できます。手順については、IP セットの作成を参照してください。
2. ネットプロファイルを作成します。手順については、ネットプロファイルの作成を参照してください。
3. ネットプロファイルをアプライアンスのエンティティにバインドします。手順については、NetScaler ADC エンティティへのネットプロファイルのバインドを参照してください。

> 注:
>
> - ネットプロファイルは、NetScaler ADC アプライアンスで SNIP および VIP として指定された IP アドレスのみを持つことができます。
> - NetScaler が開始したパケットでは、送信元 IP パーシステンスは適用されません。

ネットプロファイルの管理

ネットプロファイル（ネットワークプロファイル）には、1 つの IP アドレスまたは IP セットが含まれます。物理サーバーまたはピアとの通信中、NetScaler ADC アプライアンスは、プロファイルで指定されたアドレスを送信元 IP アドレスとして使用します。

- ネットワークプロファイルの作成手順については、ネットワークプロファイルの作成を参照してください。
- ネットワークプロファイルを NetScaler ADC エンティティにバインドする手順については、NetScaler ADC エンティティへのネットプロファイルのバインドを参照してください。

IP セットを作成する

IP セットは、NetScaler ADC アプライアンスでサブネット IP アドレス（SNIP）または仮想 IP アドレス（VIP）として構成される IP アドレスのセットです。IP セットには、そのセットに含まれる IP アドレスの用途を識別するためのわかりやすい名前を付けます。IP セットを作成するには、IP アドレスセットを追加し、NetScaler ADC 所有の IP アドレスをバインドします。SNIP アドレスと VIP アドレスは、同じ IP セット内に存在できます。

CLI を使用して IP セットを作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add ipset <name>

bind ipset <name> <IPAddress>
<!--NeedCopy-->
```

または

```
bind ipset <name> <IPAddress>

show ipset [<name>]
<!--NeedCopy-->
```

上記のコマンドは、名前を渡さない場合に、アプライアンス上のすべての IP セットの名前を表示します。名前を渡すと、指定した IP セットにバインドされた IP アドレスが表示されます。

例

```
1.
> add ipset skpnwipset
 Done
> bind ipset skpnwipset 21.21.20.1
 Done

2.
 > add ipset testnwipset
 Done
```



```
> bind ipset testnwipset 21.21.21.[21-25]
 IPAddress "21.21.21.21" bound
 IPAddress "21.21.21.22" bound
 IPAddress "21.21.21.23" bound
 IPAddress "21.21.21.24" bound
 IPAddress "21.21.21.25" bound
 Done

3.
 > bind ipset skpipset 11.11.11.101
 ERROR: Invalid IP address
[This IP address could not be added because this is not an IP address
   owned by the NetScaler appliance]
 > add ns ip 11.11.11.101 255.255.255.0 -type SNIP
 ip "11.11.11.101" added
 Done
 > bind ipset skpipset 11.11.11.101
 IPAddress "11.11.11.101" bound
 Done
4.
> sh ipset
1) Name: ipset-1
2) Name: ipset-2
3) Name: ipset-3
4) Name: skpnewipset
 Done

5.
> sh ipset skpnewipset
IP:21.21.21.21
IP:21.21.21.22
IP:21.21.21.23
IP:21.21.21.24
IP:21.21.21.25
 Done
<!--NeedCopy-->
```

GUI を使用して **IP** セットを作成するには

[システム] **>** [ネットワーク] **>** [**IP** セット] に移動し、IP セットを作成します。

ネットプロファイルの作成

ネットプロファイル（ネットワークプロファイル）は、Citrix
ADC アプライアンスの 1 つ以上の SNIP アドレスまたは VIP アドレスで構成されます。

CLI を使用してネットプロファイルを作成するには

コマンドプロンプトで入力します。

```
add netprofile <name> [-srcIp <srcIpVal>]
<!--NeedCopy-->
```

このコマンドで srcipVal が指定されていない場合は、後で`set netprofile`コマンドを使用して指定することができます。

例

```
add netprofile skpnetprofile1 -srcIp 21.21.20.1
Done

add netprofile baksnp -srcIp bakipset
Done

set netprofile yahnp -srcIp 12.12.23.1
Done

set netprofile citkbnp -srcIp citkbipset
Done
<!--NeedCopy-->
```

ネットプロファイルを **NetScaler ADC** エンティティにバインドする

ネットプロファイルは、負荷分散仮想サーバー、サービス、サービスグループ、またはモニターにバインドできます。

注：NetScaler ADC エンティティの作成時にネットプロファイルをバインドすることも、既存のエンティティにバインドすることもできます。

コマンドラインインターフェイスを使用してネットプロファイルをサーバーにバインドするには

ネットプロファイルは、負荷分散仮想サーバーとコンテンツスイッチング仮想サーバーにバインドできます。適切な仮想サーバを指定します。

コマンドプロンプトで入力します。

```
set lb vserver <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

または

```
set cs vserver <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```


例

```
set lb vserver skpnwvs1 -netProfile gntnp
 Done
set cs vserver mmdcsv -netProfile mmdnp
 Done
<!--NeedCopy-->
```

GUI を使用してネットプロファイルを仮想サーバーにバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [プロファイル] をクリックし、ネットプロファイルを設定します。

CLI を使用してネットプロファイルをサービスにバインドするには

コマンドプロンプトで入力します。

```
set service <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

例

```
set service brnssvc1 -netProfile brnsnp
 Done
<!--NeedCopy-->
```

GUI を使用してネットプロファイルをサービスにバインドするには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で [プロファイル] をクリックし、ネットプロファイルを設定します。

CLI を使用してネットプロファイルをサービスグループにバインドするには

コマンドプロンプトで入力します。

```
set servicegroup <serviceGroupName> -netProfile <net_profile_name>
<!--NeedCopy-->
```

例

```
set servicegroup ndhsvcgrp -netProfile ndhnp
 Done
<!--NeedCopy-->
```


GUI を使用してネットプロファイルをサービスグループにバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [サービスグループ] に移動し、サービスグループを開きます。
2. [詳細設定] で [プロファイル] をクリックし、ネットプロファイルを設定します。

CLI を使用してネットプロファイルをモニターにバインドするには

コマンドプロンプトで入力します。

```
set monitor <monitor_name> -netProfile <net_profile_name>
```

例

```
set monitor brnsecvmon1 -netProfile brnsmonnp
 Done
<!--NeedCopy-->
```

GUI を使用してネットプロファイルをモニターにバインドするには

1. **Traffic Management > Load Balancing > Monitors** に移動します。
2. モニターを開き、ネットプロファイルを設定します。

アイドル状態のクライアント接続のタイムアウト値を設定する

August 15, 2023

設定したタイムアウト期間 (秒単位) が経過すると、アイドル状態のクライアント接続をすべて終了するように仮想サーバーを構成できます。この設定を構成すると、NetScaler ADC アプライアンスは指定した時間待機し、その時間以降にクライアントがアイドル状態になると、クライアント接続を閉じます。デフォルトでは、クライアントのアイドルタイムアウト値は 180 秒に設定されています。

CLI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -cltTimeout <Value>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -cltTimeout 100
<!--NeedCopy-->
```


GUI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、 [トラフィック設定] をクリックし、クライアントのアイドルタイムアウト値を秒単位で設定します。

RTSP 接続を管理する

August 15, 2023

NetScaler アプライアンスは、NAT-On モードと NAT-Off モードの 2 つのトポロジーのいずれかを使用して RTSP サーバーの負荷分散を行うことができます。NAT-on モードでは、ネットワークアドレス変換（NAT）が有効化され、アプライアンス上で設定されます。RTSP 要求と応答は両方ともアプライアンスを通過します。したがって、ネットワークアドレス変換（NAT）を実行してデータ接続を識別するようにアプライアンスを設定する必要があります。

NAT の有効化と設定の詳細については、IP アドレッシングを参照してください。

NAT オフモードでは、NAT は有効になっておらず、設定されていません。アプライアンスはクライアントから RTSP 要求を受信し、設定された負荷分散方式を使用して選択したサービスにルーティングします。負荷分散された RTSP サーバーは、アプライアンスをバイパスして応答をクライアントに直接送信します。したがって、ダイレクトサーバーリターン (DSR) モードを使用するようにアプライアンスを設定し、DNS でパブリックにアクセス可能な FQDN をロードバランシングされた RTSP サーバーに割り当てる必要があります。

DSR モードの有効化と構成の詳細については、「ダイレクトサーバーリターンモードでの負荷分散の構成」を参照してください。DNS の構成の詳細については、「ドメインネームシステム」を参照してください。いずれの場合も、RTSP ロードバランシングを設定する場合は、ロードバランシング設定のトポロジと一致するように rtspNat も設定する必要があります。

CLI を使用して RTSP NAT を設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name>  -RTSPNAT <ValueOfRTSPNAT>
<!--NeedCopy-->
```

例:

```
set lb vserver vserver-LB-1  -RTSPNAT ON
<!--NeedCopy-->
```


GUI を使用して **RTSP NAT** を設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、RTSP タイプの仮想サーバーを開きます。
2. [詳細設定] で [トラフィック設定] をクリックし、[**RTSP Natting**] を選択します。

トラフィックレートに基づいてクライアントトラフィックを管理する

August 15, 2023

負荷分散仮想サーバーを通過するトラフィックのレートを監視し、トラフィックレートに基づいて NetScaler ADC アプライアンスの動作を制御できます。次に例を示します:

- トラフィックフローが高すぎる場合は、トラフィックフローをスロットリングします。
- トラフィックレートに基づいて情報をキャッシュします。
- トラフィックレートが高すぎる場合は、余分なトラフィックを別の負荷分散仮想サーバーにリダイレクトします。
- HTTP およびドメインネームシステム (DNS) 要求にレートベースのモニタリングを適用します。

レートベースのポリシーの詳細については、レート制限を参照してください。

レイヤー **2** パラメーターとの接続を特定する

August 15, 2023

一般に、NetScaler ADC アプライアンスは、接続を識別するために、クライアント IP アドレス、クライアントポート、宛先 IP アドレス、宛先ポートの 4 タプルを使用します。[L2 接続] オプションを有効にすると、通常の 4 タプルに加えて、接続のレイヤ 2 パラメータ（チャネル番号、MAC アドレス、VLAN ID）が使用されます。

負荷分散仮想サーバーの L2Conn パラメーターを有効にすると、同じ 4 タプル（<source IP>:::<source port><destination IP>:<destination port>）の複数の TCP 接続と非 TCP 接続を NetScaler アプライアンス上で共存させることができます。アプライアンスは、4 タプルとレイヤー 2 の両方のパラメーターを使用して、TCP 接続と非 TCP 接続を識別します。

L2Conn オプションは次のシナリオで有効にできます。

- NetScaler アプライアンスには複数の VLAN が構成され、VLAN ごとにファイアウォールが設定されます。
- ある VLAN のサーバから発信され、別の VLAN の仮想サーバに向かうトラフィックが、両方の VLAN に設定されたファイアウォールを通過する必要があります。

そのため、1 つ以上の負荷分散仮想サーバーに L2Conn パラメーターが設定されている nCore NetScaler アプライアンスをクラシックビルドまたは L2Conn パラメーターをサポートしない nCore ビルドにダウングレードすると、L2Conn パラメーターを使用する負荷分散構成は無効になります。

CLI を使用して L2 接続オプションを設定するには

コマンドプロンプトで入力します。

```
add lb vserver <name> <serviceType> <IPAddress>@ <port> -l2Conn ON
<!--NeedCopy-->
```

例

```
 add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -l2Conn ON
<!--NeedCopy-->
```

GUI を使用して L2 接続オプションを設定するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[レイヤ 2 パラメータ] を選択します。

[ダイレクトルートの優先] オプションを構成する

August 15, 2023

ワイルドカード負荷分散仮想サーバーでは、宛先へのルートを明示的に構成した場合、デフォルトでは、NetScaler ADC アプライアンスは設定されたルートに従ってトラフィックを転送します。設定したルートをアプライアンスに検索させたくない場合は、「ダイレクトルート優先」オプションを「いいえ」に設定できます。

デバイスが NetScaler アプライアンスに直接接続されている場合、アプライアンスはトラフィックをデバイスに直接転送します。たとえば、パケットの宛先がファイアウォールの場合、パケットを別のファイアウォール経由でルーティングする必要はありません。ただし、デバイスが直接接続されていても、トラフィックがファイアウォールを通過したい場合があります。このような場合は、[直接ルートを優先] オプションを [いいえ] に設定できます。

注：preferDirectRoute 設定は、NetScaler ADC アプライアンス上のすべてのワイルドカード仮想サーバーに適用されます。

CLI を使用して **[**ダイレクトルートの優先**]** オプションを設定するには

コマンドプロンプトで入力します。

```
set lb parameter -preferDirectRoute (YES | NO)
<!--NeedCopy-->
```

例**:**

```
set lb parameter -preferDirectRoute YES
<!--NeedCopy-->
```

GUI を使用して **[**ダイレクトルートを優先**]** オプションを設定するには

1. [トラフィック管理**] > [**負荷分散**] > [**ロードバランシングパラメータの設定] に移動します。
2. [直接ルートを優先] を選択します。

バックエンド通信に指定したポート範囲の送信元ポートを使用する

August 15, 2023

デフォルトでは、USIP オプションが無効になっているか、USIP でプロキシポートオプションを使用する構成の場合、NetScaler ADC アプライアンスはランダムなソースポート（1024 以上）からサーバーと通信します。

アプライアンスは、指定されたポート範囲の送信元ポートを使用して、サーバーとの通信をサポートします。この機能の使用例の 1 つは、ロギングおよびモニタリングを目的として、送信元ポートに基づいて特定のセットに属する受信トラフィックを識別するように設定されたサーバを対象としています。たとえば、ロギング目的で内部および外部トラフィックを識別します。

サーバーとの通信にポート範囲のソースポートを使用するように NetScaler アプライアンスを構成するには、次のタスクで構成されます。

- ネットプロファイルを作成し、送信元ポート範囲パラメータを設定します。送信元ポート範囲パラメータは、1 つ以上のポート範囲を指定します。アプライアンスは、指定されたポート範囲から空きポートの 1 つをランダムに選択し、それをサーバーへの各接続のソースポートとして使用します。
- ネットプロファイルをロードバランシング仮想サーバ、サービス、またはサービスグループにバインドする: ソースポート範囲が設定されたネットプロファイルを、ロードバランシング構成の仮想サーバ、サービス、またはサービスグループにバインドできます。仮想サーバーへの接続の場合、アプライアンスはネットプロファイルの指定されたポート範囲から空きポートの 1 つをランダムに選択し、このポートをバインドされたサーバーの 1 つに接続するための送信元ポートとして使用します。

CLI を使用して送信元ポート範囲を指定するには

コマンドプロンプトで入力します。

```
bind netProfile <name> (-srcPortRange <int[-int]> ...)

show netprofile <name>
<!--NeedCopy-->
```

GUI を使用して **1** つまたは複数の送信元ポートの範囲を指定するには

1. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
2. NetProfiles を追加または変更するときに、[ソースポート範囲] パラメータを設定します。

構成例

次の構成例では、ネットプロファイルの PARTIAL-NAT-1 には部分的な NAT 設定があり、ANY タイプの負荷分散仮想サーバー LBVS-1 にバインドされています。LBVS-1 で 192.0.0.0/8 から受信したパケットの場合、NetScaler アプライアンスはパケットの送信元 IP アドレスの最後のオクテットを 100 に変換します。たとえば、送信元 IP アドレスが 192.0.2.30 のパケットが LBVS-1 で受信された場合、NetScaler アプライアンスは、送信元 IP アドレスを 100.0.2.30 に変換してから、バインドされたサーバーの 1 つを送信します。

```
```
> add netprofile CUSTOM-SRCPORT-NP-1
Done
> bind netprofile CUSTOM-SRCPRT-NP-1 - srcportrange 2000-3000

Done
> bind netprofile CUSTOM-SRCPRT-NP-1 - srcportrange 5000-6000

Done
> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1

Done
<!--NeedCopy--> ```
```

## バックエンド通信の送信元 **IP** パーシステンシーを構成する

August 15, 2023

デフォルトでは、USIP オプションを無効にし、ネットプロファイルを仮想サーバーまたはサービスグループにバインドした負荷分散構成の場合、NetScaler ADC アプライアンスはラウンドロビンアルゴリズムを使用して、サーバ

 
```

ーと通信するためのネットプロファイルから IP アドレスを選択します。この選択方法により、選択される IP アドレスは、特定のクライアントのセッションによって異なる場合があります。

場合によっては、NetScaler ADC アプライアンスがトラフィックをサーバーに送信するときに、特定のクライアントのトラフィックをすべて同じ IP アドレスからルーティングする必要があります。たとえば、サーバは、ロギングおよびモニタリングを目的として、特定のセットに属するトラフィックを識別できます。

ネットプロファイルのソース IP 永続性オプションを使用すると、NetScaler アプライアンスはネットプロファイルで指定されている同じアドレスを使用して、特定のクライアントから仮想サーバーに対して開始されたすべてのセッションについてサーバーと通信できます。

CLI を使用してネットプロファイルのソース IP パーシステンシーを有効にするには

ネットプロファイルの追加時にソース IP パーシステンシスを有効にするには、コマンドプロンプトで次のように入力します。

```
add netProfile <name> -srcippersistency ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

既存のネットプロファイルでソース IP パーシステンシーを有効にするには、コマンドプロンプトで次のように入力します。

```
set netProfile <name> -srcippersistency ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

GUI を使用してネットプロファイルのソース IP パーシステンシーを有効にするには

1. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
2. ネットプロファイルを追加または変更する際は、「ソース IP パーシステンシー」を選択します。

例

次の設定例では、ネットプロファイル NETPROFILE-IPPRSTNCY-1 が送信元 IP 永続性オプションが有効になっており、ロードバランシング仮想サーバー LBVS-1 にバインドされています。

NetScaler ADC アプライアンスは、特定のクライアントから仮想サーバーへのすべてのセッションについて、LBVS-1 にバインドされたサーバーと通信するために、常に同じ IP アドレス（この例では 192.0.2.11）を使用します。


```
```
> add ipset IPSET-1

Done
> bind ipset IPSET-1 192.0.2.[11-15]
 IPAddress "192.0.2.11" bound
 IPAddress "192.0.2.12" bound
 IPAddress "192.0.2.13" bound
 IPAddress "192.0.2.14" bound
 IPAddress "192.0.2.15" bound
 Done
> add netprofile NETPROFILE-IPPRSTNCY-1 -srcIp IPSET-1 -
   srcippersistency ENABLED

Done
> set lb vserver LBVS-1 -netprofile NETPROFILE-IPPRSTNCY-1

 Done
<!--NeedCopy--> ```
```

## ロードバランシングセットアップのサーバー側で **IPv6** リンクローカルアドレスを使用する

August 15, 2023

IPv6 リンクローカルアドレスは、ロードバランシング設定のサービス、サービスグループ、およびサーバでサポートされます。サービス、サービスグループ、およびサーバ設定で、関連する VLAN ID とともにリンクローカル IPv6 アドレスを指定できます。NetScaler ADC アプライアンスは、サービス、サービスグループ、およびサーバー構成で指定された同じ VLAN のリンクローカル SNIP6 アドレスを使用して、それらと通信します。

リンクローカル IPv6 アドレスおよび関連する VLAN ID は、サービス、サービスグループ、およびサーバの設定で次の形式で指定されます。<IPv6_Addrs>%<vlan_id>

たとえば、fe80:123:4567::a%2048:, fe80:123:4567::a はリンクローカルアドレスで、2048 は VLAN ID です。

```
> add service SERVICE-1 fe80:123:4567::a%2048 HTTP 80

Done
> bind servicegroup SERVICE-GROUP-1 fe80::1%24 80

Done
> add server SERVER-1 fe80:b:c:d::e:f:a/64%1028

Done
```


```

# 高度な負荷分散設定

August 15, 2023

仮想サーバーの構成に加えて、サービスの詳細設定を構成できます。

負荷分散の詳細設定を行うには、以下のセクションを参照してください。



## 仮想サーバレベルの低速スタートにより、新しいサービスの負荷を段階的にステップアップ

August 15, 2023

NetScaler ADC アプライアンスは、サービスが負荷分散構成に追加された直後、または DOWN から UP への状態が変化した直後に、サービスの負荷（サービスが毎秒受信する要求の数）を徐々に増加するように構成できます（このドキュメントでは、「新しいサービス」という用語は、どちらの状況にも使用されます）。選択した負荷値と間隔（手動スロースタート）で手動で負荷を増やしたり、サービスが設定内の他のサービスと同じ数の要求を受信するまで、指定した間隔（自動スロースタート）で負荷を増やしたりするようにアプライアンスを設定できます。新しいサービスの立ち上げ期間中、アプライアンスは設定された負荷分散方式を使用します。

この機能はグローバルには利用できません。仮想サーバーごとに構成する必要があります。この機能は、次の負荷分散方法のいずれかを使用する仮想サーバーでのみ使用できます。

- ラウンドロビン
- 最小接続数
- 最短の応答時間
- 最小帯域幅
- 最小パケット数
- LRTM (最小応答時間法)
- カスタムロード

この機能を使用するには、次のパラメータを設定する必要があります。

- 新しいサービスリクエストレートとは、レートが上がるたびに新しいサービスに送信されるリクエストの数またはパーセンテージを増やす量です。つまり、1 秒あたりのリクエスト数か、その時点で既存のサービスが負担する負荷のパーセンテージのいずれかで増分のサイズを指定します。この値を 0 (ゼロ) に設定すると、新しいサービスではスロースタートは実行されません。

  注: 自動スロースタートモードでは、指定した値が他のサービスよりも新しいサービスに負荷がかかる場合、最終的な増分は指定された値より小さくなります。

- 増分間隔（秒単位）。この値を 0 (ゼロ) に設定すると、負荷は自動的に増加しません。手動でインクリメントする必要があります。

自動スロースタートでは、次のいずれかの条件が適用されると、サービスはスロースタートフェーズから取り出されます。

- 実際の要求レートが、新しいサービス要求レートよりも低くなっています。
- このサービスは、3 回連続して増加するトラフィックを受信しません。
- リクエスト率が 200 倍に増加しました。
- 新しいサービスが受信しなければならないトラフィックの割合は 100 以上です。

手動スロースタートでは、サービスはそのフェーズを終了するまでスロースタートフェーズのままになります。

### 手動スロースタート

新しいサービスの負荷を手動で増やす場合は、負荷分散仮想サーバーの増分間隔を指定しないでください。新しいサービスリクエスト率と単位のみを指定してください。間隔が指定されていない場合、アプライアンスは定期的に負荷

を増加させません。いずれかのパラメータを手動で変更しない限り、新しいサービスの負荷は、新しいサービスリクエスト率と単位の組み合わせで指定された値に維持されます。たとえば、新しいサービスリクエスト率とユニットパラメータをそれぞれ 25 と「1 秒あたり」に設定した場合、いずれかのパラメータを変更するまで、アプライアンスは新しいサービスの負荷を 1 秒あたり 25 リクエストで維持します。新しいサービスがスロースタートモードを終了し、既存のサービスと同じ数のリクエストを受信するようにするには、新しいサービスリクエスト率パラメータを 0 に設定します。

例として、仮想サーバーを使用して Service1 と Service2 の 2 つのサービスをラウンドロビンモードでロードバランシングしているとします。さらに、仮想サーバーが 1 秒あたり 240 のリクエストを受信し、負荷をサービス全体に均等に分散していると仮定します。Service3 という新しいサービスが構成に追加されたら、負荷のすべてをそのサービスに送信する前に、1 秒あたり 10、20、40 リクエストという値で手動で負荷を増やしたくなる場合があります。次の表は、3 つのパラメータを設定する値を示しています。

表 1. パラメータ値

| パラメーター | 値 |
|---|---|
| 間隔 (秒単位) | 0 |
| 新規サービスリクエスト率 | 10、20、40、0 (任意の間隔で) |
| 新しいサービスリクエストレートの単位 | 1 秒あたりのリクエスト数 |

新しいサービスリクエスト率パラメータを 0 に設定すると、Service3 は新しいサービスとは見なされなくなり、負荷の全部分を受けます。

Service3 の立ち上げ期間中に、Service4 という別のサービスを追加すると仮定します。この例では、新しいサービスリクエスト率パラメータが 40 に設定されたときに Service4 が追加されます。そのため、Service4 は 1 秒あたり 40 件のリクエストの受信を開始します。

次の表は、この例で説明されている期間におけるサービスの負荷分散を示しています。

表 2. 手動で負荷をステップアップする場合のサービスの負荷分散

| | 新規サービスリクエストレート = 10 リクエスト/秒 (サービス 3 追加) | 新規サービスリクエスト率 = 20 リクエスト/秒 | 新規サービスリクエストレート = 40 リクエスト/秒 (サービス 4 追加) | 新しいサービス要求レート = 0 req/秒 (新しいサービスはスロースタートモードを終了します) |
|---|---|---|---|---|
| **Service1** | 115 | 110 | 80 | 60 |
| **Service2** | 115 | 110 | 80 | 60 |
| **Service3** | 10 | 20 | 40 | 60 |
| **Service4** | - | - | 40 | 60 |

| | 新規サービスリクエストレート = 10 リクエスト/秒 (サービス 3 追加) | 新規サービスリクエスト率 = 20 リクエスト/秒 | 新規サービスリクエストレート = 40 リクエスト/秒 (サービス 4 追加) | 新しいサービス要求レート = 0 req/秒 (新しいサービスはスロースタートモードを終了します) |
|---|---|---|---|---|
| 合計要件/秒 (仮想サーバーの負荷) | 240 | 240 | 240 | 240 |

### 自動スロースタート

アプライアンスが新しいサービスの負荷を一定の間隔で自動的に増加させたい場合は、新しいサービスリクエストレートパラメータ、units パラメータ、および増分間隔を設定します。すべてのパラメータが 0 以外の値に設定されている場合、アプライアンスは、指定された間隔で、サービスが負荷の全シェアを受け取るまで、新しいサービスへの負荷を新しいサービスリクエストレートの値だけ増やします。

たとえば、サービス 1、サービス 2、サービス 3、サービス 4 の 4 つのサービスがロードバランシング仮想サーバー vserver1 にバインドされているとします。さらに、vserver1 が 1 秒あたり 100 のリクエストを受信し、その負荷をサービス全体に均等に分散すると仮定します (サービスあたり 1 秒あたり 25 リクエスト)。5 つ目のサービス Service5 を設定に追加すると、アプライアンスが新しいサービスに、最初の 10 秒は 1 秒あたり 4 リクエスト、次の 10 秒間は 1 秒あたり 8 リクエストというように送信し、1 秒あたり 20 リクエストを受信するようにしたい場合があります。この要件について、3 つのパラメータを設定する値を次の表に示します。

表 3. パラメータ値

| パラメーター | 値 |
|---|---|
| 間隔 (秒単位) | 10 |
| インクリメント値 | 4 |
| 新しいサービスリクエストレートの単位 | 1 秒あたりのリクエスト数 |

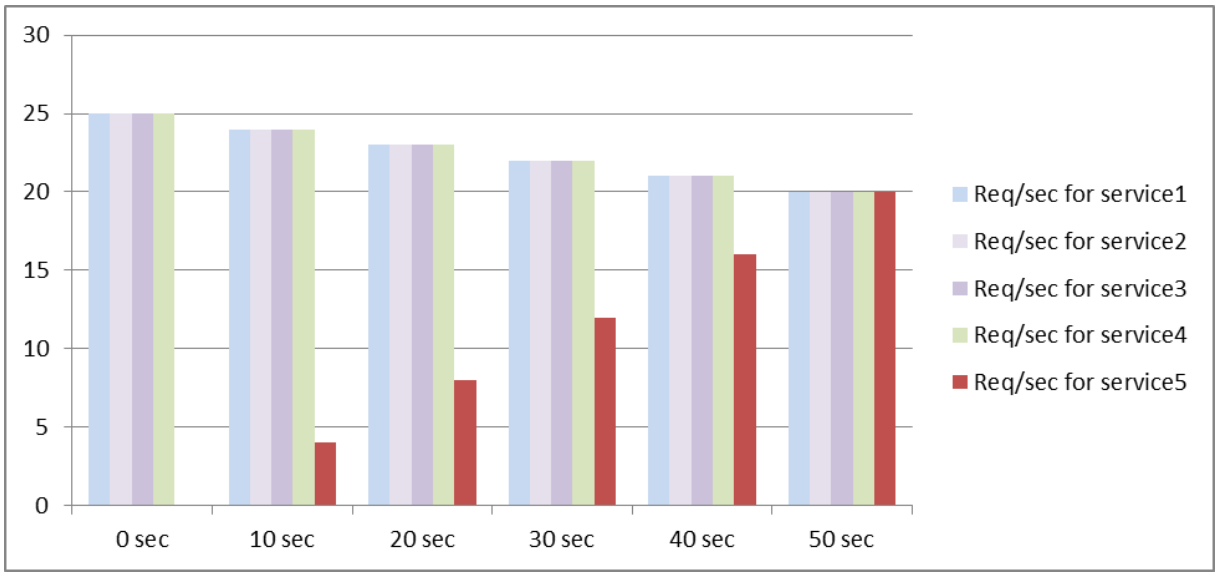
この構成では、新しいサービスは、追加されたか、状態が DOWN から UP に変更されてから 50 秒後に、既存のサービスと同じ数のリクエストの受信を開始します。この期間の各間隔の間に、アプライアンスは、段階的に増加しないと新しいサービスに送信されるはずの超過リクエストを既存のサーバーに分散します。たとえば、段階的な増加がない場合、Service5 を含む各サービスは 1 秒あたり 20 件のリクエストを受信することになります。段階的に増分すると、Service5 が 1 秒あたり 4 つの要求しか受信しない最初の 10 秒間に、アプライアンスは 1 秒あたり 16 個を超える要求を既存のサービスに分配します。その結果、50 秒間にわたって次の表と図に示す分散パターンになります。50 秒が経過すると、Service5 は新しいサービスとは見なされなくなり、通常のトラフィックシェアを受け取ります。

表 4. Service5 追加直後の 50 秒間の全サービスの負荷分散パターン

| | 0 秒 | 10 秒 | 20 秒 | 30 秒 | 40 秒 | 50 秒 |
|---|---|---|---|---|---|---|
| サービス **1** の要求/秒 | 25 | 24 | 23 | 22 | 21 | 20 |
| サービス **2** の要求/秒 | 25 | 24 | 23 | 22 | 21 | 20 |
| サービス **3** の要求/秒 | 25 | 24 | 23 | 22 | 21 | 20 |
| サービス **4** の要求/秒 | 25 | 24 | 23 | 22 | 21 | 20 |
| サービス **5** の要求/秒 | 0 | 4 | 8 | 12 | 16 | 20 |
| 合計要件/秒 **(**仮想サーバーの負荷**)** | 100 | 100 | 100 | 100 | 100 | 100 |

図 1: Service5 追加直後の 50 秒間の全サービス負荷分散パターンのグラフ




別の要件として、アプライアンスが既存のサービスに対する負荷の Service5 25% を、最初の 5 秒間に 50% を、次の 5 秒間に 50% を、1 秒あたり 20 のリクエストを受信するまで送信するということが挙げられます。この要件について、3 つのパラメータを設定する値を次の表に示します。

表 5. パラメータ値

| パラメーター | 値 |
| --- | --- |
| 間隔 (秒単位) | 5 |
| インクリメント値 | 25 |
| 新しいサービスリクエストレートの単位 | パーセント |

この構成では、サービスは追加されてから 20 秒後、または状態が DOWN から UP に変更されてから 20 秒後に、既存のサービスと同じ数のリクエストの受信を開始します。新サービスの立ち上げ期間中のトラフィック配分は、前述のステップ増加の単位が「1 秒あたりのリクエスト数」だった場合と同じです。

### スロースタートパラメーターの設定

スロースタートパラメータは、`set lb vserver`または`add lb vserver`コマンドを使用して設定します。次のコマンドは、仮想サーバーを追加するときにスロースタートパラメータを設定するためのものです。

コマンドラインインターフェイスを使用して新しいサービスの段階的なロードインクリメントを設定するには

コマンドプロンプトで次のコマンドを入力して、サービスの負荷を段階的に増やすように構成し、構成を確認します。

```
add lb vserver <name> <serviceType> <IPAddress> <port> [-
    newServiceRequest <positive_integer>] [<newServiceRequestUnit>] [-
    newServiceRequestIncrementInterval <positive_integer>]

show lb vserver <name>
<!--NeedCopy-->
```

例

```
set lb vserver BR_LB -newServiceRequest 5 PER_SECOND -
    newServiceRequestIncrementInterval 10
Done

show lb vserver BR_LB
BR_LB (192.0.2.33:80) - HTTP Type: ADDRESS
State: UP
...
...
New Service Startup Request Rate: 5 PER_SECOND, Increment Interval: 10
...
...
Done
<!--NeedCopy-->
```


構成ユーティリティを使用して新しいサービスの段階的なロードインクリメントを設定するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で [方法] を選択し、次のスロースタートパラメータを設定します。

    - 新規サービス起動リクエスト率
    - 新しいサービスリクエストユニット。
    - インクリメント間隔。

サービスの監視なしオプション

August 15, 2023

外部システムを使用してサービスのヘルスチェックを実行し、NetScaler アプライアンスにサービスのヘルスを監視させたくない場合は、サービスの監視なしオプションを設定できます。その場合、アプライアンスはサービスの状態を確認するためのプローブを送信せず、サービスを UP と表示します。サービスがダウンしても、アプライアンスは負荷分散方式で指定されたとおりにクライアントからサービスにトラフィックを送信し続けます。

no-monitor オプションを設定すると、モニターは ENABLED または DISABLED 状態になり、no-monitor オプションを削除すると、モニターの以前の状態が再開されます。

サービスの作成時に、サービスの監視なしオプションを設定できます。既存のサービスに no-monitor オプションを設定することもできます。

監視なしオプションを設定した場合の結果は次のとおりです。

- 監視なしオプションを有効にしたサービスが停止しても、アプライアンスは引き続きサービスを稼働中として表示し、トラフィックをサービスに転送し続けます。サービスへの永続的な接続は状況を悪化させる可能性があります。その場合、または UP と表示されている多くのサービスが実際に DOWN の場合、システムに障害が発生する可能性があります。このような状況を回避するには、サービスを監視する外部メカニズムがサービスを DOWN と報告するときに、NetScaler ADC 構成からサービスを削除します。
- サービスで no-monitor オプションを構成する場合は、ダイレクトサーバーリターン (DSR) モードでロードバランシングを構成できません。既存のサービスでは、no-monitor オプションを設定すると、そのサービスの DSR モードを設定できません。

CLI を使用して新しいサービスの監視なしオプションを設定するには

コマンドプロンプトで次のコマンドを入力して、ヘルスモニターオプションを使用してサービスを作成し、構成を確認します。

```
add service <serviceName> <IP | serverName> <serviceType> <port> -
   healthMonitor (YES|NO)
```



```
<!--NeedCopy-->
```

例:

```
add service nomonsrvc 10.102.21.21 http 80 -healthMonitor no
Done

show service nomonsrvc
nomonsrvc (10.102.21.21:80) - HTTP
State: UP
Last state change was at Mon Nov 15 22:41:29 2010
Time since last state change: 0 days, 00:00:00.970
Server Name: 10.102.21.21
 Server ID : 0 Monitor Threshold : 0
...
Access Down Service: NO
...
Down state flush: ENABLED
Health monitoring: OFF

1 bound monitor:
1) Monitor Name: tcp-default
State: UNKNOWN Weight: 1
Probes: 3 Failed [Total: 3 Current: 3]
Last response: Probe skipped - Health monitoring is turned off.
Response Time: N/A
Done
<!--NeedCopy-->
```

CLI を使用して既存のサービスの監視なしオプションを設定するには

コマンドプロンプトで次のコマンドを入力して、ヘルスモニターオプションを設定します。

```
set service <name> -healthMonitor (YES|NO)
<!--NeedCopy-->
```

例:

```
By default, the state of a service and the state of the corresponding
   monitor are UP.
 >show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP


1) Monitor Name: http-ecv
 State: UP Weight: 1
Probes: 99992 Failed [Total: 0 Current: 0]
Last response: Success - Pattern found in response.
Response Time: 3.76 millisec
Done
```



```
13
14 When the no-monitor option is set on a service, the state of the
       monitor changes to UNKNOWN.
15 set service LB-SVC1 -healthMonitor NO
16 Done
17
18 show service LB-SVC1
19 LB-SVC1 (10.102.29.5:80) - HTTP
20 State: UP
21 Last state change was at Fri Dec 10 10:17:37 2010.
22 Time since last state change: 5 days, 18:55:48.710
23 Health monitoring: OFF
24
25 1) Monitor Name: http-ecv
26    State: UNKNOWN Weight: 1
27          Probes: 100028 Failed [Total: 0 Current: 0]
28    Last response: Probe skipped - Health monitoring is turned off.
29      Response Time: 0.0 millisec
30   Done
31 When the no-monitor option is removed, the earlier state of the monitor
       is resumed.
32 > set service LB-SVC1 -healthMonitor YES
33 Done
34 >show service LB-SVC1
35 LB-SVC1 (10.102.29.5:80) - HTTP
36 State: UP
37 Last state change was at Fri Dec 10 10:17:37 2010
38 Time since last state change: 5 days, 18:57:47.880
39 1) Monitor Name: http-ecv
40    State: UP Weight: 1
41    Probes: 100029 Failed [Total: 0 Current: 0]
42    Last response: Success - Pattern found in response.
43    Response Time: 5.690 millisec
44    Done
45 <!--NeedCopy-->
```

GUI を使用してサービスの監視なしオプションを設定するには

1. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
2. サービスを開き、ヘルスモニタリングをクリアします。

保護されたサーバー上のアプリケーションをトラフィックの急増から保護する

August 15, 2023

NetScaler アプライアンスには、サーバーまたはキャッシュの容量を維持するためのサージ保護オプションがあります。アプライアンスは、サーバーへのクライアント要求のフローを制御し、サーバーに同時にアクセスできるクラ

イアントの数を制御します。アプライアンスはサーバーに渡されたサージをブロックし、サーバーの過負荷を防ぎます。

サージ保護を正しく機能させるには、サージ保護をグローバルに有効にする必要があります。サージ保護の詳細については、サージ保護を参照してください。

CLI を使用してサービスにサージ保護を設定するには

コマンドプロンプトで入力します。

```
set service <name> -sp <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -sp ON
<!--NeedCopy-->
```

GUI を使用してサービスにサージ保護を設定するには

1. [トラフィック管理**] > [**負荷分散**] > [**サービス] に移動し、ソースを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[サージ保護] を選択します。

仮想サーバーとサービス接続のクリーンアップを有効にする

August 15, 2023

仮想サーバーの状態は、それにバインドされているサービスの状態によって異なります。各サービスの状態は、そのサービスにバインドされたモニターから送信されたプローブまたはヘルスチェックに対する負荷分散サーバーの応答によって異なります。負荷分散されたサーバーが応答しないことがあります。サーバーが遅いかビジー状態の場合、モニタリングプローブがタイムアウトする可能性があります。設定されたタイムアウト期間内に繰り返しモニタリングプローブに応答しない場合、サービスには DOWN とマークされます。サービスまたは仮想サーバーが DOWN とマークされている場合は、サーバーとクライアント側の接続をフラッシュする必要があります。既存の接続を終了すると、リソースが解放され、場合によっては負荷分散セットアップの回復速度が速くなります。

特定の条件下では、サービスまたは仮想サーバーが DOWN とマークされたときに、既存の接続をただちに終了するように **downStateFlush** 設定を構成できます。トランザクションを完了する必要があるアプリケーションサーバーでは、DownStateFlush 設定を有効にしないでください。この設定は、DOWN とマークされたときに接続を安全に終了できる Web サーバー上で有効にできます。

次の表は、仮想サーバー VServer-LB-1 に Service-1 という 1 つのサービスをバインドした構成例に対するこの設定の影響をまとめたものです。表内の E と D は DownStateFlush 設定の状態を示しています。E は有効、D は無効を意味します。

Vserver-LB-1	Service-1	接続の状態
E	E	クライアント接続とサーバー接続の両方が終了します。
E	D	NetScaler アプライアンスが接続の再利用をサポートしていない TCP などの一部のサービスタイプでは、クライアント接続とサーバー接続の両方が終了します。アプライアンスが接続の再利用をサポートする HTTP などのサービスタイプでは、クライアント接続とサーバー接続の両方が、それらの接続でトランザクションがアクティブである場合にのみ終了されます。トランザクションがアクティブでない場合、クライアント接続のみが終了します。
D	E	NetScaler アプライアンスが接続の再利用をサポートしていない TCP などの一部のサービスタイプでは、クライアント接続とサーバー接続の両方が終了します。アプライアンスが接続の再利用をサポートする HTTP などのサービスタイプでは、クライアント接続とサーバー接続の両方が、それらの接続でトランザクションがアクティブである場合にのみ終了されます。トランザクションがアクティブでない場合、サーバー接続のみが終了します。
D	D	クライアント接続もサーバー接続も終了しません。

サーバーまたはクライアントによって確立されたすべての接続が閉じられた場合にのみサービスを無効にする場合は、正常なシャットダウンオプションを使用できます。サービスのグレースフルシャットダウンの詳細については、「サービスのグレースフルシャットダウン」を参照してください。

CLI を使用してサービスでダウン状態のフラッシュを設定するには

コマンドプロンプトで入力します。

```
set service <name>  -downStateFlush (ENABLED | DISABLED )
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -downStateFlush enabled
<!--NeedCopy-->
```

GUI を使用してサービスにダウンステートフラッシュを設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[ダウン状態フラッシュ] を選択します。

CLI を使用して仮想サーバーにダウンステートフラッシュを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name>  -downStateFlush (ENABLED | DISABLED )
<!--NeedCopy-->
```

例:

```
set lb vserver vsvr1 -downStateFlush enabled
<!--NeedCopy-->
```

GUI を使用して仮想サーバーにダウンステートフラッシュを設定するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[ダウン状態フラッシュ] を選択します。

サービスの正常なシャットダウン

August 15, 2023

システムのアップグレードやハードウェアのメンテナンスなど、ネットワークが定期的に停止している間は、一部のサービスを終了または無効にしなければならない場合があります。後で「enable service」<name> コマンドを使用してサービスを有効にできます。

確立されたセッションが中断されないようにするには、次のいずれかを実行してサービスをトランジションアウトオブサービス (TROFS) 状態にします。

- TROFS コードまたは文字列をモニターに追加する-モニタープローブに応答して特定のコードまたは文字列を送信するようにサーバーを設定します。
- サービスを明示的に無効にし、
  - 遅延を秒単位で設定します。
  - グレースフルシャットダウンを有効にします。

TROFS コードまたは文字列の追加

1 つのサービスにバインドするモニタが 1 つだけで、そのモニタが TROFS 対応の場合、モニタプローブに対するサーバの応答に基づいて、サービスを TROFS 状態に設定できます。この応答は、HTTP モニターの TrofsCode パラメーターの値または HTTP-ECV モニターまたは TCP-ECV モニターの TrofsString パラメーターの値と比較されます。コードが一致すると、サービスは TROFS 状態になります。この状態では、持続的な接続を引き続き尊重します。

複数のモニターがサービスにバインドされている場合、サービスの有効状態は、サービスにバインドされているすべてのモニターの状態に基づいて計算されます。TROFS 応答を受信すると、この計算のために TROFS 対応モニターの状態が UP と見なされます。NetScaler ADC アプライアンスがサービスを UP として指定する方法の詳細については、「サービスにバインドされたモニターのしきい値を設定する」を参照してください。

重要:

- 1 つのサービスに複数のモニターをバインドできますが、複数の TROFS を有効にすることはできません。
- TROFS 対応モニターを TROFS 対応でないモニターに変換することはできますが、その逆はできません。

コマンドラインインターフェイスを使用してモニターの **TROFS** コードまたは文字列を設定するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
add lb monitor <monitor-name> HTTP -trofsCode <respcode>

add lb monitor <monitor-name> HTTP-ECV  -trofsString <resp string>

add lb monitor <monitor-name> TCP-ECV  -trofsString <resp string>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **TROFS** コードまたは文字列を変更するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
set lb monitor <trofs monitorname> HTTP -trofscode <newcode>

```



```
set lb monitor <trofs monitorname> HTTP-ECV -trofsstring <new string>

set lb monitor <trofs monitorname> TCP-ECV -trofsstring <new string>
<!--NeedCopy-->
```

注:set コマンドを使用できるのは、TROFS 対応のモニターを以前に追加した場合に限られます。このコマンドを使用して、TROFS に対応していないモニターに TROFS コードまたは文字列を設定することはできません。

構成ユーティリティを使用してモニターの **TROFS** コードまたは文字列を構成するには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. モニターペインで「追加」をクリックし、次のいずれかを実行します。
   - 「タイプ」に「HTTP」を選択し、TROFS コードを指定します。
   - [タイプ] に [HTTP-ECV] または [TCP-ECV] を選択し、TROFS 文字列を指定します。

サービスを無効にする

ただし、多くの場合、サービスへのすべての接続が既存のトランザクションを完了するのにかかる時間を見積もることはできません。待機時間が過ぎてもトランザクションが完了していない場合、サービスをシャットダウンするとデータが失われる可能性があります。この場合、サービスのグレースフルシャットダウンを指定して、現在アクティブなクライアント接続がすべてサーバーまたはクライアントによって閉じられたときにのみサービスが無効になるようにします。グレースフルシャットダウンに加えて待機時間を指定した場合の動作については、次の表を参照してください。

グレースフルシャットダウンを有効にしても、指定した方法に従って持続性が維持されます。モニタによるチェックの結果として正常なシャットダウン状態の間にサービスが DOWN とマークされない限り、システムは、クライアントからの新しい接続を含む、すべての永続的なクライアントに対して引き続きサービスを提供します。

次の表では、グレースフルシャットダウンオプションについて説明します。

状態	結果
グレースフルシャットダウンが有効で、待機時間が指定されています。	待機時間が経過していなくても、現在のアクティブなクライアント接続のうち最後の接続が処理されると、サービスは停止されます。アプライアンスは接続の状態を 1 秒に 1 回チェックします。待機時間が終了すると、開いているセッションはすべて閉じられます。
グレースフルシャットダウンは無効で、待機時間が指定されます。	確立されたすべての接続が期限切れ前に処理された場合でも、待機時間が終了した後にのみサービスが停止されます。

状態	結果
グレースフルシャットダウンは有効で、待機時間は指定されていません。	サービスは、最後の接続の処理にかかった時間に関係なく、以前に確立された最後の接続が提供された後にのみシャットダウンされます。
グレースフルシャットダウンは無効で、待機時間は指定されていません。	正常にシャットダウンできない。disable オプションを選択するか、disable コマンドが発行された直後に、サービスがシャットダウンされます。(デフォルトの待機時間は 0 秒です)。

サービスまたは仮想サーバーが DOWN とマークされている場合に既存の接続を終了するには、[Down State Flush] オプションを使用します。詳細については、「仮想サーバー接続のクリーンアップの有効化」を参照してください。

コマンドラインインターフェイスを使用してサービスの正常なシャットダウンを構成するには

コマンドプロンプトで次のコマンドを入力してサービスを正常にシャットダウンし、構成を確認します。

```
disable service <name> [<delay>] [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```

例:

```
> disable service svc1 6000 -graceFul YES
Done
>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: GOING OUT OF SERVICE (Graceful, Out Of Service in 5998 seconds)
Last state change was at Mon Nov 15 22:44:15 2010
Time since last state change: 0 days, 00:00:01.160
...
Down state flush: ENABLED

1 bound monitor:
1) Monitor Name: tcp-default
State: UP              Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - live traffic to service.
Response Time: N/A
Done

>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: OUT OF SERVICE
Last state change was at Mon Nov 15 22:44:19 2010
Time since last state change: 0 days, 00:00:03.250
```



```
Down state flush: ENABLED

1 bound monitor:
1)  Monitor Name: tcp-default
State: UNKNOWN           Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - service state OFS.
Response Time: N/A
Done
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスのグレースフルシャットダウンを設定するには

1. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
2. サービスを開き、アクションリストから「無効」をクリックします。待機時間を入力し、[Graceful] を選択します。

TROFS サービスでパーシステンスセッションを有効または無効にする

August 15, 2023

TrofsPersistence フラグを設定して、トランジションアウトオブサービス (TROFS) 状態のサービスが永続セッションを維持する必要があるかどうかを指定できます。モニタが TROFS が有効な場合、モニタプローブに対するサーバの応答に基づいて、サービスを TROFS 状態にすることができます。この応答は、HTTP モニターの TrofsCode パラメーターの値または HTTP-ECV モニターまたは TCP-ECV モニターの TrofsString パラメーターの値と比較されます。コードが一致すると、サービスは TROFS 状態になります。この状態でも、アクティブなクライアント接続を引き続き尊重します。場合によっては、有効なアクティブなセッションに永続的なセッションを含める必要があります。ただし、その他の場合、特に長寿命の永続セッションやカスタムサーバー ID などの永続性メソッドを含む場合、永続セッションを尊重すると、サービスがアウトオブサービス状態に移行するのを防ぐことができます。

trofsPersistence フラグを ENABLED に設定すると、永続セッションが優先されます。DISABLED に設定しても、そうではありません。

コマンドラインインターフェイスを使用して **troFsPersistence** フラグを設定するには

コマンドプロンプトで次のコマンドのいずれかを入力して、`trofsPersistence` 新しい仮想サーバーまたは既存の仮想サーバーのフラグを設定するか、設定をデフォルト値に戻します。

```
add lb vserver <name> [-trofsPersistence ( ENABLED | DISABLED )]

set lb vserver <name> [-trofsPersistence ( ENABLED | DISABLED )]

```



```
unset lb vserver <name> [-trofsPersistence]
<!--NeedCopy-->
```

引数

trofsPersistence。サービスが TROFS 状態の場合、現在のアクティブなクライアント接続と永続セッションでの新しい要求を尊重します。

指定可能な値: 有効、無効。デフォルト：ENABLED。

例:

```
add lb vserver v1 http 10.102.217.42 80 -persistencetype SOURCEIP -
    trofsPersistence ENABLED

set lb vserver v1 -trofsPersistence DISABLED

unset lb vserver v1 -trofsPersistence
<!--NeedCopy-->
```

カスタム **Web** ページへのリクエストの直接作成

August 15, 2023

> 警告
>
> SureConnect（SC）は NetScaler 12.0 ビルド 56.20 以降から廃止され、別の方法として、AppQoE 機能を使用することをお勧めします。詳細については、「AppQoE」を参照してください。

SureConnect が正しく機能するには、グローバルに設定する必要があります。NetScaler には、アプリケーションからの応答を確実にする SureConnect オプションが用意されています。

CLI を使用してサービスに **SureConnect** を設定するには

コマンドプロンプトで入力します。

```
set service <name> -sc <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -sc ON
<!--NeedCopy-->
```


GUI を使用してサービスに **SureConnect** を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[確実接続] を選択します。

サービスダウン時のアクセスを有効にする

August 15, 2023

サービスへのアクセスは、サービスに送信されるパケットをレイヤー 2 モードを使用してブリッジするように NetScaler アプライアンスを構成することで、サービスが無効またはダウン状態のときに有効にできます。通常、リクエストがダウンしているサービスに転送されると、リクエストパケットはドロップされます。ただし、[**Access Down**] 設定を有効にすると、これらの要求パケットは負荷分散されたサーバに直接送信されます。

レイヤ 2 モードおよびレイヤ 3 モードの詳細については、IP アドレッシングを参照してください。

アプライアンスが DOWN サービスに送信されたパケットをブリッジするには、accessDown パラメータを使用してレイヤ 2 モードを有効にします。

CLI を使用してサービスへのアクセスダウンを有効にするには

コマンドプロンプトで入力します。

```
set service <name> -accessDown <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -accessDown YES
<!--NeedCopy-->
```

GUI を使用してサービスへのアクセスダウンを有効にするには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[アクセスダウン] を選択します。

応答の **TCP** バッファリングを有効にする

August 15, 2023

NetScaler アプライアンスには、負荷分散サーバーからの応答のみをバッファリングする TCP バッファリングオプションがあります。これにより、アプライアンスは、クライアントが受け入れることができる最大速度でクライアントへのサーバー応答を配信できます。アプライアンスは、TCP バッファリングに 0 ～4095 MB（MB）のメモリを割り当てて、接続ごとに 4 ～20480 キロバイト（KB）のメモリを割り当てます。

注：サービスレベルで設定された TCP バッファリングは、グローバル設定よりも優先されます。
TCP バッファリングのグローバル設定の詳細については、TCP バッファリングを参照してください。

CLI を使用してサービスで TCP バッファリングを有効にするには

コマンドプロンプトで入力します。

```
set service <name> -TCPB <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -TCPB YES
<!--NeedCopy-->
```

GUI を使用してサービスの TCP バッファリングを有効にするには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[**TCP** バッファリング] を選択します。

圧縮を有効にする

August 15, 2023

NetScaler アプライアンスには、組み込みの圧縮ポリシーセットを使用して HTML ファイルとテキストファイルを透過的に圧縮する圧縮オプションがあります。圧縮により、必要な帯域幅が減少し、帯域幅に制約のある設定でのサーバーの応答性を大幅に向上させることができます。圧縮ポリシーは、仮想サーバーにバインドされたサービスに関連付けられます。ポリシーは、応答を圧縮できるかどうかを決定し、圧縮可能なコンテンツをアプライアンスに送信します。アプライアンスはそれを圧縮してクライアントに送信します。

注：圧縮を正しく機能させるには、グローバルに有効にする必要があります。圧縮をグローバルに設定する方法の詳細については、「圧縮」を参照してください。

CLI を使用して特定のサービスの圧縮を有効にするには

コマンドプロンプトで入力します。


```
set service <name> -CMP <YES | NO>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -CMP YES
<!--NeedCopy-->
```

GUI を使用して特定のサービスの圧縮を有効にするには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[圧縮] を選択します。

UDP 仮想サーバーの外部 TCP 正常性チェックを有効にする

August 15, 2023

パブリッククラウドでは、ネイティブロードバランサを第 1 層として使用する場合、NetScaler ADC アプライアンスを第 2 層ロードバランサーとして使用できます。ネイティブロードバランサは、アプリケーションロードバランサ（ALB）またはネットワークロードバランサ（NLB）になります。ほとんどのパブリッククラウドは、ネイティブロードバランサーで UDP ヘルスプローブをサポートしていません。UDP アプリケーションの正常性を監視するために、パブリッククラウドでは、サービスに TCP ベースのエンドポイントを追加することをお勧めします。エンドポイントは UDP アプリケーションの正常性を反映します。

NetScaler ADC アプライアンスは、UDP 仮想サーバーの外部 TCP ベースのヘルスチェックをサポートします。この機能により、仮想サーバーの VIP と構成済みポートに TCP リスナーが導入されます。TCP リスナーは、仮想サーバーのステータスを反映します。

CLI で UDP 仮想サーバの外部 TCP ヘルスチェックを有効にするには

コマンドプロンプトで次のコマンドを入力して、TcpProbeport オプションを指定して外部 TCP ヘルスチェックを有効にします。

```
add lb vserver <name> <serviceType> <IPAddress> <port> -tcpProbePort <
   tcpProbePort>
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-UDP-1 UDP 10.102.29.60 80 tcpProbePort 5000
<!--NeedCopy-->
```


GUI で **UDP** 仮想サーバの外部 **TCP** ヘルスチェックを有効にするには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを作成します。
2. [追加] をクリックして、仮想サーバーを作成します。
3. [基本設定] ペインの [**TCP** プローブポート] フィールドにポート番号を追加します。
4. [**OK**] をクリックします。

複数のクライアント要求に対するクライアント接続を維持する

August 15, 2023

Client Keep-Alive パラメータを設定して、HTTP または SSL サービスを構成して、Web サイトへのクライアント接続を複数のクライアント要求にわたって開いたままにできます。クライアントのキープアライブが有効になっている場合、負荷分散された Web サーバーが接続を閉じた場合でも、NetScaler ADC アプライアンスはクライアントとそれ自体の間の接続を開いたままにします。この設定では、サービスは 1 つのクライアント接続で複数のクライアント要求を処理できます。

この設定を有効にしない場合、クライアントは Web サイトに送信するリクエストごとに新しい接続を開きます。クライアントのキープアライブ設定により、接続の確立と終了に必要なパケットの往復時間が節約されます。この設定により、各トランザクションを完了するまでの時間も短縮されます。Client Keep-Alive は、HTTP または SSL サービスタイプでのみ有効にできます。

サービスレベルで設定された Client Keep-Alive は、グローバル Client Keep-Alive 設定よりも優先されます。Client Keep-Alive の詳細については、「Client Keep-Alive」を参照してください。

CLI を使用してサービスで **Client Keep-Alive** を有効にするには

コマンドプロンプトで入力します。

```
set service <name> -CKA <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -CKA YES
<!--NeedCopy-->
```

GUI を使用してサービスで **Client Keep-Alive** を有効にするには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[**Client Keep-Alive**] を選択します。

クライアントの IP アドレスを要求ヘッダーに挿入する

August 15, 2023

NetScaler は、サブネット IP（SNIP）アドレスを使用してサーバーに接続します。サーバーはクライアントを認識する必要はありません。

ただし、状況によっては、サーバーがサービスを提供するクライアントを認識する必要があります。クライアント IP 設定を有効にすると、アプライアンスは要求をサーバーに転送する際にクライアントの IPv4 または IPv6 アドレスを挿入します。サーバーは、このクライアント IP を応答のヘッダーに挿入します。したがって、サーバーはクライアントを認識します。

> 注: 複数のヘッダーを挿入するには、次のいずれかを実行する必要があります。
>
> - CLIENT.IS_SSL をチェックし、適切なヘッダーを挿入する書き換えポリシーを追加します。
> - タイプに基づいて、各仮想サーバに適切な書き換えポリシーをバインドします。

CLI を使用してクライアント要求にクライアント IP アドレスを挿入するには

コマンドプロンプトで入力します。

```
set service <name> -CIP <Value> <cipHeader>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -CIP enabled X-Forwarded-For
<!--NeedCopy-->
```

GUI を使用してクライアント要求にクライアント IP アドレスを挿入するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを編集します。
2. サービス設定ペインで、編集アイコンをクリックします。
3. 「負荷分散サービス」ペインで、「クライアント IP アドレスを挿入」チェックボックスを選択します。

位置情報データベースを使用して、ユーザー IP アドレスから場所の詳細を取得する

August 15, 2023

> 注：この機能は、NetScaler ADC リリース 12.1 ビルド 50.x 以降から使用できます。

NetScaler ADC アプライアンスは、大陸、郡、都市などのユーザーの場所の詳細を取得できます。地理位置データベースからの任意のパブリック IP アドレス用。これは、高度なポリシーインフラストラクチャを使用して実行されます。取得した場所の詳細は、次のユースケースを実行するための書き換えアクションまたはレスポンダーアクションで使用されます。

- クライアント要求をバックエンドサーバーに送信するときに、ユーザーの場所の詳細 (国、都市情報など) を含む HTTP ヘッダーを挿入します。
- 無効なユーザーの HTML ページ応答に国名を追加します。

アプライアンスは、監査ロギングメカニズムを使用して場所の詳細をログに記録することもできます。

位置情報関数を使用したユーザーの位置情報の取得

コンポーネントは、次のように相互作用します。

1. ユーザーは、特定の地理的場所からクライアント要求を送信します。
2. NetScaler ADC アプライアンスは、クライアント要求からユーザー IP アドレスを検索し、地理的な場所の詳細を取得します。詳細には、大陸、国、地域、都市、ISP、組織、または地理位置情報データベースのカスタム詳細が含まれます。
3. 場所の詳細が取得されると、アプライアンスはレスポンダポリシーまたは書き換えポリシーのいずれかを使用して要求を評価します。
4. 書き換えポリシーでは、アプライアンスは地理的位置の詳細を含むヘッダーを追加し、バックエンドサーバーに送信します。たとえば、国情報を含むカスタム HTTP ヘッダーを挿入します。
5. レスポンダポリシーでは、アプライアンスは HTTP 要求を評価し、ポリシー評価に基づいて、ユーザへのアクセスを許可するか、またはユーザをエラーページにリダイレクトします。これは、アプリケーションにアクセスしているリージョンがアクセスできないことを示します。

位置情報データベースのセットアップ

前提条件として、NetScaler ADC アプライアンスで実行する地理位置情報データベースが必要です。地理位置情報データベースファイルは、NetScaler ADC ファームウェアで使用できます。ベンダーからデータベースファイルをダウンロードするには、NetScaler ADC 形式に変換してアプライアンスにインポートします。
位置情報データベースの詳細については、「静的近接データベースを作成するためのロケーションファイルの追加 」トピックを参照してください。

ジオロケーション関数

次の表に、パブリック IP アドレスの場所の詳細を取得する位置情報関数の一覧を示します。これらの関数は、書き換えポリシーまたはレスポンダーポリシーで使用できます。

ジオロケーション機能	例
CLIENT.IP.SRC.LOCATION	Asia.In.Karnataka.Bangalore
CLIENT.IP.SRC.LOCATION_GET (1).LOCATION_LONG	India
CLIENT.IP.SRC.LOCATION(3)	Asia.In.Karnataka
CLIENT.IP.SRC.LAT_LONG	12,77
クライアント.IPV6.SRC. ロケーション	North America.US.California.Santa Clara.Verizon.Citrix
クライアント.IPV6.SRC. ロケーション (3)	North America.US.California
CLIENT.IPV6.SRC.LOCATION_GET (1).LOCATION_LONG	米国
CLIENT.IPV6.SRC.LOCATION.GET(3)	California
CLIENT.IPV6.SRC.LAT_LONG	36, -119

位置情報機能の設定

高度なポリシーインフラストラクチャを使用して位置情報機能を設定するには、負荷分散、書き換え、およびレスポンダ機能を有効にして、次のユースケースを完了する必要があります。

負荷分散、レスポンダー、書き換え機能を有効にする

NetScaler ADC アプライアンスで特定の地理的場所からのユーザーアクセスを承認する場合は、負荷分散、書き換え、およびレスポンダー機能を有効にする必要があります。

```
enable ns feature loadbalancing rewrite responder
<!--NeedCopy-->
```

ユースケース **1:** 無効なユーザーを地理的位置の外部にリダイレクトするための位置情報機能の設定

インドのユーザーがウェブページへのアクセスをリクエストしているときは、リクエストをブロックし、国名の HTML ページで応答します。

次の手順は、このユースケースの設定を完了するのに役立ちます。

- 応答側アクションの追加
- レスポンダーポリシーの追加
- レスポンダーポリシーを負荷分散サーバーにバインドする

書き換えアクションと書き換えポリシー構成の GUI 手順の詳細については、Responder のトピックを参照してください。

応答側アクションの追加　国名の HTML ページで応答するレスポンダーアクションを追加します。
コマンドプロンプトで、次のように入力します。

```
add responder action <name> <type> (<target> | <htmlpage>) [-comment <string>] [-responseStatusCode <positive_integer>][-reasonPhrase <string>]
<!--NeedCopy-->
```

例:

```
add responder action responder_act respondwith "HTTP.REQ.VERSION + " 304 Requested Page not allowed in your country - " + CLIENT.IP.SRC.LOCATION.GET (1).LOCATION_LONG + "\r\n""
<!--NeedCopy-->
```

監査ログメッセージアクションの追加　syslog 形式のみ、または syslog とnewnslog形式の両方で、さまざまなログレベルでメッセージをログに記録するように監査メッセージアクションを設定できます。監査メッセージアクションでは、式を使用して監査メッセージの形式を指定します。
コマンドラインインターフェイスを使用して監査メッセージアクションを作成するには

コマンドプロンプトで入力します。

```
add audit messageaction <name> <logLevel> <stringBuilderExpr> [-logtoNewnslog (YES|NO)]
```

例:

```
add audit messageaction msg1 DEBUG ""Request Location: "+CLIENT.IP.SRC.LOCATION"
<!--NeedCopy-->
```

レスポンダーポリシーの追加　レスポンダーポリシーを追加して、インドからのリクエストを識別し、レスポンダーアクションをこのポリシーに関連付けます。
コマンドプロンプトで、次のように入力します。

```
add responder policy <name> <rule> <action> [<undefAction>] [-comment <string>] [-logAction <string>] [-appflowAction <string>]
<!--NeedCopy-->
```

例:

```
add responder policy responder_pol CLIENT.IP.SRC.MATCHES_LOCATION("Asia.India.\*.\*.\*.\*") responder_act -logaction msg1
<!--NeedCopy-->
```


レスポンダーポリシーを負荷分散サーバーにバインドする　レスポンダーポリシーを HTTP/SSL タイプの負荷分散仮想サーバーにバインドします。
コマンドプロンプトで、次のように入力します。

```
bind lb vserver <vserver name> -policyName < policy_name >  -priority <> -type <L7InlineREQUEST | L4Inline-REQUEST>
<!--NeedCopy-->
```

例:

```
bind lb vserver http_vserver -policyName responder_pol -priority 100 -type REQUEST
<!--NeedCopy-->
```

ユースケース **2:** バックエンドが応答するための場所の詳細を含む新しい **HTTP** ヘッダーを挿入するジオロケーション関数の設定

NetScaler ADC アプライアンスがアプリケーションサーバーに送信される要求の HTTP ヘッダーにユーザーの場所を挿入して、サーバーがビジネスロジックの情報を使用できるようにする必要があるシナリオを考えてみましょう。次の手順は、このユースケースの設定を完了するのに役立ちます。

- 書き換えアクションを追加
- 書き換えポリシーを追加する
- リライト・ポリシーをロード・バランシングにバインドする

書き換えアクションおよび書き換えポリシー構成の GUI 手順の詳細については、Responder のトピックを参照してください。

書き換えアクションを追加　リライトアクションを追加して、リクエストにユーザージオロケーションの詳細を含むカスタム HTTP ヘッダーを挿入し、バックエンドサーバーに送信します。
コマンドプロンプトで、次のように入力します。

```
add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-search <expression>] [-refineSearch <string>][-comment <string>]
<!--NeedCopy-->
```

例:

```
add rewrite action rewrite_act insert_http_header "User_location" CLIENT.IP.SRC.LOCATION
<!--NeedCopy-->
```

書き換えポリシーを追加する　書き換えポリシーを追加して、書き換えアクションを実行する必要があるかどうかを評価します。この場合、アプリケーションサーバーに送信されるすべてのリクエストにはカスタム HTTP ヘッダーが

必要なため、ルールは「true」になります。
コマンドプロンプトで、次のように入力します。

```
add rewrite policy <name> <rule> <action> [<undefAction>] [-comment <
    string>] [-logAction <string>]
<!--NeedCopy-->
```

例:

```
add rewrite policy rewrite_pol true rewrite_act -logaction log_act
<!--NeedCopy-->
```

リライト・ポリシーをロード・バランシングにバインドする　書き換えポリシーを、HTTP/SSL タイプの必要な負荷分散仮想サーバーにバインドします。
コマンドプロンプトで、次のように入力します。

```
bind lb vserver <vserver name> -policyName < policy_name >  -priority
    <> -type <L7InlineREQUEST | L4Inline-REQUEST>
<!--NeedCopy-->
```

例:

```
bind lb vserver http_vserver -policyName rewrite_pol -priority 100 -
    type REQUEST
<!--NeedCopy-->
```

位置情報詳細のロギングに対する **syslog** サポート（オプション）

ユーザーの地理的位置情報の詳細を記録する場合は、リクエストがポリシーに一致したときに実行される SYSLOG アクションを指定する必要があります。アプライアンスは、詳細をログ・メッセージとして ns.log ファイルに保存します。
SYSLOG および NSLOG 監査の詳細については、監査ログのトピックを参照してください。

ユーザーの地理位置情報の詳細の出力

次の出力は、バンガロールの場所からアプリケーションにアクセスしようとして、アプライアンスがジオロケーション機能「CLIENT.IP.SRC.LOCATION」を使用する場合、SYSLOG またはnewnslogアクションを使用してアプライアンスに記録されます。

```
Asia.India.Karnataka.Banglore
<!--NeedCopy-->
```

出力ログの例:


```
07/23/2018:19:03:54 GMT  Debug 0-PPE-0 : default REWRITE Message 22 0 :
     "Request Location: asia.in.karnataka.bangalore.\*.\*"
07/23/2018:19:23:55 GMT  Debug 0-PPE-0 : default RESPONDER Message 32 0
Done
<!--NeedCopy-->
```

サーバーへの接続時にクライアントの送信元 IP アドレスを使用する

August 15, 2023

送信元 IP アドレスを変更せずに、クライアントからサーバーにパケットを転送するように NetScaler アプライアンスを構成できます。これは、HTTP 以外のサービスを使用する場合など、クライアントの IP アドレスをヘッダーに挿入できない場合に便利です。

USIP をグローバルに設定する方法の詳細については、「送信元 IP モードの使用の有効化」を参照してください。

CLI を使用してサービスの USIP モードを有効にするには

コマンドプロンプトで入力します。

```
set service <name> -usip (YES | NO)
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -usip YES
<!--NeedCopy-->
```

GUI を使用してサービスの USIP モードを有効にするには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを開きます。
2. [詳細設定] の [サービス設定] セクションで、[ソース IP アドレスを使用] を選択します。

v4-v6 ロードバランシング構成でのバックエンド通信にクライアント送信元 IP アドレスを使用する

August 15, 2023

v4 から v6 への負荷分散構成では、USIP が無効になっているサービスの場合、NetScaler ADC アプライアンスは、構成済みの IPv6 SNIP（SNIP6）アドレスのいずれかから関連サーバーと通信します。

USIP が有効になっているサービスの場合は、リクエストパケットのクライアントの IP アドレスを関連サーバーに認識させるために、グローバル USIP NAT プレフィックスパラメータを設定する必要があります。USIP NAT プレフィックスは、NetScaler アプライアンス上で構成された長さ 32/40/48/56/64/96 ビットのグローバル IPv6 プレフィックスです。

USIP が有効になっている負荷分散サービスの場合、アプライアンスは IPv4 要求パケットを IPv6 パケットに変換し、変換された IPv6 パケットの送信元 IP アドレスを次の連結に設定します。

- 長さが 32/40/48/56/64/96 ビットの USIP NAT プレフィックス。
- USIP NAT プレフィックスの長さが 96 ビット未満の場合はゼロが埋められます。ゼロで埋められたビット数 = 96-USIP NAT プレフィックス長。たとえば、USIP NAT プレフィクス長が 64 の場合、ゼロで埋められるビット数は 96-64 = 32 です。
- 要求パケットで受信された IPv4 送信元アドレス [32 ビット]。つまり、送信元 IPv6 アドレスの最後の 32 ビットは、クライアントの IPv4 アドレスに設定されます。

サーバーから IPv6 応答パケットを受信すると、NetScaler アプライアンスは IPv6 パケットを IPv4 パケットに変換し、変換された IPv4 パケットの宛先 IP アドレスを IPv6 パケットの宛先 IP アドレスの最後の 32 ビットに設定します。

> 注：この機能は、NetScaler Gateway の構成、コンテンツスイッチング、およびキャッシュリダイレクトの負荷分散構成ではサポートされていません。

構成の手順

v4 から v6 への負荷分散構成の USIP の設定は、次のタスクで構成されます。

- グローバル **USIP NAT** プレフィックスを追加します。これは、アプライアンスで設定する長さが 32/40/48/56/64/96 ビットのグローバル IPv6 プレフィックスです。
- グローバル **USIP** モードを有効にします。詳細については、「送信元 IP モードの使用を有効にする」を参照してください。
- 負荷分散サービスの **USIP** モードを有効にします。詳細については、「サーバーに接続するときにクライアントの送信元 IP アドレスを使用する」を参照してください。

CLI を使用してグローバル **USIP NAT** プレフィクスを追加するには、次の手順を実行します。

- `set ipv6 -usipnatprefix <prefix/prefix_length>`
- `show ipv6`

GUI を使用してグローバル **USIP NAT** プレフィックスを追加するには：

1. [システム] > [ネットワーク] に移動し、[**IPv6** 設定の変更] をクリックします。
2. **IPV6** の構成画面で、**USIP NAT** プレフィックスパラメータを設定します。

設定例

```
> set ipv6 -usipnatprefix 2001:DB8:90::/64
Done

> enable ns mode USIP
Done

> add lb vserver LBVS-1 HTTP 203.0.113.90 80
Done

> add service SVC-1 2001:DB8:5001::30 HTTP 80 -usip yes
Done

> add service SVC-2 2001:DB8:5001::60 HTTP 80 -usip yes
Done

> bind lb vserver LBVS-1 SVC-1
Done

> bind lb vserver LBVS-1 SVC-2
Done

<!--NeedCopy-->
```

サーバー側接続の送信元ポートを構成する

August 15, 2023

NetScaler ADC アプライアンスが物理サーバーに接続するときに、クライアントのリクエストからのソースポートを使用するか、接続のソースポートとしてプロキシポートを使用できます。Use Proxy Port パラメーターを YES に設定すると、次のような状況を処理できます。

- NetScaler アプライアンスは、LBVS1 と LBVS2 の 2 つの負荷分散仮想サーバーで構成されています。
- 両方の仮想サーバーは同じサービス（S-ANY）にバインドされています。
- (クライアントの) 送信元 IP アドレス (USIP) の使用がサービスで有効になっています。
- クライアント C1 は、同じサービスに対して、Req1 と Req2 の 2 つの要求を送信します。
- LBVS1 は Req1 を受信し、LBVS2 は Req2 を受信します。
- LBVS1 と LBVS2 は要求を S-ANY に転送し、S-ANY が応答を送信すると、LBVS1 と LBVS2 は応答をクライアントに転送します。
- 次の 2 つのケースを検討します。
  - クライアントポートを使用してください。アプライアンスがクライアント・ポートを使用する場合、仮想サーバーは両方ともクライアントの IP アドレス（USIP が ON のため）とサーバーへの接続時にクラ

イアントのポートを使用します。したがって、サービスが応答を送信すると、アプライアンスは応答を受信する必要がある仮想サーバを特定できません。

- プロキシポートを使用してください。アプライアンスがプロキシ・ポートを使用する場合、仮想サーバーはクライアントの IP アドレスを使用します（USIP が ON であるため）。ただし、サーバーへの接続時にはポートは異なります。したがって、サービスが応答を送信すると、ポート番号は応答を受信する必要がある仮想サーバーを識別します。

ただし、完全に透過的なキャッシュのリダイレクト構成など、完全に透過的な構成が必要な場合は、NetScaler ADC アプライアンスがクライアントの要求からソースポートを使用できるように、プロキシポート設定の使用を無効にする必要があります。

「ソース IP (USIP) を使用」オプションが有効になっている場合は、「プロキシポートを使用」オプションが適切になります。TCP、HTTP、SSL などの TCP ベースのサービスタイプでは、このオプションはデフォルトで有効になっています。UDP や DNS など ANY を含む UDP ベースのサービスタイプの場合、このオプションはデフォルトで無効になっています。USIP オプションの詳細については、「ソース IP モードの使用を有効にする」を参照してください。

プロキシポートの使用（**Use Proxy Port**）設定は、グローバルに、または特定のサービスで設定できます。

サービスのプロキシポートの使用設定を構成する

グローバル設定を上書きする場合は、サービスの [プロキシポートを使用] 設定を構成します。

CLI を使用してサービスの **[**プロキシポートを使用**]** 設定を構成するには

コマンドプロンプトで入力します。

```
set service <name> -useProxyPort (YES | NO)
<!--NeedCopy-->
```

例:

```
set service svc1 -useproxyport YES
Done

show service svc1
svc1 (10.102.29.30:80) - HTTP
State: UP
. . .
Use Source IP: YES Use Proxy Port: YES
. . .
Done
<!--NeedCopy-->
```


GUI を使用してサービスの **Use Proxy Port** 設定を構成するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[プロキシポートを使用] を選択します。

プロキシポートの使用設定をグローバルに設定

NetScaler ADC アプライアンス上のすべてのサービスに設定を適用する場合は、［プロキシポートを使用する］設定をグローバルに構成します。サービス固有の [プロキシポートを使用] 設定は、グローバル設定よりも優先されます。

CLI を使用して **[**プロキシポートを使用**]** 設定をグローバルに設定するには

コマンドプロンプトで次のコマンドを入力して、[プロキシポートを使用] 設定をグローバルに設定し、構成を確認します。

```
set ns param -useproxyport ( ENABLED | DISABLED )`
show ns param`
<!--NeedCopy-->
```

例**:**

```
set ns param -useproxyport ENABLED

Done

show ns param
Global configuration settings:
. . .
Use Proxy Port: ENABLED
Done
<!--NeedCopy-->
```

GUI を使用して **Use Proxy Port** 設定をグローバルに設定するには

[システム**] > [**設定**] > [**グローバルシステム設定の変更] に移動し、[プロキシポートを使用] を選択または選択解除します。

クライアント接続の数に制限を設定する

August 15, 2023

負荷分散された各サーバーが処理できるクライアント接続の最大数を指定できます。NetScaler アプライアンスは、この制限に達するまでサーバーへのクライアント接続のみを開きます。負荷分散されたサーバーが限界に達すると、監視プローブはスキップされ、既存の接続の処理が完了して容量が解放されるまで、サーバーは負荷分散に使用されません。

最大クライアント設定の詳細については、「ドメイン名ベースのサービスの負荷分散」を参照してください。

注: クローズ処理中の接続は、この制限では考慮されません。

CLI を使用してクライアント接続数の制限を設定するには

コマンドプロンプトで入力します。

```
set service <name> -maxclient <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -maxClient 1000
<!--NeedCopy-->
```

GUI を使用してクライアント接続数の制限を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[最大クライアント数] を選択します。

サーバーへの接続あたりの要求数の制限を設定する

August 15, 2023

NetScaler ADC アプライアンスは、接続を再利用してパフォーマンスを向上させるように構成できます。ただし、一部のシナリオでは、負荷分散された Web サーバーで多数の要求に対して接続が再利用されるときに問題が発生することがあります。HTTP サービスまたは SSL サービスの場合は、max request オプションを使用して、負荷分散された Web サーバーへの 1 つの接続を介して送信される要求の数を制限します。

注: 最大リクエストオプションを設定できるのは HTTP または SSL サービスのみです。

CLI を使用して接続ごとのクライアントリクエスト数を制限するには

コマンドプロンプトで入力します。


```
set service <ServiceName> -maxReq <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -maxReq 100
<!--NeedCopy-->
```

GUI を使用して接続ごとのクライアント要求数を制限するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[最大要求数] を選択します。

サービスにバインドされたモニターのしきい値を設定する

August 15, 2023

NetScaler アプライアンスは、そのサービスにバインドされていて稼働しているすべてのモニターの重みの合計が、サービスに設定されているしきい値以上である場合にのみ、サービスを UP として指定します。モニターの重みによって、そのモニターがバインドされているサービスを UP として指定するうえで、そのモニターがどの程度貢献できるかが決まります。

デフォルトでは、モニターのしきい値は 0 に設定され、モニターの重みは 1 に設定されます。その場合、すべてのモニターの重みは等しく、いずれかのモニターがダウンするとサービスが停止する可能性があります。

たとえば、それぞれ Monitor-HTTP-1、Monitor-HTTP-2、Monitor-HTTP-3 という名前の 3 つのモニターがサービス HTTP-1 にバインドされており、サービスに設定されているしきい値が 3 だとします。各モニターに次の重みが割り当てられているとします。

- モニタ-HTTP-1 の重みは 1 です。
- モニター-HTTP-2 の重みは 3 です。
- モニター HTTP-3 の重みは 1 です。

次のいずれかに当てはまる場合にのみ、サービスが UP とマークされます。

- モニタ-HTTP-2 は稼働しています。
- モニター HTTP-2 とモニター HTTP-1 またはモニター HTTP-3 が稼働している
- 3 台のモニタはすべて稼働しています。

CLI を使用してサービスの監視しきい値を設定するには

コマンドプロンプトで入力します。


```
set service <name> -monThreshold <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -monThreshold 100
<!--NeedCopy-->
```

GUI を使用してサービスの監視しきい値を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[モニタのしきい値] を選択します。

アイドル状態のクライアント接続のタイムアウト値を設定する

August 15, 2023

タイムアウト値を使用してサービスを構成し、設定した時間が経過したときにアイドル状態のクライアント接続をすべて終了できます。設定した時間内にクライアントがアイドル状態になると、NetScaler アプライアンスはクライアント接続を閉じます。

CLI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

コマンドプロンプトで入力します。

```
set service <name> -cltTimeout <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -cltTimeout 100
<!--NeedCopy-->
```

GUI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[クライアントアイドルタイムアウト] を選択します。

アイドル状態のサーバー接続のタイムアウト値を設定する

August 15, 2023

タイムアウト値を使用してサービスを構成し、設定した時間 (秒単位) が経過したときに、アイドル状態のサーバー接続をすべて終了できます。サーバーが設定された時間アイドル状態になると、NetScaler アプライアンスはサーバー接続を閉じます。

CLI を使用してアイドル状態のサーバー接続のタイムアウト値を設定するには

コマンドプロンプトで入力します。

```
set service <name> -svrTimeout <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -svrTimeout 100
<!--NeedCopy-->
```

GUI を使用してアイドル状態のサーバー接続のタイムアウト値を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[サーバーアイドルタイムアウト] を選択します。

クライアントによる帯域幅の使用に制限を設定する

August 15, 2023

場合によっては、サーバーの帯域幅が制限されてクライアント要求を処理し、過負荷になることがあります。サーバの過負荷を防ぐために、サーバによって処理される帯域幅の上限を Kbps 単位で指定できます。NetScaler アプライアンスは、この制限に達するまでのみ、要求を負荷分散されたサーバーに転送します。

CLI を使用してサービスの最大帯域幅制限を設定するには

コマンドプロンプトで入力します。

```
set service <name> -maxBandwidth <Value>
<!--NeedCopy-->
```


例:

```
set service Service-HTTP-1 -maxBandwidth 100
<!--NeedCopy-->
```

GUI を使用してサービスの最大帯域幅制限を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[最大帯域幅] を選択します。

クライアント要求をキャッシュにリダイレクトする

August 15, 2023

クライアント要求をキャッシュにリダイレクトし、キャッシュできない要求を、設定された負荷分散方法で選択されたサービスに転送するようにサービスを構成できます。

CLI を使用してサービスにキャッシュリダイレクトを設定するには

コマンドプロンプトで入力します。

```
set service <name> -cacheable <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -cacheable YES
<!--NeedCopy-->
```

GUI を使用してサービスにキャッシュリダイレクトを設定するには

1. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。
2. サービスを開き、キャッシュタイプを設定します。

VLAN の透過性のために **VLAN ID** を保持する

August 15, 2023

サーバーに転送されるパケットにクライアントの VLAN 識別子を保持するように負荷分散仮想サーバーを構成できます。仮想サーバーは、ANY タイプのワイルドカード仮想サーバーで、MAC モードで機能している必要があります。

CLI を使用してクライアント **VLAN ID** を保持するように負荷分散仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、クライアント VLAN ID を保持するように負荷分散仮想サーバーを構成し、構成を確認します。

```
set lb vserver <name> -m MAC -macmodeRetainvlan ENABLED

show lb vserver <name>
<!--NeedCopy-->
```

> 注
>
> -m MACオプションが有効になっている仮想サーバーにバインドされているサービスの場合は、非ユーザーモニターをバインドする必要があります。

GUI を使用してクライアント **VLAN ID** を保持するようにロードバランシング仮想サーバを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[**VLAN ID** を保持] を選択します。

バインドしたサービスの正常性のパーセンテージに基づいて自動状態遷移を構成する

August 15, 2023

アクティブなサービスの割合が設定されたしきい値を下回った場合に、自動的に稼働状態から停止状態に移行するように負荷分散仮想サーバーを構成できます。たとえば、10 のサービスを負荷分散仮想サーバーにバインドし、その仮想サーバーのしきい値を 50% に設定した場合、6 つ以上のサービスが停止すると、仮想サーバーは UP から DOWN に移行します。ヘルスのパーセンテージがしきい値を上回ると、仮想サーバーは UP 状態に戻ります。

バインドされたサービスの状態パーセンテージによって仮想サーバーの状態が変化したときに NetScaler アプライアンスから通知を受けさせたい場合は、ENTITY-STATE という SNMP アラームを有効にすることもできます。

CLI を使用してパーセンテージベースの自動状態移行を設定するには

コマンドプロンプトで次のコマンドを入力して、仮想サーバーの自動状態遷移を構成し、構成を確認します。

```
set lb vserver <name> -healthThreshold <positive_integer>

```



```
show lb vserver <name>
<!--NeedCopy-->
```

GUI を使用してパーセンテージに基づく自動状態遷移を構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[ヘルスしきい値] を設定します。

CLI を使用して **ENTITY-STATE** アラームを有効にするには

コマンドプロンプトで次のコマンドを入力して ENTITY-STATE SNMP アラームを有効にし、構成を確認します。

```
enable snmp alarm ENTITY-STATE

show snmp alarm
<!--NeedCopy-->
```

GUI を使用して **ENTITY-STATE** アラームを有効にするには

1. [システム**] > [SNMP] > [**アラーム] に移動します。
2. [**ENTITY-STATE**] を選択し、[アクション] リストで [有効] を選択します。

NetScaler ロケーションに基づく静的近接性

August 15, 2023

> 注
>
> セルフからの近接パラメータは、リリース 13.1 ビルド 48.x 以降で使用できます。

静的近接負荷分散方式を構成すると、NetScaler ループバック IP アドレスではなくクライアント IP アドレスに基づいてサーバーが選択されます。その結果、応答時間が長くなる可能性があります。「self」パラメーターを有効にすると、NetScaler のループバック IP アドレスを使用して、NetScaler に最も近いサーバーにリクエストが送信されます。このパラメーターを「YES」に設定すると、サーバーがクライアントよりも NetScaler に近い場所にある場合、応答時間が短縮されます。

前提要件

負荷分散方法として静的近接を選択

CLI を使用して ProximityFromSelf パラメータを設定するには

コマンドプロンプトで次のコマンドを入力して ProximityFromSelf パラメーターを構成し、構成を確認します。

```
set lbparameter -proximityFromSelf <NO/YES>
show lbparameter

<!--NeedCopy-->
```

例:

```
set lbparameter -proximityFromSelf Yes
<!--NeedCopy-->
```

GUI を使用して「自己からの距離」パラメータを設定するには

1. [トラフィック管理] > [負荷分散] に移動します。
2. 「設定」セクションの「負荷分散」ページで、「負荷分散パラメーターの変更」をクリックします。
3. 「自分」から「近さ」を選択します。
4. [**OK**] をクリックします。

内蔵モニター

August 15, 2023

NetScaler ADC アプライアンスには、サービスを監視するために使用できるさまざまな組み込みモニターが含まれています。これらの組み込みモニタは、一般的なプロトコルのほとんどを処理します。これらは、要件に合わせて間隔、応答タイムアウトなどの一部のパラメータを変更するオプションを提供します。ただし、モニタ名とプロトコルは変更できません。詳細については、「モニターの変更」を参照してください。組み込みモニターをサービスにバインドし、サービスからバインド解除することもできます。

> 注:
>
> 組み込みモニターに基づいてカスタムモニターを作成できます。カスタムモニターの作成方法については、「負荷分散セットアップでのモニターの構成」を参照してください。

TCP ベースのアプリケーション監視

August 15, 2023

NetScaler ADC アプライアンスには、TCP ベースのアプリケーションを監視する 2 つの組み込みモニターがあります。`tcp-default`および`ping-default`。サービスを作成すると、適切なデフォルトモニターが自動的にバインドされるため、サービスが UP であればすぐに使用できます。tcp-default モニタは、すべての TCP サービスにバインドされます。ping-default モニタは、TCP 以外のすべてのサービスにバインドされます。

既定のモニターを削除または変更することはできません。他のモニタを TCP サービスにバインドすると、デフォルトモニタはサービスからバインド解除されます。次の表に、モニタタイプ、および各タイプに関連付けられたパラメータとモニタリングプロセスを示します。

モニタタイプ	特定のパラメータ	プロセス
tcp	該当なし	NetScaler ADC アプライアンスは、モニターの宛先との 3 ウェイハンドシェイクを確立し、接続を閉じます。アプライアンスが宛先への TCP トラフィックを監視する場合、TCP モニタリング要求は送信されません。これは、LRTM が無効になっている場合に発生します。デフォルトでは、このモニタでは LRTM は無効になっています。
http	httprerequest [“HEAD/”]-サービスに送信される HTTP リクエスト。respcode [200]-一連の HTTP レスポンスコードがサービスから送信されることが期待されます。	NetScaler ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは HTTP 要求を送信し、応答コードと構成済みの応答コードのセットを比較します。
tcp-ecv	send [””]-サービスに送信されるデータです。文字列の最大許容長は 512 バイトです。recv [””]-サービスからの応答が期待されます。文字列の最大許容長は 128 バイトです。最後の文字は NULL 終端です。	NetScaler ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは送信パラメータを使用して特定のデータをサービスに送信し、受信パラメータを介して特定の応答を期待します。サーバーによって異なるサイズのセグメントが送信されます。ただし、パターンは 16 個の TCP セグメント内にある必要があります。

モニタタイプ	特定のパラメータ	プロセス
http-ecv	send [”]-サービスに送信される HTTP データ、recv [” “]-サービスからの期待される HTTP レスポンスデータ	NetScaler ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは send パラメータを使用して HTTP データをサービスに送信し、receive パラメータで指定された HTTP 応答を期待します。(HTTP ヘッダーを含まない HTTP ボディパーツ)。空の応答データは、任意の応答と一致します。期待されるデータは、レスポンスの HTTP 本体の最初の 24 K バイトの任意の場所にある可能性があります。
ping	該当なし	NetScaler ADC アプライアンスは、モニタの宛先に ICMP エコー要求を送信し、ICMP エコー応答を期待します。

TCP ベースのアプリケーションの組み込みモニターを構成するには、負荷分散セットアップでのモニタの構成を参照してください。

CLI を使用して TCP ベースのモニタを構成するには

次のコマンドを入力します。

```
add lb monitor <monitorName> <type> -respCode <int[-int]> -httpRequest
    <string> -resptimeout <integer> [<units>] -retries <integer> -
    downTime <integer> [<units>] -action <action>
<!--NeedCopy-->
```

TCP モニタタイプの例**:**

```
add lb monitor Exch2010-RPC-AddressBook TCP -LRTM ENABLED -interval 10
    -resptimeout 5 -destPort 59601
<!--NeedCopy-->
```

HTTP モニタタイプの例**:**

```
add lb monitor Mon_S4B_FE_2 HTTP -respCode 200 -httpRequest "GET /
    Autodiscover/XFrame/XFrame.html" -LRTM ENABLED -retries 10 -secure
    YES
<!--NeedCopy-->
```


HTTP-ECV モニタタイプの例**:**

```
add lb monitor STM_EXC2016_SSLBridge_MON HTTP-ECV -send "GET /owa/
   healthcheck.htm" -recv "200 OK" -LRTM ENABLED -destPort 443 -secure
   YES
<!--NeedCopy-->
```

PING モニタタイプの例**:**

```
add lb monitor lbmon-localhost-ping PING -LRTM DISABLED -destIP
   127.0.0.1
<!--NeedCopy-->
```

SSL サービスの監視

December 8, 2023

NetScaler アプライアンスには、セキュアモニター、TCPS および HTTPS が組み込まれています。セキュア・モニターを使用して、HTTP トラフィックと非 HTTP トラフィックをモニタリングできます。セキュア HTTP モニタを設定するには、モニタータイプを HTTP として選択し、セキュアフラグを設定します。セキュア TCP モニタを設定するには、モニタータイプを TCP として選択し、セキュアフラグを設定します。セキュア・モニターは次のように機能します。

- 安全な **TCP** モニタリング。NetScaler アプライアンスは TCP 接続を確立します。接続が確立されると、アプライアンスはサーバーと SSL ハンドシェイクを実行します。ハンドシェイクが終了すると、アプライアンスは接続を閉じます。
- 安全な **HTTP** モニタリング。NetScaler アプライアンスは TCP 接続を確立します。接続が確立されると、アプライアンスはサーバーと SSL ハンドシェイクを実行します。SSL 接続が確立されると、アプライアンスは暗号化されたチャネルを介して HTTP リクエストを送信し、応答コードを確認します。

次の表は、SSL サービスの監視に使用できる組み込みモニターを示しています。

モニタタイプ	プローブ	達成基準 (直接条件)
TCP	TCP 接続; SSL ハンドシェイク	TCP 接続が正常に確立され、SSL ハンドシェイクが成功しました。
HTTP	TCP 接続、SSL ハンドシェイク、暗号化された HTTP リクエスト	TCP 接続が成功し、SSL ハンドシェイクが成功し、サーバーの HTTP 応答で期待される HTTP 応答コードが暗号化されます。

モニタタイプ	プローブ	達成基準 (直接条件)
TCP-ECV	TCP 接続。SSL ハンドシェイク (サーバーに送信されるデータは暗号化されます。)	成功した TCP 接続が確立され、正常な SSL ハンドシェイクが実行され、予期された TCP データがサーバーから受信されます。
HTTP-ECV	TCP 接続; SSL ハンドシェイク (暗号化された HTTP リクエスト)	TCP 接続が成功し、SSL ハンドシェイクが成功し、期待どおりの HTTP データがサーバーから受信されます。

HTTP-ECV ヘルスチェックモニターの設定例

HTTP サービスには、拡張コンテンツ検証 (ECV) が可能なモニターがあらかじめ定義されています。
これらのモニターは、TCP 接続の成功後に検証が必要な場合に使用されます。これらのモニターは、以下の条件がすべて満たされた場合に、サービスが稼働中であることを検証します。

- TCP 接続が成功しました。
- 特定のタイプのリクエストを生成する必要があります。
- 受信文字列から特定のメッセージが返信されることが期待されます。

これらのモニターでは、要求文字列と応答文字列が設定されます。NetScaler モニターが受信した応答文字列が設定された文字列と一致する場合、サービスは UP とマークされます。

GUI を使用してモニターをサービスにバインドする

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを作成して、プロトコルを **SSL** として指定します。[**OK**] をクリックします。
2. 「サービスから負荷分散へのモニターのバインディング」ペインをクリックし、「バインドの追加」をクリックします。
3. モニタータイプを HTTP-ECV として選択し、**[**編集**]** をクリックします。
4. 「基本パラメータ」タブの「モニタの構成」ペインで、次のパラメータの値を入力します。
    - 送信文字列—モニターがサービスに送信する必要がある文字列。
    - 受信文字列—サービスを稼働中としてマークするためにモニターが受信する必要がある文字列。
5. 「**OK**」をクリックして、モニターの設定を完了します。
6. [**Select**] をクリックします。
7. 「バインド」をクリックして、**HTTP-ECV** モニターをサービスにバインドします。
8. [閉じる] をクリックします。

CLI を使用してモニターを作成してサービスにバインドする

コマンドプロンプトで入力します:

```
add lb monitor <monitor-name> http-ecv
bind service <servicename> -monitorName <monitor-name>
<!--NeedCopy-->
```

例:

```
add lb monitor monitor-1 http-ecv
bind service services1 -monitorName monitor-1
<!--NeedCopy-->
```

HTTP/2 サービスの監視

August 15, 2023

NetScaler ADC アプライアンスは、HTTP/2 サービスのヘルスステータスを監視するための HTTP/2 モニターをサポートしています。

HTTP/2 モニタは、2 つの異なる方法で設定できます。トラフィックタイプに応じて、HTTP/2 モニタを設定できます。

- **HTTP/2** ダイレクト。HTTP/2 Direct は、非セキュアな HTTP/2 サービスを監視するように設定できます。
- **HTTP/2 SSL**。SSL を介したセキュアなトラフィックをモニタリングするために HTTP/2 SSL を設定できます。HTTP/2 で secure flag パラメータを有効にして、SSL トラフィックを監視します。

http2direct と http2ssl は、HTTP/2 プロトコルでサポートされている 2 つの異なる組み込みモニターです。

次の表に、各タイプに関連付けられた構成タイプ、およびモニタリングプロセスを示します。

構成タイプ	プローブ	成功基準
HTTP/2 ダイレクト	TCP 接続; HTTP2 接続の序文 & 設定ネゴシエーション; HTTP2 要求	HTTP/2 応答ステータスコードは、設定されたレスポンスコードと一致する必要があります。
HTTP/2 SSL	TCP 接続; SSL ハンドシェイク; HTTP2 接続の序文と設定ネゴシエーション; HTTP2 要求	サーバーは常に HTTP/2 プロトコルでALPNを選択する必要があり、HTTP/2 応答ステータスコードは、設定された応答コードと一致する必要があります。

CLI を使用して **HTTP/2** モニタをサービスにバインドする

コマンドプロンプトで入力します。

- `bind service <servicename> -monitorName <name>`
- `bind service <servicename> -monitorName <name>`

例**:**

- `bind service s1 -monitorName http2direct`
- `bind service s2 -monitorName http2ssl`

プロキシプロトコルサービスの監視

August 15, 2023

プロキシプロトコルを備えた NetScaler ADC アプライアンスは、モニターチェックをサポートしています。モニタチェックでは、バックエンドサーバーがプロキシプロトコルもサポートしていることを確認します。NetScaler ADC アプライアンスには、HTTP または TCP 関連サービス用の 4 つのビルトインモニタタイプ（HTTP、HTTPS、HTTP-ECV、TCP-ECV）があります。

次の表に、モニタタイプ、および各タイプに関連付けられたパラメータとモニタリングプロセスを示します。

構成タイプ	プローブ	成功基準
HTTP	`httprequest` [「HEAD/」]-サービスに送信される HTTP リクエスト。`respcode` [200]-サービスからの一連の HTTP 応答コードが返されることが期待されます。	NetScaler ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは HTTP 要求を送信し、応答コードと構成済みの応答コードのセットを比較します。
HTTPS	`httprequest` [「HEAD/」]-サービスに送信される HTTPS リクエスト。`respcode` [200]-サービスからの一連の HTTPS 応答コードが返されることが期待されます。	NetScaler ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは HTTPS 要求を送信し、応答コードを設定済みの応答コードセットと比較します。

構成タイプ	プローブ	成功基準
HTTP-ECV	send [」]-サービスに送信される HTTP データ。Received [””] - the expected HTTP response data from the service	NetScaler ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは send パラメータを使用して HTTP データをサービスに送信し、receive パラメータで指定された HTTP 応答を期待します。(HTTP ヘッダーを含まない HTTP ボディパーツ)。空の応答データは、任意の応答と一致します。期待されるデータは、レスポンスの HTTP 本体の最初の 24 K バイトの任意の場所にある可能性があります。
TCP-ECV	send [」]-サービスに送信されるデータです。文字列の最大許容長は 512 K バイトです。received [」] はサービスからの期待される応答です。文字列の最大許容長は 128 K バイトです。	NetScaler ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは送信パラメータを使用して特定のデータをサービスに送信し、受信パラメータを介して特定の応答を期待します。サーバーによって異なるサイズのセグメントが送信されます。ただし、パターンは 16 個の TCP セグメント内にある必要があります。

プロキシプロトコルモニタは、`netprofile`を使用して設定できます。

CLI を使用したプロキシプロトコルモニタの設定

コマンドプロンプトで入力します。

1. プロキシプロトコルを有効にしてネットプロファイルを追加する

```
add netprofile <name> -proxyProtocol ( ENABLED | DISABLED )
```

例:

```
1 add netprofile profile1 -proxyProtocol ENABLED
```


1. ネットプロファイルをサービスにバインドします。

   ```
   set service <name> -netprofile <netprofile-name>
   ```

例:

```
1 set service S1  - netprofile profile1
```

> 注
>
> ネットプロファイルをサービスにバインドする場合は、前述のコマンドを実行できます。

1. ネットプロファイルをモニターにバインドします。

   ```
   set lb monitor <monitor-name> <type> -netprofile <netprofile-name
   >
   ```

例:

```
1 set lb monitor http1 HTTPS  - netprofile profile1
```

> 注
>
> - ネットプロファイルをモニタにバインドする場合は、前述のコマンドを実行できます。
> - お好みのモニタタイプを選択できます。HTTP、HTTPS、TCP-ECV、または HTTP-ECV のいずれかになります。
>
> 重要
>
> - 一般に、サービスにバインドされたネットプロファイル（プロキシプロトコルが有効）が考慮されます。
> - ネットプロファイルがモニターとサービスの両方にバインドされている場合、モニターにバインドされたネットプロファイルが考慮されます。サービスにバインドされたネットプロファイルは無視されます。

FTP サービスの監視

August 15, 2023

FTP サービスを監視するために、NetScaler アプライアンスは FTP サーバーへの 2 つの接続を開きます。まず、クライアントと FTP サーバー間でコマンドを転送するために使用される制御ポートに接続します。期待される応答を受け取ると、データポートに接続します。データポートは、クライアントと FTP サーバー間のファイル転送に使用されます。FTP サーバーが両方の接続で期待どおりに応答した場合にのみ、UP とマークされます。

注：監視プローブは、NSIP アドレスから発信されます。

NetScaler アプライアンスには、FTP サービス用の 2 つのビルトインモニター（FTP モニターと FTP-EXTENDED モニター）があります。FTP-EXTENDED モニターはスクリプト可能なモニターです。nsftp.pl スクリプトを使用し

ています。FTP-EXTENDED 監視スクリプトが拡張され、安全なプローブを FTP サービスに送信できるようになりました。FTP-EXTENDED タイプのモニターを作成できます。nsftp.pl スクリプトは、デフォルトディレクトリから自動的に取得されます。

CLI を使用してセキュア FTP プローブを FTP サービスに送信するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <type> -username <string> -password <
    string> -filename <filename>
<!--NeedCopy-->
```

例

```
add monitor mon1 FTP-EXTENDED -username root -password freebsd -
    filename fsdf
<!--NeedCopy-->
```

GUI を使用してセキュア FTP プローブを FTP サービスに送信するには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. モニタータイプを **FTP-EXTENDED** に指定し、パラメータを設定します。
3.「特殊パラメータ」で、ファイル名、ユーザー名、および パスワードを指定します。

FTP サービスの状態をチェックするように組み込みモニターを構成するには、負荷分散セットアップでのモニタの構成を参照してください。

SFTP を使用したサーバーの安全な監視

August 15, 2023

SSH ファイル転送プロトコル（SFTP）監視をサポートするために、ユーザースクリプト「nssftp.pl」が追加されました。これは、組み込みの NetScaler ADC ユーザーモニターの現在のリストにあり、/netscaler/モニターディレクトリにあります。SFTP モニタは、指定したユーザ名とパスワードを使用して、ファイルがサーバに存在するかどうかを確認します。

CLI を使用して SFTP を使用してセキュアモニタリングを設定するには

コマンドプロンプトで入力します。


```
add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
    string> -secure ( YES | NO )
<!--NeedCopy-->
```

例:

```
add monitor SFTP_MON USER -scriptname nssftp.pl -scriptargs "file=
    example.txt;user=sam;password=sam_passwd"
<!--NeedCopy-->
```

GUI を使用して **SFTP** によるセキュアモニタリングを設定するには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動し、[タイプ] に [ユーザー] を指定します。
2. 「特殊パラメータ」の「スクリプト名」で、「nssftp.pl」を選択します。
3. スクリプト引数を指定します。

安全なモニターで **SSL** パラメーターを設定する

August 15, 2023

> 重要
>
> この機能は、新しいデフォルトプロファイルでのみサポートされます。これらのプロファイルの詳細については、「拡張 SSL プロファイルインフラストラクチャの概要」を参照してください。

モニタは、グローバル設定またはバインドされているサービスの設定のいずれかを継承します。モニターが SSL_BRIDGE などの非 SSL または非 SSL_TCP サービスにバインドされている場合、プロトコルバージョンや使用する暗号などの SSL 設定を使用してモニターを構成することはできません。そのため、デプロイメントでバックエンドサーバーを SSL ベースで監視する必要がある場合、監視は効果がありません。

SSL プロファイルをモニターにバインドすることで、バックエンドサーバーの SSL ベースの監視をより詳細に制御できます。SSL プロファイルには、SSL パラメータ、暗号バインディング、および ECC バインディングが含まれます。たとえば、SSL プロファイルでサーバー認証、暗号、およびプロトコルのバージョンを設定し、そのプロファイルをモニターにバインドできます。サーバ認証を実行するには、CA 証明書をモニタにバインドする必要もあります。クライアント認証を実行するには、クライアント証明書をモニタにバインドする必要があります。「bind lb monitor」コマンドの新しいパラメータにより、これを実現できます。

> 注

> SSL 設定は、セキュアモニターを追加した場合にのみ有効になります。また、SSL プロファイルタイプは **BackEnd** でなければなりません。

SSL プロファイルをサポートするモニタータイプ

SSL プロファイルは、次のモニタータイプにバインドできます。

- HTTP
- HTTP-ECV
- TCP
- TCP-ECV
- HTTP-INLINE

コマンドラインを使用してモニターを追加するときに **SSL** プロファイルを指定するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <type> -secure  YES   -sslprofile <string>

set lb monitor <monitorName> <type> -secure YES -sslprofile <string>
<!--NeedCopy-->
```

例:

```
add ssl profile prof1 -sslProfileType  BackEnd

add lb monitor mon1 HTTP -secure  YES -sslprofile prof1
<!--NeedCopy-->
```

コマンドラインを使用して証明書とキーのペアをモニターにバインドするには

コマンドプロンプトで入力します。

```
bind monitor  <monitor name> -certkeyName <string>  [(-CA  [-crlCheck (
    Mandatory | Optional ) | -ocspCheck ( Mandatory | Optional )]
<!--NeedCopy-->
```

SIP サービスの監視

August 15, 2023

NetScaler には、SIP サービスの監視に使用できる SIP-UDP モニターと SIP-TCP モニターという 2 つの組み込みモニターがあります。 SIP モニターは、SIP リクエストメソッドを SIP サービスに送信することにより、SIP モニターがバインドされている SIP サービスを定期的にチェックします。SIP サービスが応答コードを返した場合、モニターはそのサービスを UP とマークします。SIP サービスが応答しない、または正しく応答しない場合は、DOWN とマークされます。

パラメーター	指定
sipURI	SIP サーバの SIP アドレッシングスキーマ。
sipmethod	SIP サービスのプローブに使用される SIP 要求のタイプ。次の方法のいずれかを指定します。INVITE、OPTION（デフォルト）、登録
respcode	SIP サービスがプローブ要求に応答する SIP 応答コード。デフォルト:200。

RADIUS サービスの監視

August 15, 2023

NetScaler アプライアンスの RADIUS モニターは、サービスに認証要求を送信することで、バインド先の RADIUS サービスの状態を定期的にチェックします。RADIUS サーバは RADIUS モニタを認証し、応答を送信します。デフォルトでは、モニタは RADIUS サーバから応答コード 2（デフォルトの Access-Accept 応答）を受信することを想定しています。モニターが適切な応答を受信している限り、サービスは稼働中とマークされます。

注:RADIUS モニタは PAP タイプの認証のみをサポートします。

- クライアントの認証に成功すると、RADIUS サーバは Access-Accept 応答を送信します。デフォルトの access-accept 応答コードは 2 で、これがアプライアンスが使用するコードです。
- クライアントが正常に認証されない場合（ユーザー名、パスワード、または秘密鍵が一致しない場合など）、RADIUS サーバは Access-Reject 応答を送信します。デフォルトのアクセス拒否応答コードは 3 で、これはアプライアンスが使用するコードです。

パラメーター	指定
userName	RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 サーバーのユーザー名。このユーザー名はプローブで使用されます。

パラメーター	指定
パスワード	RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP サーバの監視に使用されるパスワード。
RadKey	RADIUS サーバがクライアント認証時に使用する共有秘密キー値。
radNASid	アクセス要求が行われたときにペイロードにカプセル化される NAS-ID。
radNASip	アクセス要求が行われたときにペイロードにカプセル化される IP アドレス。RadnaSip が構成されていない場合、NetScaler アプライアンスはマップされた IP アドレス（MIP）を NAS IP アドレスとして RADIUS サーバーに送信します。

RADIUS サービスを監視するには、バインド先の RADIUS サーバを次のように設定する必要があります。

1. モニタが認証に使用するクライアントのユーザ名とパスワードを RADIUS 認証データベースに追加します。
2. クライアントの IP アドレスと秘密キーを適切な RADIUS データベースに追加します。
3. アプライアンスが RADIUS パケットを RADIUS データベースに送信するために使用する IP アドレスを追加します。NetScaler ADC アプライアンスに複数のマッピング IP アドレスがある場合、またはサブネット IP アドレス（SNIP）を使用する場合は、すべての IP アドレスに同じ秘密キーを追加する必要があります。

   注意：アプライアンスで使用される IP アドレスが RADIUS データベースに追加されない場合、RADIUS サーバはすべてのパケットを廃棄します。

RADIUS サーバの状態を確認するように組み込みモニタを構成するには、ロードバランシングセットアップでのモニタの設定を参照してください。

RADIUS サーバーからのアカウンティング情報の配信を監視する

August 15, 2023

RADIUS アカウンティングモニタと呼ばれるモニタを構成して、認証、認可、アカウンティング（NetScaler AAA）に使用される RADIUS サーバーが想定どおりにアカウンティング情報を配信しているかどうかを判断できます。モニタのタイプは RADIUS_ACCOUNTING です。プローブは、/nsconfig/monitors/ ディレクトリにある nsbmradius.pl という名前の Perl スクリプトによって生成されます。このスクリプトは、アカウンティング要求プローブを連続して RADIUS サーバに送信します。プローブが成功したと見なされるのは、RADIUS アカウンティン

グサーバが Code フィールドが 5 に設定されたパケット（RFC 2866 によると、Accounting-Response パケット）で応答した場合だけです。

RADIUS アカウンティングモニタを設定する場合、秘密キーを指定する必要があります。オプションのパラメータを指定できます。各パラメータは、Acct-Status-Type や Framed-IP アドレスなどの RADIUS 属性を表します。これらの属性の詳細については、RFC 2865「リモート認証ダイヤルインユーザーサービス (RADIUS)」および RFC 2866「RADIUS アカウンティング」を参照してください。

コマンドラインインターフェイスを使用して **RADIUS** アカウンティングモニタを設定するには

コマンドプロンプトで次のコマンドを入力して、RADIUS アカウンティングモニターを構成し、構成を確認します。

```
add lb monitor <monitorName> RADIUS_ACCOUNTING [-userName <string>] {
 -password  }
  {
 -radKey  }
  [-radNASip <ip_addr>] [-radAccountType <positive_integer>] [-
     radFramedIP <ip_addr>] [-radAPN <string>] [-radMSISDN <string>] [-
     radAccountSession <string>]

show lb monitor <monitorName>
<!--NeedCopy-->
```

例

```
add lb monitor radAccntMon RADIUS_ACCOUNTING -radKey "8d#>9jr4rV)L7%a2-
   zW13sM"
<!--NeedCopy-->
```

DNS および **DNS-TCP** サービスの監視

August 15, 2023

NetScaler アプライアンスには、DNS サービスの監視に使用できる DNS と DNS-TCP の 2 つのビルトインモニターがあります。サービスにバインドされると、いずれかのモニターが DNS クエリを送信して、その DNS サービスの状態を定期的にチェックします。クエリは IPv4 または IPv6 アドレスに解決されます。次に、その IP アドレスが、設定したテスト IP アドレスのリストと照合されます。リストには、最大 5 つの IP アドレスを含めることができます。解決された IP アドレスがリストの少なくとも 1 つの IP アドレスと一致する場合、DNS サービスは稼働中としてマークされます。解決された IP がリストのどの IP アドレスとも一致しない場合、DNS サービスは停止中とマークされます。

パラメーター	説明
問い合わせ	監視対象の DNS サービスに送信された DNS クエリ (ドメイン名)。デフォルト値:「\ 007」DNS クエリが成功すると、サービスは UP としてマークされます。それ以外の場合は、DOWN とマークされます。リバースモニタの場合、DNS クエリが成功すると、サービスは DOWN としてマークされます。それ以外の場合は、UP とマークされます。応答が受信されない場合、サービスは DOWN としてマークされます。
クエリータイプ	送信される DNS クエリのタイプ。可能な値:Possible values: Address, Zone.、ゾーン。
IPAddress	DNS モニタリングプローブに対する応答に対してチェックされる IP アドレスのリスト。
IPv6	IP アドレスが IPv6 形式を使用する場合は、このチェックボックスをオンにします。

組み込み DNS または DNS-TCP モニタを構成するには、ロードバランシングセットアップでのモニタの構成を参照してください。

LDAP サービスの監視

August 15, 2023

NetScaler アプライアンスには、LDAP サービスの監視に使用できる組み込みモニターが 1 つあります。それが LDAP モニターです。認証して検索クエリを送信することで、バインド先の LDAP サービスを定期的にチェックします。検索が成功すると、サービスは UP とマークされます。LDAP サーバがエントリを見つけられない場合、LDAP モニタに障害メッセージが送信され、サービスに DOWN とマークされます。

LDAP モニタを設定して、クエリの送信時に実行する必要のある検索を定義します。Base DN パラメータを使用して、LDAP サーバーがテストクエリを開始する必要があるディレクトリ階層内の場所を指定できます。Attribute パラメーターを使用して、ターゲットエンティティの属性を指定できます。

注: 監視プローブは、NSIP アドレスから発信されます。

パラメーター	Specifies
ベース DN	LDAP 検索の開始元である LDAP モニターのベース名。LDAP サーバーがローカルで実行されている場合、base のデフォルト値は dc =netscaler、dc=com.
bindDN	LDAP モニターの BDN 名。
filter	LDAP モニタのフィルタ。クエリで filter パラメーターを使用して、結果の数を制限します。クエリでこのパラメーターを指定しない場合、フィルタはオブジェクトクラス全体に適用されます。これは、CPU 使用率が高くなるなど、コストのかかる操作になる可能性があります。
パスワード	LDAP サーバーの監視に使用するパスワード。
属性	LDAP モニタの属性。

組み込み LDAP モニタを設定するには、ロードバランシングセットアップでのモニタの設定を参照してください。

MySQL サービスの監視

August 15, 2023

NetScaler アプライアンスには、MySQL サービスの監視に使用できる組み込みモニターが 1 つあります。それが MySQL モニターです。検索クエリを送信することで、バインド先の MySQL サービスを定期的にチェックします。検索が成功すると、サービスは UP とマークされます。MySQL サーバーが応答しない場合や検索が失敗した場合、MySQL モニターに障害メッセージが送信され、サービスが DOWN とマークされます。

注：監視プローブは、NSIP アドレスから発信されます。

パラメーター	指定
データベース	MySQL モニターに使用されるデータベース。
SQL クエリー	MySQL モニターに使用される SQL クエリ。

組み込み MySQL モニターを構成するには、負荷分散セットアップでのモニタの設定を参照してください。

CLI を使用して MySQL モニターを設定するには

次のコマンドを入力します。


```
add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
    string>
<!--NeedCopy-->
```

例:

```
add lb monitor mysql1 USER -scriptName nsmysql.pl -scriptArgs "database
    =cloud;user=cloud;password=password;query=show tables from cloud"
<!--NeedCopy-->
```

SNMP サービスの監視

August 15, 2023

NetScaler アプライアンスには、SMNP サービスの監視に使用できる SNMP モニターという 1 つの組み込みモニターがあります。監視対象に設定したエンタープライズ ID (OID) のクエリを送信することで、バインド先のサービスの SNMP エージェントを定期的にチェックします。クエリが成功すると、サービスは UP とマークされます。SNMP サービスが指定した OID を検出すると、クエリは成功し、SNMP モニタはそのサービスを UP とマークします。OID が見つからない場合、クエリは失敗し、SNMP モニタはサービスダウンとマークします。

注: 監視プローブは、NSIP アドレスから発信されます。

パラメーター	指定
SNMPOID	SNMP モニターに使用される OID。
snmpCommunity	SNMP モニターに使用されるコミュニティ。
snmpThreshold	SNMP モニターに使用されるしきい値。
snmpVersion	負荷監視に使用される SNMP バージョン。可能な値:V1、V2。

組み込み SNMP モニタを設定するには、ロードバランシングセットアップでのモニタの設定を参照してください。

NNTP サービスの監視

August 15, 2023

NetScaler アプライアンスには、NNTP サービスの監視に使用できる組み込みモニターが 1 つあります。それが NNTP モニターです。サービスに接続し、指定したニュースグループの存在を確認することで、バインド先の NNTP

サービスを定期的にチェックします。ニュースグループが存在する場合、検索は成功し、サービスは UP とマークされます。NNTP サービスが応答しない場合、または検索が失敗した場合、そのサービスは DOWN とマークされます。

注：監視プローブは、NSIP アドレスから発信されます。

NNTP モニターは、オプションでニュースグループにテストメッセージを投稿するように構成することもできます。

パラメーター	指定
userName	RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 サーバーのユーザー名。このユーザー名はプローブで使用されます。
パスワード	RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP サーバの監視に使用されるパスワード。
グループ	Group name to be queried for NNTP monitor.

組み込み NNTP モニタを設定するには、ロードバランシングセットアップでのモニタの設定を参照してください。

POP3 サービスの監視

August 15, 2023

NetScaler アプライアンスには、POP3 サービスの監視に使用できる POP3 モニターという 1 つのビルトインモニターがあります。POP3 サーバーとの接続を開くことで、バインドされている POP3 サービスを定期的にチェックします。POP3 サーバーが設定された時間内に正しい応答コードで応答すると、サービスが UP とマークされます。POP3 サービスが応答しない場合、または正しく応答しない場合は、サービスを DOWN とマークします。

注：監視プローブは、NSIP アドレスから発信されます。

パラメーター	指定
ユーザー名	ユーザー名 POP3 サーバー。このユーザー名はプローブで使用されます。
パスワード	POP3 サーバーの監視に使用するパスワード。
scriptName	実行するスクリプトのパスと名前。
dispatcherIP	プローブの送信先となるディスパッチャーの IP アドレス。
ディスパッチャーポート	プローブの送信先であるディスパッチャのポート。

パラメーター	指定

組み込み POP3 モニターを構成するには、負荷分散セットアップでのモニタの構成を参照してください。

CLI を使用して POP3 モニターを構成するには

次のコマンドを入力します。

```
add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
   string>
<!--NeedCopy-->
```

例:

```
add lb monitor pop31 USER -scriptName nspop3.pl -scriptArgs "user=
   test@lbmon1.net;password=Freebsd123"

<!--NeedCopy-->
```

SMTP サービスの監視

August 15, 2023

NetScaler ADC アプライアンスには、SMTP サービス（SMTP モニター）を監視するために使用できるビルトインモニターがあります。モニタは、バインドされている SMTP サービスをチェックし、サーバが正常に動作していることを確認するための一連のハンドシェイクを実行します。SMTP サービスがハンドシェイクを正しく完了すると、モニタはサービスを UP とマークします。それ以外の場合は、SMTP サービスが応答しない、または正しく応答しない場合、サービスが DOWN とマークされます。

注：監視プローブは、NSIP アドレスから発信されます。

パラメーター	指定
scriptName	実行するスクリプトのパスと名前。
dispatcherIP	プローブの送信先となるディスパッチャーの IP アドレス。
ディスパッチャーポート	プローブの送信先であるディスパッチャのポート。

組み込み SMTP モニタを構成するには、ロードバランシングセットアップでのモニタの構成を参照してください。

RTSP サービスの監視

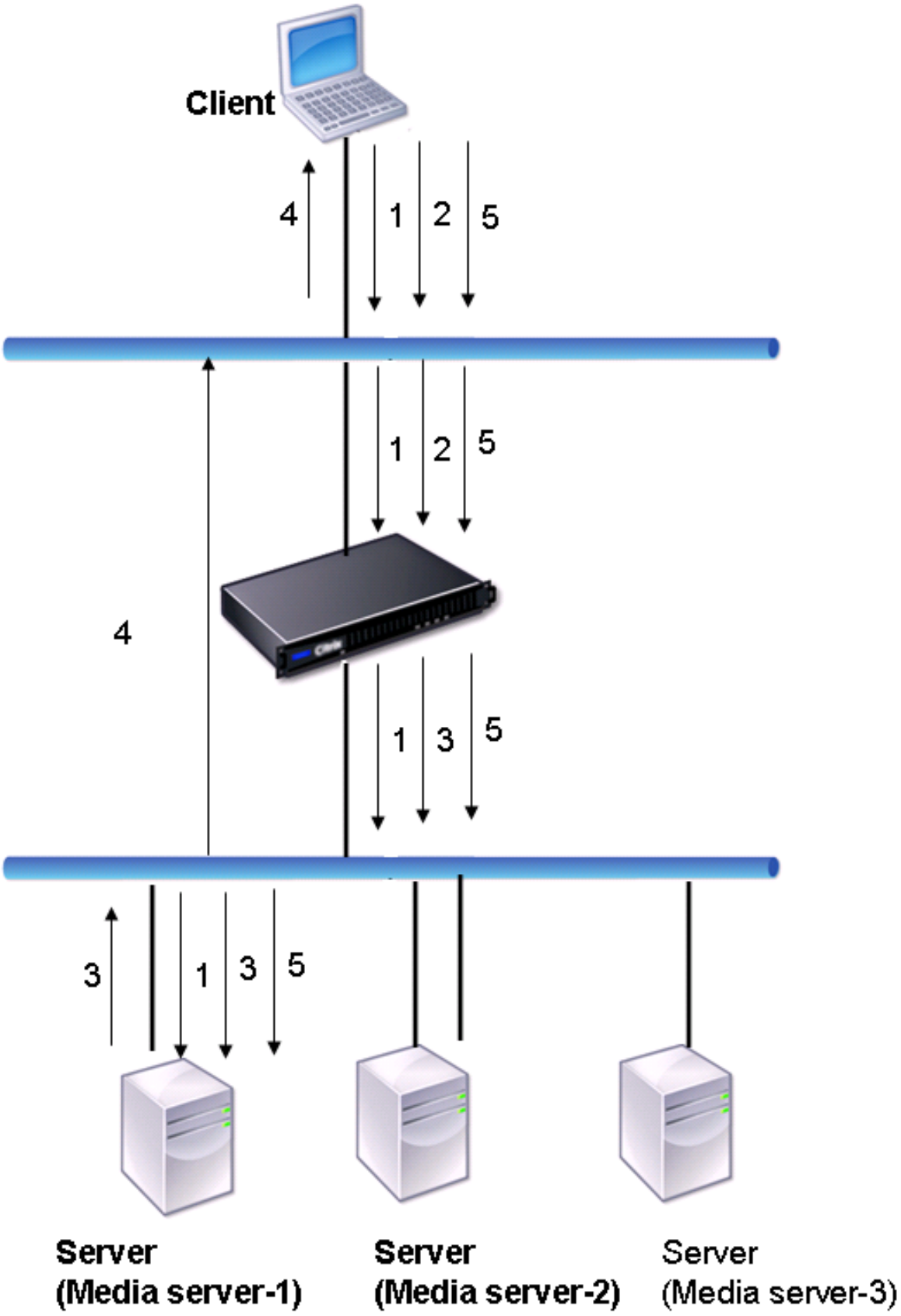
August 15, 2023

NetScaler アプライアンスには、RTSP サービスの監視に使用できる組み込みモニターが 1 つあります。それが RTSP モニターです。負荷分散された RTSP サーバーとの接続を開くことで、バインドされている RTSP サービスを定期的にチェックします。開く接続のタイプと期待される応答は、ネットワーク構成によって異なります。RTSP サービスが、設定した時間内に期待どおりに応答すると、サービスが UP とマークされます。サービスが応答しない、または正しく応答しない場合は、サービスを DOWN とマークします。

NetScaler アプライアンスは、NAT-Off と NAT-on の 2 つのトポロジーを使用して RTSP サーバーの負荷分散を行うように構成できます。RTSP サーバーは、アプライアンスをバイパスして応答をクライアントに直接送信します。ネットワークが使用するトポロジに応じて、RTSP サービスを異なる方法で監視するようにアプライアンスを設定する必要があります。アプライアンスは、NAT-off モードと NAT on モードの両方で、インラインモードでも非インラインモードでも導入できます。

NAT-off モードでは、アプライアンスはルーターとして動作します。クライアントから RTSP 要求を受信し、設定された負荷分散方式を使用して選択したサービスにルーティングします。負荷分散された RTSP サーバーに DNS で公開されている FQDN が割り当てられている場合、負荷分散されたサーバーは、アプライアンスをバイパスして応答をクライアントに直接送信します。次の図は、この構成を示しています。

図 1: RTSP in NAT-off Mode



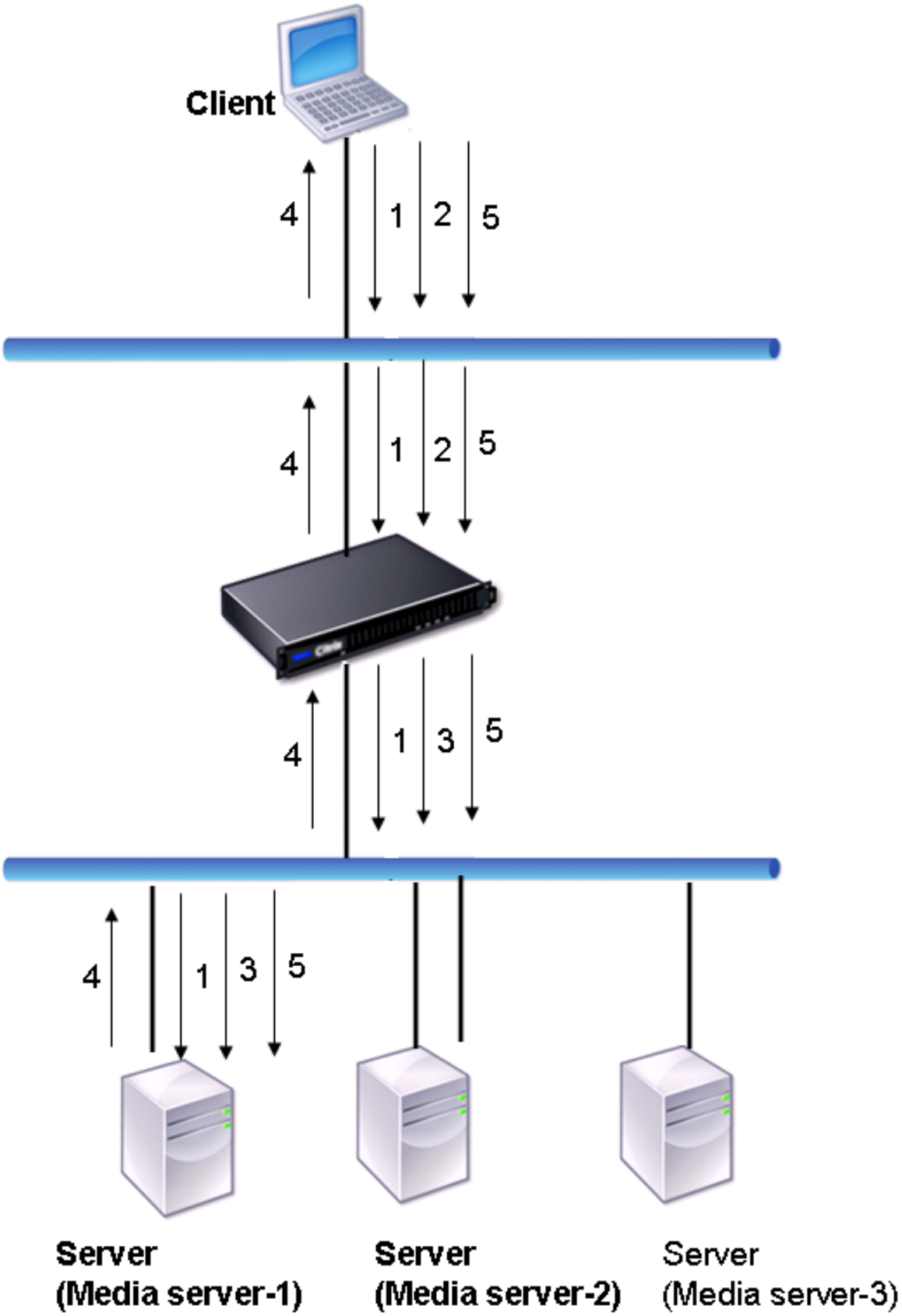
このシナリオでの要求と応答の流れは次のとおりです。

1. クライアントは DESCRIBE 要求をアプライアンスに送信します。アプライアンスは、設定された負荷分散方式を使用してサービスを選択し、要求を Media Server-1 にルーティングします。
2. クライアントは SETUP 要求をアプライアンスに送信します。DESCRIBE 要求で RTSP セッション ID が交換された場合、アプライアンスは RTSPSID パーシステンスを使用して要求をメディアサーバ 1 にルーティングします。SETUP リクエストで RTSP セッション ID が交換されると、アプライアンスは次のいずれかを実行します。
   - RTSP 要求が同じ TCP 接続で送信された場合、要求は永続性を維持したまま Media Server-1 にルーティングされます。
   - 要求が別の TCP 接続に着信した場合、設定されたロードバランシング方式を使用してサービスを選択し、永続性を維持せず、そのサービスに要求を送信します。これは、リクエストが別のサービスに送信される可能性があることを意味します。
3. Media Server-1 は、アプライアンスから SETUP 要求を受信し、RTSP 要求を処理するためのリソースを割り当てて、適切なセッション ID をクライアントに送信します。

   注:RTSP 接続はバイパスされるため、アプライアンスは RTSP 接続を識別するために NAT を実行しません。
4. 以降の要求では、クライアントはセッション ID を使用してセッションを識別し、制御メッセージをメディアサーバーに送信します。Media Server-1 は、再生、転送、巻き戻しなど、要求されたアクションを実行します。

NAT-on モードでは、アプライアンスはクライアントから RTSP 要求を受信し、設定された負荷分散方式を使用してそれらの要求を適切なメディアサーバーにルーティングします。次に、メディアサーバーは、次の図に示すように、アプライアンスを介してクライアントに応答を送信します。

図 2: NAT オンモードでの RTSP



このシナリオでの要求と応答の流れは次のとおりです。

1. クライアントは DESCRIBE 要求をアプライアンスに送信します。アプライアンスは、設定された負荷分散方式を使用してサービスを選択し、要求を Media Server-1 にルーティングします。
2. クライアントは SETUP 要求をアプライアンスに送信します。DESCRIBE 要求で RTSP セッション ID が交換された場合、アプライアンスは RTSPSID パーシステンスを使用して要求をメディアサーバー 1 にルーティングします。SETUP リクエストで RTSP セッション ID が交換されると、アプライアンスは次のいずれかを実行します。
    - RTSP 要求が同じ TCP 接続で送信された場合、要求は永続性を維持したまま Media Server-1 にルーティングされます。
    - 要求が別の TCP 接続に着信した場合、設定されたロードバランシング方式を使用してサービスを選択し、永続性を維持せず、そのサービスに要求を送信します。これは、リクエストが別のサービスに送信される可能性があることを意味します。
3. Media Server-1 は、アプライアンスから SETUP 要求を受信し、RTSP 要求を処理するためのリソースを割り当てて、適切なセッション ID をクライアントに送信します。
4. アプライアンスは NAT を実行して RTSP データ接続用のクライアントを識別し、RTSP 接続はアプライアンスを経由して正しいクライアントにルーティングされます。
5. 以降の要求では、クライアントはセッション ID を使用してセッションを識別し、アプライアンスに制御メッセージを送信します。アプライアンスは RTSPSID パーシステンスを使用して適切なサービスを識別し、要求を Media Server-1 にルーティングします。Media Server-1 は、再生、転送、巻き戻しなど、要求されたアクションを実行します。

RTSP モニタは RTSP プロトコルを使用して RTSP サービスの状態を評価します。RTSP モニターは RTSP サーバーに接続し、一連のハンドシェイクを実行してサーバーが正しく動作していることを確認します。

パラメーター	指定
rtspRequest	RTSP サーバーに送信される RTSP リクエスト文字列 (たとえば、OPTIONS*)。デフォルト値は 07 です。リクエストの長さは 163 文字を超えてはなりません。
respCode	サービスから期待される応答コードのセット。

RTSP モニタの構成手順については、ロードバランシングセットアップでのモニタの構成を参照してください。

ARP 要求の監視

August 15, 2023

NetScaler アプライアンスには、ARP リクエストの監視に使用できる組み込みモニターが 1 つあります。それが ARP モニターです。このモニターは、バインドされているサービスに ARP 要求を定期的に送信し、期待される応答を待ちます。期待どおりの応答を受け取ると、サービスを UP とマークします。応答がないか、間違った応答が返された場合は、サービスを DOWN とマークします。

ネットワーク層アドレスだけがわかっている場合、ARP は負荷分散サーバーのハードウェアアドレスを検索します。ARP は IPv4 と互換性があり、IP アドレスをイーサネット MAC アドレスに変換します。ARP モニタリングは IPv6 ネットワークには関係しないため、これらのネットワークではサポートされません。

ARP モニタには特別なパラメータはありません。

ARP モニタの設定手順については、ロードバランシングセットアップでのモニタの設定を参照してください。

Citrix Virtual Desktops Delivery Controller サービス監視

August 15, 2023

デスクトップ仮想化では、NetScaler ADC アプライアンスを使用して、Citrix Virtual Desktops 環境によってデプロイされた Citrix Virtual Desktops Delivery Controller サーバーの負荷分散を行うことができます。NetScaler ADC アプライアンスには、Citrix Virtual DesktopsDelivery Controller サーバーを監視するCITRIX-XD-DDCモニターという組み込みモニターがあります。ヘルスチェックに加えて、プローブが Citrix Virtual Desktops Delivery Controller サーバーの有効なユーザーによって送信されたかどうかを確認することもできます。

モニターは、XML メッセージ形式で Citrix Virtual Desktops Delivery Controller サーバーにプローブを送信します。サーバがサーバファームの ID を使用してプローブに応答した場合、プローブは成功したとみなされ、サーバのステータスは UP とマークされます。HTTP 応答に成功コードがない場合や、サーバーファームの ID が応答に含まれていない場合、プローブは失敗と見なされ、サーバーのステータスは DOWN とマークされます。

資格情報の検証オプションでは、モニターから Citrix Virtual Desktops Delivery Controller サーバーに送信するプローブ、つまりサーバー名のみを要求するか、ログイン資格情報も検証するかを決定します。

注:
CITRIX-XD-DDC モニターでユーザーの資格情報（ユーザー名、パスワード、ドメイン）が指定されているかどうかに関係なく、Citrix Virtual Desktops Delivery Controller サーバーは、資格情報を検証するオプションがモニターで有効になっている場合にのみユーザーの資格情報を検証します。

ウィザードを使用して Citrix Virtual Desktops サーバーの負荷分散を構成すると、CITRIX-XD-DDC モニターが自動的に作成され、Citrix Virtual Desktops Delivery Controller サービスにバインドされます。

コマンドラインインターフェイスを使用して、資格情報の検証オプションを指定して **XD-DDC** モニターを追加するには

コマンドプロンプトで次のコマンドを入力して XD-DDC モニターを追加し、構成を確認します。


```
add lb monitor <monitorName> <monitorType> -userName <userName> -
    password <password> -domain <domain_name> -validateCred YES

show lb monitor <monitorName>
<!--NeedCopy-->
```

例:

```
 > add lb monitor xdddcmon Citrix-xd-ddc -userName Administrator -
    password E12Dc35450a1 -domain dhop -validateCred YES
 Done
> show lb monitor xdddcmon
1) Name......:xdddcmon Type......:CITRIX-XD-DDC State......: ENABLED

Standard parameters:
Interval......:..5 sec...Retries......:..3
Response timeout......:..2 sec...Down time......:..30 sec
Reverse......:..NO...Transparent......:..NO
Secure......:..NO...LRTM......:..ENABLED
Action......:..Not applicable...Deviation......:..0 sec
Destination IP......:..Bound service
Destination port......:..Bound service
Iptunnel......:..NO
TOS......:..NO...TOS ID......:..0
SNMP Alert Retries......:..0...Success Retries......:..1
Failure Retries......:..0

Special parameters:
User Name..........:"Administrator"
Password..........:*****
DDC Domain.........: "dhop"
Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **XD-DDC** モニターで認証情報の検証オプションを指定するには

コマンドプロンプトで入力します。

```
set lb monitor <monitorName> <monitorType> -userName -password -domain
    <domain_name> -validateCred YES
<!--NeedCopy-->
```

例:

```
 set lb monitor XD_DDC_21.21.21.22_443_mn CITRIX-xd-ddc -userName
    Administrator -password D123S1R2A123 -domain dhop -validateCred YES
Done
<!--NeedCopy-->
```


構成ユーティリティを使用して **XD-DDC** モニターの資格情報の検証オプションを構成するには

[トラフィック管理] **>** [負荷分散] **>** [モニタ] に移動し、タイプ`Citrix-XD-DDC`のモニタを作成します。

Citrix StoreFront ストアの監視

August 15, 2023

Citrix StoreFront ストアのユーザーモニターを構成できます。モニターは、アカウントサービス、ディスカバリサービス、および認証エンドポイント（Citrix StoreFront Store が認証ストアの場合）を連続して調べることによって、StoreFront ストアの状態を決定します。これらのサービスのいずれかがプローブに応答しない場合、モニタープローブは失敗し、StoreFront ストアは DOWN としてマークされます。モニタは、バインドされたサービスの IP アドレスとポートにプローブを送信します。詳細については、「Citrix StoreFront ストアサービス API」を参照してください。

注: 監視プローブは、NSIP アドレスから発信されます。ただし、StoreFront サーバーのサブネットがアプライアンスのサブネットと異なる場合は、サブネット IP（SNIP）アドレスが使用されます。

リリース 10.1 ビルド 120.13 以降では、StoreFront モニターをサービスグループにバインドすることもできます。モニターはサービスグループの各メンバーにバインドされ、プローブはバインドされたメンバー（サービス）の IP アドレスとポートに送信されます。また、サービスグループの各メンバーはメンバーの IP アドレスを使用して監視されるようになったため、StoreFront モニターを使用して、サービスグループのメンバーとして追加された StoreFront クラスターノードを監視できるようになりました。

以前のリリースでは、StoreFront モニターが匿名ストアの認証を試みました。その結果、サービスが停止中としてマークされ、負荷分散仮想サーバーの URL を使用して Citrix Virtual Apps と Citrix Virtual Desktops を起動できなくなります。

ビルド 64.x から、プローブの順序が変更されました。モニターは、アカウントサービス、検出ドキュメント、認証サービスを連続的に調査して StoreFront ストアの状態を判断し、匿名ストアの認証をスキップします。

StoreFront モニターのホスト名パラメータは非推奨です。HTTP（デフォルト）と HTTPS のどちらを使用してモニタプローブを送信するかを決定するために、secure パラメータが使用されるようになりました。

HTTPS を使用するには、セキュリティオプションを Yes に設定します。

コマンドラインインターフェイスを使用して **StoreFront** モニターを作成するには

コマンドプロンプトで次のコマンドを入力して StoreFront モニターを構成し、構成を確認します。

```
1 add lb monitor <monitorName> STOREFRONT <string> -storeName <string> [-
      storefrontacctservice ( YES | NO )] -secure ( YES | NO )
2
```



```
show lb monitor <monitorName>
<!--NeedCopy-->
```

例

```
add lb monitor storefront_ssl STOREFRONT -storename myStore -
    storefrontacctservice YES -secure YES
<!--NeedCopy-->
```

構成ユーティリティを使用して **StoreFront** モニターを作成するには

トラフィック管理 **>** 負荷分散 **>** モニターに移動し、**STOREFRONT** タイプのモニターを作成します。

> 注
>
> StoreFront モニターの詳細については、StoreFront StoreFront のドキュメントを参照してください。

Oracle ECV サービスモニタリング

August 15, 2023

NetScaler の拡張コンテンツ検証（ECV）モニターを使用して Oracle データベースを監視できます。負荷分散された各データベースサーバーの状態をリアルタイムで追跡するには、Oracle ECV モニターを各サービスにバインドする必要があります。モニターは、SQL クエリの形式で定期的にプローブをサービスに送信してサービスをテストします。これをヘルスチェックと呼ぶこともあります。Oracle ECV モニタがプローブへの応答を適時に受け取り、設定した式が true と評価されると、サービスは稼働中とマークされます。指定された数のプローブに対してタイムリーに応答しなかったり、設定した式が false と評価された場合は、サービスを DOWN とマークします。
Netscaler Oracle ECV モニタリングは、21c までのすべての Oracle バージョンとすべてのパスワードベースの認証プロトコルをサポートしています。

サポートされていないセキュリティ機能

NetScaler Oracle ECV モニターは、パスワードベースの認証のみをサポートしています。セキュリティ関連の機能や機能をすべてサポートしているわけではありません。

次のセキュリティ機能はサポートされていません。

- データ暗号化 (SQLNet.Encryption_server= 必須)
- データ整合性 (SQLNET.crypto_checksum_server= 必須)
- ロング識別子 (O8L_LI)

- TLS 認証/暗号化
- ケルベロスや Radius などの外部認証サービス
- 圧縮
- オラクルウォレット

カスタムモニター

August 15, 2023

組み込みのモニタに加えて、カスタムモニタを使用してサービスの状態を確認することもできます。NetScaler ADC アプライアンスは、NetScaler ADC オペレーティングシステムに含まれるスクリプトに基づいて、いくつかのタイプのカスタムモニターを提供します。スクリプトを使用して、サービスに送信されるサービスまたはネットワークトラフィックの負荷に基づいて、サービスの状態を判断できます。カスタムモニタは、インラインモニタ、ユーザモニタ、および負荷モニタです。

これらのタイプのモニターでは、提供された機能を使用するか、独自のスクリプトを作成してこれらのスクリプトを使用して、モニタがバインドされているサービスの状態を判断できます。

HTTP インラインモニターを構成する

August 15, 2023

インラインモニタは、これらのサービスがクライアント要求を受信した場合にのみ、バインドされているサービスからの応答を分析し、プローブします。インラインモニタのタイプは HTTP-INLINE で、HTTP および HTTPS サービスでのみ設定できます。インラインモニタは、バインドされているサービスが UP であると判断し、送信される要求に対する応答をチェックします。クライアントリクエストがサービスに送信されない場合、インラインモニターは設定された URL を使用してサービスを調査します。

注: インラインモニターは HTTP または HTTPS のグローバルサーバー負荷分散 (GSLB) のリモートまたはローカルサービスにバインドできません。これらのサービスは実際の負荷分散された Web サーバーではなく仮想サーバーであるためです。

インラインモニターには、プローブが失敗したときのタイムアウト値と再試行回数があります。障害発生時に NetScaler アプライアンスが実行するアクションタイプは、次のいずれかを選択できます。

- なし。明示的なアクションは実行されません。サービスとモニターを表示でき、モニターには現在の連続エラー応答の数とチェックされた累積応答数が表示されます。
- ログ。ns/syslog にイベントを記録し、カウンタを表示します。

- ダウン。サービスをマークダウンし、トラフィックをサービスに転送しません。この設定により、サービスへの永続的な接続がすべて切断されます。このアクションはまた、イベントをログに記録し、カウンタを表示します。

サービスがダウンした後、設定されたダウンタイムはサービスが DOWN のままになります。ダウンタイムが経過すると、インラインモニタは設定された URL を使用してサービスをプローブし、サービスが再び利用可能かどうかを調べます。プローブが成功すると、サービスの状態が UP に変わります。トラフィックはサービスに送信され、監視は以前と同じように再開されます。

インラインモニタを構成するには、負荷分散セットアップでのモニタの構成を参照してください。

CLI を使用して HTTP インラインモニタを設定するには

次のコマンドを入力します。

```
add lb monitor <monitorName> <type> -respCode <int[-int]> -httpRequest
    <string> -resptimeout <integer> [<units>] -retries <integer> -
    downTime <integer> [<units>] -action <action>
<!--NeedCopy-->
```

例:

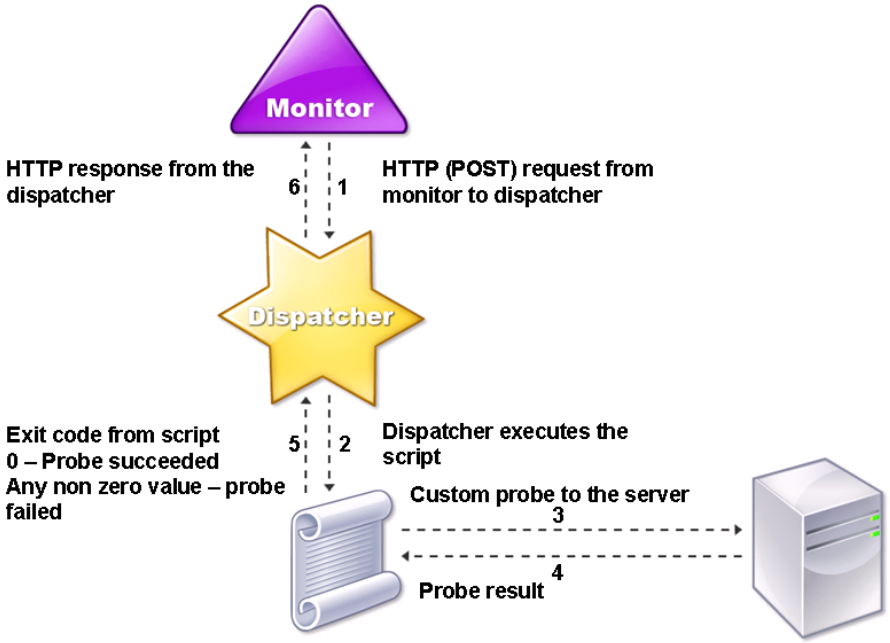
```
add lb monitor http_inline HTTP-INLINE -respCode 200 304 -httpRequest "
    HEAD /var/static/empty.htm" -resptimeout 4 -retries 1 -downTime 2 -
    action NONE
<!--NeedCopy-->
```

ユーザーモニターを理解する

August 29, 2023

ユーザーモニターは、カスタムモニターの範囲を拡張します。ユーザーモニターを作成して、NetScaler ADC アプライアンスがサポートしていないカスタマイズされたアプリケーションとプロトコルの健全性を追跡できます。次の図は、ユーザーモニターの動作を示しています。

図 1: ユーザーモニター



ユーザーモニターには、次のコンポーネントが必要です。

ディスパッチャー。モニタリング要求をリッスンするアプライアンス上のプロセス。ディスパッチャは、ループバック IP アドレス（127.0.0.1）およびポート 3013 上に置くことができます。ディスパッチャは、内部ディスパッチャとしても知られています。ディスパッチャは、Common Gateway Interface (CGI) をサポートする Web サーバーでも使用できます。このようなディスパッチャは、外部ディスパッチャとも呼ばれます。これらは、.NET スクリプトなど、FreeBSD 環境では動作しないカスタムスクリプトに使用されます。

> 注:
>
> モニターで「セキュア」オプションを有効にして、HTTP の代わりに HTTPS を使用するようにモニターとディスパッチャを設定し、外部ディスパッチャとして設定できます。ただし、内部ディスパッチャは HTTP のみを認識し、HTTPS を使用することはできません。

高可用性セットアップでは、ディスパッチャはプライマリとセカンダリの両方の NetScaler ADC アプライアンスで実行されます。ディスパッチャは、セカンダリアプライアンスで非アクティブのままです。

[スクリプト]。スクリプトは、負荷分散サーバーにカスタムプローブを送信し、ディスパッチャに応答コードを返すプログラムです。スクリプトはディスパッチャに任意の値を返すことができますが、プローブが成功した場合、スクリプトは値 0 を返さなければなりません。ディスパッチャは、その他の値をプローブ障害と見なします。

NetScaler ADC アプライアンスには、一般的に使用されるプロトコルのサンプルスクリプトがバンドルされていま

す。これらのスクリプトは、/nsconfig/monititors ディレクトリに存在します。スクリプトを追加する場合は、そこにスクリプトを追加します。既存のスクリプトをカスタマイズするには、新しい名前でコピーを作成し、それを修正します。

重要:

- NetScaler ADC リリース 13.0 ビルド 41.20 以降、`nsntlm-lwp.pl`スクリプトを使用して安全な NTLM サーバーを監視するためのモニターを作成できます。
- リリース 10.1 ビルド 122.17 以降、ユーザモニタ用のスクリプトファイルが新しい場所にあります。

  MPX または VPX 仮想アプライアンスをリリース 10.1 ビルド 122.17 以降にアップグレードする場合、変更点は次のとおりです。

  - /nsconfig/monitors/ に conflicts という名前の新しいディレクトリが作成され、以前のビルドのすべての組み込みスクリプトがこのディレクトリに移動されます。
  - すべての新しい組み込みスクリプトは、/netscaler/monitors/ ディレクトリで利用できます。すべてのカスタムスクリプトは、/nsconfig/monitors/ ディレクトリにあります。
  - 新しいカスタムスクリプトを /nsconfig/monitors/ ディレクトリに保存します。
  - アップグレードの完了後、カスタムスクリプトが作成され、組み込みスクリプトと同じ名前で /nsconfig/monitors/ ディレクトリに保存されると、/netscaler/monitors/ ディレクトリ内のスクリプトが優先されます。カスタムスクリプトは実行されません。

  リリース 10.1 ビルド 122.17 以降で仮想アプライアンスをプロビジョニングする場合、変更内容は次のとおりです。

  - すべての組み込みスクリプトは、/netscaler/monitors/ ディレクトリにあります。
  - /nsconfig/monitors/ ディレクトリが空です。
  - カスタムスクリプトを作成する場合は、/nsconfig/monitors/ ディレクトリに保存する必要があります。

スクリプトが正しく機能するには、次の手順を実行します。

- スクリプト名の最大文字数は 63 文字以下でなければなりません。
- スクリプトに指定できるスクリプト引数の最大数は 512 を超えないようにしてください。
- パラメータースクリプト引数に指定できる最大文字数は 639 文字以下でなければなりません。

スクリプトをデバッグするには、CLI から nsumon-debug.pl スクリプトを使用してスクリプトを実行する必要があります。nsumon-debug.pl スクリプトの引数として、スクリプト名 (引数付き)、IP アドレス、およびポートを使用します。ユーザーは、nsumon-debug.pl スクリプトのスクリプト名、IP アドレス、ポート、タイムアウト、およびスクリプト引数を使用する必要があります。

CLI で、次のように入力します。

```
1 nsumon-debug.pl <scriptname> <IP> <port> <timeout> <partitionID> [
     scriptarguments][is_secure]
2 <!--NeedCopy-->
```


重要: リリース 10.5 ビルド 57.x 以降、ユーザーモニター用の 11.0 スクリプトファイルは IPv6 アドレスをサポートし、次の変更を含めました。

- 次のプロトコルでは、IPv6 サポート用の新しい`pm files`が含まれています。
  - RADIUS
  - NNTP
  - POP3
  - SMTP
- /netscaler/monitors/ の次のサンプルスクリプトが IPv6 サポート用に更新されました。
  - nsbmradius.pl
  - nsldap.pl
  - nsnntp.pl
  - nspop3 nssf.pl
  - nssnmp.pl
  - nswi.pl
  - nstftp.pl
  - nssmtp.pl
  - nsrdp.pl
  - nsntlm-lwp.pl
  - nsftp.pl
  - nsappc.pl

  リリース 10.5 ビルド 57.x または 11.0 にアップグレードした後、IPv6 サービスで既存のカスタムスクリプトを使用する場合は、の更新されたサンプルスクリプトで提供された変更で既存のカスタムスクリプトを更新してください。/netscaler/monitors/.

> 注:
>
> サンプルスクリプト nsmysql.pl は IPv6 アドレスをサポートしていません。IPv6 サービスが nsmysql.pl を使用するユーザーモニターにバインドされている場合、プローブは失敗します。

- IPv6 アドレスをサポートするために、次の LB モニタタイプが更新されました。
  - USER
  - SMTP
  - NNTP

- LDAP
- SNMP
- POP3
- FTP_EXTENDED
- StoreFront
- APPC
- CITRIX_WI_EXTENDED

これらの LB モニタタイプのいずれかを使用するカスタムスクリプトを作成する場合は、カスタムスクリプトに IPv6 サポートを含めるようにしてください。IPv6 サポート用のカスタムスクリプトで行う必要がある変更については、/netscaler/monitors/ の関連するサンプルスクリプトを参照してください。

サーバーのステータスを追跡するために、モニターは設定されたディスパッチャに HTTP POST 要求を送信します。この POST リクエストには、サーバーの IP アドレスとポート、および実行する必要があるスクリプトが含まれます。ディスパッチャは、ユーザー定義パラメータ (存在する場合) を使用して、スクリプトを子プロセスとして実行します。次に、スクリプトはプローブをサーバーに送信します。スクリプトは、プローブのステータス (応答コード) をディスパッチャに送信します。ディスパッチャは、レスポンスコードを HTTP レスポンスに変換し、モニタに送信します。HTTP 応答に基づいて、モニタはサービスをアップまたはダウンとしてマークします。

NetScaler ADC アプライアンスは、エラーメッセージを /var/nslog/nsumond.log ユーザーモニタープローブが失敗した場合のファイル。これらの詳細なエラーメッセージは、GUI および`show service`/`service group`コマンドの CLI に表示されます。

次の表は、ユーザーモニターと、考えられる失敗の原因の一覧です。

ユーザーモニターの種類	プローブ失敗の理由
SMTP	モニタがサーバへの接続を確立できない。
NNTP	モニタがサーバへの接続を確立できない。
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。 モニタが NNTP グループを見つけられない。
LDAP	モニタがサーバへの接続を確立できない。
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。 モニタが LDAP サーバにバインドできない。
	モニタが LDAP サーバでターゲットエンティティのエントリを見つけない。

ユーザーモニターの種類	プローブ失敗の理由
FTP	サーバーへの接続がタイムアウトします。
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。 ログオンが失敗する。
	モニタがサーバ上のファイルを検出できない。
POP3	モニタがデータベースへの接続を確立できない。
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。 ログオンが失敗する。
	SQL クエリの準備が失敗します。
	SQL クエリの実行が失敗します。
SNMP	モニタがデータベースへの接続を確立できない。
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。 ログオンが失敗する。
	モニタが SNMP セッションを作成できない。
	モニタがオブジェクト識別子を見つけられない。
	モニタのしきい値の設定は、モニタの実際のしきい値以上です。
RDP (Windows ターミナルサーバー)	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。 モニタがソケットを作成できない。
	バージョンの不一致。
	モニタが接続を確認できない。

次のコマンドを使用して、CLI からログファイルを表示できます。

```
1 > shell
2 root@ns# cat /var/nslog/nsumond.log
3 root@ns# exit
4 >
5 <!--NeedCopy-->
```

このコマンドは BSD シェルを開き、ログファイルを画面に表示し、BSD シェルを閉じて CLI に戻ります。

NetScaler ADC リリース 13.0 ビルド 52.X より前の`show service/service group`コマンドでは、ユ

ーザーモニターのプローブの失敗の原因として「プローブに失敗しました」という一般的なエラーメッセージが表示されました。

例:

```
show service ftp

Monitor Name: mon2
State: UNKNOWN  Weight: 1   Passive: 0
Probes: 3    Failed [Total: 0 Current: 0]
Last response: Failure - Probe failed.
Response Time: 1071.838 millisec
<!--NeedCopy-->
```

NetScaler ADC リリース 13.0 ビルド 52.X 以降では、`show service/service group`コマンドがユーザーユーザーモニタープローブの失敗の実際の原因を表示します。

例:

```
show service ftp

Monitor Name: mon2
State: DOWN   Weight: 1   Passive: 0
Probes: 729   Failed [Total: 726 Current: 726]
Last response: Failure - Login failed.
Response Time: 8000.0 millisec
<!--NeedCopy-->
```

ユーザーモニターには、タイムアウト値とプローブ障害の再試行回数もあります。ユーザーモニターは、ユーザーモニター以外のモニターでも使用できます。CPU 使用率が高い場合、非ユーザモニタを使用すると、サーバ障害をより迅速に検出できます。

CPU 使用率が高いときにユーザモニタプローブがタイムアウトしても、サービスの状態は変更されません。

Example1:

```
add lb monitor <name> USER  -scriptname <script-name> -resptimeout 5
    seconds
<!--NeedCopy-->
```

> 注:
>
> スクリプト可能なモニタの場合、応答タイムアウトは、予想されるタイムアウト + 1 秒に等しい値に設定する必要があります。たとえば、タイムアウトが 4 秒になると予想される場合、応答タイムアウトを 5 秒に設定します。

Example2:

```
add lb monitor <name> USER  -scriptname <script-name> -scriptargs <
    Arguments> -secureargs <Arguments>
<!--NeedCopy-->
```


> 注:
>
> スクリプトに関連する機密データには、scriptargsパラメーターの代わりにsecureargsパラメーターを使用することをお勧めします。

ユーザーモニターを使用して **Web** サイトを確認する方法

August 15, 2023

特定の HTTP コードを使用して HTTP サーバーによって報告される特定の Web サイトの問題をチェックするように、ユーザーモニターを構成できます。次の表は、このユーザーモニターが期待する HTTP 応答コードの一覧です。

HTTP レスポンスコード	意味
200-成功	プローブ成功。
503-サービスは利用できません	プローブに障害が発生しました。
404-見つかりません	スクリプトが見つからないか、実行できません。
500-内部サーバエラー	ディスパッチャーの内部エラー/リソース制約 (メモリ不足、接続が多すぎる、予期しないシステムエラー、プロセスが多すぎる)。サービスには DOWN のマークが付いていません。
400-不正なリクエスト	HTTP リクエストの解析中にエラーが発生しました。
502-不正なゲートウェイ	スクリプトの応答をデコード中にエラーが発生しました。

HTTP 用のユーザーモニターは、次のパラメーターを使用して構成します。

パラメーター	指定
scriptName	実行するスクリプトのパスと名前。
scriptArgs	POST データに追加される文字列。これらはそのままリクエストにコピーされます。
dispatcherIP	プローブの送信先となるディスパッチャーの IP アドレス。
ディスパッチャーポート	プローブの送信先であるディスパッチャのポート。
localfileName	ローカルシステム上の監視スクリプトファイルの名前。

パラメーター	指定
destPath	アップロードされたローカルファイルが保存されている NetScaler ADC アプライアンス上の特定の場所。

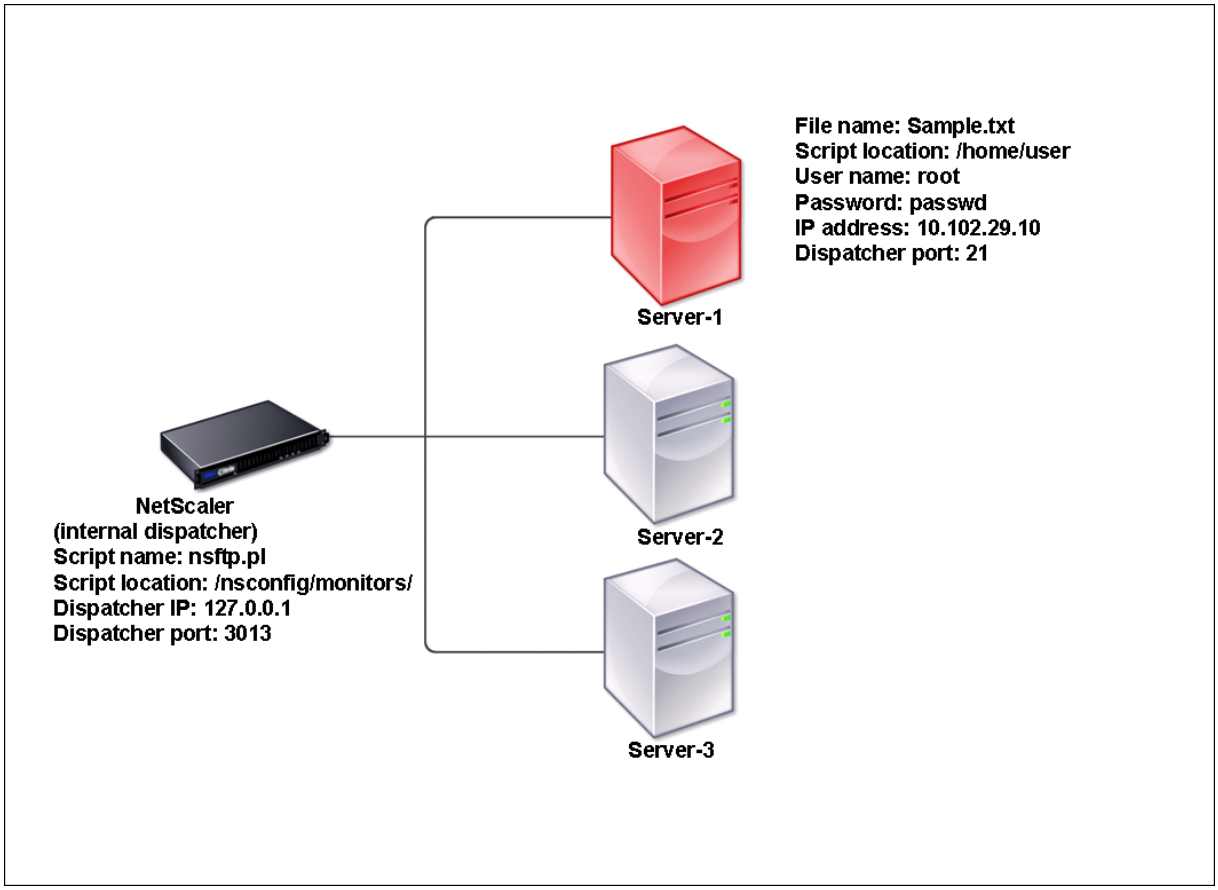
HTTP を監視するユーザーモニターを作成するには、負荷分散セットアップでのモニタの構成を参照してください。

内部ディスパッチャーを理解する

August 15, 2023

内部ディスパッチャーでカスタムユーザーモニターを使用できます。サーバー上のファイルの有無に基づいてサーバーの状態を追跡する必要がある場合を考えてみましょう。次の図は、このシナリオを示しています。

図 1：内部ディスパッチャを使用したユーザーモニタの使用

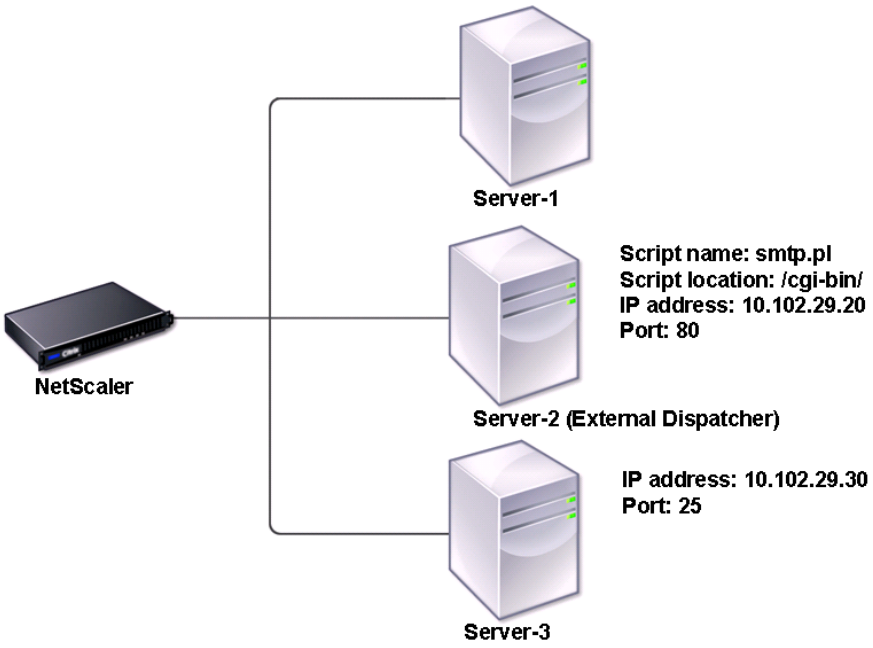


考えられる解決策は、サーバーとの FTP セッションを開始し、ファイルの存在をチェックする Perl スクリプトを使

用することです。その後、Perl スクリプトを使用するユーザーモニターを作成できます。NetScaler アプライアンスの/nsconfig/monitors/ディレクトリには、このような Perl スクリプト（nsftp.pl）が含まれています。

ユーザーモニターは外部ディスパッチャーで使用できます。別のサーバー上の SMTP サービスの状態に基づいてサーバーの状態を追跡する必要がある場合を考えてみましょう。このシナリオを次の図に示します。

図 2：外部ディスパッチャを使用したユーザーモニタの使用



可能な解決策は、サーバー上の SMTP サービスの状態をチェックする Perl スクリプトを作成することです。その後、Perl スクリプトを使用するユーザーモニターを作成できます。

ユーザーモニターの設定

August 15, 2023

ユーザーモニターは、NetScaler ADC アプライアンスがサポートしていないカスタムアプリケーションとプロトコルの健全性を追跡します。これは、カスタムモニターの拡張スコープです。ユーザモニタを設定するには、次の手順を実行する必要があります。

- バインドされたサービスを監視できるスクリプトを記述します。

- スクリプトを NetScaler ADC アプライアンスの/nsconfig/monitorsディレクトリにアップロードします。
- スクリプトに実行権限を与えます。

モニタタイプがアプライアンスでサポートされていないプロトコルの場合は、タイプ **USER** のモニタを使用する必要があります。ユーザーモニターは Perl および Bash タイプのスクリプトのみをサポートします。Python スクリプトはサポートしていません。

> 注
>
> モニタープローブは NSIP アドレスから発信されます。モニタタイプ **USER** に設定されたscriptargsは、実行構成ファイルと ns.conf ファイルに表示されます。

モニターの詳細については、「モニターの構成」を参照してください。

CLI を使用してユーザモニタを設定するには

コマンドプロンプトで入力します。

```
1 add lb monitor <monitorName> USER -scriptname <NameOfScript> -
     scriptargs <Arguments> -secureargs <Arguments>
2 <!--NeedCopy-->
```

Example1:

```
1 add monitor Monitor-User-1 USER -scriptname nsftp.pl -scriptargs "file
     =/home/user/
2 sample.txt;user=root;password=passwd"
3 <!--NeedCopy-->
```

Example2:

```
1 add monitor Monitor-User-1 USER -scriptname nsftp.pl -scriptargs "file
     =/home/user/
2 sample.txt -secureargs "user=root;password=passwd"
3 <!--NeedCopy-->
```

> 注
>
> secureargsパラメータは、スクリプト引数をプレーンテキスト形式ではなく暗号化された形式で格納します。スクリプトに関連する機密データ（ユーザー名やパスワードなど）には、scriptargs パラメーターの代わりにパラメーターsecureargsを使用することをお勧めします。両方のパラメータを一緒に使用する場合、-scriptnameで指定されたスクリプトは、<scriptargs> <secureargs>の順序で引数を受け入れる必要があります。最初のいくつかの引数を<scriptargs>パラメータに指定し、残りの引数を<secureargs>パラメータに指定します。つまり、引数に定義された順序を維持します。Secure 引数は

> 内部ディスパッチャにのみ適用されます。外部ディスパッチャーを使用する場合は、スクリプト内の脆弱なデータを保護することをお勧めします。

例 **3:**

scriptargsパラメータに「a=b; c=d; e=f」という引数を設定しているとします。

```
add monitor mon1 USER -scriptargs "a=b;c=d;e=f"
<!--NeedCopy-->
```

scriptargsパラメータの代わりにsecureargsパラメータを使用する場合は、次の操作を行います。

- scriptargsパラメータを無効にします。
- secureargsパラメータの下にすべての引数を指定します。

```
set monitor mon1 USER -scriptargs "" -secureargs "a=b;c=d;e=f"
<!--NeedCopy-->
```

GUI を使用してユーザモニタを設定するには

1. [トラフィック管理] **>** [負荷分散] **>** [モニタ] に移動し、[追加] をクリックします。
2. [モニターの作成] ページで、次の操作を行います。
   - モニタータイプとして **USER** を選択します。
   - ドロップダウンメニューからスクリプトを選択するか、独自のスクリプトをアップロードします。
   - [スクリプト引数] フィールドと [** セキュア引数 **] フィールドに適切な値を入力します。
   - [作成] をクリックします。

   ユーザーモニターが作成されます。

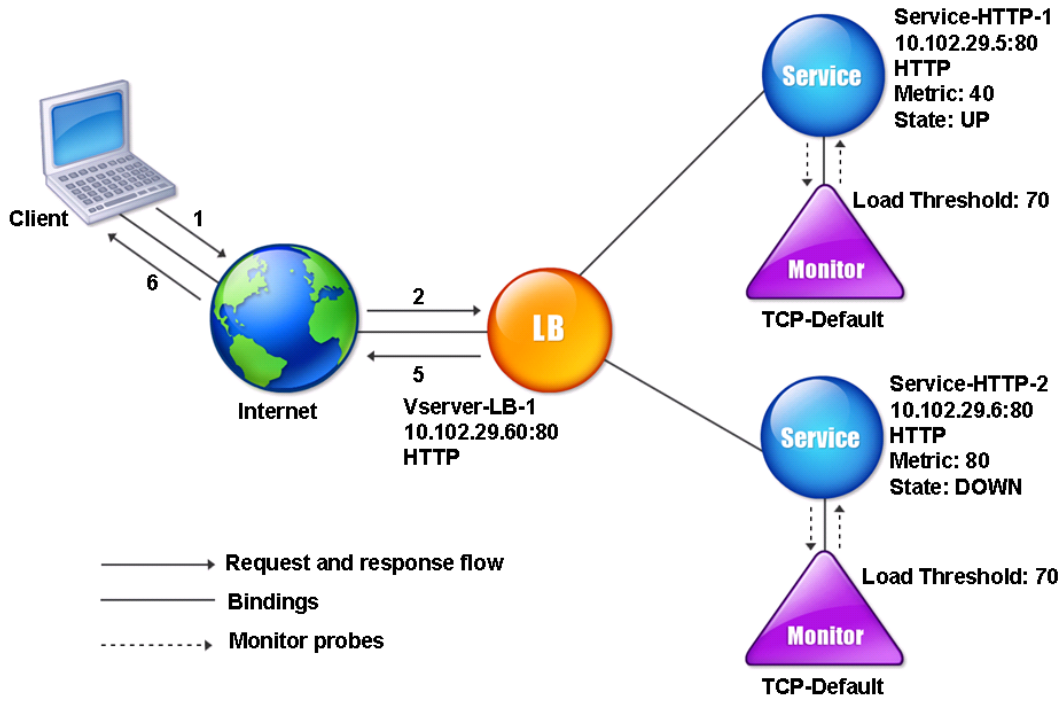
負荷モニターを理解する

August 15, 2023

負荷モニターは、SNMP ポーリングされた OID を使用して負荷を計算します。ロードモニターは、バインド先のサービスの IP アドレス (宛先 IP アドレス) をポーリングに使用します。メトリックスの OID を指定して、SNMP クエリをサービスに送信します。メトリックには、CPU、メモリ、またはサーバー接続数を指定できます。サーバーはクエリにメトリック値で応答します。応答のメトリック値は、しきい値と比較されます。NetScaler アプライアンスは、メトリックがしきい値未満の場合にのみ、サービスを負荷分散の対象と見なします。負荷値が最も小さいサービスが最初に考慮されます。

次の図は、基本負荷分散の設定で説明されている基本的な負荷分散設定で説明されているサービスに対して構成された負荷モニタを示しています。

図 1：負荷モニタの動作



注：負荷モニターは、サービスの状態を決定しません。これによって、アプライアンスは負荷分散のためのサービスを検討することしかできません。

負荷モニターを構成したら、モニターが使用するメトリックを設定する必要があります。負荷評価では、負荷モニターはメトリックと呼ばれるサーバーパラメーターを考慮します。これらのパラメーターは、アプライアンス構成のメトリックテーブル内で定義されます。メトリックテーブルには次の 2 つのタイプがあります。

- ローカル。デフォルトでは、このテーブルはアプライアンスに存在します。接続、パケット、応答時間、帯域幅の 4 つのメトリックで構成されています。アプライアンスはサービスのこれらのメトリックを指定し、これらのサービスについては SNMP クエリは送信されません。これらの指標は変更できません。
- カスタム。ユーザー定義のテーブル。各メトリックは OID に関連付けられています。

デフォルトでは、アプライアンスは次のテーブルを生成します。

- NetScaler
- RADWARE
- CISCO-CSS
- LOCAL
- FOUNDRY
- ALTEON

次の表に示すように、アプライアンスで生成されたメトリックテーブルを追加することも、自分で選択したテーブルを追加することもできます。メトリックテーブルの値は例としてのみ提供されています。実際のシナリオでは、メトリックの実際の値を検討してください。

指標名	OID	重要度	しきい値
CPU	1.2.3.4	2	70
メモリ	4.5.6.7	3	80
接続	5.6.7.8	4	90

1 つ以上のメトリックの負荷を計算するには、各メトリックスに重みを割り当てます。デフォルトの重みは 1 です。重みは各指標に与えられた優先順位を表します。重量が大きい場合は、優先度が高くなります。アプライアンスは SOURCEIPDESTIP ハッシュアルゴリズムに基づいてサービスを選択します。

各指標のしきい値を設定することもできます。このしきい値により、サービスのメトリック値がしきい値より小さい場合に、アプライアンスは負荷分散の対象となるサービスを選択できます。しきい値によって各サービスの負荷も決まります。

負荷モニターを構成する

August 15, 2023

ロードモニタを設定するには、最初にロードモニタを作成します。モニターの作成手順については、モニターの作成を参照してください。次に、メトリックテーブルを選択するか、作成して、サーバーの状態を決定するメトリックのセットを定義し、（メトリックテーブルを作成した場合）各メトリックをメトリックテーブルにバインドします。

コマンドラインインターフェイスを使用してメトリックテーブルを作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add lb metricTable <metricTableName>

bind lb metricTable <metricTableName> <metric> <SNMPOID>
<!--NeedCopy-->
```

例:

```
add lb metricTable Table-Custom-1

bind lb metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5 11
<!--NeedCopy-->
```


設定ユーティリティを使用してメトリックテーブルを作成し、メトリックをそのテーブルにバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [メトリックテーブル] に移動し、メトリックテーブルを作成します。
2. メトリックをバインドするには、[バインド] をクリックし、メトリックと SNMP OID を指定します。

メトリックテーブルからメトリックのバインドを解除する

August 15, 2023

メトリクスを変更する必要がある場合、またはメトリクステーブルを完全に削除したい場合は、メトリクステーブルからメトリクスをアンバインドできます。

コマンドラインインターフェイスを使用してメトリクステーブルからメトリクスをアンバインドするには

コマンドプロンプトで入力します。

```
unbind lb metricTable <metricTable> <metric>
<!--NeedCopy-->
```

例**:**

```
unbind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5
<!--NeedCopy-->
```

設定ユーティリティを使用してメトリクステーブルからメトリクスをアンバインドするには

1. [トラフィック管理] > [ロードバランシング] > [メトリックテーブル] に移動します。
2. 指標テーブルを開き、指標を選択して [削除] をクリックします。

名前やタイプなど、設定されているすべてのメトリックテーブルの詳細を表示して、メトリックテーブルが内部的なものか、作成および構成されたものかを判断できます。

サービスのリバース監視を構成する

August 15, 2023

リバースモニターは、プローブ基準が満たされている場合はサービスを DOWN としてマークし、満たさない場合は UP とマークします。たとえば、プライマリサービスがダウンしているときだけバックアップサービスがトラフィッ

クを受信するようにしたい場合は、リバースモニターをセカンダリサービスにバインドして、プライマリサービスをプローブするように設定できます。

NetScaler アプライアンスは、次のリバースモニターをサポートしています。

- HTTP
- ICMP
- TCP (リリース 11.1 ビルド 49.x 以降)

サービスの **HTTP** リバースモニタリングの設定

次の表は、サービスの HTTP ダイレクトモニタリングとリバースモニタリングの条件を示しています。

条件	ダイレクト	リバース
接続が確立されていません。	失敗	失敗
HTTP 応答コードはプローブの仕様と一致します。	成功	失敗
HTTP 応答コードがプローブの仕様と一致しません。	失敗	成功
プローブがタイムアウトしました。	失敗	失敗

CLI を使用してサービスの **HTTP** リバースモニタリングを設定するには

コマンドプロンプトで入力します。

```
add lb monitor <Monitor_Name> HTTP -respCode 200 -httpRequest "HEAD /"
   -destIP <Primary_Service_IP_Address> -destPort 80 -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

サービスの **ICMP** リバースモニタリングの設定

次の表は、サービスの ICMP 直接監視と逆監視の条件を示しています。

条件	ダイレクト	リバース
ICMP エコー応答が受信されました。	成功	失敗
プローブがタイムアウトしました。	失敗	成功

CLI を使用してサービスの **ICMP** リバースモニタリングを設定するには

コマンドプロンプトで入力します。

```
add lb monitor <Monitor_Name> PING -destIP <Primary_Service_IP_Address>
    -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

サービスの **TCP** リバースモニタリングの設定

ダイレクト TCP モニターがモニタープローブへの応答として RESET を受信すると、そのサービスは DOWN とマークされます。ただし、リバース TCP モニターが RESET 応答を受け取ると、プローブは成功したと見なされ、サービスは UP とマークされます。

次の表は、サービスの TCP リバースモニタリングの条件を示しています。

条件	ダイレクト	リバース
TCP 接続が確立されます。	成功	失敗
プローブがタイムアウトしました。	失敗	失敗
プローブへの応答は RESET です。	失敗	成功

CLI を使用してサービスの **TCP** リバースモニタリングを設定するには

コマンドプロンプトで入力します。

```
add lb monitor <Monitor_Name> TCP  - destip <Primary_Service_IP_Address>
    -destport <primary_service_port>  - reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

GUI を使用してリバースモニタリングを設定するには

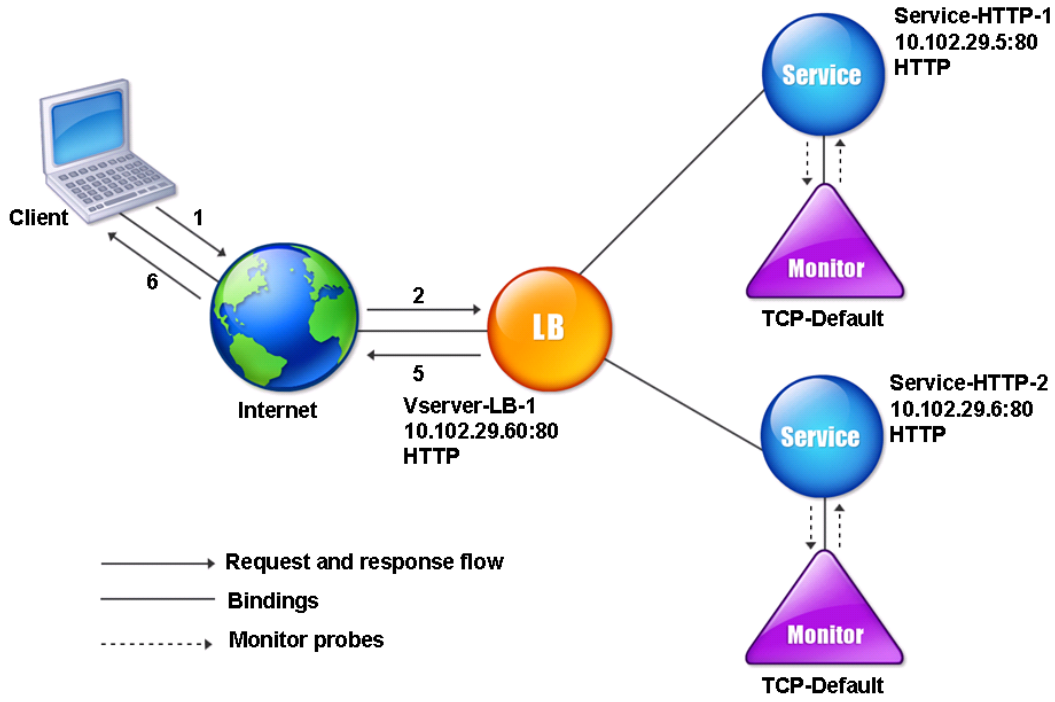
1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. **HTTP**、**ICMP**、または **TCP** モニターを作成し、「リバース」を選択します。

負荷分散セットアップでモニターを構成する

August 15, 2023

Web サイトでモニターを構成するには、まずビルトインモニターを使用するか、独自のモニターを作成するかを決定します。モニターを作成する場合は、組み込みモニターに基づいてモニターを作成するか、作成したスクリプトを使用してサービスをモニターするカスタムモニターを作成するかを選択できます。カスタムモニターの作成の詳細については、「カスタムモニター」を参照してください。モニターを選択または作成したら、それを適切なサービスにバインドします。モニター名の長さは最大 255 文字です。次の概念図は、モニターを使用した基本的な負荷分散設定を示しています。

図 1: モニターの動作方法



図のように、各サービスにはモニタがバインドされています。モニターは、サービスを介して負荷分散サーバーをプローブします。負荷分散されたサーバーがプローブに応答する限り、モニターはサーバーにマークを付けます。負荷分散されたサーバが、指定された時間内に指定された数のプローブに応答しなかった場合、モニタはそのプローブを DOWN とマークします。

ここでは、次の詳細について説明します。

- モニターの作成

- サービスの状態を判断するための監視パラメータの設定
- モニターをサービスにバインドする
- モニターの変更
- モニターの有効化と無効化
- モニターのバインド解除
- モニターの削除
- モニターの表示
- モニター接続を閉じる
- モニタプローブのクライアント接続の上限を無視する

モニターを作成する

December 8, 2023

NetScaler アプライアンスには、一連のビルトインモニターが用意されています。また、内蔵モニターを基に、またはゼロからカスタムモニターを作成することもできます。

CLI を使用してモニターを作成するには

コマンドプロンプトで入力します:

```
add lb monitor <monitorName> <monitorType> [<interval>]

add lb mon monitor-HTTP-1 HTTP

add lb mon monitor-HTTP-2 TCP 2
<!--NeedCopy-->
```

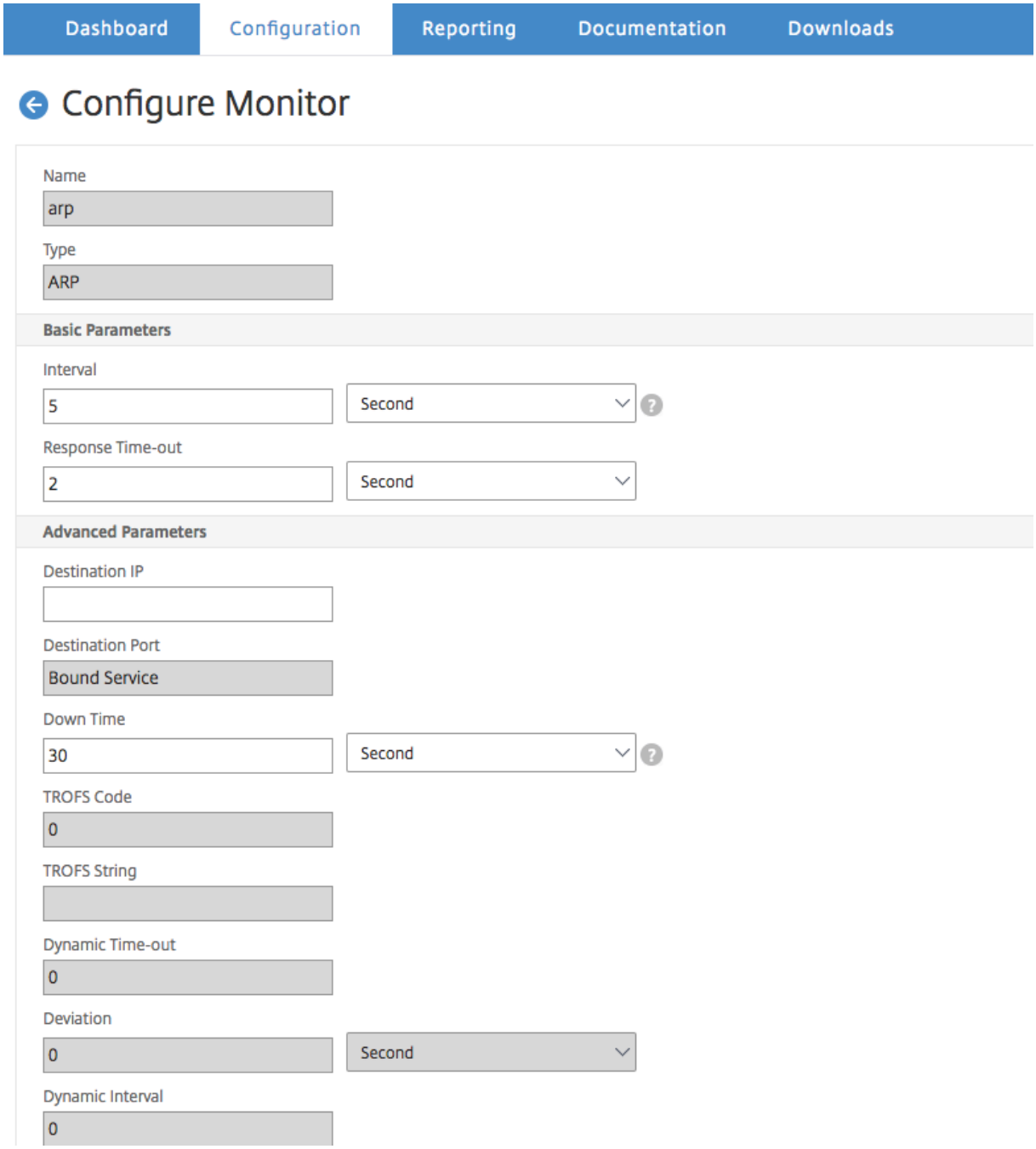
GUI を使用してモニターを作成するには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. [追加] をクリックして、要件を満たすモニタータイプを作成します。

モニターの作成画面には、** 基本パラメーターと詳細パラメーターの **2 つのセクションがあります。

モニタータイプに応じて、**Basic Parameters** セクションには各モニターに設定する必要のあるパラメーターが含まれます。「詳細パラメータ」セクションには、高度な使用事例で使用できるパラメータが含まれています。

次の図は、ARP モニタータイプの [モニターの作成] ページのサンプルです。



> 注
>
> NetScaler リリース 12.0 ビルド 56.20 より前のバージョンでは、基本パラメーターと詳細パラメーターはそれぞれ標準パラメーターと特殊パラメーターという名前でした。

サービス正常性を判断するモニターパラメーターを構成する

August 15, 2023

次のモニタリングパラメータを設定して、モニタリングプローブに基づいてサービスを DOWN としてマークできます。

再試行

監視プローブが失敗したサービスの状態を確立するために送信するプローブの最大数。

失敗**/**再試行

サービスが DOWN としてマークされるために、Retries パラメーターに指定された数のうち、失敗する必要がある再試行の数。たとえば、Retries パラメータが 10 に設定され、Failure Retries パラメータが 6 に設定されている場合、送信された 10 個のプローブのうち、サービスが DOWN としてマークされる場合は、少なくとも 6 つのプローブが失敗する必要があります。

alertRetries

アプライアンスが monProbeFailed と呼ばれる SNMP トラップを生成した後の連続したプローブ障害の数。

alertRetries を再試行値よりも大きい値に設定する

NetScaler ADC アプライアンスが monProbeFailed と呼ばれる SNMP トラップを生成した後の連続する監視プローブ障害の最大数を指定する alertRetries パラメーターを、Retries 値（送信するプローブの最大数を指定する）よりも高い値に設定できるようになりました。監視プローブが失敗したサービスの状態を確立するため）。AlertRetries の値が [再試行] の値よりも大きい場合、SNMP トラップはサービスが停止するまで送信されません。

たとえば、[再試行] を 3、[AlertRetries] を 12、時間間隔を 5 秒に設定した場合、サービスは 15 秒 (35*)* 後に *DOWN* とマークされますが、アラートは生成されません。*60* 秒（*12*、5）経過してもモニタープローブに障害が発生する場合、NetScaler アプライアンスは MonProbeFailed トラップを生成します。15 秒から 60 秒の間でプローブが成功すると、サービスは UP とマークされ、アラートは生成されません。

AlertRetries 値を Retries 値よりも大きい値に設定すると、正規のアラートのみを生成し、予定された再起動時に誤検出が発生するのを防ぐのに役立ちます。

コマンドラインインターフェイスを使用して **AlertRetries** パラメータ値を **Retries** 値よりも高い値に設定するには　コマンドプロンプトで入力します。

```
1 add lb monitor <monitorName> [-retries <integer>] [-alertRetries <
     integer>]
2 <!--NeedCopy-->
```

例**:**

LB モニターモニターを追加**-HTTP-1** HTTP-リトライ 3-アラートリトライ 12

GUI を使用して **AlertRetries** パラメータ値を **[**再試行**]** の値よりも大きい値に設定するには

1. [設定] > [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. [追加] をクリックして新しいモニターを追加するか、既存のモニターを選択して [編集] をクリックします。
3.「再試行」ボックスに、「再試行」パラメータの値を入力します。
4.「**SNMP** アラート再試行」ボックスに、パラメータの値を入力します。`alertRetries`

モニターをサービスにバインドする

August 15, 2023

モニターを作成したら、それをサービスにバインドします。1 つまたは複数のモニターをサービスにバインドできます。1 つのモニターをサービスにバインドすると、そのモニターがサービスのマークが UP か DOWN かを判断します。

複数のモニターを 1 つのサービスにバインドすると、NetScaler アプライアンスはすべてのモニターの状態をチェックし、サービスの状態を判断します。モニターにさまざまな重みを設定できます。モニターの重みによって、そのモニターがサービスを UP または DOWN として指定するうえでどの程度貢献できるかが決まります。重量が大きいモニターほど、サービスを「アップ」または「ダウン」とマークする優先順位が高くなります。デフォルトの重みは 1 です。したがって、モニターの 1 つに障害が発生した場合でも、サービスは DOWN としてマークされます。詳細については、「サービスにバインドされたモニターのしきい値を設定する」を参照してください。

注意: モニタープローブの宛先 IP アドレスは、サーバーの IP アドレスおよびポートとは異なる場合があります。

CLI を使用してモニタをサービスにバインドするには

コマンドプロンプトで入力します。

```
1 bind service <name> (-monitorName <string>)
2 <!--NeedCopy-->
```

例**:**


```
bind service s1 -monitorName tcp
<!--NeedCopy-->
```

GUI を使用してモニタをサービスにバインドするには

1. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。
2. サービスを開き、モニターを追加します。

モニターを変更する

August 15, 2023

作成したすべてのモニターの設定を変更できます。

注: モニターには 2 つのパラメーターが適用されます。1 つはタイプに関係なくすべてのモニターに適用されるパラメーターで、もう 1 つはモニターの種類に固有のパラメーターです。特定のモニター・タイプのパラメーターについては、そのタイプのモニターの説明を参照してください。

CLI を使用して既存のモニタを変更するには

コマンドプロンプトで入力します。

```
set lb monitor <monitorName> <type> -interval <interval> -resptimeout <
   resptimeout>
<!--NeedCopy-->
```

例**:**

```
set mon monitor-HTTP-1 HTTP -interval 50 milli
-resptimeout 20 milli
<!--NeedCopy-->
```

GUI を使用して既存のモニタを変更するには

[トラフィック管理] **>** [負荷分散] **>** [モニタ] に移動し、変更するモニタを開きます。

モニターの有効化と無効化

August 15, 2023

デフォルトでは、サービスとサービスグループにバインドされたモニターは有効になっています。モニターを有効にすると、モニターはバインドされているサービスの調査を開始します。サービスにバインドされたモニターを無効にした場合、サービスの状態は、そのサービスにバインドされている他のモニターを使用して決定されます。サービスが 1 つのモニターのみにバインドされている場合、およびモニターを無効にすると、サービスの状態はデフォルトのモニターを使用して決定されます。

CLI を使用してモニタを有効にするには

コマンドプロンプトで入力します。

```
enable lb monitor <monitorName>
<!--NeedCopy-->
```

例**:**

```
enable lb mon monitor-HTTP-1
<!--NeedCopy-->
```

GUI を使用してモニタを有効にするには

1. [トラフィック管理**] > [**負荷分散**] > [**モニター] に移動します。
2. モニターを選択し、[アクション] リストから [有効] または [無効] を選択します。

CLI を使用してモニタを無効にするには

コマンドプロンプトで入力します。

```
disable lb monitor <monitorName>
<!--NeedCopy-->
```

例**:**

```
disable lb mon monitor-HTTP-1
<!--NeedCopy-->
```

モニタのバインド解除

August 15, 2023

サービスおよびサービスグループからモニターをアンバインドできます。サービスグループからモニターをバインド解除すると、モニターはサービスグループを構成する個々のサービスからバインド解除されます。サービスまたはサ

ービスグループからモニターをバインド解除すると、モニターはサービスまたはサービスグループをプローブしません。

注: ユーザー設定のすべてのモニターをサービスまたはサービスグループからバインド解除すると、デフォルトのモニターがサービスとサービスグループにバインドされます。次に、デフォルトのモニターがサービスまたはサービスグループをプローブします。

CLI を使用してサービスからモニタをバインド解除するには

コマンドプロンプトで入力します。

```
unbind lb monitor <monitorName>
<!--NeedCopy-->
```

例:

```
unbind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスからモニタをバインド解除するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、変更するサービスを開きます。
2. [モニター] セクションをクリックし、モニタを選択し、[バインド解除] をクリックします。

モニターを削除する

August 15, 2023

サービスから作成したモニターをアンバインドしたら、そのモニターを NetScaler 構成から削除できます。(モニターがサービスにバインドされている場合、そのモニターは削除できません)。

注: サービスにバインドされたモニターを削除すると、デフォルトのモニターがサービスにバインドされます。デフォルトのモニターを削除することはできません。

CLI を使用してモニタを削除するには

コマンドプロンプトで入力します。

```
rm lb monitor <monitorName> <type>
<!--NeedCopy-->
```

例:


```
rm lb monitor monitor-HTTP-1 HTTP
<!--NeedCopy-->
```

GUI を使用してモニターを削除するには

1. [トラフィック管理] **>** [負荷分散] **>** [モニター] に移動します。
2. モニタを選択し、[削除] をクリックします。

モニターを表示する

August 15, 2023

モニターにバインドされているサービスとサービスグループを表示できます。モニターの設定を確認して、NetScaler 構成のトラブルシューティングを行うことができます。次の手順では、モニターのサービスおよびサービスグループへのバインディングを表示する手順について説明します。

CLI を使用してモニタバインディングを表示するには

コマンドプロンプトで入力します。

```
show lb monbindings <MonitorName>
<!--NeedCopy-->
```

例**:**

```
show lb monbindings monitor-HTTP-1
<!--NeedCopy-->
```

GUI を使用してモニタバインディングを表示するには

1. [トラフィック管理] **>** [負荷分散] **>** [モニター] に移動します。
2. モニターを選択し、[アクション] リストで、[バインドの表示] をクリックします。

CLI を使用してモニターを表示するには

コマンドプロンプトで入力します。

```
show lb monitor <monitorName>
<!--NeedCopy-->
```


例**:**

```
show lb mon monitor-HTTP-1
<!--NeedCopy-->
```

GUI を使用してモニターを表示するには

1. [トラフィック管理] **>** **[**負荷分散**] >** **[**モニター] に移動します。使用可能なモニターの詳細がモニターペインに表示されます。

モニター接続を閉じる

August 15, 2023

NetScaler アプライアンスは、サービスにバインドされたモニターを介してサービスにプローブを送信します。デフォルトでは、アプライアンスと物理サーバーのモニターは、モニタープローブの場合でも完全なハンドシェイク手順に従います。ただし、この手順では、モニタリングプロセスにオーバーヘッドが加わり、必ずしも必要になるとは限りません。

TCP タイプのモニターの場合、サービスから SYN-ACK を受信した後、モニターとプローブの接続を閉じるようにアプライアンスを構成できます。そのためには、MonitorConnectionClose パラメーターの値を RESET に設定します。モニタとプローブの接続を最後まで実行したい場合は、値を FIN に設定します。

注:MonitorConnectionClose 設定は、TCP タイプと TCP デフォルトタイプのモニターにのみ適用されます。

コマンドラインインターフェイスを使用してモニター接続クロージャーを設定するには:

コマンドプロンプトで入力します。

```
set lb parameter -monitorConnectionClose <monitor_conn_close_option>
<!--NeedCopy-->
```

例

```
set lb parameter -monitorConnectionClose RESET
<!--NeedCopy-->
```

構成ユーティリティを使用してモニター接続クロージャーを設定するには:

1. [トラフィック管理] **>** **[**負荷分散**] >** **[**ロードバランシングパラメータの設定] に移動します。
2. [**FIN**] または [リセット] を選択します。

サービスまたはサービスグループレベルでの監視接続の終了

MonConnectionClose パラメータを設定して、サービスおよびサービスグループレベルでモニタプローブ接続を閉じるようにアプライアンスを設定することもできます。このパラメータが設定されていない場合、グローバル負荷分散パラメータに設定された値を使用してモニタ接続が閉じられます。このパラメータがサービスまたはサービスグループレベルで設定されている場合、FIN または RESET ビットが設定された接続終了メッセージをサービスまたはサービスグループに送信することにより、モニタ接続が閉じられます。

CLI を使用してサービスレベルでモニタ接続終了を設定するには

コマンドプロンプトで入力します。

```
set service <service_name> -monConnectionClose ( RESET | FIN )
<!--NeedCopy-->
```

CLI を使用してサービスグループレベルでモニタ接続終了を設定するには

コマンドプロンプトで入力します。

```
 set serviceGroup <service_name> -monConnectionClose ( RESET | FIN )
<!--NeedCopy-->
```

GUI を使用してサービスレベルでモニタ接続クロージャを設定するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. サービスを追加または編集し、**** 基本設定で監視接続のクローズビットを設定します **。

GUI を使用してサービスグループレベルでモニタ接続終了を設定するには

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. サービスグループを追加または編集し、**** 基本設定で監視接続のクローズビットを設定します **。

> 注**:** グローバルロードバランシングパラメータを使用してモニタプローブ接続を閉じる場合は、MonitorConnectionClose を FIN または RESET に設定できます。MonitorConnectionClose パラメーターを次のように設定すると、
>
> - FIN: アプライアンスは完全な TCP ハンドシェイクを実行します。
> - RESET: アプライアンスは、サービスから SYN-ACK を受信した後、接続を閉じます。
>
> NetScaler CPX のより軽いバージョンでは、MonitorConnectionClose パラメータ値はデフォルトでリセットに設定され、グローバルレベルで FIN に変更することはできません。ただし、サービスレベルで

> MonitorConnectionClose パラメーターを FIN に変更することはできます。

モニタープローブのクライアント接続の上限を無視する

August 15, 2023

物理サーバーの容量などの考慮事項に応じて、どのサービスに対して行われるクライアント接続の最大数にも制限を指定できます。サービスにこのような制限を設定した場合、NetScaler アプライアンスはしきい値に達するとサービスへの要求の送信を停止し、既存の接続数が制限内に収まるとサービスへの接続の送信を再開します。モニタプローブ接続をサービスに送信するときにこのチェックをスキップするようにアプライアンスを設定できます。

注: 個々のサービスの最大クライアント接続数のチェックをスキップすることはできません。このオプションを指定すると、NetScaler ADC アプライアンスで構成されているすべてのサービスにバインドされているすべてのモニターに適用されます。

CLI を使用して **[**モニタ接続の **MaxClients** をスキップ**]** オプションを設定するには

コマンドプロンプトで入力します。

```
set lb parameter -monitorSkipMaxClient (ENABLED|DISABLED)
<!--NeedCopy-->
```

例**:**

```
set lb parameter -monitorSkipMaxClient enabled
<!--NeedCopy-->
```

GUI を使用して **[**モニタ接続の **MaxClients** をスキップ**]** オプションを設定するには

1. [トラフィック管理**] > [**負荷分散**] > [**ロードバランシングパラメータの設定] に移動します。
2. [接続を監視する **MaxClients** をスキップ] を選択します

大規模環境を管理する

August 15, 2023

NetScaler アプライアンスには、大規模な負荷分散環境を構成する際に役立ついくつかの機能が含まれています。仮想サーバーとサービスを個別に設定する代わりに、仮想サーバーとサービスのグループを作成できます。また、さま

ざまな仮想サーバーとサービスを作成したり、仮想サーバーとサービスの IP アドレスを変換またはマスクしたりできます。

仮想サーバーのグループに永続性を設定できます。モニタをサービスのグループにバインドできます。同じタイプの仮想サーバーとサービスの範囲を作成すると、それらのサーバーを 1 つの手順で設定および構成できます。これにより、これらの仮想サーバーとサービスの構成にかかる時間が大幅に短縮されます。

IP アドレスを変換またはマスキングすることで、仮想サーバーとサービスを停止できます。これにより、サービスおよび仮想サーバー定義を広範囲に再構成することなく、インフラストラクチャに変更を加えることができます。

仮想サーバーとサービスの範囲

August 15, 2023

負荷分散を設定すると、仮想サーバーとサービスの範囲を作成できるため、仮想サーバーとサービスを個別に設定する必要がなくなります。たとえば、1 つの手順を使用して、対応する 3 つの IP アドレスを持つ 3 つの仮想サーバーを作成できます。複数の引数が 1 つの範囲を使用する場合、範囲は同じサイズでなければなりません。

次に、サービスおよび仮想サーバーを構成に追加するときに指定できる範囲のタイプを示します。

- 数値の範囲。1 つの数値を入力する代わりに、連続する数値の範囲を指定できます。

  たとえば、開始 IP アドレス (10.102.29.30 など) を指定し、その範囲を示す最後のバイト値 (34 など) を入力することで、仮想サーバーの範囲を作成できます。この例では、10.102.29.30 から 10.102.29.34 の範囲の IP アドレスを使用して 5 つの仮想サーバーが作成されます。

  注: 仮想サーバーとサービスの IP アドレスは連続している必要があります。

- アルファベットの範囲。リテラル文字を入力する代わりに、[C-G] のように任意の 1 文字を範囲に置き換えることができます。これにより、範囲 内のすべての文字（この場合は C、D、E、F、G）が含まれます。

  たとえば、`Vserver-x`、`Vserver-y`、`Vserver-z`という名前の仮想サーバーが 3 つある場合は、別々に構成する代わりに、「`vserver [x-z]`」と入力してすべてを構成できます。

さまざまな仮想サーバの作成

CLI を使用して仮想サーバーの範囲を作成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<
    port>]

add lb vserver <name>@[<rangeValue>]> <protocol> <IPAddress[<rangeValue
    >]> [<port>]
```



```
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
<!--NeedCopy-->
```

または

```
add lb vserver vserver[P-R] http 10.102.29.[26-28] 80

vserver "vserverP" added

vserver "vserverQ" added

vserver "vserverR" added

Done
<!--NeedCopy-->
```

CLI を使用して仮想サーバーの範囲を作成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<
    port>]

add lb vserver <name>@\*\*[\*\*<rangeValue>\*\*]\*\*> <protocol> <
    IPAddress[<rangeValue>]> [<port>]
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
<!--NeedCopy-->
```

または

```
add lb vserver vserver[P-R] http 10.102.29.[26-28] 80
vserver "vserverP" added
vserver "vserverQ" added
vserver "vserverR" added
Done
<!--NeedCopy-->
```

GUI を使用して仮想サーバーの範囲を作成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを追加し、範囲を指定します。

さまざまなサービスの作成

サービス名の範囲を指定する場合は、IP アドレスの範囲も指定します。

CLI を使用してサービスの範囲を作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add service <name>@ <IP>@ <protocol> <port>
<!--NeedCopy-->
```

例**:**

```
> add service serv[1-3] 10.102.29.[102-104] http 80
service "serv1" added
service "serv2" added
service "serv3" added
 Done
<!--NeedCopy-->
```

サービスグループを構成する

August 15, 2023

サービスグループを設定すると、サービスグループを 1 つのサービスと同じくらい簡単に管理できます。たとえば、サービスグループに対して圧縮、ヘルスモニタリング、グレースフルシャットダウンなどのオプションを有効または無効にすると、サービスグループのすべてのメンバーに対してオプションが有効になります。

サービスグループを作成したら、仮想サーバにバインドし、そのグループにサービスを追加できます。モニタをサービスグループにバインドすることもできます。

> 注**:**
>
> 同じ IP アドレスとポートを持つサービスとサービスグループを同じ仮想サーバーにバインドすることはできません。

サービスグループのメンバーは、IP アドレスまたはサーバ名で識別されます。

ドメインネームベースのサービス（DBS）グループメンバーを使用すると、メンバーの IP アドレスが変更された場合に NetScaler ADC アプライアンスのメンバーを再構成する必要がないため、便利です。アプライアンスは、設定されたネームサーバを介してこのような変更を自動的に検出します。この機能は、サービスプロバイダーが物理サーバーを変更したり、サービスの IP アドレスを変更したりできるクラウドシナリオで役立ちます。DBS グループメンバーを指定すると、アプライアンスはその IP アドレスを動的に学習します。

IP ベースのメンバーと DBS メンバーの両方を同じサービスグループにバインドできます。

注：DBS サービスグループメンバーを使用する場合は、ネームサーバーが指定されているか、NetScaler ADC アプライアンスで DNS サーバーが設定されていることを確認してください。ドメイン名は、対応するアドレスレコードがアプライアンスまたはネームサーバに存在する場合のみ、IP アドレスに解決されます。

サービスグループの作成

NetScaler ADC アプライアンスでは最大 8192 のサービスグループを構成できます。

コマンドラインを使用してサービスグループを作成するには

コマンドプロンプトで入力します。

```
add servicegroup <ServiceGroupName> <Protocol>
<!--NeedCopy-->
```

例:

```
add servicegroup Service-Group-1 HTTP
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを作成するには

[トラフィック管理] > [負荷分散] > [サービスグループ] に移動し、サービスグループを追加します。

サービスグループを仮想サーバーにバインドする

サービスグループを仮想サーバーにバインドすると、メンバーサービスは仮想サーバーにバインドされます。

コマンドラインインターフェイスを使用してサービスグループを仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name>@ <serviceGroupName>
<!--NeedCopy-->
```

例:

```
bind lb vserver Vserver-LB-1 Service-Group-1
<!--NeedCopy-->
```


GUI を使用してサービスグループを仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[サービスグループ] を選択します。

メンバーをサービスグループにバインドする

サービスグループにサービスを追加すると、サービスグループがサーバーを管理できるようになります。IP アドレスまたはサーバ名を指定して、サーバをサービスグループに追加できます。

GUI で、ドメイン名ベースのサービスグループメンバーを追加する場合は、[**Server Based**] を選択します。

このオプションを使用すると、名前が IP アドレスであるか、ユーザーによって割り当てられた名前であるかに関係なく、名前が割り当てられた任意のサーバーを追加できます。

コマンドラインインターフェイスを使用してサービスグループにメンバーを追加するには

サービスグループを構成するには、コマンドプロンプトで次のように入力します。

```
bind servicegroup <serviceGroupName> (<IP>@ | <serverName>) <port>
<!--NeedCopy-->
```

例:

```
bind servicegroup Service-Group-1 10.102.29.30 80

bind servicegroup Service-Group-2 1000:0000:0000:0000:0005:0600:700a
    :888b 80

bind servicegroup CitrixEdu s1.citrite.net
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループにメンバーを追加するには

1. [トラフィック管理] **> [**負荷分散**] > [**サービスグループ] に移動し、サービスグループを開きます。
2. [Service Group] セクションをクリックし、次のいずれかの操作を行います。
   - IP ベースのサービスグループメンバーを追加するには、[IP ベース] を選択します。
   - サーバ名ベースのサービスグループメンバーを追加するには、[Server Based] を選択します。

   ドメイン名ベースのサービスグループメンバーを追加する場合は、[サーバベース] を選択します。このオプションを使用すると、名前が IP アドレスであるか、ユーザーによって割り当てられた名前であるかに関係なく、名前が割り当てられた任意のサーバーを追加できます。

3. 新しい IP ベースのメンバーを追加する場合は、IP アドレステキストボックスに IP アドレスを入力します。IP アドレスが IPv6 形式を使用している場合は、IPv6 チェックボックスを選択して、IP アドレステキストボックスにアドレスを入力します。

   注:IP アドレスの範囲を追加できます。範囲内の IP アドレスは連続している必要があります。IP アドレステキストボックスに開始 IP アドレスを入力して範囲を指定します (たとえば、10.102.29.30)。Range の下のテキストボックスに IP アドレス範囲の終了バイトを指定します (例:35)。「ポート」テキストボックスにポート (たとえば、80) を入力し、「追加」をクリックします。

4. ［作成］をクリックします。

モニターをサービスグループにバインドする

サービスグループを作成すると、そのグループに適したタイプのデフォルトモニターが自動的にバインドされます。モニターは、バインドされているサービスグループ内のサーバーを定期的に調査し、サービスグループの状態を更新します。

自分で選択した別のモニターをサービスグループにバインドできます。

コマンドラインインターフェイスを使用してモニターをサービスグループにバインドするには

コマンドプロンプトで入力します。

```
bind serviceGroup <serviceGroupName> -monitorName <string> -monState (
    ENABLED | DISABLED)
<!--NeedCopy-->
```

例:

```
bind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してモニターをサービスグループにバインドするには

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. サービスグループを開き、[詳細設定] で [モニター] をクリックします。

仮想サーバを無効化して有効にした後も、サービスグループメンバーの元の状態を維持

ビルド 64.x から、新しいグローバルオプション—retainDisableServer により、サーバを無効化して再び有効化しても、サービスグループメンバーの状態を保持できます。

以前は、以下の条件でメンバーのステータスが DISABLED から ENABLED に変わっていました。

- 2 つのアプリケーションが仮想サーバーの同じポートにデプロイされます。
- 共通メンバーを持つ 2 つのサービスグループがこの仮想サーバーにバインドされ、共通メンバーは一方のグループで有効になり、もう一方のグループでは無効になります。
- サーバーは無効になり、再び有効になります。

これらの条件下では、サーバを無効にすると、すべてのサービスグループメンバーが無効になり、サーバを再度有効にすると、以前の状態に関係なく、デフォルトですべてのメンバーが有効になります。メンバーを元の状態に戻すには、サービスグループ内のメンバーを手動で無効にする必要があります。これは面倒な作業であり、エラーが発生しやすい。

サービスグループを管理する

August 15, 2023

サービスグループ内のサービスの設定を変更したり、サービスグループの有効化、無効化、削除などのタスクを実行できます。サービスグループからメンバーをバインド解除することもできます。サービスグループの詳細については、「サービスグループの構成」を参照してください。

サービスグループの変更

サービスグループメンバーの属性を変更できます。最大クライアント数や圧縮数など、サービスグループの複数の属性を設定できます。属性は、サービスグループの個々のサーバに設定されます。トランスポート情報（IP アドレスとポート）、重み、サーバ ID などのパラメータをサービスグループに設定することはできません。

注: サービスグループに設定したパラメータは、個々のサービスではなく、グループ内のメンバーサーバに適用されます。

コマンドラインインターフェイスを使用してサービスグループを変更するには

コマンドプロンプトで、1 つ以上のオプションパラメータを指定して、次のコマンドを入力します。

```
set servicegroup <serviceGroupName> [-type <type>] [-maxClient <
    maxClient>] [-maxReq <maxReq>] [-cacheable (YES|NO)] [-cip (ENABLED|
    DISABLED)] [-cipHeader <cipHeader>] [-usip (YES|NO)] [-sc (ON|OFF)]
    [-sp (ON|OFF)] [-cltTimeout <cltTimeout>] [-svrTimeout <svrTimeout>]
     [-cka (YES|NO)] [-TCPB (YES|NO)] [-CMP (\*\*YES\*\*|\*\*NO\*\*)] [-
    maxBandwidth <maxBandwidth>] [-maxThreshold <maxThreshold>] [-state
    (ENABLED|DISABLED)] [-downStateFlush (ENABLED|DISABLED)
<!--NeedCopy-->
```

例:


```
set servicegroup Service-Group-1 -type TRANSPARENT

set servicegroup Service-Group-1 -maxClient 4096

set servicegroup Service-Group-1 -maxReq 16384

set servicegroup Service-Group-1 -cacheable YES
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを変更するには

[トラフィック管理] > [負荷分散] > [サービスグループ] に移動し、変更するサービスグループを開きます。

サービスグループを削除する

サービスグループを削除すると、グループにバインドされたサーバーは個別の設定を保持し、NetScaler ADC アプライアンスに引き続き存在します。

コマンドラインインターフェイスを使用してサービスグループを削除するには

コマンドプロンプトで入力します。

```
rm servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

例**:**

```
rm servicegroup Service-Group-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを削除するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを選択し、[**Delete**] をクリックします。

サービスグループからメンバーをバインド解除する

サービスグループからメンバーをバインド解除すると、サービスグループに設定された属性は、バインド解除したメンバーに適用されなくなります。ただし、メンバーサービスは個々の設定を保持し、NetScaler ADC アプライアンスに引き続き存在します。

コマンドラインインターフェイスを使用してサービスグループからメンバーをバインド解除するには

コマンドプロンプトで入力します。

```
unbind servicegroup <serviceGroupName> <IP>@ [<port>]
<!--NeedCopy-->
```

例:

```
unbind servicegroup Service-Group-1 10.102.29.30 80
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループからメンバーをバインド解除するには

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. サービスグループを開き、[サービスグループメンバー] セクションをクリックします。
3. サービスグループメンバーを選択し、[**Unbind**] をクリックします。

仮想サーバからサービスグループをバインド解除する

仮想サーバーからサービスグループをバインド解除すると、メンバーサービスは仮想サーバーからバインド解除され、NetScaler ADC アプライアンスに引き続き存在します。

コマンドラインインターフェイスを使用して仮想サーバからサービスグループをバインド解除するには

コマンドプロンプトで入力します。

```
unbind lb vserver <name>@ <ServiceGroupName>
<!--NeedCopy-->
```

例:

```
unbind lb vserver Vserver-LB-1 Service-Group-1
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバからサービスグループをバインド解除するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバーを開き、[Service Group] セクションをクリックします。
3. サービスグループを選択し、[**Unbind**] をクリックします。

サービスグループからのモニタのバインド解除

サービスグループからモニタをバインド解除すると、バインドを解除したモニタは、グループを構成する個々のサービスを監視しなくなります。

コマンドラインインターフェイスを使用してサービスグループからモニタをバインド解除するには

コマンドプロンプトで入力します。

```
unbind serviceGroup <serviceGroupName> -monitorName <string>
<!--NeedCopy-->
```

例:

```
unbind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループからモニタをバインド解除するには

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. サービスグループを開き、[モニター] セクションをクリックします。
3. モニタを選択し、[バインド解除] をクリックします。

サービスグループの有効化または無効化

サービスグループとサーバを有効にすると、そのサービスグループに属するサービスが有効になります。同様に、サービスグループに属するサービスを有効にすると、サービスグループとサービスが有効になります。デフォルトでは、サービスグループは有効になっています。

有効なサービスを無効にした後、構成ユーティリティまたはコマンドラインを使用してサービスを表示し、サービスが停止するまでの残り時間を確認できます。

コマンドラインインターフェイスを使用してサービスグループを無効にするには

コマンドプロンプトで入力します。

```
disable servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

例:

```
disable servicegroup Service-Group-1
<!--NeedCopy-->
```


構成ユーティリティを使用してサービスグループを無効にするには

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. サービスグループを選択し、[アクション] リストで [無効] をクリックします。

コマンドラインインターフェイスを使用してサービスグループを有効にするには

コマンドプロンプトで入力します。

```
enable servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

例:

```
enable servicegroup Service-Group-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを有効にするには

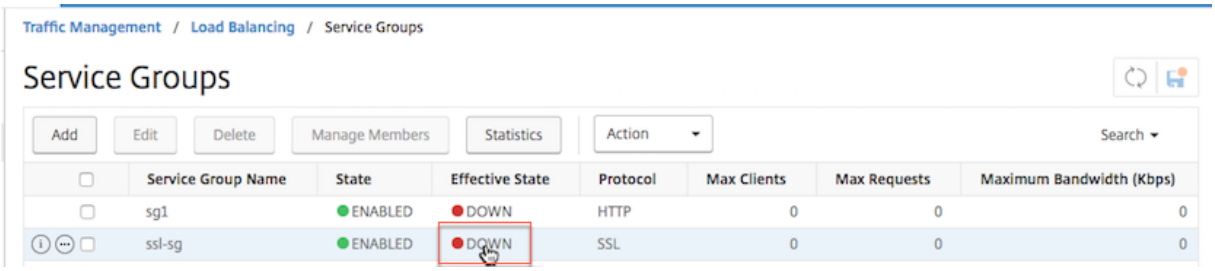
1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. サービスグループを選択し、[アクション] リストで [有効] をクリックします。

サービスグループメンバーのステータスの表示

[トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。

[サービスグループ] ページの [有効な状態] 列に、サービスグループのステータスが表示されます。[有効状態] 列のステータス UP/DOWN はクリック可能です。ステータスをクリックすると、同じビューでメンバーのリストとそのステータスが取得できます。メンバーを選択し、[詳細の監視] ボタンをクリックして、ステータスが [DOWN] になっている理由を表示します。

注: NetScaler リリース 12.0 ビルド 56.20 より前のリリースでは、[有効な状態] 列のステータスはクリック可能ではありませんでした。



サービスグループのプロパティの表示

構成されたサービスグループの次の設定を表示できます。

- 名前
- IP アドレス
- 状態
- プロトコル
- 最大クライアント接続数
- 接続あたりの最大要求数
- 最大帯域幅
- モニタしきい値

設定の詳細を表示すると、設定のトラブルシューティングに役立ちます。

コマンドラインインターフェイスを使用してサービスグループのプロパティを表示するには

コマンドプロンプトで、次のいずれかのコマンドを入力して、グループのプロパティ、またはプロパティとグループメンバーを表示します。

```
show servicegroup <ServiceGroupName>

show servicegroup <ServiceGroupName> -includemembers
<!--NeedCopy-->
```

例:

```
show servicegroup Service-Group-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループのプロパティを表示するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループの横にある矢印をクリックします。

サービスグループの統計情報の表示

要求レート、応答、要求バイト数、応答バイト数などのサービスグループの統計データを表示できます。NetScaler ADC アプライアンスは、サービスグループの統計を使用してサービスの負荷を分散します。

コマンドラインインターフェイスを使用してサービスグループの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

例:

```
stat servicegroup Service-Group-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループの統計情報を表示するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを選択し、[**Statistics**] をクリックします。

サービスグループにバインドされた仮想サーバーの負荷分散

大規模な展開では、同じサービスグループを複数の負荷分散仮想サーバーにバインドできます。このような場合、各仮想サーバーを表示してそれがバインドされているサービスグループを表示する代わりに、サービスグループにバインドされているすべての負荷分散仮想サーバーの一覧を表示できます。各仮想サーバーの次の詳細を表示できます。

- 名前
- 状態
- IP アドレス
- ポート

コマンドラインインターフェイスを使用してサービスグループにバインドされた仮想サーバーを表示するには

コマンドプロンプトで次のコマンドを入力して、サービスグループにバインドされた仮想サーバーを表示します。

```
show servicegroupbindings <serviceGroupName>
<!--NeedCopy-->
```

例:

```
> show servicegroupbindings SVCGRPDTLS
SVCGRPDTLS - State :ENABLED
1) Test-pers (10.10.10.3:80) - State : DOWN
2) BRVSERV (10.10.1.1:80) - State : DOWN
3) OneMore (10.102.29.136:80) - State : DOWN
4) LBVIP1 (10.102.29.66:80) - State : UP
Done
>
```



```
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループにバインドされた仮想サーバーを表示するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを選択し、[アクション] リストで [バインドの表示] をクリックします。

1 回の NITRO API 呼び出しで、サービスグループに必要なサービスグループメンバーのセットを構成する

October 25, 2023

1 回の NITRO API コールで、サービスグループに必要な一連のサービスグループメンバーを設定するためのサポートが追加されました。この設定をサポートするために、新しい API である Desired State API が追加されました。望ましい状態 API を使用すると、次のことができます。

- 「servicegroup_servicegroupgroupmemberlist_binding」リソースに対する単一の PUT リクエストで、サービスグループメンバーのリストを提供します。
- その PUT リクエストで、その重みと状態 (オプション) を指定します。
- アプライアンスの構成を、アプリケーションサーバーに関する展開の変更と効果的に同期します。

NetScaler ADC アプライアンスは、要求された目的のメンバセットと構成済みのメンバセットを比較します。その後、新しいメンバーが自動的にバインドされ、リクエストに存在しないメンバーのバインドが解除されます。

注:

- この機能は、次のタイプのサービスグループでのみサポートされます。`API`.
- Desired State API を使用して IP アドレスベースのサービスのみをバインドできます。ドメイン名ベースのサービスは許可されません。
- 以前は、NITRO コールにバインドできるサービスグループメンバーは 1 つだけです。

> 重要
>
> ServiceGroup メンバーシップに必要な状態 API は、NetScaler ADC クラスタ展開でサポートされています。

ユースケース：**Kubernetes** などの大規模な展開で、展開の変更を **NetScaler ADC** アプライアンスに同期する

大規模で非常に動的なデプロイ（Kubernetes など）では、アプライアンスの構成をデプロイの変更率に合わせて最新の状態に保ち、アプリケーショントラフィックを正確に処理することが課題となります。このような展開では、コ

ントローラ（入力コントローラまたは E-W コントローラ）が ADC 構成の更新を担当します。デプロイメントに変更があるときはいつでも、kube-api serverは「Endpoints イベント」を介して有効なエンドポイントセットをコントローラに送信します。Controller は、読み取り-デルタ-変更アプローチを使用して、次の処理を実行します。

- サービスの現在設定されているエンドポイントセット（サービスグループのサービスグループメンバーセット）を ADC アプライアンスから取得します。
- 設定されたエンドポイントセットと、受信したイベントのセットを比較します。
- 新しいエンドポイント（サービスグループメンバー）をバインドするか、削除したエンドポイントをバインド解除します。

この環境では変更率とサービスのサイズが高いため、この設定方法は効率的ではなく、設定の更新が遅れる可能性があります。

望ましい状態 API は、単一の API でサービスグループの意図されたメンバセットを受け入れることで問題を解決し、構成を効果的に更新します。

CLI を使用してタイプ API のサービスグループを作成する

コマンドプロンプトで次を入力します。

```
add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale <
    autoScale>]
```

例:

```
add serviceGroup svg1 HTTP -autoScale API
```

autoDisablegracefulパラメーター、autoDisabledelayパラメーター、autoScaleパラメーターを設定するには、add serviceGroup コマンドまたは set serviceGroup コマンドを使用します。

```
add serviceGroup <serviceGroupName>@ <serviceType> \[-autoScale <
    autoScale>] \[-autoDisablegraceful \( YES | NO)] \[-autoDisabledelay
     <secs>]

add serviceGroup <serviceGroupName>@ <serviceType> \[-autoScale \(API |
    CLOUD | DISABLED| DNS |POLICY)]

set serviceGroup <serviceGroupName \[-autoDisablegraceful \( YES | NO)]
     \[-autoDisabledelay <secs>]

set serviceGroup <serviceGroupName \[-autoScale \(API |CLOUD | DISABLED
    | DNS |POLICY)]
```

例:

```
add serviceGroup svg1 HTTP autoDisablegraceful YES -autoDisabledelay
    100
```



```
2
3 add serviceGroup svg1 HTTP -autoScale API
4
5 set serviceGroup svg1 -autoDisablegraceful  YES -autoDisabledelay 100
6
7 set serviceGroup svg1 -autoScale API
```

引数

自動無効化グレースフル　サービスが正常にシャットダウンされたことを示します。このオプションを有効にすると、アプライアンスはこのサービスへの未処理の接続がすべて閉じられるのを待ってから、サービスを削除します。システム上にすでに永続的なセッションがあるクライアントでは、新しい接続または要求が引き続きこのサービスに送信されます。サービスメンバが削除されるのは、未処理の接続がない場合だけです。デフォルト値:NO

AutoDisableDelay　グレースフルシャットダウンに許容される時間（秒単位）を示します。この期間中、システムに永続的セッションがあるクライアントに対しては、新しい接続または要求がこのサービスに送信され続けます。システムにパーシステンスセッションがない新しいクライアントからの接続または要求は、サービスに送信されません。代わりに、利用可能な他のサービス間で負荷が分散されます。遅延時間が経過すると、サービスメンバーは削除されます。

Autoscale API　Autoscale API 引数を使用すると、Desired State API を使用してメンバセットを目的のサービスグループにバインドできます。指定された条件がすべて一致する場合、サービスグループを Desired State API の非自動スケールタイプから Autoscale タイプに設定できます。

必要な状態 API は、サービスグループメンバーの IP アドレスが既存のサーバに関連付けられているかどうかを確認します。IP アドレスが既存のサーバーと一致する場合、API は既存のサーバーの IP アドレスと名前を再利用します。IP アドレスが既存のサーバと一致しない場合、API はサーバを作成し、IP アドレス自体をサーバ名として割り当てます。

例:

NetScaler ADC アプライアンスに存在する、IP アドレスが 2.2.2.2 で、名前が myserver であるサーバーを考えてみましょう。目的の状態 API を使用して、IP アドレスが 2.2.2.1 ～2.2.2.3 の範囲のサービスグループメンバーのセットをバインドします。

IP アドレス 2.2.2.2 は既存のサーバに関連付けられているため、API は IP アドレスと名前（2.2.2.2 と myserver）を再利用します。IP アドレス 2.2.2.1、2.2.2.3 を持つ既存のサーバーはないため、API はこれらの IP アドレスを持つサーバーを作成します。API は IP アドレス自体をサーバー名として割り当てます。

目的の状態コマンドで指定された IP アドレスが、CS 仮想サーバーなどの他の NetScaler ADC エンティティと競合すると、競合が発生します。失敗の理由を含むエラーメッセージが表示されます。障害が発生したメンバーのリストのうち、最初のサービスグループメンバーの IP アドレスがエラーメッセージに表示されます。

例:

IP アドレスが 2.2.2.8 のサーバが LB サーバとして使用されているとします。目的の状態 API を使用して、IP アドレスが 2.2.2.2 ～2.2.2.11 の範囲のサービスグループメンバーのセットをバインドしようとします。

2.2.2.8 はすでに LB サービスに使用されているため、競合が発生します。失敗の理由とメンバーバインディングの失敗を含む次のエラーメッセージが表示されます。

```
{
  "errorcode": 304, "message": "Address already in use", "severity": "
     ERROR", "servicegroup_servicegroupmemberlist_binding": {
  "servicegroupname": "sg1", "failedmembers": [ {
  "ip": "2.2.2.8", "port": 80  }
 , {
  "ip": "2.2.2.9", "port": 80  }
  ]  }
   }

<!--NeedCopy-->
```

エラーコード 304 は、障害が発生したメンバーのリストのうち、最初のサービスグループメンバーである 2.2.2.8 を表示します。

既存のメンバーバインディングが次のいずれかの条件を満たすと、`set serviceGroup Autoscale` コマンドが失敗することがあります。

- サービスグループにバインドされているサーバがネームサーバまたはドメインベースのサーバである場合。
- ループバックサーバー名が 127.0.0.1 または 0000:0000:0000:0000:0000:0000:0000:0001 以外である場合。
- set ServiceGroup コマンドで異なるタイプの Autoscale (クラウド、API、DNS、ポリシー) を選択し、serviceGroup コマンドを追加する場合。

重要:

- AutoDisableGraceful および AutoDisableDelay パラメータは、Autoscale タイプ「API」および「クラウド」のサービスグループにのみ適用されます。
- AutoDisableGraceful パラメーターまたは AutoDisableDelay パラメーターが設定されていない場合、サービスメンバーはすぐに削除されます。

サービスグループメンバーを正常にバインド解除する

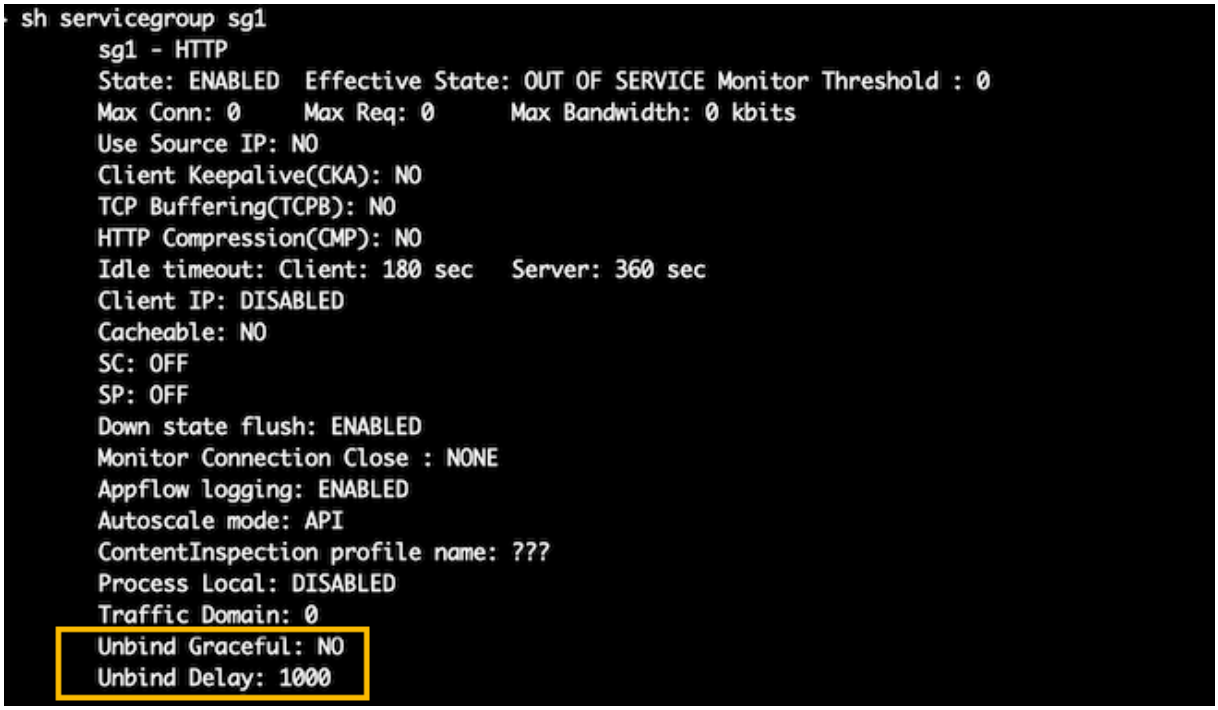
目的のステートリストにないサービスグループメンバーがある場合、`autoDisablegraceful`または`autoDisabledelay`パラメータの設定に基づいて、それらのメンバーは正常にバインド解除されます。

- これらのパラメータの 1 つが設定されている場合、サービスグループメンバーは正常にバインド解除されます。
- これらのパラメータがいずれも設定されていない場合、サービスグループメンバーはすぐにバインド解除されます。

注:

- グレースフルバインド解除のために識別されたサービスグループメンバーは、show service group コマンドの実行時にだけ表示されます。
- 正常なアンバインドとして識別されたサービスグループメンバーに対して、操作（set、unset など）を実行できません。

次の図に、show service group コマンドの例を示します。



GUI を使用して **API** タイプのサービスグループを作成する

1. [トラフィック管理] **>** [負荷分散] **>** [サービスグループ] に移動し、[追加] をクリックします。
2. [**AutoScale** モード] で、[**API**] を選択します。

GUI を使用して、**API** タイプのサービスグループにグレースフルシャットダウンまたは遅延時間を設定します

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Service Groups**] の順に移動します。
2. [**AutoScale** モード] で、[**API**] を選択します。
3. [グレースフルの自動無効化] で [はい] を選択します
4. [自動無効化遅延] に、グレースフルシャットダウンの待機時間を入力します。

注:［自動無効化］または［自動表示遅延］フィールドは、**AutoScale** モードで［**API** ］または［**CLOUD**］を選択した場合にのみ有効になります。

サービスグループのドメインベースの自動スケーリングを構成する

December 8, 2023

ドメインベースのサービスグループは、サービスグループにバインドされたサーバーのドメイン名を解決することによって取得される IP アドレスを持つメンバーで構成されます。ドメイン名は、アプライアンス上で詳細を設定するネームサーバによって解決されます。ドメインベースのサービスグループには、IP アドレスベースのメンバーを含めることもできます。

ドメインベースのサーバーの名前解決のプロセスでは、複数の IP アドレスが返されることがあります。DNS 応答に含まれる IP アドレスの数は、ネームサーバ上のドメイン名に設定されたアドレス (A) レコードの数によって決まります。名前解決プロセスが複数の IP アドレスを返す場合でも、1 つの IP アドレスだけがサービスグループにバインドされます。サービスグループをスケールアップまたはスケールダウンするには、サービスグループに対して他のドメインベースのサーバーを手動でバインドおよびバインド解除する必要があります。

ただし、ドメインベースのサーバーの DNS ネームサーバーから返される IP アドレスの完全なセットに基づいて自動的にスケーリングするように、ドメインベースのサービスグループを構成できます。自動スケーリングを設定するには、ドメインベースのサーバをサービスグループにバインドするときに、[Automatic Scaling] オプションを有効にします。自動的にスケーリングするドメインベースのサービスグループを設定する手順は、次のとおりです。

- ドメイン名を解決するためのネームサーバーを追加します。アプライアンスでのネームサーバの設定の詳細については、ネームサーバの追加を参照してください。
- ドメインベースのサーバーを追加します。ドメインベースのサーバーを追加する方法については、「サーバーオブジェクトの構成」を参照してください。
- サービスグループを追加し、ドメインベースのサーバーをサービスグループに関連付けます。[自動スケール] オプションは [DNS] に設定されています。サービスグループの追加については、サービスグループの構成を参照してください。

ドメインベースのサーバーがサービスグループにバインドされ、自動スケーリングオプションがバインディングに設定されている場合、UDP モニターと TCP モニターが自動的に作成され、ドメインベースのサーバーにバインドされます。2 台のモニタはリゾルバとして機能します。TCP モニターはデフォルトで無効になっており、アプライアンスは UDP モニターを使用して DNS クエリをネームサーバーに送信し、ドメイン名を解決します。DNS 応答が切り捨てられた（TC フラグが 1 に設定されている）場合、アプライアンスは TCP にフォールバックし、TCP モニタを使用して TCP 経由で DNS クエリを送信します。その後、アプライアンスは引き続き TCP モニターのみを使用します。

ネームサーバからの DNS 応答には、ドメイン名に対して複数の IP アドレスが含まれる場合があります。自動スケーリングオプションが設定されている場合、アプライアンスはデフォルトのモニターを使用して各 IP アドレスをポーリングし、稼働していて使用可能な IP アドレスのみをサービスグループに含めます。Time To Live（TTL; 存続可能

時間）値で定義された IP アドレスレコードの有効期限が切れると、UDP モニタ（または、アプライアンスが TCP モニタを使用するようにフォールバックした場合は TCP モニタ）はネームサーバにドメイン解決を照会し、新しい IP アドレスをサービスグループに含めます。サービスグループの一部である IP アドレスが DNS 応答に存在しない場合、アプライアンスはグループメンバーへの既存の接続を正常に閉じた後、そのアドレスをサービスグループから削除します。このプロセスでは、メンバーとの新しい接続を確立できません。過去に正常に解決されたドメイン名が NXDOMAIN 応答になった場合、そのドメインに関連付けられているすべてのサービスグループメンバーが削除されます。

スタティック（IP アドレスベース）メンバとダイナミックスケーリングドメインベースメンバは、1 つのサービスグループに共存できます。また、Automatic Scaling オプションを設定して、異なるドメイン名を持つメンバーを 1 つのサービスグループにバインドすることもできます。ただし、サービスグループに関連付けられている各ドメイン名は、サービスグループ内で一意である必要があります。サービスグループの自動スケーリングに使用するドメインベースのサーバごとに、[Automatic Scaling] オプションを有効にする必要があります。IP アドレスが 1 つ以上のドメインに共通する場合、その IP アドレスはサービスグループに 1 回だけ追加されます。

> 重要
>
> - DNS Autoscale はクラスタ展開でサポートされています。
> - Autoscale サービスグループのパス監視は、クラスタ展開ではサポートされていません。

コマンドラインインターフェイスを使用してサービスグループを自動的にスケーリングするように構成するには

コマンドプロンプトで次のコマンドを入力して、サービスグループを構成し、構成を確認します。

```
add servicegroup <serviceGroupName> <serviceType> -autoscale DNS
<!--NeedCopy-->
```

例

次の例では、server1 はドメインベースのサーバーです。DNS 応答には複数の IP アドレスが含まれています。5 つのアドレスが使用可能で、サービスグループに追加されます。

```
> add serviceGroup servGroup -autoScale YES
 Done
> sh servicegroup servGroup
        servGroup - HTTP
         State: ENABLED  Monitor Threshold : 0
                . . .
                . . .
         1)   192.0.2.31:80   State: UP       Server Name: server1 (Auto
              scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
```



```
                Last response: Success - TCP syn+ack received.

        2)   192.0.2.32:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        3)   192.0.2.36:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        4)   192.0.2.55:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        5)   192.0.2.80:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを自動的にスケーリングするように構成するには

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. サービスグループを作成し、Autoscale モードを DNS に設定します。

TTL 値の上書き

> 注:
>
> このオプションは、NetScaler 12.1 ビルド 51.xx 以降でサポートされています。

NetScaler アプライアンスは、アプリケーションの起動時に、アプリケーションに関連付けられた SRV レコードの更新について DNS サーバーに定期的にクエリを実行するように構成されています。デフォルトでは、このクエリの周期性は SRV レコードにパブリッシュされた TTL によって異なります。マイクロサービスまたはクラウドワールドアプリケーションでは、デプロイメントはより動的に変化します。そのため、プロキシはアプリケーションのデプロイメントに対する変更をより迅速に吸収する必要があります。したがって、ドメインベースのサービス TTL パラメー

タは、SRV レコード TTL よりも小さく、展開に最適な値に明示的に設定することをお勧めします。TTL 値は、次の 2 つの方法で上書きできます。

- メンバーをサービスグループにバインドしているとき
- set lb パラメータコマンドを使用して、TTL 値をグローバルに設定します。

TTL 値がサービスグループメンバーのバインド時とグローバルの両方で設定されている場合、サービスグループメンバーのバインド時に指定された TTL 値が優先されます。
サービスグループメンバーのバインド中またはグローバルレベルで TTL 値が指定されていない場合、DBS モニタ間隔は DNS 応答の TTL 値から導出されます。

CLI を使用した **TTL** 値の上書き

- バインド中に TTL 値を上書きするには、コマンドプロンプトで次のように入力します。

```
bind serviceGroup <serviceGroupName> (<serverName> [-dbsTTL <secs
    >])
<!--NeedCopy-->
```

例:

```
bind servicegroup svc_grp_1 web_serv -dbsTTL 10
<!--NeedCopy-->
```

- TTL 値をグローバルに上書きするには、コマンドプロンプトで次のように入力します。

```
set lb parameter [-dbsTTL <secs>]
<!--NeedCopy-->
```

例:

```
set lb parameter -dbsTTL 15
<!--NeedCopy-->
```

GUI を使用した **TTL** 値の上書き

バインド中に **TTL** 値を上書きするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. [**Service Groups**] ページで、作成したサービスグループを選択し、[**Edit**] をクリックします。
3. [負荷分散サービスグループ] ページで、[サービスグループメンバー] をクリックします。
4. 「サービスグループメンバーのバインド」ページで、作成したサーバーを選択し、「編集」をクリックします。
5. [ドメインベースサービス **TTL**] に、TTL 値を入力します。

TTL 値をグローバルレベルで上書きするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [負荷分散パラメータの変更] に移動します。
2. [ドメインベースサービス **TTL**] に、TTL 値を入力します。

> 注**:**
>
> ドメインベースのサーバの TTL 値が 0 に設定されている場合は、データパケットの TTL 値が使用されます。

サービスグループとドメイン名のバインディングに異なるネームサーバを指定する

> 注:
>
> このオプションは、NetScaler 12.1 ビルド 51.xx 以降でサポートされています。

特定のグループ内の異なるドメイン名に対して、異なるネームサーバを設定できます。DBS サーバをサービスグループにバインドする場合、NameServer パラメータの設定は任意です。メンバーをサービスグループにバインドする際にネームサーバーが指定されない場合、グローバルに設定されたネームサーバーが考慮されます。

CLI を使用してサーバをサービスグループにバインドする際にネームサーバを指定する　コマンドプロンプトで入力します:

```
bind serviceGroup <serviceGroupName> (<serverName> [-nameServer <
    ip_addr>] [-dbsTTL <secs>])
<!--NeedCopy-->
```

例:

```
bind servicegroup svc_grp_1 web_serv  -ns.nameserver.com 10.102.27.155
    -dbsTTL 10
<!--NeedCopy-->
```

GUI を使用してサーバをサービスグループにバインドする際にネームサーバを指定する

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. [**Service Groups**] ページで、作成したサービスグループを選択し、[**Edit**] をクリックします。
3. [負荷分散サービスグループ] ページで、[サービスグループメンバー] をクリックします。
4. 「サービスグループメンバーのバインド」ページで、作成したサーバーを選択し、「編集」をクリックします。
5. [ネームサーバー] で、バインドされたドメインのクエリの送信先となるネームサーバー名を指定します。

自動遅延 **TROFS**

DNS 応答から IP アドレスが削除されたときに、サービスグループのメンバーが TROFS 状態に正常に移行するように設定できます。自動遅延 TROFS オプションを有効にすると、NetScaler は、メンバーを TROFS 状態に移行する前に、サービスグループに接続されているすべてのモニターで最も長い応答タイムアウトを待ちます。

このオプションは、新しい IP アドレスセットによって既存の IP アドレスセットが完全に置き換えられ、新しい IP アドレスを追加する前に接続を確認する必要がある場合に便利です。

> 注:
>
> `-autoDelayedTrofs`オプションは、NetScaler 13.1 ビルド 37.xx 以降でサポートされています。

CLI を使用して自動遅延 **TROFS** を設定する

コマンドプロンプトで、次のコマンドを入力します:

```
add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale <
    autoScale>] [-autoDelayedTrofs ( YES | NO)]
<!--NeedCopy-->
```

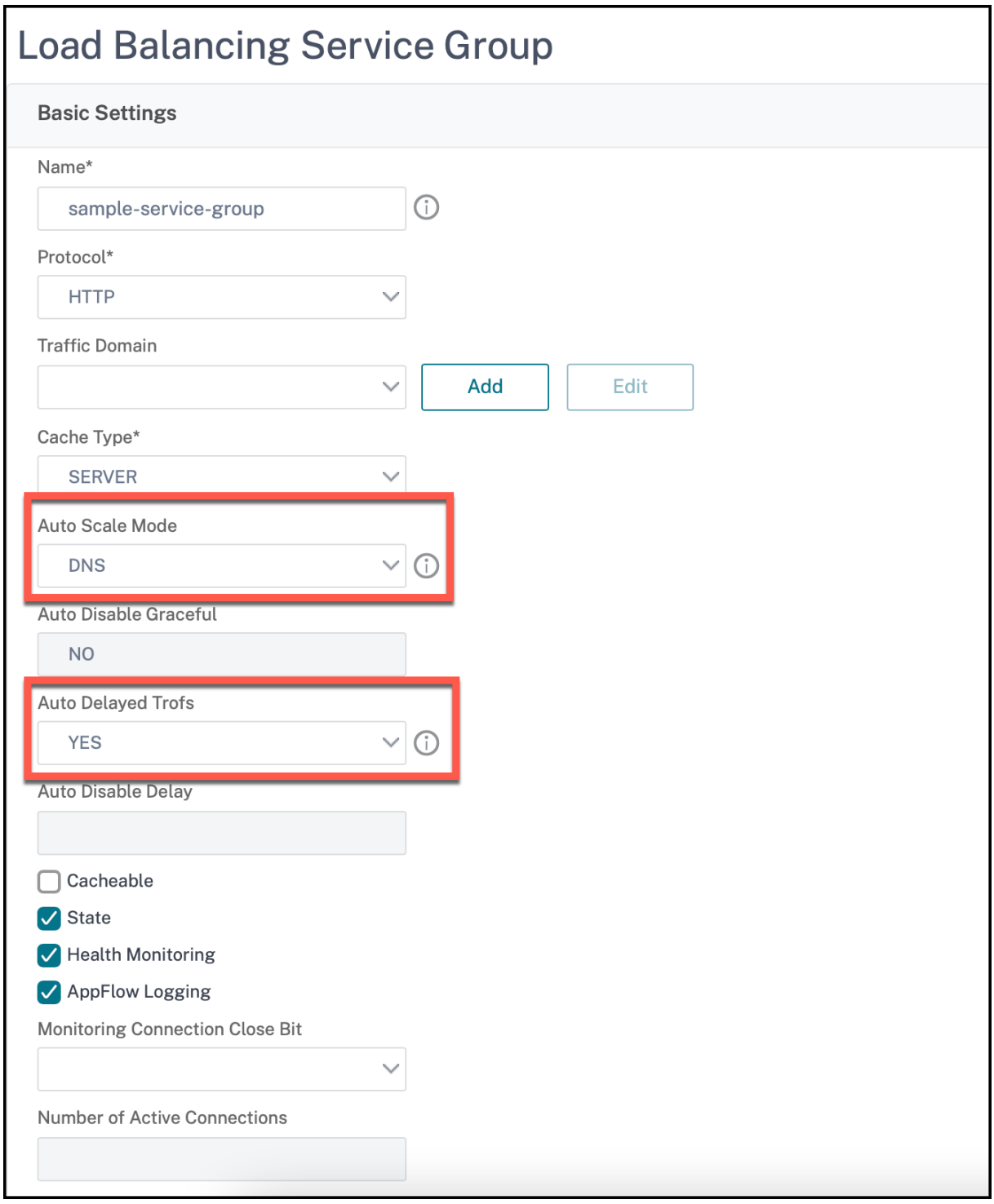
例

```
> add serviceGroup sg1 HTTP -autoScale DNS -autoDelayedTrofs YES
<!--NeedCopy-->
```

GUI を使用して自動遅延 **TROFS** を設定する

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Service Groups**] の順に移動します。
2. [**AutoScale Mode**] で、[**DNS**] を選択します。
3. 「自動遅延 **Trofs**」で「はい」を選択します。

> 注:
>
> 自動遅延 Trofs オプションは、オートスケールモードで DNS を選択した場合にのみ有効になります。



DNS サービスレコードを使用したサービス検出

October 25, 2023

SRV レコード（サービスレコード）は、場所、つまり特定のサービスのサーバーのホスト名とポート番号を定義するドメインネームシステム内のデータの仕様です。このレコードには、各サーバーの重みと優先順位も定義されます。

SRV レコードの例**:**

_http。_tcp.example.com。100 IN SRV 10 60 5060 a.example.com。

次の表に、SRV レコードの各項目を示します。

Service	Protocol	Name	TTL	Class	SRV	Priority	Weight	Port	Target
HTTP	TCP	example.com	100	IN	SRV	10	60	5060	a.example.com

DNS SRV レコードを使用して、サービスエンドポイントを検出できます。NetScaler アプライアンスは、サービスに関連付けられた SRV レコードを使用して DNS サーバーに定期的にクエリを実行するように構成されています。SRV レコードを受信すると、SRV レコードに公開されている各ターゲットホストは、サービスに関連するサービスグループにバインドされます。各バインディングは、SRV レコードからポート、優先度、および重みを継承します。サービスを展開するたびに、ユーザーは起動時に NetScaler アプライアンスを一度構成する必要があるため、アプリケーションをワンタッチで展開できます。

重要：動的に学習されるサービスグループメンバーの体重は、CLI または GUI を使用して変更することはできません。

使用事例**:** 負荷分散マイクロサービス

アプリケーションはモノリシックアーキテクチャからマイクロサービスアーキテクチャに移行しています。マイクロサービス・アーキテクチャとバックエンド・サーバの Autoscale ソリューションへの移行により、アプリケーションの展開がよりダイナミックになります。このような動的な展開をサポートするには、プロキシまたは ADC がバックエンドアプリケーションまたはサービスインスタンスを動的に検出し、それらをプロキシ構成に吸収できる必要があります。
DNS SRV レコードを使用したサービス検出機能は、このような動的な展開シナリオにおける NetScaler アプライアンスの構成に役立ちます。アプリケーション開発者は、オーケストレーションプラットフォームの一部を使用してアプリケーションをデプロイできます。オーケストレーションプラットフォームは、アプリケーションのデプロイ時にコンテナをインスタンス化するときに、これらのコンテナごとにプロトコル固有の標準ポートを割り当てない場合があります。このようなシナリオでは、ポート情報を検出することが NetScaler ADC アプライアンスを構成するための鍵となります。このようなシナリオでは、SRV レコードが役立ちます。優先度や重みなどの SRV レコードパラメータを使用して、アプリケーションのロードバランシングを向上させることができます。

- 優先度パラメータを使用して、サーバープールの優先順位を指定できます。
- 重みパラメータはバックエンドサービスインスタンスの容量を決定するために使用できるため、加重負荷分散に使用できます。
- バックエンドサーバープールに変更があった場合 (たとえば、バックエンドインスタンスがプールから削除されるなど)、既存のクライアント接続がすべて受け入れられた後にのみ、インスタンスは正常に削除されます。

> 注:
>
> - A/AAAA レコードベースのサービス検出では、解決対象のドメインに重みを割り当てるため、解決された IP アドレスはすべて同じ重みになります。
> - SRV 応答の重みが 100 を超える場合、サービスは作成されません。

SRV レコードを使用した優先度ベースのロードバランシング

SRV レコードを使用して、優先度ベースのロードバランシングを実行できます。優先度ベースのサーバープールは、バックアップ仮想サーバーの代わりに使用できます。ns.conf ファイルは、バックアップ仮想サーバーと比較して最小限の構成で済みます。

SRV レコードを使用した優先順位ベースの負荷分散では、各サーバプールに優先順位番号が割り当てられます。最小の数値が最も高い優先度を持ちます。優先順位が最も高いプール内のサーバの 1 つが、サーバの正常性と可用性に基づいて負荷分散対象として選択されます。優先順位が最も高いサーバープール内のすべてのサーバーがダウンした場合、次に優先順位の高いサーバーが負荷分散の対象として選択されます。ただし、最も優先順位の高いサーバプール内のサーバが再び稼働すると、そのサーバは優先順位が最も高いプールから再び選択されます。

ある優先サーバープールから別のサーバープールへの切り替えは、既存のクライアントトランザクションを流出させることによってスムーズに行われます。したがって、現在のクライアントでは、アプリケーションアクセスが中断されることはありません。

CLI を使用して SRV レコードのクエリを有効にするには

SRV レコードのクエリを有効にするには、次のタスクを実行します。

1. クエリータイプパラメーターに SRV を指定してサーバーを作成します。

   コマンドプロンプトで入力します。

   ```
   add server <name> <domain> [-queryType <queryType>])
   <!--NeedCopy-->
   ```

   例:

   ```
   add server web_serv example.com -queryType SRV
   <!--NeedCopy-->
   ```

   注:

   - デフォルトでは、IPv4 クエリが送信されます。IPv6 クエリを送信するには、IPv6 ドメインを有効にする必要があります。
   - SRV ターゲットドメイン名は 127 文字を超えてはなりません。

2. オートスケールモードを DNS としてサービスグループを作成します。

   コマンドプロンプトで入力します。

   ```
   add serviceGroup <serviceGroupName> <serviceType> [-autoScale <
       autoScale>]
   <!--NeedCopy-->
   ```

   例:

   ```
   add servicegroup svc_grp_1 http -autoscale dns
   <!--NeedCopy-->
   ```

3. ステップ 1 で作成したサーバーをメンバーとしてサービスグループにバインドします。

   コマンドプロンプトで入力します。

   ```
   bind serviceGroup <serviceGroupName> <serverName>
   <!--NeedCopy-->
   ```

   例:

   ```
   bind servicegroup svc_grp_1 web_serv
   <!--NeedCopy-->
   ```

> 注:
>
> - サーバをサービスグループのメンバーにバインドする場合、SRV サーバタイプのポート番号を入力する必要はありません。SRV サーバータイプにポート番号を指定すると、エラーメッセージが表示されます。
> - サーバーをサービスグループにバインドする際に、オプションでネームサーバーと TTL 値を指定できます。

GUI を使用して **SRV** レコードをクエリできるようにするには

サーバーの作成

1. [トラフィック管理] > [負荷分散] > [サーバー] に移動し、[追加] をクリックします。
2. 「サーバーの作成」ページで、ドメイン名を選択します。
3. すべての必須パラメータの詳細を入力します。
4. 「クエリー・タイプ」で、「**SRV**」を選択します。
5. ［作成］をクリックします。

オートスケールモードを **DNS** とするサービスグループを作成

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。

2. 負荷分散サービスグループページで、必要なすべてのパラメータの詳細を入力します。
3. [**AutoScale Mode]** で、[**DNS**] を選択します。
4. [**OK**] をクリックします。

サーバーをサービスグループメンバーにバインド

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. [**Service Groups**] ページで、作成したサービスグループを選択し、[**Edit**] をクリックします。
3. [負荷分散サービスグループ] ページで、[サービスグループメンバー] をクリックします。
4. 「サービスグループメンバーのバインド」ページで、作成したサーバーを選択し、「閉じる」をクリックします。

> 注:
>
> - バインド中は、SRV サーバタイプのポート番号を入力する必要はありません。SRV サーバタイプのポート番号を入力すると、エラーメッセージが表示されます。
> - サーバーをサービスグループにバインドする際に、オプションでネームサーバーと TTL 値を指定できます。

TTL 値の上書き

NetScaler ADC アプライアンスは、アプリケーションの起動時に、アプリケーションに関連付けられた SRV レコードの更新について DNS サーバーに定期的にクエリを実行するように構成されています。デフォルトでは、このクエリの周期性は SRV レコードにパブリッシュされた TTL によって異なります。マイクロサービスまたはクラウドワールドアプリケーションでは、デプロイメントはより動的に変化します。そのため、プロキシはアプリケーションのデプロイメントに対する変更をより迅速に吸収する必要があります。したがって、ドメインベースのサービス TTL パラメータは、SRV レコード TTL よりも小さく、展開に最適な値に明示的に設定することをお勧めします。TTL 値は、次の 2 つの方法で上書きできます。

- メンバーをサービスグループにバインドしているとき
- set lb パラメータコマンドを使用して、TTL 値をグローバルに設定します。

TTL 値がサービスグループメンバーのバインド時とグローバルの両方で設定されている場合、サービスグループメンバーのバインド時に指定された TTL 値が優先されます。
サービスグループメンバーのバインド中またはグローバルレベルで TTL 値が指定されていない場合、DBS モニタ間隔は DNS 応答の TTL 値から導出されます。

CLI を使用した **TTL** 値の上書き

- バインド中に TTL 値を上書きするには、コマンドプロンプトで次のように入力します。


```
bind serviceGroup <serviceGroupName> (<serverName> [-dbsTTL <secs
    >])
<!--NeedCopy-->
```

例:

```
bind servicegroup svc_grp_1 web_serv -dbsTTL 10
<!--NeedCopy-->
```

- TTL 値をグローバルに上書きするには、コマンドプロンプトで次のように入力します。

```
set lb parameter [-dbsTTL <secs>]
<!--NeedCopy-->
```

例:

```
set lb parameter -dbsTTL 15
<!--NeedCopy-->
```

GUI を使用した **TTL** 値の上書き

バインド中に **TTL** 値を上書きするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. [**Service Groups**] ページで、作成したサービスグループを選択し、[**Edit**] をクリックします。
3. [負荷分散サービスグループ] ページで、[サービスグループメンバー] をクリックします。
4.「サービスグループメンバーのバインド」ページで、作成したサーバーを選択し、「編集」をクリックします。
5. [ドメインベースサービス **TTL**] に、TTL 値を入力します。

TTL 値をグローバルレベルで上書きするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [負荷分散パラメータの変更] に移動します。
2. [ドメインベースサービス **TTL**] に、TTL 値を入力します。

> 注: ドメインベースのサーバーの TTL 値が 0 に設定されている場合、データパケットの TTL 値が使用されます。

サービスグループとドメイン名のバインディングに異なるネームサーバを指定する

特定のグループ内の異なるドメイン名に対して、異なるネームサーバを設定できます。DBS サーバをサービスグループにバインドする場合、NameServer パラメータの設定は任意です。メンバーをサービスグループにバインドする際にネームサーバーが指定されない場合、グローバルに設定されたネームサーバーが考慮されます。

CLI を使用してサーバをサービスグループにバインドする際にネームサーバを指定する

コマンドプロンプトで入力します。

```
bind serviceGroup <serviceGroupName> (<serverName> [-nameServer <
    ip_addr>] [-dbsTTL <secs>])
<!--NeedCopy-->
```

例:

```
bind servicegroup svc_grp_1 web_serv  -ns.nameserver.com 10.102.27.155
    -dbsTTL 10
<!--NeedCopy-->
```

GUI を使用してサーバをサービスグループにバインドする際にネームサーバを指定する

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. [**Service Groups**] ページで、作成したサービスグループを選択し、[**Edit**] をクリックします。
3. [負荷分散サービスグループ] ページで、[サービスグループメンバー] をクリックします。
4. 「サービスグループメンバーのバインド」ページで、作成したサーバーを選択し、「編集」をクリックします。
5. [ネームサーバー] で、バインドされたドメインのクエリの送信先となるネームサーバー名を指定します。

ドメインベースのサーバーの **IP** アドレスを変換する

August 15, 2023

NetScaler アプライアンスとそれに接続されているドメインベースのサーバーのメンテナンスを簡素化するために、IP アドレスマスクと翻訳 IP アドレスを構成できます。これらの関数は連携して着信 DNS パケットを解析し、DNS で解決された IP アドレスを新しい IP アドレスに置き換えます。

ドメインベースのサーバ用に設定されている場合、IP アドレス変換により、メンテナンスのためにサーバを停止したり、サーバに影響するその他のインフラストラクチャの変更を行った場合に、アプライアンスは代替サーバの IP アドレスを特定できます。

マスクを設定するときは、標準の IP マスク値（2 の累乗、1 を引いた値）とゼロ（255.255.0.0 など）を使用する必要があります。ゼロ以外の値は、最初のオクテットでのみ使用できます。

サーバに変換 IP を設定すると、サーバ IP アドレスと、その IP アドレスの先頭または末尾のオクテットを共有する代替サーバとの間に 1 対 1 の対応関係が確立されます。マスクは、元のサーバーの IP アドレスの特定のオクテットをブロックします。DNS で解決された IP アドレスは、変換 IP アドレスと変換マスクを適用することによって新しい IP アドレスに変換されます。

たとえば、変換 IP アドレスを 10.20.0.0 に、トランスレーションマスクを 255.255.0.0 に設定できます。DNS で解決されたサーバーの IP アドレスが 40.50.27.3 の場合、このアドレスは 10.20.27.3 に変換されます。この場合、変換 IP アドレスは新しいアドレスの最初の 2 オクテットを供給し、マスクは元の IP アドレスの最後の 2 オクテットを通過します。DNS によって解決された元の IP アドレスへの参照は失われます。サーバーがバインドされているすべてのサービスのモニターは、変換された IP アドレスについても報告します。

ドメインベースのサーバーの変換 IP アドレスを設定する場合、DNS で解決された IP アドレスの変換先となるマスクと IP アドレスを指定します。

注:IP アドレスの変換は、ドメインベースのサーバーでのみ可能です。この機能は IP ベースのサーバーには使用できません。アドレスパターンは IPv4 アドレスのみに基づいて設定できます。

コマンドラインインターフェイスを使用してサーバーの翻訳 **IP** アドレスを設定するには

コマンドプロンプトで入力します。

```
add server <name>@ <serverDomainName> -translationIp <
   translationIPAddress> -translationMask <netMask> -state <ENABLED|
   DISABLED>
<!--NeedCopy-->
```

例**:**

```
add server myMaskedServer www.example.com -translationIp 10.10.10.10 -
   translationMask
255.255.0.0 -state ENABLED
<!--NeedCopy-->
```

構成ユーティリティを使用してサーバーの翻訳 **IP** アドレスを構成するには

[トラフィック管理] **>** [負荷分散] **>** [サーバー] に移動し、ドメインベースのサーバーを作成し、変換 IP アドレスを指定します。

仮想サーバーの **IP** アドレスのマスク

August 15, 2023

仮想サーバには、固定 IP アドレスの代わりにマスクとパターンを設定できます。これにより、マスクとパターンに一致する任意の IP アドレス宛てのトラフィックを、特定の仮想サーバに再ルーティングできます。たとえば、IP アドレスの最初の 3 オクテットを可変にできるマスクを設定して、111.11.11.198、22.22.22.198、および 33.33.33.198 へのトラフィックがすべて同じ仮想サーバに送信されるようにすることができます。

仮想サーバ IP アドレスのマスクを設定すると、ルーティングの変更やインフラストラクチャの変更による仮想サーバの再構成を回避できます。このマスクにより、仮想サーバを大規模に再構成することなく、トラフィックのフローを継続できます。

仮想サーバの IP アドレスのマスクは、ドメインベースサーバの IP アドレスの変換で説明されているサーバの IPパターン定義とは異なります。仮想サーバの IP アドレスマスクの場合、ゼロ以外のマスクは考慮されるオクテットとして解釈されます。サービスの場合、ゼロ以外の値はブロックされます。

また、仮想サーバーの IP アドレスマスクの場合は、先頭または末尾の値を考慮できます。仮想サーバの IP アドレスマスクが IP アドレスの左側の値を考慮する場合、これをフォワードマスクと呼びます。マスクがアドレスの右側の値を考慮する場合、これはリバースマスクと呼ばれます。

注：NetScaler ADC アプライアンスは、リバースマスク仮想サーバーを評価する前に、すべての前方マスク仮想サーバーを評価します。

仮想サーバの IP アドレスをマスキングする場合は、着信トラフィックを正しい仮想サーバと照合するための IP アドレスパターンも作成する必要があります。アプライアンスは、着信 IP パケットを受信すると、パケット内の宛先 IP アドレスと IP アドレスパターンで考慮されるビットとを照合し、一致が見つかった後、IP アドレスマスクを適用して最終的な宛先 IP アドレスを構築します。

次の例を考えてみましょう。

- 着信パケットの宛先 IP アドレス：10.102.27.189
- IP アドレスパターン:10.102.0.0
- IP マスク：255.255.0.0
- 構築された（最終的な）宛先 IP アドレス:10.102.27.189。

この場合、元の宛先 IP アドレスの最初の 16 ビットがこの仮想サーバーの IP アドレスパターンと一致するため、この着信パケットはこの仮想サーバーにルーティングされます。

宛先 IP アドレスが複数の仮想サーバの IP パターンと一致する場合は、最長の一致が優先されます。次の例を考えてみましょう。

- 仮想サーバ 1：IP パターン 10.10.0.0、IP マスク 255.255.0.0
- 仮想サーバ 2：IP パターン 10.10.10.0、IP マスク 255.255.255.0
- パケット内の宛先 IP アドレス：10.10.10.45。
- 選択した仮想サーバー: 仮想サーバー 2。

仮想サーバ 2 に関連付けられたパターンは、仮想サーバ 1 に関連付けられたビットよりも多くのビットと一致するため、一致する IP は仮想サーバ 2 に送信されます。

注：タイブレーカーが必要な場合は、ポートも考慮されます。

コマンドラインインターフェイスを使用して仮想サーバーの **IP** アドレスマスクを構成するには

コマンドプロンプトで入力します。


```
add lb vserver <name>@ http -ipPattern <ipAddressPattern> -ipMask <
    ipMask> <listenPort>
<!--NeedCopy-->
```

例**:**

プレフィックスオクテットに基づくパターンマッチング:

```
add lb vserver myLBVserver http -ippattern 10.102.0.0 -ipmask
    255.255.0.0 80
<!--NeedCopy-->
```

末尾のオクテットに基づくパターンマッチング:

```
add lb vserver myLBVserver1 http -ippattern 0.0.22.74 -ipmask
    0.0.255.255 80
<!--NeedCopy-->
```

パターンベースの仮想サーバーを変更します。

```
set lb vserver myLBVserver1 -ippattern 0.0.22.74 -ipmask 0.0.255.255
<!--NeedCopy-->
```

仮想サーバ 1 を次のように設定するとします。

```
add lb vserver vs1 HTTP -ippattern 100.1.1.0 -ipmask 255.255.255.0 80
<!--NeedCopy-->
```

NetScaler ADC アプライアンスは、すべての IP アドレスに対する ARP 要求に応答しません。ただし、このパターン内のすべての IP アドレスにルーティングされる仮想サーバトラフィックに応答します。

構成ユーティリティを使用して仮想サーバの **IP** アドレスマスクを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. [アドレスタイプ] リストで、[IP パターン] を選択し、IP パターンと IP マスクを指定します。

一般的に使用されるプロトコルの負荷分散を構成する

August 15, 2023

ウェブサイトや Web ベースのアプリケーションに加えて、他の一般的なプロトコルを使用する他のタイプのネットワーク展開アプリケーションは、多くの場合、大量のトラフィックを受信するため、ロードバランシングの恩恵を受けます。これらのプロトコルのいくつかは、ロードバランシングを適切に動作させるために特定の設定を必要とします。その中には、FTP、DNS、SIP、および RTSP があります。

NetScaler ADC アプライアンスを IP ではなくサーバーのドメイン名を使用するように構成する場合は、それらのサーバーの IP 変換とマスキングも設定する必要があります。

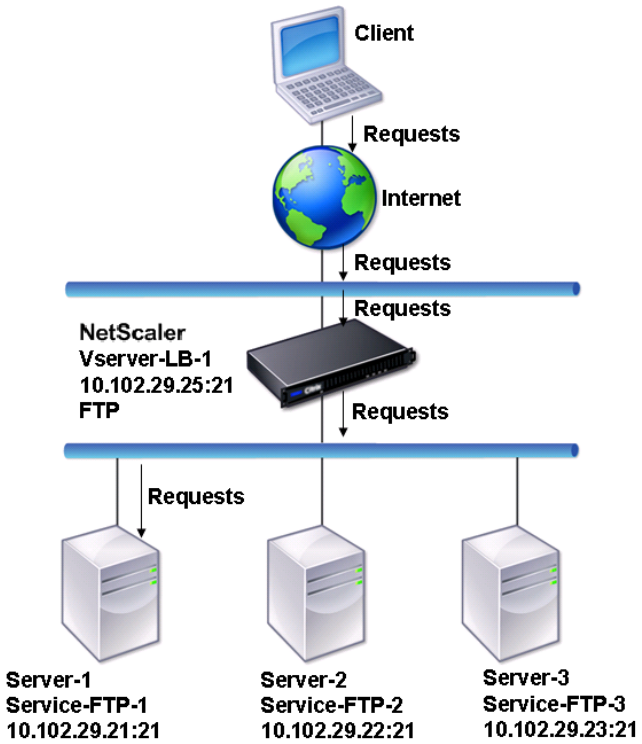
FTP サーバーのグループを負荷分散する

August 15, 2023

NetScaler アプライアンスは FTP サーバーの負荷分散に使用できます。FTP では、ユーザーが同じサーバーへの 2 つの異なるポートで 2 つの接続を開始する必要があります。1 つはクライアントがサーバーにコマンドを送信する制御接続で、もう 1 つはサーバーがクライアントにデータを送信するデータ接続です。クライアントが FTP サーバーへの制御接続を開いて FTP セッションを開始すると、アプライアンスは設定された負荷分散方法を使用して FTP サービスを選択し、制御接続を FTP に転送します。次に、負荷分散された FTP サーバーは、情報交換のためにクライアントへのデータ接続を開きます。

次の図は、FTP サーバーのグループの負荷分散構成のトポロジを示しています。

図 1：FTP サーバの基本的なロードバランシングトポロジ



この図では、Service-FTP-1、Service-FTP-2、および Service-FTP-3 が仮想サーバ Vserver-LB-1 にバインドされています。VServer-LB-1 は、クライアントの接続要求を最小限の接続負荷分散方法を使用してサービスの 1 つに

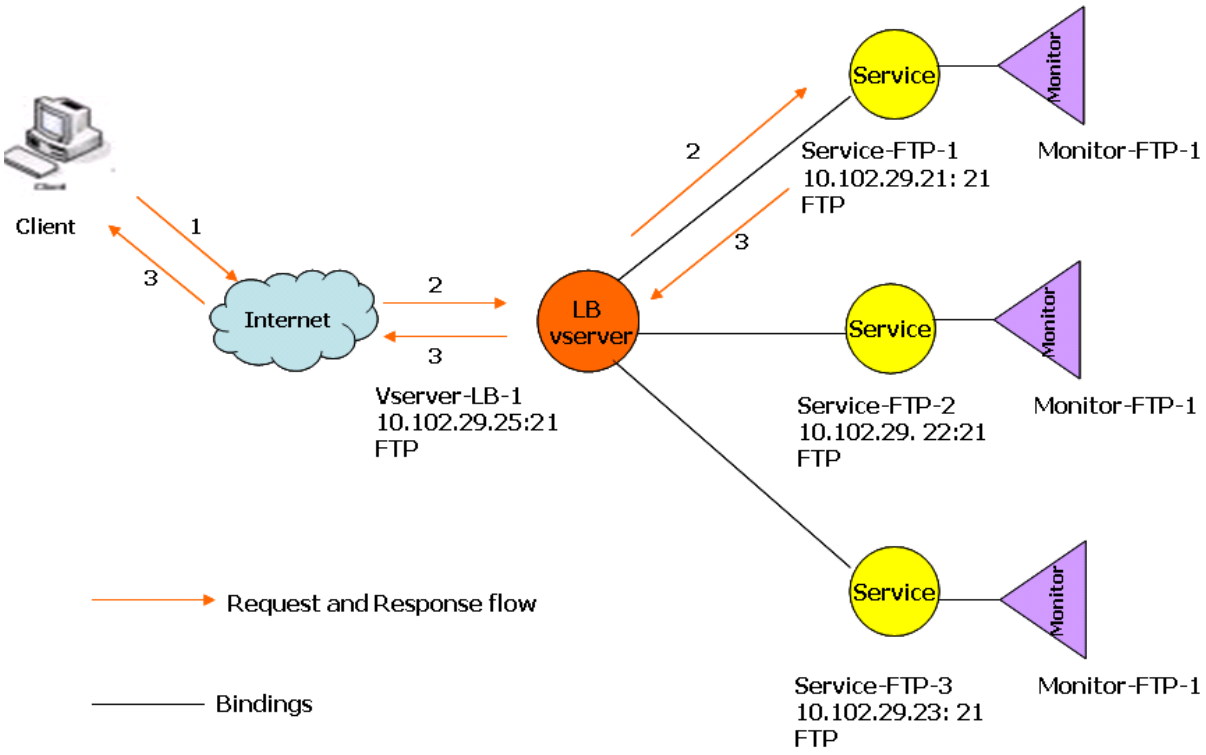
転送します。それ以降のリクエストは、アプライアンスが最初に負荷分散用に選択したサービスに転送されます。

次の表は、アプライアンスに設定されている基本エンティティの名前と値を示しています。

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
Vserver	Vserver-LB-1	10.102.29.25	21	FTP
Services	Service-FTP-1	10.102.29.21	21	FTP
	Service-FTP-2	10.102.29.22	21	FTP
	Service-FTP-3	10.102.29.23	21	FTP
モニター	FTP	なし	なし	なし

次の図は、負荷分散エンティティと、アプライアンスで設定する必要があるパラメータの値を示しています。

図 2: 負荷分散 FTP サーバーエンティティモデル



アプライアンスは、ファイアウォールの外側から FTP サーバーにアクセスするためのパッシブ FTP オプションを提供することもできます。クライアントがパッシブ FTP オプションを使用して FTP サーバーへの制御接続を開始すると、FTP サーバーはクライアントへの制御接続も開始します。その後、ファイアウォール経由でファイルを転送するためのデータ接続を開始します。

FTP タイプのサービスおよび仮想サーバーを作成するには、基本的な負荷分散の設定を参照してください。エンティティに名前を付け、パラメータを前の表の列で説明した値に設定します。基本的な負荷分散セットアップを構成する

と、デフォルトのモニターがサービスにバインドされます。

次に、「モニタをサービスへのバインド」の項で説明されている手順に従って、FTP モニタをサービスにバインドします。

CLI を使用して FTP モニタを作成するには

コマンドプロンプトで入力します。

```
add lb monitor <MonitorName> FTP -interval <Interval> -userName <
    UserName> -password <Password>
<!--NeedCopy-->
```

例:

```
add lb monitor monitor-FTP-1 FTP -interval 360 -userName User -password
    User
<!--NeedCopy-->
```

GUI を使用して FTP モニタを作成するには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. FTP タイプのモニターを作成し、「特殊パラメータ」にユーザー名とパスワードを指定します。

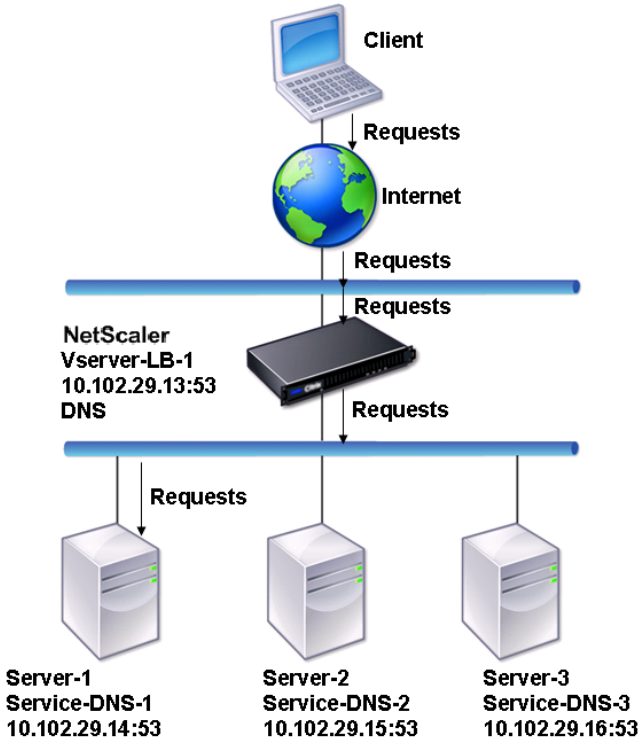
DNS サーバーを負荷分散する

August 15, 2023

ドメイン名の DNS 解決をリクエストすると、NetScaler アプライアンスは構成済みの負荷分散方法を使用して DNS サービスを選択します。次に、サービスがバインドされている DNS サーバーがドメイン名を解決し、応答として IP アドレスを返します。アプライアンスは DNS 応答をキャッシュし、キャッシュされた情報を使用して、同じドメイン名の解決を求める将来の要求に応答することもできます。DNS サーバーの負荷分散 DNS 応答時間が短縮されます。

次の図は、DNS サービスのグループを負荷分散する負荷分散構成のトポロジを示しています。

図 1: DNS サーバーの基本的な負荷分散トポロジ

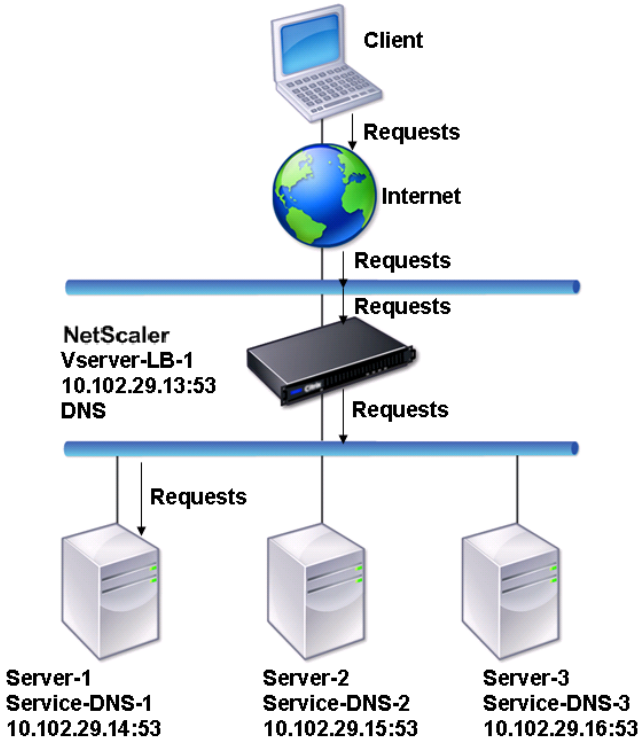


この図では、サービス DNS-1、サービス DNS-2、およびサービス DNS-3 が仮想サーバ Vserver-LB-1 にバインドされています。仮想サーバー Vserver-LB-1 は、最小接続負荷分散方式を使用してクライアント要求をサービスに転送します。次の表は、アプライアンスに設定されている基本エンティティの名前と値を示しています。

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.13	53	DNS
Services	Service-DNS-1	10.102.29.14	53	DNS
	Service-DNS-2	10.102.29.15	53	DNS
	Service-DNS-3	10.102.29.16	53	DNS
モニター	monitor-DNS-1	なし	なし	なし

次の図は、負荷分散エンティティと、アプライアンスで設定する必要があるパラメータの値を示しています。

図 2: 負荷分散 DNS サーバーエンティティモデル



基本的な DNS 負荷分散設定を構成するには、基本的な負荷分散の設定を参照してください。手順に従って、DNS タイプのサービスと仮想サーバーを作成し、エンティティに名前を付けて、前の表で説明した値を使用してパラメータを設定します。基本的な負荷分散設定を構成すると、デフォルトの ping モニターがサービスにバインドされます。DNS モニタを DNS サービスにバインドする手順については、「モニタをサービスへのバインド」を参照してください。

次の手順では、クエリに基づいてドメイン名を IP アドレスにマッピングするモニターを作成する手順について説明します。

CLI を使用して DNS モニタを設定するには

コマンドプロンプトで入力します。

```
1 add lb monitor <monitorName> DNS -query <domainName> -queryType <
     Address|ZONE> -IPAddress <ipAddress>
2 <!--NeedCopy-->
```

例:

```
1 add lb monitor monitor-DNS-1 DNS -query www.citrix.com -queryType
     Address -IPAddress 10.102.29.66
2
```



```
add lb monitor monitor-DNS-2 DNS -query www.citrix2.com -queryType
   Address -IPAddress
1000:0000:0000:0000:0005:0600:700a::888b-888d
<!--NeedCopy-->
```

GUI を使用して **DNS** モニタを構成するには

1. [トラフィック管理] **>** [負荷分散] **>** [モニター] に移動します。
2. DNS タイプのモニターを作成し、「特殊パラメータ」でクエリとクエリタイプを指定します。

ドメイン名ベースのサービスを負荷分散する

August 15, 2023

負荷分散用のサービスを作成するときに、IP アドレスを指定できます。または、ドメイン名を使用してサーバーを作成することもできます。サーバー名（ドメイン名）は、IPv4 または IPv6 ネームサーバーを使用するか、権限のある DNS レコード（IPv4 の場合は A レコードまたは IPv6 の場合は AAAA レコード）を NetScaler 構成に追加することで解決できます。

IP アドレスではなくドメイン名を使用してサービスを構成し、ネームサーバーがドメイン名を新しい IP アドレスに解決する場合、サービスにバインドされたモニターは新しい IP アドレスに対してヘルスチェックを実行し、IP アドレスが正常であることが検出された場合にのみサービス IP アドレスを更新します。モニタは、サービスにバインドされたデフォルトのモニタにすることも、サポートされているその他のモニタをバインドすることもできます。監視パラメータで定義された一定の間隔でサービスをプローブします。ドメイン名が新しい IP アドレスに変換されると、モニターは新しいプローブを送信してサービスの状態を確認します。それ以降のプローブはすべて、事前に定義された間隔で行われます。

注: サーバーの IP アドレスを変更すると、最初のクライアントリクエストに対応するサービスがマークダウンされます。ネームサーバーは、次のリクエスト用にサービスの IP アドレスを変更された IP アドレスに変換し、サービスが UP とマークされます。

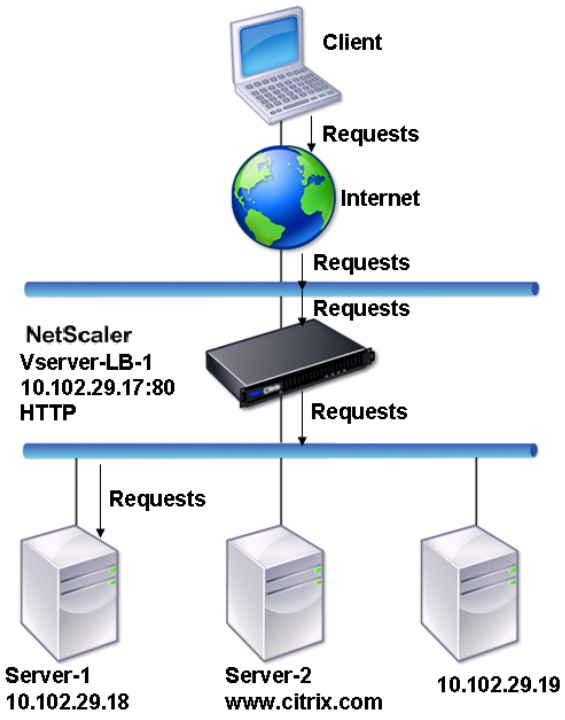
ドメイン名ベースのサービスには次の制限があります。

- ドメイン名の最大長は 255 文字です。
- Maximum Client パラメータは、ドメイン名ベースのサーバーを表すサービスを構成するために使用されます。たとえば、仮想サーバーにバインドされたサービスには MaxClient が 1000 に設定されます。仮想サーバーの接続数が 2000 に達すると、DNS リゾルバーはサービスの IP アドレスを変更します。ただし、サービスの接続カウンタはリセットされないため、仮想サーバーは古い接続がすべて閉じられるまで新しい接続を確立できません。
- サービスの IP アドレスが変更されると、永続性を維持するのが難しくなります。
- タイムアウトによりドメイン名解決が失敗した場合、アプライアンスは古い情報（IP アドレス）を使用します。

- 監視によってサービスが停止していることが検出されると、アプライアンスはサービス（ドメイン名ベースのサーバーを表す）に対して DNS 解決を実行して新しい IP アドレスを取得します。
- 統計情報はサービスで収集され、IP アドレスが変更されてもリセットされません。
- DNS 解決が「名前エラー」（3）のコードを返した場合、アプライアンスはサービスをマークダウンし、IP アドレスをゼロに変更します。

アプライアンスは、サービスの要求を受信すると、ターゲットサービスを選択します。このようにして、アプライアンスはサービスの負荷のバランスをとります。次の図は、ドメインネームベースサーバー (DBS) のグループをロードバランシングするロードバランシング構成のトポロジを示しています。

図 1：DBS サーバの基本的なロードバランシングトポロジ



Service-HTTP-1、Service-HTTP-2、および Service-HTTP-3 は、仮想サーバ Vserver-LB-1 にバインドされます。仮想サーバー vserver-LB-1 は、サービスの選択に最小接続負荷分散方式を使用します。サービスの IP アドレスは、ネームサーバー Vserver-LB-2 を使用して解決されます。

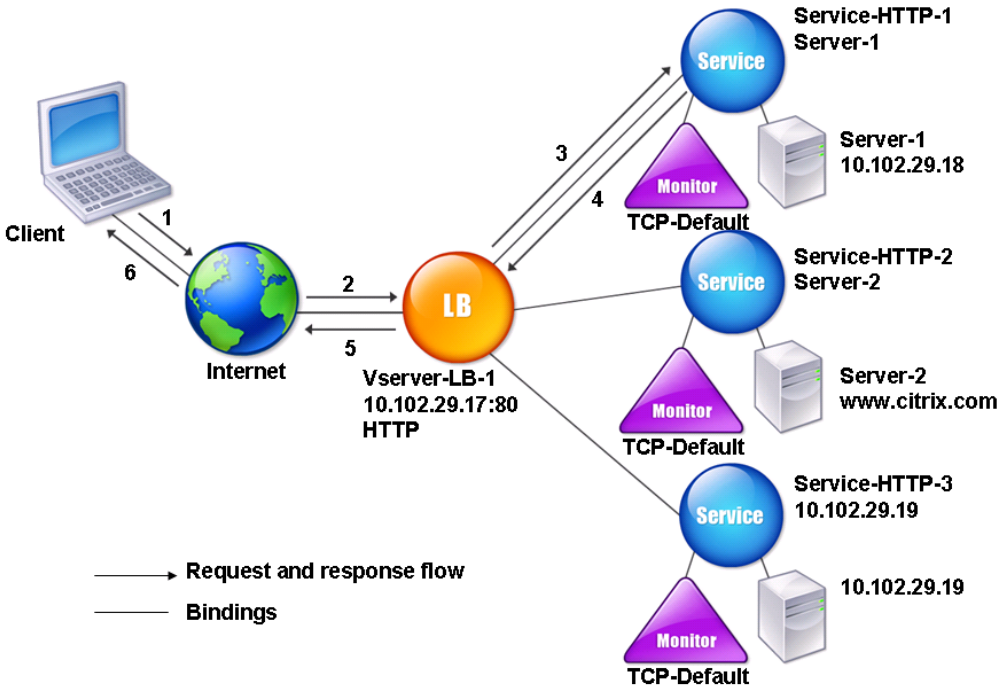
次の表は、アプライアンスに設定されている基本エンティティの名前と値を示しています。

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.17	80	HTTP

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
	Vserver-LB-2	10.102.29.20	53	DNS
サーバー	server-1	10.102.29.18	80	HTTP
	server-2	www.citrix.com	80	HTTP
Services	Service-HTTP-1	server-1	80	HTTP
	Service-HTTP-2	server-2	80	HTTP
	Service-HTTP-2	10.102.29.19	80	HTTP
モニター	デフォルト	なし	なし	なし
ネームサーバー	なし	10.102.29.19	なし	なし

次の図は、負荷分散エンティティと、アプライアンスで設定する必要があるパラメータの値を示しています。

図 2: 負荷分散 DBS サーバーエンティティモデル



基本的な負荷分散設定を構成するには、基本負荷分散の設定を参照してください。HTTP タイプのサービスと仮想サーバーを作成し、エンティティに名前を付け、前の表で説明した値を使用してパラメータを設定します。

外部ネームサーバーを追加、削除、有効化、および無効化することができます。IP アドレスを指定してネームサーバーを作成するか、既存の仮想サーバーをネームサーバーとして設定できます。

コマンドラインインターフェイスを使用してネームサーバーを追加するには

コマンドプロンプトで入力します。

```
add dns nameServer <dnsVserverName>
<!--NeedCopy-->
```

例**:**

```
add dns nameServer Vserver-LB-2
<!--NeedCopy-->
```

構成ユーティリティを使用してネームサーバーを追加するには

1. [****** トラフィック管理 ******] > [**DNS**] > [****** ネームサーバー ******] に移動します。
2. DNS 仮想サーバータイプの DNS ネームサーバーを作成し、DNS 仮想サーバーリストからサーバーを選択します。

ドメイン名を IP アドレスに解決する権限のあるネームサーバーを追加することもできます。

> 注
>
> TCP、UDP、または UDP_TCP タイプのネームサーバーをリゾルバ DBS プローブに追加できます。ただし、TCP と UDP ネームサーバーが共存し、UDP ネームサーバーが切り捨てられたビットの応答を受信した場合、この応答は TCP ネームサーバー上で再試行されません。

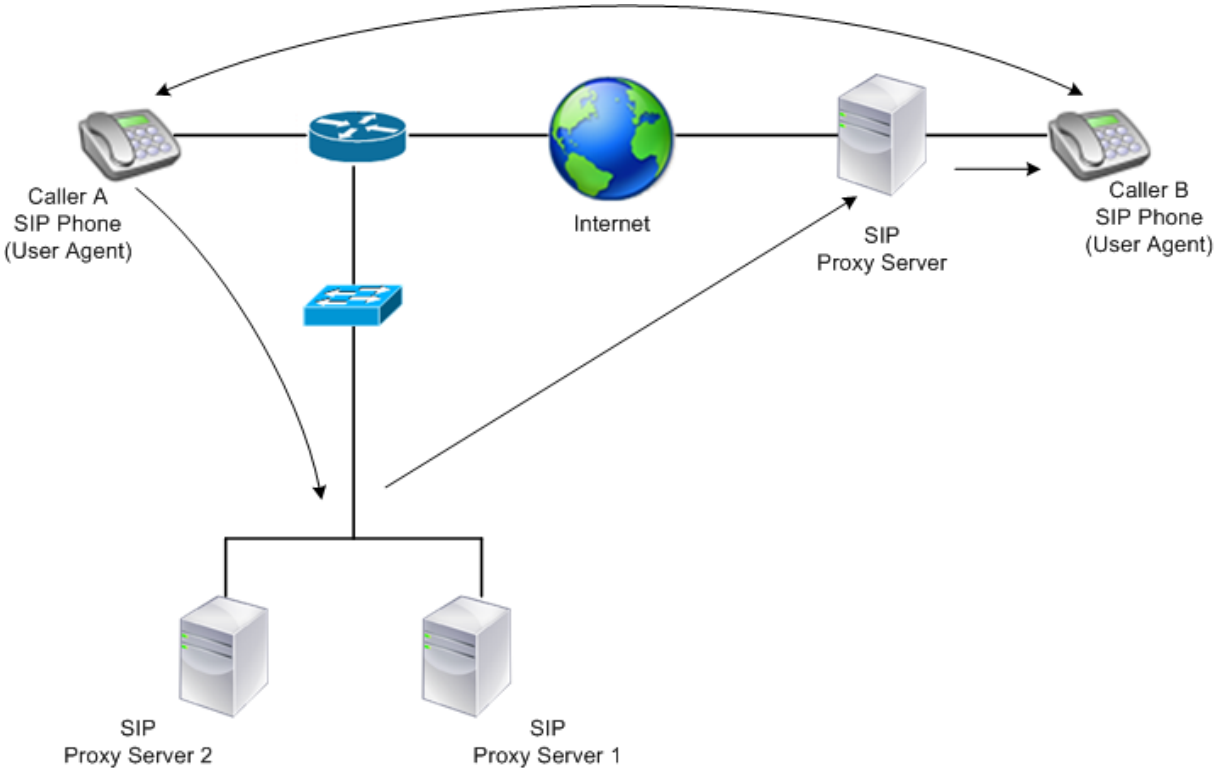
SIP サーバーのグループを負荷分散する

August 15, 2023

セッション開始プロトコル (SIP) は、マルチメディア通信セッションを開始、管理、および終了するように設計されています。インターネットテレフォニー（VoIP）の標準として登場しました。SIP メッセージは TCP または UDP 経由で送信できます。SIP メッセージには、要求メッセージと応答メッセージの 2 種類があります。

SIP ベースの通信システムのトラフィックは、専用のデバイスとアプリケーション（エンティティ）を介してルーティングされます。マルチメディア通信セッションでは、これらのエンティティはメッセージを交換します。次の図は、基本的な SIP ベースの通信システムを示しています。

図 1: SIP ベースの通信システム



NetScaler ADC を使用すると、UDP または TCP（TLS を含む）経由で SIP メッセージを負荷分散できます。NetScaler は、SIP リクエストを SIP プロキシサーバーのグループに負荷分散するように構成できます。そのためには、負荷分散方法と永続性のタイプを次の組み合わせのいずれかに設定した負荷分散仮想サーバーを作成します。

- パーシスタンス設定なしの Call-ID ハッシュ負荷分散方式
- 最小接続またはラウンドロビン負荷分散方式によるコール ID ベースのパーシスタンス
- 最小接続またはラウンドロビン負荷分散方式によるルールベースのパーシステンス

また、デフォルトでは、NetScaler ADC は SIP 要求のヘッダーを介して RPORT を追加し、サーバーがリクエストの送信元 IP アドレスとポートに応答を返信します。

注：ロードバランシングが機能するには、プライベート IP アドレスまたはプライベートドメインが SIP ヘッダー/ペイロードに追加されないように SIP プロキシを設定する必要があります。SIP プロキシは、SIP 仮想サーバーの IP アドレスに対応するドメイン名を SIP ヘッダーに追加する必要があります。また、SIP プロキシは共通のデータベースと通信して登録情報を共有する必要があります。

サーバー開始トラフィック

SIP サーバーから開始されるアウトバウンドトラフィックの場合は、クライアントが使用するプライベート IP アドレスがパブリック IP アドレスに変換されるように、NetScaler で RNAT を構成します。

RNAT 送信元ポートまたは宛先ポートを含む SIP パラメータを設定した場合、アプライアンスは要求パケットの送信元ポートと宛先ポートの値を RNAT 送信元ポートおよび RNAT 宛先ポートと比較します。いずれかの値が一致する

と、アプライアンスは VIA ヘッダーを RPORT で更新します。その後、クライアントからの SIP 応答は要求と同じパスを通過します。

サーバーから開始される SSL トラフィックの場合、NetScaler は組み込みの証明書とキーのペアを使用します。カスタムの証明書とキーのペアを使用する場合は、カスタム証明書とキーのペアを **nsrnatsip-127.0.0.1-5061** という名前の **NetScaler** 内部サービスにバインドします。

ポリシーと式のサポート

NetScaler デフォルト式言語には、セッション開始プロトコル（SIP）接続で動作するいくつかの式が含まれています。これらの式は、SIP ベース (sip_udp、sip_tcp、sip_ssl) 仮想サーバーおよびグローバルバインドポイントにのみバインドできます。これらの表現は、コンテンツスイッチング、レート制限、レスポンダー、およびリライトポリシーで使用できます。

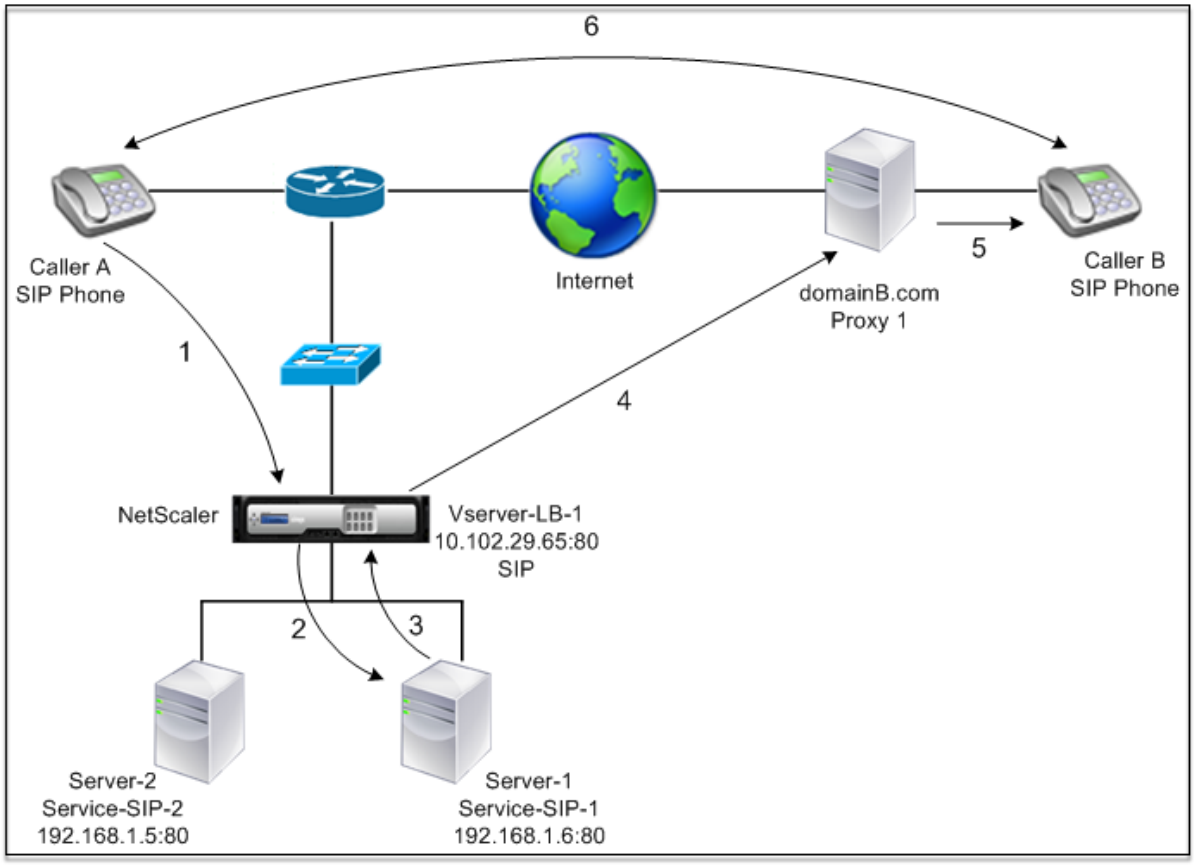
TCP または **UDP** 経由の **SIP** シグナリングトラフィックのロードバランシングの設定

NetScaler は、TLS で保護された TCP トラフィックを含め、UDP または TCP を介してリクエストを送信する SIP サーバーの負荷分散を行うことができます。ADC は、SIP サーバーの負荷分散用に次のサービスタイプを提供します。

- SIP_UDP —SIP サーバーが UDP 経由で SIP メッセージを送信する場合に使用されます。
- SIP_TCP —SIP サーバーが TCP 経由で SIP メッセージを送信する場合に使用されます。
- SIP_SSL —SSL または TLS を使用して TCP 経由の SIP シグナリングトラフィックを保護するために使用されます。NetScaler は以下のモードをサポートしています。
  - クライアント、ADC、SIP サーバー間のエンドツーエンド TLS 接続。
  - クライアントと ADC 間の TLS 接続、および ADC と SIP サーバー間の TCP 接続。
  - クライアントと ADC 間の TCP 接続、および ADC と SIP サーバー間の TLS 接続。

次の図は、TCP または UDP を介して SIP メッセージを送信する SIP サーバーのグループを負荷分散するように構成されたセットアップのトポロジを示しています。

図 2: SIP ロードバランシングトポロジ



エンティティタイプ	名前	IP アドレス	ポート	サービスタイプ/プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.65	80	SIP_UDP /SIP_TCP /SIP_SSL
Services	Service-SIP-1	192.168.1.6	80	SIP_UDP /SIP_TCP /SIP_SSL
	Service-SIP-2	192.168.1.5	80	SIP_UDP /SIP_TCP /SIP_SSL
モニター	デフォルト	なし	80	SIP_UDP /SIP_TCP /SIP_SSL

SIP トラフィックの基本的なロードバランシング設定の概要は次のとおりです。

1. サービスを構成し、負荷分散する SIP トラフィックの種類ごとに仮想サーバーを構成します。

- **SIP_UDP** —**UDP** 経由で SIP トラフィックの負荷分散を行う場合。
- **SIP_TCP** —**TCP** 経由で SIP トラフィックの負荷分散を行う場合。
- **SIP_SSL** —**TCP** 経由で SIP トラフィックのロードバランシングとセキュリティ保護を行う場合。

注:SIP_SSL を使用する場合は、必ず SSL 証明書とキーのペアを作成してください。詳細については、「証明書キーペアの追加」を参照してください。

2. サービスを仮想サーバーにバインドします。
3. デフォルト (**tcp-default**) 以外のモニターでサービスの状態を監視する場合は、カスタムモニターを作成してサービスにバインドします。NetScaler には、****SIP** サービスを監視するための **SIP-UDP** と **SIP-TCP** の **2** 種類のカスタムモニタータイプが用意されています **。
4. SIP_SSL 仮想サーバーを使用する場合は、SSL 証明書とキーのペアを仮想サーバーにバインドします。
5. 展開環境内の SIP サーバーのゲートウェイとして NetScaler を使用している場合は、RNAT を構成します。
6. SIP サーバから送信される SIP メッセージに RPORT を追加する場合は、SIP パラメータを設定します。

コマンドラインインターフェイスを使用して **SIP** トラフィックの基本的な負荷分散設定を構成するには

1 つ以上のサービスを作成します。コマンドプロンプトで入力します。

```
add service <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
<!--NeedCopy-->
```

例:

```
add service Service-SIP-UDP-1 192.0.2.5 SIP_UDP 80
<!--NeedCopy-->
```

作成したサービスを処理するために必要な数の仮想サーバーを作成します。仮想サーバのタイプは、バインドするサービスのタイプと一致する必要があります。コマンドプロンプトで入力します。

```
add lb vserver <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 SIP_UDP 10.102.29.60 80
<!--NeedCopy-->
```

各サービスを仮想サーバーにバインドします。コマンドプロンプトで入力します。

```
bind lb vserver <name> <serverName>
<!--NeedCopy-->
```

例:


```
bind lb vserver Vserver-LB-1 Service-SIP-UDP-1
<!--NeedCopy-->
```

(オプション) SIP-UDP または SIP-TCP タイプのカスタムモニターを作成し、そのモニターをサービスにバインドします。コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <monitorType> [<interval>]

bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

例:

```
add lb monitor mon1 sip-UDP -sipMethod REGISTER -sipuRI sip:mon@test.
   com -sipregURI sip:mon@test.com -respcode 200

bind monitor mon1 Service-SIP-UDP-1
<!--NeedCopy-->
```

SIP_SSL 仮想サーバーを作成した場合は、SSL 証明書キーペアを仮想サーバーにバインドします。コマンドプロンプトで、次のように入力します。コマンドプロンプトで、次のように入力します。

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -
   CA  - skipCAName
<!--NeedCopy-->
```

例:

```
bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1
<!--NeedCopy-->
```

ネットワークトポロジの必要に応じて RNAT を設定します。コマンドプロンプトで、次のコマンドのいずれかを入力して、ネットワークアドレスを条件として、SNIP を NAT IP アドレスとして使用する RNAT エントリ、条件としてネットワークアドレスを使用し、NAT IP アドレスとして固有の IP アドレスを使用する RNAT エントリ、条件として ACL を使用し、SNIP を NAT IP アドレスとして使用する RNAT エントリ、または ACL を使用する RNAT エントリをそれぞれ作成します。条件として、また固有の IP アドレスを NAT IP アドレスとして指定すると、

```
add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))

bind rnat <name> <natIP>@ …

show rnat
<!--NeedCopy-->
```

例:

```
add rnat RNAT-1 192.168.1.0 255.255.255.0

bind rnat RNAT-1 -natip 10.102.29.50
<!--NeedCopy-->
```


カスタムの証明書とキーのペアを使用する場合は、カスタム証明書とキーのペアを nsrnatsip-127.0.0.1-5061 という名前の NetScaler 内部サービスにバインドします。

```
add ssl certKey <certkeyName> -cert <string> [-key <string>]

bind ssl service <serviceName> -certkeyName <string>
<!--NeedCopy-->
```

例:

```
add ssl certKey c1 -cert cert.epm -key key.ky

bind ssl service nsrnatsip-127.0.0.1-5061 -certkeyName c1
<!--NeedCopy-->
```

SIP サーバーが開始する SIP メッセージに RPORT を追加する場合は、コマンドプロンプトで次のコマンドを入力します。

```
set lb sipParameters -rnatSrcPort <rnatSrcPort> -rnatDstPort<
    rnatDstPort> -retryDur <integer> -addRportVip <addRportVip> -
    sip503RateThreshold <sip503_rate_threshold_value>
<!--NeedCopy-->
```

UDP 経由の SIP トラフィックのロードバランシングの設定例

```
add service service-UDP-1 10.102.29.5 SIP_UDP 80

Done

add lb vserver vserver-LB-1 SIP_UDP 10.102.29.60 80

Done

bind lb vserver vserver-LB-1 service-UDP-1

Done

add lb mon mon1 sip-udp -sipMethod REGISTER -sipuRI sip:mon@test.com -
    sipregURI sip:mon@test.com -respcode 200

Done

bind mon mon1 service-UDP-1

Done

add rnat RNAT-1 192.168.1.0 255.255.255.0

Done

```



```
set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
    -addRportVip ENABLED -sip503RateThreshold 1000

Done
<!--NeedCopy-->
```

TCP 経由の **SIP** トラフィックのロードバランシングの設定例

```
add service service-TCP-1 10.102.29.5 SIP_TCP 80

Done

add lb vserver vserver-LB-1 SIP_TCP 10.102.29.60 80

Done

bind lb vserver vserver-LB-1 service-TCP-1

Done

add lb mon mon1 sip-tcp -sipMethod REGISTER -sipuRI sip:mon@test.com -
   sipregURI sip:mon@test.com -respcode 200

Done

bind mon mon1 service-TCP-1

Done

add rnat RNAT-1 192.168.1.0 255.255.255.0

Done

set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
    -addRportVip ENABLED -sip503RateThreshold 1000

Done
<!--NeedCopy-->
```

TCP を介した **SIP** トラフィックのロードバランシングおよびセキュリティ保護の設定例

```
add service service-SIP-SSL-1 10.102.29.5 SIP_SSL 80

Done

add lb vserver vserver-LB-1 SIP_SSL 10.102.29.60 80

Done

```



```
bind lb vserver vserver-LB-1 service-SIP-SSL

Done

add lb mon mon1 sip-tCP -sipMethod REGISTER -sipuRI sip:mon@test.com -
    sipregURI sip:mon@test.com -respcode 200

Done

bind mon mon1 service-SIP-SSL

Done

bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1

Done

add rnat RNAT-1 192.168.1.0 255.255.255.0

Done

set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
     -addRportVip ENABLED -sip503RateThreshold 1000

Done
<!--NeedCopy-->
```

GUI を使用して **SIP** トラフィックの基本的なロードバランシング設定を構成するには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、タイプ SIP_UDP、SIP_TCP、または SIP_SSL の仮想サーバーを追加します。
2. [サービス] セクションをクリックし、タイプ SIP_UDP、SIP_TCP、または SIP_SSL のサービスを追加します。
3. （オプション）[**Monitor**] セクションをクリックし、タイプ SIP-UDP または SIP-TCP のモニタを追加します。
4. モニターをサービスにバインドし、サービスを仮想サーバーにバインドします。
5. SIP_SSL 仮想サーバーを作成した場合は、SSL 証明書キーペアを仮想サーバーにバインドします。証明書セクションをクリックし、証明書キーペアを仮想サーバーにバインドします。
6. ネットワークトポロジの必要に応じて RNAT を設定します。RNAT を設定するには:

   a) [システム] **>** [ネットワーク] **>** [ルート] に移動します。
   b) [ルート] ページで、[**RNAT**] タブをクリックします。
   c) 詳細ウィンドウで、[**RNAT** の構成] をクリックします。
   d) RNAT の設定ダイアログボックスで、次のいずれかを実行します。

- RNAT エントリを作成するための条件としてネットワークアドレスを使用する場合は、[ネットワーク] をクリックし、次のパラメータを設定します。
  - ネットワーク
  - ネットマスク
- 拡張 ACL を RNAT エントリを作成するための条件として使用する場合は、[**ACL**] をクリックし、次のパラメータを設定します。
  - ACL 名
  - リダイレクトポート

e) SNIP アドレスを NAT IP アドレスとして設定するには、ステップ 7 に進みます。

f) 一意の IP アドレスを NAT IP として設定するには、[使用可能な NAT IP] の一覧で、NAT IP として設定する IP アドレスを選択し、[追加] をクリックします。選択した NAT IP が [構成された NAT IP] リストに表示されます。

g) ［Create］をクリックしてから、［Close］をクリックします。

カスタムの証明書とキーのペアを使用する場合は、カスタム証明書とキーのペアを **nsrnatsip-127.0.0.1-5061** という名前の **NetScaler** 内部サービスにバインドします。ペアをバインドするには:

a) [トラフィック管理**] > [**負荷分散**] > [**サービス] に移動し、[内部サービス] タブをクリックします。

b) Select nsrnatsip-127.0.0.1-5061 and click **Edit**.

c) **Certificates** セクションをクリックし、証明書キーペアを内部サービスにバインドします。

7. SIP サーバが開始する SIP メッセージに RPORT を追加する場合は、SIP パラメータを設定します。[トラフィック管理**] > [**ロードバランシング] に移動し、[SIP 設定の変更] をクリックし、さまざまな SIP パラメータを設定します。

SIP 表現とポリシーの例**:** クライアント要求での圧縮の有効化

NetScaler は圧縮されたクライアント SIP 要求を処理できないため、クライアント SIP 要求は失敗します。

クライアントからの SIP NEGITIATE メッセージをインターセプトし、圧縮ヘッダーを探すレスポンダーポリシーを設定できます。メッセージに圧縮ヘッダーが含まれている場合、ポリシーは「400 Bad Request」で応答するため、クライアントはリクエストを圧縮せずに再送信します。

コマンドプロンプトで、次のコマンドを入力してレスポンダーポリシーを作成します。

```
add responder action sipaction1 respondwith q{
 "SIP/2.0 400 Bad Requestrnrn" }


Done

add responder policy sippol1

add responder policy sippol1 "SIP.REQ.METHOD.EQ("NEGOTIATE")&&SIP.REQ.
   HEADER("Compression").EXISTS" sipaction1
```



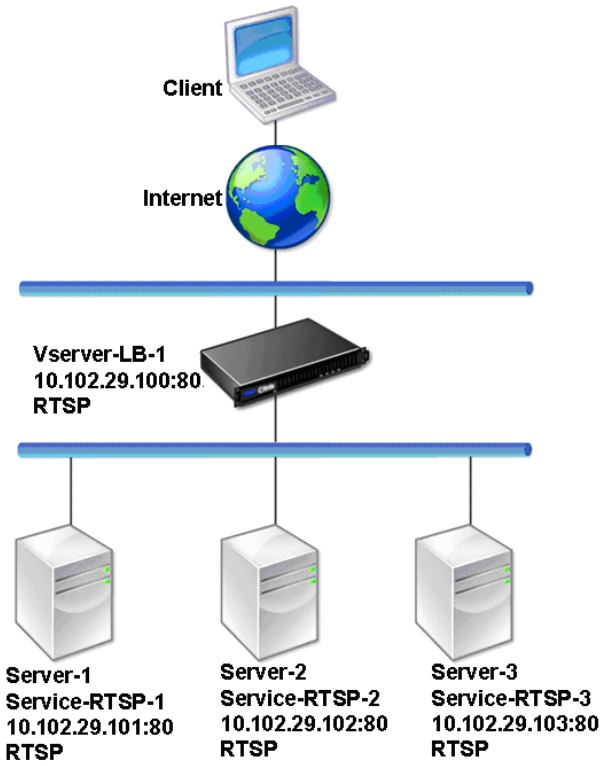
```
<!--NeedCopy-->
```

RTSP サーバーを負荷分散する

August 15, 2023

NetScaler ADC アプライアンスは、RTSP サーバーの負荷を分散して、ネットワークを介したオーディオおよびビデオストリームのパフォーマンスを向上させることができます。次の図は、RTSP サーバのグループを負荷分散するように構成されたロードバランシングセットアップのトポロジを示しています。

図 1: RTSP のロードバランシングトポロジー



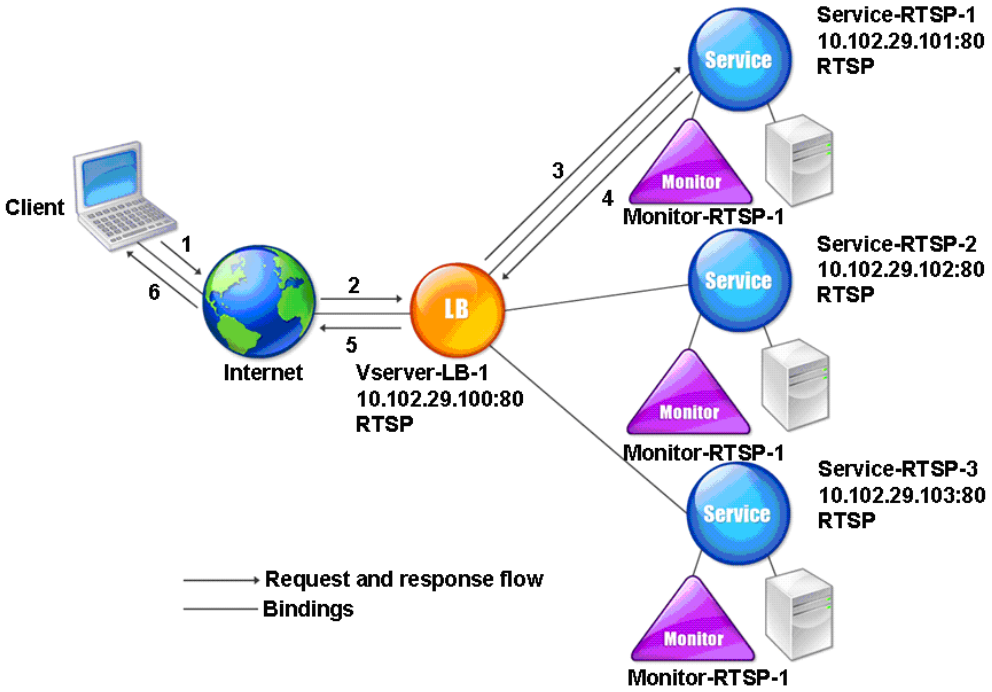
この例では、サービス-RTSP-1、サービス-RTSP-2、およびサービス-RTSP-3 というサービスが仮想サーバー vserver-LB-1 にバインドされています。次の表は、サンプルエンティティの名前と値を示しています。

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.100	554	RTSP

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
Services	Service-RTSP-1	10.102.29.101	554	RTSP
	Service-RTSP-2	10.102.29.102	554	RTSP
	Service-RTSP-3	10.102.29.103	554	RTSP
モニター	Monitor-RTSP-1	なし	554	RTSP

次の図は、RTSP 設定で使用されるロードバランシングエンティティを示しています。

図 2: ロードバランシング RTSP サーバエンティティモデル



RTSP サーバーの基本的な負荷分散設定を構成するには、基本負荷分散の設定を参照してください。RTSP タイプのサービスおよび仮想サーバを作成します。基本的な負荷分散セットアップを構成すると、デフォルトの TCP デフォルトモニターがサービスにバインドされます。RTSP モニタをこれらのサービスにバインドするには、モニタをサービスへのバインドを参照してください。次の手順では、RTSP サーバをチェックするモニタの作成方法について説明します。

CLI を使用して RTSP モニタを設定するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <type>
<!--NeedCopy-->
```

例:

```
add lb monitor Monitor-RTSP-1 RTSP
<!--NeedCopy-->
```

GUI を使用して RTSP モニタを設定するには

[トラフィック管理] > [ロードバランシング] > [モニター] に移動し、RTSP タイプのモニタを作成します。

リモートデスクトッププロトコルサーバーの負荷分散

August 15, 2023

リモートデスクトッププロトコル (RDP) はマルチチャネル対応のプロトコルで、プレゼンテーションデータ、シリアルデバイス通信、ライセンス情報、高度に暗号化されたデータ (キーボードとマウスの操作) などを転送するための個別の仮想チャネルを可能にします。

RDP は、ネットワーク上の別のコンピュータに GUI を提供するために使用されます。RDP は Windows ターミナルサーバーで使用され、低帯域幅の接続でもマウスの動きやキーの押下をほぼリアルタイムで送信し、高速アクセスを実現します。

リモートデスクトップサービスを提供するために複数のターミナルサーバーを展開する場合、NetScaler アプライアンスはターミナルサーバー（Windows 2003 および 2008 Server Enterprise Edition）の負荷分散を行います。アプリケーションにリモートでアクセスしているユーザーが、アプリケーションをリモートマシンで実行したままにして、ローカルマシンをシャットダウンしたい場合があります。したがって、ユーザはリモートアプリケーションからログアウトせずにローカルアプリケーションを閉じます。リモートマシンに再接続した後、ユーザーはリモートアプリケーションを続行できる必要があります。この機能を提供するために、NetScaler RDP 実装では、ターミナルサービスセッションディレクトリまたはブローカーによって設定されたルーティングトークン（Cookie）が優先されるため、クライアントは以前に接続されていたターミナルサーバーに再接続できます。Windows 2003 ターミナルサーバーに実装されたセッションディレクトリは、Windows 2008 ターミナルサーバーではブローカーと呼ばれます。

クライアントと負荷分散仮想サーバー間で TCP 接続が確立されると、NetScaler は指定された負荷分散方法を適用し、要求をいずれかのターミナルサーバーに転送します。ターミナルサーバーはセッションディレクトリをチェックして、クライアントのセッションがドメイン内の他のターミナルサーバーで実行されているかどうかを判断します。

他のターミナルサーバーにアクティブなセッションがない場合、ターミナルサーバーはクライアント要求を処理して応答し、NetScaler ADC アプライアンスはクライアントに応答を転送します。

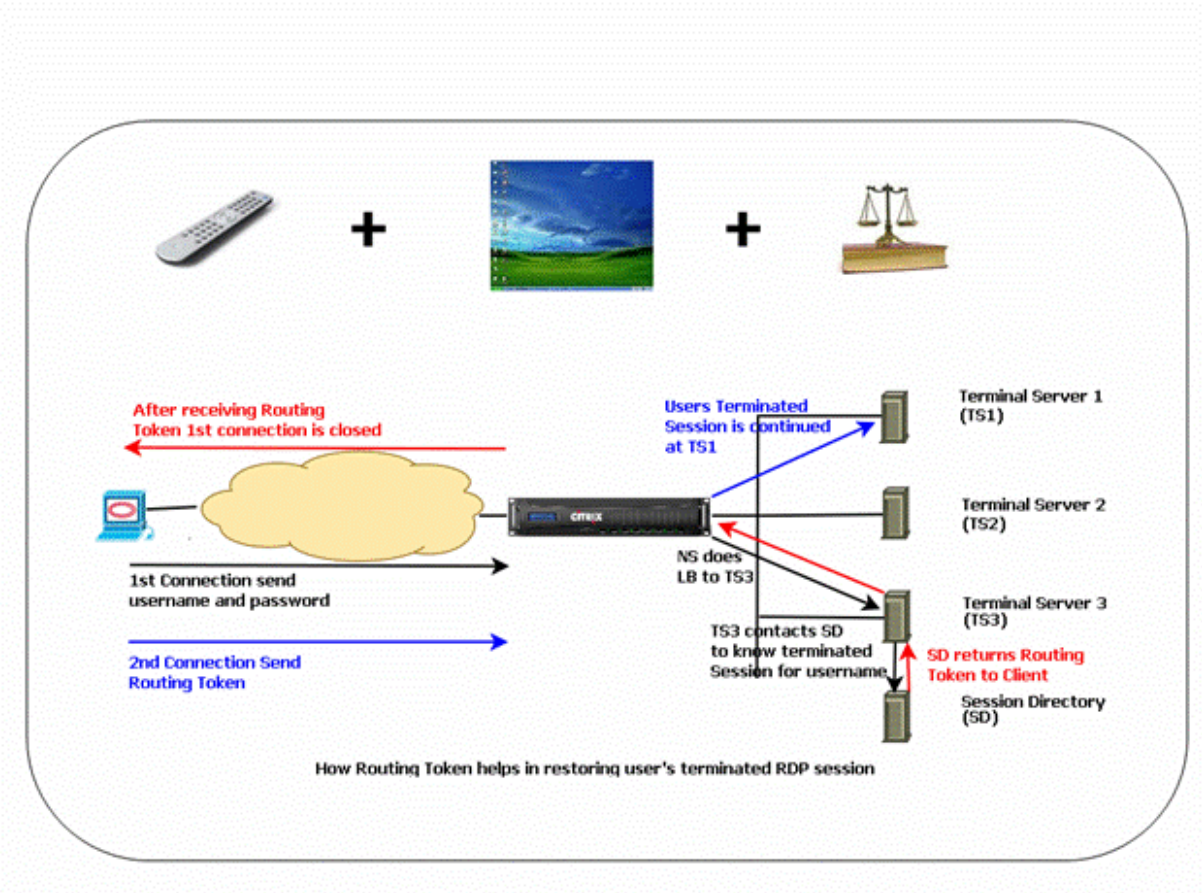
他のターミナルサーバーにアクティブなセッションがある場合、要求を受信するターミナルサーバーはアクティブなセッションの詳細を含むクッキー（ルーティングトークンと呼ばれる）を挿入し、パケットを NetScaler ADC アプライアンスに返し、パケットをクライアントに返します。サーバーはクライアントとの接続を閉じます。クライアントが接続を再試行すると、NetScaler は Cookie 情報を読み取り、クライアントのセッションがアクティブなターミナルサーバーにパケットを転送します。

クライアントマシン上のユーザーにはサービスが継続されるため、特別なアクションを実行する必要はありません。

注:Windows セッションディレクトリ機能には、Windows XP で最初にリリースされたリモートデスクトップクライアントが必要です。Windows 2000 または Windows NT 4.0 ターミナルサーバクライアントとのセッションが切断され、クライアントが再接続すると、接続を確立するサーバが負荷分散アルゴリズムによって選択されます。

次の図は、RDP 負荷分散を示しています。

図 1：RDP の負荷分散トポロジー



注

- RDP サービスを設定すると、ルーティングトークンを使用して永続性が自動的に維持されます。永続性

> を明示的に有効にする必要はありません。
> - NetScaler アプライアンスは IP ベースのクッキーのみをサポートします。
> - nsrdp.pl スクリプトは、現在のバージョンの Windows サーバではサポートされていません。

RDP セッションがログアウトせずに切断された場合に 2 つのターミナルサーバー間でフラッピングが発生しないように、切断された RDP セッションがバックエンドのターミナルサーバーでクリアされていることを確認してください。詳しくは、https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc758177(v=ws.10)#BKMK_2を参照してください。

RDP サービスを追加すると、NetScaler はデフォルトで TCP タイプのモニターを追加し、サービスにバインドします。デフォルトモニターは、RDP サービスに指定されたサーバーの 3389 ポートにリスニングプロセスが存在するかどうかをチェックする単純な TCP モニターです。3389 でリスニングプロセスがある場合、NetScaler はこのサービスを稼働中としてマークし、リスニングプロセスがない場合はサービスを停止中とマークします。

RDP サービスをより効率的に監視するために、デフォルトのモニターに加えて、RDP プロトコル用のスクリプトモニターを構成できます。スクリプトモニターを構成すると、NetScaler は指定されたサーバーへの TCP 接続を開き、RDP パケットを送信します。モニターは、物理サーバーから接続の確認を受け取った場合にのみ、サービスを UP とマークします。そのため、NetScaler はスクリプトモニターから、RDP サービスがリクエストを処理する準備ができているかどうかを知ることができます。

モニターはユーザータイプのモニターで、スクリプトは NetScaler の/nsconfig/monitors/nsrdp.pl にあります。ユーザーモニターを構成すると、NetScaler はスクリプトを自動的に実行します。スクリプトモニターを設定するには、モニターを追加して RDP サービスにバインドします。

RDP 負荷分散を設定するには、RDP タイプのサービスを作成し、RDP 仮想サーバーにバインドします。

コマンドラインインターフェイスを使用して **RDP** 負荷分散サービスを構成するには

コマンドプロンプトで次のコマンドを入力して RDP 負荷分散セットアップを構成し、構成を確認します。

```
add service <name>@ <serverName> <serviceType> <port>
<!--NeedCopy-->
```

注: サービスをさらに追加するには、前述のコマンドを繰り返します。

例

```
> add service ser1 10.102.27.182 RDP 3389
Done
> add service ser2 10.102.27.183 RDP 3389
Done
>show service ser1
ser1 (10.102. 27.182:3389) - RDP
        State: UP
…
            Server Name: 10.102.27.182
```



```
10              Server ID : 0            Monitor Threshold : 0
11          Down state flush: ENABLED
12  …
13  1)      Monitor Name: tcp-default
14                  State: UP       Weight: 1
15  …
16                  Response Time: 4.152 millisec
17   Done
18  <!--NeedCopy-->
```

構成ユーティリティを使用して **RDP** 負荷分散サービスを構成するには

[トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、RDP タイプのサービスを作成します。

コマンドラインインターフェイスを使用して **RDP** 負荷分散仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して RDP 負荷分散仮想サーバーを構成し、構成を確認します。

```
1  add lb vserver <name>@ <serviceType> <ipAddress> <port>
2
3  bind lb vserver <name>@ <serviceName>
4
5  Bind all the RDP services to be load balanced to the virtual server.
6  <!--NeedCopy-->
```

例:

この例では、2 つの RDP サービスが RDP 仮想サーバーにバインドされています。

```
1  add lb vs v1 rdP 10.102.27.186 3389
2  Done
3
4  bind lb vs v1 ser1
5  service "ser1" bound
6
7  bind lb vs v1 ser2
8  service "ser2" bound
9  Done
10
11 sh lb vs v1
12 v1 (10.102.27.186:3389) - RDP   Type: ADDRESS
13 State: UP
14  …
15 No. of Bound Services :  2 (Total)       2 (Active)
16 Configured Method: LEASTCONNECTION
17   Current Method: Round Robin, Reason: A new service is bound
18 Mode: IP
19 Persistence: NONE
20   L2Conn: OFF
21
```



```
1) ser1 (10.102.27.182: 3389) - RDPState: UP   Weight: 1
2) ser2 (10.102.27.183: 3389) - RDPState: UP   Weight: 1
Done
<!--NeedCopy-->
```

構成ユーティリティを使用して **RDP** 負荷分散仮想サーバーを構成するには

[トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、RDP タイプの仮想サーバーを作成し、RDP サービスをこの仮想サーバーにバインドします。

コマンドラインインターフェイスを使用して **RDP** サービスのスクリプトモニターを構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add lb monitor <monitorName> USER -scriptName nsrdp.pl

bind lb monitor <monitorName> <rdpServiceName>
<!--NeedCopy-->
```

例**:**

```
add service ser1 10.102.27.182 RDP 3389

add lb monitor RDP_MON USER -scriptName nsrdp.pl

bind lb monitor RDP_MON ser1

<!--NeedCopy-->
```

構成ユーティリティを使用して **RDP** サービスのスクリプトモニターを構成するには

1. [トラフィック管理] **>** [負荷分散] **>** [モニター] に移動し、USER タイプのモニターを作成します。
2. 「特殊パラメータ」の「スクリプト名」リストで「nsrdp.pl」を選択し、このモニターを RDP サービスにバインドします。

負荷分散サービスの優先順位

October 25, 2023

サービスの優先順位機能を使用すると、負荷分散の選択プリファレンスに基づいて、サービスまたはサービスグループの順序に優先順位を付けることができます。次の操作を行うと、優先順位を設定できます。

- サービスを負荷分散仮想サーバーにバインドします。
- サービスグループを負荷分散仮想サーバにバインドします。
- サービスグループメンバーを負荷分散サービスグループにバインドします。

現在、サービスの優先順位は次の方法で構成できます。ただし、これらのアプローチには次の制限があります。

- バックアップ仮想サーバーチェーンの構成: 構成行の数が多いため、各仮想サーバーのすべての LB サービスの状態を確認するには、showコマンドを複数回実行する必要があります。
- 優先ロケーションの設定: すべてのアプリケーションエンドポイントに対してロケーションエントリを作成する必要があります。

サービスの優先順位付け機能は、設定コマンドの数を減らして前述の制限に対処し、すべての負荷分散サービスの IP アドレスをロケーションで表現しなくても、優先ロケーションの設定を行うことができます。

負荷分散サービスの優先順位を設定する

負荷分散サービスの優先順位を設定するために、-order <number>パラメータが bind コマンドに追加されます。

> 注:
>
> 順序番号が小さいほど優先度が高くなります。

コマンド:

```
bind lb vserver <vservername> <servicename/servicegroupname> -order <number>
```

たとえば、負荷分散仮想サーバー (vs1) にバインドされた一連のサービスを考えてみます。
- order <number>パラメータを使用すると、次のように、サービスの選択順序に優先順位を付けることができます。

- vs1 にバインドされたセット 1 (s1, s2) —オーダ 1
- vs1 にバインドされたセット 2 (s3, s4) —オーダー 2
- vs1 にバインドされた 3 (s5, s6) をセット—オーダ 3

サービスを vs1 にバインドし、vs1 がクライアントトラフィックを受信すると、サービスの選択順序は次のようになります。

- 仮想サーバ (vs1) は、最初にセット 1 (s1 と s2) のサービスを選択します。これは、このセットには最小の順序番号が割り当てられているためです。既定では、最小の順序番号が優先されます。
- セット 1 のすべてのサービスが DOWN の場合、vs1 は順序番号 2 のセット 2 (s3 と s4) を選択します。
- セット 1 とセット 2 のすべてのサービスがダウンしている場合、vs1 は順序番号 3 のセット 3 (s5 と s6) を選択します。

CLI を使用して負荷分散サービスの優先順位を設定する

負荷分散サービスの優先順位を構成するには、コマンドプロンプトで次のコマンドを入力します。

1. LB 仮想サーバを追加します。

   ```
   add lb vserver vs1 HTTP 1.1.1.1 80
   ```

2. LB サービスを追加します。

   ```
   add service s[1-6] 2.2.2.[1-6] HTTP 80
   ```

3. 注文番号を設定し、サービスを LB 仮想サーバーにバインドします。

   ```
   bind lb vserver vs1 s1 -order 1
   bind lb vserver vs1 s2 -order 1
   bind lb vserver vs1 s3 -order 2
   bind lb vserver vs1 s4 -order 2
   bind lb vserver vs1 s5 -order 3
   bind lb vserver vs1 s6 -order 3
   ```

GUI を使用した負荷分散サービスの優先順位の設定

前提条件:

- これで、負荷分散仮想サーバーが作成されました。
- サービスを作成しました。

負荷分散サービスの優先順位を設定し、仮想サーバにバインドするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、負荷分散仮想サーバーをダブルクリックします。
2. [負荷分散仮想サーバー] の [サービスとサービスグループ] セクションで、[負荷分散仮想サーバーサービスバインド] をクリックします。
3. [負荷分散仮想サーバーサービスバインド] ダイアログボックスで、[バインドの追加] をクリックします。
4. [サービスバインディング] ダイアログボックスで、サービスを選択します。
5. [順序] フィールドに番号を入力して、サービスの優先順位を設定します。
6. 「バインド」をクリックします。
7. ステップ 1 ～6 を繰り返して、サービスごとに異なる優先順位番号を設定します。

LB policy コマンドを使用して、負荷分散サービスの優先順位を設定します

既定では、最小の順序番号が優先されます。ただし、新しい LB action コマンドと policy コマンドを使用すると、このデフォルトの動作を延期できます。着信クライアントトラフィックまたはクライアントデータに基づいて、サービスの選択順序を設定できます。

たとえば、仮想サーバ (vs1) にバインドされた一連のサービスを考えてみます。- `order <number>`パラメータを使用して、サービスの優先順位を次のように設定しました。

- vs1 にバインドされたセット 1 (s1, s2) —オーダ 1
- vs1 にバインドされたセット 2 (s3, s4) —オーダー 2
- vs1 にバインドされた 3 (s5, s6) をセット—オーダ 3

既定では、最小の順序番号が優先されます。したがって、set 1、set2、set3 のサービスのデフォルトの優先順位は、それぞれ 1、2、3 です。ただし、特定のクライアントトラフィックについては、優先順位を 3、1、2 に変更する必要があります。これを実現するには、LB ポリシーを追加して vs1 にバインドします。

LB ポリシーコマンドは、ルールとアクションの 2 つの要素で構成されます。ルールはアクションに関連付けられ、リクエストがルールに一致した場合に実行されるアクションです。

> 注:
>
> LB ポリシーコマンドは、LB と GSLB の両方の構成に共通であり、NetScaler ADC アプライアンスによって処理される要求に適用されます。

LB アクション

** 表現:**

```
add lb action <name> <type> <string>
```

** 例:**

```
add lb action act1 -type SELECTIONORDER -value 3 2 1
```

パラメーター:

- `name`: アクションの名前。
- `type`: アクションのタイプ。
- `string`: 指定したアクションの値。

LB ポリシー

** 表現:**


```
add lb policy <name> <rule> <action> <undefaction>
```

** 例:**

```
add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1
```

パラメーター:

- name: ポリシーの名前。
- rule: ルールは 1 つ以上の式で構成されます。ルールはアクションに関連付けられ、リクエストがルールに一致した場合に実行されるアクションです。
- action: DROP、NOLBACTION、およびリセットがサポートされています。
- undefaction: NetScaler ADC アプライアンスは、要求がポリシーと一致しない場合、未定義イベント（UNDEF イベント）を生成します。
  `set lb param -undefAction <action>`コマンドを使用して、未定義のアクションを設定できます。これらのアクションは、DROP、NOLBACTION、RESET といった未定義のイベントに割り当てることができます。

次のように、LB アクション、LB ポリシーを追加し、そのポリシーを負荷分散仮想サーバー (vs1) にバインドする例を考えてみましょう。

```
add lb action act1 -type SELECTIONORDER -value 3 1 2
add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1
bind lb vserver vs1 -policyName pol1 -priority 10
```

ルールは IP アドレス、`8.8.8.8`と一致するクライアントトラフィックを選択し、そのトラフィックを vs1 に送信します。LB アクションタイプ (SELECTIONORDER) は、サービスの選択順序を定義します。LB ポリシーを vs1 にバインドした後、vs1 が IP アドレス`8.8.8.8`からクライアントトラフィックを受信すると、次の順序でサービスが選択されます。

1. 仮想サーバ (vs1) は、セット 3 (s5 と s6) のサービスを優先順位 3 で選択します。
2. セット 3 のすべてのサービスが DOWN の場合、vs1 は優先順位が 2 のセット 1 (s1 と s2) を選択します。
3. セット 3 とセット 2 のすべてのサービスがダウンしている場合、vs1 は順序 1 のセット 1（s1 と s2）を選択します。

CLI を使用して LB policy コマンドを使用して負荷分散サービスの優先順位を設定する

LB policy コマンドを使用して負荷分散サービスの優先順位を構成するには、コマンドプロンプトで次のコマンドを入力します。

1. LB アクションを追加します。

   ```
   add lb action act1 -type SELECTIONORDER -value 3 1 2
   ```


2. LB ポリシーを追加します。

```
add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1
```

3. LB 仮想サーバを追加します。

```
add lb vserver vs1 HTTP 1.1.1.1 80
```

4. LB ポリシーを LB 仮想サーバーにバインドします。

```
bind lb vs vs1 -policyName pol1 -priority 10
```

5. LB サービスを追加します。

```
add service s[1-6] 2.2.2.[1-6] HTTP 80
```

6. 順序を設定し、サービスを LB 仮想サーバーにバインドします。

```
bind lb vserver vs1 s1 -order 1
bind lb vserver vs1 s2 -order 1
bind lb vserver vs1 s3 -order 2
bind lb vserver vs1 s4 -order 2
bind lb vserver vs1 s5 -order 3
bind lb vserver vs1 s6 -order 3
```

GUI を使用して **LB policy** コマンドを使用して、負荷分散サービスの優先順位を設定します

前提条件:

- これで、負荷分散仮想サーバーが作成されました。
- サービスを作成しました。

ステップ **1-LB** アクションを作成します。

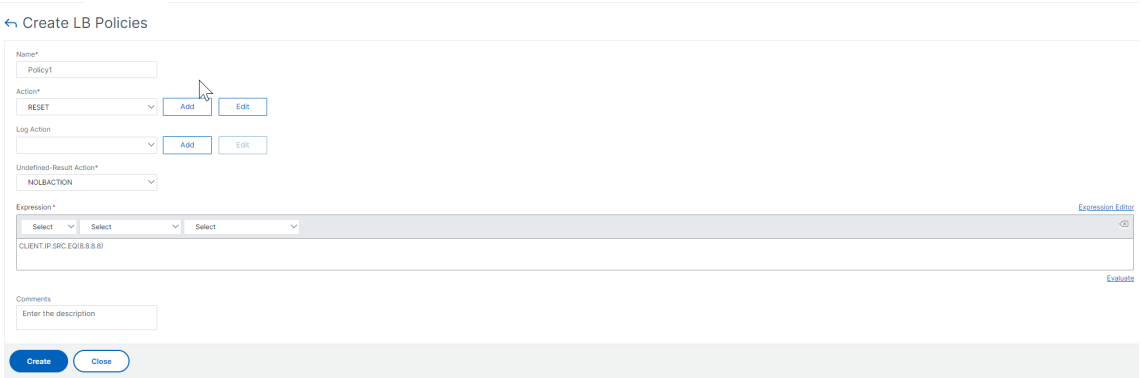
1. **AppExpert > LB >** アクションに移動します。
2. [**LB** アクション] で、[追加] をクリックします。
3. [**Create LB Actions**] ダイアログボックスで、次のパラメータの値を指定します。
    - **[**アクション名**]**
    - 種類
    - 値

> 注:
>
> **Value** フィールドの数字はスペースで区切られます。

4. [作成] をクリックします。

ステップ **2-LB** ポリシーを作成します。

1. **AppExpert > LB >** ポリシーに移動します。
2. [**LB** ポリシー] で [追加] をクリックします。
3. [**Create LB Policies**] ダイアログボックスで、次のパラメータの値を指定します。
   - 名前:pol1
   - アクション:act1
   - 未定義の結果アクション:NOLBACTION
   - 式:CLIENT.IP.SRC.EQ(8.8.8.8)



4. 「作成」をクリックします。

ステップ **3-LB** ポリシーを **LB** 仮想サーバーにバインドします。

1. [トラフィック管理**] > [LB] > [**仮想サーバー] に移動し、仮想サーバーをダブルクリックします。
2. [詳細設定**]** で [ポリシー] をクリックします。
3. [**Policies**] セクションで、プラス (+) アイコンをクリックします。
4. 「タイプを選択」 (Choose Type) ダイアログボックスで、次のパラメータの値を指定します。
   - ポリシーを選択:LB
   - タイプを選択: リクエスト
5. [バインドを追加**]** をクリックします。
6. [ポリシーバインド] ダイアログボックスで、次のパラメータの値を指定します。
   - ポリシーを選択

- 優先度
- **Goto** 式
- ラベルタイプを呼び出す

7. [**Bind**] をクリックします。

ステップ **4-**負荷分散サービスの優先順位を設定します。

負荷分散サービスの優先順位を設定するには、「**GUI** を使用した負荷分散サービスの優先順位の設定」の手順を参照してください。

サービスのパーシスタンス設定

サービスに永続性が設定されている場合、デフォルトでは常に永続性が優先されます。

たとえば、パーシステンスが設定され、優先順位が 1 のサービスを考えてみます。優先順位が 0 のサービスが UP の場合、優先順位が 1 のサービスが常に優先されます。

ただし、次の CLI コマンドを使用すると、このデフォルトの動作を上書きできます。

```
set lb param -overridePersistencyforOrder <YES/NO>
```

次の例を考えてみましょう。

一連のサービスは、次の優先順位で仮想サーバ (vs1) にバインドされます。

- vs1 にバインドされたセット 1 (s1, s2) —オーダ 1
- vs1 にバインドされたセット 2 (s3, s4) —オーダー 2

永続性を上書きするには、コマンドプロンプトで次のコマンドを入力します。

```
set lb parameter -overridePersistencyforOrder YES
```

set 1 (永続性を持つサービスが構成されている) が DOWN の場合、set 2 のサービスが set 1 のサービスが UP になるまで、すべての要求を処理します。優先度 2 の持続性エントリが作成されます。

しばらくすると、set 1 のサービスが起動したと仮定します。これで、set 1 と set 2 の両方のサービスがリクエストを処理するために UP になりました。このシナリオでは、高次のサービスが稼働すると、新しい負荷分散の決定が下されます。パーシステンスエントリは、新しい負荷分散エントリで上書きされます。

優先度切り替え

優先度切り替え機能を使用すると、優先順位の高いサービスのバージョンアップグレード中に、すべてのトラフィックを優先度の低いサービスに切り替えることができます。以下のコマンドを使用して、優先度を切り替えることができます。

- `set lb vserver -toggleorder<Ascending/Descending>`

- `set lb vserver v1 -orderthreshold 80`

たとえば、次の優先順位を持つ 2 つのサービスがあるとします。

- Service 1- order 0
- サービス 2 —注文 1

デフォルトでは、サービス 1 がすべてのトラフィックを処理します。サービス 1 をアップグレードする必要がある場合は、トラフィックをサービス 2 に再ルーティングする必要があります。

コマンドプロンプトで次のコマンドを入力して、優先度を切り替えます。

```
set lb vserver -toggleorder Descending
```

既定では、0 の方が優先度が高くなります。ただし、優先度を切り替えた後は、1 の方が優先度が高いと見なされます。サービスにパーシステンスエントリが存在する場合、パーシステンスプリファレンスの動作は、「サービスのパーシスタンス設定」セクションで説明されているとおりになります。

使用例 **1**：**SMPP** 負荷分散

August 15, 2023

Short Message Peer to Peer（SMPP）プロトコルを使用して、個人と銀行、広告主、ディレクトリサービスなどの付加価値サービスプロバイダーとの間で毎日何百万ものショートメッセージが交換されています。多くの場合、サーバーが過負荷になり、トラフィックがサーバー間で最適に分散されないため、メッセージの配信が遅れます。NetScaler は SMPP 負荷分散をサポートし、サーバー全体にメッセージを最適に分散することで、パフォーマンスの低下や停止を防ぎます。

NetScaler は、クライアントからメッセージを受信したときはサーバー側で、サーバーからメッセージを受信したときはクライアント側で負荷分散を実行します。

NetScaler による SMPP メッセージの負荷分散には次の利点があります。

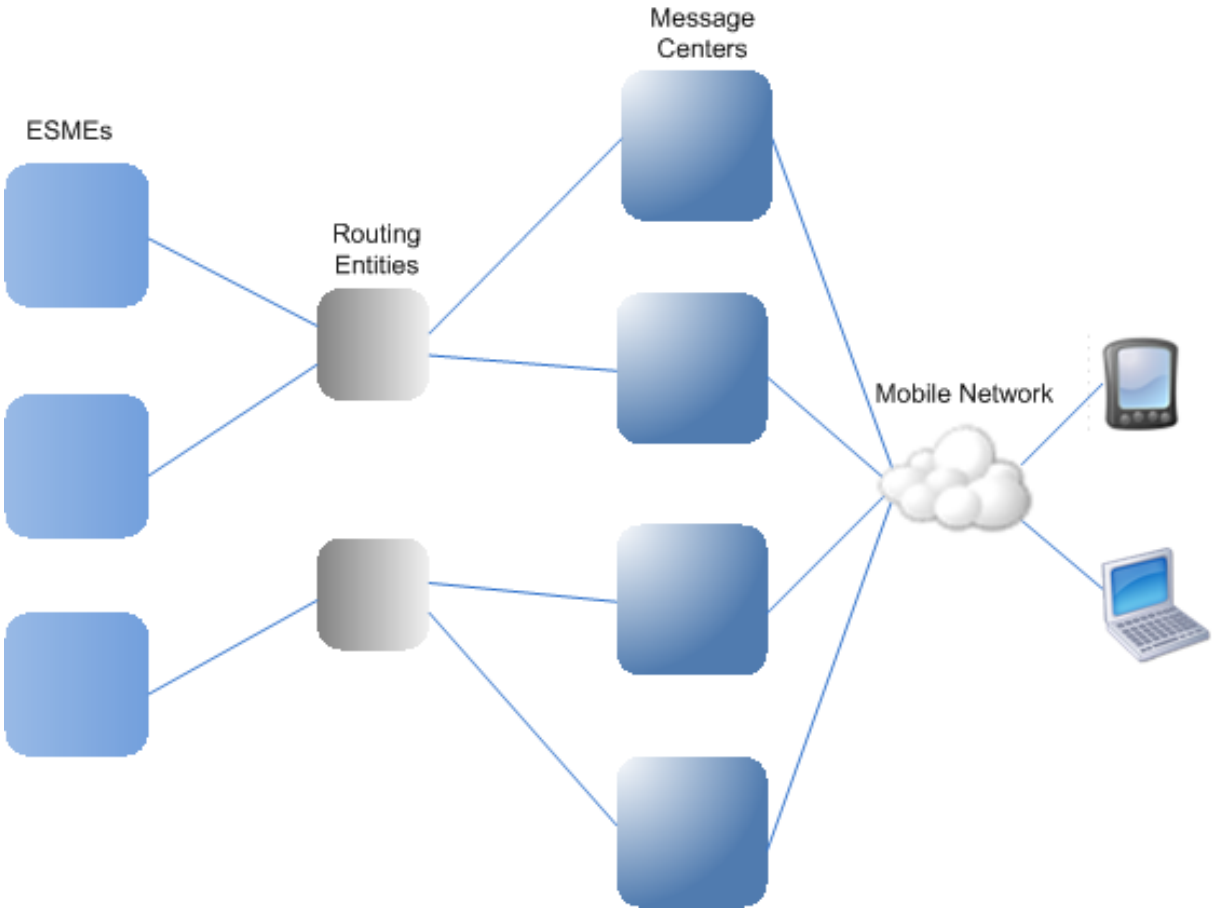
- サーバーの負荷分散が改善され、エンドユーザーへの応答時間が短縮されます
- サーバーのヘルスモニタリングとフェイルオーバー機能の向上
- クライアント構成を変更せずに、新しいサーバー (メッセージセンター) をすばやく簡単に追加できます
- 高可用性

SMPP の概要

SMPP は、長寿命の TCP 接続を介した外部ショートメッセージエンティティ（ESME）、ルーティングエンティティ（RE）、およびメッセージセンター（MC）間でショートメッセージを転送するためのアプリケーション層プロトコルです。友人、連絡先、銀行（モバイルバンキング）、広告主（モバイルコマース）、ディレクトリサービスなどの第

三者との間でショートメッセージサービス（SMS）メッセージを送信するために使用されます。ESME（非モバイルエンティティ）からのメッセージは MC に到着し、MC はそれらを携帯電話などのショートメッセージエンティティ（SME）に配信します。SMPP は、中小企業が第三者にショートメッセージを送信するためにも使用されます（たとえば、製品の購入、請求書の支払い、送金など）。これらのメッセージは MC に到着し、宛先 MC または ESME に転送されます。

次の図は、モバイルネットワーク内の ESME、RES、および MC のさまざまな SMPP エンティティを示しています。



モバイルネットワークにおけるさまざまな **SMPP** エンティティのアーキテクチャの概要

注:「クライアント」と「ESME」という用語は、ドキュメント全体で同じ意味で使用されています。

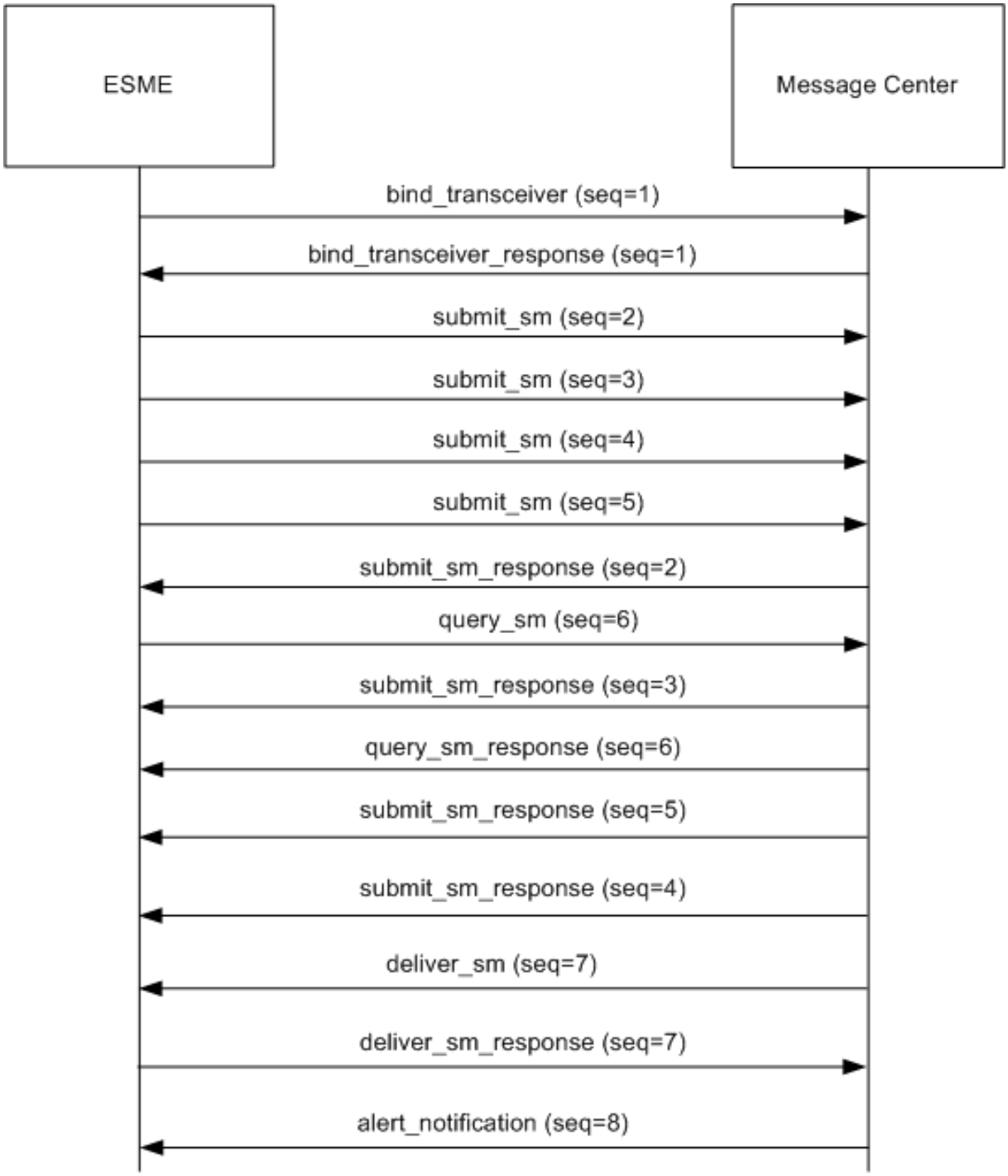
ESME (クライアント) は、送信機、受信機、またはトランシーバの 3 つのモードのいずれかで MC への接続を開きます。送信機としては、配信用のメッセージのみを送信できます。受信側としては、メッセージのみを受信できます。トランシーバーとして、ESME はメッセージを送受信できます。ESME は MC に、バインド送信機、バインドレシーバー、またはバインドトランシーバーの 3 つのメッセージ (PDU とも呼ばれます) のうちの 1 つを MC に送信します。MC は、要求に応じて bind_transmitter_resp、bind_receiver_resp、または bind_transceiver_resp で応答します。

接続が確立されると、ESME は MC にバインドされているモードに応じて、submit_sm または data_sm メッセージを送信したり、deliver_sm または data_sm メッセージを受信したり、これらのタイプのメッセージのいずれかを送受信したりできます。ESME は、query_sm、replace_sm、cancel_sm などの補助メッセージを送信して、以前のメッセージ配信のステータスを照会したり、以前のメッセージを新しいメッセージに置き換えたり、未配信のメッセージをキャンセルしたりすることもできます。

ESME が利用できない、またはモバイルユーザがオンラインではないためにメッセージが配信されない場合、メッセージはキューに入れられます。その後、MC はモバイル加入者が接続可能になったことを検出すると、受信側またはトランシーバセッション経由で alert_notification PDU を ESME に送信し、キューに入っているメッセージの配信を要求します。

各リクエスト PDU には固有のシーケンス番号があります。レスポンス PDU のシーケンス番号は元のリクエストと同じです。SMPP を介したメッセージ交換は非同期モードで行うことができるため、ESME または MC は一度に複数の要求を送信できます。シーケンス番号は、同じ SMPP セッションで応答を返す上で重要な役割を果たします。つまり、シーケンス番号によってリクエストとレスポンスのマッチングが可能になります。

次の図は、ESME がトランシーバとしてバインドされるときに、トラフィックフローがさまざまな PDU を使用する方法を示しています。



制限事項:

NetScaler ADC アプライアンスはアウトバウンド操作をサポートしていません。つまり、メッセージセンターは、NetScaler ADC アプライアンスを介して ESME との SMPP セッションを開始できません。

NetScaler での **SMPP** 負荷分散の仕組み

ESME（クライアント）は、NetScaler への接続を開くためのバインドメッセージを送信します。ADC は各 ESME を認証し、認証に成功すると、適切なメッセージで応答します。NetScaler は各メッセージセンターとの接続を確立し、これらのメッセージセンター間ですべてのメッセージの負荷分散を行います。ADC はクライアントからメッセージを受信すると、開いている接続をメッセージセンターに再利用するか、開いている接続が利用できない場合はメッセージセンターにバインド要求を送信します。

ADC は、クライアントとサーバーから発信されるメッセージのロードバランシングを行うことができます。メッセージセンターの状態を監視し、連結されたメッセージを処理できます。また、メッセージセンターのコンテンツスイッチングもサポートしています。

ESME から送信されるメッセージ

認証を行うには、各 ESME を NetScaler にユーザーとして追加する必要があります。クライアントは、バインド要求を送信することにより、ADC に設定された SMPP 仮想サーバーとの TCP 接続を確立します。ADC はクライアントを認証し、成功するとバインドメッセージを解析します。次に、ADC は、設定された負荷分散方式で選択されたメッセージセンターに要求を送信します。メッセージセンターへの接続を再利用できない場合、ADC はメッセージセンターに新しいバインド要求を送信して、メッセージセンターとの TCP 接続を開きます。

メッセージ・センターからクライアントに応答（submit_sm_resp または data_sm_resp）を転送する前に、ADC はメッセージ ID にカスタム・サーバー ID を追加して、クライアントによるメッセージのクエリ、置換、キャンセルなどの補助操作のためにメッセージ・センターを識別します。他のクライアントからのリクエストも同様に負荷分散されます。

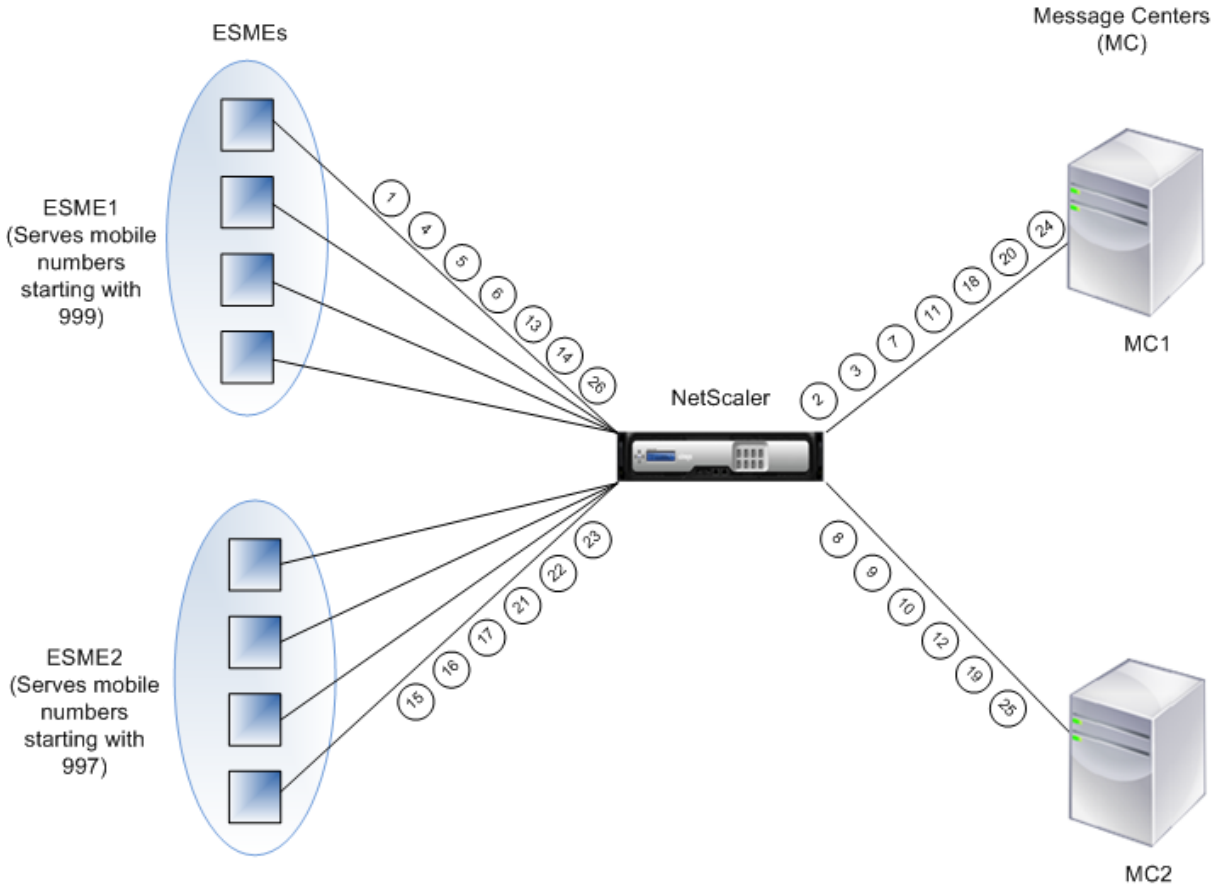
元のバインドリクエストでは、クライアントは処理できるアドレス範囲を指定します。この範囲は、メッセージセンターからクライアントに deliver_sm または data_sm メッセージを転送するために使用されます。

メッセージセンターから送信されたメッセージ

特定のアドレス範囲を処理できる ESME は、クラスターにグループ化されます。クラスター内のすべてのノードが同じ認証情報を提供します。クラスター内では、ロードバランシングにはラウンドロビン方式のみが使用されます。モバイル発信（MO）メッセージを配信するために、メッセージセンターは deliver_sm メッセージを NetScaler に送信します。宛先アドレス範囲 (たとえば、998 で始まる番号) に対応できるクラスターが ADC にバインドされている場合、ADC はそのクラスターを選択し、そのクラスター内の ESME ノード間でメッセージの負荷分散を行います。

そのアドレス範囲の deliver_sm メッセージを提供できる ESME が ADC にバインドされておらず、メッセージキューイングが有効になっている場合、メッセージはそのようなクライアントがレシーバーまたはトランシーバモードで ADC にバインドされるまでキューに入れられます。キューのサイズを指定できます。

次の図は、ESME、NetScaler、およびメッセージセンター間の PDU の内部フローを示しています。簡単にするために、2 つの ESME と 2 つのメッセージセンターのみが表示されます。



メッセージ（PDU）のフロー：

1. ESME1 が NetScaler にバインドリクエストを送信
2. NetScaler が MC1 にバインドリクエストを送信
3. MC1 がバインドレスポンスを NetScaler に送信
4. NetScaler が ESME1 にバインドレスポンスを送信
5. ESME1 が submit_sm (1) を NetScaler に送信する
6. ESME1 が submit_sm (2) を NetScaler に送信する
7. NetScaler submit_sm (1) を MC1 に転送します
8. NetScaler が MC2 にバインドリクエストを送信
9. MC2 がバインドレスポンスを NetScaler に送信
10. NetScaler submit_sm (2) を MC2 に転送します
11. MC1 が submit_sm_resp (1) を NetScaler に送信する
12. MC2 が submit_sm_resp (2) を NetScaler に送信する
13. NetScaler は submit_sm_resp (1) を ESME1 に転送します
14. NetScaler は submit_sm_resp (2) を ESME1 に転送します
15. ESME2 が NetScaler にバインドリクエストを送信
16. NetScaler が ESME2 にバインドレスポンスを送信
17. ESME2 が submit_sm (3) を NetScaler に送信する

18. NetScaler submit_sm (3) を MC1 に転送します
19. MC2 は deliver_sm を NetScaler に送信します（ESME2 はメッセージで指定されたアドレス範囲を処理します）
20. MC1 が submit_sm_resp (3) を NetScaler に送信する
21. NetScaler は submit_sm_resp (3) を ESME2 に転送します
22. NetScaler deliver_sm を ESME2 に転送します
23. ESME2 が deliver_sm_resp を NetScaler に送信
24. MC1 はアラート通知を NetScaler に送信します（ESME1 はメッセージで指定されたアドレス範囲を処理します）
25. NetScaler deliver_sm_resp を MC2 に転送します
26. NetScaler はアラート通知を ESME1 に転送します

メッセージセンターのヘルスモニタリング

デフォルトでは、TCP_DEFAULT モニターは SMPP サービスにバインドされますが、SMPP タイプのカスタムモニターをバインドすることもできます。カスタムモニターはメッセージセンターへの TCP 接続を開き、enquire_link パケットを送信します。プローブの成功または失敗に応じて、サービスは UP または DOWN とマークされます。

メッセージセンターのコンテンツスイッチング

メッセージセンターは、ESME からの複数の接続（またはバインド要求）を受け付けることができます。SMPP バインドパラメータに基づいて、これらの要求をコンテンツスイッチするように NetScaler ADC を構成できます。メッセージセンターを選択するためのメソッドを設定するための一般的な式を次に示します。

- アドレス範囲に基づく: 次のサンプル式では、アドレス範囲が 988 から始まる場合、ADC は特定のメッセージセンターを選択します。

例**:**

SMPP.BINDINFO.ADDRESS_RANGE. CONTAINS (「^988」)

- ESME ID に基づいて: 次のサンプル式では、ESME ID が ESME1 と等しい場合、ADC は特定のメッセージセンターを選択します。

例**:**

SMPP.BINDINFO.SYSTEM_ID.EQ(“ESME1”)

- ESME タイプに基づく: 次のサンプル式では、ESME タイプが VMS の場合、ADC は特定のメッセージセンターを選択します。VMS はボイスメールシステムの略です。

例**:**

SMPP.BINDINFO.SYSTEM_TYPE.EQ(“VMS”)

- ESME の数値のタイプ（TON）に基づいて、次の式例では、TON が 1 に等しい場合（1 は国際番号）、ADC は特定のメッセージセンターを選択します。

例**:**

SMPP.BINDINFO.ADDR_TON.EQ(1)

- ESME のナンバー・プラン・インジケータ（NPI）に基づいて、次の式では、NPI が 0（0 は未知の接続）の場合、ADC は特定のメッセージ・センターを選択します。

例**:**

SMPP.BINDINFO.ADDR_NPI.EQ(0)

- バインドタイプに基づく。次のサンプル式では、バインドタイプが TRANSCEIVER の場合、ADC は特定のメッセージセンターを選択します。（トランシーバはメッセージを送受信できます。）

例**:**

SMPP.BINDINFO.TYPE.EQ(TRANSCEIVER)

連結メッセージ処理

SMS には最大 140 バイトまで保存できます。長いメッセージは小さな部分に分割する必要があります。宛先の携帯電話が対応している場合、メッセージは結合され、1 つの長い SMS として配信されます。NetScaler はメッセージの断片を同じメッセージセンターに転送します。各メッセージには、参照番号、シーケンス番号、およびフラグメントの総数が含まれます。参照番号は、長いメッセージの各フラグメントで同じです。シーケンス番号は、メッセージ全体における特定のフラグメントの位置を指定します。すべてのフラグメントが受信されると、ESME はフラグメントを 1 つの長いメッセージにまとめ、そのメッセージをモバイルユーザに配信します。

クライアントがアクティブな接続を切断しても、メッセージセンターへの接続は閉じられません。他のクライアントからのリクエストに再利用されます。

制限事項

メッセージセンターからの 59 バイトを超えるメッセージ ID はサポートされていません。メッセージセンターから返されたメッセージ ID の長さが 59 バイトを超える場合、補助操作は失敗し、NetScaler はエラーメッセージで応答します。

NetScaler での **SMPP** 負荷分散の構成

ADC で SMPP ロードバランシングを設定するには、次のタスクを実行します。

1. SMPP ユーザーを追加します。ADC は、ユーザーからのバインド要求を受け入れる前にユーザーを認証します。ユーザーは通常 ESME です。

2. プロトコルを SMPP として指定して、負荷分散仮想サーバーを追加します。
3. プロトコルを SMPP として指定し、各サーバに固有のカスタムサーバ ID を指定してサービスを追加します。サービスを以前に作成した負荷分散仮想サーバーにバインドします。
4. オプションで、サービスグループを作成し、そのサービスグループにサービスを追加します。
5. オプションで、SMPP-ECV タイプのモニターを追加してサービスにバインドします。TCP-Default モニターはデフォルトでバインドされます。
6. クライアントモードやメッセージキューなどの SMPP パラメータを設定します。

コマンドラインを使用して **SMPP** ロードバランシングを設定するには

コマンドプロンプトで入力します。

```
add smpp user <username> -password <password>
add service <name> <IP> SMPP <port>  - customserverID <customserverID>
add lb vserver <name> <IP> SMPP <port>
bind lb vserver <name> <service name>
set smpp param
<!--NeedCopy-->
```

例

```
add smpp user smppclient1 -password c03ebb540695b6110eb31172f32245a1 -
    encrypted -encryptmethod ENCMTHD_2
add smpp user smppclient2 -password c03ebb540695b6110eb31172f32245a1 -
    encrypted -encryptmethod ENCMTHD_2
add service smmpsvc 10.102.84.140 SMPP 2775 -gslb NONE -maxClient 0 -
    maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
     180 -svrTimeout 360 -CustomServerID ab -CKA NO -TCPB NO -CMP NO
add service smmpsvc2 10.102.81.175 SMPP 2775 -gslb NONE -maxClient 0 -
    maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
     180 -svrTimeout 360 -CustomServerID xy -CKA NO -TCPB NO -CMP NO
add lb vserver smppvs SMPP 10.102.239.179 2775 -persistenceType NONE -
    cltTimeout 180
bind lb vserver smppvs smmpsvc2
bind lb vserver smppvs smmpsvc
set smpp param -addrrange "d*"
<!--NeedCopy-->
```

構成ユーティリティを使用して **SMPP** 負荷分散を設定するには

1. [システム] **>** [ユーザ管理] **>** [**SMPP** ユーザ] に移動し、SMPP ユーザを追加します。
2. [トラフィック管理] **>** [負荷分散] **>** [**SMPP** パラメータの設定] に移動し、展開で必要に応じてパラメータを設定します。
3. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、タイプ SMPP の仮想サーバーを追加します。
4. サービスセクションをクリックし、SMPP タイプのサービスを追加し、サーバー ID を指定します。

ユースケース 2：TCP バイトストリームの名前と値のペアに基づいて規則に基づくパーシステンスを構成する

August 15, 2023

プロトコルによっては、名前と値のペアを TCP バイトストリームで送信します。この例の TCP バイトストリームのプロトコルは、財務情報交換 (FIX) プロトコルです。XML 以外の実装では、FIX プロトコルにより、ネットワークを介して通信する 2 つのホストが、名前と値のペア（「FIX フィールド」と呼ばれる）のリストとしてビジネスまたは取引関連の情報を交換できます。フィールドの形式は<tag>=<value><delimiter>です。この従来のタグ値形式により、FIX プロトコルはユースケースに最適です。

FIX フィールドのタグは、フィールドの意味を示す数値識別子です。この例では、

- タグ 35 はメッセージタイプを示します。
- 等号の後の値は、指定されたタグの特定の意味を保持し、データ型に関連付けられています。タグ 35 の値が A の場合は、メッセージがログオンメッセージであることを示します。
- 区切り文字は、印刷されない「ヘッダーの開始」(SOH) ASCII 文字 (0x01) で、キャレット記号 (^) です。
- 各フィールドには名前も割り当てられます。タグ 35 のフィールドは MsgType フィールドです。

ログオンメッセージの例を次に示します。

8=FIX.4.1 9=61 35=A 49=INVMGR 56=BRKR 34=1 52= 20000426-12:05:06 98=0 108=30 10=157

上記のようなタグ値リストのパーシスタンスタイプの選択は、リストから特定の文字列を抽出するために利用できるオプションによって決まります。トークンベースの永続性メソッドでは、ペイロードから抽出するトークンのオフセットと長さを指定する必要があります。FIX プロトコルでは、特定のフィールドのオフセットとその値の長さがメッセージによって異なる可能性があるため、これを行うことはできません。この変化は、メッセージタイプ、前のフィールド、および前の値の長さによって異なります。また、カスタムフィールドが定義されているかどうかによって、実装によっても異なります。このようなバリエーションにより、特定のフィールドの正確なオフセットを予測したり、トークンとして抽出される値の長さを指定したりすることは不可能になります。この場合、ルールベースの永続性が優先されるパーシステンスタイプです。

仮想サーバー fixlb1 が、Fix 対応アプリケーションのインスタンスをホストしているサーバーのファームへの TCP 接続の負荷分散を行うと仮定します。メッセージを送信する会社を識別する SenderCompid フィールドの値に基づいて、接続の永続性を構成します。この FIX フィールドのタグは 49 です (前述のログオンメッセージの例に示されています)。

負荷分散仮想サーバーのルールベースの永続性を構成するには、負荷分散仮想サーバーの永続性タイプを RULE に設定し、式を使用してルールパラメーターを構成します。式は、TCP ペイロードの senderCompid フィールドが見つかるはずの部分を抽出し、結果の文字列を区切り文字に基づいて名前と値のリストに型キャストし、SenderCompid フィールドの値 (タグ 49) を次のように抽出する式でなければなりません。

```
set lb vserver fixlb1 -persistenceType RULE -rule "CLIENT.TCP.PAYLOAD
(300).TYPECAST_NVLIST_T('=','^').VALUE("\49\")"
```


注: これは CLI コマンドなので、式にはバックスラッシュ文字が使用されています。設定ユーティリティを使用している場合は、バックスラッシュ文字を入力しないでください。

クライアントが前述のログオンメッセージ例の名前と値のリストを含む FIX メッセージを送信すると、式によって INVMGR の値が抽出され、NetScaler アプライアンスはこの値に基づいて永続セッションを作成します。

PAYLOAD () 関数の引数は、関数によって抽出された文字列に SenderCompID フィールドを含めるために必要な大きさにすることができます。オプションで、フィールドの値を抽出するときにアプライアンスで大文字と小文字を区別する場合は、SET_TEXT_MODE（IGNORECASE）関数を使用し、抽出された値のハッシュに基づいて永続セッションを作成するために HASH 関数を使用できます。次の式では、SET_TEXT_MODE (IGNORECASE) 関数と HASH 関数を使用します。

```
CLIENT.TCP.PAYLOAD(500).TYPECAST_NVLIST_T('=','^').SET_TEXT_MODE(
IGNORECASE).VALUE("49").HASH
```

FIX 接続のパーシステンスを設定するために使用できるルールの他の例を以下に示します (<tag> 値を抽出したいフィールドのタグに置き換えてください)。

- TCP ペイロードの最初の 300 バイトにある任意の FIX フィールドの値を抽出するには、CLIENT.TCP.PAYLOAD(300).BEFORE_STR(“^”).AFTER_STR(“<tag>=”). という式を使用できます。
- オフセット 80 で 20 バイトの長さの文字列を抽出するには、その文字列を名前と値のリストにキャストしてから、必要なフィールドの値を抽出します。CLIENT.TCP.PAYLOAD(100).SUBSTR(80,20).TYPECAST_NVLIST_T(‘=’ ,’ ^’).VALUE(“<tag> “) という式を使用します。
- TCP ペイロードの最初の 100 バイトを抽出し、その文字列を名前と値のリストにキャストし、必要なフィールドが 3 番目に出現する位置の値を抽出するには、CLIENT.TCP.PAYLOAD(100).TYPECAST_NVLIST_T(‘=’ ,’ ^’).VALUE(“<tag> “,2) という式を使用します。
  注:
  VALUE () 関数に渡される 2 番目の引数が
  n の場合、カウントはゼロ
  (
  0) から始まるため、アプライアンスはフィールドの (n+1)
  th インスタンスの値を抽出します。

パーシスタンスの設定に使用できるルールの他の例を以下に示します。FIX プロトコルを介して送信されるデータを評価できるのは、ペイロードベースの式だけです。他の表現は、下位ネットワークプロトコルに基づいてパーシステンスを設定するための、より一般的な表現です。

- CLIENT.TCP.PAYLOAD(100)
- CLIENT.TCP.PAYLOAD(100).HASH
- CLIENT.TCP.PAYLOAD(100).SUBSTR(5,10)
- CLIENT.TCP.SRCPORT
- CLIENT.TCP.DSTPORT
- CLIENT.IP.SRC

- CLIENT.IP.DST
- CLIENT.IP.SRC.GET4
- CLIENT.IP.DST.GET4
- CLIENT.ETHER.SRCMAC.GET6
- CLIENT.ETHER.DSTMAC.GET5
- CLIENT.VLAN.ID

ユースケース **3**：負荷分散を直接サーバー応答モードで構成する

August 15, 2023

Direct Server Return（DSR）モードの負荷分散により、サーバーは NetScaler アプライアンスを経由しないリターンパスを使用してクライアントに直接応答できます。ただし、DSR モードでは、アプライアンスはサービスのヘルスチェックを引き続き実行できます。データ量が多い環境では、DSR モードでサーバートラフィックをクライアントに直接送信すると、パケットがアプライアンスを通過しないため、アプライアンスの全体的なパケット処理能力が向上します。

DSR モードには次の機能と制限があります。

- ワンアームモードとインラインモードをサポートしています。
- アプライアンスは、アイドルタイムアウトに基づいてセッションを期限切れにします。
- アプライアンスは TCP 接続をプロキシしない（つまり、クライアントに SYN-ACK を送信しない）ため、SYN 攻撃は排除されません。SYN パケットレートフィルターを使用すると、サーバーに対する SYN のレートを制御できます。SYN のレートを制御するには、SYN のレートのしきい値を設定します。SYN 攻撃から保護するには、TCP 接続をプロキシするようにアプライアンスを設定する必要があります。ただし、そのためには、逆方向トラフィックがアプライアンスを通過する必要があります。
- DSR 構成では、NetScaler アプライアンスは負荷分散仮想サーバーの IP アドレスを宛先サーバーの IP アドレスに置き換えません。代わりに、サーバーの MAC アドレスを使用してパケットをサービスに転送します。VIP をサーバーに構成し、サーバーに構成されている VIP に対して ARP を無効にする必要があります。これにより、アプライアンスがワンアームモードで構成されている場合、クライアント要求がアプライアンスをバイパスしなくなります。たとえば、ユーザーはループバックインターフェイスで VIP を構成し、この VIP に対して ARP を無効にする必要があります。
- アプライアンスは、サービスにバインドされたモニターからサーバーの MAC アドレスを取得します。ただし、NetScaler アプライアンスに保存されているスクリプトを使用するカスタムユーザーモニター（USER タイプのモニター）では、サーバーの MAC アドレスは学習されません。DSR 構成でカスタムモニターのみを使用する場合、仮想サーバーが受信する要求ごとに、アプライアンスは（ARP 要求を送信して）宛先 IP アドレスを MAC アドレスに解決しようとします。宛先 IP アドレスは NetScaler アプライアンスが所有する仮想 IP アドレスであるため、ARP 要求は常に NetScaler インターフェイスの MAC アドレスに解決されます。したがって、仮想サーバーが受信したすべてのトラフィックは、アプライアンスにループバックされます。DSR 構

成でユーザーモニターを使用する場合は、サービス用に別のタイプのモニター（PING モニターなど）も設定する必要があります。理想的には、プローブの間隔を長くして、サーバーの MAC アドレスを学習できるようにします。

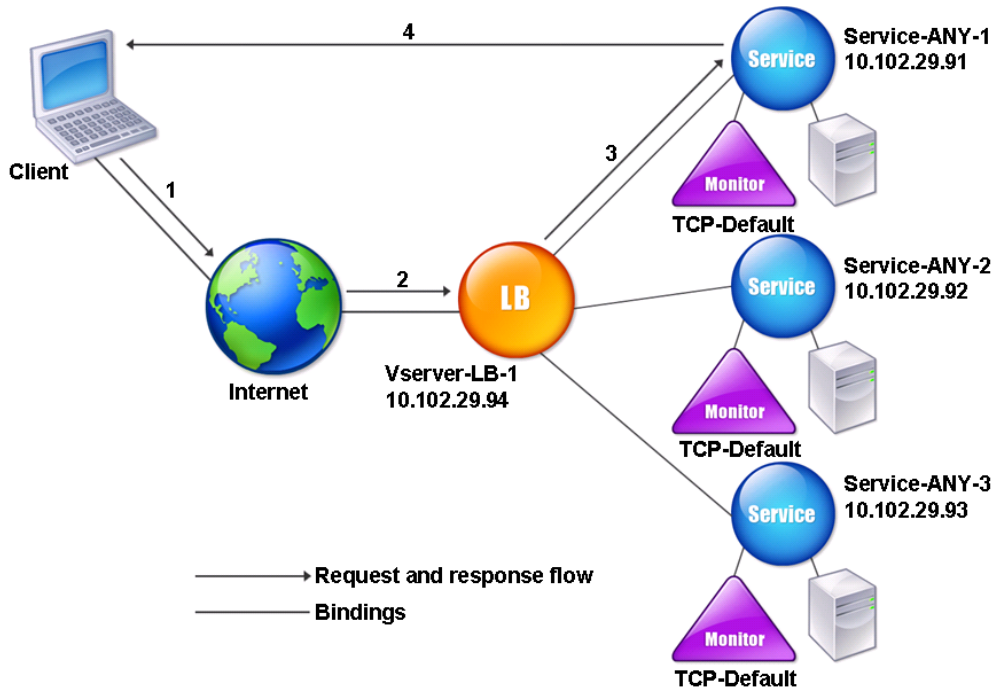
- NetScaler アプライアンスは、サービスにバインドされたモニターからサーバー L2 パラメーターを学習します。UDP-ECV モニターの場合は、受信文字列を設定して、アプライアンスがサーバーの L2 パラメーターを学習できるようにします。受信文字列が設定されておらず、サーバーが応答しない場合、アプライアンスは L2 パラメーターを学習しませんが、サービスは UP に設定されます。このサービスのトラフィックはブラックホール化されています。

この例のシナリオでは、Service-ANY-1、Service-ANY-2、Service-ANY-3 が作成され、仮想サーバ Vserver-LB-1 にバインドされます。仮想サーバーはサービスへのクライアント要求の負荷分散を行い、サービスは NetScaler アプライアンスをバイパスしてクライアントに直接応答します。次の表は、DSR モードの NetScaler アプライアンスで構成されているエンティティの名前と値を示しています。

エンティティタイプ	名前	IP アドレス	プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.94	ANY
Services	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY
	Service-ANY-3	10.102.29.93	ANY
モニター	TCP	なし	なし

次の図は、アプライアンスに設定する負荷分散エンティティとパラメーターの値を示しています。

図 1：DSR モデルでの負荷分散のためのエンティティモデル



アプライアンスが DSR モードで正常に機能するためには、クライアント要求の宛先 IP を変更しないでください。代わりに、アプライアンスは宛先 MAC を選択したサーバーの MAC に変更します。この設定により、サーバーは、サーバーをバイパスしながらクライアントに要求を転送するためのクライアント MAC アドレスを決定できます。

次に、「基本的な負荷分散の設定」の説明に従って、基本的な負荷分散設定を構成し、前の表で説明した値を使用してエンティティに名前を付けて、パラメーターを設定します。

基本的な負荷分散セットアップを構成したら、DSR モード用にカスタマイズする必要があります。これを行うには、セッションレス仮想サーバーでのソース IP ハッシュ法など、サポートされている負荷分散方法を構成します。また、リダイレクトモードを設定して、サーバーが応答を転送するためのクライアント MAC アドレスを決定し、アプライアンスをバイパスできるようにする必要があります。

負荷分散方法とリダイレクトモードを設定したら、各サービスで USIP モードを有効にする必要があります。その後、サービスは応答を転送するときに送信元 IP アドレスを使用します。

コマンドラインインターフェイスを使用してセッションレス仮想サーバーの負荷分散方法とリダイレクトモードを構成するには

コマンドプロンプトで入力します。


```
set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <
    RedirectionMode> -sessionless <Value>
<!--NeedCopy-->
```

例

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPHash -m MAC -sessionless
    enabled
<!--NeedCopy-->
```

> 注
>
> -m MAC オプションが有効になっている仮想サーバーにバインドされているサービスの場合は、非ユーザーモニターをバインドする必要があります。

構成ユーティリティを使用してセッションレス仮想サーバーの負荷分散方法とリダイレクトモードを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、[リダイレクトモード] を [MAC ベース]、[方法] を [SOURCEIPHASH] として選択します。
3. [トラフィック設定] で、[セッションレス負荷分散] を選択します。

コマンドラインインターフェイスを使用して送信元 **IP** アドレスを使用するようにサービスを構成するには

コマンドプロンプトで入力します。

```
set service <ServiceName> -usip <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-ANY-1 -usip yes
<!--NeedCopy-->
```

構成ユーティリティを使用して送信元 **IP** アドレスを使用するようにサービスを構成するには

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
2. サービスを開き、[トラフィック設定] で、**[**送信元 **IP** アドレスの使用**]** を選択します。

特定の状況では、後続のセクションで説明するように、いくつかの追加手順が必要です。

ユースケース 4：LINUX サーバーを DSR モードで構成する

August 15, 2023

Linux オペレーティングシステムでは、DSR クラスター内の各負荷分散サーバーに、NetScaler アプライアンスの仮想 IP アドレス（VIP）を使用してループバックインターフェイスを設定する必要があります。

Linux サーバーを DSR モードに設定するには

負荷分散された各サーバー上の NetScaler ADC アプライアンスの VIP とのループバックインターフェイスを作成するには、Linux OS プロンプトで次のコマンドを入力します。

```
ifconfig dummy0 up

ifconfig dummy0:0 inet <netscaler vip> netmask 255.255.255.255 up

echo 1 > /proc/sys/net/ipv4/conf/dummy0/arp_ignore

echo 2 > /proc/sys/net/ipv4/conf/dummy0/arp_announce
<!--NeedCopy-->
```

次に、TOS ID を VIP に再マップするソフトウェアを実行します。

注: ソフトウェアを実行する前に、正しいマッピングをソフトウェアに追加してください。前述のコマンドでは、Linux サーバーは dummy0 を使用してネットワークに接続します。このコマンドを使用するときは、Linux サーバーがネットワークに接続するために使用するインターフェースの名前を入力します。

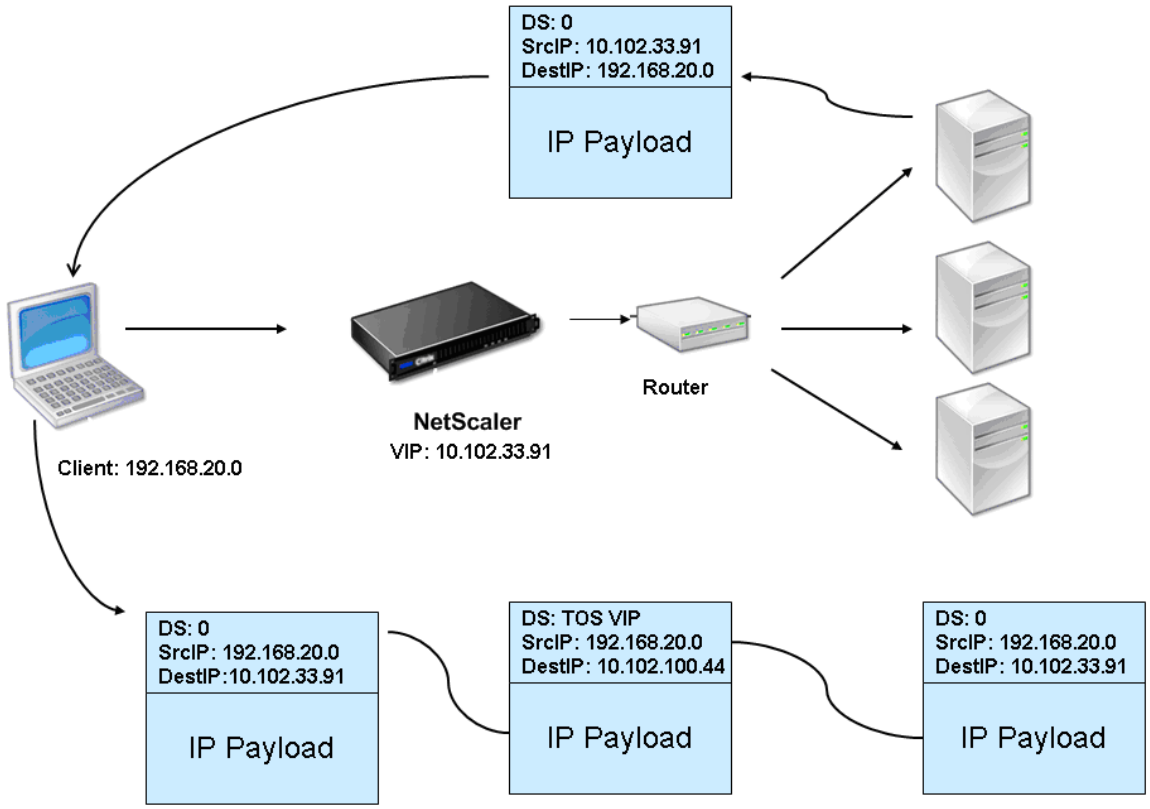
ユースケース 5：TOS の使用時に DSR モードを構成する

August 15, 2023

TOS（タイプオブサービス）とも呼ばれるディファレンシエーテッドサービス（DS）は、IPv4 パケットヘッダーの一部であるフィールドです。IPv6 ヘッダーの対応するフィールドは「トラフィッククラス」です。上位層プロトコルでは TOS を使用してパケットのパスを最適化します。TOS 情報は NetScaler アプライアンスの仮想 IP アドレス（VIP）をエンコードし、負荷分散サーバーはそこから VIP を抽出します。

次のシナリオでは、アプライアンスはパケットの **TOS** フィールドに VIP を追加し、そのパケットを負荷分散サーバーに転送します。次に、負荷分散されたサーバーは、次の図に示すように、アプライアンスをバイパスしてクライアントに直接応答します。

図 1：DSR モードの NetScaler ADC アプライアンス（TOS 付き）



TOS 機能は、制御された環境向けに次のようにカスタマイズされています。

- 環境には、アプライアンスと負荷分散サーバーの間のパスに、ステートフルファイアウォールや TCP ゲートウェイなどのステートフルデバイスがあってはなりません。
- 負荷分散されたサーバーがアプライアンスによって追加されたフィールドと別の TOS フィールドを混同しないように、ネットワークへのすべてのエントリポイントにあるルーターは、すべての受信パケットから TOS フィールドを削除する必要があります。
- 各サーバには 63 個の VIP しか設定できません。
- 中間ルータは、フラグメンテーションに関する ICMP エラーメッセージを送信してはなりません。ソース IP アドレスは負荷分散サーバーの IP アドレスであり、NetScaler VIP ではないため、クライアントはメッセージを認識しません。
- TOS は IP ベースのサービスにのみ有効です。ドメイン名ベースのサービスは TOS では使用できません。

この例では、Service-Any-1 が作成され、仮想サーバー Vserver-LB-1 にバインドされています。仮想サーバーはサービスへのクライアント要求の負荷分散を行い、サービスはアプライアンスをバイパスしてクライアントに直接応答します。次の表は、DSR モードでアプライアンスに設定されているエンティティの名前と値を示しています。

エンティティの種類	名前	IP アドレス	プロトコル
仮想サーバー	Vserver-LB-1	10.102.33.91	ANY
Services	Service-ANY-1	10.102.100.44	ANY
モニター	PING	なし	なし

TOS を使用した DSR では、レイヤー 3 で負荷分散を設定する必要があります。レイヤ 3 の基本的なロードバランシング設定を構成するには、基本ロードバランシングの設定を参照してください。エンティティに名前を付け、前の表で説明した値を使用してパラメータを設定します。

ロード・バランシング・セットアップを構成したら、サーバーがデータ・パケットをカプセル化解除し、クライアントに直接応答してアプライアンスをバイパスできるようにリダイレクション・モードを構成して、DSR モードのロード・バランシング・セットアップをカスタマイズする必要があります。

リダイレクションモードを指定したら、オプションでアプライアンスがサーバーを透過的に監視できるようにすることができます。これにより、アプライアンスは負荷分散されたサーバーを透過的に監視できます。

コマンドラインインターフェイスを使用して仮想サーバーのリダイレクトモードを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <vServerName> -m <Value> -tosId <Value>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -m TOS -tosId 3
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーのリダイレクトモードを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、リダイレクトモードで TOS ID を選択します。

コマンドラインインターフェイスを使用して **TOS** のトランスペアレントモニターを設定するには

コマンドプロンプトで入力します。

```
add monitor <MonitorName> <Type> -destip <DestinationIP> -tos <Value> -
    tosId <Value>
<!--NeedCopy-->
```


例:

```
add monitor mon1 PING -destip 10.102.33.91 -tos Yes  -tosId 3
<!--NeedCopy-->
```

構成ユーティリティを使用して **TOS** 用の透明モニターを作成するには

1. [トラフィック管理] **>** [負荷分散] **>** [モニター] に移動します。
2. モニターを作成し、TOS を選択し、仮想サーバーに指定した TOS ID を入力します。

ワイルドカード **TOS** モニター

TOS フィールドを使用した DSR モードのロードバランシング設定では、サービスをモニタリングするには、TOS モニタを作成し、これらのサービスにバインドする必要があります。TOS モニタでは、VIP アドレスのエンコード値を作成するために VIP アドレスと TOS ID が必要になるため、[TOS] フィールドを使用して DSR モードのロードバランシング設定ごとに個別の TOS モニタが必要です。モニターは、**TOS** フィールドが VIP アドレスのエンコードされた値に設定されているプローブパケットを作成します。次に、プローブパケットをロードバランシング構成のサービスによって表されるサーバーに送信します。

多くの負荷分散構成では、構成ごとに個別のカスタム TOS モニターを作成することは非常に面倒な作業です。これらの TOS モニターの管理も重要な作業です。これで、ワイルドカード TOS モニターを作成できます。同じプロトコル（TCP や UDP など）を使用するすべての負荷分散構成に対してワイルドカード TOS モニターを 1 つだけ作成します。

ワイルドカード TOS モニターには次の必須設定があります。

- タイプ = <protocol>
- TOS = Yes

次のパラメータは、値に設定することも、空白のままにすることもできます。

- 接続先 IP
- 送信先ポート
- TOS ID

DSR サービスにバインドされたワイルドカード TOS モニタ（宛先 IP、宛先ポート、および TOS ID が設定されていない）は、ロードバランシング仮想サーバの TOS ID および VIP アドレスを自動的に学習します。モニタは、エンコードされた VIP アドレスに設定された TOS フィールドを使用してプローブパケットを作成し、DSR サービスによって表されるサーバにプローブパケットを送信します。

CLI を使用してワイルドカード **TOS** モニタを作成するには

コマンドプロンプトで入力します。


```
add lb monitor <monitorName> <Type> -tos YES

show lb monitor <monitorName>
<!--NeedCopy-->
```

CLI を使用してワイルドカード **TOS** モニターをサービスにバインドするには

コマンドプロンプトで入力します。

```
bind lb monitor <monitorName> <serviceName>

show lb monitor <monitorName>
<!--NeedCopy-->
```

GUI を使用してワイルドカード **TOS** モニターを作成するには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. 次のパラメーター設定でモニターを追加します。
   - タイプ = <protocol>
   - TOS = YES

GUI を使用してワイルドカード **TOS** モニターをサービスにバインドするには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. サービスを開き、ワイルドカード TOS モニターをそのサービスにバインドします。

次の構成例では、V1、V2、V3 は ANY タイプの負荷分散仮想サーバーで、TOS ID はそれぞれ 1、2、3 に設定されています。S1、S2、S3、S4、および S5 は ANY タイプのサービスです。S1 と S2 は V1 と V2 の両方にバインドされます。S3、S4、S5、および V1 と V3 の両方にバインドされています。WLCD-TOS-MON は TCP タイプのワイルドカード TOS モニタで、S1、S2、S3、S4、S5 にバインドされています。

WLCD-TOS-MON は、S1、S2、S3、S4、および S5 にバインドされている仮想サーバの TOD ID および VIP アドレスを自動的に学習します。

S1 は V1 と V2 にバインドされているため、WLCD-TOS-MON は S1 の 2 種類のプローブパケットを作成します。1 つは、**TOS** フィールドが V1 のエンコードされた VIP アドレス（203.0.113.1）に設定され、もう 1 つは V2 の VIP アドレス（203.0.113.2）です。その後、NetScaler ADC はこれらのプローブパケットを S1 で表されるサーバーに送信します。同様に、WLCD-TOS-MON は S2、S3、S4、および S5 のプローブパケットを作成します。

```
add lb monitor WLCD-TOS-MON  TCP -tos YES

Done

```



```
add lb vserver V1 ANY 203.0.113.1 * -m TOS  - tosID 1

Done

add lb vserver V2 ANY 203.0.113.2 * -m TOS  - tosID 2

Done

add lb vserver V3 ANY 203.0.113.3 * -m TOS  - tosID 3

Done

add service S1 198.51.100.1 ANY *

Done

add service S2 198.51.100.2 ANY *

Done

add service S3 198.51.100.3 ANY *

Done

add service S4 198.51.100.4 ANY *

Done

add service S5 198.51.100.5 ANY *

Done

bind lb monitor WLCD-TOS-MON S1

Done

bind lb monitor WLCD-TOS-MON S2

Done

bind lb monitor WLCD-TOS-MON S3

Done

bind lb monitor WLCD-TOS-MON S4

Done

bind lb monitor WLCD-TOS-MON S5

Done

bind lb vserver V1 S1, S2, S3, S4, S5
```



```
58
59 Done
60
61 bind lb vserver V2, S1, S2
62
63 Done
64
65 bind lb vserver V3 S3, S4, S5
66
67 Done
68 <!--NeedCopy-->
```

ユースケース 6：TOS フィールドを使用して、DSR モードで IPv6 ネットワークの負荷分散を構成する

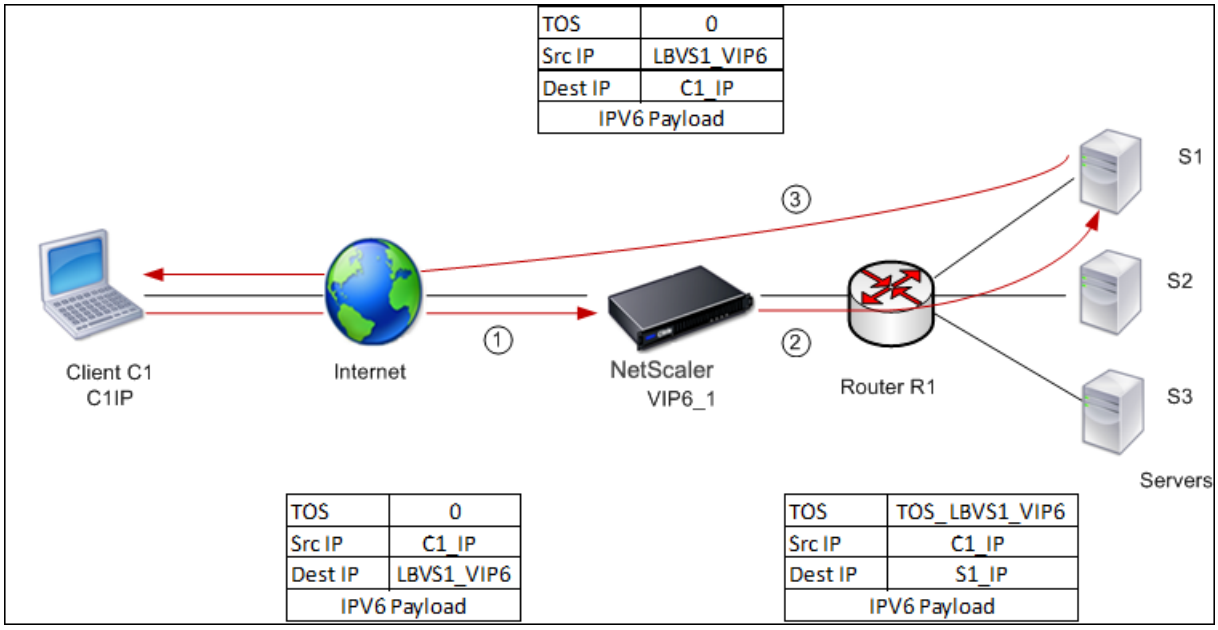
August 15, 2023

NetScaler アプライアンスとサーバーが異なるネットワークにある場合は、サービスの種類（TOS）フィールドを使用して、IPv6 ネットワークのダイレクトサーバーリターン（DSR）モードの負荷分散を構成できます。

注**:** TOS フィールドは、トラフィッククラスフィールドとも呼ばれます。

DSR モードでは、クライアントが NetScaler アプライアンスの VIP6 アドレスに要求を送信すると、アプライアンスはパケットの宛先 IPv6 アドレスをサーバーの IPv6 アドレスに変更してこの要求をサーバーに転送し、IPv6 ヘッダーの TOS（トラフィッククラスとも呼ばれます）フィールドに VIP6 アドレスのエンコードされた値を設定します。TOS フィールドの情報を使用してエンコードされた値から VIP6 アドレスを取得し、それを応答パケットの送信元 IP アドレスとして使用するようにサーバを設定できます。レスポンストラフィックは、アプライアンスをバイパスしてクライアントに直接送信されます。

NetScaler アプライアンス NS1 上に構成された負荷分散仮想サーバー LBVS1 を使用して、サーバー S1、S2、S3 間でトラフィックの負荷分散を行う例を考えてみましょう。NetScaler アプライアンス NS1 とサーバー S1、S2、S3 は異なるネットワークにあるため、ルーター R1 は NS1 とサーバーの間に展開されます。



次の表に、この例で使用されている設定の一覧を示します。

エンティティ	名前
クライアント C1 の IPv6 アドレス	C1_IP (参照のみを目的としています)
NS1 の負荷分散仮想サーバ	LBVS1
LBVS1 の IPv6 アドレス	LBVS1_VIP6 (参照のみを目的としています)
TOS バリュー	TOS_LBVS1_VIP6 (参考用です)
NS1 上のサーバ S1 用サービス	SVC_S1
サーバー S1 の IPv6 アドレス	S1_IP (参照用のみ)
NS1 上のサーバ S2 用サービス	SVC_S2
サーバー S1 の IPv6 アドレス	S2_IP (参照用のみ)
NS1 上のサーバー S3 用サービス	SVC_S3
サーバー S1 の IPv6 アドレス	S3_IP (参照用のみ)

シナリオ例のトラフィックフローは次のとおりです。

1. クライアント C1 は、仮想サーバー LBVS1 に要求を送信します。
2. LBVS1 の負荷分散アルゴリズムはサーバー S1 を選択し、アプライアンスは S1 への接続を開きます。NS1 は、次の内容で要求を S1 に送信します。
   - TOS フィールドは TOS_LBVS1_VIP6 に設定されています。
   - 送信元 IP アドレスは C1_IP です。

3. サーバ S1 は、要求を受信すると、TOS フィールドの情報を使用して、NS1 上の仮想サーバ LBVS1 の IP アドレスである LBVS1_VIP6 アドレスを導出します。サーバーは、アプライアンスをバイパスして、次のように応答を C1 に直接送信します。

    - ソース IP アドレスは DeriveDLBVS1_VIP6 アドレスに設定されます。これにより、クライアントは NS1 上の仮想サーバー LBVS1 と通信し、サーバー S1 とは通信しません。

TOS を使用して **DSR** モードでロードバランシングを設定するには、アプライアンスで次の手順を実行します

1. USIP モードをグローバルに有効にします。
2. サーバーをサービスとして追加します。
3. TOS 値を使用して負荷分散仮想サーバーを構成します。
4. サービスを仮想サーバーにバインドします。

コマンドラインインターフェイスを使用して **TOS** を使用して **DSR** モードのロードバランシングを設定するには

コマンドプロンプトで入力します。

```
enable ns mode USIP

add service <serviceName> <IP> <serviceType> <port>
<!--NeedCopy-->
```

NetScaler ADC アプライアンスに各サーバーをサービスとして追加するには、前述のコマンドを必要な回数だけ繰り返します。

```
add lb vserver <name> <serviceType> <ip> <port> -m <redirectionMode> -
    tosId <positive_integer>

bind lb vserver <vserverName> <serviceName>
<!--NeedCopy-->
```

構成ユーティリティを使用して **USIP** モードを有効にするには

[システム] **>** [設定] **>** [モードの設定] に移動し、[送信元 **IP** アドレスの使用] を選択します。

構成ユーティリティを使用してサービスを作成するには

[トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、サービスを作成します。

構成ユーティリティを使用して負荷分散仮想サーバーとバインドサービスを作成するには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを作成します。
2. サービスセクションをクリックして、サービスをこの仮想サーバーにバインドします。

ユースケース **7**：**IP Over IP** を使用して、**DSR** モードで負荷分散を構成する

August 15, 2023

IP トンネリング（*IP over* IP 構成とも呼ばれる）を使用して、レイヤー 3 ネットワーク全体でダイレクトサーバーリターン（DSR）モードを使用するように NetScaler ADC アプライアンスを構成できます。DSR モードの標準の負荷分散構成と同様に、サーバーは NetScaler ADC アプライアンスを経由するリターンパスを使用する代わりに、クライアントに直接応答できます。これにより、応答時間とスループットが向上します。標準の DSR モードと同様に、NetScaler ADC アプライアンスはサーバーを監視し、アプリケーションポートでヘルスチェックを実行します。

IP over IP 構成では、NetScaler ADC アプライアンスとサーバーが同じレイヤー 2 サブネット上にある必要はありません。代わりに、NetScaler ADC アプライアンスはパケットをカプセル化してから宛先サーバーに送信します。宛先サーバはパケットを受信した後、パケットのカプセル化を解除し、応答をクライアントに直接送信します。これはしばしば L3DSR と呼ばれます。

NetScaler ADC アプライアンスで L3-DSR モードを構成するには：

- 負荷分散仮想サーバーを作成します。モードを IPTUNNEL に設定し、セッションレストラッキングを有効にします。
- サービスを作成します。バックエンドアプリケーションごとにサービスを作成し、サービスを仮想サーバーにバインドします。
- カプセル化解除を設定します。NetScaler ADC アプライアンスまたはバックエンドサーバーのいずれかをカプセル化解除機能として構成します。

  > 注：
  >
  > NetScaler ADC アプライアンスを使用する場合、カプセル化解除の設定は、バックエンドが L2DSR を実サーバーに対して行う ADC アプライアンス間の IP トンネルです。

負荷分散仮想サーバーの構成

アプリケーションへの要求を処理するように仮想サーバーを設定します。サービスと一致するサービスタイプを割り当てるか、複数のサービスに対して ANY のタイプを使用します。
転送方式を IPTUNNEL に設定し、仮想サーバがセッションレスモードで動作できるようにします。使用する負荷分散方式を設定します。

コマンドラインインターフェイスを使用して **IP over IP DSR** 用の負荷分散仮想サーバーを作成および構成するには

コマンドプロンプトで次のコマンドを入力して、IP over IP DSR の負荷分散仮想サーバーを構成し、構成を確認します。

```
add lb vserver <name> serviceType <serviceType> IPAddress <ip> Port <
    port> -lbMethod <method> -m <ipTunnelTag> -sessionless [ENABLED |
    DISABLED]

show lb vserver <name>
<!--NeedCopy-->
```

例**:**

次の例では、負荷分散方式を sourceiPhash として選択し、セッションレス負荷分散を設定しています。

```
add lb vserver Vserver-LB-1 ANY 1.1.1.80 * -lbMethod SourceIPHash -m
    IPTUNNEL -sessionless ENABLED
<!--NeedCopy-->
```

GUI を使用して **IP over IP DSR** の負荷分散仮想サーバーを作成および構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバを作成し、[リダイレクションモード] を [**IP** トンネルベース] に指定します。

IP 経由の **IP DSR** のサービスを構成する

負荷分散サーバーを作成したら、アプリケーションごとに 1 つのサービスを構成します。このサービスは、NetScaler ADC アプライアンスからこれらのアプリケーションへのトラフィックを処理し、NetScaler ADC アプライアンスが各アプリケーションの正常性を監視できるようにします。

USIP モードを使用するようにサービスを割り当て、トンネルベースのモニタリングのために IPTUNNEL タイプのモニタをサービスにバインドします。

コマンドラインインターフェイスを使用して **IP over IP DSR** 用のサービスを作成および構成するには

コマンドプロンプトで次のコマンドを入力してサービスを作成し、オプションでモニターを作成してサービスにバインドします。

```
add service <serviceName> <serverName> <serviceType> <port> -usip <usip
    >

add monitor <monitorName> <monitorType> -destip <ip> -iptunnel <
    iptunnel>

```



```
bind service <serviceName> -monitorName <monitorName>
<!--NeedCopy-->
```

例:

次の例では、タイプ IPTUNNEL のモニタが作成されます。

```
add monitor mon_DSR PING -destip 1.1.1.80 -iptunnel yes
add service svc_DSR01 2.2.2.100 ANY * -usip yes
bind service svc_DSR01 -monitorName mon_DSR
<!--NeedCopy-->
```

サーバと ADC アプライアンスの両方でルーティングを簡素化する別の方法として、ADC とサーバの両方が同じサブネットからの IP を使用するように設定する方法があります。そうすることで、トンネルエンドポイントの宛先を持つすべてのトラフィックがトンネルを介して送信されます。この例では、10.0.1.0/30 が使用されています。

> 注:
>
> モニタの目的は、IP トンネルを介して各サーバのループバックに到達することにより、トンネルがアクティブであることを確認することです。サービスが稼働していない場合は、ADC とサーバ間の外部 IP ルーティングが良好かどうかを確認します。また、内部 IP アドレスが IP トンネルを介して到達可能かどうかも確認します。サーバでルートが必要になる場合や、選択した実装に応じて PBR が ADC に追加されます。

例:

```
add ns ip 10.0.1.2 255.255.255.252 -vServer DISABLED
add netProfile netProfile_DSR -srcIP 10.0.1.2
add lb monitor mon_DSR PING -LRTM DISABLED -destIP 1.1.1.80 -ipTunnel
    YES -netProfile netProfile_DSR
<!--NeedCopy-->
```

GUI を使用してモニターを構成するには

1. **Traffic Management > Load Balancing > Monitors** に移動します。
2. モニターを作成し、[**IP** トンネル] を選択します。

GUI を使用して **IP over IP DSR** のサービスを作成および構成するには

1. **Traffic Management > Load Balancing > Services** に移動します。
2. サービスを作成し、[設定] タブで [送信元 **IP** アドレスを使用] を選択します。

コマンドラインインターフェイスを使用してサービスを負荷分散仮想サーバーにバインドするには

コマンドプロンプトで次のコマンドを入力します。


```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

```
bind lb vserver Vserver-LB-1 Service-DSR-1
<!--NeedCopy-->
```

GUI を使用してサービスを負荷分散仮想サーバーにバインドするには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、[サービス] セクションをクリックして、サービスを仮想サーバーにバインドします。

トンネルパケットの **Outer** ヘッダーでのクライアント **IP** アドレスの使用

NetScaler ADC は、IP トンネリングを使用したダイレクトサーバーリターンモードに関連するトンネルパケットの外側ヘッダーでクライアントソース IP アドレスを、送信元 IP アドレスとして使用することをサポートしています。この機能は、IPv4 を使用した DSR と IPv6 トンネリングモードを使用した DSR でサポートされます。この機能を有効にするには、IPv4 または IPv6 の **use client** 送信元 **IP** アドレスパラメータを有効にします。この設定は、IP トンネリングを使用するすべての DSR 設定にグローバルに適用されます。

CLI を使用してクライアント**-**送信元 **IP** アドレスを送信元 **IP** アドレスとして使用するには

コマンドプロンプトで入力します。

- `set iptunnelparam -useclientsourceip [YES | NO]`
- `show iptunnelparam`

GUI を使用してクライアントの送信元 **IP** アドレスを送信元 **IP** アドレスとして使用するには

1. **[**システム**] > [**ネットワーク**]** に移動します。
2. [設定] タブで、[**IPv4** トンネルのグローバル設定**]** をクリックします。
3. [**IPv4** トンネルグローバルパラメータの設定] ページで、[クライアントソース **IP** を使用] チェックボックスをオンにします。
4. [**OK**] をクリックします。

CLI を使用してクライアントの送信元 **IP** アドレスを送信元 **IP** アドレスとして使用するには

コマンドプロンプトで入力します。

- `set ip6tunnelparam -useclientsourceip [YES | NO]`
- `show ip6tunnelparam`

GUI を使用してクライアントの送信元 **IP** アドレスを送信元 **IP** アドレスとして使用するには

1. **[**システム**] > [**ネットワーク**]** に移動します。
2. [設定] タブで、[**IPv6** トンネルのグローバル設定**]** をクリックします。
3. [**IPv6** トンネルグローバルパラメータの設定] ページで、[クライアントソース **IP** を使用] チェックボックスをオンにします。
4. [**OK**] をクリックします。

カプセル化解除設定

NetScaler ADC アプライアンスまたはバックエンドサーバーのいずれかをカプセル化解除として構成できます。

NetScaler カプセル化解除

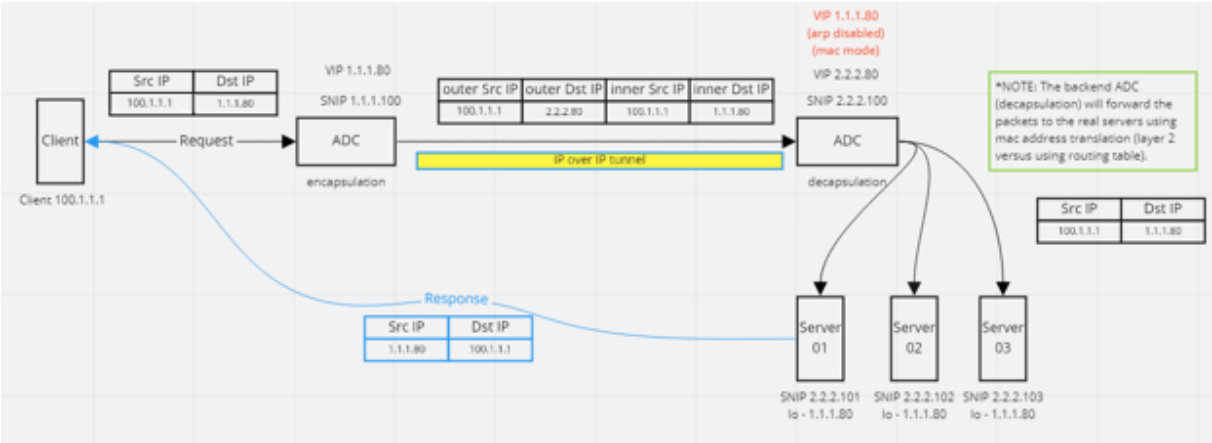
NetScaler ADC アプライアンスをカプセル化解除として使用する場合は、NetScaler ADC アプライアンスに IP トンネルを作成する必要があります。詳細については、IP トンネルの設定を参照してください。

NetScaler カプセル化解除セットアップは、次の 2 つの仮想サーバーで構成されます。

- 最初の仮想サーバがカプセル化されたパケットを受信し、外部 IP カプセル化を削除します。
- 2 番目の仮想サーバは、フロントエンド ADC 上の元のサービスの IP を持ち、MAC 変換を使用して、バインドされたサービスの MAC アドレスを使用してパケットをバックエンドに転送します。このセットアップは、通常 L2DSR と呼ばれます。この仮想サーバで ARP を無効にします。

設定例**:**

次の図は、ADC アプライアンスを使用したカプセル化解除の設定を示しています。



セットアップに必要な完全な構成は次のとおりです。

フロントエンド **ADC** 構成**:**

```
add service svc_DSR01 2.2.2.80 ANY * -usip YES -useproxyport NO
add lb vserver vip_DSR_ENCAP ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
    IPTUNNEL -sessionless ENABLED
bind lb vserver vip_DSR_ENCAP svc_DSR01
<!--NeedCopy-->
```

バックエンド **ADC** 構成**:**

```
add ipTunnel DSR-IPIP 1.1.1.100 255.255.255.255 *

add service svc_DSR01_01 2.2.2.101 ANY * -usip YES -useproxyport NO
add service svc_DSR01_02 2.2.2.102 ANY * -usip YES -useproxyport NO
add service svc_DSR01_03 2.2.2.103 ANY * -usip YES -useproxyport NO

add lb vserver vs_DSR_DECAP ANY 2.2.2.80 * -lbMethod SOURCEIPHASH -m
    IPTUNNEL -sessionless ENABLED -netProfile netProf_DSR_MBF_noIP

add ns ip 1.1.1.80 255.255.255.255 -type VIP -arp DISABLED -snmp
    DISABLED
add lb vserver vs_DSR_Relay ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
    MAC -sessionless ENABLED

bind lb vserver vs_DSR_DECAP svc_DSR01_01
bind lb vserver vs_DSR_DECAP svc_DSR01_02
bind lb vserver vs_DSR_DECAP svc_DSR01_03

bind lb vserver vip_DSR_Relay svc_DSR01_01
bind lb vserver vip_DSR_Relay svc_DSR01_02
bind lb vserver vip_DSR_Relay svc_DSR01_03

add netProfile netProf_DSR_MBF_noIP -MBF ENABLED
add lb monitor mon_DSR_MAC PING -netProfile netProf_DSR_MBF_noIP
bind service svc_DSR01_01 -monitorName mon_DSR_MAC
bind service svc_DSR01_02 -monitorName mon_DSR_MAC
bind service svc_DSR01_03 -monitorName mon_DSR_MAC
<!--NeedCopy-->
```

次の例は、apache2 を実行している Ubuntu サーバーと Red Hat サーバーを使用したテストセットアップを示しています。これらのコマンドは、各バックエンドサーバーで設定されます。

```
sudo ip addr add 1.1.1.80 255.255.255.255 dev lo
sudo sysctl net.ipv4.conf.all.arp_ignore=1
sudo sysctl net.ipv4.conf.all.arp_announce=2
sudo sysctl net.ipv4.conf.eth4.rp_filter=2 (The interface has the
    external IP with route towards the ADC)
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo ip link set dev lo arp on
<!--NeedCopy-->
```

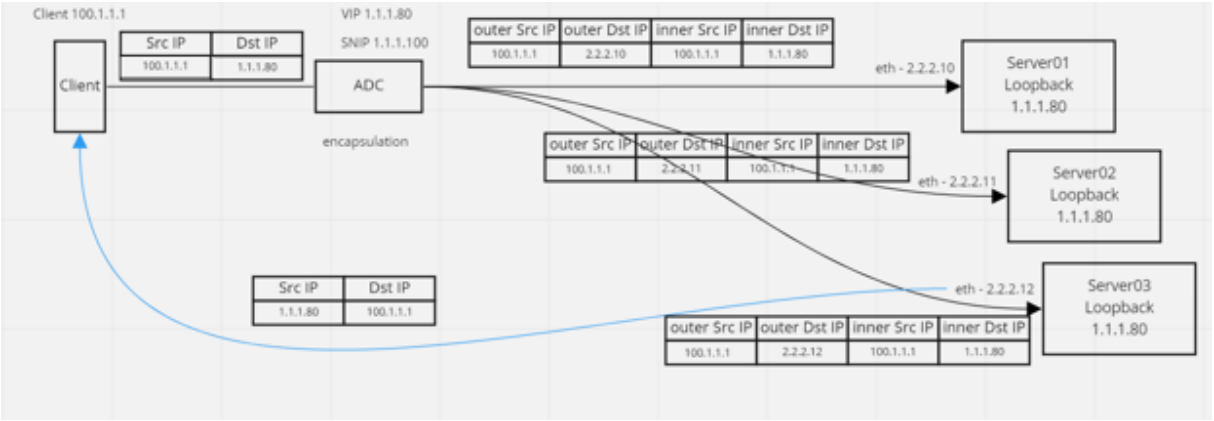

バックエンドサーバーのカプセル化解除

バックエンドサーバーをカプセル化解除として使用する場合、バックエンドの構成はサーバーの OS タイプによって異なります。次の手順に従って、バックエンドサーバーをカプセル化解除として構成できます。

1. サービス IP の IP を使用してループバックインターフェイスを設定します。
2. トンネルインターフェイスを作成します。
3. トンネルインターフェイスを介したルートを追加します。
4. トラフィックに必要なインターフェイス設定を構成します。

> 注:
>
> Windows OS サーバでは IP トンネリングをネイティブに実行できないため、Linux ベースのシステムの例としてコマンドが提供されています。Windows OS サーバーではサードパーティプラグインを使用できますが、この例の範囲外です。

次の図は、バックエンドサーバーを使用したカプセル化解除の設定を示しています。



設定例:

この例では、1.1.1.80 は NetScaler ADC 仮想 IP（VIP）アドレスで、2.2.2.10-2.2.2.12 はバックエンドサーバーの IP アドレスです。VIP アドレスはループバックインターフェイスで設定され、ルートはトンネルインターフェイスを介して追加されます。モニタはサーバ IP を使用し、トンネルエンドポイントを使用して IP トンネル経由でモニタパケットをトンネリングします。

セットアップに必要な完全な構成は次のとおりです。

フロントエンド **ADC** 構成**:**

次の設定では、トンネルエンドポイントをソースとして使用するモニタが作成されます。次に、トンネル経由でサービス IP アドレスに ping を送信します。

```
1 add ns ip 10.0.1.2 255.255.255.252 -vServer DISABLED
2 add netProfile netProfile_DSR -srcIP 10.0.1.2
3 add lb monitor mon_DSR PING -LRTM DISABLED -destIP 1.1.1.80 -ipTunnel
     YES -netProfile netProfile_DSR
4 <!--NeedCopy-->
```


次の設定では、元の送信元 IP アドレスを使用するサービスの VIP を作成します。次に、IP トンネル経由でバックエンドサーバーにトラフィックを転送します。

```
add service svc_DSR01 2.2.2.10 ANY * -usip YES -useproxyport NO
bind service svc_DSR01 -monitorName mon_DSR

add service svc_DSR02 2.2.2.11 ANY * -usip YES -useproxyport NO
bind service svc_DSR02 -monitorName mon_DSR

add service svc_DSR03 2.2.2.12 ANY * -usip YES -useproxyport NO
bind service svc_DSR03 -monitorName mon_DSR

add lb vserver vip_DSR_ENCAP ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
    IPTUNNEL -sessionless ENABLED
bind lb vserver vip_DSR_ENCAP svc_DSR01
bind lb vserver vip_DSR_ENCAP svc_DSR02
bind lb vserver vip_DSR_ENCAP svc_DSR03
<!--NeedCopy-->
```

各サーバーのバックエンドサーバー構成**:**

次のコマンドは、バックエンドサーバーが IPIP パケットを受信し、外部カプセル化を削除し、ループバックから元のクライアント IP に応答するために必要です。そうすることで、クライアントが受信したパケット内の IP アドレスが元の要求の IP アドレスと一致するようにします。

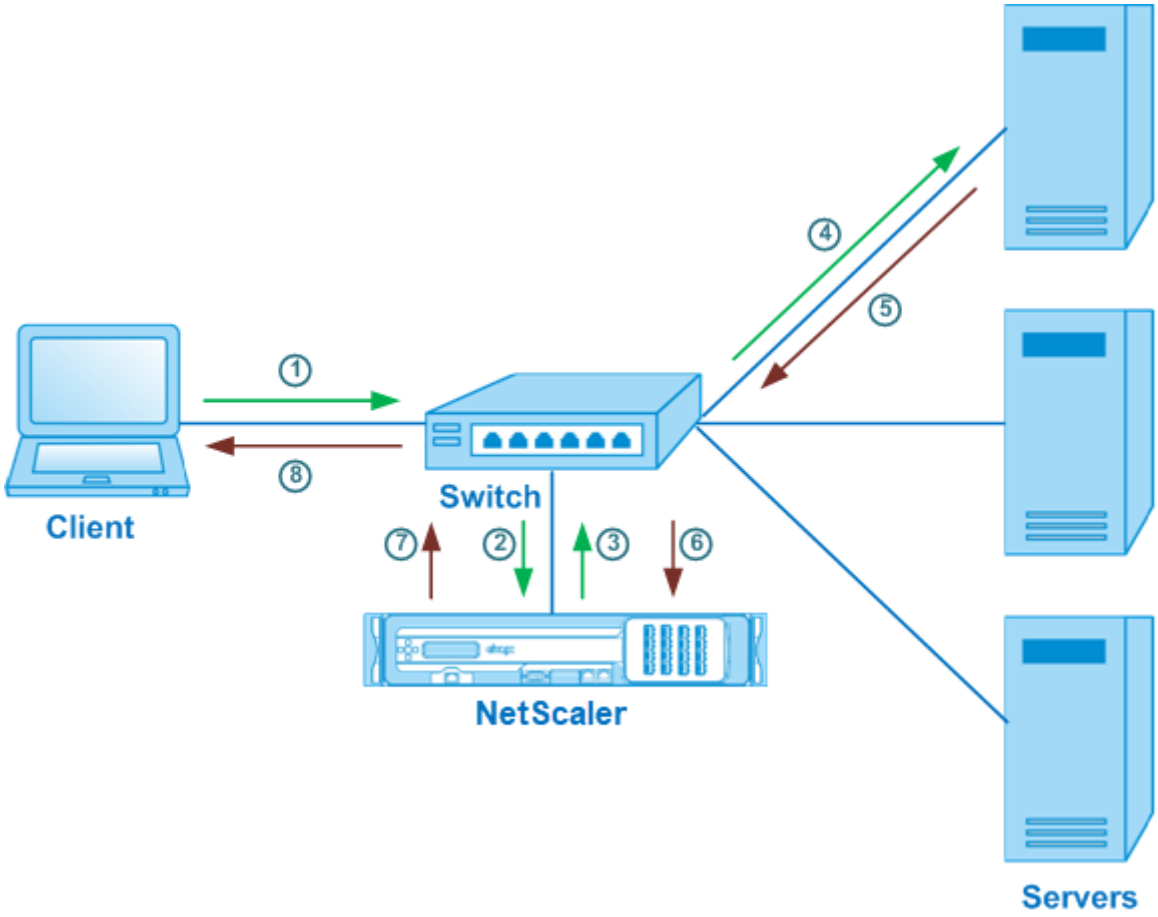
```
modprobe ipip
sudo ip addr add 1.1.1.80 255.255.255.255 dev lo
nmcli connection add type ip-tunnel ip-tunnel.mode ipip con-name tun0
ifname tun0 remote 198.51.100.5 local 203.0.113.10
nmcli connection modify tun0 ipv4.addresses '10.0.1.1/30'
nmcli connection up tun0
sudo sysctl net.ipv4.conf.all.arp_ignore=1
sudo sysctl net.ipv4.conf.all.arp_announce=2
sudo sysctl net.ipv4.conf.tun0.rp_filter=2
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo ip link set dev lo arp off
<!--NeedCopy-->
```

ユースケース **8**: ワンアームモードで負荷分散を構成する

August 15, 2023

ワンアームセットアップでは、NetScaler アプライアンスを単一の VLAN を介してネットワークに接続します。アプライアンスは、単一の VLAN 上のクライアントから要求を受信し、その要求を同じ VLAN 上のサーバに送信します。これは、ルータ、サーバ、アプライアンスをすべて同じスイッチに接続する最も簡単な導入シナリオの 1 つです。スイッチでのクライアント要求はアプライアンスに転送され、アプライアンスは設定されたロードバランシング方式を使用してサービスを選択します。

図 1: ワンアームモードでのロードバランシング



この例のシナリオでは、Service-ANY-1、Service-ANY-2、Service-ANY-3 が作成され、仮想サーバ Vserver-LB-1 にバインドされます。仮想サーバーは、クライアント要求をサービスにロードバランシングします。次の表は、ワンアームモードでアプライアンスに設定されているエンティティの名前と値を示しています。

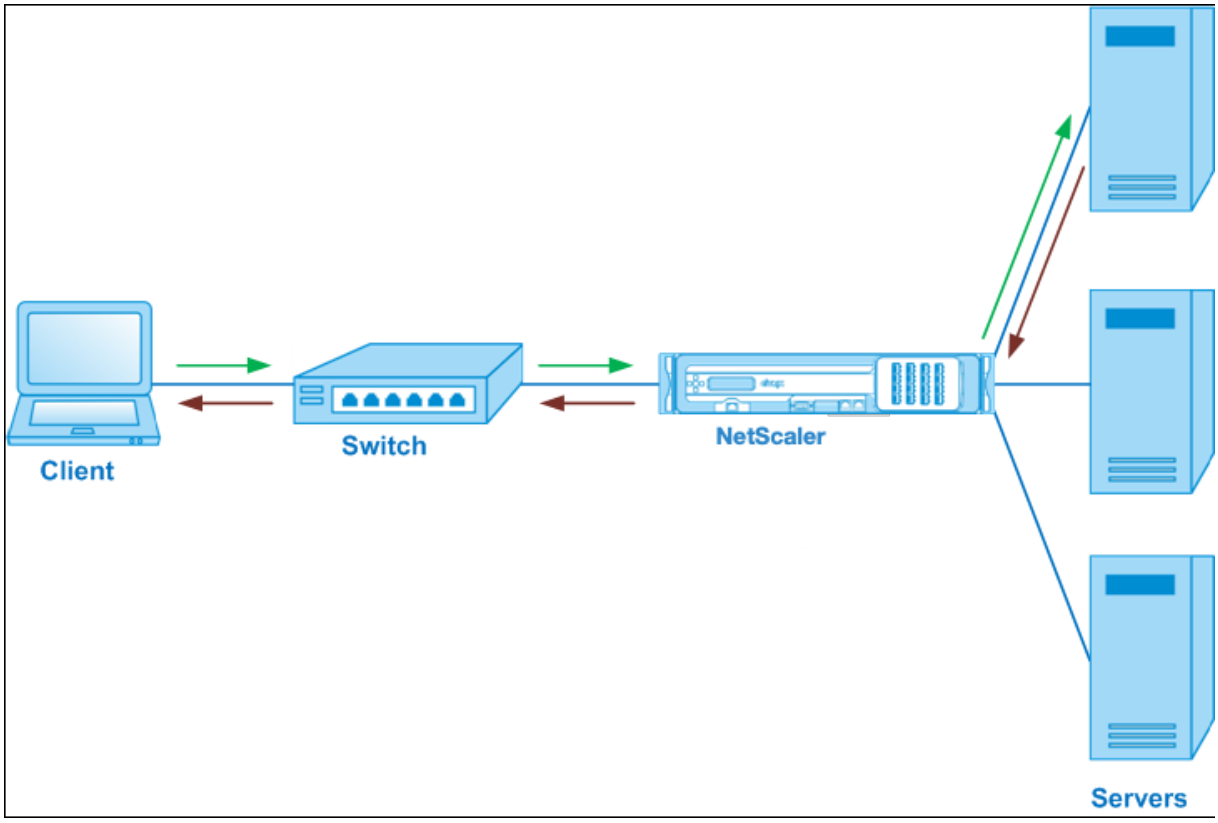
エンティティタイプ	名前	IP アドレス	プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.94	ANY
Services	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY
	Service-ANY-3	10.102.29.93	ANY
モニター	TCP	なし	なし

ワンアームモードでロードバランシング設定を構成するには、基本ロードバランシングの設定を参照してください。

ユースケース 9：インラインモードで負荷分散を構成する

August 15, 2023

インラインモード（ツーアームモードとも呼ばれる）セットアップでは、NetScaler アプライアンスを複数の VLAN を介してネットワークに接続します。アプライアンスは、ある VLAN 上のクライアントから要求を受信し、別の VLAN 上のサーバーに要求を送信します。ツーアーム構成では、アプライアンスはサーバーとクライアント間で接続されます。スイッチでのクライアント要求はアプライアンスに転送され、アプライアンスは設定されたロードバランシング方式を使用してサービスを選択します。



インラインモードの設定とエンティティ図は、ワンアームモードでのロードバランシングの設定で説明されているものと同じです。

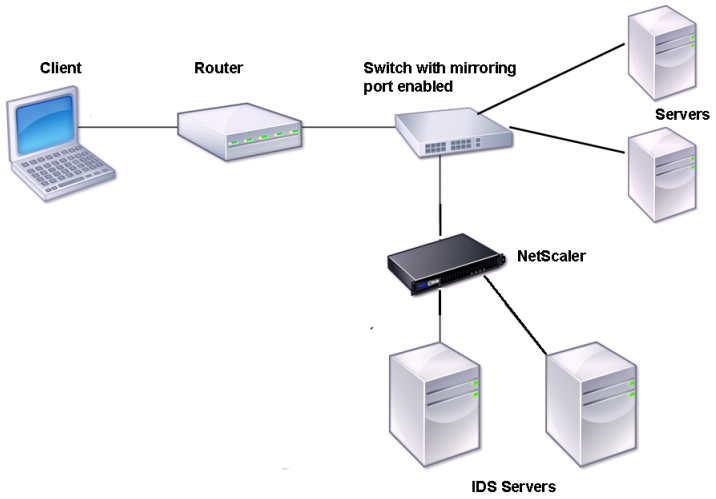
ユースケース 10：侵入検知システムサーバーの負荷分散

August 15, 2023

NetScaler アプライアンスが侵入検知システム（IDS）サーバーの負荷分散をサポートできるようにするには、IDS サーバーとクライアントをポートミラーリングが有効になっているスイッチを介して接続する必要があります。クラ

イアントはサーバーに要求を送信します。スイッチでポートミラーリングが有効になっているため、要求パケットは NetScaler アプライアンスの仮想サーバーポートにコピーまたは送信されます。次に、アプライアンスは、次の図に示すように、設定された負荷分散方法を使用して IDS サーバーを選択します。

図 1: ロードバランシングされた IDS サーバのトポロジ



注: 現在、アプライアンスはパッシブ IDS デバイスのロードバランシングのみをサポートしています。

上の図に示すように、IDS ロードバランシング設定は次のように機能します。

1. クライアントの要求は IDS サーバに送信され、ミラーリングポートが有効になっているスイッチがこれらのパケットを IDS サーバに転送します。送信元 IP アドレスはクライアントの IP アドレスで、宛先 IP アドレスはサーバの IP アドレスです。送信元 MAC アドレスはルータの MAC アドレスで、宛先 MAC アドレスはサーバの MAC アドレスです。
2. スイッチを通過するトラフィックは、アプライアンスにミラーリングされます。アプライアンスは、レイヤ 3 情報（送信元 IP アドレスと宛先 IP アドレス）を使用して、送信元 IP アドレスまたは宛先 IP アドレスを変更せずに、選択した IDS サーバにパケットを転送します。送信元 MAC アドレスと宛先 MAC アドレスを、選択した IDS サーバの MAC アドレスに変更します。

注:IDS サーバーの負荷分散を行う場合、SRCIPHASH、DESTIPHASH、または SRCIPDESTIPHASH のロードバランシング方法を設定できます。クライアントからアプライアンス上のサービスに流れるパケットは単一の IDS サーバーに送信する必要があるため、SRCIPDESTIPHASH 方式が推奨されます。

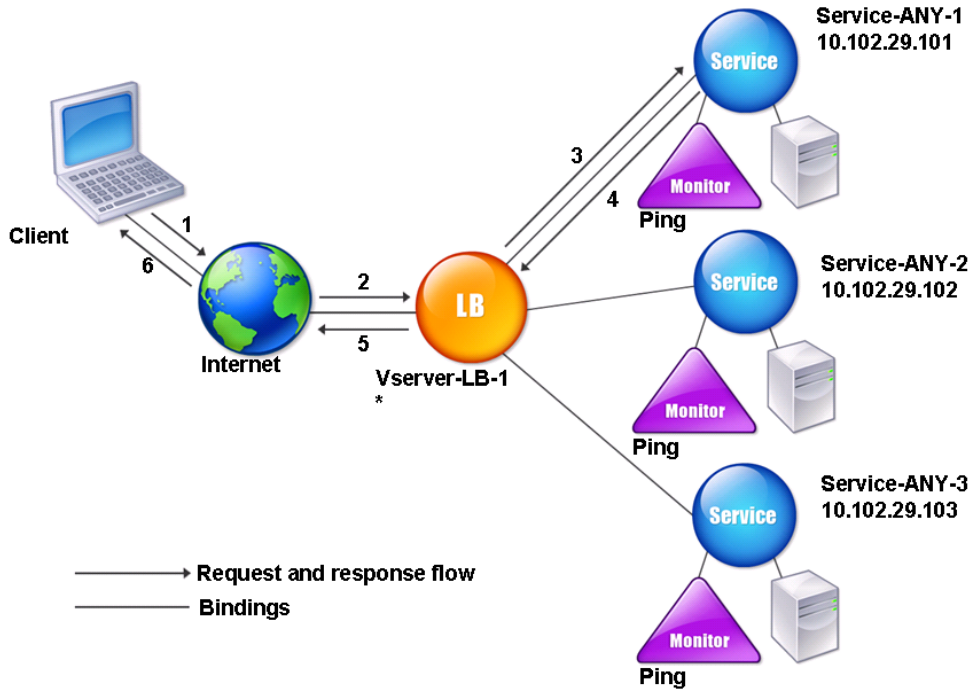
サービス任意-1、サービス任意-2、およびサービス任意-3 が作成され、VServer-LB-1 にバインドされているとします。仮想サーバーはサービスの負荷を分散します。次の表は、アプライアンスに設定されているエンティティの名前と値を示しています。

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
仮想サーバー	Vserver-LB-1	*	*	ANY
Services	Service-ANY-1	10.102.29.101	*	ANY
	Service-ANY-2	10.102.29.102	*	ANY
	Service-ANY-3	10.102.29.103	*	ANY
モニター	Ping	なし	なし	なし

注:IDS ロードバランシングの設定には、インラインモードまたはワンアームモードを使用できます。

次の図は、アプライアンスに設定する負荷分散エンティティとパラメーターの値を示しています。

図 2: ロードバランシング IDS サーバのエンティティモデル



IDS ロードバランシング設定を設定するには、最初に MAC ベースの転送を有効にする必要があります。アプライアンスのレイヤ 2 モードとレイヤ 3 モードも無効にします。

コマンドラインインターフェイスを使用して **MAC** ベースの転送を有効にするには

コマンドプロンプトで入力します。

```
enable ns mode <ConfigureMode>
<!--NeedCopy-->
```

例:

```
enable ns mode MAC
<!--NeedCopy-->
```

設定ユーティリティを使用して **MAC** ベースの転送を有効にするには

[システム] **>** [設定] **>** [モードの設定] に移動し、[**MAC** ベースの転送] を選択します。

次に、基本的な負荷分散設定を構成するには、「基本負荷分散の設定」を参照してください。

基本的なロードバランシング設定を設定したら、サポートされているロードバランシング方式（セッションレス仮想サーバ上の SRCIPDESTIP Hash 方式など）を設定し、MAC モードを有効にして IDS 用にカスタマイズする必要があります。アプライアンスは接続の状態を維持せず、パケットを処理せずに IDS サーバーに転送するだけです。仮想サーバが MAC モードであるため、宛先 IP アドレスとポートは変更されません。

コマンドラインインターフェイスを使用してセッションレス仮想サーバーの負荷分散方式とリダイレクトモードを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <
    RedirectionMode> -sessionless <Value>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPDestIPHash -m MAC -
    sessionless enabled
<!--NeedCopy-->
```

> 注
>
> -m MAC オプションが有効になっている仮想サーバーにバインドされているサービスの場合は、非ユーザーモニターをバインドする必要があります。

構成ユーティリティを使用してセッションレス仮想サーバーの負荷分散方式とリダイレクトモードを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、「リダイレクションモード」で「MAC ベース」を選択します。
3. 「詳細設定」で、「メソッド」をクリックし、「SRCIPDESTIPHASH」を選択します。[トラフィックの設定] をクリックし、[セッションレスロードバランシング] を選択します。

コマンドラインインターフェイスを使用して送信元 **IP** アドレスを使用するようにサービスを設定するには

コマンドプロンプトで入力します。

```
set service <ServiceName> -usip <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-ANY-1 -usip yes
<!--NeedCopy-->
```

構成ユーティリティを使用して送信元 **IP** アドレスを使用するようにサービスを設定するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. サービスを開き、[設定] で [ソース **IP** アドレスを使用] を選択します。

USIP を正しく機能させるには、グローバルに設定する必要があります。USIP をグローバルに設定する方法の詳細については、「IP アドレッシング」を参照してください。

ユースケース **11**: リッスンポリシーを使用してネットワークトラフィックを分離する

August 15, 2023

> 注**:**
>
> シャドウ仮想サーバーを使用してマルチテナント分離をシミュレートするトラフィック分離ソリューションは推奨されなくなりました。または、このような展開には NetScaler ADC AdminPartitioning 機能を使用することをお勧めします。詳細については、「管理者のパーティショニング」を参照してください。

データセンターの一般的なセキュリティ要件は、さまざまなアプリケーションまたはテナントのトラフィック間のネットワークパスの分離を維持することです。あるアプリケーションまたはテナントのトラフィックは、他のアプリケーションまたはテナントのトラフィックから分離する必要があります。たとえば、ある金融サービス会社は、保険部

門のアプリケーションのトラフィックと金融サービスアプリケーションのトラフィックを分離したいと思うでしょう。これまでは、ファイアウォール、ロードバランサー、IdP などのネットワークサービスデバイスを物理的に分離し、スイッチングファブリックでネットワークを監視して論理的に分離することで簡単に実現できました。

データセンターのアーキテクチャがマルチテナントの仮想化データセンターへと進化するにつれて、データセンターの集約レイヤーにおけるネットワークサービスが統合されつつあります。この開発により、ネットワーク・パス・アイソレーションはネットワーク・サービス・デバイスにとって重要なコンポーネントとなり、ADC は L4 から L7 のレベルでトラフィックを分離できる必要性が高まっています。さらに、特定のテナントのすべてのトラフィックは、サービスレイヤに到達する前にファイアウォールを通過する必要があります。

ネットワークパスを分離する要件に対処するために、NetScaler ADC アプライアンスはネットワークドメインを識別し、ドメイン間のトラフィックを制御します。NetScaler ADC ソリューションには、リッスンポリシーとシャドウ仮想サーバーの 2 つの主要コンポーネントがあります。

隔離される各ネットワークパスには仮想サーバーが割り当てられ、その仮想サーバーは指定されたネットワークドメインからのトラフィックのみを受信するようにリッスンポリシーが定義されます。

トラフィックを分離するために、リッスンポリシーを複数のクライアントパラメータまたはそれらの組み合わせに基づいて設定し、ポリシーに優先順位を割り当てることができます。次の表は、トラフィックを識別するためのリッスンポリシーで使用できるパラメータを示しています。

カテゴリ	パラメーター
イーサネットプロトコル	送信元 MAC アドレス、宛先 MAC アドレス
ネットワークインターフェイス	ネットワーク ID、受信スループット、送信スループット、送信スループット
IP プロトコル	送信元 IP アドレス、宛先 IP アドレス
IPv6 プロトコル	送信元 IPv6 アドレス、宛先 IPv6 アドレス
TCP プロトコル	送信元ポート、宛先ポート、最大セグメントサイズ、ペイロード、およびその他のオプション
UDP プロトコル	送信元ポート、宛先ポート
VLAN	ID

表 1. リッスンポリシーの定義に使用されるクライアントパラメータ

NetScaler アプライアンスでは、仮想サーバーがドメインごとに構成され、仮想サーバーがそのドメインのトラフィックのみを受信するように指定するリッスンポリシーが設定されます。また、各ドメインには、任意のドメイン宛てのトラフィックを受信するシャドウ負荷分散仮想サーバーが構成されています。各シャドウロードバランシング仮想サーバーにはワイルドカード (*) IP アドレスとポートがあり、そのサービスタイプは ANY に設定されています。

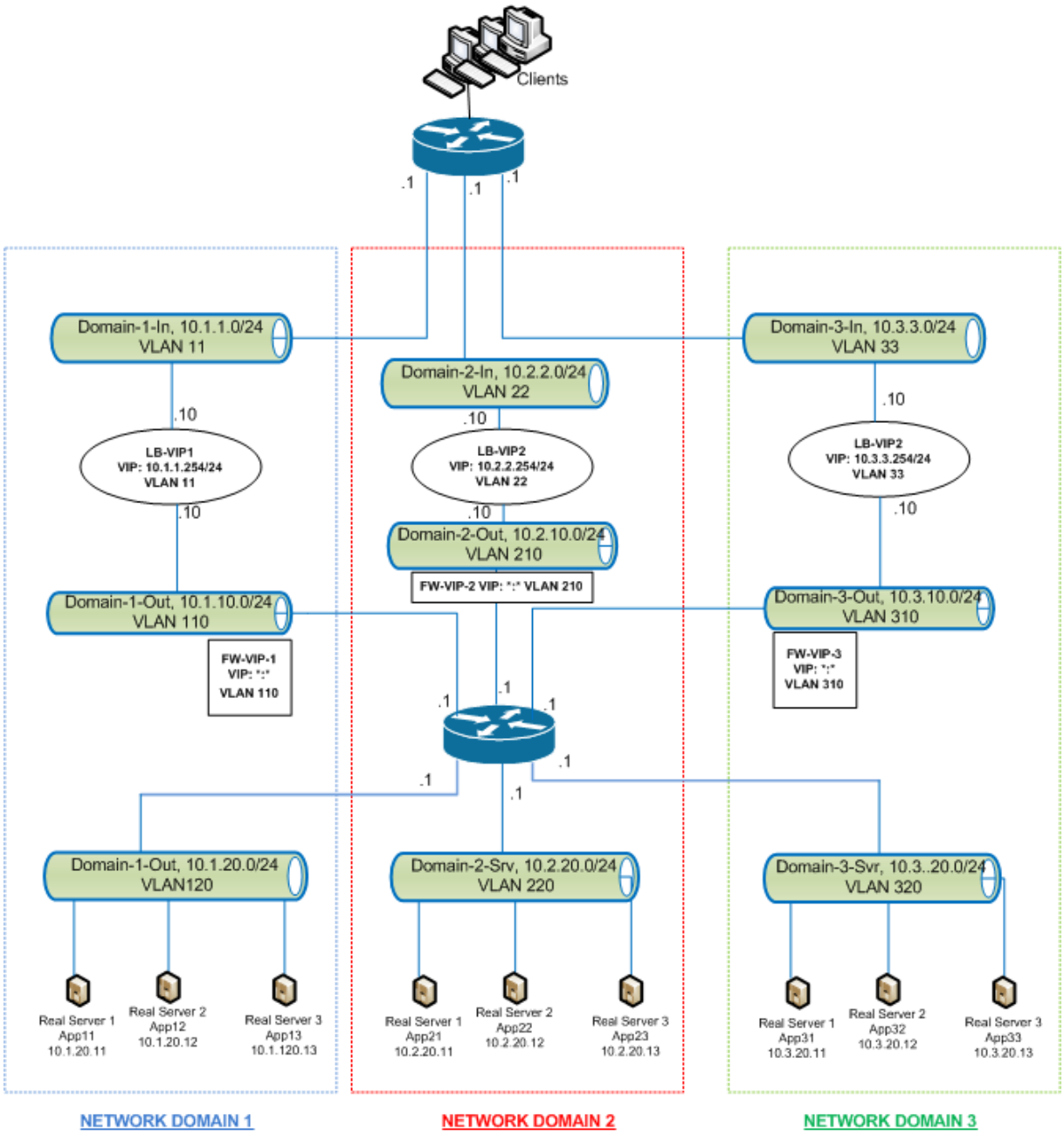
各ドメインでは、ドメインのファイアウォールがサービスとしてシャドウロードバランシング仮想サーバーにバインドされ、シャドウロードバランシング仮想サーバーはすべてのトラフィックをファイアウォール経由で転送します。

ローカルトラフィックは宛先に転送され、別のドメイン宛てのトラフィックはそのドメインのファイアウォールに転送されます。シャドウロードバランシング仮想サーバーは MAC モードリダイレクト用に設定されています。

ネットワークパスの分離方法

次の図は、ドメイン全体の一般的なトラフィックフローを示しています。ネットワークドメイン 1 内、およびネットワークドメイン 1 とネットワークドメイン 2 の間のトラフィックフローを考えてみましょう。

図 1: ネットワークパスの分離



ネットワークドメイン **1** 内のトラフィック

ネットワークドメイン 1 には、VLAN 11、VLAN110、VLAN120 の 3 つの VLAN があります。次の手順では、トラフィックフローについて説明します。

- VLAN 11 のクライアントは、VLAN 120 のサービスプールから利用可能なサービスに対する要求を送信します。
- VLAN 11 からのトラフィックを受信するように設定された負荷分散仮想サーバー LB-VIP1 は、要求を受信し、要求を VLAN 110 に転送します。VLAN110 の仮想サーバは、要求をシャドウロードバランシング仮想サーバ FW-VIP-1 に転送します。
- VLAN 110 からのトラフィックを受信するように設定された FW-VIP-1 は、要求を受信して VLAN 120 に転送します。
- VLAN 120 の負荷分散仮想サーバーは、App11、App12、または App13 のいずれかの物理サーバーに要求を負荷分散します。
- 物理サーバから送信された応答は、同じパスで VLAN 11 のクライアントに返されます。

この構成により、クライアントから送信されるすべてのトラフィックについて、トラフィックが常に NetScaler 内で分離されます。

ネットワークドメイン **1** とネットワークドメイン **2** の間のトラフィック

ネットワークドメイン 1 には、VLAN 11、VLAN110、VLAN120 の 3 つの VLAN があります。ネットワークドメイン 2 には、VLAN 22、VLAN 210、VLAN 220 の 3 つの VLAN もあります。次の手順では、VALN 11 から VLAN 22 へのトラフィックフローについて説明します。

- ネットワークドメイン 1 に属する VLAN 11 のクライアントが、ネットワークドメイン 2 に属する VLAN 220 のサービスプールから利用可能なサービスに対する要求を送信します。
- ネットワークドメイン 1 では、VLAN 11 からのトラフィックを受信するように構成された負荷分散仮想サーバー LB-VIP1 が要求を受信し、その要求を VLAN110 に転送します。
- 他のドメイン宛ての VLAN 110 トラフィックを受信するように構成されたシャドウロードバランシング仮想サーバー FW-VIP-1 は、要求を受信してファイアウォール仮想サーバー FW-VIP-2 に転送します。これは、要求の宛先がネットワークドメイン 2 の物理サーバーであるためです。
- ネットワークドメイン 2 では、FW-VIP-2 は VLAN 220 に要求を転送します。
- VLAN 220 の負荷分散仮想サーバーは、App21、App22、または App23 のいずれかの物理サーバーに要求を負荷分散します。
- 物理サーバーから送信された応答は、同じパスでネットワークドメイン 2 のファイアウォールを経由して返され、次にネットワークドメイン 1 に戻って VLAN 11 のクライアントに到達します。

構成の手順

リッスンポリシーを使用してネットワークパスアイソレーションを設定するには、以下を実行してください。

- リッスンポリシー表現を追加します。各式は、トラフィックの宛先となるドメインを指定します。VLAN ID またはその他のパラメータを使用してトラフィックを識別できます。
- ネットワークドメインごとに、次のように 2 つの仮想サーバーを構成します。
  - このドメイン宛てのトラフィックを識別するリッスンポリシーを指定する、負荷分散仮想サーバーを作成します。前に作成した式の名前を指定することも、仮想サーバーの作成中に式を作成することもできます。
  - シャドウ仮想サーバーと呼ばれる別の負荷分散仮想サーバーを作成します。このサーバーには、任意のドメイン宛てのトラフィックに適用されるリッスンポリシー式を指定します。この仮想サーバーでは、サービスタイプを ANY に、IP アドレスとポートをアスタリスク (*) に設定します。この仮想サーバーで MAC ベースの転送を有効にします。
  - 両方の仮想サーバーで L2 接続オプションを有効にします。

    一般に、NetScaler ADC アプライアンスは、接続を識別するために、クライアント IP アドレス、クライアントポート、宛先 IP アドレス、宛先ポートの 4 タプルを使用します。[L2 接続] オプションを有効にすると、通常の 4 タプルに加えて、接続のレイヤ 2 パラメータ（チャネル番号、MAC アドレス、VLAN ID）が使用されます。
- ドメイン内のサーバプールを表すサービスを追加し、仮想サーバにバインドします。
- 各ドメインのファイアウォールをサービスとして構成し、すべてのファイアウォールサービスをシャドウ仮想サーバーにバインドします。

コマンドラインインターフェイスを使用してネットワークトラフィックを分離するには

コマンドプロンプトで、次のコマンドを入力します。

```
add policy expression <expressionName> <listenPolicyExpression>

add lb vserver <name> <serviceType> <ip> <port> -l2conn ON -
    listenPolicy <expressionName>
<!--NeedCopy-->
```

ドメインごとに負荷分散仮想サーバーを追加します。この仮想サーバーは、同じドメインのトラフィック用です。

```
add lb vserver <name> ANY * * -l2conn ON -m MAC -listenPolicy <
    expressionName>
<!--NeedCopy-->
```

ドメインごとにシャドウ負荷分散仮想サーバーを追加します。この仮想サーバーは、他のドメインのトラフィック用です。

例**:**

```
add policy expression e110 client.vlan.id==110
add policy expression e210 client.vlan.id==210
add policy expression e310 client.vlan.id==310
```



```
4  add policy expression e11 client.vlan.id==11
5  add policy expression e22 client.vlan.id==22
6  add policy expression e33 client.vlan.id==33
7
8  add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -persistenceType NONE -
       listenPolicy e11
9  -cltTimeout 180 -l2Conn ON
10
11 add lb vserver LB-VIP2 HTTP 10.2.2.254 80 -persistenceType NONE -
       listenPolicy e22
12 -cltTimeout 180 -l2Conn ON
13
14 add lb vserver LB-VIP3 HTTP 10.3.3.254 80 -persistenceType NONE -
       listenPolicy e33
15 -cltTimeout 180 -l2Conn ON
16
17
18 add lb vserver FW-VIP-1 ANY * * -persistenceType NONE -lbMethod
       ROUNDROBIN - listenPolicy e110 -Listenpriority 1 -m MAC -cltTimeout
       120
19
20 add lb vserver FW-VIP-2 ANY * * -persistenceType NONE -lbMethod
       ROUNDROBIN - listenPolicy e210 -Listenpriority 2 -m MAC -cltTimeout
       120
21
22 add lb vserver FW-VIP-3 ANY * * -persistenceType NONE -lbMethod
       ROUNDROBIN - listenPolicy e310 -Listenpriority 3 -m MAC -cltTimeout
       120
23
24
25 add service RD-1 10.1.1.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
       DISABLED
26 -usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
       NO -TCPB NO -CMP NO
27
28 add service RD-2 10.2.2.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
       DISABLED
29 -usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
       NO -TCPB NO -CMP NO
30
31 add service RD-3 10.3.3.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
       DISABLED
32 -usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
       NO -TCPB NO -CMP NO
33
34
35 bind lb vserver FW-VIP-1 RD-1
36
37 bind lb vserver FW-VIP-2 RD-2
38
39 bind lb vserver FW-VIP-3 RD-3
40 <!--NeedCopy-->
```


構成ユーティリティを使用してネットワークトラフィックを分離するには

1. サービスの作成の説明に従って、サーバを表すサービスを追加します。
2. 各ファイアウォールをサービスとして追加します。

   a) ［**Traffic Management**］**>** ［**Load Balancing**］**>** ［**Services**］の順に移動します。
   b) プロトコルを ANY、サーバをファイアウォールの IP アドレス、ポートを 80 に指定して、サービスを作成します。

3. 負荷分散仮想サーバーを構成します。
4. シャドウロードバランシング仮想サーバーを設定します。
5. ネットワークドメインごとに、手順 3 と 4 を繰り返します。
6. [負荷分散仮想サーバー] ウィンドウで、作成した仮想サーバーを開き、設定を確認します。

ユースケース **12**：負荷分散用の **Citrix** 仮想デスクトップの構成

August 15, 2023

仮想デスクトップアプリケーションの配信パフォーマンスを向上させるには、NetScaler ADC アプライアンスを Citrix Virtual Desktops と統合し、NetScaler ADC 負荷分散機能を使用して Desktop Delivery Controller（DDC）サーバー全体に負荷を分散します。

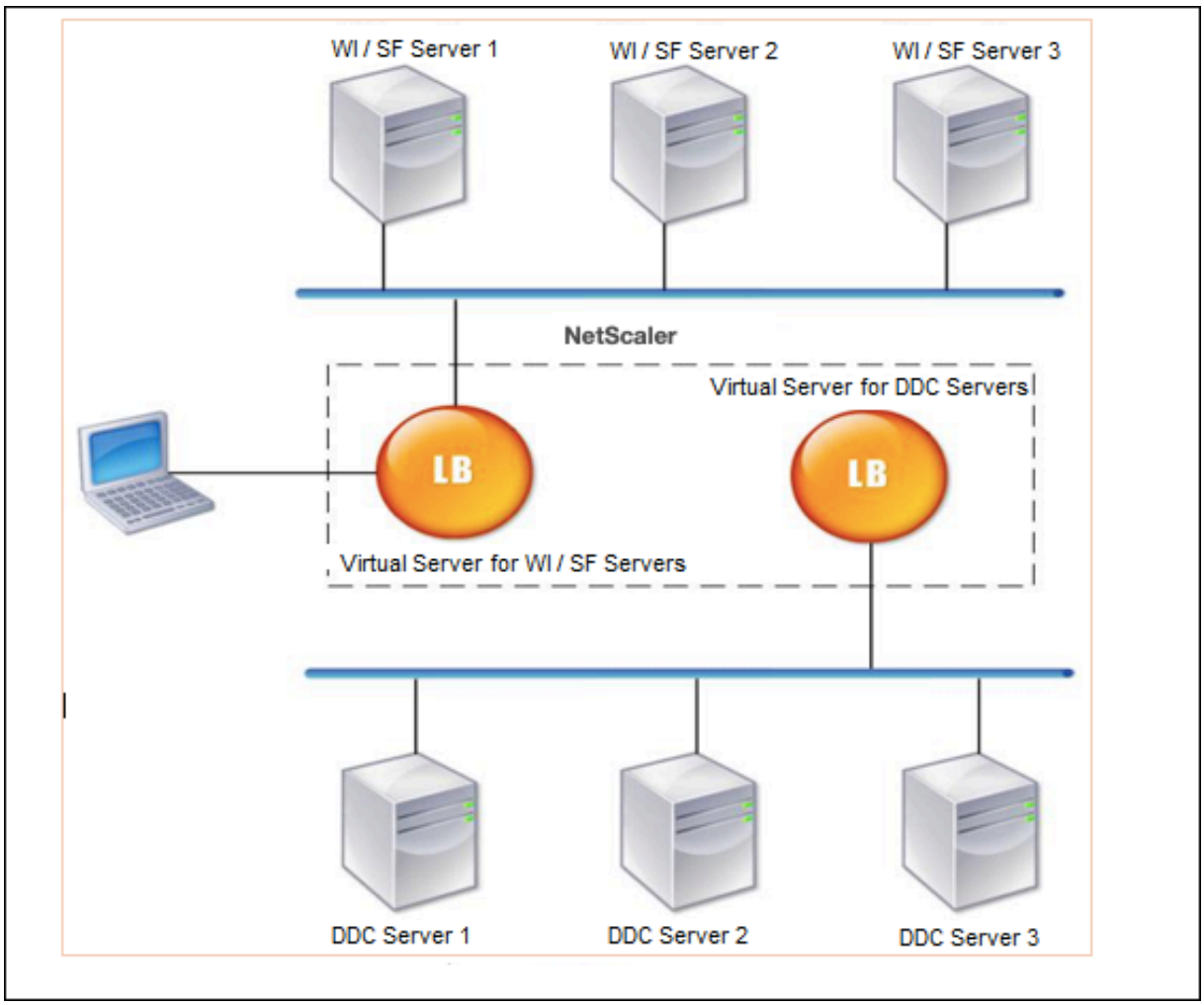
一般に、Citrix Virtual Desktops は、アプリケーションがターミナルサーバーまたは仮想アプリでの実行と互換性がない場合や、各仮想デスクトップに固有の要件がある場合に使用します。このような場合、接続するユーザーごとに 1 つのデスクトップホストが必要です。ただし、ホストはプールできるので、現在接続しているユーザーごとに必要なホストは 1 つだけです。

Citrix Virtual Desktops に導入されるコアアプリケーションサービスは、Desktop Delivery Controller（DDC）です。DDC はサーバーにインストールされ、その主な機能はデスクトップホストを登録し、それらへのクライアント接続を仲介することです。

また、DDC は、デスクトップの状態を制御し、デスクトップを起動および停止することにより、ユーザーを認証し、ユーザーの仮想デスクトップ環境の構成を管理します。

一般的に、可用性を高めるために複数の DDC がインストールされます。

次の図は、Citrix Virtual Desktops と連携する NetScaler ADC アプライアンスのトポロジーを示しています。



注:

HTTP プロトコルを使用することもできますが、クライアントと NetScaler ADC アプライアンス間の通信には SSL を使用することをお勧めします。クライアントとの通信には SSL プロトコルを使用していますが、NetScaler ADC と DDC サーバー間の通信には HTTP プロトコルを使用できます。

GUI を使用して Citrix Virtual Desktops の負荷分散を構成するには

1. サービスを作成します。
   a) [設定] > [トラフィック管理] > [負荷分散] > [サービス] に移動し、[追加] をクリックします。
   b) 名前、IP アドレス、ポート、プロトコルタイプを指定してサービスを作成し、「**OK**」をクリックします。
2. 負荷分散仮想サーバーを作成します。
   a) [設定] > [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、[追加] をクリックします。
   b) 名前、IP アドレス、ポート、プロトコルタイプを指定して仮想サーバーを作成し、「**OK**」をクリックします。

3. サービスを負荷分散仮想サーバーにバインドします。
4. [設定] > [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、サーバーを選択します。

    a) ［編集］をクリックします。
    b)「サービス」と「サービスグループ」で、「****** 」をクリックし、「****** バインドの追加」をクリックします。
    c) バインドするサービスを選択し、重み値を入力します。
    d) ［**Bind**］をクリックします。

コマンドラインインターフェイスを使用して **Citrix Virtual Desktops** の負荷分散を構成するには

- サービスを作成するには、コマンドプロンプトで次のように入力します。

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

例:

```
add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

- 仮想サーバーを作成するには、コマンドプロンプトで次のように入力します。

```
add lb vserver <name> <serviceType> <ip> <port>
<!--NeedCopy-->
```

例:

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

- サービスを負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

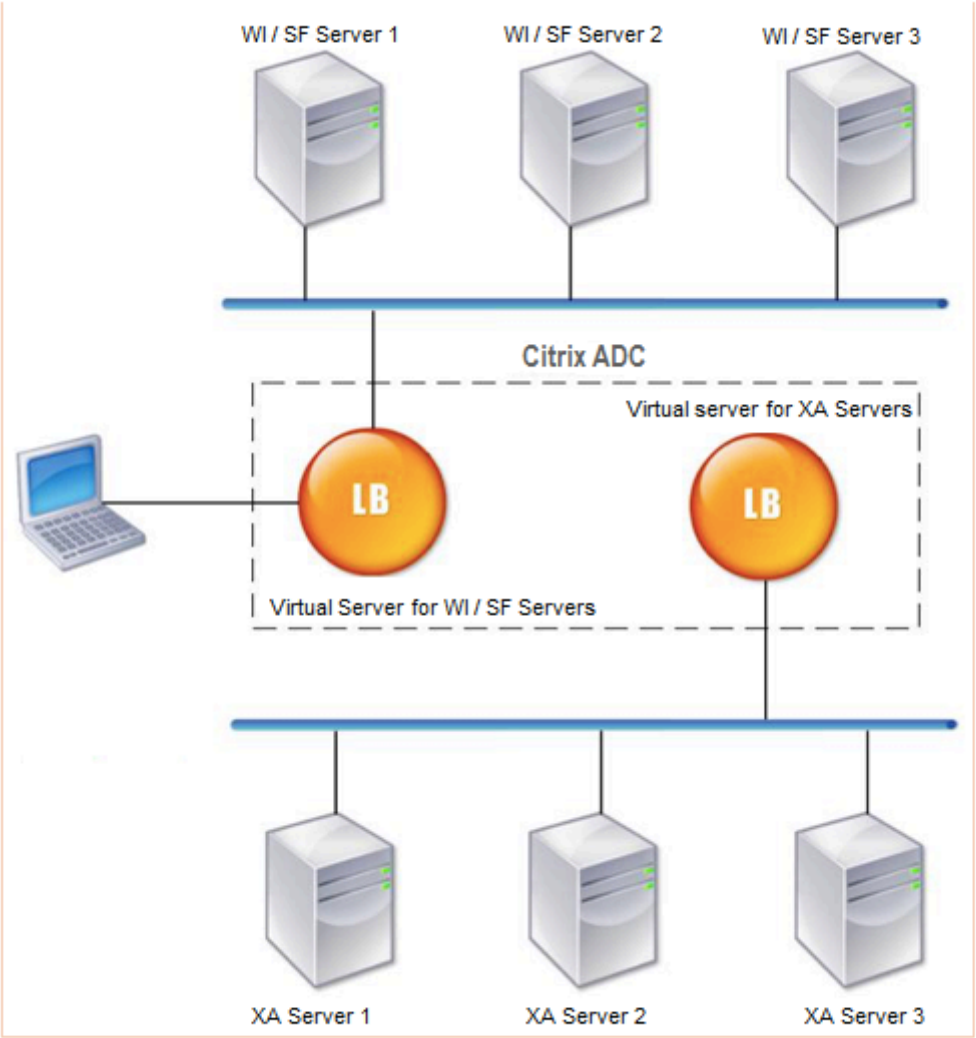
例:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

ユースケース **13:** 負荷分散のための **Citrix Virtual Apps** の構成

December 8, 2023

アプリケーションを効率的に配信するには、NetScaler アプライアンスを Citrix Virtual Apps と統合し、NetScaler 負荷分散機能を使用して Citrix Virtual Apps サーバーファーム全体に負荷を分散できます。次の図は、このような設定のトポロジー図です。



Web Interface サーバーは、ユーザーの Web ブラウザから Citrix Virtual Apps アプリケーションリソースへの安全なアクセスを提供します。Web Interface クライアントは、Citrix Virtual Apps サーバーファームで使用可能なアプリケーション、コンテンツ、デスクトップなどのすべてのリソースをユーザーに提供します。ユーザーは、標準の Web ブラウザまたは Citrix オンラインプラグインを介して公開リソースにアクセスできます。

ユーザーのデバイス上の Web ブラウザーは、Web サーバーに情報を送信します。Web サーバーは、サーバーファーム上のサーバーと通信して、ユーザーがリソースにアクセスできるようにします。

Web インターフェイスと XML ブローカは補完的なサービスです。Web Interface はユーザーにアプリケーションへのアクセスを提供し、XML Broker はユーザーの権限を評価して Web Interface に表示されるアプリケーションを決定します。

XML サービスは、サーバーファーム内のすべてのサーバーにインストールされます。Web インターフェイスで指定された XML サービスは、XML ブローカとして機能します。Web Interface サーバーによって渡されたユーザー資格情報に基づいて、XML Broker サーバーはユーザーがアクセスできるアプリケーションのリストを送信します。

複数の Web Interface サーバーと XML Broker サーバーを展開する大企業では、NetScaler アプライアンスを使用してこれらのサーバーの負荷を分散することをお勧めします。Web Interface サーバーを負荷分散するように 1 つの仮想サーバーを構成し、XML Broker サーバー用に別の仮想サーバーを構成します。負荷分散方式およびその他の機能は、必要に応じて仮想サーバ上で設定できます。

> 注
>
> HTTP プロトコルを使用することもできますが、クライアントと NetScaler 間の通信には SSL を使用することをお勧めします。クライアントとの通信に SSL プロトコルを使用する場合でも、NetScaler と WI サーバー間の通信には HTTP プロトコルを使用できます。

GUI を使用して Citrix Virtual Apps の負荷分散を構成するには

1. サービスを作成します。
   a) [設定] > [トラフィック管理] > [負荷分散] > [サービス] に移動し、[追加] をクリックします。
   b) 名前、IP アドレス、ポート、プロトコルタイプを指定してサービスを作成し、「**OK**」をクリックします。
2. 負荷分散仮想サーバーを作成します。
   a) [設定] > [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、[追加] をクリックします。
   b) 名前、IP アドレス、ポート、プロトコルタイプを指定して仮想サーバーを作成し、「**OK**」をクリックします。
3. サービスを負荷分散仮想サーバーにバインドします。
4. [設定] > [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、サーバーを選択します。
   a) ［編集］をクリックします。
   b)「サービス」と「サービスグループ」で、「**** 」をクリックし、「**** バインドの追加」をクリックします。
   c) バインドするサービスを選択し、重量値を入力します。
   d) ［**Bind**］をクリックします。

コマンドラインインターフェイスを使用して Citrix Virtual Apps の負荷分散を構成するには

- サービスを作成するには、コマンドプロンプトで次のように入力します。

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

例:


```
add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

- 仮想サーバーを作成するには、コマンドプロンプトで次のように入力します。

```
add lb vserver <name> <serviceType> <ip> <port>
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

- サービスを負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

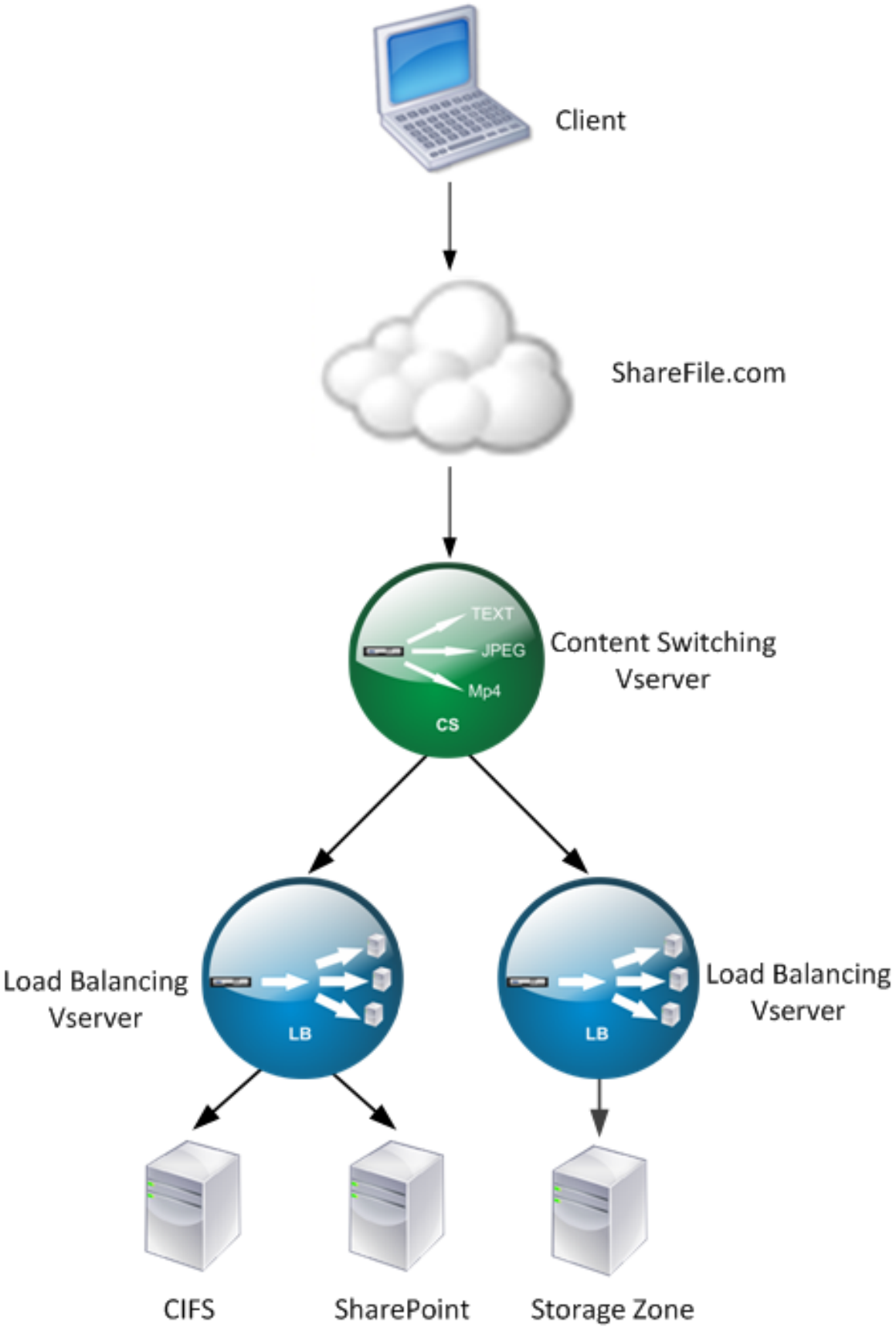
ユースケース 14: Citrix ShareFile の負荷分散のための ShareFile ウィザード

December 8, 2023

ウィザードを使用して Citrix ShareFile の負荷分散を構成できます。Citrix ShareFile ウィザードは、要求されたコンテンツの種類に基づいて ShareFile サイトの負荷分散構成を設定するのに役立ちます。コンテンツスイッチングサーバは、リクエストが StorageZone、CIFS、SharePoint のいずれのリクエストであるかに基づいてリクエストを送信します。コンテンツの切り替えはポリシーに基づいています。ウィザードは、リクエストが StorageZone、CIFS、SharePoint のいずれを対象としているかを識別するポリシーを自動生成します。コンテンツスイッチング仮想サーバーは、これらのポリシーを使用して、要求を適切な負荷分散サーバーに送信します。

一般的なデータフローは、次の図のように表すことができます。

図 1. ShareFile データロードバランシング



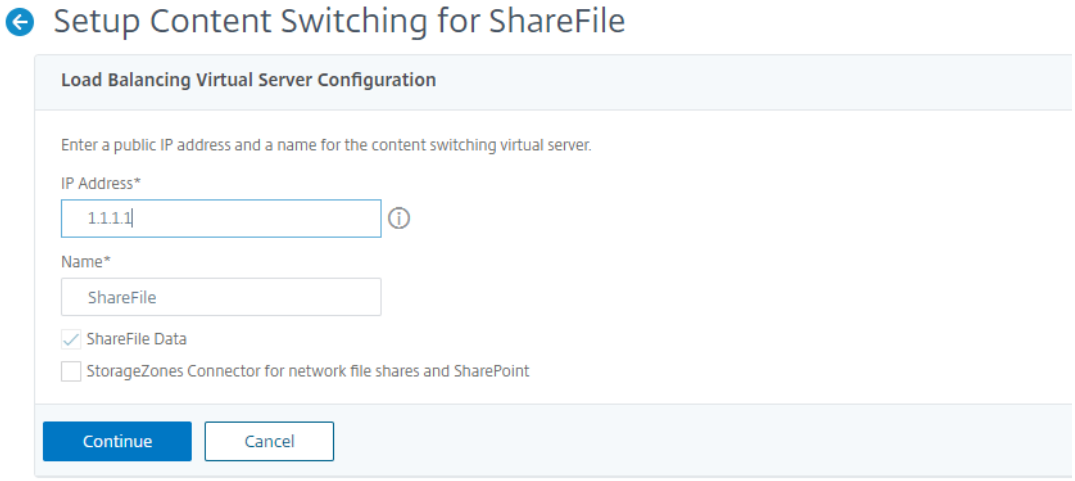
ShareFile ウィザードが作成する負荷分散仮想サーバーを表示するには、[トラフィック管理] **> [** ** 仮想サーバーとサービス] > [仮想サーバー **] に移動します。ShareFile ウィザードを使用して作成した仮想サーバーを手動で削除することはできません。ウィザードを使用して仮想サーバーを削除します。

NetScaler は、シェアポイントまたは CIFS リクエストに LDAP 認証を使用します。ハッシュ認証は、ストレージゾーンのリクエストを認証するために使用されます。

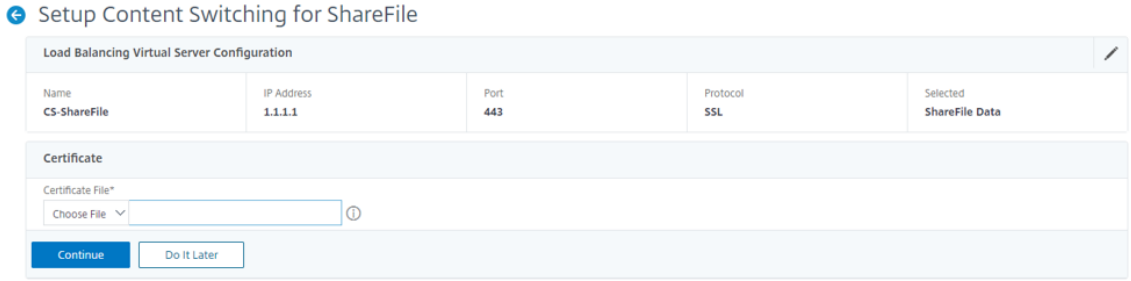
NetScaler アプライアンスを負荷分散するように構成するには、**Citrix ShareFile**

1. ナビゲーションペインで、[トラフィック管理] をクリックします。
2. 「**Citrix ShareFile** 」セクションで、「SShareFile 用 **NetScaler** のセットアップ」をクリックします。
3. **ShareFile** のコンテンツスイッチングの設定ページで、次の情報を入力します。

   - IP アドレス: コンテンツスイッチング仮想サーバーの IP アドレス。
   - 名前: コンテンツスイッチング仮想サーバーの名前。
   - CIFS または SharePoint の負荷分散を設定する場合は 、ネットワークファイル用 の StorageZone コネクタをクリックします Shares/SharePoint チェックボックスをオンにして、[続行] をクリックします。既定では、[**ShareFile** データ] チェックボックスはオンになっています。



4. 有効な証明書を提供してください。証明書をお持ちの場合は、「証明書を選択」をクリックし、ドロップダウンリストから証明書を選択します。証明書をインストールする必要がある場合は、「証明書をインストール」をクリックし、証明書とキーのペアを指定します。



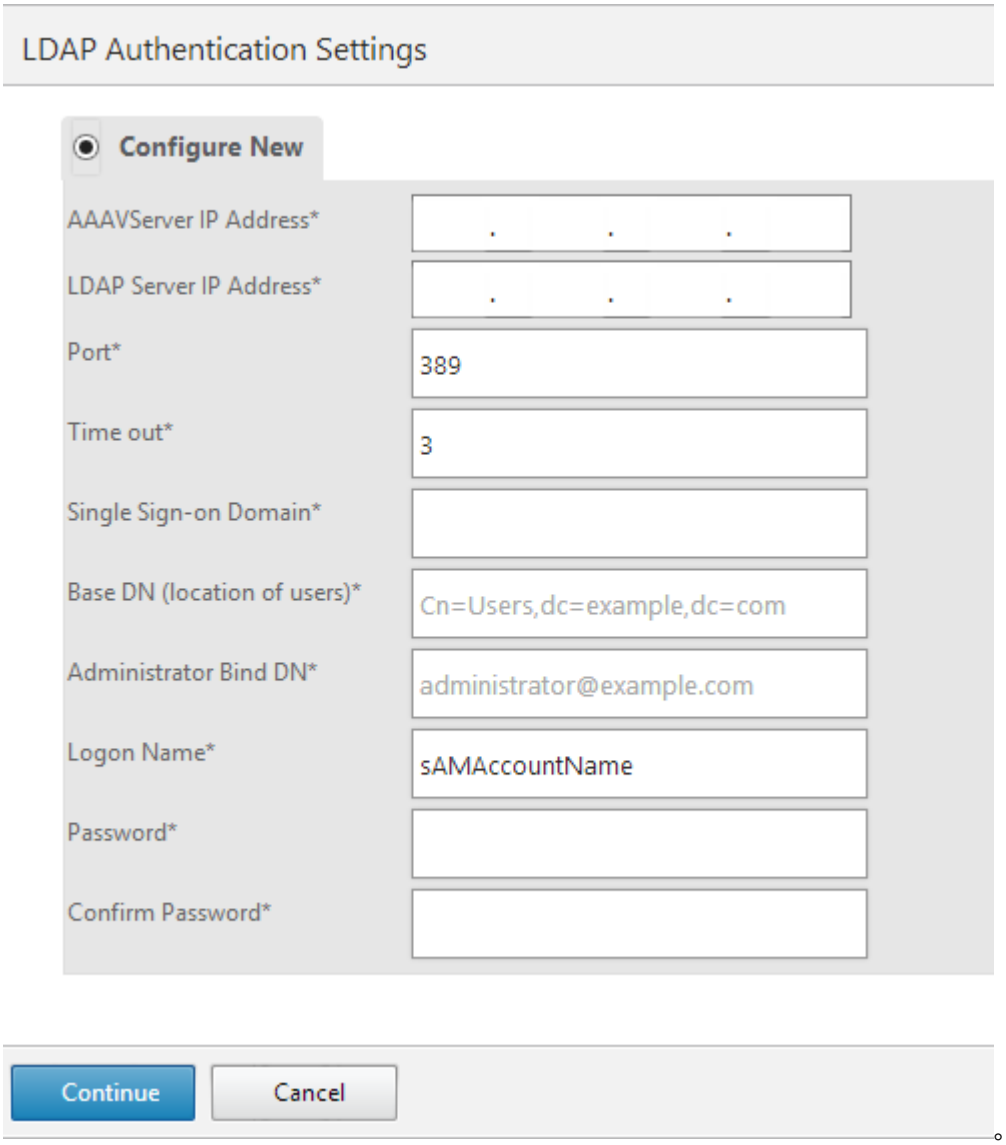
5. [続行] をクリックします。

6.「**StorageZone** コントローラの新規追加」ダイアログ・ボックスで、次のパラメータの値を指定します。

- StorageZone コントローラの IP アドレス—IP アドレス
- ポート：ポート番号。デフォルト値は 443 です。
- プロトコル：HTTPS または HTTP

7. [作成] をクリック してから、[完了] をクリックします。ウィザードは自動的にサービスを作成し、サービスの名前を自動生成します。
8. 手順 4.c で CIFS または SharePoint のロード・バランシングを選択した場合は、LDAP 認証設定の値を指定します。

- NetScaler AAA 仮想サーバーの IP アドレス：NetScaler AAA 仮想サーバーの IP アドレス
- LDAP サーバの IP アドレス：LDAP サーバの IP アドレス
- ポート：ポート番号。デフォルト値は 389 です。
- Timeout：タイムアウト値（分）
- シングルサインオンドメイン-シングルサインオンドメイン名
- ベース DN —ベースドメイン名
- 管理者バインド DN —ドメイン名の LDAP アカウント名 (例:adminstrator@domainname.com)
- ログオン名-ログオン名は sAMAccountName です
- パスワードとパスワードの確認：パスワードを入力し、パスワード



9. [続行] をクリック してから、[完了] をクリックします。

ShareFile の負荷分散構成を削除するには

1. ナビゲーションペインで、[トラフィック管理] をクリックします。
2.「**Citrix ShareFile** 」セクションで、「**ShareFile** 構成の削除」をクリックします。

ユースケース **15**: **NetScaler ADC** アプライアンスでレイヤー **4** の負荷分散を構成する

October 25, 2023

レイヤー 4 ロードバランサ (TCP および UDP ポート) は、ネットワークトランスポート層で提供される情報を使用して、サーバーグループ間でクライアント要求をルーティングします。

クライアントとサーバ間でレイヤ 4 接続が確立されると、クライアントとサーバ間で交換されるトラフィックのパケットビューが表示されます。レイヤー 4 ロードバランサは、TCP ストリームの最初の数個のパケットから抽出されたアドレス情報に基づいてルーティングを決定し、パケットの内容を検査しません。したがって、レイヤ 4 負荷分散は、接続ベースの負荷分散とも呼ばれます。

レイヤー 4 ロードバランサーは、サーバーの正常性を監視します。DOWN の場合、トラフィックはサーバにルーティングされません。

レイヤ 4 負荷分散は、TCP または UDP ペイロードを使用するさまざまなアプリケーションで役立ちます。このようなプロトコルは、TCP ペイロードとしてデータを交換し、従うべき特定の構造を持ちません。

コマンドラインインターフェイスを使用してレイヤ **4** 負荷分散を設定するには

コマンドプロンプトで入力します。

```
add service <name> <serverName> <serviceType> <port>
add lb vserver <name> <serviceType> <ip> <port>
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

```
add service TCPservice 192.0.2.3 TCP 1
add lb vserver TCPserver TCP 192.0.2.4 1
bind lb vserver TCPserver TCPservice
<!--NeedCopy-->
```

GUI を使用してレイヤ **4** 負荷分散を設定するには

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
2. サービスを作成するには、[追加] をクリックします。
3. [サービス名] と [**IP** アドレス] に必要な詳細を指定します。
4. [プロトコル] で [**TCP**] または [**UDP**] を選択します。
5. [**OK**] をクリックします。
6. [完了] をクリックします。

   サービスが作成されます。

   UDP をトランスポート層プロトコルとして使用してサービスを作成すると、ping モニター (組み込みモニター) が自動的にサービスにバインドされます。トランスポート層プロトコルとして TCP を使用してサービスを作成すると、**tcp_default** モニターが自動的にサービスにバインドされます。

負荷分散の設定では、サービスを別のタイプのモニターまたは複数のモニターにバインドできます。高度なモニタリング要件については、**tcp-ecv** モニタを使用して、要求メッセージと応答メッセージを設定できます。

7. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
8. [追加] をクリックして、新しい仮想サーバーを作成します。

   負荷分散が構成されると、仮想サーバーの IP アドレスまたは FQDN を使用して、負荷分散された Web サイト、アプリケーション、またはサーバーに接続できます。
9. **[**** 名前]、[**IP** アドレスタイプ**]**、および **[IP** アドレス**]** に必要な詳細を指定します。**
10. [プロトコル] で [**TCP**] または [**UDP**] を選択します。
11. **[**ポート**]** にポート番号 **(**サービスタイプに基づいて **0** ～**1023)** を入力します。
12. [**OK**] をクリックします。
13. [サービスとサービスグループ**]** で **[**** 負荷分散仮想サーバーサービスバインドなし ******] をクリックします。
14. [サービスバインド] ページで、[サービスの選択 ****]** で **[**クリックして選択] を選択します ******。
15. バインドするサービスを選択し、[選択（**Select**）] をクリックします。
16. [バインド] をクリックして、サービスを仮想サーバーにバインドします。
17. [続行] をクリックします。
18. [完了] をクリックします。

    レイヤ 4 負荷分散仮想サーバの設定が完了しました。

トラブルシューティング

August 15, 2023

構成後に負荷分散が期待どおりに機能しない場合は、いくつかの一般的なツールを使用して NetScaler リソースにアクセスし、問題を診断できます。

負荷分散のトラブルシューティングに関するリソース

最良の結果を得るには、次のリソースを使用して NetScaler アプライアンスのコンテンツスイッチングの問題をトラブルシューティングしてください。

- 最新の ns.conf ファイル
- 関連`newnslog`ファイル
- 可能であれば、アプライアンスおよび関連するクライアントに記録される Ethereal パケットトレース

- ns.log ファイル

上記のリソースに加えて、次のツールを使用するとトラブルシューティングが容易になります。

- HTTP ヘッダーを表示できるブラウザアドオンツール。これは永続性関連の問題のトラブルシューティングに使用できます。
- NetScaler トレースファイル用にカスタマイズされた Wireshark アプリケーション。

負荷分散問題のトラブルシューティング

- 問題

CPU 使用率が、-m MAC オプションが有効になっている仮想サーバーにバインドされているサービスにユーザーモニターがバインドされている場合、100% に達します。

- 解像度

ユーザー以外のモニターをサービスにバインドします。

- 問題

  監視用のユーザースクリプトを作成しましたが、動作しません。

  解像度

  スクリプトの引数の数を確認してください。上限は 512 です。512 個を超える引数を持つスクリプトは正しく動作しない可能性があります。CLI から nsumon-debug.pl スクリプトを使用してスクリプトをデバッグします。

- 問題

  多くのモニタプローブがあり、ネットワークトラフィックが不必要に増加しているようです。モニタープローブを外す方法はありますか?

  解像度

  モニタを無効にするか、set service コマンドの HealthMonitor パラメータの値を NO に設定することで、モニタプローブ接続をオフに設定できます。NO オプションを使用すると、アプライアンスは常に UP としてサービスを表示します。

- 問題

  サービスのモニターをセットアップしましたが、接続はまだダウンしているサーバーに送られます。

  解像度

  モニタープローブの間隔を短くする必要があるかもしれません。NetScaler アプライアンスは、モニターがプローブを送信するまでダウン状態を検出しません。

- 問題

  モニターにバインドされたメトリックは、ローカルメトリックテーブルとカスタムメトリックテーブルに存在します。

  解像度

  メトリックがローカルメトリックテーブルから選択される場合は、メトリック名にローカルプレフィックスを追加します。ただし、指標がカスタムテーブルから選択される場合は、プレフィックスを追加する必要はありません。

- 問題

  サービスへのモニタプローブがサービスに到達していません。

  解像度

  サービスの接続数に制限を設定しているかどうかを確認してください。「はい」の場合は、MonitorSkipMaxClient パラメーターを ENABLED に設定して、モニターとプローブの接続をこの制限から除外してください。

- 問題

  サーバーに ping はできますが、サービスの状態は常に DOWN と表示されます。

  解像度

  設定されているモニターのタイプを確認してください。たとえば、サーバーが SSL 用に構成されておらず、HTTPS モニターを使用する場合、サービスの状態は DOWN としてマークされます。この場合、TCP モニタを使用すると、サービスの状態を UP に変更する必要があります。

- 問題

  負荷モニターの重みを設定しても、サービスの状態を判断するのに役立ちません。

  解像度

  ロードモニターはサービスの状態を判断できません。したがって、ロードモニターに重みを設定することは不適切です。

- 問題

  サービスが安定していません。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - 正しいサーバーがサービスにバインドされていることを確認します。
  - サービスにバインドされているモニターのタイプを確認してください。
  - モニタの障害の原因を確認します。[サービス] ページからサービスを開き、[サービスの構成] ダイアログボックスの [Monitors] タブで、モニタのプローブの数、障害、および最後の応答ステータスの詳細を確認できます。詳細を表示するには、構成されているモニタをクリックします。

- カスタムモニターの場合は、TCP または ping モニターをサービスにバインドし、モニターのステータスを確認します。これで問題が解決した場合は、カスタムモニターに何らかの問題があり、モニターについてさらに調査する必要があります。
- NetScaler アプライアンスにパケットトレースを記録し、監視プローブとサーバーの応答を確認してさらに調査することができます。

- 問題

  仮想 IP (VIP) アドレスが安定していないか、ステータスが DOWN と表示される。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - 負荷分散機能がライセンスされていることを確認します。
  - この機能が有効になっていることを確認します。
  - 適切なサービスが仮想サーバーにバインドされていることを確認します。
  - VIP アドレスのステータスが DOWN と表示される場合は、管理者がサービスを有効にしていることを確認します。そうでない場合、サービスのステータスは Out-of-Service である必要があります。このような場合は、サービスを有効にし、問題が解決されたかどうかを確認する必要があります。
  - 仮想サーバーにバインドされているサービスを確認し、サービスが安定していない問題に関するトラブルシューティング手順を完了します。
  - VIP アドレスが安定していない場合、仮想サーバーにバインドされているすべてのサービスが失敗する必要があります。したがって、すべてのサービスが同時に失敗しているかどうかを確認します。その場合は、NetScaler アプライアンスとサーバー間のネットワークに問題があります。

- 問題

  サイトの負荷分散が不均一です。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - アプライアンスに設定されている負荷分散方法を確認します。
  - サービスに関連付けられている重みが期待どおりであることを確認します。
  - ロードバランシング方式がラウンドロビン以外の場合は、`newnslog`ファイルにログインしているサーバへの接続数を確認します。次のコマンドを実行して、`newnslog`ファイル内の番号を確認できます。

    ```
    # nsconmsg -K <newnslog_file> -s ConLb=2 -d oldconmsg
    ```

    特定の仮想サーバのサービスを確認し、応答時間、オープン確立接続（OE）、要求数、永続要求、および永続レート（P）をチェックして、問題をさらにトラブルシューティングします。

- ロードバランシング方式がラウンドロビンの場合は、前の手順で説明した永続リクエストを確認します。さらに、サービスが安定していないかどうかを確認します。そうでない場合は、「サービスが安定しない」の問題に記載されているトラブルシューティング手順を実行してください
- アプライアンスで永続性が構成されているかどうかを確認します。
- 安定していないサービスがあるかどうかを確認します。「はい」の場合は、「サービスが安定していない」という問題について記載されているトラブルシューティング手順を完了します。

- 問題

  サービスのステータスは DOWN と表示されます。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - SNIP アドレスが設定されているかどうかを確認します。
  - 適切なモニターがサービスにバインドされていることを確認します。
  - カスタムモニターがサービスにバインドされている場合は、TCP モニターまたは ping モニターをサービスにバインドし、モニターのステータスを確認します。これで問題が解決した場合は、カスタムモニターに何らかの問題があり、モニターについてさらに調査する必要があります。
  - 別のサブネットにあるサーバーのサービスのステータスが DOWN と表示されているかどうかを確認します。[はい] の場合は、[サブネット IP] (USNIP) で問題が解決するかどうかを確認します。これは、MIP アドレスがサーバと通信できないことが原因であるためです。

- 問題

  応答時間に問題があります。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - 次のコマンドを実行して、サービス統計からサーバーの応答時間を確認します。

    ```
    # nsconmsg -K <newnslog_file> -s ConLb=2 -d oldconmsg
    ```

  - サービスが安定していないか、サービスのステータスが DOWN の問題として表示されていないか確認してください。

- 問題

  一方のサーバーは、他の負荷分散サーバーよりも多くのリクエストを処理しています。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - 負荷分散方法を確認してください。ラウンドロビン方式を使用して、サーバーの負荷に関係なくクライアント要求を均等に分散します。

  - ロードバランシング設定に対してパーシステンスが有効になっているかどうかを確認します。永続性が有効になっている場合、特に永続セッションが長い場合、特定のサーバーがそのセッションを維持するためにより重い負荷がかかっている可能性があります。
  - 各サービスに重みが割り当てられているかどうかを確認します。適切な重みを割り当てると、適切な荷重分散に役立ちます。

- 問題

  特定の負荷分散サーバーへの接続が停止します。たとえば、1 つの Outlook サーバーへのすべての接続が停止する可能性があります。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - ロードバランシング方法を確認してください。ラウンドロビンの場合は、接続数を最小限に抑える方法に変更することを検討してください。
  - モニターのタイムアウト期間を短縮することを検討してください。タイムアウト期間を短くすると、サービスをより早く DOWN（ダウン）としてマークできます。これにより、機能しているサーバにトラフィックを誘導するのに役立ちます。
  - 接続が長時間停止すると、サージキューが構築される可能性があります。サーバーの負荷が急増するのを避けるために、サージキューをフラッシュすることを検討してください。
  - サーバが最大レベルで動作している場合は、パフォーマンスを向上させるために新しいサーバを追加することを検討してください。

- 問題

  負荷分散のための最小接続方法が設定されている場合でも、接続の大部分は特定のサーバーに向けられます。

  解像度

  パーシスタンスが設定されていて、タイプがソース IP であるかどうかを確認します。接続数が最も少ない方法でも送信元 IP パーシステンスが設定されている場合、要求は特定のサーバーに送信されます。セッション情報を維持するには、サーバーの IP アドレスが必要です。HTTP Cookie ベースの永続性を使用することを検討してください。

- トラブルシューティングのヒントその他の問題については
  、上記に記載されていない問題のトラブルシューティングを行うために、次のヒントを検討してください。

  - 複数の負荷モニターが 1 つのサービスにバインドされている場合、サービスの負荷は、そのサービスにバインドされている負荷モニターのすべての値の合計になります。負荷分散が適切に機能するには、同じモニターセットをすべてのサービスにバインドする必要があります。
  - サービスにバインドされた負荷モニターを無効にし、サービスが仮想サーバーにバインドされている場合、仮想サーバーはロードバランシングにラウンドロビン方式を使用します。
  - 負荷分散方式が CUSTOMLOAD で、サービスのステータスが UP である仮想サーバーにサービスをバインドすると、仮想サーバーは負荷分散に初期ラウンドロビン方式を使用します。サービスにカスタム

ロードモニターがない場合、またはカスタムロードモニターの少なくとも 1 つのステータスが UP でない場合は、引き続きラウンドロビン状態になります。
- 負荷分散方式が CUSTOMLOAD である仮想サーバーにバインドされているすべてのサービスでは、サービスには負荷監視がバインドされている必要があります。
- CUSTOMLOAD ロードバランシング方式は、スタートアップラウンドロビンにも従います。
- メトリックベースのバインディングを無効にし、これが最後のアクティブなメトリックである場合、特定の仮想サーバーはロードバランシングにラウンドロビン方式を使用します。メトリックのしきい値をゼロに設定すると、メトリックは無効になります。
- モニターにバインドされたメトリックがしきい値を超えると、その特定のサービスは負荷分散の対象とは見なされません。すべてのサービスがしきい値に達すると、仮想サーバーはラウンドロビン方式を使用して負荷分散を行い、「5xx-server busy error」というエラーメッセージが表示されます。
- カスタムテーブルから最大 10 個のメトリックスをモニターにバインドできます。
- OID はスカラー変数でなければなりません。
- 負荷分散を正常に行うには、間隔をできるだけ短くする必要があります。間隔が長いと、負荷値を取得する時間が長くなります。その結果、不適切な値を使用して負荷分散が行われます。
- ユーザーはローカルテーブルを変更できません。

負荷分散に関する **FAQ**

August 15, 2023

NetScaler アプライアンスで作成できるさまざまな負荷分散ポリシーにはどのようなものがありますか

NetScaler アプライアンスでは、次の種類の負荷分散ポリシーを作成できます。

- 最小接続数
- ラウンドロビン
- 最短の応答時間
- 最小帯域幅
- 最小パケット数
- URL ハッシュ
- ドメイン名ハッシュ
- 送信元 IP アドレスハッシュ
- 宛先 IP アドレスハッシュ
- 送信元 IP-宛先 IP ハッシュ
- トークン
- LRTM

NetScaler アプライアンスを使用して負荷分散を実装することで、**Web** ファームのセキュリティを実現できますか

はい。NetScaler アプライアンスを使用して負荷分散を実装することで、Web ファームのセキュリティを実現できます。NetScaler アプライアンスでは、負荷分散機能の次のオプションを実装できます。

- IP アドレスの非表示: セキュリティ上の理由と IP アドレスの保護のため、実際のサーバーをプライベート IP アドレス空間にインストールできます。NetScaler アプライアンスはサーバーに代わってリクエストを受け入れるため、このプロセスはエンドユーザーには意識されません。アドレス隠蔽モードでは、アプライアンスは 2 つのネットワークを完全に分離します。したがって、クライアントは、FTP や Telnet サーバーなどのプライベートサブネットで実行されているサービスに、そのサービスのアプライアンス上の別の VIP を介してアクセスできます。
- ポートマッピング: セキュリティ上の理由から、実際の TCP サービスを非標準ポートでホストできるようにします。このプロセスは、NetScaler ADC アプライアンスが標準のアドバタイズ IP アドレスとポート番号でサーバーに代わって要求を受け入れるため、エンドユーザーには透過的です。

NetScaler ADC アプライアンスの負荷分散に使用できるさまざまなデバイスは何ですか

NetScaler ADC アプライアンスを使用して、次のデバイスを負荷分散できます。

- サーバーファーム
- キャッシュまたはリバースプロキシ
- ファイアウォールデバイス
- 侵入検知システム
- SSL オフロードデバイス
- 圧縮デバイス
- コンテンツ検査サーバー

Web サイトの負荷分散機能を実装するのはなぜですか

Web サイトの負荷分散機能を実装すると、次の利点があります。

- 応答時間の短縮: ウェブサイトに負荷分散機能を実装する場合の大きな利点の 1 つは、ロード時間が大幅に短縮されることです。2 つ以上のサーバーが Web トラフィックの負荷を分散しているため、各サーバーのトラフィック負荷は 1 台のサーバーだけの場合よりも少なくなります。つまり、クライアントの要求を満たすために利用できるリソースが増えるということです。これにより、ウェブサイトが速くなります。
- 冗長性: 負荷分散機能を実装すると、多少の冗長性が生じます。たとえば、ウェブサイトが 3 つのサーバー間でバランスが取れていて、そのうちの 1 つがまったく応答しない場合、他の 2 つは実行し続けることができ、ウェブサイトの訪問者もダウンタイムに気付かない。負荷分散ソリューションは、利用できないバックエンドサーバーへのトラフィックの送信を直ちに停止します。

リンクロードバランシング **(LLB)** の **Mac** ベースフォワーディング **(MBF)** オプションを無効にする必要があるのはなぜですか**?**

- MBF オプションを有効にすると、NetScaler ADC アプライアンスは、クライアントからの着信トラフィックと同じクライアントへの発信トラフィックが同じアップストリームルーターを通過すると見なします。ただし、LLB 機能では、リターントラフィックに最適なパスを選択する必要があります。
- MBF オプションを有効にすると、着信クライアントトラフィックを転送したルータを介して発信トラフィックを送信することによって、このトポロジ設計が中断されます。

ネットワーク

August 15, 2023

以下のトピックでは、NetScaler アプライアンスのさまざまなネットワークコンポーネントを構成するための概念的な参考資料と手順について説明します。

IP アドレス	NetScaler が所有するさまざまなタイプの IP アドレスと、それらを作成、カスタマイズ、削除する方法について説明します。
インターフェイス	利用開始時に実行する必要がある基本的なネットワーク構成のいくつかを構成します。
アクセス制御リスト（ACL）	さまざまな種類のアクセス制御リストと、それらを作成、カスタマイズ、および削除する方法を構成します。
IP ルーティング	NetScaler アプライアンスの静的ルーティング機能と動的ルーティング機能を学習して構成します。
インターネットプロトコルバージョン 6（IPv6）	NetScaler アプライアンスが IPv6 をどのようにサポートしているかをご覧ください。
トラフィックドメイン	さまざまなアプリケーションのネットワークトラフィックをセグメント化するために、トラフィックドメインを学習および構成します。
VXLAN	データセンターにおけるスケーラビリティのニーズを満たすように、VXLAN を学習および構成します。

IP アドレス

August 15, 2023

NetScaler アプライアンスを構成する前に、管理 IP アドレスとも呼ばれる NSIP アドレスを割り当てる必要があります。NetScaler が所有する他の IP アドレスを作成して、サーバーを抽象化し、サーバーとの接続を確立することもできます。このタイプの構成では、アプライアンスは抽象化されたサーバーのプロキシとして機能します。ネットワークアドレス変換 (INAT と RNAT) を使用して接続をプロキシすることもできます。接続をプロキシする場合、アプライアンスはブリッジ（レイヤー 2）デバイスまたはパケット転送（レイヤー 3）デバイスとして動作できます。パケット転送をより効率的にするために、スタティック ARP エントリを設定できます。IPv6 では、ネイバーディスカバリ (ND) を設定できます。

NetScaler ADC 所有の IP アドレスの構成

August 15, 2023

NetScaler が所有する IP アドレス（NSIP アドレス、仮想 IP アドレス（VIP)、サブネット IP アドレス（SNIP)、およびグローバルサーバー負荷分散サイト IP アドレス（GSLBIP)）は、NetScaler アプライアンスにのみ存在します。NSIP はネットワーク上の NetScaler を一意に識別し、アプライアンスへのアクセスを提供します。VIP は、クライアントがリクエストを送信するパブリック IP アドレスです。NetScaler は VIP でクライアント接続を終了し、サーバーとの接続を開始します。この新しい接続では、サーバーに転送されるパケットの送信元 IP アドレスとして SNIP または MIP を使用します。地理的に分散した複数のデータセンターがある場合、各データセンターは一意の GSLBIP で識別できます。管理アプリケーションへのアクセスを提供するために、NetScaler ADC が所有する IP アドレスを構成できます。

NSIP アドレスの構成

August 15, 2023

NSIP アドレスは、管理目的で NetScaler アプライアンスにアクセスする IP アドレスです。アプライアンスには、管理 IP アドレスとも呼ばれる NSIP を 1 つだけ設定できます。NetScaler を初めて構成するときは、この IP アドレスを追加する必要があります。NSIP アドレスは削除できません。セキュリティ上の理由から、NSIP は組織の LAN 上のルーティング不可能な IP アドレスにする必要があります。

このアドレスを変更する場合は、NetScaler アプライアンスを再起動する必要があります。新しい NSIP アドレスのサブネットアドレスが前のものと異なる場合は、LAN 上の他のネットワークから新しい NSIP アドレスにアクセスできるように、このサブネットにデフォルトルートを追加する必要があります。

> 重要
>
> NSIP アドレスの設定は必須です。

NetScaler アプライアンスの NSIP アドレスの変更は、次のタスクで構成されます。

- NSIP アドレスを変更します。
- NSIP アドレスのサブネットアドレスにデフォルトルートがない場合は、そのルートを追加します。
- 構成を保存します。
- アプライアンスを再起動します。

コマンドラインプロシージャ

CLI を使用して NSIP アドレスを変更するには:

コマンドプロンプトで入力します。

- **set ns config -IPAddress** <ip_addr> **-netmask** <netmask>
- **show ns config**

CLI を使用してデフォルトルートを追加するには:

コマンドプロンプトで入力します。

- **add route 0 0** <gateway IP address>
- ルートを表示

CLI を使用して設定を保存するには:

コマンドプロンプトで入力します。

- 設定を保存

CLI を使用して NetScaler アプライアンスを再起動するには:

コマンドプロンプトで入力します。

- **reboot**

GUI 手順

GUI を使用して NSIP アドレスを設定するには:

1. 設定ページの右上隅にある歯車アイコンをクリックします。
2. **NSIP** アドレスペインをクリックします。
3. **NSIP** アドレスページで、次のパラメータを設定し、「完了」をクリックします。

- NSIP アドレス
- ネットマスク

GUI を使用してデフォルトルートを追加するには:

[システム] > [ネットワーク] > [ルート] に移動し、[基本] タブで次のパラメータ設定を含むデフォルトルートを追加し、[作成] をクリックします。

- ネットワーク (0 に設定)
- ネットマスク (0 に設定)
- ゲートウェイ (ゲートウェイの IP アドレス)

GUI を使用して NetScaler を再起動するには:

1. システムノードの「システム情報」タブページで、「再起動」をクリックします。
2. 再起動を促すメッセージが表示されたら、[設定を保存] を選択して、構成が失われないようにします。

設定例

次の例では、NetScaler アプライアンスの NSIP アドレスが 192.0.2.90 に変更されています。このアドレスは、以前の NSIP アドレスとはサブネットアドレス（192.0.2.0/24）が異なります。したがって、新しい NSIP アドレスが他のネットワークから到達可能になるように、このサブネットにデフォルトルートが追加されます。

```
> set nsconfig -ipAddress 192.0.2.90 -netmask 255.255.255.0

 Warning: The configuration must be saved and the system rebooted for
    these settings to take effect
> add route 0 0 192.0.2.1

 Warning: The configuration must be saved and the system rebooted for
    these settings to take effect
> save config

 Done
> reboot
```

仮想 IP（VIP）アドレスの構成と管理

August 15, 2023

NetScaler の初期構成では、仮想サーバーの IP（VIP）アドレスの構成は必須ではありません。ロードバランシングを構成するときは、VIP アドレスを仮想サーバーに割り当てます。

ロードバランシングの設定の詳細については、「負荷分散」を参照してください。

場合によっては、VIP 属性をカスタマイズするか、VIP アドレスを有効または無効にする必要があります。通常、VIP アドレスは仮想サーバーに関連付けられ、一部の VIP 属性は仮想サーバーの要件に合わせてカスタマイズされます。ARP と ICMP の属性を使用して、同じブロードキャストドメインにある複数の NetScaler アプライアンスで同じ仮想サーバーをホストできます。VIP（または任意の IP アドレス）を追加すると、アプライアンスは ARP 要求を送信し、応答します。VIP は、無効化できる唯一の NetScaler 所有 IP アドレスです。VIP アドレスが無効になると、そのアドレスを使用する仮想サーバーがダウンし、ARP、ICMP、または L4 サービスリクエストに応答しなくなります。VIP アドレスを 1 つずつ作成する代わりに、連続した VIP アドレスの範囲を指定できます。

CLI を使用して VIP アドレスを作成するには

コマンドプロンプトで入力します。

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

例:

```
> add ns ip 10.102.29.59 255.255.255.0 -type VIP
 Done
<!--NeedCopy-->
```

CLI を使用して VIP アドレスの範囲を作成するには:

コマンドプロンプトで入力します。

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

例:

```
> add ns ip 10.102.29.[60-64] 255.255.255.0 -type VIP
ip "10.102.29.60" added
ip "10.102.29.61" added
ip "10.102.29.62" added
ip "10.102.29.63" added
ip "10.102.29.64" added
 Done
<!--NeedCopy-->
```

CLI を使用して IPv4 VIP アドレスを有効または無効にするには:

コマンドプロンプトで、次のいずれかのコマンドセットを入力して VIP を有効または無効にし、構成を確認します。

- enable ns ip <IPAddress>
- show ns ip <IPAddress>
- disable ns ip <IPAddress>
- show ns ip <IPAddress>

例:


```
> enable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Enabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled
 Done
> disable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Disabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled

 Done
<!--NeedCopy-->
```

GUI を使用して VIP アドレスを設定するには:

[** システム] > [ネットワーク] > [IP] > [**IPv4] に移動し、新しい IP アドレスを追加するか、既存のアドレスを編集します。**

GUI を使用して VIP アドレスの範囲を作成するには:

1. **System** > **Network** > **IPs** > **IPV4s** に移動します。

2. アクションリストで、「範囲を追加」を選択します。

GUI を使用して VIP アドレスを有効または無効にするには:

1. **System** > **Network** > **IPs** > **IPV4s** に移動します。
2. 次のいずれかを行います:
   - VIP アドレスを選択します。
   - **Ctrl** キーを押したまま、複数のサーバーアドレスエントリを選択します。
   - **Shift** キーを押したまま、サーバーアドレスエントリの範囲を選択します。
   - ヘッダー行の左側にあるチェックボックスを選択して、すべてのアドレスを選択します。
3. アクションリストから、「無効化」または「有効化 **」を選択します。

TTL アップデートによる **UDP** 負荷分散セットアップでの **NetScaler** アプライアンスの検出

次の表は、NetScaler アプライアンスがさまざまな機能で受信パケットの TTL 値をどのように処理するかを示しています。

機能	TTL 値
仮想サーバー	リクエストをバックエンドサーバーに転送する場合、TTL は 255 に設定されます。応答をクライアントに転送すると、TTL が 1 ずつ減ります。
L2 モード	TTL は変更されません。
L3 モード	TTL は 255 に設定されています。
INAT	リクエストをバックエンドサーバーに転送する際、TTL は 255 に設定されます。応答をクライアントに転送すると、TTL が 1 ずつ減ります。

監視アプリケーションを実行している企業やシナリオによっては、負荷分散設定の NetScaler アプライアンスをトレースルートのホップの 1 つとして検出する必要があります。負荷分散セットアップの NetScaler アプライアンスは、リクエストをバックエンドサーバーに転送するときに、デフォルトで TTL 値を減らさずに 255 に設定するため、トレースルートで検出されません。

この要件を満たすには、**VIP** アドレスの **Decrement TTL** パラメータを使用できます。このパラメータは、この VIP を使用するすべての UDP 仮想サーバーに適用されます。

VIP の **Decrement TTL** パラメーターを有効にすると、NetScaler アプライアンスは、この VIP を使用する UDP 仮想サーバーで受信したリクエストを転送するときに、TTL 値を 255 に設定するのではなく、1 ずつ減らします。

traceroute データを使用してアプリケーションを監視しているときに、UDP 負荷分散セットアップの NetScaler アプライアンスの存在を検出できるようになりました。

始める前に

負荷分散設定のトレースルートで検出されるように NetScaler アプライアンスを構成する前に、次の点に注意してください。

- Decrement TTL パラメータは UDP 負荷分散仮想サーバーでのみサポートされます。
- デクリメント TTL パラメータは、IPv4 VIP アドレスと IPv6 VIP (VIP6) アドレスでサポートされています。
- Decrement TTL パラメーターは、スタンドアロンの NetScaler アプライアンスだけでなく、高可用性（HA）やクラスターのセットアップでもサポートされています。

構成の手順

UDP 負荷分散設定の traceroute で検出されるように NetScaler アプライアンスを構成するには、次のタスクで構成されます。

- UDP 負荷分散構成の作成
- VIP アドレスのデクリメント TTL パラメータを有効にする

CLI 手順

CLI を使用して VIP アドレスのデクリメント TTL オプションを有効にするには:

- VIP アドレスを追加するときに VIP アドレスの Drecrement TTL オプションを有効にするには、コマンドプロンプトで次のように入力します。
  - **add ns ip** <ip> <mask> -**type VIP** -**decrementTTL ENABLED**
  - **show ns ip** <VIP address>
- 既存の VIP アドレスの Decrement TTL オプションを有効にするには、コマンドプロンプトで次のように入力します。
  - **set ns ip** <ip> <mask> -**decrementTTL ENABLED**
  - **show ns ip** <VIP address>

CLI を使用して VIP6 アドレスのデクリメント TTL オプションを有効にするには:

- VIP6 アドレスを追加するときに VIP6 アドレスの Drecrement TTL オプションを有効にするには、コマンドプロンプトで次のように入力します。
  - **add ns ip6** <IP6/prefix> <mask> -**type VIP** -**decrementTTL ENABLED**
  - **show ns ip6** <VIP6/prefix>
- 既存の VIP6 アドレスのデクリメント TTL オプションを有効にするには、コマンドプロンプトで次のように入力します。

- **set ns ip6** <ip6/prefix> <mask> -**decrementTTL ENABLED**
- **show ns ip6** <VIP6 address>

```
    > add ns ip 203.0.113.30 -type VIP -decrementTTL ENABLED
     Done

    > add ns ip6  2001:DB8:5001::30 -type VIP -decrementTTL ENABLED
     Done
<!--NeedCopy-->
```

GUI 手順

GUI を使用して VIP アドレスのデクリメント TTL オプションを有効にするには:

[** システム] > [ネットワーク] > [IP] > [**IPv4]** に移動し、新しい **VIP** アドレスを追加するか、既存のアドレスを編集するときに **Decrement TTL** パラメータを有効にします。**

GUI を使用して VIP6 アドレスのデクリメント TTL オプションを有効にするには:

[** システム] > [ネットワーク] > [IP] > [**IPv6]** に移動し、新しい **VIP6** アドレスを追加するか、既存のアドレスを編集するときに **Decrement TTL** パラメータを有効にします。**

仮想 **IP** アドレス（**VIP**）の **ARP** 抑制の構成

August 15, 2023

NetScaler アプライアンスは、その VIP に関連付けられている仮想サーバーの状態に基づいて、仮想 IP（VIP）アドレスに対する ARP 要求に応答するか、応答しないかを構成できます。

たとえば、HTTP タイプの仮想サーバー V1 と HTTPS タイプの V2 が NetScaler アプライアンスの VIP アドレス 10.102.29.45 を共有している場合、V1 と V2 の両方がダウン状態の場合、VIP10.102.29.45 の ARP 要求に応答しないようにアプライアンスを構成できます。

仮想 IP アドレスの ARP 応答抑制を設定するには、次の 3 つのオプションを使用できます。

- なし。NetScaler アプライアンスは、そのアドレスに関連付けられている仮想サーバーの状態に関係なく、VIP アドレスを求めるすべての ARP 要求に応答します。
- **1** つの仮想サーバー。NetScaler アプライアンスは、関連する仮想サーバーの少なくとも 1 つが稼働状態の場合、VIP アドレスの ARP 要求に応答します。
- すべての仮想サーバー。NetScaler アプライアンスは、関連するすべての仮想サーバーが稼働状態の場合、VIP アドレスの ARP 要求に応答します。

次の表は、2 つの仮想サーバーで構成された VIP での NetScaler アプライアンスの動作例を示しています。

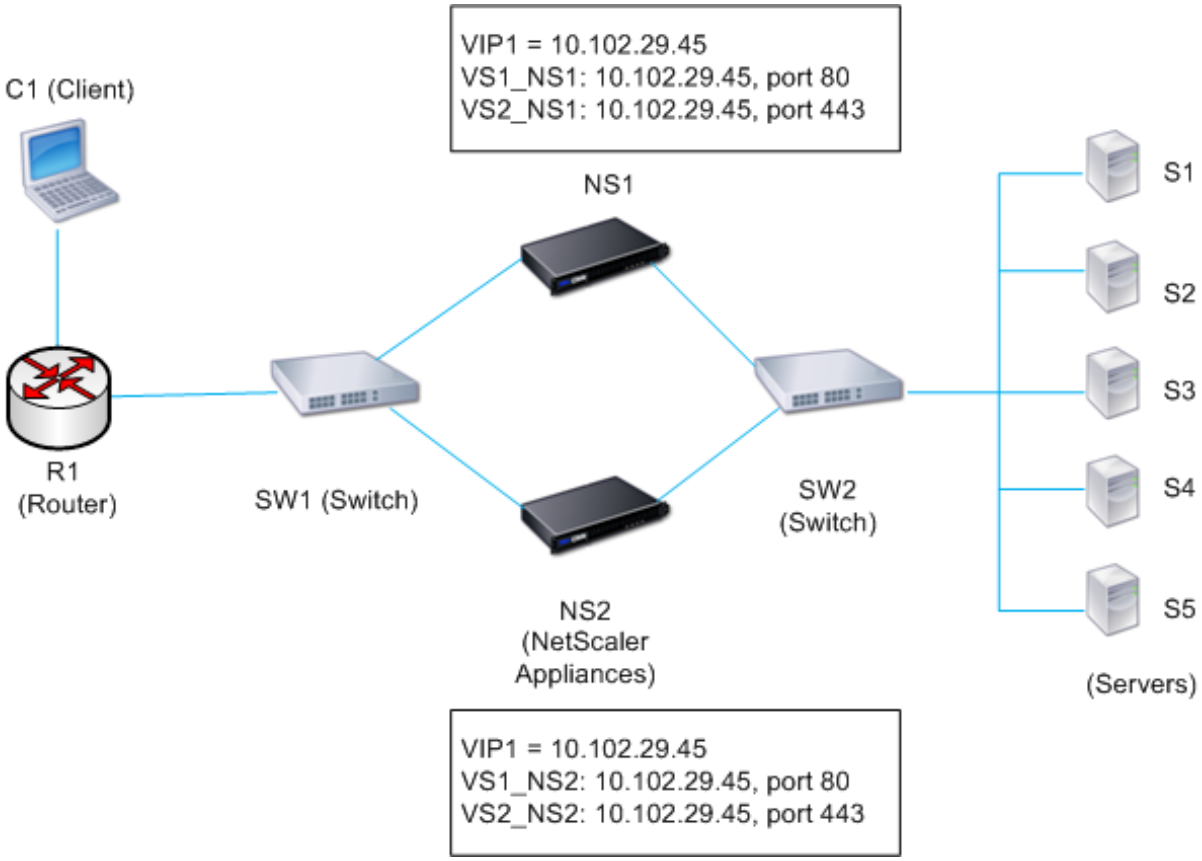
VIP に関連する仮想サーバー	ステート 1	ステート 2	ステート 3	ステート 4
NONE				
V1	上へ	上へ	DOWN	DOWN
V2	上へ	DOWN	上へ	DOWN
この VIP の ARP リクエストに応答しますか?	はい	はい	はい	はい
ONE VSERVER				
V1	上へ	上へ	DOWN	DOWN
V2	上へ	DOWN	上へ	DOWN
この VIP の ARP リクエストに応答しますか?	はい	はい	はい	いいえ
ALL VSERVER				
V1	上へ	上へ	DOWN	DOWN
V2	上へ	DOWN	上へ	DOWN
この VIP の ARP リクエストに応答しますか?	はい	いいえ	いいえ	いいえ

V1 と V2 の 2 つの仮想サーバーのパフォーマンスをテストする例を考えてみましょう。これらの仮想サーバーの VIP アドレスは同じですが、タイプが異なり、それぞれ NetScaler アプライアンス NS1 と NS2 で構成されています。共有 *VIP* アドレスを *VIP1* と呼びましょう。

V1 は、サーバー S1、S2、および S3 のロードバランシングを行います。V2 はサーバー S4 と S5 のロードバランシングを行います。

NS1 と NS2 の両方で、VIP1 の場合、ARP 抑制パラメータは ALL_VSERVER に設定されています。NS1 の V1 と V2 のパフォーマンスをテストする場合は、NS2 が VIP1 の ARP 要求に応答しないように、NS2 の V1 と V2 を手動で無効にする必要があります。

図 1:



実行フローは次のとおりです。

1. クライアント C1 は V1 に要求を送信します。リクエストは R1 に達します。
2. R1 には V1 の IP アドレス（VIP1）の APR エントリがないため、R1 は VIP1 の ARP 要求をブロードキャストします。
3. NS1 は、送信元 MAC アドレス MAC1 と送信元 IP アドレス VIP1 で応答します。NS2 は ARP 要求に応答しません。
4. SW1 は ARP 応答から VIP1 のポートを学習してブリッジテーブルを更新し、R1 は ARP エントリを MAC1 と VIP1 で更新します。
5. R1 は NS1 のアドレス VIP1 にパケットを転送します。
6. NS1 の負荷分散アルゴリズムがサーバー S2 を選択し、NS1 は SNIP アドレスの 1 つと S2 間の接続を開きます。S2 がクライアントに応答を送信すると、応答は同じパスで返されます。
7. 次に、NS2 で V1 と V2 のパフォーマンスをテストします。そこで、NS2 で V1 と V2 を有効にし、NS1 で無効にします。NS2 は VIP1 の ARP メッセージをブロードキャストするようになりました。メッセージでは、MAC2 は送信元 MAC アドレスで、VIP1 は送信元 IP アドレスです。
8. SW1 は ARP ブロードキャストから MAC2 に到達するためのポート番号を学習し、ブリッジテーブルを更新して以降の VIP1 へのクライアント要求を NS2 に送信します。R1 は ARP テーブルを更新します。
9. ここで、R1 の ARP テーブルで VIP1 の ARP エントリがタイムアウトし、クライアント C1 が V1 への要求を送信したとします。R1 には VIP1 の APR エントリがないため、VIP1 の ARP 要求をブロードキャストします。
10. NS2 は、送信元 MAC アドレスと VIP1 を送信元 IP アドレスとして応答します。NS1 は ARP 要求に応答しま

せん。

CLI を使用して ARP 応答抑制を設定するには:

コマンドプロンプトで入力します。

- **set ns ip** -**arpResponse** <arpResponse>]
- **sh ns ip** <IPAddress>

例**:**

```
1 > set ns ip 10.102.29.96 -arpResponse ALL_VSERVERS
2  Done
3 <!--NeedCopy-->
```

GUI を使用して ARP 応答抑制を設定するには:

1. **System** > **Network** > **IPs** > **IPV4s** に移動します。
2. IP アドレスエントリを開き、ARP レスポンスのタイプを選択します。

サブネット **IP** アドレス（**SNIP**）の構成

August 15, 2023

サブネット IP アドレス（SNIP）は、NetScaler がサーバーと通信するために使用する NetScaler 所有の IP アドレスです。

NetScaler は、サブネット IP アドレスをソース IP アドレスとして使用して、クライアント接続をサーバーにプロキシします。また、動的ルーティングプロトコルに関連するパケットなどの独自のパケットを生成する場合や、監視プローブを送信してサーバーの状態を確認する場合にもサブネット IP アドレスを使用します。ネットワークトポロジによっては、さまざまなシナリオで 1 つ以上の SNIP を設定する必要がある場合があります。

NetScaler で SNIP アドレスを構成するには、SNIP アドレスを追加し、グローバルサブネット IP（USNIP）使用モードを有効にします。SNIP を 1 つずつ作成する代わりに、SNIP の範囲を連続して指定することもできます。

CLI を使用して SNIP アドレスを設定するには:

コマンドプロンプトで入力します。

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

例**:**

```
1 > add ns ip 10.102.29.203 255.255.255.0 -type SNIP
2  Done
3 <!--NeedCopy-->
```


CLI を使用してさまざまな SNIP アドレスを作成するには:

コマンドプロンプトで入力します。

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

例:

```
> add ns ip 10.102.29.[205-209] 255.255.255.0 -type SNIP
ip "10.102.29.205" added
ip "10.102.29.206" added
ip "10.102.29.207" added
ip "10.102.29.208" added
ip "10.102.29.209" added
 Done
<!--NeedCopy-->
```

CLI を使用して USNIP モードを有効または無効にするには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- enable ns modeUSNIP
- disable ns modeUSNIP

GUI を使用して SNIP アドレスを設定するには:

[システム] > [ネットワーク] > [IP] > [IPv4] に移動し、新しい SNIP アドレスを追加するか、既存のアドレスを編集します。

GUI を使用してさまざまな SNIP アドレスを作成するには:

1. [システム] > [ネットワーク] > [IP] > [IPv4] に移動します。
2. アクションリストで、「範囲を追加」を選択します。

CLI を使用して USNIP モードを有効または無効にするには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- enable ns mode USNIP
- disable ns mode USNIP

GUI を使用して USNIP モードを有効または無効にするには:

1. [システム] > [設定] に移動し、[モードと機能] グループで [モードの変更] をクリックします。
2. 「サブネット IP を使用」オプションを選択または選択解除します。

直接接続されたサーバーサブネットでの **SNIP** の使用

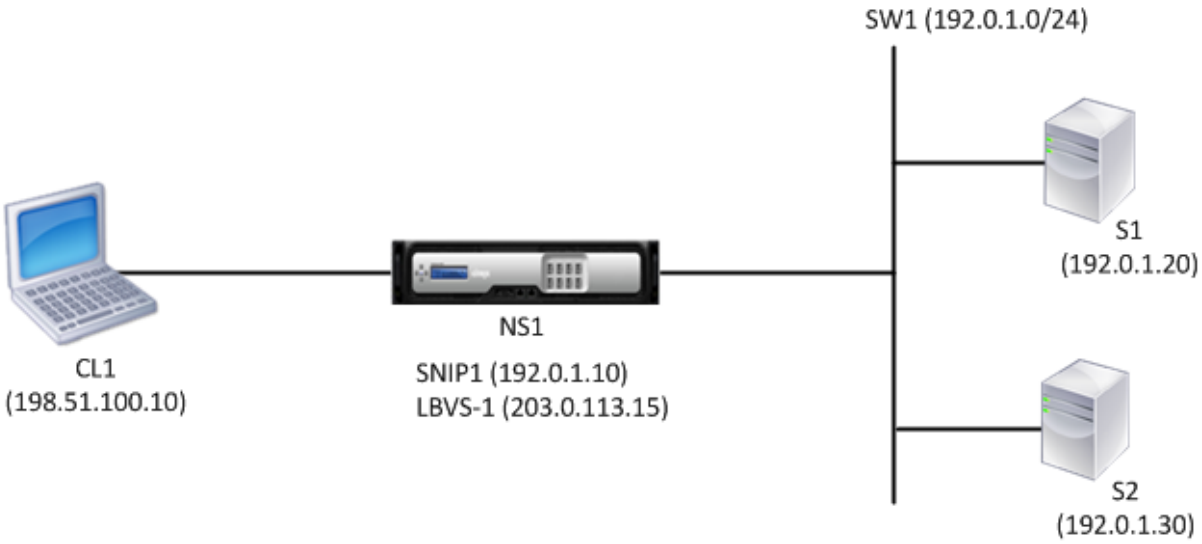
NetScaler と NetScaler に直接接続されているサーバーまたは L2 スイッチのみを介して接続されているサーバー間の通信を有効にするには、サーバーのサブネットに属するサブネット IP アドレスを構成する必要があります。NSIP を介して接続された直接接続管理サブネットを除き、直接接続されたサブネットごとに少なくとも 1 つのサブネット IP アドレスを設定する必要があります。

NetScaler NS1 の負荷分散仮想サーバー LBVS1 を使用して、L2 スイッチ SW1 を介して NS1 に接続されているサーバー S1 と S2 の負荷分散を行う負荷分散設定の例を考えてみましょう。S1 と S2 は同じサブネットに属します。

SNIP アドレス SNIP1 は、S1 および S2 と同じサブネットに属し、NS1 で設定されます。SNIP1 が設定されるとすぐに、NS1 は SNIP1 の ARP パケットをブロードキャストします。

NS1 上のサービス SVC-S1 と SVC-S2 は S1 と S2 を表します。これらのサービスが設定されるとすぐに、NS1 は S1 と S2 に ARP 要求をブロードキャストして、IP と MAC のマッピングを解決します。S1 と S2 が応答した後、NS1 はアドレス SNIP1 から定期的に監視プローブを送信し、正常性をチェックします。

NetScaler ADC での負荷分散の構成の詳細については、「負荷分散」を参照してください。



次に、この例のトラフィックフローを示します。

1. クライアント C1 は LBVS-1 に要求パケットを送信します。リクエストパケットには次のものが含まれます。
   - ソース IP = クライアントの IP アドレス (198.51.100.10)
   - 宛先 IP = LBVS-1 の IP アドレス (203.0.113.15)
2. NS1 の LBVS1 は要求パケットを受信します。
3. LBVS1 の負荷分散アルゴリズムはサーバー S2 を選択します。
4. S2 は NS1 に直接接続されており、SNIP1（192.0.1.10）は S2 と同じサブネットに属する唯一の IP アドレスであるため、NS1 は SNIP1 と S2 間の接続を開きます。
5. NS1 は SNIP1 から S2 にリクエストパケットを送信します。リクエストパケットには次のものが含まれます。

- ソース IP = SNIP1 (192.0.1.10)
- デスティネーション IP = S2 の IP アドレス (192.0.1.30)

6. S2 の応答は同じパスで返されます。

ルーター経由で接続されたサーバーサブネットに **SNIP** を使用する

NetScaler とルーターを介して接続されたサブネット内のサーバー間の通信を可能にするには、ルーターに直接接続されたインターフェイスのサブネットに属するサブネット IP アドレスを少なくとも 1 つ構成する必要があります。ADC は、このサブネット IP アドレスを使用して、ルーター経由でアクセスできるサブネット内のサーバーと通信します。

NetScaler NS1 の負荷分散仮想サーバー LBVS1 を使用して、ルーター R1 を介して NS1 に接続されているサーバー S1、S2、S3、S4 の負荷分散を行う負荷分散設定の例を考えてみましょう。

S1 と S2 は同じサブネット 192.0.2.0/24 に属し、L2 スイッチ SW1 を介して R1 に接続されています。S3 と S4 は別のサブネット（192.0.3.0/24）に属し、L2 スイッチ SW2 を介して R1 に接続されています。

NetScaler NS1 は、サブネット 192.0.1.0/24 を介してルーター R1 に接続されています。SNIP1 は、ルータに直接接続されたインターフェイス（192.0.1.0/24）と同じサブネットに属し、NS1 に設定されています。NS1 は、このアドレスを使用して、サーバ S1 および S2 と通信し、サーバ S3 および S4 と通信します。

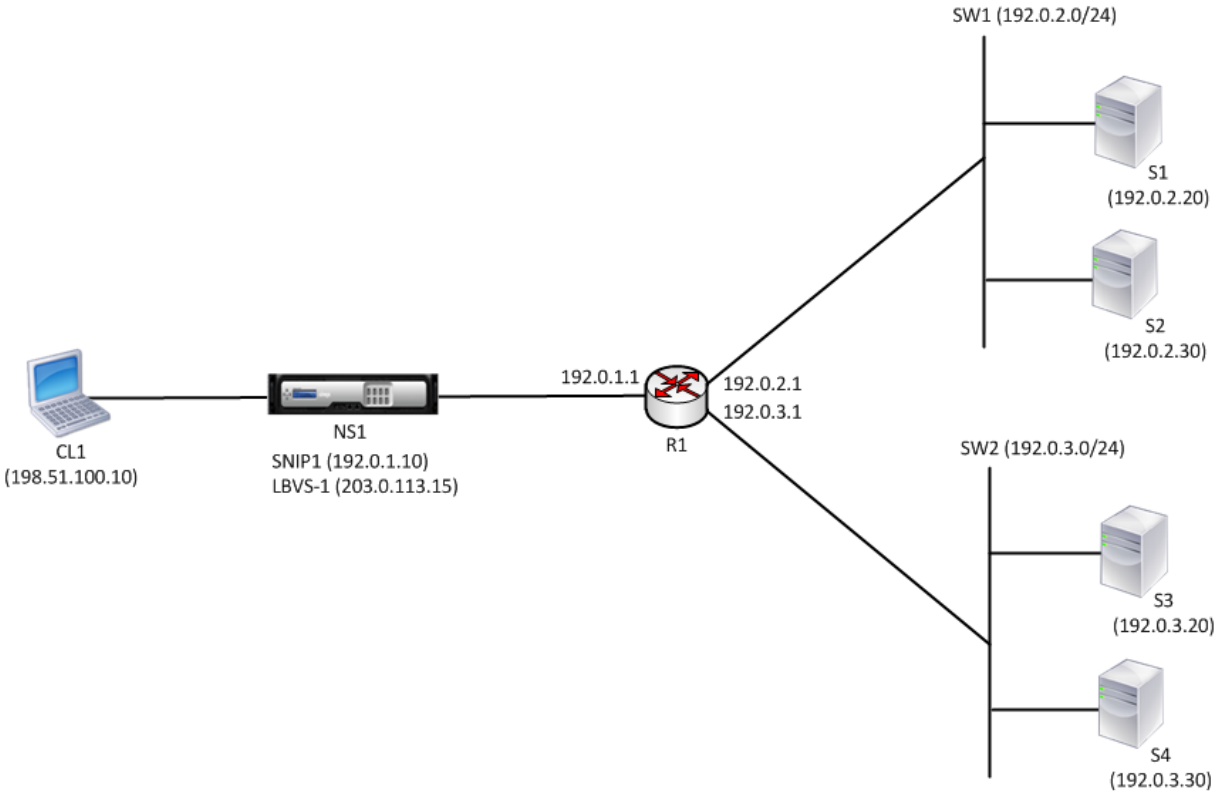
NetScaler ADC での負荷分散の構成の詳細については、「負荷分散」を参照してください。

アドレス SNIP1 が設定されると、NS1 は SNIP1 の ARP アナウンスパケットをブロードキャストします。

NS1 のルーティングテーブルは、S1、S2、S3、および S4 から R1 までのルートエントリで構成されています。これらのルートエントリは、スタティックルートエントリであるか、ダイナミックルーティングプロトコルを使用して R1 から NS1 にアドバタイズされます。

NS1 上の SVC-S1、SVC-S2、SVC-S3、および SVC-S4 の各サービスは、サーバ S1、S2、S3、および S4 を表します。NS1 は、ルーティング・テーブルで、これらのサーバに R1 経由でアクセス可能であることを検出します。NS1 は、アドレス SNIP1 から定期的に監視プローブを送信し、正常性をチェックします。

NetScaler ADC での IP ルーティングの詳細については、「IP ルーティング」を参照してください。



次に、この例のトラフィックフローを示します。

1. クライアント C1 は LBVS-1 に要求パケットを送信します。リクエストパケットには次のものが含まれます。
   - ソース IP = クライアントの IP アドレス (198.51.100.10)
   - 宛先 IP = LBVS-1 の IP アドレス (203.0.113.15)
2. NS1 の LBVS1 は要求パケットを受信します。
3. LBVS1 の負荷分散アルゴリズムはサーバー S3 を選択します。
4. NS1 はルーティングテーブルをチェックし、S3 に R1 経由でアクセスできることを確認します。SNIP1（192.0.1.10）は、ルーター R1 と同じサブネットに属する NS1 の唯一の IP アドレスです。NS1 は、R1 を介して SNIP1 と S3 間の接続を開きます。
5. NS1 は SNIP1 から R1 にリクエストパケットを送信します。リクエストパケットには次のものが含まれます。
   - 送信元 IP アドレス = SNIP1 (192.0.1.10)
   - ターゲット IP アドレス = S3 の IP アドレス (192.0.3.20)
6. 要求は R1 に到達し、R1 はルーティングテーブルをチェックして、要求パケットを S3 に転送します。
7. S3 の応答は同じパスで返されます。

L2 スイッチ上の複数のサーバサブネット（VLAN）に SNIP を使用する

NetScaler に接続された L2 スイッチに複数のサーバーサブネット（VLAN）がある場合は、NetScaler がこれらのサーバーサブネットと通信できるように、各サーバーサブネットに少なくとも 1 つの SNIP アドレスを構成する必要

があります。

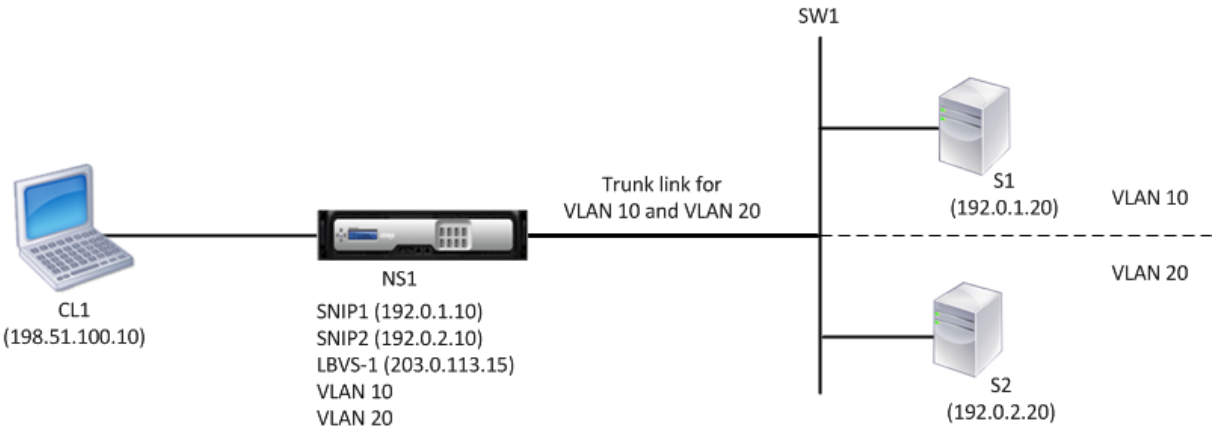
NetScaler NS1 の負荷分散仮想サーバー LBVS1 を使用して、L2 スイッチ SW1 を介して NS1 に接続されているサーバー S1 と S2 の負荷分散を行う負荷分散設定の例を考えてみましょう。S1 と S2 は異なるサブネットに属し、それぞれ VLAN 10 と VLAN20 の一部です。NS1 と SW1 の間のリンクはトランクリンクであり、VLAN10 と VLAN20 によって共有されます。

NetScaler ADC での負荷分散の構成の詳細については、「負荷分散」を参照してください。

サブネット IP アドレス SNIP1（参照目的のみ）および SNIP2（参照目的のみ）は、NS1 上で構成されます。NS1 は (VLAN 10 の）SNIP1 を使用してサーバ S1 と通信し、SNIP2（VLAN 20 上）を使用して S2 と通信します。SNIP1 と SNIP2 が構成されると、NS1 は SNIP1 と SNIP2 の ARP アナウンスメント・パケットをブロードキャストします。

NetScaler ADC での VLAN の構成の詳細については、「VLAN の構成」を参照してください。

NS1 上のサービス SVC-S1 および SVC-S2 は、サーバ S1 および S2 を表します。これらのサービスが設定されるとすぐに、NS1 はそれらのサービスに対する ARP 要求をブロードキャストします。S1 と S2 が応答すると、NS1 は定期的に監視プローブを送信して状態を確認します。NS1 は、アドレス SNIP1 から S1 に、アドレス SNIP2 から S2 にモニタリングプローブを送信します。



次に、この例のトラフィックフローを示します。

1. クライアント C1 は LBVS-1 に要求パケットを送信します。リクエストパケットには次のものが含まれます。
   - ソース IP = クライアントの IP アドレス (198.51.100.10)
   - 宛先 IP = LBVS-1 の IP アドレス (203.0.113.15)
2. NS1 の LBVS1 は要求パケットを受信します。
3. LBVS1 の負荷分散アルゴリズムはサーバー S2 を選択します。
4. S2 は NS1 に直接接続されており、S2 と同じサブネットに属する NS1 上の IP アドレスは SNIP2(192.0.2.10) だけなので、NS1 は SNIP2 と S2 間の接続を開きます。
   注:S1 を選択すると、NS1 は SNIP1 と S1 の間の接続を開きます。
5. NS1 は SNIP2 から S2 にリクエストパケットを送信します。リクエストパケットには次のものが含まれます。

- ソース IP = SNIP1 (192.0.2.10)
- ターゲット IP = S2 の IP アドレス (192.0.2.20)

6. S2 の応答は同じパスで返されます。

GSLB サイト IP アドレス（GSLBIP）の構成

August 15, 2023

GSLB サイト IP (GSLBIP) アドレスは、GSLB サイトに関連付けられた IP アドレスです。NetScaler アプライアンスを最初に構成するときに、GSLBIP アドレスを指定する必要はありません。GSLBIP アドレスは、GSLB サイトを作成する場合にのみ使用されます。

GSLB サイトの IP アドレスの作成の詳細については、「グローバルサーバーの負荷分散」を参照してください。

NetScaler ADC が所有する IP アドレスを削除する

August 15, 2023

NSIP 以外のすべての IP アドレスを削除できます。次の表は、さまざまなタイプの IP アドレスを削除するために実行する必要のあるプロセスをまとめたものです。VIP を削除する前に、関連する仮想サーバーを削除してください。

IP アドレスタイプ	含意
サブネット IP アドレス (SNIP)	削除する IP アドレスがサブネット内の最後の IP アドレスの場合、関連するルートはルートテーブルから削除されます。削除する IP アドレスが対応するルートエントリのゲートウェイである場合、そのサブネットルートのゲートウェイは NetScaler が所有する別の IP アドレスに変更されます。
仮想サーバーの IP アドレス (VIP)	VIP を削除する前に、まず VIP に関連付けられた仮想サーバを削除する必要があります。仮想サーバーの削除の詳細については、「負荷分散」を参照してください。
GSLB サイトの IP アドレス	GSLB サイトの IP アドレスを削除する前に、それに関連付けられているサイトを削除する必要があります。サイトの削除の詳細については、「グローバルサーバーの負荷分散」を参照してください。

CLI を使用して IP アドレスを削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

rm ns ip <IPaddress>

例**:**

```
1 > rm ns ip 10.102.29.54
2 Done
3 <!--NeedCopy-->
```

GUI を使用して IP アドレスを削除するには

[システム] > [ネットワーク] > [**IP**] > [**IPv4]** に移動し、**IP** アドレスを削除します。

アプリケーションアクセス制御の構成

August 15, 2023

管理アクセス制御とも呼ばれるアプリケーションアクセス制御は、ユーザー認証を管理し、アプリケーションとデータへのユーザーアクセスを決定するルールを実装するための統一されたメカニズムを形成します。管理アプリケーションへのアクセスを提供するように SNIP を構成できます。NSIP の管理アクセスはデフォルトで有効になっており、無効にすることはできません。ただし、ACL を使用して制御することはできます。

ACL の使用の詳細については、アクセス制御リスト (ACL)を参照してください。

NetScaler ADC アプライアンスは、VIP への管理アクセスをサポートしていません。

次の表は、管理アクセスと Telnet の特定のサービス設定との相互作用の概要を示しています。

管理アクセス	Telnet (NetScaler で設定されている状態)	Telnet（IP レベルでの有効状態）
有効化	有効化	有効化
有効化	無効化	無効化
無効化	有効化	無効化
無効化	無効化	無効化

次の表は、アウトバウンドトラフィックの送信元 IP アドレスとして使用される IP アドレスの概要を示しています。

アプリケーション/ IP	NSIP	SNIP	VIP
ARP	はい	はい	いいえ
サーバー側トラフィック	いいえ	はい	いいえ

アプリケーション/ IP	NSIP	SNIP	VIP
RNAT	いいえ	はい	はい
ICMP PING	はい	はい	いいえ
動的ルーティング	はい	はい	はい

次の表は、これらの IP アドレスで利用できるアプリケーションの概要を示しています。

アプリケーション/ IP	NSIP	SNIP	VIP
SNMP	はい	はい	はい
システムアクセス	はい	はい	いいえ

Telnet、SSH、GUI、FTP などのアプリケーションを使用して NetScaler にアクセスし、管理できます。

注：セキュリティ上の理由から、NetScaler では Telnet と FTP は無効になっています。有効にするには、カスタマーサポートに連絡してください。アプリケーションを有効にすると、IP レベルで制御を適用できます。

これらのアプリケーションに応答するように NetScaler を構成するには、特定の管理アプリケーションを有効にする必要があります。IP アドレスの管理アクセスを無効にすると、その IP アドレスを使用する既存の接続は終了しませんが、新しい接続を開始することはできません。

また、基盤となる FreeBSD オペレーティングシステムで実行されている管理以外のアプリケーションはプロトコル攻撃を受けやすいため、これらのアプリケーションは NetScaler アプライアンスの攻撃防止機能を活用しません。

SNIP または NSIP 上のこれらの非管理アプリケーションへのアクセスをブロックできます。アクセスがブロックされると、SNIP または NSIP を使用して NetScaler に接続しているユーザーは、基盤となるオペレーティングシステムで実行されている非管理アプリケーションにアクセスできなくなります。

CLI を使用して IP アドレスの管理アクセスを設定するには:

コマンドプロンプトで入力します。

set ns ip <IPAddress> -mgmtAccess <value> -**telnet** <value> -**ftp** <value> -**gui** <value> -**ssh** <value> -**snmp** <value> -**restrictAccess** (**ENABLED** | **DISABLED**)

例**:**

```
1   > set ns ip 10.102.29.54 -mgmtAccess enabled -restrictAccess ENABLED
2    Done
3  <!--NeedCopy-->
```

GUI を使用して IP アドレスの管理アクセスを有効にするには:

1. **System** > **Network** > **IPs** > **IPV4s** に移動します。

2. IP アドレスエントリを開き、[一覧表示されたアプリケーションをサポートするために 管理アクセス制御を有効にする] オプションを選択します。

サブネット **IP** アドレス（**SNIP**）を使用して **NetScaler ADC GUI** への安全なアクセスを有効にする

NetScaler IP（NSIP）では、NetScaler GUI へのセキュアアクセスがデフォルトで有効になっています。アプライアンスのサブネット IP アドレスを使用して、NetScaler ADC アプライアンスへの安全なアクセスを有効にすることもできます。

ハイアベイラビリティペアへのセキュアアクセス用の SNIP アドレスを設定した後、SNIP アドレスにアクセスすると、プライマリアプライアンスがセキュアアクセスできるようになります。

NetScaler LI 手順

CLI を使用してサブネット IP アドレス（SNIP）を使用して NetScaler ADC GUI への安全なアクセスを有効にするには:

コマンドプロンプトで入力します。

ns IP タイプに設定 <SNIP_Address>**SNIP-gui** セキュアオンリー**-mgmtAccess** 有効

例**:**

```
1   > set ns ip 203.0.113.99 -mgmtAccess enabled -restrictAccess ENABLED
2
3    Done
4  <!--NeedCopy-->
```

NetScaler ADC が接続をプロキシする方法

August 15, 2023

クライアントが接続を開始すると、NetScaler アプライアンスはクライアント接続を終了し、適切なサーバーへの接続を開始して、パケットをサーバーに送信します。アプライアンスは、サービスタイプ UDP または ANY に対してこのアクションを実行しません。

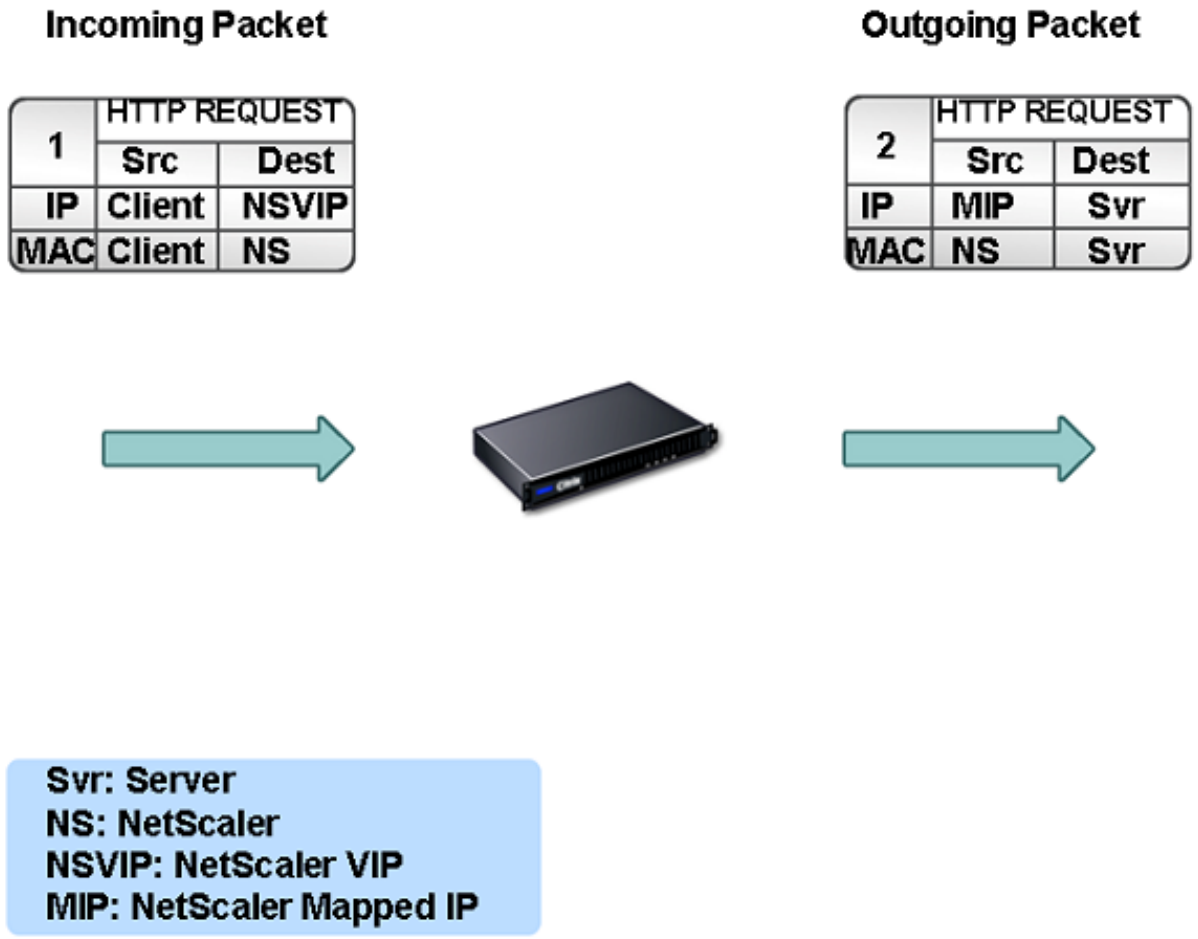
サービスタイプの詳細については、「負荷分散」を参照してください。

サーバーとの接続を開始する前に、パケットを処理するように NetScaler ADC を構成できます。デフォルトの動作では、パケットをサーバーに送信する前に、パケットの送信元と宛先の IP アドレスを変更します。送信元 IP モードの使用を有効化することで、パケットの送信元 IP アドレスを保持するように NetScaler を構成できます。

宛先 **IP** アドレスの選択方法

NetScaler アプライアンスに送信されるトラフィックは、仮想サーバーまたはサービスに送信できます。アプライアンスは、仮想サーバーとサービスへのトラフィックを異なる方法で処理します。NetScaler は、次の図に示すように、仮想サーバーの IP（VIP）アドレスで受信したトラフィックを終了し、宛先 IP アドレスをサーバーの IP アドレスに変更してから、トラフィックをサーバーに転送します。

図 1：VIP へのプロキシ接続



サービス宛てのパケットは適切なサーバーに直接送信され、NetScaler は宛先 IP アドレスを変更しません。この場合、NetScaler はプロキシとして機能します。

送信元 **IP** アドレスの選択方法

NetScaler アプライアンスが物理サーバーまたはピアデバイスと通信する場合、デフォルトではクライアントの IP アドレスは使用されません。NetScaler はサブネット IP アドレス（SNIP）のプールを管理し、このプールから IP

アドレスを選択して、物理サーバーへの接続のソース IP アドレスとして使用します。物理サーバーが配置されているサブネットに応じて、NetScaler は特定の SNIP アドレスを選択します。

注:「ソース IP（USIP）の使用」オプションが有効になっている場合、アプライアンスはクライアントの IP アドレスを使用します。

送信元 **IP** モードの使用を有効にする

August 15, 2023

NetScaler ADC アプライアンスが物理サーバーまたはピアデバイスと通信する場合、デフォルトでは、独自の IP アドレスの 1 つをソース IP として使用します。アプライアンスはサブネット IP アドレス（SNIP）のプールを管理し、このプールから IP アドレスを選択して、物理サーバーへの接続の送信元 IP アドレスとして使用します。SNIP アドレスを選択するかどうかは、物理サーバーが置かれているサブネットによって異なります。

必要に応じて、クライアントの IP アドレスをソース IP として使用するように NetScaler ADC アプライアンスを構成できます。アプリケーションによっては、クライアントの実際の IP アドレスが必要です。以下のユースケースはその一例です。

- Web アクセスログ内のクライアントの IP アドレスは、請求または使用状況分析に使用されます。
- クライアントの IP アドレスは、クライアントの出身国またはクライアントの発信元 ISP を決定するために使用されます。たとえば、Google などの多くの検索エンジンは、ユーザーが属する場所に関連するコンテンツを提供しています。
- アプリケーションは、リクエストが信頼できる送信元からのものであることを確認するために、クライアントの IP アドレスを知っている必要があります。
- アプリケーションサーバーがクライアントの IP アドレスを必要としない場合でも、アプリケーションサーバーと NetScaler ADC の間に配置されたファイアウォールは、トラフィックをフィルタリングするためにクライアントの IP アドレスを必要とする場合があります。

NetScaler ADC がサーバーの通信にクライアントの IP アドレスを使用するようにするには、「送信元 IP モード（USIP）モードを使用する」を有効にします。

次の図は、アプライアンスが USIP モードで IP アドレスをどのように使用するかを示しています。

はじめに

USIP モードを有効にする前に、次の点に注意してください。

- 次の状況では USIP を有効にします。
  - 侵入検知システム (IDS) サーバーの負荷分散
  - SMTP ロードバランシング
  - ステートレス接続フェイルオーバー
  - セッションレスロードバランシング
  - ダイレクトサーバーリターン (DSR) モードを使用する場合
- USIP グローバル設定は、USIP グローバル設定が行われた後に作成されたサービスにのみ適用されます。つまり、USIP グローバル設定を行うと、USIP グローバル設定は既存のサービスには適用されません。たとえば、USIP をグローバルに無効にしても、既存のサービスの USIP は無効になりません。ただし、それ以降に作成されるサービスでは USIP が自動的に有効化されなくなります。

  既存のサービスのセットで USIP を有効または無効にするには、これらの各サービスで USIP を有効または無効にする必要があります。
- USIP が有効になっている場合は、サーバーの応答が常に NetScaler ADC アプライアンスを経由するように、サーバーのゲートウェイを NetScaler ADC が所有する IP アドレス（サブネット IP（SNIP）タイプ）のいずれかに設定する必要があります。
- USIP を有効にする場合は、サーバー接続のアイドルタイムアウトをデフォルト値よりも低い値に設定して、アイドル接続がサーバー側ですぐにクリアされるようにします。
- 透過的なキャッシュリダイレクトを行うには、USIP を有効にする場合は L2CONN も有効にします。
- USIP が有効になっていると HTTP 接続は再利用されないため、サーバー側接続が大量に蓄積される可能性があります。アイドル状態のサーバー接続は、他のクライアントの接続をブロックする可能性があります。そのため、サービスへの最大接続数に制限を設定してください。また、Citrix では、USIP が有効になっているサ

ービスの HTTP サーバーのタイムアウト値をデフォルトよりも低い値に設定することを推奨しています。これにより、アイドル状態の接続がサーバー側ですぐにクリアされます。

- USIP モードの代わりに、クライアントの IP アドレスを必要とするアプリケーションサーバーのサーバー側接続のリクエストヘッダーにクライアントの IP アドレス (CIP) を挿入することもできます。
- 以前の NetScaler ADC リリースでは、USIP モードにはサーバー側接続用の次のソースポートオプションがありました。
  - クライアントのポートを使用してください。このオプションでは、接続を再利用することはできません。クライアントからのリクエストごとに、物理サーバーとの新しい接続が確立されます。
  - プロキシポートを使用してください。このオプションを使用すると、同じクライアントからのすべてのリクエストで接続を再利用できます。

  それ以降の NetScaler ADC リリースでは、USIP が有効になっている場合、デフォルトではサーバー側の接続にはプロキシポートを使用し、接続を再利用しません。接続を再利用しなくても、接続の確立速度には影響しない場合があります。

  デフォルトでは、USIP モードが有効になっている場合、「プロキシポートを使用する」オプションは有効になります。

  注**:** USIP モードを有効にする場合は、[プロキシポートを使用] オプションを有効にすることをお勧めします。

  [プロキシポートを使用] オプションの詳細については、「サーバー側接続の送信元ポートを構成する」を参照してください。

構成の手順

NetScaler ADC がクライアントの IP アドレスを使用してサーバーと通信できるようにするには、「送信元 IP モード（USIP）モードを使用する」を有効にします。デフォルトでは、USIP モードは無効になっています。USIP モードは、NetScaler または特定のサービスでグローバルに有効にできます。グローバルに有効にすると、その後作成されるすべてのサービスで USIP がデフォルトで有効になります。特定のサービスで USIP を有効にすると、クライアントの IP アドレスはそのサービスに向けられたトラフィックにのみ使用されます。

CLI のプロシージャ

CLI を使用して USIP モードをグローバルに有効または無効にするには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- **NS** モード **USIP** を有効にする
- **NS** モード **USIP** を無効にする

CLI を使用してサービスの USIP モードを有効にするには:

コマンドプロンプトで入力します。

** サービス @ を設定 ** <name>-**usip** (** はい **) | **NO**)

例**:**

```
1 > set service Service-HTTP-1 -usip YES
2 Done
3 <!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して **USIP** モードをグローバルに有効にするには:

1. [システム] > [設定**]** に移動し、[モードと機能] グループで [モードの変更] をクリックします。
2. 「送信元 **IP** を使用」オプションを選択します。

GUI を使用してサービスの **USIP** モードを有効にするには:

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを編集します。
2. [詳細設定] で [サービス設定] を選択し、[ソース **IP** アドレスを使用] を選択します。

ネットワークアドレス変換の構成

August 15, 2023

ネットワークアドレス変換（NAT）では、NetScaler アプライアンスを通過する IP パケットの送信元または宛先の IP アドレスおよび/または TCP/UDP ポート番号を変更します。アプライアンスで NAT を有効にすると、データが NetScaler を通過するときにネットワークのソース IP アドレスを変更することで、プライベートネットワークのセキュリティが強化され、インターネットなどのパブリックネットワークから保護されます。また、NAT エントリを使用すると、プライベートネットワーク全体をいくつかの共有パブリック IP アドレスで表現できます。NetScaler は、次の種類のネットワークアドレス変換をサポートしています。

- インバウンド **NAT (INAT)**。NetScaler は、クライアントによって生成されたパケット内の宛先 IP アドレスを、サーバーのプライベート IP アドレスに置き換えます。
- リバース **NAT (RNAT)**。NetScaler は、サーバーによって生成されたパケットの送信元 IP アドレスをパブリック NAT IP アドレスに置き換えます。

インバウンドネットワークアドレス変換

August 15, 2023

クライアントがインバウンドネットワークアドレス変換（INAT）用に構成された NetScaler アプライアンスにパケットを送信すると、アプライアンスはパケットのパブリック宛先 IP アドレスをプライベート宛先 IP アドレスに変換し、そのアドレスのサーバーに転送します。

次の構成がサポートされています。

- **IPv4-IPv4** マッピング：NetScaler アプライアンスのパブリック IPv4 アドレスは、プライベート IPv4 サーバーに代わって接続要求を受信します。NetScaler アプライアンスは、パケットのパブリック宛先 IP アドレスをサーバーの宛先 IP アドレスに変換します。次に、アプライアンスはパケットをそのアドレスのサーバーに転送します。
- **IPv4-IPv6** マッピング：NetScaler アプライアンスのパブリック IPv4 アドレスは、プライベート IPv6 サーバーに代わって接続要求を受信します。NetScaler アプライアンスは、IPv6 サーバーの IP アドレスを宛先 IP アドレスとして IPv6 リクエストパケットを作成します。
- **IPv6-IPv4** マッピング：NetScaler アプライアンスのパブリック IPv6 アドレスは、プライベート IPv4 サーバーに代わって接続要求を受信します。NetScaler アプライアンスは、IPv4 サーバーの IP アドレスを宛先 IP アドレスとして IPv4 リクエストパケットを作成します。
- **IPv6-IPv6** マッピング：NetScaler アプライアンスのパブリック IPv6 アドレスは、プライベート IPv6 サーバーに代わって接続要求を受信します。NetScaler アプライアンスは、パケットのパブリック宛先 IP アドレスをサーバーの宛先 IP アドレスに変換します。次に、アプライアンスはパケットをそのアドレスのサーバーに転送します。

アプライアンスがパケットをサーバーに転送すると、パケットに割り当てられる送信元 IP アドレスは次のように決定されます。

- サブネット IP（USNIP）の使用モードが有効で、ソース IP（USIP）モードの使用が無効になっている場合、アプライアンスは送信元 IP アドレスとしてサブネット IP アドレス（SNIP）を使用します。
- USIP モードが有効で、USNIP モードが無効の場合、アプライアンスはクライアント IP（CIP）アドレスを送信元 IP アドレスとして使用します。
- USIP モードと USNIP モードの両方が有効な場合は、USIP モードが優先されます。
- ProxyIP パラメーターを設定して、ソース IP アドレスとして一意の IP アドレスを使用するように NetScaler を構成することもできます。
- 上記のモードのいずれも有効になっておらず、固有の IP アドレスが指定されていない場合、NetScaler は MIP をソース IP アドレスとして使用しようとします。
- USIP モードと USNIP モードの両方が有効で、固有の IP アドレスが指定されている場合、優先順位は次のようになります。つまり、USIP 固有 IP-USnip-MIP-Error です。

NetScaler を DoS 攻撃から保護するために、TCP プロキシを有効にすることができます。ただし、ネットワークで他の保護メカニズムが使用されている場合は、それらを無効にできます。

INAT ルールの設定

INAT エントリを作成、変更、または削除できます。

CLI のプロシージャ

CLI を使用して INAT エントリを作成するには:

コマンドプロンプトで、次のコマンドを入力して INAT エントリを作成し、その構成を確認します。

- **add inat** <name> <publicIP> <privateIP> [-**tcpproxy** (**ENABLED** | **DISABLED**)] [-**ftp** (**ENABLED** | **DISABLED**)] [-**usip** (**ON** | **OFF**)] [-**usnip** (**ON** | **OFF**)] [-**proxyIP** <**ip_addr > ipv6_addr>**]
- **show inat** [\<name>]

例**:**

```
1 > add inat ip4-ip4 172.16.1.2 192.168.1.1 -proxyip 10.102.29.171
2  Done
3 <!--NeedCopy-->
```

CLI を使用して INAT エントリを変更するには:

INAT エントリを変更するには、`set inat` コマンド、エントリの名前、変更するパラメータを新しい値とともに入力します。

CLI を使用して INAT 設定を削除するには:

コマンドプロンプトで入力します。

- **rm inat** <name>

例**:**

```
1 > rm inat ip4-ip4
2  Done
3 <!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して INAT エントリを設定するには:

[システム] > [ネットワーク] > [ルート] > [**INAT**] に移動し、INAT エントリを追加するか、既存の INAT エントリを編集します。

GUI を使用して INAT 設定を削除するには:

[システム] > [ネットワーク] > [ルート] > [**INAT**] に移動し、INAT 設定を削除します。

INAT ルールの接続フェイルオーバー

接続フェイルオーバーまたは接続ミラーリングにより、プライマリノードは接続と永続性の情報を高可用性でセカンダリノードに複製できます。接続ミラーリングが有効になっている場合、接続の状態情報は定期的にセカンダリノードと共有されます。

接続フェイルオーバーを有効にすると、信頼性は高くなりますが、状態情報の共有にシステム時間がいくらか消費されるという代償が伴います。接続データは、パケットまたはフローステートが更新されるたびにスタンバイユニットに同期されます。したがって、接続レベルの信頼性が最も重要な場所でのみ使用する必要があります。

NetScaler アプライアンスの高可用性設定は、INAT 接続の接続フェイルオーバーをサポートします。プライマリノードは、INAT マッピングとその他の INAT 関連接続情報を定期的にセカンダリノードに送信します。セカンダリアプライアンスは、フェイルオーバーが発生した場合にのみマッピングと接続情報を使用します。

フェイルオーバーが発生すると、新しいプライマリノードには、フェイルオーバー前に確立された INAT 接続に関する情報が格納されます。そのため、フェイルオーバー後もこれらの接続は引き続き提供されます。

クライアントから見ると、フェイルオーバーは透過的です。移行期間中、クライアントとサーバーで短時間の中断と再送信が発生する可能性があります。接続フェイルオーバーは INAT ルールごとに有効にできます。

INAT ルールで接続フェイルオーバーを有効にするには、CLI を使用してその特定の RNAT `connFailover` ルールのパラメータを有効にします。

CLI 手順

CLI を使用して INAT ルールの接続フェイルオーバーを有効にするには:

INAT ルールの追加中に接続フェイルオーバーを有効にするには、コマンドプロンプトで次のように入力します。

- **add inat** <name> <publicIP> <privateIP> [-**tcpproxy** (**ENABLED** | **DISABLED**)] [-**ftp** (**ENABLED** | **DISABLED**)] [-**usip** (**ON** | **OFF**)] [-**usnip** (**ON** | **OFF**)] [-**proxyIP** <ip_addr|ipv6_addr>] -**connfailover** (**ENABLED** | **DISABLED**)
- **show inat** <name>

既存の INAT ルールの変更中に接続フェイルオーバーを有効にするには、コマンドプロンプトで次のように入力します。

set inat -connfailover (**ENABLED** **DISABLED**)

-
- **show inat** <name>

INAT と仮想サーバーの共存

August 15, 2023

INAT と RNAT の両方が設定されている場合、INAT ルールが RNAT ルールよりも優先されます。RNAT にネットワークアドレス変換 IP（NAT IP）アドレスが設定されている場合、NAT IP アドレスがその RNAT クライアントの送信元 IP アドレスとして選択されます。

INAT 構成のデフォルトのパブリック宛先 IP は、NetScaler デバイスの仮想 IP（VIP）アドレスです。仮想サーバーも VIP を使用します。INAT と仮想サーバーの両方が同じ IP アドレスを使用する場合、Vserver 構成は INAT 構成よりも優先されます。

次に、構成設定のシナリオとその効果の例をいくつか示します。

ケース	結果
特定の NetScaler ポートで受信したすべてのデータパケットをサーバーに直接送信するように仮想サーバーとサービスを構成しました。また、INAT を設定し、TCP を有効にしました。このように INAT を設定すると、TCP エンジンを介して受信したすべてのデータパケットがサーバーに送信される前に送信されます。	NetScaler で受信されたすべてのパケットは、指定されたポートで受信されたものを除き、TCP エンジンを通過します。
NetScaler の特定のポートで受信したサービスタイプ TCP のすべてのデータパケットを、TCP エンジンを通過した後にサーバーに送信するように仮想サーバーとサービスを構成しました。INAT を設定し、TCP を無効にしました。このように INAT を設定すると、受信したデータパケットがサーバーに直接送信されます。	指定されたポートで受信されたパケットのみが TCP エンジンを通過します。
受信したすべてのデータパケットを 2 つのサーバーのどちらかに送信するように、仮想サーバーとサービスを構成しました。受信したすべてのデータパケットを別のサーバーに送信するように INAT を設定しようとしています。	INAT 設定は許可されていません。
受信したすべてのデータパケットをサーバーに直接送信するように INAT を設定しました。受信したすべてのデータパケットを 2 つの異なるサーバーに送信するように、仮想サーバーとサービスを構成しようとしています。	仮想サーバーの設定は許可されていません。

ステートレス **NAT46**

August 15, 2023

ステートレス NAT46 機能により、NetScaler アプライアンスのセッション情報を維持することなく、IPv4 から IPv6 へのパケット変換、またはその逆のパケット変換による IPv4 ネットワークと IPv6 ネットワーク間の通信が可能になります。

ステートレス NAT46 構成の場合、アプライアンスは RFC 6145 および 2765 で定義されているように、IPv4 パケットを IPv6 に変換するか、IPv6 パケットを IPv4 に変換します。

NetScaler アプライアンスのステートレス NAT46 構成には、次のコンポーネントがあります。

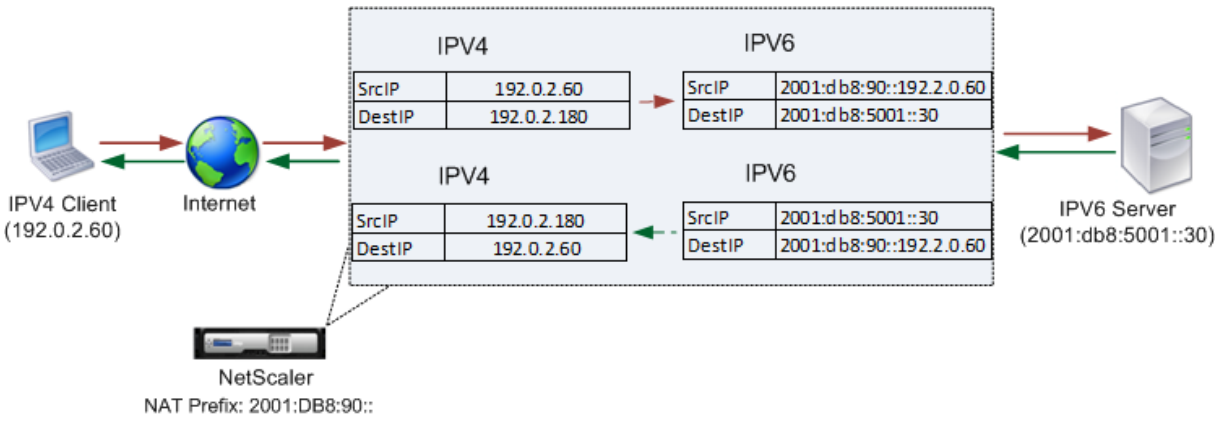
- **IPv4-IPv6 INAT** エントリ。IPv4 アドレスと IPv6 アドレスの間の 1:1 の関係を定義する INAT エントリ。つまり、アプライアンス上の IPv4 アドレスは、IPv6 サーバーに代わって接続要求を受信します。この IPv4 アドレスの IPv4 要求パケットが IPv6 パケットに変換され、IPv6 パケットが IPv6 サーバーに送信されます。

  アプライアンスは、IPv6 応答パケットを、送信元 IP アドレスフィールドが INAT エントリで指定された IPv4 アドレスとして設定された IPv4 応答パケットに変換します。その後、変換されたパケットはクライアントに送信されます。

- **NAT46 IPv6 prefix.** アプライアンスで設定されている長さ 96 ビット（128-32=96）のグローバル IPv6 プレフィクス。IPv4 パケットから IPv6 パケットへの変換中、アプライアンスは変換された IPv6 パケットの送信元 IP アドレスを、要求パケットで受信した NAT46 IPv6 プレフィックス [96 ビット] と IPv4 送信元アドレス [32 ビット] を連結して設定します。

  IPv6 パケットから IPv4 パケットへの変換中、アプライアンスは、変換された IPv4 パケットの宛先 IP アドレスを、IPv6 パケットの宛先 IP アドレスの最後の 32 ビットに設定します。

企業が IPv6 アドレスを持つサーバー S1 にサイト www.example.com をホストしている例を考えてみましょう。IPv4 クライアントと IPv6 サーバー S1 間の通信を可能にするために、NetScaler アプライアンス NS1 は、サーバー S1 の IPv4-IPv6 INAT エントリと NAT46 プレフィックスを含むステートレス NAT46 構成で展開されます。INAT エントリには、アプライアンスが IPv6 サーバー S1 の代わりに IPv4 クライアントからの接続要求を受信する IPv4 アドレスが含まれています。



次の表に、この例で使用されている設定の一覧を示します。

エンティティ	名前	値
クライアントの IP アドレス	クライアント _IPv4 (参照用のみ)	192.0.2.60
サーバーの IPv6 アドレス	SEVR_IPv6 (参照のみを目的としています)	2001:DB8:5001::30
IPv6 サーバー S1 の INAT エントリで定義されている IPv4 アドレス	マップ-SevR-IPv4 (参照用のみ)	192.0.2.180
NAT 46 変換用の IPv6 プレフィックス	NAT46_ プレフィックス (参照用のみ)	2001:DB8:90:

次に、この例のトラフィックフローを示します。

1. IPv4 クライアント CL1 は、NetScaler アプライアンス上の Map-SevR-IPv4（192.0.2.180）アドレスにリクエストパケットを送信します。
2. アプライアンスは要求パケットを受信し、NAT46 INAT エントリを検索して、MAP-SEVR-IPv4 (192.0.2.180) アドレスにマップされた IPv6 アドレスを検索します。SEVR-IPv6 (2001: DB 8:5001::30) アドレスを検索します。
3. アプライアンスは、次のように変換された IPv6 リクエストパケットを作成します。
    - 宛先 IP アドレスフィールド = SEVR-IPv6 = 2001: DB 8:5001:30
    - 送信元 IP アドレスフィールド = NAT プレフィックス (最初の 96 ビット) と Client_IPv4 (最後の 32 ビット) を連結したもの = 2001: DB 8:90:: 192.0.2.60
4. アプライアンスは、変換された IPv6 要求を SevR-IPv6 に送信します。
5. IPv6 サーバー S1 は、次のように IPv6 パケットを NetScaler アプライアンスに送信することで応答します。
    - 宛先 IP アドレスフィールド = NAT プレフィックス (最初の 96 ビット) と Client_IPv4 (最後の 32 ビット) を連結したもの = 2001: DB 8:90:: 192.0.2.60
    - ソース IP アドレスフィールド = SEVR-IPv6 = 2001: DB 8:5001:30
6. アプライアンスは IPv6 応答パケットを受信し、その宛先 IP アドレスがアプライアンスに設定されている NAT46 プレフィックスと一致することを確認します。宛先アドレスは NAT46 プレフィックスと一致するため、アプライアンスは NAT46 INAT エントリを検索して SEVR-IPv6 アドレス（2001：DB 8:5001：30）に関連付けられた IPv4 アドレスを検索します。マップサービスの IPv4 アドレス (192.0.2.180) を検索します。
7. アプライアンスは以下を含む IPv4 応答パケットを作成します。
    - 宛先 IP アドレスフィールド = IPv6 応答の宛先アドレスから削除された NAT46 プレフィックス = Client_IPv4 (192.0.2.60)
    - 送信元 IP アドレスフィールド = MAP-SEVR-IPv4 アドレス (192.0.2.180)
8. アプライアンスは、変換された IPv4 応答をクライアント CL1 に送信します。

ステートレス **NAT46** の制限事項

ステートレス NAT46 には次の制限が適用されます。

- IPv4 オプションの変換はサポートされていません。
- IPv6 ルーティングヘッダーの変換はサポートされていません。
- IPv6 パケットのホップバイホップ拡張ヘッダーの変換はサポートされていません。
- IPv4 パケットの ESP および EH ヘッダーの変換はサポートされていません。
- マルチキャストパケットの変換はサポートされていません。
- 宛先オプションヘッダーとソースルーティングヘッダーの変換はサポートされていません。
- UDP チェックサムを含まない断片化された IPv4 UDP パケットの変換はサポートされていません。

ステートレス **NAT46** の設定

NetScaler アプライアンスでステートレス NAT46 構成に必要なエンティティを作成するには、次の手順が必要です。

1. ステートレスモードを有効にして IPv4-IPv6 マッピング INAT エントリを作成します。
2. NAT46 IPv6 プレフィックスを作成します。

CLI のプロシージャ

CLI を使用して INAT マッピングエントリを設定するには:

コマンドプロンプトで入力します。

- add inat <name> <publicIPv4> <privateIPv6> -mode STATELESS
- show inat <name>

CLI を使用して NAT46 プレフィックスを作成するには:

コマンドプロンプトで入力します。

---

set inatparam -nat46v6Prefix <ipv6_addr *>

-
- show inatparam

例**:**

```
 > add inat exmpl-com-stls-nat46 192.0.2.180
2001:DB8:5001::30 -mode stateless
Done
```



```
4
5  > set inatparam -nat46v6Prefix 2001:DB8:90::/96
6  Done
7  <!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して INAT マッピングエントリを作成するには:

1. [システム] > [ネットワーク] > [ルート] > [INAT] に移動します。
2. 新しい INAT エントリを追加するか、既存の INAT エントリを編集します。
3. 次のパラメーターを設定します。

    - 名前 *
    - パブリック IP アドレス *
    - プライベート IP アドレス * (IPv6 チェックボックスを選択し、IPv6 形式でアドレスを入力します)
    - モード (ドロップダウンリストから [ステートレス] を選択します。)

    * 必須パラメータ

GUI を使用して NAT46 プレフィックスを作成するには:

[システム] > [ネットワーク] に移動し、[設定] グループで [**INAT** パラメータの設定] をクリックし、[プレフィックス] パラメータを設定します。

ステートレス **NAT46** のグローバルパラメータの設定

アプライアンスには、ステートレス NAT46 構成用のオプションのグローバルパラメータがいくつか用意されています。

CLI を使用してステートレス NAT46 のグローバルパラメータを設定するには:

コマンドプロンプトで入力します。

---

set inatparam [-**nat46IgnoreTOS** (**YES** **NO**)] [-**nat46ZeroCheckSum** (**ENABLED** **DISABLED**)] [-**nat46v6Mtu**

-
- **show inatparam**

例**:**


```
> set inatparam -nat46IgnoreTOS YES -nat46ZeroCheckSum DISABLED -
    nat46v6Mtu 1400 -nat46FragHeader DISABLED
 Done
<!--NeedCopy-->
```

GUI を使用してステートレス NAT46 のグローバルパラメータを設定するには:

[システム] > [ネットワーク] に移動し、[設定] グループで [**INAT** パラメータの設定] をクリックします。

DNS64

August 15, 2023

NetScaler DNS64 機能は、IPv4 のみのドメインに対して AAAA リクエストを送信する IPv6 クライアントに対して、合成された DNS AAAA レコードで応答します。DNS64 機能を NAT64 機能と併用すると、IPv6 のみのクライアントと IPv4 のみのサーバー間のシームレスな通信が可能になります。DNS64 は IPV6 のみのクライアントによる IPv4 ドメインの検出を可能にし、NAT64 はクライアントとサーバー間の通信を可能にします。

AAAA レコードを合成するために、NetScaler アプライアンスは DNS サーバーから DNS A レコードを取得します。DNS64 プレフィックスは、NetScaler アプライアンスで構成された 96 ビットの IPv6 プレフィックスです。NetScaler アプライアンスは、DNS64 プレフィックス（96 ビット）と IPv4 アドレス（32 ビット）を連結して AAAA レコードを合成します。

IPv6 クライアントと IPv4 サーバー間の通信を可能にするために、DNS64 および NAT64 構成の NetScaler アプライアンスを IPv6 クライアント側または IPv4 サーバー側に展開できます。どちらの場合も、NetScaler アプライアンスの DNS64 構成は似ており、DNS サーバーのプロキシサーバーとして機能する負荷分散仮想サーバーが含まれています。NetScaler アプライアンスをクライアント側に展開する場合、IPv6 クライアント上の負荷分散仮想サーバーをドメインのネームサーバーとして指定する必要があります。

DNS64 と NAT64 の構成を備えた NetScaler アプライアンスが IPv4 側で構成されている例を考えてみましょう。この例では、企業は IPv4 アドレスを持つサーバー S1 でサイト www.example.com をホストしています。IPv6 クライアントと IPv4 サーバー S1 間の通信を可能にするために、NetScaler アプライアンス NS1 は DNS64 とステートフル NAT64 構成でデプロイされます。

DNS64 構成には、DNS64 オプションが有効になっている DNS 負荷分散仮想サーバー LBVS-DNS64-1 が含まれます。DNS64-Policy-1 という名前の DNS64 ポリシーと、DNS64-Action-1 という名前の関連する DNS64 アクションも NS1 で設定されており、DNS64-Policy-1 は LBVS-DNS64-1 にバインドされています。LBVS-DNS64-1 は DNS サーバー DNS-1 と DNS-2 の DNS プロキシサーバーとして機能します。

LBVS-DNS64-1 に到着するトラフィックが DNS64-Policy-1 で指定された条件と一致する場合、トラフィックは DNS64-Action-1 の設定に従って処理されます。DNS64-Action-1 では、AAAA レコードを合成する際に使用する DNS64 プレフィックスと DNS サーバーから受信した A レコードを指定します。

NetScaler アプライアンスではグローバル DNS パラメーター cacherecords が有効になっているため、アプライアンスは DNS レコードをキャッシュします。この設定は、DNS64 が正常に動作するために必要です。

次の表に、上記の例で使用した設定を示します。DNS64 の設定例。

次に、この例のトラフィックフローを示します。

1. IPv6 クライアント CL1 は、サイト www.example.com の IPv6 アドレスに対して DNS AAAA リクエストを送信します。
2. リクエストは、NetScaler アプライアンス NS1 上の DNS 負荷分散仮想サーバー LBVS-DNS64-1 によって受信されます。
3. NS1 は DNS キャッシュレコードに要求された AAAA レコードがないかどうかを確認し、サイト www.example.com の AAAA レコードが DNS キャッシュに存在しないことを検出します。
4. LBVS-DNS64-1 のロードバランシングアルゴリズムは、DNS サーバー DNS-1 を選択し、AAAA リクエストをそのサーバーに転送します。
5. サイト www.example.com は IPv4 サーバーでホストされているため、DNS サーバー DNS-1 にはサイト www.example.com の AAAA レコードはありません。
6. DNS-1 は、空の DNS AAAA 応答またはエラーメッセージを LBVS-DNS64-1 に送信します。
7. LBVS-DNS64-1 では DNS64 オプションが有効になっていて、CL1 からの AAAA リクエストは DNS64-Policy-1 で指定された条件と一致するため、NS1 は www.example.com の IPv4 アドレスの DNS A リクエストを DNS-1 に送信します。
8. DNS-1 は www.example.com の DNS A レコードを LBVS-DNS64-1 に送信することで応答します。A レコードには www.example.com の IPv4 アドレスが含まれています。
9. NS1 はサイト www.example.com の AAAA レコードを次のように合成します。
    - サイト www.example.com の IPv6 アドレス = 関連する DNS64Action で指定されている DNS64 プレフィックス (96 ビット) と DNS A レコードの IPv4 アドレス (32 ビット) を連結したもの = 2001: DB 8:300:: 192.0.2.60
10. NS1 は、合成された AAAA レコードを IPv6 クライアント CL1 に送信します。NS1 は A レコードもメモリにキャッシュします。NS1 はキャッシュされた A レコードを使用して、後続の AAAA リクエスト用に AAAA レコードを合成します。

DNS64 構成で考慮すべきポイント

NetScaler アプライアンスで DNS64 を構成する前に、次の点を考慮してください。

- NetScaler アプライアンスの DNS64 機能は RFC 6174 に準拠しています。
- NetScaler アプライアンスの DNS64 機能は DNSSEC をサポートしていません。NetScaler アプライアンスは、DNS サーバーから受信した DNSSEC 応答から AAAA レコードを合成しません。応答は、RRSIG レコードを含む場合にのみ DNSSEC 応答として分類されます。
- NetScaler アプライアンスは、96 ビットの長さの DNS64 プレフィックスしかサポートしていません。

- DNS64 機能は NAT64 機能とともに使用されますが、DNS64 と NAT64 の構成は NetScaler アプライアンスとは独立しています。特定のフローでは、クライアントが受信した合成 IPv6 アドレスが特定の NAT64 設定にルーティングされるように、DNS64 プレフィクスと NAT64 プレフィクスパラメータに同じ IPv6 プレフィクス値を指定する必要があります。NetScaler ADC アプライアンスでの NAT64 の構成の詳細については、ステートフル NAT64 を参照してください。
- NetScaler ADC アプライアンスによる DN64 処理の異なるケースを次に示します。
  - DNS サーバーからの AAAA レスポンスに AAAA レコードが含まれている場合、レスポンス内の各レコードが、NetScaler アプライアンスで特定の DNS64 構成に対して構成されている除外ルールのセットと照合されます。NetScaler は、プレフィックスが除外ルールと一致する IPv6 アドレスを応答から削除します。結果の応答に少なくとも 1 つの IPv6 レコードが含まれている場合、NetScaler アプライアンスはこの応答をクライアントに転送します。それ以外の場合、アプライアンスはドメインの A レコードから AAAA 応答を合成して IPv6 クライアントに送信します。
  - DNS サーバからの AAAA 応答が空の応答である場合、アプライアンスは同じドメイン名の A リソースレコードを要求するか、アプライアンスがそのドメインの本物のドメインネームサーバかどうかを独自のレコードで検索します。リクエストの結果、回答またはエラーが返された場合は、その回答がクライアントに転送されます。
  - DNS サーバーからの応答に RCODE=1（フォーマットエラー）が含まれる場合、NetScaler アプライアンスはそれをクライアントに転送します。タイムアウト前に応答がない場合、NetScaler アプライアンスは RCODE=2（サーバー障害）の応答をクライアントに送信します。
  - DNS サーバーからの応答に CNAME が含まれている場合は、終端の A または AAAA レコードに到達するまでチェーンが続きます。CNAME に AAAA リソースレコードがない場合、NetScaler アプライアンスは AAAA レコードの合成に使用する DNS A レコードを取得します。CNAME チェーンは、合成された AAAA レコードとともに回答セクションに追加され、クライアントに送信されます。
- NetScaler アプライアンスの DNS64 機能は、PTR 要求への応答もサポートしています。IPv6 アドレスのドメインに対する PTR 要求がアプライアンスで受信され、IPv6 アドレスが設定された DNS64 プレフィックスのいずれかと一致すると、アプライアンスは IP6-ARPA ドメインを対応する IN-ADDR にマッピングする CNAME レコードを作成します。ARPA ドメインと新しく形成された IN-ADDR.ARPA ドメインが解決に使用されます。アプライアンスはローカルの PTR レコードを検索し、レコードが存在しない場合、アプライアンスは IN-ADDR.ARPA ドメインの PTR 要求を DNS サーバーに送信します。NetScaler アプライアンスは、DNS サーバーからの応答を使用して、最初の PTR 要求に対する応答を合成します。

構成の手順

NetScaler アプライアンスでステートフル NAT64 構成に必要なエンティティを作成するには、次の手順が必要です。

- **DNS** サービスを追加します。DNS サービスは、NetScaler ADC アプライアンスが DNS プロキシサーバー

として機能する DNS サーバーを論理的に表現したものです。サービスのオプションパラメータの設定の詳細については、「負荷分散」を参照してください。

- **DNS64** アクションと **DNS64** ポリシーを追加し、**DNS64** アクションを **DNS64** ポリシーにバインドします。DNS64 ポリシーは、関連する DNS64 アクションの設定に従って、DNS64 処理のトラフィックと照合する条件を指定します。DNS64 アクションでは、必須の DNS64 プレフィックスと、オプションの除外ルールとマップルール設定を指定します。
- **DNS** 負荷分散仮想サーバーを作成し、**DNS** サービスと **DNS64** ポリシーをそれにバインドします。DNS 負荷分散仮想サーバーは、バインドされた DNS サービスに代表される DNS サーバーの DNS プロキシサーバーとして機能します。仮想サーバに着信するトラフィックは、DNS64 処理用のバインドされた DNS64 ポリシーと照合されます。負荷分散仮想サーバーのオプションパラメータの設定の詳細については、「負荷分散」を参照してください。

  注：CLI には、これら 2 つのタスクに対して個別のコマンドがありますが、GUI では 1 つのダイアログボックスにまとめられています。

  DNS レコードのキャッシュを有効にします。NetScaler ADC アプライアンスのグローバルパラメータを有効にして、DNS プロキシ操作によって取得される DNS レコードをキャッシュします。DNS レコードのキャッシュの有効化の詳細については、「ドメインネームシステム」を参照してください。

CLI のプロシージャ

CLI を使用して DNS タイプのサービスを作成するには:

コマンドプロンプトで入力します。

- add service <name> <IP> <serviceType> <port> …

CLI を使用して DNS64 アクションを作成するには:

コマンドプロンプトで入力します。

- add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule \] \[-excludeRule \\]

CLI を使用して DNS64 ポリシーを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add dns policy64 <name> -rule <expression> -action <string>

CLI を使用して DNS 負荷分散仮想サーバーを作成するには:

コマンドプロンプトで入力します。

- add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED) [-bypassAAAA (YES | NO)] …

CLI を使用して DNS サービスと DNS64 ポリシーを DNS ロードバランシング仮想サーバにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- bind lb vserver <name> <serviceName> …
- bind lb vserver <name> -policyName <string> -priority <positive_integer> …

GUI のプロシージャ

GUI を使用して DNS タイプのサービスを作成するには:

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、新しいサービスを追加します。
2. 次のパラメーターを設定します。
    - サービス名 *
    - サーバー *
    - プロトコル * (ドロップダウンリストから DNS を選択)
    - ポート *

GUI を使用して DNS64 アクションを作成するには:

[トラフィック管理] > [DNS] > [アクション] に移動し、[DNS アクション 64] タブで新しい DNS64 アクションを追加します。

GUI を使用して DNS64 ポリシーを作成するには:

[トラフィック管理] > [DNS] > [ポリシー] に移動し、[DNS ポリシー 64] タブで新しい DNS64 ポリシーを追加します。

DNS 負荷分散仮想サーバーを作成し、GUI を使用して DNS サービスと DNS64 ポリシーをそれにバインドするには:

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、新しい仮想サーバーを追加します。
2. 次のパラメーターを設定します。
    - 名前 *
    - IP アドレス *
    - プロトコル * (ドロップダウンリストから DNS を選択)
    - ポート *
3. 「DNS64 を有効にする」オプションを選択します。
4. サービスペインで、サービスを仮想サーバーにバインドします。
5. ポリシーペインで、ポリシーを仮想サーバーにバインドします。

構成例

```
 > add service SVC-DNS-1 203.0.113.50 DNS 53
 Done

> add service SVC-DNS-2 203.0.113.60 DNS 53
 Done

> add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
 Done

> add dns Policy64 DNS64-Policy-1 -rule "CLIENT.IPv6.SRC.IN_SUBNET
   (2001:DB8:5001::/64)"
-action DNS64-Action-1
 Done

> add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-1
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-2
 Done

> bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
 Done

<!--NeedCopy-->
```

ステートフル **NAT64** の変換

August 16, 2023

ステートフル NAT64 機能により、NetScaler アプライアンスのセッション情報を維持したまま、IPv6 から IPv4 へのパケット変換、またはその逆のパケット変換による IPv6 クライアントと IPv4 サーバー間の通信が可能になります。

NetScaler アプライアンスのステートフル NAT64 構成には、次のコンポーネントが含まれます。

- **NAT64** ルール—**ACL6** ルールとネットプロファイルで構成されるエントリ。ネットプロファイルは、NetScaler が所有する SNIP アドレスのプールで構成されます。
- **NAT64 IPv6** プレフィックス—アプライアンスに設定された長さ 96 ビット (128-32=96) のグローバル IPv6 プレフィックス。
  注：現在、NetScaler アプライアンスは、すべての NAT 64 ルールで共通に使用される 1 つのプレフィックスしかサポートしていません。

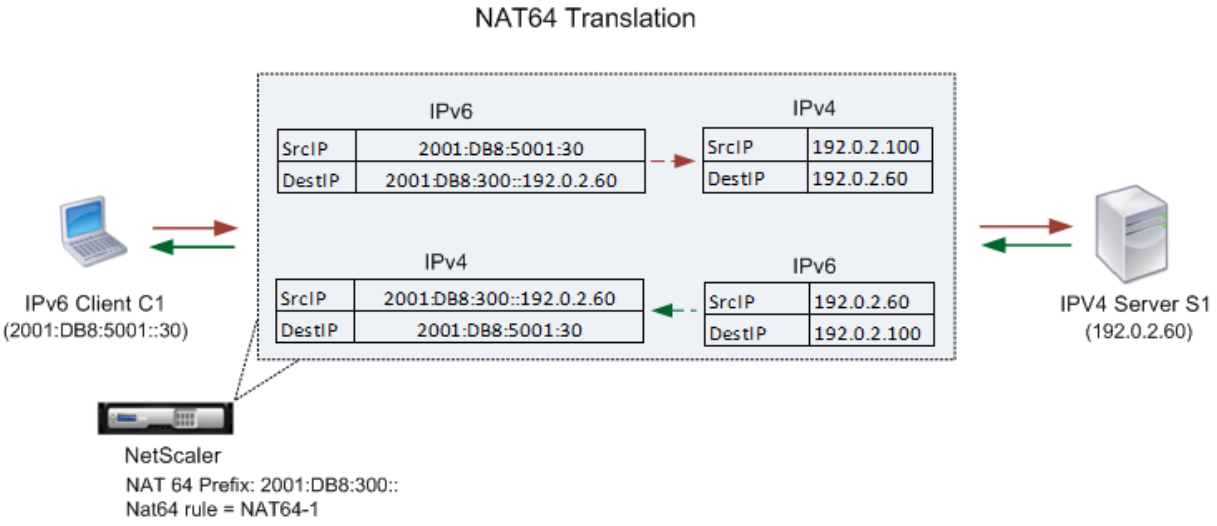
NetScaler アプライアンスは、次の条件がすべて満たされている場合、受信 IPv6 パケットを NAT64 変換の対象と見なします。

- 着信 IPv6 パケットは、NAT64 ルールにバインドされた ACL6 ルールと一致します。
- IPv6 パケットの宛先 IP アドレスは NAT64 IPv6 プレフィックスと一致します。

NetScaler アプライアンスが受信した IPv6 要求パケットが NAT64 ルールで定義されている ACL6 と一致し、パケットの宛先 IP が NAT64 IPv6 プレフィックスと一致する場合、NetScaler アプライアンスは IPv6 パケットを変換対象と見なします。

アプライアンスは、この IPv6 パケットを、NAT64 ルールで定義されているネットプロファイルにバインドされた IP アドレスのいずれかに一致する送信元 IP アドレスと、IPv6 要求パケットの宛先 IPv6 アドレスの最後の 32 ビットで構成される宛先 IP アドレスを持つ IPv4 パケットに変換します。NetScaler アプライアンスは、この特定のフローの NAT64 セッションを作成し、パケットを IPv4 サーバーに転送します。IPv4 サーバーからのその後の応答と IPv6 クライアントからの要求は、特定の NAT64 セッションの情報に基づいて、アプライアンスによってそれに応じて変換されます。

企業が IPv4 アドレスを持つサーバー S1 にサイト www.example.com をホストしている例を考えてみましょう。IPv6 クライアントと IPv4 サーバー S1 間の通信を可能にするために、NetScaler ADC アプライアンス NS1 は、NAT64 ルールと NAT64 プレフィックスを含むステートフル NAT64 構成で展開されます。サーバー S1 のマッピングされた IPv6 アドレスは、NAT64 IPv6 プレフィックス [96 ビット] と IPv4 ソースアドレス [32 ビット] を連結することによって形成されます。このマッピングされた IPv6 アドレスは、DNS サーバで手動で構成されます。IPv6 クライアントは、マッピングされた IPv6 アドレスを DNS サーバーから取得して、IPv4 サーバー S1 と通信します。



次の表に、この例で使用される設定を示します。ステートフル NAT64 変換の例の設定。

次に、この例のトラフィックフローを示します。

1. IPv6 クライアント CL1 は、MAP-SEVR-IPv6 (2001: DB 8:300:: 192.0.2.60) アドレスに要求パケットを送信します。

2. NetScaler アプライアンスは要求パケットを受信します。要求パケットが NAT64 ルールで定義されている ACL6 と一致し、パケットの宛先 IP アドレスが NAT64 IPv6 プレフィックスと一致する場合、NetScaler は IPv6 パケットを変換対象と見なします。
3. アプライアンスは、次のように変換された IPv4 リクエストパケットを作成します。
    - IPv6 リクエストの宛先アドレスから削除された NAT64 プレフィックスを含む宛先 IP アドレスフィールド (SEVR_IPv4 = 192.0.2.60)
    - Netprofile-1 (この場合は 192.0.2.100) にバインドされた IPv4 アドレスのいずれかを含む送信元 IP アドレスフィールド
4. NetScaler アプライアンスはこのフローの NAT64 セッションを作成し、変換された IPv4 要求をサーバー S1 に送信します。
5. IPv64 サーバー S1 は、次のように IPv4 パケットを NetScaler アプライアンスに送信することで応答します。
    - 192.0.2.100 を含む宛先 IP アドレスフィールド
    - SEVR_IPv4 (192.0.2.60) のアドレスを含む送信元 IP アドレスフィールド
6. アプライアンスは IPv4 応答パケットを受信し、すべてのセッションエントリを検索し、IPv6 応答パケットが手順 4 で作成した NAT64 セッションエントリと一致することを確認します。アプライアンスは IPv4 パケットを変換対象と見なします。
7. アプライアンスは、次のように変換された IPv6 応答パケットを作成します。
    - 宛先 IP アドレスフィールド =CLIENT_IPv6=2001: DB 8:5001:30
    - 送信元 IP アドレスフィールド = NAT64 プレフィックス (最初の 96 ビット) と SEVR_IPv4 (最後の 32 ビット) を連結したもの =2001: DB 8:300:: 192.0.2.60
8. アプライアンスは、変換された IPv6 応答をクライアント CL1 に送信します。

ステートフル **NAT64** の制限事項

ステートフル NAT64 には次の制限が適用されます。

- IPv4 オプションの変換はサポートされていません。
- IPv6 ルーティングヘッダーの変換はサポートされていません。
- IPv6 パケットのホップバイホップ拡張ヘッダーの変換はサポートされていません。
- IPv6 パケットの ESP および EH ヘッダーの変換はサポートされていません。
- マルチキャストパケットの変換はサポートされていません。
- ストリーム制御転送プロトコル (SCTP)、データグラム輻輳制御プロトコル (DCCP)、および IPsec のパケットは変換されません。

ステートフル **NAT64** の設定

NetScaler アプライアンスでステートフル NAT64 構成に必要なエンティティを作成するには、次の手順が必要です。

1. アクション ALLOW を含む ACL6 ルールを追加します。
2. 複数の IP アドレスをバインドする ipset を追加します。
3. ネットプロファイルを追加し、それに IP セットをバインドします。IP アドレスを 1 つだけバインドする場合は、ipset エンティティを作成する必要はありません。その場合は、IP アドレスをネットプロファイルに直接バインドしてください。
4. ACl6 ルールとネットプロファイルを NAT 64 ルールにバインドすることを含む NAT64 ルールを追加します。
5. NAT64 IPv6 プレフィックスを追加します。

CLI のプロシージャ

CLI を使用して ACL6 ルールを追加するには:

コマンドプロンプトで入力します。

- add ns acl6 <acl6name> <acl6action> …

CLI を使用して IP セットを追加し、複数の IP をそのセットにバインドするには:

コマンドプロンプトで入力します。

- add ipset <name>
- bind ipset <name> <IPaddress …>

CLI を使用してネットプロファイルを追加するには:

コマンドプロンプトで入力します。

- add netprofile <name> -srcIP <IPaddress or IPset>

CLI を使用して NAT64 ルールを追加するには:

コマンドプロンプトで入力します。

- add nat64 <name> <acl6name> -netProfile <string>

CLI を使用して NAT64 プレフィックスを追加するには:

コマンドプロンプトで入力します。

set ipv6 -natprefix <ipv6_addr *>

-

例:

```
 > add acl6 ACL6-1 ALLOW -srcIPv6 2001:DB8:5001::30
 Done

> apply acls6
 Done

> add ip 192.0.2.100 255.255.255.0  - type SNIP
 Done

> add ip 192.0.2.102 255.255.255.0   - type SNIP
 Done

> add ipset IPset-1
 Done

> bind ipset IPset-1 192.0.2.100 192.0.2.102
IPAddress "192.0.2.100" bound
IPAddress "192.0.2.102" bound
 Done

> add netprofile Netprofile-1 -srcIP IPset-1
 Done

> add nat64 NAT64-1 ACL6-1 -netprofile Netprofile-1
 Done

> set ipv6 -natprefix 2001:DB8:300::/96
 Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して NAT64 ルールを追加するには:

[システム] > [ネットワーク] > [ルート] > [NAT64] に移動して新しい NAT64 ルールを作成するか、既存のルールを編集します。

GUI を使用して NAT64 プレフィックスを追加するには:

[システム] > [ネットワーク] に移動し、[設定] グループで [INAT パラメータの設定] をクリックし、[プレフィックス] パラメータを設定します。

RNAT

August 15, 2023

逆ネットワークアドレス変換（RNAT）では、NetScaler アプライアンスは、サーバーによって生成されたパケット内の送信元 IP アドレスをパブリック NAT IP アドレスに置き換えます。デフォルトでは、アプライアンスは NAT IP アドレスとして SNIP アドレスを使用します。サブネットごとに一意の NAT IP アドレスを使用するようにアプライアンスを構成することもできます。アクセスコントロールリスト（ACL）を使用して RNAT を設定することもできます。ソース IP を使用 (USIP)、サブネット IP を使用 (USNIP)、およびリンクロードバランシング (LLB) の各モードは、RNAT の動作に影響します。統計情報を表示して RNAT を監視できます。

注： NetScaler アプライアンスの RNAT のエフェメラルポート範囲は 1024～65535 です。

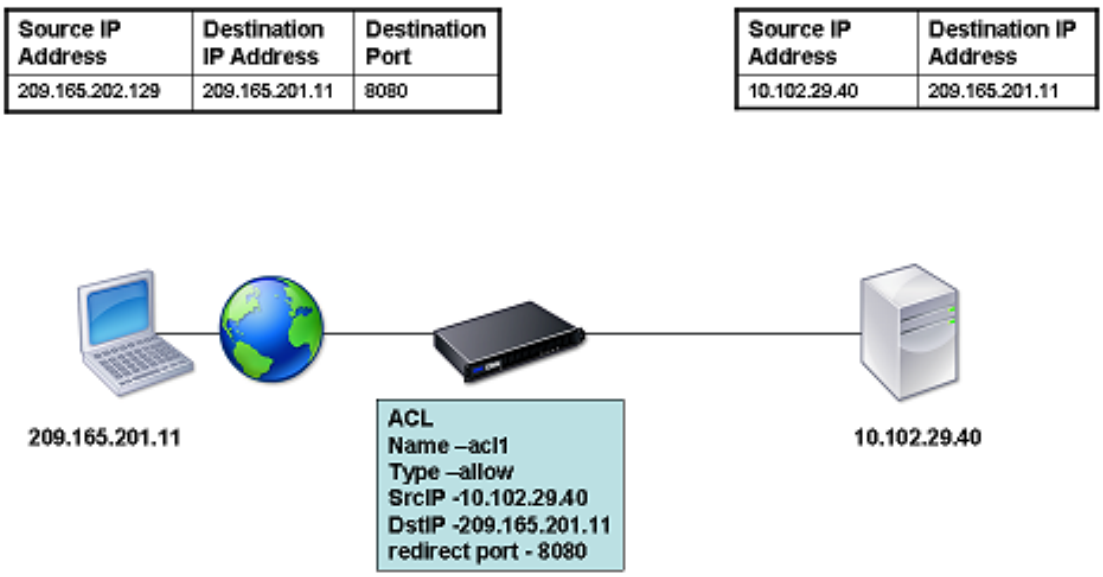
ネットワークアドレスまたは拡張 ACL のいずれかを RNAT エントリの条件として使用できます。

- ネットワークアドレスを使用する。ネットワークアドレスを使用すると、指定されたネットワークから送信されるすべてのパケットで RNAT 処理が実行されます。
- 拡張 **ACL** を使用する。ACL を使用すると、その ACL と一致するすべてのパケットで RNAT 処理が実行されます。ACL と一致するトラフィックに固有の IP アドレスを使用するように NetScaler アプライアンスを構成するには、次の 3 つのタスクを実行する必要があります。

  1. ACL を設定します。
  2. 送信元 IP アドレスと宛先ポートを変更するように RNAT を設定します。
  3. ACL を適用します。

  次の図は、ACL で設定された RNAT を示しています。

  図 1：ACL を使用した RNAT

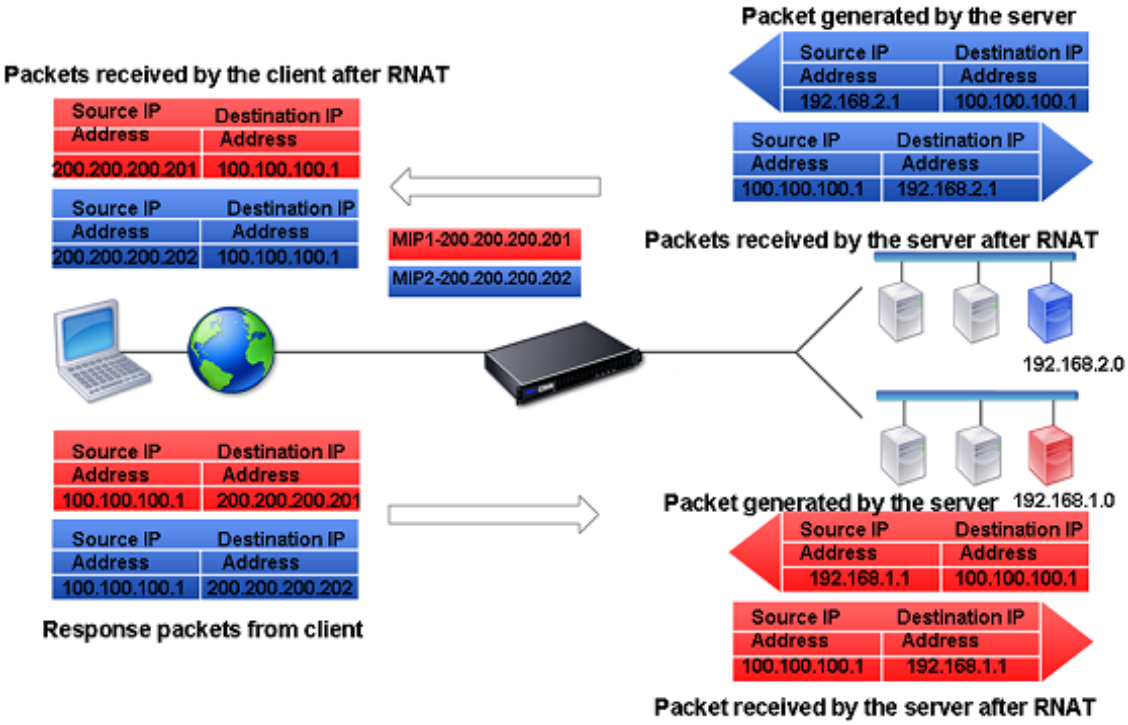


NAT IP アドレスのタイプには、次の基本的な選択肢があります。

- **SNIP** を **NAT IP** アドレスとして使用します。SNIP を NAT IP アドレスとして使用する場合、NetScaler アプライアンスはサーバーで生成されたパケットの送信元 IP アドレスを SNIP に置き換えます。したがって、SNIP アドレスはパブリック IP アドレスでなければなりません。[サブネット IP（USNIP）を使用] モードが有効になっている場合、NetScaler はサブネット IP アドレス（SNIP）を NAT IP アドレスとして使用できます。
- 固有の **IP** アドレスを **NAT IP** アドレスとして使用します。NAT IP アドレスとして固有の IP アドレスを使用する場合、NetScaler アプライアンスはサーバーが生成したパケットの送信元 IP アドレスを、指定された固有の IP アドレスに置き換えます。固有の IP アドレスは、NetScaler が所有するパブリック IP アドレスである必要があります。サブネットに複数の NAT IP アドレスが設定されている場合、NAT IP 選択はラウンドロビンアルゴリズムを使用します。

  この構成を次の図に示します。

  図 2：固有の IP アドレスを NAT IP アドレスとして使用する



はじめに

RNAT ルールを設定する前に、次の点を考慮してください。

- NetScaler アプライアンスで RNAT と Use Source IP（USIP）の両方が構成されている場合、RNAT が優先されます。つまり、RNAT ルールに一致するパケットの送信元 IP アドレスは、RNAT ルールの設定に従って置き換えられます。
- NetScaler アプライアンスがサーバーからのトラフィックに対してリンク負荷分散（LLB）と RNAT の両方を実行するトポロジーでは、アプライアンスはルーターに基づいて送信元 IP アドレスを選択します。LLB 設

定によって、ルータの選択が決まります。LLB の詳細については、リンクロードバランシングを参照してください。

RNAT の設定

以下の手順では、さまざまな条件と種類の NAT IP アドレスを使用する RNAT エントリを作成するための個別のコマンドライン手順を示しています。GUI では、すべてのバリエーションを同じダイアログボックスで設定できるため、GUI ユーザー向けの手順は 1 つだけです。

CLI のプロシージャ

CLI を使用して RNAT ルールを作成するには:

コマンドプロンプトで、ルールを作成して構成を確認するには、次のように入力します。

- `add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))`
- `bind rnat <name> <natIP>@ …`
- `show rnat`

CLI を使用して RNAT ルールを変更または削除するには:

- RNAT ルールを変更するには:

  `set rnat <name> (<aclname> [-redirectPort <port>])`

- RNAT ルールを削除するには、コマンドを入力します。

  `rm rnat <name>`

次のコマンドを使用して構成を確認します。

- `show rnat`

例:

```
A network address as the condition and a SNIP address as the NAT IP
    address:

> add rnat RNAT-1 192.168.1.0 255.255.255.0
Done

A network address as the condition and a unique IP address as the NAT
    IP address:

> add rnat RNAT-2 192.168.1.0 255.255.255.0
Done

> bind rnat RNAT-2 -natip 10.102.29.50
```



```
Done

If instead of a single NAT IP address you specify a range, RNAT entries are created with all the NetScaler-owned IP addresses, except the NSIP, that fall within the range specified:

> add rnat RNAT-3 192.168.1.0 255.255.255.0
Done

> bind rnat RNAT-3 -natip 10.102.29.[50-110]
Done


An ACL as the condition and a SNIP address as the NAT IP address:

> add rnat RNAT-4 acl1
Done

An ACL as a condition and a unique IP address as the NAT IP address:

> add rnat RNAT-4 acl1
Done

> bind rnat RNAT-4 -natip 10.102.29.50
Done

If instead of a single NAT IP address you specify a range, RNAT entries are created with all the NetScaler-owned IP addresses, except the NSIP, that fall within the range specified:

> add rnat RNAT-5 acl1
Done

> bind rnat RNAT-5 -natip 10.102.29.[50-70]
Done

<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して RNAT エントリを作成するには:

[システム] > [ネットワーク] > [**NAT**] に移動し、[**RNAT**] タブをクリックして新しい RNAT ルールを追加するか、既存のルールを編集します。

RNAT をモニタリング

RNAT 統計情報を表示して、IP アドレス変換に関連する問題をトラブルシューティングできます。

次の表は、RNAT および RNAT IP に関連する統計情報をまとめたものです。

統計量	説明
受信バイト数	RNAT セッション中に受信したバイト数
送信バイト数	RNAT セッション中に送信されたバイト数
受信したパケット	RNAT セッション中に受信したパケット
送信されたパケット	RNAT セッション中に送信されるパケット
Syn sent	RNAT セッション中に送信される接続要求
現在のセッション	現在アクティブな RNAT セッション

CLI を使用して RNAT 統計を表示するには:

コマンドプロンプトで入力します。

- **stat rnat**

例**:**

```
> stat rnat

RNAT summary
                                Rate (/s)            Total
Bytes Received                    0                    0
Bytes Sent                        0                    0
Packets Received                  0                    0
Packets Sent                      0                    0
Syn Sent                          0                    0
Current RNAT sessions            --                    0
 Done
>
<!--NeedCopy-->
```

GUI を使用して RNAT を監視するには:

[** システム] > [ネットワーク] > [**NAT**] に移動し、[**RNAT**] タブをクリックして、[統計] をクリックします。******

RNAT6 を設定して下さい

IPv6 パケットの逆ネットワークアドレス変換 (RNAT) ルールは RNAT6 と呼ばれます。サーバーによって生成された IPv6 パケットが RNAT6 ルールで指定された条件と一致すると、アプライアンスは IPv6 パケットの送信元 IPv6 アドレスを設定された NAT IPv6 アドレスに置き換えてから宛先に転送します。NAT IPv6 アドレスは、NetScaler が所有する SNIP6 または VIP6 アドレスの 1 つです。

RNAT6 ルールを設定する場合、条件として IPv6 プレフィックスまたは ACL6 のいずれかを指定できます。

- **IPv6** ネットワークアドレスを使用する。IPv6 プレフィックスを使用する場合、アプライアンスは IPv6 アドレスがプレフィックスと一致する IPv6 パケットに対して RNAT 処理を実行します。
- **ACL6** を使用する。ACL6 を使用する場合、アプライアンスは ACL6 で指定された条件と一致する IPv6 パケットに対して RNAT 処理を実行します。

NAT IP アドレスを設定するには、次のいずれかのオプションがあります。

- RNAT6 ルールには、NetScaler が所有する SNIP6 アドレスと VIP6 アドレスのセットを指定します。NetScaler アプライアンスは、このセットの IPv6 アドレスのいずれか 1 つを各セッションの NAT IP アドレスとして使用します。選択はラウンドロビンアルゴリズムに基づいており、セッションごとに行われます。
- RNAT6 ルールには、NetScaler が所有する SNIP6 アドレスまたは VIP6 アドレスを指定しないでください。NetScaler アプライアンスは、NetScaler が所有する SNIP6 または VIP6 アドレスのいずれかを NAT IP アドレスとして使用します。選択は、RNAT ルールに一致する IPv6 パケットの宛先となるネクストホップネットワークに基づいて行われます。

CLI のプロシージャ

CLI を使用して RNAT6 ルールを作成するには:

コマンドプロンプトで、ルールを作成して構成を確認するには、次のように入力します。

- **add rnat6** <name> (<network> | (<acl6name> [-**redirectPort** \]))
- **bind rnat6** <name> <natIP6>@ …
- **show rnat6**

CLI を使用して RNAT6 ルールを変更または削除するには:

- 条件が **ACL6** である **RNAT6** ルールを変更するには、set rnat6 コマンドを入力し、続けて **RedirectPort** パラメータに新しい値を入力します。 <name>
- <name>RNAT6 ルールを削除するには、**clear rnat6** コマンドを入力します。

GUI のプロシージャ

GUI を使用して RNAT6 ルールを設定するには:

[システム] > [ネットワーク] > [**NAT**] に移動し、[**RNAT6**] タブをクリックして新しい RNAT6 ルールを追加するか、既存のルールを編集します。

RNAT 6 モニター

RNAT6 機能に関連する統計情報を表示して、パフォーマンスを監視したり、RNAT6 機能に関連する問題をトラブルシューティングしたりできます。RNAT6 ルールまたは特定の RNAT6 ルールの統計の概要を表示できます。統計カ

ウンターには、NetScaler アプライアンスが最後に再起動されてからのイベントが反映されます。NetScaler アプライアンスを再起動すると、これらのカウンタはすべて 0 にリセットされます。

RNAT6 機能に関連する統計カウンタの一部を以下に示します。

- 受信バイト数 -RNAT6 セッション中に受信した合計バイト数。
- 送信バイト数 -RNAT6 セッション中に送信されたバイトの総数。
- 受信パケット -RNAT6 セッション中に受信したパケットの総数。
- 送信されたパケット -RNAT6 セッション中に送信されたパケットの総数。
- **Syn** 送信済み -RNAT6 セッション中に送信された接続リクエストの総数
- 現在のセッション -現在アクティブな RNAT6 セッション

CLI を使用してすべての RNAT6 ルールの要約統計を表示するには:

コマンドプロンプトで入力します。

- **stat rnat6**

CLI を使用して指定した RNAT6 ルールの統計情報を表示するには:

コマンドプロンプトで入力します。

- **stat rnat6** [\<rnat6 rule name>]

GUI を使用して RNAT6 統計情報を表示するには、次の手順を実行します。

[** システム] > [ネットワーク] > [**NAT] に移動し、[**RNAT6**] タブをクリックして、[統計] をクリックします。 **

```
> stat rnat6

RNAT6 summary

                           Rate (/s)                Total

Bytes Received              178                 20644

Bytes Sent                  178                 20644

Packets Received             5                   401

Packets Sent                 5                   401

Syn Sent                     0                     2

Current RNAT6 sessions      --                     1

Done

<!--NeedCopy-->
```


RNAT ログエントリに開始時刻と接続終了理由を記録

NetScaler アプライアンスは、RNAT に関連する問題の診断やトラブルシューティングのために、終了するたびに RNAT セッションをログに記録します。

RNAT セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元となる NetScaler 所有の IP アドレス（NSIP アドレスまたは SNIP アドレス）
- ログ作成のタイムスタンプ
- RNAT セッションのプロトコル
- 送信元 IP アドレス
- RNAT IP アドレス
- 宛先 IP アドレス
- RNAT セッションの開始時刻
- RNAT セッションの終了時刻
- この RNAT セッションで NetScaler アプライアンスによって送信された合計バイト数
- この RNAT セッションで NetScaler アプライアンスが受信した合計バイト数
- RNAT セッションの終了の理由。NetScaler アプライアンスは、アプライアンスの TCP プロキシを使用しない（TCP プロキシが無効になっている）TCP RNAT セッションの終了理由を記録します。TCP RNAT セッションについてログに記録されるクローズ理由の種類は次のとおりです。
  - **TCP** フィン。送信元または宛先デバイスのいずれかから TCP FIN が送信されたため、RNAT セッションが終了しました。
  - **TCP RST**。送信元デバイスまたは宛先デバイスのいずれかから TCP リセットが送信されたため、RNAT セッションが終了しました。
  - タイムアウト。RNAT セッションがタイムアウトしました。

次の表は、RNAT セッションのログエントリのサンプルを示しています。

エントリのタイプ	サンプルログエントリ
UDP RNAT セッションのサンプルログエントリ	Dec 1 15:28:12 10.102.53.114 12/01/2015:15:28:12 GMT 0-PPE-0 : default UDP NAT_OTHERCONN_DELINK 154 0 : Source 1.2.2.5:23431 - Destination 192.168.123.122:22 - NatIP 192.168.123.1:4045 - Destination 192.168.123.122:22 - Start Time 12/01/2015:15:26:58 GMT - Delink Time 12/01/2015:15:28:12 GMT - Total_bytes_send 2511 - Total_bytes_recv 3725

エントリのタイプ	サンプルログエントリ
TCP RNAT セッションのサンプルログエントリ。ログエントリには、TCP リセットによりセッションが終了したことが示されます	Dec 1 15:29:59 10.102.53.114 12/01/2015:15:27:59 GMT 0-PPE-0 : default TCP NAT_OTHERCONN_DELINK 152 0 : Source 1.2.2.5:33826 - Destination 192.168.123.122:22 - NatIP 192.168.123.1:2384 - Destination 192.168.123.122:22 - Start Time 12/01/2015:15:27:40 GMT - Delink Time 12/01/2015:15:27:59 GMT - Total_bytes_send 2147 - Total_bytes_recv 3257 - Closure Reason TCP RST
TCP RNAT セッションのサンプルログエントリ。ログエントリには、セッションがタイムアウトしたことが示されます。	Dec 1 15:30:12 10.102.53.114 12/01/2015:15:30:12 GMT 0-PPE-0 : default TCP NAT_OTHERCONN_DELINK 155 0 : Source 1.2.2.5:64976 - Destination 192.168.123.115:22 - NatIP 192.168.123.1:19636 - Destination 192.168.123.115:22 - Start Time 12/01/2015:15:27:25 GMT - Delink Time 12/01/2015:15:30:12 GMT - Total_bytes_send 0 - Total_bytes_recv 0 - Closure Reason TIMEOUT

RNAT のステートフル接続フェイルオーバー

接続フェイルオーバーは、分散環境に展開されているアプリケーションへのアクセスの中断を防ぐのに役立ちます。NetScaler アプライアンスは、NetScaler 高可用性（HA）セットアップの RNAT ルールに関連する接続のステートフル接続フェイルオーバーをサポートするようになりました。HA セットアップでは、接続フェイルオーバー (または接続ミラーリング) とは、フェイルオーバーが発生しても、確立された TCP または UDP 接続をアクティブに保つプロセスを指します。

プライマリアプライアンスはセカンダリアプライアンスにメッセージを送信して、RNAT 接続に関する現在の情報を同期します。セカンダリアプライアンスは、フェイルオーバーが発生した場合にのみこの接続情報を使用します。フェイルオーバーが発生すると、新しいプライマリ NetScaler アプライアンスはフェイルオーバー前に確立された接続に関する情報を保持するため、フェイルオーバー後もそれらの接続を引き続き提供します。クライアントから見ると、このフェイルオーバーは透過的です。移行期間中、クライアントとサーバーで短時間の中断と再送信が発生する可能性があります。

接続フェイルオーバーは、RNAT ルールごとに有効にできます。RNAT ルールで接続フェイルオーバーを有効にするには、CLI または GUI を使用して特定の RNAT ルールの ConnFailover (接続フェイルオーバー) パラメータを有効にします。

CLI を使用して RNAT ルールの接続フェイルオーバーを有効にするには:

コマンドプロンプトで入力します。

- `set rnat <name> -connfailover (ENABLED | DISABLED)`
- `show rnat`

GUI を使用して RNAT ルールの接続フェイルオーバーを有効にするには:

1. Navigate to **System** > **Network** > **NATs**, and then click the **RNAT** tab.
2. 新しい RNAT 規則の追加中、または既存の規則の編集中に [接続フェールオーバー] を選択します。

サーバーへの **RNAT** 接続用の送信元ポートを予約する

1 つ以上の RNAT IP アドレスと Use Proxy port パラメーターが無効になっている RNAT 構成にヒットするリクエストの場合、NetScaler アプライアンスは RNAT IP アドレスと RNAT リクエストの送信元ポートのいずれかをサーバーに接続します。13.0 47.x ビルドより前のバージョンでは、同じソースポートが他の接続で既に使用されている場合、サーバーへの RNAT 接続 (RNAT クライアントのソースポートを使用) は失敗します。

- 送信元ポートが **1024** 未満です。デフォルトでは、NetScaler アプライアンスは NetScaler が所有する IP アドレス（RNAT IP アドレスを含む）の最初の 1024 ポートを予約します。13.0 47.x ビルドより前のバージョンでは、RNAT 要求の送信元ポートが 1024 以下の場合、サーバーへの RNAT 接続 (RNAT クライアントのソースポートを使用) は失敗します。13.0 47.x ビルドでは、RNAT 要求の送信元ポートが 1024 以下であっても、サーバーへの RNAT 接続 (RNAT クライアントのソースポートを使用) は成功します。
- 送信元ポートが **1024** を超えています。13.0 47.x ビルドより前のバージョンでは、同じソースポートが他の接続で既に使用されている場合、サーバーへの RNAT 接続 (RNAT クライアントのソースポートを使用) は失敗します。13.0 47.x ビルドでは、RNAT 設定の一部として、`Retain Source Port range` (`retainsourceportrange`) パラメータで RNAT クライアントソースポートの範囲を指定できます。NetScaler アプライアンスは、RNAT IP アドレス上のこれらの RNAT クライアント送信元ポートを、サーバーへの RNAT 接続にのみ使用するように予約します。

RNAT セッションの削除

不要または非効率的な RNAT セッションを NetScaler アプライアンスから削除できます。アプライアンスは、これらのセッションに割り当てられたリソース（NAT IP アドレスのポート、メモリなど）をすぐに解放し、リソースを新しいセッションで使用できるようにします。アプライアンスは、削除されたセッションに関連する後続のパケットもすべてドロップします。NetScaler アプライアンスからすべてまたは選択した RNAT セッションを削除できます。

CLI を使用してすべての RNAT セッションをクリアするには:

コマンドプロンプトで入力します。

- フラッシュランセッション

CLI を使用して選択的な RNAT セッションをクリアするには:

コマンドプロンプトで入力します。

- **flush rnatsession** ((-**network** <ip_addr> -**netmask** <netmask>) | -**natIP** <ip_addr> | -**aclname** <string>)

GUI を使用してすべての RNAT セッションまたは選択的な RNAT セッションをクリアするには:

1. [システム] > [ネットワーク] > [**NAT**] に移動し、[**RNAT**] タブをクリックします。
2. アクションメニューで、「**Flash RNAT Sessions** 」をクリックして、すべてまたは一部の RNAT セッションを削除します（たとえば、特定の RNAT IP、特定のネットワークまたは ACL ベースの RNAT ルールに属する RNAT セッションを削除するなど）。

構成例:

```
    Clear all RNAT sessions existing on a NetScaler appliance

    > flush rnatsession

    Done

    Clear all RNAT sessions belonging to network based RNAT rules that
       has 203.0.113.0/24 network as the matching condition.

    > flush rnatsession -network 203.0.113.0 -netmask 255.255.255.0

    Done

    Clear all RNAT sessions with RNAT IP 192.0.2.90.

    > flush rnatsession -natIP 192.0.2.90

    Done

    Clear all RNAT sessions belonging to ACL based RNAT rules that has
       ACL-RNAT-1 as the matching condition.

    > flush rnatsession -aclname ACL-RNAT-1

    Done
<!--NeedCopy-->
```

プレフィックスベースの **IPv6-IPv4** 変換の構成

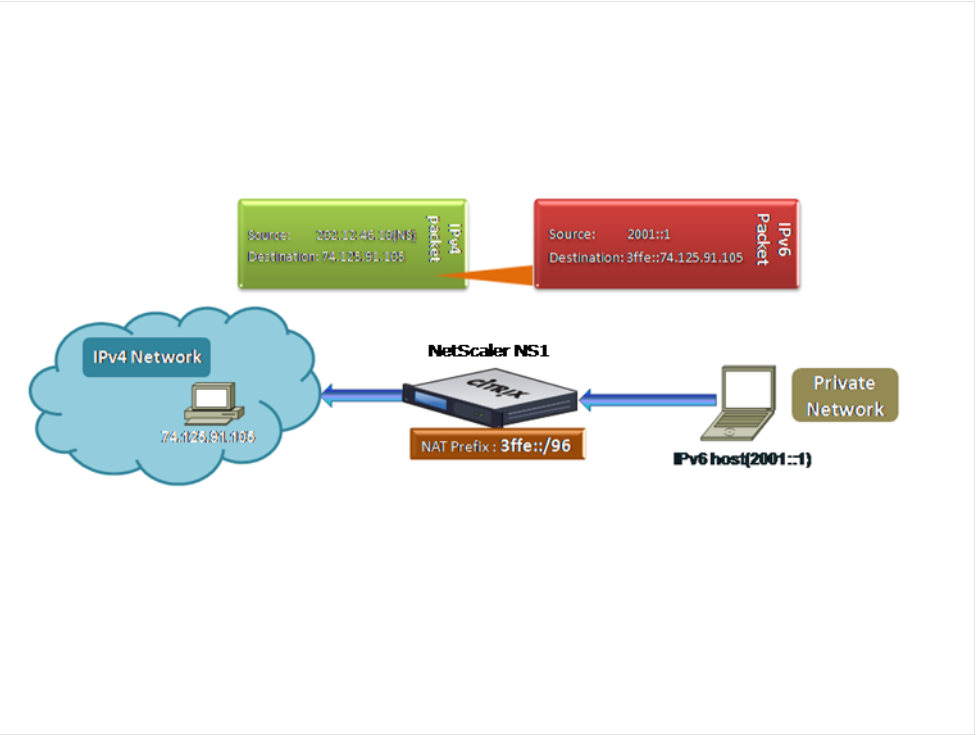
August 15, 2023

プレフィックスベースの変換は、NetScaler アプライアンスで構成された IPv6 プレフィックスを使用して、プライベート IPv6 サーバーから送信されたパケットを IPv4 パケットに変換するプロセスです。このプレフィックスの長さは 96 ビット (128-32=96) です。IPv6 サーバーは、IPv6 パケットの宛先 IP アドレスフィールドの最後の 32 ビットに IPv4 サーバーまたはホストの宛先 IP アドレスを埋め込みます。宛先 IP アドレスフィールドの最初の 96 ビットは IPv6 NAT プレフィックスとして設定されます。

NetScaler アプライアンスは、すべての受信 IPv6 パケットの宛先 IP アドレスの最初の 96 ビットを、構成されたプレフィックスと比較します。一致する場合、NetScaler アプライアンスは IPv4 パケットを生成し、一致した IPv6 パケットの宛先 IP アドレスの最後の 32 ビットとして宛先 IP アドレスを設定します。このプレフィックス宛ての IPv6 パケットは、NetScaler が IPv6-IPv4 変換を行うように NetScaler にルーティングする必要があります。

次の図では、3ffe:: /96 が NetScaler NS1 の IPv6 NAT プレフィックスとして構成されています。IPv6 ホストは、宛先 IP アドレス 3ffe:: 74.125.91.105 の IPv6 パケットを送信します。NS1 は、すべての着信 IPv6 パケットの宛先 IP アドレスの最初の 96 ビットを、設定されたプレフィックスと比較し、一致させます。次に、NS1 は IPv4 パケットを生成し、宛先 IP アドレスを 74.125.91.105 に設定します。

図 1: IPv6-IPv4 プレフィックスベースの変換



CLI を使用してプレフィックスベースの IPv6-IPv4 変換を設定するには:

コマンドプロンプトで入力します。

```
set ipv6 [-natprefix <ipv6_addr                    *>]
```

-

- show ipv6

例**:**

```
> set ipv6 -natprefix 3ffe::/96
 Done
<!--NeedCopy-->
```

GUI を使用してプレフィックスベースの IPv6-IPv4 変換を設定するには:

[システム] > [ネットワーク] に移動し、[設定] グループで [INAT パラメータの設定] をクリックし、[プレフィックス] パラメータを設定します。

IP プレフィックス NAT

August 15, 2023

NetScaler アプライアンスは、アプライアンスで受信したパケットの完全なアドレスではなく、送信元 IP アドレスの一部の変換をサポートします。IP プレフィックス NAT には、送信元 IP アドレスの 1 つまたは複数のオクテットまたはビットの変更が含まれます。

NetScaler アプライアンスは、ANY、UDP、DNS、TCP、および HTTP のタイプの負荷分散構成で IP プレフィックス NAT をサポートします。

使用事例：**NetScaler** アプライアンスと最適化デバイスの導入におけるクライアントのゾーン化

IP プレフィックス NAT は、NetScaler アプライアンスと最適化デバイス（Citrix ByteMobile など）を含む展開環境で非常に役立ちます。このタイプの展開では、地理的に異なるクライアントネットワークがあり、同じネットワークアドレスを共有しています。NetScaler アプライアンスは、宛先に転送する前に、各クライアントネットワークから受信したトラフィックを最適化デバイスに送信する必要があります。

デバイスは最適化されたトラフィックを NetScaler アプライアンスに送り返します。最適化要件はクライアントネットワークからのトラフィックごとに異なるため、最適化デバイスは受信する各パケットのクライアントネットワークを認識する必要があります。解決策は、VLAN を使用して各クライアントネットワークのトラフィックを異なるゾーンに分離することです。ゾーンごとに異なる設定の IP プレフィックス NAT が設定されます。NetScaler アプライアンスはすべてのパケットの送信元 IP アドレスの最後のオクテットを変換し、変換されたオクテット値はゾーンごとに異なります。

ネットワークアドレス 192.0.2.0/24 を共有する Z1 と Z2 の 2 つのゾーンの例を考えてみましょう。NetScaler アプライアンスでは、natrule-1 と natrule-2 という名前の IP プレフィックス NAT エンティティがこれら 2 つのゾーンに設定されています。アプライアンスが Z1 からパケットを転送する前に、natrule-1 はパケットの送信元 IP アドレスの最後のオクテットを 100 に変換します。同様に、Z2 からのパケットの場合、natrule-2 は送信元 IP アドレスの最後のオクテットを 200 に変換します。ゾーン Z1 の CL1-Z1 とゾーン Z2 の CL1-Z2 の 2 つのクライアント

で、それぞれ IP アドレスが 192.0.2.30 の場合、NetScaler アプライアンスは CL1-Z1 のパケットの送信元 IP アドレスを 100.0.2.30 に、CL1-Z2 のパケットの送信元 IP アドレスを 200.0.2.30 に変換します。NetScaler アプライアンスが変換されたパケットを送信する最適化デバイスは、パケットのソース IP アドレスを使用してゾーンを認識するように構成されているため、パケットの発信元のゾーンに設定された適切な最適化が適用されます。

構成の手順

IP プレフィックス NAT の設定は、次の手順で構成されます。

- ネットプロファイルを作成し、ネットプロファイルの **NAT Rule** パラメータを設定します。NAT ルールは 2 つの IP アドレスと 1 つのネットマスクを指定します。最初の IP アドレス (IP Address パラメータで指定) は、2 番目の IP アドレス (IP Rewrite パラメータで指定) で変換される送信元 IP アドレスです。ネットマスクは、送信元 IP アドレスのうち、2 番目の IP アドレスの同じ部分に変換される部分を指定します。
- ネットプロファイルを負荷分散仮想サーバーまたはサービスにバインドします。NAT ルールが設定されたネットプロファイルは、ANY、UDP、DNS、TCP、および HTTP タイプの仮想サーバーまたはサービスにバインドできます。ネットプロファイルを仮想サーバーまたはサービスにバインドすると、NetScaler アプライアンスは仮想サーバーまたはサービスに関連する受信パケットの送信元 IP アドレスと NAT ルール設定を照合します。次に、NetScaler は NAT ルールに一致するパケットに対して IP プレフィックス NAT を実行します。

コマンドラインを使用して IP プレフィックス NAT 変換を設定するには:

コマンドプロンプトで入力します。

- **bind netProfile** <name> (-**natRule** <ip_addr> <netmask> <rewriteIp>)
- **show netprofile** <name>

GUI を使用して IP プレフィックス NAT を設定するには:

1. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
2. NetProfiles を追加または変更するときに、NAT ルールで次のパラメータを設定します。
    - IP アドレス
    - ネットマスク
    - リライト IP

設定例

次の構成例では、ネットプロファイル PARTIAL-NAT-1 には IP プレフィックス NAT 設定があり、ANY タイプの負荷分散仮想サーバー LBVS-1 にバインドされています。LBVS-1 で 192.0.0.0/8 から受信したパケットの場合、NetScaler アプライアンスはパケットの送信元 IP アドレスの最後のオクテットを 100 に変換します。たとえば、送信元 IP アドレスが 192.0.2.30 のパケットが LBVS-1 で受信された場合、NetScaler アプライアンスは、送信元 IP アドレスを 100.0.2.30 に変換してから、バインドされたサーバーの 1 つを送信します。


```
> add netprofile PARTIAL-NAT-1
Done

> bind netprofile PARTIAL-NAT-1  - natrule 192.0.0.0 255.0.0.0 100.0.0.0
Done

> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1
Done
<!--NeedCopy-->
```

スタティック **ARP**

August 15, 2023

ARP テーブルにスタティック ARP エントリを追加したり、ARP テーブルからスタティック ARP エントリを削除したりできます。エントリを追加したら、設定を確認する必要があります。スタティック ARP エントリを作成した後に IP アドレス、ポート、または MAC アドレスが変更された場合は、スタティックエントリを削除するか、手動で調整する必要があります。したがって、必要な場合を除いて、スタティック ARP エントリを作成することはお勧めしません。

CLI を使用してスタティック ARP エントリを追加するには:

コマンドプロンプトで入力します。

- **add arp -IPAddress** <ip_addr> -**mac**<mac_addr> -**ifnum** <interface_name>
- **show arp** <IPAddress>

例**:**

```
> add arp -ip 10.102.29.6 -mac 00:24:e8:73:ca:ec -ifnum 1/1
 Done
<!--NeedCopy-->
```

CLI を使用してスタティック ARP エントリを削除するには:

コマンドプロンプトで、**rm arp** コマンドと IP アドレスを入力します。

GUI を使用してスタティック ARP エントリを追加するには:

[システム] > [ネットワーク] > [**ARP** テーブル] に移動し、スタティック ARP エントリを追加します。

スタティック **ARP** エントリに **VLAN** を指定する

スタティック ARP エントリでは、宛先デバイスにアクセスできる VLAN を指定できます。この機能は、スタティック ARP エントリで指定されたインターフェイスが複数のタグ付き VLAN の一部であり、その VLAN の 1 つを経由し

て宛先にアクセスできる場合に便利です。NetScaler アプライアンスは、静的 ARP エントリと一致する送信パケットに、指定された VLAN ID を含めます。ARP エントリに VLAN ID を指定せず、指定したインターフェイスが複数のタグ付き VLAN の一部である場合、アプライアンスはインターフェイスのネイティブ VLAN を ARP エントリに割り当てます。

たとえば、NetScaler インターフェイス 1/2 がネイティブ VLAN2 の一部であり、タグ付き VLAN3 と 4 の一部である場合、ネットワークデバイス A の静的 ARP エントリを追加するとします。このエントリは VLAN 3 の一部であり、インターフェイス 1/2 からアクセス可能です。ネットワークデバイス A の ARP エントリに VLAN 3 を指定する必要があります。NetScaler アプライアンスは、ネットワークデバイス A 宛のすべてのパケットにタグ付き VLAN 3 を含めて、インターフェイス 1/2 から送信します。

VLAN ID を指定しない場合、NetScaler アプライアンスは ARP エントリにネイティブ VLAN 2 を割り当てます。デバイス A 宛てのパケットは、デバイス A の VLAN であるタグ付き VLAN 3 が指定されていないため、ネットワークパスでドロップされます。

CLI を使用してスタティック ARP エントリに VLAN を指定するには:

コマンドプロンプトで入力します。

- **add arp -IPAddress** <ip_addr> **-mac**<mac_addr> **-ifnum** <interface_name> [-**vlan** \<positive_integer>]
- **show arp** <IPAddress>

例**:**

```
1 > add arp -ip 198.51.100.91 -mac 36:db:4b:f6:12:15 -ifnum 1/2 -vlan 3
2 Done
3 <!--NeedCopy-->
```

ダイナミック **ARP** エントリのタイムアウトの設定

August 15, 2023

動的に学習した ARP エントリのエージングタイム（タイムアウト値）をグローバルに設定できます。新しい値は、新しい値の設定後に動的に学習される ARP エントリにのみ適用されます。以前に存在していた ARP エントリは、以前に設定されたエージングタイムが経過すると期限切れになります。ARP タイムアウト値は 1 ～1200 秒の範囲で指定できます。

CLI を使用してダイナミック ARP エントリのタイムアウトを設定するには:

コマンドプロンプトで入力します。

- **set arpparam** -**timeout** <positive_integer>]
- **show arpparam**

例:

```
> set arpparam -timeout 500
 Done
<!--NeedCopy-->
```

CLI を使用してダイナミック ARP エントリのタイムアウトをデフォルト値に設定するには:

コマンドプロンプトで入力します。

- **unset arpparam**
- **show arpparam**

例:

```
> unset arpparam
 Done
<!--NeedCopy-->
```

GUI を使用してダイナミック ARP エントリのタイムアウトを設定するには:

[システム] > [ネットワーク] に移動し、[設定] グループで [**ARP** グローバルパラメータの構成] をクリックし、[**ARP** テーブルエントリタイムアウト] パラメータを設定します。

ネイバーディスカバリー

August 15, 2023

ネイバーディスカバリー (ND) は IPv6 の最も重要なプロトコルの 1 つです。アドレス解決プロトコル (ARP)、インターネット制御メッセージプロトコル (ICMP)、およびルーターディスカバリーの機能を組み合わせたメッセージベースのプロトコルです。ND を使用すると、ノードはリンク層アドレスをアドバタイズし、隣接ノードの MAC アドレスまたはリンク層アドレスを取得できます。このプロセスは近傍探索プロトコル (ND6) によって実行されます。

近傍探索では次の機能を実行できます。

- ルーター検出: ホストが接続されたリンク上のローカルルーターを検出し、デフォルトルーターを自動的に構成できるようにします。
- プレフィックス検出: ホストがローカル宛先のネットワークプレフィックスを検出できるようにします。

  注: NetScaler アプライアンスはプレフィックス検出をサポートしていません。
- パラメータ検出: ホストが MTU やアウトバウンドトラフィックのデフォルトホップ制限など、追加の動作パラメータを検出できるようにします。
- アドレス自動構成:DHCPv6 などのステートフルアドレス構成サービスの有無にかかわらず、ホストがインターフェイスの IP アドレスを自動的に構成できるようにします。NetScaler は、グローバル IPv6 アドレスのアドレス自動構成をサポートしていません。

- アドレス解決:IPv4 の ARP と同等で、ノードが隣接ノードの IPv6 アドレスをリンク層アドレスに変換できるようにします。
- ネイバー到達不能検知: ノードがネイバーの到達可能性状態を判断できるようにします。
- 重複アドレス検出:NSIP アドレスが隣接ノードですでに使用されているかどうかをノードが判断できるようにします。
- リダイレクト:IPv4 ICMP リダイレクトメッセージと同等で、ルーターがホストをより適切なファーストホップ IPv6 アドレスにリダイレクトして宛先に到達できるようにします。

注: NetScaler アプライアンスは IPv6 リダイレクトをサポートしていません。

構成の手順

ネイバーディスカバリの設定は、次のタスクで構成されます。

- IPv6 ネイバーの追加
- （オプション）IPv6 ネイバーの削除

CLI のプロシージャ

CLI を使用して IPv6 ネイバーを追加するには:

コマンドプロンプトで入力します。

- **add nd6** <neighbor> <mac> <ifnum> [-**vlan** \<integer>]
- **sh nd6**

例**:**

```
1  > add nd6 2001::1 00:04:23:be:3c:06 1/1  - vlan 1
2   Done
3
4  > show nd6
5     Neighbor                  MAC-Address(Vlan, Interface)        State
                TIME
6     --------                  ----------------------------        -----
                --------
7  1) ::1                       00:d0:68:0b:58:da(   1,  LO/1)    REACHABLE
          PERMANENT
8  2) fe80::2d0:68ff:fe0b:58da 00:d0:68:0b:58:da(   1,  LO/1)    REACHABLE
          PERMANENT
9  3) 2001::1                   00:04:23:be:3c:06(   1,   1/1)    REACHABLE
            STATIC
10  Done
11 <!--NeedCopy-->
```


CLI を使用してネイバーディスカバリエントリを削除するには:

コマンドプロンプトで入力します。

- **rm nd6** <Neighbor> -**vlan** <VLANID>

例**:**

```
1   rm nd6 3ffe:100:100::1 -vlan 1
2 <!--NeedCopy-->
```

CLI を使用してすべてのネイバーディスカバリエントリを削除するには:

コマンドプロンプトで入力します。

- **clear nd6**

GUI のプロシージャ

GUI を使用して IPv6 ネイバーを追加するには:

[システム] > [ネットワーク] > [**IPv6** ネイバー] に移動し、新しい IPv6 ネイバーを追加します。

GUI を使用してネイバーディスカバリエントリを削除するには:

[システム] > [ネットワーク] > [**IPv6** ネイバー**]** に移動し、**IPv6** ネイバーを削除します。

GUI を使用してすべてのネイバーディスカバリエントリを削除するには:

[** システム] > [ネットワーク] > [**IPv6** ネイバー] に移動し、[クリア] をクリックします。******

IP トンネル

August 15, 2023

IP トンネルは、ルーティングパスのない 2 つのネットワーク間で、カプセル化技術を使用して作成できる通信チャネルです。2 つのネットワーク間で共有されるすべての IP パケットは、別のパケットにカプセル化され、トンネル経由で送信されます。

NetScaler ADC アプライアンスは、次の方法で IP トンネリングを実装します。

- カプセル化ツールとしての **NetScaler ADC**（**DSR** モードによる負荷分散）: さまざまな国に複数のデータセンターを持つ組織を考えてみましょう。NetScaler ADC はある場所にあり、バックエンドサーバーは別の国にあります。基本的に、NetScaler ADC とバックエンドサーバーは異なるネットワーク上にあり、ルーターを介して接続されています。

この NetScaler ADC でダイレクトサーバーリターン（DSR）を構成すると、ソースサブネットから送信されるパケットは NetScaler ADC によってカプセル化され、ルーターとトンネルを介して適切なバックエンドサーバーに送信されます。バックエンドサーバーは、パケットが NetScaler ADC を通過することを許可せずに、パケットをカプセル化解除してクライアントに直接応答します。

- デカプスレーターとしての **NetScaler**：複数のデータセンターがあり、それぞれに NetScaler とバックエンドサーバーがある組織を考えてみましょう。パケットがデータセンター A からデータセンター B に送信される場合、通常、ルーターや別の NetScaler ADC などの仲介者を介して送信されます。NetScaler ADC はパケットを処理し、パケットをバックエンドサーバーに転送します。ただし、カプセル化されたパケットを送信する場合、NetScaler ADC はパケットをバックエンドサーバーに送信する前にパケットをカプセル化解除できる必要があります。NetScaler ADC をデカプセル化装置として機能させるには、ルーターと NetScaler ADC の間にトンネルを追加します。追加のヘッダー情報を含むカプセル化されたパケットが NetScaler ADC に到達すると、データパケットはカプセル化解除されます。つまり、追加のヘッダー情報が削除され、パケットは適切なバックエンドサーバーに転送されます。

  NetScaler ADC は、負荷分散機能のデカプセル化機能としても使用できます。特に、仮想サーバー上の接続数がしきい値を超え、すべての新しい接続がバックアップ仮想サーバーに転送されるシナリオではそうです。

IP トンネル機能は、NetScaler プレミアムエディションライセンスで使用できます。NetScaler エディションのライセンスと NetScaler ADC 機能マトリックスの詳細については、NetScaler エディションのデータシートを参照してください。

IP トンネルの設定

NetScaler ADC アプライアンスでの IP トンネルの構成は、IP トンネルエンティティの作成で構成されます。IP トンネルエンティティは、ローカルおよびリモートのトンネルエンドポイント IP アドレスと、IP トンネルに使用するプロトコルを指定します。

注**:** クラスタ設定で IP トンネルを設定する場合、ローカル IP アドレスはストライプされた SNIP アドレスでなければなりません。

CLI のプロシージャ

CLI を使用して IP トンネルを作成するには:

コマンドプロンプトで次のように入力します。

- **add iptunnel** <name> <remote> <remoteSubnetMask> <local> -**type** -**protocol** (**ipoverip** | **GRE**)
- **show iptunnel**

CLI を使用して IP トンネルを削除するには:

IP トンネルを削除するには、**rm iptunnel** コマンドとトンネルの名前を入力します。

CLI を使用して IPv6 トンネルを作成するには:

コマンドプロンプトで次のように入力します。

- **add ip6tunnel** <name> <remoteIp> <local>
- **show ip6tunnel**

CLI を使用して IPv6 トンネルを削除するには:

IPv6 トンネルを削除するには、**rm ip6tunnel** コマンドとトンネルの名前を入力します。

GUI のプロシージャ

GUI を使用して IP トンネルを作成するには:

[システム] > [ネットワーク] > [**IP** トンネル] に移動し、新しい IP トンネルを追加します。

GUI を使用して IPv6 トンネルを作成するには:

[システム] > [ネットワーク] > [**IP** トンネル] > [**IPv6** トンネル] に移動し、新しい IPv6 トンネルを追加します。

IP トンネルをグローバルにカスタマイズする

送信元 IP アドレスをグローバルに指定することで、すべてのトンネルに共通の送信元 IP アドレスを割り当てることができます。また、フラグメンテーションは CPU を大量に消費するため、NetScaler ADC アプライアンスがフラグメンテーションを必要とするパケットをドロップするようにグローバルに指定できます。また、CPU のしきい値に達していない限りすべてのパケットをフラグメント化したい場合は、CPU しきい値をグローバルに指定できます。

CLI のプロシージャ

CLI を使用して IP トンネルをグローバルにカスタマイズするには:

コマンドプロンプトで入力します。

- **set ipTunnelParam** -**srcIP** <sourceIPAddress> -**srcIPRoundRobin** (**YES** | **NO**)-**dropFrag** [**YES** | [**NO**]-** ドロップフラグ CPU しきい値 ** <Positive integer>
- **show ipTunnelParam**

例**:**

```
1 > set iptunnelparam  - srcIP 12.12.12.22 -dropFrag Yes  -
     dropFragCpuThreshold 50
2  Done
3
4 > set iptunnelparam -srcIPRoundRobin YES -dropFrag Yes  -
     dropFragCpuThreshold 50
```



```
5   Done
6  <!--NeedCopy-->
```

CLI を使用して IPv6 トンネルをグローバルにカスタマイズするには:

コマンドプロンプトで入力します。

- **set ip6tunnelparam** -**srcIP** <IPv6Address> -**srcIPRoundRobin** (**YES** | NO)-**dropFrag** [**YES** | [**NO**]-** ドロップフラグ CPU しきい値 ** <Positive integer>
- **show ip6tunnelparam**

GUI のプロシージャ

GUI を使用して IP トンネルをグローバルにカスタマイズするには:

[システム] > [ネットワーク] に移動し、[設定] グループで [**IPv4** トンネルのグローバル設定] をクリックします。

1. [システム] > [ネットワーク] に移動し、[設定] グループで [**IPv6** トンネルのグローバル設定] をクリックします。
2. **IP** トンネルグローバルパラメータの設定ダイアログボックスで、パラメータを設定します。

GUI を使用して IPv6 トンネルをグローバルにカスタマイズするには:

1. [システム] > [ネットワーク] に移動し、[設定] グループで [**IPv6** トンネルのグローバル設定] をクリックします。
2. **IP** トンネルグローバルパラメータの設定ダイアログボックスで、パラメータを設定します。

GRE IP トンネルの **GRE** ペイロードオプション

設定された GRE IP トンネルの場合、NetScaler ADC アプライアンスは、イーサネットヘッダーと VLAN ヘッダー（dot1q VLAN タグ）を含むレイヤー 2 パケット全体をカプセル化します。NetScaler ADC アプライアンスと一部のサードパーティデバイス間の IP GRE トンネルは、これらのサードパーティデバイスがレイヤー 2 パケットヘッダーの一部または一部を処理するようにプログラムされていないため、安定していない可能性があります。NetScaler ADC アプライアンスとサードパーティデバイス間の安定した IP GRE トンネルを設定するには、GRE IP トンネルコマンドセットの GRE ペイロードパラメータを使用できます。GRE ペイロード設定は、IPsec トンネルを使用する GRE にも適用できます。

パケットが GRE トンネルを介して送信される前に、次のいずれかを実行するように GRE ペイロードパラメータを設定できます。

- **DOT1Q** 搭載のイーサネットイーサネットヘッダーと VLAN ヘッダーを伝送します。これがデフォルトの設定です。ネットブリッジにバインドされたトンネルの場合、内部イーサネットヘッダーと VLAN ヘッダーには、NetScaler ADC アプライアンスの ARP とブリッジテーブルからの情報が含まれます。PBR ルールのネクストホップとして設定されたトンネルでは、内部イーサネットの宛先 MAC アドレスは 0 に設定され、VLAN へ

ッダーはデフォルト VLAN を指定します。NetScaler トンネルエンドポイントから送信されるカプセル化（GRE）パケットの形式は次のとおりです。

Outer Ethernet Header	Outer IP Header	GRE Header	Inner Ethernet	Inner VLAN header	Inner IP/IPv6/ARP header	Inner TCP/UDP Header	Payload

- イーサネット。イーサネットヘッダーは伝送しますが、VLAN ヘッダーはドロップします。パケットはトンネル内で VLAN 情報を伝送しないため、この設定でネットブリッジにバインドされているトンネルの場合、適切な VLAN をネットブリッジにバインドして、トンネル上でパケットを受信すると、NetScaler ADC がこれらのパケットを指定された VLAN に転送できるようにする必要があります。トンネルが PBR ルールでネクストホップとして設定されている場合、NetScaler ADC はトンネルで受信したパケットをルーティングします。NetScaler トンネルエンドポイントから送信されるカプセル化（GRE）パケットの形式は次のとおりです。

Outer Ethernet header	Outer IP header	GRE Header	Inner Ethernet header	Inner IP/IPv6/ARP header	Inner TCP/UDP header	Payload

- **IP**。イーサネットヘッダーと VLAN ヘッダーをドロップします。この設定のトンネルはレイヤ 2 ヘッダーを伝送しないため、これらのトンネルをネットブリッジにバインドすることはできませんが、PBR ルールではネクストホップとして設定できます。パケットを受信したピアトンネルエンドポイントデバイスは、パケットを消費またはルーティングします。NetScaler トンネルエンドポイントから送信されるカプセル化（GRE）パケットの形式は次のとおりです。

Outer Ethernet header	Outer IP header	GRE header	Inner IP/IPv6 header	Inner TCP/UDP header	Payload

CLI を使用して GRE IP トンネル内のパケットのレイヤ 2 ヘッダーをドロップするには、次の手順を実行します。

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> [-**protocol** \<GRE> [-**vlan** \<positive_integer>]] [-**grepayload** \<grepayload>] [-**ipsecProfileName** \<string>]
- **show iptunnel** <tunnelname>

例**:**

```
> add iptunnel IPTUNNEL-1 203.0.113.133 255.255.255.0 198.51.100.15 -protocol GRE -grepayload Ethernet -ipsecProfileName IPTUNNEL-IPSEC-1
Done
<!--NeedCopy-->
```

GRE IPV4 トンネル経由の **IPv6** トラフィック

NetScaler ADC アプライアンスは、IPV4 GRE トンネルを介した IPv6 トラフィックの転送をサポートしています。この機能を使用すると、隔離された IPv6 ネットワーク間の IPv4 インフラストラクチャをアップグレードしなくて

も、隔離された IPv6 ネットワーク間の通信が可能になります。

この機能を設定するには、PBR6 ルールを、NetScaler ADC が IPv6 トラフィックを送受信できるようにする構成済みの IPv4 GRE トンネルに関連付けます。PBR6 ルールの送信元 IPv6 アドレスと宛先 IPv6 アドレスのパラメータは、トラフィックが IPv4 GRE トンネルを通過する IPv6 ネットワークを指定します。

注**:** IPsec プロトコルは、IPv6 パケットを転送するように設定された GRE IPv4 トンネルではサポートされていません。

CLI を使用して GRE IPv4 トンネルを作成するには:

コマンドプロンプトで入力します。

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> -**protocol GRE**
- **show ipTunnel** <name>

CLI を使用して PBR6 ルールを GRE IPv4 トンネルに関連付けるには:

- **add ns pbr6** <pbrName> **ALLOW** -**srcIPv6** <network-range> -**dstIPv6** <network-range> -**ipTunnel** <tunnelName>
- **show pbr**

設定例

次の設定例では、リモートトンネルエンドポイント IP アドレス 10.10.6.30 とローカルトンネルエンドポイント IP アドレス 10.10.5.30 で GRE IP トンネル V6onv4 が作成されています。その後、トンネルは pbr6 PBR6-v6onv4 にバインドされます。SRCIPv6 はローカルエンドポイントに接続された IPv6 ネットワークを指定し、DestIPv6 はリモートエンドポイントに接続された IPv6 ネットワークを指定します。これらの Ipv6 ネットワークからのトラフィックは、GRE IPv4 トンネルを通過できます。

```
1 > add ipTunnel TUNNEL-V6onV4 10.10.6.30 255.255.255.255 10.10.5.30 -
     protocol GRE
2 -ipsecProfileName None
3 Done
4 > add ns pbr6 PBR6-V6onV4 ALLOW -srcIPv6 = 2001:0db8:1::1-2001:0db8
     :1::255 -destIPv6 =
5 1-2001:0db8:4::255 -ipTunnel TUNNEL-V6onV4
6 <!--NeedCopy-->
```

IP-IP トンネル経由の応答トラフィックの送信

NetScaler ADC アプライアンスは、応答トラフィックを送信元に戻すのではなく、IP-IP トンネルを介して送信するように構成できます。デフォルトでは、アプライアンスは IP-IP トンネルを介して別の NetScaler ADC またはサードパーティデバイスからの要求を受信すると、応答トラフィックをトンネル経由で送信する代わりにルーティングし

ます。ポリシーベースルート (PBR) を使用するか、MAC ベース転送 (MBF) を有効にして、トンネル経由で応答を送信できます。

PBR ルールでは、トラフィックがトンネルを通過する両方のエンドポイントのサブネットを指定します。また、ネクストホップをトンネル名として設定します。応答トラフィックが PBR ルールと一致すると、NetScaler ADC アプライアンスはトラフィックをトンネル経由で送信します。

または、MBF を有効にしてこの要件を満たすこともできますが、機能は NetScaler ADC アプライアンスがセッション情報を保存するトラフィック（負荷分散や RNAT 構成に関連するトラフィックなど）に限定されます。アプライアンスはセッション情報を使用して、トンネルを介して応答トラフィックを送信します。

CLI のプロシージャ

CLI を使用して PBR ルールを作成し、IP-IP トンネルをそれに関連付けるには:

コマンドプロンプトで入力します。

- **add ns pbr** <pbr_name> **ALLOW** -**srcIP** = <local_subnet_range> -**destIP** = <remote_subnet_range> -**ipTunnel** <tunnel_name>
- **apply ns pbrs**
- **show ns pbr** <pbr_name>

CLI を使用して MAC ベースの転送を有効にするには:

コマンドプロンプトで入力します。

- **enable ns mode MBF**
- **show ns mode**

GUI のプロシージャ

GUI を使用して PBR ルールを作成し、その PBR ルールに IP-IP トンネルを関連付けるには:

1. [システム] > [ネットワーク] > [**PBR**] に移動します。**PBR** タブで **PBR** ルールを作成します。
2. PBR を作成するときに、ネクストホップタイプを **IP** トンネルに、**IP** トンネル名を設定した IP-IP トンネル名に設定します。

GUI を使用して MAC ベースの転送を有効にするには:

1. [システム] > [設定**]** に移動し、[モードと機能] で [モードの設定] をクリックします。
2.「モードの設定」ページで、「**MAC** ベースの転送」を選択します。

サンプル構成

NS1-NS2-IPIP という IPIP トンネルの例を考えてみましょう。このトンネルは、2 つの NetScaler ADC アプライアンス NS1 と NS2 の間にセットアップされています。

デフォルトでは、NS2 がトンネルを介して受信したリクエストでは、応答トラフィックをトンネル経由で（NS1 に）送信するのではなく、ソースにルーティングします。

ポリシーベースルート (PBR) を設定するか、NS2 で MAC ベース転送 (MBF) を有効にして、トンネル経由で応答を送信できるようにすることができます。

次の NS2 の設定例では、NS1-NS2-IPIP は IPIP トンネルで、NS1-NS2-IPIP-PBR は PBR ルールです。NS2 がトンネル経由で受信した要求（内部送信元 IP アドレスが 10.102.147.0-10.102.147.255 の範囲で、内部宛先 IP アドレスが 10.102.147.0-10.102.147.255 の範囲にある）の場合、NS2 は対応する応答を送信元に転送する代わりに、対応する応答をトンネル経由で（NS1 に）送信します。この機能は、PBR ルールに一致するトラフィックに限定されます。

```
> add iptunnel NS1-NS2-IPIP 192.0.2.99 255.255.255.255 203.0.113.99 -
    protocol IPIP

Done
> add pbr NS1-NS2-IPIP-PBR -srcIP 10.102.147.0-10.102.147.255  - destIP
    10.20.1.0-10.20.1.255  - ipTunnel NS1-NS2-IPIP

Done
> apply pbrs

Done
<!--NeedCopy-->
```

または、NS2 で MBF を有効にすることもできます。この機能は、NS2 がセッション情報を格納するトラフィック（たとえば、ロード・バランシングや RNAT 構成に関連するトラフィック）に限定されます。

```
> enable ns mode MBF

Done
<!--NeedCopy-->
```

クラス **E IPv4** パケット

August 15, 2023

デフォルトでは、NetScaler アプライアンスは、送信元 IP または宛先 IP フィールドにクラス E IPv4 アドレスが含まれているパケットをすべてドロップします。クラス E IPv4 アドレスを使用している場合は、クラス E IPv4 パケットを処理するように NetScaler アプライアンスを構成できます。

はじめに

クラス E IPv4 パケットを処理するように NetScaler アプライアンスを構成する前に、次の点に注意してください。

- NetScaler アプライアンスは、NetScaler が所有する IPv4 アドレス（SNIP や VIP など）をクラス E 範囲に設定することをサポートしていません。NetScaler アプライアンスは、クラス E IPv4 パケットの処理のみをサポートします。
- NetScaler アプライアンスは、IPv6 機能にクラス E の IPv4 アドレスを内部で使用します。NetScaler アプライアンスは、同時に動作する両方の機能（クラス E IPv4 パケットの処理と IPv6 サポート）をサポートしていません。NetScaler アプライアンスは、クラス E IPv4 パケットの処理が有効になっている場合は IPv6 機能を有効にしないという制限を課します。その逆も同様です。

構成の手順

クラス E IPv4 パケットを処理するように NetScaler アプライアンスを構成するには、**IPv4** クラス **E** アドレスクライアント（AllowClasseIPv4**）レイヤー 3 パラメーターを有効にするタスクが必要です。

CLI のプロシージャ

CLI を使用してクラス E IPv4 パケットを処理するように NetScaler アプライアンスを構成するには:

コマンドプロンプトで入力します。

set l3param -allowClassEIPv4 (ENABLED DISABLED)

-
- **show l3param**

設定例**:**

```
> set l3param -allowClassEIPv4 ENABLED

Done

> sh l3param

    Network L3 related Configuration Parameters

    icmpgen_rate_threshold      : 100

    srcnat                      : ENABLED

```



```
    override_rnat                 : DISABLED

    drop_df_flag                  : DISABLED

    .

    .

    .

    IPv6DynamicRouting            : DISABLED

    allowClassEIPv4               : ENABLED

Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用してクラス E IPv4 パケットを処理するように NetScaler アプライアンスを構成するには:

1. [システム] > [ネットワーク] に移動し、[設定] セクションで [レイヤ **3** パラメータの設定] をクリックします。
2. **IPv4** クラス **E** アドレスクライアントを選択し、**OK** をクリックします。

NetScaler ADC アプライアンスで使用可能な空きポートを監視して、新しいバックエンド接続がないか確認する

August 15, 2023

NetScaler ADC アプライアンスは、物理サーバーまたは他のピアデバイスとの通信に、Citrix が所有する IP アドレスをソース IP アドレスとして使用します。NetScaler ADC アプライアンスは IP アドレスのプールを維持し、サーバーとの接続中に IP アドレスを動的に選択します。物理サーバが配置されているサブネットに応じて、アプライアンスは使用する IP アドレスを決定します。このアドレスプールは、トラフィックおよびモニタプローブの送信に使用されます。

新しいバックエンド接続で NetScaler ADC が所有する IP アドレスで使用可能な空きポートの総数を表示できます。この情報は、使用可能な空きポートがほぼ使い果たされそうになった場合に、NetScaler が所有する IP アドレスを増やす必要があるかどうかを判断するのに役立ちます。

NetScaler ADC アプライアンスに次の情報を入力して、新しいバックエンド接続に使用できる空きポートの総数を計算できます。

- Citrix が所有する IP アドレス（オプション）

- 宛先 IP アドレス
- Destination port
- TCP または非 TCP プロトコル

Citrix が所有する IP アドレスを指定する以外のすべての情報を指定すると、次のようになります。

- NetScaler アプライアンスはルート検索を実行して、宛先 IP アドレスに接続できる NetScaler 所有の IP アドレスをすべて検索します。次に、アプライアンスは、指定された新しいバックエンド接続について、NetScaler が所有するこれらの IP アドレスで使用可能な空きポートの総数を検出して表示します。

  > 注:
  >
  > NetScaler アプライアンスは、宛先 IP アドレスに接続できる NetScaler 所有の IP アドレスを見つけるための ECMP ルックアップ、LLB ルックアップパス、または PBR ルックアップパスを実行しません。

Citrix が所有する IP アドレスを指定するなど、すべての情報を指定すると、次のようになります。

- NetScaler ADC アプライアンスには、指定された新しいバックエンド接続について、指定された IP アドレスで使用可能な空きポートの数が表示されます。

はじめに

新しいバックエンド接続で使用できる空きポートの総数を表示する前に、次の点に注意してください。

- NetScaler アプライアンスは、宛先 IP アドレスに接続できる NetScaler 所有の IP アドレスを見つけるための ECMP ルックアップ、LLB ルックアップパス、または PBR ルックアップパスを実行しません。
- NetScaler ADC アプライアンスは、リンクローカル IP アドレスで使用可能な空きポートの表示をサポートしていません。

NetScaler ADC アプライアンスで新しいバックエンド接続に使用できる空きポート数を表示する手順

NetScaler ADC アプライアンスで新しいバックエンド接続に使用できる空きポートの総数を表示するには:

コマンドプロンプトで次のように入力します。

• **show portallocation** [-**srcIP** <ip_addr	ipv6_addr>] -**destIP** <ip_addr	ipv6_addr> -**destPort** -**protocol** <1 for TCP, 0 for non-TCP protocol>

例-スタンドアロン **NetScaler ADC** アプライアンスで使用可能な空きポートの総数:


```
> show portallocation -destip 198.51.100.30 -destport 80 -protocol 1

        Freeports available : 64505
 Done


> show portallocation -srcip 192.0.2.30 -destip 198.51.100.30 -destport
    80 -protocol 1

        Freeports available for IPAddress 192.0.2.30 : 20505
 Done
<!--NeedCopy-->
```

例-クラスタ設定で使用可能な空きポートの総数:

次の出力例は、2 ノードクラスタセットアップの各ノードで使用可能な空きポートの総数を示しています。

```
> show portallocation -destip 198.51.100.30 -destport 80 -protocol 1

    Node Id: 1
    Freeports available : 32321

    Node Id: 0
    Freeports available : 32184

Done
<!--NeedCopy-->
```

SNMP を使用して、NetScaler ADC アプライアンスのバックエンド接続のポート使用量を監視する

PORT-ALLOC-EXCEED SNMP アラームを使用して、NetScaler ADC アプライアンスのバックエンド接続のポート使用率を監視できます。

PORT-ALLOC-EXCEED SNMP アラームには、NetScaler が所有する IP アドレスに割り当てられたポートの合計数をパーセンテージで指定するhigh-thresholdおよびnormal-thresholdパラメータが含まれます。たとえば、high-threshold パラメータが 90 に設定されている場合、NetScaler ADC アプライアンスは次のイベントが発生するとトラップメッセージを生成して送信します。

- NetScaler ADC が所有するバックエンド接続の IP アドレスのいずれかで、ポート割り当ての割合が 90%を超える場合

SNMP アラートは、使用可能な空きポートがほぼ使い果たされそうになった場合に、NetScaler が所有する IP アドレスを増やす必要があるかどうかを判断するのに役立ちます。

SNMP を使用して NetScaler ADC アプライアンスのバックエンド接続のポート使用量を監視するには

コマンドプロンプトで次のように入力します。

• **set snmp alarm PORT-ALLOC-EXCEED -logging** (ENABLED	DISABLED) -**severity** -**state** (ENABLED	DISABLED) -**thresholdValue** [-**normalValue** \] -**time**

- **sh snmp** アラームポート割り当て超過

例:

```
1  > set snmp alarm PORT-ALLOC-EXCEED -logging ENABLED  -severity Major -
       state ENABLED -thresholdValue 90 -time 1200
2  Done
3
4  > sh snmp alarm port-alloc-EXCEED
5
6  Alarm                    Alarm Threshold     Normal Threshold    Time
         State       Severity        Logging
7  -----                    ---------------     ----------------    ----
         --------    -------------   --------
8  1) PORT-ALLOC-EXCEED     80                  80                  7200
         ENABLED     Major           ENABLED
9  Done
10
11 <!--NeedCopy-->
```

SNMP アラームと SNMP トラップリスナーの構成について詳しくは、「SNMP トラップを生成するための NetScaler ADC 構成」を参照してください。

インターフェイス

August 15, 2023

インターフェイスの構成を開始する前に、構成で MAC ベースの転送モードを使用できるかどうかを決定し、それに応じてこのシステム設定を有効または無効にします。構成内のインターフェイスの数は、NetScaler アプライアンスのモデルによって異なります。個々のインターフェイスを設定できるだけでなく、VLAN を使用してインターフェイスセット内のデータフローを制限したり、リンクをチャネルに集約したりして、インターフェイスを論理的にグループ化できます。高可用性セットアップでは、必要に応じて仮想 MAC アドレスを設定できます。L2 モードを使用する場合は、ブリッジテーブルのエージングを変更したい場合があります。

構成が完了したら、パス MTU 検出のシステム設定を有効にするかどうかを決定します。NetScaler アプライアンスは、VRRP を使用してアクティブ/アクティブモードで展開できます。アクティブ-アクティブ展開では、ダウンタイムを防ぐだけでなく、展開環境内のすべての NetScaler アプライアンスを効率的に使用できます。Network Visualizer ツールを使用して、NetScaler 環境のネットワーク構成を表示し、インターフェイス、チャネル、VLAN、およびブリッジグループを構成できます。

MAC ベース転送の構成

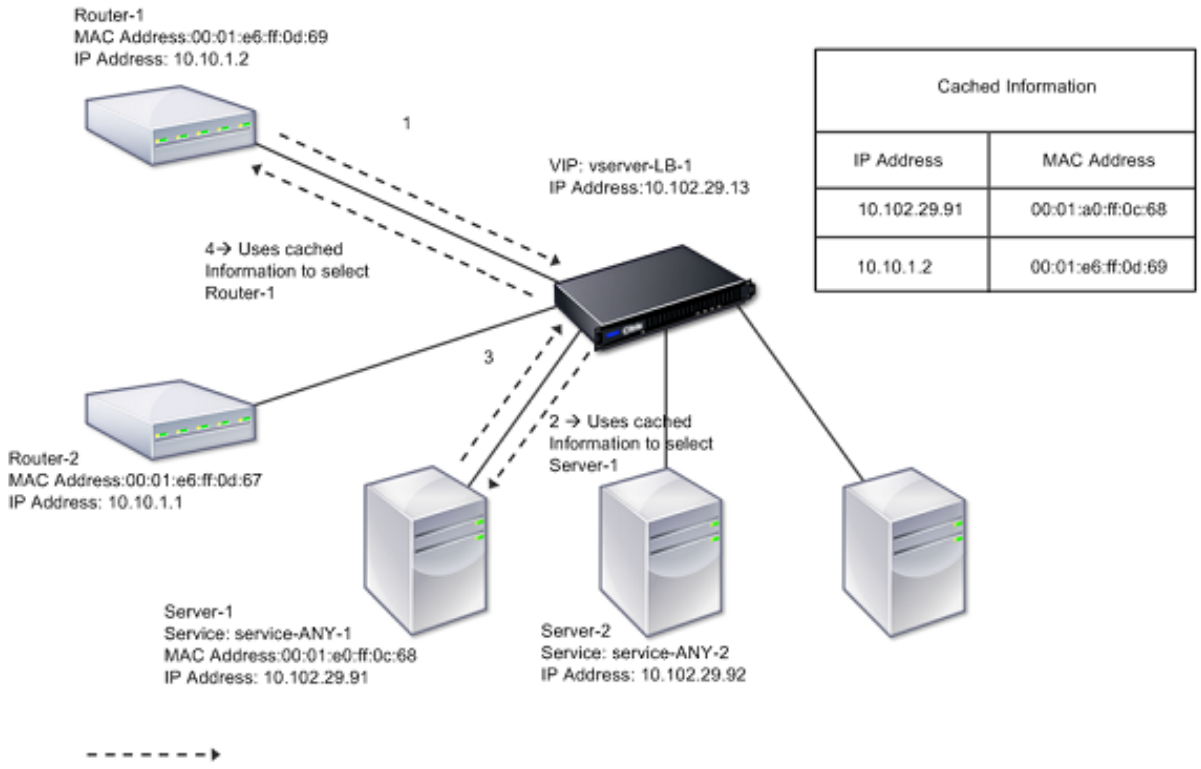
August 15, 2023

MAC ベースの転送（MBF）を有効にすると、要求が NetScaler アプライアンスに届くと、アプライアンスはフレームの送信元 MAC アドレスを記憶し、それを返信の宛先 MAC アドレスとして使用します。MAC ベースの転送を使用すると、複数のルート/ARP ルックアップを回避したり、パケットフローが非対称になるのを防ぐことができます。NetScaler が VPN やファイアウォールなどの複数のステートフルデバイスに接続されている場合、MAC ベースの転送が必要になることがあります。これは、リターントラフィックが最初のトラフィックの送信元と同じデバイスに送信されることを保証するためです。

MAC ベースの転送は、VPN を経由するすべてのトラフィックが同じ VPN デバイスを経由することが保証されるため、VPN デバイスを使用する場合に便利です。

次のトポロジー図は、MAC ベースの転送プロセスを示しています。

図 1：MAC ベース転送モード



MAC ベースの転送（MBF）が有効な場合、NetScaler は次の MAC アドレスをキャッシュします。

- 受信接続のソース（ルーター、ファイアウォール、VPN デバイスなどの通信デバイス）
- 要求に応答するサーバー

サーバーが NetScaler アプライアンスを介して応答すると、アプライアンスは応答パケットの宛先 MAC アドレスをキャッシュされたアドレスに設定し、トラフィックが対称的に流れるようにしてから、応答をクライアントに転送

します。このプロセスでは、ルートテーブルのルックアップ機能と ARP ルックアップ機能が回避されます。ただし、NetScaler は接続を開始すると、ルックアップ機能にルートテーブルと ARP テーブルを使用します。ダイレクトサーバリターン設定では、MAC ベースの転送を有効にする必要があります。

ダイレクトサーバーのリターン構成の詳細については、「負荷分散」を参照してください。

一部の展開トポロジでは、着信パスと発信パスが異なるルータを経由する必要があります。MAC ベースの転送は、このトポロジー設計を壊すことになります。

MBF は、次の状況で無効にする必要があります。

- サーバーが **LACP (802.1ad** リンクアグリゲーション**)** を使用せずにネットワークインターフェイスカード **(NIC)** チーミングを使用する場合。このような状況で MAC ベースの転送を有効にするには、NetScaler とサーバーの間にレイヤー 3 デバイスを使用する必要があります。
  注: 仮想インターフェイスは 1 つの MAC アドレスを使用するため、サーバが LACP と NIC チーミングを使用する場合、MBF を有効にできます。
- ファイアウォール・クラスタリングを使用する場合。ファイアウォールクラスタリングでは、ARP を使用してインバウンドトラフィックの MAC アドレスを解決することを前提としています。インバウンド MAC アドレスがクラスタ化されていない MAC アドレスである場合があり、インバウンドパケットの処理には使用しないでください。

MBF が無効になっている場合、アプライアンスは L2 または L3 接続を使用してサーバーからの応答をクライアントに転送します。ルートテーブルによっては、送信接続と着信接続に使用されるルーターが異なる場合があります。リバーストラフィック (サーバーからの応答) の場合:

- 送信元と宛先が異なる IP サブネットにある場合、アプライアンスはルート検索を使用して宛先を特定します。
- ソースがターゲットと同じサブネットにある場合、NetScaler は ARP テーブルを検索してネットワークインターフェイスを見つけ、トラフィックをそのテーブルに転送します。ARP テーブルが存在しない場合、NetScaler は ARP エントリを要求します。

CLI を使用して MAC ベースの転送を有効または無効にするには:

コマンドプロンプトで入力します。

- **enable ns mode MBF**
- **disable ns mode MBF**

GUI を使用して MAC ベースの転送を有効または無効にするには:

1. [システム] > [設定**]** に移動し、[モードと機能**]** グループで [モードの設定] をクリックします。
2. **MAC** ベースの転送オプションを選択または選択解除します。

MAC ベースの転送による負荷分散設定

一部の負荷分散設定では、NetScaler アプライアンスがこれらの設定でグローバル MBF（有効になっている場合）をバイパスし、代わりにルート/ARP ルックアップを使用して宛先にパケットを送信する必要があります。

ネットプロファイルの MBF パラメータは、特定の負荷分散構成の MBF を有効または無効にするために使用されます。MBF は、ネットプロファイル（MBF の有効または無効）を仮想サーバーとサービスにバインドすることにより、負荷分散構成のサーバー側だけでなくクライアント側にも設定できます。

たとえば、MBF が無効になっているネットプロファイルが負荷分散構成の仮想サーバーにバインドされている場合、NetScaler アプライアンスはグローバル MBF をバイパスし（有効になっている場合）、代わりにルート/ARP ルックアップを使用して応答パケットをクライアントに送信します。

はじめに

負荷分散構成用に MBF を設定する前に、次の点に注意してください。

- 負荷分散構成では、クライアント側 (仮想サーバー) とサーバー側 (サービス/サービスグループ) で異なる MBF 設定を使用できます。
- 仮想サーバーとサービスにバインドされたネットプロファイルに MBF が明示的に設定されていない場合、負荷分散構成はグローバル MBF 設定を継承します。
- 負荷分散構成では、ネットプロファイルがサービスにバインドされていない場合、サーバー側 (サービス) は仮想サーバーにバインドされたネットプロファイルのクライアント側 MBF 設定を継承します。
- Direct Server Return モードの負荷分散構成では、クライアント側はサービスにバインドされたネットプロファイルの MBF 設定を継承します。
- コンテンツスイッチング構成では、クライアント側は、ターゲットの負荷分散仮想サーバーからではなく、コンテンツスイッチング仮想サーバーにバインドされたネットプロファイルの MBF 設定を取得します。

制限事項

ロードバランシング構成用に MBF を設定する前に、次の制限事項に注意してください。

- 負荷分散構成の MBF 設定は、クラスター設定ではサポートされていません。
- MAC モードまたは L2Conn 設定の負荷分散仮想サーバーでは、仮想サーバーにバインドされたネットプロファイルの MBF 設定に関係なく MBF が有効になります。
- NetScaler アプライアンスは、ネットプロファイルを使用する負荷分散モニターの MBF の設定をサポートしていません。つまり、ネットプロファイルの MBF 設定は、ネットプロファイルがバインドされているモニターには適用されません。グローバル MBF 設定は、バインドされたネットプロファイルの MBF 設定に関係なくモニターに適用されます。

ロードバランシング構成用の **MBF** の設定

ロードバランシング構成用の MBF の設定は、次のタスクで構成されます。

- ネットプロファイルの MBF パラメータを有効にします。
- ネットプロファイルを負荷分散仮想サーバーまたはサービスにバインドします。

CLI を使用してネットプロファイルで MBF を有効にするには:

- ネットプロファイルの追加時に MBF を有効にするには、コマンドプロンプトで次のように入力します。
  - **add netProfile** <name> -**MBF** (**ENABLED** | ** 無効 **)
  - **show netprofile** <name>
- 既存のネットプロファイルで MBF を有効にするには、コマンドプロンプトで次のように入力します。
  - **set netProfile** <name> -**MBF** (**ENABLED** | ** 無効 **)
  - **show netprofile** <name>

GUI を使用してネットプロファイルの MBF を有効にするには **

1. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
2. ネットプロファイルを追加または変更するときは、**MBF** パラメータを有効にします。

次の構成例では、ネットプロファイルの NETPROFILE-MBF-LBVS で MBF が有効になっており、負荷分散仮想サーバー LBVS-1 にバインドされています。また、ネットプロファイルの NETPROFILE-MBF-SVC では MBF が有効になっており、ロードバランシングサービス SVC-1 にバインドされています。

```
> add netprofile NETPROFILE-MBF-LBVS -MBF ENABLED

Done

> add netprofile NETPROFILE-MBF-SVC -MBF ENABLED

Done

> set lb vserver LBVS-1 -netprofile NETPROFILE-MBF-LBVS

Done

> set service SVC-1 -netprofile NETPROFILE-MBF-SVC

Done

<!--NeedCopy-->
```

ネットワークインターフェースの設定

December 8, 2023

NetScaler アプライアンスのネットワークインターフェースには、表記法で番号が付けられています。<slot>:<port> インターフェイスを構成したら、インターフェイスとその設定を表示して構成を確認します。この情報を表示して、設定の問題のトラブルシューティングを行うこともできます。

ネットワークインターフェースを管理するには、次の操作を行います。

- 一部のインターフェイスを有効にし、他のインターフェイスを無効にします。
- インターフェイスをリセットして設定を再ネゴシエートします。
- インターフェイスの蓄積された統計情報を消去します。

設定を確認するには、インターフェイス設定を表示できます。インターフェイスの統計情報を表示して、その状態を評価できます。

ネットワークインターフェースのパラメータを設定します

ネットワークインターフェースの設定は同期も伝達もされません。HA ペアの場合は、各ユニットで個別に設定を実行する必要があります。

> 注:
>
> HA セットアップでは、`tag all` パラメータが有効になっている場合、HA パケットにも VLAN トラフィックのタグが付けられます。ただし、高可用性通信には問題がある可能性があります。そのため、HA トラフィックには NSVLAN または HA SYNC VLAN を設定することをお勧めします。
>
> - NSVLAN の設定については、「NSVLAN の設定」を参照してください。
> - HA シンク VLAN の設定については、「HA シンク VLAN の設定」を参照してください。

CLI を使用してネットワーク・インタフェース・パラメータを設定するには:

コマンドプロンプトで入力します:

```
1 -  set interface <id> [-speed <speed>] [-duplex <duplex>] [-flowControl
      <flowControl>] [-autoneg ( DISABLED | ENABLED )] [-haMonitor ( ON |
      OFF )] [ ( ON | OFF )] [-tagall ( ON | OFF )] [-lacpMode <lacpMode
     >] [-lacpKey<positive_integer>] [-lacpPriority <positive_integer>]
     [-lacpTimeout (LONG | SHORT )] [-ifAlias <string>] [-throughput <
     positive_integer>][-bandwidthHigh <positive_integer> [-
     bandwidthNormal <positive_integer>]]
2 -  show interface [<id>]
3 <!--NeedCopy-->
```

例:

```
1 > set interface 1/8 -duplex full
2  Done
3 <!--NeedCopy-->
```

GUI を使用してネットワーク・インタフェース・パラメータを設定するには:

[システム] **>** [ネットワーク] ****[**インターフェイス] に移動し、変更するネットワークインターフェイス **(**たとえば **1/8)** を選択し、**[**** 編集] をクリックしてパラメータを設定します。

インターフェイスの受信リングサイズとリングタイプの設定

NetScaler MPX および SDX プラットフォームでは、IX、F1X、F2X、または F4X インターフェイスの受信リングサイズとリングタイプを増やすことができます。

リングサイズを大きくすると、バーストトラフィックを処理するクッション性が高まりますが、パフォーマンスに影響する可能性があります。IX インターフェイスでは最大 8192 のリングサイズがサポートされます。F1X、F2X、および F4X インターフェイスでは、最大 4096 のリングサイズがサポートされます。デフォルトのリングサイズは 2048 のままです。

インターフェイスリングタイプはデフォルトではエラスティックです。パケットの到着率に応じてサイズが増減します。リングタイプを「固定」に設定できます。その場合、リングサイズはトラフィックレートに基づいて変化しません。

注: この機能はリリース 13.0 ビルド 41.x からサポートされ、IX、F1X、F2X、または F4X インターフェイスを備えたプラットフォームでもサポートされています。

`show hardware` コマンドを使用して、アプライアンスに IX、F1X、F2X、または F4X のいずれのインターフェイスがあるかを確認します。

例:

次のモデルには、16 の F1X (10G) インターフェイスと 4 つの F4X (40G) インターフェイスがあります。

```
1  > sh hardware
2          Platform: NSMPX-25000-40G 20\*CPU+16\*F1X+4\*F4X+2\*E1K+2*CVM
               N3 250040
3          Manufactured on: 12/16/2016
4          CPU: 2800MHZ
5          Host Id: 234913926
6          Serial no: N43RJCRV3X
7          Encoded serial no: N43RJCRV3X
8          Netscaler UUID: 336a32d6-2cfa-11e8-bf01-00e0ed5dd23c
9          BMC Revision: 4.08
10  Done
11 <!--NeedCopy-->
```

次のモデルには 2 つの 1X (10G) インターフェイスがあります。

```
1  > sh hardware
2          Platform: NSMPX-10500 8\*CPU+2\*E1K+8\*E1K+2\*IX+8*CVM 1620
               760100
3          Manufactured on: 12/27/2010
4          CPU: 2832MHZ
5          Host Id: 1707114630
6          Serial no: 7VZZV1ZXJ4
7          Encoded serial no: 7VZZV1ZXJ4
8          Netscaler UUID: eb1bfd72-5176-11e7-ba18-00e0ed1b0d12
9   Done
10 <!--NeedCopy-->
```


CLI を使用してリングサイズとリングタイプを設定するには、
コマンドラインで次のように入力します。

```
set interface <id> -ringsize <positive_integer> -ringtype ( Elastic | Fixed )
<!--NeedCopy-->
```

パラメーター:

ringsize:

インターフェイスの受信リングサイズ。数値が大きいほど、受信トラフィックを処理するバッファが多くなります。

デフォルト値:2048
最小値:512
最大値:16384

ringtype:

インターフェイスの受信リングタイプ。固定リングタイプでは、トラフィックレートに関係なく、設定した数のバッファが事前に割り当てられます。一方、エラスティックリングは、着信トラフィックレートに応じて拡大および縮小します。

設定可能な値: エラスティック、固定

デフォルト値: エラスティック

例:

```
> set interface 40/2 -ringsize 4096 -ringtype Fixed
 Done
> show interface 40/2

1)      Interface 40/2 (40G Ethernet, CR4, 40 Gbit) #21  flags=0xc020 <ENABLED, UP, UP, autoneg, HAMON, HEARTBEAT, 802.1q> MTU=1500, native vlan=10, MAC=00:e0:ed:75:14:2a, uptime 119h26m32s
        Requested: media AUTO, speed AUTO, duplex AUTO, fctl OFF, throughput 0
        Actual: media UTP, speed 40000, duplex FULL, fctl OFF, throughput 40000
        LLDP Mode: NONE, LR Priority: 1024
        RX: Pkts(1443972660032) Bytes(1457207315336105) Errs(0) Drops (53319) Stalls(0)
        TX: Pkts(1452311431262) Bytes(1458534011197761) Errs(0) Drops (788) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
        Rx Ring: Configured size=4096, Actual size=4096, Type: Fixed
 Done
<!--NeedCopy-->
```

最後の行には、設定済みのリングサイズと実際のリングサイズ、およびリングタイプが表示されます。

GUI を使用してリングサイズとリングタイプを設定するには:

1. [システム**] > [**ネットワーク**] > [**インターフェイス] に移動します。
2. インターフェイスを選択して [編集] をクリックします。
3. 「リングサイズ」で、次のいずれかを指定します。
   - **IX** インターフェイス**:** 512、1024、2048、4096、または 8192。
   - **F1X**、**F2X**、または **F4X** インターフェイス**:512**、 1024、2048、または 4096。
4. [リングタイプ] で、[エラスティック] または [固定] を選択します。
5. [**OK**] をクリックします。

ネットワークインターフェースの有効化と無効化

デフォルトでは、ネットワークインターフェースは有効になっています。ネットワークに接続されていないネットワークインターフェースをすべて無効にして、パケットを送受信できないようにします。高可用性セットアップでネットワークに接続されているネットワークインターフェイスを無効にすると、フェールオーバーが発生する可能性があります。

高可用性の詳細については、「高可用性」を参照してください。

CLI を使用してネットワークインターフェイスを有効または無効にするには、次の手順を実行します。

コマンドプロンプトで入力します:

```
-  enable interface <interface_num>
-  show interface <interface_num>
-  disable interface <interface_num>
-  show interface <interface_num>
<!--NeedCopy-->
```

例:

```
> enable interface 1/8
 Done
> show interface 1/8
        Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
        flags=0x4004000 <ENABLED, DOWN, BOUND to LA/1, down, autoneg,
           802.1q>
        MTU=1514, MAC=00:d0:68:15:fd:3d, downtime 906h58m40s
        Requested: media UTP, speed AUTO, duplex FULL, fctl OFF,
           throughput 0
        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
Done
<!--NeedCopy-->
```


GUI を使用してネットワークインターフェースを有効または無効にするには:

1. [システム] > [ネットワーク] > [インターフェイス] に移動します。
2. ネットワークインターフェースを選択し、アクションリストで「有効化」または「無効化」を選択します。

ネットワークインターフェースをリセット

ネットワークインターフェース設定は、デュプレックスや速度などのプロパティを制御します。ネットワークインターフェースの設定を再ネゴシエートするには、リセットする必要があります。

CLI を使用してネットワークインターフェースをリセットするには:

コマンドプロンプトで入力します:

```
-  reset interface <interface_num>
-  show interface <interface_num>
<!--NeedCopy-->
```

例:

```
> reset interface 1/8
 Done
<!--NeedCopy-->
```

GUI を使用してネットワークインターフェースをリセットするには:

1. [システム] > [ネットワーク] > [インターフェイス] に移動します。
2. ネットワークインターフェースを選択し、アクションリストで「インターフェースをリセット」を選択します。

ネットワークインターフェースの監視

ネットワークインターフェースの統計情報を表示してパラメータを監視し、その情報を使用してネットワークインターフェースの状態を確認できます。送信パケットと受信パケット、スループット、LACP（リンク集約制御プロトコル）データユニット、エラーなどのパラメータを監視できます。ネットワークインターフェースの統計情報を消去して、統計情報が消去された時点からの統計を監視できます。

CLI を使用してネットワークインターフェースの統計情報を表示するには:

コマンドプロンプトで入力します:

```
-  stat interface <interface_num>
<!--NeedCopy-->
```

例:

```
> stat interface 1/8
 Done
<!--NeedCopy-->
```


CLI を使用してネットワークインターフェースの統計情報を消去するには:

コマンドプロンプトで入力します:

```
1 -  clear interface <interface_num>
2 <!--NeedCopy-->
```

例:

```
1 > clear interface 1/8
2  Done
3 <!--NeedCopy-->
```

GUI を使用してインターフェイスの統計情報を表示するには:

[システム] > [ネットワーク] > [インターフェイス] に移動し、ネットワークインターフェイスを選択して、[インターフェイス統計] をクリックします。

GUI を使用してネットワークインターフェースの統計情報を消去するには:

1. [システム] > [ネットワーク] > [インターフェイス] に移動します。
2. ネットワークインターフェースを選択し、アクションリストで「統計をクリア」を選択します。

セッション転送規則の構成

August 15, 2023

デフォルトでは、NetScaler アプライアンスは転送するだけのトラフィックのセッションエントリを作成しません (L3 モード)。アプライアンスがサーバーに転送するクライアント要求の結果、同じパスで応答を返さなければならない場合は、転送セッションルールを作成できます。転送セッションルールは、特定のネットワークから発信された、または特定のネットワークを宛先とし、NetScaler によって転送されるトラフィックの転送セッションエントリを作成します。IPv4 トラフィックと IPv6 トラフィックの転送セッションルールを作成できます。

IPv4 転送セッションルールを設定する場合、転送セッションエントリを作成する IPv4 トラフィックを識別する条件として、IPv4 ネットワークアドレスまたは拡張 ACL のいずれかを指定できます。

- ネットワークアドレス。IPv4 ネットワークアドレスを指定すると、アプライアンスは、送信元または宛先がネットワークアドレスと一致する IPv4 トラフィックの転送セッションを作成します。
- 拡張 **ACL** ルール。拡張 ACL ルールを指定すると、アプライアンスは拡張 ACL ルールで指定された条件と一致する IPv4 トラフィックの転送セッションを作成します。

IPv6 転送セッションルールを設定する場合、転送セッションエントリを作成する IPv6 トラフィックを識別する条件として、IPv6 プレフィックスまたは ACL6 のいずれかを指定できます。

- **IPv6** プレフィックス。IPv6 プレフィックスを指定すると、アプライアンスは、送信元または宛先が IPv6 プレフィックスと一致する IPv6 トラフィックの転送セッションを作成します。

- **ACL6** ルール。ACL6 ルールを指定すると、アプライアンスは ACL6 ルールで指定された条件と一致する IPv6 トラフィックの転送セッションを作成します。

CLI を使用して IPv4 転送セッションルールを作成するには:

コマンドプロンプトで次のコマンドを入力して、転送セッションルールを作成し、構成を確認します。

- add forwardingSession <name> [\ \\ ] | \[-aclname \\] -connfailover \(ENABLED | DISABLED)
- show forwardingSession

例:

```
A network address as the condition:

> add forwardingSession fs-nw-1 10.102.105.51 255.255.255.255
 Done

An ACL as the condition:

> add forwardingSession fs-acl-1 acl1
 Done
<!--NeedCopy-->
```

GUI を使用して IPv4 転送セッションルールを設定するには:

[システム] > [ネットワーク] > [転送セッション] に移動するか、新しい IPv4 転送セッションを追加するか、既存の転送セッションを編集します。

CLI を使用して IPv6 転送セッションルールを作成するには:

- コマンドプロンプトで次のコマンドを入力して、転送セッションルールを作成し、構成を確認します。
  - add forwardingSession <name> [\] | \[-acl6name \\]
  - show forwardingSession

  例:

```
    An IPv6 prefix as the condition:

    > add forwardingSession fsv6-pfx-1 3ffe::/64
     Done

    An ACL6 rule as the condition:

    > add forwardingSession fsv6-acl6-1  -acl6name ACL6-FS
     Done
<!--NeedCopy-->
```

GUI を使用して IPv6 転送セッションルールを設定するには:

[システム] > [ネットワーク] > [転送セッション] に移動するか、新しい IPv6 転送セッションを追加するか、既存の転送セッションを編集します。

既存の転送セッションルールへの **ACL** ルールの割り当て

ACL ルールをネットワークアドレス/IPv6 プレフィックスベースの転送セッションルールに割り当てることができます。その場合、ACL ベースの転送セッションルールになります。既存の ACL ルールを ACL ベースの転送セッションルール内の別の ACL ルールに変更することもできます。既存の関連転送セッションエントリ（存在する場合）がタイムアウトすると、ルールは新しく割り当てられた ACL を使用して、転送セッションエントリを作成する対象の IPv4/IPv6 トラフィックを識別し始めます。

CLI を使用して拡張 ACL ルールを既存の IPv4 転送セッションルールに割り当てるには：

コマンドプロンプトで、次のように入力します。

- `set forwardingSession <name> [-aclname <string>]`
- `show forwardingSession <name>`

CLI を使用して既存の IPv6 転送セッションルールに ACL6 ルールを割り当てるには：

コマンドプロンプトで、次のように入力します。

- `set forwardingSession <name> [-acl6name <string>]`
- `show forwardingSession <name>`

例**:**

```
1 > add forwardingSession FS-1 -aclname ACL-9
2 Done
3
4 > add forwardingSession FS6-1  - acl6name ACL6-9
5 Done
```

クラスタセットアップでの転送セッションのステアリングの無効化

NetScaler クラスターのデフォルトの動作では、トラフィックを受信するノード（フローレシーバー）がトラフィックを処理する別のノード（フロープロセッサー）にトラフィックを転送します。フローレシーバからフロープロセッサへのトラフィックの転送は、クラスタバックプレーンを介して行われ、ステアリングと呼ばれます。

リアルタイム処理やセットアップに高遅延のリンクが含まれる場合、ステアリングはオーバーヘッドになる可能性があります。

転送セッションのステアリングを無効にして、処理をフローレシーバーに対してローカルに行うことができるようになりました。つまり、フローレシーバーがフロープロセッサになります。

はじめに

クラスタ設定で転送セッションルールを設定する前に、次の点に注意してください。

- 転送セッションに使用するリンクセットを設定する必要があります。
- クラスタ設定で MAC ベースフォワーディング (MBF) を有効にする必要があります。

クラスタ設定での転送セッションルールの設定

クラスター設定での転送セッションルールのステアリングの無効化は、次の 2 つのレベルで行うことができます。

- 特定の転送セッションルールレベル。新しい転送セッションルールを追加したり、既存の転送セッションルールを編集したりするときは、Process Local パラメータを有効にします。
- グローバルレベル。新しいクラスターインスタンスを追加したり、既存のクラスターインスタンスを編集したりするときに、Process Local パラメーターを有効にします。グローバル設定は、転送セッションのルール設定よりも優先されます。

CLI のプロシージャ　CLI を使用してクラスターセットアップの転送セッションルールのステアリングを無効にするには:

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

- 新しい転送セッション規則を追加する場合:
  - **add forwardingSession** <name> ((<network> [\]) | -**acl6name** \ | -**aclname** \) -**processLocal ENABLED**
  - **show forwardingSession** <name>
- 既存の転送セッションルールを再設定する場合:
  - **set forwardingSession** <name> -**processLocal ENABLED**
  - **show forwardingSession** <name>

CLI を使用してクラスタ設定上のすべての（グローバルレベルの）転送セッションルールのステアリングを無効にするには:

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

- 新しいクラスターインスタンスを追加する場合:
  - クラスターインスタンスの追加 <clid>-プロセスローカル有効
  - クラスタインスタンスを表示 <clid>
- 既存のクラスターインスタンスを再構成する場合:
  - クラスタインスタンスの設定 <clid>-プロセスローカル有効
  - クラスタインスタンスを表示 <clid>

設定例:

次に、転送セッションのルールレベルでステアリングを無効にする 2 つの例と、グローバルレベルでステアリングを無効にする例を示します。

```
An IPv4 forwarding session rule:

> add forwardingSession FWD-SESSN-PROCSS-LOCL-IPV4-1 10.102.105.51
   255.255.255.255 -processLocal Enabled
Done

An IPv6 forwarding session rule:

> add forwardingSession FWD-SESSN-PROCSS-LOCL-IPV6-1  - acl6name ACL6-
   FWD-SESSN-1 -processLocal Enabled
Done

A cluster setup, with an instance ID 10, has steering disabled at
   global level:

> set cluster instance 10 -processLocal Enabled
Done
<!--NeedCopy-->
```

GUI のプロシージャ　GUI を使用してクラスターセットアップの転送セッションルールのステアリングを無効にするには:

[システム] > [ネットワーク] > [転送セッション] に移動し、新しい転送セッションルールを追加するか、既存の転送セッションルールを編集するときに [ローカル処理] を選択します。

GUI を使用してクラスタ設定上のすべての (グローバルレベルの) 転送セッションルールのステアリングを無効にするには:

クラスタ構成を追加するか、既存のクラスタ構成を変更するときに、[システム] > [クラスタ] に移動し、[**Process Local**] を選択します。

VLAN について

August 15, 2023

NetScaler アプライアンスは、レイヤー 2 ポートと IEEE 802.1q タグ付き VLAN をサポートします。VLAN 構成は、トラフィックを特定のワークステーショングループだけに制限しなければならない場合に便利です。IEEE 802.1q タギングを使用して、ネットワークインターフェイスを複数の VLAN の一部として設定できます。

VLAN を設定して IP サブネットにバインドできます。次に、NetScaler ADC はこれらの VLAN 間で IP 転送を実行します（これらのサブネット上のホストのデフォルトルーターとして構成されている場合）。

NetScaler は、次のタイプの VLAN をサポートしています。

- ポートベースの **VLAN**。ポートベースの VLAN のメンバーシップは、共通の排他的レイヤ 2 ブロードキャストドメインを共有する一連のネットワークインターフェイスによって定義されます。複数のポートベース VLAN を構成できます。デフォルトでは、NetScaler 上のすべてのネットワークインターフェイスは VLAN 1 のメンバーです。

  ポートに 802.1q タグを適用すると、ネットワークインターフェイスはポートベースの VLAN に属します。レイヤ 2 トラフィックはポートベースの VLAN 内でブリッジされ、レイヤ 2 モードが有効な場合、レイヤ 2 ブロードキャストは VLAN のすべてのメンバーに送信されます。タグなしネットワークインターフェイスを新しい VLAN のメンバーとして追加すると、現在の VLAN から削除されます。

- デフォルト **VLAN**。デフォルトでは、NetScaler 上のネットワークインターフェイスは、タグなしネットワークインターフェイスとして単一のポートベースの VLAN に含まれます。この VLAN がデフォルト VLAN です。このデバイスの VLAN ID (VID) は 1 です。この VLAN は永続的に存在します。デフォルト VLAN を削除したり、その VID を変更したりすることはできません。

  ネットワークインターフェイスをタグなしメンバーとして別の VLAN に追加すると、そのネットワークインターフェイスはデフォルト VLAN から自動的に削除されます。ネットワークインターフェイスを現在のポートベースの VLAN からバインド解除すると、そのネットワークインターフェイスはデフォルト VLAN に再び追加されます。

- タグ付き **VLAN**。802.1q タギング（IEEE 802.1q 規格で定義されている）を使用すると、ネットワークデバイス（NetScaler など）がレイヤー 2 のフレームに情報を追加して、フレームの VLAN メンバーシップを識別できます。タグ付けにより、ネットワーク環境に複数のデバイスにまたがる VLAN を設定できます。パケットを受信したデバイスはタグを読み取り、フレームが属する VLAN を認識します。一部のネットワークデバイス、特に Force10 スイッチは、同じネットワークインターフェイス上でタグ付きパケットとタグなしパケットの両方を受信することをサポートしていません。このような場合は、カスタマーサポートに連絡して支援を受ける必要があります。

  ネットワークインターフェースは、VLAN のタグ付きメンバーでもタグなしメンバーでもかまいません。各ネットワークインターフェイスは、1 つの VLAN（ネイティブ VLAN）のタグなしメンバーにすぎません。このネットワークインターフェースは、ネイティブ VLAN のフレームをタグなしフレームとして送信します。他の VLAN にタグが付けられていれば、1 つのネットワークインターフェースを複数の VLAN の一部にすることができます。

  タギングを設定するときは、リンクの両端の VLAN の設定を必ず一致させてください。NetScaler が接続するポートは、NetScaler ネットワークインターフェイスと同じ VLAN 上にある必要があります。

  注**:** この VLAN 設定は同期も伝達もされないため、HA ペアの各ユニットで個別に設定を実行する必要があります。

フレーム分類へのルールの適用

VLAN には、フレームを分類するための次の 2 種類のルールがあります。

- 進入ルール。イングレス・ルールでは、各フレームは 1 つの VLAN のみに属するものとして分類されます。ネットワークインターフェースでフレームを受信すると、次のルールが適用されてフレームが分類されます。
  - フレームがタグなしの場合、またはタグ値が 0 の場合、フレームの VID は、ネイティブ VLAN に属するものとして分類される受信側インターフェイスのポート VID（PVID）に設定されます。（PVID は IEEE 802.1q 規格で定義されています）。
  - フレームのタグ値が FFF と等しい場合、そのフレームはドロップされます。
  - フレームの VID に、受信側のネットワークインターフェイスがメンバーではない VLAN が指定されている場合、フレームはドロップされます。たとえば、VLAN ID 12 に関連付けられたサブネットから VLAN ID 10 に関連付けられたサブネットにパケットが送信された場合、そのパケットはドロップされます。VID 9 のタグなしパケットが VLAN ID 10 に関連付けられたサブネットからネットワークインターフェイス PVID 9 に送信されると、そのパケットはドロップされます。
- エグレス・ルール以下のエグレス・ルールが適用されます。
  - フレームの VID が、送信ネットワークインタフェースがメンバーではない VLAN を指定した場合、そのフレームは破棄されます。
  - 学習プロセス（IEEE 802.1q 規格で定義されています）では、Src MAC と VID を使用して NetScaler のブリッジルックアップテーブルが更新されます。
  - VID にメンバーがいない VLAN が指定されている場合、フレームは破棄されます。(メンバーを定義するには、ネットワーク・インタフェースを VLAN にバインドします)。

NetScaler での **VLAN** とパケット転送

NetScaler アプライアンスの転送プロセスは、他の標準スイッチの転送プロセスと同様です。ただし、NetScaler はレイヤー 2 モードがオンの場合にのみ転送を実行します。転送プロセスの主な機能は次のとおりです。

- トポロジーの制限が適用されます。強制適用には、VLAN 内の各ネットワークインターフェイスを送信ポートとして選択すること（ネットワークインターフェイスの状態に応じて）、ブリッジ制限（受信側のネットワークインターフェイスでは転送しない）、および MTU 制限が含まれます。
- フレームは、NetScaler の転送データベース（FDB）テーブルのブリッジテーブル検索の情報に基づいてフィルタリングされます。ブリッジテーブルの検索は、宛先 MAC と VID に基づいています。NetScaler の MAC アドレス宛のパケットは、上位レイヤーで処理されます。
- すべてのブロードキャストフレームとマルチキャストフレームは、VLAN のメンバーである各ネットワークインターフェイスに転送されますが、転送は L2 モードが有効な場合にのみ行われます。L2 モードが無効の場合、ブロードキャストパケットとマルチキャストパケットはドロップされます。これは、現在ブリッジテーブルにない MAC アドレスにも当てはまります。
- VLAN エントリには、タグなしメンバーセットに含まれるメンバーネットワークインターフェイスのリストが含まれます。これらのネットワークインターフェイスにフレームを転送する場合、タグはフレームに挿入されません。
- ネットワークインターフェースがこの VLAN のタグ付きメンバーである場合、フレームが転送されるときにタグがフレームに挿入されます。

ユーザーが VLAN を識別せずにブロードキャストまたはマルチキャストパケットを送信した場合、つまり NSIP の重複アドレス検出（DAD）またはルートのネクストホップの ND6 中に、そのパケットは入力ルールと出力ルールのいずれかに基づいて適切なタグ付けを付けて、すべてのネットワークインターフェイスに送信されます。ND6 は通常、VLAN を識別し、データパケットは VLAN 上でのみ送信されます。ポートベースの VLAN は IPv4 と IPv6 に共通です。IPv6 の場合、NetScaler ADC はプレフィックスベースの VLAN をサポートします。

VLAN の設定

August 15, 2023

VLAN は次の環境で実装できます。

- 単一サブネット
- 複数のサブネット
- シングル LAN
- VLAN (タグ付けなし)
- VLANs (802.1q tagging)

タグ付けされていないネットワークインターフェースのみをメンバーとする VLAN を構成する場合、使用可能な VLAN の総数は NetScaler で使用可能なネットワークインターフェースの数に制限されます。VLAN 構成でより多くの IP サブネットが必要な場合は、802.1q タギングを使用する必要があります。

ネットワークインターフェースを VLAN にバインドすると、そのネットワークインターフェースはデフォルト VLAN から削除されます。ネットワークインターフェイスを複数の VLAN の一部にする必要がある場合は、ネットワークインターフェイスをタグ付きメンバーとして VLAN にバインドできます。

レイヤー 3 の VLAN 間でトラフィックを転送するように NetScaler を構成できます。この場合、VLAN は単一の IP サブネットに関連付けられます。1 つのサブネットに属する VLAN 内のホストは、同じサブネットマスクと、そのサブネットに接続された 1 つ以上のデフォルトゲートウェイを使用します。VLAN のレイヤ 3 の設定は任意です。レイヤ 3 は IP 転送（VLAN 間ルーティング）に使用されます。各 VLAN には、VLAN の IP サブネットを定義する固有の IP アドレスとサブネットマスクがあります。HA 構成では、この IP アドレスは他の NetScaler アプライアンスと共有されます。NetScaler は、構成済みの IP サブネット（VLAN）間でパケットを転送します。

NetScaler を構成するときは、重複する IP サブネットを作成しないでください。そうすると、レイヤ 3 の機能が妨げられます。

各 VLAN は固有のレイヤ 2 ブロードキャストドメインです。それぞれ別々の IP サブネットにバインドされた 2 つの VLAN を 1 つのブロードキャストドメインにまとめることはできません。2 つの VLAN 間でトラフィックを転送するには、NetScaler アプライアンスなどのレイヤー 3 転送（ルーティング）デバイスが必要です。

HA セットアップでの VLAN の設定

高可用性セットアップの VLAN 構成では、NetScaler アプライアンスのハードウェア構成が同じで、その上に構成されている VLAN がミラーイメージである必要があります。

NetScaler アプライアンス間で構成が同期されると、正しい VLAN 構成が自動的に実装されます。その結果、すべてのアプライアンスで同じアクションが実行されます。たとえば、ネットワークインターフェイス 0/1 を VLAN2 に追加すると、このネットワークインターフェイスが高可用性セットアップに含まれるすべてのアプライアンスの VLAN 2 に追加されます。

注：HA セットアップでネットワークインターフェイス固有のコマンドを使用する場合、作成した構成は他の NetScaler アプライアンスに伝達されません。HA ペアの各アプライアンスでこれらのコマンドを実行して、HA ペア内の 2 つのアプライアンスの設定の同期を維持する必要があります。

VLAN の作成または変更

VLAN を設定するには、VLAN エンティティを作成し、ネットワークインターフェイスと IP アドレスを VLAN にバインドします。VLAN を削除すると、そのメンバーインターフェイスがデフォルト VLAN に追加されます。

CLI のプロシージャ

CLI を使用して VLAN を作成するには：

コマンドプロンプトで入力します。

- `add vlan <id> [-aliasName <string>] [-ipv6DynamicRouting (ENABLED|DISABLED)]`
- `sh vlan <id>`

例：

```
  > add vlan 2  -aliasName "Network A"   Done
<!--NeedCopy-->
```

CLI を使用してインターフェイスを VLAN にバインドするには：

コマンドプロンプトで入力します。

- `bind vlan <id> -ifnum <slot/port>`
- `sh vlan <id>`

例：

```
  > bind vlan 2 -ifnum 1/8   Done
<!--NeedCopy-->
```


CLI を使用して IP アドレスを VLAN にバインドするには:

コマンドプロンプトで入力します。

- `bind vlan <id> -IPAddress <IPAddress> <netMask>`
- `sh vlan <id>`

例:

```
  > bind vlan 2 -IPAddress 10.102.29.54 255.255.255.0   Done
<!--NeedCopy-->
```

CLI を使用して VLAN を削除するには:

コマンドプロンプトで入力します。

- `rm vlan <id>`

GUI のプロシージャ

GUI を使用して VLAN を設定するには:

1. [システム] > [ネットワーク] > [VLAN] に移動するか、新しい VLAN を追加するか、既存の VLAN を編集します。
2. IP アドレスを VLAN にバインドするには、「IP バインディング」で、VLAN にバインドする IP アドレスに対応する「アクティブ」オプション（たとえば、10.102.29.54）を選択します。「タイプ」列には、「IP アドレス」列の各 IP アドレスの IP アドレスのタイプ (マップ IP、仮想 IP、サブネット IP など) が表示されます。
3. ネットワークインターフェースを VLAN にバインドするには、「インターフェースバインディング」で、VLAN にバインドするインターフェースに対応する「アクティブ」オプションを選択します。

VLAN のモニタリング

受信パケット、受信バイト、送信パケット、送信バイトなどの VLAN 統計情報を表示し、その情報を使用して異常を特定したり、VLAN をデバッグしたりできます。

CLI を使用して VLAN の統計情報を表示するには:

コマンドプロンプトで入力します。

- `stat vlan <vlanID>`

例:

```
  stat vlan 2
<!--NeedCopy-->
```

GUI を使用して VLAN の統計情報を表示するには:

1. [システム] > [ネットワーク] > [VLAN] に移動します。
2. VLAN を選択し、[統計情報] をクリックします。

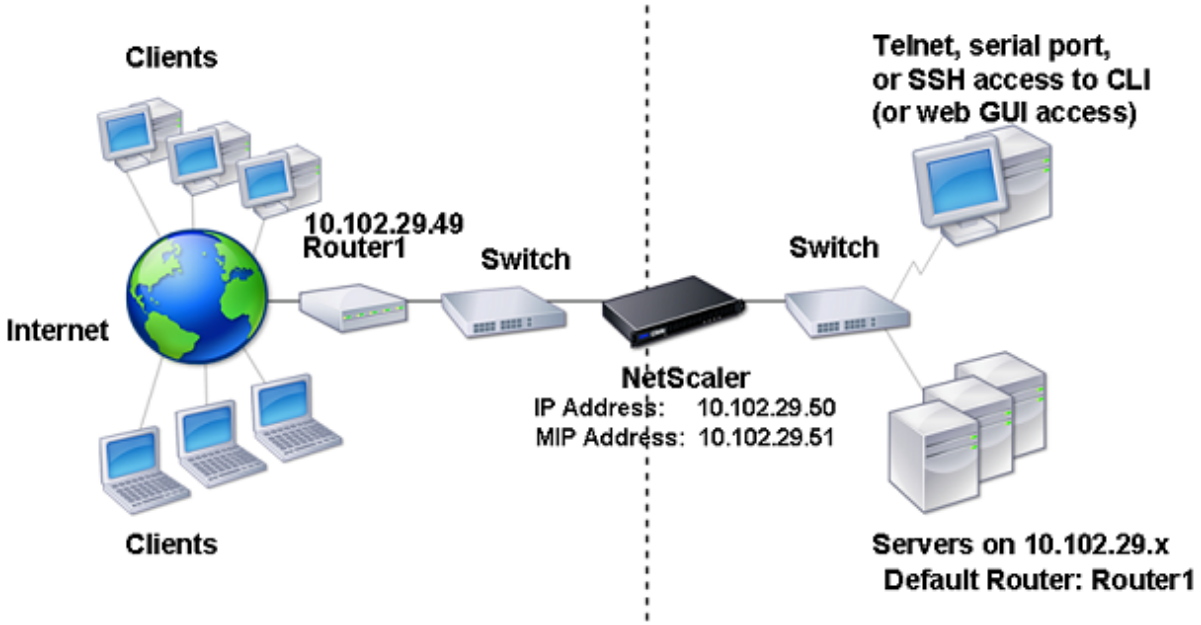
単一のサブネットでの **VLAN** の構成

August 15, 2023

1 つのサブネットに VLAN を設定する前に、レイヤ 2 モードが有効になっていることを確認してください。

次の図は、単一のサブネット環境を示しています。

図 1：単一サブネット上の VLAN



上の図では、

1. NetScaler とサーバーのデフォルトルーターはルーター 1 です。
2. NetScaler がサーバーに直接アクセスするには、NetScaler でレイヤー 2 モードを有効にする必要があります。
3. このサブネットでは、NetScaler ADC アプライアンスの負荷分散用に仮想サーバーを構成できます。

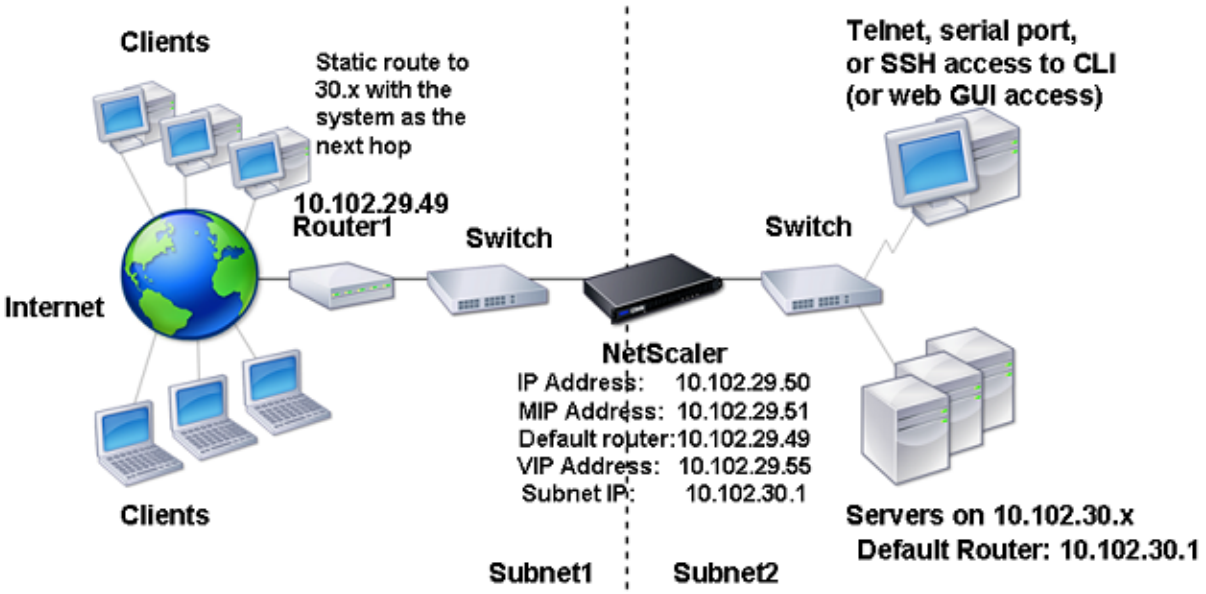
1 つのサブネットに VLAN を設定するには、VLAN の設定で説明されている手順に従います。

複数のサブネットでの **VLAN** の構成

August 15, 2023

複数のサブネットにまたがる単一の VLAN を設定するには、VLAN に VIP を追加し、ルーティングを適切に設定する必要があります。次の図は、複数のサブネットにまたがって設定された単一の VLAN を示しています。

図 1: 単一 VLAN 内の複数のサブネット



複数のサブネットにまたがる単一の VLAN を設定するには、次のタスクを実行します。

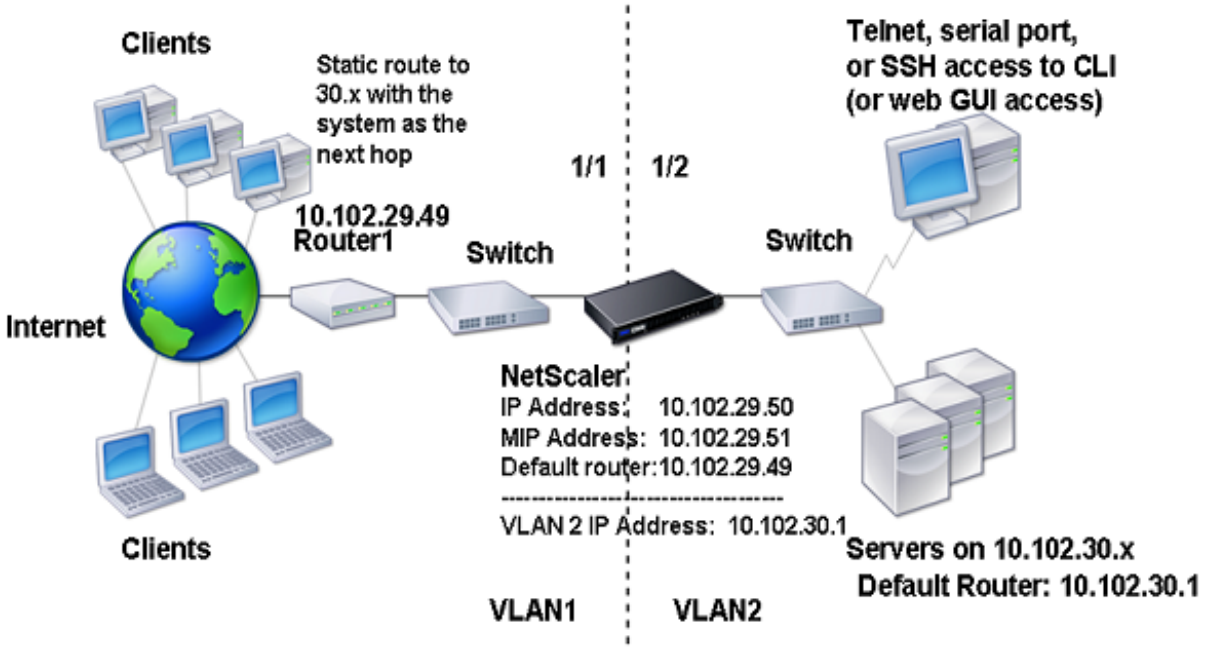
1. レイヤ 2 モードを無効にします。レイヤ 2 モードを無効にする手順については、パケット転送モードを参照してください。
2. VIP アドレスを追加します。VIP アドレスを追加する手順については、仮想 IP アドレス (VIP) の設定と管理を参照してください。
3. RNAT ルールを設定します。RNAT ID を設定する手順については、「RNAT の設定」を参照してください。

複数のサブネットにまたがる複数のタグなし **VLAN** の構成

August 15, 2023

複数のサブネットに複数のタグなし VLAN がある環境では、IP サブネットごとに 1 つの VLAN が設定されます。ネットワークインターフェイスは 1 つの VLAN にのみバインドされます。次の図は、この構成を示しています。

図 1: VLAN を含む複数のサブネット-タグ付けなし



上の図に示す構成を実装するには、次のタスクを実行します。

1. VLAN 2 を追加します。
2. NetScaler の 1/2 ネットワークインターフェースをタグなしネットワークインターフェースとして VLAN 2 にバインドします。
3. IP アドレスとサブネットマスクを VLAN 2 にバインドします。

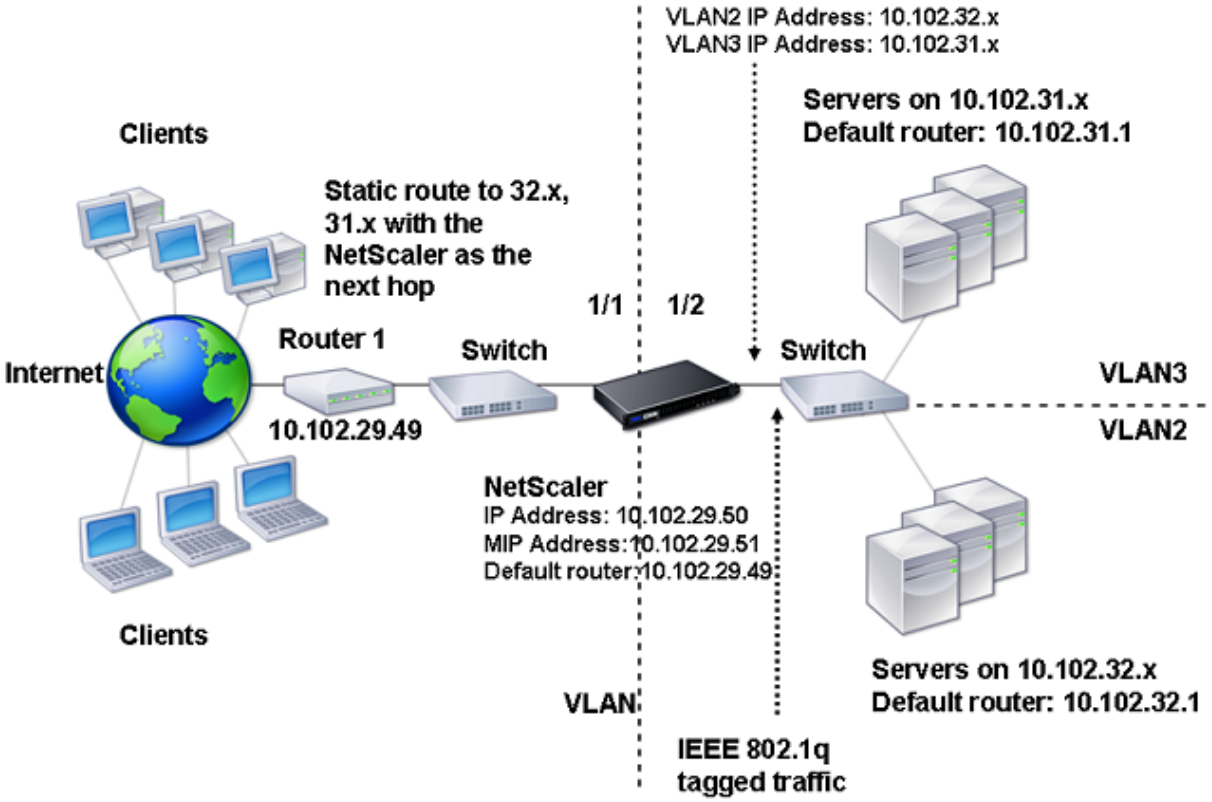
これらのタスクの手順については、VLAN の設定を参照してください。

802.1q タグ付けを使用した複数の VLAN の構成

August 15, 2023

802.1q タグが設定された複数の VLAN では、各 VLAN は異なる IP サブネットで設定されます。各ネットワークインターフェイスは 1 つの VLAN にあります。VLAN の 1 つがタグ付きに設定されています。次の図は、この構成を示しています。

図 1: IEEE 802.1q タギングを備えた複数の VLAN



上の図に示す構成を実装するには、次のタスクを実行します。

1. VLAN 2 を追加します。
2. NetScaler の 1/2 ネットワークインターフェースをタグなしネットワークインターフェースとして VLAN 2 にバインドします。
3. IP アドレスとネットマスクを VLAN 2 にバインドします。
4. VLAN 3 を追加します。
5. NetScaler の 1/2 ネットワークインターフェースをタグ付きネットワークインターフェースとして VLAN 3 にバインドします。
6. IP アドレスとネットマスクを VLAN 3 にバインドします。

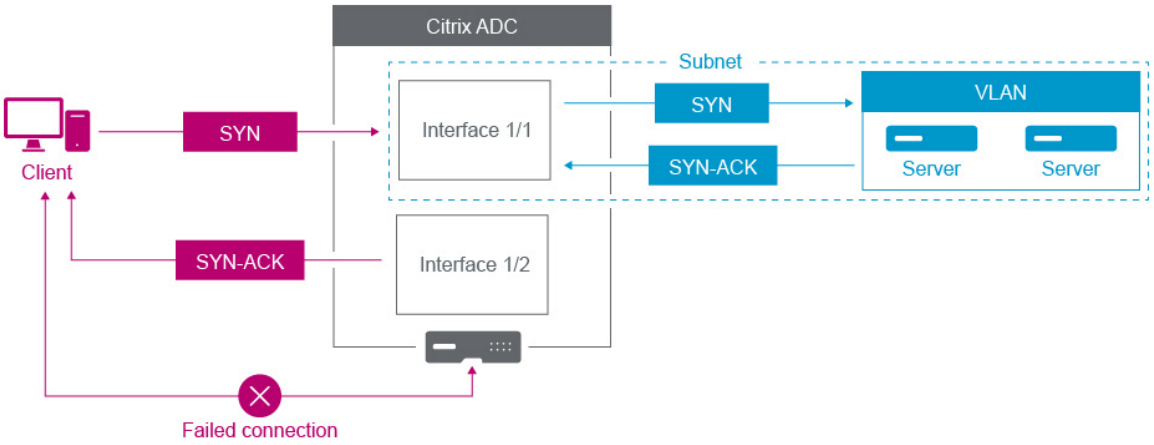
これらのタスクの手順については 、VLAN の設定を参照してください。

VLAN を使用して IP サブネットを NetScaler インターフェイスに関連付ける

August 15, 2023

デフォルトでは、NetScaler ADC アプライアンスはネットワークインターフェースを区別しません。アプライアンスは、スイッチよりもネットワークハブのように機能します。これにより、重複したトラフィックが複数のインターフェイスで送信されるレイヤ 3 ネットワークループが発生する可能性があります。

このようなシナリオでは、ネットワークの設計によっては、要求が 1 つのインターフェイスで送信され、対応する応答が別のインターフェイスで受信されることがあります。



たとえば、アプライアンスは元の SYN パケットを送信したのと同じインターフェイスで SYN-ACK を受信することを想定しているため、あるインターフェイスで送信された SYN パケットと別のインターフェイスで SYN-ACK 応答が受信されると、接続が失敗する可能性があります。

このような問題を解決するために、アプライアンスは内部または外部の VLAN を使用して特定のサブネットをインターフェイスに関連付けることができます。

はじめに

VLAN を使用して IP サブネットを NetScaler インターフェイスに関連付ける前に、次の点に注意してください。

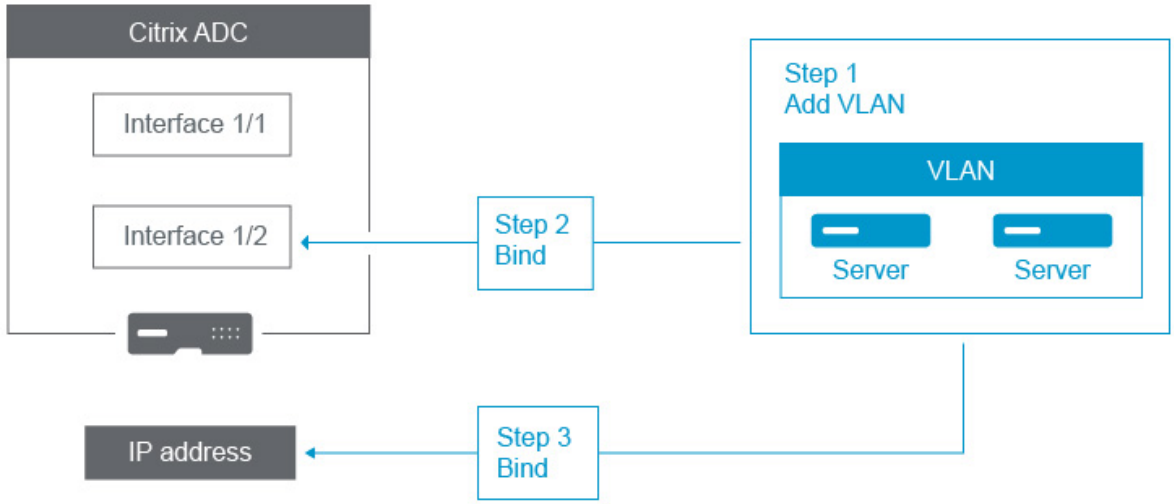
- NetScaler GUI またはコマンドラインインターフェイスへのアクセスに現在使用されているサブネットまたはインターフェイスに VLAN を関連付けると、ネットワーク接続が誤って失われることがあります。そのため、このようなシナリオでは、物理 NetScaler アプライアンスのシリアルコンソールまたは NetSNetScaler VPX 仮想シリアルコンソールからコマンドラインインターフェイスにアクセスして変更を行うことを強くお勧めします。
- NetScaler 管理インターフェイスには特定のハードウェア最適化機能が欠けているため、本番データトラフィックでの使用にはあまり適していません。そのため、NetScaler は管理（NSIP）トラフィックに管理インターフェイスのみを使用するように構成することをお勧めします。デフォルト構成では、ハードウェア NetScaler の管理インターフェイスとデータインターフェイスの間に論理的な違いはありません。この目標を達成するには、NSIP をデータトラフィックとは別の VLAN に配置することを推奨します。これにより、管理トラフィックは別のインターフェイスに配置できます。

  概念は同じですが、NSIP アドレスを含むサブネットの VLAN アソシエーションを変更するには、以下の手順の代わりに NSVLAN を設定する必要があります。また、このような変更を有効にするには、NetScaler ADC を再起動する必要があります。詳細については、NSVLAN の設定を参照してください。
- NetScaler SDX では、各インスタンスの NSIP を、SDX の SVM（管理サービス GUI）および XenServer と同じサブネットおよび VLAN 上に置くことを強くお勧めします。SVM はネットワーク経由でインスタンスと

通信します。SVM、XenServer、およびインスタンスが同じ VLAN とサブネット上にない場合、管理トラフィックは SDX の外部を流れる必要があります。このような状況では、ネットワークの問題によりインスタンスの状態が黄色または赤で表示され、NetScaler インスタンスの管理と構成の変更ができなくなる可能性があります。

構成の手順

IP サブネットを NetScaler インターフェイスに関連付けるには、次のタスクが含まれます。



VLAN を追加します。VLAN を追加する際、VLAN にタグを付ける場合は、関連するスイッチポートのネットワークスイッチで定義されている VLAN 番号を選択する必要があります。VLAN にタグが付いておらず、アプライアンスの内部にある場合は、簡単に参照できるように、スイッチ構成で使用可能な VLAN 番号を選択することをお勧めします。

インターフェイスを **VLAN** にバインドします。バインディング中にリンクアグリゲーションを使用している場合は、VLAN を物理インターフェイスではなく LA チャネル（たとえば、LA/1）に関連付けます。VLAN は 1 つのネットワークインターフェースにのみ関連付ける必要があります。

インターフェイス上のトラフィックにタグを付ける場合は、tagged (Tag) オプションを使用してください。それ以外の場合、トラフィックはアプライアンスにタグ付けされず、スイッチポートのネイティブ VLAN に関連付けられます。

IP アドレスを **VLAN** にバインドします。バインド中に、同じサブネットから複数の IP アドレスをバインドすると、エラーが発生します。IP アドレスを VLAN に関連付けると、そのサブネット内のすべての IP アドレスが自動的に VLAN に関連付けられます。

> 注:
>
> 高可用性（HA）セットアップでは、これらの VLAN 構成は、HA 同期中にプライマリノードからセカンダリノードに自動的に追加されます。高可用性設定の詳細については、「高可用性」を参照してください。

CLI のプロシージャ

CLI を使用して VLAN を追加するには:

コマンドプロンプトで入力します。

- **add vlan** <id>
- **sh vlan** <id>

CLI を使用してインターフェイスを VLAN にバインドするには:

コマンドプロンプトで入力します。

- **bind vlan** <id> -**ifnum** <slot/port>
- **sh vlan** <id>

CLI を使用して IP アドレスを VLAN にバインドするには:

コマンドプロンプトで入力します。

- **bind vlan** <id> -**IPAddress** <IPAddress> <netMask>
- **sh vlan** <id>

例**:**

```
> add vlan 100

> bind vlan 100 -ifnum 1/1

> bind vlan 100 -ipAddress 10.0.1.0 255.255.255.0
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して VLAN を設定するには:

1. [システム] > [ネットワーク] > [**VLAN**] に移動し、新しい VLAN を追加します。
2. ネットワークインターフェースを VLAN にバインドするには、「インターフェースバインディング」で、VLAN にバインドするインターフェースに対応する「アクティブ」オプションを選択します。
3. IP アドレスを VLAN にバインドするには、「**IP** バインディング」で、VLAN にバインドする IP アドレスに対応する「アクティブ」オプション（たとえば、10.102.29.54）を選択します。「タイプ」列には、「IP アドレス」列の各 IP アドレスの **IP** アドレスタイプが表示されます。

NetScaler アプライアンスのネットワークと **VLAN** のベストプラクティス

August 15, 2023

NetScaler アプライアンスは VLAN を使用して、どのインターフェイスをどのトラフィックに使用する必要があるかを判断します。また、NetScaler アプライアンスはスパニングツリーに参加しません。適切な VLAN 構成がないと、NetScaler アプライアンスはどのインターフェイスを使用するかを判断できず、スイッチやルーターというよりは HUB のように機能する可能性があります。つまり、NetScaler アプライアンスは各会話ですべてのインターフェイスを使用できます。

VLAN の設定ミスの症状

VLAN の構成ミスの問題は、パフォーマンスの問題、接続を確立できない、セッションがランダムに切断される、深刻な状況では NetScaler アプライアンス自体とは無関係に見えるネットワークの中断など、さまざまな形で現れます。NetScaler アプライアンスは、ネットワークとのやりとりの性質によっては、MAC の移動、インターフェイスのミュート、管理インターフェイスの送受信バッファオーバーフローを報告する場合もあります。

MAC 移動（**nic_tot_bdg_mac_moved**）：この問題は、NetScaler アプライアンスが使用するインターフェイスを適切に決定できなかったため、同じデバイス（MAC アドレス）との通信に複数のインターフェイスを使用していることを示しています。

ミュートされたインターフェイス（**nic_err_bdg_muted**）：この問題は、NetScaler アプライアンスが VLAN 構成の問題によりルーティングループを作成していることを検出し、ネットワークの停止を防ぐために問題のある 1 つ以上のインターフェイスをシャットダウンしたことを示しています。

インターフェイスバッファオーバーフロー。通常は管理インターフェイスを指します **(nic_err_tx_overflow):** この問題は、管理インターフェイスを介して送信されるトラフィックが多すぎる場合に発生する可能性があります。NetScaler アプライアンスの管理インターフェイスは、大量のトラフィックを処理するようには設計されていません。これは、NetScaler アプライアンスが運用データトラフィックに管理インターフェイスを使用するきっかけとなるネットワークや VLAN の設定ミスが原因である可能性があります。これは、NetScaler アプライアンスが NSIP（NSVLAN）の VLAN/サブネット上のトラフィックを通常の本番トラフィックと区別する方法がないために発生することがよくあります。NSIP は、ワークステーションやサーバなどの本番デバイスとは別の VLAN とサブネットに配置することを強く推奨します。

オーファン **ACK**（**tcp_err_orphan_ack** カウンター）：この問題は、NetScaler アプライアンスが予期していなかった ACK パケットを受信したことを示します。通常、ACK のトラフィックの送信元とは異なるインターフェイスで送信されます。この状況は、NetScaler アプライアンスが、ターゲットデバイスが NetScaler アプライアンスとの通信に通常使用するインターフェイスとは異なるインターフェイスで送信する（MAC の移動と関連してよく見られる）VLAN の構成ミスが原因である可能性があります。

再送信または再送信のギブアップ率が高い(カウンター:**tcp_err_retransmit_giveups**、**tcp_err_7th_retransmit**、その他のさまざまな再送信カウンター)：**NetScaler** アプライアンスは、**TCP** パケットの再送信を合計 **7** 回試行し

てから、接続をあきらめて終了します。この状況はネットワークの状態によって発生することもありますが、多くの場合、VLAN とインターフェイスの設定ミスが原因で発生します。

高可用性スプリットブレーン：スプリットブレインとは、両方の高可用性ノードが自分たちをプライマリと見なす状態で、IP アドレスが重複し、NetScaler アプライアンスの機能が失われます。これは、2 つの高可用性ノードが、NSIP を使用して UDP ポート 3003 で高可用性ハートビートを使用して、どのインターフェイスでも相互に通信できない場合に発生します。これは通常、NetScaler アプライアンスインターフェイスのネイティブ VLAN が NetScaler アプライアンス間で接続されていないという VLAN の構成ミスが原因です。

VLAN とネットワーク構成のベストプラクティス

1. 各サブネットは VLAN に関連付ける必要があります。
2. （ネットワークの設計に応じて）複数のサブネットを同じ VLAN に関連付けることができます。
3. 各 VLAN は 1 つのインターフェイスにのみ関連付ける必要があります（この説明では、LA チャネルは 1 つのインターフェイスとしてカウントされます）。
4. インターフェイスに複数のサブネットを関連付ける必要がある場合は、サブネットにタグを付ける必要があります。
5. 一般に信じられていることとは反対に、NetScaler アプライアンスの Mac ベース転送（MBF）機能はこの種の問題を軽減するようには設計されていません。MBF は、主に NetScaler アプライアンスの DSR（Direct Server Return）モード向けに設計されており、ほとんどの環境ではほとんど使用されません（バックエンドサーバーからのリターンパスでトラフィックが NetScaler アプライアンスを意図的にバイパスできるように設計されています）。MBF は VLAN の問題を隠す場合がありますが、この種の問題を解決するうえで信頼すべきではありません。
6. NetScaler アプライアンスのすべてのインターフェイスにはネイティブ VLAN が必要です（ネイティブ VLAN がオプションである Cisco とは異なります）。ただし、インターフェイスの tagAll 設定を使用して、タグ付けされていないトラフィックが問題のインターフェイスから出ないようにできます。
7. ネットワーク設計に必要な場合は、ネイティブ VLAN にタグを付けることができます (これはインターフェイスの tagAll オプションです)。
8. NetScaler アプライアンスの NSIP のサブネットの VLAN は特殊なケースです。これは NSVLAN と呼ばれます。概念は同じですが、構成するコマンドが異なり、NSVLAN への変更を有効にするには、NetScaler アプライアンスを再起動する必要があります。NSIP と同じサブネットを共有する SNIP に VLAN をバインドしようとすると、「操作は許可されていません」というメッセージが表示されます。これは、代わりに NSVLAN コマンドを使用する必要があるためです。また、一部のファームウェアバージョンでは、add VLAN コマンドを使用して NSVLAN 番号が存在する場合、NSVLAN を設定できません。VLAN を削除してから、NSVLAN を再設定するだけです。
9. 高可用性ハートビートは常にそれぞれのインターフェースのネイティブ VLAN を使用します (インターフェースで tagAll オプションが設定されている場合はオプションでタグ付けできます)。

10. 高可用性ペアの 2 つのノード上の少なくとも 1 つのネイティブ VLAN セット間で通信が必要です（直接通信することも、ルーター経由で行うこともできます）。ネイティブ VLAN は高可用性ハートビートに使用されます。NetScaler アプライアンスがどのインターフェイス上のネイティブ VLAN 間でも通信できない場合、高可用性フェイルオーバーが発生し、両方の NetScaler アプライアンスが自分たちをプライマリと見なすスプリットブレイン状態になる可能性があります（とりわけ IP アドレスが重複する）。

11. NetScaler アプライアンスはスパニングツリーに参加しません。そのため、NetScaler アプライアンスを使用する場合、スパニングツリーを使用してインターフェイスの冗長性を提供することはできません。代わりに、この目的にはリンクアグリゲーション（LACP または手動 LAG）を使用してください。

    注: 複数の物理スイッチ間でリンクアグリゲーションを行う場合は、Cisco のスイッチスタックなどの機能を使用して、スイッチを仮想スイッチとして構成する必要があります。

12. 高可用性同期とコマンド伝播は、デフォルトで NSIP/NSVLAN を使用します。これらを別の VLAN に分離するには、コマンドの SyncVLAN オプションを使用できます。`set HA node`

13. NetScaler アプライアンスのデフォルト構成には、管理インターフェイス（0/1 または 0/2）が管理トラフィックのみに制限されていることを示すようなビルトインはありません。この制限は、エンドユーザが VLAN 設定を通じて適用する必要があります。管理インターフェイスはデータトラフィックを処理するようには設計されていないため、ネットワーク設計ではこの点を考慮する必要があります。NetScaler アプライアンスのマザーボードに含まれる管理インターフェイスには、CRC オフロード、より大きなパケットバッファ、その他の最適化などのさまざまなオフロード機能がないため、大量のトラフィックの処理効率が大幅に低下します。本番データトラフィックと管理トラフィックを分離するには、NSIP をデータトラフィックと同じサブネット/VLAN に配置しないでください。

14. 管理インターフェイスを使用して管理トラフィックを伝送する場合は、デフォルトルートを NSIP（NSVLAN）のサブネット以外のサブネットに配置するのがベストプラクティスです。

    多くの構成では、（インターネットシナリオの）ワークステーション通信にはデフォルトルートが使用されます。デフォルトルートが NSIP と同じサブネット上にある場合、ADC アプライアンスは管理インターフェイスを使用してデータトラフィックを送受信できます。このようなデータトラフィックの使用は、管理インターフェイスに過負荷をかける可能性があります。

15. また、SDX: SVM、XenServer、およびすべての NetScaler インスタンスの NSIP は、同じ VLAN とサブネット上にある必要があります。SDX アプライアンスには、**SVM/Xen/**インスタンス間の通信を可能にするバックプレーンはありません。これらが同じ VLAN /サブネット/インターフェイス上にない場合、それらの間のトラフィックは物理ハードウェアから出て、ネットワーク上でルーティングされてから戻る必要があります。

    この設定では、インスタンスと SVM 間の接続に明らかな問題が発生する可能性があるため、お勧めしません。この場合によく見られる症状は、問題の VPX インスタンスの SVM に黄色のインスタンス状態インジケータが表示され、SVM を使用して VPX インスタンスを再構成できないことです。

16. 一部の VLAN がサブネットにバインドされ、一部がバインドされていない場合、高可用性フェールオーバー中に、VLAN にバインドされていないサブネットの IP アドレスには GARP パケットは送信されません。この構成では、高可用性フェイルオーバー中に接続が切断されたり、接続の問題が発生したりする可能性がありま

す。この問題は、NetScaler アプライアンスが VMAC 構成でない NetScaler アプライアンスのネットワーク MAC 所有権 IP アドレスの変更を通知できないために発生します。

この症状としては、高可用性フェイルオーバー中または後に、以前のプライマリ NetScaler アプライアンスの ip_tot_floating_ip_err カウンターが数秒以上増加します。これは、ネットワークが GARP パケットを受信または処理せず、ネットワークが新しいセカンダリ NetScaler アプライアンスにデータを送信し続けていることを示しています。

NSVLAN の構成

August 15, 2023

NSVLAN は、NetScaler ADC 管理 IP（NSIP）アドレスのサブネットがバインドされている VLAN です。NSIP サブネットは、NSVLAN に関連付けられたインターフェイスでのみ使用できます。デフォルトでは、NSVLAN は VLAN 1 ですが、別の VLAN を NSVLAN として指定できます。その場合、変更を有効にするために NetScaler ADC アプライアンスを再起動する必要があります。再起動後、NSIP サブネットトラフィックは新しい NSVLAN に制限されます。

NetScaler IP サブネットからのトラフィックには、NSVLAN に指定された VLAN ID でタグ付け（802.1q）できます。接続されているスイッチインターフェイスに、接続されているインターフェイスでこれと同じ VLAN ID をタグ付けして許可するように設定する必要があります。NSVLAN 構成を削除すると、NSIP サブネットは自動的に VLAN1 にバインドされ、デフォルトの NSVLAN が復元されます。

CLI を使用して **NSVLAN** を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

---

set ns config -nsvlan -**ifnum** …[-**tagged**　　NO)]
(YES

-
- **show ns config**

> 注:
>
> この構成は、NetScaler ADC アプライアンスの再起動後に有効になります。

例:

```
> set ns config -nsvlan 300 -ifnum 1/1 1/2 1/3 -tagged YES
 Done

```



```
> save config
 Done
<!--NeedCopy-->
```

CLI を使用してデフォルトの **NSVLAN** 構成を復元するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **ns** コンフィグの設定解除 -**nsvlan**
- **show ns config**

例**:**

```
> unset ns config -nsvlan
 Done
<!--NeedCopy-->
```

GUI を使用して **NSVLAN** を設定するには、次の手順を実行します。

[システム**] > [**設定**]** に移動し、[設定] グループで [**NSVLAN** 設定の変更] をクリックします。

NSVLAN で **MTU** を設定する

デフォルトでは、NSVLAN の MTU は 1500 バイトに設定されています。この設定を変更して、スループットとネットワークパフォーマンスを最適化できます。たとえば、ジャンボフレームを処理するように NSVLAN を設定できます。

CLI を使用して NSVLAN の MTU を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **VLAN** を設定 <id>**-mtu** <positive_integer>
- **show vlan** <id>

GUI を使用して NSVLAN の MTU を設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [**VLAN**] に移動し、NSVLAN を開き、[最大伝送単位] パラメータを設定します。

設定例:

次の設定例では、VLAN 100 が NSVLAN です。

```
> set ns config -nsvlan 100 -ifnum 1/1 -tagged no

  Warning: The configuration must be saved and the system rebooted for
     these settings to take effect

> set vlan 100 -mtu 1600

  Done

```



```
> sh vlan

  1)  VLAN ID: 1

      Link-local IPv6 addr:
    fe80::947b:52ff:fead:12d5/64

      Interfaces : 1/2 LO/1

  2)  VLAN ID: 100    VLAN Alias Name:

      MTU: 1600

      Interfaces : 1/1

      IPs :

          10.102.53.114     Mask: 255.255.255.0

  Done

> save config

  Done
<!--NeedCopy-->
```

VLAN 許可リストの構成

August 15, 2023

NetScaler アプライアンスで VLAN が明示的に構成され、インターフェイスが VLAN にバインドされている場合、NetScaler はインターフェイス上の VLAN のタグ付きパケットを受け入れて送信します。一部の展開（Bump in the Wire など）では、NetScaler アプライアンスが多数の VLAN に関連するタグ付きパケットを受け入れて転送するための透過的なデバイスとして機能する必要があります。この要件では、多数の VLAN を設定および管理することは現実的な解決策ではありません。

インターフェイスで許可される VLAN リストは、VLAN のリストを指定します。インターフェイスは、アプライアンス上でこれらの VLAN を明示的に設定しなくても、指定した VLAN に関連するタグ付きパケットを透過的に受け入れて送信します。

許可 VLAN リストを設定する前に考慮すべきポイント

許可 VLAN リストを設定する前に、次の点を考慮してください。

- 高可用性設定では、許可された VLAN リストは伝播も同期もされません。そのため、両方のノードで許可される VLAN リストを設定する必要があります。

- ネイティブ VLAN のトラフィックは、許可 VLAN リストでネイティブ VLAN を指定している非メンバーインターフェイスに漏れる可能性があります。
- インターフェイスで許可される VLAN リストには、最大 60 の VLAN 範囲を指定できます。
- NetScaler ADC アプライアンスは、リンク集約チャネルまたは冗長インターフェイスセットの一部であるインターフェイス上の許可された VLAN リストをサポートしていません。冗長インターフェイスセットの詳細については、冗長インターフェイスセットを参照してください。
- 許可された VLAN リストは、NetScaler ADC クラスタ構成ではサポートされません。
- NetScaler アプライアンスは、ブリッジグループの許可 VLAN リストをサポートしていません。
- NetScaler アプライアンスは、VXLAN の許可 VLAN リストをサポートしていません。

VLAN 許可リストの構成

CLI を使用して許可された VLAN リストを設定するには:

コマンドプロンプトで入力します。

- **set interface** <id> **-trunkmode** (ON|OFF) **-trunkAllowedVlan** <int[-int]> …
- インターフェースを表示 < id>

GUI を使用して許可された VLAN リストを設定するには:

[システム] > [ネットワーク] > [インターフェイス] に移動し、ネットワークインターフェイスを選択して [編集] をクリックし、次のパラメータを設定します。

- トランクモード
- トランク許可 VLAN

設定例**:**

次の設定例では、100 ～120、190 ～200、300 ～330 の範囲の VLAN がインターフェイス 1/2 で許可される VLAN リストの一部として指定されています。

```
> set int 1/2 -trunkmode on -trunkallowedVlan 100-120 190-200 300-330

Done

> sh int 1/2

1)      Interface 1/2 (Gig Ethernet 10/100/1000 MBits) #6
        flags=0xc020

        <ENABLED, UP, UP, AUTONEG OFF, HEARTBEAT, 802.1q, trunkmode>

        Trunk Allowed Vlans:  100-120 190-200 300-330

Done

<!--NeedCopy-->
```


ブリッジグループの構成

August 15, 2023

通常、2 つ以上の VLAN を 1 つのドメインに統合する場合は、個別のドメイン内のすべてのデバイスの VLAN 設定を変更します。これは面倒な作業です。複数の VLAN を 1 つのブロードキャストドメインに簡単に統合するには、ブリッジグループを使用できます。

ブリッジグループ機能は VLAN と同じように機能します。複数の VLAN を 1 つのブリッジグループにバインドでき、同じブリッジグループにバインドされたすべての VLAN は 1 つのブロードキャストドメインを形成します。1 つのブリッジグループにバインドできるのは、レイヤ 2 VLAN だけです。レイヤ 3 機能を使用するには、ブリッジグループに IP アドレスを割り当てる必要があります。

レイヤ 2 モードでは、特定の VLAN に属するインターフェイスで受信したブロードキャストパケットは、同じブリッジグループに属する他の VLAN にブリッジされます。ユニキャストパケットの場合、NetScaler アプライアンスはブリッジテーブルを検索して、同じブリッジグループに属するすべての VLAN の学習済み MAC アドレスを探します。

レイヤ 3 転送モードでは、IP サブネットはブリッジグループにバインドされます。NetScaler は、バインドされたサブネットに属する受信パケットを受け入れ、ブリッジグループにバインドされた VLAN でのみパケットを転送します。

IPv6 ルーティングは、設定されたブリッジグループで有効にできます。

> 注
>
> ブリッジグループ機能とブリッジ BPDU モードは連携できません。

構成の手順

ブリッジグループを設定するには、次の手順を実行します。

- レイヤ 2 モードを有効にする
- ブリッジグループを追加し、そのブリッジグループに VLAN をバインドします。

CLI のプロシージャ

CLI を使用してレイヤ 2 モードを有効にするには、
コマンドプロンプトで次のように入力します。

- **ns** モード **l2** を有効にする
- **show ns mode**

CLI を使用してブリッジグループを追加し、VLAN をバインドするには:

コマンドプロンプトで入力します。

- **add bridgegroup** <id> [-**ipv6DynamicRouting** (**ENABLED** | ** 無効 **)]
- **bind bridgegroup** <id> -**vlan** <positive_integer>
- ブリッジグループを表示 <id>

例**:**

```
1 > add bridgegroup 12
2  Done
3 <!--NeedCopy-->
```

CLI を使用してブリッジグループを削除するには:

コマンドプロンプトで入力します。

- **RM** ブリッジグループ <id>

例**:**

```
1  rm bridgegroup 12
2 <!--NeedCopy-->
```

GUI 手順

GUI を使用してブリッジグループを設定するには:

[システム] > [ネットワーク] > [ブリッジグループ] に移動し、新しいブリッジグループを追加して VLAN をブリッジグループにバインドするか、既存のブリッジグループを編集します。

仮想 **MAC** の設定

August 15, 2023

高可用性 (HA) セットアップのプライマリノードとセカンダリノードは、仮想 MAC アドレスのフローティングエンティティを共有します。プライマリノードはフローティング IP アドレス (MIP、SNIP、VIP など) を所有し、これらの IP アドレスに対する ARP 要求に独自の MAC アドレスで応答します。そのため、アップストリームルーターなどの外部デバイスの ARP テーブルは、プライマリノードのフローティング IP アドレスと MAC アドレスで更新されます。

フェイルオーバーが発生すると、セカンダリノードが新しいプライマリノードとして引き継がれます。前のセカンダリノードは、Gratuitous ARP (GARP) を使用して、古いプライマリノードから学習したフローティング IP アドレスをアドバタイズします。新しいプライマリノードが通知する MAC アドレスは、自身のネットワークインターフェイスの MAC アドレスです。一部のデバイス（一部のルータ）は、これらの GARP メッセージを受け入れません。したがって、これらの外部デバイスには、古いプライマリノードが通知していた IP アドレスと MAC アドレスのマッピングが保持されます。これにより、GSLB サイトがダウンする可能性があります。

そのため、HA ペアの両方のノードに仮想 MAC を設定する必要があります。つまり、両方のノードの MAC アドレスは同じです。フェールオーバーが発生しても、セカンダリノードの MAC アドレスは変更されず、外部デバイスの ARP テーブルを更新する必要はありません。

仮想 MAC を設定する手順については、仮想 MAC アドレスの設定を参照してください。

リンクアグリゲーションの構成

August 15, 2023

リンクアグリゲーションは、複数のポートからのデータを 1 つの高速リンクに結合します。リンクアグリゲーションを構成すると、NetScaler アプライアンスと他の接続デバイス間の通信チャネルの容量と可用性が向上します。集約されたリンクは「チャネル」とも呼ばれます。チャネルは手動で設定することも、Link Aggregation Control Protocol (LACP) を使用することもできます。LACP を手動で設定したチャネルに適用したり、LACP によって作成されたチャネルを手動で設定したりすることはできません。

ネットワークインターフェイスをチャネルにバインドした場合、チャネルのパラメーターは、ネットワークインターフェイスのパラメーターよりも優先されます。(つまり、ネットワークインターフェイスパラメータは無視されます)。ネットワークインターフェイスは 1 つのチャネルにのみバインドできます。

ネットワークインターフェイスがチャネルにバインドされると、その VLAN 設定は破棄されます。ネットワークインターフェースが手動で、または LACP によってチャネルにバインドされると、元々所属していた VLAN から削除され、デフォルト VLAN に追加されます。ただし、チャネルを元の VLAN や新しい VLAN にバインドすることができます。たとえば、ネットワークインターフェイス 1/2 と 1/3 を ID 2 の VLAN にバインドし、次にチャネル LA/1 にバインドすると、ネットワークインターフェイスはデフォルト VLAN に移動されますが、VLAN 2 にバインドし直すことができます。

リンクアグリゲーションの手動設定

リンクアグリゲーションチャネルを作成すると、アクティブなインターフェイスをそのチャネルにバインドするまでは DOWN 状態になります。チャンネルはいつでも変更できます。チャンネルを削除することも、有効化/無効化することもできます。

CLI のプロシージャ

CLI を使用してリンク・アグリゲーション・チャネルを作成するには:

コマンドプロンプトで入力します。

- add channel <id> [-ifnum \ ...] \[-state \(ENABLED | DISABLED)] \[-speed \\] \[-flowControl \\] \[-haMonitor \(ON | OFF)]\[-tagall \(ON | OFF)] \[-ifAlias \\] \[-throughput \\<positive_integer>] \[-bandwidthHigh \\<positive_integer> \[-bandwidthNormal \\<positive_integer>]]
- チャンネルを表示

例:

```
> add channel LA/1 -ifnum 1/8
Done
<!--NeedCopy-->
```

CLI を使用して既存のリンクアグリゲーションチャネルにインターフェイスをバインドしたり、既存のリンクアグリゲーションチャネルからインターフェイスをバインド解除したりするには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- bind channel <id> <interfaceName>
- unbind channel <id> <interfaceName>

例:

```
bind channel LA/1 1/8
<!--NeedCopy-->
```

CLI を使用してリンクアグリゲーションチャネルを変更するには:

コマンドプロンプトで、
set channel コマンド、チャネル ID、および変更するパラメータを新しい値とともに入力します。

CLI を使用してリンクアグリゲーションチャネルを削除するには:

重要: チャネルを削除すると、そのチャネルにバインドされているネットワークインターフェイスがネットワークループを引き起こし、ネットワークのパフォーマンスが低下します。チャネルを削除する前に、ネットワークインターフェイスを無効にする必要があります。

コマンドプロンプトで入力します。

- rm channel <id>

例:

```
> rm channel LA/1
 Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用してリンクアグリゲーションチャネルを設定するには:

[システム] > [ネットワーク] > [チャネル] に移動するか、新しいチャネルを追加するか、既存のチャネルを編集します。

GUI を使用してリンクアグリゲーションチャネルを削除するには:

> 重要:
>
> チャネルが削除されると、そのチャネルにバインドされたネットワークインターフェイスがネットワークループを引き起こし、ネットワークのパフォーマンスが低下します。チャネルを削除する前に、ネットワークインターフェイスを無効にする必要があります。

[システム] > [ネットワーク] > [チャネル] に移動し、削除するチャネルを選択して [削除] をクリックします。

Link Aggregation Control Protocol を使用したリンクアグリゲーションの構成

Link Aggregation Control Protocol（LACP）を使用すると、ネットワーク・デバイスは LACP データ・ユニット（LACPDU）を交換することによってリンク・アグリゲーション情報を交換できます。そのため、手動で作成したチャネルのメンバーであるネットワークインターフェイスでは LACP を有効にできません。

LACP を使用してリンクアグリゲーションを設定する場合、リンクアグリゲーションチャネルの変更には、リンクアグリゲーションチャネルの作成とは異なるコマンドとパラメータを使用します。チャネルを削除するには、そのチャネルに含まれるすべてのインターフェイスで LACP を無効にする必要があります。

注**:** 高可用性構成では、LACP 構成は伝播も同期もされません。

LACP システムプライオリティの設定

LACP システムプライオリティによって、LACP LA チャネルのどのピアデバイスが LA チャネルを制御できるかが決まります。この番号は、アプライアンスのすべての LACP チャネルにグローバルに適用されます。この値が小さいほど優先度が高くなります。

CLI を使用して LACP システムプライオリティを設定するには:

コマンドプロンプトで次のコマンドを入力して、スタンドアロンアプライアンスの優先順位を設定し、構成を確認します。

- set lacp -sysPriority <positive_integer>
- show lacp

例**:**

```
set lacp -sysPriority 50
<!--NeedCopy-->
```

特定のクラスターノードの優先順位を設定するには、クラスター IP アドレスにログオンし、コマンドプロンプトで次のコマンドを入力します。

- set lacp -sysPriority <positive_integer> -ownerNode <positive_integer>
- show lacp

例:

```
set lacp -sysPriority 50 -ownerNode 2
<!--NeedCopy-->
```

GUI を使用して LACP システムプライオリティを設定するには:

1. [システム] > [ネットワーク] > [インターフェイス] に移動し、[アクション] リストで [LACP の設定] を選択します。
2. システム優先度と所有者ノード (クラスター設定にのみ適用) を指定します。

リンクアグリゲーションチャネルの作成

LACP を使用してリンクアグリゲーションチャネルを作成するには、LACP を有効にし、チャネルに含めたい各インターフェイスで同じ LACP キーを指定する必要があります。たとえば、LACP を有効にし、インターフェイス 1/1 と 1/2 で LACP キーを 3 に設定すると、リンクアグリゲーションチャネル LA/3 が作成され、インターフェイス 1/1 と 1/2 が自動的にそれにバインドされます。

注:

- ネットワークインターフェイスで LACP を有効にする場合は、LACP キーを指定する必要があります。
- デフォルトでは、LACP はすべてのネットワークインターフェースで無効になっています。

CLI を使用して LACP チャネルを作成するには:

コマンドプロンプトで入力します。

- set interface <id> [-lacpMode \] \[-lacpKey\\<positive_integer>] \[-lacpPriority \\<positive_integer>] \[-lacpTimeout \(LONG | SHORT)]
- show interface [\<id>]

GUI を使用して LACP チャネルを作成するには:

[システム] > [ネットワーク] > [インターフェイス] に移動し、ネットワークインターフェイスを開き、パラメータを設定します。

リンクアグリゲーションチャネルの変更

インターフェイスを指定して LACP チャネルを作成したら、チャネルのプロパティを変更できます。

CLI を使用して LACP チャネルを変更するには:

コマンドプロンプトで入力します。

- set channel <id> [-ifnum \ ...] \[-state \(ENABLED | DISABLED)] \[-speed \\] \[-flowControl \\] \[-haMonitor \(ON | OFF)] \[-ifAlias \\] \[-throughput \\<positive_integer>] \[-tagall \(ON | OFF)] \[-bandwidthHigh \\<positive_integer> \[-bandwidthNormal \\<positive_integer>]]
- チャンネルを表示

例**:**

```
> set channel LA/3 -state ENABLED -speed 10000
 Done
<!--NeedCopy-->
```

GUI を使用して LACP チャネルを変更するには:

[システム] > [ネットワーク] > [チャネル] に移動し、既存の LACP チャネルを変更します。

リンクアグリゲーションチャネルの削除

LACP を使用して作成されたリンクアグリゲーションチャネルを削除するには、チャネルに含まれるすべてのインターフェイスで LACP を無効にする必要があります。

CLI を使用して LACP チャネルを削除するには:

コマンドプロンプトで入力します。

- set interface <id> -lacpMode Disable
- show interface [\<id>]

GUI を使用して LACP チャネルを削除するには:

[システム] > [ネットワーク] > [インターフェイス] に移動し、ネットワークインターフェイスを開き、[LACP を有効にする] オプションをオフにします。

LACP チャネルを使用したリンク冗長性

LACP チャネルを使用したリンク冗長性により、NetScaler は LACP チャネルを論理サブチャネルに分割し、1 つのサブチャネルをアクティブにし、他のサブチャネルをスタンバイモードにすることができます。アクティブなサブチャネルがスループットの最小しきい値に達しない場合、スタンバイサブチャネルのいずれかがアクティブになり、処理を引き継ぎます。

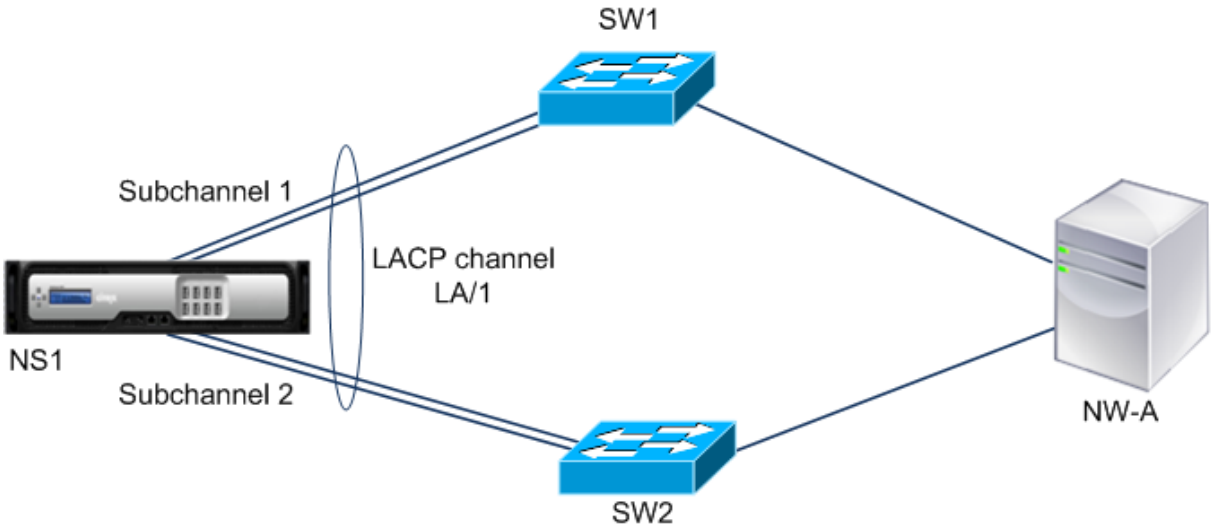
サブチャネルは、LACP チャネルの一部であり、特定のデバイスに接続されているリンクから作成されます。たとえば、NetScaler 上の 4 つのインターフェイスを備えた LACP チャネルで、2 つのインターフェイスがデバイス A に接続され、残りの 2 つがデバイス B に接続されている場合、ADC は 2 つの論理サブチャネルを作成します。1 つのサブチャネルはデバイス A への 2 つのリンクを持ち、もう 1 つはデバイス B への 2 つのリンクを持つサブチャネルです。

LACP チャネルのリンク冗長性を設定するには、LRMinThroughput パラメータを設定します。このパラメータには、アクティブなサブチャネルが満たす最小スループットしきい値（Mbps 単位）を指定します。このパラメーターを設定すると、サブチャンネルが自動的に作成されます。アクティブチャネルでサポートされる最大スループットが lrMinThroughput 値を下回ると、リンクフェールオーバーが発生し、スタンバイサブチャネルがアクティブになります。

LACP チャネルの lrMinThroughput パラメータの設定を解除するか、値をゼロに設定すると、そのチャネルのリンク冗長性はデフォルト設定である無効になります。

例

NetScaler NS1 とスイッチ SW1 および SW2 の間に構成されたリンク冗長性の例を考えてみましょう。



NS1 は、SW1 と SW2 を介してネットワークデバイス NW-A に接続されます。
NS1 では、LACP チャネル LA/1 はインターフェイス 1/1、1/2、1/3、および 1/4 から作成されます。NS1 のインタフェース 1/1 と 1/2 は SW1 に接続され、インタフェース 1/3 と 1/4 は SW2 に接続されます。4 つのリンクはそれぞれ 1000 Mbps の最大スループットをサポートします。
lrMinThroughput パラメータを何らかの値 (たとえば 2000) に設定すると、NS1 は LA/1 から 2 つの論理サブチャネルを作成します。1 つのサブチャネル (たとえばサブチャネル 1) はインターフェイス 1/1 と 1/2 (SW1 に接続) を使用し、もう 1 つのサブチャネル (サブチャネル 2) はインターフェイス 1/3 と 1/4 (SW2 に接続) を使用します。

NS1 は、一方のサブチャネル（たとえばサブチャネル 1）をアクティブにし、もう一方をスタンバイにするアルゴリズムを適用します。NS1 とネットワークデバイス NW-A は、アクティブなサブチャネルを介してのみ相互にアクセスできます。

サブチャネル 1 がアクティブで、サポートされる最大スループットが lrMinThroughput 値を下回っているとします（たとえば、リンクの 1 つに障害が発生し、サポートされる最大スループットが 1000 Mbps に低下した場合など）。サブチャネル 2 がアクティブになり、引き継ぎます。

高可用性セットアップでの **LACP** チャネルを使用したリンク冗長性

高可用性 (HA) 構成で、LACP チャネルにスループット (スループットパラメータ) ベースの HA フェイルオーバーとリンク冗長性 (lrMinThroughput パラメータ) を設定する場合は、スループットパラメータを lrmInThroughput パラメータと同じかそれ以下の値に設定する必要があります。

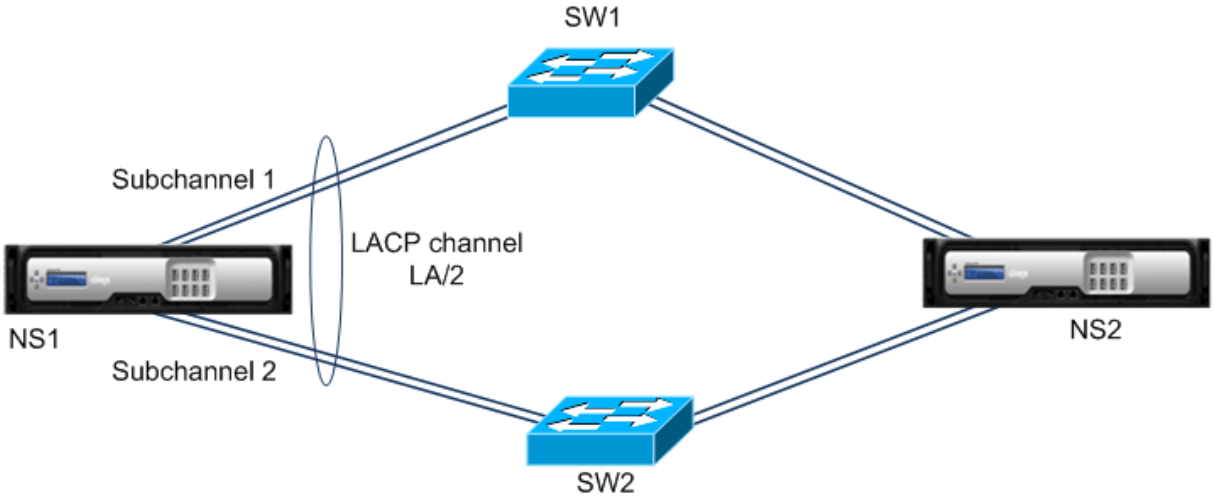
LACP チャネルでサポートされる最大スループットは、アクティブなサブチャネルでサポートされる最大スループットとして計算されます。

スループットパラメータ値が lrminthroughput パラメータ値と同じかそれ以下の場合、次の両方の条件が同時に存在すると HA フェイルオーバーが発生します。

- サブチャネルのサポートされる最大スループットのいずれも、lrMinThroughput パラメーター値を満たしていません。
- LACP チャネルでサポートされる最大スループットがスループットパラメータ値を満たしていません

NetScalers NS1 と NS2 を使用し、スイッチ SW1 と SW2 を備えた HA セットアップの例を考えてみましょう。NS1 は SW1 と SW2 を介して NS2 に接続されています。

NS1 では、LACP チャネル LA/1 はインターフェイス 1/1、1/2、1/3、および 1/4 から作成されます。NS1 のインタフェース 1/1 と 1/2 は SW1 に接続され、インタフェース 1/3 と 1/4 は SW2 に接続されます。4 つのリンクはそれぞれ 1000 Mbps の最大スループットをサポートします。



この例の LACP パラメータ設定は次のとおりです。

パラメーター	値
スループット	2000
lrminthroughput	2000

NS1 は LA/1 から 2 つのサブチャネルを形成します。1 つのサブチャネル（たとえばサブチャネル 1）はインターフェイス 1/1 と 1/2（SW1 に接続）を使用し、もう 1 つのサブチャネル（サブチャネル 2）はインターフェイス 1/3 と 1/4（SW2 に接続）を使用します。2 つのサブチャネルはそれぞれ 2000 Mbps の最大スループットをサポートします。NS1 はアルゴリズムを適用して、一方のサブチャネル（たとえばサブチャネル 1）をアクティブにし、もう一方をスタンバイにします。

サブチャネル 1 がアクティブで、サポートされる最大スループットが lrMinThroughput 値を下回っているとします（たとえば、リンクの 1 つに障害が発生し、サポートされる最大スループットが 1000 Mbps に低下した場合など）。サブチャネル 2 がアクティブになり、引き継ぎます。LACP チャネルでサポートされる最大スループットがスループットパラメータ値を下回らないため、HA フェイルオーバーは発生しません。

LACP チャネルでサポートされる最大スループット = アクティブチャネルでサポートされる最大スループット = サブチャネル 2 でサポートされる最大スループット = 2000 Mbps

サブチャネル 2 でサポートされる最大スループットも lrminthroughput 値を下回ると（たとえば、リンクの 1 つに障害が発生し、サポートされる最大スループットが 1000 Mbps に低下した場合）、LACP チャネルの最大サポートスループットがスループットパラメータ値よりも小さくなるため、HA フェールオーバーが発生します。

LACP チャネルを使用したリンク冗長性の設定

CLI を使用して LACP チャネルのリンク冗長性を設定するには:

コマンドプロンプトで次のコマンドを入力してチャネルを構成し、構成を確認します。

- **set channel** <id> -lrMinThroughput <positive_integer>
- チャンネルを表示

例**:**

```
1 > set channel la/1 - lrMinThroughput 2000
2 Done
3 > set channel la/2 - throughput 2000 - lrMinThroughput 2000
4 Done
5 <!--NeedCopy-->
```

GUI を使用して LACP チャネルのリンク冗長性を設定するには

1. [システム] > [ネットワーク] > [チャネル] に移動します。
2. 詳細ペインで、リンク冗長性を設定する LACP チャネルを選択し、[編集] をクリックします。
3. LACP チャネルの設定ダイアログボックスで、lrMinThroughput パラメータを設定します。
4. [閉じる] をクリックします。

冗長インターフェイスセット

August 15, 2023

> 注:
>
> リンク冗長構成は、NetScaler SDX アプライアンスでホストされている NetScaler VPX インスタンスではサポートされていません。

冗長インターフェイスセットは、インターフェイスの 1 つがアクティブで、残りがスタンバイになっているインターフェイスのセットです。アクティブインターフェイスに障害が発生すると、スタンバイインターフェイスのいずれかが引き継ぎ、アクティブになります。

冗長インターフェイスセットを使用する主な利点は次のとおりです。

- 冗長インターフェイスセットは、NetScaler アプライアンスとピアデバイス間のバックアップリンクを提供することにより、それらの間の接続の信頼性を確保します。
- LACP を使用するリンク冗長性とは異なり、ピアデバイスで冗長インターフェイスセットを設定する必要はありません。ピアデバイスから見ると、冗長インターフェイスセットは個別のインターフェイスとして認識され、セットやコレクションとして認識されません。
- 高可用性構成（HA）では、冗長インターフェイスセットにより HA フェイルオーバーの数を最小限に抑えることができます。

> 注
>
> 冗長インターフェイスセットは、10.5 リリースで初めて導入された当初は「NIC バンドル」と呼ばれていました。

冗長インターフェイスセットの仕組み

冗長インターフェイスセットの場合、NetScaler アプライアンスは内部アルゴリズムに基づいて MAC アドレスを取得し、それを冗長インターフェイスセットに割り当てます。この MAC アドレスはすべてのメンバーインターフェイスで共有され、一度に使用されるのはアクティブなインターフェイスだけです。アクティブインターフェイスは GARP メッセージをブロードキャストします。GARP メッセージには、インターフェイス自体の物理 MAC アドレスではなく、冗長インターフェイスセットに割り当てられた MAC アドレスが含まれます。現在のアクティブインターフェイスに障害が発生して別のインターフェイスに引き継がれると、新しいアクティブインターフェイスが GARP メッセージを送信します。ピアデバイスは、新しいアクティブインターフェイス情報で転送テーブルを更新します。スタンバイインターフェイスは GARP メッセージを送信しません。スタンバイインターフェイスはパケットを送信せず、受信したパケットもドロップします。

冗長インターフェイスセットでは、メンバーインターフェイスをアクティブにするかどうかは、次のいずれかの要因に基づいて決定されます。

- 冗長インターフェイスプライオリティ。これはインタフェースのパラメータであり、アクティブメンバー選択用の冗長インタフェースセットにおけるインタフェースの優先順位を定義します。このパラメータは正の整数を指定します。値を小さくすると、アクティブなメンバー選択の優先度が高くなります。優先順位が最も高い（最も低い値）メンバーインターフェイスが、冗長インターフェイスセットのアクティブインターフェイスとして選択されます。
- メンバーインターフェースのバインディング順序。すべてのメンバーインターフェイスの冗長インターフェイスプライオリティが同じ場合、冗長インターフェイスセットに最初にバインドされたメンバーインターフェイスが冗長インターフェイスセットのアクティブインターフェイスとして選択されます。

冗長インターフェイスセットでは、次のいずれかのイベントでアクティブインターフェイス選択がトリガーされます。

- 現在アクティブなインターフェイスに障害が発生した場合、またはそれを無効にしたとき。
- スタンバイインターフェイスのプライオリティを現在のアクティブインターフェイスのプライオリティよりも低い値に設定した場合。スタンバイインターフェイスがアクティブインターフェイスとして引き継ぎます。
- 現在のアクティブなインターフェイスよりも優先順位が低いインターフェイスをバインドする場合。新しくバインドされたインターフェイスがアクティブインターフェイスとして引き継がれます。

冗長インターフェイスセットの設定に関する考慮事項

冗長インターフェイスセットを設定する前に、次の点を考慮してください。

- スタンドアロンアプライアンスまたは高可用性設定のアプライアンスでは、リンク冗長セットは LR/X 表記で指定されます。X の範囲は 1 ～4 です。たとえば、LR/1 と入力します。
- 高可用性構成では、冗長インターフェイスセット構成はセカンダリノードに伝播または同期されません。
- NetScaler アプライアンスでは、最大 4 つの冗長インターフェイスセットを構成できます。
- 冗長インターフェイスセットには、最大 16 のインターフェイスをバインドできます。
- 冗長インターフェイスセットのメンバーインターフェイスは、別の冗長インターフェイスセットにバインドできません。
- 冗長インターフェイスセットのメンバーインターフェイスをリンクアグリゲート（LA）チャネルにバインドすることはできません。
- LA チャネルを冗長インターフェイスセットにバインドすることはできません。
- 冗長インターフェイスセットを LA チャネルにバインドすることはできません。
- クラスタ設定の場合:
  - 冗長インターフェイスセットをクラスタリンクアグリゲーションにバインドすることはできません。
  - リンク冗長セットは N/LR/X 表記 (たとえば、1/LR/3) で指定されます。ここで、
    N は冗長インターフェイスセットを作成するクラスターノードの ID です。
    X はクラスタノード上のリンク冗長セット識別子です。X の範囲は 1 ～4 です。
  - クラスタリンクアグリゲーションを冗長インターフェイスセットにバインドすることはできません。
  - 冗長インターフェイスセットには、冗長インターフェイスセットが属するノードのインターフェイスのみを含めることができます。

- スタンドアロンアプライアンスの既存の elink 冗長セット構成は、アプライアンスをクラスタ設定に追加すると、自動的にクラスタ表記（N/LR/X）に変更されます。

構成の手順

NetScaler アプライアンスでの冗長インターフェイスセットの構成は、次のタスクで構成されます。

- 冗長インターフェイスセットを作成します。channel コマンド操作を使用して、冗長インターフェイスセットを作成します。

  スタンドアロンアプライアンスまたは高可用性設定のアプライアンスでは、リンク冗長セットは LR/X 表記で指定されます。X の範囲は 1 ～4 です。たとえば、LR/1 と入力します。

  クラスタ設定では、リンク冗長セットは N/LR/X（たとえば、1/LR/3）で指定されます。ここで、N は冗長インターフェイスセットを作成するクラスタノードの ID、X はクラスタノード上のリンク冗長セット識別子です。X の範囲は 1 ～4 です。

- インターフェイスを冗長インターフェイスセットにバインドします。必要なインターフェイスを冗長インターフェイスセットに関連付けます。1 つのインターフェイスを複数の冗長インターフェイスセットの一部にすることはできません。
- （オプション）メンバーインターフェイスに冗長インターフェイスプライオリティを設定します。インターフェイスコマンド操作を使用して、冗長インターフェイスセットの目的のメンバーインターフェイスに冗長インターフェイスプライオリティを設定します。

CLI を使用して冗長インターフェイスセットを作成するには：

コマンドプロンプトで、次の操作を行います。

- add channel <ID>
- show channel <ID>

CLI を使用してインターフェイスを冗長インターフェイスセットにバインドするには：

コマンドプロンプトで、次の操作を行います。

- bind channel <ID> <ifnum>
- show channel <ID>

CLI を使用してインターフェイスの冗長インターフェイスプライオリティを設定するには：

コマンドプロンプトで、次の操作を行います。

- set interface <ID> -lrsetpriority <positive_integer>
- show interface <ID>

設定例 **1:**

次の例では、冗長インターフェイスセット LR/1 が作成され、インターフェイス 1/1、1/2、1/3、1/4 が LR/1 にバインドされます。これらすべてのメンバーインターフェイスの冗長インターフェイスプライオリティは、デフォルト値の 1024 に設定されています。show channel コマンドの出力では、インターフェイス 1/1 が冗長インターフェイスセット lr/1 の現在のアクティブインターフェイスであることがわかります。

```
> add channel lr/1
 Done
> bind channel lr/1 1/1 1/2 1/3 1/4
 Done
> show channel
1)    Interface LR/1 (Link Redundant) #23
        flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
        MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
        Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
                  throughput 0
        Actual: throughput 1000
        LLDP Mode: NONE,
        RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
        TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
                1/1: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Active Member
                1/2: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/3: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/4: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
Done
<!--NeedCopy-->
```

設定例 **2:**

次の例では、メンバーインターフェイス 1/4 の冗長インターフェイスプライオリティが 100 に設定されています。これは、LR/1 の他のすべてのメンバーインターフェイスに設定されている冗長インターフェイスプライオリティよりも低くなっています。

show channel コマンドの出力では、インターフェイス 1/4 が冗長インターフェイスセット LR/1 の現在のアクティブインターフェイスであることがわかります。

```
> set interface 1/4 -lrsetPriority 100
Done
> show channel
1)      Interface LR/1 (Link Redundant) #23
        flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
        MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
        Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
                  throughput 0
        Actual: throughput 1000
```



```
10          LLDP Mode: NONE,
11          RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
12          TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
13          NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
14          Bandwidth thresholds are not set.
15                  1/1: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                       Inactive Member
16                  1/2: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                       Inactive Member
17                  1/3: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                       Inactive Member
18                  1/4: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                       Active Member
19  Done
20  <!--NeedCopy-->
```

設定例 **3:**

4 つのノード N1、N2、N3、N4 のクラスタ構成を考えてみましょう。この例では、冗長インターフェイスセット 1/LR/3 がノード N1 上に作成され、インターフェイス 1/1/1、1/1/2、および 1/1/3 がそれにバインドされます。これらすべてのメンバーインターフェイスの冗長インターフェイスプライオリティは、デフォルト値の 1024 に設定されています。show channel コマンドの出力から、インターフェイス 1/1/1 が冗長インターフェイスセット 1/LR/3 の現在のアクティブインターフェイスであることがわかります。

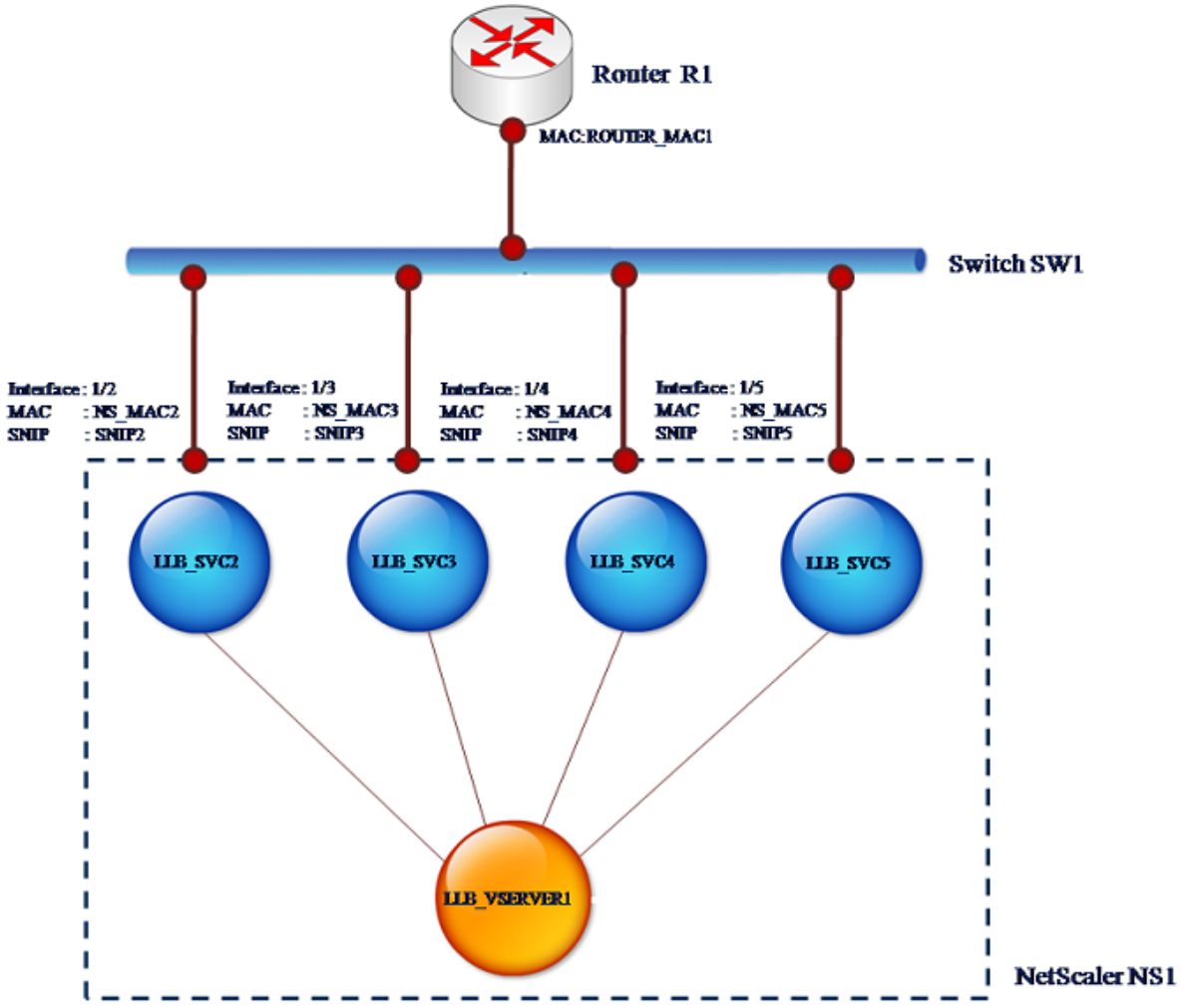
```
1       > add channel 1/LR/3
2
3       Done
4       > bind channel 1/LR/3 1/1/1 1/1/2 1/1/3
5
6       Done
7       > show channel
8       1)    Interface 1/LR/3 (Link Redundant) #14
9               flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON,
                   802.1q>
10              MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0
                   h00m00s
11              Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
12              throughput 0
13              Actual: throughput 1000
14              LLDP Mode: NONE,
15              RX: Pkts(66) Bytes(4406) Errs(0) Drops(82) Stalls(0)
16              TX: Pkts(55) Bytes(2626) Errs(0) Drops(145) Stalls(0)
17              NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted
                   (0)
18              Bandwidth thresholds are not set.
19
20              1/1/1: UTP-1000-FULL-OFF UP  0h14m06s LR Active Member
21              1/1/2: UTP-1000-FULL-OFF UP  0h14m06s LR Inactive Member
22              1/1/3: UTP-1000-FULL-OFF UP  0h14m06s LR Inactive Member
23
24      Done
25  <!--NeedCopy-->
```


SNIP アドレスをインターフェイスにバインド

August 15, 2023

レイヤー 3 VLAN を使用せずに、NetScaler が所有する SNIP アドレスをインターフェイスにバインドできるようになりました。SNIP アドレスに関連するパケットは、バインドされたインターフェイスのみを経由します。

この機能は、アップストリームスイッチがリンクアグリゲーションチャネルをサポートしていない場合に、次の図に示すように、サーバーから発信されたトラフィックを NetScaler アプライアンスにアップストリームスイッチへの 4 つのリンクにわたって負荷分散させたい場合に役立ちます。



次の表は、シナリオの設定例を示しています。

エンティティ	名前	値
NS1 の SNIP アドレス	SNIP2 (参照用のみ)	10.10.10.2

エンティティ	名前	値
		SNIP3 (参照用のみ)
		SNIP4 (参照用のみ)
		SNIP5 (参照のみを目的としています)
NS1 上の LLB 仮想サーバ	LLB_VSERVER1	-
NS1 の透明モニター	TRANS_MON	-
NS1 の LLB サービス	LLB_SVC2	10.10.10.240
		LLB_SVC3
		LLB_SVC4
		LLB_SVC5
NS1 のインターフェイス 1/2 の MAC アドレス	NS_MAC_2 (参照のみを目的としています)	00:e0:ed:0f:bc:e0
NS1 のインタフェース 1/3 の MAC アドレス	NS_MAC_3 (参照のみを目的としています)	00:e0: ed:0f: bc: df
NS1 のインタフェース 1/4 の MAC アドレス	NS_MAC_4 (参照のみを目的としています)	00:e0:ed:0f:bc:de
NS1 のインターフェイス 1/5 の MAC アドレス	NS_MAC_5 (参照のみを目的としています)	00:e0:ed:1c:89:53
ルータ R1 の IP アドレス	ルーター _IP (参照用のみ)	10.10.10.1
R1 のインターフェイスの MAC アドレス	ルーター _MAC1 (参照用のみ)	00:21:a1:2d:db:cc

設定例を設定するには:

1. 異なるサブネット範囲に 4 つの異なる SNIP を追加します。これは、ARP が 4 つの異なるリンクで解決されるためです。SNIP アドレスの作成の詳細については、「サブネット IP アドレス (SNIP) の設定」を参照してください。

   CLI の例:

```
> add ns ip 10.10.10.2 255.255.255.0 -type SNIP
 Done
> add ns ip 10.10.10.3 255.255.255.128  - type SNIP
Done
> add ns ip 10.10.10.4 255.255.255.192  - type SNIP
Done
> add ns ip 10.10.10.5 255.255.255.224  - type SNIP
Done
```



```
<!--NeedCopy-->
```

2. 追加した SNIP サブネットに 4 つの異なるダミーサービスを追加します。これは、4 つの設定済み SNIP の 1 つとして、送信元 IP を使用してトラフィックが送信されるようにするためです。サービスの作成の詳細については、「基本的な負荷分散の設定」を参照してください。

   CLI の例:

```
> add service LLB_SVC2 10.10.10.240 any *
 Done
> add service LLB_SVC3 10.10.10.120 any *
 Done
> add service LLB_SVC4 10.10.10.60 any *
 Done
> add service LLB_SVC5 10.10.10.30 any *
 Done
<!--NeedCopy-->
```

3. ゲートウェイを監視するためのトランスペアレント PING モニターを追加します。モニターを設定済みの各ダミーサービスにバインドします。これは、サービスの状態を UP にすることです。トランスペアレントモニタの作成の詳細については、「ロードバランシングセットアップでのモニタの設定」を参照してください。

   CLI の例:

```
> add monitor TRANS_MON ping -destIP 10.10.10.1 -transparent YES
 Done
> bind monitor TRANS_MON LLB_SVC2
 Done
> bind monitor TRANS_MON LLB_SVC3
 Done
> bind monitor TRANS_MON LLB_SVC4
 Done
> bind monitor TRANS_MON LLB_SVC5
 Done
<!--NeedCopy-->
```

4. リンクロードバランシング (LLB) 仮想サーバーを追加し、ダミーサービスをバインドします。LLB 仮想サーバーの作成の詳細については、基本的な LLB セットアップの構成を参照してください。

   CLI の例:

```
> add lb vserver LLB_VSERVER1 any
 Done
> set lb vserver LLB_VSERVER1 -lbmethod ROUNDROBIN
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
```



```
12   Done
13  <!--NeedCopy-->
```

5. LLB 仮想サーバをデフォルトの LLB ルートとして追加します。LLB ルートの作成の詳細については、基本的な LLB セットアップの設定を参照してください。

   CLI の例:

```
1  > add lb route 0.0.0.0 0.0.0.0  LLB_VSERVER1
2   Done
3  <!--NeedCopy-->
```

6. ゲートウェイの MAC アドレスを使用して、ダミーサービスごとに ARP エントリを追加します。このようにして、これらのダミーサービスを介して Gateway に到達できます。ARP エントリの追加の詳細については、スタティック ARP の設定を参照してください。

   CLI の例:

```
1  > add arp -ipaddress 10.10.10.240 -mac 00:21:a1:2d:db:cc -ifnum
      1/2
2   Done
3  > add arp -ipaddress 10.10.10.120 -mac 00:21:a1:2d:db:cc -ifnum
      1/3
4   Done
5  > add arp -ipaddress 10.10.10.60 -mac 00:21:a1:2d:db:cc -ifnum 1/4
6   Done
7  > add arp -ipaddress 10.10.10.30 -mac 00:21:a1:2d:db:cc -ifnum 1/5
8   Done
9  <!--NeedCopy-->
```

7. これらの SNIP ごとに ARP エントリを追加して、特定のインターフェイスを SNIP にバインドします。これは、応答トラフィックが、要求が送信されたのと同じインターフェイスに到達するようにするためです。ARP エントリの追加の詳細については、スタティック ARP の設定を参照してください。

   CLI の例:

```
1  > add arp -ipAddress 10.10.10.2 -mac 00:e0:ed:0f:bc:e0 -ifnum 1/2
2   Done
3  > add arp -ipAddress 10.10.10.3 -mac 00:e0:ed:0f:bc:df -ifnum 1/3
4   Done
5  > add arp -ipAddress 10.10.10.4 -mac 00:e0:ed:0f:bc:de -ifnum 1/4
6   Done
7  > add arp -ipAddress 10.10.10.5 -mac 00:e0:ed:1c:89:53 -ifnum 1/5
8   Done
9  <!--NeedCopy-->
```


ブリッジテーブルの監視とエージングタイムの変更

August 15, 2023

NetScaler アプライアンスは、宛先 MAC アドレスと VLAN ID のブリッジテーブル検索に基づいてフレームをブリッジします。ただし、アプライアンスはレイヤ 2 モードが有効になっている場合にのみ転送を実行します。

ブリッジテーブルは動的に生成されますが、表示したり、ブリッジテーブルのエージングタイムを変更したり、ブリッジ統計情報を表示したりできます。ブリッジテーブル内のすべての MAC エントリは、エージングタイムとともに更新されます。

CLI を使用してブリッジテーブルエントリのエージングタイムを設定するには:

コマンドプロンプトで入力します。

- **set l2param** -**bridgeagetimeout** <positive_integer>
- **show l2param**

例**:**

```
1 > set l2param -bridgeagetimeout 90
2  Done
3 <!--NeedCopy-->
```

CLI を使用してブリッジテーブルの統計情報を表示するには:

コマンドプロンプトで入力します。

- **stat bridge**

GUI を使用してブリッジテーブルエントリのエージングタイムを設定するには:

[システム] > [ネットワーク] に移動します。[ネットワーク] ページの [設定] セクションで、[レイヤ **2** パラメータの設定] をクリックし、[ブリッジテーブルエントリのタイムアウト値 **(**秒**)**] パラメータを設定します。

GUI を使用してブリッジテーブルの統計情報を表示するには:

[システム] > [ネットワーク] > [ブリッジテーブル] に移動し、MAC アドレスを選択して [統計] をクリックします。

VRRP を使用したアクティブ**/**アクティブモードでの **NetScaler ADC** アプライアンス

August 15, 2023

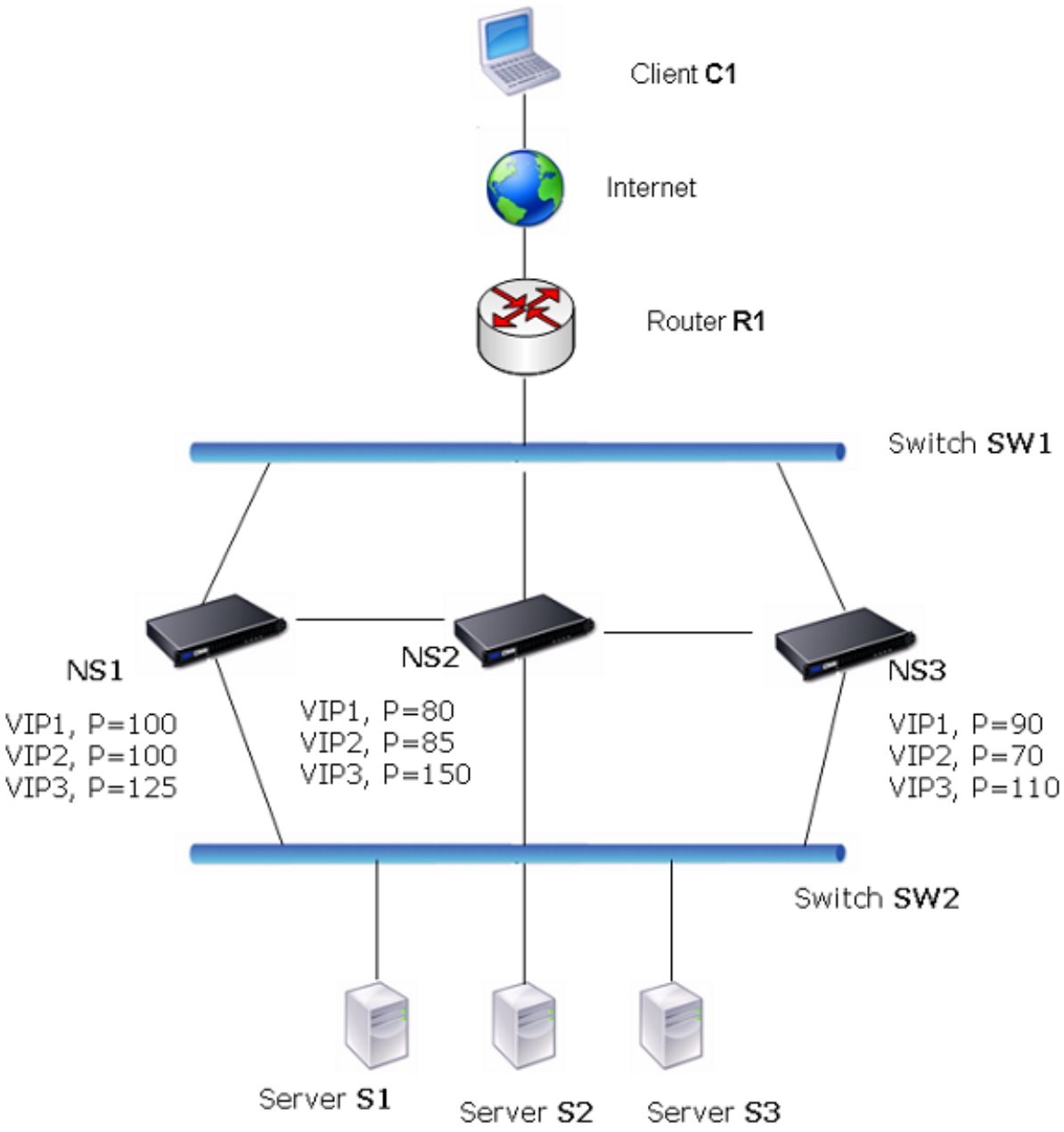
アクティブ-アクティブ展開では、ダウンタイムを防ぐだけでなく、展開環境内のすべての NetScaler アプライアンスを効率的に使用できます。アクティブ-アクティブ展開モードでは、同じ VIP が構成内のすべての NetScaler アプ

ライアンスに異なる優先順位で設定されるため、特定の VIP は一度に 1 つのアプライアンスでのみアクティブになります。

アクティブな VIP はマスター VIP と呼ばれ、他の NetScaler アプライアンスの対応する VIP はバックアップ VIP と呼ばれます。マスター VIP に障害が発生すると、優先順位が最も高いバックアップ VIP が引き継いでマスター VIP になります。アクティブ/アクティブ展開のすべての NetScaler アプライアンスは、仮想ルーター冗長プロトコル（VRRP）プロトコルを使用して、VIP とそれに対応する優先順位を定期的にアドバタイズします。

アクティブ/アクティブモードの NetScaler アプライアンスは、NetScaler がアイドル状態にならないように構成できます。この構成では、NetScaler ごとに異なる VIP セットがアクティブになります。たとえば、次の図では、VIP1、VIP2、VIP3、VIP4 がアプライアンス NS1、NS2、NS3 で設定されています。優先順位の関係で、VIP1 と VIP 2 は NS1 でアクティブになり、VIP3 は NS2 でアクティブになり、VIP 4 は NS3 でアクティブになります。たとえば、NS1 に障害が発生すると、NS3 の VIP1 と NS2 の VIP2 がアクティブになります。

図 1：アクティブ-アクティブ構成



上の図の NetScaler アプライアンスは、次のようにトラフィックを処理します。

1. クライアント C1 は VIP1 に要求を送信します。リクエストは R1 に達します。
2. R1 には VIP1 の ARP エントリがないため、VIP1 の ARP 要求をブロードキャストします。
3. VIP1 は NS1 でアクティブなので、NS1 は VIP1 に関連付けられた仮想 MAC（仮想 MAC1 など）として送信元 MAC アドレスを、送信元 IP アドレスとして VIP1 を返信します。
4. SW1 は ARP 応答から VIP1 のポートを学習し、ブリッジテーブルを更新します。
5. R1 は仮想 MAC1 と VIP1 を使用して ARP エントリを更新します。

6. R1 は NS1 上の VIP1 にパケットを転送します。
7. NS1 の負荷分散アルゴリズムがサーバー S2 を選択し、NS1 は SNIP アドレスの 1 つと S2 間の接続を開きます。
8. S2 は NetScaler 上の SNIP に応答します。
9. NS1 は S2 の応答をクライアントに送信します。応答では、NS1 は物理インタフェースの MAC アドレスを送信元の MAC アドレスとして挿入し、VIP1 を送信元の IP アドレスとして挿入します。
10. NS1 に障害が発生した場合、NetScaler アプライアンスは VRRP プロトコルを使用して優先順位が最も高い VIP1 を選択します。この場合、NS3 の VIP1 がアクティブになり、次の 2 つの手順でアクティブ-アクティブ構成が更新されます。
11. NS3 は VIP1 の GARP メッセージをブロードキャストします。メッセージでは、仮想 MAC1 が送信元 MAC アドレスで、VIP1 が送信元 IP アドレスです。
12. SW1 は、GARP ブロードキャストから仮想 MAC1 の新しいポートを学習し、ブリッジテーブルを更新して、VIP1 に対するその後のクライアント要求を NS3 に送信します。R1 は ARP テーブルを更新します。

VIP の優先度はヘルストラッキングで変更できます。ヘルストラッキングを有効にする場合は、優先度が低い VIP が別の VIP にプリエンプションされるように、プリエンプションも有効にする必要があります。

状況によっては、トラフィックがバックアップ VIP に到達することがあります。このようなトラフィックがドロップされないようにするには、アクティブ/アクティブ構成を作成するときに、ノード単位で共有を有効化できます。または、マスターへのグローバル送信オプションを有効にすることもできます。共有が有効になっているノードでは、マスターへの送信よりも共有が優先されます。

ヘルストラッキング

通常、ベースプライオリティ（BP 範囲 1 ～255）によってどの VIP がマスター VIP になるかが決まりますが、実効プライオリティ（EP）も決定に影響します。

たとえば、NS1 の VIP の優先度が 101 で、NS2 の同じ VIP の優先順位が 99 の場合、NS1 の VIP はアクティブです。ただし、2 台の仮想サーバが NS1 で VIP を使用していて、そのうちの 1 つがダウンした場合、ヘルストラッキングによって NS1 の VIP の EP が減少する可能性があります。その後、VRRP は NS2 の VIP をアクティブな VIP にします。

EP を変更するためのヘルストラッキングオプションは次のとおりです。

- なし。追跡なし。EP = BP
- **ALL**。すべての仮想サーバーが稼働している場合、EP = BP です。それ以外の場合は、EP = 0 になります。
- 一。少なくとも 1 台の仮想サーバーが稼働している場合、EP = BP です。それ以外の場合は、EP = 0 になります。
- **PROGRESSIVE.** すべての仮想サーバーが稼働している場合、EP = BP です。すべての仮想サーバーがダウンしている場合、EP = 0 になります。それ以外の場合、EP = BP (1-K/N)。ここで、N は VIP に関連付けられている仮想サーバの総数、k はダウンしている仮想サーバの数です。

注**:** NONE 以外の値を指定する場合は、プリエンプションを有効にする必要があります。これにより、マスター VIP の優先順位がダウングレードされた場合に、優先順位が最も高いバックアップ VIP がアクティブになります。

プリエンプション

プライオリティが高い別の VIP によるアクティブ VIP のプリエンプションはデフォルトで有効になっており、通常は有効になっているはずです。ただし、場合によっては無効にしたい場合もあります。プリエンプションは各 VIP のノード単位の設定です。

プリエンプションは次のような状況で発生する可能性があります。

- アクティブな VIP がダウンし、優先度の低い VIP が代わりになります。優先順位の高い VIP がオンラインに戻ると、現在アクティブな VIP をプリエンプションします。
- ヘルストラッキングにより、バックアップ VIP の優先順位がアクティブ VIP の優先度よりも高くなります。次に、バックアップ VIP がアクティブ VIP をプリエンプションします。

共有

トラフィックがバックアップ VIP に到達した場合、バックアップ VIP で共有オプションが有効になっていない限り、トラフィックはドロップされます。この動作は各 VIP のノードごとの設定で、デフォルトでは無効になっています。

図では、アクティブ/アクティブ構成 NS1 の VIP1 がアクティブで、NS2 と NS3 の VIP1 VIP がバックアップです。特定の状況下では、トラフィックが NS2 の VIP1 に到達することがあります。NS2 で共有が有効になっている場合、このトラフィックはドロップされずに処理されます。

アクティブ/アクティブモードの構成

August 15, 2023

アクティブ/アクティブモードで展開する各 NetScaler アプライアンスに、仮想 MAC を追加し、その仮想 MAC を VIP にバインドする必要があります。特定の VIP の仮想 MAC は、各アプライアンスで同じである必要があります。たとえば、アプライアンスで VIP 10.102.29.5 を作成する場合、仮想ルーター ID（VRID）を各 NetScaler で作成し、各 NetScaler の VIP 10.102.29.5 にバインドする必要があります。仮想 MAC を VIP にバインドすると、アプライアンスはその VIP にバインドされている各 VLAN に VRRP アドバタイズメントを送信します。仮想 MAC は、同じ NetScaler 上に構成されているさまざまな VIP と共有できます。

IPv4 アクティブアクティブモードの設定

アクティブ/アクティブ構成に含める各 NetScaler アプライアンスで次のタスクを実行します。

- 仮想 **MAC** アドレスを追加します。VRID を追加して仮想 MAC アドレスを追加します。また、優先順位を指定したり、この VRID アドレスでプリエンプションと共有を有効または無効にしたりすることもできます。
- **VIP** アドレスを追加し、仮想 **MAC** の **VRID** を関連付けます。VIP アドレスを追加し、新しく作成した VRID に VRID パラメータを設定します。VRID の属性（プライオリティやプリエンプションなど）は、この VIP アドレスにバインドされます。
  注：他のすべての NetScaler アプライアンスにも同じ VIP アドレスを追加する必要があります。

CLI を使用して仮想 MAC アドレスを追加するには

コマンドプロンプトで入力します。

- **add vrID** <id> [-**priority** \<positive_integer>] [-**preemption** (**ENABLED**|**DISABLED**)][-**sharing** (**ENABLED**|**DISABLED**)] [-**tracking** \]
- **show vrid**

CLI を使用して VIP アドレスを追加するには

コマンドプロンプトで入力します。

- IP <IPv4Address> タイプ **VIP** グリッドの追加 <value>
- **show ns ip**

GUI を使用して仮想 MAC を設定するには：

1. [システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC**] タブで新しい仮想 MAC を追加するか、既存の仮想 MAC を編集します。
2. 次のパラメーターを設定します。
   - 仮想ルーター ID
   - 優先度
   - トラッキング
   - プリエンプション
   - 共有

GUI を使用して VIP アドレスを設定し、VRID をそれに関連付けるには：

1. **[**** システム] > [ネットワーク] > [IP] に移動し、[**IPv4]** タブで **VIP** タイプの IP アドレスを追加します。**
2. IP アドレスを追加するときに、「仮想ルーター ID」ドロップダウンボックスから仮想ルーター **ID** を選択します。

設定例**:**

次の構成例は、NetScaler アプライアンス NS1 と NS2 を IPv4 アクティブ-アクティブモードで展開するためのものです。VIP アドレス 203.0.113.10 は NS1 と NS2 の両方で設定され、アプライアンスごとに優先順位値が異なります。各アプライアンスでは、この VIP アドレスは仮想 MAC アドレスにバインドされます。NS2 の優先度（200）が NS1（100）よりも高いため、203.0.113.10 は NS2 のマスターです。


```
    Settings on NS1

    > add vrid 10 -Priority 100 -Preemption Enabled -sharing Enabled

    Done

    > add ns ip 203.0.113.10 -type VIP -vrid 10

    Done

    Settings on NS2

    > add vrid 10 -Priority 200 -Preemption Enabled -sharing Enabled

    Done

    > add ns ip 203.0.113.10 -type VIP -vrid 10

    Done
<!--NeedCopy-->
```

IPv6 アクティブ-アクティブモードの設定

アクティブ/アクティブ構成に含める各 NetScaler アプライアンスで次のタスクを実行します。

- 仮想 **MAC6** アドレスを追加します。VRID6 を追加して仮想 MAC6 アドレスを追加します。優先順位を指定し、この VRID6 アドレスでのプリエンプションと共有を有効または無効にすることもできます。
- **VIP6** アドレスを追加します。VIP6 アドレスを追加します。VRID6 パラメータを新しく作成した仮想 MAC6 の VRID6 に設定します。仮想 MAC6 の属性（プライオリティやプリエンプションなど）は、この VIP6 アドレスにバインドされます。
  注：他のすべての NetScaler アプライアンスにも同じ VIP6 アドレスを追加する必要があります。

CLI を使用して仮想 MAC6 アドレスを追加するには：

コマンドプロンプトで入力します。

- **add vrID6** <id> [-**priority** \<positive_integer>] [-**preemption** (**ENABLED** | **DISABLED**)] [-**sharing** (**ENABLED** | ** 無効 **)]
- **show vrID6**

CLI を使用して VIP6 アドレスを追加するには：

コマンドプロンプトで入力します。

- **IP6** の追加-VIP タイプ **** <IPv6Address**-グリッド ** <value>
- **show ns ip6**

GUI を使用して仮想 MAC6 を設定するには：

1. **[** システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC6**] タブで、新しい仮想 MAC6 を追加するか、既存の VMAC6 を編集します。**
2. 次のパラメーターを設定します。

    - 仮想ルーター ID
    - 優先度
    - プリエンプション
    - 共有

GUI を使用して VIP6 アドレスを設定し、それに VRID を関連付けるには:

1. [システム] > [ネットワーク] > [**IP]** に移動し、**[IPv6]** タブに **VIP** タイプの IPv6 アドレスを追加します。
2. VIP6 アドレスを追加するときに、仮想ルータ ID ドロップダウンボックスから VRID6 を選択します。

設定例**:**

次の構成例は、NetScaler アプライアンス NS1 と NS2 を IPv6 アクティブ-アクティブモードで展開するためのものです。VIP6 アドレス 2001: db8:: 5001 は NS1 と NS2 の両方で設定され、アプライアンスごとに異なる優先順位値が設定されます。各アプライアンスでは、この VIP6 アドレスは仮想 MAC6 アドレスにバインドされます。NS2 の優先度 (200) が NS1 (100) よりも高いため、NS2 のマスターは 2001: db8:: 5001 は NS2 のマスターです。

```
    Settings on NS1
    > add vrid6 10  - Priority 100  - Preemption Enable  - sharing Enable

    Done
    > add ns ip6 2001:db8::5001  - type VIP  - vrid6 10

    Done
    Settings on NS2
    > add vrid6 10  - Priority 200  - Preemption Enable  - sharing Enable

    Done
    > add ns ip6 2001:db8::5001  - type VIP  - vrid6 10

    Done
<!--NeedCopy-->
```

マスターへの送信の構成

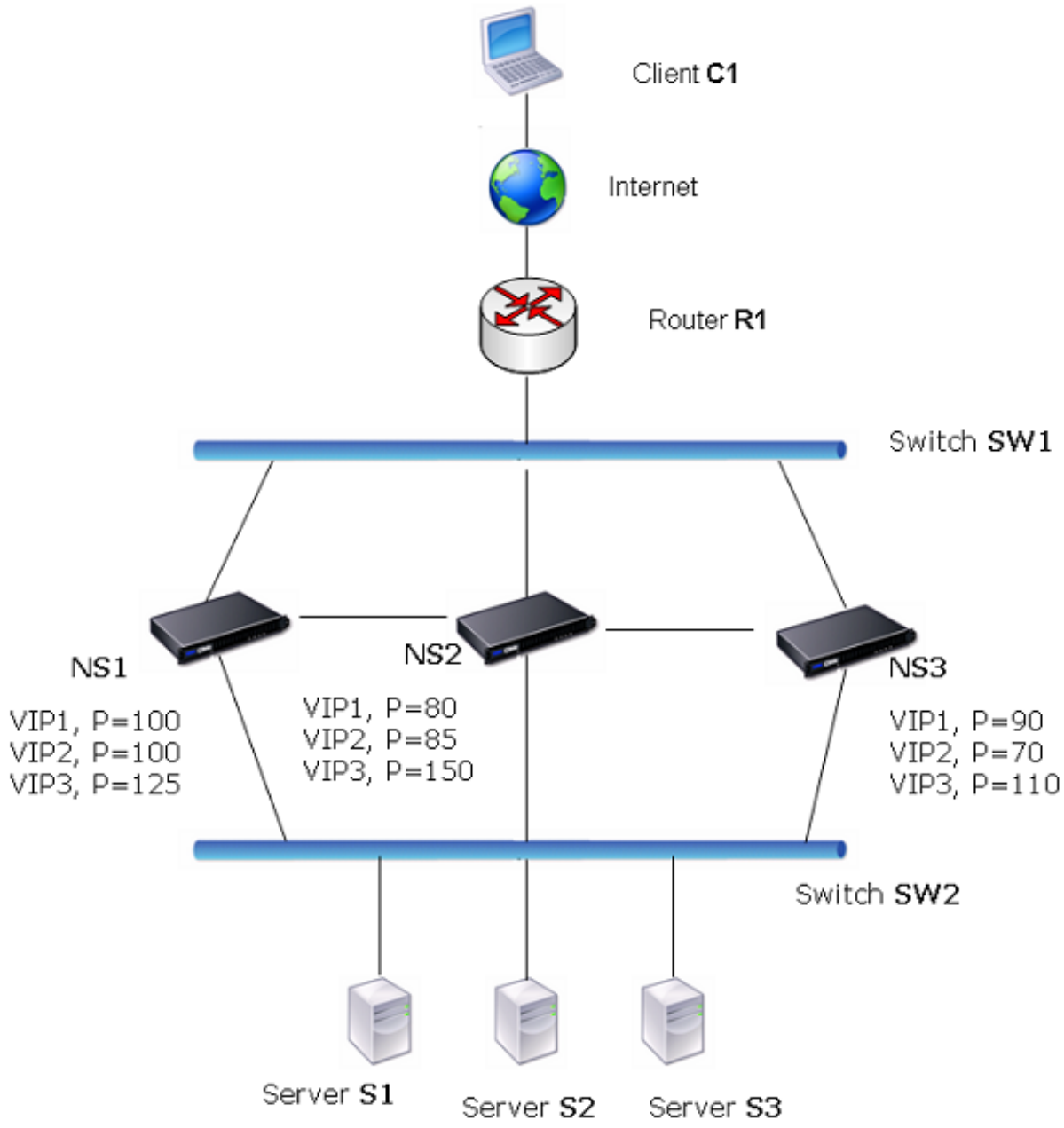
August 15, 2023

通常、VIP 宛てのトラフィックは VIP がアクティブな NetScaler アプライアンスに到達します。これは、VIP とそのアプライアンス上の仮想 MAC を含む ARP 要求が上流のルーターに到達したためです。ただし、VIP サブネットの上流ルーターに設定された静的ルートや、このルートをブロックするトポロジなど、場合によっては、VIP がバックアップ状態の NetScaler アプライアンスにトラフィックが到達することがあります。このアプライアンスで VIP がア

クティブなアプライアンスにデータパケットを転送するには、Send to master オプションを有効にする必要があります。この動作はノードごとの設定で、デフォルトでは無効になっています。

たとえば、次の図では、VIP1 は NS1、NS2、NS3 で設定され、NS1 ではアクティブになっています。特定の状況下では、VIP1（NS1 でアクティブ）のトラフィックが NS3 の VIP1 に到達することがあります。NS3 でマスターへの送信オプションが有効になっている場合、NS3 は NS1 のルートエントリを使用して NS2 経由で NS1 にトラフィックを転送します。

図 1:「マスターに送信」オプションが有効なアクティブ-アクティブ構成



CLI を使用してマスターへの送信を有効にするには:

コマンドプロンプトで入力します。

set vrIDParam -sendToMaster (ENABLED DISABLED)

例:

```
> set vrIDParam -sendToMaster ENABLED
 Done
<!--NeedCopy-->
```

GUI を使用してマスターへの送信を有効にするには:

1. [システム] > [ネットワーク] に移動し、[設定] グループで [仮想ルーターパラメーター] をクリックします。
2. 「マスターに送信」オプションを選択します。

VRRP 通信間隔の構成

August 15, 2023

アクティブ/アクティブ展開では、すべての NetScaler ノードが仮想ルーター冗長プロトコル（VRRP）を使用して、マスター VIP アドレスとそれに対応する優先順位を VRRP 広告パケット（Hello メッセージ）で定期的にアドバタイズします。

VRRP は次の通信間隔を使用します。

- こんにちはインターバル。マスター VIP アドレスのノードがピアノードに送信する VRRP hello メッセージの間隔。
- デッドインターバル。マスター VIP アドレスのノードから VRRP hello メッセージが受信されない場合に、バックアップ VIP アドレスのノードがマスター VIP アドレスの状態を DOWN と見なすまでの時間。デッドインターバルの後、バックアップ VIP アドレスが引き継ぎ、マスター VIP アドレスになります。

これらの間隔は希望の値に変更できます。これらの通信間隔は両方とも、そのノードのすべての VIP アドレスのノードごとの設定です。

CLI を使用して VRRP 通信間隔を設定するには:

コマンドプロンプトで入力します。

- **set vrIDParam** [-**helloInterval** \<msecs>] [-**deadInterval** \<secs>]
- **sh vrIDParam**

例:


```
> set vrIDParam -helloInterval 500 -deadInterval 2
 Done
<!--NeedCopy-->
```

GUI を使用して VRRP 通信間隔を設定するには:

1. [システム] > [ネットワーク] に移動し、[設定] グループで [仮想ルーターパラメーター] をクリックします。
2. 「仮想ルーターパラメータの設定」で、「**Hello Interval** 」と「**Dead Interval** 」パラメータを設定します。
3. [**OK**] をクリックします。

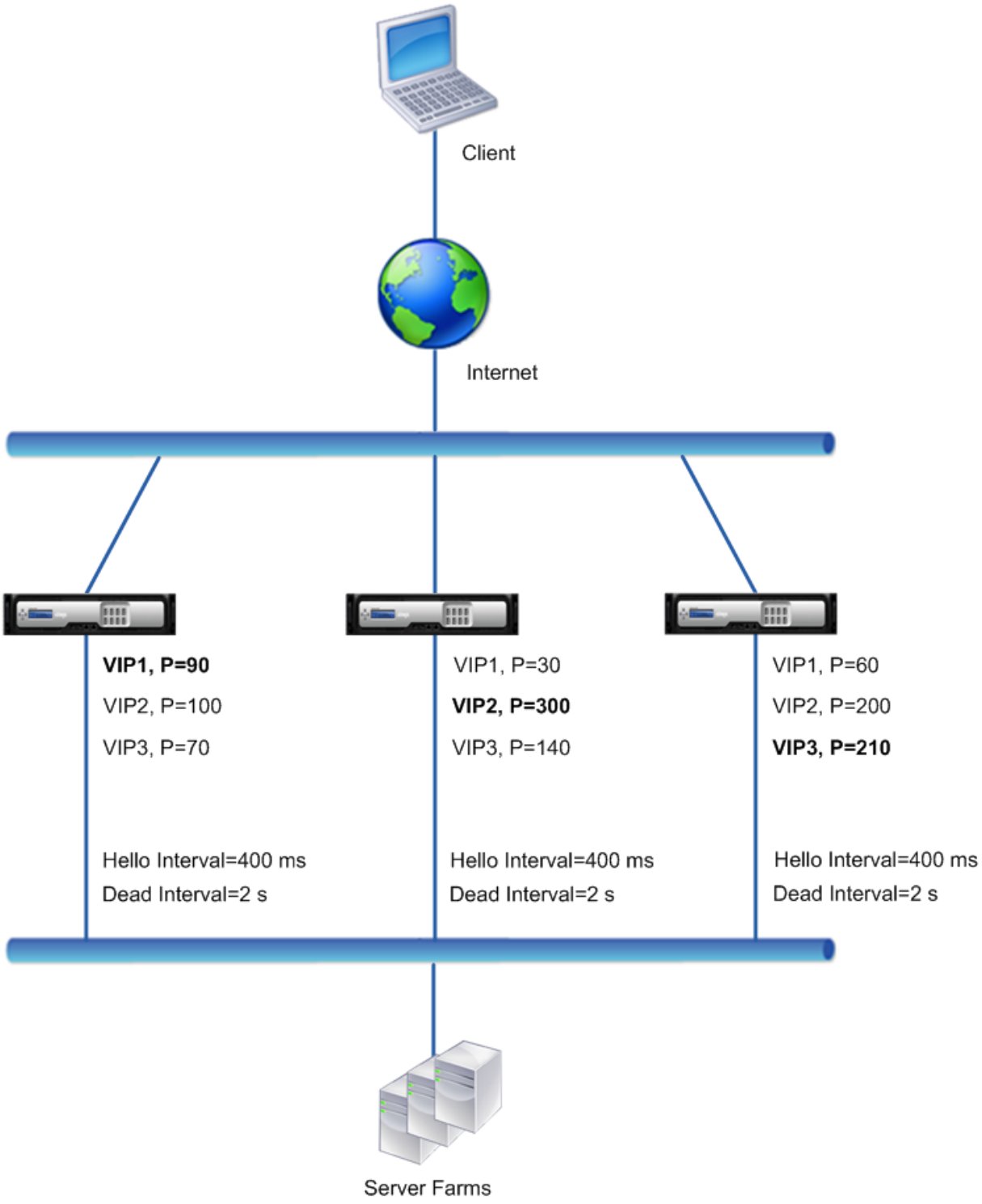
例 **1: VRRP** デッドインターバルが同じノード

NetScalers NS1、NS2、NS3 で構成されるアクティブ/アクティブ環境を考えてみましょう。これらの ADC のそれぞれには、仮想 IP アドレス VIP1、VIP2、VIP3 が設定されています。優先順位の関係で、VIP1 は NS1 でアクティブ、VIP2 は NS2 で、VIP3 は NS3 でアクティブになります。

下の表に示すように、デッドインターバルは 3 つのノードすべてで同じ値 (2 秒) に設定されます。ノードの VRRP 通信間隔（hello インターバルとデッドインターバル）は、そのノードに設定されているすべての VRID に適用され、次にノード上の VRID に関連付けられているすべての VIP アドレスに適用されます。

各ノードでは、そのノードでアクティブ（マスター）の VIP アドレスが hello インターバルを使用し、デッドインターバルはそのノードで非アクティブ（バックアップ）の VIP アドレスに使用されます。3 つのノードすべての VIP アドレスのプリエンプションは無効です。

次の表は、この例で使用される設定の一覧です。VRRP 間隔の例 1 の設定。



実行フローは次のとおりです。

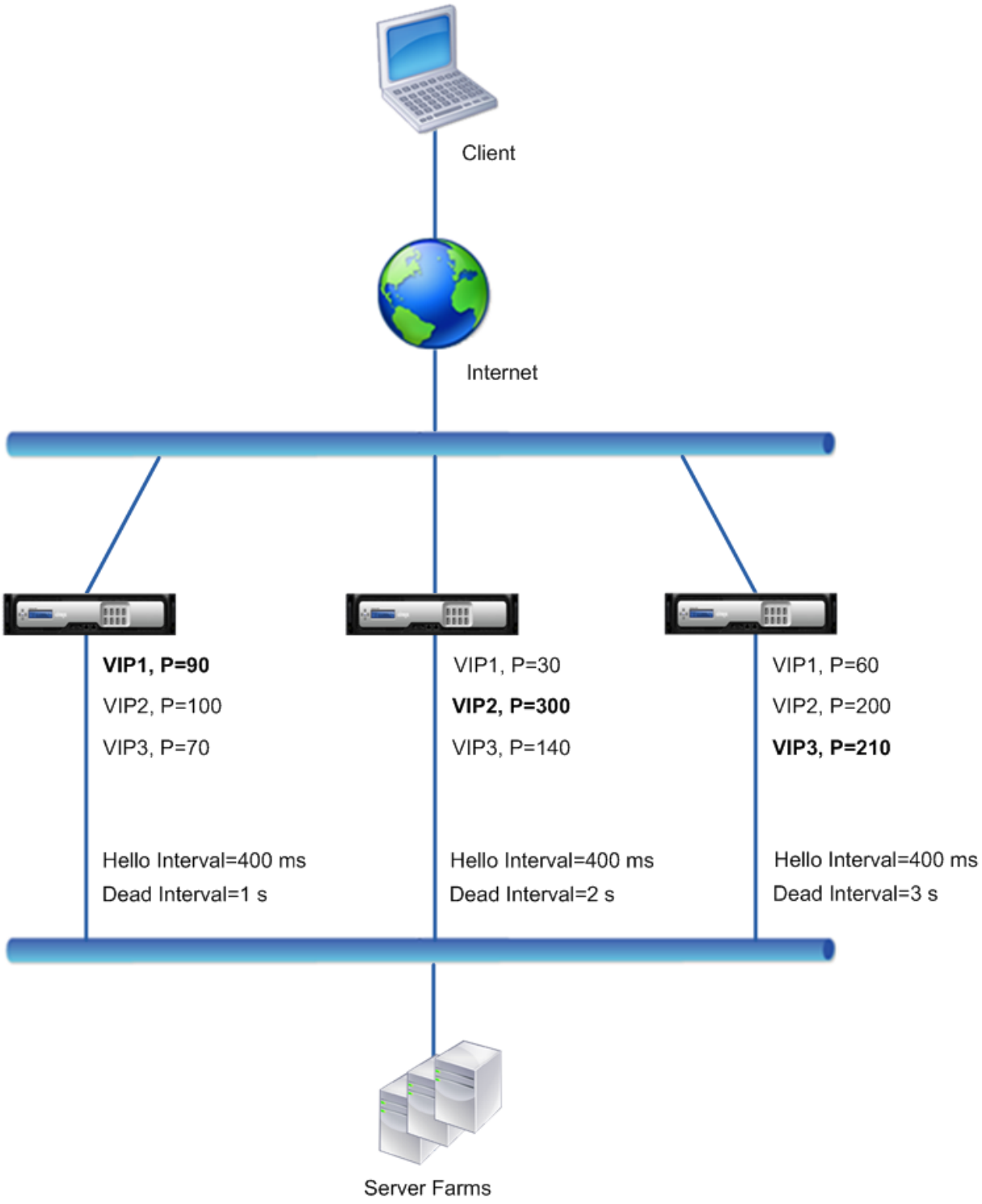
1. NS1 では VIP1 がアクティブ（マスター）であるため、NS1 は 400 ミリ秒に設定された hello 間隔で VIP1 アドレスの NS2 と NS3 に hello メッセージを送信します。同様に、NS2 は VIP2 にハローメッセージを送信し、NS3 は VIP3 にハローメッセージを送信します。

2. NS1 では、VIP2 と VIP3 は NS1 では非アクティブ（バックアップ）であるため、設定したデッドインターバルは VIP2 と VIP3 に適用されます。同様に、NS2 では、設定されたデッドインターバルは VIP1 と VIP3 に適用され、NS3 では、設定されたデッドインターバルは VIP1 と VIP2 に適用されます。
3. NS1 がダウンした場合、NS1 と NS3 は NS1 から hello メッセージを 2 秒間（デッドインターバル）受信しない場合、NS1 がダウンしていると見なします。NS3 の VIP1 が引き継いでアクティブ（マスター）になります。これは、その VRID プライオリティ（60）が NS2 の VIP1 の VRID プライオリティ（30）よりも高いためです。

例 **2: VRRP** デッドインターバルが異なるノード

例 1 で説明した導入と似ているが、各ノード（NS1、NS2、NS3）でデッドインターバルが異なる VRRP 導入を考えてみましょう。3 つのノードすべての VIP アドレスのプリエンプションは無効です。

次の表は、この例で使用される設定の一覧です。VRRP 間隔の例 2 の設定。



NS1 がダウンしたときの実行フローは次のとおりです。

1. NS2 は、NS1 からの Hello メッセージを 2 秒間（NS2 のデッドインターバル）受信しなかった後、NS1 がダウンしていると見なします。
2. NS2 の VIP1 が引き継ぎ、アクティブ（マスター）になります。NS2 が VIP1 への hello メッセージの送信を

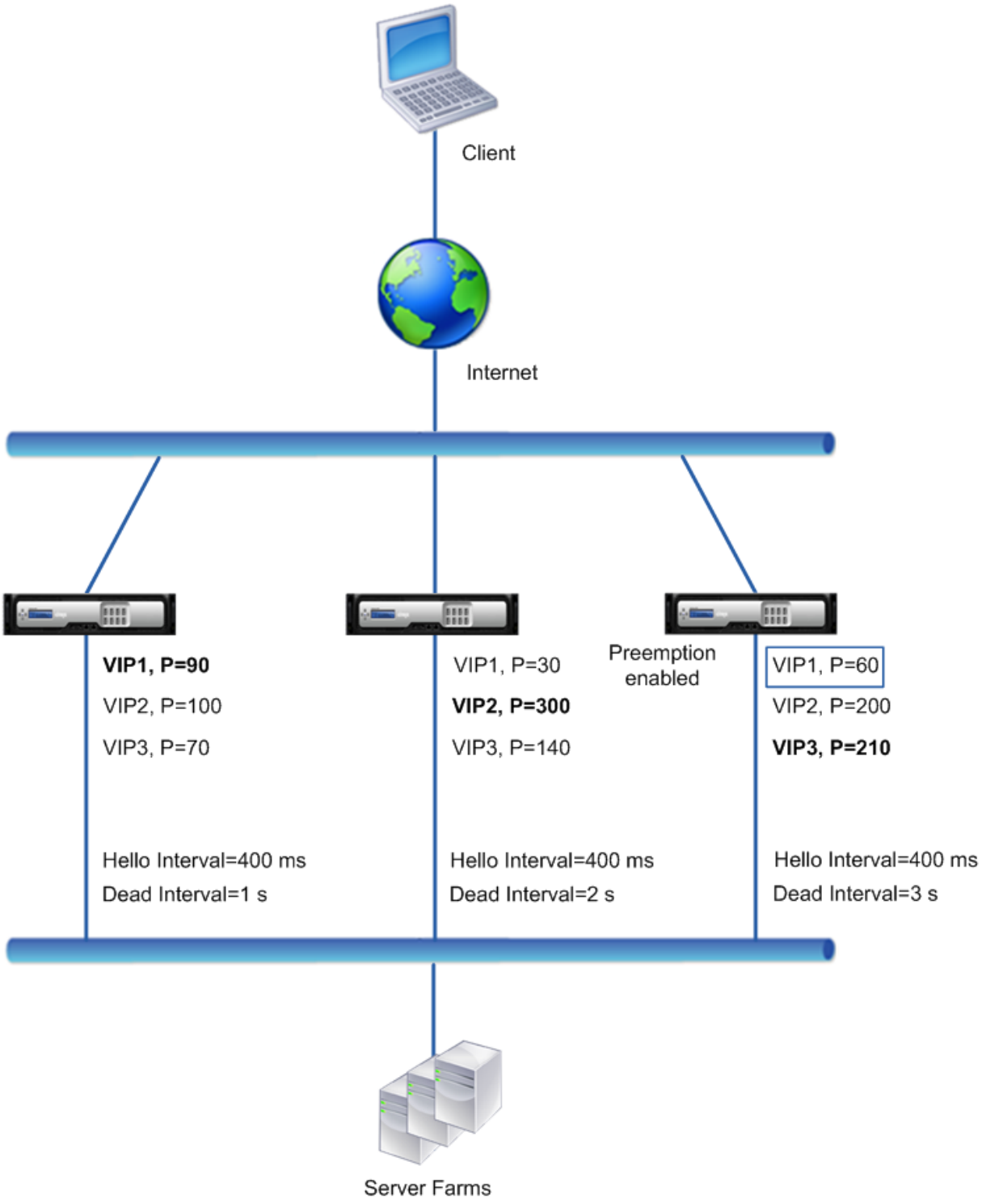
開始するようになりました。

NS3 の VIP1 は NS2 の VIP1 の VRIP 優先度（30）よりも高いですが、NS3 のデッドインターバルが大きいため（NS2 の 2 秒に対して 3 秒）、NS2 の VIP1 がすでに引き継ぐ前に NS3 上の VIP1 が引き継ぐことができません。

例 **3:** デッドインターバルが異なり、プリエンプションが有効になっているノード

例 1 で説明した配置と似ていますが、NS1、NS2、および NS3 の 3 つのノードで dead インターバルが異なり、NS3 の VIP1 アドレスに対してプリエンプションが有効になっている VRRP 配置を検討してください。

次の表は、この例で使用される設定の一覧です。VRRP 間隔の例 3 の設定。



NS1 がダウンしたときの実行フローは次のとおりです。

1. NS2 は、NS1 からの Hello メッセージを 2 秒間（NS2 が設定したデッドインターバル）受信しなかった後、NS1 がダウンしていると見なします。現時点では、デッドインターバルが 3 秒の NS3 は NS1 がダウンしているとは見なしません。

2. NS2 の VIP1 が引き継ぎ、アクティブ（マスター）になります。NS2 が VIP1 への hello メッセージの送信を開始するようになりました。
3. NS2 から VIP1 の hello メッセージを受信すると、NS3 は VIP1 の NS2 をプリエンプションします。これは、NS3 の VIP1 に対してプリエンプションが有効になっていて、NS3 の VIP1 の VRID プライオリティ（60）が NS2 の VIP1 の VRID プライオリティ（30）よりも高いためです。
4. NS3 の VIP1 が引き継ぎ、アクティブ（マスター）になります。NS3 が VIP1 に hello メッセージの送信を開始するようになりました。

インターフェイスの状態に基づいた正常性追跡の構成

August 15, 2023

現在のマスター VIP アドレスのノードが完全にダウンする前にバックアップ VIP アドレスがマスター VIP を引き継ぐようにするには、ノード上のインターフェイスの状態が変化したときに VIP アドレスの優先順位を変更するようにノードを設定できます。たとえば、ノードは、インターフェイスの状態が DOWN に変わると VIP アドレスの優先順位を下げ、インターフェイスの状態が UP に変わると VIP アドレスの優先順位を上げます。この機能は、各 VIP アドレスのノード単位の設定です。

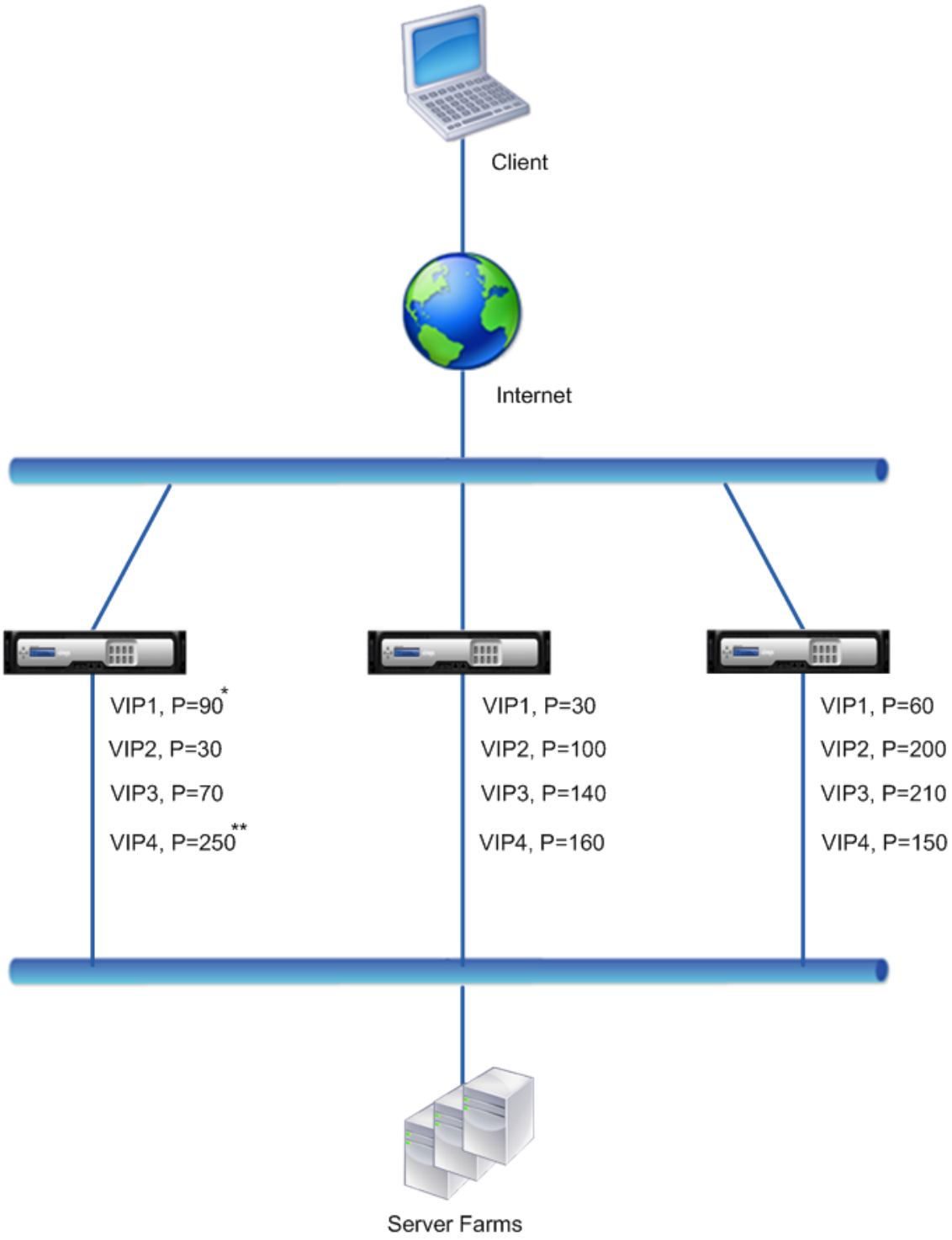
例

NetScalers NS1、NS2、NS3 で構成されるアクティブ/アクティブ環境を考えてみましょう。これらの ADC のそれぞれには、仮想 IP アドレス VIP1、VIP2、VIP3、VIP4 が設定されています。優先順位が異なるため、VIP1 と VIP4 は NS1 でアクティブになり、VIP2 は NS2 で、VIP3 は NS3 でアクティブになります。

NS1 のアクティブな VIP アドレスが NS2 または NS3 のいずれかに引き継がれるように、NS1 の VIP1 アドレスと VIP4 アドレスに対してインターフェイスベースのヘルストラッキングが設定されています。VIP アドレスのインターフェイスベースのヘルストラッキングの設定には、目的のインターフェイスを関連付け、VIP アドレスの関連する VRID の優先度を下げる（TrackIfNumPriority）パラメータを設定することが含まれます。たとえば、NS1 では、インターフェイス 1/2、1/3、1/5 が VIP1 の VRID に関連付けられ、優先度の低下は 20 に設定されます。

3 つのノードすべてで、これらの VIP アドレスに対してプリエンプションが有効になります。

次の表に、この例で使用される設定を示します。ヘルストラッキングの設定例。



* Packet Interfaces = 1/2, 1/3, 1/5
Reduced Priority = 20

** Packet Interfaces = 1/5, 1/7
Reduced Priority = 55

NS1 の複数のインターフェイスがダウンしたときの NS1 の実行フローは次のとおりです。

1. インターフェイス 1/3 がダウンすると、インターフェイス 1/3 が VIP1 に関連付けられているため、アドレス VIP1 のプライオリティは 20（VIP1 のプライオリティ値が下がる）減少します。
    - VIP1 の有効優先度 = (現在の優先度-優先度の低下) = (90-20) = 70
2. 同様に、インターフェイス 1/5 がダウンすると、アドレス VIP1 のプライオリティはさらに低下します。
    - VIP1 の有効優先度 = (現在の優先度-優先度の低下) = (70-20) = 50
3. この時点で、NS1 上の VIP1 の実効優先度は NS3 上の VIP1 の優先度よりも低くなっています。NS3 は VIP1 に対して NS1 をプリエンプションします。NS3 の VIP1 が引き継ぎ、アクティブ（マスター）になります。
4. また、インターフェイス 1/5 も VIP4 に関連付けられているため、VIP4 のプライオリティは VIP4 のプライオリティ値（55）だけ下がります。
    - VIP4 の有効優先度 = (250-55) = 195
5. インターフェイス 1/7 がダウンすると、VIP4 の優先順位はさらに下がります。
    - VIP4 の有効優先度 = (現在の優先度-優先度の低下) = (195-55) = 145
6. この時点で、NS1 上の VIP4 の実効優先度は NS2 上の VIP4 の優先度よりも低くなっています。NS2 は VIP4 に対して NS1 をプリエンプションします。NS3 の VIP4 が引き継ぎ、アクティブ（マスター）になります。この構成では、NS1 が完全にダウンする前に、4 つの VIP アドレスのいずれも NS1 上でアクティブにならないようにします。

IPv4 アクティブ**-**アクティブモードの設定手順

VIP アドレスのノードにこの機能を設定するには、優先度を下げる（TrackIfNumPriority）パラメータを設定し、状態を追跡するインターフェイスを関連付けて VIP アドレスの優先順位を変更します。関連するインターフェイスのいずれかの状態が DOWN または UP に変わると、ノードは VIP アドレスの優先度を、設定されている優先度を下げる（TrackIfNumPriority）値だけ下げたり上げたりします。

CLI を使用して低優先度を設定し、インターフェイスを仮想ルータ ID にバインドするには:

コマンドプロンプトで入力します。

- **set vrID** <id> [-**trackifNumPriority** \<positive_integer>]
- **bind vrID** <id> -**trackifNum** <interface_name>
- **show vrID** <id>

例**:**

```
> set vrID 125 -trackifNumPriority 10
Done

> bind vrID 125 -trackifNum 1/4 1/5
```



```
    Done
<!--NeedCopy-->
```

GUI を使用して低優先度を設定し、インターフェイスを仮想ルータ ID にバインドするには:

1. システムに移動 > 通信網 > **VMAC**。
2. **[**VMACs]** タブで仮想ルーター ID を選択し、[編集] をクリックします。**
3. [仮想 **MAC** の設定] で、[優先度の低下] パラメータを設定します。
4. **VRID** オプションで追跡するインターフェイスを選択し、[アソシエイトインターフェイス] で仮想ルータ ID にインターフェイスを追加します。

IPv6 アクティブ/アクティブモードの設定手順

VIP6 アドレスのノードでこの機能を設定するには、優先度を下げる (TrackIfNumPriority) パラメータを設定し、状態を追跡するインターフェイスを関連付けて VIP6 アドレスの優先順位を変更します。関連するインターフェイスのいずれかの状態が DOWN または UP に変わると、ノードは VIP6 アドレスの優先順位を、設定されている優先度を下げる（TrackIfNumPriority）値だけ下げたり上げたりします。

CLI を使用して VIP アドレスの優先度を自動的に変更するには:

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

- 新しい仮想 MAC6 を追加する場合:
  - **add vrID6** <id> [-**trackifNumPriority** \<positive_integer>]
  - **bind vrID6** <id> -**trackifNum** <interface_name>
  - **show vrID6** <id>
- 既存の仮想 MAC6 を再設定する場合:
  - **set vrID6** <id> [-**trackifNumPriority** \<positive_integer>]
  - **bind vrID6** <id> -**trackifNum** <interface_name>
  - **show vrID6** <id>

例**:**

```
    > set vrID6 130 -trackifNumPriority 10
    Done

    > bind vrID6 130 -trackifNum 1/4 1/5
    Done
<!--NeedCopy-->
```


優先設定の遅延

August 15, 2023

デフォルトでは、バックアップ VIP アドレスのプライオリティがマスター VIP アドレスのプライオリティよりも高くなった直後に、そのマスター VIP アドレスがプリエンプションされます。バックアップ VIP アドレスを設定する場合、プリエンプションを遅延させる時間を指定できます。プリエンプション遅延時間は、各バックアップ VIP アドレスのノード単位の設定です。

バックアップ VIP のプリエンプション遅延設定は、次の状況では適用されません。

- マスター VIP のノードがダウンします。この場合、バックアップ VIP のノードに設定されたデッドインターバルの後、バックアップ VIP がマスター VIP を引き継ぎます。
- マスター VIP のプライオリティは 0 に設定されます。バックアップ VIP のノードに設定されたデッドインターバルの後、バックアップ VIP がマスター VIP を引き継ぎます。

例**:** プリエンプションの遅延

NetScaler アプライアンス NS1 と NS2 で構成されるアクティブ-アクティブ展開を考えてみましょう。仮想 IP アドレス VIP1 は、これらの各アプライアンスで設定されます。優先度が高いため、VIP1 は NS2 のマスターになります。これら 2 つのノードでは、プリエンプションが有効で、VIP1 のプリエンプション遅延時間が設定されています。

次の表に、この例で使用されている設定の一覧を示します。

エンティティとパラメータ	NS1 での設定	NS2 での設定
VIP1 (参照のみを目的としています)	**IP** アドレス**:192.0.1.10**、****VRID: 10**、プライオリティ**:100**、プリエンプション**:** 有効、プリエンプション遅延時間**:**1000 秒 **	**IP** アドレス**:192.0.1.10**、****VRID: 10**、プライオリティ**:200**、プリエンプション**:** 有効、プリエンプション遅延時間**:**2000 秒 **
デッドインターバル	1 秒	2 秒

この設定で発生する可能性のあるプリエンプション動作の例を以下に示します。

- NS1 の VIP1 のプライオリティが NS2 の VIP1 のプライオリティよりも高い値（たとえば 210）に設定されている場合、NS1 上の VIP1 は、設定したプリエンプション遅延時間（1000 秒）が過ぎるとマスターを引き継ぎます。
- 次の VRRP 設定を持つ 3 番目のノード NS3 をこの導入環境に追加すると、NS3 上の VIP1 は、設定されたプリエンプション遅延時間（3000 秒）が経過するとマスターになります。
  - VIP1

    * グリッド:30
    * IP アドレス:
    * 優先度 = 300
    * プリエンプション遅延時間 = 3000 秒

- NS2 がダウンすると、1 秒後に NS1 の VIP1 がマスターを引き継ぎます（NS1 にデッドインターバルを設定）。この場合、NS1 上の VIP1 のプリエンプション遅延時間は適用されません。
- NS2 がダウンして NS1 が再起動した場合、NS1 の VIP1 は NS1 が起動してから 1 秒後にマスターになります（NS1 ではデッドインターバルを設定）。この場合、NS1 上の VIP1 のプリエンプション遅延時間は適用されません。
- NS2 の VIP1 のプライオリティが 0 に設定されている場合、VIP1 はスタンバイモードに移行します。NS1 の VIP1 は 1 秒後にマスターを引き継ぎます（NS1 にデッドインターバルを設定）。この場合、NS1 上の VIP1 のプリエンプション遅延時間は適用されません。

IPv4 アクティブ/アクティブモードの遅延プリエンプションの設定

VIP アドレスのプリエンプション遅延時間を設定するには、関連する仮想 MAC アドレスのプリエンプション遅延タイマーパラメータを設定します。アドレスを追加するときにこのパラメータを設定することも、既存の仮想 MAC アドレスを変更することもできます。

CLI を使用してプリエンプション遅延時間を設定するには:

- 仮想 MAC の追加時にプリエンプション遅延時間を設定するには、コマンドプロンプトで次のように入力します。
    - **add vrID** <id> -**preemptiondelaytimer** <secs>
    - **VRID** を表示
- 仮想 MAC の変更中にプリエンプション遅延時間を設定するには、コマンドプロンプトで次のように入力します。
    - **set vrID** <id> -**preemptiondelaytimer** <secs>
    - **VRID** を表示

GUI を使用してプリエンプション遅延時間を設定するには:

1. [システム] > [ネットワーク] > [**VMAC**] に移動します。
2. [**VMAC**] タブにあります。新しい仮想 MAC を追加するとき、または既存の仮想 MAC を編集するときに、プリエンプション遅延タイマーパラメータを設定します。

設定例:

次の構成では、「例: プリエンプションの遅延」の表に記載されている設定を使用しています。


```
    Settings on NS1

    > set vrid param  -deadInterval 1

    Done

    > add ns ip 192.0.1.10 255.255.255.255  -type VIP

    Done

    > add vrid 10  -Priority 100  -Preemption Enable  -
       preemptiondelaytimer 1000

    Done

    > bind ns ip 192.0.1.10 255.255.255.255  -vrid 10

    Done

    Settings on NS2

    > set vrid param  -deadInterval 2

    Done

    > add ns ip 192.0.1.10 255.255.255.255  -type VIP

    Done

    > add vrid 20  -Priority 200  -Preemption Enable  -
       preemptiondelaytimer 2000

    Done

    > set ns ip 192.0.1.10 255.255.255.255  -vrid 10

    Done
<!--NeedCopy-->
```

IPv6 アクティブ/アクティブモードの遅延プリエンプションの設定

VIP6 アドレスのプリエンプション遅延時間を設定するには、関連する仮想 MAC6 アドレスのプリエンプション遅延タイマーパラメータを設定します。このパラメータは、仮想 MAC6 アドレスを追加するときに設定することも、既存の仮想 MAC6 アドレスを変更することもできます。

CLI を使用してプリエンプション遅延時間を設定するには:

- 仮想 MAC6 の追加時にプリエンプション遅延時間を設定するには、コマンドプロンプトで次のように入力します。

  - **add vrID6** <id> -**preemptiondelaytimer** <secs>
  - **show vrID6**

- 仮想 MAC6 の変更中にプリエンプション遅延時間を設定するには、コマンドプロンプトで次のように入力します。

  - **set vrID6** <id> -**preemptiondelaytimer** <secs>
  - **show vrID6**

GUI を使用してプリエンプション遅延時間を設定するには:

1. [システム] > [ネットワーク] > [**VMAC**] に移動します。
2. **VMAC6** タブにあります。仮想 MAC6 アドレスを追加するとき、または既存の仮想 MAC6 アドレスを編集するときに、プリエンプション遅延タイマーパラメータを設定します。

VIP アドレスのバックアップ状態を保持

August 15, 2023

VIP アドレスを常にバックアップ状態に保つように強制できます。この操作は、VRRP 導入環境のメンテナンスやテストに役立ちます。

VIP アドレスが強制的にバックアップ状態のままでいると、その VIP アドレスは VRRP 状態遷移には関与しません。また、他のノードがすべてダウンしてもマスターになることはできません。

VIP アドレスを強制的にバックアップ状態のままにするには、関連する仮想 MAC アドレスの優先順位を 0 に設定します。ノードのメンテナンスプロセス中にノードの VIP アドレスがトラフィックを処理しないようにするには、すべての優先度をゼロに設定します。

アドレスを追加または変更するときに、仮想 MAC アドレスの優先順位を設定できます。

CLI を使用して VIP アドレスを強制的にバックアップ状態のままにするには:

- 仮想 MAC を追加する際に優先順位を設定するには、コマンドプロンプトで次のように入力します。

  - **add vrID** <id> -**priority** 0
  - **VRID** を表示

- 仮想 MAC の変更中に優先順位を設定するには、コマンドプロンプトで次のように入力します。

  - **set vrID** <id> -**priority** 0
  - **VRID** を表示

GUI を使用して VIP アドレスを強制的にバックアップ状態に保つには:

1. [システム] > [ネットワーク] > [**VMAC**] に移動します。

2. [**VMAC**] タブで、新しい仮想 MAC の追加または既存の仮想 MAC の編集中に、**Priority** パラメータをゼロに設定します。

ネットワークビジュアライザー

August 15, 2023

ネットワークビジュアライザーには、NetScaler アプライアンス上のすべてのインターフェイス、チャネル、VLAN、IP アドレス、VLAN へのバインディングがグラフィカルに表示されます。有効なインターフェイスまたはチャネルには黒いラベルが表示されます。無効になっているインターフェイスまたはチャネルには赤いラベルが表示されます。

アプライアンスのネットワーク接続の全体像は、ネットワーク設計の欠陥を検出したり、ネットワークを最適化したりするのに役立ちます。また、新しい管理者がアプライアンスのネットワーク構成を簡単に理解するのにも役立ちます。

ネットワークビジュアライザーを開くには:

[システム] > [ネットワーク] に移動します。「モニター接続」で、「ネットワークビジュアライザー」をクリックします。

Link Layer Discovery Protocol（LLDP）の構成

August 15, 2023

NetScaler は、業界標準（IEEE 802.1AB）のリンク層検出プロトコル（LLDP）をサポートしています。LLDP はレイヤー 2 プロトコルで、NetScaler がその ID と機能を直接接続されたデバイスにアドバタイズしたり、これらの隣接デバイスの ID と機能を学習したりできるようにします。

> 注:
> リンク層検出プロトコル（LLDP）は、NetScaler MPX プラットフォームでのみサポートされています。

NetScaler は LLDP を使用して、LLDP パケットデータユニット（LLDPU）と呼ばれる LLDP メッセージの形式で情報を送受信します。LLDPU は、タイプ、長さ、値 (TLV) 情報要素のシーケンスです。各 TLV には、LLDPDU を送信するデバイスに関する特定の種類の情報が保持されます。NetScaler は、各 LLDPU で次の TLV を送信します。

- シャーシ ID
- ポート ID
- 存続期間の価値
- システム名

- システム説明
- ポートの説明
- システム機能
- 管理住所
- ポート VLAN ID
- リンクアグリゲーション

注**:** LLDP メッセージで送信する TLV は指定できません。

NetScaler インターフェイスは、次の LLDP モードをサポートしています。

- なし。インターフェイスは、直接接続されたデバイスから LLDP メッセージを受信することも、送信することもありません。
- **TRANSMITTER**. インターフェイスは直接接続されたデバイスに LLDP メッセージを送信しますが、直接接続されたデバイスから LLDP メッセージは受信しません。
- **RECEIVER**. インターフェイスは直接接続されたデバイスから LLDP メッセージを受信しますが、直接接続されたデバイスには LLDP メッセージは送信しません。
- **TRANSCEIVER**. インターフェイスは、直接接続されたデバイスと LLDP メッセージを送受信します。

インターフェイスの LLDP モードは、グローバルレベルとインターフェイスレベルで設定された LLDP モードによって異なります。次の表に、グローバルレベルとインターフェイスレベル設定の組み合わせによるモードを示します。インターフェイスレベルとグローバルレベル LLDP モード。

NetScaler ADC によって送受信される LLDP メッセージに関連する次の点に注意してください。

- **LLDP** メッセージの送信。NetScaler は、トランスミッターまたはトランシーバー LLDP モードのどちらかで動作しているインターフェイスから LLDPU を送信します。

  NetScaler のグローバル LLDP 送信パラメータは次のとおりです。

  - タイマー。NetScaler が直接接続されたデバイスに LLDPU を送信する間隔（秒単位）。
  - ホールドタイムマルチプライヤー。受信側デバイスが LLDP 情報を破棄または削除する前に LLDP 情報をデータベースに保存する期間を計算するための乗数です。持続時間は、「ホールドタイムマルチプライヤー」パラメータ値に「タイマー」パラメータ値を掛けて計算されます。

- **LLDP** メッセージの受信。NetScaler は、LLDPDU 情報を管理情報ベース（MIB）に保存します。保存されている LLDP 情報は、LLDPDU を受信したインターフェイスの ID に基づいて分類またはグループ化されます。NetScaler は、受信した LLDPU で指定された期間中、この LLDP 情報を保持します。

  ADC がインターフェイス上で別の LLDPDU を受信すると、そのインターフェイスについて保存されている LLDP 情報が破棄される前に、ADC はそのインターフェイスに保存されている LLDP 情報を新しい LLDPDU の情報に置き換えます。

構成の手順

NetScaler アプライアンスでの LLDP の構成は、次のタスクで構成されます。

1. グローバルレベルの **LLDP** パラメータを設定します。このタスクでは、LLDP タイマー、ホールドタイムマルチプライヤ、LLDP モードなどのグローバル LLDP パラメータを設定します。
2. インターフェイスレベルの **LLDP** パラメータを設定します。このタスクでは、インターフェイスの LLDP モードを設定します。
3. （オプション）ネイバーデバイス情報を表示します。NetScaler のすべてのインターフェイスで収集されたネイバーデバイスの LLDP 情報を表示することも、特定のインターフェイスで収集された LLDP 情報のみを表示することもできます。インターフェイスを指定しない場合、すべてのインターフェイスの情報が表示されます。

NetScaler で LLDP を構成するための前提条件は次のとおりです。

1. 標準の LLDP プロトコル (IEEE 802.1AB) を理解していることを確認してください。
2. 直接接続された目的のデバイスに LLDP が設定されていることを確認します。

CLI のプロシージャ

CLI を使用してグローバルレベルの LLDP パラメータを設定するには：

コマンドプロンプトで入力します。

- set lldp param [-[-holdtimeTxMult <positive_integer>][-timer <positive_integer>] [-Mode \<Mode>]
- show lldp param

CLI を使用して LLDP のインターフェイスを設定するには：

コマンドプロンプトで入力します。

- set interface <id> -lldpmode <lldpmode>
- show interface <id>

CLI を使用してネイバーデバイス情報を表示するには：

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- show lldp neighbors
- show lldp neighbors <ifnum>

GUI のプロシージャ

GUI を使用してグローバルレベルの LLDP パラメータを設定するには：

1. [システム] > [ネットワーク] に移動し、[LLDP パラメータの設定] をクリックします。
2. 次のパラメーターを設定します。

    - ホールドタイマーマルチプライヤー
    - タイマー
    - Mode

GUI を使用して LLDP のインターフェイスを設定するには:

[システム] > [ネットワーク] > [インターフェイス] に移動し、インターフェイスを開き、LLDP モードパラメータを設定します。

GUI を使用してネイバーデバイス情報を表示するには:

[システム] > [ネットワーク] > [インターフェイス] に移動し、[アクション] リストで [LLDP ネイバーの表示] を選択します。

クラスタ設定での **LLDP** サポート

クラスタ設定では、GUI または CLI にクラスタ IP アドレス（CLIP）を介してアクセスすると、すべてまたは特定のクラスタノードの LLDP ネイバー設定が GUI と CLI に表示されます。グローバルレベルの LLDP モードに加えられた変更は、各クラスタノードのグローバルレベルの LLDP モードに適用されます。

NS1、NS2、NS3 の 3 つのノードで構成されるクラスタ設定の例を考えてみましょう。これらのノードはそれぞれ、ルータ 1 とルータ 2 の両方に接続されています。クラスタセットアップのクラスタ IP アドレス（CLIP）を使用してアクセスするクラスタ CLI で **show lldp neighbor-summary** 操作を実行すると、次の出力が表示されます。出力には、これらすべてのノードの LLDP ネイバー情報が表示されます。

```
> show lldp neighbor -summary

Node Id: 1
------------------------------------------------------------
    Interface    ChassisId           PortId     System name
------------------------------------------------------------
1     1/1/1      fe:c7:3b:13:bd:11    1/1         Router-1

2     1/1/2      12:68:7b:9e:4c:11    1/1         Router-2

Node Id: 2
------------------------------------------------------------
    Interface    ChassisId           PortId     System name
------------------------------------------------------------
1     2/1/1      fe:c7:3b:13:bd:12    1/2        Router-1

2     2/1/2      12:68:7b:9e:4c:12    1/2        Router-2

Node Id: 3
------------------------------------------------------------
```



```
    Interface    ChassisId            PortId     System name
-------------------------------------------------------------

1     3/1/1       fe:c7:3b:13:bd:13    1/3        Router-1

2     3/1/2       12:68:7b:9e:4c:13    1/3        Router-2

Done
<!--NeedCopy-->
```

ジャンボフレーム

August 15, 2023

NetScaler アプライアンスは、最大 9216 バイトの IP データを含むジャンボフレームの送受信をサポートします。ジャンボフレームでは、標準の IP MTU サイズ（1500 バイト）を使用するよりも効率的に大きなファイルを送信することができます。

NetScaler アプライアンスは、以下の展開シナリオでジャンボフレームを使用することができます。

- ジャンボで受信/ジャンボで送信。アプライアンスがデータをジャンボフレームで受信し、それをジャンボフレームで送信します。
- 非ジャンボで受信/ジャンボで送信。アプライアンスがデータを通常のフレームで受信し、それをジャンボフレームで送信します。
- ジャンボで受信/非ジャンボで送信。アプライアンスがデータをジャンボフレームで受信し、それを通常のフレームで送信します。

NetScaler アプライアンスは、次のプロトコルの負荷分散構成でジャンボフレームをサポートします。

- TCP
- TCP 経由の任意のプロトコル (HTTP など)
- SIP
- RADIUS

NetScaler ADC アプライアンスでのジャンボフレームサポートの構成

August 15, 2023

NetScaler アプライアンスがジャンボフレームをサポートできるようにするには、インターフェイスまたは LA チャネル、および NetScaler アプライアンスにジャンボフレームをサポートさせたい VLAN の MTU を 1500 以上に設定します。

NetScaler アプライアンスのインターフェイス、LA チャネル、または VLAN の MTU を設定する前に考慮すべきポイント

1. LA チャネルを作成すると、チャネルに MTU が指定されていない場合、チャネルは最初にバインドされたインターフェイスの MTU を使用します。
2. チャネルの MTU は、バインドされたすべてのインターフェイスに伝播されます。
3. MTU がインターフェイスの MTU と異なるチャネルにインターフェイスがバインドされると、そのインターフェイスは非アクティブリストに追加されます。
4. メンバーインターフェイスの MTU を変更すると、そのインターフェイスは非アクティブリストに追加されます。
5. インターフェイスがチャネルからバインド解除されても、インターフェイスはチャネルの MTU 値を保持します。
6. インターフェイス、チャネル、または VLAN の MTU を 1500～9216 の範囲の値に設定できます。
7. デフォルト VLAN には MTU を設定できません。NetScaler アプライアンスはインターフェースの MTU を使用して、デフォルト VLAN との間でデータを送受信します。
8. NetScaler アプライアンスの負荷分散構成の TCP ベースのトラフィックの場合、ジャンボフレームをサポートするように各エンドポイントでそれに応じて MSS が設定されます。
    - クライアントと NetScaler アプライアンス上の負荷分散仮想サーバー間の接続では、NetScaler アプライアンスの MSS が TCP プロファイルで設定され、負荷分散仮想サーバーにバインドされます。
    - NetScaler アプライアンスとサーバー間の接続では、NS1 の MSS が TCP プロファイルで設定され、NetScaler アプライアンス上のサーバーを表すサービスにバインドされます。
    - デフォルトでは、TCP プロファイル nstcp_default_profile は NetScaler アプライアンス上のすべての TCP ベースの負荷分散サーバーとサービスにバインドされます。
    - ジャンボフレームをサポートするには、TCP プロファイル nstcp_default_profile の MSS 値を変更するか、カスタム TCP プロファイルを作成してそれに応じて MSS を設定し、カスタム TCP プロファイルを目的の負荷分散仮想サーバーとサービスにバインドできます。
    - すべての TCP プロファイルのデフォルトの MSS 値は 1460 です。

CLI のプロシージャ

CLI を使用してインターフェイスの MTU を設定するには:

コマンドプロンプトで入力します。

- set interface <id> -mtu <positive_integer>
- show interface <id>

例**:**


```
> set interface 10/1  -mtu 9000
Done
<!--NeedCopy-->
```

CLI を使用してチャネルの MTU を設定するには:

コマンドプロンプトで入力します。

- set channel <id> -mtu <positive_integer>
- show channel <id>

例:

```
> set channel LA/1  -mtu 9000
Done
<!--NeedCopy-->
```

CLI を使用して VLAN の MTU を設定するには:

コマンドプロンプトで入力します。

- add vlan <id> -mtu <positive_integer>
- show vlan <id>

例:

```
> set vlan 20  -mtu 9000
Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用してインターフェイスの MTU を設定するには:

[システム] > [ネットワーク] > [インターフェイス] に移動し、インターフェイスを開き、[最大伝送単位] パラメーターを設定します。

GUI を使用してチャネルの MTU を設定するには:

[システム] > [ネットワーク] > [チャネル] に移動し、チャネルを開いて、[最大伝送単位] パラメータを設定します。

GUI を使用して VLAN の MTU を設定するには:

[システム] > [ネットワーク] > [VLAN] に移動し、VLAN を開いて、[最大転送単位] パラメータを設定します。

ユースケース 1 –ジャンボからジャンボへのセットアップ

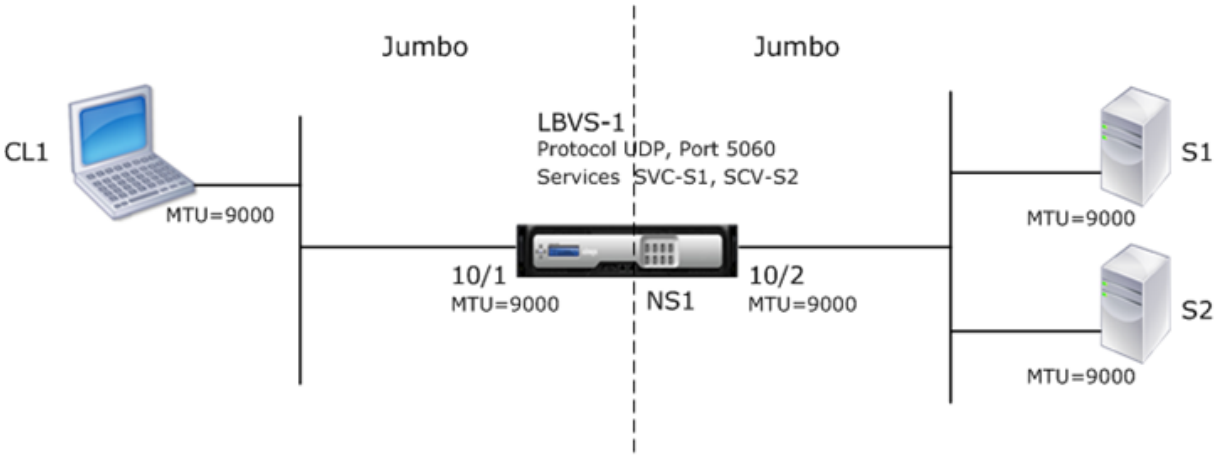
August 15, 2023

ジャンボツージャンボ設定の例を考えてみましょう。NetScaler アプライアンス NS1 上に構成された SIP 負荷分散仮想サーバー LBVS-1 を使用して、サーバー S1 と S2 間で SIP トラフィックの負荷分散を行います。クライアント CL1 と NS1 間の接続、および NS1 とサーバ間の接続は、ジャンボフレームをサポートします。

NS1 のインターフェイス 10/1 は、クライアント CL1 との間でトラフィックを送受信します。NS1 のインターフェイス 10/2 は、サーバ S1 または S2 との間でトラフィックを送受信します。NS1 のインターフェイス 10/1 と 10/2 は、それぞれ VLAN 10 と VLAN 20 の一部です。

ジャンボフレームをサポートするため、NS1 では MTU を 9216 に設定し、インターフェイス 10/1、10/2、および VLAN 10、VLAN 20 では MTU を 9216 に設定します。

このセットアップ例では、CL1、S1、S2 などの他のすべてのネットワークデバイスも、ジャンボフレームをサポートするように設定されています。



次の表は、例で使用される設定の一覧です。

エンティティ	名前	詳細
クライアント CL1 の IP アドレス	-	192.0.2.10
サーバの IP アドレス	S1	198.51.100.19
		S2
NS1 の SNIP アドレス		198.51.100.18
NS1 のインターフェイスと VLAN に指定された MTU	10/1	9000
		10/2
		VLAN 10
		VLAN 20
NS1 上のサーバを表すサービス	SVC-S1	**IP** アドレス**:198.51.100.19**、**** プロトコル:SIP、ポート:5060**

エンティティ	名前	詳細
VLAN 10 上の負荷分散仮想サーバ	LBVS-1	SVC-S2 **IP** アドレス**:203.0.113.15, **** プロトコル**:SIP,** ポート**:5060,** バウンドサービス**:SVC-S1, SVC-S2****

CL1 の NS1 への要求のトラフィックフローは次のとおりです。

1. CL1 は 20000 バイトの SIP リクエストを作成して NS1 の LBVS-1 に送信します。
2. CL1 は IP フラグメントで要求データを LBVS-1 に送信します。各 IP フラグメントのサイズは、CL1 がこれらのフラグメントを NS1 に送信するインターフェイスに設定された MTU（9000）以下になります。
    - 最初の IP フラグメントのサイズ = [IP ヘッダー + UDP ヘッダー + SIP データセグメント] = [20 + 8 + 8972] = 9000
    - 2 番目の IP フラグメントのサイズ = [IP ヘッダー + SIP データセグメント] = [20 + 8980] = 9000
    - 最後の IP フラグメントのサイズ = [IP ヘッダー + SIP データセグメント] = [20 + 2048] = 2068
3. NS1 は、インターフェイス 10/1 で要求 IP フラグメントを受信します。NS1 はこれらのフラグメントを受け入れます。これは、各フラグメントのサイズがインターフェイス 10/1 の MTU（9000）以下であるためです。
4. NS1 はこれらの IP フラグメントを再構成して、20000 バイトの SIP リクエストを形成します。NS1 はこの要求を処理します。
5. LBVS-1 の負荷分散アルゴリズムにより、サーバー S1 が選択されます。
6. NS1 は IP フラグメント内の要求データを S1 に送信します。各 IP フラグメントのサイズは、NS1 がこれらのフラグメントを S1 に送信するインターフェイス 10/2 の MTU（9000）と同じかそれ以下です。IP パケットは、NS1 の SNIP アドレスで発信されます。
    - 最初の IP フラグメントのサイズ = [IP ヘッダー + UDP ヘッダー + SIP データセグメント] = [20 + 8 + 8972] = 9000
    - 2 番目の IP フラグメントのサイズ = [IP ヘッダー + SIP データセグメント] = [20 + 8980] = 9000
    - 最後の IP フラグメントのサイズ = [IP ヘッダー + SIP データセグメント] = [20 + 2048] = 2068

この例の CL1 に対する S1 の応答のトラフィックフローを次に示します。

1. サーバ S1 は 30000 バイトの SIP 応答を作成し、NS1 の SNIP アドレスに送信します。
2. S1 は、応答データを IP フラグメントで NS1 の SNIP アドレスに送信します。各 IP フラグメントのサイズは、S1 がこれらのフラグメントを NS1 に送信するインターフェイスに設定されている MTU（9000）以下になります。
    - 最初の IP フラグメントのサイズ = [IP ヘッダー + UDP ヘッダー + SIP データセグメント] = [20 + 8 + 8972] = 9000

- 2 番目と 3 番目の IP フラグメントのサイズ = [IP ヘッダー + SIP データセグメント] = [20 + 8980] = 9000
- 最後の IP フラグメントのサイズ = [IP ヘッダー + SIP データセグメント] = [20 + 3068] = 3088

3. NS1 は、インターフェイス 10/2 で応答 IP フラグメントを受信します。NS1 はこれらのフラグメントを受け入れます。これは、各フラグメントのサイズがインターフェイス 10/2 の MTU（9000）以下であるためです。
4. NS1 はこれらの IP フラグメントを再構成して、30000 バイトの SIP 応答を形成します。NS1 はこのレスポンスを処理します。
5. NS1 は IP フラグメント内の応答データを CL1 に送信します。各 IP フラグメントのサイズは、NS1 がこれらのフラグメントを CL1 に送信するインターフェイス 10/1 の MTU（9000）と等しいか、それより小さくなります。IP フラグメントは LBVS-1 の IP アドレスから発信されます。

    - 最初の IP フラグメントのサイズ = [IP ヘッダー + UDP ヘッダー + SIP データセグメント] = [20 + 8 + 8972] = 9000
    - 2 番目と 3 番目の IP フラグメントのサイズ = [IP ヘッダー + SIP データセグメント] = [20 + 8980] = 9000
    - 最後の IP フラグメントのサイズ = [IP ヘッダー + SIP データセグメント] = [20 + 3068] = 3088

構成タスク

次の表は、NetScaler アプライアンスで必要な構成を作成するためのタスク、NetScaler コマンド、および例を示しています。

タスク	NetScaler コマンド構文	例
ジャンボフレームをサポートするために必要なインターフェイスの MTU を設定します。	set interface -mtu , show interface	set int 10/1 -mtu 9000 set int 10/2 -mtu 9000
VLAN を作成し、ジャンボフレームをサポートするために必要な VLAN の MTU を設定します	VLAN を追加 -mtu 、VLAN を表示	add vlan 10 -mtu 9000 add vlan 20 -mtu 9000
インターフェイスを VLAN にバインド	bind vlan -ifnum , show vlan	バインド VLAN 10-ifnum 10/1 バインド VLAN 20-ifnum 10/2
SNIP アドレスを追加する	add ns ip -type SNIP, show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP
SIP サーバーを表すサービスの作成	add service SIP_UDP , show service	add service SVC-S1 198.51.100.19 SIP_UDP 5060 add service SVC-S2 198.51.100.20 SIP_UDP 5060

タスク	NetScaler コマンド構文	例
SIP 負荷分散仮想サーバを作成し、そのサーバにサービスをバインドする	add lb vserver SIP_UDP bind lb vserver , show lb vserver	add lb vserver LBVS-1 SIP_UDP 203.0.113.15 5060 bind lb vserver LBVS-1 SVC-S1 bind lb vserver LBVS-1 SVC-S2
構成を保存します	save ns config, show ns config	

ユースケース **2** –ジャンボ以外からジャンボへのセットアップ

August 15, 2023

NetScaler アプライアンス NS1 上に構成された負荷分散仮想サーバー LBVS-1 を使用して、サーバー S1 と S2 のトラフィックの負荷分散を行う通常のセットアップからジャンボセットアップの例を考えてみましょう。クライアント CL1 と NS1 間の接続は通常のフレームをサポートし、NS1 とサーバ間の接続はジャンボフレームをサポートします。

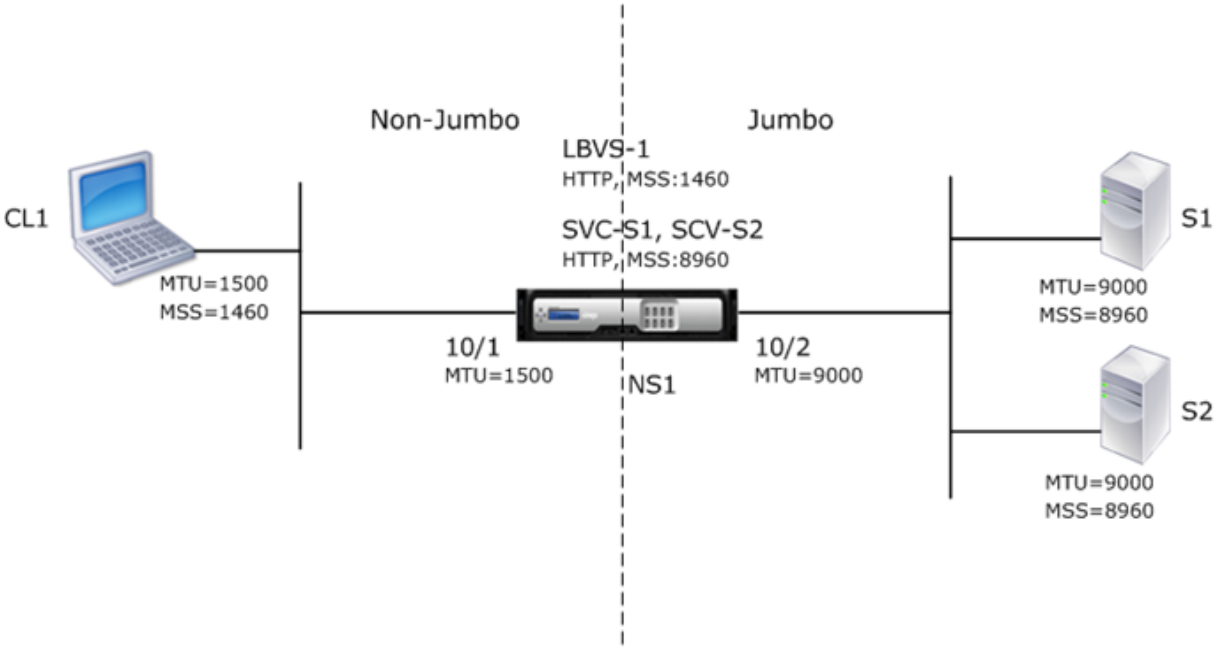
NS1 のインターフェイス 10/1 は、クライアント CL1 との間でトラフィックを送受信します。NS1 のインターフェイス 10/2 は、サーバ S1 または S2 との間でトラフィックを送受信します。

NS1 のインターフェイス 10/1 と 10/2 は、それぞれ VLAN 10 と VLAN 20 の一部です。CL1 と NS1 の間の通常のフレームのみをサポートするため、MTU はインターフェイス 10/1 と VLAN 10 の両方でデフォルト値の 1500 に設定されます。

NS1 とサーバ間のジャンボフレームをサポートするために、インターフェイス 10/2 と VLAN 20 の MTU は 9000 に設定されます。NS1 とサーバ間のサーバおよびその他すべてのネットワーク・デバイスも、ジャンボ・フレームをサポートするように構成されています。

HTTP トラフィックは TCP に基づいているため、ジャンボフレームをサポートするために、各エンドポイントで MSS が適切に設定されます。

- NS1 と S1 または S2 の SNIP アドレス間の接続でジャンボフレームをサポートする場合、NS1 の MSS はカスタム TCP プロファイルで適切に設定されます。このプロファイルは、NS1 上の S1 と S2 を表すサービス（SVC-S1 と SVC-S1）にバインドされます。
- CL1 と NS1 の仮想サーバ LBVS-1 間の接続で通常のフレームのみをサポートするには、デフォルトの TCP プロファイル nstcp_default_profile が使用されます。このプロファイルは、デフォルトで LBVS-1 にバインドされ、MSS がデフォルト値の 1460 に設定されています。



次の表に、この例で使用されている設定の一覧を示します。

エンティティ	名前	詳細
クライアント CL1 の IP アドレス		192.0.2.10
サーバの IP アドレス	S1	198.51.100.19
		S2
NS1 の SNIP アドレス		198.51.100.18
NS1 のインターフェイスと VLAN に指定された MTU	10/1	1500
		10/2
		VLAN 10
		VLAN 20
デフォルトの TCP プロファイル	nstcp_default_profile	MSS:1460
カスタム TCP プロファイル	NS1-SERVERS-JUMBO	MSS: 8960
NS1 上のサーバを表すサービス	SVC-S1	IP アドレス:198.51.100.19、プロトコル:HTTP、ポート:80、TCP プロファイル:NS1-SERVERS-JUMBO (MSS: 8960) SVC-S2

エンティティ	名前	詳細
VLAN 10 上の負荷分散仮想サーバ	LBVS-1	IP アドレス = 203.0.113.15、プロトコル:HTTP、ポート:80、バウンドサービス:SVC-S1、SVC-S2、TCP プロファイル:nstcp_default_profile (MSS: 1460)

この例では、CL1 が S1 に要求するトラフィックフローを次に示します。

1. クライアント CL1 は、NS1 の仮想サーバー LBVS-1 に送信する 200 バイトの HTTP 要求を作成します。
2. CL1 は NS1 の LBVS-1 への接続をオープンします。CL1 と NS1 は、接続の確立中にそれぞれの TCP MSS 値を交換します。
3. NS1 の MSS は HTTP 要求よりも大きいので、CL1 は 1 つの IP パケットで要求データを NS1 に送信します。

   リクエストパケットのサイズ = [IP ヘッダー + TCP ヘッダー + TCP リクエスト] = [20 + 20 + 200] = 240
4. NS1 は、インターフェイス 10/1 で要求パケットを受信し、パケット内の HTTP 要求データを処理します。
5. LBVS-1 の負荷分散アルゴリズムはサーバー S1 を選択し、NS1 はその SNIP アドレスの 1 つと S1 の間の接続を開きます。NS1 と CL1 は、接続の確立中にそれぞれの TCP MSS 値を交換します。
6. S1 の MSS は HTTP 要求よりも大きいので、NS1 は 1 つの IP パケットで要求データを S1 に送信します。

   要求パケットのサイズ = [IP ヘッダー + TCP ヘッダー + [TCP 要求] = [20 + 20 + 200] = 240

次に、この例の CL1 に対する S1 の応答のトラフィックフローを示します。

1. サーバー S1 は、NS1 の SNIP アドレスに送信する 18000 バイトの HTTP 応答を作成します。
2. S1 は応答データを NS1 の MSS の倍数に分割し、これらのセグメントを IP パケットとして NS1 に送信します。これらの IP パケットは、S1 の IP アドレスから送信され、NS1 の SNIP アドレスを宛先とします。

   - 最初の 2 つのパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (TCP セグメント = NS1 の MSS サイズ)] = [20 + 20 + 8960] = 9000
   - 最後のパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (残りの TCP セグメント)] = [20 + 20 + 2080] = 2120
3. NS1 は、インターフェイス 10/2 で応答パケットを受信します。
4. NS1 は、これらの IP パケットからすべての TCP セグメントを組み立てて、18000 バイトの HTTP 応答データを形成します。NS1 はこのレスポンスを処理します。
5. NS1 は、応答データを CL1 の MSS の倍数にセグメント化し、これらのセグメントを IP パケットとして、インターフェイス 10/1 から CL1 に送信します。これらの IP パケットは LBVS-1 の IP アドレスから発信され、CL1 の IP アドレスを宛先とします。

- 最後のパケットを除くすべてのパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (TCP ペイロード = CL1 の MSS サイズ)] = [20 + 20 + 1460] = 1500
- 最後のパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (残りの TCP セグメント)] = [20 + 20 + 480] = 520

構成タスク

次の表は、NetScaler アプライアンスで必要な構成を作成するためのタスク、NetScaler コマンド、および例を示しています。

タスク	CLI シンタックス	例
ジャンボフレームをサポートするために必要なインターフェイスの MTU を設定します。	set interface -mtu , show interface	set int 10/1 -mtu 1500 set int 10/2 -mtu 9000
VLAN を作成し、ジャンボフレームをサポートするために必要な VLAN の MTU を設定します	VLAN を追加 -mtu 、VLAN を表示	add vlan 10 -mtu 1500 add vlan 20 -mtu 9000
インターフェイスを VLAN にバインド	bind vlan -ifnum , show vlan	バインド VLAN 10-ifnum 10/1 バインド VLAN 20-ifnum 10/2
SNIP アドレスを追加する	add ns ip -type SNIP, show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP
HTTP サーバを表すサービスを作成する	add service HTTP , show service	add service SVC-S1 198.51.100.19 http 80, add service SVC-S2 198.51.100.20 http 80
HTTP 負荷分散仮想サーバーを作成し、サービスをバインドします。	add lb vserver HTTP , bind lb vserver , show lb vserver	lb vserver LVS-1 http 203.0.113.15 80、lb vserver LVS-1 SVC-S1 を追加、lb vserver LVS-1 SVC-S2 をバインド
カスタム TCP プロファイルを作成し、その MSS をジャンボフレームをサポートするように設定します。	add tcpProfile -mss , show tcpProfile	add tcpprofile NS1-SERVERS-JUMBO -mss 8960
カスタム TCP プロファイルを目的のサービスにバインドします。	set service -tcpProfileName , show service	サービス SVC-S1-TCP プロファイル名 NS1-SERVERS-JUMBO を設定、サービス SVC-S2-TCP プロファイル名 NS1-SERVERS-JUMBO を設定
構成を保存します	save ns config, show ns config	

ユースケース 3 –同じインターフェイスセットでのジャンボフローとジャンボ以外のフローの共存

August 15, 2023

負荷分散仮想サーバー LBVS-1 と LBVS-2 が NetScaler アプライアンス NS1 で構成されている例を考えてみましょう。LBVS-1 はサーバー S1 と S2 の HTTP トラフィックの負荷分散に使用され、LBVS-2 はサーバー S3 と S4 のトラフィックの負荷分散に使用されます。

CL1 は VLAN 10 に、S1 と S2 は VLAN 20 に、CL2 は VLAN 30 に、S3 と S4 は VLAN 40 上にあります。VLAN 10 と VLAN 20 はジャンボフレームをサポートし、VLAN 30 と VLAN 40 は通常のフレームのみをサポートします。

つまり、CL1 と NS1 間の接続、および NS1 とサーバ S1 または S2 の間の接続は、ジャンボフレームをサポートします。CL2 と NS1 間の接続、および NS1 とサーバー S3 または S4 間の接続は、通常のフレームのみをサポートします。

NS1 のインターフェイス 10/1 は、クライアントとの間でトラフィックを送受信します。NS1 のインターフェイス 10/2 は、サーバとの間でトラフィックを送受信します。

インターフェイス 10/1 はタグ付きインターフェイスとして VLAN 10 と VLAN 30 の両方にバインドされ、インターフェイス 10/2 はタグ付きインターフェイスとして VLAN 20 と VLAN 40 の両方にバインドされます。

ジャンボフレームをサポートするために、インターフェイス 10/1 および 10/2 の MTU は 9216 に設定されます。

NS1 では、ジャンボフレームをサポートするための VLAN 10 と VLAN 20 の MTU は 9000 に設定され、通常のフレームのみをサポートする VLAN30 と VLAN 40 では MTU がデフォルト値の 1500 に設定されています。

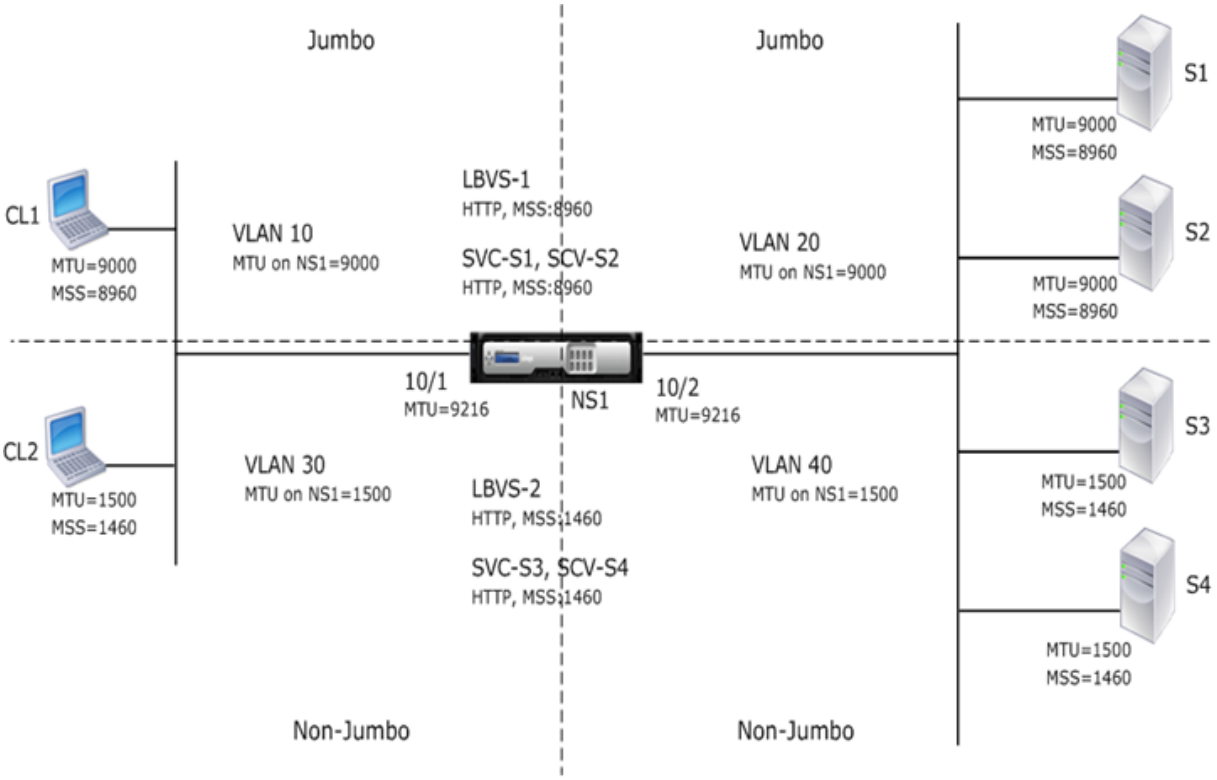
NetScaler インターフェイスでの VLAN タグ付きパケットの有効な MTU は、インターフェイスの MTU または VLAN の MTU のどちらか小さい方になります。次に例を示します:

- インターフェイス 10/1 の MTU は 9216 です。VLAN 10 の MTU は 9000 です。インターフェイス 10/1 では、VLAN 10 タグ付きパケットの MTU は 9000 です。
- インターフェイス 10/2 の MTU は 9216 です。VLAN 20 の MTU は 9000 です。インターフェイス 10/2 では、VLAN 20 タグ付きパケットの MTU は 9000 です。
- インターフェイス 10/1 の MTU は 9216 です。VLAN 30 の MTU は 1500 です。インターフェイス 10/1 では、VLAN 30 のタグ付きパケットの MTU は 1500 です。
- インターフェイス 10/2 の MTU は 9216 です。VLAN 40 の MTU は 1500 です。インターフェイス 10/2 では、VLAN 40 タグ付きパケットの MTU は 9000 です。

CL1、S1、S2、および CL1 と S1 または S2 の間にあるすべてのネットワークデバイスが、ジャンボフレーム用に設定されます。

HTTP トラフィックは TCP に基づいているため、ジャンボフレームをサポートするために、各エンドポイントで MSS が適切に設定されます。

- CL1 と NS1 の仮想サーバー LBVS-1 間の接続では、NS1 の MSS が TCP プロファイルで設定され、LBVS-1 にバインドされます。
- NS1 と S1 の SNIP アドレス間の接続では、NS1 上の MSS が TCP プロファイルに設定され、NS1 上の S1 を表すサービス (SVC-S1) にバインドされます。



次の表に、この例で使用される設定を示します。ジャンボフレームの使用例 3 の設定例。

次に、CL1 から S1 への要求のトラフィックフローを示します。

1. クライアント CL1 は、NS1 の仮想サーバ LBVS-1 に送信する 20000 バイトの HTTP 要求を作成します。
2. CL1 は NS1 の LBVS-1 への接続をオープンします。CL1 と NS1 は、接続の確立中に TCP MSS 値を交換します。
3. NS1 の MSS 値は HTTP 要求よりも小さいので、CL1 は要求データを NS1 の MSS の倍数に分割し、VLAN 10 としてタグ付けされた IP パケットでこれらのセグメントを NS1 に送信します。
   - 最初の 2 つのパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (TCP セグメント = NS1 MSS)] = [20 + 20 + 8960] = 9000
   - 最後のパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (残りの TCP セグメント)] = [20 + 20 + 2080] = 2120
4. NS1 は、これらのパケットをインタフェース 10/1 で受信します。NS1 はこれらのパケットを受け入れるのは、これらのパケットのサイズが VLAN 10 タグ付きパケットのインターフェイス 10/1 の実効 MTU（9000）以下であるためです。

5. NS1 は IP パケットからすべての TCP セグメントをアセンブルし、20000 バイトの HTTP 要求を形成します。NS1 はこの要求を処理します。
6. LBVS-1 の負荷分散アルゴリズムはサーバー S1 を選択し、NS1 はその SNIP アドレスの 1 つと S1 の間の接続を開きます。NS1 と CL1 は、接続の確立中にそれぞれの TCP MSS 値を交換します。
7. NS1 は、要求データを S1 の MSS の倍数に分割し、これらのセグメントを VLAN 20 とタグ付けされた IP パケットで S1 に送信します。

    - 最初の 2 つのパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (TCP ペイロード = S1 MSS)] = [20 + 20 + 8960] = 9000
    - 最後のパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (残りの TCP セグメント)] = [20 + 20 + 2080] = 2120

CL1 に対する S1 の応答のトラフィックフローは次のとおりです。

1. サーバ S1 は 30000 バイトの HTTP 応答を作成し、NS1 の SNIP アドレスに送信します。
2. S1 は応答データを NS1 の MSS の倍数に分割し、これらのセグメントを VLAN 20 としてタグ付けされた IP パケットで NS1 に送信します。これらの IP パケットは、S1 の IP アドレスから送信され、NS1 の SNIP アドレスを宛先とします。

    - 最初の 3 つのパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (TCP セグメント = NS1 の MSS サイズ)] = [20 + 20 + 8960] = 9000
    - 最後のパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (残りの TCP セグメント)] = [20 + 20 + 3120] = 3160

3. NS1 は、インターフェイス 10/2 で応答パケットを受信します。NS1 はこれらのパケットを受け入れます。これは、そのサイズが VLAN 20 タグ付きパケットのインターフェイス 10/2 の実効 MTU 値（9000）以下であるためです。
4. NS1 は、これらの IP パケットからすべての TCP セグメントを組み立てて、30000 バイトの HTTP 応答を形成します。NS1 はこのレスポンスを処理します。
5. NS1 は、応答データを CL1 の MSS の倍数にセグメント化し、これらのセグメントを VLAN 10 としてタグ付けされた IP パケットで、インターフェイス 10/1 から CL1 に送信します。これらの IP パケットは LBVS の IP アドレスから発信され、CL1 の IP アドレスを宛先とします。

    - 最初の 3 つのパケットのサイズ = [IP ヘッダー + TCP ヘッダー + [(TCP ペイロード = CL1 の MSS サイズ)] = [20 + 20 + 8960] = 9000
    - 最後のパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (残りの TCP セグメント)] = [20 + 20 + 3120] = 3160

構成タスク

次の表に、NetScaler ADC アプライアンスで必要な構成を作成するためのタスク、コマンド、および例を示します。ジャンボフレームのユースケース 3 構成タスク。

NetScaler ADC の Microsoft Direct Access 展開のサポート

August 15, 2023

Microsoft Direct Access は、リモートユーザーが別の VPN 接続を確立しなくても、企業の内部ネットワークにシームレスかつ安全に接続できるようにするテクノロジーです。接続の開閉にユーザーの操作を必要とする VPN 接続とは異なり、Direct Access 対応のクライアントは、クライアントがインターネットに接続するたびに企業の内部ネットワークに自動的に接続します。

Manage-Out は Microsoft Direct Access の機能で、企業ネットワーク内の管理者がネットワーク外の Direct Access クライアントに接続して管理 (たとえば、サービス更新のスケジュール設定などの管理タスクの実行、リモートサポートの提供など) を行うことができます。

Direct Access 環境では、NetScaler アプライアンスは高い可用性、スケーラビリティ、高パフォーマンス、およびセキュリティを提供します。NetScaler の負荷分散機能は、クライアントトラフィックを最適なサーバー経由で送信します。アプライアンスは、Manage-Out トラフィックを正しいパスで転送してクライアントに到達することもできます。

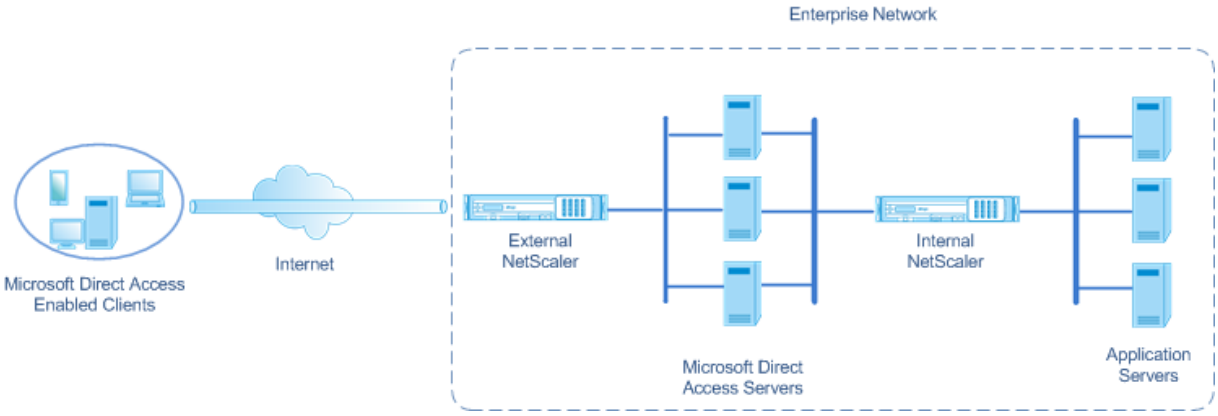
アーキテクチャ

Microsoft Direct Access 環境のアーキテクチャは、ダイレクトアクセス対応のクライアント、ダイレクトアクセスサーバー、アプリケーションサーバー、および内部および外部の NetScaler アプライアンスで構成されています。クライアントは、Direct Access サーバーを介してアプリケーションサーバーに接続します。外部の NetScaler アプライアンスは Direct Access サーバーへのクライアントトラフィックの負荷分散を行い、内部 NetScaler アプライアンスは Direct Access サーバーから宛先のアプリケーションサーバーにクライアントトラフィックを転送します。ダイレクトアクセスは、IPv4 ネットワークを介してクライアントの IPv6 トラフィックをトンネリングするために使用されます。外部 NetScaler アプライアンス上の IPv4 負荷分散仮想サーバーは、クライアントのトンネリングされたトラフィックをいずれかの Direct Access サーバーにロードバランシングします。ダイレクトアクセスサーバーは、受信したクライアントの IPv4 パケットから IPv6 パケットを抽出し、内部の NetScaler アプライアンスを介して宛先アプリケーションサーバーに送信します。内部 NetScaler アプライアンスには、Direct Access Server からのクライアントのトラフィックに関するレイヤー 2 およびレイヤー 3 の接続情報を保存するためのソースルートキャッシュオプションを有効にした転送セッションルールがあります。NetScaler アプライアンスは、ソースルートキャッシュテーブルと呼ばれるテーブルに次のレイヤー 2 およびレイヤー 3 の情報を保存します。

- 受信したパケットの送信元 IP アドレス
- パケットを送信したダイレクトアクセスサーバーの MAC アドレス
- パケットを受信した NetScaler アプライアンスの VLAN ID
- パケットを受信した NetScaler アプライアンスのインターフェイス ID

NetScaler アプライアンスは、クライアントに到達するためのトンネリング情報を持っているため、ソースルートキャッシュテーブルの情報を使用して同じ Direct Access サーバーに応答を転送します。また、内部アプライアンスは

ソースルートキャッシュテーブルを使用して、アプリケーションサーバーの管理アウトトラフィックを適切なダイレクトアクセスサーバーに転送し、特定のクライアントに到達します。



Microsoft ダイレクトアクセス環境における内部 **NetScaler** アプライアンスの構成

アプリケーションサーバーの応答と管理トラフィックを適切な Direct Access Gateway に転送するように内部 NetScaler アプライアンスを構成するには、転送セッションルールを構成します。各ルールで、ソースルートキャッシュパラメータを ENABLED に設定します。

CLI を使用して転送セッションルールを作成するには:

コマンドプロンプトで入力します。

- **add forwardingSession** <name> ((<network> [\]) | **-acl6name** \ | **-aclname** \) **-sourceroutecache** \(**ENABLED** | **DISABLED**]
- **show forwardingsession** <name>

設定例:

次の例では、転送セッションルール MS-DA-FW-1 が内部の NetScaler アプライアンスに作成されます。転送セッションでは、送信元 IPv6 プレフィクス 2001:DB8:: /96 と一致するダイレクトアクセスサーバからの着信 IPv6 パケットのレイヤ 2 およびレイヤ 3 情報が格納されます。

```
> add forwardingSession MS-DA-FW-1 2001:DB8::/96 -sourceroutecache -
    ENABLED
 Done
```

ソースルートキャッシュテーブルの表示

ソースルートキャッシュテーブルを表示して、ダイレクトアクセスサーバーとアプリケーションサーバー間の不要な接続を監視または検出できます。

CLI を使用してソースルートキャッシュテーブルを表示するには:

コマンドプロンプトで入力します。

- ソースルートキャッシュテーブルを表示

例**:**

```
> show sourceroutecachetable
SOURCEIP            MAC                 VLAN    INTERFACE
2001:DB8:5001:10    56:53:24:3d:02:eb    30        1/2
2001:DB8:5003:30    60:54:35:3e:04:bd    60        1/3
Done
```

ソースルートキャッシュテーブルのクリア

NetScaler アプライアンスのソースルートキャッシュテーブルからすべてのエントリを消去できます。

CLI を使用してソースルートキャッシュテーブルをクリアするには:

コマンドプロンプトで入力します。

- **NS** ソースルートキャッシュテーブルをフラッシュ

アクセス制御リスト

August 15, 2023

アクセス制御リスト (ACL) は IP トラフィックをフィルタリングし、ネットワークを不正アクセスから保護します。ACL は、NetScaler が評価してアクセスを許可するかどうかを決定する一連の条件です。たとえば、財務部門はおそらく、人事や文書などの他の部門が自分のリソースにアクセスすることを許可したくないと考えており、それらの部門はデータへのアクセスを制限したいと考えています。

NetScaler はデータパケットを受信すると、データパケット内の情報を ACL で指定された条件と比較し、アクセスを許可または拒否します。組織の管理者は、ACL を次の処理モードで機能するように設定できます。

- 許可-パケットを処理します。
- ブリッジ：パケットを処理せずに宛先にブリッジします。パケットは、レイヤ 2 とレイヤ 3 の転送によって直接送信されます。
- 拒否：パケットをドロップします。

ACL ルールは NetScaler の第 1 レベルの防御です。

NetScaler ADC では、次のタイプの ACL がサポートされています。

- シンプルな **ACL** は、送信元 IP アドレス、およびオプションでプロトコル、宛先ポート、またはトラフィックドメインに基づいてパケットをフィルタリングします。ACL で指定された特性を持つパケットはすべてドロップされます。

- 拡張 **ACL** は、送信元 IP アドレス、送信元ポート、アクション、プロトコルなどのさまざまなパラメータに基づいてデータパケットをフィルタリングします。拡張 ACL は、NetScaler がパケットを処理したり、パケットをブリッジしたり、パケットをドロップしたりするためにパケットが満たす必要がある条件を定義します。

命名法

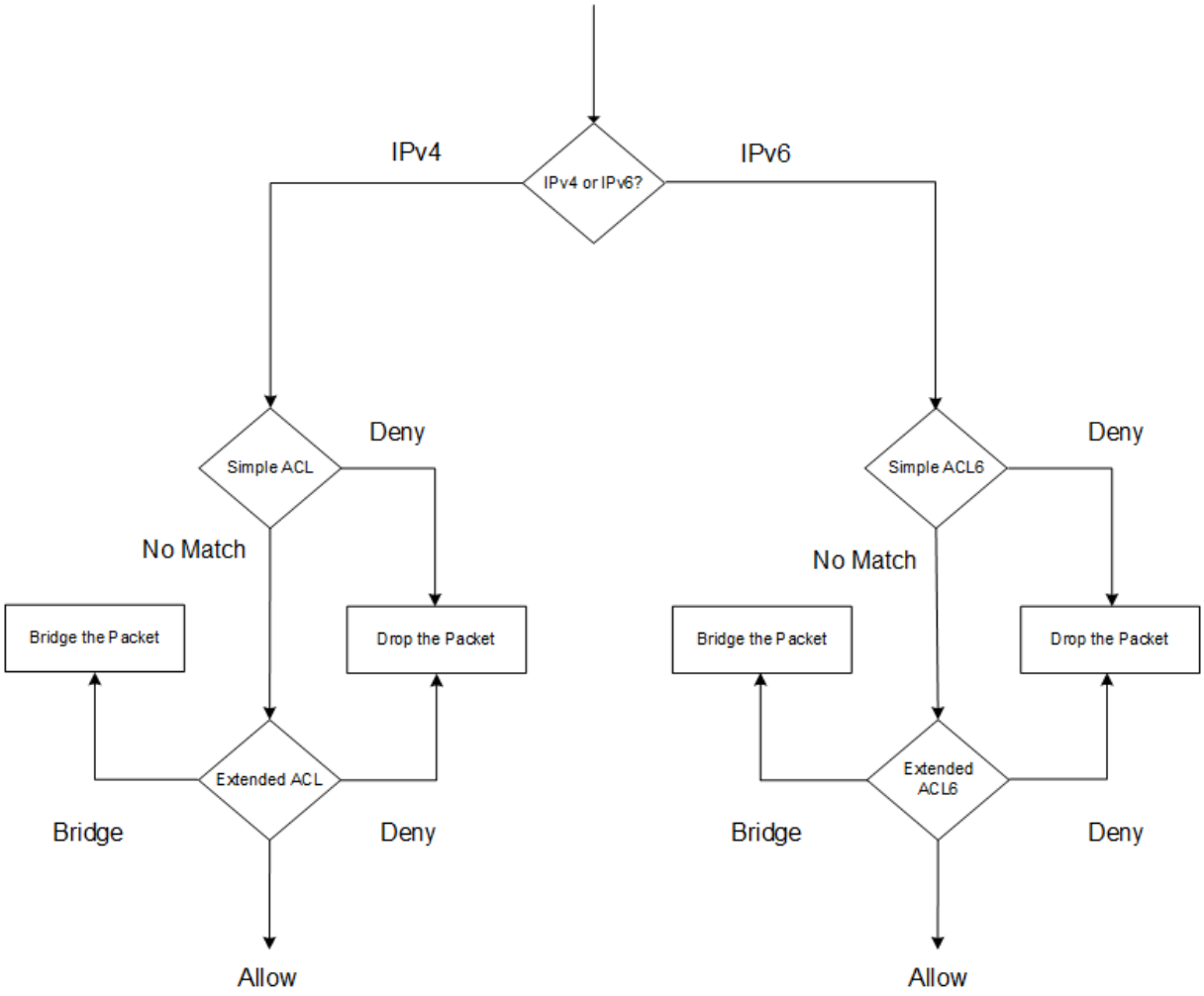
NetScaler ユーザーインターフェイスでは、シンプル ACL と拡張 ACL という用語は、IPv4 パケットを処理する ACL を指します。IPv6 パケットを処理する ACL は、単純 ACL6 または拡張 ACL6 と呼ばれます。両方のタイプについて説明する際、このドキュメントでは両方を単純 ACL または拡張 ACL と呼ぶことがあります。

ACL 優先順位

シンプル ACL と拡張 ACL の両方が設定されている場合、着信パケットは最初にシンプル ACL と比較されます。

NetScaler はまず、受信パケットが IPv4 パケットか IPv6 パケットかを判断し、次にパケットの特性を単純 ACL または単純 ACL6 と比較します。一致するものが見つかると、パケットはドロップされます。一致するものが見つからない場合、パケットは拡張 ACL または拡張 ACL6 と比較されます。その比較の結果が一致した場合、パケットは ACL で指定されたとおりに処理されます。パケットはブリッジ、ドロップ、または許可できます。一致するものが見つからない場合、パケットは許可されます。

図 1: 簡易 ACL および拡張 ACL フローシーケンス



シンプル **ACL** とシンプル **ACL6**

August 15, 2023

単純な ACL または単純な ACL6 では、パラメータはほとんど使用されないため、IP パケットをドロップするようにだけ設定できます。パケットは、送信元 IP アドレス、およびオプションでプロトコル、宛先ポート、またはトラフィックドメインに基づいてドロップできます。

単純な ACL または単純な ACL6 を作成する場合、存続可能時間（TTL）を秒単位で指定できます。この時間が経過すると、ACL が期限切れになります。TTL が設定された ACL は、設定を保存しても保存されません。単純な ACL と単純な ACL6 を表示して設定を確認したり、それらの統計情報を表示したりできます。

シンプル **ACL** とシンプル **ACL6** の設定

NetScaler でシンプルな ACL またはシンプルな ACL6 を構成するには、次のタスクが含まれる場合があります。

- 単純な **ACL** または単純な **ACL6** を作成します。送信元 IP アドレス、およびオプションでプロトコル、宛先ポート、またはトラフィックドメインに基づいてパケットをドロップ（拒否）するシンプル ACL またはシンプルな ACL 6 を作成します。
- 単純な **ACL** または単純な **ACL6** を削除します。これらの ACL は、一度作成すると変更できません。単純な ACL または単純な ACL6 を変更する必要がある場合は、それを削除して作成する必要があります。

CLI のプロシージャ

CLI を使用して簡単な ACL を作成するには:

コマンドプロンプトで入力します。

```
1 -  ns simpleacl <aclname> DENY -srcIP <ip_addr> [-destPort <port> -
     protocol ( TCP | UDP )] [-TTL <positive_integer>]
2 -  show ns simpleacl [<aclname>]
3 <!--NeedCopy-->
```

例**:**

```
1 > add simpleacl rule1 DENY -srcIP 10.102.29.5 -TTL 600
2 Done
3 <!--NeedCopy-->
```

CLI を使用して簡単な ACL6 を作成するには:

コマンドプロンプトで入力します。

```
1 -  add ns simpleacl6 <aclname> DENY - srcIPv6 <ipv6_addr|null> [-
     destPort <port> -protocol ( TCP | UDP )] [-TTL <positive_integer>]
2 -  show ns simpleacl6 [<aclname>]
3 <!--NeedCopy-->
```

例**:**

```
1 >  add ns simpleacl6 rule1 DENY  - srcIPv6 3ffe:192:168:215::82 -
     destPort 80 -Protocol TCP -TTL 9000
2  Done
3 <!--NeedCopy-->
```

CLI を使用して単純な ACL を 1 つ削除するには:

コマンドプロンプトで入力します。

- **rm ns simpleacl** <aclname>
- **show ns simpleacl**

CLI を使用して単純な ACL6 を 1 つ削除するには:

コマンドプロンプトで入力します。

- **rm ns simpleacl6**<aclname>
- **show ns simpleacl6**

CLI を使用して単純な ACL をすべて削除するには:

コマンドプロンプトで入力します。

- **clear ns simpleacl**
- **show ns simpleacl**

CLI を使用して単純な ACL6 をすべて削除するには:

コマンドプロンプトで入力します。

- **clear ns simpleacl6**
- **show ns simpleacl6**

GUI のプロシージャ

GUI を使用して簡単な ACL を作成するには:

[システム**] > [**ネットワーク**] > [ACL**] に移動し、[シンプル **ACL**] タブで、新しいシンプル ACL を追加します。

GUI を使用して簡単な ACL6 を作成するには:

[システム**] > [**ネットワーク**] > [ACL**] に移動し、[シンプル **ACL6**] タブで、新しいシンプル ACL6 を追加します。

GUI を使用して単純な ACL を 1 つ削除するには:

[システム**] > [**ネットワーク**] > [ACL**] に移動し、[シンプル **ACL**] タブでシンプル ACL を削除します。

GUI を使用して単純な ACL6 を 1 つ削除するには:

[システム**] > [**ネットワーク**] > [ACL**] に移動し、[シンプル **ACL6**] タブで、単純な ACL6 を削除します。

GUI を使用して単純な ACL をすべて削除するには:

1. [システム**] > [**ネットワーク**] > [ACL**] に移動します。
2.「シンプル **ACL** 」タブの「アクション」リストで、「クリア」をクリックします。

GUI を使用して単純な ACL6 をすべて削除するには:

1. [システム**] > [**ネットワーク**] > [ACL**] に移動します。
2.「Simple ACL6s」タブの「アクション」リストで、「クリア」をクリックします。

簡易 **ACL** および簡易 **ACL6** 統計情報の表示

単純な ACL（または簡易 ACL6）統計情報を表示できます。これには、一致の数、ミス数、および設定された単純 ACL の数が含まれます。

次の表に、シンプル ACL およびシンプル ACL 6 について表示できる統計情報を示します。

統計	内容
ACL マッチ	ACL に一致するパケット
ACL ミス	どの ACL とも一致しないパケット
ACL count	設定されている ACL の数

CLI のプロシージャ

CLI を使用して簡単な ACL 統計を表示するには:

コマンドプロンプトで入力します。

- **stat ns simpleacl**

例**:**

```
1  > stat ns simpleacl
2
3  SimpleACL Statistics
4
5                                                Rate (/s)
                                                    Total
6  SimpleACL hits                                       0
                         0
7  SimpleACL misses                                     0
      51872
8  SimpleACLs count                                    --
                         2
9  Done
10 <!--NeedCopy-->
```

CLI を使用して簡単な ACL6 統計情報を表示するには:

コマンドプロンプトで入力します。

- **stat ns simpleacl6**

GUI のプロシージャ

GUI を使用して簡単な ACL 統計情報を表示するには:

[** システム] > [ネットワーク] > [**ACL**] に移動し、[シンプル **ACL**] タブで ACL を選択し、[統計] をクリックします。**

GUI を使用して簡単な ACL6 統計情報を表示するには:

[システム**] > [**ネットワーク**] > [ACL**] に移動し、[シンプル **ACL6**] タブで単純な ACL6 を選択し、[統計] をクリックします。

確立された接続の終了

シンプルな ACL またはシンプルな ACL6 の場合、NetScaler は ACL で指定された条件に一致する新しい接続をすべてブロックします。ACL が作成される前に確立された既存の接続に関連するパケットはブロックされません。既存の ACL と一致する以前に確立された接続を終了するには、CLI または GUI からフラッシュ操作を実行できます。

フラッシュは次のような場合に便利です。

- ブラックリストに登録された IP アドレスのリストを受け取り、それらの IP アドレスによる NetScaler へのアクセスを完全にブロックしたいと考えています。この場合、単純な ACL または単純な ACL6 を作成して、これらの IP アドレスからの新しい接続をブロックし、それらのアドレスに関連付けられた既存の接続をフラッシュします。
- 特定のネットワークからの多数の接続を 1 つずつ終端する時間をとらずに終端したい。

はじめに

- フラッシュを実行すると、NetScaler ADC は確立されているすべての接続を検索し、ADC で構成された単純な ACL のいずれかに指定された条件に一致する接続を終了します。
- 複数の単純な ACL を作成し、それらのいずれかに一致する既存の接続をフラッシュする場合は、まずすべての単純な ACL を作成し、フラッシュを 1 回だけ実行することで、パフォーマンスへの影響を最小限に抑えることができます。

CLI のプロシージャ

CLI を使用して、設定したシンプルな ACL のいずれかに一致する確立済みの IPv4 接続をすべて終了するには:

コマンドプロンプトで入力します。

- シンプル **ACL-EST** セッションをフラッシュ

CLI を使用して、設定したシンプルな ACL6 のいずれかに一致する確立済みの IPv6 接続をすべて終了するには:

コマンドプロンプトで入力します。

- シンプル **ACL 6-EST** セッションをフラッシュ

GUI のプロシージャ

設定したシンプル ACL のいずれかに一致する確立された IPv4 接続を GUI を使用してすべて終了するには:

1. [システム**] > [**ネットワーク**] > [ACL**] に移動します。
2.「シンプル **ACL** 」タブの「アクション」リストで、「フラッシュ」をクリックします。

設定したシンプル ACL6 のいずれかに一致する確立済みの IPv6 接続をすべて GUI を使用して終了するには:

1. [システム**] > [**ネットワーク**] > [ACL**] に移動します。
2.「シンプル ACL6s」タブの「アクション」リストで、「フラッシュ」をクリックします。

拡張 **ACL** と拡張 **ACL6**

August 15, 2023

拡張 ACL および拡張 ACL6 は、単純な ACL では使用できないパラメータとアクションを提供します。送信元 IP アドレス、送信元ポート、アクション、プロトコルなどのパラメータに基づいてデータをフィルタリングできます。パケットを許可する、パケットを拒否する、またはパケットをブリッジするタスクを指定できます。

拡張 ACL および ACL 6 は、作成後に変更できます。また、プライオリティを再番号付けして、評価の順序を指定できます。

注**:** シンプル ACL と拡張 ACL の両方を設定する場合、シンプル ACL は拡張 ACL よりも優先されます。

拡張 ACL および ACL 6 に対して次のアクションを実行できます。変更、適用、無効化、有効、削除、および番号付け(優先度)。拡張 ACL と ACL 6 を表示して設定を確認し、統計情報を表示できます。

拡張 ACL に一致するパケットの詳細を記録するように NetScaler ADC を構成できます。

拡張 **ACL** と拡張 **ACL6** の適用: 単純な ACL および ACL6 とは異なり、NetScaler で作成された拡張 ACL および ACL6 は、適用されるまで機能しません。また、拡張 ACL または ACL の無効化、優先度の変更、ACL の削除など、拡張 ACL または ACL 6 に変更を加えた場合は、拡張 ACL または ACL 6 を再適用する必要があります。ログを有効にした後に再適用する必要があります。拡張 ACL または ACL6 を適用する手順は、これらすべてを再適用します。たとえば、拡張 ACL ルール 1 ～10 を適用し、次にルール 11 を作成して適用すると、最初の 10 個のルールが新たに適用されます。

セッションに関連した DENY ACL がある場合、ACL を適用すると、そのセッションが終了します。

拡張 ACL と ACL 6 はデフォルトで有効になっています。それらが適用されると、NetScaler ADC は受信パケットと受信パケットを比較し始めます。ただし、無効にした場合は、再適用された場合でも、再度有効にするまでは使用されません。

拡張 **ACL** および拡張 **ACL 6** のプライオリティの再番号付け: プライオリティ番号によって、拡張 ACL または ACL 6 がパケットに対して照合される順序が決まります。プライオリティ番号が低い ACL のプライオリティが高くなりま

す。これは、プライオリティ番号が高い（優先度が低い）ACL の前に評価され、パケットに一致する最初の ACL によって、パケットに適用されるアクションが決まります。

拡張 ACL または ACL6 を作成すると、特に指定しない限り、NetScaler ADC は 10 の倍数の優先度番号を自動的に割り当てます。たとえば、2 つの拡張 ACL のプライオリティがそれぞれ 20 と 30 で、3 番目の ACL にそれらの数値の間に値を持たせたい場合は、値 25 を割り当てます。ACL の評価順序を後で保持し、番号を 10 の倍数に復元する場合は、再番号付け手順を使用できます。

拡張 **ACL** および拡張 **ACL 6** の設定

NetScaler ADC で拡張 ACL または ACL6 を構成するには、次のタスクがあります。

- 拡張 **ACL** または **ACL6** を作成します。拡張 ACL または ACL6 を作成して、パケットを許可、拒否、ブリッジします。パケットの送信元または宛先 IP アドレスと照合する IP アドレスまたは IP アドレスの範囲を指定できます。着信パケットのプロトコルと照合するプロトコルを指定できます。
- （任意）拡張 **ACL** または **ACL6** を変更します。以前に作成した拡張 ACL または ACL 6 を変更できます。または、一時的に使用を中止する場合は、無効にして後で再度有効にすることができます。
- 拡張 **ACL** または **ACL 6** を適用します。拡張 ACL または ACL6 を作成、変更、無効化、または再有効化、または削除した後、拡張 ACL または ACL 6 を適用してアクティブ化する必要があります。
- （任意）拡張 **ACL** または **ACL 6** のプライオリティを再番号付けします。10 の倍数ではないプライオリティで ACL を設定していて、番号付けを 10 の倍数に戻す場合は、再番号付け手順を使用します。

CLI のプロシージャ

CLI を使用して拡張 **ACL** を作成するには:

コマンドプロンプトで入力します。

- **add ns acl** <aclname> <aclaction> [-**srcIP** [\] \] \[-**srcPort** \[\\] \] \[-**destIP** \[\\] \] \[-**destPort** \[\\] \] \[-**TTL** \\<positive_integer>] \[-**srcMac** \\<mac_addr>] \[\(-**protocol** \\ \[-established]) | -**protocolNumber** \<positive_integer>] \[-**vlan** \\<positive_integer>] \[-**interface** \\<interface_name>] \[-**icmpType** \\<positive_integer> \[-**icmpCode** \\<positive_integer>]] \[-**priority** \\<positive_integer>] \[-**state** \(ENABLED | DISABLED)] \[-**logstate** \(ENABLED | DISABLED) \[-**ratelimit** \\<positive_integer>]]
- **show ns acl** [\<aclName>]

CLI を使用して拡張 **ACL6** を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns acl6** <acl6name> <acl6action> [-**srcIPv6** [\] \] \[-**srcPort** \[\\] \] \[-**destIPv6** \[\\] \] \[-**destPort** \[\\] \] \[-**TTL** \\<positive_integer>] \[-**srcMac** \\<mac_addr>] \[\(-**protocol** \\ \[-established]) | -**protocolNumber** \<positive_integer>] \[-**vlan** \\<positive_integer>] \[-**interface** \\<interface_name>] \[-**icmpType** \\<positive_integer> \[-**icmpCode** \\<positive_integer>]] \[-**priority** \\<positive_integer>] \[-**state** \(ENABLED | DISABLED)]
- **show ns acl6** [\<aclName>]

CLI を使用して拡張 **ACL** を変更するには、次の手順を実行します。

拡張 ACL を変更するには、**set ns acl** コマンド、拡張 ACL の名前、および変更するパラメータを新しい値で入力します。

CLI を使用して拡張 **ACL6** を変更するには、次の手順を実行します。

拡張 ACL6 を変更するには、**set ns acl6** コマンド、拡張 ACL6 の名前、および変更するパラメータを新しい値で入力します。

CLI を使用して拡張 **ACL** を無効または有効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- **disable ns acl** <aclname>
- **enable ns acl** <aclname>

CLI を使用して拡張 **ACL6** を無効または有効にするには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- **disable ns acl6** <aclname>
- **enable ns acl6** <aclname>

CLI を使用して拡張 **ACL** を適用するには:

コマンドプロンプトで入力します。

- **apply ns acls**

CLI を使用して拡張 **ACL6** を適用するには:

コマンドプロンプトで入力します。

- **apply ns acls6**

CLI を使用して拡張 **ACL** の優先順位を変更するには:

コマンドプロンプトで入力します。

- **renumber ns acls**

CLI を使用して拡張 **ACL6** の優先順位を変更するには:

コマンドプロンプトで入力します。

- **renumber ns acls6**

GUI のプロシージャ

GUI を使用して拡張 **ACL** を設定するには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL**] タブで、新しい拡張 ACL を追加するか、既存の拡張 ACL を編集します。既存の拡張 ACL を有効または無効にするには、その拡張 ACL を選択し、[アクション] リストから [有効] または [無効] を選択します。

GUI を使用して拡張 **ACL6** を設定するには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL6**] タブで、新しい拡張 ACL6 を追加するか、既存の拡張 ACL6 を編集します。既存の拡張 ACL6 を有効または無効にするには、それを選択し、[操作] リストから [有効] または [無効] を選択します。

GUI を使用して拡張 **ACL** を適用するには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL**] タブの [アクション] リストで、[適用] をクリックします。

GUI を使用して拡張 **ACL6** を適用するには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL 6**] タブの [アクション] リストで、[適用] をクリックします。

GUI を使用して拡張 **ACL** のプライオリティを再番号付けするには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL**] タブの [アクション] リストで、[優先度の再番号付け] をクリックします。

GUI を使用して拡張 **ACL6** の優先順位を再番号付けするには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL 6**] タブの [アクション] リストで、[優先度の再番号付け] をクリックします。

設定例

次の表に、コマンドラインインターフェイスを介して拡張 ACL ルールを設定する例を示します。ACL の設定例。

拡張 **ACL** のロギング

拡張 ACL に一致するパケットの詳細を記録するように NetScaler ADC を構成できます。

ACL 名に加えて、ログに記録される詳細には、送信元および宛先 IP アドレスなどのパケット固有の情報が含まれます。この情報は、有効になっているグローバルロギング（syslog or nslog）のタイプに応じて、syslog ファイルまたはnslogファイルに保存されます。

ロギングは、グローバルレベルと ACL レベルの両方で有効にする必要があります。グローバル設定が優先されます。

ロギングを最適化するために、同じフローからの複数のパケットが ACL に一致すると、最初のパケットの詳細だけがログに記録され、同じフローに属するすべてのパケットに対してカウンタが増加します。フローは、送信元 IP アドレス、宛先 IP アドレス、送信元ポート、宛先ポート、およびプロトコルパラメータに対して同じ値を持つパケットのセットとして定義されます。ログメッセージのフラッディングを避けるために、NetScaler ADC は、同じフローに属するパケットが繰り返しログに記録されないように内部レート制限を実行します。任意の時点でログに記録できる異なるフローの総数は 10,000 に制限されています。

注**:** ロギングを有効にした後に ACL を適用する必要があります。

CLI のプロシージャ

CLI を使用して拡張 **ACL** ロギングを設定するには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、ログを構成し、構成を確認します。

- **set ns acl** <aclName> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** \<positive_integer>]
- **apply acls**
- **show ns acl** [\<aclName>]

GUI のプロシージャ

GUI を使用して拡張 **ACL** ロギングを設定するには、次の手順を実行します。

1. [システム**] > [**ネットワーク**] > [ACL**] に移動し、[拡張 **ACL**] タブで拡張 ACL を開きます。
2. 次のパラメーターを設定します。
    - [**Log State**]：拡張 ACL 規則に関連するイベントのロギングを有効または無効にします。ログメッセージは、構成されたsyslog or auditlogサーバに保存されます。
    - [**Log Rate Limit**]：1 秒間に生成されるログメッセージの最大数。このパラメータを設定する場合は、[Log State] パラメーターを有効にする必要があります。

設定例

```
> set ns acl restrict -logstate ENABLED -ratelimit 120
Warning: ACL modified, apply ACLs to activate change

> apply ns acls
Done
<!--NeedCopy-->
```

拡張 **ACL6** のロギング

拡張 ACL6 ルールに一致するパケットの詳細を記録するように NetScaler ADC アプライアンスを構成できます。ACL6 名に加えて、ログに記録される詳細には、送信元および宛先 IP アドレスなどのパケット固有の情報が含まれます。この情報は、NetScaler ADC アプライアンスで構成したログの種類（`syslog or nslog`）に応じて、syslog または`nslog`ファイルに保存されます。

ロギングを最適化するために、同じフローからの複数のパケットが ACL6 に一致する場合、最初のパケットの詳細のみがロギングされます。カウンタは、同じフローに属する他のすべてのパケットに対して増分されます。フローは、次のパラメータに対して同じ値を持つパケットのセットとして定義されます。

- 接続元 IP
- 接続先 IP
- 送信元ポート
- Destination port
- プロトコル (TCP または UDP)

着信パケットが同じフローからのものでない場合、新しいフローが作成されます。任意の時点でログに記録できる異なるフローの総数は 10,000 に制限されています。

CLI のプロシージャ

CLI を使用して拡張 **ACl6** ルールのロギングを設定するには、次の手順を実行します。

- 拡張 ACL6 ルールの追加時にログを構成するには、コマンドプロンプトで次のように入力します。
  - **add acl6** <acl6Name> <acl6action> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** \<positive_integer>]
  - **apply acls6**
  - **show acl6** [\<acl6Name>]
- 既存の拡張 ACL6 ルールのログを構成するには、コマンドプロンプトで次のように入力します。
  - **set acl6** <acl6Name> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** \<positive_integer>]

- **show acl6** [\<acl6Name>]
- **apply acls6**

GUI のプロシージャ

GUI を使用して拡張 **ACL6** ロギングを設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL 6**] タブをクリックします。
2. 既存の拡張 ACL6 ルールの追加または変更時に、次のパラメータを設定します。
   - ログ状態: 拡張 ACL6s ルールに関連するイベントのロギングを有効または無効にします。ログメッセージは、設定された syslog または`auditlog`サーバに保存されます。
   - [**Log Rate Limit**]: 1 秒間に生成されるログメッセージの最大数。このパラメータを設定する場合は、[**Log State**] パラメーターを有効にする必要があります。

設定例

```
> set acl6 ACL6-1 -logstate ENABLED -ratelimit 120
Done

> apply acls6
Done
<!--NeedCopy-->
```

拡張 **ACL** および拡張 **ACL 6** の統計情報の表示

拡張 ACL および ACL 6 の統計情報を表示できます。

次の表に、拡張 ACL および ACL 6 に関連する統計情報とその説明を示します。

統計情報	指定
ACL の一致を許可	処理モードが ALLOW に設定された ACL と一致するパケット。NetScaler ADC はこれらのパケットを処理します。
NAT ACL が一致する	NAT ACL に一致するパケット。その結果、NAT セッションが発生します。
ACL の一致を拒否する	処理モードが DENY に設定された ACL と一致するため、ドロップされたパケット。
ブリッジ ACL マッチ	ブリッジ ACL に一致するパケット。トランスペアレントモードでは、サービス処理をバイパスします。
ACL マッチ	ACL に一致するパケット。

統計情報	指定
ACL ミス	どの ACL とも一致しないパケット。
ACL カウント	ユーザによって設定された ACL ルールの総数。
有効な ACL カウント	内部で設定された有効な ACL の総数。IP アドレスの範囲を持つ拡張 ACL の場合、NetScaler ADC アプライアンスは各 IP アドレスに対して内部的に拡張 ACL を作成します。たとえば、1000 の IPv4 アドレス（範囲またはデータセット）を持つ拡張 ACL の場合、NetScaler ADC は内部的に 1000 個の拡張 ACL を作成します。

CLI のプロシージャ

CLI を使用してすべての拡張 **ACL** の統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **stat ns acl**

CLI を使用してすべての拡張 **ACL6** の統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **stat ns acl6**

GUI のプロシージャ

GUI を使用して拡張 **ACL** の統計情報を表示するには、次の手順を実行します。

- [システム**] > [**ネットワーク**] > [ACL**] に移動し、[拡張 **ACL**] タブで拡張 ACL を選択し、[統計情報] をクリックします。

GUI を使用して拡張 **ACL6** の統計情報を表示するには、次の手順を実行します。

- [システム**] > [**ネットワーク**] > [ACL**] に移動し、[拡張 **ACL 6**] タブで拡張 ACL を選択し、[統計情報] をクリックします。

ステートフル **ACL**

ステートフル ACL ルールは、要求がルールに一致したときにセッションを作成し、これらの応答が NetScaler ADC アプライアンスの拒否 ACL ルールと一致していても結果の応答を許可します。ステートフル ACL は、これらの特定の応答を許可するために、より多くの ACL ルール/転送セッションルールを作成する作業をオフロードします。

ステートフル ACL は、次の要件を持つ NetScaler ADC アプライアンスのエッジファイアウォール展開に最適です。

- NetScaler ADC アプライアンスは、内部クライアントから開始された要求とインターネットからの関連する応答を許可する必要があります。
- アプライアンスは、クライアント接続に関連しないパケットをインターネットからドロップする必要があります。

はじめに

ステートフル ACL 規則を設定する前に、次の点に注意してください。

- NetScaler ADC アプライアンスは、ステートフル ACL ルールとステートフル ACL6 ルールをサポートしています。
- 高可用性セットアップでは、ステートフル ACL ルールのセッションは 2 次ノードに同期されません。
- ルールが NetScaler ADC NAT 構成にバインドされている場合、ACL ルールをステートフルとして構成することはできません。NetScaler NAT 構成の例を次に示します。
  - RNAT
  - 大規模 NAT（ラージスケール NAT44、DS-Lite、large scale NAT64）
  - NAT64
  - 転送セッション
- この ACL ルールに TTL パラメータと Established パラメータが設定されている場合、ACL ルールをステートフルとして設定することはできません。
- ステートフル ACL ルール用に作成されたセッションは、次の ACL 操作に関係なく、タイムアウトするまで存続します。
  - ACL の削除
  - ACL を無効にする
  - ACL をクリアする
- ステートフル ACL は、次のプロトコルではサポートされていません。
  - アクティブ FTP
  - TFTP

ステートフル **IPv4 ACL** ルールの設定

ステートフル ACL ルールの設定は、ACL ルールのステートフルパラメータを有効にすることで構成されます。

CLI を使用して **ACL** ルールのステートフルパラメータを有効にするには、次の手順を実行します。

- ACL ルールの追加中にステートフルパラメータを有効にするには、コマンドプロンプトで次のように入力します。
  - **add acl** <lname> ALLOW -**stateful** (ENABLED | DISABLED)
  - **apply acls**
  - **show acl** <name>
- 既存の ACL ルールのステートフルパラメータを有効にするには、コマンドプロンプトで次のように入力します。
  - **set acl** <name> -**stateful** (ENABLED | DISABLED)
  - **apply acls**
  - **show acl** <name>

GUI を使用して **ACL** ルールのステートフルパラメータを有効にするには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL**] タブをクリックします。
2. 既存の ACL ルールの追加または変更中に **Stateful** パラメータを有効にします。

設定例

```
> add acl ACL-1 allow -srciP 1.1.1.1 -stateful Yes

Done

> apply acls

Done

> show acl

1)         Name: ACL-1

    Action: ALLOW                          Hits: 0

    srcIP = 1.1.1.1

    destIP

    srcMac:

    Protocol:

    Vlan:                                 Interface:

    Active Status: ENABLED                 Applied Status: NOTAPPLIED

    Priority: 10                           NAT: NO
```



```
28
29      TTL:
30
31      Log Status: DISABLED
32
33      Forward Session: NO
34
35      Stateful: YES
36  <!--NeedCopy-->
```

ステートフル **ACL6** ルールの設定

ステートフル ACL6 ルールの設定は、ACL6 ルールのステートフルパラメータを有効にすることで構成されます。

CLI を使用して **ACL6** ルールのステートフルパラメータを有効にするには、次の手順を実行します。

- ACL6 ルールの追加中にステートフルパラメータを有効にするには、コマンドプロンプトで次のように入力します。
  - **add acl6** <name> ALLOW -stateful (ENABLED | DISABLD)
  - **apply acls6**
  - **show acl6** <name>
- 既存の ACL6 ルールのステートフルパラメータを有効にするには、コマンドプロンプトで次のように入力します。
  - **set acl6** <name> -**stateful** (ENABLED | DISABLED)
  - **apply acls6**
  - **show acl6** <name>

GUI を使用して **ACL6** ルールのステートフルパラメータを有効にするには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL6**] タブをクリックします。
2. 既存の ACL6 ルールの追加または変更中に **Stateful** パラメータを有効にします。

設定例

```
1  >  add acl6 ACL6-1 allow -srcipv6 1000::1  - stateful Yes
2
3  Done
4
5  >  apply acls6
6
7  Done
8
9  > show acl6
10
```



```
1)    Name: ACL6-1

    Action: ALLOW                          Hits: 0

    srcIPv6 = 1000::1

    destIPv6

    srcMac:

    Protocol:

    Vlan:                                 Interface:

    Active Status: ENABLED                 Applied Status: NOTAPPLIED

    Priority: 10                           NAT: NO

    TTL:

    Forward Session: NO

    Stateful: YES
<!--NeedCopy-->
```

データセットベースの拡張 **ACL**

企業では多くの ACL が必要です。多くの ACL の設定と管理は、頻繁に変更が必要な場合、困難で面倒です。

NetScaler ADC アプライアンスは、拡張 ACL のデータセットをサポートします。データセットは、NetScaler ADC アプライアンスの既存の機能です。データセットは、数値（整数）、IPv4 アドレス、または IPv6 アドレスのインデックス付きパターンの配列です。

拡張 ACL でのデータセットのサポートは、共通の ACL パラメータを必要とする複数の ACL ルールを作成する場合に役立ちます。

ACL ルールの作成時に、共通パラメータを指定する代わりに、これらの共通パラメータを含むデータセットを指定できます。

データセットに加えられた変更は、このデータセットを使用している ACL ルールに自動的に反映されます。データセットを持つ ACL は、構成と管理が簡単です。また、従来の ACL よりも小さく、読みやすくなっています。

現在、NetScaler ADC アプライアンスは、拡張 ACL に対して次のタイプのデータセットのみをサポートしています。

- IPv4 アドレス（ACL ルールの送信元 IP アドレス、宛先 IP アドレス、またはその両方を指定する場合）
- 番号（ACL ルールの送信元ポート、宛先ポート、またはその両方を指定する場合）

はじめに

データセットベースの拡張 ACL ルールを構成する前に、次の点に注意してください。

- NetScaler ADC アプライアンスのデータセット機能に精通していることを確認してください。データセットの詳細については、「パターンセットとデータセット」を参照してください。
- NetScaler ADC アプライアンスは、IPv4 拡張 ACL のデータセットのみをサポートします。
- NetScaler ADC アプライアンスは、拡張 ACL に対して次のタイプのデータセットのみをサポートします。
  - IPv4 アドレス
  - 番号
- NetScaler ADC アプライアンスは、すべての NetScaler ADC セットアップ（スタンドアロン、高可用性、クラスター）でデータセットベースの拡張 ACL をサポートします。
- 範囲を含むデータセットを含む拡張 ACL の場合、NetScaler ADC アプライアンスはデータセット値の組み合わせごとに内部的に拡張 ACL を作成します。
  - 例 **1**: データセットにバインドされた 1000 の IPv4 アドレスを持つ IPv4 データセットベースの拡張 ACL で、データセットがソース IP パラメータに設定されている場合、NetScaler ADC アプライアンスは内部的に 1000 個の拡張 ACL を作成します。
  - 例 **2**: 次のパラメータが設定されたデータセットベースの拡張 ACL
    - 送信元 IP は、5 つの IP アドレスを含むデータセットに設定されます。
    - 宛先 IP は、5 つの IP アドレスを含むデータセットに設定されます。
    - 送信元ポートは、5 つのポートを含むデータセットに設定されます。
    - 宛先ポートは、5 つのポートを含むデータセットに設定されます。

    NetScaler ADC アプライアンスは、内部的に 625 の拡張 ACL を作成します。これらの内部 ACL にはそれぞれ、上記の 4 つのパラメータ値の一意の組み合わせが含まれています。
  - NetScaler ADC アプライアンスは、最大 10K の拡張 ACL をサポートします。データセットにバインドされた IP アドレスの範囲を持つ IPv4 データセットベースの拡張 ACL の場合、拡張 ACL の合計数が上限に達すると、NetScaler ADC アプライアンスは内部 ACL の作成を停止します。
  - 拡張 ACL 統計情報の一部として、次のカウンタがあります。
    - **ACL** カウント。ユーザによって設定された ACL ルールの総数。
    - 有効な **ACL** カウント。NetScaler ADC アプライアンスが内部で構成する有効な ACL ルールの総数。

    詳細については、拡張 ACL および拡張 ACL6 の統計情報の表示を参照してください。
- NetScaler ADC アプライアンスは、次の `set` および `unset` 操作をサポートしていません。associating/dissociating 拡張 ACL のパラメーターを持つデータセット。ACL パラメータをデータセットに設定できるのは、`add` 操作中に限られます。

データセットベースの拡張 **ACL** を構成する

データセットベースの拡張 ACL ルールの構成は、次のタスクで構成されます。

- データセットを追加します。データセットは、数値（整数）、IPv4 アドレス、または IPv6 アドレスのインデックス付きパターンの配列です。このタスクでは、IPv4 タイプのデータセットなど、データセットのタイプを作成します。
- 値をデータセットにバインドします。データセットの値または値の範囲を指定します。指定する値は、データセットタイプと同じタイプである必要があります。たとえば、IPv4 アドレス、IPv4 アドレス範囲、または IPv4 アドレス範囲を CIDR 表記で IPv4 データセットに指定できます。
- 拡張 **ACL** を追加し、**ACL** パラメーターをデータセットに設定します。拡張 ACL を追加し、必要な ACL パラメータをデータセットに設定します。この設定により、パラメータはデータセットで指定された値に設定されます。
- 拡張 **ACL** を適用します。ACL を適用して、新規または変更された拡張 ACL をアクティブにします。

CLI を使用してポリシーデータセットを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add policy dataset** <name> <type>
- **show policy dataset**

CLI を使用してパターンをデータセットにバインドするには:

コマンドプロンプトで入力します。

- **bind policy dataset** <name> <value> [-endRange \<string>]
- **show policy dataset**

CLI を使用して拡張 **ACL** を追加し、**ACL** パラメーターをデータセットに設定するには:

コマンドプロンプトで入力します。

- **add ns acl** <aclname> <aclaction> [-**srcIP** [\<operator>] <srcIPVal>] [-**srcPort** [\<operator>] <srcPortVal>] [-**destIP** [\<operator>] <destIPVal>] [-**destPort** [\<operator>] <destPortVal>] …
- **show acls**

CLI を使用して拡張 **ACL** を適用するには:

コマンドプロンプトで入力します。

- **apply acls**

設定例

次のデータセットベースの拡張 ACL の設定例では、2 つの IPv4 データセットDATASET_IP_ACL_1とDATASET_IP_ACL_2が作成されます。2 つのポートデータセットDATASET_PORT_ACL_1とDATASET_PORT_ACL_1が作成されます。

192.0.2.30 と 192.0.2.60 の 2 つの IPv4 アドレスがDATASET_IP_ACL_1にバインドされています。(198.51.100.15-45) と (203.0.113.60-90) の 2 つの IPv4 アドレス範囲がDATASET_IP_ACL_2にバインドされています。DATASET_IP_ACL_1は、srcIPパラメータ、および DATASET_IP_ACL_1を拡張ACLACL-1のdestIPパラメータに指定されます。

2001 と 2004 の 2 つのポート番号がDATASET_PORT_ACL_1にバインドされています。(5001-5040) と (8001-8040) の 2 つのポート範囲がバインドされています DATASET-PORT-ACL-2。DATASET_IP_ACL_1 は、srcIP パラメータ、DATASET_IP_ACL_1 および拡張 ACL destIPACL-1のパラメータに指定されます。

```
add policy dataset DATASET_IP_ACL_1 IPV4
add policy dataset DATASET_IP_ACL_2 IPV4

add policy dataset DATASET_PORT_ACL_1 NUM
add policy dataset DATASET_PORT_ACL_2 NUM

bind dataset DATASET_IP_ACL_1 192.0.2.30
bind dataset DATASET_IP_ACL_1 192.0.2.60
bind dataset DATASET_IP_ACL_2 198.51.100.15 -endrange 198.51.100.45
bind dataset DATASET_IP_ACL_2 203.0.113.1/24

bind dataset DATASET_PORT_ACL_1 2001
bind dataset DATASET_PORT_ACL_1 2004
bind dataset DATASET_PORT_ACL_2 5001 -endrange 5040
bind dataset DATASET_PORT_ACL_2 8001 -endrange 8040

add ns acl ACL-1 ALLOW -srcIP DATASET_IP_ACL_1 -destIP DATASET_IP_ACL_2
-srcPort DATASET_PORT_ACL_1 -destPort DATASET_PORT_ACL_2  - protocol TCP
<!--NeedCopy-->
```

ACL の MAC アドレスのワイルドカードマスク

August 15, 2023

拡張 ACL と ACL6 にワイルドカードマスクパラメータが導入されました。ワイルドカードマスクパラメータは、送信元 MAC アドレスパラメータと共に使用して、着信パケットの送信元 MAC アドレスと照合する MAC アドレスの範囲を定義します。

ワイルドカードマスクは、使用する MAC アドレスの 16 進数と、無視する 16 進数を指定します。ワイルドカードマスクパラメータは一連の 1 と 0 を指定し、長さは 12 桁です。各桁は、MAC アドレスの対応する 16 進数のマスクで

す。ワイルドカードマスクのゼロ桁は、MAC アドレスの対応する 16 進数を考慮する必要があることを示し、1 桁は対応する 16 進数を無視することを示します。

ワイルドカードマスクは、次の条件を満たす必要があります。

- ゼロのシリーズが 1 つしかありません
- シリーズは 1 つだけです
- 一連の 0 から始める

有効なワイルドカードマスクの例を以下に示します。

- 000000111111
- 000000011111
- 000011111111

次に、無効なワイルドカードマスクの例をいくつか示します。

- 000000111100
- 111110000000
- 010101010101

ACL の場合、MAC アドレス 96: fa: 95:1 d: 67:4 a のワイルドカードマスク 000000111111 は、MAC アドレス範囲 96: FA: 95:00:00:00-96: FA: 95: FF: FF: FF: FF: FF: FF を定義します。この MAC アドレス範囲は、着信パケットの送信元 MAC アドレスと照合されます。

CLI を使用して ACL ルール内の送信元 MAC アドレスの範囲を指定するには:

コマンドプロンプトで入力します。

```
1 -  add ns acl <name> <action> -srcMac <mac_addr> -srcMacMask <string>
2 -  show ns acl <aclname>
3 <!--NeedCopy-->
```

例:

```
1 add ns acl ACL-1 ALLOW  -protocol TCP  -srcport 2000-3000 -srcMac 96:fa
     :95:1d:67:4a
2   -srcMacMask 000000111111
3 Done
4 <!--NeedCopy-->
```

CLI を使用して ACL6 ルール内の送信元 MAC アドレスの範囲を指定するには:

コマンドプロンプトで入力します。

```
1 -  add ns acl6 <name> <action> -srcMac <mac_addr> -srcMacMask <string>
2 -  show ns acl6 <acl6name>
3 <!--NeedCopy-->
```

例:


```
> add ns acl6 ACL6-1 ALLOW -destIPv6 2001::45 -srcMac 96:fa:90:1d:67:4a
 - srcMacMask 000000001111
Done
<!--NeedCopy-->
```

内部ポートでのトラフィックのブロック

August 15, 2023

デフォルトでは、NetScaler ADC アプライアンスは、ACL ルールを使用しても一部の種類の内部トラフィックをブロックしません。

次の表は、NetScaler ADC アプライアンスが ACL ルールを使用してもブロックしない内部トラフィックの種類を示しています。

NetScaler セットアップ	プロトコル	送信先ポート	宛先 IP アドレス
すべて	TCP	3008-3011	NSIP または SNIP
すべて	TCP	179	NSIP または SNIP
すべて	UDP	520	NSIP または SNIP
高可用性	UDP	3003	NSIP
高可用性	TCP	22	NSIP
クラスター	UDP	7000	NSIP

前述のタイプのトラフィックをブロックしないというこの機能は、グローバルレイヤ 3 `Implicit ACL Allow`（`implicitACLAllow`）パラメータのデフォルト設定で指定されます。

ACL ルールを使用して前述のトラフィックタイプをブロックする場合は、このパラメータを無効にできます。高可用性セットアップのアプライアンスは、そのパートナー（プライマリまたはセカンダリ）ノードの例外を作成します。そのノードからのトラフィックはブロックされません。

CLI を使用してこのパラメータを無効または有効にするには:

コマンドプロンプトで入力します。

set l3param -implicitACLAllow [ENABLED	無効]

-

- **sh l3param**

注: パラメータ implicitaclAllow はデフォルトで有効になっています。

例:

```
> set l3param -implicitACLAllow DISABLED
Done
<!--NeedCopy-->
```

IP ルーティング

August 15, 2023

NetScaler アプライアンスは、動的ルーティングと静的ルーティングの両方をサポートしています。シンプルなルーティングは NetScaler の主な役割ではないため、動的ルーティングプロトコルを実行する主な目的は、ルートヘルスインジェクション（RHI）を有効にして、上流ルーターが地形的に分散した仮想サーバーへの複数のルートの中から最適なルートを選択できるようにすることです。

NetScaler のほとんどの実装では、ルーティングのオーバーヘッドを減らすためにいくつかの静的ルートを使用しています。バックアップスタティックルートとモニタールートを作成して、スタティックルートがダウンした場合に自動スイッチオーバーを有効にできます。また、重みを割り当ててスタティックルート間のロードバランシングを容易にしたり、ヌルルートを作成してルーティングループを防止したり、IPv6 スタティックルートを設定したりすることもできます。指定した基準に基づいてルーティングを決定するポリシーベースルート（PBR）を設定できます。

動的ルートの構成

August 15, 2023

動的ルーティングプロトコルが有効な場合、対応するルーティングプロセスがルートの更新を監視し、ルートをアドバタイズします。ルーティングプロトコルにより、アップストリームルーターは等価コストマルチパス（ECMP）技術を使用して、2 つのスタンドアロン NetScaler アプライアンスでホストされている同一の仮想サーバーにトラフィックを負荷分散できます。NetScaler アプライアンスの動的ルーティングは、3 つのルーティングテーブルを使用します。高可用性設定では、セカンダリアプライアンスのルーティングテーブルがプライマリアプライアンスのルーティングテーブルとミラーリングします。

ダイナミックルーティングプロトコルに関するコマンドリファレンスガイドおよびサポートされていないコマンドについては、ダイナミックルーティングプロトコルコマンドリファレンスガイドおよびサポートされていないコマンドを参照してください。

NetScaler ADC では、次のプロトコルがサポートされています。

- ルーティング情報プロトコル (RIP) バージョン 2
- オープン・ショート・パス・ファースト（OSPF）バージョン 2
- Border Gateway Protocol（BGP）
- IPv6 用次世代ルーティング情報プロトコル (RIPNG)
- IPv6 用オープン・ショーテスト・パス・ファースト (OSPF) バージョン 3
- ISIS プロトコル

複数のプロトコルを同時に有効にできます。

NetScaler のルーティングテーブル

NetScaler アプライアンスでは、NetScaler カーネルルーティングテーブル、FreeBSD カーネルルーティングテーブル、および NSM FIB ルーティングテーブルはそれぞれ異なるルートセットを保持し、異なる目的を果たします。UNIX ルーティングソケットを使用して相互に通信します。ルートの更新は、あるルーティングテーブルから別のルーティングテーブルに自動的に伝播されません。ルーティングテーブルごとにルート更新の伝播を設定する必要があります。

NS カーネルルーティングテーブル

NS カーネル・ルーティング・テーブルには、NSIP および各 SNIP および MIP に対応するサブネット・ルートが格納されます。通常、VIP に対応するルートは NS カーネルのルーティングテーブルには存在しません。例外は add ns ip コマンドを使用して追加され、255.255.255.255 以外のサブネットマスクで設定された VIP です。同じサブネットに属する複数の IP アドレスがある場合、それらは単一のサブネットルートとして抽象化されます。さらに、このテーブルには、ループバックネットワーク（127.0.0.0）へのルートと、CLI（CLI）を介して追加されたすべてのスタティックルートが含まれています。この表のエントリは、NetScaler がパケット転送に使用します。CLI から show route コマンドを使用してこれらを検査できます。

FreeBSD ルーティングテーブル

FreeBSD ルーティングテーブルの唯一の目的は、管理トラフィック (telnet、ssh など) の開始と終了を容易にすることです。NetScaler アプライアンスでは、これらのアプリケーションは FreeBSD と密接に結びついているため、FreeBSD がこれらのアプリケーションとの間のトラフィックを処理するために必要な情報を持っていることが不可欠です。このルーティングテーブルには、NSIP サブネットへのルートとデフォルトルートが含まれています。さらに、NetScaler がローカルネットワーク上のホストへの接続を確立すると、FreeBSD は WASCloned (W) タイプのルートを追加します。このルーティングテーブルのエントリは非常に特殊化されているため、NS カーネルと NSM FIB ルーティングテーブルからの他のすべてのルート更新は FreeBSD ルーティングテーブルをバイパスします。route コマンドで変更しないでください。FreeBSD のルーティングテーブルは、どの UNIX シェルからでも netstat コマンドを使って調べることができます。

ネットワークサービスモジュール **(NSM) FIB**

NSM FIB ルーティングテーブルには、ダイナミックルーティングプロトコルによってネットワーク内のピアに配信されるアドバタイズ可能なルートが含まれています。以下が含まれる場合があります。

- 接続ルート。NetScaler から直接アクセスできる IP サブネット。通常、NSIP サブネットとルーティングプロトコルが有効なサブネットに対応するルートは、接続されたルートとして NSM FIB に存在します。
- カーネルルート。-HostRoute オプションが有効なすべての VIP アドレスは、必要な RHI レベルを満たしていれば、カーネルルートとして NSM FIB に存在します。さらに、NSM FIB には、CLI で設定された-advertise オプションが有効になっているすべてのスタティックルートが含まれます。または、NetScaler が静的ルートアドバタイズメント（SRADV）モードで動作している場合、CLI で構成されたすべての静的ルートが NSM FIB に存在します。これらのスタティックルートは、実際には NS カーネルルーティングテーブルに属しているため、NSM FIB ではカーネルルートとしてマークされます。
- スタティックルート。通常、VTYSH で設定されたスタティックルートはすべて NSM FIB に存在します。プロトコルのアドミニストレーティブディスタンスが変更されても、常にそうなるとは限りません。注意すべき重要な点は、これらのルートは NS カーネルのルーティングテーブルには絶対に入らないということです。
- 学習したルート。NetScaler がルートを動的に学習するように構成されている場合、NSM FIB にはさまざまな動的ルーティングプロトコルによって学習されたルートが含まれます。ただし、OSPF が学習したルートには特別な処理が必要です。OSPF プロセスで fib-install オプションが有効になっている場合にのみ、FIB にダウンロードされます。これは VTYSH のルータ設定ビューから実行できます。

高可用性セットアップでの動的ルーティング

高可用性セットアップでは、プライマリノードがルーティングプロセスを実行し、ルーティングテーブルの更新をセカンダリノードに伝播します。2 次ノードのルーティングテーブルは、1 次ノードのルーティングテーブルをミラーリングします。

ノンストップフォワーディング

フェイルオーバー後、セカンダリノードはプロトコルの起動、ルートの学習、ルーティングテーブルの更新にしばらく時間がかかります。ただし、これはルーティングには影響しません。セカンダリノードのルーティングテーブルは、プライマリノードのルーティングテーブルと同一だからです。この操作モードはノンストップフォワーディングと呼ばれます。

ブラックホール回避メカニズム

フェールオーバー後、新しいプライマリノードはすべての VIP ルートを上流ルータに注入します。ただし、そのルーターは古いプライマリノードのルートを 180 秒間保持します。ルータはフェールオーバーを認識しないため、2 つのノード間のトラフィックの負荷分散を試みます。古いルートが期限切れになる前の 180 秒間、ルータはトラフィックの半分を古い非アクティブなプライマリノードに送信します。これは事実上、ブラックホールです。

これを防ぐために、新しいプライマリノードは、ルートを挿入するときに、古いプライマリノードで指定されたメトリックよりもわずかに低いメトリックを割り当てます。

ダイナミックルーティングを設定するためのインターフェイス

動的ルーティングを設定するには、GUI またはコマンドラインインターフェイスを使用できます。NetScaler は、CLI と仮想テレタイプシェル（VTYSH）という 2 つの独立したコマンドラインインターフェイスをサポートしています。CLI はアプライアンスのネイティブシェルです。VTYSH は ZeboS によって公開されています。NetScaler のルーティングスイートは、GNU Zebra の商用バージョンである ZeboS をベースにしています。

> 注:
>
> Citrix では、CLI でのみ設定できるコマンドを除くすべてのコマンドに VTYSH を使用することをお勧めします。通常、CLI の使用は、ルーティングプロトコルの有効化、ホストルートアドバタイズの設定、およびパケット転送用のスタティックルートの追加を行うコマンドに限定する必要があります。

ダイナミックルーティングプロトコルコマンドリファレンスガイドおよびサポートされていないコマンド

次の表に、NetScaler ADC アプライアンスでのさまざまな動的ルーティングプロトコル、およびサポートされていないコマンドに関するコマンドリファレンスガイドのリンクを示します：動的ルーティングプロトコルリファレンスガイドおよびサポートされていないコマンド。

RIP の構成

August 15, 2023

ルーティング情報プロトコル (RIP) はディスタンスベクトルプロトコルです。NetScaler は、RFC 1058 および RFC 2453 で定義されている RIP をサポートしています。RIP はどのサブネットでも実行できます。

RIP を有効にしたら、RIP ルートのアドバタイズメントを設定する必要があります。トラブルシューティングでは、RIP の伝播を制限できます。RIP 設定を表示して構成を確認できます。

RIP の有効化と無効化

次のいずれかの手順を使用して、RIP を有効または無効にします。RIP を有効にすると、NetScaler アプライアンスは RIP プロセスを開始します。RIP を無効にすると、アプライアンスは RIP プロセスを停止します。

CLI を使用して RIP ルーティングを有効または無効にするには:

コマンドプロンプトで、次のコマンドのいずれかを入力して RIP を有効または無効にします。

- **enable ns feature RIP**
- **disable ns feature RIP**

GUI を使用して RIP ルーティングを有効または無効にするには:

1. [システム**] > [**設定**]** に移動し、[モードと機能] グループで [詳細機能の変更] をクリックします。
2. **RIP** ルーティングオプションを選択または選択解除します。

広告ルート

RIP により、アップストリームルーターは、2 つのスタンドアロン NetScaler アプライアンスでホストされている 2 つの同一の仮想サーバー間でトラフィックの負荷分散を行うことができます。ルートアドバタイズメントにより、アップストリームルーターは NetScaler の背後にあるネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用してルートをアドバタイズするように RIP を設定するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router rip	RIP ルーティングプロセスを開始し、ルーティングプロセスの構成モードに入ります。
redistribute static	スタティックルートを再配布します。
redistribute kernel	カーネルルートを再配布する。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

RIP 伝播の制限

設定のトラブルシューティングが必要な場合は、任意のインターフェイスでリッスン専用モードを設定できます。

VTYSH コマンドラインを使用して RIP の伝播を制限するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router rip	RIP ルーティングプロセスを開始し、ルーティングプロセスの構成モードに入ります。
passive-interface < vlan_name>	指定した VLAN にバインドされたインターフェイスのルーティングアップデートを抑制します。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

RIP 設定の検証

ルーティングテーブルと他の RIP 設定を表示できます。

VTYSH コマンドラインを使用して RIP 設定を表示するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
sh rip	更新された RIP ルーティングテーブルを表示します。
sh リッピングインターフェース	指定した VLAN の RIP 情報を表示します。

例:

```
NS# VTYSH
NS# sh rip
NS# sh rip interface VLAN0
<!--NeedCopy-->
```


OSPF の構成

August 15, 2023

NetScaler は、オープン最短パスファースト（OSPF）バージョン 2（RFC 2328）をサポートしています。NetScaler ADC 上の OSPF の機能は次のとおりです。

- vserver がアクティブな場合、vserver へのホストルートをルーティングプロトコルに挿入できます。
- OSPF はどのサブネットでも実行できます。
- 隣接する OSPF ルーターによってアドバタイズされるルートラーニングは、NetScaler で無効にできます。
- NetScaler は、すべてのルートのタイプ 1 またはタイプ 2 の外部メトリックをアドバタイズできます。
- NetScaler は、VIP ルートのユーザー指定のメトリック設定をアドバタイズできます。たとえば、特別なルートマップを使用せずに VIP ごとにメトリックを設定できます。
- NetScaler の OSPF エリア ID を指定できます。
- NetScaler は、それほどスタブではないエリア（NSSA）をサポートしています。NSSA は OSPF スタブエリアに似ていますが、外部ルートを限定的にスタブエリアに挿入できます。NSSA をサポートするために、リンクステートアドバタイズメント（LSA）領域の新しいオプションビット（N ビット）と新しいタイプ（タイプ 7）が定義されています。タイプ 7 LSA は、NSSA 内の外部ルート情報をサポートします。NSSA エリアボーダールータ（ABR）は、タイプ 7 LSA をタイプ 5 LSA に変換し、OSPF ドメインに伝播します。OSPF 仕様では、次の一般クラスのエリア設定のみが定義されています。
  - タイプ 5 LSA：エリア内部のルータから発信されたルータは、AS Boarder Router（ASBR）によってドメインにフラッディングされます。
  - スタブ：タイプ 5 LSA をエリア内またはエリア全体に伝搬することはできず、代わりに外部宛先へのデフォルトルーティングに依存します。

OSPF を有効にしたら、OSPF ルートのアドバタイズメントを設定する必要があります。トラブルシューティングのために、OSPF の伝播を制限できます。OSPF 設定を表示して、設定を確認できます。

OSPF の有効化と無効化

OSPF を有効または無効にするには、CLI または GUI を使用する必要があります。OSPF が有効になっている場合、NetScaler は OSPF プロセスを開始します。OSPF が無効になっている場合、NetScaler は OSPF ルーティングプロセスを停止します。

CLI を使用して OSPF ルーティングを有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

1. **enable ns feature OSPF**
2. **disable ns feature OSPF**

GUI を使用して OSPF ルーティングを有効または無効にするには

1. [システム] > [設定] に移動し、[モードと機能] グループで [詳細機能の変更] をクリックします。
2. [**OSPF** ルーティング] オプションをオンまたはオフにします。

OSPF ルートの宣伝

OSPF を使用すると、アップストリームルーターは、2 つのスタンドアロン NetScaler ADC アプライアンスでホストされている 2 つの同一の仮想サーバー間でトラフィックの負荷を分散できます。ルートアドバタイズにより、アップストリームルーターは NetScaler ADC 背後に位置するネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用してルートをアドバタイズするように OSPF を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router OSPF	OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
network A.B.C.D/M area <0-4294967295>	IP ネットワーク上でルーティングを有効にします。
redistribute static	スタティックルートを再配布します。
redistribute kernel	カーネルルートを再配布する。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# network 10.102.29.0/24 area 0
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

OSPF 伝搬の制限

設定のトラブルシューティングが必要な場合は、任意の VLAN でリッスン専用モードを設定できます。

VTYSH コマンドラインを使用して OSPF 伝播を制限するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router OSPF	OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
passive-interface < vlan_name>	指定した VLAN にバインドされたインターフェイスのルーティングアップデートを抑制します。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

OSPF 設定の確認

現在の OSPF ネイバーと OSPF ルートを表示できます。

VTYSH コマンドラインを使用して OSPF 設定を表示するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
sh OSPF neighbor	現在のネイバーを表示します。
sh OSPF route	OSPF ルートを表示します。

例:

```
>VTYSH
NS# sh ip OSPF neighbor
NS# sh ip OSPF route
<!--NeedCopy-->
```

OSPF のグレースフルリスタートの設定

ルーティングプロトコルが設定されている INC 以外の High Availability（HA; 高可用性）設定では、フェールオーバー後にルーティングプロトコルがコンバージされ、新しいプライマリノードと隣接ネイバールータ間のルートが学

習されます。ルート学習が完了するまでに時間がかかります。この間、パケットの転送が遅れ、ネットワークパフォーマンスが低下し、パケットがドロップされる可能性があります。

グレースフルリスタートにより、フェールオーバー中の HA セットアップによって、古いプライマリノードの学習済みルートがルーティングデータベースから削除されないように隣接ルータに指示できます。新しいプライマリノードと隣接ルータは、古いプライマリノードのルーティング情報を使用して、ネットワークパフォーマンスを低下させることなく、直ちにパケットの転送を開始します。

> 注:
>
> INC モードの高可用性セットアップでは、グレースフルリスタートはサポートされていません。

VTYSH コマンドラインを使用して OSPF のグレースフルリスタートを設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
VTYSH	VTYSH	VTYSH コマンドプロンプトに入ります。
configure terminal	NS# configure terminal	グローバル構成モードを開始します。
router-id	NS(config)# router-id 1.1.1.1	NetScaler ADC アプライアンスのルーター識別子を設定します。この ID は、すべてのダイナミックルーティングプロトコルに対して設定されます。HA セットアップでグレースフルリスタートが正しく機能するためには、高可用性セットアップの他のノードに同じ ID を指定する必要があります。
ospf restart grace-period <1-1800>	NS(config)# ospf restart grace-period 170	ヘルパーデバイスでルートが保持される猶予期間を秒単位で指定します。デフォルト値:120 秒。
ospf restart helper max-grace-period <1-1800>	NS(config)# ospf restart helper max-grace-period 180	これは、NetScaler アプライアンスがヘルパーモードになる最大猶予期間を制限するオプションコマンドです。NetScaler アプライアンスは、設定されたヘルパー最大猶予期間よりも長い猶予期間を持つ不透明な LSA を受信した場合、LSA は破棄され、NetScaler はヘルパーモードになりません。

コマンド	例	コマンドの説明
router ospf	NS(config)# router ospf	OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
network A.B.C.D/M area <0-4294967295>	NS(config-router)# network 192.0.2.0/24 area 0	IP ネットワーク上でルーティングを有効にします。
capability restart graceful	NS(config-router)# capability restart graceful	OSPF ルーティングプロセスでグレースフルリスタートを有効にします。
redistribute kernel	NS(config-router)# redistribute kernel	カーネルルートを再配布します。

BGP の設定

August 15, 2023

NetScaler アプライアンスは BGP（RFC 4271）をサポートしています。NetScaler ADC での BGP の機能は次のとおりです。

- NetScaler ADC は BGP ピアにルートをアドバタイズします。
- NetScaler ADC は、基盤となる仮想サーバーの正常性によって決定される仮想 IP アドレス（VIP）にホストルートを挿入します。
- NetScaler ADC は、HA 構成でのフェイルオーバー後、セカンダリノードで BGP を実行するための構成ファイルを生成します。
- このプロトコルは IPv6 ルート交換をサポートします。
- As-Override Support in Border Gateway Protocol の As-Override サポート

BGP を有効にしたら、BGP ルートのアドバタイズメントを設定する必要があります。トラブルシューティングのために、BGP 伝播を制限できます。BGP 設定を表示して、設定を確認できます。

BGP の有効化と無効化

BGP を有効または無効にするには、CLI または GUI を使用する必要があります。BGP が有効になると、NetScaler ADC アプライアンスは BGP プロセスを開始します。BGP が無効になっている場合、アプライアンスは BGP プロセスを停止します。

CLI を使用して BGP ルーティングを有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- ns 機能 BGP を有効にする
- ns 機能 BGP を無効にする

GUI を使用して BGP ルーティングを有効または無効にするには

1. [システム] > [設定] に移動し、[モードと機能] グループで [詳細機能の変更] をクリックします。
2. [BGP ルーティング] オプションをオンまたはオフにします。

IPv4 ルートの宣伝

NetScaler ADC アプライアンスは、ホストルートを VIP にアドバタイズし、ルートをダウンストリームネットワークにアドバタイズするように構成できます。

VTYSH コマンドラインを使用して IPv4 ルートをアドバタイズするように BGP を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。指定可能な値:1 から 4,294,967,295 です。
ネイバーリモート < IPv4 address> AS < as-number>	IPv4 BGP ネイバーテーブルを、指定した自律システム内のネイバーのリンクローカル IPv4 アドレスで更新します。
アドレスファミリ ipv4	アドレスファミリ構成モードを開始します。
< IPv4 address> 近隣アクティブ化	リンクローカルアドレスを使用して、ピアとローカルノード間で IPv4 ルータファミリのプレフィクスを交換します。
redistribute kernel	カーネルルートを再配布する。
redistribute static	スタティックルートを再配布します。

例**:**

```
1 >VTYSH
2 NS# configure terminal
3 NS(config)# router BGP 5
4 NS(config-router)# Neighbor 10.102.29.170 remote-as 100
5 NS(config-router)# Address-family ipv4
6 NS(config-router-af)# Neighbor 10.102.29.170 activate
```



```
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
<!--NeedCopy-->
```

IPv6 BGP ルートの宣伝

Border Gateway Protocol（BGP）を使用すると、アップストリームルーターは、2 つのスタンドアロンの NetScaler ADC アプライアンスでホストされている 2 つの同一の仮想サーバー間でトラフィックの負荷を分散できます。ルートアドバタイズにより、アップストリームルーターは NetScaler ADC 背後に位置するネットワークエンティティを追跡できます。

IPv6 BGP の前提条件

IPv6 BGP の設定を開始する前に、次の操作を行います。

- IPv6 BGP プロトコルを理解していることを確認します。
- IPv6 機能を有効にします。

構成の手順

VTYSH コマンドラインを使用して IPv6 ルートをアドバタイズするように BGP を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。指定可能な値:1 から 4,294,967,295 です。
ネイバーリモート < IPv6 address> AS < as-number>	IPv6 BGP ネイバーテーブルを、指定した自律システム内のネイバーのリンクローカル IPv6 アドレスで更新します。
アドレスファミリ ipv6	アドレスファミリ構成モードを開始します。
近隣アクティブ化 < IPv6 address>	リンクローカルアドレスを使用して、ピアとローカルノード間で IPv6 ルーターファミリのプレフィクスを交換します。
redistribute kernel	カーネルルートを再配布する。

コマンド	指定
redistribute static	スタティックルートを再配布します。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor a1bc::102 remote-as 100
NS(config-router)# Address-family ipv6
NS(config-router-af)# Neighbor a1bc::102 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
<!--NeedCopy-->
```

BGP 設定の確認

VTYSH を使用して BGP 設定を表示できます。

VTYSH コマンドラインを使用して BGP 設定を表示するには

コマンドプロンプトで入力します。

```
VTYSH
You are now in the VTYSH command prompt. An output similar to the
    following appears:
NS170#
At the VTYSH command prompt, type:
NS170# sh ip BGP
NS170# sh BGP
NS170# sh ip BGP neighbors
NS170# sh ip BGP summary
NS170# sh ip BGP route-map <map-tag>
<!--NeedCopy-->
```

As-Override Support in Border Gateway Protocol の As-Override サポート

BGP ループ防止機能の一部として、自律システム（AS）パスでルータの自律システム番号（ASN）を含む BGP パケットをルータが受信すると、ルータはそのパケットをドロップします。パケットはルータから発信され、発信元の場所に到達したと想定されます。

企業に同じ ASN を持つサイトが複数ある場合、BGP ループ防止により、同じ ASN を持つサイトは別の ASN によってリンクされません。ルーティングアップデート（BGP パケット）は、別のサイトが受信するとドロップされます。

この問題を解決するために、NetScaler ZeboS BGP ルーティングモジュールに BGP AS-Override 機能が追加されました。

ピアデバイスで AS-Override が有効になっている場合、NetScaler ADC アプライアンスがピアに転送する BGP パケットを受信し、パケットの ASN がピアの ASN と一致すると、アプライアンスは BGP パケットの ASN を独自の ASN 番号に置き換えてからパケットを転送します。

VTYSH コマンドラインを使用して、特定のネイバーまたはネイバーのグループ（ピアグループ）に対して AS オーバーライドを有効にできます。

VTYSH コマンドラインを使用して IPv4 ネイバーの BGP AS オーバーライドを設定するには、次の手順を実行します。

コマンド	指定
configure terminal	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。
ネイバーリモート < IPv4 address> AS < as-number>	IPv4 BGP ネイバーテーブルを、指定した自律システム内のネイバーの IPv4 アドレスで更新します。
ネイバーとしてオーバーライド	指定したネイバーの BGP as-override を有効にします。

```
1      > VTYSH NS# configure terminal
2      NS(config)# router BGP 200
3      NS(config-router)# Neighbor 192.0.2.100 remote-as 100
4      NS(config-router)# Neighbor 10.102.29.100 as-override
5  <!--NeedCopy-->
```

VTYSH コマンドラインを使用して IPv4 BGP ピアグループの BGP AS オーバーライドを設定するには、次の手順を実行します。

コマンド	指定
configure terminal	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。
** ネイバーピアグループ **	BGP ピアグループを作成します。
** ネイバー・ピア・グループ **	ネイバーを指定されたピアグループに関連付けます。
ネイバーリモート AS < as-number>	IPv4 BGP ネイバーテーブルを、指定した自律システム内のネイバーの IPv4 アドレスで更新します。
ネイバーとしてオーバーライド	指定したピアグループに関連付けられているすべてのネイバーに対して、BGP as-override を有効にします。

```
1      > VTYSH NS# configure terminal
```



```
    NS(config)# router BGP 200
    NS(config-router)# neighbor external-peers-1 peer-group
    NS(config-router)# neighbor 192.0.2.101 peer-group external-peers-1
    NS(config-router)# neighbor 192.0.2.102 peer-group external-peers-1
    NS(config-router)# neighbor 192.0.2.103 peer-group external-peers-1
    NS(config-router)# Neighbor external-peers-1 remote-as 100
    NS(config-router)# Neighbor external-peers-1 as-override
<!--NeedCopy-->
```

VTYSH コマンドラインを使用して IPv6 ネイバーの BGP AS オーバーライドを設定するには、次の手順を実行します。

コマンド	指定
configure terminal	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。
ネイバーリモート < IPv6 address> AS < as-number>	IPv4 BGP ネイバーテーブルを、指定した自律システム内のネイバーの IPv4 アドレスで更新します。
ネイバーとしてオーバーライド	指定したネイバーの BGP as-override を有効にします。
アドレスファミリ **ipv6**	アドレスファミリ構成モードを開始します。
近隣アクティブ化 < IPv6 address>	リンクローカルアドレスを使用して、指定されたネイバーと NetScaler ADC との間で、IPv6 ルーターファミリのプレフィックスを交換します。
ネイバーとしてオーバーライド	指定したネイバーの BGP as-override を有効にします。

```
    > VTYSH NS# configure terminal
    NS(config)# router BGP 200
    NS(config-router)# Neighbor a1bc::102 remote-as 100
    NS(config-router)# Neighbor a1bc::102 as-override
    NS(config-router)# Address-family ipv6
    NS(config-router-af)# Neighbor a1bc::102 activate
    NS(config-router)# Neighbor a1bc::102 as-override
<!--NeedCopy-->
```

VTYSH コマンドラインを使用して IPv6 ピアグループの BGP AS オーバーライドを設定するには、次の手順を実行します。

コマンド	指定
configure terminal	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。

コマンド	指定
** ネイバーピアグループ **	BGP ピアグループを作成します。
** ネイバーピアグループ **	ネイバーを指定されたピアグループに関連付けます。
ネイバーリモート AS < as-number>	IPv4 BGP ネイバーテーブルを、指定した自律システム内のネイバーの IPv4 アドレスで更新します。
ネイバーとしてオーバーライド	指定したピアグループに関連付けられているすべてのネイバーに対して、BGP as-override を有効にします。
アドレスファミリ **ipv6**	アドレスファミリ構成モードを開始します。
近隣アクティブ化	リンクローカルアドレスを使用して、指定されたピアグループのネイバーと NetScaler ADC 間で IPv6 ルーターファミリのプレフィックスを交換します。
ネイバーとしてオーバーライド	指定したピアグループに関連付けられているすべてのネイバーに対して、BGP as-override を有効にします。

```
1      > VTYSH NS# configure terminal
2      NS(config)# router BGP 200
3      NS(config-router)# neighbor external-peers-2 peer-group
4      NS(config-router)# neighbor 2001::1 peer-group external-peers-2
5      NS(config-router)# neighbor 2001::2 peer-group external-peers-2
6      NS(config-router)# Neighbor external-peers-2 remote-as 100
7      NS(config-router)# Neighbor external-peers-2 as-override
8      NS(config-router)# Address-family ipv6
9      NS(config-router-af)# Neighbor external-peers-2 activate
10     NS(config-router)# Neighbor external-peers-2 as-override
11 <!--NeedCopy-->
```

グレースフルリスタート

ルーティングプロトコルが設定されている INC 以外の High Availability（HA; 高可用性）設定では、フェールオーバー後にルーティングプロトコルがコンバージされ、新しいプライマリノードと隣接ネイバールータ間のルートが学習されます。ルート学習が完了するまでに時間がかかります。この間、パケットの転送が遅れ、ネットワークパフォーマンスが低下し、パケットがドロップされる可能性があります。

グレースフルリスタートにより、フェールオーバー中の HA セットアップによって、古いプライマリノードの学習済みルートがルーティングデータベースから削除されないように隣接ルータに指示できます。新しいプライマリノードと隣接ルータは、古いプライマリノードのルーティング情報を使用して、ネットワークパフォーマンスを低下させることなく、直ちにパケットの転送を開始します。

> 注:
>
> INC モードの高可用性セットアップでは、グレースフルリスタートはサポートされていません。

BGP のグレースフルリスタートの設定

VTYSH コマンドラインを使用して BGP のグレースフルリスタートを設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
VTYSH	VTYSH	VTYSH コマンドプロンプトに入ります。
configure terminal	NS# configure terminal	グローバル構成モードを開始します。
router-id	NS(config)# router-id 1.1.1.1	NetScaler ADC アプライアンスのルーター識別子。この ID は、すべてのダイナミックルーティングプロトコルに対して設定されます。グレースフルリスタートを正しく機能させるには、高可用性セットアップの他のノードに同じ ID を指定する必要があります。
router bgp	NS(config)# router bgp 5	BGP 構成モードを開始します。
bgp graceful-restart	NS(config)# bgp graceful-restart	BGP ルーティングプロセスでグレースフルリスタートを有効にします。
bgp graceful-restart restart-time <1-1800>	NS(config-router)# bgp graceful-restart restart-time 170	フェールオーバー後、ヘルパールータが新しいプライマリノードからの TCP 接続を待機する猶予期間を秒単位で指定します。この時間の間、ヘルパールーターはルートを保持します。
bgp graceful-restart stalepath-time <1-1800>	NS(config-router)# bgp graceful-restart stalepath-time 180	ヘルパーモードの NetScaler ADC アプライアンスがネイバールーターを再起動するための古いルートを保持する時間を秒単位で指定します。デフォルト値は 360 秒です。
neighbor remote-as	NS(config-router)# neighbor 192.0.2.30 remote-as 2	指定されたネイバールータデバイスとの BGP ピアリングを確立します。
neighbor capability graceful-restart	NS(config-router)# neighbor 192.0.2.30 capability graceful-restart	指定したネイバーとのグレースフルリスタートを有効にします。
redistribute kernel	NS(config-router)# redistribute kernel	カーネルルートを再配布します。

IPv6 BGP のグレースフルリスタートの設定

VTYSH コマンドラインを使用して IPv6 BGP のグレースフルリスタートを設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
VTYSH	VTYSH	VTYSH コマンドプロンプトに入ります。
configure terminal	NS# configure terminal	グローバル構成モードを開始します。
router-id	NS(config)# router-id 1.1.1.1	NetScaler ADC アプライアンスのルーター識別子を設定します。この ID は、すべてのダイナミックルーティングプロトコルに対して設定されます。グレースフルリスタートを正しく機能させるには、高可用性セットアップの他のノードに同じ ID を指定する必要があります。
router bgp	NS(config)# router bgp 5	BGP プロトコルの構成モードを開始します。
bgp graceful-restart	NS(config)# bgp graceful-restart	BGP ルーティングプロセスでグレースフルリスタートを有効にします。
bgp graceful-restart restart-time <1-1800>	NS(config-router)# bgp graceful-restart restart-time 170	フェールオーバー後、ヘルパールータが新しいプライマリノードからの TCP 接続を待機する猶予期間を秒単位で指定します。この時間の間、ヘルパールーターはルートを保持します。デフォルト値は 360 秒です。
bgp graceful-restart stalepath-time <1-1800>	NS(config-router)# bgp graceful-restart stalepath-time 180	ヘルパーモードの NetScaler ADC アプライアンスがネイバールーターを再起動するための古いルートを保持する時間を秒単位で指定します。デフォルト値は 360 秒です。
neighbor remote-as	NS(config-router)# neighbor 2001:db8::10 remote-as 2	指定されたネイバールータデバイスとの BGP ピアリングを確立します。
address-family ipv6	NS(config-router)#address-family ipv6	アドレスファミリ構成モードを開始します。
neighbor activate	NS(config-router-af)#neighbor 2001:db8::10 activate	指定したネイバールータデバイスとのアドレスファミリルートの交換を有効にします。

コマンド	例	コマンドの説明
neighbor capability graceful-restart	NS(config-router-af)#neighbor 2001:db8::10 capability graceful-restart	Enables graceful restart with the specified neighbor router device.
redistribute kernel	NS(config-router-af)#redistribute kernel	カーネルルートを再配布します。
exit-address-family	NS(config-router-af)#exit-address-family	アドレスファミリ構成モードを終了します。

IPv4 BGP 用の MD5 認証の設定

NetScaler ADC アプライアンスは、Border Gateway Protocol（BGP）の MD5 認証をサポートしています。認証を有効にすると、NetScaler ADC アプライアンスとそのピアデバイス間で交換される BGP に属する TCP セグメントは、認証が成功した場合にのみ検証され、受け入れられます。認証を成功させるには、両方のピアに同じ MD5 パスワードを設定する必要があります。認証に失敗すると、BGP ネイバー関係は確立されません。NetScaler ADC アプライアンスでの BGP に対する MD5 認証のサポートは、RFC 2385 に準拠しています。

はじめに

BGP MD5 認証の設定を開始する前に、次の点を考慮してください。

- RFC 2385 で説明されている BGP MD5 認証のさまざまなコンポーネントを理解していることを確認します。
- BGP MD5 認証は、NetScaler 管理パーティションではサポートされていません。
- BGP MD5 認証は IPv6 BGP 構成ではサポートされていません。
- BGP MD5 認証は、NetScaler クラスター構成と高可用性構成でサポートされています。
- FreeBSD には次のような問題があるため、レイヤ 2 の高可用性構成では、キープライブとホールドタイムの値を低く設定し（5 と 15 など）、BGP セッションのグレースフルリスタートを構成することをお勧めします。そうしないと、MD5 認証を有効にすると、フェールオーバー後に BGP がネイバーとの接続を再確立するのに時間がかかることがあります。
  - FreeBSD からの最後の ACK には md5 ダイジェストが含まれていません:
    * https://forums.freebsd.org/threads/11170/
    * http://support.pfsense.narkive.com/povrH5HI/bgp-md5-weird-behavior-when-connection-closes

構成の手順

VTYSH コマンドラインを使用して IPv4 BGP の MD5 認証を設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	指定
vtysh	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
**router bgp **	BGP プロトコルの構成モードを開始します。は BGP 自律システム番号で、は必須パラメータです。
neighbor remote-as < AS-number >	IPv4 BGP テーブルを、指定した自律システム内のネイバーの IPv4 アドレスで更新します。
neighbor < neighbour IPv4 address > password < password in double quotes>	指定された MD5 パスワードを使用して、指定されたネイバーの MD5 認証を設定します。MD5 認証を正常に実行するには、NetScaler ADC アプライアンスとネイバーアプライアンスで同じ MD5 パスワードを構成する必要があります。

```
> vtysh

ns# configure terminal

ns(config)#router bgp 5

ns(config-router)#neighbor 20.20.20.138 remote-as 1

ns(config-router)#neighbor 20.20.20.138 password  “secret”

ns(config-router)#redistribute kernel

ns(config-router)#exit

<!--NeedCopy-->
```

4 バイト BGP ASN を asplain および asdot 形式で設定する

NetScaler ADC アプライアンスは、RFC 5396 で定義されているように、4 バイトの BGP 自律システム番号（ASN）を asplain または asdot 形式で構成および表示することをサポートしています。

- 説明する。2 バイトと 4 バイトの ASN の両方が 10 進数値で表される 10 進数値表記。たとえば、65527 は 2 バイト ASN で、234567 は 4 バイト ASN です。
- **asdot** 自律システムのドット表記。2 バイトの ASN は 10 進数で表され (asplain と同じ)、4 バイト ASN はドット表記で表されます。たとえば、65527 は 2 バイト ASN で、3.37959 は 4 バイト ASN です（3.37959 は 234567 の 10 進数のドット形式です）。

asplain および **asdot** 形式の **BGP ASN** 設定の例

デフォルトでは、NetScaler ADC アプライアンスは BGP ASN を asplain 形式で表示しますが、asdot 形式で表示するように構成できます。ローカルおよびリモートの BGP ASN は、asplain または asdot 形式で設定できます。

次に、asplain および asdot 形式の BGP ASN 設定の例をいくつか示します。

- BGP AS 番号を asplain 形式で表示します。デフォルトでは、NetScaler ADC アプライアンスは BGP AS 番号をプレーン形式で表示します。

```
ns#conf t
ns(config)# router bgp 196908
ns(config-router)# end
ns#
ns# sh run router bgp
!
router bgp 196908
!
<!--NeedCopy-->
```

- BGP AS 番号を asdot 形式で表示します。bgp asnotation-dotコマンドを実行して、BGP AS 番号を asdot 形式で表示します。

```
ns#conf t
ns(config)#router bgp 196908
ns(config-router)#bgp asnotation-dot
ns(config-router)#end
ns#
ns#sh run router bgp
!
router bgp 3.300
bgp asnotation-dot
!
<!--NeedCopy-->
```

- BGP AS 番号を asdot 形式で設定して表示します。bgp asnotation-dotコマンドを実行して、BGP AS 番号を asdot 形式で表示します。

```
ns# conf t
ns(config)# router bgp 3.300
ns(config-router)# bgp asnotation-dot
ns#
ns# sh run router bgp
!
router bgp 3.300
bgp asnotation-dot
!
<!--NeedCopy-->
```

- BGP AS 番号を asdot 形式から asplain 形式に戻して表示します。bgp no asnotation-dot コマンドを実行して、BGP AS 番号を asplain 形式に戻して表示します。


```
ns#conf t
ns(config)#router bgp 3.300
ns(config-router)#no bgp asnotation-dot
ns(config-router)#end
ns#

ns#sh run router bgp
!
router bgp 196908
!
<!--NeedCopy-->
```

- リモート as-number を asdot 形式で設定して表示します。bgp asnotation-dotコマンドを実行します。サンプル設定では、リモート as-number 80000 が asdot 形式 1.14464 で設定されています。

```
ns# conf t
ns(config)# router bgp 3.300
ns(config-router)# bgp asnotation-dot
ns(config-router)# neighbor 192.168.1.2 remote-as 1.14464
ns(config-router)#end
ns#
ns#
ns#sh run router bgp
!
router bgp 3.300
bgp asnotation-dot
neighbor 192.168.1.2 remote-as 1.14464
!
ns#
<!--NeedCopy-->
```

- BGP ローカルおよびリモート AS 番号を asdot 形式から asplain 形式に戻して表示します。bgp no asnotation-dot コマンドを実行します。

```
ns#conf t
ns(config)#router bgp 3.300
ns(config-router)#no bgp asnotation-dot
ns(config-router)#end
ns#

ns#sh run router bgp
!
router bgp 196908
neighbor 192.168.1.2 remote-as 80000
!
ns#
<!--NeedCopy-->
```

> 注:
>
> 個々の BGP ネイバー用に設定する代わりに、同じ asplain または asdot 設定を BGP ピアグループにも使用

> できます。

IPv6 RIP の構成

August 15, 2023

IPv6 ルーティング情報プロトコル (RIP) または RIPng はディスタンスベクトルプロトコルです。このプロトコルは、IPv6 をサポートするための RIP の拡張です。IPv6 RIP を有効にしたら、IPv6 RIP ルートのアドバタイズメントを設定する必要があります。トラブルシューティングでは、IPv6 RIP の伝播を制限できます。IPv6 RIP 設定を表示して構成を確認できます。

IPv6 RIP の前提条件

IPv6 RIP の設定を開始する前に、次の操作を行います。

- IPv6 RIP プロトコルを理解していることを確認してください。
- NetScaler ADC アプライアンスに IPv6pt ライセンスをインストールします。
- IPv6 機能を有効にします。

IPv6 RIP ルートのアドバタイジング

IPv6 RIP により、アップストリームルーターは、2 つのスタンドアロン NetScaler デバイスでホストされている 2 つの同一の仮想サーバー間でトラフィックの負荷分散を行うことができます。ルートアドバタイズメントにより、アップストリームルーターは NetScaler の背後にあるネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用して IPv6 ルートをアドバタイズするように IPv6 RIP を設定するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ipv6 rip	IPv6 RIP ルーティングプロセスを開始し、ルーティングプロセスの設定モードに入ります。
redistribute static	スタティックルートを再配布します。
redistribute kernel	カーネルルートを再配布する。

例**:**


```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

IPv6 RIP プロパゲーションの制限

設定をトラブルシューティングする必要がある場合は、任意のインターフェイスでリッスン専用モードを設定できます。

VTYSH コマンドラインを使用して IPv6 RIP の伝播を制限するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ipv6 rip	IPv6 RIP ルーティングプロセスを開始し、ルーティングプロセスの設定モードに入ります。
passive-interface < vlan_name>	指定した VLAN にバインドされたインターフェイスのルーティングアップデートを抑制します。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

IPv6 RIP 構成の検証

VTYSH を使用すると、指定した VLAN の IPv6 RIP ルーティングテーブルと IPv6 RIP 情報を表示できます。

VTYSH コマンドラインを使用して IPv6 RIP 設定を表示するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

Commands	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
sh ipv6 rip	更新された IPv6 RIP ルーティングテーブルを表示します。
sh ipv6 rip interface	指定された VLAN の IPv6 RIP 情報を表示します。

例**:**

```
NS# VTYSH
NS# sh ipv6 rip
NS# sh ipv6 rip interface VLAN0
<!--NeedCopy-->
```

IPv6 OSPF の構成

August 15, 2023

IPv6 OSPF または OSPF バージョン 3（OSPF v3）は、IPv6 ルーティング情報の交換に使用されるリンクステートプロトコルです。IPv6 OSPF を有効にした後は、IPv6 OSPF ルートのアドバタイズメントを設定する必要があります。トラブルシューティングのために、IPv6 OSPF の伝播を制限できます。IPv6 OSPF 設定を表示して、設定を確認できます。

IPv6 OSPF の前提条件

IPv6 OSPF の設定を開始する前に、次の操作を行います。

- IPv6 OSPF プロトコルを理解していることを確認します。
- NetScaler ADC アプライアンスに IPv6pt ライセンスをインストールします。
- IPv6 機能を有効にします。

IPv6 ルートの宣伝

IPv6 OSPF により、アップストリームルーターは、2 つのスタンドアロン NetScaler デバイスでホストされている 2 つの同一の仮想サーバー間でトラフィックの負荷分散を行うことができます。ルートアドバタイズにより、アップストリームルーターは NetScaler ADC 背後に位置するネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用して IPv6 ルートをアドバタイズするように IPv6 OSPF を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

Commands	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ipv6 OSPF	IPv6 OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
redistribute static	スタティックルートを再配布します。
redistribute kernel	カーネルルートを再配布する。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

IPv6 OSPF 伝搬の制限

設定のトラブルシューティングが必要な場合は、VTYSH を使用して、任意の VLAN でリッスン専用モードを設定します。

VTYSH コマンドラインを使用して IPv6 OSPF 伝播を制限するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

Commands	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ipv6 OSPF	IPv6 OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
passive-interface < vlan_name >	指定した VLAN にバインドされたインターフェイスのルーティングアップデートを抑制します。

例:

```
>VTYSH
NS# configure terminal
```



```
NS(config)# router ipv6 OSPF
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

IPv6 OSPF 設定の確認

VTYSH を使用して、IPv6 OSPF の現在のネイバーと IPv6 OSPF ルートを表示します。

VTYSH コマンドラインを使用して IPv6 OSPF 設定を表示するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
sh ipv6 OSPF neighbor	現在のネイバーを表示します。
sh ipv6 OSPF route	IPv6 OSPF ルートを表示します。

例:

```
>VTYSH
NS# sh ipv6 OSPF neighbor
NS# sh ipv6 OSPF route
<!--NeedCopy-->
```

OSPFv3 認証

OSPFv3 パケットの整合性、データ発信元認証、およびデータの機密性を確保するには、OSPFv3 ピアに OSPFv3 認証を設定する必要があります。

NetScaler アプライアンスは OSPFv3 認証をサポートしており、RFC 4552 に部分的に準拠しています。OSPFv3 認証は、認証ヘッダー (AH) とカプセル化セキュリティペイロード (ESP) の 2 つの IPsec プロトコルに基づいています。NetScaler ADC アプライアンスは、OSPFv3 認証用に AH プロトコルのみをサポートします。

OSPFv3 認証では、OSPFv3 ピア間で手動で定義された IPSec Security Associations（SA; セキュリティアソシエーション）が使用され、ダイナミック SA の形成に I 手動 SA は、ピア間で使用されるセキュリティパラメータ Index（SPI）値、アルゴリズム、およびキーを定義します。手動 SA では、ピア間でネゴシエーションを行う必要がないため、両方のピアで同じ SA を定義する必要があります。

OSPFv3 認証は、VLAN または OSPFv3 エリアに対して設定できます。VLAN に対してを設定すると、VLAN のメンバーであるすべてのインターフェイスに設定が適用されます。OSPF エリアに OSPFv3 認証を設定すると、そのエリア内のすべての VLAN に設定が適用されます。この設定は、これらの VLAN のメンバーであるすべてのインターフェイスに適用されます。これらの設定は、OSPFv3 認証を直接設定したメンバ VLAN には適用されません。

NetScaler アプライアンスで OSPFv3 認証を構成する前に、次の点と制限を考慮してください。

- RFC 4552 で説明されている OSPFv3 認証のさまざまなコンポーネントを理解していることを確認します。
- OSPFv3 認証では、認証ヘッダープロトコルのみがサポートされています。セキュリティペイロード (ESP) のカプセル化はサポートされていません。
- ピアインターフェイスには、同じ設定で SA を定義する必要があります。
- 手動キーのキー再生成はサポートされていません。

VTYSH コマンドラインを使用して VLAN 上で OSPFv3 認証を設定するには、次の手順を実行します。

コマンドプロンプトで、OSPFv3 認証 VLAN コマンドの順に次のコマンドを入力します。

例:

```
1 > VTYSH NS# configure terminal
2 NS(config)# interface vlan2
3 NS(config-if)# ipv6 ospf authentication ipsec spi 256 md5 123456789
     ABCDEF0123456789ABCDEF0
4 <!--NeedCopy-->
```

VTYSH コマンドラインを使用して OSPF エリアで OSPFv3 認証を設定するには、次の手順を実行します。

コマンドプロンプトで、OSPFv3 認証 OSPF エリアコマンドの順に次のコマンドを入力します。

例:

```
1 > VTYSH NS# configure terminal
2 ns(config)#router ipv6 ospf 30
3 ns(config-router)# area 1 authentication ipsec spi 256
     md5123456789ABCDEF0123456789ABCDEF0
4 <!--NeedCopy-->
```

IPv6 OSPF のグレースフルリスタートの設定

ルーティングプロトコルが設定されている INC 以外の High Availability（HA; 高可用性）設定では、フェールオーバー後にルーティングプロトコルがコンバージされ、新しいプライマリノードと隣接ネイバールータ間のルートが学習されます。ルート学習が完了するまでに時間がかかります。この間、パケットの転送が遅れ、ネットワークパフォーマンスが低下し、パケットがドロップされる可能性があります。

グレースフルリスタートにより、フェールオーバー中の HA セットアップによって、古いプライマリノードの学習済みルートがルーティングデータベースから削除されないように隣接ルータに指示できます。新しいプライマリノードと隣接ルータは、古いプライマリノードのルーティング情報を使用して、ネットワークパフォーマンスを低下させることなく、直ちにパケットの転送を開始します。

> 注:
>
> INC モードの高可用性セットアップでは、グレースフルリスタートはサポートされていません。

VTYSH コマンドラインを使用して IPv6 OSPF のグレースフルリスタートを設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
VTYSH	VTYSH	VTYSH コマンドプロンプトに入ります。
configure terminal	NS# configure terminal	グローバル構成モードを開始します。
router-id id>	NS(config)#router-id 1.1.1.1	NetScaler ADC アプライアンスのルーター識別子を設定します。この ID は、すべてのダイナミックルーティングプロトコルに対して設定されます。HA セットアップでグレースフルリスタートが正しく機能するためには、高可用性セットアップの他のノードに同じ ID を指定する必要があります。
IPv6ospf restart grace-period <1-1800>	NS(config)# IPv6ospf restart grace-period 170	ヘルパーデバイスでルートが保持される猶予期間を秒単位で指定します。デフォルト値:120 秒。
IPv6 ospf restart helper max-grace-period <1-1800>	NS(config)# IPv6 ospf restart helper max-grace-period 180	これは、NetScaler アプライアンスがヘルパーモードになる最大猶予期間を制限するオプションコマンドです。NetScaler アプライアンスは、設定されたヘルパー最大猶予期間よりも長い猶予期間を持つ不透明な LSA を受信した場合、LSA は破棄され、NetScaler はヘルパーモードになりません。
interface	NS(config)#interface vlan3	VLAN 構成モードを開始します。
ipv6 router ospf area tag	NS(config-if)#ipv6 router ospf area 0 tag 1	VLAN 上で IPv6 OSPF ルーティングプロセスを開始します。
exit	NS(config-if)#exit	VLAN 構成モードを終了します。
router ipv6 ospf	NS(config)# router ipv6 ospf 1	IPv6 OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
capability restart graceful	NS(config-router)#capability restart graceful	IPv6 OSPF ルーティングプロセスでグレースフルリスタートを有効にします。

コマンド	例	コマンドの説明
redistribute kernel	NS(config-router)# redistribute kernel	カーネルルートを再配布します。

ISIS の構成

August 15, 2023

NetScaler アプライアンスは、中間システムから中間システム（IS-IS または ISIS）への動的ルーティングプロトコルをサポートしています。このプロトコルは、IPv4 と IPv6 のルート交換をサポートします。IS-IS はリンクステートプロトコルなので、ルーティングループが発生しにくいです。ISIS は、コンバージェンスが速く、大規模なネットワークをサポートできるという利点があるため、インターネットサービスプロバイダー (ISP) ネットワークで非常に役立ちます。

ISIS を設定するための前提条件

ISIS の設定を開始する前に、次の操作を行います。

- ISIS プロトコルを理解していることを確認してください。
- IPV6 ルートの場合は、以下を有効にします。
  - IPv6 プロトコル変換機能。
  - ISIS プロトコルを実行する VLAN の IPv6 動的ルーティングオプション。

ISIS を有効にする

次のいずれかの手順を使用して、NetScaler アプライアンスの ISIS ルーティング機能を有効にします。

CLI を使用して ISIS ルーティングを有効にするには:

コマンドプロンプトで入力します。

enable ns feature ISIS

GUI を使用して ISIS ルーティングを有効にするには:

1. [システム] > [設定] に移動し、[モードと機能] グループで [詳細機能の変更] をクリックします。
2. ISIS ルーティングオプションを選択または選択解除します。

ISIS ルーティングプロセスの作成と **VLAN** 上での起動

ISIS ルーティングプロセスを作成するには、VTYSH コマンドラインを使用する必要があります。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	説明
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードに移行します。
router ISIS [tag]	ルーティングプロセスの ISIS ルーティングプロセスおよび構成モードを作成します。
net XX….XXXX.YYYY.YYYY.YYYY.00	ルーティングプロセスの **NET** 値を指定します。ここで、·XX….XXXX はエリアアドレス **(1** ～**13** バイト**)**、·YYYY.YYYY はシステム ID (6 バイト)、·00 は **N** セレクタ **(1** バイト**)** です。
is-type (level-1|level-1-2|level-2-only)	ISIS ルーティングプロセスを指定されたルーティングレベルに設定します。Default: level-1-2.
ns IPv6-routing	IPv6 動的ルーティングデーモンを起動します。
interface	VLAN 構成モードを開始します。
ip router ISIS	IPv4 ルート交換の VLAN 上の ISIS ルーティングプロセスを有効にします。
ipv6 router ISIS	IPv6 ルート交換用の VLAN 上の ISIS ルーティングプロセスを有効にします。

例**:**

```
> VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# net 15.aabb.ccdd.0097.00
NS(config-router)# is-type level-1
NS(config-router)# exit
NS(config)# ns IPv6-routing
NS(config)# interface vlan0
NS(config-if)#  ip router isis 11
NS(config-if)# ipv6 router isis 11
<!--NeedCopy-->
```


広告ルート

ルートアドバタイズメントにより、アップストリームルーターは NetScaler アプライアンスの背後にあるネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用してルートをアドバタイズするように ISIS を設定するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	説明
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードに移行します。
router ISIS [tag]	ISIS ルーティングインスタンスを起動し、ルーティングプロセスの構成モードを開始します。
接続済み再配布 (レベル 1、レベル 1-2、レベル 2)	接続されたルートを再配布します。レベル **1:** 接続されたルートをレベル 1 に再配分、レベル 1-2: 接続されたルートをレベル 1 とレベル 2 に再配分、レベル 2**:** 接続されたルートをレベル **2** に再配布します。
カーネルの再配布 (レベル 1、レベル 1-2、レベル 2)	カーネルルートを再配布します。レベル **1:** カーネルルートをレベル 1 に再配布、レベル 1-2: カーネルルートをレベル 1 とレベル 2 に再配布、レベル 2**:** カーネルルートをレベル **2** に再配布します。

例**:**

```
1 >VTYSH
2 NS# configure terminal
3 NS(config)# router isis 11
4 NS(config-router)# redistribute connected level-1
5 NS(config-router)# redistribute kernel level-1
6 <!--NeedCopy-->
```

ISIS プロパゲーションの制限

設定をトラブルシューティングする必要がある場合は、任意の VLAN にリッスン専用モードを設定できます。

VTYSH コマンドラインを使用して ISIS の伝播を制限するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	説明
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードに移行します。
router isis [tag]	ルーティングプロセスの構成モードを開始します。
passive-interface	指定した VLAN にバインドされたインターフェイスのルーティングアップデートを抑制します。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

ISIS 設定の検証

VTYSH を使用すると、指定した VLAN の ISIS ルーティングテーブルと ISIS 情報を表示できます。

VTYSH コマンドラインを使用して ISIS 設定を表示するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

Commands	説明
VTYSH	VTYSH コマンドプロンプトを表示します。
show ip isis route	更新された IPv4 ISIS ルーティングテーブルを表示します。
show ipv6 isis route	更新された IPv6 ISIS ルーティングテーブルを表示します。
sh isis interface	指定された VLAN の IPv6 ISIS 情報を表示します。

例:

```
NS# VTYSH
NS#  show ip isis route
NS#  show ipv6 isis route
NS# sh isis interface VLAN0
<!--NeedCopy-->
```


NetScaler ルーティングテーブルへのルートのインストール

August 15, 2023

NetScaler ADC アプライアンスは、アプライアンスのルーティングテーブルにルートをインストールした後、さまざまなルーティングプロトコルによって学習されたルートを使用できます。

VTYSH コマンドラインを使用して内部ルーティングテーブルにさまざまなルートをインストールするには:

CLI で、インストールするルートに応じて次のコマンドを入力します。

Commands	指定
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
ns route-install Default	IPv4 デフォルトルートを内部ルーティングテーブルにインストールします。
ns route-install RIP	IPv4 RIP 固有のルートを内部ルーティングテーブルにインストールします。
ns route-install BGP	IPv4 BGP 固有のルートを内部ルーティングテーブルにインストールします。
ns route-install OSPF	IPv4 OSPF 固有のルートを内部ルーティングテーブルにインストールします。
ns route-install IPv6 Default	IPv6 デフォルトルートを内部ルーティングテーブルにインストールします。
ns route-install IPv6 RIP	IPv6 RIP 固有のルートを内部ルーティングテーブルにインストールします。
ns route-install IPv6 BGP	IPv6 BGP 固有のルートを内部ルーティングテーブルにインストールします。
ns route-install IPv6 OSPF	IPv6 OSPF 固有のルートを内部ルーティングテーブルにインストールします。

例**:**

```
>VTYSH
NS# configure terminal
NS# ns route-install Default
NS(config)# ns route-install RIP
NS(config)# ns route-install BGP
NS(config)# ns route-install OSPF
NS# ns route-install IPv6 Default
NS(config)# ns route-install IPv6 RIP
NS(config)# ns route-install IPv6 BGP
```



```
NS(config)# ns route-install IPv6 OSPF
<!--NeedCopy-->
```

NetScaler ADC アプライアンスでサポートされる ECMP ルートの最大数

NetScaler ADC アプライアンスでは、最大 32 の ECMP（等価コストマルチパス）ルートがサポートされます。ルートの選択は 5 つのタプルに基づいています。詳細については、「5 つのタプルに基づくルート選択」を参照してください。

選択エリアへの SNIP および VIP ルートのアドバタイズ

August 15, 2023

一部の SNIP アドレスを特定の領域にアドバタイズするには、DRADV モードを有効にしたり、接続 ZebOS 操作を再配布したりすることはできません。これは、これらの操作が接続されたすべてのルートを ZebOS に送信するためです。また、必要なサブネットにダミーの静的ルートを ZebOS に追加したり、ZebOS に ACL を追加して不要な接続ルートをフィルタリングしたりするのは、面倒で面倒な作業です。

ネットワークルートとタグオプションはこの問題を解決します。ネットワークルートオプションを有効にできるのは、サブネットごとに 1 つの SNIP アドレスだけです。その SNIP アドレスの接続ルートは、カーネルルートとして ZebOS に送信されます。

VIP アドレスと SNIP アドレスの場合、タグには 1 から 4294967295 までの整数を割り当てることができます。このパラメータは、ホストルートまたはネットワークルートが VIP アドレスまたは SNIP アドレスで有効になっている場合にのみ設定できます。VIP アドレスと SNIP アドレスに関連付けられたタグ値も、そのルートとともに ZeBos に送信されます。VIP ルートと SNIP ルートには、異なる値のタグを設定できます。その後、これらのタグ値を ZeBOS のルートマップで照合し、特定のエリアにアドバタイズできます。

選択エリアへの SNIP ルートのアドバタイズ

CLI を使用して SNIP アドレスのネットワークルートとタグパラメータを設定するには:

コマンドプロンプトで入力します。

- 新しい SNIP アドレスを追加する場合:
  - **add ns ip** <IPAddress>@ <netmask> -**type SNIP** -**networkroute** (**ENABLED** | ** 無効 **)-** タグ ** <positive_integer>
  - **show ns ip** <IPAddress>
- 既存の SNIP アドレスを再設定する場合:

- **set ns ip** <IPAddress>@ <netmask> -**type SNIP** - **networkroute** (**ENABLED** | ** 無効 **)-** タグ ** <positive_integer>
- **show ns ip** <IPAddress>

GUI を使用して SNIP アドレスのネットワークルートとタグパラメータを設定するには:

1. [システム] > [ネットワーク] > [**IP] > [IPV4**] に移動します。
2. サブネット IP (SNIP) アドレスを追加したり、既存のサブネット IP アドレスを変更したりするときに、ネットワークルート と タグ パラメータを設定します。

特定のエリアへの **VIP** ルートをアドバタイズする

CLI を使用して VIP アドレスのホストルートとタグパラメータを設定するには:

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

- 新しい VIP アドレスを追加する場合:
  - **add ns ip** <IPAddress>@ <netmask> -**type VIP** -**hostRoute** (**ENABLED** | ** 無効 **)-** タグ ** <positive_integer>
  - **show ns ip** <IPAddress>
- 既存の VIP アドレスを再設定する場合:
  - **set ns ip** <IPAddress>@ <netmask> -**type VIP** -**hostRoute** (**ENABLED** | ** 無効 **)-** タグ ** <positive_integer>
  - **show ns ip** <IPAddress>

GUI を使用して VIP アドレスのネットワークルートとタグパラメータを設定するには:

1. **System** > **Network** > **IPs** > **IPV4s** に移動します。
2. VIP アドレスを追加したり、既存の VIP アドレスを変更したりするときに、**Host Route** パラメータと **Tag** パラメータを設定します。

双方向転送検出の構成

August 15, 2023

双方向転送検知 (BFD) プロトコルは、転送パスの障害を迅速に検出するためのメカニズムです。BFD はパス障害をミリ秒単位で検出します。BFD はダイナミックルーティングプロトコルで使用されます。

BFD の動作では、ルーティングピアはネゴシエートされた間隔で BFD パケットを交換します。ネゴシエートされた間隔と猶予間隔を足した時間内にピアからパケットが受信されなかった場合、そのピアは停止していると見なされ、

登録されたルーティングプロトコルのセットに通知が送信されます。次に、ルーティングプロトコルはベストパスを再計算し、ルーティングテーブルを再プログラムします。BFD は、ルーティングプロトコルが提供するタイマーよりも短い時間間隔をサポートしているため、障害をより迅速に検出できます。

NetScaler アプライアンスは、BGP（IPv4 および IPv6）、OSPFv2（IPv4）、および OSPFv3（IPv6）のルーティングプロトコルの BFD をサポートしています。NetScaler アプライアンスの BFD サポートは、RFC 5880、5881、および 5883 に準拠しています。

双方向転送検出の設定に関する考慮事項

BFD の設定を開始する前に、次の点を考慮してください。

- RFC 5880、5881、5883 で説明されている BFD のさまざまなコンポーネントについて理解していることを確認してください。
- NetScaler アプライアンスの BFD は、次のルーティングプロトコルでサポートされています。
  - BGP (IPv4 および IPv6)
  - OSPFv2 (IPv4)
  - OSPFv3 (IPv6)
- NetScaler アプライアンスの BFD は、次のルーティングプロトコルではサポートされていません。
  - ISIS
  - RIP (IPv4)
  - RIPng (IPv6)
- 次の BFD 機能は NetScaler アプライアンスではサポートされていません。
  - BFD エコーモード
  - BFD 認証
  - BFD デマンド非同期モード
- BFD 間隔と BFD Rx タイマーの最小値は 100 ミリ秒です。
- BFD を共有 IP アドレスを使用するトポロジー（SNIP アドレスを使用するレイヤ 2 高可用性設定やストライピング IP アドレスを使用するクラスタ設定など）で使用すると、BFD 障害検出時間（ミリ秒単位）が HA フェールオーバー検出間隔（3 ～4 秒）よりも短いため、BFD はフェールオーバー中にアクティブセッションを停止させます。そのため、Citrix では、フェイルオーバー処理中もルートが保持されるため、レイヤー 2 HA トポロジではグレースフルリスタートを使用することを推奨しています。

構成の手順

NetScaler アプライアンスでの BFD の設定は、次のタスクで構成されます。

- BFD パラメータの設定
- ダイナミックルーティングプロトコルの BFD サポートの設定

BFD パラメータの設定

NetScaler アプライアンスは、シングルホップセッション、IPv4 マルチホップセッション、および IPv6 マルチホップセッションに個別の BFD セッションパラメーターを提供します。あるタイプのセッションに BFD パラメータを設定しない場合、そのセッションにはデフォルト値が適用されます。

各 BFD パラメータのデフォルト値は、シングルホップセッション、IPv4 マルチホップセッション、および IPv6 マルチホップセッションで同じです。次の表は、各 BFD パラメータのデフォルト値を示しています。

BFD パラメータ名	デフォルト値
Interval	750 ミリ秒
最小 Rx	500 ミリ秒
マルチプライヤー	3

> 重要:
>
> NetScaler ADC アプライアンスの MellanoxNIC の初期化には約 1500 ミリ秒かかります。MellanoxNIC を搭載した NetScaler アプライアンスの場合は、BFD タイマーを 1500 ミリ秒以上に設定する必要があります。Citrix では、BFD タイマーを 3000 ミリ秒に設定することを推奨しています。
>
> - 送信間隔 = 600 ミリ秒
> - 最小レックス = 600 ミリ秒
> - マルチプラー = 5

シングルホップセッションの BFD パラメータの設定

VTYSH コマンドラインを使用してシングルホップセッションの BFD パラメータを設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	指定
`vtysh`	VTYSH コマンドプロンプトを表示します。
`configure terminal`	グローバル構成モードを開始します。
`interface vlan ID>`	インターフェイス構成モードに入ります。
`bfd singlehop-peer interval <num> minrx <num> multiplier <num>`	指定したインターフェイスの BFD パラメータを設定します。

設定例:


```
> vtysh

ns# configure terminal

ns(config)# interface vlan3

ns(config-if)# bfd singlehop-peer interval 200 minrx 200 multiplier 5

ns(config-if)# exit
<!--NeedCopy-->
```

IPv4 マルチホップセッションの **BFD** パラメータの設定

VTYSH コマンドラインを使用して IPv4 マルチホップセッションの BFD パラメータを設定するには、コマンドプロンプトで次のコマンドを順番に入力します。

コマンド	指定
`vtysh`	VTYSH コマンドプロンプトを表示します。
`configure terminal`	グローバル構成モードを開始します。
`bfd multihop-peer <ipv4addr> interval <num> minrx <num> multiplier <num>`	IPv4 マルチホップセッションの BFD パラメータを設定します。

設定例:

```
    > vtysh

    ns# configure terminal

    ns(config)# bfd multihop-peer 20.20.20.138 interval 300 minrx 300
       multiplier 5

    ns(config)# exit
<!--NeedCopy-->
```

IPv6 マルチホップセッションの **BFD** パラメータの設定

VTYSH コマンドラインを使用して IPv6 マルチホップセッションの BFD パラメータを設定するには、コマンドプロンプトで次のコマンドを順番に入力します。

コマンド	指定
vtysh	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
bfd multihop-peer ipv6 <ipv6addr> interval <num> minrx <num> multiplier <num>	IPv6 マルチホップセッションの BFD パラメータを設定します。

設定例:

```
1      > vtysh
2
3      ns(config)# bfd multihop-peer ipv6 20fe:125::138 interval 500 minrx
           500 multiplier 5
4
5      ns(config)# exit
6  <!--NeedCopy-->
```

ダイナミックルーティングプロトコルの **BFD** サポートの設定

ピアとのセッションの種類に応じて、ダイナミックルーティングプロトコルの BFD を有効にできます。たとえば、シングルホップとマルチホップ。NetScaler アプライアンスは、関連する BFD パラメーター設定をセッションに適用します。

IPv4 BGP シングルホップセッションの **BFD** の設定

VTYSH コマンドラインを使用して IPv4 BGP シングルホップセッションの BFD を設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	指定
vtysh	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router bgp <asnumber>	BGP 自律システム。asnumber は必須のパラメータです。
neighbor <ipv4addr> remote-as <num>	IPv4 BGP テーブルを、指定された自律システム内のネイバーの IPv4 アドレスで更新します。
neighbor <ipv4addr> fall-over bfd	指定されたネイバーの BFD を有効にします。

設定例:

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)#router bgp 1
6
7      ns(config-router)#neighbor 20.20.20.138 remote-as 1
8
9      ns(config-router)#neighbor 20.20.20.138 fall-over bfd
10
11     ns(config-router)#redistribute kernel
12
13     ns(config-router)#exit
14 <!--NeedCopy-->
```

IPv4 BGP マルチホップセッションの **BFD** の設定

VTYSH コマンドラインを使用して IPv4 BGP マルチホップセッションの BFD を設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	指定
vtysh	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router bgp <asnumber>	BGP 自律システム。asnumber は必須のパラメータです。
neighbor <ipv4addr> remote-as <num>	IPv4 BGP テーブルを、指定された自律システム内のネイバーの IPv4 アドレスで更新します。
neighbor <ipv4addr> fall-over bfd multihop	指定されたネイバーの BFD を有効にします。

設定例:

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)#router bgp 1
6
7      ns(config-router)#neighbor 20.20.20.138 remote-as 1
8
9      ns(config-router)#neighbor 20.20.20.138 fall-over bfd multihop
10
```



```
11      ns(config-router)#redistribute kernel
12
13      ns(config-router)#exit
14  <!--NeedCopy-->
```

IPv6 BGP シングルホップセッションの **BFD** の設定

VTYSH コマンドラインを使用して IPv6 BGP シングルホップセッションの BFD を設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	指定
vtysh	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router bgp <asnumber>	BGP 自律システム。asnumber は必須のパラメータです。
neighbor <ipv6addr> remote-as <num>	IPv6 BGP テーブルを、指定された自律システム内のネイバーのリンクローカル IPv6 アドレスで更新します。
neighbor <ipv6addr> fall-over bfd	指定されたネイバーの BFD を有効にします。
address-family ipv6	アドレスファミリ構成モードを開始します。
neighbor <ipv6addr> activate	リンクローカルアドレスを使用して、ピアとローカルノード間で IPv6 ルーターファミリのプレフィクスを交換します。

設定例:

```
1  > vtysh
2
3  ns# configure terminal ns(config)#router bgp 1
4
5  ns(config-router)#neighbor 30fe:123::124 remote-as 1
6
7  ns(config-router)#neighbor 30fe:123::124 fall-over bfd
8
9  ns(config-router)#address-family ipv6
10
11 ns(config-router-af)#neighbor 30fe:123::124 activate
12
13 ns(config-router-af)#redistribute kernel
14
15 ns(config-router-af)#exit
16
17 <!--NeedCopy-->
```


IPv6 BGP マルチホップセッションの **BFD** の設定

VTYSH コマンドラインを使用して IPv6 BGP マルチホップセッションの BFD を設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	指定
vtysh	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router bgp <asnumber>	BGP 自律システム。asnumber は必須のパラメータです。
neighbor <ipv6addr> remote-as <num>	IPv6 BGP テーブルを、指定された自律システム内のネイバーのリンクローカル IPv6 アドレスで更新します。
neighbor <ipv6addr> fall-over bfd multihop	指定されたネイバーの BFD を有効にします。
address-family ipv6	アドレスファミリ構成モードを開始します。
neighbor <ipv6addr> activate	リンクローカルアドレスを使用して、ピアとローカルノード間で IPv6 ルーターファミリーのプレフィックスを交換します。

設定例:

```
> vtysh

ns# configure terminal

ns(config)# bfd multihop-peer ipv6 20fe:125::138 interval 500 minrx 500
    multiplier 5

ns(config)#router bgp 1

ns(config-router)#neighbor 20fe:125::138 remote-as 1

ns(config-router)#neighbor 20fe:125::138 fall-over bfd multihop

ns(config-router)#address-family ipv6

ns(config-router-af)#neighbor 20fe:125::138 activate

ns(config-router-af)#redistribute kernel

ns(config-router-af)#end

<!--NeedCopy-->
```


インターフェイスでの **OSPFv2**（**IPv4**）の **BFD** の設定

BFD は、OSPFv2 プロトコルを使用するすべてのインターフェイスまたは特定のインターフェイスで有効にできます。

VTYSHコマンドラインを使用してすべてのインターフェイスで **OSPFv2** の **BFD** を設定するには：

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
vtysh	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ospf <process tag>	OSPFv2 構成モードを開始します。
bfd all-interfaces	OSPFv2 を使用するすべてのインターフェイスで BFD を有効にします。

設定例：

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)#router ospf 1
6
7      ns(config-router)#bfd all-interfaces
8
9      ns(config-router)#redistribute kernel
10
11     ns(config-router)#exit
12 <!--NeedCopy-->
```

VTYSHコマンドラインを使用して特定のインターフェイスで **OSPFv2** の **BFD** を設定するには：

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
vtysh	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
interface <vlan ID>	インターフェイス構成モードに入ります。
ip ospf bfd	OSPFv2 を使用する指定のインターフェイスで BFD を有効にします。

設定例：


```
    > vtysh

    ns# configure terminal

    ns(config)# interface vlan5

    ns(config-if)# ip ospf bfd

    ns(config-if)# exit
<!--NeedCopy-->
```

インターフェイスでの **OSPFv3**（**IPv6**）の **BFD** の設定

BFD は、OSPFv3 プロトコルを使用するすべてのインターフェイスまたは特定のインターフェイスで有効にできます。

VTYSHコマンドラインを使用してすべてのインターフェイスで **OSPFv3** の **BFD** を設定するには：

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
`vtysh`	VTYSH コマンドプロンプトを表示します。
`configure terminal`	グローバル構成モードを開始します。
`router ipv6 ospf <process tag>`	OSPFv3 構成モードを開始します。
`bfd all-interfaces`	OSPFv3 を使用するすべてのインターフェイスで BFD を有効にします。

設定例：

```
    > vtysh

    ns# configure terminal

    ns(config)#router ipv6 ospf 10

    ns(config-router)#bfd all-interfaces

    ns(config-router)#redistribute kernel

    ns(config-router)#exit
<!--NeedCopy-->
```

VTYSHコマンドラインを使用して特定のインターフェイスに **OSPFv3** の **BFD** を設定するには：

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	指定
vtysh	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
interface <vlan ID>	インターフェイス構成モードに入ります。
ipv6 ospf bfd	OSPFv3 を使用する指定のインターフェイスで BFD を有効にします。

設定例:

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)# interface vlan15
6
7      ns(config-if)# ipv6 ospf bfd
8
9      ns(config-if)# exit
10 <!--NeedCopy-->
```

静的ルートの構成

August 15, 2023

スタティックルートはネットワークのパフォーマンスを向上させるために手動で作成されます。スタティックルートを監視してサービスの中断を防ぐことができます。また、ECMP ルートに重みを割り当てたり、ヌルルートを作成してルーティングループを防ぐこともできます。

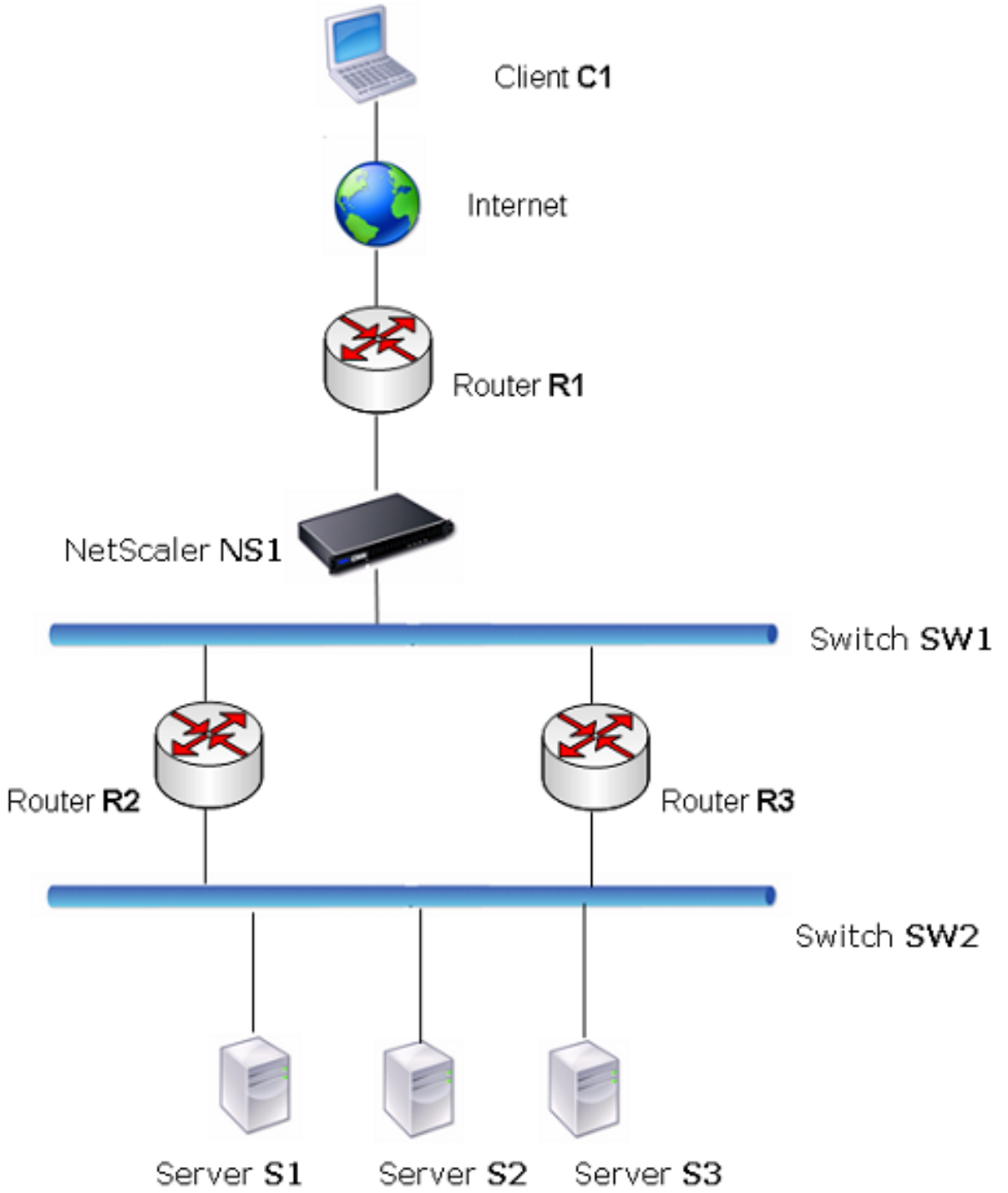
監視対象のスタティックルート。手動で作成した（静的）ルートがダウンしても、バックアップルートは自動的にアクティブ化されません。非アクティブなプライマリスタティックルートは手動で削除する必要があります。ただし、静的ルートを監視対象ルートとして構成すると、NetScaler アプライアンスは自動的にバックアップルートをアクティブ化できます。

スタティックルートモニタリングは、サブネットのアクセシビリティに基づいて行うこともできます。サブネットは通常 1 つのインターフェイスに接続されますが、他のインターフェイスを介して論理的にアクセスすることもできます。VLAN にバインドされたサブネットには、VLAN が稼働している場合にのみアクセスできます。VLAN は、NetScaler がパケットを送受信するための論理インターフェイスです。ネクストホップが到達不能なサブネットにある場合、スタティックルートは DOWN とマークされます。

注**:** 高可用性 (HA) セットアップでは、セカンダリノードの監視対象ステートルート (MSR) のデフォルト値は UP です。この値は、フェールオーバー時に状態遷移のギャップが生じないように設定されます。その結果、それらのルー

トでパケットがドロップされる可能性があります。

NetScaler が複数のサーバーにまたがるサイトへのトラフィックの負荷分散を行う、次の単純なトポロジーを考えてみましょう。



ルーター R1 は、クライアントと NetScaler アプライアンスの間でトラフィックを移動します。アプライアンスは、ルータ R2 または R3 を介してサーバ S1 と S2 にアクセスできます。サーバーのサブネットに到達するための 2 つ

の静的ルートがあります。1 つは R2 をゲートウェイ、もう 1 つは R3 をゲートウェイとして使用します。これらのルートは両方ともモニタリングが有効になっています。ゲートウェイ R2 のスタティックルートのアドミニストレーティブディスタンスは、ゲートウェイ R3 を使用するスタティックルートのアドミニストレーティブディスタンスよりも低くなります。そのため、トラフィックをサーバに転送するには、R3 よりも R2 の方が優先されます。また、NetScaler のデフォルトルートは R1 を指しているため、すべてのインターネットトラフィックは正常に出ます。

R2 をゲートウェイとして使用する静的ルートで監視が有効になっているときに R2 に障害が発生すると、NetScaler はそのルートを DOWN とマークします。NetScaler は R3 の静的ルートをゲートウェイとして使用し、トラフィックを R3 経由でサーバーに転送するようになりました。

NetScaler は、IPv4 および IPv6 の静的ルートの監視をサポートしています。新しい ARP モニターまたは PING モニターを作成するか、既存の ARP または PING モニターを使用して、IPv4 静的ルートを監視するように NetScaler を構成できます。IPv6（ND6）または PING モニター用の新しいネイバーディスカバリーを作成するか、既存の ND6 または PING モニターを使用して、IPv6 静的ルートを監視するように NetScaler を構成できます。

加重スタティックルート。NetScaler アプライアンスは、距離とコストが等しいルート、つまり Equal Cost Multi-Path（ECMP）ルートを含むルーティングを決定する場合、送信元 IP アドレスと宛先 IP アドレスに基づくハッシュメカニズムを使用して、ルート間の負荷を分散します。ただし、ECMP ルートの場合は、重み値を設定できます。次に、NetScaler は重みとハッシュ値の両方を使用して負荷を分散します。

Null Routes. ルーティング決定で選択したルートが非アクティブの場合、NetScaler アプライアンスはバックアップルートを選択します。すべてのバックアップルートにアクセスできなくなると、アプライアンスがパケットを送信者に再ルーティングする可能性があり、その結果、ルーティングループが発生してネットワークが混雑する可能性があります。このような状況を防ぐには、ヌルルートを作成して、ゲートウェイとしてヌルインターフェイスを追加します。ヌルルートは、他のスタティックルートよりもアドミニストレーティブディスタンスが高いため、決して推奨ルートにはなりません。ただし、他のスタティックルートにアクセスできなくなった場合は選択されます。その場合、アプライアンスはパケットをドロップし、ルーティングループを防ぎます。

IPv4 スタティックルートの設定

いくつかのパラメータを設定して単純なスタティックルートまたはヌルルートを追加することも、追加のパラメータを設定してモニタリング対象または監視対象および加重スタティックルートを設定することもできます。スタティックルートのパラメータは変更できます。たとえば、重み付けされていないルートに重みを割り当てたい場合や、監視対象ルートの監視を無効にしたい場合があります。

CLI のプロシージャ

CLI を使用して静的ルートを作成するには:

コマンドプロンプトで入力します。

- add route <network> <netmask> <gateway>[-cost \<positive_integer>] [-advertise (DISABLED | 有効)]

- show route [\<network> \<netmask> [\<gateway>]] [\<routeType>] [-detail]

例:

```
1 > add route 10.102.29.0 255.255.255.0 10.102.29.2 -cost 2 -advertise
     ENABLED
2  Done
3 <!--NeedCopy-->
```

CLI を使用して監視対象のスタティックルートを作成するには:

コマンドプロンプトで次のコマンドを入力して、監視対象のスタティックルートを作成し、設定を確認します。

- add route <network> <netmask> <gateway> [-distance \<positive_integer>] [-weight \<positive_integer>][-msr (ENABLED | DISABLED) [-monitor \]]
- show route [\<network> \<netmask> [\<gateway>]] [\<routeType>] [-detail]

例:

```
1 > add route 10.102.29.0 255.255.255.0 10.102.29.3 -distance 5 -weight 6
     -msr ENBLED -monitor PING
2  Done
3 <!--NeedCopy-->
```

CLI を使用してヌルルートを作成するには:

コマンドプロンプトで次のように入力します。

- add route <network> <netmask> null
- show route <network> <netmask>

例:

```
1 > add route 10.102.29.0 255.255.255.0 null
2  Done
3 <!--NeedCopy-->
```

CLI を使用してスタティックルートを削除するには:

コマンドプロンプトで入力します。

rm route <network> <netmask> <gateway>

例:

```
1 > rm route 10.102.29.0 255.255.255.0 10.102.29.3
2  Done
3 <!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用してスタティックルートを設定するには:

[システム] > [ネットワーク] > [ルート] に移動し、[基本] タブで新しい静的ルートを追加するか、既存の静的ルートを編集します。

GUI を使用してルートを削除するには:

[システム] > [ネットワーク] > [ルート] に移動し、[基本] タブで静的ルートを削除します。

IPv6 スタティックルートの設定

最大 6 つのデフォルト IPv6 静的ルートを設定できます。IPv6 ルートは、宛先デバイスの MAC アドレスに到達可能かどうかに基づいて選択されます。これは IPv6 ネイバーディスカバリ機能を使用して判断できます。ルートは負荷分散され、送信元/宛先ベースのハッシュメカニズムのみが使用されます。そのため、ラウンドロビンなどのルート選択メカニズムはサポートされていません。デフォルトルートのネクストホップアドレスは、NSIP サブネットに属している必要はありません。

CLI のプロシージャ

CLI を使用して IPv6 ルートを作成するには:

コマンドプロンプトで、次のコマンドを入力して IPv6 ルートを作成し、構成を確認します。

- add route6 <network> <gateway> [-vlan \<positive_integer>]
- show route6 [\<network> [\<gateway>]

例:

```
1 > add route6 ::/0 FE80::67 -vlan 5
2  Done
3 <!--NeedCopy-->
```

CLI を使用して監視対象の IPv6 静的ルートを作成するには:

コマンドプロンプトで、次のコマンドを入力して、監視対象の IPv6 静的ルートを作成し、構成を確認します。

- add route6 <network> <gateway> [-msr (ENABLED | DISABLED) [-monitor \]
- show route6 [\<network> [\<gateway>]

例:

```
1 > add route6 ::/0 2004::1 -msr ENABLED -monitor PING
2  Done
3 <!--NeedCopy-->
```

CLI を使用して IPv6 ルートを削除するには:

コマンドプロンプトで入力します。

rm route6 <network> <gateway>

例:

```
> rm route6 ::/0 FE80::67
 Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して IPv6 ルートを設定するには:

[システム] > [ネットワーク] > [ルート] に移動し、[IPV6] タブで新しい IPv6 ルートを追加するか、既存の IPv6 ルートを編集します。

GUI を使用して IPv6 ルートを削除するには:

[システム] > [ネットワーク] > [ルート] に移動し、[IPV6] タブで IPv6 ルートを削除します。

仮想サーバー設定に基づくルートヘルスインジェクション

August 15, 2023

VIP アドレスのルートをアドバタイズする NetScaler アプライアンスのルートヘルスインジェクション（RHI）機能を制御するために、次のオプションとパラメーターが導入されました。

- **VSVR_CNTRLD**. これは VIP アドレスの (Vserver RHI レベル) パラメータのオプションです。このオプションを Vserver RHI Level パラメータに設定すると、VIP アドレスのルートをアドバタイズするための RHI の動作は、その VIP アドレスに関連するすべての仮想サーバとその状態の RHI STATE パラメータ設定によって異なります。
- **RHI STATE**. 仮想サーバーのパラメーターです。RHI STATE パラメータは、パッシブまたはアクティブのいずれかに設定できます。デフォルトでは、RHI STATE パラメータはパッシブに設定されています。

VIP アドレスの場合、RHI (Vserver RHI レベル) パラメータが VSVR_CNTRLD に設定されている場合、その VIP アドレスに関連付けられた仮想サーバの RHI STATE 設定に基づいて、VIP アドレスの RHI の動作が次のように異なります。

- すべての仮想サーバーで RHI STATE を PASSIVE に設定すると、NetScaler は常に VIP アドレスのルートをアドバタイズします。
- すべての仮想サーバーで RHI STATE を ACTIVE に設定した場合、関連する仮想サーバーの少なくとも 1 つが稼働状態の場合、NetScaler は VIP アドレスのルートをアドバタイズします。
- RHI STATE を ACTIVE に設定し、他の仮想サーバーを PASSIVE に設定した場合、NetScaler は、RHI STATE が ACTIVE に設定されている関連仮想サーバーの少なくとも 1 つが稼働状態であれば、VIP アドレスのルートをアドバタイズします。

次の表は、VIP アドレスに関連付けられた仮想サーバーの RHI STATE 設定に基づく、VIP アドレスの RHI 動作の例を示しています。NetScaler アプライアンスには、VIP アドレスに関連付けられた 2 つの仮想サーバー V1 と V2 があります。

VIP に関連する仮想サーバー	ステート 1	ステート 2	ステート 3	ステート 4
すべての仮想サーバーで **RHI** ステートが **PASSIVE** に設定されている				
V1	上へ	上へ	DOWN	DOWN
V2	上へ	DOWN	上へ	DOWN
この VIP アドレスのルートを宣伝しますか?	はい	はい	はい	はい
すべての仮想サーバーで **RHI** 状態が **ACTIVE** に設定されている				
V1	上へ	上へ	DOWN	DOWN
V2	上へ	DOWN	上へ	DOWN
この VIP アドレスのルートを宣伝しますか?	はい	はい	はい	いいえ
一方の仮想サーバーでは **RHI** の状態が **ACTIVE** に設定され、もう一方の仮想サーバーでは **PASSIVE** に設定されています				
V1 (RHI 状態 = アクティブ)	上へ	上へ	DOWN	DOWN
V2 (RHI ステート = パッシブ)	上へ	DOWN	上へ	DOWN

この VIP アドレスのルートを宣伝しますか?	はい	はい	いいえ	いいえ

関連する仮想サーバーの RHI (RHI State) パラメータ設定に基づいて VIP アドレスの RHI を設定するには、次の手順を実行します。

- VIP アドレスの RHI (仮想サーバ RHI レベル) パラメータを VSVR_CNTRLD に設定します。
- VIP アドレスに関連付けられた各仮想サーバーの RHI State パラメータを設定します。

CLI を使用して VIP アドレスの仮想サーバー RHI レベルを設定するには:

コマンドプロンプトで入力します。

- **set ns ip** <IPAddress> [-**vserverRHILevel** \<vserverRHILevel>]

CLI を使用して仮想サーバの RHI State パラメータを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set lb vserver** <name> [-**RHIstate** (**PASSIVE** | **ACTIVE**)]

GUI を使用して VIP アドレスの vServer RHI レベルを設定するには

1. [システム] > [ネットワーク] > [**IP**] に移動します。
2. VIP アドレスを選択し、[編集] をクリックします。
3. 仮想サーバの **RHI** レベルパラメータを **VSVR_CNTRLD** に設定し、「**OK**」をクリックします。

GUI を使用して仮想サーバーの RHI State パラメータを設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. **RHI State** パラメータを設定し、**OK** をクリックします。

ポリシーベースルートの構成

August 15, 2023

ポリシーベースのルーティングは、指定した基準に基づいてルーティングを決定します。ポリシーベースルート（PBR）は、パケットの選択基準と、通常は選択したパケットの送信先となるネクストホップを指定します。たとえば、特定の IP アドレスまたは範囲からの送信パケットを特定のネクストホップルーターにルーティングするように NetScaler アプライアンスを構成できます。各パケットは、一致するものが見つかるまで、指定された優先順位で決定された順序で、設定された各 PBR と照合されます。一致するものが見つからない場合、または一致する PBR が

DENY アクションを指定している場合、NetScaler は通常の宛先ベースのルーティングにルーティングテーブルを適用します。

PBR は、送信元 IP アドレス、送信元ポート、宛先 IP アドレス、宛先ポート、プロトコル、送信元 MAC アドレスなどのパラメータに基づいてデータパケットのルーティングを決定します。PBR は、NetScaler がパケットをルーティングするためにパケットが満たす必要がある条件を定義します。これらのアクションは「処理モード」と呼ばれます。処理モードは以下のとおりです。

- 許す。アプライアンスは、指定されたネクストホップルーターにパケットを送信します。
- 拒否。NetScaler は、通常の宛先ベースのルーティングにルーティングテーブルを適用します。

IPv4 および IPv6 の送信トラフィック用の PBR を作成できます。

多くのユーザーは、まず PBR を作成してから変更します。新しい PBR をアクティブ化するには、それを適用する必要があります。PBR を非アクティブ化するには、削除するか無効にすることができます。PBR のプライオリティ番号を変更して、優先順位を高くしたり低くしたりできます。

IPv4 トラフィックのポリシーベースルート (PBR)

August 15, 2023

PBR の設定には次のタスクが含まれます。

- PBR を作成します。
- PBR を適用してください。
- （オプション）PBR を無効または有効にします。
- （オプション）PBR のプライオリティの番号を変更します。

PBR の作成または変更

同じパラメータで 2 つの PBR を作成することはできません。複製を作成しようとすると、エラーメッセージが表示されます。

PBR のプライオリティを設定できます。優先度（整数値）は、NetScaler アプライアンスが PBR を評価する順序を定義します。優先度を指定せずに PBR を作成すると、NetScaler は自動的に 10 の倍数の優先度を割り当てます。

パケットが PBR で定義された条件と一致すると、NetScaler はアクションを実行します。パケットが PBR で定義された条件と一致しない場合、NetScaler はそのパケットを次に優先順位の高い PBR と比較します。

選択したパケットをネクストホップルータに送信する代わりに、複数のネクストホップをバインドしたリンクロードバランシング仮想サーバにパケットを送信するように PBR を設定できます。この設定は、ネクストホップリンクに障害が発生した場合のバックアップとなります。

次の例を考えてみましょう。2 つの PBR（p1 と p2）が NetScaler 上に構成され、優先順位 20 と 30 が自動的に割り当てられます。最初の PBR p1 の直後に評価対象となる 3 番目の PBR p3 を追加する必要があります。新しい PBR p3 の優先順位は 20 ～30 でなければなりません。この場合、優先度を 25 に指定できます。

CLI のプロシージャ

CLI を使用して PBR を作成するには:

コマンドプロンプトで入力します。

- add ns pbr <name> <action> [-srcIP [\] \] \[-srcPort \[\\] \] \[-destIP \[\\] \] \[-destPort \[\\] \] \[-nextHop \\] \[-srcMac \\<mac_addr>] \[-protocol \ |-protocolNumber \<positive_integer>] \[-vlan \\<positive_integer>] \[-interface \\<interface_name>] \[-priority \\<positive_integer>] \[-msr \(ENABLED | DISABLED) \[-monitor \\]] \[-state \(ENABLED | DISABLED)]
- show ns pbr

例**:**

```
> add ns pbr pbr1  allow -srcip 10.102.37.252 -destip 10.10.10.2 -
   nexthop 10.102.29.77
 Done
<!--NeedCopy-->
```

CLI を使用して PBR の優先度を変更するには:

コマンドプロンプトで次のコマンドを入力して、優先度を変更し、構成を確認します。

- set ns pbr <name> [-action (ALLOW | DENY)] [-srcIP [\] \] \[-srcPort \[\\] \] \[-destIP \[\\] \] \[-destPort \[\\] \] \[-nextHop \\] \[-srcMac \\<mac_addr>] \[-protocol \ | -protocolNumber \<positive_integer>] \[-vlan \\<positive_integer>] \[-interface \\<interface_name>] \[-priority \\<positive_integer>] \[-msr \(ENABLED | DISABLED) \[-monitor \\]] \[-state \(ENABLED | DISABLED)]
- show ns pbr [\<name>]

例**:**

```
> set ns pbr pbr1 -priority 23
 Done
<!--NeedCopy-->
```

CLI を使用して 1 つまたはすべての PBR を削除するには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- rm ns pbr <name>
- clear ns pbrs

例**:**


```
> rm ns pbr pbr1
 Done

> clear ns PBRs
 Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して PBR を作成するには:

[システム] > [ネットワーク] > [PBR] に移動し、[PBR] タブで新しい PBR を追加するか、既存の PBR を編集します。

GUI を使用して 1 つまたはすべての PBR を削除するには:

[システム] > [ネットワーク] > [PBR] に移動し、[PBR] タブで PBR を削除します。

PBR を適用する

PBR をアクティブ化するには、PBR を適用する必要があります。次の手順では、無効にしていないすべての PBR を再適用します。PBR はメモリツリー (ルックアップテーブル) を構成します。たとえば、10 個の PBR（p1 ～p10）を作成してから、別の PBR（p11）を作成して適用すると、すべての PBR（p1 ～p11）が新たに適用され、新しい参照テーブルが作成されます。セッションに DENY PBR が関連付けられている場合、そのセッションは破棄されます。

PBR を変更するたびに、この手順を適用する必要があります。たとえば、PBR を無効にした後は次の手順を実行する必要があります。

注: NetScaler アプライアンスで作成された PBR は、適用されるまで機能しません。

CLI を使用して PBR を適用するには:

コマンドプロンプトで入力します。

PBR に適用

GUI を使用して PBR を適用するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [PBR] に移動します。
2. [PBR] タブで PBR を選択し、[アクション] リストで [適用] を選択します。

PBR の有効化または無効化

デフォルトでは、PBR は有効になっています。つまり、PBR が適用されると、NetScaler アプライアンスは受信パケットを構成済みの PBR と自動的に比較します。PBR をルックアップテーブルに含める必要はないが、設定には保

持する必要がある場合は、PBR を適用する前に無効にする必要があります。PBR が適用された後、NetScaler は受信パケットと無効になっている PBR を比較しません。

CLI を使用して PBR を有効または無効にするには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- enable ns pbr <name>
- disable ns pbr <name>

例:

```
> enable ns PBR pbr1
 Done
> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                           Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                                 Protocol:
        Vlan:                                   Interface:
        Active Status: ENABLED                  Applied Status: APPLIED
        Priority: 10
        NextHop: 10.102.29.77

 Done

> disable ns PBR pbr1
Warning: PBR modified, use 'apply pbrs' to commit this operation

> apply pbrs
 Done

> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                           Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                                 Protocol:
        Vlan:                                   Interface:
        Active Status: DISABLED                 Applied Status:
           NOTAPPLIED
        Priority: 10
        NextHop: 10.102.29.77
Done
<!--NeedCopy-->
```

GUI を使用して PBR を有効または無効にするには:

1. [システム] > [ネットワーク] > [PBR] に移動します。
2. [PBR] タブで PBR を選択し、[アクション] リストで [有効化] または [無効化] を選択します。

PBR の番号変更

PBR の番号を自動的に変更して、優先順位を 10 の倍数に設定できます。

CLI を使用して PBR に番号を付け直すには:

コマンドプロンプトで入力します。

- nspbrs に番号を付け直す

GUI を使用して PBR に番号を付け直すには:

[システム] > [ネットワーク] > [PBR] に移動し、[PBR] タブの [アクションリスト] で [優先順位の再番号付け] を選択します。

使用事例-複数ホップの PBR

NetScaler アプライアンス NS1 に PBR1 と PBR2 の 2 つの PBR が構成されているシナリオを考えてみましょう。PBR1 は、送信元 IP アドレスが 10.102.29.30 のすべての送信パケットをネクストホップルータ R1 にルーティングします。PBR2 は、送信元 IP アドレスが 10.102.29.90 のすべての送信パケットをネクストホップルータ R2 にルーティングします。R3 は NS1 に接続されたもう 1 つのネクストホップルータです。

ルータ R1 に障害が発生すると、PBR1 と一致したすべての発信パケットがドロップされます。このような状況を回避するには、PBR を作成または変更するときに、ネクストホップフィールドにリンクロードバランシング（LLB）仮想サーバーを指定できます。複数のネクストホップがサービスとして LLB 仮想サーバにバインドされます（R1、R2、R3 など）。これで、R1 に障害が発生すると、PBR1 と一致したすべてのパケットが LLB 仮想サーバに設定された LB 方式によって決定された R2 または R3 にルーティングされます。

次の場合に LLB 仮想サーバーをネクストホップとして PBR を作成しようとすると、NetScaler アプライアンスはエラーをスローします。

- 同じ LLB 仮想サーバで別の PBR を追加します。
- 存在しない LLB 仮想サーバーを指定する。
- バインドされたサービスがネクストホップではない LLB 仮想サーバーを指定する。
- LB 方式が次のいずれかに設定されていない LLB 仮想サーバを指定する。
  - ROUNDROBIN
  - DESTINATIONIPHASH
  - SOURCEIPHASH
  - SRCIPDESTIPHASH
  - LEASTPACKETS
  - LEASTBANDWIDTH
  - LTRM
  - CALLIDHASH
  - CUSTOM LOAD

- LB パーシステンスタイプが次のいずれかに設定されていない LLB 仮想サーバーを指定します。
  - DESTIP
  - SOURCEIP
  - SRCDSTIP

次の表は、NetScaler アプライアンスで構成されているエンティティの名前と値を示しています。

エンティティの種類	名前	IP アドレス
リンク負荷分散仮想サーバー	LLB1	-
サービス (ネクストホップ)	Router1	1.1.1.254
	Router2	2.2.2.254
	Router3	3.3.3.254
PBR	PBR1	-
	PBR2	-

表 1. エンティティ作成のサンプル値

上記の構成を実装するには、次のことが必要です。

1. ネクストホップルータ R1、R2、R3 を表すルータ 1、ルータ 2、ルータ 3 のサービスを作成します。
2. リンク負荷分散仮想サーバー LLB1 を作成し、それにサービス Router1、Router2、Router3 をバインドします。
3. PBR PBR1 と PBR2 を作成し、ネクストホップフィールドをそれぞれ LLB1 と 2.2.254（ルータ R2 の IP アドレス）に設定します。

CLI を使用してサービスを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add service <name> <IP> <serviceType> <port>
- show service <name>

例:

```
1 > add service Router1 1.1.1.254 ANY *
2  Done
3 > add service Router2 2.2.2.254 ANY *
4  Done
5 > add service Router3 3.3.3.254 ANY *
6  Done
7 <!--NeedCopy-->
```


GUI を使用してサービスを作成するには、次の手順を実行します。

[トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを作成します。

CLI を使用してリンク負荷分散仮想サーバーを作成し、サービスをバインドするには:

コマンドプロンプトで入力します。

- add lb vserver <name> <serviceType>
- bind lb vserver < name> <serviceName>
- show lb vserver < name>

例:

```
> add lb vserver LLB1 ANY
 Done
> bind lb vserver LLB1 Router1 Router2 Router3
 Done
<!--NeedCopy-->
```

GUI を使用してリンク負荷分散仮想サーバーを作成し、サービスをバインドするには:

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、リンク負荷分散用の仮想サーバーを作成します。「プロトコル」フィールドに「ANY」を指定します。
   注:「
   直接アドレス指定可能」がオフになっていることを確認してください。
2. 「サービス」タブの「アクティブ」列で、仮想サーバーにバインドするサービスのチェックボックスを選択します。

CLI を使用して PBR を作成するには:

コマンドプロンプトで入力します。

- add ns pbr <name> <action> [-srcIP [\<operator>] <srcIPVal>] [-nextHop \<nextHopVal>]
- show ns pbr

例:

```
> add pbr PBR1 ALLOW -srcIP 10.102.29.30 -nextHop LLB1
 Done
> add pbr PBR2 ALLOW -srcIP 10.102.29.90 -nextHop 2.2.2.254
 Done
<!--NeedCopy-->
```

GUI を使用して PBR を作成するには:

[システム] > [ネットワーク] > [PBR] に移動し、[PBR] タブで新しい PBR を追加します。

IPv6 トラフィック用のポリシーベースルート (PBR6)

August 15, 2023

PBR6 の設定には次のタスクが含まれます。

- PBR6 を作成します。
- PBR6 を適用してください。
- （オプション）PBR6 を無効または有効にします。
- （オプション）PBR6 のプライオリティの番号を変更します。

PBR6 の作成または変更

同じパラメータで 2 つの PBR6 を作成することはできません。複製を作成しようとすると、エラーメッセージが表示されます。

PBR6 のプライオリティを設定できます。優先度（整数値）は、NetScaler アプライアンスが PBR6 を評価する順序を定義します。優先度を指定せずに PBR6 を作成すると、NetScaler は自動的に 10 の倍数の優先度を割り当てます。

パケットが PBR6 で定義されている条件と一致すると、NetScaler はアクションを実行します。パケットが PBR6 で定義されている条件と一致しない場合、NetScaler はそのパケットを次に優先順位の高い PBR6 と比較します。

CLI のプロシージャ

CLI を使用して PBR6 を作成するには:

コマンドプロンプトで入力します。

- **add ns pbr6** <name> <action> [-srcIPv6 [\] \] \[-srcPort \[\\] \] \[-destIPv6 \[\\] \] \[-destPort \[\\] \] \[-srcMac \\<mac_addr>] \[-protocol \ |-protocolNumber \<positive_integer>] \[-vlan \\<positive_integer>] \[-interface \\<interface_name>] \[-priority \\<positive_integer>] \[-state \(ENABLED | DISABLED)] \[-msr \(ENABLED | DISABLED) \[-monitor \\]] \[-nextHop \\] \[-nextHopVlan \\<positive_integer>]
- show ns pbr

CLI を使用して PBR6 を変更または削除するには:

PBR6 を変更するには、**set pbr6** <name> コマンドと変更するパラメーターを新しい値とともに入力します。

CLI を使用して 1 つまたはすべての PBR6 を削除するには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- **rm ns pbr6** <name>
- **clear ns pbr6**

GUI のプロシージャ

GUI を使用して PBR6 を作成または変更するには:

[システム] > [ネットワーク] > [PBR] に移動し、[PBR6s] タブで新しい PBR6 を追加するか、既存の PBR6 を編集します。

GUI を使用して 1 つまたはすべての PBR6 を削除するには:

[システム] > [ネットワーク] > [PBR] に移動し、[PBR6s] タブで PBR6 を削除します。

PBR6 の適用

PBR6 をアクティブ化するには、PBR6 を適用する必要があります。次の手順では、無効にしていないすべての PBR6 を再適用します。PBR6 はメモリツリー (ルックアップテーブル) を構成します。たとえば、10 個の PBR6 (p6_1-p6_10) を作成してから、別の PBR6 (p6_11) を作成して適用すると、すべての PBR6 (p6_1-p6_11) が新たに適用され、新しい参照テーブルが作成されます。セッションに関連する DENY PBR6 がある場合、そのセッションは破棄されます。

PBR6 を変更するたびに、この手順を適用する必要があります。たとえば、PBR6 を無効にした後は、次の手順に従う必要があります。

注: NetScaler アプライアンスで作成された PBR6 は、適用されるまで機能しません。

CLI を使用して PBR6 を適用するには:

コマンドプロンプトで入力します。

- **apply ns PBR6**

GUI を使用して PBR6 を適用するには:

1. [システム] > [ネットワーク] > [PBR] に移動します。
2. 「PBR6s」タブで PBR6 を選択し、「アクション」リストで「適用」を選択します。

PBR6 の有効化または無効化

デフォルトでは、PBR6 は有効になっています。つまり、PBR6 が適用されると、NetScaler アプライアンスは送信 IPv6 パケットを構成済みの PBR6 と自動的に比較します。PBR6 をルックアップテーブルに含める必要はないが、設定に保持する必要がある場合は、PBR6 を適用する前に無効にする必要があります。PBR6 が適用された後、NetScaler は受信パケットを無効化された PBR6 と比較しません。

CLI を使用して PBR6 を有効または無効にするには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- **enable ns pbr** <name>
- **nspbr** を無効にする < 名前 >

GUI を使用して PBR6 を有効または無効にするには:

1. [システム] > [ネットワーク] > [PBR] に移動します。
2.「PBR6s」タブで PBR6 を選択し、「アクション」リストで「有効化」または「無効化」を選択します。

PBR6 の番号付け直し

PBR6 の番号を自動的に変更して、優先順位を 10 の倍数に設定できます。

CLI を使用して PBR6 に番号を付け直すには:

コマンドプロンプトで入力します。

- **renumber ns pbr6**

GUI を使用して PBR6 に番号を付け直すには:

[システム] > [ネットワーク] > [PBR] に移動し、[PBR6s] タブの [アクション] リストで [優先順位の再番号付け] を選択します。

PBR の **MAC** アドレスワイルドカードマスク

August 15, 2023

拡張 PBR と PBR6 にワイルドカードマスクパラメータが導入されました。ワイルドカードマスクパラメータは、送信パケットの送信元 MAC アドレスと照合する MAC アドレスの範囲を定義するために送信元 MAC アドレスパラメータとともに使用されます。

ワイルドカードマスクは、使用する MAC アドレスの 16 進数と、無視する 16 進数を指定します。ワイルドカードマスクパラメータは一連の 1 と 0 を指定し、長さは 12 桁です。各桁は、MAC アドレスの対応する 16 進数のマスクです。ワイルドカードマスクのゼロ桁は、MAC アドレスの対応する 16 進数を考慮する必要があることを示し、1 桁は対応する 16 進数を無視することを示します。

ワイルドカードマスクは次の条件を満たす必要があります。

- ゼロのシリーズが 1 つしかありません
- シリーズは 1 つだけです
- 一連の 0 から始める

有効なワイルドカードマスクの例を以下に示します。

- 000000111111
- 000000011111
- 000011111111

次に、無効なワイルドカードマスクの例をいくつか示します。

- 000000111100
- 111110000000
- 010101010101

PBR ルールの場合、MAC アドレス 96: fa: 95:1 d: 67:4 のワイルドカードマスクは 0000001111 であり、MAC アドレス範囲 96:FA: 95:00:00-96: FA: 95: FF: FF: FF: FF を定義します。この MAC アドレス範囲は、送信パケットの送信元 MAC アドレスと照合されます。

CLI を使用して PBR ルール内の送信元 MAC アドレスの範囲を指定するには:

コマンドプロンプトで入力します。

- **add ns pbr** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show ns pbr** <pbrname>

例**:**

```
1  > add ns pbr PBR-1 ALLOW -srcip 192.0.2.34 -srcMac 96:fa:95:1d:67:4a
       - srcMacMask 000000111111 -nexthop 198.51.100.1
2
3   Done
```

CLI を使用して PBR6 ルール内の送信元 MAC アドレスの範囲を指定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns pbr6** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show pbr6** <pbr6name>

例**:**

```
1  > add ns pbr6 PBR6-1 ALLOW  - srcipv6 2001:db8:0::7 -srcMac 96:fa:95:1d
       :67:4a   - srcMacMask 000000001111 -nexthop 2001:db8:0::1
2   Done
```

NULL ポリシーベースルートを使用して送信パケットをドロップする

August 15, 2023

状況によっては、たとえばテストケースや展開の移行中に、NetScaler アプライアンスが特定の送信パケットをルーティングするのではなく、ドロップする必要がある場合があります。

NULL ポリシーベースのルートを使用して、特定の送信パケットをドロップできます。NULL PBR は、ネクストホップパラメータが NULL に設定されている PBR の一種です。NetScaler アプライアンスは、NULL PBR と一致する送信パケットをドロップします。

IPv4 パケット用の **NULL PBR** の設定

CLI を使用して NULL PBR を作成するには:

コマンドプロンプトで入力します。

- **add ns pbr** <name> ALLOW [**-td** \<positive_integer>] [-**srcIP** [\] \] \[-**srcPort** \[\\] \] \[-**destIP** \[\\] \] \[-**destPort** \[\\] \] \(-**nextHop NULL**) \[**srcMac** \\<mac_addr> \[-**srcMacMask** \\]] \[-**protocol** \ | -**protocolNumber** \<positive_integer>] \[-**vlan** \<positive_integer> | -**vxlan** \<positive_integer>] \[-**interface** \\<interface_name>] \[-**priority** \\<positive_integer>] \[-**msr** \(**ENABLED** | **DISABLED**) \[**-monitor** \\]] \[-**state** \(**ENABLED** | **DISABLED**)\[-**ownerGroup** \\]
- **apply ns pbrs**
- **show ns pbr**<id>

GUI を使用して NULL PBR を設定するには:

[システム] > [ネットワーク] > [**PBR**] に移動し、[**PBR]** タブで新しい **NULL PBR** を追加するか、既存の **NULL PBR** を編集します。

設定例

次の設定例では、インターフェイス 1/5 からの発信 IPv6 パケットをドロップするために、NULL PBR6 PBR6-NULL-EXAMPLE-1 が設定されています。

```
> add ns pbr PBR6-NULL-EXAMPLE-1 ALLOW  -nextHop NULL -interface 1/5
 Done

> apply ns pbr6
 Done
```


5 つのタプル情報に基づく複数のルートでのトラフィック分散

August 15, 2023

負荷分散のセットアップでは、NetScaler ADC アプライアンスは、パケットを宛先に送信するための複数のルートを持つことができます。たとえば、サーバーに、クライアントに。

NetScaler ADC アプライアンスは、ハッシュアルゴリズムを使用して、パケットを宛先に送信するルートを選択します。

ハッシュアルゴリズムは、次の 2 つのパケットタプルを使用してハッシュを計算し、それに基づいて NetScaler ADC アプライアンスがパケットのルートを選択します。

- 送信元 IP アドレス
- 宛先 IP アドレス

2 つのタプル情報に基づいてルートを選択すると、利用可能なルートのトラフィックが不均等に分散する可能性があります。このトラフィックの不均一な分布は、一部のルートでトラフィックの過負荷につながります。

この問題を解決するために、ビルド 13.0 71.x から、NetScaler ADC アプライアンスはハッシュアルゴリズムでパケットの次の 5 つのタプル情報を使用して、パケットのルートを選択します。

- 送信元 IP アドレス (クライアント IP)
- 送信元ポート (クライアントポート)
- 宛先 IP アドレス (サービス IP)
- 宛先ポート (サービスポート)
- プロトコル番号

5 つのタプル情報に基づいてルートを選択すると、利用可能なルート上のトラフィックが均等に分散されます。このようにトラフィックが均等に分散されるため、ルート内のトラフィックの過負荷を防ぐことができます。

クライアントが VIP アドレスに要求を送信する負荷分散設定の例を考えてみましょう。NetScaler ADC アプライアンスは、次の 5 つのタプル情報を使用して、負荷分散されたサーバーに要求パケットを送信するルートを選択します。

- 送信元 IP アドレス (クライアント IP アドレス)
- 送信元ポート (クライアントポート)
- 宛先 IP アドレス (サービス IP アドレス)
- 宛先ポート (サービスポート番号)
- プロトコル番号

Use source IP (USIP) モードが有効になっている場合、5 つのタプルすべてがルートを選択するためのハッシュの入力とみなされます。Use Subnet IP (USNIP) モードが有効になっている場合は、SNIP と送信元ポートの両方がルート選択後に選択されるため、入力とは見なされません。USIP および USNIP モードの設定方法については、「送信元 IP モードの使用を有効にする 」および「サブネット IP アドレス (SNIP) を設定する」を参照してください。

> 注:
>
> ビルド 13.1 30.x から、NetScaler ADC アプライアンスは、2 つのタプルハッシュアルゴリズムの代わりに 5 つのタプルハッシュアルゴリズムを使用して、負荷分散モニタープローブのルートを選択します。

他のルート選択に基づく **NetScaler ADC** 機能に関する優先順位

このセクションでは、NetScaler ADC アプライアンスの 5 つのタプル機能とその他のルート選択関連機能に基づくルート選択の優先順位について説明します。

- ポリシーベースルート **(PBR)**。PBR ルールは、5 つのタプルに基づくルート選択よりも常に優先されます。
- **Mac** ベースの転送 **(MBF)**。ロード・バランシング構成では、次の場合に、5 つのタプルに基づく MBF またはルート選択が優先されます。
  - NetScaler ADC アプライアンスの負荷分散構成の VIP アドレスへのクライアント開始トラフィックの場合:
    * 負荷分散されたサーバー宛てのトラフィックを要求します。5 つのタプルに基づくルート選択は、MBF よりも優先されます。
    * クライアント宛ての応答トラフィック。MBF は 5 つのタプルに基づくルート選択よりも優先されます。
  - NetScaler ADC アプライアンスの SNIP アドレスへのサーバー開始トラフィックの場合:
    * クライアント宛ての応答トラフィック。5 つのタプルに基づくルート選択は、MBF よりも優先されます。
    * 負荷分散されたサーバー宛てのトラフィックを要求します。MBF は 5 つのタプルに基づくルート選択よりも優先されます。

ルーティングの問題のトラブルシューティング

August 15, 2023

トラブルシューティングプロセスを可能な限り効率的にするには、まずネットワークに関する情報を収集することから始めます。NetScaler アプライアンスとネットワーク内の他のシステムに関する次の情報を取得する必要があります。

- インターフェース接続と中間スイッチの詳細を含む完全なトポロジー図
- 実行中の設定。show running コマンドを使用すると、ns.conf と Zebos.conf の実行構成を取得できます。
- 問題が発生したときに設定が変更されたかどうかを判断するための History コマンドの出力。
- Top および ps-ax コマンドの出力。ルーティングデーモンが CPU を過剰に使用しているか、動作が誤っているかを判断します。

- /var/core にあるルーティング関連のコアファイル (nsm、bgpd、ospfd、ripd) タイムスタンプをチェックして、関連性があるかどうかを確認してください。
- /var/log の dr_error.log ファイルと dr_info.log ファイル。
- 関連するすべてのシステムの日付コマンドと時刻の詳細の出力。ログメッセージの時刻をさまざまなイベントと関連付けることができるように、すべてのデバイスの日付を次々に表示します。
- 関連する ns.log、ニュースログファイル。
- アップストリームおよびダウンストリームのルータからの設定ファイル、ログファイル、およびコマンド履歴の詳細。

汎用ルーティングに関するよくある質問

August 15, 2023

一般的なルーティング問題のトラブルシューティング方法について、ユーザーから次のような質問が寄せられます。

- 設定ファイルを保存するにはどうすればよいですか?

  VTYSH からの書き込みコマンドは Zebos.conf だけを保存します。CLI から ns 構成の保存コマンドを実行して、ns.conf ファイルと Zebos.conf ファイルの両方を保存します。

- スタティックなデフォルトルートと動的に学習されたデフォルトルートの両方を設定した場合、どちらが優先されますか。

  動的に学習されたルートが優先デフォルトルートです。この動作はデフォルトルートに固有のものです。ただし、ネットワークサービスモジュール（NSM）の場合は、アドミニストレーティブディスタンスが変更されない限り、動的ルートよりも RIB で静的に設定されたルートの方が優先されます。NSM FIB にダウンロードされるルートはスタティックルートです。

- デフォルトルートのアドバタイズメントをブロックするにはどうすればよいですか。

  デフォルトルートは ZebOS には注入されません。

- ネットワークデーモンのデバッグ出力を見るにはどうすればいいですか?

  ネットワークデーモンからのデバッグ出力をファイルに書き込むには、VTYSH のグローバル設定ビューから次のログファイルコマンドを入力します。

  ```
  ns(config)# log file /var/ZebOS.log
  <!--NeedCopy-->
  ```

  VTYSH ユーザビューから terminal monitor コマンドを入力すると、デバッグ出力をコンソールに送ることができます。

  ```
  ns# terminal monitor
  <!--NeedCopy-->
  ```


- 実行中のデーモンのコアを集めるにはどうすればいいですか?

  gcore ユーティリティを使用すると、実行中のデーモンのコアを収集して gdb で処理できます。これは、ルーティング操作全体を停止させることなく、誤動作するデーモンをデバッグするのに役立つ場合があります。

```
gcore [-s] [-c core] [executable] pid
<!--NeedCopy-->
```

  -s オプションは、コアイメージの収集中にデーモンを一時的に停止します。この方法を使うと、生成される画像に一貫した状態のコアが確実に表示されるので、このオプションをお勧めします。

```
root@ns#gcore -s -c nsm.core /netscaler/nsm 342
<!--NeedCopy-->
```

- ZebOS コマンドのバッチ実行方法を教えてください。

  <file-name>VTYSH-f コマンドを入力すると、ファイルから複数の ZebOS コマンドを実行できます。これは実行構成を置き換えるのではなく、それに追加されます。ただし、バッチファイル内の既存の構成を削除し、それを新しい目的の構成に追加するコマンドを含めることで、このメカニズムを使用して特定の構成を置き換えることができます。

```
!
router bgp 234
network 1.1.1.1 255.255.255.0
!
route-map bgp-out2 permit 10
 set metric 9900
 set community 8602:300
!
<!--NeedCopy-->
```

OSPF 固有の問題のトラブルシューティング

August 15, 2023

OSPF 固有の問題のデバッグを開始する前に、NetScaler アプライアンスと、影響を受ける LAN 内のすべてのシステム（上流および下流のルーターを含む）から情報を収集する必要があります。まず、次のコマンドを入力します。

1. show interface from both nscli and VTYSH
2. show ip ospf interface
3. show ip ospf neighbor detail
4. show ip route
5. show ip ospf route

6. show ip ospf database summary

   - データベース内の LSA が少ない場合は、show ip ospf データベースルータ、show ip ospf database A. network、show ip ospf database external、およびその他のコマンドを入力して LSA の詳細情報を取得してください。
   - データベースに多数の LSA がある場合は、show ip ospf database 自己生成コマンドを入力します。

7. show ip ospf
8. show ns ip. これにより、関心のあるすべての VIP の詳細が含まれるようになります。
9. ピアリングデバイスからログを取得し、次のコマンドを実行します。

   ```
   gcore -s -c xyz.core /netscaler/ospfd <pid>
   ```

   注**:** gcore コマンドは中断を伴いません。

NetScaler から以下のように追加情報を収集します。

1. VTYSH のグローバル設定ビューから次のコマンドを入力して、エラーメッセージのロギングを有効にします。

   ```
   ns(config)# log file /var/ospf.log
   <!--NeedCopy-->
   ```

2. OSPF イベントのデバッグを有効にし、次のコマンドを使用してログに記録します。

   ```
   ns(config) #log file /var/ospf.log
   <!--NeedCopy-->
   ```

   データベース内の LSA の数が比較的少ない（500 未満）場合にのみ、OSPF LSA パケットのデバッグを有効にします。

インターネットプロトコルバージョン **6**（**IPv6**）

August 15, 2023

NetScaler アプライアンスはサーバー側とクライアント側の両方の IPv6 をサポートしているため、IPv6 ノードとして機能できます。IPv6 ノード (ホストとルーターの両方) と IPv4 ノードからの接続を受け付けることができ、トラフィックをサービスに送信する前にプロトコル変換 (RFC 2765) を実行できます。

次の表は、NetScaler アプライアンスがサポートする IPv6 機能の一部を示しています。

表 1. サポートされている IPv6 機能の一部

IPv6 の機能
SNIP (NSIP6、VIP6、SNIP6) の IPv6 アドレス
ネイバーディスカバリ（アドレス解決、重複アドレス検出、ネイバー到達不能検出、ルータディスカバリ）
管理アプリケーション (ping6、telnet6、ssh6)
スタティックルーティングとダイナミックルーティング (OSPF、BGP、RIPng、ISIS)
ポートベースの VLAN
IPv6 アドレスのアクセス制御リスト (ACL6)
IPv6 プロトコル (TCP6、UDP6、ICMP6)
サーバーサイドサポート (仮想サーバー、サービスの IPv6 アドレス)
IPv6 の USIP (ソース IP を使用) および DSR (ダイレクトサーバーリターン)
IPv6 用の SNMP と VPN
ネイティブ IPv6 ノードアドレスを持つ HA
MIP 用の IPv6 アドレス
IPv6 のパス MTU ディスカバリ

IPv6 サポートの実装

IPv6 機能を使用または構成する前に、NetScaler アプライアンスで IPv6 機能を有効にする必要があります。IPv6 が無効になっている場合、NetScaler は IPv6 パケットを処理しません。サポートされていないコマンドを実行すると、次の警告が表示されます。

```
"Warning: Feature(s) not enabled [IPv6PT]"
<!--NeedCopy-->
```

IPv6 を有効または無効にするには、次のいずれかの手順に従います。

CLI のプロシージャ

CLI を使用して IPv6 を有効または無効にするには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- NS 機能 ipv6pt を有効にする
- NS 機能 ipv6pt を無効にする

GUI のプロシージャ

GUI を使用して IPv6 を有効または無効にするには:

1. [システム] > [設定] に移動し、[モードと機能] グループで [拡張機能の設定] をクリックします。
2. **IPv6** プロトコル変換オプションを選択または選択解除します。

VLAN のサポート

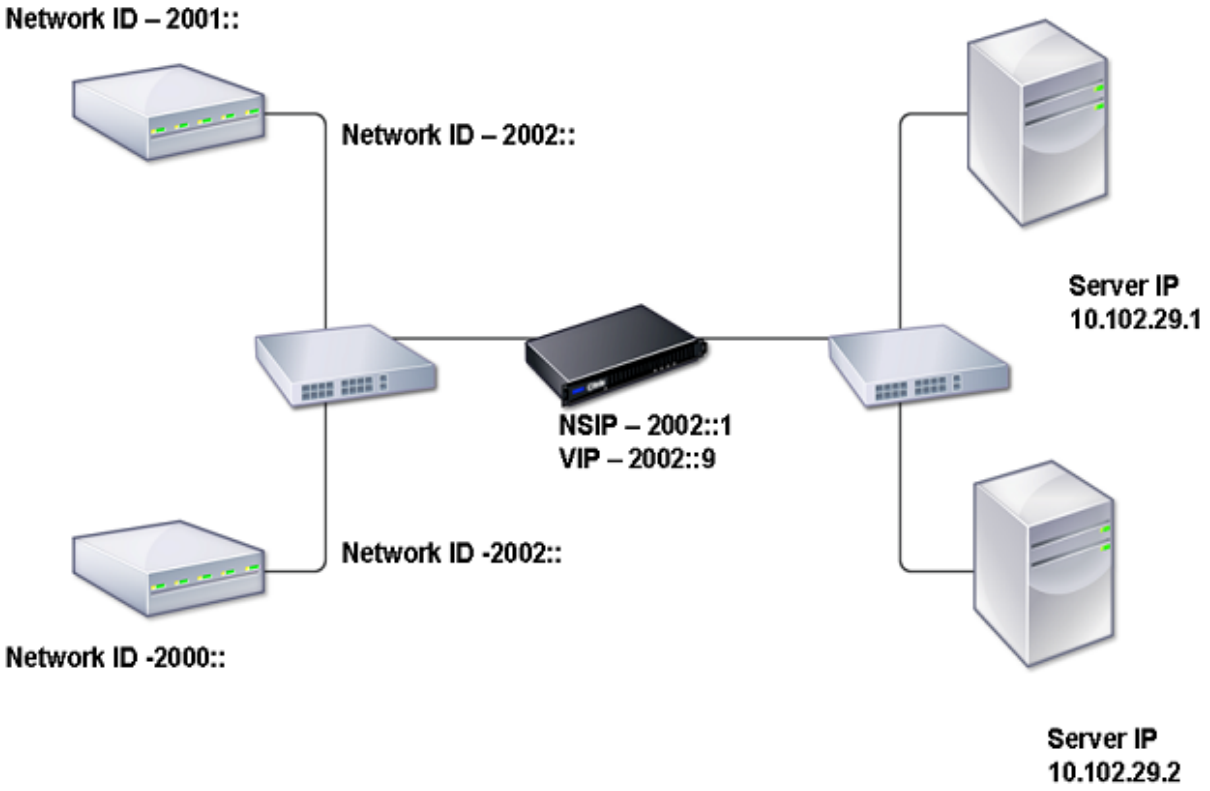
VLAN を特定せずにブロードキャストまたはマルチキャストパケットを送信する必要がある場合（たとえば、NSIP の場合は DAD 中、ルートのネクストホップの場合は ND6 中）、NetScaler アプライアンスをすべてのインターフェイスに適切なタグを付けてパケットを送信するように構成できます。VLAN は ND6 によって識別され、データパケットは VLAN 上でのみ送信されます。ND6 および VLAN の詳細については、ネイバー探索の設定を参照してください。

ポートベースの VLAN は、IPv4 および IPv6 に共通です。IPv6 では、プレフィックスベースの VLAN がサポートされています。

簡単な導入シナリオ

以下は、次のトポロジー図に示すように、IPv6 仮想サーバーと IPv4 サービスで構成される単純な負荷分散設定の例です。

図 1: IPv6 サンプルトポロジー



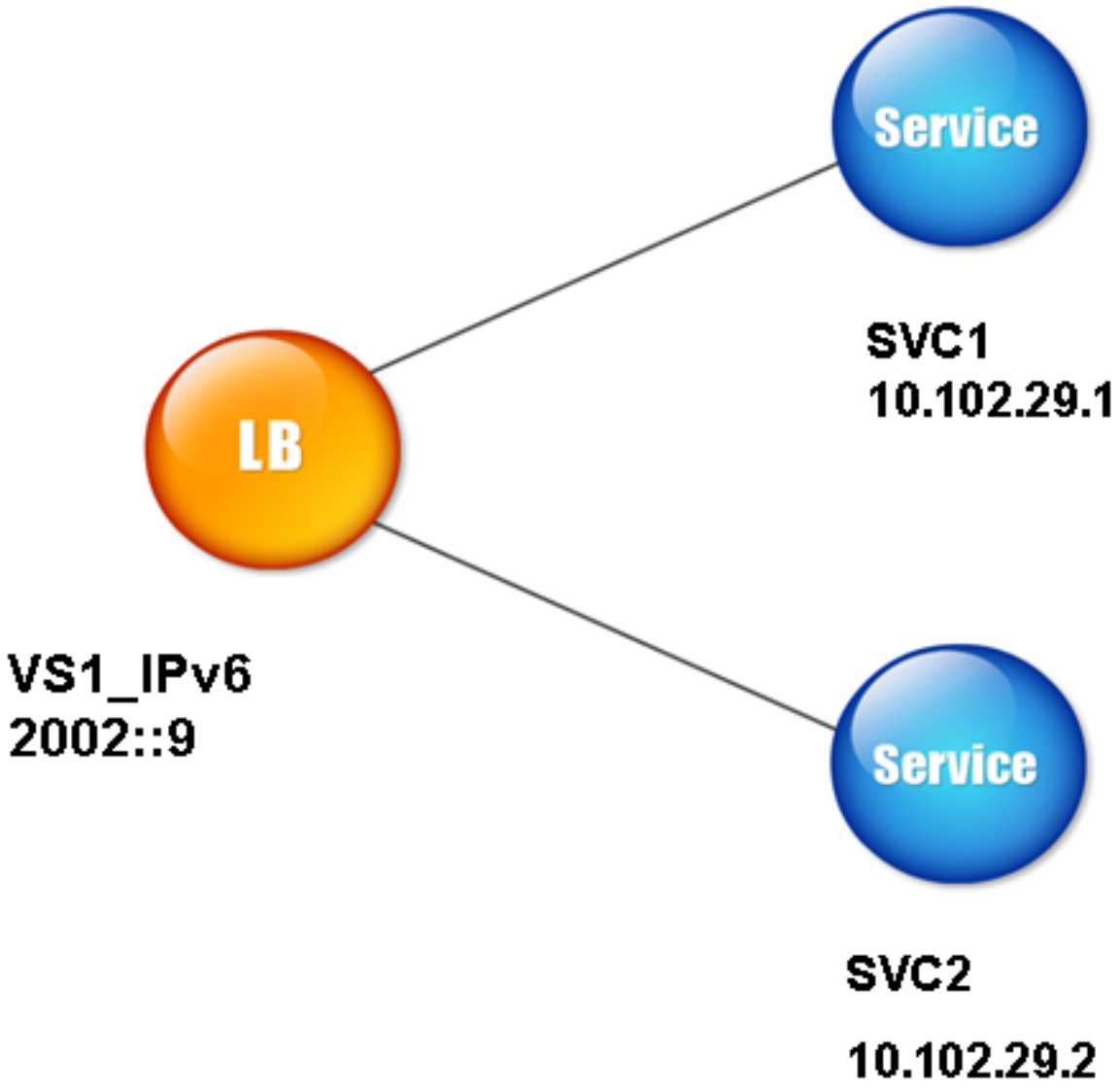
次の表は、NetScaler で構成する必要があるエンティティの名前と値をまとめたものです。

表 2. エンティティ作成のサンプル値

エンティティタイプ	名前	値
LB 仮想サーバー	VS1_IPv6	2002::9
Services	SVC1	10.102.29.1
	SVC2	10.102.29.2

次の図は、NetScaler で構成されるパラメーターのエンティティと値を示しています。

図 2: IPv6 エンティティダイアグラム



このデプロイシナリオを設定するには、以下を実行する必要があります。

1. IPv6 サービスを作成します。
2. IPv6 LB 仮想サーバーを作成します。
3. サービスを仮想サーバーにバインドします。

CLI のプロシージャ

CLI を使用して IPv4 サービスを作成するには:

コマンドプロンプトで入力します。

- **add service** <Name> <IPAddress> <Protocol> <Port>
- **sh service** <Name>

例**:**

```
> add service SVC1 10.102.29.1 HTTP 80
Done

>add service SVC2 10.102.29.2 HTTP 80
Done
<!--NeedCopy-->
```

CLI を使用して IPv6 仮想サーバーを作成するには:

コマンドプロンプトで入力します。

- **add lb vserver** <Name> <IPAddress> <Protocol> <Port>
- **sh lb vserver** <Name>

例**:**

```
> add lb vserver VS1_IPv6 2002::9 HTTP 80
Done
<!--NeedCopy-->
```

CLI を使用してサービスを LB 仮想サーバーにバインドするには:

コマンドプロンプトで入力します。

- **bind lb vserver** <name> <service>
- **sh lb vserver** <name>

例**:**

```
> bind lb vserver VS1_IPv6 SVC1
Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して IPv4 サービスを作成するには:

[トラフィック管理] > [負荷分散] > [サービス] に移動し、[追加] をクリックして次のパラメータを設定します。

- サービス名
- IP アドレス
- プロトコル
- ポート

GUI を使用して IPv6 仮想サーバーを作成するには:

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、[追加] をクリックして [**IPv6**] チェックボックスを選択します。
2. 次のパラメーターを設定します。

    - 名前
    - プロトコル
    - IP アドレスの種類
    - IP アドレス
    - ポート

GUI を使用してサービスを LB 仮想サーバーにバインドするには:

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2.「負荷分散仮想サーバー」ページで、サービスをバインドする仮想サーバー（VS1_IPv6 など）を選択します。
3.［開く］をクリックします。
4. [仮想サーバー **(**負荷分散**)** の設定] ダイアログボックスの [サービス] タブで、仮想サーバーにバインドするサービス (たとえば、SVC1) に対応する [アクティブ] チェックボックスを選択します。
5.［**OK**］をクリックします。
6. 手順 1～4 を繰り返してサービスをバインドします（たとえば、SVC2 を仮想サーバーにバインドします）。

ホストヘッダーの変更

HTTP 要求のホストヘッダーに IPv6 アドレスが含まれていて、サーバーが IPv6 アドレスを認識しない場合は、IPv6 アドレスを IPv4 アドレスにマッピングする必要があります。次に、IPv4 アドレスは、仮想サーバーに送信される HTTP 要求のホストヘッダーで使用されます。

CLI のプロシージャ

CLI を使用してホストヘッダーの IPv6 アドレスを IPv4 アドレスに変更するには:

コマンドプロンプトで入力します。

- **set ns ip6** <IPv6Address> -**map** <IPAddress>
- **sh ns ip6** <IPv6Address>

例**:**

```
1 > set ns ip6 2002::9 -map 200.200.200.200
2 Done
3 <!--NeedCopy-->
```


GUI のプロシージャ

GUI を使用してホストヘッダーの IPv6 アドレスを IPv4 アドレスに変更するには:

1. **[**** システム] > [ネットワーク] > [IP] に移動し、[**IPv6**] タブで、マッピング IP アドレスを設定する IP アドレス (たとえば、2002:0:0:0:0:0:0:9) を選択し、[編集] をクリックします。**
2. 「マップされた **IP** 」テキストボックスに、構成するマッピングされた IP アドレス（200.200.200.200.200 など）を入力します。

VIP インサーション

IPv6 アドレスが IPv4 ベースのサーバーに送信された場合、サーバーは HTTP ヘッダー内の IP アドレスを理解できず、エラーを生成する可能性があります。これを回避するには、IPv4 アドレスを IPv6 VIP にマッピングできます。次に、VIP 挿入を有効にして、サーバーに送信される HTTP 要求に IPv4 VIP アドレスとポート番号を挿入できるようにします。

CLI のプロシージャ

CLI を使用してマップ IPv6 アドレスを設定するには:

コマンドプロンプトで入力します。

set ns ip6 <IPv6Address> -**map** <IPAddress>

例**:**

```
1 > set ns ip6 2002::9 -map 200.200.200.200
2  Done
3 <!--NeedCopy-->
```

CLI を使用して VIP 挿入を有効にするには:

コマンドプロンプトで入力します。

- **set lb vserver** <name> -**insertVserverIPPort** <Value>
- **sh lb vserver** <name>

例**:**

```
1 > set lb vserver VS1_IPv6 -insertVserverIPPort ON
2  Done
3
4 <!--NeedCopy-->
```


GUI のプロシージャ

GUI を使用してマップ IPv6 アドレスを設定するには:

1. **[** システム] > [ネットワーク] > [IP] に移動し、[**IPv6**] タブで、マップ IP アドレスを設定する IP アドレス (たとえば、2002:0:0:0:0:0:0:9) を選択し、[編集] をクリックします。**
2. 「マップされた **IP** 」テキストボックスに、構成するマップ IP アドレス（たとえば、200.200.200.200）を入力します。

GUI を使用して VIP 挿入を有効にするには:

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、ポート挿入を有効にする仮想サーバーを選択して、[編集] をクリックします。
2. **[** 詳細設定] タブの [トラフィック設定] の [仮想サーバー **IP** ポート挿入**]** ドロップダウンリストボックスで、**[VIPADDR**] を選択します。**
3. 「**Vserver IP** ポート挿入」テキストボックスに、vip ヘッダーを入力します。

トラフィックドメイン

August 15, 2023

> 警告
>
> トラフィックドメインではなく、管理パーティションを使用することをお勧めします。詳細については、「管理者パーティショニング 」ページを参照してください。

トラフィックドメインは、さまざまなアプリケーションのネットワークトラフィックをセグメント化する方法です。トラフィックドメインを使用して、NetScaler ADC アプライアンス内に複数の隔離された環境を作成できます。特定のトラフィックドメインに属するアプリケーションは、エンティティと通信し、そのドメイン内のトラフィックを処理します。あるトラフィックドメインに属するトラフィックは、別のトラフィックドメインの境界を越えることはできません。

トラフィックドメインを使用する利点

NetScaler ADC アプライアンスでトラフィックドメインを使用する主な利点は次のとおりです。

- ネットワークでの重複した **IP** アドレスの使用。トラフィックドメインを使用すると、ネットワーク上で重複した IP アドレスを使用できます。重複するアドレスのそれぞれが異なるトラフィックドメインに属している限り、ネットワーク上の複数のデバイス、または NetScaler ADC アプライアンス上の複数のエンティティに同じ IP アドレスまたはネットワークアドレスを割り当てることができます。

- **NetScaler ADC** アプライアンスでの重複エンティティの使用。トラフィックドメインでは、アプライアンスで重複した NetScaler ADC 機能エンティティを使用することもできます。各エンティティが別のトラフィックドメインに割り当てられている限り、同じ設定でエンティティを作成できます。
  注: 同じ名前の重複エンティティはサポートされていません。
- マルチテナンシー。トラフィックドメインを使用すると、ネットワーク上の定義済みのアドレス空間内で各顧客のアプリケーショントラフィックのタイプを分離することで、複数の顧客にホスティングサービスを提供できます。

トラフィックドメインは、整数値である識別子によって一意に識別されます。各トラフィックドメインには、1 つまたは VLAN のセットが必要です。トラフィックドメインの分離機能は、トラフィックドメインにバインドされた VLAN によって異なります。1 つのトラフィックドメインに複数の VLAN をバインドできますが、同じ VLAN を複数のトラフィックドメインに含めることはできません。したがって、作成できるトラフィックドメインの最大数は、アプライアンスに設定されている VLAN の数によって異なります。

デフォルトトラフィックドメイン

NetScaler ADC アプライアンスには、
デフォルトのトラフィックドメインと呼ばれる事前構成されたトラフィックドメインがあり、ID は 0 です。すべての工場出荷時の設定と設定は、デフォルトのトラフィックドメインの一部です。他のトラフィックドメインを作成し、デフォルトのトラフィックドメインと他の各トラフィックドメインの間でトラフィックをセグメント化できます。NetScaler ADC アプライアンスからデフォルトのトラフィックドメインを削除することはできません。トラフィックドメイン ID を設定せずに作成したフィーチャエンティティは、自動的にデフォルトのトラフィックドメインに関連付けられます。

注**:** 一部の機能と設定は、デフォルトのトラフィックドメインでのみサポートされています。デフォルト以外のトラフィックドメインでは機能しません。すべてのトラフィックドメインでサポートされている機能のリストについては、「トラフィックドメインでサポートされる NetScaler ADC 機能」を参照してください。

トラフィックドメインの仕組み

トラフィックドメインの例として、ID 1 と 2 の 2 つのトラフィックドメインが NetScaler ADC アプライアンス NS1 で構成されている例を考えてみましょう。

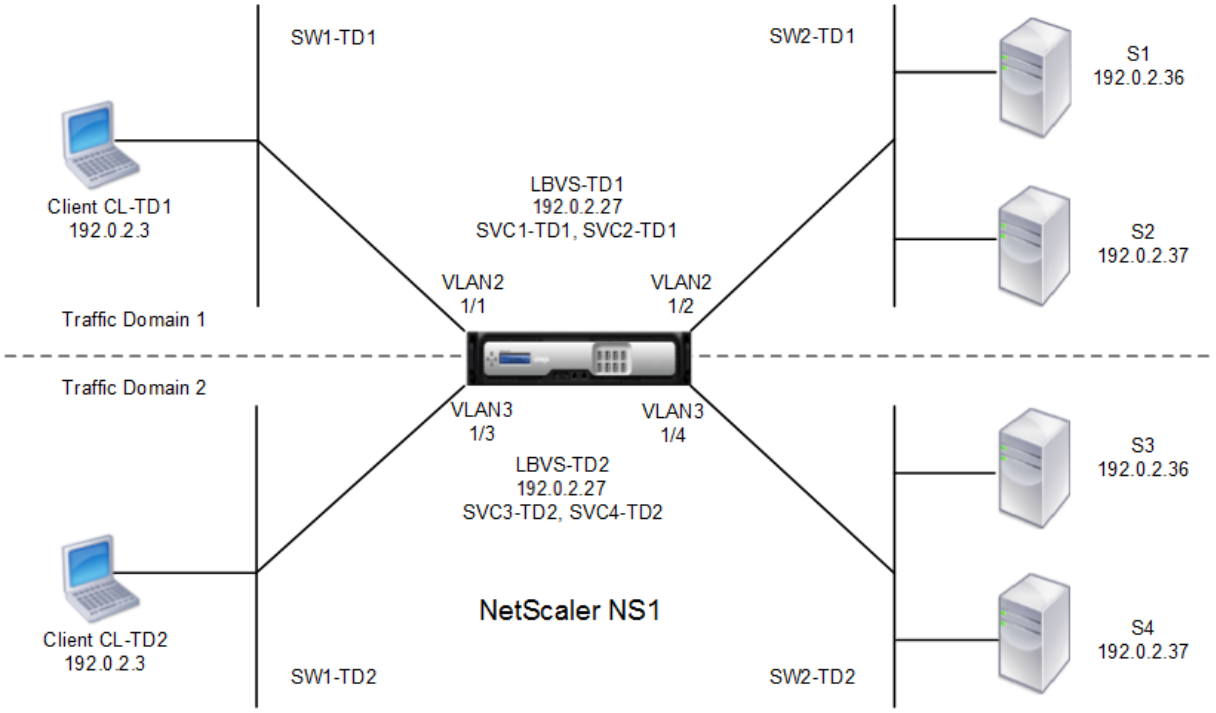
トラフィックドメイン 1 では、負荷分散仮想サーバ LBVS-TD1 は、サーバ S1 および S2 間でトラフィックを負荷分散するように設定されます。NetScaler ADC アプライアンスでは、サーバー S1 と S2 は、それぞれサービス SVC1-TD1 と SVC2-TD1 で表されます。サーバ S1 と S2 は、L2 スイッチ SW2-TD1 を介して NS1 に接続されています。クライアント CL-TD1 は、L2 スイッチ SW1-TD1 を介して NS1 に接続されたプライベートネットワーク上にあります。SW1-TD1 と SW2-TD1 は NS1 の VLAN 2 に接続されています。VLAN 2 はトラフィックドメイン 1 にバインドされます。これは、クライアント CL-TD1 とサーバ S1 と S2 がトラフィックドメイン 1 の一部であることを意味します。

同様に、トラフィックドメイン 2 では、負荷分散仮想サーバー LBVS-TD2 が S3 と S4 間でトラフィックを負荷分散するように構成されています。NetScaler ADC アプライアンスでは、サーバー S3 と S4 は、それぞれサービス SVC3-TD2 と SVC4-TD2 で表されます。サーバ S3 と S4 は、L2 スイッチ SW2-TD2 を介して NS1 に接続されています。クライアント CL-TD2 は、L2 スイッチ SW1-TD2 を介して NS1 に接続されたプライベートネットワーク上にあります。SW1-TD2 と SW2-TD2 は NS1 の VLAN 3 に接続されています。VLAN 3 はトラフィックドメイン 2 にバインドされています。つまり、クライアント CL-TD2 とサーバ S3 と S4 はトラフィックドメイン 2 の一部です。

NetScaler ADC アプライアンスでは、エンティティ LBVS-TD1 と LBVS-TD2 は、IP アドレスを含む同じ設定を共有します。SVC1-TD1 と SVC3-TD2 についても同様であり、SVC2-TD1 と SVC4-TD2 についても同様である。これは、これらのエンティティが異なるトラフィックドメインにあるために可能です。

同様に、サーバー S1 と S3、S2 と S4 は同じ IP アドレスを共有し、クライアント CL-TD1 と CL-TD2 はそれぞれ同じ IP アドレスを持ちます。

図 1: トラフィックドメインの仕組み



次の表は、例で使用される設定の一覧です。

エンティティ	名前	詳細
トラフィックドメイン **1** の設定		
トラフィックドメイン 1 にバインドされた VLAN	VLAN 2	VLAN ID: 2 つのインターフェイスバインド:1/1、1/2
TD1 に接続されたクライアント	CL-TD1（参照目的のみ）	IP アドレス:192.0.2.3
TD1 の仮想サーバーの負荷分散	LBVS-TD1	IP アドレス: 192.0.2.27

エンティティ	名前	詳細
仮想サーバー LBVS-TD1 にバインドされたサービス	SVC1-TD1	IP アドレス：192.0.2.36
仮想サーバー LBVS-TD1 にバインドされたサービス	SVC2-TD1	IP アドレス:192.0.2.37
SNIP	SNIP-TD1（参照目的のみ）	IP アドレス：192.0.2.27
トラフィックドメイン **2** の設定		
トラフィックドメイン 2 にバインドされた VLAN	VLAN 3	VLAN ID: 3 つのインターフェイスバインド:1/3、1/4
クライアントは TD2 に接続されています	CL-TD2 (参考目的のみ)	IP アドレス:192.0.2.3
TD2 の仮想サーバーの負荷分散	LBVS-TD2	IP アドレス：192.0.2.27
仮想サーバー LBVS-TD2 にバインドされたサービス	SVC3-TD2	IP アドレス：192.0.2.36
仮想サーバー LBVS-TD2 にバインドされたサービス	SVC4-TD2	IP アドレス:192.0.2.37
SNIP in TD2	SNIP-TD2（参考目的のみ）	IP アドレス：192.0.2.29

次に、トラフィックドメイン 1 のトラフィックフローを示します。

1. クライアント CL-TD1 は、L2 スイッチ SW1-TD1 を介して IP アドレス 192.0.2.27 の ARP 要求をブロードキャストします。
2. ARP 要求は、VLAN 2 にバインドされているインターフェイス 1/1 の NS1 に到達します。VLAN 2 はトラフィックドメイン 1 にバインドされているため、NS1 はクライアント CL-TD1 の IP アドレスのトラフィックドメイン 1 の ARP テーブルを更新します。
3. ARP 要求はトラフィックドメイン 1 で受信されるため、NS1 は IP アドレスが 192.0.2.27 のトラフィックドメイン 1 に設定されているエンティティを探します。NS1 は、負荷分散仮想サーバー LBVS-TD1 がトラフィックドメイン 1 で構成され、IP アドレスが 192.0.2.27 であることを検出します。
4. NS1 は、インターフェイス 1/1 の MAC アドレスを指定して ARP 応答を送信します。
5. ARP 応答は CL-TD1 に到達します。CL-TD1 は、LBVS-TD1 の IP アドレスの ARP テーブルを NS1 のインターフェイス 1/1 の MAC アドレスで更新します。
6. クライアント CL-TD1 は 192.0.2.27 に要求を送信します。要求は、NS1 のポート 1/1 の LBVS-TD1 によって受信されます。
7. LBVS-TD1 の負荷分散アルゴリズムはサーバ S2 を選択し、NS1 はトラフィックドメイン 1 の SNIP（192.0.2.27）と S2 の間の接続を開きます。
8. S2 は NS1 で SNIP 192.0.2.27 に返信する。
9. NS1 は S2 の応答をクライアント CL-TD1 に送信します。

トラフィックドメイン 2 のトラフィックフローを次に示します。

1. クライアント CL-TD2 は、L2 スイッチ SW1-TD2 を介して、IP アドレス 192.0.2.27 の ARP 要求をブロードキャストします。
2. ARP 要求は、VLAN 3 にバインドされているインターフェイス 1/3 の NS1 に到達します。VLAN 3 はトラフィックドメイン 2 にバインドされているため、同じ IP アドレス（CL-TD1）の ARP エントリがトラフィックドメイン 1 の ARP テーブルにすでに存在していても、NS1 はクライアント CL-TD2 の IP アドレスのトラフィックドメイン 2 の ARP テーブルエントリを更新します。
3. ARP 要求はトラフィックドメイン 2 で受信されるため、NS1 は IP アドレスが 192.0.2.27 のエンティティをトラフィックドメイン 2 で検索します。NS1 は、負荷分散仮想サーバー LBVS-TD2 がトラフィックドメイン 2 で構成され、IP アドレスが 192.0.2.27 であることを検出します。NS1 は、LBVS-TD2 と同じ IP アドレスを持つにもかかわらず、トラフィックドメイン 1 の LBVS-TD1 を無視します。
4. NS1 は、インターフェイス 1/3 の MAC アドレスを指定して ARP 応答を送信します。
5. ARP 応答は CL-TD2 に到達します。CL-TD2 は、LBVS-TD2 の IP アドレスの ARP テーブルエントリを、NS1 のインターフェイス 1/3 の MAC アドレスで更新します。
6. クライアント CL-TD2 は 192.0.2.27 に要求を送信します。要求は、NS1 のインターフェイス 1/3 の LBVS-TD2 によって受信されます。
7. LBVS-TD2 の負荷分散アルゴリズムはサーバー S3 を選択し、NS1 はトラフィックドメイン 2 の SNIP (192.0.2.29) と S3 の間の接続を開きます。
8. S2 は NS1 で SNIP 192.0.2.29 に返信する。
9. NS1 は S2 の応答をクライアント CL-TD2 に送信します。

トラフィックドメインでサポートされる **NetScaler ADC** 機能

次のリストの NetScaler ADC 機能は、すべてのトラフィックドメインでサポートされています。

> 重要
>
> 以下に記載されていない NetScaler ADC 機能は、デフォルトのトラフィックドメインでのみサポートされています。

- ARP テーブル
- ND6 テーブル
- ブリッジテーブル
- すべてのタイプの IPv4 アドレスと IPv6 アドレス
- IPv4 および IPv6 ルート
- ACL と ACL6
- PBR & PBR6
- INAT
- RNAT
- RNAT6
- MSR

- MSR6
- ネットプロファイル
- SNMP MIB
- フラグメンテーション
- モニター (スクリプト可能なモニターはサポートされていません)
- コンテンツ スイッチ
- キャッシュリダイレクト
- 永続性 (永続性グループはサポートされていません)
- サービス（ドメインベースのサービスはサポートされていません）
- サービスグループ (ドメインベースのサービスグループはサポートされていません)
- ポリシー (*)
- PING
- TRACEROUTE
- PMTU
- 高可用性 (接続ミラーリングはサポートされていません)
- クラスタ (L2 クラスタでサポートされています。L3 クラスタではサポートされていません)
- クッキー永続性
- MSS
- ロギング (Syslog はサポートされていません)
- サージ保護
- 負荷分散 (次のタイプはサポートされていません)
  - TFTP
  - RTSP
  - Diameter
  - SIP
  - SMPP
- NAT46
- NAT64
- DNS64
- 転送セッションルール
- SNMP

> 注
>
> - * ポリシーには、トラフィックドメインのグローバルバインディングポイントはありません。ただし、ポリシーは、トラフィックドメインの特定の負荷分散仮想サーバーにバインドできます。
> - NetScaler ADC のグローバルサーバー負荷分散 (GSLB) および ADNS 機能は、トラフィックドメインを認識しません。GSLB 設定をすべてのトラフィックドメインで共有する必要がある場合、GSLB 方式静的近接およびラウンドトリップ時間 (RTT) は機能しません。このシナリオの回避策として、RTT および静的

> 近接以外の GSLB 方式を使用できます。詳しくは、http://support.citrix.com/article/CTX202277を参照してください。

トラフィックドメインの設定

NetScaler ADC アプライアンスでのトラフィックドメインの構成は、次のタスクで構成されます。

- **VLAN** を追加します。VLAN を作成し、指定されたインターフェイスをそれらにバインドします。
- トラフィックドメインエンティティを作成し、**VLAN** をバインドします。これには、次の 2 つのタスクが含まれます。
  - ID（整数）で一意に識別されるトラフィックドメインエンティティを作成します。
  - 指定した VLAN をトラフィックドメインエンティティにバインドします。指定された VLAN にバインドされているすべてのインターフェイスは、トラフィックドメインに関連付けられます。1 つのトラフィックドメインに複数の VLAN をバインドできますが、1 つの VLAN を複数のトラフィックドメインの一部にすることはできません。
- トラフィックドメインにフィーチャエンティティを作成します。トラフィックドメインに必要なフィーチャエンティティを作成します。デフォルト以外のトラフィックドメインでサポートされているすべての機能の CLI コマンドおよび設定ダイアログボックスには、トラフィックドメイン識別子（td）と呼ばれるパラメータが含まれています。機能エンティティを設定するときに、エンティティを特定のトラフィックドメインに関連付ける場合は、td を指定する必要があります。td を設定せずに作成したフィーチャエンティティは、自動的にデフォルトのトラフィックドメインに関連付けられます。

このトピックでは、フィーチャエンティティがトラフィックドメインにどのように関連付けられているかを理解するために、「トラフィックドメインの仕組み」という図で説明されているすべてのエンティティの構成手順について説明します。

CLI には、これら 2 つのタスクに対して 2 つのコマンドがありますが、GUI では 1 つのダイアログボックスにこれらのコマンドが組み合わされます。

CLI のプロシージャ

CLI を使用して VLAN を作成し、そのインターフェイスにインターフェイスをバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add vlan** <id>
- **bind vlan** <id> -**ifnum** <slot/port>
- **show vlan** <id>

CLI を使用してトラフィックドメインエンティティを作成し、VLAN をそのエンティティにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns trafficdomain** <td>
- **bind ns trafficdomain** <td> -**vlan** <id>
- **nstrafficdomain** を表示 <td>

CLI を使用してサービスを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add service** <name> <IP> <serviceType> <port> -**td** <id>
- ショーサービス < 名前 >

CLI を使用して負荷分散仮想サーバーを作成し、そのサーバーにサービスをバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add lb vserver** <name> <serviceType> <IPAddress> <port> -**td** <id>
- **bind lb vserver** <name> <serviceName>
- **show lb vserver** <name>

GUI のプロシージャ

GUI を使用して VLAN を作成するには、次の手順を実行します。

[システム] > [ネットワーク] > [**VLAN**] に移動し、[追加] をクリックして、パラメータを設定します。

GUI を使用してトラフィックドメインエンティティを作成するには、次の手順を実行します。

[システム] > [ネットワーク] > [トラフィックドメイン] に移動し、[追加] をクリックし、[トラフィックドメインの作成] ダイアログボックスでパラメータを設定します。

GUI を使用してサービスを作成するには、次の手順を実行します。

[トラフィック管理] > [負荷分散] > [サービス] に移動し、[追加] をクリックしてパラメータを設定します。

GUI を使用して負荷分散仮想サーバーを作成するには

[トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、[追加] をクリックしてパラメータを設定します。

トラフィックドメイン間のエンティティのバインディング

August 15, 2023

あるトラフィックドメインのサービスを、別のトラフィックドメインの仮想サーバーにバインドできます。別のトラフィックドメインの仮想サーバーにバインドされるすべてのサービスは、同じトラフィックドメインに存在する必要があります。

このサポートを設定するには、既存の bind lb vserver コマンドまたは関連する GUI プロシージャを使用します。

この機能により、異なるトラフィックドメイン間の相互作用が容易になります。企業では、サーバーをさまざまなトラフィックドメインにグループ化できます。仮想サーバーは、インターネットに面したトラフィックドメインに作成されます。このトラフィックドメインの仮想サーバーは、別のトラフィックドメインのサーバーを負荷分散するように構成できます。この仮想サーバーは、インターネットからの接続要求を受信して、バインドされたサーバーに転送します。

NetScaler をクラウドインフラストラクチャで使用する場合、各テナントに個別のトラフィックドメインを割り当て、テナントのすべてのリソース（サーバーを含む）をテナントのトラフィックドメインにまとめることができます。テナントごとに、トラフィックドメイン内の負荷分散サーバー用の仮想サーバーが作成されます。これらの仮想サーバーはすべて、インターネットに面した単一のトラフィックドメインにまとめられています。

クラウドサービスプロバイダーの Example-Cloud-A が、NetScaler アプライアンス NS1 に ID が 10、20、30 の 3 つのトラフィックドメインを構成している例を考えてみましょう。

Example-Org-A と Example-Org-B は Example-Cloud-A のテナントです。テナント A にはトラフィックドメイン 20 が割り当てられ、テナント B にはドメイン 30 が割り当てられます。サーバ S1 と S2 はトラフィックドメイン 20 にあり、サーバ S3 と S4 はトラフィックドメイン 30 にあります。

トラフィックドメイン 10 はインターネットに面しています。仮想サーバー LBVS-1 と LBVS-2 はトラフィックドメイン 10 に作成されます。トラフィックドメイン 10 の LBVS-1 は、トラフィックドメイン 20 にあるサーバ S1 と S2 を負荷分散するように設定されています。トラフィックドメイン 10 の LBVS-2 は、トラフィックドメイン 30 にあるサーバー S3 と S4 を負荷分散するように設定されています。

したがって、これらの仮想サーバーは、仮想サーバーとは異なるトラフィックドメインにあるサーバーのインターネット接続要求を受け入れます。

仮想 **MAC** ベースのトラフィックドメイン

August 15, 2023

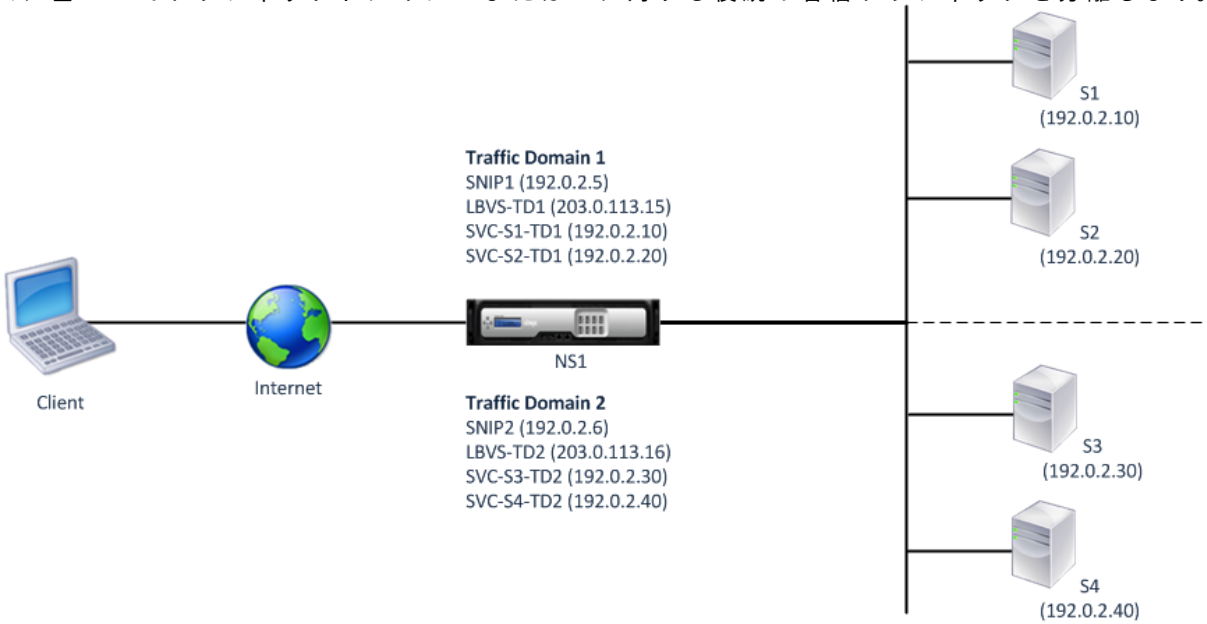
トラフィックドメインを VLAN の代わりに仮想 MAC アドレスに関連付けることができます。次に、NetScaler は、そのドメイン内のネットワークエンティティに対する ARP クエリへのすべての応答として、トラフィックドメインの仮想 MAC アドレスを送信します。その結果、宛先 MAC アドレスはトラフィックドメインの仮想 MAC アドレスであるため、ADC はその後の着信トラフィックを宛先 MAC アドレスに基づいてトラフィックドメインごとに分離できます。トラフィックドメインにエンティティを作成したら、トラフィックドメインレベルの操作を実行することでエンティティを簡単に管理および監視できます。

NetScaler アプライアンス NS1 に ID が 1 と 2 の 2 つのトラフィックドメインが構成されている例を考えてみましょう。NetScaler は仮想 MAC アドレス（仮想 MAC1）を作成し、それをトラフィックドメイン 1 に関連付けます。同様に、NetScaler は別の仮想 MAC アドレス（仮想 MAC2）を作成し、トラフィックドメイン 2 に関連付けます。

トラフィックドメイン 1 では、負荷分散仮想サーバ LBVS-TD1 は、サーバ S1 および S2 間でトラフィックを負荷分散するように設定されます。NetScaler ADC アプライアンスでは、サーバー S1 と S2 は、それぞれサービス SVC1-TD1 と SVC2-TD1 で表されます。サブネット IP アドレス（SNIP）SNIP1 は、NetScaler が S1 および S2 と通信できるように構成されています。仮想 MAC1 はトラフィックドメイン 1 に関連付けられているため、アプライアンスは LBVS-TD1 と SNIP1 のすべての ARP アナウンスと ARP 応答で MAC アドレスとして仮想 MAC1 を送信します。

同様に、トラフィックドメイン 2 では、負荷分散仮想サーバー LBVS-TD2 が S3 と S4 間でトラフィックを負荷分散するように構成されています。NetScaler ADC アプライアンスでは、サーバー S3 と S4 は、それぞれサービス SVC3-TD2 と SVC4-TD2 で表されます。SNIP アドレス SNIP2 は、NetScaler が S3 および S4 と通信できるように構成されています。仮想 MAC2 はトラフィックドメイン 2 に関連付けられているため、アプライアンスは、すべての ARP アナウンスメントおよび LBVS-TD2 および SNIP2 に対する ARP 応答で、仮想 MAC2 を MAC アドレスとして送信します。

NetScaler ADC は、宛先 MAC アドレスが仮想 MAC1 または仮想 MAC2 の場合、宛先 MAC アドレスに基づいてトラフィックドメイン 1 または 2 に対する後続の着信トラフィックを分離します。



次の表に、例で使用される設定を示します。仮想 MAC ベースのトラフィックドメインの設定例。

はじめに

仮想 MAC ベースのトラフィックドメインを設定する前に考慮すべき点は次のとおりです。

1. 仮想 MAC ベースのトラフィックドメインは、ネットワークトラフィックの分離を実現する最も簡単な方法で

す。

2. 仮想 MAC ベースのトラフィックドメインは、VLAN ではなく仮想 MAC アドレスに基づいてネットワークトラフィックを分離するため、NetScaler 上の異なる仮想 MAC ベースのトラフィックドメインに重複した IP アドレスを作成することはできません。
3. NetScaler が L2 モードのみで展開されている場合、仮想 MAC ベースのトラフィックドメインは機能しません。
4. NetScaler では、VLAN ベースのトラフィックドメインと仮想 MAC ベースのトラフィックドメインの両方を共存できます。仮想 MAC ベースのトラフィックドメインは、実際には VLAN ベースのトラフィックドメインにバインドされていないすべての VLAN 上で実行されます。

構成の手順

NetScaler アプライアンスでの仮想 MAC ベースのトラフィックドメインの設定は、次のタスクで構成されます。

- トラフィックドメインエンティティを作成し、仮想 MAC オプションを有効にします。ID（整数値）で一意に識別されるトラフィックドメインエンティティを作成し、仮想 MAC オプションを有効にします。トラフィックドメインエンティティを作成すると、NetScaler は仮想 MAC アドレスを作成し、それをトラフィックドメインエンティティに関連付けます。
- トラフィックドメインにフィーチャエンティティを作成します。これらの機能エンティティを設定するときにトラフィックドメイン識別子 (td) を指定して、トラフィックドメインに必要な機能エンティティを作成します。仮想 MAC ベースのトラフィックドメインで作成された NetScaler 所有のネットワークエンティティは、トラフィックドメインに関連付けられている仮想 MAC アドレスに関連付けられます。次に、NetScaler はトラフィックドメインの仮想 MAC アドレスを ARP アナウンスと ARP レスポンスでこれらのネットワークエンティティに送信します。

CLI のプロシージャ

CLI を使用して仮想 MAC ベースのトラフィックドメインを作成するには:

コマンドプロンプトで入力します。

```
add ns trafficDomain <td> [-vmac ( ENABLED          DISABLED )]
```

-
- nstrafficdomain を表示 <td>

CLI を使用して SNIP アドレスを設定するには:

コマンドプロンプトで入力します。

- add ns ip <IPAddress> <netmask> -type SNIP –td <id>

- show ns ip <IPAddress> -td <id>

CLI を使用してサービスを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add service <name> <IP> <serviceType> <port> -td <id>
- show service <name> -td <id>

CLI を使用して負荷分散仮想サーバーを作成し、そのサーバーにサービスをバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- add lb vserver <name> <serviceType> <IPAddress> <port> -td <id>
- bind lb vserver <name> <serviceName>
- show lb vserver <name> -td <id>

例:

```
> add ns trafficDomain 1 -vmac ENABLED
 Done
> add ns trafficDomain 2 -vmac ENABLED
 Done

> add ns ip 192.0.2.5 255.255.255.0 -type -SNIP -td 1
 Done
> add service  SVC-S1-TD1  192.0.2.10 HTTP 80 -td 1
 Done
> add service  SVC-S2-TD1  192.0.2.20 HTTP 80 -td 1
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.15 80 -td 1
 Done
> bind lb vserver  LBVS-TD1 SVC-S1-TD1
 Done
> bind lb vserver  LBVS-TD1 SVC-S2-TD1
 Done

> add ns ip 192.0.2.6 255.255.255.0 -type -SNIP -td 2
 Done
> add service  SVC-S3-TD2  192.0.2.30 HTTP 80 -td 2
 Done
> add service  SVC-S4-TD2  192.0.2.40 HTTP 80 -td 2
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.16 80 -td 1
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
<!--NeedCopy-->
```


GUI のプロシージャ

GUI を使用して仮想 MAC ベースのトラフィックドメインを作成するには:

1. ［System］> ［Network］> ［Interfaces］に移動します。
2. 詳細ペインで、［Add］をクリックします。
3.「トラフィックドメインの作成」ページで、次のパラメータを設定します。

    - トラフィックドメイン ID*
    - Mac を有効にする

4. ［作成］をクリックします。

GUI を使用して SNIP アドレスを設定するには:

1. [システム] > [ネットワーク] > [IP] > [IPv4] に移動します
2. [ネットワーク] > [IP] > [IPv4] に移動します
3. 詳細ウィンドウで、[追加] をクリックします。
4. Create IP ページで、次のパラメータを設定します。パラメーターの説明を表示するには、対応するフィールドの上にマウスカーソルを合わせます。

    - IP アドレス
    - ネットマスク
    - IP タイプ
    - トラフィックドメイン ID

5. ［作成］をクリックします。

GUI を使用してサービスを作成するには、次の手順を実行します。

1. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
2. 詳細ペインで、［Add］をクリックします。
3. 基本設定ページで、次のパラメータを設定します。パラメーターの説明を表示するには、対応するフィールドの上にマウスカーソルを合わせます。

    - サービス名
    - サーバー
    - プロトコル
    - ポート
    - トラフィックドメイン ID

4. [続行] をクリックし、[完了] をクリックします。
5. 手順 2 ～4 を繰り返して、別のサービスを作成します。
6. ［閉じる］をクリックします。

GUI を使用して負荷分散仮想サーバーを作成し、それにサービスをバインドするには:

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. [負荷分散仮想サーバー] ペインで、[追加] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次のパラメーターを設定します。パラメーターの説明を表示するには、対応するフィールドの上にマウスカーソルを合わせます。

    - 名前
    - IP アドレス
    - プロトコル
    - ポート
    - トラフィックドメイン ID

4. [続行] をクリックし、サービスペインで [>] をクリックします。
5.「サービス」ページで「挿入」をクリックし、仮想サーバーにバインドするサービスのチェックボックスを選択します。
6. [続行] をクリックし、[完了] をクリックします。
7. 手順 2 ～5 を繰り返して、別の仮想サーバーを作成します。

VXLAN

August 15, 2023

NetScaler アプライアンスは、拡張可能な仮想ローカルエリアネットワーク（VXLAN）をサポートしています。VXLAN は、レイヤ 2 フレームを UDP パケットにカプセル化することにより、レイヤ 2 ネットワークをレイヤ 3 インフラストラクチャにオーバーレイします。各オーバーレイネットワークは VXLAN セグメントと呼ばれ、VXLAN ネットワーク識別子 (VNI) と呼ばれる一意の 24 ビットの識別子で識別されます。同じ VXLAN 内のネットワークデバイスだけが相互に通信できます。

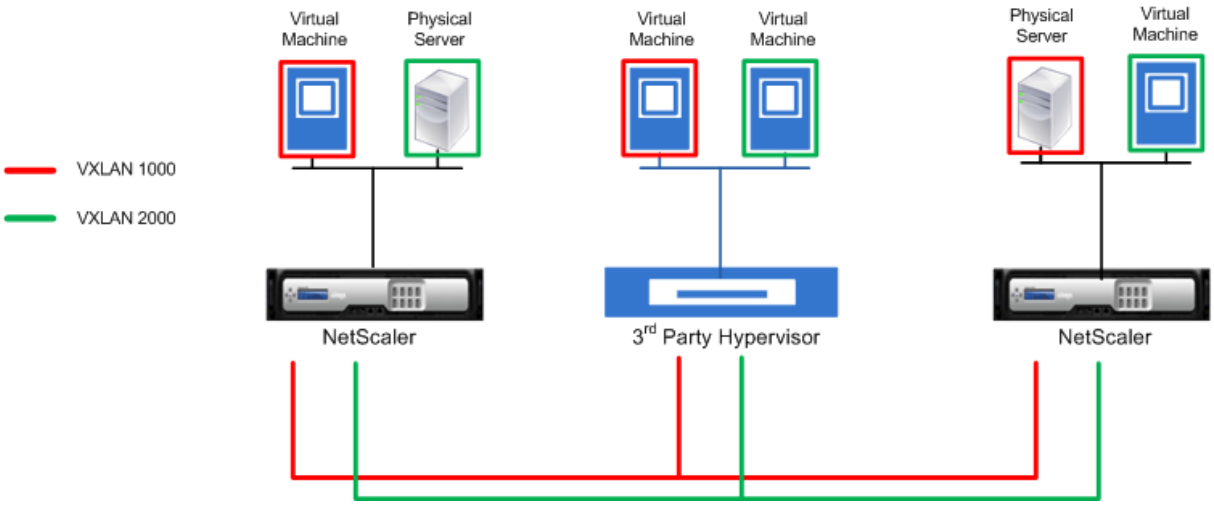
VXLAN は VLAN と同じイーサネットレイヤ 2 ネットワークサービスを提供しますが、拡張性と柔軟性が高くなっています。VXLAN を使用する主な利点は次の 2 つです。

- より高いスケーラビリティ。サーバーの仮想化とクラウドコンピューティングアーキテクチャにより、データセンター内の独立したレイヤー 2 ネットワークの需要が劇的に高まっています。VLAN 仕様では、12 ビットの VLAN ID を使用してレイヤー 2 ネットワークを識別するため、4094 を超える VLAN を拡張することはできません。何千もの分離されたレイヤ 2 ネットワークが必要な場合、この数では不十分な場合があります。24 ビット VNI は、同じ管理ドメイン内で最大 1,600 万の VXLAN セグメントに対応します。
- より高い柔軟性。VXLAN はレイヤー 3 パケットを介してレイヤー 2 データフレームを伝送するため、VXLAN はデータセンターのさまざまな部分や地理的に離れたデータセンターに L2 ネットワークを拡張します。データセンターのさまざまな部分や異なるデータセンターでホストされているが、同じ VXLAN の一部であるアプリケーションは、1 つの隣接するネットワークとして認識されます。

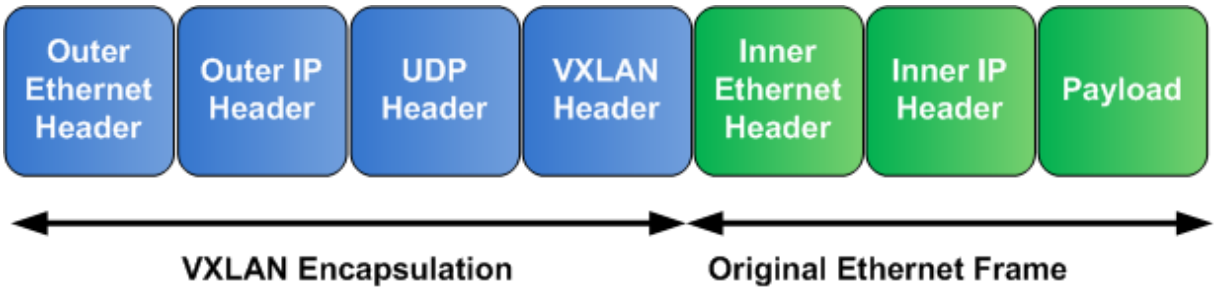
VXLAN の仕組み

VXLAN セグメントは VXLAN トンネルエンドポイント（VTEP）間で作成されます。VTEP は VXLAN プロトコルをサポートし、VXLAN のカプセル化とカプセル化解除を行います。VXLAN セグメントは 2 つの VTEP 間のトンネルと考えることができます。1 つの VTEP が UDP ヘッダーと IP ヘッダーでレイヤ 2 フレームをカプセル化し、トンネル経由で送信します。もう一方の VTEP はパケットを受信してカプセル化を解除し、レイヤ 2 フレームを取得します。NetScaler VTEP の一例です。その他の例としては、サードパーティのハイパーバイザー、VXLAN 対応仮想マシン、VXLAN 対応スイッチがあります。

次の図は、VXLAN トンネルを介して接続された仮想マシンと物理サーバーを示しています。



次の図は、VXLAN パケットの形式を示しています。



NetScaler 上の VXLAN は、レイヤー 2 メカニズムを使用してブロードキャストフレーム、マルチキャストフレーム、および未知のユニキャストフレームを送信します。VXLAN は、これらの L2 フレームを送信するための次のモードをサポートしています。

- ユニキャストモード：このモードでは、NetScaler で VXLAN を構成する際に VTEP の IP アドレスを指定します。NetScaler は、ブロードキャストフレーム、マルチキャストフレーム、および未知のユニキャストフレームをレイヤー 3 を介してこの VXLAN のすべての VTEP に送信します。
- マルチキャストモード：このモードでは、NetScaler で VXLAN を構成するときにマルチキャストグループの IP アドレスを指定します。NetScalers はインターネットグループ管理プロトコル（IGMP）プロトコルをサポートしていません。NetScaler は、アップストリームルーターを利用して、共通のマルチキャストグループ

IP アドレスを共有するマルチキャストグループに参加します。NetScaler は、ブロードキャスト、マルチキャスト、および未知のユニキャストフレームをレイヤー 3 を介してこの VXLAN のマルチキャストグループ IP アドレスに送信します。

レイヤー 2 ブリッジテーブルと同様に、NetScalers は受信した VXLAN パケットの内部ヘッダーと外部ヘッダーに基づいて VXLAN マッピングテーブルを管理します。このテーブルは、リモートホストの MAC アドレスを特定の VXLAN の VTEP IP アドレスにマッピングします。NetScaler は、VXLAN マッピングテーブルを使用してレイヤー 2 フレームの宛先 MAC アドレスを検索します。この MAC アドレスのエントリが VXLAN テーブルに存在する場合、NetScaler は VXLAN プロトコルを使用してレイヤー 3 上のレイヤー 2 フレームを、VXLAN のマッピングエントリに指定されているマッピングされた VTEP IP アドレスに送信します。

VXLAN は VLAN と同様に機能するため、分類パラメーターとして VLAN をサポートする NetScaler 機能のほとんどは VXLAN をサポートしています。これらの機能には、VXLAN VNI を指定するオプションの VXLAN パラメータ設定が含まれます。

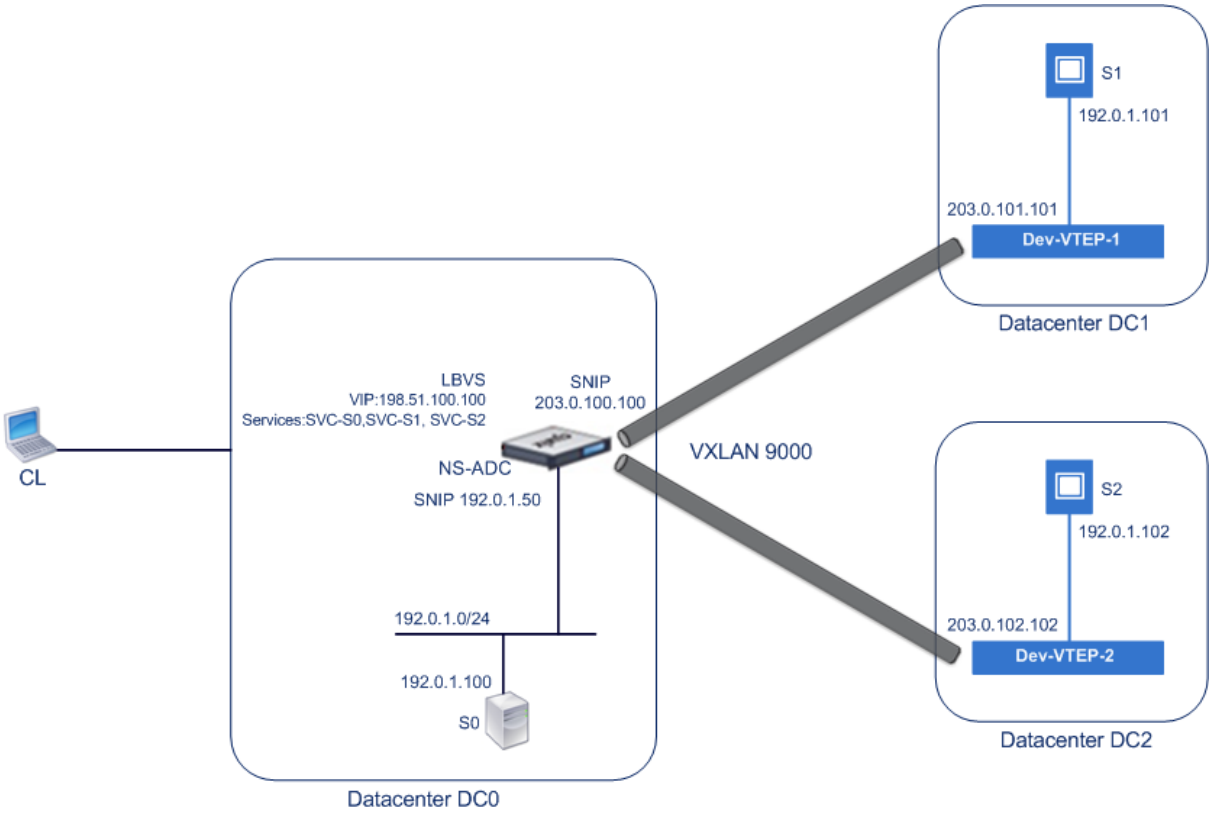
高可用性 (HA) 構成では、VXLAN 構成はセカンダリノードに伝播または同期されます。

VXLAN の使用事例**:** データセンター間の負荷分散

NetScaler の VXLAN 機能を理解するには、Example Corp が www.example.com のサイトをホストしている例を考えてみましょう。アプリケーションの可用性を確保するため、サイトは S0、S1、S2 の 3 台のサーバーでホストされています。NetScaler NS-ADC 上の負荷分散仮想サーバー（LBVS）は、これらのサーバーの負荷分散に使用されます。S0、S1、S2 はそれぞれデータセンター DC0、DC1、DC2 にあります。DC0 では、サーバー S0 は NS-ADC に接続されています。

S0 は物理サーバ、S1 と S2 は仮想マシン (VM) です。S1 はデータセンター DC1 の仮想化ホストデバイス Dev-VTEP-1 で実行され、S2 は DC2 のホストデバイス Dev-VTEP-2 上で実行されます。NS-ADC、Dev-VTEP-1、および Dev-VTEP-2 は VXLAN プロトコルをサポートしています。

S0、S1、S2 は同じプライベートサブネット 192.0.1.0/24 の一部です。S0、S1、S2 は共通のブロードキャストドメインの一部であり、VXLAN 9000 は NS-ADC、Dev-VTEP-1、および Dev-VTEP-2 で設定されています。サーバ S1 と S2 はそれぞれ Dev-VTEP-1 と Dev-VTEP-2 の VXLAN9000 の一部になります。



次の表に、この例で使用される設定を示します。
VXLAN 設定。

NS-ADC 上のサービス SVC-S0、SVC-S1、および SVC-S2 は、S0、S1、および S2 を表します。これらのサービスが設定されるとすぐに、NS-ADC は S0、S1、S2 の ARP 要求をブロードキャストして IP と MAC のマッピングを解決します。これらの ARP リクエストは VXLAN 9000 経由で Dev-VTEP-1 と Dev-VTEP-2 にも送信されます。

S2 の ARP 要求を解決するためのトラフィックフローは次のとおりです。

1. NS-ADC は、IP と MAC のマッピングを解決するために S2 に ARP 要求をブロードキャストします。このパケットには次の内容が含まれます。
   - 送信元 IP アドレス = サーバー用のサブネット IP アドレス SNIP（192.0.1.50）
   - 送信元 MAC アドレス = パケットの送信元となる NS-ADC のインタフェースの MAC アドレス = NS-MAC-1
2. NS-ADC は、パケットを次のヘッダーでカプセル化して、VXLAN 9000 経由で送信する ARP パケットを準備します。
   - ID (VNI) が 9000 の VXLAN ヘッダー
   - 標準 UDP ヘッダー、UDP チェックサムは 0×0000、宛先ポートは 4789 に設定されています。
3. NS-ADC は、生成されたカプセル化されたパケットを VXLAN-9000 の Dev-VTEP-1 と Dev-VTEP-2 に送信します。カプセル化されたパケットには次のものが含まれます。

- 送信元 IP アドレス = SNIP-VTEP-0 (203.0.100.100)。

4. Dev-VTEP-2 は UDP パケットを受信して UDP ヘッダーをカプセル化解除します。Dev-VTEP-2 は、このパケットが VXLAN 関連のパケットであることを学習します。次に、Dev-VTEP-2 は VXLAN ヘッダーのカプセル化を解除し、パケットの VXLAN ID を学習します。結果のパケットは、ステップ 1 と同じ S2 の ARP 要求パケットです。
5. Dev-VTEP-2 は VXLAN パケットの内部ヘッダーと外部ヘッダーから、VXLAN9000 の MAC アドレス (NS-MAC-1) と SNIP-VTEP-0 (203.0.100.100) のマッピングを示すエントリを VXLAN マッピングテーブルに作成します。
6. Dev-VTEP-2 は ARP パケットを S2 に送信します。S2 の応答パケットは Dev-VTEP-2 に到達します。Dev-VTEP-2 は VXLAN マッピングテーブルを検索し、宛先 MAC アドレス NS-MAC-1 と一致するものを取得します。Dev-VTEP-2 は、VXLAN 9000 経由で SNIP-VTEP-0 (203.0.100.100) 経由で NS-MAC-1 にアクセスできることを認識するようになりました。
7. S2 はその MAC アドレス（MAC-S2）で応答します。ARP 応答パケットには次の内容が含まれます。

   - 送信先 IP アドレス = サブネット IP アドレスサーバー用 SNIP アドレス (192.0.1.50)
   - 宛先 MAC アドレス = NS-MAC-1

8. S2 の応答パケットは Dev-VTEP-2 に到達します。Dev-VTEP-2 は VXLAN マッピングテーブルを検索し、宛先 MAC アドレス NS-MAC-1 と一致するものを取得します。Dev-VTEP-2 は、VXLAN 9000 経由で SNIP-VTEP-0 (203.0.100.100) 経由で NS-MAC-1 にアクセスできることを認識するようになりました。Dev-VTEP-2 は ARP 応答を VXLAN ヘッダーと UDP ヘッダーでカプセル化し、結果のパケットを NS-ADC の SNIP-VTEP-0 (203.0.100.100) に送信します。
9. NS-ADC は、パケットを受信すると、VXLAN と UDP ヘッダーを削除してパケットのカプセル化を解除します。結果のパケットは S2 の ARP 応答です。NS-ADC は、S2 の MAC アドレス (MAC-S2) の VXLAN マッピングテーブルを VXLAN 9000 の Dev-VTEP-2 の IP アドレス (203.0.102.102) に更新します。NS-ADC は、S2 の IP アドレス（192.0.1.102）の ARP テーブルも S2 の MAC アドレス（MAC-S2）で更新します。

この例では、仮想サーバー LBVS の負荷分散のトラフィックフローは次のとおりです。

1. クライアント CL は、NS-ADC の LBVS に要求パケットを送信します。リクエストパケットには次のものが含まれます。

   - 送信元 IP アドレス = クライアント CL の IP アドレス (198.51.100.90)
   - 宛先 IP アドレス = LBVS の IP アドレス (VIP) = 198.51.110.100

2. NS-ADC の LBVS は要求パケットを受信し、その負荷分散アルゴリズムがデータセンター DC2 のサーバー S2 を選択します。
3. NS-ADC は要求パケットを処理し、宛先 IP アドレスを S2 の IP アドレスに変更し、送信元 IP アドレスを NS-ADC で設定されたサブネット IP（SNIP）アドレスのいずれかに変更します。リクエストパケットには次のものが含まれます。

   - 送信元 IP アドレス = NS-ADC 上のサブネット IP アドレス = SNIP-for-server (192.0.1.50)
   - 宛先 IP アドレス = S2 の IP アドレス (192.0.1.102)

4. NS-ADC は、ブリッジテーブルで S2 の VXLAN マッピングエントリを検出します。このエントリは、VXLAN 9000 経由の Dev-VTEP-2 経由で S2 にアクセスできることを示しています。
5. NS-ADC は、パケットを次のヘッダーでカプセル化して、VXLAN 9000 経由で送信するパケットを準備します。

    - ID (VNI) が 9000 の VXLAN ヘッダー
    - 標準 UDP ヘッダー、UDP チェックサムは 0×0000、宛先ポートは 4789 に設定されています。

6. NS-ADC は、生成されたカプセル化されたパケットを Dev-VTEP-2 に送信します。リクエストパケットには次のものが含まれます。

    - 送信元 IP アドレス = SNIP アドレス = SNIP-VTEP-0 (203.0.100.100)
    - 宛先 IP アドレス = Dev-VTEP-2 (203.0.102.102) の IP アドレス

7. Dev-VTEP-2 は UDP パケットを受信して UDP ヘッダーをカプセル化解除します。Dev-VTEP-2 は、このパケットが VXLAN 関連のパケットであることを学習します。次に、Dev-VTEP-2 は VXLAN ヘッダーのカプセル化を解除し、パケットの VXLAN ID を学習します。結果のパケットは、ステップ 3 と同じパケットです。
8. 次に、Dev-VTEP-2 はパケットを S2 に転送します。
9. S2 は要求パケットを処理し、NS-ADC の SNIP アドレスに応答を送信します。応答パケットには次の内容が含まれます。

    - ソース IP アドレス = S2 の IP アドレス (192.0.1.102)
    - 宛先 IP アドレス = NS-ADC 上のサブネット IP アドレス = SNIP-for-Servers (192.0.1.50)

10. Dev-VTEP-2 は、NS-ADC が手順 4 と 5 で要求パケットをカプセル化したのと同じ方法で応答パケットをカプセル化します。次に、Dev-VTEP-2 はカプセル化された UDP パケットを NS-ADC の SNIP アドレス SNIP サーバ用 SNIP アドレス (192.0.1.50) に送信します。
11. NS-ADC は、カプセル化された UDP パケットを受信すると、Dev-VTEP-2 がステップ 7 でパケットをカプセル化解除したのと同じ方法で、UDP ヘッダーと VXLAN ヘッダーを削除してパケットのカプセル化を解除します。生成されるパケットは、ステップ 9 と同じ応答パケットです。
12. 次に、NS-ADC はセッションテーブルを使用して仮想サーバー LBVS の負荷分散を行い、応答パケットをクライアント CL に転送します。応答パケットには次の内容が含まれます。

    - 送信元 IP アドレス = クライアント CL の IP アドレス (198.51.100.90)
    - 宛先 IP アドレス = LBVS (198.51.110.100) の IP アドレス (VIP)

VXLAN を設定する際に考慮すべきポイント

NetScaler で VXLAN を構成する前に、次の点を考慮してください。

- NetScaler では最大 2048 の VXLAN を構成できます。
- VXLAN はクラスタではサポートされません。
- リンクローカル IPv6 アドレスは、VXLAN ごとに設定できません。

- NetScalers は、マルチキャストグループを形成するためのインターネットグループ管理プロトコル（IGMP）プロトコルをサポートしていません。NetScalers は、アップストリームルーターの IGMP プロトコルを利用して、共通のマルチキャストグループ IP アドレスを共有するマルチキャストグループに参加します。VXLAN ブリッジテーブルエントリを作成するときにマルチキャストグループ IP アドレスを指定できますが、マルチキャストグループはアップストリームルータで設定する必要があります。NetScaler は、ブロードキャスト、マルチキャスト、および未知のユニキャストフレームをレイヤー 3 を介してこの VXLAN のマルチキャストグループ IP アドレスに送信します。次に、アップストリームルータは、マルチキャストグループに含まれるすべての VTEP にパケットを転送します。
- VXLAN のカプセル化により、各パケットに 50 バイトのオーバーヘッドが追加されます。

  外部イーサネットヘッダー (14) + UDP ヘッダー (8) + IP ヘッダー (20) + VXLAN ヘッダー (8) = 50 バイト

  フラグメンテーションやパフォーマンスの低下を防ぐには、VXLAN パケットの 50 バイトのオーバーヘッドを処理できるように、VXLAN VTEP デバイスを含む VXLAN パスウェイ内のすべてのネットワークデバイスの MTU 設定を調整する必要があります。

  重要：ジャンボフレームは、NetScaler VPX 仮想アプライアンス、NetScaler SDX アプライアンス、および NetScaler MPX 15000/17000 アプライアンスではサポートされていません。これらのアプライアンスは 1500 バイトの MTU サイズしかサポートしておらず、VXLAN パケットの 50 バイトのオーバーヘッドを処理するように調整することはできません。これらのアプライアンスのいずれかが VXLAN 経路にあるか、VXLAN VTEP デバイスとして機能している場合、VXLAN トラフィックは断片化されるか、パフォーマンスが低下する可能性があります。
- NetScaler SDX アプライアンスでは、VXLAN パケットに対して VLAN フィルタリングは機能しません。
- VXLAN に MTU 値を設定することはできません。
- インターフェイスを VXLAN にバインドすることはできません。

構成の手順

NetScaler アプライアンスでの VXLAN の構成は、次のタスクで構成されます。

- **VXLAN** エンティティを追加します。正の整数で一意に識別される VXLAN エンティティを作成します。これは VXLAN ネットワーク識別子 (VNI) とも呼ばれます。このステップでは、VXLAN プロトコルが実行されているリモート VTEP の宛先 UDP ポートを指定することもできます。デフォルトでは、VXLAN エンティティの宛先 UDP ポートパラメータは 4789 に設定されています。この UDP ポート設定は、この VXLAN のすべてのリモート VTEP の設定と一致する必要があります。VLAN をこの VXLAN にバインドすることもできます。バインドされたすべての VLAN のトラフィック（ブロードキャスト、マルチキャスト、未知のユニキャストを含む）は、この VXLAN 上で許可されます。VXLAN にバインドされている VLAN がない場合、NetScaler はこの VXLAN 上の他の VXLAN に含まれていないすべての VLAN のトラフィックを許可します。
- ローカル **VTEP IP** アドレスと **VXLAN** エンティティをバインドします。設定した SNIP アドレスのいずれかを VXLAN にバインドして、発信 VXLAN パケットを送信します。

- ブリッジ可能なエントリを追加します。作成する VXLAN の VXLAN ID とリモート VTEP IP アドレスを指定するブリッジ可能なエントリを追加します。
- （オプション）さまざまな機能エンティティを設定済みの **VXLAN** にバインドします。VXLAN は VLAN と同様に機能しますが、分類パラメーターとして VLAN をサポートする NetScaler 機能のほとんどは、VXLAN もサポートしています。これらの機能には、VXLAN VNI を指定するオプションの VXLAN パラメータ設定が含まれます。
- （オプション）**VXLAN** マッピングテーブルを表示します。リモートホストの MAC アドレスから特定の VXLAN の VTEP IP アドレスへのマッピングエントリを含む VXLAN マッピングテーブルを表示します。つまり、VXLAN マッピングとは、特定の VXLAN 上の VTEP を介してホストにアクセスできるということです。NetScaler は VXLAN マッピングを学習し、受信した VXLAN パケットからマッピングテーブルを更新します。NetScaler は、VXLAN マッピングテーブルを使用してレイヤー 2 フレームの宛先 MAC アドレスを検索します。この MAC アドレスのエントリが VXLAN テーブルに存在する場合、NetScaler は VXLAN プロトコルを使用してレイヤー 3 上のレイヤー 2 フレームを、VXLAN のマッピングエントリに指定されているマッピングされた VTEP IP アドレスに送信します。

CLI のプロシージャ

CLI を使用して VXLAN エンティティを追加するには:

コマンドプロンプトで、次のように入力します。

- **add vxlan** <id>
- **show vxlan**<id>

CLI を使用してローカル VTEP IP アドレスを VXLAN にバインドするには:

コマンドプロンプトで、次のように入力します。

- **bind vxlan** <id> -**SrcIP** <IPaddress>
- **show vxlan** <id>

CLI を使用してブリッジテーブルを追加するには:

コマンドプロンプトで、次のように入力します。

- **add bridgetable** -**mac** <macaddress> -**vxlan** <ID> -**vtep** <IPaddress>
- ブリッジテーブルを表示

コマンドラインを使用して VXLAN 転送テーブルを表示するには:

コマンドプロンプトで入力します。

- ブリッジテーブルを表示

GUI のプロシージャ

VXLAN エンティティを追加し、GUI を使用してローカル VTEP IP アドレスをバインドするには:

[システム] > [ネットワーク] > [**VXLAN**] に移動し、新しい VXLAN エンティティを追加するか、既存の VXLAN エンティティを変更します。

GUI を使用してブリッジテーブルを追加するには:

[システム] > [ネットワーク] > [ブリッジテーブル] に移動し、VXLAN ブリッジテーブルエントリを追加または変更するときに次のパラメータを設定します。

- MAC
- VTEP
- VXLAN ID

GUI を使用して VXLAN 転送テーブルを表示するには:

[システム**]** > **[**ネットワーク**]**> **[**ブリッジテーブル**]** に移動します。

```
Example
> add vxlan 9000
Done
> bind vxlan 9000 -srcIP 203.0.100.100

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.101.101

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.102.102

Done
```

VXLAN での **IPv6** ダイナミックルーティングプロトコルのサポート

NetScaler アプライアンスは、VXLAN の IPv6 動的ルーティングプロトコルをサポートしています。VTYSH コマンドラインから VXLAN 上にさまざまな IPv6 ダイナミックルーティングプロトコル（OSPFv3、RIPng、BGP など）を設定できます。VXLAN の IPv6 動的ルーティングプロトコルを有効または無効にするためのオプション IPv6 動的ルーティングプロトコルが VXLAN コマンドセットに追加されました。VXLAN で IPv6 動的ルーティングプロトコルを有効にした後、IPv6 動的ルーティングプロトコルに関連するプロセスを VTYSH コマンドラインを使用して VXLAN 上で起動する必要があります。

CLI を使用して VXLAN で IPv6 ダイナミックルーティングプロトコルを有効にするには、次の手順を実行します。

- **add vxlan** <ID> [-**ipv6DynamicRouting** (**ENABLED** | ** 無効 **)]
- **show vxlan**


```
In the following sample configuration, VXLAN-9000 is created and has
    IPv6 dynamic routing protocols enabled on it. Then, using the VTYSH
    command line, process for the IPv6 OSPF protocol is started on the
    VXLAN.

> add vxlan 9000 -ipv6DynamicRouting ENABLED

Done
> bind vxlan 9000 -srcIP 203.0.100.100

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.101.101

Done
> VTYSH
NS# configure terminal
NS(config)# ns IPv6-routing
NS(config)# interface VXLAN-9000
NS(config-if)# ipv6 router OSPF area 3
```

VXLAN-VLAN マップを使用した複数の企業からクラウドへの **VLAN** の拡張

CloudBridge Connector トンネルは、企業の VLAN をクラウドに拡張するために使用されます。複数の企業から拡張された VLAN には、VLAN ID が重複している場合があります。クラウド内の固有の VXLAN にマッピングすることで、各企業の VLAN を分離できます。クラウド内の CloudBridge Connector エンドポイントである NetScaler アプライアンスでは、企業の VLAN をクラウド内の固有の VXLAN にリンクする VXLAN-VLAN マップを構成できます。VXLAN は、企業の複数の VLAN を CloudBridge Connector から同じ VXLAN に拡張するための VLAN タギングをサポートしています。

複数の企業の VLAN をクラウドに拡張するには、次のタスクを実行します。

1. VXLAN-VLAN マップを作成します。
2. VXLAN-VLAN マップを、クラウド上の NetScaler アプライアンス上のネットワークブリッジベースまたは PBR ベースの CloudBridge Connector トンネル構成にバインドします。
3. （オプション）VXLAN 設定で VLAN タギングを有効にします。

CLI のプロシージャ

CLI を使用して VXLAN-VLAN マップを追加するには:

- **VXLAN** マップの追加 <name>
- **show vxlanVlanMap** <name>

CLI を使用して VXLAN および VLAN を VXLAN VLAN マップにバインドするには、次の手順を実行します。

- **bind vxlanVlanMap** <name> [-**vxlan** \<positive_integer> -**vlan** \<int[-int]> …]
- **show vxlanVlanMap** <name>

CLI を使用して VXLAN-VLAN マップをネットワークブリッジベースの CloudBridge Connector トンネルにバインドするには:

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

新しいネットワークブリッジを追加する場合:

- **add netbridge** <name> [-**vxlanVlanMap** \<string>]
- **show netbridge** <name>

既存のネットワークブリッジを再設定する場合:

- **set netbridge** <name> [-**vxlanVlanMap** \<string>]
- **show netbridge** <name>

CLI を使用して VXLAN-VLAN マップを PBR ベースの CloudBridge Connector トンネルにバインドするには:

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

新しい PBR を追加する場合:

- **add pbr** <name> **ALLOW** (-**ipTunnel** <ipTunnelName> [-**vxlanVlanMap** \<name>])
- **show pbr** <name>

既存の PBR を再設定する場合:

- **set pbr** <name> **ALLOW** (-**ipTunnel** <ipTunnelName> [-**vxlanVlanMap** \<name>])
- **show pbr** <name>

CLI を使用して VXLAN に関連するパケットに VLAN タグを含めるには:

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

新しい VXLAN を追加する場合:

- **add vxlan** <vnid> -**vlanTag** (**ENABLED** | **DISABLED**)
- **show vxlan** <vnid>

既存の VXLAN を再設定する場合:

- **set vxlan** <vnid> -**vlanTag** (**ENABLED** | **DISABLED**)
- **show vxlan** <vnid>

GUI のプロシージャ

GUI を使用して VXLAN-VLAN マップを追加するには:

[システム] > [ネットワーク] > [**VXLAN VLAN** マップ] に移動し、VXLAN VLAN マップを追加します。

GUI を使用して VXLAN-VLAN マップをネットブリッジベースの CloudBridge Connector トンネルにバインドするには:

[システム] > [**CloudBridge Connector**] > [ネットワークブリッジ] に移動し、新しいネットワークブリッジを追加するか、既存のネットワークブリッジを再設定するときに、**VXLAN VLAN** ドロップダウンリストから **VXLAN-VLAN** マップを選択します。

GUI を使用して VXLAN-VLAN マップを PBR ベースの CloudBridge Connector トンネルにバインドするには:

新しい PBR を追加するか、既存の PBR を再設定するときに、**[**** システム] > [ネットワーク] > [PBR] に移動し、[ポリシーベースルーティング (PBR)] タブで **VXLAN VLAN** ドロップダウンリストから VXLAN-VLAN** マップを選択します。

GUI を使用して VXLAN に関連するパケットに VLAN タグを含めるには:

[システム] > [ネットワーク] > [**VXLAN**] に移動し、新しい VXLAN を追加するときに 内部 **VLAN** タグ付け を有効化するか、既存の VXLAN を再構成します。

```
> add vxlanVlanMap VXLANVLAN-DC1

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 3000 -vlan 3

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 3500 -vlan 4

Done

>add vxlanVlanMap VXLANVLAN-DC2

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 8000 -vlan 3 4

Done

> set pbr PBR-CBC-DC-1-CLOUD ALLOW  -ipTunnel  CBC-DC-1-CLOUD -
    vxlanVlanMap VXLANVLAN-DC1

Done

> set pbr PBR-CBC-DC-2-CLOUD ALLOW  -ipTunnel  CBC-DC-2-CLOUD -
    vxlanVlanMap VXLANVLAN-DC2

```



```
27 Done
```

Geneve トンネル

August 15, 2023

NetScaler ADC アプライアンスは、RFC 8926 で定義されている汎用ネットワーク仮想化カプセル化（Geneve）プロトコルをサポートしています。
サーバ仮想化とクラウドコンピューティングアーキテクチャにより、データセンターにおける分離されたレイヤ 2 ネットワークの需要が高まっています。

VLAN 制限の 4094 は不十分であることが判明しており、この制限を克服するために VXLAN や NVGRE などのカプセル化プロトコルが導入されました。これらのプロトコルは、主にコントロールプレーンの実装が異なります。Geneve プロトコルは、コントロールプレーンの仕様を定義していません。このプロトコルは、コントロールプレーンの仕様を定義するために実装に任されています。

Geneve プロトコルは、レイヤー 2 フレームを UDP パケットにカプセル化することにより、レイヤー 3 インフラストラクチャ上でレイヤー 2 オーバーレイネットワークを作成することを目的としたカプセル化テクノロジーです。

VNID と呼ばれる一意の 24 ビット識別子は、各 VLAN を識別します。同じセグメント ID (VNID) 内でのみ相互に通信できます。NetScaler ADC アプライアンスは、UDP ポート 6081 での Geneve カプセル化をサポートします。

作成できる Geneve トンネルには次の 2 つのタイプがあります。

- トンネルは、L2 または L3 モードで既存の VLAN を拡張できます。L2 モードでは、VLAN とトンネルの間でブリッジングが行われ、ブリッジテーブルのエントリが更新されます。

  L3 モードでは、プロキシ ARP が有効になり、クライアント/サーバアドレスの MAC アドレスとトンネル情報を学習します。ARP テーブルには、対応する MAC とトンネルの情報が含まれます。

- Geneve トンネルは、ポリシーベースルート（PBR）を使用することにより、L3 モードでさまざまな VLAN と連携できます。
  Geneve トンネルセグメントで到達可能なホストにパケットを送信する必要がある場合、NetScaler ADC アプライアンスはそのパケットを Geneve トンネルヘッダーにカプセル化し、トンネルエンドポイントに送信します。

NetScaler ADC は、トンネルエンドポイントとしても機能します。トンネルエンドポイントは、Geneve トンネルを発信および終端します。レイヤー 2 モードがオンになっている場合、NetScaler ADC アプライアンスはトンネルエンドポイントとして機能し、VLAN と Geneve トンネル間でパケットをブリッジします。NetScaler ADC は、MAC アドレスに到達可能な VNID とトンネルエンドポイントを学習します。次に、この情報をブリッジングテーブルに保存します。

Geneve トンネルは、NetScaler ADC 管理パーティション、NetScaler ADC の高可用性セットアップ、および NetScaler ADC クラスタセットアップでサポートされています。

高可用性セットアップでは、Geneve トンネル構成がセカンダリノードに伝播または同期されます。クラスタセットアップでは、Geneve トンネル構成 (ストライプ) は同一であり、すべてのクラスタノードに存在します。

Geneve トンネルの構成

NetScaler ADC アプライアンスでの Geneve トンネルの構成は、次のタスクで構成されます。

- プロトコル付き IP トンネルを追加する
- ネットブリッジを追加する
- Geneve トンネルをネットブリッジに縛り付けろ

CLI を使用して **Geneve** プロトコルで **IP** トンネルを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add iptunnel** <name> <remote> <remoteSubnetMask> <local> -**protocol** <Geneve> -**destPort** <port> -**tosInherit** (**ENABLED** | **DISABLED**) -**vlanTagging** (**ENABLED** | **DISABLED**) -**vnid** <positive_integer>
- **show iptunnel**

CLI を使用してネットブリッジを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add netbridge** <name>
- **show netbridge**

CLI を使用して **Geneve** トンネルを **netbridge** にバインドするには、以下を実行します。

コマンドプロンプトで入力します。

- **bind netbridge** <name> -**vlan** <Vlan ID> -**tunnel** <tunnel name>
- **show netbridge**

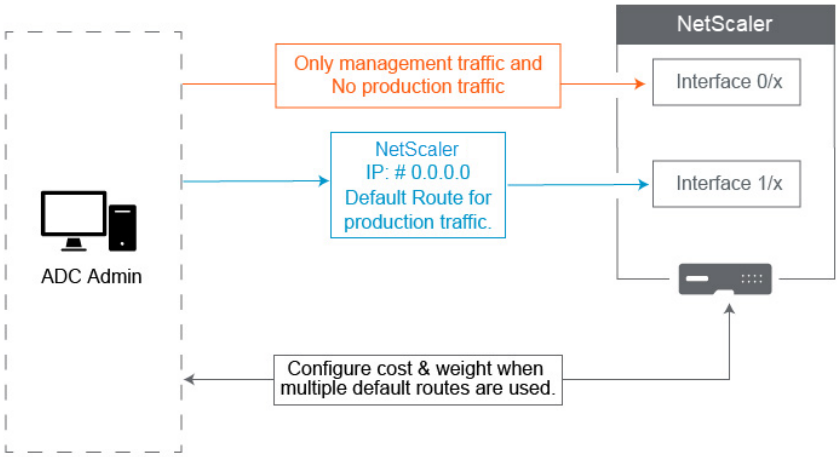
ネットワーク構成のベストプラクティス

August 15, 2023

次のセクションでは、NetScaler アプライアンスのネットワーク機能を構成するためのベストプラクティスについて説明します。

ルーティングとデフォルトルート

NetScaler アプライアンスでレイヤー 3 機能を構成するためのベストプラクティスを以下に示します。



- **NetScaler** アプライアンスまたは **NetScaler SDX** アプライアンスのインターフェイス**0/x**をプロダクショントラフィックに使用しないでください。MPX または SDX では、0/x 名前の付いたインターフェイスは管理インターフェイスを指します。だからといって、管理にこれらのインターフェースを使用しなければならないわけではありません。つまり、これらのインターフェースはプロダクショントラフィック用には設計されていないということです。1 Gbps のスループットを維持するのに必要なハードウェアバッファや最適化機能はありません。したがって、デフォルトルートが NSIP と同じサブネットにある場合は、デフォルトルートを変更するか、管理ネットワークの1/xインターフェイスを使用する必要があります。1/xインターフェイスは運用環境の 1 Gbps トラフィック用に完全に最適化されているためです。

  > 注:
  >
  > これは NetScaler VPX アプライアンスには適用されません。

  - オプション **1**。インターフェイスに接続しないでください 0/x —0/1インターフェイスからケーブルを取り外します。NetScaler は他のインターフェースで NSIP を受信します。(注: SVM と XenServer は0/xインターフェイスとしか通信できないため、これは SDX ではオプションではありません)
  - オプション **2**: 次のセクションで説明するように、デフォルトルートを別のインターフェイスに変更します。

- デフォルトゲートウェイ **(**ルート **0.0.0.0)** は、どの**0/x**インターフェイスにも配置せず、プロダクションネットワークに配置する必要があります。NetScaler を初めてセットアップするときに、NSIP、サブネットマスク、ゲートウェイアドレスの入力を求められます。これが管理者にもたらす問題は、インターフェイス 0/1 を使用して管理ネットワーク上にデフォルトルートを設定しただけということです。

  - ルートを確認するには、CLI で実行します。デフォルトゲートウェイは、`show route` ネットワークとネットマスクが 0.0.0.0 の行の IP です。ゲートウェイが回線 1 にある例を次に示します。


```
> sh route
        Network          Netmask          Gateway/OwnedIP
           State   Traffic Domain  Type
        -------          -------          ---------------
           -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.25.213.65     UP
         0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        UP
         0               PERMANENT
3)      10.25.213.64     255.255.255.192  10.25.213.68     UP
         0               DIRECT
4)      172.16.0.0       255.255.255.0    172.16.0.1       UP
         0               DIRECT

<!--NeedCopy-->
```

- デフォルトゲートウェイに使用されているインターフェイスと VLAN を確認するには、CLI の`sh arp`を使用して ARP テーブルを確認します。`show arp | grep 10.25.213.65`を使用して特定の IP を検索することもできます。ゲートウェイ 10.25.213.65 がインターフェイス 1/1 と VLAN 1 を使用している例を次に示します。

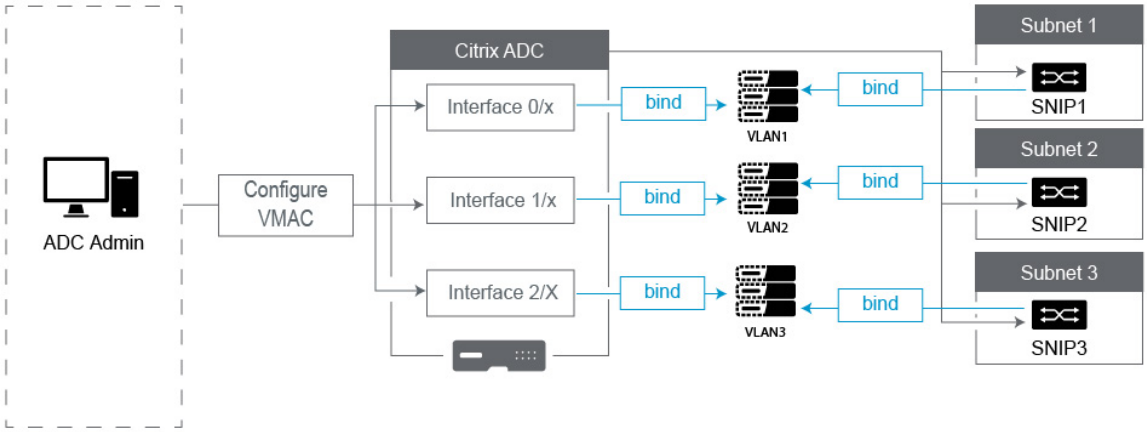
```
> sh arp
        IP               MAC                Iface VLAN
           Origin     TTL     Traffic Domain
        --               ---                ----- ----
           ------     ---     --------------
1)      127.0.0.1        02:00:18:a4:00:1e  LO/1  1
   PERMANENT  N/A    0
2)      10.25.213.70     02:00:0f:46:00:28  1/1   1
   DYNAMIC    967    0
3)      10.25.213.68     02:00:18:a4:00:1e  LO/1  1
   PERMANENT  N/A    0
4)      10.25.213.67     02:00:0f:46:00:28  1/1   1
   DYNAMIC    641    0
5)      10.25.213.65     00:08:e3:ff:fd:90  1/1   1
   DYNAMIC    483    0
<!--NeedCopy-->
```

- デフォルトルートをプロダクションサブネットとインターフェースのゲートウェイを使用するように変更します。管理ネットワークが 10.0.0.0/24 でゲートウェイ 10.0.0.1 で、運用ネットワークが 10.1.1.0/24 でゲートウェイ 10.1.1.1 であると仮定します。以下のように設定してください。
  - SNIP: (管理アクセスが無効) 10.1.1.2
  - NSIP: (管理アクセスが有効) 10.0.0.2
  - デフォルトルート:0.0.0.0 0.0.0 10.1.1.1 ([システム] > [ネットワーク] > [ルート])。これは NSIP ネットワークの代わりに SNIP ネットワーク上のルーターを使用します。

> 注記:
>
> デフォルトゲートウェイを変更すると、静的ルート、ポリシーベースルートを設定するか、MAC ベースの転送を有効にしない限り、管理トラフィックが中断する可能性があります。

インターフェイス、チャネル、**VLAN**

NetScaler アプライアンスでレイヤー 2 機能を構成するためのベストプラクティスを以下に示します。



- VLAN 1 を含め、複数のインターフェイス/チャネルを同じ **VLAN** に接続しないでください。
  - VLAN を適切に設定しないと、ネットワーク内で予期しないパケットルーティングが発生し、同じ VLAN（ネイティブまたはタグ付き）のアクティブインターフェイスが複数ある場合にレイヤ 2 ループが発生する可能性があります。
  - デフォルトでは、すべてのインターフェイスとチャネルはネイティブ VLAN 1 上にあります。これにより、次の 2 つの問題が発生する可能性があります。
    * NetScaler は、受信したすべてのトラフィックが同じネットワーク上にあると見なすため、任意のインターフェイスを使用してトラフィックを送信します。データを送信したインターフェイスに別のネイティブ VLAN がある場合、トラフィックは期待どおりにルーティングされません。
    * NetScaler は、あるポートでブロードキャストパケットを受信すると、別のポートで再送信する可能性があります。両方のスイッチポートが同じ VLAN 上にある場合は、レイヤ 2 ループが作成されたことになります。
  - VLAN 1 からインターフェイス/チャネルを削除するには:
    * スイッチインターフェイス/ポートチャネルでネイティブ VLAN を使用していない場合。NetScaler インターフェイス/チャネルのネイティブ VLAN を 999 などの未使用の VLAN 番号に変更します。

複数のチャネルまたはインターフェイスに同じ未使用の VLAN 番号を使用しないでください。レイヤ 2 ループが発生するためです。

* スイッチインターフェイス/ポートチャネルでネイティブ VLAN を使用している場合。NetScaler インターフェイス/チャネルのネイティブ VLAN をそれに合わせて変更します。ただし、同じ VLAN に複数のアクティブなインターフェイスまたはチャネルがあると、レイヤ 2 ループが発生しないよう注意してください。

* ネイティブ VLAN は削除できません。代わりに、これを変更したり、インターフェイスまたはチャネルに tagAll を設定したりできます。スイッチポートにタグなしのネイティブ VLAN が設定されていない場合は、インターフェイスで tagall を有効にして High Availability ハートビートパケットがタグ付けされるようにします。

– インターフェイス上のネイティブ VLAN を表示するには、CLI で`sh interface`を実行します。これにより、インターフェースが TAGALL オプションを使用しているかどうかもわかります。

- インターフェイスを **VLAN** にバインドする -NetScaler は、デフォルトでは新しい VLAN をインターフェイスに接続しません。つまり、インターフェイスにバインドするまでは VLAN は使用されません。新しい VLAN がインターフェイスにバインドされておらず、その VLAN がタグ付けされている場合、NetScaler はその VLAN からのすべての受信トラフィックをドロップします。また、同じ VLAN を複数のインターフェイスにバインドしないでください。

  – サブネットを VLAN にバインドします。NetScaler は一般的なルーターのようには機能しません。ほとんどのルータはインターフェイスに IP をアタッチします。NetScaler では、特に構成されていない限り、IP はどのインターフェイスでもフローティングします。そのため、NetScaler が特定の VLAN を介して送信するようにしたいサブネットは、特に NetScaler がそのトラフィックを開始する場合は、そのサブネット内の SNIP を VLAN にバインドする必要があります。

  – これに対してよく聞かれる議論は、以前は正常に機能していましたが、現在はサブネットを VLAN にバインドしないと機能しないというものです。これは、NetScaler がトラフィックを送信する VLAN を学習するためによく発生しますが、ARP テーブルの作成には時間がかかる場合があります。再起動またはファームウェアのアップグレード後、ARP テーブルの構築が再開されると、最初は学習されるため、デフォルトルートなど、希望とは異なるパスを使用している場合があります。SNIP を VLAN にバインドして、どのパスを取るかを指示するのが一番です。SNIP が VLAN にバインドされると、その SNIP のサブネット全体が VLAN にバインドされます。

  – すべての SNIP が VLAN にバインドされ（サブネットに複数の SNIP がある場合は、1 つだけをバインドする必要があります）、VLAN が 1 つのインターフェイスまたはチャネルにのみバインドされていることを確認します。また、すべてのサブネットに SNIP を設定するのが最善ですが、SNIP のない宛先サブネットには最も具体的なルートが使用されるため、必須ではありません。

- サブネットが使用する VLAN とインターフェイスを識別するには:

  1. [** システム] > [ネットワーク] **[VLAN] に移動します。

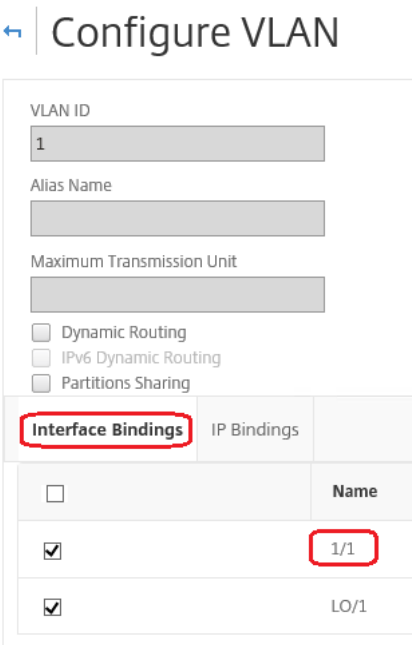
2. 次のステップで説明するように、正しい IP アドレスが見つかるまで、設定した各 VLAN を順番に編集します。
3. IP バインディングタブをクリックすると、どの IP、つまりどのサブネットがバインドされ、この VLAN を使用しているかを確認できます。
4. IP がバインドされている VLAN（その IP がデフォルトルートのサブネット内にある場合）を特定したら、インターフェイスバインディングをクリックします。この VLAN にバインドされた各インターフェイスまたはチャネルが使用されます。

例

デフォルトルートは`0.0.0.0 0.0.0.0 10.1.1.1`であると仮定します。

10.0.0.5 と 10.1.1.69 の 2 つの SNIP があるとします。10.1.1.69 はデフォルトルートのサブネットにあるので、探したいのはそのルートです。下のスクリーンショットでは、VLAN 1 を確認していますが、IP 10.1.1.69 がこの VLAN にバインドされていることがわかるので、正しい VLAN を確認していることがわかります。

次に、「インターフェースバインディング」をクリックします。VLAN インターフェースのバインディングを見ると、`1/1` このサブネットにはインターフェースが使用されており、したがってデフォルトルートにも使用されていることがわかります。



> 注意:
>
> VLAN に IP がバインドされていない場合、デフォルトでは VLAN 1 に送信されるため、その場合はどのインターフェイスが VLAN 1 にバインドされているかを確認してください。つまり、NetScaler は、IP を新しい VLAN にバインドしない限り、構成済みの VLAN を起動するトラフィックに使用しません。

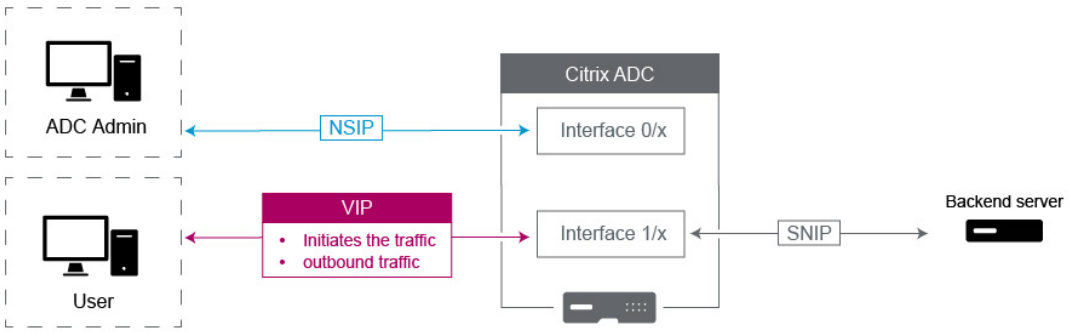
無償の **ARP**

GARP が機能しない場合は、VMAC を使用してください-デフォルトでは、NetScaler は GARP を使用して IP と MAC アドレスのバインディングを他のネットワークデバイスにアドバタイズします。これは通常問題なく機能しますが、NetScaler でより多くのサービスを作成するにつれて、HA ペアでフェイルオーバーするときに問題が発生する可能性があります。最も一般的な問題は、一部のネットワークデバイスが ARP テーブルを新しい MAC アドレスで更新していないために、フェイルオーバー先の NetScaler でサービスが停止したままになることです。ARP テーブルをチェックして、MAC アドレスが現在プライマリとなっている NetScaler 上のアドレスと一致するかどうかを確認することで、これを簡単に確認できます。この場合、一部のネットワークデバイスが、受け入れる GARP アドバタイズメントの数を制限している可能性が高くなります。この場合、すべてのアクティブなインターフェイスやチャネルに VMAC を設定する必要があります。NetScaler で大規模な構成を行うことが予想される場合は、初期展開時にすべてのインターフェイスとチャネルに対して VMAC を構成するのが最適な場合があります。

> 注: デフォルトルートで使用されるインターフェイスまたはチャネルに VMAC
>
> を設定することを忘れないでください。

NetScaler が所有する **IP** アドレス

このセクションでは、NetScaler が所有する IP アドレスを構成するためのベストプラクティスについて説明します。



- **NetScaler IP**（**NSIP**）：この IP は、HA またはクラスター環境の個々の NetScaler に固有の唯一の IP であるため、一般的に管理に使用されます。また、LDAP、RADIUS、およびユーザースクリプトモニターのトラフィック（LDAP モニターや StoreFront モニターなど）は NSIP から送信され、NSIP がバインドされている VLAN とインターフェイス（デフォルトのネイティブ VLAN 1）を介してルーティングされるという点にも注意してください。LDAP トラフィックと RADIUS トラフィックを SNIP から送信する必要がある場合は、バックエンドサーバー用の LB 仮想サーバーを作成します。

- サブネット **IP (SNIP)**: この IP アドレスは、バックエンドサーバーとの通信を開始するために使用され、常にトラフィックを開始します。とはいえ、次のような場合はトラフィックの宛先になる可能性があります。
  - NetScaler でレイヤー 3 ルーティングを行うときに、他のデバイスのゲートウェイアドレスとして使用できます。
  - 有効にすると、GUI、SSH、SNMP へのアクセスなどの管理サービスを受け入れることができます。
- 仮想 **IP (VIP)**: VIP は、アウトバウンドトラフィックの開始には決して使用されないという点で独特です。トラフィックのみを受信することを目的としています。トラフィックを受信すると、応答してトラフィックをクライアントに送り返します。つまり、VIP アドレスはアウトバウンドトラフィックを開始しません。

これは、たとえば LB 仮想サーバーで使用されるバックエンドサーバーとの通信のソースとして使用されないことも意味することにも注意してください。

SNIP アドレスから NetScaler ADC FreeBSD データトラフィックをソースするように構成する

August 16, 2023

一部の Citrix ADC データ機能は、Citrix ADC OS ではなく、基盤となる FreeBSD OS 上で実行されます。このため、これらの機能は、SNIP アドレスからではなく、NetScaler IP（NSIP）アドレスから送信されるトラフィックを送信します。設定にすべての管理トラフィックとデータトラフィックを分離する構成がある場合は、NSIP アドレスからデータトラフィックを調達することは望ましくありません。

以下の NetScaler データ機能は、基盤となる FreeBSD OS 上で動作し、NetScaler IP (NSIP) アドレスからトラフィックを送信します。

- 負荷分散スクリプタブルモニター
- GSLB 自動同期

この問題を解決するには、グローバル Layer-2 パラメータを使用できます。`useNetprofileBSDtraffic` このパラメーターを有効にすると、NetScaler の機能は、機能に関連付けられたネットプロファイル内の SNIP アドレスのいずれかからトラフィックを送信します。

はじめに

SNIP アドレスから NetScaler 機能関連のトラフィックを送信するように NetScaler アプライアンスを構成する前に、次の点に注意してください。

- 現在、グローバルレイヤー 2 パラメータ`useNetprofileBSDtraffic`は、負荷分散スクリプタブルモニターでのみサポートされています。

SNIP アドレスから GSLB 自動同期トラフィックを送信するように NetScaler アプライアンスを構成する場合、回避策として拡張 ACL ルールと RNAT ルールを使用できます。

- useNetprofileBSDtraffic 負荷分散スクリプタブルモニターのサポートは、関連サービスにバインドされたネットプロファイルにのみ適用されます。useNetprofileBSDtraffic このサポートは、関連するサービスグループにバインドされたネットプロファイルには適用されません。

  つまり、NetScaler ADC アプライアンスは、サービスグループにバインドされたネットプロファイルの SNIP アドレスを使用して、負荷分散のスクリプト可能なモニタートラフィックを調達しません。

- useNetprofileBSDtraffic このサポートは SSL サービスには適用されません。

  つまり、NetScaler ADC アプライアンスは、負荷分散スクリプト可能なモニタトラフィックをソースするために、SSL サービスにバインドされたネットプロファイルからの SNIP アドレスを使用しません。

SNIP アドレスからスクリプト可能なモニタートラフィックを送信するように **NetScaler ADC** アプライアンスを構成する

SNIP アドレスからスクリプタブルモニタートラフィックを送信するように NetScaler アプライアンスを構成するには、次のタスクで構成されます。

- グローバル Layer-2 パラメータを有効にします。useNetprofileBSDtraffic
- ネットプロファイルを作成し、少なくとも 1 つの SNIP アドレスをそれにバインドします。
- ネットプロファイルを、スクリプタブルモニターを使用している負荷分散サービスにバインドします。

レイヤ **2** パラメータを有効にするには、**CLI** を使用して **NetProfileBSDTraffic** を使用してください。

コマンドプロンプトで入力します。

- **set l2param** -**useNetprofileBSDtraffic** (**ENABLED** / **DISABLED**)
- **show l2param**

CLI を使用してネットプロファイルを作成し、**SNIP** アドレスをそれにバインドするには:

コマンドプロンプトで入力します。

- **add netProfile** <name> -**srcIP** <string>
- **show netProfile**

CLI を使用してネットプロファイルを負荷分散サービスにバインドするには:

コマンドプロンプトで入力します。

- **set service** <name> -**netProfile** <string>
- ショーサービス < 名前 >

設定例

次の構成例により、NetScaler アプライアンスは SNIP アドレスからスクリプタブルモニタートラフィックを送信できるようになります。ネットプロファイル NETPROFILE-1 には SNIP アドレス 198.51.100.20 がバインドされて設定されています。ユーザー/スクリプタブルモニター USER-MONITOR-1 が作成され、負荷分散サービス SERVICE-1 にバインドされます。NETPROFILE-1 is bound to SERVICE-1. NetScaler アプライアンスは、USER-MONITOR-1 のすべてのスクリプト可能なモニターパケットを SNIP アドレス 198.51.100.20 から送信します。

```
set l2param -useNetprofileBSDtraffic ENABLED

set netprofile NETPROFILE-1 -srcip 198.51.100.20

add lb monitor USER-MONITOR-1 USER -scriptName nsftp.pl -scriptArgs "
    file=Index.png;user=nsroot;password=nsroot" -dispatcherIP 127.0.0.1
    -dispatcherPort 3013 -destIP 203.0.113.90 -destPort 21

bind service SERVICE-1 -monitorName USER-MONITOR-1

set service SERVICE-1 -netProfile NETPROFILE-1

<!--NeedCopy-->
```

SNIP アドレスから **GSLB** 自動同期トラフィックを送信するように **NetScaler** アプライアンスを構成する

SNIP アドレスから GSLB 自動同期トラフィックを送信するように NetScaler アプライアンスを構成するには、次の回避タスクが必要です。

- 拡張 **ACL** ルールを作成します。拡張 ACL ルールは GSLB 自動同期パケットを識別します。この識別は、送信元 IP アドレスと宛先 IP アドレスに基づいています。
- **ACL** を適用します。ACL を適用すると、新しく作成された ACL ルールがアクティブになります。
- **ACL** ベースの **RNAT** ルールを作成します。RNAT ルールは、これらのパケットの送信元 IP アドレスを NSIP アドレスから SNIP アドレスに変更します。

> 注:
>
> 高可用性またはクラスタ設定では、セットアップのすべての NSIP アドレスに ACL および RNAT ルールを追加する必要があります。

CLI を使用して拡張 **ACL** を作成するには**:**

コマンドプロンプトで入力します。

- **add acl** <aclname> **ALLOW** -**srcIP** = <NSIP address> -**destIP** = <destination IP address of the packets>

- **show acl** <aclName>

CLI を使用して拡張 **ACL** を適用するには:

コマンドプロンプトで入力します。

- **apply acls**

CLI を使用して **ACL** ベースの **RNAT** ルールを作成するには**:**

コマンドプロンプトで入力します。

- **add rnat** <name> <aclname>
- **bind rnat** <name> -**natIP** <SNIP address - source IP address for the packets>
- **show rnat** <name>

設定例

次の構成例により、NetScaler アプライアンスは SNIP アドレスから GSLB 自動同期トラフィックを送信できるようになります。ACL-2 は、NSIP アドレス 192.0.1.20 から送信され、GSLB サイトの IP アドレス 203.0.113.20 を宛先とする GSLB 自動同期パケットを識別します。RNAT-2 は、これらの識別されたパケットの送信元 IP アドレスを SNIP アドレス 198.51.100.20 に変更します。

```
add acl ACL-2 ALLOW -srcIP = 192.0.1.20  -destIP = 203.0.113.20

apply acls

add rnat RNAT-2 ACL-2

bind rnat RNAT-2 -natIP 198.51.100.20
<!--NeedCopy-->
```

オブザーバビリティ

March 20, 2024

現代のアプリケーションはますます複雑化しているため、アプリケーションの監視とトラブルシューティングは IT チームにとってますます困難になっています。また、インフラストラクチャとアプリケーションの動作を可視化することは、ソフトウェア開発チームにとってより重要です。オブザーバビリティは、インフラストラクチャ全体に関するより深い洞察を提供することで、このギャップを埋めます。オブザーバビリティツールは、さまざまな IT インフラストラクチャコンポーネントと統合することでアプリケーションまたはシステムのパフォーマンステレメトリを継続的に収集し、IT インフラストラクチャを全体的に可視化します。

オブザーバビリティの利点は以下のように要約できます。

- 迅速なトラブルシューティング: オブザーバビリティツールから得られる詳細なデータインサイトは、システムの問題を迅速に診断してトラブルシューティングするのに役立ちます。
- アプリケーションパフォーマンスの向上: 主要な指標を監視し、問題を特定することで、開発者はデータ主導の意思決定を行い、アプリケーションのパフォーマンスを向上させることができます。
- 信頼性の向上とユーザーエクスペリエンスの向上: オブザーバビリティデータにより、開発者はユーザー体験を妨げる可能性のあるシステム障害をプロアクティブに解決できます。

オブザーバビリティとは

オブザーバビリティとは、システムが生成するデータ (ログ、メトリクス、トレース、イベントなど) を分析することで、システムの内部状態を理解する能力です。オブザーバビリティにより、障害発生時のシステムの動作に関する特定の質問に理解し、回答することができます。システムを深く理解していれば、未知なことに備えることができます。たとえば、速度が遅いか速いか、何が壊れているか、システムパフォーマンスを向上させるために何をすべきかを追跡できます。

メトリクス、ログ、トレースはオブザーバビリティの重要な柱です。

- 指標: 指標は、一定期間にわたって測定されたデータを数値で表したものです。メトリックスデータは、システムの状態を経時的に追跡するのに役立ちます。これらの数値測定には、CPU 使用率、メモリ使用量、エラー率が含まれます。
- ログ: ログは、特定の時点で発生したイベントを説明するメッセージまたはレコードです。通常、これらのメッセージまたはレコードはアプリケーションまたはシステムによって生成されます。
- トレース: トレースは、リクエストが分散システムのさまざまな部分を通過する過程を表します。トレースは、リクエストがどのように処理され、完了するまでにかかる時間を記録します。このデータは、ボトルネックやその他の待ち時間の問題を特定するのに役立ちます。

モニタリングとオブザーバビリティ

モニタリングは、何か問題が発生したときに通知するためのツールまたはソリューションのセットです。オブザーバビリティがあれば、何が起こっているのかを特定し、問題の根本をすばやく特定して原因を突き止めることができます。監視によって生成された事実とデータを統合して、システムのパフォーマンスと状態を詳細に把握できるようにします。オブザーバビリティを使用すると、データを自動的に分析し、迅速かつ正確な入力に基づいてユーザーエクスペリエンスを向上させることができます。

NetScaler によるオブザーバビリティ

NetScaler をアプリケーション展開のプロキシとして展開する場合、NetScaler は各ユーザーの要求または応答を検査して、グローバルルーティングとローカルデータセンタールーティングを確認します。NetScaler が提供する何千ものログとカウンターを使用して、HTTP、TCP、SSL、および DNS パケットに関する詳細な情報を得ることがで

きます。NetScaler の豊富なデータやインサイトを活用して、問題のトラブルシューティングや特定を行うことができます。NetScaler から任意のオブザーバビリティエンドポイントにデータをエクスポートして視覚化を行い、リアルタイムで詳細なアプリケーションインサイトを得ることができます。

NetScaler は、Prometheus、Splunk、ElasticSerach、Kafka などの一般的なオブザーバビリティツールとの統合が可能です。
NetScaler は Prometheus と直接統合できます。直接統合すれば、データをエクスポートしてニーズに合わせてカスタマイズしたダッシュボードを構築するために、追加のエージェントやノードをデプロイする必要はありません。Prometheus は、すべてのエンティティから数値メトリックを収集する時系列データモニタリングに焦点を当てています。

NetScaler Console には、SSL インサイト、Web トランザクションインサイト、API インサイトなど、いくつかのオブザーバビリティ機能が組み込まれています。

NetScaler は、オブザーバビリティの一部として次の 3 種類のインサイトを提供できます:

- アプリケーションと API に関するインサイト: アプリケーションヘルスインサイトは、どのアプリケーション Web サイトの待ち時間が長いか、エラー数が多いか、パフォーマンスが標準以下かをトラブルシューティングするのに役立ちます。また、エラー、トラフィック、遅延、飽和度の指標の監視も含まれています。これらの信号を総称して、アプリケーションの状態を監視するためのゴールデンシグナルと呼ばれます。
- アプリケーションと API のセキュリティに関するインサイト: アプリケーションセキュリティのインサイトには、トラフィック全体と比較して検出または防止された WAF 違反、WAF または BOT 違反の影響を受けた上位アプリケーション、CVE、良いボットと悪いボットなどの BOT 分類が含まれ、攻撃者に関する情報も得られます。
- ネットワークインフラストラクチャに関するインサイト: NetScaler インフラストラクチャのインサイトには、CPU 使用率、メモリとディスクの使用量、ネットワークインターフェーステレメトリなどの NetScaler に関する情報が含まれます。また、SSL、GSLB、マルチパス TCP (MPTCP) などの特定の機能レベルのインサイトや、証明書の有効期限の詳細、使用されているプロトコル、暗号強度などの SSL TLS モニタリングに関するインサイトも得られます。

NetScaler から Prometheus にメトリックを直接エクスポートする方法について詳しくは、「Prometheus による NetScaler、アプリケーション、およびアプリケーションセキュリティの監視」を参照してください。

Prometheus による **NetScaler**、アプリケーション、およびアプリケーションセキュリティの監視

August 29, 2023

指標は、特定の期間に測定されたデータを数値で表したものです。メトリックデータは、
システムの状態を経時的に追跡するのに役立ちます。Prometheus は、メトリクスデータを収集し、データが記録された時点のタイムスタンプを付けてそのデータを保存するオープンソースの監視ツールです。

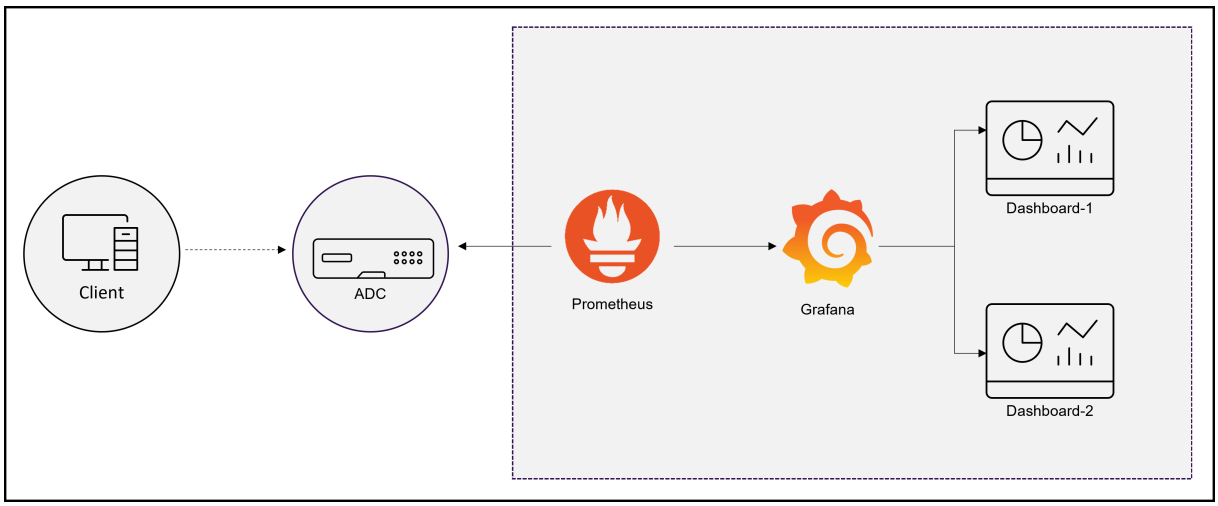
メトリクスを監視および分析することで、アプリケーションの状態を追跡し、異常を検出し、アラートを作成し、必要な是正措置を講じて、ソフトウェアを確実に配信できます。

NetScaler は、Prometheus へのメトリクスの直接エクスポートをサポートするようになりました。NetScaler ADC が提供する豊富なメトリックセットを使用して、NetScaler の状態とアプリケーションの状態を監視できます。たとえば、CPU とメモリの使用状況に関するメトリックを収集して、NetScaler の状態を知ることができます。同様に、1 秒あたりに受信した HTTP リクエストの数やアクティブなクライアントの数などの指標を使用して、アプリケーションの状態を監視できます。

NetScaler から **Prometheus** へのメトリクスのエクスポート

NetScaler は、Prometheus のプルモードとプッシュモードをサポートしています。プルモードでは、Prometheus が定期的にクエリを実行し、エクスポーターリソースを挟まずにメトリクスデータを直接取得する時系列プロファイルを設定する必要があります。プルモードでは、スーパーユーザー権限を持たないユーザーが Prometheus にメトリックをエクスポートするための読み取り専用アクセスを有効にできます。Grafana を使用すると、Prometheus にエクスポートされた NetScaler メトリクスを視覚化して、解釈や理解が容易になります。

次の図は、Prometheus と Grafana と NetScaler の統合を示しています。



NetScaler から **Prometheus** へのメトリクスのエクスポートと **Grafana** を使用した視覚化を設定します

NetScaler から Prometheus へのメトリックのエクスポートを設定し、Grafana を使用して視覚化するには、次の手順を実行する必要があります。

1. メトリクスを Prometheus にエクスポートするための時系列分析プロファイルを使用して NetScaler を構成します。
2. Prometheus をインストールし、NetScaler 固有のパラメーターを使用して構成します。
3. Grafana のデータソースとして Prometheus を追加します。

4. Grafana でビジュアライゼーションを作成

Prometheus プルモードをサポートするように **NetScaler** で時系列分析プロファイルを構成する

NetScaler CLI を使用してプルモードを構成するには、次の手順を実行します。

1. タイプが時系列である分析プロファイルを作成します。必要な NetScaler メトリックを含むスキーマファイルを指定します。

```
add analytics profile <timeseries_profile_name> -type timeseries -
    schemaFile <name_of_schema_file>
-outputMode Prometheus -serveMode PULL -metrics ENABLED
```

このコマンドでは、

- timeseries_profile_name: 時系列プロファイル名を指定します。
- schemaFile: NetScaler カウンタでスキーマファイルの名前を指定します。デフォルトでは、カウンターのリストを含むスキーマファイル/var/metrics_conf/schema.jsonが設定されます。サポートされているすべてのカウンタを含む参照スキーマファイルreference_schema.jsonもパス/var/metrics_conf/の下にあります。このスキーマファイルは、カウンターのカスタムリストを作成するための参照として使用できます。スキーマファイルを指定すると、スキーマファイル/var/metrics_conf/のパスが自動的に追加されるため、指定する必要があるのはスキーマファイル名だけです。たとえば、/var/metrics_conf/にカウンタのカスタムリストを含むスキーマファイルschema1.jsonを作成した場合、ファイル名だけをschema1.jsonとして指定する必要があります。
- outputMode: 出力モードを Prometheus に設定します。
- serveMode: Prometheus のプルモードを指定します。
- metrics: NetScaler からのメトリクスの収集を有効にします。

> 注:
>
> add コマンドを使用して、必要なすべてのパラメーターを含む分析プロファイルを設定できます。プロファイルの作成後に変更を加える必要がある場合は、set コマンドを使用して、メトリックの無効化やサーバーモードの変更などの適切なアクションを実行できます。非スーパーユーザーには、読み取り専用の Prometheus アクセスを設定できます。詳細については、「スーパーユーザー以外の読み取り専用の Prometheus アクセスを設定する」を参照してください。

NetScaler からメトリックをエクスポートするための **Prometheus** のインストールと設定

Prometheus は、DockerHub や Quay などのリポジトリや、Prometheus の公式リポジトリからダウンロードできます。

Prometheus を Docker コンテナーとして実行するには、以下のコマンドを使用します。

```
1     docker run -dp 39090:9090 -v /tmp/prometheus.yml:/etc/prometheus/
         prometheus.yml --name native_prom prom/prometheus:latest  > **
         注： ** > > ここでは、`/tmp/prometheus.yml`が`prometheus.yml`ファ
         イルへのパスとして使用されます。その代わりに、仮想マシンのパスを
         指定できます。
```

`prometheus.yml` NetScaler のパラメータを使用して編集する必要があります。

NetScaler からメトリックをエクスポートするには、**Prometheus** YAML スクレイプ構成セクションで次の NetScaler 固有のパラメーターを指定する必要があります。スクレイプ構成セクションでは、ターゲットとそれらをスクレイピングする方法を説明する構成パラメーターのセットを指定します。

- `metrics_path` : NetScaler の HTTP リソースパス (/nitro/v1/config/systemfile) を指定してメトリックを取得します。
- `username`: NetScaler ユーザー名を指定します。
- `password`: NetScaler のパスワードを指定します。
- `targets`: メトリクス、メトリックのエクスポート元となる NetScaler の IP アドレス
、および公開するポートを指定します。
- `filename`: `metrics_prom_<timeseries_profile_name>.log`ファイルで`timeseries_profi`
の代わりに、設定した時系列プロファイルの名前を指定します。
- `filelocation`: ファイルの場所を`/var/nslog`として指定します。

以下は、Prometheus YAML のスクラップ構成セクションで、NetScaler IP アドレスを Prometheus のターゲットとして追加してメトリックをエクスポートするためのものです。ここでは、HTTP がスキームとして使用されています。HTTP または HTTPS のいずれかを使用できます。

```
1  scrape_configs:
2    - job_name: 'vpx2_metrics_direct'
3      metrics_path: /nitro/v1/config/systemfile
4      params:
5        args: ['filename:metrics_prom_ns_analytics_time_series_profile.
            log,filelocation:/var/nslog']
6        format: ['prometheus']
7      basic_auth:
8       username: 'prom_user'
9       password: 'user_password'
10     scheme: http
11     scrape_interval: 30s
12     static_configs:
13      - targets: ['10.102.34.231:80']
14 <!--NeedCopy-->
```


Prometheus を **Grafana** データソースとして追加

Grafana ダッシュボードを使用してメトリクスを視覚化する必要がある場合は、Grafana のデータソースとして Prometheus を追加する必要があります。詳細については、「Grafana のデータソースとして Prometheus を追加する」を参照してください。

Grafana でメトリクスのビジュアライゼーションを作成

Grafana ダッシュボードを作成し、主要な指標と適切なビジュアライゼーションタイプを選択できます。

以下の手順は、Grafana パネルに指標を追加し、サンプルビジュアライゼーションダッシュボードを作成する方法を示しています。

1. パネルタイトルを指定します。
2. 「クエリ」タブで、クエリ A に必要なメトリックを指定します。
3. 「設定」タブで、「ビジュアライゼーション」タイプを選択します。

Grafana では、データとその表現を変更できます。詳細については、Grafana のドキュメントを参照してください。

以下は、いくつかの NetScaler ADC メトリックを含む Grafana ダッシュボードのサンプルです。

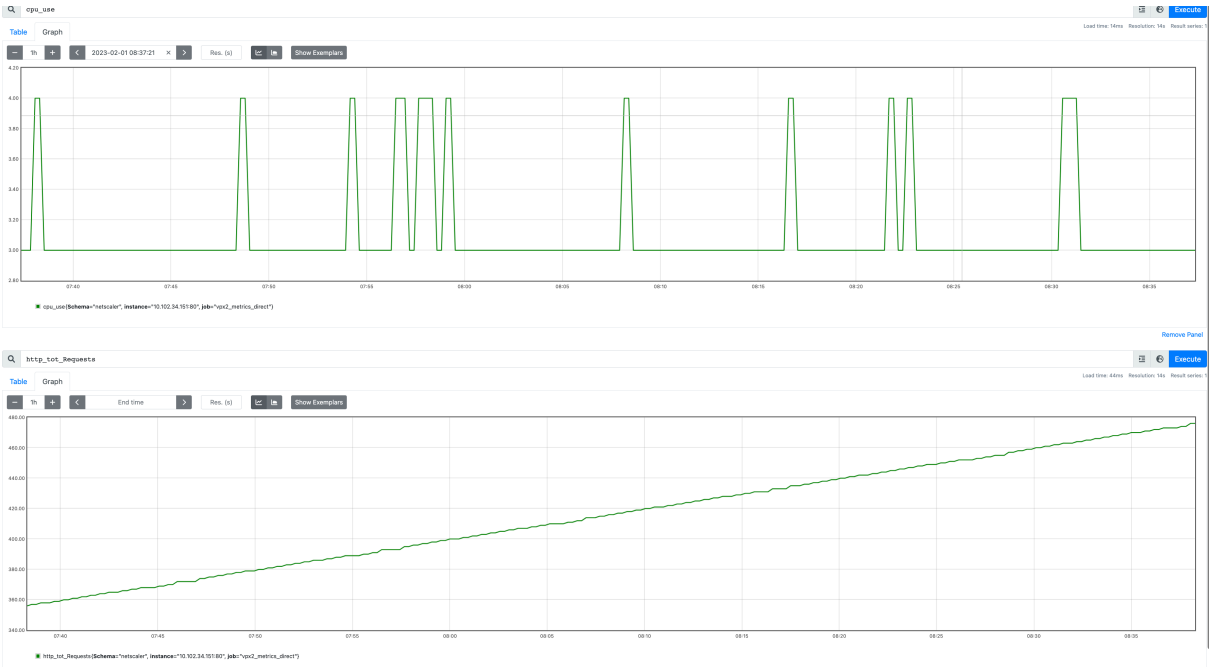
このダッシュボードには、次のようなさまざまな NetScaler メトリックのグラフが表示されます。

- `vsvr_tot_Hits`: 仮想サーバーが受信した要求の数を示します。
- `cc_cpu_use`: CPU 使用率を表示します。
- `http_tot_Requests`: 受信した HTTP リクエストを表示します。
- `serv_tot_serviced`: 処理中のリクエストを表示します。
- `mem_cur_used_size`: NetScaler アプライアンスの現在使用されているメモリが表示されます。

Prometheus グラフのサンプル

Prometheus エクスプレッションブラウザを使用すると、Prometheus サーバーによって収集された時系列メトリックを表示できます。ブラウザで`prometheu-server-ip-address/graph` をポイントすると、エクスプレッションブラウザにアクセスできます。式を入力すると、その結果を表またはグラフとして時系列で表示できます。「Expression」フィールドに指標名を入力して、表示する正確な指標を指定します。さまざまなパネルを使用して複数のカウンターを指定できます。

次の図は、2 つの NetScaler メトリックとに関する Prometheus グラフを示しています。`cpu_use` `http_tot_requests`



Grafana ダッシュボードのサンプル

サンプルダッシュボードは NetScaler のダウンロードページからダウンロードできます。

以下は、NetScaler の状態、仮想サーバーのステータス、アプリケーションの状態（HTTP および TCP アプリ）、アプリケーションセキュリティ（SSL アプリ）など、インフラストラクチャ全体のさまざまなメトリックを 1 か所に表示するオプションを備えた Grafana ダッシュボードのサンプルです。

ダッシュボードの対応するセクションを展開すると、HTTP アプリ、SSL アプリ、TCP アプリ、仮想サーバー統計(vStats)、NetScaler 統計情報などの各セクションの詳細な視覚化を表示できます。

次の図は、NetScaler の統計情報を拡張した Grafana ダッシュボードのサンプルを示しています。

追加情報

エクスポートに必要な **NetScaler** カウンタを含むスキーマ

メトリクスコレクターは、設定されたスキーマファイルに存在するカウンターをエクスポートします。`/var/metrics_conf/schema.json` このファイルは、分析プロファイルで設定されたデフォルトのスキーマファイルです。

スキーマファイルは、エンティティタイプと関連するカウンターのリストです。スキーマでは、netscalerすべてのグローバルまたはシステムレベルのカウンタがエンティティタイプ別にグループ化されます。グローバルカウンタには、CPU 使用率 (cpu_use)、管理 CPU 使用率 (mgmt_cpu_use)、受信した HTTP リクエストの総数 (http_tot_Requests) があります。サービスグループ固有のカウンタ、lbvserver、csvserverなどは、それぞれのエンティティタイプの下に一覧表示されます。

以下は、認証仮想サーバー (vserver_authn) エンティティのschema.jsonファイル内のカウンターのサンプルです。

```
"vserver_authn":
  [
       {
 "name":"si_tot_Requests","rate":"True" }
 ,
       {
 "name":"si_tot_Responses","rate":"True" }
 ,
       {
 "name":"si_tot_RequestBytes","rate":"True" }
 ,
       {
 "name":"si_cur_state","rate":"False" }
 ,
       {
 "name":"si_tot_ResponseBytes","rate":"True" }
 ,
       {
 "name":"si_peer_port","rate":"True" }
 ,
       {
 "name":"vsvr_Protocol","rate":"False" }

  ]
```

次の表では、このサンプルに記載されているカウンタについて説明します。

カウンター名	説明
si_tot_Requests	このサービスまたは仮想サーバーで受信したリクエストの総数。
si_tot_Responses	このサービスまたは仮想サーバーで受信した応答の総数。
si_tot_RequestBytes	このサービスまたは仮想サーバーで受信したリクエストバイトの総数。
si_cur_state	仮想サーバーの現在の状態。
si_tot_ResponseBytes	このサービスまたは仮想サーバーで受信した応答バイトの総数。
si_peer_port	サービスが実行されているポート。

カウンター名	説明
vsvr_Protocol	仮想サーバーに関連するプロトコル。

rateフィールドは、カウンターのレート値をエクスポートする必要がある場合、Trueとして設定できます。たとえば、rateをsi_tot_RequestsのTrueに設定すると、si_tot_Requestsのレートがエクスポートされます。

以下は、netscaler エンティティのカウンターのサンプルです。

```
 "netscaler":
   [
        {
"name":"cpu_use","rate":"False" }
,
        {
"name":"mgmt_cpu_use","rate":"False" }
,
        {
"name":"tcp_tot_rxpkts","rate":"True" }
,
        {
"name":"tcp_tot_rxbytes","rate":"True" }
,
        {
"name":"tcp_tot_txpkts","rate":"True" }
,
        {
"name":"tcp_tot_txbytes","rate":"True" }
,
        {
"name":"tcp_cur_ClientConnEst","rate":"False" }
,
        {
"name":"tcp_cur_ServerConnEst","rate":"False" }
,
        {
"name":"tcp_cur_ClientConn","rate":"False" }
,
        {
"name":"tcp_cur_ClientConnClosing","rate":"False" }
,
        {
"name":"tcp_tot_ClientOpen","rate":"True" }
,
        {
"name":"tcp_cur_ServerConn","rate":"False" }
,
        {
"name":"tcp_cur_ServerConnClosing","rate":"False" }
,
```



```
        {
"name":"http_tot_Requests","rate":"True" }
,
        {
"name":"http_tot_Responses","rate":"True" }
,
        {
"name":"http_tot_Gets","rate":"True" }
,
        {
"name":"http_tot_Posts","rate":"True" }
,
        {
"name":"http_tot_Others","rate":"True" }
,
   ]
```

次の表では、このサンプルに記載されているカウンタについて説明します。

カウンター名	説明
cpu_use	CPU 使用率を追跡します (CPU 使用率 * 10)。
tcp_tot_rxpkts	TCP パケットを受信しました。
tcp_tot_rxbytes	受信した TCP データのバイト数。
tcp_tot_txpkts	TCP パケットが送信されました。
tcp_tot_txbytes	送信された TCP データのバイト数。
tcp_cur_ClientConnEst	現在のクライアント接続が「確立済み」状態です。これは、NetScaler アプライアンスとクライアント間でデータ転送が可能であることを示しています。
tcp_cur_ServerConnEst	現在のサーバー接続が「確立済み」状態です。これは、NetScaler アプライアンスとサーバー間でデータ転送が可能であることを示しています。
tcp_cur_ClientConn	クライアント接続 ([開始]、[確立]、[終了] 状態の接続を含む)。サーバー接続 (「開始」、「確立」、および「終了」状態の接続を含む)。
tcp_cur_ClientConnClosing	クライアント接続が閉じている状態です。これは、接続終了プロセスが開始されたがまだ完了していないことを示します。
tcp_cur_ServerConn	サーバー接続 (「開始」、「確立」、および「終了」状態の接続を含む)。
tcp_cur_ServerConnClosing	サーバー接続が「クローズ」状態になっています。これは、接続終了プロセスが開始されたがまだ完了していないことを示します。

カウンター名	説明
http_tot_Requests	このカウンタは、GET メソッドを使用して受信した HTTP リクエストを追跡します。
http_tot_Responses	このカウンタは、POST メソッドを使用して受信した HTTP リクエストを追跡します。
http_tot_Gets	このカウンタは、GET メソッドを使用して受信した HTTP リクエストを追跡します。
http_tot_Posts	このカウンタは、受信した HTTP リクエストを追跡します。
http_tot_Others	このカウンタは、GET と POST 以外のメソッドを使用して受信した HTTP リクエストを追跡します。

以下は、vserver_ssl エンティティのカウンターのサンプルです。

```
"vserver_ssl":
[
    {
 "name":"ssl_ctx_tot_session_hits","rate":"True" }
 ,
    {
 "name":"ssl_ctx_tot_session_new","rate":"True" }
 ,
    {
 "name":"ssl_ctx_tot_enc_bytes","rate":"True" }
 ,
    {
 "name":"ssl_ctx_tot_dec_bytes","rate":"True" }
 ,
]
```

次の表では、このサンプルに記載されている SSL カウンタについて説明します。

カウンター名	説明
ssl_ctx_tot_session_hits	このカウンタはセッションヒット数を追跡します。
ssl_ctx_tot_session_new	このカウンタは、作成された新しいセッションの数を追跡します。
ssl_ctx_tot_enc_bytes	このカウンタは、SSL 仮想サーバーごとの暗号化されたバイト数を追跡します。
ssl_ctx_tot_dec_bytes	このカウンタは、SSL 仮想サーバーごとに復号化されたバイト数を追跡します。

非スーパーユーザーには読み取り専用の **Prometheus** アクセスを設定する

スーパーユーザー以外のユーザーに読み取り専用の Prometheus アクセスを設定するには、次の手順を実行します。

1. NetScaler アプライアンスに新しいユーザーを追加します。

```
add system user <ns_user_name> <ns_user's_password> -externalAuth
    enabled -promptString user-%u-at-%T logging enaBLED
```

例:

```
add system user nspaul nspaul -externalAuth enabled -promptString
    user-%u-at-%T logging enaBLED
```

2. 読み取り専用ユーザー用のコマンドポリシーを作成します。このコマンドポリシーでは、/var/nslog/ directoryにあるすべてのファイルからの読み取り専用アクセスを許可します。

```
add system cmdPolicy read-only-prometheus ALLOW \"\(^man.\*)|\(^
    show\s+\(?!system)\(?!configstatus)\(?!ns ns\.conf)\(?!ns
    savedconfig)\(?!ns runningConfig)\(?!gslb runningConfig)\(?!
    audit messages)\(?!techsupport).\*)|\(^stat.\*)|\(show system
    file .\* -filelocation \"/var/nslog\")\"
```

3. メトリクスを特定のファイルにのみ書き込む場合は、その特定のファイルのみを取得できるようにユーザーアクセスを制限することもできます。

```
add system cmdPolicy read-only-prometheus ALLOW \"\(^man.\*)|\(^
    show\s+\(!system)\(!configstatus)\(!ns ns\.conf)\(!ns
    savedconfig)
\(!ns runningConfig)\(!gslb   runningConfig)\(!audit messages)\(!
    techsupport).\*)|\(^stat.\*)
|\(show system file metrics\_prom\_<name\_of\_timeseries\_profile
    >.log -filelocation \"/var/nslog\")\"
```

> 注:
>
> show system fileコマンドで、name_of_ timeseries_profileの代わりに設定した時系列プロファイルの名前を指定します。

4. ユーザーをコマンドポリシーでバインドします。

```
bind system user <userName> \(\(<policyName> <priority>) | -
    partitionName <string>)
```

例えば:

```
bind system user user1 read-only-prometheus 0
```

ユーザーをバインド解除してコマンドポリシーから削除するには、次のコマンドを使用します。

1. 設定したユーザーをシステムコマンドポリシーからバインド解除します。

```
unbind system user <userName> \(<policyName> | -partitionName <
    string>
```

例:

```
unbind system user user1 read-only-prometheus
```

2. NetScaler からコマンドポリシーを削除します。

```
rm system cmdPolicy read-only-prometheus
```

複数の時系列プロファイルのカウンターの購読

NetScaler は複数の時系列プロファイルの作成をサポートし、プロファイルごとに異なるカウンターセットを指定できるようになりました。また、要件に応じてカウンターのみをエクスポートできます。

複数の時系列プロファイルを構成するには、必要なカウンタを固有の名前と拡張子.jsonで含む複数のschema.jsonファイルを作成する必要があります。参照スキーマファイルreference_schema.jsonはパス/var/metrics_conf/の下にあります。要件に応じて参照スキーマを変更し、それに応じて使用できます。

2 つの新しい時系列プロファイルの構成は次のとおりです。

```
add analytics profile ns_analytics_timeseries_profile_1   -type
    timeseries -schemaFile schema1.json

set analytics profile  ns_analytics_timeseries_profile_1 -outputMode
    prometheus -serveMode PULL -metrics ENABLED

add analytics profile ns_analytics_timeseries_profile_2   -type
    timeseries -schemaFile schema2.json

set analytics profile  ns_analytics_timeseries_profile_2 -outputMode
    prometheus -serveMode PULL -metrics ENABLED
```

この例では、schema1.jsonと schema2.json でカウンターのセットが異なります。

Prometheus 構成

prometheus.yml サンプルファイルの設定は次のとおりです。

```
scrape_configs:
  - job_name: 'vpx2_metrics_direct'
    metrics_path: /nitro/v1/config/systemfile
    params:
      args: ['filename:metrics_prom_ns_analytics_time_series_profile.
          log,filelocation:/var/nslog']
```



```
      format: ['prometheus']
    basic_auth:
     username: 'prom_user'
     password: 'user_password'
    scheme: https
    scrape_interval: 30s
    static_configs:
     - targets: ['<ADC1-ip>:<port>', '<ADC2-ip>:<port>']
<!--NeedCopy-->
```

監査ログとイベントを **NetScaler** から **Splunk** に直接エクスポートする

October 25, 2023

監査ログを使用すると、NetScaler のさまざまなモジュールによって収集された NetScaler の状態とステータス情報を記録できます。ログを確認することで、問題やエラーをトラブルシューティングして修正できます。

NetScaler から Splunk などの業界標準のログアグリゲータープラットフォームに監査ログとイベントをエクスポートして、有意義な洞察を得ることができるようになりました。

NetScaler から Splunk に監査ログをエクスポートする方法は複数あります。Splunk は、Syslog サーバーまたは HTTP サーバーとして構成できます。このトピックでは、Splunk HTTP イベントコレクターを使用して Splunk を HTTP サーバーとして構成する方法について説明します。HTTP イベントコレクターを使用すると、監査ログを HTTP（または HTTPS）経由で NetScaler から Splunk プラットフォームに直接送信できます。

> 注:
>
> NetScaler から Splunk への非パケットエンジン（PE）ログのエクスポートは、現在サポートされていません。

NetScaler から **Splunk** への監査ログのエクスポートの設定

監査ログのエクスポートを設定するには、次の手順を実行する必要があります。

1. Splunk で HTTP イベントコレクターを設定します。
2. NetScaler でコレクターサービスと時系列分析プロファイルを作成します。

Splunk で **HTTP** イベントコレクターを設定

HTTP イベントコレクターを設定することで、監査ログを Splunk に転送できます。

HTTP イベントコレクターの設定方法については、Splunk のマニュアルを参照してください 。

HTTP イベントコレクターを設定したら、認証トークンをコピーして参照用に保存します。NetScaler で分析プロファイルを構成する際には、このトークンを指定する必要があります。

NetScaler での時系列分析プロファイルの設定

NetScaler 監査ログを Splunk にエクスポートするには、次の手順を実行してください。

1. Splunk 用のコレクターサービスを作成します。

```
add service <collector> <splunk-server-ip-address> <protocol> <
    port>
```

例:

```
add service splunk_service 10.102.34.155 HTTP 8088
```

この構成では、

- ip-address: Splunk サーバの IP アドレスを指定します。
- collector-name: コレクターを指定します。
- protocol: プロトコルを HTTP または HTTPS として指定します。
- port: ポート番号を指定します。

2. 時系列分析プロファイルを作成します。

```
    add analytics profile <profile-name> -type time series -
        auditlog enabled -collectors <collector-name>  -
        analyticsAuthToken <"auth-tocken">
    -analyticsEndpointContentType <"Application/json"> -
        analyticsEndpointMetadata <"meta-data-for-endpoint:"> -
        analyticsEndpointUrl <"endpoint-url">
```

例:

```
    add analytics profile audit_profile -type timeseries -auditlog
         enabled -collectors splunk -analyticsAuthToken "
        1234-5678-12345" -analyticsEndpointContentType "Application
        /json" -analyticsEndpointMetadata "Event:" -
        analyticsEndpointUrl "/services/collector/event"
```

この構成では、

- auditlog: enabled 監査ログを有効にするには値を指定します。
- analyticsAuthToken: Splunk にログを送信する際に認証ヘッダーに含める認証トークンを指定します。このトークンは、HTTP イベントコレクターの設定時に Splunk サーバーで作成される認証トークンです。
- analyticsEndpointContentType: ログの形式を指定します。
- analyticsEndpointMetadata: エンドポイント固有のメタデータを指定します。
- analyticsEndpointUrl: ログをエクスポートするエンドポイントの場所を指定します。

> 注:
>
> 時系列分析プロファイルのパラメーターは、`set analytics profile` コマンドを使用して変更できます。

3. show analytics profile コマンドを使用して、アナリティクスプロファイルの設定を確認します。

```
# show analytics profile audit_profile

1)    Name: audit_profile
      Collector: splunk
      Profile-type: timeseries
            Output Mode: avro
            Metrics: DISABLED
              Schema File: schema.json
              Metrics Export Frequency: 30
            Events: DISABLED
            Auditlog: ENABLED
            Serve mode: Push
       Authentication Token: <auth-tocken>
       Endpoint URL: /services/collector/event
       Endpoint Content-type: Application/json
       Endpoint Metadata: Event:
       Reference Count: 0
```

設定が完了すると、監査ログは HTTP ペイロードとして Splunk に送信され、Splunk アプリケーションのユーザーインターフェイスで表示できます。

優先負荷分散

August 15, 2023

優先負荷分散機能を使用すると、優先負荷分散仮想サーバーにバインドされている各サービスまたはサービスグループに優先順位番号を割り当てることができます。番号が最も小さいサービスまたはサービスグループの優先順位が最も高くなります。アプリケーショントラフィックは、このサービスまたはサービスグループが稼働している限り、このサービスまたはサービスグループにのみ分散されます。次のプライオリティ番号が割り当てられたサービスまたはサービスグループは、プライオリティが最も高いサービスまたはメンバーがすべてダウンしている場合にのみ動作可能になります。ただし、優先順位が最も高いサービスまたはサービスグループのメンバーが再び使用可能になると、トラフィックはそのサービスまたはサービスグループにリダイレクトされます。

たとえば、優先負荷分散仮想サーバーにバインドされているサービスグループ SVG1、SVG2、SVG3 を考えてみましょう。優先グループの最大数は 3 に設定されています。次のように各グループに優先度を割り当てます。

- SVG1 —優先度 1
- SVG2 —優先度 2

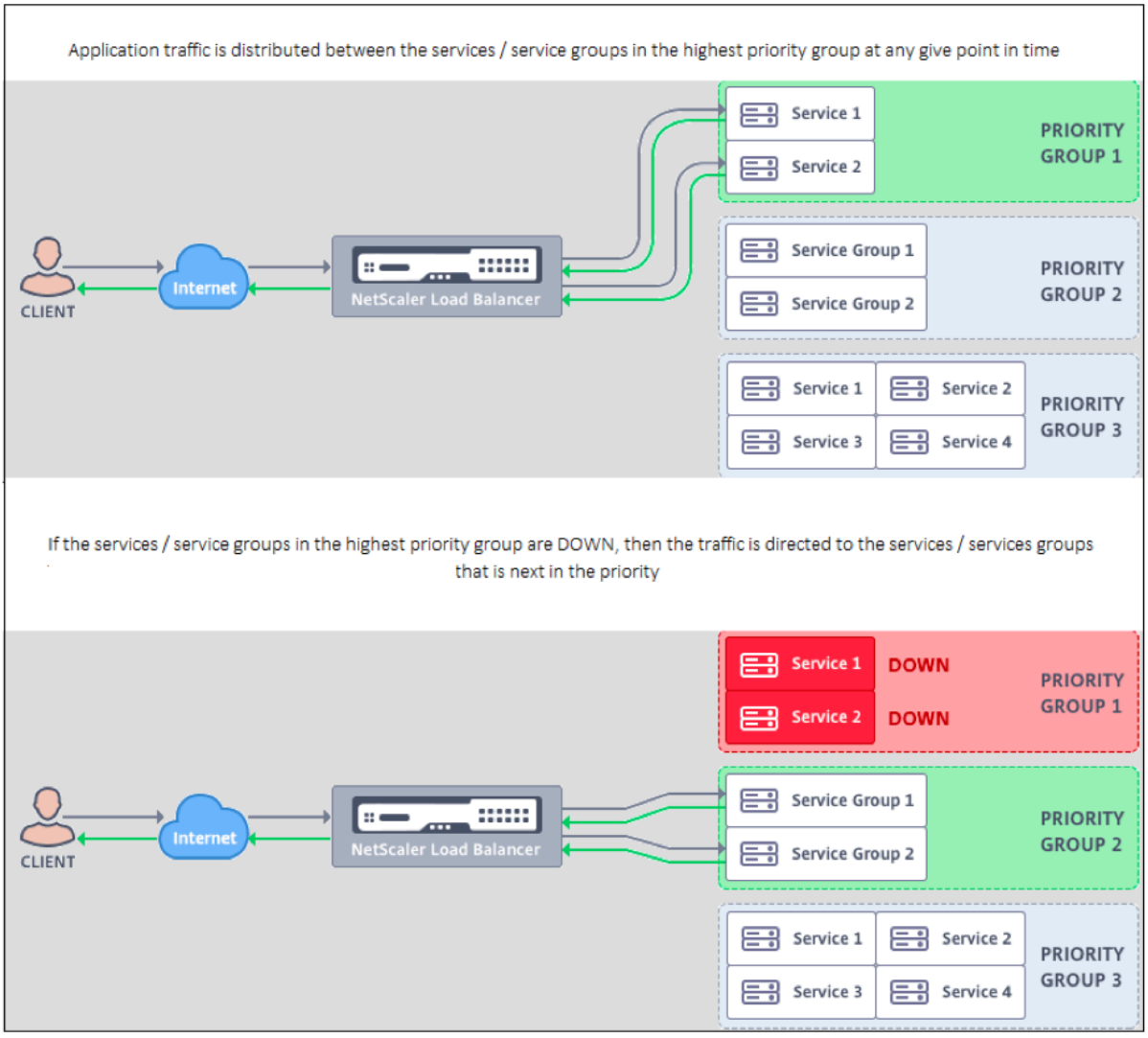
- SVG3 —優先度 3

このシナリオでは、アプリケーショントラフィックはサービスグループ SVG1 に転送されます。これは、このグループには最も低い優先順位番号が割り当てられているためです。SVG1 のすべてのメンバーが DOWN の場合、トラフィックはサービスグループ SVG2 に分散されます。これは、このグループに 1 つ低い優先順位番号が割り当てられるためです。SVG2 のすべてのメンバーも DOWN の場合、トラフィックは SVG3 に分散されます。ただし、SVG1 のメンバーのいずれかが UP の場合、SVG1 には最も小さい番号が割り当てられ、優先順位が最も高いため、トラフィックは SVG1 にリダイレクトされます。

サービスまたはサービスグループにプライオリティを割り当てて、必要に応じていつでもプロダクショントラフィックへの影響を最小限に抑えながら、プライオリティが最も高い特定のサービスまたはサービスグループをアップグレードできます。

また、アップグレードが失敗した場合でも、本番トラフィックへの影響を最小限に抑えながら、次の優先順位のサービスまたはサービスグループに安全に切り替えることができます。

次の図は、優先ロードバランシング機能を示しています。



優先ロードバランシングの設定

> 注
>
> NetScaler の優先負荷分散構成は、GUI でのみサポートされます。CLI を使用してプライオリティロードバランシングを設定することはできません。

1. [トラフィック管理] > [優先負荷分散] > [仮想 ***** サーバー] に移動し、仮想サーバーのプロトコル、IP アドレス、および仮想サーバーのポート番号を指定します。
2. 「最大優先グループ」ボックスに、この仮想サーバーにバインドできる優先サービスまたはサービスグループの数を入力します。デフォルト値は 2 で、設定できる最大優先度は 10 です。このパラメータは、設定後は編集できません。

   注:

   優先グループの最大数を指定して「**OK**」をクリックすると、コンテンツスイッチング仮想サーバーと「n」個のバックアップ負荷分散仮想サーバーが作成されます。アルファベット「n」は優先グループの最大数を表します。

   たとえば、仮想サーバー名を vs1 と入力し、最大優先度グループを 5 に設定した場合、`_Pri.LB#vs1#MaxPri=5` その名前のコンテンツスイッチング仮想サーバーと、次の 5 つの負荷分散仮想サーバーが作成されます。

   - `_Pri.LB#vs1#MaxPri=5_LB1`
   - `_Pri.LB#vs1#MaxPri=5_LB2`
   - `_Pri.LB#vs1#MaxPri=5_LB3`
   - `_Pri.LB#vs1#MaxPri=5_LB4`
   - `_Pri.LB#vs1#MaxPri=5_LB5`
3. 優先グループの最大数を指定して「**OK**」をクリックすると、このコンテンツスイッチング仮想サーバーにバインドする必要がある 1 つまたは複数のサービスグループを選択するように求められます。

   - サービスを仮想サーバーにバインドするには、「サービス」セクションの「挿入」をクリックします。次に、既存のサービスを選択するか、サービスを作成してこのサービスの優先度を設定します。また、このサービスをバインドするプライオリティ番号も設定してください。
   - サービスグループを仮想サーバーにバインドするには、「サービスグループ」セクションの「挿入」をクリックします。次に、既存のサービスグループを選択するか、サービスグループを作成して、このサービスグループの優先順位を設定します。また、このサービスグループをバインドするプライオリティ番号も設定してください。

   入力した優先グループの最大数に応じて、手順 3 を繰り返します。

   注:

   - 優先順位が最も高いサービスまたはサービスグループは、最も優先順位の高い負荷分散仮想サーバーにバインドされます。

たとえば、サービスグループSG_App1 and SG_App2にそれぞれ優先度 1 と 2 を割り当てた場合、SG_App1 はvirtual server _Priにバインドされます。LB#vs1#MaxPri=5_LB1 and SG_App2はvirtual server _Priにバインドされます。LB#vs1#MaxPri=5_LB2は ステップ 2 で作成しました。

- サービスグループまたはサービスの優先度を変更するには、優先度負荷分散仮想サーバーページの編集アイコンをクリックし、必要に応じて優先度を変更します。
- すべての負荷分散仮想サーバーの構成が同じであるため、各仮想サーバーの負荷分散方法と永続性を明示的に設定することはできません。

4.「詳細設定」セクションから、要件を満たす他の構成を完了します。

> 重要:
>
> 優先ロードバランシングの設定中に作成されたエンティティは、GUI の他のタブや CLI から変更してはなりません。優先度負荷分散エンティティは、「優先度負荷分散」タブからのみ変更することをお勧めします。

NetScaler ADC 拡張機能

August 15, 2023

NetScaler 拡張機能を使用して、拡張コードを記述することで NetScaler アプライアンスをカスタマイズできます。現在、ポリシー拡張とプロトコル拡張がサポートされています。ポリシー拡張を使用してポリシー言語を拡張できます。プロトコル拡張を使用して、NetScaler アプライアンスにカスタムプロトコルのサポートを追加できます。

NetScaler 拡張機能は NetScaler CPX でもサポートされています。

このドキュメントでは、次の内容について説明します。

- NetScaler 拡張機能-言語の概要
- NetScaler 拡張機能-ライブラリリファレンス
- NetScaler 拡張 API リファレンス
- プロトコル拡張
- ポリシー拡張

NetScaler の拡張 - 言語の概要

August 15, 2023

拡張言語は Lua 5.2 プログラミング言語に基づいています。Lua は、NetScaler ソフトウェアなどの C プログラムへの組み込み向けに設計された、パフォーマンスの高いコンパクトな実行エンジンを提供します。

拡張言語は動的に型付けされます。つまり、各オブジェクトには独自の型情報が含まれます。どの変数も実行中いつでも任意の型を保持できるため、変数型は宣言されません。

言語も自由形式で、トークン間の空白は無視されます。文はセミコロンで区切ってもかまいませんが、必須ではなく、通常はセミコロンで区切ることもありません。ステートメントのブロックは通常、最後で終了します。C や Java では {と} のようにブロックを括弧で囲むことはできません。

識別子は、文字 (a ～z および A ～Z)、数字 (0 ～9)、およびアンダースコア (_) のシーケンスで、数字で始まりません。識別子は大文字と小文字が区別されるため、var、VAR、Var はすべて異なる識別子です。

コメントは--で始まります。--以降はすべて行末まで無視されます。例:

```
-- This is a comment.
```

単純型

August 15, 2023

この言語では、次の単純なタイプの値を使用できます。

- 数字
- ストリング
- ブーリアン型
- ゼロ
- その他のタイプ

数字

すべての数値 (整数も含む) は IEEE 754 浮動小数点値で表されます。2^54 までの整数は正確に表現されます。数値は次のように表すことができます。

- 符号付きおよび符号なしの 10 進整数 (例:10、-5)
- 小数点を含む実数 (10.5、3.14159)
- 指数を含む実数 (1.0e+10)
- Hexadecimals (0xffff0000)

NetScaler のポリシー表現には 3 つの数値タイプがあります。

- 32 ビットの整数 (num_at)
- 64 ビットの整数 (符号なし長整数)
- 64 ビット浮動小数点 (double_at)

これらはすべて、拡張関数に渡されると数値型に変換され、数値は返されるときに必要なポリシー数値型に変換されます。

ストリング

文字列は任意の長さのバイトシーケンスです。これらはポリシーの **text_at** タイプに対応しています。文字列には NULL (0x00) バイトを含めることができます。任意のバイナリデータを、任意の文字コード表現 (UTF-8 や完全な Unicode など) を含む文字列に保持できます。ただし、**string.upper ()** のような文字列関数は 8 ビットの ASCII を前提としています。

文字列は使用時に自動的に割り当てられます。文字列にバッファを明示的に割り当てる必要はありません (あるいは方法さえありません)。また、文字列は使用されなくなると、ガベージコレクションによって自動的に割り当てが解除されます。文字列を明示的に解放する必要はありません (あるいは方法さえありません)。この自動割り当てと割り当て解除により、メモリリークやダングリングポインタなど、C 言語でよく発生する問題を回避できます。

文字列リテラルは、二重引用符または一重引用符で囲まれた文字列です。2 種類の引用符に違いはありません。「文字列リテラル」は「文字列リテラル」と同じです。通常のバックスラッシュエスケープには、\ s (ベル)、\ b (バックスペース)、\ f (フォームフィード)、\ n (改行/改行フィード)、\ t (水平タブ)、\\ (バックスラッシュ)、\ “(二重引用符)、\ ‘(一重引用符) があります。10 進数のバイト値は、バックスラッシュと 1 ～3 桁 (\ d、\ dd、\ ddd) で入力できます。16 進数のバイト値は、バックスラッシュ、x、および 2 桁の 16 進数 (\ xhh) で入力できます。

長い複数行の文字列リテラルには、長括弧表記という特別な構文を使用できます。この表記法では、文字列を二重の角括弧で囲み、括弧の間に 0 個以上の等号を付けます。つまり、文字列にはない括弧と等号の組み合わせを考え出すというものです。文字列にはエスケープシーケンスは指定されません。例:

[[これは長括弧表記を使用した複数行の文字列です。]]

[= [[and]] とエスケープされていない長い表記を使った複数行の文字列です。] =]

長い括弧表記は、複数行のコメントを作成するために使用することができます。例:

```
-[[
これは複数行のコメントです。
-]]
```

ブーリアン型

通常の true と false のブーリアン値が表示されます。0 が false、0 以外の値が true と見なされる C とは対照的に、ブール値は数値とは異なることに注意してください。

ゼロ

nil は「値なし」を意味する特別な値です。NULL がゼロと定義されている C とは対照的に、これは独自の型であり、他の値とは等しくありません。

その他のタイプ

他にユーザーデータとスレッドの 2 つのタイプがあります。これらは高度なトピックであり、ここでは説明しません。

変数

August 15, 2023

変数は、拡張機能の実行中に変更される可能性のある値を保持します。動的型付けのため、どの変数にも任意の型の値を保持できます。変数には型宣言はありません。代わりに、変数のタイプは実行時に決定されます。実際、変数の値の型は実行中に変わる可能性がありますが、これはお勧めできません。変数の初期値は nil です。

変数名は識別子であり、数字で始まらない文字、数字、アンダースコアも識別子です。例: ヘッダー、結合ヘッダー。

グローバル変数

Lua では、特に宣言されていない変数はプログラム内でグローバルです。ただし、グローバル変数はポリシー拡張関数では使用できません。これは、関数を実行できるパケットエンジンが複数あり、各パケットエンジンには独自のメモリがあるためです。

エクステンションでグローバル変数を使用すると、実行時エラーが発生します。**/var/log/ns.log** で報告されたグローバル変数を更新または作成しようとしたときです。

変数名の入力ミスは潜在的な問題となります。なぜなら、タイプミスがある変数は別のグローバル変数として解釈され、C や Java のような言語のように構文エラーを引き起こすことはないからです。前述のように、代わりにランタイムエラーが発生します。

ローカル変数

変数は、関数などのステートメントのブロックに対してローカルであると宣言できます。これはローカル変数名によって行われます。変数はブロックにスコープされます。つまり、変数はブロック内にのみ存在します。ローカル宣言は、オプションで変数に値を割り当てることができます。

例:

ローカルヘッダー **= {}**
ローカル複合ヘッダー **= {}**

式

August 15, 2023

式は変数とリテラル値から値を計算します。

- 算術演算
- リレーショナルオペレーション
- ロジカルオペレーション
- 連結
- 長さ
- 優先順位

算術演算

算術演算は数値に対して実行されます。文字列値を算術演算に使用すると、数値に変換されます。変換に失敗すると、エラーが返されます。

a + b	a と b を追加
a-b	a から b を引く
a * b	a と b を掛ける
a/b	a を b で割る
a% b	modulo = a - math.floor(a/b)*b
a ^ b	a を b 乗する。b には任意の数を指定できる
-a	否定する

リレーショナルオペレーション

関係演算は 2 つの値を比較し、関係が満たされている場合は true を返し、そうでない場合は false を返します。リレーショナル操作は、どのタイプの値でも実行できます。値が同じタイプでない場合は、false が返されます。数値は通常の方法で比較されます。文字列は、現在のロケールの照合順序を使用して比較されます。

a == b	a は b と等しい

a ~= b	a は b と等しくない
a < b	a は b より小さい
a > b	a は b より大きい
a <= b	a が b と等しいか、それより小さい
a >= b	a が b より大きいか、または等しい

ロジカルオペレーション

論理演算は従来、ブール値に対して実行されていましたが、この言語では任意の 2 つの値に対して実行できます。nil と false は false と見なされ、その他の値は true と見なされます。論理演算ではショートカット評価を使用します。最初の値が操作の結果を決定する場合、2 番目の値は評価されません。

a と b	a が false または nil の場合は a を返し、それ以外の場合は b を返します。
a or b	a が false ではなく nil でもない場合は a を返し、それ以外の場合は b を返します。
not a	a が false でないか nil の場合は false を返し、それ以外の場合は true を返します

and および or 演算は、式内の条件付き評価に使用できます。

a or b	a が初期化されていない (nil) 場合は、これを使用してデフォルト値 b を指定できます。これは関数のオプションパラメータに役立ちます。
a と b または c	条件 a に基づいて、nil 以外の b または c を選択できます。a が true の場合、a と b は b を返し、b または c は b を返します。a が false の場合、a と b は false を返し、false または c は c を返します。これは? と同等です。b: C プログラミング言語の c。

連結

文字列の連結は s1.. s2 です。これにより、s1 と s2 の内容を保持するのに十分な大きさの新しい文字列が作成され、その内容が新しい文字列にコピーされます。s1 または s2 が文字列でない場合、エラーが発生します。連結を繰り返すと、かなりのコピーオーバーヘッドが発生する可能性があることに注意してください。一度に 1 バイトずつ連結して n バイトの文字列を作成すると、n* (n+1) /2 バイトがコピーされます。パフォーマンスを向上させるには、文字列の一部を連結してテーブル (後述) に入れ、table.concat () 関数を使用できます。この例が COMBINE_HEADERS () の例に示されています。

長さ

文字列 s の長さは #s によって返されます。＃演算子は、後で説明するように、配列テーブルでも使用されます。

優先順位

演算子の優先順位によって、式で実行される演算の順序が決まります。優先順位の高い操作は、優先順位の低い操作より先に実行されます。優先順位は通常どおり、括弧で上書きできます。たとえば、in `a + b \* c`, * は + よりも優先順位が高いため、`a + (b \* c)` 式は次のように評価されます。

最高	
-	not # - (unary)
-	• / %
-	..
-	= ~= < > <= >=
-	変更前
最低	または

同じ優先順位の演算は左から右 (左連想) に実行されますが、^ と.. は右から左 (右連想) に実行されます。したがって、^ b ^ c は a^（b ^ c）として評価されます。

割り当て

August 15, 2023

代入ステートメントは式を評価し、結果の値を変数に割り当てます。

```
variable = expression
```

前述のように、どの型の値も任意の変数に割り当てることができるため、次のことが許可されます。

```
local v1 = "a string literal"
v1 = 10
```

代入ステートメントでは、実際には次の形式で複数の変数を設定できます。

variable1, variable2, ···= expression1, expression2, ···

式よりも多くの変数がある場合、余分な変数には nil が割り当てられます。変数よりも式の方が多い場合、余分な式の値は破棄されます。式はすべて代入前に評価されるので、これを使って 2 つの変数の値を簡潔に交換できます。

v1, v2 = v2, v1

は次と等価です。

tmp = v1
v2 = v1
v1 = tmp

テーブル

August 15, 2023

テーブルは、キーと値を含むエントリのコレクションです。これらは提供される唯一の集約データ構造です。その他のデータ構造 (配列、リスト、セットなど) はすべてテーブルから構築されます。テーブルのキーと値は、他のテーブルを含め、どのタイプでもかまいません。同じテーブル内のキーと値には型が混在する可能性があります。

- テーブルコンストラクター
- テーブル使用量
- 配列としてのテーブル
- レコードとしてのテーブル

テーブルコンストラクター

テーブルコンストラクターを使用すると、キーと関連する値を含むテーブルを指定できます。構文は次のとおりです。

{[キー 1] = 値 1, [キー 2] = 値 2,···}

キーと値は式です。キーが予約語ではない文字列の場合、キーを囲む括弧と引用符は省略できます。例:

{key1 = “value1”, key2 = “value2”, key3 = “value3”}

空のテーブルは {} だけで指定できます。

テーブルコンストラクターを代入に使用して、テーブルを参照する変数を設定できます。例:

local t1 = {} –set t1 to an empty table
local t2 = {key1 = “value1”, key2 = “value2”, key3 = “value3”}

テーブル自体は匿名であることに注意してください。複数の変数が同じテーブルを参照する場合があります。上記の例を続けます。

ローカル t3 = t2–t2 と t3 の両方が同じテーブルを参照しています

テーブル使用量

期待どおりに、キーを使用してテーブル内の値を検索できます。構文は table [key] です。ここで table はテーブル参照 (通常はテーブルに割り当てられた変数) で、key はキーを提供する式です。これが式で使用され、キーがテーブルに存在する場合、キーに関連付けられた値が返されます。キーがテーブルにない場合は nil を返します。これを代入の変数として使用しても、キーがテーブルに存在しない場合、キーと値の新しいエントリが作成されます。キーがテーブルにすでに存在する場合、キーの値を新しい値に置き換えます。例:

local t = {}–t を空のテーブル
t [“k1”] =「v1” に設定します–キー「k1」と値「v1” = t [” k1”] のエントリを作成します–v1 をキー「
k1」=「v1” t [” k1”] =「new_v1”–キー「k1” の値を設定します」から「
新規 _v1」

配列としてのテーブル

従来の配列は、整数キーをインデックスとするテーブルを使用して実装できます。配列には負のインデックスを含めて任意のインデックスを付けることができますが、慣例としては、配列はインデックス 1 から始まります (C や Java のような言語のように 0 ではありません)。このような配列には特別な用途のテーブルコンストラクタがあります。

{value1, value2, value3, ···}

その場合、配列参照は配列 [インデックス] になります。

長さ演算子 # は、1 から始まる連続したインデックスを持つ配列の要素の数を返します。例:

local a = { “value1”, “value2”, “value3” }
local length = #a –sets length to the length of array a = 3

配列は、定義された要素のみが割り当てられるスパース配列でもかまいません。ただし、# は、連続しないインデックスを持つスパース配列では使用できません。例:

local sparse_array = {}–空の配列をセットアップ
sparse_array [1] =「value1”–インデックス 1 に要素を追加
sparse_array [99] =「value99」–インデックス 99 に要素を追加

多次元配列は、テーブルのテーブルとして設定できます。たとえば、3x3 マトリックスは次のように設定できます。

local m = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9} }
local v22 = m[2][2] –sets v22 to 5

レコードとしてのテーブル

フィールドを持つレコードは、フィールド名キーを持つテーブルとして実装できます。参照フォームの table.field はテーブル [“field”] に使用できます。例:

local person = {name = “John Smith”, phone = “777-777-7777”}
local name = person.name –sets name to “John Smith”

テーブルの配列をレコードのシーケンスに使用できます。例:

local people = {
{name = “John Smith”, phone = “777-777-7777”},
{name = “Jane Doe”, phone = “888-888-8888”}
…
}

名前 = 人 [2] .name–名前を「Jane Doe」に設定します

制御構造

August 15, 2023

拡張関数言語には、プログラムの実行を制御する通常のステートメントが用意されています。

- もしそうなら
- 実行しながら繰り返す
- 数値形式
- ブレーク
- Goto

もしそうなら

If ステートメントは、1 つ以上の条件に基づいて実行するステートメントのブロックを選択します。次の 3 つの形式があります。

If then Form

```
if expression then
     statements to execute if expression is not false or nil
end
<!--NeedCopy-->
```

If then else Form

```
if expression then
     statements to execute if expression is not false or nil
else
     statements to execute if expression is false or nil
end
<!--NeedCopy-->
```

If then elseif else Form

```
if expression1 then
     statements to execute if expression1 is not false or nil
     elseif expression2 then
          statements to execute if expression2 is not false or nil
. . .
else
     statements to execute if all expressions are false or nil
end
<!--NeedCopy-->
```

例:

```
if headers[name] then

     local next_value_index = #(headers[name]) + 1
     headers[name][next_value_index] = value

else

     headers[name] = {
 name .. ":" .. value }


end
<!--NeedCopy-->
```

注:

- C や Java の場合のように、式は括弧で囲まれていません。

- C/Java スイッチステートメントと同等のものはありません。同等の処理を行うには、一連の if elseif ステートメントを使用する必要があります。

実行しながら繰り返す

while ステートメントと **repeat** ステートメントでは、エクスプレッションによってループを制御できます。

```
while expression do
     statements to execute while expression is not false or nil
end

repeat

     statements to execute until expression is not false or nil

until expression
<!--NeedCopy-->
```

while の例**:**

```
local a = {
 1, 2, 3, 4 }

local sum, i = 0, 1 -- multiple assignment initializing sum and i
while i <= #a do -- check if at the end of the array
     sum = sum + a[i] -- add array element with index i to sum
     i = i + 1 -- move to the next element
end
<!--NeedCopy-->
```

リピートの例**:**

```
sum, i = 0, 1 -- multiple assignment initializing sum and i
repeat
     sum = sum + a[i] -- add array element with index i to sum
     i = i + 1 -- move to the next element
until i > #a -- check if past the end of the array
<!--NeedCopy-->
```

もちろん、これらの例のいずれかで i = i + 1 ステートメントを省略した場合など、終了しないループを書くことも可能です。このような関数が実行されると、NetScaler はその関数が妥当な時間内に完了しなかったことを検出し、実行時エラーを出して強制終了します。

```
Cpu limit reached. Terminating extension execution in [[string "
function extension function..."]]: line line-number.
```

/var/log/ns.log にレポートされます。

数値形式

for ループには 2 つのタイプがあります。1 つ目は数値の for です。これは C や Java で通常の for 文の使い方と似ています。numeric for ステートメントは変数を初期化し、変数が最終値を渡したかどうかをテストします。渡されなかった場合は、ステートメントのブロックを実行して変数をインクリメントして繰り返します。数値の for ループの構文は以下のとおりです。

```
for variable = initial, final, increment do

     statements in the loop body

end
<!--NeedCopy-->
```

ここで、initial、final、および increment はすべて数値を生成する (または数値に変換できる) 式です。variable は for ループステートメントブロックのローカルと見なされ、ループ外では使用できません。increment は省略できます。デフォルトは 1 です。式はループの開始時に 1 回評価されます。終了条件は、インクリメントが正の場合は変数 > final で、インクリメントが負の場合は変数 < final です。インクリメントが 0 の場合、ループはすぐに終了します。

例 (前のセクションの while ループと repeat ループと同等):

```
sum = 0
for i = 1, #a do -- increment defaults to 1
     sum = sum + a[i]
end
<!--NeedCopy-->
```

for ループの 2 番目のタイプは汎用の for で、より柔軟なタイプのループに使用できます。これには関数の使用が含まれるため、関数が導入された後に説明します。

ブレーク

break ステートメントは while、repeat、for ループの中で使用します。ループを終了し、ループの後の最初のステートメントで実行を再開します。例 (前述の while ループ、repeat ループ、for ループと同等です):

```
sum, i = 0, 1
while true do
     if i > #a then
          break
     end
     sum = sum + a[i]
     i = i + 1
end
<!--NeedCopy-->
```


Goto

goto ステートメントを使用すると、ラベルに移動したり戻ったりできます。ラベルは識別子で、構文は:: label:: です。goto ステートメントは goto ラベルです。例 (ここでも前述のループと同等):

```
sum, i = 0, 1
::start_loop::
     if i > #a then
          goto end_loop -- forward jump
     end
     sum = sum + a[i]
     i = i + 1
     goto start_loop -- backwards jump
::end_loop::
. . .
<!--NeedCopy-->
```

プログラミングで goto を使うことについては、長い間論争が続いています。一般的には、関数を読みやすく信頼性のあるものにするために、他の制御構造を使用するようにしてください。しかし、ときどき goto を賢く使うことで、より良いプログラムにつながるかもしれません。特に、gotos はエラーの処理に役立ちます。

関数

August 15, 2023

関数はプログラミングの基本的な構成要素であり、タスクを実行するステートメントをグループ化するための便利で強力な方法です。これらは、NetScaler ADC アプライアンスと拡張コードの間のインターフェイスです。ポリシーの場合は、ポリシー拡張関数を定義します。プロトコルの場合は、プロトコル動作のコールバック関数を実装します。関数は、関数との間で渡される値とその関数に対して実行されるステートメントを指定する関数定義、および特定の入力データで関数を実行し、関数から結果を取得する関数呼び出しで構成されます。

プロトコル動作コールバック関数

TCP クライアントの動作は、TCP クライアントデータストリームイベントを処理するコールバック関数（on_data）で構成されます。TCP ベースのプロトコルのメッセージベース負荷分散 (MBLB) を実装するには、このコールバック関数のコードを追加して、クライアントからの TCP データストリームを処理し、バイトストリームを解析してプロトコルメッセージにします。

ビヘイビアのコールバック関数は、処理モジュールの状態であるコンテキストとともに呼び出されます。コンテキストは、処理モジュールのインスタンスです。たとえば、TCP クライアント動作コールバックは、クライアント TCP 接続ごとに異なるコンテキストで呼び出されます。

コンテキストに加えて、ビヘイビアコールバックは他の引数を持つことができます。通常、残りの引数はペイロードとして渡されます。ペイロードはすべての引数のコレクションです。したがって、プログラマブルプロセッシングモジュールのインスタンスは、インスタンス状態とイベントコールバック関数の組み合わせ、つまりコンテキストと動作の組み合わせと見なすことができます。トラフィックはイベントペイロードとしてパイプラインを通過します。

TCP クライアントコールバック関数のプロトタイプ:

```
1
2                   Function                  client on_data (ctxt, payload)
3
4                                                     //.code
5
6                   end
7
8
9   <!--NeedCopy-->
```

各項目の意味は次のとおりです。

- ctxt -TCP クライアント処理コンテキスト
- ペイロード—イベントペイロード
  - payload.data -受信した TCP データ、バイトストリームとして利用可能

ポリシー拡張関数

NetScaler ポリシー式言語は型指定されているため、拡張関数の定義では、入力の種類と戻り値を指定する必要があります。**Lua** 関数定義は、次の型を含むように拡張されました。

```
1  function self-type: function-name(parameter1: parameter1-type, and so
       on): return-type
2        statements
3  end
4
5  <!--NeedCopy-->
```

各項目の意味は次のとおりです。

型は NSTEXT、NSNUM、NSBOOL、または NSDOUBLE。

自己型は、関数に渡される暗黙的な自己パラメータの型です。拡張関数が NetScaler ADC ポリシー式で使用される場合、これは関数の左側の式によって生成される値です。これを表示するもう 1 つの方法は、関数がそのタイプを NetScaler ADC ポリシー言語で拡張することです。

parameter-types は、ポリシー式の拡張関数呼び出しで指定された各パラメータの型です。拡張関数は 0 個以上のパラメータを持つことができます。

return-type は、拡張関数の呼び出しによって返される値の型です。これは、ポリシー式の部分（存在する場合）への入力であり、関数の右側にある場合、または式の結果の値です。

例:

```
function NSTEXT:COMBINE_HEADERS(): NSTEXT
```

ポリシー式での拡張関数の使用:

```
HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n").COMBINE_HEADERS()
```

ここで、自己パラメータは`HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n")`の結果であり、これはテキスト値です。COMBINE_HEADERS () 呼び出しの結果はテキストであり、この呼び出しの右側に何もないので、式全体の結果はテキストになります。

ローカル関数定義

拡張関数のほかに、拡張ファイルにグローバル関数を定義することはできません。しかし、ローカル関数は、通常の Lua 関数ステートメントを使用して拡張関数内で定義できます。これは、関数の名前とそのパラメータの名前 (引数とも呼ばれます) を宣言し、Lua のすべての宣言と同様に、型を指定しません。この構文は次のとおりです。

```
local function function-name(parameter1-name, parameter2-name, and so
   on)
     statements
end

<!--NeedCopy-->
```

関数名とパラメータ名はすべて識別子です。(関数名は実際には変数であり、関数ステートメントはローカル関数名 = 関数（parameter1 など）の省略形ですが、関数を使用するためにこの微妙なことを理解する必要はありません。)

ここでは、通常の…の代わりにパラメータ名のパターンの継続に使用されていることに注意してください。これは、…自体が実際には可変パラメータリストを意味するためです。ここでは説明しません。

関数本体とリターン

function ステートメントと end ステートメントの間のステートメントのブロックは、関数本体です。関数本体では、関数パラメータはローカル変数のように動作し、前述のように関数呼び出しによって提供される値を持ちます。

return 文は、関数の呼び出し元に返される値を提供します。これは、ブロックの最後 (関数内、if then、for ループなど) に出現しなければなりません。それはそれ自身のブロックにあることができます do return…end)。戻り値を指定しない、1 つ、または複数指定する。

```
return -- returns nil
return expression -- one return value
return expression1, expression2, ... -- multiple return values

<!--NeedCopy-->
```

例:


```
local function fsum(a)
     local sum = 0
     for i = 1, #a do
          sum = sum + a[i]
     end
     return sum
end

Local function fsum_and_average(a)
     local sum = 0
     for i = 1, #a do
          sum = sum + a[i]
     end
     return sum, sum/#a
end

<!--NeedCopy-->
```

関数呼び出し

関数呼び出しは、関数の本体を実行し、そのパラメータの値を指定し、結果を受け取ります。関数呼び出しの構文は関数名 (式 1、式 2 など) で、関数パラメーターは対応する式に設定されます。式とパラメータの数は同じである必要はありません。パラメータより式の数が少ない場合、残りのパラメータは nil に設定されます。したがって、呼び出しの最後に 1 つ以上のパラメータをオプションにすることができ、関数はそれらが nil でないかどうかをチェックすることによって指定されているかどうかをチェックできます。これを行う一般的な方法は OR 演算です。

```
function f(p1, p2) -- p2 is optional
     p2 = p2 or 0 -- if p2 is nil, set to a default of 0
     . . .
end

<!--NeedCopy-->
```

パラメータよりも多くのエクスプレッションがある場合、残りのエクスプレッション値は無視されます。

前述のとおり、関数は複数の値を返すことができます。これらの戻り値は、複数の代入ステートメントで使用できます。例:

```
local my_array = {
 1, 2, 3, 4 }

local my_sum, my_ave = sum_and_average(my_array)

<!--NeedCopy-->
```


イテレータ関数とジェネリック **for** ループ

関数を導入したので、ジェネリックな for ループについて話すことができます。ジェネリック for ループ (変数が 1 つ) の構文は次のとおりです。

```
for variable in iterator(parameter1, parameter2, and so on) do
     statements in the for loop body
end

<!--NeedCopy-->
```

iterator () は、ループ本体の各反復で変数の値を提供する 0 個以上のパラメーターを持つ関数です。イテレータ関数は、クロージャと呼ばれる手法を使用して、反復のどこにあるかを追跡します。ここで心配する必要はありません。nil を返すことで、反復の終了を知らせます。イテレータ関数は、複数代入で使用するために、複数の値を返すことができます。

イテレータ関数の記述はこのホワイトペーパーの範囲を超えていますが、この概念を説明する便利なビルトインイテレータはほとんどありません。1 つは pairs () イテレータで、テーブル内のエントリを反復処理し、次のエントリのキーと値の 2 つの値を返します。

例**:**

```
local t = {
 k1 = "v1", k2 = "v2", k3 = "v3" }

local a = {
  }
  -- array to accumulate key-value pairs
local n = 0 -- number of key-value pairs
for key, value in pairs(t) do
     n = n + 1
     a[n] = key.. " = ".. Value -- add key-value pair to the array
end
local s = table.concat(a, ";") -- concatenate all key-value pairs into
   one string

<!--NeedCopy-->
```

もう 1 つの便利なイテレータは、`string.gmatch()COMBINE_HEADERS()` 次の例で使用される関数です。

NetScaler の拡張 **-** ライブラリリファレンス

August 15, 2023

ポリシー拡張でサポートされているライブラリのリスト。

- 基本ライブラリ
- String ライブラリ
- 正規表現パターン-文字クラス
- 正規表現パターン-パターンアイテム
- Table ライブラリ
- 数学ライブラリ
- ビットワイズライブラリ
- オペレーティング・システム・ライブラリ
- NetScaler ライブラリ

基本ライブラリ

assert(v[,message])	v が false の場合、オプションのメッセージとともにエラーを発行します。
error(message)	関数を終了し、エラーメッセージを報告します。
ipairs(a)	Iterator for an array a. 各反復のインデックスと値を返します。
pairs(t)	テーブル t のイテレータ。各反復のキーと値を返します。
tonumber(e[,base])	e を数値に変換し、オプションの基数を指定します。
tostring(v)	v を文字列に変換します
type(v)	v のタイプ (数値、文字列、ブール値、テーブルなど) を返します。
getmetatable (object)	オブジェクトにメタテーブルがない場合は nil を返します。それ以外の場合、オブジェクトのメタテーブルに「__metatable」フィールドがある場合は、関連する値を返します。それ以外の場合は、指定されたオブジェクトのメタテーブルを返します。
setmetatable (table, metatable)	指定されたテーブルのメタテーブルを設定します。(Lua から他のタイプのメタテーブルを変更することはできません。C からしか変更できません) メタテーブルが nil の場合、指定されたテーブルのメタテーブルを削除します。元のメタテーブルに「__metatable」フィールドがある場合、エラーが発生します。
select (index, …)	引数番号インデックスの後のすべての引数を返します。インデックスが文字列「#」の場合は、それが受け取った余分な引数の合計数を返します。

pcall (f [, arg1, ···])	プロテクトモードで指定された引数で関数 f を呼び出します。最初の結果として、呼び出しが成功したかどうかを示すステータスコードが返されます。呼び出しが成功した場合は、ステータスコードとともに呼び出しのすべての結果も返し、それ以外の場合はエラーメッセージを返します。
xpcall (f, msgh [, arg1, ···])	この関数は pcall に似ていますが、エラー処理用の引数も取る点が異なります。
_VERSION	現在のインタプリタのバージョンを返します。

String ライブラリ

string.byte(s[,i[,j]])	s [i] から s [j] までのバイト値を返します。デフォルト i = 1 および j = i
string.char(···)	整数パラメータで構成された文字列を返します。
string.find(s,pattern[,init[,plain])	s 内の正規表現パターンが最初に一致するものを検索します。match または nil の最初と最後のインデックスを返します。init は開始時のインデックスで、デフォルトは 1 です。plain = true はパターンが正規表現ではないことを意味します。
string.format(form,···)	パラメータのフォーマットされたバージョンを返します。
string.gmatch(s,pattern)	正規表現パターンで s を検索するためのイテレータ。一致する値を返します。
string.gsub(s,pattern,repl[,n])	パターンのすべての（または n）出現が repl に置き換えられた s のコピーを返します。
string.len(s)	文字列の長さを返します。
string.lower(s)	小文字に変換された文字列のコピーを返します。
string.match(s,pattern[,init])	s の正規表現パターンの最初の一致を検索し、キャプチャまたはパターン全体を返します。init は開始するインデックス、デフォルトは 1 です。
string.rep(s,n[,sep])	セパレータ sep、デフォルトなしセパレータで、s の n 個のコピーである文字列を返します
string.reverse(s)	逆にされた文字列を返します。

string.sub(s,i[,j])	s [i] から s [j] までの部分文字列を返します。デフォルトの j は文字列の末尾です。
string.upper(s)	大文字に変換された文字列のコピーを返します。
string.dump (function)	指定された関数のバイナリ表現を含む文字列を返します。

正規表現パターン-文字クラス

x	文字 x (マジック文字 ^$ ()%. [] *+-?)
.	任意の文字
%a	任意の文字
%c	任意の制御文字
%d	任意の数字
%g	スペースを除くすべての印刷可能な文字
%l	任意の小文字
%p	任意の句読点
%s	任意の空白文字
%u	任意の大文字
%w	任意の英数字
%x	エスケープされたマジック文字 x (例:%%)
[set]	文字セット: 個々の文字のシーケンス、範囲 x、y、% クラス
[^set]	セットにない文字。

正規表現パターン-パターンアイテム

X	文字クラス
X*	X 内の文字の最長反復回数 0 回以上

X+	X 内の 1 回以上の文字の繰り返し
X-	X 内の文字の最短の繰り返し 0 回以上
X?	X の 0 または 1 文字
%n	n=1 ～9; キャプチャされた n 番目の文字列に一致します
%bxy	バランスの取れた 2 つの文字 x と y の間の部分文字列をマッチさせます。例%b () は、2 つの平衡括弧の間の部分文字列を照合します。
%f[set]	次の文字が set に属し、前の文字が set に属さないような任意の位置の空文字列にマッチします。

パターンは一連のパターンアイテムです。^pattern は文字列の先頭と一致し、pattern$ は文字列の末尾と一致します。

一致した部分文字列は (pattern) を使用してキャプチャできます。パターン () のない括弧は、現在の文字列の位置 (数値) をキャプチャします。

Table ライブラリ

table.concat(list[,sep[,i[,j]]])	文字列リスト [i].. sep.. list [i+1].. sep…list [j] を返します。デフォルト sep は空の文字列です。既定値 i は 1、j は #list です。
table.insert(list,[pos,]value)	インデックス pos のリストに値を挿入します。pos のデフォルトは #list (リストの最後) です。
table.pack(…)	インデックス 1 から始まるパラメータと、パラメータの総数を示すキー n を含む配列を返します。
table.remove(list[,pos])	位置を埋めるために要素をシフトし、位置 pos で要素をリストから削除します。削除された要素を返します。posis #list のデフォルト (リストの最後)
table.sort(list[,comp])	リストの要素をソートします。comp は、使用する比較関数です。Default for comp is <.
table.unpack(list[,i[,j]])	リスト [i] からリスト [j] までを返します。i のデフォルトは 1 で、j は #list です <c/ode>。

数学ライブラリ

さまざまな三角関数と対数関数は示されていません。

math.abs(x)	x の絶対値を返します。
math.ceil(x)	x 以上の最小の整数を返します。
math.floor(x)	最大整数 <= x を返します。
math.fmod (x, y)	商をゼロに向かって四捨五入した x/y の余りを返します。
math.huge	値 >= その他の数値。
math.max(x,···)	最大引数を返します。
math.min(x,···)	最小引数を返します。
math.modf(x)	x の整数部と小数部を返します。
math.random()	0 と 1 の間の疑似乱数を返します。
math.random(m)	1 から m までの疑似乱数整数を返します。
math.random(m, n)	m と n の間の疑似乱数整数を返します。
math.randomseed(x)	疑似乱数ジェネレータセットを x に設定します。
math.sqrt(x)	x (x^0.5) の平方根を返します
math.acos(x)	x のアークコサインを返します (ラジアン)。
math.asin(x)	x のアークサインを返します (ラジアン)。
math.atan(x)	x のアークタンジェント (ラジアン) を返します。
math.atan2(y, x)	y/x のアークタンジェント (ラジアン) を返します。
math.cos(x)	x のコサインを返します。
math.cosh(x)	x の双曲線余弦を返します。
math.sin(x)	x の正弦を返します。
math.sinh(x)	x の双曲線正弦を返します。
math.tan(x)	x のタンジェントを返します。
math.tanh(x)	x の双曲線タンジェントを返します。
math.deg(x)	角度 x (ラジアンで指定) を度単位で返します。
math.exp(x)	値 e^x を返します。
math.frexp (x)	x = m2e、e が整数で m の絶対値が [0.5, 1) の範囲に収まるように m と e を返します。

math.ldexp (m, e)	m2e (e は整数でなければなりません) を返します。
math.log (x [, base])	指定された底の x の対数を返します。base のデフォルトは e です。
math.pow (x, y)	x^y を返します。
math.rad (x)	角度 x (度で指定) をラジアン単位で返します。
math.pi	π の値。

ビットワイズライブラリ

特に明記されていない限り:

- すべての関数は、範囲（-2^51、+2^51）の数値引数を受け入れます。
- 各引数は 2^32 で除算された余りまで正規化され、(なんらかの方法で) 整数に切り捨てられ、最終値は [0,2^32-1] の範囲に収まります。
- すべての結果は [0,2^32-1] の範囲にあります。

bit32.arshift(x,disp)	右 (+disp) または左 (-disp) に算術的にシフトされた x 個の disp ビットを返します。
bit32.band(…)	引数のビット単位の論理和を返します。
bit32.bnot(x)	x のビット単位の否定を返します。
bit32.bor(…)	引数のビット単位の OR を返します。
bit32.btest(…)	引数のビット単位の AND がゼロでない場合は true を返します。
bit32.bxor(…)	引数のビット単位排他的論理和を返します。
bit32.extract(n,field[,width])	n の field から field + width - 1 までのビットを返します（最上位から最下位までのビット数)。幅のデフォルトは 1 です。
bit32.replace(n,v,field[,width])	n のコピーを、フィールドからフィールドへのビット数 + 幅 -1 を v に置き換えたものを返します。既定の幅は 1 です。
bit32.lrotate(x,disp)	左 (+disp) または右 (-disp) に回転した x 個のディスクビットを返します。

bit32.lshift(x,disp)	x 個のディスクビットを左 (+disp) または右 (-disp) にシフトして返します。
bit32.rrotate(x,disp)	右 (+disp) または左 (-disp) に回転した x 個のディスクビットを返します。
bit32.rshift(x,disp)	x 個のディスクビットを右 (+disp) または左 (-disp) にシフトして返します。

オペレーティング・システム・ライブラリ

os.clock ()	CPU 時間の秒単位の近似値を返します。
os.date ([format [, time]])	指定された文字列フォーマットに従ってフォーマットされた文字列または日付と時刻を含むテーブルを返します。
os.time ([table])	引数なしで呼び出されたときの現在の時刻、または指定されたテーブルで指定された日付と時刻を表す時刻を返します。
os.difftime (t2, t1)	時間 t1 から時間 t2 までの秒数を返します。

NetScaler ライブラリ

ns.logger:level(message)	レベルが「緊急」、「アラート」、「緊急」、「エラー」、「警告」、「通知」、「情報」、「デバッグ」のメッセージを記録します。パラメータは、C printf () 関数と同じです。書式文字列と、書式文字列の% 指定子に値を指定するための可変数の引数です。

NetScaler の拡張 **- API** リファレンス

August 15, 2023

動作とは、NetScaler アプライアンスで使用できる一般的なプログラム可能なパターンを形式化したものです。たとえば、TCP 仮想サーバーは TCP クライアントの動作と TCP サーバーの動作をサポートします。動作とは、あらかじめ定義されたコールバック関数のセットです。コールバック関数を提供することで動作を実装できます。たとえば、TCP クライアントの動作は TCP データストリームを処理する on_data 関数で構成できます。

TCP クライアントの動作

on_data -TCP クライアントデータイベントの関数コールバック。コールバックは次の 2 つの引数を取ります。

- **ctxt** -TCP クライアント処理コンテキスト
- ペイロード—イベントペイロード
  - **payload.data** -受信した TCP データ、バイトストリームとして利用可能

TCP サーバーの動作

on_data -TCP サーバーデータイベントの関数コールバック。コールバックは 2 つの引数を取ります。

- **ctxt** -TCP サーバ処理コンテキスト
- ペイロード—イベントペイロード
  - **payload.data** -受信した TCP データ、バイトストリームとして利用可能

TCP クライアントコンテキスト

TCP クライアントイベントコールバックに渡されるコンテキスト:

- **ctxt.output** -パイプライン内の次の処理コンテキスト。拡張コールバックハンドラーは、イベント DATA（メッセージの一部）またはプロトコルメッセージの終了を意味する EOM を使用して、ns.tcp.stream タイプのデータを ctxt.output に送信できます。EOM イベントには TCP データが含まれている場合と含まれていない場合があります。TCP データを含む EOM イベントを先に DATA イベントなしで送信すると、プロトコルメッセージデータ全体を送信してメッセージの終了を知らせることができます。負荷分散の決定は、最初に受信したデータに基づいて、負荷分散仮想サーバーによって下流で行われます。EOM メッセージの受信後に、新しい負荷分散の決定が行われます。そのため、プロトコルメッセージデータをストリーミングするには、最後のイベントを EOM として複数の DATA イベントを送信します。連続するすべての DATA イベントと次の EOM イベントは、シーケンスの最初の DATA イベントの負荷分散決定によって選択された同じサーバー接続に送信されます。
- **ctxt.input** -TCP ストリームデータの送信元であるパイプライン内の以前の処理コンテキスト。
- **ctx: hold (**データ**)** -将来の処理に備えてデータを保存する関数。データで hold を呼び出すと、データはコンテキストに保存されます。その後、同じコンテキストでさらにデータを受信すると、新しく受信したデータ

が以前に保存されたデータに追加され、結合されたデータストリームが on_data コールバック関数に渡されます。保留を呼び出すと、データ参照は使用できなくなり、どの使用でもエラーになります。

- **ctxt.vserver-**仮想サーバーのコンテキスト。
- **ctxt.client** —クライアント接続処理コンテキスト。この処理コンテキストを使用して、クライアントにデータを送信したり、IP アドレス、送信元、宛先ポートなどの接続関連情報を取得したりできます。
- **ctx: close ()** —FIN をクライアントに送信してクライアント接続を閉じます。この API を呼び出すと、クライアント処理コンテキストは使用できなくなり、どのような使用でもエラーが発生します。

TCP サーバーコンテキスト

TCP サーバーのイベントコールバックに渡されるコンテキスト:

- **ctxt.output** —パイプライン内の次の処理コンテキスト。拡張コールバックハンドラーは、イベント DATA（メッセージの一部）またはプロトコルメッセージの終了を意味する EOM を使用して、ns.tcp.stream タイプのデータを ctxt.output に送信できます。
- **ctxt.input** -TCP ストリームデータの送信元であるパイプライン内の以前の処理コンテキスト。
- **ctx: hold (**データ**)** -将来の処理に備えてデータを保存する関数。データで hold を呼び出すと、データはコンテキストに保存されます。その後、同じコンテキストでさらにデータを受信すると、新しく受信したデータが以前に保存されたデータに追加され、結合されたデータストリームが on_data コールバック関数に渡されます。保留を呼び出すと、データ参照は使用できなくなり、どの使用でもエラーになります。
- **ctxt.vserver-**仮想サーバーのコンテキスト。
- **ctxt.server-**サーバー接続処理コンテキスト。この処理コンテキストを使用して、データをサーバーに送信したり、IP アドレス、送信元ポート、宛先ポートなどの接続関連情報を取得したりできます。
- **ctx: reuse_server_connection ()** -この API を使用すると、サーバー接続をサーバーコンテキスト内の他のクライアント接続にのみ再利用できます。この API は、(ns.send () API で) EOM イベントを使用してクライアントコンテキストでデータを送信する場合にのみ使用できます。そうしないと、ADC アプライアンスはエラーを投げます。

  サーバー接続を他のクライアントが再利用できるようにするには、各応答メッセージの最後にこの API を呼び出す必要があります。この API を呼び出した後、このサーバー接続でさらにデータを受信した場合、これはエラーとして扱われ、サーバー接続は閉じられます。この API を使用しない場合、サーバー接続は、その接続が開かれたクライアントでのみ使用できます。また、そのクライアントに対して別の負荷分散を決定するために同じサーバーを選択した場合、同じサーバー接続を使用してクライアントデータを送信します。この API を使用すると、サーバー接続は開かれたクライアント接続に関連付けられなくなり、他のクライアント接続の新しい負荷分散の決定に再利用できます。この API を呼び出すと、サーバーコンテキストは使用できなくなり、使用するとエラーが発生します。

  注：この API は NetScaler 12.1 ビルド 49.xx 以降で使用できます。

- **ctx: close ()** —FIN をサーバーに送信してサーバー接続を閉じます。この API を呼び出すと、クライアント処理コンテキストは使用できなくなり、どのような使用でもエラーが表示されます。

  注：この API は、NetScaler 12.1 ビルド 50.xx 以降で使用できます。

仮想サーバーコンテキスト

コールバックに渡されるコンテキストを介して利用可能なユーザー仮想サーバーコンテキスト:

- **vserver: counter_increment (counter_name)** -引数として渡された仮想サーバーカウンターの値をインクリメントします。現在、以下の組み込みカウンターがサポートされています。
  - **-invalid_messages** —この仮想サーバー上の無効なリクエスト/レスポンスの数。
  - **-invalid_messages_dropped** —この仮想サーバーによってドロップされた無効なリクエスト/レスポンスの数。
- **vserver.params-**ユーザー仮想サーバーに設定されたパラメーター。パラメータは拡張の設定を可能にします。拡張コードは、CLI で指定されたパラメータにアクセスして、ユーザー仮想サーバーを追加できます。

クライアント接続コンテキスト

接続関連情報を取得するためのクライアント接続処理コンテキスト。

- クライアント**.ssl** —**SSL** コンテキスト
- クライアント**.tcp** —**TCP** コンテキスト
- **client.is_ssl** —クライアント接続が **SSL** ベースの場合は True

サーバー接続コンテキスト

接続関連情報を取得するためのサーバー接続処理コンテキスト。

- **server.ssl** –SSL context
- **server.tcp** –TCP context
- **server.is_ssl** —サーバー接続が **SSL** ベースの場合は True

TCP コンテキスト

TCP コンテキストは TCP プロトコルで動作します。

- **tcp.srcport** —送信元ポートを数値で表したもの
- **tcp.dstport-**宛先ポートを数値で表したもの

IP コンテキスト

IP コンテキストは IP または IPv6 プロトコルデータで機能します。

- **ip.src** -送信元 IP アドレスコンテキスト。
- **ip.dst** -宛先 IP アドレスコンテキスト。

注：この API は NetScaler 12.1 ビルド 51.xx 以降で使用できます。

IP アドレスコンテキスト

IP アドレスコンテキストは IP または IPv6 アドレスデータで機能します。

- **<address>.to_s** -適切な ASCII 表記のアドレス文字列。
- **<address>.to_n** -ネットワーク順のバイト列で表したアドレスの数値 (IPv4 は 4 バイト、IPv6 は 16 バイト)。
- **<address>.version** -IPv4 の場合は 4、IPv6 の場合は 6 を返します。
- **<address>:subnet(<prefix value>)** -プレフィックス番号を適用した後のサブネットアドレス文字列を返します。
  - IPv4 アドレスの場合、値は 0 から 32 の間でなければなりません
  - IPv6 アドレスの場合、値は 0 から 128 の間でなければなりません。
- **<address>:apply_mask(<mask string>)** -マスク文字列を適用した後のアドレス文字列を返します。API は引数のバージョンを検証し、適切なエラーチェックを行います。
- **address>:eq(<address string>)** -引数がアドレスオブジェクトと等しいかどうかに基づいて true または false を返します。API は引数のバージョンを検証します。

注：この API は NetScaler 12.1 ビルド 51.xx 以降で使用できます。

SSL コンテキスト

SSL コンテキストは、フロントエンド SSL 接続に関連する情報を提供します。

- **ssl.cert** —SSL 証明書コンテキスト。クライアント接続の場合はクライアント証明書コンテキストを提供し、サーバー接続の場合はサーバー証明書コンテキストを提供します。
- **ssl.version** -現在のトランザクションの SSL プロトコルバージョンを表す数値。次のようになります。

```
- - 0: The transaction is not SSL-based
- - 0x002: The transaction is SSLv2
- - 0x300: The transaction is SSLv3
- - 0x301: The transaction is TLSv1
- - 0x302: The transaction is TLSv1.1
```



```
- - 0x303: The transaction is TLSv1.2
```

- **ssl.cipher_name** -SSL 接続から呼び出された場合は SSL 暗号名を文字列とし、それ以外の場合は NULL 文字列を返します。
- **ssl.cipher_bits** —暗号化キーのビット数。

SSL 証明書コンテキスト

- 証明書バージョン—証明書のバージョン番号。接続が SSL ベースでない場合は、0 を返します。
- **Cert.Valid_not_Before** —文字列形式の日付で、それより前の証明書は無効です。
- **Cert.valid_not_After** —文字列形式の日付。この日付を過ぎると証明書は無効になります。
- **Cert.days_to_Expire** —証明書が有効になるまでの日数。期限切れの証明書の場合は -1 を返します。
- **cert.to_PEM** —バイナリ形式の証明書。
- **cert.issuer** -証明書内の発行者の識別名 (DN) を名前/値リストとして表示します。等号 (「=」) は名前と値の区切り文字で、スラッシュ (「/」) は名前と値のペアを区切る区切り文字です。

  返される DN の例を次に示します。
  /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@m
- **cert.auth_keyid** —X.509 V3 証明書のオーソリティキー識別子エクステンションのコンテキスト。
  - **auth_keyid.exists** -証明書にオーソリティキー識別子の拡張子が含まれている場合は TRUE。
  - **auth_keyid.issuer_name-**証明書内の発行者識別名を名前**/**値リストとして指定します。
    等号 (「=」) は名前と値の区切り文字で、スラッシュ (「/」) は名前と値のペアを区切る区切り文字です。

  以下は例です:
  /C =us/o=mycompany/ou=www.mycompany.com/cn=www.mycompany.com/emailAddress=
  myuserid@mycompany.com
  - **auth_keyid.keyid-**ブロブとしてのオーソリティキー識別子のキー識別子フィールド
  - **auth_keyid.cert_serialnumber-**ブロブ形式のオーソリティキー識別子のシリアル番号フィールド。
- **cert.pk_algorithm** -証明書で使用される公開鍵アルゴリズムの名前。
- **cert.pk_size** -証明書で使用されるパブリックキーのサイズ。
- **cert.** シリアル番号**-**クライアント証明書のシリアル番号。これが SSL 以外のトランザクションであるか、証明書にエラーがある場合は、空の文字列が返されます。
- **cert.signature_algorithm** -CA がこの証明書に署名するために使用する暗号化アルゴリズムの名前。

- **cert.subject_keyid-**クライアント証明書のサブジェクトキー **ID** 。Subject KeyID がない場合は、長さ 0 のテキストオブジェクトになります。
- **cert.subject-**サブジェクトの識別名を名前**/**値として指定します。等号 (「=」) は名前と値を区切り、スラッシュ (「/」) は名前と値のペアを区切ります。

以下は例です:
/C =us/o=mycompany/ou=www.mycompany.com/cn=www.mycompany.com/emailAddress=myuserid@mycompany.com

NetScaler ライブラリ

- **ns.tcp.stream** -TCP データをバイトストリームとして処理するための文字列のようなライブラリ。これらの API が動作できる TCP ストリームデータの最大サイズは 128 KB です。ns.tcp.stream ライブラリ関数は、通常の拡張オブジェクト指向の呼び出し方法で呼び出すこともできます。たとえば、data: len () は ns.tcp.stream.len (データ) と同じです
  - **ns.tcp.stream.len (**データ**)** -Lua の文字列.len と同様に、データの長さをバイト単位で返します
  - **ns.tcp.stream.find (data, pattern [, init])-Lua** の文字列.find に似た関数さらに、データの最後で部分的なマッチングも行います。部分的に一致すると、開始インデックスが返され、終了インデックスは nil になります。
  - **ns.tcp.stream.split (**データ、長さ**)-**データを 2 つのチャンクに分割します。最初のチャンクは指定された長さになります。分割が成功すると、元のデータは TCP データストリームとして使用できなくなります。その方法で使用しようとするとエラーが発生します。
  - **ns.tcp.stream.byte (data [, i [, j]])-Lua** の文字列**.byte** に似た関数。データ [i]、データ [i+1]、…、データ [j] の内部数値コードを返します。
  - n**s.tcp.stream.sub (data, i [, j])-Lua** の string.sub に似た関数。i から始まり j まで続く s の部分文字列を返します。
  - **ns.tcp.stream.match (**データ、パターン、**[, init])-Lua** の文字列マッチに似た関数。文字列 s のパターンに最初に一致するものを探します。
- **ns.send (processing_ctxt**、**event_name**、**event_data**)-処理コンテキストにイベントを送信する汎用関数。イベントデータは、どのような内容でも含めることができる Lua テーブルです。内容はイベントによって異なります。ns.send () API が呼び出されると、データ参照は使用できなくなります。これを使用しようとするとエラーが発生します。
- **ns.pipe (src_ctxt**、**dest_ctxt)-pipe ()** API の呼び出しを使用して、拡張コードはソースコンテキストを宛先コンテキストに接続できます。パイプを呼び出すと、ソースコンテキストからパイプライン内の次のモジュールに送信されるすべてのイベントは、宛先コンテキストに直接送信されます。この API は通常、pipe () 呼び出しを行うモジュールがパイプラインから自身を削除するために使用されます。
- **ns.inet** —インターネットアドレス用ライブラリ。

- **ns.inet.apply_mask (address_str, mask_str)-**マスク文字列を適用した後のアドレス文字列を返します。
- **ns.inet.aton (address_str)** -アドレスの数値をネットワーク順のバイト列として返します (IPv4 の場合は 4 バイト、IPv6 の場合は 16 バイト)。
- **ns.inet.ntoa (byte_str)-**数値バイト値をバイト文字列としてアドレス文字列に変換します。
- **ns.inet.ntohs (**数値**)** -指定されたネットワークのバイトオーダーをホストのバイトオーダーに変換します。入力が 2^16-1 より大きい場合、エラーが発生します。
- **ns.inet.htons (**数値**)** -指定されたホストのバイトオーダーをネットワークのバイトオーダーに変換します。入力が 2^16-1 より大きい場合、エラーが発生します。
- **ns.inet.ntohl (**数値**)** -指定されたネットワークバイトオーダーをホストのバイトオーダーに変換します。入力が 2^32-1 より大きい場合、エラーが発生します。
- **ns.inet.htonl (**数値**)** -指定されたホストのバイトオーダーをネットワークのバイトオーダーに変換します。入力が 2^32-1 より大きい場合、エラーが発生します。
- **ns.inet.subnet (address_str**、**subnet_value)** —指定されたサブネットを適用した後のサブネットアドレス文字列を返します。

プロトコル拡張

August 15, 2023

NetScaler アプライアンスは、HTTP などのプロトコルをネイティブにサポートしています。これに加えて、プロトコル拡張を使用してカスタムプロトコルのサポートを追加することもできます。現在、メッセージキューテレメトリトランスポート (MQTT) プロトコルなど、TCP ベースのカスタムプロトコルのみがサポートされています。安全なトランザクションを実現するために、SSL 経由の TCP もサポートされています。

NetScaler アプライアンスのプロトコル拡張は、NetScaler アプライアンスで利用できる高レベルのスクリプトインフラストラクチャの一部です。スクリプト言語は Lua 5.2 プログラミング言語に基づいています。NetScaler アプライアンスにカスタムプロトコルを追加するには、ユーザーは該当する動作を実装する拡張コードを記述する必要があります。たとえば、ns.tcp.client と ns.tcp.server の動作は TCP ベースのプロトコルにも適用できます。動作を実装するには、カスタマイズするコールバックのみを実装します。コールバックが実装されていない場合、そのデフォルトが有効になります。スクリプト言語の詳細については、NetScaler ADC 拡張機能-言語の概要を参照してください。動作の詳細については、NetScaler ADC 拡張機能の API リファレンスを参照してください。

NetScaler ADC プロトコル拡張は、次の用途に使用できます。

- 拡張機能を使用して、新しいプロトコルサポートを NetScaler アプライアンスにプログラムで追加します。
- プロトコルトラフィックを解析し、プロトコル固有のメッセージベースのロードバランシング (MBLB) を行います。
- ユーザ定義のロードバランシングの永続性を設定します。

プロトコル拡張 - アーキテクチャ

August 15, 2023

トラフィックレベルの拡張性を実現するために、NetScaler アプライアンスでのトラフィック処理は、個別の処理モジュールのパイプラインとして公開されます。トラフィックは、入力から出力へと処理されるときにそれらを通過します。パイプラインのこれらのモジュールは、何も共有しないモデルを採用しています。メッセージパッシングは、パイプライン内の 1 つのモジュールから次のモジュールにトラフィックデータを送信するために使用されます。

トラフィック処理パイプラインの特定のポイントは拡張可能になっているため、コードを追加して NetScaler の動作をカスタマイズできます。

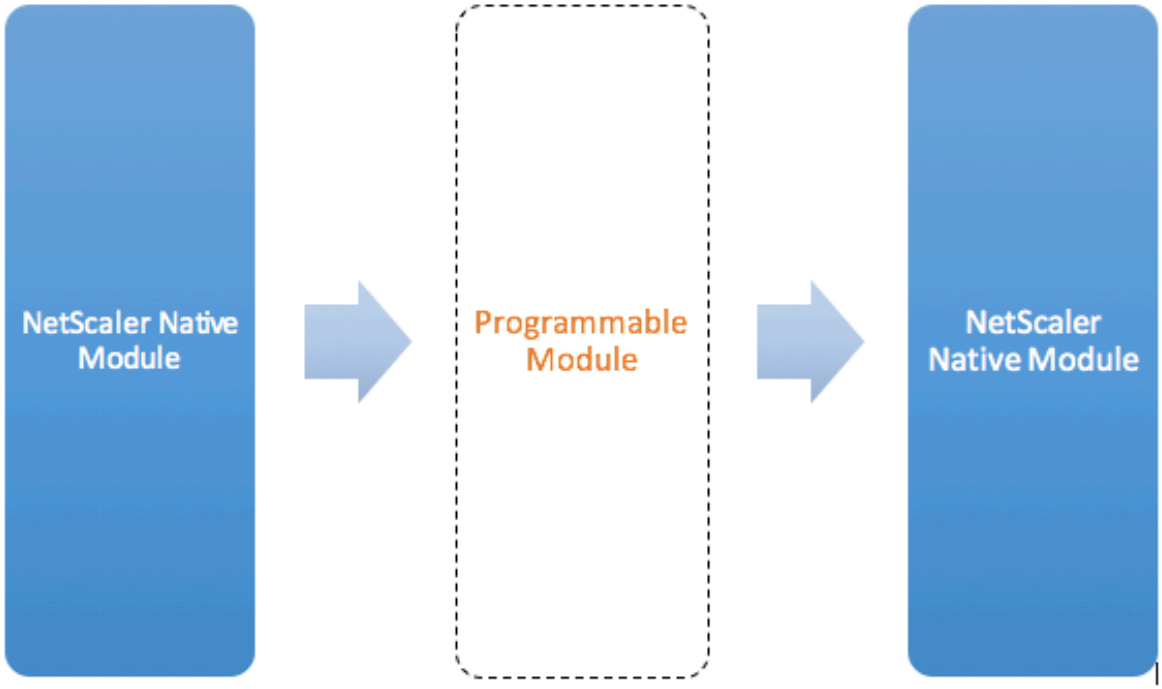


Figure: A Programmable Module In the Traffic Pipeline

デフォルトでは、トラフィックはコードを追加しないプログラマブルモジュールをバイパスします。

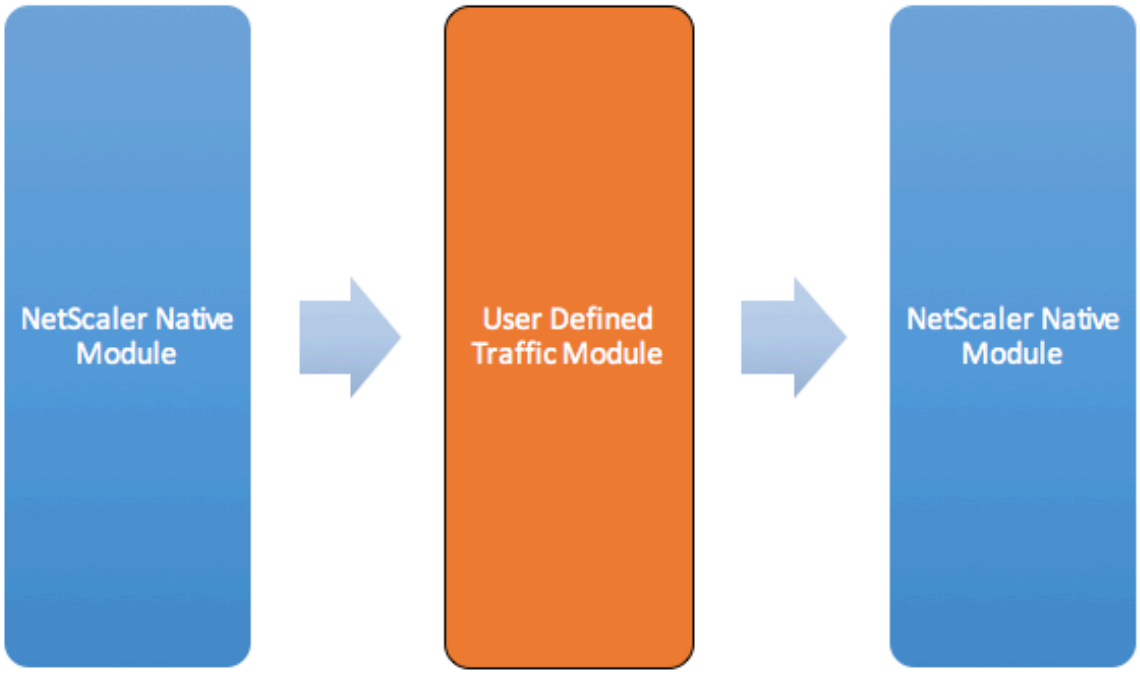


Figure: User Defined Traffic Module

行動

トラフィック処理をカスタマイズするためのプログラム可能なインターフェースは、ビヘイビアと呼ばれます。動作は基本的に、NetScaler アプライアンスで使用できる一般的なプログラム可能なパターンを形式化したものです。動作は、事前に定義された一連のイベントコールバック関数で構成されています。動作に対応するコールバック関数を提供することで、動作を実装できます。

たとえば、TCP クライアントの動作は、TCP クライアントデータストリームイベントを処理するコールバック関数 (on_data) で構成されています。TCP ベースのプロトコルのメッセージベース負荷分散 (MBLB) を実装するには、このコールバック関数のコードを追加して、クライアントからの TCP データストリームを処理し、バイトストリームを解析してプロトコルメッセージにします。

コンテキスト**:**

ビヘイビアのコールバック関数は、処理モジュールの状態であるコンテキストとともに呼び出されます。コンテキストは、処理モジュールのインスタンスです。たとえば、TCP クライアント動作コールバックは、クライアント TCP 接続ごとに異なるコンテキストで呼び出されます。

ペイロード**:**

コンテキストに加えて、ビヘイビアコールバックは他の引数を持つことができます。通常、残りの引数はペイロードとして渡されます。ペイロードはすべての引数のコレクションです。

したがって、プログラマブルプロセッシングモジュールのインスタンスは、インスタンス状態とイベントコールバック関数の組み合わせ、つまりコンテキストと動作の組み合わせと見なすことができます。トラフィックはイベントペイロードとしてパイプラインを通過します。

NetScaler API 拡張機能については、NetScaler ADC 拡張 API リファレンスを参照してください。

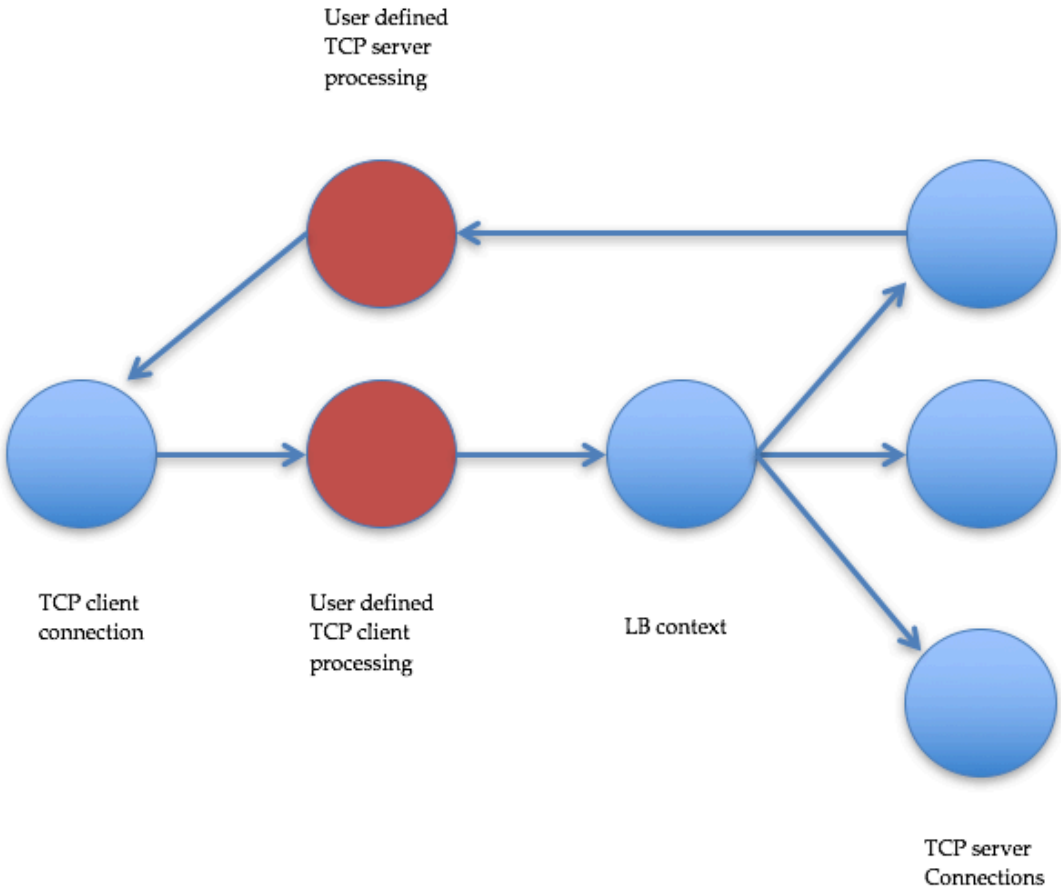
次のコードスニペットは、TCP クライアントデータストリームイベントを処理するユーザー定義関数を示しています。コンテキストとペイロードは NetScaler コードによって関数に渡されます。このコードは、すべての呼び出しで受信した TCP データを、パイプライン内の次の処理モジュールコンテキストに転送するだけです。この場合、次のモジュールは負荷分散（LB）コンテキストであり、これは NetScaler ADC ネイティブモジュールです。

```
function client.on_data(ctxt, payload)
    ns.send(ctxt.output, "DATA", {
 data = payload.data }
 )
end
<!--NeedCopy-->
```

プロトコル拡張 - ユーザー定義 TCP クライアントとサーバーの動作のトラフィックパイプライン

August 15, 2023

次の図は、サンプルプロトコル拡張（ユーザー定義の TCP クライアントとサーバーの動作用のトラフィックパイプライン）を示しています。



Traffic Pipeline For User Defined TCP Client And Server Behaviors

プロトコル拡張を使用してカスタムプロトコルを追加する

カスタムプロトコルのコマンドラインインターフェイス (CLI) コマンドでは、「user」というキーワードを使用して、基礎となる構成エンティティがユーザー定義であることを表します。拡張コードを使用すると、システムに新しいユーザープロトコルを追加したり、ユーザー定義プロトコル用のユーザー仮想サーバーを追加したりできます。ユーザー仮想サーバーは、パラメーターを設定することで順番に設定できます。仮想サーバーパラメーターの設定値は、拡張コードで確認できます。

次の例は、新しいプロトコルのサポートを追加するためのユーザーフローを示しています。この例では、MQTT プロトコルのサポートをシステムに追加しています。MQTT はマシン間の「モノのインターネット」接続プロトコルです。これは軽量なパブリッシュ/サブスクライブメッセージングトランスポートです。遠隔地との接続に便利なこのプロトコルは、クライアントとブローカーのツールを使用してメッセージをサブスクライバーに公開します。

1. MQTT プロトコル拡張実装ファイルを NetScaler システムにインポートします。mqtt.lua のコードリストは以下の通りです。以下の例では、Web サーバーでホストされている MQTT 拡張ファイルをインポートしま

す。

```
import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
```

2. 拡張機能を使用して、新しいユーザー TCP ベースのプロトコルをシステムに追加します。

```
add user protocol MQTT -transport TCP -extension mqtt_code
```

3. ユーザー負荷分散仮想サーバーを追加し、それにバックエンドサービスをバインドします。

```
add service mqtt_svr1 10.217.24.48 USER_TCP 1501
add service mqtt_svr2 10.217.24.48 USER_TCP 1502
add lb vserver mqtt_lb USER_TCP  - lbmethod USER_TOKEN
bind lb vserver mqtt_lb mqtt_svr1
bind lb vserver mqtt_lb mqtt_svr2
<!--NeedCopy-->
```

4. 新しく追加されたプロトコル用のユーザー仮想サーバーを追加します。defaultlb を上記で設定した LB 仮想サーバーに設定します。

```
add user vserver mqtt_vs MQTT 10.217.24.28 8765 -defaultLb
mqtt_lb
```

5. オプションで、ClientID に基づいて MQTT セッションパーシステンスを有効にし、パーシステンスタイプを USERSESSION に設定します。

```
set lb vserver mqtt_lb -persistenceType USERSESSION
```

プロトコル拡張 - ユースケース

August 15, 2023

プロトコル拡張は、次のユースケースに使用できます。

- メッセージベースの負荷分散 (MBLB)
- ストリーミング
- トークンベースの負荷分散
- ロード・バランシング・パーシステンス
- TCP 接続ベースのロードバランシング
- コンテンツベースの負荷分散
- SSL
- トラフィックを変更
- クライアントまたはサーバーへのトラフィックの発信
- 接続確立時のデータ処理

メッセージベースの負荷分散

プロトコル拡張はメッセージベースの負荷分散（MBLB）をサポートしており、NetScaler アプライアンス上の任意のプロトコルを解析し、1 つのクライアント接続で届くプロトコルメッセージの負荷分散、つまりメッセージを複数のサーバー接続に分散できます。MBLB は、クライアントの TCP データストリームを解析するユーザーコードによって実現されます。

TCP データストリームは、クライアントとサーバーの動作の on_data コールバックに渡されます。TCP データストリームは、Lua 文字列のようなインターフェイスを介して拡張機能で使用できます。Lua 文字列 API に似た API を使用して TCP データストリームを解析できます。

便利な API には以下が含まれます。

```
data:len()
```

```
data:find()
```

```
data:byte()
```

```
data:sub()
```

```
data:split()
```

TCP データストリームが解析されてプロトコルメッセージになったら、ユーザーコードは、クライアントの on_data コールバックに渡されたコンテキストから利用可能な次のコンテキストにプロトコルメッセージを送信するだけで、負荷分散を実現します。

ns.send () API は、他の処理モジュールにメッセージを送信するために使用されます。送信先コンテキストに加えて、送信 API はイベント名とオプションのペイロードを引数として取ります。イベント名と動作のコールバック関数名は 1 対 1 で対応しています。<event_name> イベントのコールバックは on_ と呼ばれます。コールバック名は小文字のみを使用します。

たとえば、TCP クライアントとサーバーの on_data コールバックは、「DATA」という名前のイベントのユーザー定義ハンドラーです。1 回の送信呼び出しでプロトコルメッセージ全体を送信するには、EOM イベントが使用されます。EOM は end of message の略で、LB コンテキストのダウンストリームへのプロトコルメッセージの終了を意味します。そのため、このメッセージに続くデータについては、新たに負荷分散の決定が行われます。

拡張コードが on_data イベントでプロトコルメッセージ全体を受け取らないことがあります。このような場合、ctx:hold () API を使用してデータを保持できます。hold API は、TCP クライアントコンテキストとサーバコールバックコンテキストの両方で使用できます。「hold with data」が呼び出されると、データはコンテキストに保存されます。同じコンテキストでさらにデータを受信すると、新しく受信したデータが以前に保存されたデータに追加され、結合されたデータを使用して on_data コールバック関数が再度呼び出されます。

注: 使用される負荷分散方法は、負荷分散コンテキストに対応する負荷分散仮想サーバーの構成によって異なります。

次のコードスニペットは、send API を使用して解析されたプロトコルメッセージを送信する方法を示しています。

例:


```
    function client.on_data(ctxt, payload)
        --
        -- code to parse payload.data into protocol message comes here
        --
        -- sending the message to lb
        ns.send(ctxt.output, "EOM", {
data = message }
)
    end -- client.on_data

    function server.on_data(ctxt, payload)
        --
        -- code to parse payload.data into protocol message comes here
        --
        -- sending the message to client
        ns.send(ctxt.output, "EOM", {
data = message }
)

    end -- server.on_data
<!--NeedCopy-->
```

ストリーミング

シナリオによっては、プロトコルメッセージ全体が収集されるまで TCP データストリームを保持する必要がない場合があります。実際、必要な場合以外はお勧めしません。データを保持すると、NetScaler アプライアンスのメモリ使用量が増加し、多くの接続で不完全なプロトコルメッセージで NetScaler アプライアンスのメモリが使い果たされ、アプライアンスが DDoS 攻撃を受けやすくなります。

ユーザーは、send API を使用して拡張コールバックハンドラーで TCP データのストリーミングを実現できます。メッセージ全体が収集されるまでデータを保持する代わりに、データをまとめて送信できます。DATA イベントを使用して ctxt.output にデータを送信すると、プロトコルメッセージの一部が送信されます。その後にさらに多くの DATA イベントが続くことがあります。プロトコルメッセージの終了を知らせるには、EOM イベントを送信する必要があります。下流の負荷分散コンテキストは、最初に受信したデータに対して負荷分散の決定を行います。EOM メッセージの受信後に、新しい負荷分散の決定が行われます。

プロトコルメッセージデータをストリーミングするには、複数の DATA イベントの後に EOM イベントを送信します。連続した DATA イベントと次の EOM イベントは、シーケンスの最初の DATA イベントの負荷分散決定によって選択された同じサーバー接続に送信されます。

クライアントへの送信コンテキストでは、下流のクライアントコンテキストによる EOM イベントの特別な処理がないため、EOM イベントと DATA イベントは実質的に同じです。

トークンベースの負荷分散

ネイティブにサポートされているプロトコルについては、NetScaler アプライアンスは PI 式を使用してトークンを作成するトークンベースの負荷分散方法をサポートしています。拡張機能の場合、プロトコルが事前にわからないため、PI 表現は使用できません。トークンベースの負荷分散では、USER_TOKEN 負荷分散メソッドを使用するようにデフォルトの負荷分散仮想サーバーを設定し、user_token フィールドを指定して send API を呼び出して拡張コードからトークン値を指定する必要があります。トークン値が送信 API から送信され、USER_TOKEN 負荷分散方法がデフォルトの負荷分散仮想サーバーで構成されている場合、負荷分散の決定はトークン値に基づいてハッシュを計算することによって行われます。トークン値の最大長は 64 バイトです。

```
add lb vserver v\\_mqttlb USER\\_TCP -lbMethod USER\\_TOKEN
```

次の例のコードスニペットは、送信 API を使用して LB トークン値を送信します。

例:

```
        -- send the message to lb



        -- user_token is set to do LB based on clientID



        ns.send(ctxt.output, "EOM", {
 data = message,
                                  user_token = token_info }
 )
<!--NeedCopy-->
```

ロード・バランシング・パーシステンス

負荷分散の永続性は、トークンベースの負荷分散と密接に関係しています。ユーザーは、永続性セッションの値をプログラムで計算し、それを負荷分散の持続性に使用できなければなりません。send API はパーシスタンスパラメータの送信に使用されます。負荷分散永続性を使用するには、デフォルトの負荷分散仮想サーバーで USERSESSION 永続性タイプを設定し、user_session フィールドを指定して send API を呼び出して拡張コードから永続性パラメーターを指定する必要があります。パーシスタンスパラメータ値の最大長は 64 バイトです。

カスタムプロトコルに複数のタイプのパーシステンスが必要な場合は、ユーザーパーシステンスタイプを定義して設定する必要があります。仮想サーバーの構成に使用されるパラメータの名前は、プロトコル実装者によって決定されます。パラメータの設定値は、拡張コードでも使用できます。

次の CLI とコードスニペットは、ロードバランシングの永続性をサポートする送信 API の使用方法を示しています。mqtt.lua のコードリストのセクションにあるコードリストでは 、user_session フィールドの使用方法も示してい

ます。

永続性については、負荷分散仮想サーバーで USERSESSION 永続性タイプを指定し、ns.send API から user_session 値を渡す必要があります。

```
add lb vserver v\\_mqttlb USER\\_TCP -persistencetype USERSESSION
```

ペイロードの user_session フィールドを ClientID に設定して、MQTT メッセージをロードバランサーに送信します。

例:

```
-- send the data so far to lb

-- user_session is set to clientID as well (it will be used to persist
   session)

ns.send(ctxt.output,  “DATA” , {
 data = data, user_session = clientID }
 )
<!--NeedCopy-->
```

TCP 接続ベースのロードバランシング

一部のプロトコルでは、MBLB が必要ない場合があります。代わりに、TCP 接続ベースの負荷分散が必要な場合があります。たとえば、MQTT プロトコルは TCP ストリームの最初の部分を解析して、負荷分散用のトークンを決定する必要があります。また、同じ TCP 接続上のすべての MQTT メッセージは、同じサーバー接続に送信する必要があります。

TCP 接続ベースの負荷分散は、DATA イベントのみで EOM を送信せずに send API を使用することで実現できます。これにより、ダウンストリームのロードバランシングコンテキストは、最初に受信したデータに基づいてロードバランシングを決定し、後続のすべてのデータをロードバランシング決定によって選択された同じサーバー接続に送信します。

また、ユースケースによっては、負荷分散の決定後に拡張処理をバイパスする機能が必要になる場合があります。拡張呼び出しをバイパスすると、トラフィックがネイティブコードだけで処理されるため、パフォーマンスが向上します。バイパスは ns.pipe () API を使用して実行できます。pipe () API 拡張コードを呼び出すと、入力コンテキストを出力コンテキストに接続できます。pipe () の呼び出し後、入力コンテキストからのすべてのイベントは直接出力コンテキストに送られます。事実上、pipe () 呼び出しが行われたモジュールはパイプラインから削除されます。

次のコードスニペットは、ストリーミングと pipe () API を使用してモジュールをバイパスする方法を示しています。mqtt.lua のコードリストのセクションにあるコードリストは、ストリーミングを行う方法と pipe () API を使用して接続上の残りのトラフィックに対してモジュールをバイパスする方法も示しています。

例:

```
        -- send the data so far to lb
        ns.send(ctxt.output, "DATA", {
```



```
 data = data,
                                        user_token = clientID }
 )
        -- pipe the subsequent traffic to the lb - to bypass the client
            on_data handler
        ns.pipe(ctxt.input, ctxt.output)
<!--NeedCopy-->
```

コンテンツベースの負荷分散

ネイティブプロトコルでは、プロトコル拡張用のコンテンツスイッチング機能のような機能がサポートされています。この機能を使用すると、データをデフォルトのロードバランサーに送信する代わりに、選択したロードバランサーにデータを送信できます。

プロトコル拡張用のコンテンツスイッチング機能は、ctxt: lb_connect () API を使用することで実現されます。<lbname> この API は TCP クライアントコンテキストで使用できます。この API を使用すると、拡張コードは、すでに構成されている負荷分散仮想サーバーに対応する負荷分散コンテキストを取得できます。その後、取得した負荷分散コンテキストで送信 API を使用できます。

lb コンテキストは NULL になることがあります。

- 仮想サーバーは存在しません
- 仮想サーバーはユーザープロトコルタイプではありません
- 仮想サーバーの状態は UP ではありません
- 仮想サーバーはユーザー仮想サーバーであり、負荷分散仮想サーバーではありません

使用中にターゲットの負荷分散仮想サーバーを削除すると、その負荷分散仮想サーバーに関連するすべての接続がリセットされます。

次のコードスニペットは、lb_connect () API の使用方法を示しています。このコードは、Lua テーブル lb_map を使用してクライアント ID を負荷分散仮想サーバー名 (lbname) にマッピングし、次に lb_connect () を使用して lbname の LB コンテキストを取得します。そして最後に send API を使用して LB コンテキストに送信します。

```
    local lb_map = {

       ["client1*"] = "lb_1",
       ["client2*"] = "lb_2",
       ["client3*"] = "lb_3",
       ["client4*"] = "lb_4"
     }


    -- map the clientID to the corresponding LB vserver and connect to
       it
    for client_pattern, lbname in pairs(lb_map) do
       local match_idx = string.find(clientID, client_pattern)
       if (match_idx == 1) then
      lb_ctxt = ctxt:lb_connect(lbname)
```



```
      if (lb_ctxt == nil) then
         error("Failed to connect to LB vserver: " .. lbname)
      end
      break
       end
    end
    if (lb_ctxt == nil) then
    -- If lb context is NULL, the user can raise an error or send data
       to default LB
       error("Failed to map LB vserver for client: " .. clientID)
    end
-- send the data so far to lb
ns.send(lb_ctxt, "DATA", {
 data = data }

<!--NeedCopy-->
```

SSL

拡張機能を使用するプロトコルの SSL は、ネイティブプロトコルの SSL のサポートと同様の方法でサポートされます。同じ解析コードを使用してカスタムプロトコルを作成すると、TCP または SSL を介してプロトコルインスタンスを作成し、それを使用して仮想サーバーを構成できます。同様に、TCP または SSL 経由でユーザーサービスを追加できます。

詳細については、MQTT の SSL オフロードの設定およびエンドツーエンド暗号化を使用したMQTT の SSL オフロードの設定を参照してください。

サーバー接続の多重化

クライアントは一度に 1 つの要求を送信し、最初の要求に対する応答がサーバーから受信された後にのみ次の要求を送信することがあります。このような場合、サーバー接続を他のクライアント接続に再利用できます。また、応答がクライアントに送信された後の同じ接続上の次のメッセージにも再利用できます。他のクライアント接続によるサーバー接続の再利用を許可するには、サーバー側コンテキストで ctxt: reuse_server_connection () API を使用する必要があります。

注：この API は NetScaler 12.1 ビルド 49.xx 以降で使用できます。

トラフィックを変更

リクエストまたはレスポンス内のデータを変更するには、高度なポリシー PI 表現を使用するネイティブの書き換え機能を使用する必要があります。拡張で PI 式を使用できないため、次の API を使用して TCP ストリームデータを変更できます。

```
data:replace(offset, length, new_string)
data:insert(offset, new_string)
```



```
data:delete(offset, length)
data:gsub(pattern, replace [,n]))
```

次のコードスニペットは、replace () API の使用方法を示しています。

```
-- Get the offset of the pattern, we want to replace
   local old_pattern =  “pattern to repalace”
local old_pattern_length = old_pattern:len()
   local pat_off, pat_end = data:find(old_pattern)
   -- pattern is not present
if (not pat_off) then
    goto send_data
   end
  -- If the data we want to modify is not completely present, then
  -- wait for more data
  if (not pat_end) then
        ctxt:hold(data)
        data = nil
     goto done
  end
data:replace(pat_off, old_pattern_length,  “new pattern” )
::send_data::
ns.send(ctxt.output,  “EOM” , {
 data = data }
 )
::done::
```

次のコードスニペットは、insert () API の使用方法を示しています。

```
data:insert(5,  “pattern to insert” )
```

次のコードスニペットは、いくつかのパターンの前または後に挿入したいときに、insert () API の使用を示しています。

```
-- Get the offset of the pattern, after or before which we want to
    insert
   local pattern =  “pattern after/before which we need to insert”
local pattern_length = pattern:len()
   local pat_off, pat_end = data:find(pattern)
-- pattern is not present
   if (not pat_off) then
    goto send_data
   end
  -- If the pattern after which we want to insert is not
  -- completely present, then wait for more data
  if (not pat_end) then
        ctxt:hold(data)
        data = nil
     goto done
  end
-- Insert after the pattern
data:insert(pat_end + 1,  “pattern to insert” )
   -- Insert before the pattern
```



```
data:insert(pat_off, “pattern to insert” )
::send_data::
    ns.send(ctxt.output, “EOM” , {
 data = data }
 )
::done::
```

次のコードスニペットは、delete () API の使用方法を示しています。

```
-- Get the offset of the pattern, we want to delete
   local delete_pattern = “pattern to delete”
local delete_pattern_length = delete_pattern:len()
   local pat_off, pat_end = data:find(old_pattern)
   -- pattern is not present
if (not pat_off) then
          goto send_data
   end
  -- If the data we want to delete is not completely present,
  -- then wait for more data
  if (not pat_end) then
       ctxt:hold(data)
       data = nil
    goto done
  end
data:delete(pat_off, delete_pattern_length)
::send_data::
ns.send(ctxt.output, “EOM” , {
 data = data }
 )
::done::
```

次のコードスニペットは、gsub () API の使用を示しています。

```
    -- Replace all the instances of the pattern with the new string
data:gsub( “old pattern” , “new string” )
-- Replace only 2 instances of “old pattern”
data:gsub( “old pattern” , “new string” , 2)
-- Insert new_string before all instances of “http”
data:gsub( “input data” , “(http)” , “new_string%1” )
-- Insert new_string after all instances of “http”
data:gsub( “input data” , “(http)” , “%1new_string” )
-- Insert new_string before only 2 instances of “http”
data:gsub( “input data” , “(http)” , “new_string%1” , 2)
```

注：この API は、NetScaler 12.1 ビルド 50.xx 以降で使用できます。

クライアントまたはサーバーへのトラフィックの発信

ns.send () API を使用して、拡張コードから生成されたデータをクライアントとバックエンドサーバーに送信できます。クライアントコンテキストからクライアントと直接応答を送受信するには、ターゲットとして ctxt.client を使用する必要があります。サーバーコンテキストからバックエンドサーバーと直接応答を送受信するには、ctxt.server

をターゲットとして使用する必要があります。ペイロード内のデータは、TCP ストリームデータでも Lua 文字列でもかまいません。

接続上のトラフィック処理を停止するには、クライアントまたはサーバーコンテキストのいずれかから ctxt: close () API を使用できます。この API は、クライアント側の接続またはそれにリンクされているサーバー接続を閉じます。

ctx: close () API を呼び出すと、拡張コードはクライアントとサーバーの接続に TCP FIN パケットを送信し、この接続でクライアントまたはサーバーからさらにデータを受信すると、アプライアンスは接続をリセットします。

次のコードスニペットは、ctxt.client と ctxt の使用方法を示しています。クローズ（）API。

```
    -- If the input packet is not MQTT CONNECT type, then
-- send some error response to the client.
function client.on_data(ctxt, payload)
    local data = payload.data
    local offset = 1
    local msg_type = 0
    local error_response =  “Missing MQTT Connect packet.”
    byte = data:byte(offset)
msg_type = bit32.rshift(byte, 4)
if (msg_type ~= 1) then
-- Send the error response
   ns.send(ctxt.client,  “DATA” , {
 data = error_response }
 )
-- Since error response has been sent, so now close the connection
    ctxt:close()
end
```

次のコードスニペットは、ユーザーが通常のトラフィックフローにデータを注入できる例を示しています。

```
-- After sending request, send some log message to the server.
function client.on_data(ctxt, payload)
local data = payload.data
local log_message =  “client id :  “..data:sub(3, 7)..”  user name :  “
    data:sub(9, 15)
-- Send the request we get from the client to backend server
ns.send(ctxt.output,  “DATA” , {
 data = data }
 )
After sending the request, also send the log message
ns.send(ctxt.output,  “DATA” , {
 data = log_message”  }
 )
end
```

次のコードスニペットは、ctxt.to_server API の使用方法を示しています。

```
-- If the HTTP response status message is  “Not Found” ,
-- then send another request to the server.
function server.on_data(ctxt, payload)
    local data = payload.data
    local request  “GET /default.html HTTP/1.1\r\n\r\n” ss
```



```
6      local start, end = data:find( “Not Found” )
7      if (start) then
8      -- Send the another request to server
9          ns.send(ctxt.server, “DATA” , {
10   data = request }
11   )
12  end
```

注：この API は、NetScaler 12.1 ビルド 50.xx 以降で使用できます。

接続確立時のデータ処理

接続確立時（最終 ACK を受信したとき）にデータを送信したいユースケースがあるかもしれません。たとえば、プロキシプロトコルでは、接続確立時にクライアントの送信元と宛先の IP アドレスとポートをバックエンドサーバーに送信したい場合があります。この場合、client.init () コールバックハンドラを使用して、接続確立時にデータを送信できます。

次のコードスニペットは、client.init () コールバックの使用方法を示しています。

```
1  -- Send a request to the next processing context
2  -- on the connection establishment.
3  function client.init(ctxt)
4     local request  “PROXY TCP4”  + ctxt.client.ip.src.to_s +  “  “ +
          ctxt.client.ip.dst.to_s +  “  “ + ctxt.client.tcp.srcport +  “  “
           +     ctxt.client.tcp.dstport
5  -- Send the another request to server
6     ns.send(ctxt.output,  “DATA” , {
7   data = request }
8   )
9    end
```

注：この API は、NetScaler 13.0 ビルド xx.xx 以降で使用できます。

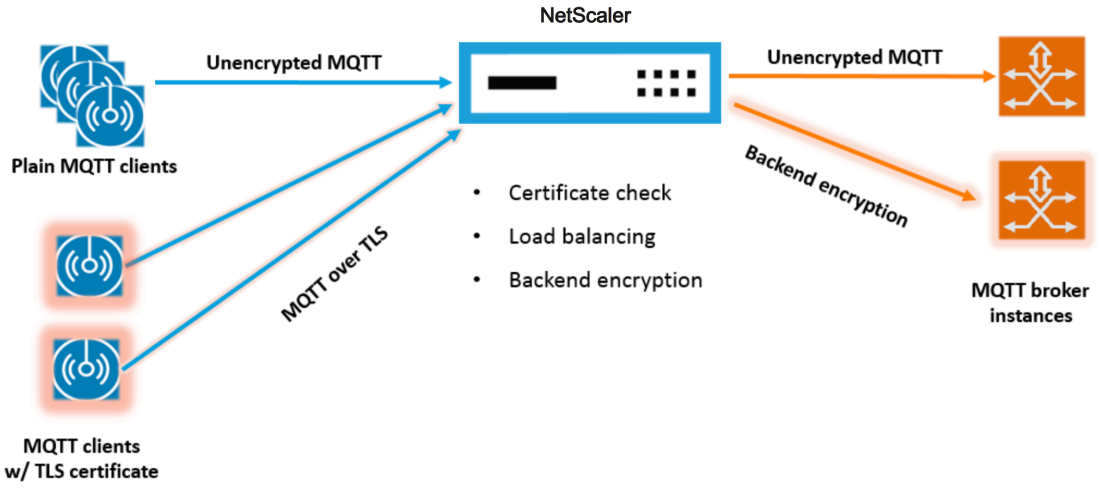
チュートリアル-プロトコル拡張を使用して **NetScaler ADC** アプライアンスに **MQTT** プロトコルを追加する

August 15, 2023

カスタムプロトコルのコマンドラインインターフェイス (CLI) コマンドでは、「user」というキーワードを使用して、基礎となる構成エンティティがユーザー定義であることを表します。拡張コードを使用すると、システムに新しいユーザープロトコルを追加したり、ユーザー定義プロトコル用のユーザー仮想サーバーを追加したりできます。ユーザー仮想サーバーは、パラメーターを設定することで順番に設定できます。仮想サーバーパラメーターの設定値は、拡張コードで確認できます。

MQTT プロトコルは説明の目的で使用されています。

次の図は、NetScaler アプライアンスと MQTT クライアントおよびブローカーツールを示しています。



mqtt.lua のコードリスト

August 15, 2023

以下のコードリスト mqtt.lua は、プロトコル拡張を使用して NetScaler に MQTT プロトコルを実装するためのコードを示しています。このコードには、TCP クライアントデータコールバック関数 client.on_data () のみが定義されています。サーバーデータの場合、コールバック関数は追加されず、サーバーからクライアントへの転送は高速なネイティブパスを取ります。クライアントデータの場合、コードは CONNECT MQTT プロトコルメッセージを解析し、クライアント ID を抽出します。次に、user_token の値に ClientID を使用します。この値を使用して、LB 仮想サーバーの LB メソッドを USER_TOKEN に設定することで、ClientID に基づいて接続のすべてのクライアントトラフィックの負荷分散を行います。user_session の値には ClientID も使用します。LB 仮想サーバーのパーシステンスタイプを USERSESSION に設定することで LB パーシステンスに使用できます。このコードは ns.send () を使用して LB を実行し、初期データを送信します。ns.pipe () API を使用して、残りのクライアントトラフィックをサーバー接続に直接送信し、拡張コールバックハンドラへの呼び出しをバイパスします。

```
--[[

  MQTT event handler for TCP client data

    ctxt - TCP client side App processing context.

    data - TCP Data stream received.

    - parse the client ID from the connect message - the first message
       should be connect

    - send the data to LB with ClientID as user token and session

```



```
13      - pipe the subsequent data to LB directly. This way the subsequent
           MQTT traffic will
14
15        bypass the tcp client on_data handler
16
17      - if a parse error is seen, throw an error so the connection is
           reset
18
19  --]]
20
21  function client.on_data(ctxt, payload)
22
23      local data = payload.data
24
25      local data_len = data:len()
26
27      local offset = 1
28
29      local byte = nil
30
31      local utf8_str_len = 0
32
33      local msg_type = 0
34
35      local multiplier = 1
36
37      local max_multiplier = 128 * 128 * 128
38
39      local rem_length = 0
40
41      local clientID = nil
42
43      -- check if MQTT fixed header is present (fixed header length is
           atleast 2 bytes)
44
45      if (data_len < 2) then
46
47          goto need_more_data
48
49      end
50
51      byte = data:byte(offset)
52
53      offset = offset + 1
54
55         -- check for connect packet - type value 1
56
57      msg_type = bit32.rshift(byte, 4)
58
59      if (msg_type ~= 1) then
60
61          error("Missing MQTT Connect packet.")
62
```



```
63      end
64
65      -- parse the remaining length
66
67      repeat
68
69          if (multiplier > max_multiplier) then
70
71             error("MQTT CONNECT packet parse error - invalid Remaining
                   Length.")
72
73          end
74
75          if (data_len < offset) then
76
77             goto need_more_data
78
79          end
80
81          byte = data:byte(offset)
82
83          offset = offset + 1
84
85          rem_length = rem_length + (bit32.band(byte, 0x7F) * multiplier)
86
87          multiplier = multiplier * 128
88
89      until (bit32.band(byte, 0x80) == 0)
90
91      -- protocol name
92
93      -- check if protocol name length is present
94
95      if (data_len < offset + 1) then
96
97          goto need_more_data
98
99      end
100
101     -- protocol name length MSB
102
103     byte = data:byte(offset)
104
105     offset = offset + 1
106
107     utf8_str_len = byte * 256
108
109     -- length LSB
110
111     byte = data:byte(offset)
112
113     offset = offset + 1
114
```



```
115      utf8_str_len = utf8_str_len + byte
116
117         -- skip the variable header for connect message
118
119      -- the four required fields (protocol name, protocol level, connect
             flags, keep alive)
120
121      offset = offset + utf8_str_len + 4
122
123      -- parse the client ID
124
125      --
126
127      -- check if client ID len is present
128
129      if (data_len < offset + 1) then
130
131         goto need_more_data
132
133      end
134
135         -- client ID length MSB
136
137      byte = data:byte(offset)
138
139      offset = offset + 1
140
141      utf8_str_len = byte * 256
142
143      -- length LSB
144
145      byte = data:byte(offset)
146
147      offset = offset + 1
148
149      utf8_str_len = utf8_str_len + byte
150
151      if (data_len < (offset + utf8_str_len - 1)) then
152
153          goto need_more_data
154
155      end
156
157      clientID = data:sub(offset, offset + utf8_str_len - 1)
158
159      -- send the data so far to lb, user_token is set to do LB based on
            clientID
160
161      -- user_session is set to clientID as well (it will be used to
            persist session)
162
163      ns.send(ctxt.output, "DATA", {
164   data = data,
```



```
165
166                                 user_token = clientID,
167
168                                 user_session = clientID }
169   )
170
171         -- pipe the subsequent traffic to the lb - to bypass the
              extension handler
172
173      ns.pipe(ctxt.input, ctxt.output)
174
175      goto parse_done
176
177      ::need_more_data::
178
179      ctxt:hold(data)
180
181      ::parse_done::
182
183      return
184
185  end
186  <!--NeedCopy-->
```

プロトコル拡張を使用して **MQTT** を構成する

August 15, 2023

次の手順では、NetScaler アプライアンスに MQTT プロトコルを追加します。

Web サーバー（HTTP を使用）またはローカルワークステーションから、拡張子ファイルを NetScaler ADC アプライアンスにインポートします。拡張ファイルのインポートの詳細については、「拡張機能のインポート」を参照してください。

```
import ns extension local:mqtt_generic_fs.lua mqtt_code
```

拡張を使用して、新しいユーザ TCP ベースのプロトコルをシステムに追加します。

```
add user protocol MQTT -transport TCP -extension mqtt_code
```

USER_TCP タイプのサービスを追加して、これがユーザー定義プロトコルであることを示します。

```
add service s1 10.102.90.112 USER_TCP 80
```

ユーザー負荷分散仮想サーバーを追加し、それにバックエンドサービスをバインドします。

```
add lb vs mysv USER_TCP

bind lb vs mysv s1
```


新しく追加したプロトコル用のユーザー仮想サーバーを追加し、前のステップで設定した負荷分散仮想サーバーをデフォルトのロードバランサーにします。

```
add user vs v_mqtt MQTT 10.217.24.28 80 -defaultlb mysv
```

オプションで、ClientID に基づいて MQTT セッションパーシステンスを有効にし、パーシステンスタイプを USERSESSION に設定します。

```
set lb vserver mqtt_lb -persistenceType USERSESSION
```

MQTT の SSL オフロードの設定

August 15, 2023

プロトコルの SSL インスタンスを追加することで、ユーザープロトコルの SSL オフロードを実装できます。以下の例は、ユーザープロトコルの SSL オフロードを行う方法を示しています。この構成では、バックエンドサービスへのトラフィックは暗号化されません。

注意: この例では、証明書とキーのペアを追加または更新し、それを仮想サーバーにバインドすることに関する詳細は提供していません。詳細については、SSL 証明書を参照してください。

次のコマンドは、mqtt.lua をトランスポート値「SSL」に含めて MQTT_SSL プロトコルを追加します。

```
import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
add user protocol MQTT_SSL -transport SSL -extension mqtt_code
<!--NeedCopy-->
```

次のコマンドは、ユーザー負荷分散仮想サーバーを追加し、それにバックエンドサービスをバインドします。

```
add service mqtt_svr1 10.217.24.48 USER_TCP 1501
add service mqtt_svr2 10.217.24.48 USER_TCP 1502
add lb vserver mqtt_lb USER_TCP  - lbMethod ROUNDROBIN
bind lb vserver mqtt_lb mqtt_svr1
bind lb vserver mqtt_lb mqtt_svr2
<!--NeedCopy-->
```

次のコマンドは、新しく追加されたプロトコル MQTT_SSL のユーザー仮想サーバーを追加します。MQTT_SSL は SSL トランスポートで構成されているため、MQTT_SSL を使用すると、NetScaler アプライアンスが SSL オフロードを実行することになります。また、このコマンドは defaultlb を前のステップで設定した負荷分散仮想サーバーに設定します。

```
add user vserver mqtt_vs MQTT_SSL 10.217.24.28 8765 -defaultLb
mqtt_lb
```

SSL オフロードの場合は、SSL 機能を有効にし、証明書キーをユーザー仮想サーバーにバインドする必要もあります。詳しくは、次のトピックを参照してください:

証明書とキーのペアを追加または更新する

証明書とキーのペアを SSL 仮想サーバーにバインドする

例:

```
enable ns feature SSL

add SSL certKey mqtt_svr_cert_key -cert server1.cert -key server1.key

bind ssl vserver mqtt_vs  -certkeyName mqtt_svr_cert_key
<!--NeedCopy-->
```

MQTT のエンドツーエンド暗号化による SSL オフロードの設定

August 15, 2023

次の例は、エンドツーエンド暗号化を使用して MQTT の SSL オフロードを行う方法を示しています。

注意: この例では、証明書とキーのペアを追加または更新し、それを仮想サーバーにバインドすることに関する詳細は提供していません。詳細については、SSL 証明書を参照してください。

次のコマンドは、拡張ファイルをインポートし、SSL トランスポートで MQTT_SSL プロトコルを追加します。

```
import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
add user protocol MQTT_SSL -transport SSL -extension mqtt_code
<!--NeedCopy-->
```

次のコマンドは、ユーザー負荷分散仮想サーバーを追加し、それにバックエンドサービスをバインドします。負荷分散仮想サーバーとサービスの両方が、サービスタイプ USER_SSL_TCP に設定されています。

```
add service mqtt_svr1 10.217.24.48 USER_SSL_TCP 1501
add service mqtt_svr2 10.217.24.48 USER_SSL_TCP 1502
add lb vserver mqtt_lb USER_SSL_TCP  - lbmethod RR
bind lb vserver mqtt_lb mqtt_svr1
bind lb vserver mqtt_lb mqtt_svr2
<!--NeedCopy-->
```

次のコマンドは、新しく追加されたプロトコル MQTT_SSL のユーザー仮想サーバーを追加します。MQTT_SSL は SSL トランスポートで構成されているため、MQTT_SSL を使用すると、NetScaler アプライアンスが SSL オフロードを実行することになります。また、このコマンドは、前のステップで設定した負荷分散仮想サーバーをデフォルトのロードバランサーにします。

```
add user vserver mqtt_vs MQTT_SSL 10.217.24.28 8765 -defaultLb
mqtt_lb
```

エンドツーエンドの暗号化を行うには、SSL 機能を有効にし、証明書キーをユーザーとデフォルトの負荷分散仮想サーバーにバインドする必要もあります。詳しくは、次のトピックを参照してください:

証明書とキーのペアを追加または更新する

証明書とキーのペアを SSL 仮想サーバーにバインドする

```
enable ns feature SSL

add SSL certKey mqtt_svr_cert_key -cert server1.cert -key server1.key

bind ssl vserver mqtt_lb  -certkeyName mqtt_svr_cert_key

bind ssl vserver mqtt_vs  -certkeyName mqtt_svr_cert_key
<!--NeedCopy-->
```

チュートリアル-プロトコル拡張を使用した **syslog** メッセージの負荷分散

August 15, 2023

NetScaler アプライアンスで使用可能な Syslog プロトコルは、NetScaler アプライアンスで生成されたメッセージに対してのみ機能します。外部ノードからのメッセージの負荷分散は行いません。このようなメッセージの負荷分散を行うには、プロトコル拡張機能を使用し、Lua 5.2 プログラミング言語を使用して syslog メッセージ解析ロジックを記述する必要があります。

Syslog メッセージを解析するためのコード

このコードには、TCP クライアントデータコールバック関数 client.on_data () のみが定義されています。サーバーデータの場合、コールバック関数は追加されず、サーバーからクライアントへの転送は高速なネイティブパスを取ります。このコードは、末尾の文字に基づいてメッセージの境界を識別します。TCP パケットに複数の syslog メッセージが含まれている場合は、末尾の文字に基づいてパケットを分割し、各メッセージをロードバランシングします。

```
--[[

  Syslog event handler for TCP client data

    ctxt - TCP client side App processing context.

    data - TCP Data stream received.

--]]

function client.on_data(ctxt, payload)

                  local message = nil

                  local data_len

                  local data = payload.data
```



```
18
19                  local trailing_character = "\n"
20
21                  ::split_message::
22
23                                  -- Get the offset of trailing
                                       character
24
25                                  local new_line_character_offset =
                                       data:find(trailing_character)
26
27                                  -- If trailing character is not
                                       found, then wait for more data.
28
29                                  if (not new_line_character_offset)
                                       then
30
31                                                  goto
                                                       need_more_data
32
33                                  end
34
35                                  -- Get the length of the current
                                       message
36
37                                  data_len = data:len()
38
39                                  -- Check whether we have more than
                                       one message
40
41                                  -- by comparing trailing character
                                       offset and
42
43                                  -- current data length
44
45                                  if (data_len >
                                       new_line_character_offset) then
46
47                                                  -- If we have
                                                       more than one
                                                       message, then
                                                       split
48
49                                                  -- the data into
                                                       two parts such
                                                        that first
                                                       part
50
51                                                  -- will contain
                                                       message upto
                                                       trailing
                                                       character
52
```



```
53                                                      -- offset and
                                                           second part
                                                           will contain
54
55                                                      -- remaining
                                                           message.
56
57                                                      message, data =
                                                           data:split(
                                                           new_line_character_offse
                                                           )
58
59                                      else
60
61                                                      message = data
62
63                                                      data = nil
64
65                                      end
66
67  -- Send the data to the backend server.
68
69                                      ns.send(ctxt.output, "EOM", {
70   data = message }
71   )
72
73                                      goto done
74
75                                      ::need_more_data::
76
77                                                      -- Wait for more
                                                           data
78
79                                                      ctxt:hold(data)
80
81                                                      data = nil
82
83                                                      goto done
84
85                                      ::done::
86
87                                                      -- If we have
                                                           more data to
                                                           parse,
88
89                                                      -- then do
                                                           parsing again.
90
91                                                      if (data) then
92
93                                                                      goto
                                                                        split_
```



```
94
95                                                          end
96
97 end
98 <!--NeedCopy-->
```

プロトコル拡張を使用した **syslog** プロトコルの設定

August 15, 2023

次の手順では、NetScaler アプライアンスにユーザー SYSLOG プロトコルを追加します。

Web サーバー（HTTP を使用）またはローカルワークステーションから、拡張子ファイルを NetScaler ADC アプライアンスにインポートします。拡張ファイルのインポートの詳細については、拡張機能のインポートを参照してください。

```
import ns extension local:syslog_parser.lua syslog_parser_code
```

拡張を使用して、新しいユーザー TCP ベースのプロトコルをシステムに追加します。

```
add user protocol USER_SYSLOG -transport TCP -extension syslog_parser_code
```

USER_TCP タイプのサービスを追加して、これがユーザー定義プロトコルであることを示します。

```
add service s1 10.102.90.112 USER_TCP 80
```

ユーザー負荷分散仮想サーバーを追加し、それにバックエンドサービスをバインドします。

```
1 add lb vs mysv USER_TCP
2
3 bind lb vs mysv s1
4 <!--NeedCopy-->
```

新しく追加したプロトコル用のユーザー仮想サーバーを追加し、前のステップで設定した負荷分散仮想サーバーをデフォルトのロードバランサーにします。

```
add user vs v_syslog USER_SYSLOG 10.217.24.28 80 -defaultlb mysv
```

プロトコル拡張 - コマンドリファレンス

August 15, 2023

次の表は、カスタムプロトコル用に追加されたすべての新しいコマンドと、カスタムプロトコル用に変更された既存のコマンドの一覧です。

```
show lb persistentSessions [<vserv-name>]
```

- **CLI** コマンド:

  ```
  add user protocol <name> -transport ( TCP | SSL )-extension <
  string> -comment <string>]>
  ```

- 説明:

  拡張機能を使用して、NetScaler アプライアンスに新しいユーザープロトコルを追加します。現在、トランスポート値が TCP または SSL のユーザープロトコルのみがサポートされています。

  > 例:
  >
  > ユーザープロトコル MQTT-トランスポート TCP-エクステンション mqtt_code の追加

- **CLI** コマンド:

  ```
  rm user protocol <name>
  ```

- 説明:

  NetScaler アプライアンスに以前に追加されたユーザープロトコルを削除します。

  > 例:
  >
  > rm ユーザープロトコル mqtt

- **CLI** コマンド:

  ```
  set user protocol <name> -comment <string>
  ```

- 説明:

  NetScaler アプライアンスに以前に追加されたユーザープロトコルの設定を変更します。

  > 例:
  >
  > ユーザープロトコル mqtt-comment「MQTT プロトコル実装」を設定

- **CLI** コマンド:

  ```
  unset user protocol <name> -comment
  ```

- 説明:

  NetScaler アプライアンスに以前に追加されたユーザープロトコルの設定を削除します。

> 例:
>
> ユーザープロトコル MQTT-コメント「MQTT プロトコル実装」を未設定

- **CLI** コマンド:

```
update ns extension <extension name>
```

- 説明:

  拡張機能を使用して、以前に追加されたユーザープロトコルの実装を更新します。

  プロトコル実装を更新できるのは、そのプロトコルがどのユーザー仮想サーバーでも使用されていない場合のみです。

> 例:
>
> ns エクステンション my-extension の更新

- **CLI** コマンド:

```
add lb vserver <name> [USER_TCP | USER_SSL_TCP] [-lbmethod
USER_TOKEN] [-persistencetype USERSESSION] [-timeout <value>]
```

- 説明:

  負荷分散仮想サーバーを NetScaler アプライアンスに追加します。これは既存の CLI コマンドです。

  ユーザー仮想サーバーの負荷分散の場合、使用するサービスタイプは USER_TCP または USER_SSL_TCP です。IP アドレスとポートは、ユーザー負荷分散仮想サーバーでは使用できません。

  ユーザー負荷分散仮想サーバーでは、ROUNDROBIN 負荷分散方式のみが許可され、トークンの値は拡張コードによって提供されます。同様に、USERSESSION パーシスタンスのみが許可され、パーシスタンス設定は拡張コードによって提供されます。

> 例:
>
> lb vserver mysv の追加 USER_TCP —lbmethod ラウンドロビン

- **CLI** コマンド:

```
add user vserver <name> <userProtocol> <IPAddress> <port> -
defaultLB <string> [-params <string>] [-comment <string>]
```

- 説明:

  拡張機能を使用してユーザープロトコル用の仮想サーバーを追加します。設定されたデフォルトのユーザー負荷分散仮想サーバーは、ctxt.output として TCP クライアントデータ拡張ハンドラーで使用できます。仮想サーバーでは、-params オプションに名前と値のペアを指定して拡張パラメーターを設定できます。対応するパラメーター値は、拡張ハンドラーが ctxt.vserver.params として利用できます。<paramName>。

> 例:
>
> ユーザーを追加 vs v_mqtt MQTT 10.217.24.28 80-defaultlb mysv

- **CLI** コマンド:

```
rm user vserver <name>
```

- 説明:

  NetScaler アプライアンスに以前に追加されたユーザー仮想サーバーを削除します。

> 例:
>
> rm ユーザー仮想サーバー v_mqtt

- **CLI** コマンド:

```
set user vserver <name> [-IPAddress <ip_addr|ipv6_addr|*>] [-
defaultLB <string>] [-params <string>] [-comment <string>]
```

- 説明:

  NetScaler アプライアンスに以前に追加されたユーザー仮想サーバーの設定を変更します。-params オプションによって拡張パラメータに新しい値が割り当てられると、古い値は上書きされます。

> 例:
>
> ユーザーと v_mqtt MQTT 10.217.24.28 を設定-defaultlb mysv-comment「MQTT プロトコル実装」

- **CLI** コマンド:

```
unset user vserver <name> [-params] [-comment]
```

- 説明:

  NetScaler アプライアンスに以前に追加されたユーザー仮想サーバーの設定を削除します。—params オプションを使用して拡張パラメーターの設定を解除すると、拡張ハンドラーで使用できる対応するパラメーター値は nil に変更されます。

> 例:
>
> ユーザーと v_mqtt MQTT 10.217.24.28-defaultlb mysv-comment「MQTT プロトコル実装」の設定解除

- **CLI** コマンド:

```
show user protocol [<name>]
```

- 説明:

  拡張機能やコールバックなど、ユーザープロトコルに関する情報を表示します。

> 例:
>
> ユーザープロトコル mqtt を表示

- **CLI** コマンド:

```
show user vserver [<name>]
```

- 説明:

  ユーザー仮想サーバーに関する情報を表示します。

> 例:
>
> ユーザー仮想サーバー vs_mqtt を表示

- **CLI** コマンド:

```
stat user vserver [<name>]
```

- 説明:

  ユーザー仮想サーバーに関する統計情報を表示します。

> 例:
>
> ユーザー仮想サーバー vs_mqtt の統計情報

- **CLI** コマンド:

```
show lb persistentSessions [<vserv-name>]
```

- 説明:

  永続セッションに関する情報を表示します。これは既存の CLI です。ユーザープロトコルの場合、パーシステンスタイプは USERSESSION と表示されます。

- **CLI** コマンド:

```
rm lb vserver <name>
```

- 説明:

  NetScaler アプライアンスに以前に追加されたユーザー LB 仮想サーバーを削除します。

> 例:
>
> rm lb vserver mysv

- **CLI** コマンド:

```
add service <name> <IPAddr> (USER_TCP | USER_SSL_TCP)<Port>
```


- 説明:

  ユーザープロトコルに使用するバックエンドサービスを追加します。これは既存の CLI コマンドで、新しいサービスタイプは USER_TCP と USER_SSL_TCP です。

  > 例:
  >
  > サービスの追加 mqtt_svr1 10.217.24.48 USER_TCP 1501
  >
  > 注: 既存の「サービスの設定とサービスの設定解除」コマンドを使用して、ユーザープロトコル用に以前に追加したサービスの設定を削除または変更できます。

- **CLI** コマンド:

  ```
  bind lb vserver <name> <serviceName>
  ```

- 説明:

  サービスをユーザー LB vserver にバインドします。USER_TCP/USER_SSL_TCP タイプの LB 仮想サーバーにバインドするには、サービスタイプを USER_TCP/USER_SSL_TCP にする必要があります。

  > 例:
  >
  > LB 仮想サーバー mysv mqtt_svr1 をバインド

- **CLI** コマンド:

  ```
  unbind lb vserver <name> <serviceName>
  ```

- 説明:

  以前にバインドされたサービスをユーザー LB vserver にバインド解除します。

  > 例:
  >
  > LB 仮想サーバー mysv mqtt_svr1 のバインド解除

- **CLI** コマンド:

  ```
  rm service <name>
  ```

- 説明:

  ユーザープロトコル用に以前に追加されたサービスを削除します。

  > 例:
  >
  > rm サービス mqtt_svr1

プロトコル拡張のトラブルシューティング

August 15, 2023

拡張関数が期待どおりに動作しない場合は、拡張トレース機能を使用して拡張関数の動作を検証できます。カスタムロギング機能を使用して拡張機能にロギングを追加することもできます。この機能では、NetScaler アプライアンスでキャプチャするログレベルを定義できます。

カスタムロギング

拡張機能に独自のロギングを追加することもできます。そのためには、組み込みの ns.logger: level () 関数を使用します。レベルは、緊急、アラート、クリティカル、エラー、警告、通知、情報、またはデバッグです。パラメータは C の printf () 関数と同じです。フォーマット文字列と、フォーマット文字列で指定された% の値を提供する可変数の引数です。たとえば、次のコードを COMBINE_HEADERS 関数に追加して、呼び出しの結果をログに記録することができます。

```
local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"

ns.logger:info("Result: %s", result_str)

return result_str
<!--NeedCopy-->
```

上記の関数は、上記の「拡張トレース」セクションの省略されたログメッセージの例に示されたサンプル入力について、次のメッセージを /var/log/ns.log に記録します。

```
... : default NSEXTENSION Message 143 0 : "Result: Host: 10.217.24.7:2000^
M H1: abcd, 1234^M User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4)
libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Accept: \*/\*^M H2: h2val1
, h2val2, h2val3^M ^M"
```

ポリシー拡張

August 15, 2023

ポリシー拡張機能を使用すると、組み込みポリシータイプの拡張関数を記述できます。拡張機能は、組み込み関数と同様にポリシー表現で使用できます。これらは、対応するポリシー表現が評価されるときに実行されます。この機能は次のような場合に便利です。

- 既存のポリシーにカスタマイズされた機能を追加します。
- 複雑な顧客要件に対応する論理構造の実装。

ポリシー拡張機能は、ユーザーが組み込みのポリシータイプ用の拡張関数を記述できるようにすることで、これらの制限に対処します。その後、組み込み関数と同様に、拡張機能をポリシー表現で使用できます。これらは、対応するポリシー表現が評価されるときに実行されます。

次の表は、エクステンションを書くときに使用できるポリシータイプとそれに関連するマッピングを示しています。

ポリシーの種類	マップポリシータイプ	出力
TEXT_T	NSTEXT	文字列
BOOL_AT	NSBOOL	ブーリアン型
NUM_AT	NSNUM	数値 (倍精度浮動小数点)
DOUBLE_AT	NSDOUBLE	数値 (倍精度浮動小数点)

ポリシー拡張を使用するための前提条件

インポートされた関数は、既存のポリシー標準に準拠している必要があります。したがって:

- 関数名は文字で始まらなければならず、数字やアンダースコアを含むことができます。
- 関数名は、NetScaler ポリシーによって大文字と小文字が区別されないように扱われます。
- 拡張言語が複数の値を返す場合でも、関数は単一の値を返さなければなりません。
- 引数の数が可変の関数はサポートされていません。

ポリシーの拡張はどのように機能しますか?

NetScaler アプライアンスの既存のポリシーは、インタープリターを使用して機能を評価し、ポリシー拡張ファイルにインポートされます。ユーザーがポリシー拡張ファイルに新しい関数をインポートすると、

1. 拡張ファイルの構文やその他の条件が検証されます。
2. 検証に失敗すると、エラーがユーザーに報告されます。
3. 検証が成功すると、拡張ファイルが NetScaler アプライアンスにインポートされ、組み込みのポリシー機能と同様に、その内容をポリシー表現で使用できます。

    a) 実行時にポリシー表現の評価でエラーが返された場合、undef イベントとして報告され、関連するエラーカウンタがインクリメントされます。

    注: ポリシー未定義イベントが発生し、ポリシー規則に 1 つ以上のポリシー拡張機能が含まれている場合、`show ns extension <name>` コマンドはそれらのポリシー拡張に適用されたときに undef ヒットを表示します。拡張関数が中止されると、中止カウンタの値が増えます。

    b) ポリシー表現の評価が成功した場合、式全体が評価されるか、エラーにより中止されるまで、式の評価が再開されます。

拡張関数の実行に時間がかかりすぎると、その拡張関数は中止され、その拡張関数に関連するエラーカウンタがインクリメントされます。拡張機能はサンドボックス化されているため、次のことが防止されます。

- NetScaler アプライアンスの CPU 使用率が高すぎる。
- NetScaler アプライアンスのメモリ使用量が多すぎる。
- 有害な組み込みライブラリまたはサードパーティのライブラリまたはバイナリの使用。
- NetScaler アプライアンスの再起動の原因となる可能性のある長時間実行のスクリプト。

ポリシー拡張の設定

August 15, 2023

ポリシー拡張ファイルの準備ができたら、NetScaler アプライアンスにインポートします。インポートプロセスでは、拡張ファイルを NetScaler アプライアンスのディレクトリにコピーし、構文エラーをチェックします。

インポート後、拡張ファイルをポリシー式で使用できるようにする必要があります。

注: import コマンドは、外部ソース \<src\>または内部ソースから NetScaler ADC ファイルシステムにファイルコンテンツをダウンロードするために使用されます。このファイルコンテンツを 1 つ以上のパケットエンジンに初めてロードするには、add コマンドを使用します。ファイルコンテンツに更新がある場合、overwrite 引数を指定してインポートコマンドを発行することで、更新されたコンテンツを NetScaler ADC ファイルシステムにダウンロードできます。このコマンドは、ファイルシステムの内容を更新します。更新されたコンテンツを 1 つ以上のパケットエンジンにロードするには、update コマンドを使用します。

CLI を使用したポリシー拡張の設定

1. ポリシー拡張ファイルを、Web サーバー（HTTP を使用）またはローカルワークステーションから NetScaler Appliance にインポートします。

   a) HTTP インポート

   使用可能な Web サーバーがある場合は、拡張ファイルを Webserver ディレクトリに保存し、NetScaler アプライアンスにインポートできます。

   ```
   import ns extension <src> <name> [-comment<string>] [-
      overwrite]
   <!--NeedCopy-->
   ```

   例:

   ```
   import ns extension http://myhost/path/to/extension
      myextension -comment  “Custom crc calculation
   <!--NeedCopy-->
   ```


b) ローカルインポート

SSH クライアントを使用して、ワークステーションから NetScaler ADC アプライアンスの/var/tmp ディレクトリに拡張ファイルをコピーできます。

```
scp extension-file-name <ns-userid@ns-ip-addr>:/var/tmp
<!--NeedCopy-->
```

各項目の意味は次のとおりです。

- extension-file-name は、クライアントマシン上の拡張ファイルの名前です。
- ns-userid NetScaler ADC アプライアンスユーザーには、/var/tmp への書き込み権限が付与されているか。
- ns-ip-addr は、NetScaler IP アドレスです。

ファイルを NetScaler ADC アプライアンスにコピーした後、NetScaler ADC アプライアンス上でインポートコマンドを実行します。

```
import ns extension local:<extension-file-name extension-name>
<!--NeedCopy-->
```

注**:** ローカル拡張ファイルをインポートするには、
import コマンドを実行して CLI を使用する必要があります。

2. 評価用にポリシー拡張をパケットエンジンに追加します。

```
add ns extension <name> [-comment <string>]
<!--NeedCopy-->
```

例**:**

```
add ns extension myextension
<!--NeedCopy-->
```

拡張ファイルをインポートした後、インポートコマンドに-overwrite パラメータを含めた場合はそのファイルを更新したり、削除したりできます。インポートした拡張ファイルの詳細を表示することもできます。

NetScaler アプライアンス上の拡張ファイルをソースから更新します

コマンドプロンプトで入力します。

```
update ns extension <name>
<!--NeedCopy-->
```

注：拡張ファイルを更新できるのは、指定した拡張ファイルを-overwrite パラメーターを使用して NetScaler アプライアンスにインポートした後だけです。

例**:**


```
update ns extension myextension
<!--NeedCopy-->
```

NetScaler ADC アプライアンスからの拡張ファイルの削除

コマンドプロンプトで次のように入力します。

```
rm ns extension <name>
<!--NeedCopy-->
```

例**:**

```
rm ns extension myextension
<!--NeedCopy-->
```

NetScaler ADC アプライアンス上で指定した拡張機能の詳細を表示します

コマンドプロンプトで入力します。

```
show ns extension <name>
<!--NeedCopy-->
```

例**:**

```
show ns extension myextension
<!--NeedCopy-->
```

GUI を使用してポリシー拡張を設定する

1. ポリシー拡張ファイルを、Web サーバー（HTTP を使用）またはローカルワークステーションから NetScaler Appliance にインポートします。
   a) [**AppExpert**] > [ポリシー拡張] に移動し、[ポリシー拡張] をクリックし、[インポート元**]** ドロップダウンリストから、インポートする拡張ファイルの場所の URL を選択します。
   b) **AppExpert** > ポリシー拡張機能、ポリシー拡張に移動し、[インポート元] ドロップダウンリストで [ファイル] を選択して拡張ファイルをインポートします。
2. 評価用にポリシー拡張をパケットエンジンに追加します。

   [**AppExpert**] > [ポリシー拡張] に移動し、[ポリシー拡張**]** タブで拡張ファイルを追加します。

NetScaler アプライアンス上の拡張ファイルをソースから更新します

[**AppExpert**] > [ポリシー拡張] に移動し、[ポリシー拡張**]** タブで拡張ファイルを更新します。

NetScaler ADC アプライアンスからの拡張ファイルの削除

AppExpert > ポリシー拡張に移動し、[ポリシー拡張] タブで、拡張ファイルを削除します。

NetScaler ADC アプライアンス上で指定した拡張機能の詳細を表示します

AppExpert > ポリシー拡張に移動し、[ポリシー拡張機能] タブで、詳細を表示する拡張機能の [クリック] ドロップダウンリスト矢印をクリックします。

ポリシー拡張 - ユースケース

August 15, 2023

特定の顧客アプリケーションには、既存のポリシーや表現では対応できない要件があります。ポリシー拡張機能により、お客様は要件に合わせてカスタマイズされた機能をアプリケーションに追加できます。

次の使用例は、NetScaler アプライアンスのポリシー拡張機能を使用して新しい機能を追加する方法を示しています。

- ケース 1: カスタムハッシュ
- ケース 2: URL の二重スラッシュを折りたたむ
- ケース 3: ヘッダーの結合

ケース **1:** カスタムハッシュ

CUSTOM_HASH 関数は、クライアントに送信される応答に任意のタイプのハッシュ値を挿入するメカニズムを提供します。このユースケースでは、ハッシュ関数を使用して HTTP 書き換えリクエストのクエリ文字列のハッシュを計算し、その計算値を含む CUSTOM_HASH という名前の HTTP ヘッダーを挿入します。CUSTOM_HASH 関数は DJB2 ハッシュアルゴリズムを実装しています。

カスタムハッシュの使用例**:**

```
1 > add rewrite action test_custom_hash insert_http_header "CUSTOM_HASH"
     "HTTP.REQ.URL.QUERY.CUSTOM_HASH"
2 <!--NeedCopy-->
```

カスタムハッシュ **()** のサンプル定義**:**

```
1     -- Extension function to compute custom hash on the text
2
3     -- Uses the djb2 string hash algorithm
4     function NSTEXT:CUSTOM_HASH() : NSTEXT
5
```



```
6           local hash = 5381
7
8           local len = string.len(self)
9
10          for i = 1, len do
11
12               hash = bit32.bxor((hash * 33), string.byte(self, i))
13
14          end
15
16          return tostring(hash)
17
18      end
19  <!--NeedCopy-->
```

上記のサンプルの **1** 行ごとの説明**:**

```
1  function NSTEXT:CUSTOM_HASH() : NSTEXT
2
3  Defines the CUSTOM_HASH() function, with text input and a text return
       value.
4
5  local hash = 5381
6  local len = string.len(self)
7
8  Declares two local variables:
9
10  - hash. Accumulates the compute hash value and is seeded with the
       number 5381
11
12  - len. Sets to the length of the self input text string, using the
       built-in string.len() function.
13
14 for i = 1, len do
15       hash = bit32.bxor((hash * 33), string.byte(self, i))
16 end
17
18 Iterates through each byte of the input string and adds the byte to the
       hash. It uses the built-in string.byte() function to get the byte
      and the built-in bit32.bxor() function to compute the XOR of the
      existing hash value (multiplied by 33) and the byte.
19
20 return tostring(hash)
21
22 Calls the built-in tostring() function to convert the numeric hash
      value to a string and returns the string as the value of the
      function.
23 <!--NeedCopy-->
```


ケース **2: URL** の二重スラッシュを折りたたむ

URL の二重スラッシュを折りたたむと、ブラウザが単一スラッシュ URL をより効率的に解析するため、Web サイトのレンダリング時間が短縮されます。単一スラッシュの URL は、二重スラッシュを受け入れないアプリケーションとの互換性を維持するためでもあります。ポリシー拡張機能により、お客様は URL の二重スラッシュをシングルスラッシュに置き換える機能を追加できます。次の例は、URL の二重スラッシュを折りたたむポリシー拡張機能の追加を示しています。

COLLAPSE_DOUBLE_SLASHES () のサンプル定義**:**

```
    -- Collapse double slashes in URL to a single slash and return the
       result
    function NSTEXT:COLLAPSE_DOUBLE_SLASHES() : NSTEXT

        local result = string.gsub(self, "//", "/")

        return result

    end
<!--NeedCopy-->
```

上記のサンプルの **1** 行ごとの説明**:**

```
function NSTEXT:COLLAPSE_DOUBLE_SLASHES() : NSTEXT

Declares the COLLAPSE_DOUBLE_SLASHES() function with text input and
   return.

local result = string.gsub(self, "//", "/")

Declares a local variable named result and uses the built-in string.
   gsub() function to replace all double slashes with single slashes in
    the self input text.

The second parameter of string.gsub() is actually a regular expression
   pattern, although here a simple string is used for the pattern.

return result

Returns the resulting string.
<!--NeedCopy-->
```

ケース **3:** ヘッダーの結合

特定のカスタマーアプリケーションでは、リクエスト内の複数のヘッダーを処理できません。また、同じヘッダー値を持つ重複したヘッダー、またはリクエスト内の同じ名前で値が異なる複数のヘッダーを解析すると、時間とネットワークリソースが消費されます。ポリシー拡張機能により、お客様はこれらのヘッダーを元の値を組み合わせた値を持つ単一のヘッダーにまとめる機能を追加できます。たとえば、ヘッダー H1 と H2 の値を組み合わせます。

オリジナルのリクエスト:

```
GET /combine_headers HTTP/1.1
User-Agent: amigo unit test
Host: myhost
H2: h2val1
H1: abcd
Accept: \*/\*
H2: h2val2
Content-Length: 0
H2: h2val3
H1: 1234
<!--NeedCopy-->
```

変更されたリクエスト:

```
GET /combine_headers HTTP/1.1
User-Agent: amigo unit test
Host: myhost
H2: h2val1, h2val2, h2val3
H1: abcd, 1234
Accept: \*/\*
Content-Length: 0
<!--NeedCopy-->
```

一般に、このタイプのリクエスト変更は、リライト機能を使用して行われます。ポリシー表現を使用して、リクエストの変更する部分（ターゲット）と実行する変更（ストリングビルダー式）を記述します。ただし、ポリシー表現には任意の数のヘッダーを反復処理することはできません。

この問題を解決するには、ポリシーファシリティの拡張が必要です。そのために、COMBINE_HEADERS という拡張関数を定義します。この関数では、次の書き換えアクションを設定できます。

```
> add rewrite action combine_headers_act replace 'HTTP.REQ.FULL_HEADER
.AFTER_STR("HTTP/1.1rn")''HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1rn
").COMBINE_HEADERS'
```

ここで、書き換え対象は HTTP.REQ.FULL_HEADER.AFTER_STR (「HTTP/1.1rN」) です。FULL_HEADER には HTTP リクエストの最初の行 (例:GET /combine_headers HTTP/1.1) が含まれているため、AFTER_STR (「HTTP/1.1rN」) が必要です。

ストリングビルダーの式は HTTP.REQ.FULL_HEADER.AFTER_STR (「HTTP/1.1rN」) .COMBINE_HEADERS です。ここで、ヘッダー (最初の行を引いたもの) が COMBINE_HEADERS 拡張関数に渡され、ヘッダーの値が結合されて返されます。

COMBINE_HEADERS () のサンプル定義:

```
    -- Extension function to combine multiple headers of the same name
       into one header.


```



```
5     function NSTEXT:COMBINE_HEADERS(): NSTEXT
6
7         local headers = {
8   }
9   -- headers
10
11        local combined_headers = {
12  }
13  -- headers with final combined values
14        -- Iterate over each header (format "name:valuer\r\n")
15
16        -- and build a list of values for each unique header name.
17
18        for name, value in string.gmatch(self, "([^:]+):([^\r\n]*)\r\n"
              ) do
19
20            if headers[name] then
21
22                local next_value_index = #(headers[name]) + 1
23
24                headers[name][next_value_index] = value
25
26            else
27
28                headers[name] = {
29  name .. ":" .. value }
30
31
32            end
33
34        end
35
36
37
38        -- iterate over the headers and concat the values with
              separator ","
39
40        for name, values in pairs(headers) do
41
42            local next_header_index = #combined_headers + 1
43
44            combined_headers[next_header_index] = table.concat(values,
                  ",")
45
46        end
47
48
49
50        -- Construct the result headers using table.concat()
51
52        local result_str = table.concat(combined_headers, "\r\n") .. "\
              r\n\r\n"
53
```



```
54          return result_str
55
56      end
57  <!--NeedCopy-->
```

上記のサンプルの **1** 行ごとの説明**:**

```
1  function NSTEXT:COMBINE_HEADERS(): NSTEXT
2
3  Defines the COMBINE_HEADERS extension function, with the text input
       into the function from the policy expression and a text return type
       to the policy expression.
4
5  local headers = {
6    }
7    -- headers
8  local combined_headers = {
9    }
10   -- headers with final combined values
11
12 Declares local variables headers and combined_headers and initialize
       these variables to empty tables. headers will be a table of arrays
       of strings, where each array holds one or more values for a header.
       combined_headers will be an array of strings, where each array
       element is a header with its combined values.
13
14 for name, value in string.gmatch(self, "([^:]+):([^\r\n]*)\r\n") do
15 . . .
16 end
17 <!--NeedCopy-->
```

この汎用の for ループは、入力の各ヘッダーを解析します。イテレータは組み込みの string.gmatch () 関数です。この関数は 2 つのパラメータを取ります。1 つは検索する文字列で、もう 1 つは文字列の一部を一致させるために使用するパターンです。検索する文字列は、関数に入力されるヘッダーのテキストである暗黙のセルフパラメーターによって提供されます。

パターンは正規表現 (略して regex) を使用して表現されます。この正規表現は、HTTP ** 標準では *name*: value\ r\ n と定義されている各ヘッダーのヘッダー名と値と一致します。** 正規表現の括弧は抽出する一致する部分を指定するので、正規表現の回路図は *(match-name): **(*match-value***)\ r\ n になります。***match-name** パターンは、コロン以外のすべての文字と一致する必要があります。これは`[^:]+`が書かれています。`[^:]`は`:`を除く任意の文字で、+は 1 回以上繰り返されます。同様に、マッチバリューパターンは、`\r\n`以外のすべての文字と一致しなければならないので、`[^\r\n]*`が書き込まれます。`[^\r\n]`は`\r`と`\n`を除く任意の文字と一致し、*は 0 回以上繰り返されます。`([^:]+):([^\r\n]*)\r\n`これで完全な正規表現が完成します。

for ステートメントは、複数の代入を使用して string.gmatch () イテレータから返された 2 つのマッチに名前と値を設定します。これらは for ループの本体内でローカル変数として暗黙的に宣言されます。

```
1  if headers[name] then
2      local next_value_index = #(headers[name]) + 1
3      headers[name][next_value_index] = value
```



```
else
     headers[name] = {
 name .. ":" .. value }

end
<!--NeedCopy-->
```

for ループ内のこれらのステートメントは、ヘッダー名と値をヘッダーテーブルに入れます。ヘッダー名が初めて解析されるとき（たとえば、入力例では H2: h2val1)、名前のヘッダーエントリはなく、headers [name] は nil です。

nil は false として扱われるので、else 節が実行されます。これは、name のヘッダエントリを 1 つの文字列値 *name*:*value* を持つ配列に設定します。

注**:** else ループの配列コンストラクターは {[1] = name と同じです。「:」.. value}: 配列の最初の要素を設定します。) 最初の H2 ヘッダーには、ヘッダー [“H2”] = { “h2: H2val1” } を設定します。

ヘッダーの後続のインスタンス (たとえば、入力例の H2: h2val2) では。headers [name] は nil ではないので、then 句が実行されます。これにより、headers [name] の配列値で次に利用できるインデックスを決定し、そのインデックスにヘッダー値を格納します。2 番目の H2 ヘッダーには、ヘッダー [“H2”] = { “h2: H2val1” ,「h2val2” } を設定します。

```
for name, values in pairs(headers) do
    local next_header_index = #combined_headers + 1
    combined_headers[next_header_index] = table.concat(values, ",")
end
<!--NeedCopy-->
```

元のヘッダーが解析され、ヘッダーテーブルが入力された後、このループは combined_headers 配列を作成します。for ループイテレータとして pairs () 関数を使用します。

pairs () を呼び出すたびに、ヘッダーテーブルの次のエントリの名前と値が返されます。

次の行は combined_headers 配列で次に使用可能なインデックスを決定し、次の行ではその配列要素を組み合わせたヘッダーに設定します。組み込みの table.concat () 関数を使用します。この関数は、文字列の配列とセパレータとして使用する文字列を引数にとり、セパレータで区切られた配列文字列を連結した文字列を返します。

たとえば、= { “h2: H2val1” ,「h2val2” } という値の場合、「h2: H2val1, h2val2」が生成されます

```
local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"
<!--NeedCopy-->
```

combined_headers 配列が作成されると、要素を 1 つの文字列に連結し、HTTP ヘッダーを終了する 2 つの\ r\ n を追加します。

```
return result_str
<!--NeedCopy-->
```

COMBINE_HEADERS 拡張関数の結果として文字列を返します。

ポリシー拡張のトラブルシューティング

August 15, 2023

拡張関数が期待どおりに動作しない場合は、拡張トレース機能を使用して拡張関数の動作を検証できます。カスタムロギング機能を使用して拡張機能にロギングを追加することもできます。この機能では、NetScaler アプライアンスでキャプチャするログレベルを定義できます。

このトピックでは、次の情報を提供します。

- エクステンショントレース
- カスタムロギング

エクステンショントレース

拡張関数の実行内容を確認するために、拡張トレース機能は関数の実行を NetScaler システムログ(/var/log/ns.log)に記録します。トレースロギングは DEBUG ログレベルを使用しますが、通常は有効になっていません。そのため、すべてのログレベルを有効にする必要があります。その後、set ns extension コマンドの-trace オプションを設定することで、トレースを有効にできます。使用可能な設定は以下のとおりです。

- off トレースをオフにする (ns extension-trace の設定を解除するのと同等)。
- 呼び出しは、引数を指定して関数呼び出しをトレースし、最初の戻り値を使用して関数呼び出しをトレースします。
- 行は上記のトレースと実行された行の行番号をトレースします。
- すべて上記と実行された行によって変更されたローカル変数をトレースします。

例:

```
set audit syslogParams -loglevel ALL

set ns extension combine_headers -trace all
<!--NeedCopy-->
```

各トレースメッセージには、次の形式があります。

```
log-header : default NSEXTENSION Message message-number 0 : "TRACE function-name CALL call-number: event"
```

各項目の意味は次のとおりです。

- ログヘッダーは、タイムスタンプ、NetScaler IP アドレス、およびパケットエンジン ID を提供します。
- メッセージ番号はログメッセージを識別する連続番号です。
- 関数名は拡張関数名です。
- call-number は、各拡張関数呼び出しの連続番号です。拡張関数呼び出しのすべてのトレースメッセージをグループ化するために使用できます。

- イベントは次のいずれかです。
  - CALL 関数名; パラメータ値は、関数が指定されたパラメータで呼び出されたことを示します。
  - RETURN FROM 関数名; return = value は、関数が指定された (最初の) 値を返したことを示します。(その他の戻り値は報告されません)。
  - LINE 行番号。変数値は、行が実行されたことを示し、値が変更されたすべての変数を一覧表示します。

各項目の意味は次のとおりです。

- 1 つまたは複数の値は
  - 小数点のある場合とない場合の数値
  - 前述のように二重引用符で囲まれてエスケープされた文字列
  - 真か偽かのブーリアン、
  - nil、
  - {[キー 1] = 値 1、[キー 2] = 値 2、…} という形式のテーブルコンストラクター。
- parameter-values is parameter1 = value1 ; parameter2 = value2 , …
- variable-values is variable1 = value1 ; variable2 = value2 , …

簡略化されたログメッセージの例:

```
>shell tail -f /var/log/ns.log | grep TRACE | more

... NSEXTENSION Message 3035 0 : "TRACE combine_headers CALL 30 : CALL COMBINE_HEADERS; self = "User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nHost: 10.217.24.7\r\nAccept: \*/\*\r\nH2: h2val1\r\nH1: abcd\r\nH2: h2val2\r\nH2: h2val3\r\n\r\n""

... NSEXTENSION Message 3036 0 : "TRACE combine_headers CALL 30 : LINE 4; headers = {
  }
 "

... NSEXTENSION Message 3037 0 : "TRACE combine_headers CALL 30 : LINE 5; combined_headers = {
  }
 "

... NSEXTENSION Message 3038 0 : "TRACE combine_headers CALL 30 : CALL gmatch"

... NSEXTENSION Message 3039 0 : "TRACE combine_headers CALL 30 : RETURN FROM gmatch; return = function 0x2bee5a80"

... NSEXTENSION Message 3040 0 : "TRACE combine_headers CALL 30 : CALL for iterator"

```



```
19 ... NSEXTENSION Message 3041 0 : "TRACE combine_headers CALL 30 :
      RETURN FROM for iterator; return = " curl/7.24.0 (amd64-portbld-
      freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3""
20
21 ... NSEXTENSION Message 3042 0 : "TRACE combine_headers CALL 30 : LINE
      9; name = "User-Agent"; value = " curl/7.24.0 (amd64-portbld-
      freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3""
22
23 ... NSEXTENSION Message 3043 0 : "TRACE combine_headers CALL 30 : LINE
      10"
24
25 ... NSEXTENSION Message 3044 0 : "TRACE combine_headers CALL 30 : LINE
      14; headers = {
26  ["User-Agent"]={
27  [1]="User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0
        OpenSSL/0.9.8y zlib/1.2.3" }
28   }
29  "
30
31 . . .
32
33 ... NSEXTENSION Message 3117 0 : "TRACE combine_headers CALL 30 : CALL
      for iterator"
34
35 ... NSEXTENSION Message 3118 0 : "TRACE combine_headers CALL 30 :
      RETURN FROM for iterator; return = nil"
36
37 ... NSEXTENSION Message 3119 0 : "TRACE combine_headers CALL 30 : LINE
      19"
38
39 ... NSEXTENSION Message 3120 0 : "TRACE combine_headers CALL 30 : CALL
      concat"
40
41 ... NSEXTENSION Message 3121 0 : "TRACE combine_headers CALL 30 :
      RETURN FROM concat; return = "User-Agent: curl/7.24.0 (amd64-portbld
      -freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\
      nAccept: \*/\*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3""
      ... NSEXTENSION Message 3122 0 : "TRACE combine_headers CALL 30 :
      LINE 25; result_str = "User-Agent: curl/7.24.0 (amd64-portbld-
      freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\
      nAccept: \*/\*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3\r\
      n\r\n""
42
43 ... NSEXTENSION Message 3123 0 : "TRACE combine_headers CALL 30 :
      RETURN FROM COMBINE_HEADERS; return = "User-Agent: curl/7.24.0 (
      amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r
      \nH1: abcd\r\nAccept: \*/\*\r\nHost: 10.217.24.7\r\nH2: h2val1,
      h2val2, h2val3\r\n\r\n""
44 <!--NeedCopy-->
```


カスタムロギング

拡張機能に独自のロギングを追加することもできます。そのためには、組み込みの ns.logger:*level*() 関数を使用します。レベルは、緊急、アラート、クリティカル、エラー、警告、通知、情報、またはデバッグです。パラメータは C の printf () 関数と同じです。フォーマット文字列と、フォーマット文字列で指定された% の値を提供する可変数の引数です。たとえば、次のコードを COMBINE_HEADERS 関数に追加して、呼び出しの結果をログに記録することができます。

```
local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"

ns.logger:info("Result: %s", result_str)

return result_str
<!--NeedCopy-->
```

上記の関数は、上記の「拡張トレース」セクションの省略されたログメッセージの例に示されたサンプル入力について、次のメッセージを /var/log/ns.log に記録します。

```
... : default NSEXTENSION Message 143 0 : "Result: Host: 10.217.24.7:2000^M H1: abcd, 1234^M User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Accept: */*^M H2: h2val1, h2val2, h2val3^M ^M"
```

最適化

August 15, 2023

NetScaler の最適化機能により、クライアントとサーバー間のトランザクション時間が短縮され、帯域幅の消費量が減少します。また、一部のタスクをオフロードし、他のタスクをより効率的にすることで、サーバーのパフォーマンスを向上させます。

機能	説明
クライアント Keep-Alive	1 つのクライアント接続で複数の要求を処理します。クライアントは、サーバーへの要求ごとに新しい接続をネゴシエートする必要はありません。
HTTP 圧縮	サーバーから圧縮対応ブラウザーに送信される HTTP 応答を圧縮します。応答が小さいほど、ダウンロード時間が短縮され、帯域幅が節約されます。

機能	説明
統合キャッシング	クライアント要求への応答を保存します。同じコンテンツに対する後続のリクエストは、オリジンサーバーに転送されるのではなく、NetScaler キャッシュから処理されます。
フロントエンドの最適化	クライアントブラウザーに提供されるコンテンツを簡素化および最適化することにより、Web ページの読み込みとレンダリング時間を短縮します。注：NetScaler 10.5 以降でサポートされています。

Client Keep-Alive

December 8, 2023

Client Keep-Alive 機能により、複数のクライアント要求を単一の接続で送信できます。この機能は、トランザクション管理に役立ちます。アプライアンスで Client Keep-Alive モードが有効で、クライアント要求に対するサーバ応答に Connection が含まれている場合：HTTP ヘッダーを閉じて、次のタスクを実行します。

- ヘッダー名の文字をシャッフルして、既存の Connection ヘッダー名の名前を変更します。
- Keep-Alive をヘッダーの値として新しい Connection: ヘッダーを追加します。

Client Keep-Alive モードを使用すると、NetScaler アプライアンスは同じソケット接続を使用して複数の要求と応答を処理できます。この機能は、サーバーがアプライアンスとの接続を閉じた後でも、クライアントとアプライアンスの間の接続（クライアント側接続）を開いたままにします。これにより、単一の接続を使用して複数のクライアント要求が可能になり、接続の開閉に関連するラウンドトリップが節約されます。Client Keep-Alive は SSL セッションで最も有益です。

Client Keep-Alive は、次のシナリオで役立ちます。

- サーバーがクライアントのキープアライブをサポートしていない場合。
- サーバーはサポートしているが、サーバー上のアプリケーションがクライアントのキープアライブをサポートしていない場合。

> 注**:**
> Client Keep-Alive は HTTP および SSL トラフィックに適用されます。Client-Keep Alive は、すべてのトラフィックを処理するようにグローバルに設定できます。また、特定のサービスで有効にすることもできます。

Client Keep-Alive 環境では、設定されたサービスがクライアントトラフィックをインターセプトし、クライアント要求はオリジンサーバーに送信されます。サーバーは応答を送信し、サーバーとアプライアンス間の接続を閉じます。

サーバーの応答に「Connection: Close」ヘッダーが存在する場合、アプライアンスはクライアント側の応答でこのヘッダーを壊し、クライアント側の接続は開いたままになります。その結果、クライアントは次のリクエストで新しい接続を開く必要がなくなります。代わりに、サーバーへの接続が再開されます。

> 注:
> サーバーが「Connection: Close」ヘッダーを 2 つ送り返しても、編集されるのは 1 つだけです。その結果、クライアントは接続が閉じられるまでオブジェクトが完全に配信されたとは想定しないため、クライアントによるオブジェクトのレンダリングが大幅に遅れます。

Client Keep-Alive の設定

NetScaler では、Client Keep-Alive はデフォルトでグローバルレベルでもサービスレベルでも無効になっています。そのため、必要な範囲で機能を有効にする必要があります。

> 注:
> クライアントのキープアライブをグローバルに有効にすると、サービスレベルで有効にするかどうかに関係なく、すべてのサービスで有効になります。また、一部の HTTP パラメータを設定して以下を指定する必要があります。
>
> - 接続再利用プールに保持される HTTP 接続の最大数。
> - 接続の多重化を有効にし、永続性を有効にします。`Etag`
>
> 注:
> Persistent `ETag` が有効な場合、`ETag` ヘッダーにはコンテンツを提供したサーバーに関する情報が含まれます。これにより、そのコンテンツに対するキャッシュ検証条件付きリクエストまたはブラウザリクエストが常に同じサーバーに到達することが保証されます。

NetScaler コマンドインターフェイスを使用して Client Keep-Alive を設定する

コマンドプロンプトで、次の操作を行います:

1. NetScaler で Client Keep-Alive を有効にします。
   - 世界レベルでは- `enable ns mode cka`
   - サービスレベルでは- `set service <name> -CKA YES`

   > 注:
   >
   > Client Keep-Alive は、HTTP および SSL サービスでのみ有効にできます。

2. 1 つ以上のサービスにバインドされている HTTP プロファイルに HTTP パラメータを設定します。

```
set ns httpProfile <name> -maxReusePool <value> -conMultiplex
    ENABLED -persistentETag ENABLED
<!--NeedCopy-->
```


> 注:
>
> `nshttp_default _profile HTTP` これらのパラメータをプロファイルに設定して、グローバルに使用できるようにします。

NetScaler GUI を使用して **Client Keep-Alive** を構成する

1. NetScaler で Client Keep-Alive を有効にします。
   - グローバルレベルで

     [システム] > [設定] に移動し、[モードの設定] をクリックして [クライアント側キープアライブ] を選択します。
   - サービスレベルで

     [トラフィック管理] > [負荷分散] > [サービス] に移動し、必要なサービスを選択します。「設定」セクションで、「**Client Keep-Alive** 」チェックボックスを選択します。
2. 1 つ以上のサービスにバインドされている HTTP プロファイルで必要な HTTP パラメータを設定します。
3. [システム] > [プロファイル] に移動し、[**HTTP** プロファイル] タブで必要なプロファイルを選択し、必要な HTTP パラメータを更新します。

HTTP 圧縮

August 16, 2023

圧縮可能なコンテンツを持つ Web サイトの場合、HTTP 圧縮機能は可逆圧縮を実装し、サーバーから圧縮対応ブラウザーに送信される HTTP 応答を圧縮することにより、レイテンシ、長いダウンロード時間、およびその他のネットワークパフォーマンスの問題を軽減します。計算負荷の高い圧縮タスクをサーバーから NetScaler ADC アプライアンスにオフロードすることで、サーバーのパフォーマンスを向上させることができます。

次の表に、HTTP 圧縮機能の機能を示します。

機能	説明
圧縮率	圧縮率は、応答内のファイルのタイプによって異なりますが、常に重要であり、ネットワーク上で送信されるデータの量を大幅に削減します。

機能	説明
ブラウザ認識	NetScaler ADC は、圧縮データを圧縮対応ブラウザにのみ提供し、クライアントとサーバー間のトランザクション時間を短縮します。最新のウェブブラウザのほとんどは HTTP 圧縮をサポートしています。
圧縮ブロッキング	組み込みアクションを適用して、圧縮を選択的にブロックするコンテンツフィルタを定義できます。
圧縮キャッシュ	統合キャッシュ機能を有効にすると、同じコンテンツに対する後続のリクエストがローカルキャッシュから処理されるため、サーバーへのラウンドトリップ回数が削減され、トランザクション時間が短縮されます。
HTTPS サポート	圧縮は、サーバーまたは NetScaler ADC アプライアンスによって暗号化され、クライアントによって復号化されるコンテンツの量を減らすため、SSL 接続で役立ちます。
インテリジェントな応答フィルタリング	NetScaler ADC 圧縮エンジンは、定義された圧縮パラメータに基づいてサーバーの応答をインテリジェントにフィルタリングします。たとえば、圧縮エンジンは、コンテンツ長ゼロの応答と圧縮された応答を検出し、圧縮しません。圧縮された応答の検出により、オリジンサイトは NetScaler ADC 圧縮機能でサーバーベースの圧縮を使用できます。
圧縮スイッチング	NetScaler ADC アプライアンスは、圧縮対応クライアントからの要求を圧縮対応サーバーに透過的に送信するため、これらのクライアントへの応答は圧縮され、他のクライアントへの応答は圧縮処理によって遅延されません。

HTTP 圧縮のしくみ

NetScaler ADC は、静的および動的に生成されたデータの両方を圧縮できます。GZIP または DEFLATE 圧縮アルゴリズムが適用されることで、無関係で反復的な情報がサーバー応答から削除され、より簡潔で効率的な形式で元の情報が表されます。この圧縮データは、クライアントのブラウザに送信され、ブラウザがサポートするアルゴリズム（GZIP または DEFLATE）によって決定されると解凍されます。

NetScaler ADC 圧縮は、静的コンテンツと動的コンテンツを別々に扱います。

- 静的ファイルは一度だけ圧縮され、圧縮されたコピーはローカルメモリに保存されます。キャッシュファイルに対する後続のクライアント要求は、そのメモリから処理されます。

- 動的ページは、クライアントが要求するたびに動的に作成されます。

クライアントがサーバーに要求を送信すると、次のようになります。

1. クライアント要求は NetScaler ADC に到着します。ADC はヘッダーを調べ、ブラウザがどのような圧縮をサポートしているか (ある場合) に関する情報を保存します。
2. ADC は要求をサーバに転送し、応答を受信します。
3. NetScaler ADC 圧縮エンジンは、ポリシーと照合することによって、サーバーの応答の圧縮性を調べます。
4. 応答が圧縮アクションに関連付けられたポリシーと一致し、クライアントブラウザーがアクションで指定された圧縮アルゴリズムをサポートしている場合、NetScaler ADC はアルゴリズムを適用し、圧縮された応答をクライアントブラウザーに送信します。
5. クライアントは、サポートされている圧縮アルゴリズムを適用して、応答を解凍します。

HTTP 圧縮を構成する

デフォルトでは、NetScaler ADC では圧縮が無効になっています。この機能を設定する前に、この機能を有効にする必要があります。この機能が有効な場合、ADC は圧縮ポリシーで指定されたサーバー要求を圧縮します。

CLI を使用して HTTP 圧縮を有効にするには

圧縮は HTTP および SSL サービスに対してのみ有効にできます。すべての HTTP および SSL サービスに適用するようにグローバルに有効にすることも、特定のサービスに対してのみ有効にすることもできます。

コマンドプロンプトで、次のコマンドのいずれかを入力して、圧縮をグローバルに、または特定のサービスに対して有効にします。

- `enable ns feature cmp`
  または
- `set service \<name\> -CMP YES`

GUI を使用して圧縮を構成するには

次のいずれかを行います:

圧縮をグローバルに有効にするには、[システム] > [設定] に移動し、[基本機能の設定] をクリックして [HTTP 圧縮] を選択します。

特定のサービスの圧縮を有効にするには、[トラフィック管理**] > [**負荷分散**] > [**サービス] に移動し、サービスを選択して [編集] をクリックします。[設定] グループで鉛筆アイコンをクリックし、[圧縮] を有効にします。

圧縮アクションの設定

圧縮アクションは、要求または応答が、アクションが関連付けられているポリシーのルール（式）に一致したときに実行するアクションを指定します。たとえば、特定のサーバーに送信される要求を識別する圧縮ポリシーを構成し、そのポリシーをサーバーの応答を圧縮するアクションに関連付けることができます。

4 つの組み込み圧縮アクションがあります。

- COMPRESS: GZIP アルゴリズムを使用して、GZIP のいずれか、または GZIP と DEFLATE の両方をサポートするブラウザーからのデータを圧縮します。DEFLATE アルゴリズムを使用して、DEFLATE アルゴリズムのみをサポートするブラウザーからのデータを圧縮します。ブラウザーがどちらのアルゴリズムもサポートしていない場合、ブラウザーの応答は圧縮されません。
- NOCOMPRESS: データを圧縮しません。
- GZIP: GZIP アルゴリズムを使用して、GZIP 圧縮をサポートするブラウザーのデータを圧縮します。ブラウザーが GZIP アルゴリズムをサポートしていない場合、ブラウザーの応答は圧縮されません。
- DEFLATE: DEFLATE アルゴリズムを使用して、DEFLATE アルゴリズムをサポートするブラウザーのデータを圧縮します。ブラウザーが DEFLATE アルゴリズムをサポートしていない場合、ブラウザーの応答は圧縮されません。アクションを作成したら、そのアクションを 1 つ以上の圧縮ポリシーに関連付けます。

コマンドプロンプトで、次のコマンドを入力して、圧縮アクションを作成します。

```
add cmp action <name> <cmpType> [-addVaryHeader <addVaryHeader> -
varyHeaderValue<string>]
```

CLI を使用して圧縮ポリシーを設定するには

圧縮ポリシーにはルールが含まれています。ルールは、NetScaler ADC アプライアンスが圧縮する必要があるトラフィックを識別できるようにする論理式です。

NetScaler ADC はサーバーから HTTP 応答を受信すると、組み込みの圧縮ポリシーとカスタム圧縮ポリシーを評価して、応答を圧縮するかどうか、圧縮する場合は適用する圧縮の種類を決定します。ポリシーに割り当てられた優先順位によって、ポリシーがリクエストに対して照合される順序が決まります。

コマンドプロンプトで次のコマンドを入力して、圧縮ポリシーを作成します。

```
add cmp policy <name> -rule <expression> -resAction <string>
```

GUI を使用して圧縮アクションを作成するには

[最適化] > [**HTTP** 圧縮] > [アクション] に移動し、[追加] をクリックして、HTTP 応答で実行する圧縮のタイプを指定する圧縮アクションを作成します。

圧縮ポリシーを構成する

圧縮ポリシーにはルールが含まれています。ルールは、NetScaler ADC アプライアンスが圧縮する必要があるトラフィックを識別できるようにする論理式です。

NetScaler ADC はサーバーから HTTP 応答を受信すると、組み込みの圧縮ポリシーとカスタム圧縮ポリシーを評価して、応答を圧縮するかどうか、圧縮する場合は適用する圧縮の種類を決定します。ポリシーに割り当てられた優先順位によって、ポリシーがリクエストに対して照合される順序が決まります。

次の表に、組み込みの HTTP 圧縮ポリシーを示します。これらのポリシーは、圧縮を有効にするとグローバルにアクティブ化されます。

組み込みのクラシックポリシーまたは詳細ポリシー	説明
ns_nocmp_mozilla_47、 ns_adv_nocmp_mozilla_47	Mozilla 4.7 ブラウザからリクエストが送信されたときに CSS ファイルの圧縮を防止します。
ns_cmp_mscss、ns_cmp_mscss	Microsoft Internet Explorer ブラウザーから要求が送信されたときに CSS ファイルを圧縮します。
ns_cmp_msapp、ns_cmp_msapp	Microsoft Office Word、Microsoft Office Excel、Microsoft Office PowerPoint のアプリケーションによって生成されたファイルを圧縮します。
ns_cmp_content_type, ns_adv_cmp_content_type	レスポンスに containsContent-type ヘッダーがあり、テキストが含まれている場合に、データを圧縮します。
ns_nocmp_xml_ie、ns_adv_nocmp_xml_ie	Microsoft Internet Explorer ブラウザーから要求が送信され、応答に Content-Type ヘッダーが含まれ、テキストまたは xml が含まれている場合に、圧縮を防止します。

圧縮ポリシーのバインド

圧縮ポリシーを有効にするには、NetScaler ADC を通過するすべてのトラフィックに適用されるようにグローバルにバインドするか、特定の仮想サーバーにバインドして、宛先がその仮想サーバーの VIP アドレスである要求にのみポリシーを適用する必要があります。

ポリシーをバインドするときは、そのポリシーにプライオリティを割り当てます。プライオリティによって、定義したポリシーが評価される順序が決まります。優先度は、任意の正の整数に設定できます。

CLI を使用して圧縮ポリシーをバインドするには

コマンドプロンプトで、次のいずれかのコマンドを入力して、圧縮ポリシーをグローバルに、または特定の仮想サーバーにバインドします。

- `bind cmp global <policyName> [-priority <positive_integer>] [-state (ENABLED|DISABLED)]…`
- `bind lb vserver <vserverName> -policyName <policyName> -type (Request|Response)-priority <positive_integer> )`

圧縮ポリシーをバインドする仮想サーバごとに、このコマンドを繰り返します。

GUI を使用して圧縮ポリシーをバインドするには

次のいずれかを行います:

グローバルレベルで [最適化] > [**HTTP** 圧縮] > [ポリシー] に移動し、[ポリシーマネージャ] をクリックし、関連するバインドポイントと接続タイプ（要求/応答）を指定して、必要なポリシーをバインドします。

仮想サーバレベル

負荷分散仮想サーバーの場合は、[トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、必要な仮想サーバーを選択し、[ポリシー] をクリックして、関連するポリシーをバインドします。

コンテンツスイッチ仮想サーバーの場合は、[トラフィック管理**] > [**コンテンツスイッチング**] > [**仮想サーバー] に移動し、必要な仮想サーバーを選択し、[ポリシー] をクリックして、関連するポリシーをバインドします。

最適なパフォーマンスを得るためのグローバル圧縮パラメータの設定

多くのユーザーはグローバル圧縮パラメータのデフォルト値を受け入れますが、これらの設定をカスタマイズすることで、より効果的な圧縮を提供できる場合があります。

> （注
> ）グローバル圧縮パラメータを設定した後は、アプライアンスをリブートする必要はありません。新しいフローに直ちに適用されます。

次の表では、NetScaler ADC で設定できる圧縮パラメータについて説明します。

圧縮パラメータ	説明
量子サイズ	サーバー応答を累積するために維持されるバッファのサイズ（KB 単位）。バッファサイズがこの値を超えると、応答は圧縮されます。たとえば、量子サイズを 50 KB に設定すると、NetScaler ADC はバッファのサイズが 50 KB を超えるとバッファの内容を圧縮します。最小値:1。最大値:63488 デフォルト:57344。
圧縮レベル	サーバー応答に適用する圧縮レベル。可能な値:[最高速度]、[最高の圧縮]、[最適]。
HTTP 応答の最小サイズ	圧縮される HTTP 応答の最小サイズ（バイト単位）。このパラメータで指定された値より小さいレスポンスは、圧縮されずに送信されます。
CPU 使用率で圧縮をバイパスする	NetScaler CPU 使用率（パーセンテージ）で、それ以上では圧縮が行われません。デフォルトは 100 です。
ポリシータイプ *	圧縮に使用されるポリシーのタイプ。指定可能な値: クラシック、高度なポリシー。デフォルト: クラシック。
サーバー側の圧縮を許可する	サーバーが圧縮データを NetScaler ADC に送信できるようにします。
プッシュパケットを圧縮する	TCP PUSH フラグの付いたパケットを受信したら、量子バッファがいっぱいになるのを待たずに、蓄積されたパケットをただちに圧縮します。

圧縮パラメータ	説明
外部キャッシュ	プライベートレスポンスディレクティブを発行して、レスポンスメッセージが単一のユーザーを対象としており、共有キャッシュまたはプロキシキャッシュによってキャッシュされてはならないことを示します。

GUI を使用して HTTP 圧縮を設定するには

次のいずれかを行います:

- 圧縮をグローバルに有効にするには、[システム] > [設定] に移動し、[基本機能の設定] をクリックして [**HTTP** 圧縮] を選択します。
- 特定のサービスの圧縮を有効にするには、[トラフィック管理] > [負荷分散] > [サービス**]** に移動し、サービスを選択して [編集] をクリックします。
- 「設定」グループで、鉛筆アイコンをクリックし、「圧縮」を有効にします。

GUI を使用して圧縮アクションを作成するには

[最適化] > [**HTTP** 圧縮] > [アクション] に移動し、[追加] をクリックして、圧縮アクションを作成して HTTP レスポンスで実行する圧縮の種類を指定します。

GUI を使用して圧縮ポリシーを作成するには

[最適化] > [**HTTP** 圧縮] > [ポリシー] に移動し、[追加] をクリックして、条件と実行する対応するアクションを指定して圧縮ポリシーを作成します。

圧縮構成の評価

圧縮統計は、ダッシュボードユーティリティまたは SNMP モニタで表示できます。ダッシュボードユーティリティは、サマリーおよび詳細な統計を表形式およびグラフィック形式で表示します。

オプションで、ポリシーベースの圧縮中にポリシーカウンタが増加する要求の数など、圧縮ポリシーの統計情報を表示することもできます。

> 注
>
> - 統計情報とグラフの詳細については、NetScaler ADC アプライアンスのダッシュボードのヘルプを参照してください。
> - SNMP の詳細については、「SNMP トピック」を参照してください。

CLI を使用して圧縮統計を表示するには

コマンドプロンプトで次のコマンドを入力して、圧縮統計情報を表示します。

1. 圧縮統計のサマリーを表示します。


```
stat cmp
```

> （注
> ）stat cmp policy コマンドは、高度なポリシー圧縮ポリシーの統計情報だけを表示します。

2. 圧縮ポリシーのヒット数と詳細を表示するには

```
show cmp policy \<name\>
```

3. 詳細な圧縮統計情報を表示するには

```
stat cmp -detail
```

ダッシュボードを使用して圧縮統計を表示するには、次の手順を実行します。

ダッシュボードユーティリティでは、次のタイプの圧縮統計を表示できます。

- [圧縮（Compression）] を選択して、圧縮統計のサマリーを表示します。
- プロトコルタイプ別の詳細な圧縮統計情報を表示するには、[Details] をクリックします。
- 圧縮機能によって処理された要求の割合を表示するには、[Graphical View] タブをクリックします。

SNMP を使用して圧縮統計情報を表示するには

SNMP ネットワーク管理アプリケーションを使用すると、次の圧縮統計情報を表示できます。

- 圧縮要求の数 (OID: 1.3.6.1.4.1.5951.4.1.1.50.1)
- 送信された圧縮バイト数（OID：1.3.6.1.4.1.5951.4.1.1.50.2）
- 受信した圧縮可能なバイト数（OID：1.3.6.1.4.1.5951.4.1.1.50.3）
- 送信された圧縮可能なパケットの数（OID：1.3.6.1.4.1.5951.4.1.1.50.4）
- 受信された圧縮可能なパケットの数（OID：1.3.6.1.4.1.5951.4.1.1.50.5）
- 圧縮可能なデータの受信データと送信された圧縮データの比率 (OID: 1.3.6.1.4.1.5951.4.1.4.1.50.6)
- 送信されたデータ合計に対する受信データの合計の比率（OID：1.3.6.1.4.1.5951.4.1.1.50.7）

GUI を使用して圧縮統計をさらに表示するには

1. HTTP 圧縮統計情報を表示するには、次の手順を実行します。

[最適化] > [**HTTP** 圧縮] に移動し、[統計] をクリックします。

1. 圧縮ポリシーの統計情報を表示します。

[最適化] > [**HTTP** 圧縮] > [ポリシー] に移動し、ポリシーを選択して [統計] をクリックします。

1. 圧縮ポリシーラベルの統計情報を表示するには
2. [最適化] > [**HTTP** 圧縮] > [ポリシー] に移動し、ポリシーラベルを選択し、[統計] をクリックします。

HTTP 圧縮のオフロード

サーバーで圧縮を実行すると、サーバーのパフォーマンスに影響する可能性があります。Web サーバーの前に配置され、HTTP 圧縮用に構成された NetScaler ADC は、静的コンテンツと動的コンテンツの両方の圧縮をオフロードし、サーバーの CPU サイクルとリソースを節約します。

Web サーバーから圧縮をオフロードするには、次の 2 つの方法があります。

Web サーバーで圧縮を無効にし、NetScaler ADC 圧縮機能をグローバルレベルで有効にし、圧縮用のサービスを構成します。

Web サーバーで圧縮機能を有効にしたままにし、すべての HTTP クライアント要求から「Accept Encoding」ヘッダーを削除するように NetScaler ADC アプライアンスを構成します。その後、サーバーは圧縮されていないレスポンスを送信します。NetScaler ADC は、サーバー応答をクライアントに送信する前に圧縮します。

> 注:
> サーバーがすべての応答を自動的に圧縮する場合、2 番目のオプションは機能しません。NetScaler ADC は、すでに圧縮されている応答を圧縮しようとしません。

Servercmp パラメーターにより、NetScaler ADC アプライアンスはオフロード HTTP 圧縮を処理できます。デフォルトでは、このパラメータは、サーバーが圧縮データを NetScaler ADC アプライアンスに送信するためにオンに設定されています。HTTP 圧縮をオフロードするには、servercmp パラメーターを OFF に設定する必要があります。コマンドプロンプトで、次のコマンドを入力します。

```
set service <service name> -CMP YES
```

圧縮を有効にするサービスごとに、このコマンドを繰り返します。

```
show service <service name>
```

サービスごとにこのコマンドを繰り返して、圧縮が有効になっていることを確認します。

```
Save config
```

```
set cmp parameter -serverCmp OFF
```

> 注:
>
> Servercmpパラメータがオンになっており、アプライアンスがサーバーから圧縮された応答を受信した場合、アプライアンスはデータをそれ以上圧縮しません。代わりに、圧縮された応答をクライアントに転送します。

統合キャッシング

December 8, 2023

統合キャッシュは、NetScaler アプライアンスのメモリ内ストレージを提供し、オリジンサーバーへの往復を必要とせずにユーザーに Web コンテンツを提供します。静的コンテンツの場合、統合キャッシュの初期設定はほとんど必要ありません。統合キャッシュ機能を有効にし、基本的なセットアップ（たとえば、キャッシュが使用できる NetScaler アプライアンスのメモリの量の決定）を実行すると、統合キャッシュは組み込みポリシーを使用して、単純な Web ページやイメージファイルなどの特定のタイプの静的コンテンツを格納および提供します。統合キャッシュを構成して、Web サーバーおよびアプリケーションサーバーによってキャッシュ不可としてマークされた動的コンテンツ（データベースレコードや株価など）を保存および提供することもできます。

> 注:
>
> 統合キャッシュという用語は、AppCache と同じ意味で使用できます。機能の観点から、両方の用語は同じ意味であることに注意してください。

要求または応答が、組み込みポリシーまたは作成したポリシーで指定されたルール (論理式) と一致する場合。NetScaler アプライアンスは、ポリシーに関連するアクションを実行します。デフォルトでは、すべてのポリシーがキャッシュされたオブジェクトを保存し、デフォルトコンテンツグループから取得します。さまざまなタイプのコンテンツ用に独自のコンテンツグループを作成できます。

アプライアンスがコンテンツグループ内のキャッシュされたオブジェクトを検索できるようにするには、セレクターを設定できます。セレクターは、キャッシュされたオブジェクトを式と照合するか、コンテンツグループ内のオブジェクトを検索するためのパラメーターを指定できます。Citrix が推奨するセレクターを使用する場合は、最初にセレクターを構成して、コンテンツグループを構成するときにセレクターを指定できるようにします。次に、追加するコンテンツグループを設定します。これにより、ポリシーを構成するときに使用できるようになります。初期設定を完了するには、各ポリシーをグローバルバインドポイントまたは仮想サーバーにバインドしてポリシーバンクを作成します。または、他のポリシーバンクから呼び出せるラベルをバインドすることもできます。

統合キャッシュは、有効期限が切れる前にキャッシュされたオブジェクトを事前にロードすることで改善できます。キャッシュされたデータの処理を管理するために、レスポンスに挿入されるキャッシュ関連のヘッダーを設定できます。統合キャッシュは、他のキャッシュサーバーのフォワードプロキシとしても機能します。

> 注:
>
> 統合キャッシュには、HTTP リクエストとレスポンスにある程度精通している必要があります。HTTP データの構造については、次の URL にある「ライブ *HTTP* ヘッダー」を参照してください。"<http://livehttpheaders.mozdev.org/>."

統合キャッシュの仕組み

統合キャッシュは、NetScaler アプライアンスを経由する HTTP および SQL 要求を監視し、要求を保存されたポリシーと比較します。結果に応じて、統合キャッシュ機能はキャッシュで応答を検索するか、要求をオリジンサーバーに転送します。HTTP リクエストの場合、統合キャッシュは、シングルバイトレンジおよびマルチパートバイトレンジのリクエストに応答して、キャッシュ内のコンテンツの一部として機能します。

クライアントが圧縮コンテンツを受け入れる場合、キャッシュされたデータは圧縮されます。コンテンツグループの有効期限を設定したり、コンテンツグループのエントリを選択的に期限切れにしたりできます。

次の表に示すように、統合キャッシュから提供されるデータはヒットし、オリジンから提供されるデータはキャッシュミスです。

取引タイプ	仕様
キャッシュヒット	NetScaler アプライアンスがキャッシュから処理する応答には、画像ファイルや静的 Web ページなどの静的オブジェクト、200 個の OK ページ、203 個の非正規応答ページ、300 個の複数選択ページ、301 個の永久移動ページ、302 個の見つかったページ、304 個の未変更ページ、これらの応答は肯定的な応答と呼ばれます。 NetScaler アプライアンスは、次の否定的な応答もキャッシュします：307 の一時リダイレクトページ、403 ページの禁止ページ、404 ページの見つからないページ、410 ページの消失ページ。パフォーマンスをさらに向上させるには、より多くの種類のコンテンツをキャッシュするように NetScaler アプライアンスを構成できます。
保存可能なキャッシュミス	保存可能なキャッシュミスの場合、NetScaler アプライアンスはオリジンサーバーからの応答を取得し、その応答をキャッシュに保存してからクライアントに提供します。
保存できないキャッシュミス	保存できないキャッシュミスはキャッシュには不適切です。デフォルトでは、201、202、204、205、206 ステータスコード、403、404、410、5xx ステータスコードを除くすべての 4xx コードを含む応答は、保存できないキャッシュミスです。

> 注:
>
> 動的キャッシュをアプリケーションインフラストラクチャに統合するには、NITRO API を使用してキャッシュコマンドをリモートで実行します。たとえば、データベーステーブルが更新されたときにキャッシュされた応答を期限切れにするトリガーを設定できます。

キャッシュされた応答をオリジンサーバー上のデータと確実に同期させるには、有効期限方法を設定します。NetScaler アプライアンスは、期限切れの応答と一致する要求を受信すると、オリジンサーバーからの応答を更新します。

> 注:
>
> NetScaler アプライアンスと 1 つ以上のバックエンドサーバーの時刻を同期することをお勧めします。

ダイナミックキャッシュの仕組み

動的キャッシュは、パラメータと値のペア、文字列、文字列パターン、またはその他のデータに基づいて HTTP 要求と応答を評価します。たとえば、ユーザーがバグ報告アプリケーションで Bug 31231 を検索したとします。ブラウザはユーザーに代わって次のリクエストを送信します。

```
1     GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&
          Template=view&TableId=1000
2
3     Host: mycompany.net
4
5     User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9)
          Gecko/2008052906 Firefox/3.0
6
7     Accept: text/html,application/xhtml+xml,application/xml;q
          =0.9,\*/\*;q=0.8
8
9     Accept-Language: en-us,en;q=0.5
10 <!--NeedCopy-->
```

この例では、このバグ報告アプリケーションの GET リクエストには常に次のパラメータが含まれています。

- 問題ページ
- レコード ID
- テンプレート
- テーブル ID

GET リクエストはデータを更新または変更しないため、キャッシュポリシーとセレクターでこれらのパラメーターを次のように設定できます。

- HTTP リクエストで mybugreportingsystem という文字列と GET メソッドを検索するキャッシュポリシーを設定します。このポリシーは、一致するリクエストをバグのコンテンツグループに転送します。
- バグのコンテンツグループでは、IssuePage、RecordID など、さまざまなパラメーターと値のペアに一致する`hit`セレクターを設定します。

> 注
>
> ブラウザは、1 つのユーザーアクションに基づいて複数の GET リクエストを送信できます。以下は、ユーザーがバグ ID に基づいてバグを検索したときにブラウザが発行する 3 つの個別の GET リクエストのシリーズです。

```
1     GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&
          Template=view&TableId=1000
2
```



```
3      GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=
          viewbtns&RecordId=31231&TableId=1000
4
5      GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=
          viewbody&RecordId=31231&tableid=1000
6  <!--NeedCopy-->
```

これらの要求を満たすために、複数の応答がユーザーのブラウザーに送信され、ユーザーに表示される Web ページは応答のアセンブリです。

ユーザーがバグレポートを更新した場合、キャッシュ内の対応する応答をオリジンサーバーからのデータで更新する必要があります。バグレポートアプリケーションは、ユーザがバグレポートを更新したときに HTTP POST リクエストを発行します。この例では、POST リクエストがキャッシュ内の無効化をトリガーするように次のように構成します。

- mybugreportingsystem という文字列と POST HTTP リクエストメソッドを検索し、一致するリクエストをバグレポート用のコンテンツグループに送るリクエストをリクエスト時無効化ポリシーです。
- RecordID パラメーターに基づいてキャッシュされたコンテンツを期限切れにするバグレポート用のコンテンツグループの無効化セレクター。このパラメータはすべてのレスポンスに表示されるので、無効化セレクターはキャッシュ内のすべての関連項目を期限切れにすることができます。

次の抜粋は、バグレポートのサンプルを更新する POST リクエストを示しています。

```
1      POST /mybugreportingsystem/mybugreport.dll?TransitionForm HTTP/1.1\
          r\n
2
3      User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
          Opera 7.23 [en]\r\n
4
5      Host: mybugreportingsystem\r\n
6
7      Cookie:ttSearch.134=%23options%3Afalse%23active%23owner%3Afalse%23
          unowned%3Afalse%23submitter%3Afalse%23incsub%3Atrue;
8
9      Cookie2: $Version=1\r\n
10
11     . . .
12
13     \r\n
14
15     ProjectId=2&RecordId=31231&TableId=1000&TransitionId=1&Action=
          Update&CopyProjectId=0&ReloadForm=0&State=&RecordLockId=49873+
          issues+in+HTTP&F43. . .
16 <!--NeedCopy-->
```

NetScaler アプライアンスはこの要求を受け取ると、次の処理を行います。

- リクエストを無効化ポリシーと照合します。
- ポリシーで指定されたコンテンツグループを検索します。

- このコンテンツグループに無効化セレクターを適用し、recordId=31231 に一致するすべての応答を期限切れにします。

ユーザーがこのバグレポートの新しいリクエストを発行すると、NetScaler アプライアンスはオリジンサーバーにアクセスして、レポートインスタンスに関連するすべての応答の更新されたコピーを取得します。応答はコンテンツグループに保存され、ユーザーのブラウザに送信されます。ブラウザはレポートを再構成して表示します。

統合キャッシュの設定

統合キャッシュを使用するには、ライセンスをインストールして機能を有効にする必要があります。統合キャッシュを有効にすると、NetScaler® アプライアンスは組み込みポリシーで指定された静的オブジェクトを自動的にキャッシュし、キャッシュの動作に関する統計を生成します。(組み込みポリシーには、ポリシー名の最初の位置にアンダースコアが付いています)。

組み込みポリシーが状況に適している場合でも、グローバル属性を変更したい場合があります。たとえば、統合キャッシュに割り当てられる NetScaler アプライアンスのメモリ容量を変更できます。

設定を変更する前にキャッシュ操作を確認する場合は、「キャッシュされたオブジェクトとキャッシュ統計の表示」を参照してください。

> 注:
>
> NetScaler キャッシュは、アプライアンスを再起動すると削除されるメモリ内ストアです。

統合キャッシュライセンスをインストールするには

- 統合キャッシュライセンスが必要です。
- Citrix からライセンスコードを取得し、コマンドラインインターフェイスにアクセスしてログインします。

コマンドラインインターフェイスで、ライセンスファイルを/nsconfig/licenseフォルダーにコピーします。

- 次のコマンドを使用して、NetScaler アプライアンスを再起動します。

```
reboot
```

統合キャッシュを有効にするには:
統合キャッシュを有効にすると、NetScaler アプライアンスはサーバー応答のキャッシュを開始します。ポリシーまたはコンテンツグループを設定していない場合、組み込みポリシーは、キャッシュされたオブジェクトをデフォルトコンテンツグループに保存します。

コマンドプロンプトで、次のいずれかのコマンドを入力して、統合キャッシュを有効または無効にします。

```
enable ns feature IC
```


キャッシュのグローバル属性を構成する

グローバル属性は、すべてのキャッシュされたデータに適用されます。統合キャッシュに割り当てられる NetScaler メモリの量は、ヘッダー挿入で指定できます。キャッシュされたオブジェクトを提供する必要があることを確認するための基準。キャッシュで許可される POST 本文の最大長、HTTP GET リクエストのポリシー評価をバイパスするかどうか、ポリシーを評価できない場合に行うアクション。

キャッシュメモリ容量は、ハードウェアアプライアンスのメモリによってのみ制限されます。また、nCore NetScaler アプライアンスのパケットエンジン（すべての着信 TCP 要求の中央配信ハブ）は、nCore NetScaler アプライアンスの他のパケットエンジンによってキャッシュされたオブジェクトを認識します。

> 注:
>
> デフォルトのグローバルメモリ制限が 0 に設定され、統合キャッシュ（IC）機能が有効になっている場合、アプライアンスはオブジェクトをキャッシュしません。キャッシュの場合は、グローバルメモリ制限を明示的に設定する必要があります。ただし、「認証、承認、監査パラメータの設定 enableStaticPageCaching」オプションを有効にすると、アプライアンスにはデフォルトのメモリがいくらか設定されます。このメモリは大きなオブジェクトをキャッシュするには不十分であるため、IC にはより高いメモリ制限を割り当てる必要があります。これを実行するには、「キャッシュパラメータの設定—MemLimit」コマンドを設定します。新しい設定は、構成を保存してアプライアンスを再起動した後にのみ適用されます。

オブジェクトをキャッシュするために設定されたグローバルメモリ制限を変更できます。ただし、グローバルメモリ制限を既存の値よりも低い値（たとえば 10 GB から 4 GB に）更新すると、アプライアンスは引き続きメモリ制限を使用します。

つまり、統合キャッシュの制限はある値に設定されていても、実際に使用される制限はもっと大きくなる可能性があります。ただし、この余分なメモリは、オブジェクトがキャッシュから削除されると解放されます。

show cache parameter コマンドの出力には、設定値（メモリ使用量の上限）と実際に使用されている値（メモリ使用量の上限（アクティブ値））が表示されます。

コマンドプロンプトで入力します:

```
1     set cache parameter [-memLimit <MBytes>] [-via <string>] [-
          verifyUsing <criterion>] [-maxPostLen <positiveInteger>] [-
          prefetchMaxPending <positiveInteger>] [-enableBypass(YES|NO)] [-
          undefAction (NOCACHE|RESET)]
2 <!--NeedCopy-->
```

NetScaler GUI による統合キャッシュの有効化

[システム] **>** [設定] に移動し、[基本機能の設定] をクリックし、[統合キャッシュ] を選択します。

NetScaler GUI を使用してキャッシュのグローバル設定を構成する

[最適化] > [統合キャッシュ] に移動し、[キャッシュ設定の変更] をクリックして、キャッシュのグローバル設定を構成します。

統合キャッシュの組み込みコンテンツグループ、パターンセット、およびポリシーを設定する

NetScaler アプライアンスには、コンテンツのキャッシュに使用できる統合キャッシュ構成が組み込まれています。構成は、ctx_cg_poc というコンテンツグループ、ctx_file_extensions というパターンセット、および統合キャッシュポリシーのセットで構成されています。コンテンツグループ ctx_cg_poc では、500 KB 以下のオブジェクトのみがキャッシュされます。コンテンツは 86000 秒間キャッシュされ、コンテンツグループのメモリ制限は 512 MB です。パターンセットは、ファイルタイプマッチングのための一般的な拡張子のインデックス付き配列です。

次の表は、組み込みの統合キャッシュポリシーの一覧です。デフォルトでは、ポリシーはどのバインドポイントにもバインドされません。NetScaler アプライアンスにポリシーと照合してトラフィックを評価させる場合は、ポリシーをバインドポイントにバインドする必要があります。ポリシーは ctx_cg_poc コンテンツグループのオブジェクトをキャッシュします。

統合キャッシュポリシー名	ポリシールール
_cacheVPNStaticObjects	HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).CONTAINS_IN “ctx_file_extensions”).BETWEEN(101,150)
_cacheTCPVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(“.css”)
_cacheOCVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(“.pdf”)
_cacheWFStaticObjects	HTTP.REQ.URL.ENDSWITH(“.js”)
_mayNoCacheReq	HTTP.RES.HEADER(“Content-Type”).CONTAINS(“application/x-javascript”)
_noCacheRest	TRUE

フラッシュキャッシュ設定

キャッシュグループ、キャッシュグループ、またはキャッシュオブジェクトロケータをフラッシュできます。キャッシュオブジェクトをフラッシュするコマンドは次のとおりです。

コマンドプロンプトで入力します:

```
flush cache contentgroup all
```

例


```
        0x00000089bae000000004 DEFAULT GET  //1.1.1.1:80/html/index.
           html?name=hello
        0x00000089bae000000005 DEFAULT GET  //1.1.1.1:80/html/index.
           html?name=hi

        Flush cache contentGroup all
        done

`flush cache contentgroup <content group name>`
<!--NeedCopy-->
```

例:

```
        0x00000089bae000000004 DEFAULT GET  //1.1.1.1:80/html/index.
           html?name=hello
        0x00000089bae000000005 DEFAULT GET  //1.1.1.1:80/html/index.
           html?name=hi

        Flush cache ob -| 0x00000089bae000000004
        done

`flush cache object (-locator <positive_integer> | (-url <URL> (-host <
   string> [-port <port>] [-groupName <string>] [-httpMethod ( GET |
   POST )])))`
<!--NeedCopy-->
```

例:

```
        0x00000089bae000000006 DEFAULT GET //1.1.1.1:80/html/index.html

        flush cache ob -URL /html/index.html -host 1.1.1.1 -groupName
           DEFAULT
        done
<!--NeedCopy-->
```

NetScaler GUI を使用してキャッシュ構成をフラッシュする

NetScaler GUI を使用してキャッシュフラッシュを設定する手順を完了します

1. [最適化] > [コンテンツグループ] に移動します。
2. コンテンツグループ詳細ペインで、「追加」をクリックします。
3. 「キャッシュコンテンツグループの作成」ページで、「その他」タブで次のパラメータを設定します。
    a) キャッシュをフラッシュします。このチェックボックスを選択すると、キャッシュオブジェクトがフラッシュされます。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

さまざまなシナリオに対応した統合キャッシュの設定

次のセクションでは、さまざまなシナリオにおける NetScaler アプライアンスの統合キャッシュの構成について説明します。

NetScaler 9.2 リリース以降、統合キャッシュにはキャッシュ用のメモリが多くなっています。統合キャッシュメモリは、ハードウェアアプライアンスで使用可能なメモリによってのみ制限されます。使用可能なメモリの最大 50% を統合キャッシュ機能に割り当てることができます。

CLI を使用してキャッシュのメモリ割り当てを設定するには

コマンドプロンプトで入力します:

```
set cache parameter -memlimit <value>
```

> 注:
>
> 統合キャッシュのデフォルトのグローバルメモリ制限はゼロです。そのため、統合キャッシュ機能を有効にしても、NetScaler アプライアンスはグローバルメモリ制限が明示的に設定されるまでオブジェクトをキャッシュしません。

次のセクションでは、さまざまなシナリオで統合キャッシュを構成する方法について説明します。

> 注:
>
> NetScaler アプライアンスのメモリ制限は、アプライアンスの起動時に識別されます。そのため、メモリ制限を変更した場合は、アプライアンスを再起動してその変更をパケットエンジン全体に適用する必要があります。

統合キャッシュが有効で、キャッシュメモリの制限がゼロ以外に設定されている

アプライアンスを起動し、統合キャッシュ機能が有効になり、グローバルメモリ制限が正の数に設定されているシナリオを考えてみましょう。以前に設定したメモリは、ブートプロセス中に統合キャッシュ機能に割り当てられます。アプライアンスで使用可能なメモリによっては、メモリ制限を別の値に変更する必要がある場合があります。

CLI を使用して設定する

1. キャッシュパラメータを表示します

```
> show cache parameter
        Integrated cache global configuration:
        Memory usage limit: 500 MBytes
        Memory usage limit (active value): 500 MBytes
        Maximum value for Memory usage limit: 843 MBytes
        Via header: NS-CACHE-9.3: 18
        Verify cached object using: HOSTNAME_AND_IP
        Max POST body size to accumulate: 0 bytes
        Current outstanding prefetches: 0
        Max outstanding prefetches: 4294967295
```



```
11                  Treat NOCACHE policies as BYPASS policies: YES
12                  Global Undef Action: NOCACHE
13  <!--NeedCopy-->
```

1. ゼロ以外のメモリ制限を設定する

```
set cache parameter -memlimit 600
```

> 注:
>
> 上記のコマンドは、次の警告メッセージを表示します。警告: 新しい統合キャッシュのメモリ制限を使用するには、構成を保存して、**NetScaler** アプライアンスを再始動します。

1. 構成を保存します

```
save config
```

1. シェルプロンプトから、次のコマンドを実行して構成ファイルで確認します。

```
root@ns# cat /nsconfig/ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 600 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 0 -enableBypass YES -undefAction NOCACHE
```

1. アプライアンスを再起動します

```
root@ns# reboot
```

1. メモリ制限の新しい値を確認します

```
1           > show cache parameter
2                   Integrated cache global configuration:
3                   Memory usage limit: 600 MBytes
4                   Memory usage limit (active value): 600 MBytes
5                   Maximum value for Memory usage limit: 843 MBytes
6                   Via header: NS-CACHE-9.3: 18
7                   Verify cached object using: HOSTNAME_AND_IP
8                   Max POST body size to accumulate: 0 bytes
9                   Current outstanding prefetches: 0
10                  Max outstanding prefetches: 4294967295
11                  Treat NOCACHE policies as BYPASS policies: YES
12                  Global Undef Action: NOCACHE
13  <!--NeedCopy-->
```

すべてのパケットエンジンが正常に起動すると、統合キャッシュ機能が構成済みのメモリをネゴシエートします。アプライアンスが構成済みのメモリを使用できない場合、メモリはそれに応じて割り当てられます。使用可能なメモリが割り当てたメモリよりも少ない場合、アプライアンスはそれよりも少ない数を推奨します。統合キャッシュ機能では、アクティブ値と同じ値を使用します。

統合キャッシュは無効で、キャッシュメモリの制限はゼロ以外に設定されています

このシナリオでは、アプライアンスを起動すると、統合キャッシュ機能が無効になり、グローバルメモリ制限が正の数に設定されます。そのため、ブートプロセス中に統合キャッシュにメモリが割り当てられることはありません。

CLI を使用して設定する

1. キャッシュパラメータを表示します

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 600 MBytes
4                  Maximum value for Memory usage limit: 843 MBytes
5                  Via header: NS-CACHE-9.3: 18
6                  Verify cached object using: HOSTNAME_AND_IP
7                  Max POST body size to accumulate: 0 bytes
8                  Current outstanding prefetches: 0
9                  Max outstanding prefetches: 4294967295
10                 Treat NOCACHE policies as BYPASS policies: YES
11                 Global Undef Action: NOCACHE
12 <!--NeedCopy-->
```

1. 新しいメモリ制限を設定する

```
set cache parameter -memlimit 500
```

> 注:
>
> 上記のコマンドを実行すると、次の警告メッセージが表示されます。警告**:** 機能が有効になっていません **[IC]**。

1. 構成を保存します

```
save config
```

1. シェルプロンプトから、次のコマンドを実行して構成ファイルで確認します

```
root@ns# cat /nsconfig/ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 500 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 0 -enableBypass YES -undefAction NOCACHE
```

1. メモリ制限の新しい値を確認します

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 500 MBytes
4                  Maximum value for Memory usage limit: 843 MBytes
```



```
                  Via header: NS-CACHE-9.3: 18
                  Verify cached object using: HOSTNAME_AND_IP
                  Max POST body size to accumulate: 0 bytes
                  Current outstanding prefetches: 0
                  Max outstanding prefetches: 4294967295
                  Treat NOCACHE policies as BYPASS policies: YES
                  Global Undef Action: NOCACHE
<!--NeedCopy-->
```

1. 統合キャッシュ機能を有効にする

```
enable ns feature IC
```

1. メモリ制限の新しい値を確認します

```
         > show cache parameter
                  Integrated cache global configuration:
                  Memory usage limit: 500 Mbytes
                  Memory usage limit (active value): 500 Mbytes
                  Maximum value for Memory usage limit: 843 MBytes
                  Via header: NS-CACHE-9.3: 18
                  Verify cached object using: HOSTNAME_AND_IP
                  Max POST body size to accumulate: 0 bytes
                  Current outstanding prefetches: 0
                  Max outstanding prefetches: 4294967295
                  Treat NOCACHE policies as BYPASS policies: YES
                  Global Undef Action: NOCACHE
<!--NeedCopy-->
```

> 注:
>
> 500MB のメモリが統合キャッシュ機能に割り当てられます。

1. 構成を保存して、アプライアンスの再起動時にメモリが自動的に機能に割り当てられるようにします。

統合キャッシュが有効で、キャッシュメモリがゼロに設定されている

このシナリオでは、アプライアンスを起動すると、統合キャッシュ機能が有効になり、グローバルメモリ制限がゼロに設定されます。そのため、ブートプロセス中に統合キャッシュにメモリが割り当てられることはありません。

CLI を使用して設定する

1. シェルプロンプトから ns.conf ファイルに設定されているメモリ制限を確認します

```
root@ns# cat ns.conf | grep memLimit
```

1. メモリ制限を変更する


```
set cache parameter -memLimit 0 -via NS-CACHE-9.3: 18 -verifyUsing
 HOSTNAME_AND_IP -maxPostLen 4096 -enableBypass YES -undefAction
NOCACHE
```

1. メモリ制限の値を確認します

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 0 Mbytes
4                  Maximum value for Memory usage limit: 843 MBytes
5                  Via header: NS-CACHE-9.3: 18
6                  Verify cached object using: HOSTNAME_AND_IP
7                  Max POST body size to accumulate: 0 bytes
8                  Current outstanding prefetches: 0
9                  Max outstanding prefetches: 4294967295
10                 Treat NOCACHE policies as BYPASS policies: YES
11                 Global Undef Action: NOCACHE
12 <!--NeedCopy-->
```

> 注:
>
> メモリ制限は 0MB に設定されており、統合キャッシュ機能にメモリは割り当てられていません。

1. 統合キャッシュ機能がオブジェクトを確実にキャッシュするようにメモリ制限を設定します

```
set cache parameter -memLimit 600
```

上記のコマンドを実行すると、アプライアンスは統合キャッシュ機能のメモリをネゴシエートし、使用可能なメモリが機能に割り当てられます。その結果、アプライアンスはアプライアンスを再起動せずにオブジェクトをキャッシュします。

1. メモリ制限の値を確認します

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 600 Mbytes
4                  Memory usage limit (active value): 600 Mbytes
5                  Maximum value for Memory usage limit: 843 MBytes
6                  Via header: NS-CACHE-9.3:
7                  Verify cached object using: HOSTNAME_AND_IP
8                  Max POST body size to accumulate: 0 bytes
9                  Current outstanding prefetches: 0
10                 Max outstanding prefetches: 4294967295
11                 Treat NOCACHE policies as BYPASS policies: YES
12                 Global Undef Action: NOCACHE
13 <!--NeedCopy-->
```

> 注:
>
> 600MB のメモリが統合キャッシュ機能に割り当てられます。

1. 構成を保存します。アプライアンスの再起動時に、メモリが自動的に機能に割り当てられるようにしてください。
2. シェルプロンプトから ns.conf ファイルに設定されているメモリ制限を確認します

```
root@ns# cat /nsconfig/ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 600 -via NS-CACHE-9.3: -verifyUsing
 HOSTNAME_AND_IP -maxPostLen 4096 -enableBypass YES -undefAction
NOCACHE
```

統合キャッシュが無効になり、キャッシュメモリがゼロに設定されます

このシナリオでは、アプライアンスを起動すると、統合キャッシュ機能が無効になり、グローバルメモリ制限がゼロに設定されます。そのため、ブートプロセス中に統合キャッシュにメモリが割り当てられることはありません。

CLI を使用して設定する

1. シェルプロンプトから ns.conf ファイルに設定されているメモリ制限を確認します

```
root@ns# cat /nsconfig/ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 0 -via NS-CACHE-9.3: 18 -verifyUsing
 HOSTNAME_AND_IP -maxPostLen 4096 -enableBypass YES -undefAction
NOCACHE
```

1. メモリ制限の値を確認します

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 0 Mbytes
4                  Maximum value for Memory usage limit: 843 MBytes
5                  Via header: NS-CACHE-9.3: 18
6                  Verify cached object using: HOSTNAME_AND_IP
7                  Max POST body size to accumulate: 0 bytes
8                  Current outstanding prefetches: 0
9                  Max outstanding prefetches: 4294967295
10                 Treat NOCACHE policies as BYPASS policies: YES
11                 Global Undef Action: NOCACHE
12 <!--NeedCopy-->
```


> 注:
>
> メモリ制限は 0MB に設定されており、統合キャッシュ機能にメモリは割り当てられていません。また、キャッシュ設定コマンドを実行すると、次の警告メッセージが表示されます。警告**:** 機能が有効になっていません **[IC]**。

1. 統合キャッシュ機能を有効にする

```
enable ns feature IC
```

> 注:
>
> この段階で、統合キャッシュ機能を有効にすると、アプライアンスはその機能にメモリを割り当てません。その結果、オブジェクトはメモリにキャッシュされません。また、キャッシュ設定コマンドを実行すると、「IC 用のメモリが設定されていません」という警告メッセージが表示されます 。**set cache** パラメータコマンドを使用してメモリ制限を設定します。

1. 統合キャッシュ機能がオブジェクトを確実にキャッシュするようにメモリ制限を設定します

```
set cache parameter -memLimit 500
```

上記のコマンドを実行すると、アプライアンスは統合キャッシュ機能のメモリをネゴシエートし、使用可能なメモリが機能に割り当てられます。その結果、アプライアンスを再起動せずに、アプライアンスがオブジェクトをキャッシュします。

> 注:
>
> この機能を有効にしてメモリ制限を設定する順序は重要です。この機能を有効にする前にメモリ制限を設定すると、次の警告メッセージが表示されます。警告**:** 機能が有効化されていません **[IC]**。

1. メモリ制限の値を確認します

```
1           > show cache parameter
2                   Integrated cache global configuration:
3                   Memory usage limit: 500 Mbytes
4                   Memory usage limit (active value): 500 Mbytes
5                   Maximum value for Memory usage limit: 843 MBytes
6                   Via header: NS-CACHE-9.3:
7                   Verify cached object using: HOSTNAME_AND_IP
8                   Max POST body size to accumulate: 0 bytes
9                   Current outstanding prefetches: 0
10                  Max outstanding prefetches: 4294967295
11                  Treat NOCACHE policies as BYPASS policies: YES
12                  Global Undef Action: NOCACHE
13  <!--NeedCopy-->
```

> 注:
>
> 500MB のメモリが統合キャッシュ機能に割り当てられます。

1. 構成を保存します


```
save config
```

1. シェルプロンプトから ns.conf ファイルに設定されているメモリ制限を確認します

```
root@ns# cat /nsconfig/ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 500 -via NS-CACHE-9.3: 18 -verifyUsing
 HOSTNAME_AND_IP -maxPostLen 4096 -enableBypass YES -undefAction
NOCACHE
```

セレクタと基本的なコンテンツグループを構成する

August 15, 2023

セレクターを設定してコンテンツグループに適用できます。セレクターを 1 つ以上のコンテンツグループに追加する場合、そのセレクターをキャッシュリクエストの識別に使用するのか、無効化する (期限切れの) キャッシュオブジェクトの識別に使用するのかを指定します。セレクターはオプションです。または、`hit` パラメータと無効化パラメータを使用するようにコンテンツグループを設定することもできます。ただし、Citrix ではセレクターを構成することをお勧めします。

セレクターを設定するか、代わりにパラメーターを使用することを決定したら、基本的なコンテンツグループを設定する準備が整います。基本コンテンツグループを作成したら、キャッシュからオブジェクトを期限切れにする方法を決定し、キャッシュの有効期限を設定する必要があります。「キャッシュパフォーマンスの向上」および「Cookie、ヘッダー、およびポーリングの設定」の説明に従って、キャッシュをさらに変更できますが、最初にキャッシュポリシーを構成する必要がある場合があります。

> 注
>
> コンテンツグループのパラメータとセレクタはリクエスト時にのみ使用され、通常は MAY_CACHE または MAY_NOCACHE アクションを使用するポリシーに関連付けます。

セレクターの利点

セレクターは、コンテンツグループ内の特定のオブジェクトを検索するフィルターです。セレクターを構成しない場合、Citrix® ADC アプライアンスはコンテンツグループ内で完全に一致するものを探します。これにより、コンテンツグループに同じオブジェクトの複数のコピーが存在する可能性があります。たとえば、セレクターのないコンテンツグループには、host1.domain.com\ mypage.htm、host2.domain.com\ mypage.htm、host3.domain.com\ mypage.htm の URL を保存する必要がある場合があります。これに対し、セレクターは URL (mypage.html、http.req.url という表現を使用) とドメイン (.com、http.req.hostname.domain という表現を使用) のみを照合できるため、同じ URL でリクエストを処理できます。

セレクター式を使用すると、パラメータを簡単に照合できます (たとえば、いくつかのクエリ文字列パラメータとその値に一致するオブジェクトを検索する場合など)。セレクター式では、ブール論理、算術演算、属性の組み合わせを使用してオブジェクト (URL ステムのセグメント、クエリ文字列、POST リクエスト本文の文字列、HTTP ヘッダーの文字列、Cookie など) を識別できます。セレクターはプログラム関数を実行してリクエスト内の情報を分析することもできます。例えば、セレクタは、POST 本文内のテキストを抽出し、テキストをリストに変換し、リストから特定の項目を抽出することができます。

式および式で指定できる内容の詳細については、「ポリシーと式」を参照してください。

セレクタの代わりにパラメータを使用する

Citrix ではコンテンツグループでセレクターを使用することを推奨していますが、`hit` 代わりにパラメーターと無効化パラメーターを構成できます。たとえば、バグレポート用のコンテンツグループに BugID、Issuer、Assignee の 3 `hit` つのパラメーターを設定するとします。リクエストに BugID=456、発行者 =Rohitv、担当者 =Robert の場合、NetScaler アプライアンスはこれらのパラメーターと値のペアに一致する応答を処理できます。

コンテンツグループの無効化パラメータは、キャッシュされたエントリを期限切れにします。たとえば、BugID が無効化パラメータで、ユーザが POST リクエストを発行してバグレポートを更新したとします。無効化ポリシーによってリクエストがこのコンテンツグループに送信され、コンテンツグループの無効化パラメータにより、BugID 値と一致するキャッシュされたすべての応答が期限切れになります。(次回ユーザーがこのレポートに対して GET リクエストを発行したときに、キャッシュポリシーにより、NetScaler アプライアンスはオリジンサーバーからのレポートのキャッシュされたエントリを更新できます。)

`hit` 同じパラメーターをパラメーターまたは無効化パラメーターとして使用できることに注意してください。

コンテンツグループは次の順序でリクエストパラメータを抽出します。

- URL クエリ
- ポストボディ
- クッキーヘッダー

パラメータが最初に出現した後は、リクエスト内のどこで発生したかに関係なく、それ以降の出現はすべて無視されます。たとえば、URL クエリと POST 本文の両方にパラメータが存在する場合、URL クエリのパラメータのみが考慮されます。

コンテンツグループにヒットパラメータと無効化パラメータを使用する場合は、コンテンツグループを設定するときにパラメータを設定します。

注: Citrix では、パラメータ化されたコンテンツグループではなくセレクターを使用することをお勧めします。セレクターはより柔軟で、より多くの種類のデータに対応できるためです。

セレクタを構成する

コンテンツグループは、ヒットセレクターを使用してキャッシュヒットを取得したり、無効化セレクターを使用して期限切れのキャッシュオブジェクトを期限切れにしてオリジンサーバーから新しいオブジェクトを取得したりできま

す。

セレクタには、高度な式と呼ばれる名前と論理式が含まれています。

高度な式の詳細については、「ポリシーと式」を参照してください。

セレクタを設定するには、セレクタに名前を割り当て、1 つ以上の式を入力します。ベストプラクティスとして、特に省略する理由がない限り、セレクター式には URL のステムとホストを含める必要があります。

CLI を使用してセレクターを設定するには

コマンドプロンプトで入力します。

```
add cache selector \<selectorName\> ( \<rule\> ... )
```

式を設定する方法については、「コマンドラインインターフェイスを使用してセレクタ式を設定するには」を参照してください。

```
1  >add cache selector product_selector "http.req.url.query.value("
       ProductId")" "http.req.url.query.value("BatchNum")" "http.req.url.
       query.value("depotLocation")"
2
3  > add cache selector batch_selector "http.req.url.query.value("
       ProductId")" "http.req.url.query.value("BatchId")" "http.req.url.
       query.value("depotLocation")"
4
5  > add cache selector product_id_selector "http.req.url.query.value("
       ProductId")"
6
7  > add cache selector batchnum_selector "http.req.url.query.value("
       BatchNum")" "http.req.url.query.value("depotLocation")"
8
9  > add cache selector batchid_selector "http.req.url.query.value("
       depotLocation")" "http.req.url.query.value("BatchId")"
10
11 <!--NeedCopy-->
```

GUI を使用してセレクタを設定するには

[最適化] **>** [統合キャッシュ] **>** [キャッシュセレクター] に移動し、キャッシュセレクターを追加します。

コンテンツグループ

コンテンツグループは、レスポンスで提供できるキャッシュされたオブジェクトのコンテナです。統合キャッシュを初めて有効にすると、キャッシュ可能なオブジェクトは Default という名前のコンテンツグループに保存されます。独自のプロパティを持つコンテンツグループを作成できます。たとえば、画像データ、バグレポート、株価情報用に個別のコンテンツグループを定義したり、株価コンテンツグループを他のグループよりも頻繁に更新するように構成したりできます。

コンテンツグループ全体またはコンテンツグループ内の選択したエントリの有効期限を設定できます。

コンテンツグループのデータは、次のように静的でも動的でもかまいません。

- 静的コンテンツグループ。リクエストの URL ステムとホスト名、およびレスポンスの URL ステムとホスト名が完全に一致するものを検索します。
- 動的コンテンツグループ。特定のパラメータと値のペア、任意の文字列、または文字列パターンを含むオブジェクトを検索します。動的コンテンツグループは、頻繁に更新されるデータ (バグレポートや株価情報など) をキャッシュする場合に便利です。

コンテンツグループからのリクエストを処理する

1. ユーザーは、バグレポートなどの項目の検索条件を入力し、HTML フォームの [検索] ボタンをクリックします。
2. ブラウザは 1 つ以上の HTTP GET リクエストを発行します。これらのリクエストにはパラメーター (バグ所有者、バグ ID など) が含まれます。
3. NetScaler アプライアンスは要求を受信すると、一致するポリシーを検索し、これらの要求と一致するキャッシュポリシーが見つかると、要求をコンテンツグループに送信します。
4. コンテンツグループは、セレクターで設定した基準に基づいて、コンテンツグループ内の適切なオブジェクトを検索します。

たとえば、コンテンツグループは`NameField=username and BugID=ID`に一致する応答を取得できます。

1. 一致するオブジェクトが見つかると、NetScaler アプライアンスはそれらをユーザーのブラウザに提供し、そこで完全な応答（バグレポートなど）にまとめられます。

コンテンツグループのオブジェクトを無効にする

1. ユーザーがデータを変更します (たとえば、ユーザーがバグレポートを変更して [送信] ボタンをクリックします)。
2. ブラウザは、このデータを 1 つ以上の HTTP リクエストの形式で送信します。たとえば、バグの所有者とバグ ID に関する情報を含む複数の HTTP POST リクエストの形式でバグレポートを送信できます。
3. NetScaler アプライアンスは、リクエストを無効化ポリシーと照合します。通常、これらのポリシーは HTTP POST メソッドを検出するように設定されます。
4. 要求が無効化ポリシーと一致する場合、NetScaler アプライアンスはこのポリシーに関連付けられているコンテンツグループを検索し、構成された無効化条件に一致する応答を期限切れにします。

たとえば、無効化セレクターは一致する応答を検索できます。`NameField=username and BugID=ID`

1. NetScaler アプライアンスがこれらの応答に対する GET 要求を次回受信すると、元のサーバーから更新されたバージョンを取得し、更新された応答をキャッシュして、これらの応答をユーザーのブラウザーに提供します。これらの応答は、完全なバグレポートにまとめられます。

基本コンテンツグループの設定

デフォルトでは、キャッシュされたデータはすべてデフォルトのコンテンツグループに保存されます。複数のコンテンツグループを設定し、これらのコンテンツグループを 1 つ以上のポリシーで指定できます。

静的コンテンツにはコンテンツグループを設定でき、動的コンテンツにはコンテンツグループを構成する必要があります。デフォルトグループを含むすべてのコンテンツグループの設定を変更できます。

コマンドラインインターフェイスを使用して基本的なコンテンツグループを設定するには

コマンドプロンプトで入力します。

```
add cache contentgroup <name> (-hitSelector <hitSelectorName> -
invalSelector <invalidationSelectorName> | -hitParams <hitParamName>
-invalParams<invalidationParamName>)-type <type> [-relExpiry <sec> |
-relExpiryMilliSec <msec>] [-heurExpiryParam <positiveInteger>]
```

```
add cache contentgroup Products_Details -hitSelector product_selector
 -invalSelector id_selector
```

```
add cache contentgroup bugrep -hitParams IssuePage RecordID Template
TableId -invalParams RecordID -relExpiry 864000
```

GUI を使用して基本的なコンテンツグループを設定するには

[最適化] **>** [統合キャッシュ] **>** [コンテンツグループ] に移動し、コンテンツグループを作成します。

キャッシュされたオブジェクトを期限切れにするかフラッシュする

応答に Expires ヘッダーまたは有効期限 (Max-Age または Smax-Age) の Cache-Control ヘッダーがない場合は、次のいずれかの方法を使用してコンテンツグループのオブジェクトを期限切れにする必要があります。

- コンテンツグループの有効期限設定を設定して、オブジェクトを保持するかどうか、また保持する期間を決定します。
- コンテンツグループの無効化ポリシーとアクションを構成します。詳細については、「キャッシュと無効化のポリシーの設定」を参照してください。
- コンテンツグループまたはコンテンツグループ内のオブジェクトを手動で期限切れにします。

キャッシュされた応答の有効期限が切れると、NetScaler アプライアンスは次回クライアントが応答を要求したときにそれを更新します。デフォルトでは、キャッシュがいっぱいになると、NetScaler アプライアンスは最も使用頻度の低い応答を最初に置き換えます。

次のリストでは、コンテンツグループの設定を使用してキャッシュされた応答を期限切れにする方法について説明します。通常、これらのメソッドはパーセントまたは秒単位で指定されます。

- マニュアル。コンテンツグループのすべての応答またはキャッシュ内のすべての応答を手動で無効にする。
- レスポンスベース。肯定応答と否定応答の特定の有効期限。レスポンスベースの有効期限は、レスポンスに Last-Modified ヘッダーがない場合にのみ考慮されます。
- ヒューリスティックな有効期限。Last-Modified ヘッダーを持つ応答の場合、ヒューリスティック有効期限は応答が変更されてからの経過時間を指定します (現在の時刻から Last-Modified 時間を引き、ヒューリスティック有効期限の値を掛けて計算されます)。たとえば、Last-Modified ヘッダーが応答が 2 時間前に更新さ

れたことを示していて、ヒューリスティック有効期限設定が 10% の場合、キャッシュされたオブジェクトは 0.2 時間後に期限切れになります。この方法では、頻繁に更新される応答は、より頻繁に期限切れになる必要があると想定しています。

- 絶対または相対。回答が毎日期限切れになる正確な（絶対）時刻を、HH: MM 形式、ローカル時間、または GMT で指定します。ローカルタイムはすべてのタイムゾーンで機能しない場合があります。

相対的な有効期限は、キャッシュミスによりオリジンサーバーにアクセスした時点から応答の有効期限が切れるまでの数秒またはミリ秒を指定します。相対有効期限をミリ秒単位で指定する場合は、10 の倍数を入力します。この形式の有効期限は、すべての肯定的な回答に適用されます。レスポンスの Last Modified、Expires、Cache-Control ヘッダーは無視されます。

絶対有効期限と相対有効期限は、レスポンス自体の有効期限情報よりも優先されます。

- ダウンロード時。「レスポンス受信完了後に期限切れ」オプションを選択すると、レスポンスがダウンロードされた時点で期限切れになります。これは、株価情報など、頻繁に更新される回答に役立ちます。デフォルトでは、このオプションは無効になっています。

「Flash Cache」と「レスポンス受信完了後に期限切れ」の両方を有効にすると、動的アプリケーションのパフォーマンスが向上します。両方のオプションを有効にすると、NetScaler アプライアンスは同時リクエストのブロックに対して 1 つの応答のみを取得します。

- 固定されています。デフォルトでは、キャッシュがいっぱいになると、NetScaler アプライアンスは最も使用頻度の低い応答を最初に置き換えます。NetScaler アプライアンスは、固定されているとマークされているコンテンツグループにはこの動作を適用しません。

コンテンツグループの有効期限設定を行わない場合、グループ内のオブジェクトを期限切れにするオプションは次のとおりです。

- コンテンツグループに適用される INVAL アクションを含むポリシーを設定します。
- INVAL アクションを使用するポリシーを設定する場合は、コンテンツグループの名前を入力します。

有効期限法の適用方法

有効期限の仕組みは、肯定的な回答と否定的な回答で異なります。肯定的反応と否定的反応は、下記の表「陽性反応の有効期限、および否定的回答の有効期限」に記載されています。

コンテンツグループに適用される有効期限方法を理解するための経験則は次のとおりです。

- オブジェクトを期限切れにするかどうかを決定するときに、NetScaler アプライアンスが応答ヘッダーを評価するかどうかを制御できます。
- 絶対有効期限と相対有効期限により、NetScaler アプライアンスは応答ヘッダーを無視します（応答の有効期限情報は上書きされます）。
- ヒューリスティックな有効期限設定と「弱陽性」および「弱陰性」の有効期限（構成ユーティリティではデフォルト値として表示）により、NetScaler アプライアンスは応答ヘッダーを調べます。これらの設定は次のように連動します。

- Expires または Cache-Control ヘッダーの値は、これらのコンテンツグループ設定よりも優先されます。
- Expires または Cache-Control ヘッダーがないが、Last-Modified ヘッダーがある肯定応答の場合、NetScaler アプライアンスはヒューリスティックな有効期限設定をヘッダー値と比較します。
- Expires、Cache-Control、Last-Modified のヘッダーがない肯定的な応答には、NetScaler アプライアンスは「弱い陽性」の値を使用します。
- Expires または Cache-Control ヘッダーのない否定的な応答の場合、NetScaler アプライアンスは「弱い負の」値を使用します。

次の表は、これらの方法の適用方法をまとめたものです。

応答タイプ	有効期限ヘッダータイプ	コンテンツグループ設定	オブジェクトがキャッシュに残っている期間
ポジティブ	すべてのヘッダー	他の設定なしでコンテンツを期限切れにする (REExpiry)	「コンテンツを期限切れにする」設定の値を使用してください。
ポジティブ	すべてのヘッダー	他の設定なしでコンテンツを期限切れにする (ABS 期限)	[コンテンツの有効期限] 設定の値から現在の日付を引きます。
ポジティブ	すべてのヘッダー	[コンテンツの有効期限] ([有効期限]) と [コンテンツの期限切れ] (ABS 期限)	コンテンツグループの設定には、2 つの値の小さい方を使用します。この表の前の行を参照してください。
ポジティブ	最終更新日 (他のヘッダーを含む)	ヒューリスティック (HeuRExpiry パラメータ) とその他の設定	現在の日付から最終更新日を引き、その結果にヒューリスティック有効期限設定の値を掛けて、100 で割ります。
ポジティブ	最終更新日 (他のヘッダーを含む)	デフォルト (正) (WeakPosREL 有効期限)、その他の設定なし	デフォルト (正) 有効期限設定の値を使用してください。
ポジティブ	有効期限切れまたはキャッシュコントロール:Max-Age ヘッダーが存在する	最終更新ヘッダーが存在しない、ヒューリスティック (HeuRespiry パラメーター)、デフォルト (正) (WeakPosREL 有効期限)、またはその両方が表示される	有効期限または日付から現在の日付を引きます。`Cache-Control: Max-Age`

応答タイプ	有効期限ヘッダータイプ	コンテンツグループ設定	オブジェクトがキャッシュに残っている期間
ポジティブ	キャッシュヘッダーなし	デフォルト (正) (WeakPosREL 有効期限) およびその他の有効期限設定	デフォルト (正) 設定の値を使用してください。
ポジティブ	キャッシュヘッダーなし	ヒューリスティック (HeuRespiry パラメーター) は存在しますが、デフォルト (正) (WeakPosREL 有効期限) 設定はありません。	Last-Modified ヘッダーがない場合、応答はキャッシュされないか、既に Expired ステータスでキャッシュされます。Last-Modified ヘッダーが存在する場合は、ヒューリスティックな有効期限値を使用してください。
ネガティブ	有効期限が切れるまたは `Cache-Control: Max-Age`	[コンテンツを期限切れにする (RELExpiry)]、[コンテンツの有効期限] (ABS Expiry)、またはその両方の設定	Expires ヘッダーの値から現在の日付を減算するか、Cache-Control: Max-Age ヘッダーの値を使用してください。
ネガティブ	有効期限またはキャッシュコントロールヘッダーがない	[コンテンツを期限切れにする (RELExpiry)]、[コンテンツの有効期限] (ABS Expiry)、またはその両方の設定	レスポンスはキャッシュされていないか、すでに期限切れのステータスでキャッシュされています。
ネガティブ	有効期限が切れるまたは `Cache-Control: Max-Age`	すべての設定	`Cache-Control: Max-Age` 有効期限または日付から現在の日付を引きます。
ネガティブ	有効期限とキャッシュコントロール:Max-Age ヘッダーはありません	デフォルト (マイナス) (弱ネグレルの有効期限)	デフォルト (マイナス) 設定の値を使用してください。
ネガティブ	有効期限とキャッシュコントロール:Max-Age ヘッダーはありません	デフォルト (マイナス) (WeakNegrel 有効期限) 以外のすべての設定	オブジェクトはキャッシュされていないか、すでに期限切れのステータスでキャッシュされています。

手動でコンテンツグループを期限切れにする

コンテンツグループのすべてのエントリを手動で期限切れにすることができます。

コマンドラインインターフェイスを使用してコンテンツグループのすべての回答を手動で期限切れにするには

コマンドプロンプトで入力します。

```
expire cache contentGroup <name>
```

GUI を使用してコンテンツグループ内のすべての応答を手動で期限切れにするには

[最適化] **>** **[**統合キャッシュ**]** **>** **[**コンテンツグループ] に移動し、コンテンツグループを選択し、[無効化] をクリックしてコンテンツグループ内のすべての応答を期限切れにします。

GUI を使用してキャッシュ内のすべての応答を手動で期限切れにするには

[最適化] **>** **[**統合キャッシュ**]** **>** **[**コンテンツグループ**]** に移動し、**[**すべて無効**]** をクリックしてキャッシュ内のすべての応答を期限切れにします。

コンテンツグループの定期的な有効期限の設定

エントリを選択的に期限切れにするか、完全に期限切れにするように、コンテンツグループを設定できます。有効期限は固定でも相対でもかまいません。

コマンドラインインターフェイスを使用してコンテンツグループの有効期限を設定するには

コマンドプロンプトで入力します。

```
set cache contentgroup \<name> (-relExpiry|-relExpiryMilliSec|-
absExpiry|-absExpiryGMT| -heurExpiryParam|-weakPosRelExpiry|-weakNegRelExpiry
| -expireAtLastBye)\<expirationValue>
```

GUI を使用してコンテンツグループの有効期限を設定するには

[最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択し、有効期限方法を指定します。

個々の回答を期限切れにする

応答の期限が切れると、NetScaler アプライアンスは更新されたコピーを元のサーバーから強制的に取得します。たとえば、ETag バリデーターや Last-Modified ヘッダーがないレスポンスは再検証できません。そのため、これらの応答をフラッシュしても、期限切れになるのと同じ効果があります。

静的データのコンテンツグループ内のキャッシュされたレスポンスを期限切れにするには、保存されている URL と一致する必要がある URL を指定できます。キャッシュされた応答がパラメータ化されたコンテンツグループの一部である場合は、グループ名と正確な URL ステムを指定する必要があります。ホスト名とポート番号は、キャッシュさ

れたレスポンスのホスト HTTP リクエストヘッダーと同じでなければなりません。ポートが指定されていない場合、ポート 80 と見なされます。

コマンドラインインターフェイスを使用してコンテンツグループ内の個々の回答を期限切れにするには

コマンドプロンプトで入力します。

```
expire cache object -url <URL> -host <hostName> [-port <port>] [-groupName<contentGroupName>] [-httpMethod GET|POST]
```

CLI を使用してコンテンツグループの個々の回答を期限切れにするには

コマンドプロンプトで、次のコマンドを入力します。

```
expire cache object -locator <positiveInteger>
```

GUI を使用してキャッシュされたレスポンスを期限切れにするには

[最適化] > [統合キャッシュ] > [キャッシュオブジェクト] に移動し、キャッシュされたレスポンスを選択して期限切れにします。

GUI を使用して応答を期限切れにするには

[最適化] > [統合キャッシュ] > [キャッシュオブジェクト] に移動し、[検索] をクリックして検索条件を設定し、必要なキャッシュ済みレスポンスを見つけて期限切れにします。

コンテンツグループ内の回答をフラッシュする

コンテンツグループ内のすべての応答、グループ内の一部の応答、またはキャッシュ内のすべての応答を削除またはフラッシュできます。キャッシュされたレスポンスをフラッシュすると、キャッシュされた新しいレスポンス用にメモリが解放されます。

> 注:
>
> 一度に複数のオブジェクトの応答をフラッシュするには、設定ユーティリティメソッドを使用します。コマンドラインインターフェイスにはこのオプションはありません。

コマンドラインインターフェイスを使用してコンテンツグループからの応答をフラッシュするには

コマンドプロンプトで入力します。

```
flush cache contentGroup <name> [-query <queryString> | [-selectorValue <selectorExpressionIDList> -host <hostName>]]
```

GUI を使用してコンテンツグループからの応答をフラッシュするには

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動します。
2. 詳細ペインで、次のように回答をフラッシュします。
   - すべてのコンテンツグループのすべての回答を消去するには、「すべて無効」をクリックし、すべての回答をフラッシュします。

- 特定のコンテンツグループの回答を消去するには、コンテンツグループを選択し、「無効化」をクリックして、すべての回答をフラッシュします。

> 注:
> このコンテンツグループがセレクターを使用している場合は、「セレクター値」テキストボックスに文字列を入力し、「ホスト」テキストボックスにホスト名を入力することで、回答を選択的にフラッシュできます。次に、「フラッシュ」と「**OK**」をクリックします。Selector の値には、パラメータ化された無効化に使用される最大 2319 文字のクエリ文字列を使用できます。

コンテンツグループが無効化パラメータを使用している場合は、クエリフィールドに文字列を入力することで回答を選択的にフラッシュできます。

コンテンツグループが無効化パラメータを使用しており、ターゲットホストに属するオブジェクトを無効化するように設定されている場合は、「クエリー」フィールドと「ホスト」フィールドに文字列を入力します。

コマンドラインインターフェイスを使用してキャッシュされた応答をフラッシュするには

コマンドプロンプトで入力します。

```
flush cache object -locator <positiveInteger> | -url <URL> -host <
hostName> [-port <port>] [-groupName <contentGroupName>] [-httpMethod
 GET|POST]
```

GUI を使用してキャッシュされたレスポンスをフラッシュするには

[最適化] > [統合キャッシュ] > [キャッシュオブジェクト] に移動し、キャッシュされたオブジェクトを選択してフラッシュします。

コンテンツグループを削除する

キャッシュにレスポンスを保存するポリシーで使用されていないコンテンツグループを削除できます。コンテンツグループがポリシーにバインドされている場合は、最初にポリシーを削除する必要があります。コンテンツグループを削除すると、そのグループに保存されているすべての回答が削除されます。

デフォルト、BASEFILE、またはデルタグループを削除することはできません。デフォルトグループには、他のコンテンツグループに属さないキャッシュされた応答が格納されます。

コマンドラインインターフェイスを使用してコンテンツグループを削除するには

コマンドプロンプトで入力します。

```
rm cache contentgroup <name>
```

GUI を使用してコンテンツグループを削除するには

[最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択して削除します。

キャッシュと無効化のポリシーを構成する

August 15, 2023

ポリシーにより、統合キャッシュは、キャッシュからの応答とオリジンのどちらからの応答の処理を試みるかを決定できます。NetScaler アプライアンスには統合キャッシュ用の組み込みポリシーが用意されており、さらに多くのポリシーを構成できます。ポリシーを設定すると、それをアクションに関連付けます。アクションは、ポリシーが適用されるオブジェクトをキャッシュするか、オブジェクトを無効（期限切れ）にします。通常、キャッシュポリシーは GET リクエストと POST リクエストの情報に基づいて設定されます。通常、無効化ポリシーは、リクエスト内の POST メソッドの存在やその他の情報に基づいて決定します。キャッシュポリシーまたは無効化ポリシーの GET または POST リクエスト内の任意の情報を使用できます。

組み込みポリシーの一部は、構成ユーティリティの統合キャッシュの「ポリシー」ノードで表示できます。組み込みポリシー名はアンダースコア (_) で始まります。

アクションは、トラフィックがポリシーと一致した場合に NetScaler アプライアンスが何をするかを決定します。実行できるアクションは次のとおりです：

- アクションをキャッシュします。CACHE アクションに関連付けるポリシーは、応答をキャッシュに保存し、キャッシュから処理します。
- 無効化アクション。INVAL アクションに関連付けるポリシーは、キャッシュされた応答をただちに期限切れにし、オリジンサーバーから更新します。Web ベースのアプリケーションの場合、無効化ポリシーは POST リクエストを評価することがよくあります。
- 「キャッシュしない」アクション。NOCACHE アクションに関連付けるポリシーは、オブジェクトをキャッシュに保存しません。
- アクションを暫定的にキャッシュします。MAYCACHE または MAYNOCACHE アクションに関連付けるポリシーは、より多くのポリシー評価の結果によって異なります。

統合キャッシュは LOCK メソッドで指定されたオブジェクトを格納しませんが、`LOCK`リクエストの受信時にキャッシュされたオブジェクトを無効にすることができます。無効化ポリシーの場合のみ、式`http.req.method.eq( “lock” )`を使用して`LOCK`をメソッドとして指定できます。NetScaler アプライアンスはこのメソッド名を文字列として認識するため、`GET`および`POST`要求のポリシーとは異なり、LOCK メソッドを引用符で囲む必要があります。

ポリシーを作成したら、リクエストとレスポンスの全体的な処理における特定のポイントにポリシーをバインドします。ポリシーをバインドする前にポリシーを作成しますが、ポリシーを作成する前に、バインドポイントが処理の順序にどのように影響するかを理解しておく必要があります。

特定のバインドポイントにバインドされたポリシーは、ポリシーバンクを構成します。goto 式を使用して、ポリシーバンクの実行順序を変更できます。他のポリシーバンクのポリシーを呼び出すこともできます。さらに、ラベルを作成してポリシーをそれらにバインドできます。このようなラベルは処理ポイントに関連付けられていませんが、そのラベルにバインドされたポリシーは他のポリシーバンクから呼び出すことができます。

統合キャッシュポリシーに関連付けるアクション

次の表は、統合キャッシュポリシーのアクションを示しています。

アクション	仕様
キャッシュ	応答の有効期限が切れていない場合は、キャッシュから応答を返します。応答を配信元サーバーから取得する必要がある場合、NetScaler アプライアンスは応答を提供する前に応答をキャッシュします。頻繁に更新されアクセスされるデータもキャッシュできます。たとえば、株価は頻繁に更新されますが、キャッシュしておくと、複数のユーザーにすばやく配信できます。必要に応じて、キャッシュされたデータはダウンロード後すぐに更新できます。CACHE アクションは組み込みポリシーで上書きできます。
NOCACHE	常にオリジンサーバーからレスポンスを取得し、そのレスポンスを保存不可としてマークします。通常、機密データやパーソナライズされたデータには NOCACHE ポリシーを設定します。

アクション	仕様
MAY_CACHE	リクエストタイムポリシーで使用すると、レスポンスタイムポリシーの評価を待つ間、レスポンスをコンテンツグループに保存することが暫定的に可能になります。次のことが可能です。1. 一致する応答時間ポリシーに CACHE アクションはあるがコンテンツグループが指定されていない場合、組み込みポリシーがこのポリシーよりも優先されない限り、応答は Default グループに保存されます。一致する応答時間ポリシーに CACHE アクションがあり、要求時ポリシー内のコンテンツグループと同じコンテンツグループが指定されている場合、組み込みポリシーがこのポリシーよりも優先されない限り、応答は指定されたコンテンツグループに保存されます。一致する応答時間ポリシーに CACHE アクションがあるが、要求時ポリシー内のコンテンツグループとは異なるコンテンツグループが指定されている場合、NOCACHE アクションが適用されます。一致するレスポンス・タイム・ポリシーに NOCACHE アクションがある場合は、NOCACHE アクションを実行します。一致する応答時間ポリシーがない場合は、組み込みポリシーがこのポリシーよりも優先されない限り、CACHE アクションが適用されます。
MAY_NOCACHE	リクエスト時ポリシーでは、この設定によりレスポンスのキャッシュが暫定的に禁止されます。応答時に、次のいずれかのアクションが実行されます。-リクエストに一致する応答時間ポリシーがない場合、最後のアクションは NOCACHE です。-一致する応答時間ポリシーに CACHE アクションが含まれている場合、組み込みポリシーがこのポリシーをオーバーライドしない限り、最後のアクションは CACHE です。-一致する応答時間ポリシーに NOCACHE アクションが含まれている場合、最後のアクションは NOCACHE です。-一致する応答時間ポリシーに CACHE アクションはあるがコンテンツグループが指定されていない場合、組み込みポリシーがこのポリシーよりも優先されない限り、最後のアクションは応答をデフォルトコンテンツグループにキャッシュすることです。

アクション	仕様
INVAL	キャッシュされた応答を期限切れにします。ポリシーとコンテンツグループの設定方法に応じて、1 つ以上のコンテンツグループのすべての応答が期限切れになるか、コンテンツグループ内の選択したオブジェクトが期限切れになります。注:INVAL アクションはリクエスト時ポリシーでのみ指定できます。

ポリシーのバインドポイント

ポリシーを次のバインドポイントのいずれかにバインドできます。

- グローバルポリシーバンク。これらは、「ポリシー評価の順序」で説明されているように、リクエスト時間のデフォルト、リクエスト時間のオーバーライド、応答時間のデフォルト、および応答時間オーバーライドポリシーバンクです。
- 仮想サーバ。仮想サーバにバインドするポリシーは、「ポリシー評価の順序」で説明されているように、グローバルオーバーライドポリシーの後およびグローバルデフォルトポリシーの前に処理されます。ポリシーを仮想サーバーにバインドするときは、ポリシーを要求時または応答時の処理にバインドします。
- アドホックポリシーラベル。ポリシーラベルは、ポリシーバンクに割り当てられる名前です。統合キャッシュには、グローバルラベルに加えて、次の 2 つのカスタムポリシーラベルが組み込まれています。
  - _reqBuiltinDefaults. このポリシーラベルは、デフォルトでは、リクエスト時のデフォルトポリシーバンクから呼び出されます。
  - _resBuiltinDefaults. このポリシーラベルは、デフォルトでは、応答時間のデフォルトポリシーバンクから呼び出されます。

新しいポリシーラベルを定義することもできます。ユーザー定義のポリシーラベルにバインドされたポリシーは、組み込みバインドポイントのいずれかのポリシーバンク内から呼び出す必要があります。

> 重要:
> INVAL アクションを含むポリシーは、要求時間オーバーライドまたは応答時間オーバーライドのバインドポイントにバインドする必要があります。ポリシーを削除するには、まずそのポリシーをバインド解除する必要があります。

ポリシー評価の順序

高度なポリシーを有効にするには、NetScaler アプライアンスのトラフィック処理中のある時点でポリシーが呼び出されていることを確認する必要があります。呼び出し時間を指定するには、ポリシーをバインドポイントに関連付けます。評価順にバインドポイントを以下に示します。

- リクエスト時間のオーバーライド。要求が要求時上書きポリシーと一致する場合、デフォルトで要求時ポリシー評価は終了し、NetScaler アプライアンスは一致するポリシーに関連付けられたアクションを保存します。
- 要求時の負荷分散仮想サーバー。すべての要求時オーバーライドポリシーを評価してもポリシー評価を完了できない場合、NetScaler アプライアンスは負荷分散仮想サーバーにバインドされた要求時ポリシーを処理します。リクエストがこれらのポリシーのいずれかに一致すると、評価が終了し、NetScaler アプライアンスは一致するポリシーに関連付けられたアクションを保存します。
- リクエスト時のコンテンツスイッチング仮想サーバー。このバインドポイントにバインドされたポリシーは、負荷分散仮想サーバーにバインドされた要求時ポリシーの後に評価されます。
- リクエスト時間のデフォルト。すべての要求時間が経過してもポリシー評価を完了できない場合、仮想サーバー固有のポリシーが評価されると、NetScaler アプライアンスは要求時のデフォルトポリシーを処理します。要求が要求時のデフォルトポリシーと一致する場合、デフォルトで要求時のポリシー評価は終了し、NetScaler アプライアンスは一致するポリシーに関連付けられたアクションを保存します。
- 応答時間のオーバーライド。リクエスト時のオーバーライドポリシー評価に似ています。
- 応答時間負荷分散仮想サーバー。要求時の仮想サーバーポリシー評価に似ています。
- 応答時間のコンテンツスイッチング仮想サーバー。要求時の仮想サーバーポリシー評価に似ています。
- 応答時間のデフォルト。リクエスト時のデフォルトポリシー評価に似ています。

各バインドポイントに複数のポリシーを関連付けることができます。バインドポイントに関連するポリシーの評価順序を制御するには、優先度レベルを設定します。他のフロー制御情報がない場合、ポリシーは、プライオリティレベルに従って評価されます。プライオリティレベルは、最も低い数値から開始します。

> 注:
> POST データまたは Cookie ヘッダーのリクエスト時間ポリシーは、統合キャッシュ内の組み込みのリクエスト時ポリシーが POST リクエストのNOCACHEアクションと Cookie を使用したリクエストに対するMAY_NOCACHEアクションを返すため、リクエスト時のオーバーライド評価時に呼び出す必要があります。パラメータ化されたコンテンツグループを指す要求時ポリシーに、MAY_CACHEまたはMAY_NOCACHEアクションを関連付けます。応答時間ポリシーによって、トランザクションがキャッシュに格納されるかどうかが決まります。

統合キャッシュのポリシーを構成する

組み込みポリシーで処理できないデータを処理するには、新しいポリシーを構成します。キャッシュ、キャッシュの発生防止、およびキャッシュされたデータの無効化には、個別のポリシーを設定します。統合キャッシュのポリシーの主な構成要素は次のとおりです。

- ルール:HTTP リクエストまたはレスポンスを評価する論理表現。
- アクション: ポリシーをアクションに関連付けて、ポリシー・ルールに一致するリクエストまたはレスポンスをどう処理するかを決定します。

コンテンツグループ: ポリシーを 1 つ以上のコンテンツグループに関連付けて、アクションを実行する場所を指定します。

コマンドラインインターフェイスを使用してキャッシュのポリシーを設定するには

コマンドプロンプトで入力します。

```
add cache policy <policyName> -rule <expression> -actionCACHE|MAY_CACHE
|NOCACHE|MAY_NOCACHE [-storeInGroup <contentGroupName>] [-undefAction
 NOCACHE|RESET]
> add cache policy image_cache -rule "http.req.url.contains(\"jpg
\")|| http.req.url.contains(\"jpeg\")"-action CACHE -storeingroup
myImages_group -undefaction NOCACHE

> add cache policy bugReportPolicy -rule "http.req.url.query.contains
(\"IssuePage\")"-action CACHE -storeInGroup bugReportGroup

> add cache policy my_form_policy -rule "http.req.header(\"Host
\")contains(\"my.company.com\")&& http.req.method.eq(\"GET\")&&
http.req.url.query.contains(\"v=7\")"-action CACHE -storeInGroup
my_form_event

> add cache policy viewproducts_policy -rule "http.req.url.contains
(\"viewproducts.aspx\")"-action CACHE -storeInGroup Product_Details
```

コマンドラインインターフェイスを使用して無効化ポリシーを設定するには

コマンドプロンプトで入力します。

```
add cache policy <policyName> -rule <expression> -action INVAL [-
    invalObjects "<contentGroupName1>[,<selectorName1>"]. . .]] | [-
    invalGroup <contentGroupName1>[, <contentGroupName2>. . .]] [-
    undefAction NOCACHE|RESET]
<!--NeedCopy-->
```

```
> add cache policy invalidation_events_policy -rule "http.req.header("
    Host")contains("my.company.com") && http.req.method.eq("GET") &&
    http.req.url.query.contains("v=8") -action INVAL -invalObjects
    my_form_event -undefaction NOCACHE
<!--NeedCopy-->
```

```
> add cache policy inval_all -rule "http.req.method.eq("POST") && http.
    req.url.contains("jpeg")" -action INVAL -invalGroups myImages_group
    myApps_group PDF_group
<!--NeedCopy-->
```

```
> add cache policy bugReportInvalidationPolicy -rule "http.req.url.
    query.contains("TransitionForm")" -action INVAL -invalObjects
    bugReport`
`> add cache policy editproducts_policy  - rule "http.req.url.contains("
    editproducts.aspx")"  - action INVAL -invalObjects "Product_Details,
    batchnum_sel" "Products_In_Depots,batchid_sel"
<!--NeedCopy-->
```


GUI を使用してキャッシュまたは無効化のポリシーを設定するには

[最適化] > [統合キャッシュ] > [ポリシー] に移動し、新しいポリシーを作成します。

統合キャッシュポリシーをグローバルにバインドする

ポリシーをグローバルにバインドすると、NetScaler アプライアンス上のすべての仮想サーバーでポリシーを使用できます。

コマンドラインインターフェイスを使用して統合キャッシュポリシーをグローバルにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
bind cache global <policy> -priority <positiveInteger> [-
    typeREQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT] [-
    gotoPriorityExpression <expression>] [-invoke <labelType> <labelName
    >]
<!--NeedCopy-->
```

```
> bind cache global myCachePolicy -priority 100 -type req_default
<!--NeedCopy-->
```

> 注:
>
> type 引数は、以前のバージョンの NetScaler アプライアンスを使用して定義したポリシーとの下位互換性を維持するために、グローバルにバインドされたポリシーの場合はオプションです。タイプを省略すると、ポリシー・ルールが応答時間式かリクエスト時間式かに応じて、ポリシーは REQ_DEFAULT または RES_DEFAULT にバインドされます。ルールにリクエスト時間と応答時間の両方のパラメーターが含まれている場合、そのルールは RES_DEFAULT にバインドされます。以下は、型を省略したバインディングの例です

以下は、型を省略したバインディングの例です。

```
> bind cache global myCache Policy 200
```

構成ユーティリティを使用して統合キャッシュポリシーをグローバルにバインドするには

[最適化] > [統合キャッシュ] に移動し、[キャッシュポリシーマネージャー] をクリックし、関連するバインドポイントと接続タイプ (リクエスト/レスポンス) を指定してポリシーをバインドします。

統合キャッシュポリシーを仮想サーバーにバインドする

ポリシーを仮想サーバにバインドすると、ポリシーに一致し、関連する仮想サーバを通過する要求と応答に対してのみ使用できます。

GUI を使用する場合、仮想サーバの設定ダイアログボックスを使用してポリシーをバインドできます。これにより、この仮想サーバーにバインドされているすべての NetScaler ADC モジュールのすべてのポリシーを表示できます。

統合キャッシュの PolicyManager 構成 ダイアログを使用することもできます。これにより、仮想サーバーにバインドされている統合キャッシュポリシーのみを表示できます。

コマンドラインインターフェイスを使用して統合キャッシュポリシーを仮想サーバーにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
bind lb vserver <name>@ -policyName <policyName> -priority <
    positiveInteger> -type(REQUEST|RESPONSE)
<!--NeedCopy-->
```

```
bind cs vserver <name>@ -policyName <policyName> -priority <
    positiveInteger> -type(REQUEST|RESPONSE)
<!--NeedCopy-->
```

構成ユーティリティを使用して統合キャッシュポリシーを仮想サーバーにバインドするには (仮想サーバー方式)

- CS Virtual Server-[トラフィック管理**] > [**コンテンツスイッチング**] > [**仮想サーバー] に移動し、仮想サーバーを選択し、関連するキャッシュポリシーをバインドします。
- LB 仮想サーバー - Traffic **Management > Load Balancing > Virtual Servers**, に移動して仮想サーバーを選択し、関連キャッシュポリシーをバインドします。

GUI を使用して統合キャッシュポリシーを仮想サーバにバインドするには（Policy Manager 方式）。

[最適化] > [統合キャッシュ] に移動し、[キャッシュポリシーマネージャ] をクリックし、関連するバインドポイントと接続タイプを指定してキャッシュポリシーをバインドします。

> 注:
> 適切なバインドポイントを選択すると、キャッシュポリシーを負荷分散仮想サーバーとコンテンツスイッチング仮想サーバーの両方にバインドできます。

ファイルの圧縮および非圧縮バージョンをキャッシュする方法

デフォルトでは、圧縮を処理できるクライアントには、圧縮されていない応答か、gzip、deflate、compress、pack200-gzip 形式の圧縮応答を提供できます。クライアントが圧縮を処理する場合、`Accept-Encoding: compression` リクエストでフォーマットヘッダーが送信されます。クライアントが受け入れる圧縮タイプは、キャッシュされたオブジェクトの圧縮タイプと一致する必要があります。たとえば、`Accept-Encoding: deflate` ヘッダーのあるリクエストに応答して`cached.gzip`ファイルを提供することはできません。

圧縮を処理できないクライアントは、キャッシュされた応答が圧縮されている場合、キャッシュミスを処理します。

動的キャッシュの場合、同じデータの圧縮データ用と非圧縮バージョン用の 2 つのコンテンツグループを構成する必要があります。次に、圧縮を処理できないクライアントにキャッシュから非圧縮ファイルを提供し、圧縮を処理できる同じファイルの圧縮バージョンをクライアントに提供するためのセレクタ、コンテンツグループ、およびポリシーを設定する例を示します。


```
add cache selector uncompressed_response_selector http.req.url "http.
req.header(\"Host\")"

add cache contentGroup uncompressed_group -hitSelector uncompressed_responst_s
 -invalSelector uncomp_resp_sel

add cache policy cache_uncompressed -rule "HTTP.REQ.URL.CONTAINS(\"
xyz\")&& !HTTP.REQ.HEADER(\"Accept-Encoding\").EXISTS"-action CACHE -
storeInGroup uncompressed_group

bind cache global cache_uncompressed -priority 100 -gotoPriorityExpression
 END -type REQ_OVERRIDE

add cache selector compressed_response_selector HTTP.REQ.URL "HTTP.
REQ.HEADER(\"Host\")""HTTP.REQ.HEADER(\"Accept-Encoding\")"

add cache contentGroup compressed_group -hitSelector compressed_response_selec


add cache policy cache_compressed -rule "HTTP.REQ.URL.CONTAINS(\"
xyz\")&& HTTP.REQ.HEADER(\"Accept-Encoding\").EXISTS"-action CACHE -
storeInGroup compressed_group

bind cache global cache_compressed -priority 200 -gotoPriorityExpression
 END -type REQ_OVERRIDE
```

キャッシュ用のポリシーバンクの構成

特定のバインドポイントに関連するすべてのポリシーは、まとめてポリシーバンクと呼ばれます。銀行内のポリシーの優先度レベルを設定するだけでなく、Goto 式を設定することで銀行内の評価順序を変更できます。現在のポリシーバンク内から外部ポリシーバンクを呼び出すことで、評価順序をさらに変更できます。また、新しいポリシーバンクを設定して、それに独自のラベルを割り当てることもできます。このようなポリシーバンクは処理サイクルのどの時点にも縛られないため、他のポリシーバンク内からのみ呼び出すことができます。便宜上、組み込みのバインドポイントに対応しないラベルを持つポリシーバンクをポリシーラベルと呼びます。

「ポリシーのバインド」で説明されているように、ポリシーをバインドしてプライオリティレベルを割り当てることによってポリシー評価の順序を制御するだけでなく、Goto 式を設定することで、ポリシーバンク内のフローを確立できます。Goto 式は、優先度レベルによって決定されるフローを上書きします。現在のバンクのエントリを評価した後に外部ポリシーバンクを呼び出すことで、評価フローを制御することもできます。評価が完了すると、評価は常に現在の銀行に返却されます。

次の表は、ポリシーバンクにおける統制評価のエントリをまとめたものです。

属性	**Specifies**
名前	ポリシーの名前、またはポリシーを評価せずに別のポリシーバンクを呼び出す場合は、キーワード NOPOLICY。ポリシーバンクでは NOPOLICY を複数回指定できますが、名前付きポリシーは 1 回しか指定できません。
優先度	整数。整数が小さいほど、優先度が高くなります。
Goto 式	次に評価するポリシーまたはポリシーバンクを決定します。次の値のいずれかを指定できます。1. 次へ: 次に優先度の高いポリシーに移動 2. 終了: 評価を停止します。3. USE_INVOCATION_RESULT: このエントリが別のポリシーバンクを呼び出す場合に適用されます。呼び出されたバンクの最後の Goto の値が END の場合、評価は停止します。最後の Goto が END 以外の場合は、現在の政策銀行が NEXT を実行します。正の数: 次に評価される政策の優先度番号 5. 数値表現: 次に評価されるポリシーの優先順位番号を生成する式。Goto は政策銀行でしか前進できない。Goto 式を省略することは、END を指定することと同じです。
呼び出しタイプ	ポリシーバンクの種類を指定します。値は次のいずれかです-1。仮想サーバーをリクエスト: 仮想サーバーに関連付けられているリクエスト時ポリシーを呼び出します。2. レスポンス仮想サーバー: 仮想サーバーに関連付けられている応答時間ポリシーを呼び出します。3. ポリシーラベル: 銀行のポリシーラベルによって識別されるように、別のポリシーバンクを呼び出します。
呼び出し名	「呼び出しタイプ」に指定した値に応じて、仮想サーバーまたはポリシーラベルの名前を指定します。

統合キャッシュには 2 つの組み込みポリシーラベルがあり、さらに多くのポリシーラベルを設定できます。

`_reqBuiltInDefaults`: このポリシーラベルは、リクエスト時のデフォルトのバインドポイントから呼び出されます。
`_resBuiltInDefaults`: このポリシーラベルは、応答時のデフォルトのバインドポイントから呼び出されます。

コマンドラインインターフェイスを使用してキャッシュポリシーバンクでポリシーラベルを呼び出すには

コマンドプロンプトで入力します。

```
bind cache policylabel <labelName> -policname<policyName> -priority<
    priority> [-gotoPriorityExpression <gotoprioritыExpression>] [-
```



```
      invoke <labelType> <labelName>]
<!--NeedCopy-->
```

GUI を使用してキャッシュポリシーバンク内のポリシーラベルを呼び出すには、次の手順を実行します。

1. [最適化] > [統合キャッシュ] に移動し、[キャッシュポリシーマネージャ] をクリックし、関連するバインドポイント ([グローバル上書き] または [デフォルトグローバル]) と接続タイプを指定して、このバインドポイントにバインドされているポリシーのリストを表示します。
2. ポリシーを評価せずにポリシーラベルを呼び出す場合は、「**NOPOLICY**」をクリックします。

> 注:
>
> 外部ポリシーバンクを呼び出すには、「Invoke Type」列のフィールドをクリックし、ポリシーバンク内のこの時点で呼び出すポリシーバンクのタイプを選択します。これはグローバルラベルでも仮想サーバーバンクでもかまいません。「Invoke Name」フィールドに、ラベルまたは仮想サーバー名を入力します。

コマンドラインインターフェイスを使用して仮想サーバーポリシーバンクのキャッシュポリシーラベルを呼び出すには

コマンドプロンプトで入力します。

```
bind lb vserver <name>@ -policyName <policyName>|<NOPOLICY-CACHE> -
    priority<positiveInteger> -gotoPriorityExpression <expression> -type
     REQUEST|RESPONSE -invoke<labelType> <labelName>
<!--NeedCopy-->
```

```
bind cs vserver <name> -policyName <policyName>|<NOPOLICY-CACHE> -
    priority<positiveInteger> -gotoPriorityExpression <expression> -type
     REQUEST|RESPONSE -invoke<labelType> <labelName>
<!--NeedCopy-->
```

GUI を使用して仮想サーバポリシーバンクのキャッシュポリシーラベルを呼び出すには

1. **[**** トラフィック管理] > [負荷分散**/**コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバーを選択して [ポリシー] をクリックします。******
2. このバンクの既存のエントリを設定する場合は、この手順をスキップしてください。このポリシーバンクに新しいポリシーを追加する場合、または「ダミー」NOPOLICY エントリを使用する場合は、[追加] をクリックして、次のいずれかの操作を行います。
   - 新しいポリシーを設定するには、「統合キャッシュでのポリシーの設定」の説明に従って、[キャッシュ] をクリックし、新しいポリシーを設定します。
   - ポリシールールを処理せずにポリシーバンクを呼び出すには、NOPOLICY-CACHEオプションを選択します。

> 注:
>
> 外部ポリシーバンクを呼び出すには、「Invoke Type」列のフィールドをクリックし、ポリシーバンク内のこの

> 時点で呼び出すポリシーバンクのタイプを選択します。これはグローバルラベルでも仮想サーバーバンクでもかまいません。「Invoke Name」フィールドに、ラベルまたは仮想サーバー名を入力します。

統合キャッシュにポリシーラベルを設定

組み込みバインドポイントまたは仮想サーバーのポリシーバンクでポリシーを構成する以外に、キャッシュポリシーラベルを作成し、これらの新しいラベルのポリシーバンクを構成できます。

統合キャッシュのポリシーラベルは、統合キャッシュの詳細ペインの Policy Manager で表示できるバインドポイントの 1 つ（リクエストの上書き、リクエストのデフォルト、応答の上書き、または応答のデフォルト）または組み込みポリシーラベル_reqBuiltinDefaultsおよび_resBuiltinDefaultsからのみ呼び出すことができます。ポリシーラベルは、一度だけ呼び出すことができるポリシーとは異なり、何度でも呼び出すことができます。

NetScaler GUI には、ポリシーラベルの名前を変更するオプションがあります。ポリシーラベルの名前を変更しても、ラベルにバインドされているポリシーの評価プロセスには影響しません。

> 注記:
>
> NOPOLICY「ダミー」ポリシーを使用して、別のポリシーバンクからポリシーラベルを呼び出すことができます。NOPOLICYエントリは、ルールを処理しないプレースホルダです。

コマンドラインインターフェイスを使用してキャッシュ用のポリシーラベルを構成するには

コマンドプロンプトで次のコマンドを入力してポリシーラベルを作成し、構成を確認します。

- `add cache policylabel <labelName> -evaluates (REQ|RES)`
- `show cache policylabel <labelName>`

このポリシーラベルは、ポリシーバンクから呼び出します。

GUI を使用してキャッシュのポリシーラベルを設定するには、次の手順を実行します。

[最適化] > [統合キャッシュ] > [ポリシーラベル] に移動し、ポリシーラベルを追加して、キャッシュされたポリシーをバインドします。

> 注:
> NetScaler ADC が適切なタイミングでポリシーラベルを処理できるようにするには、組み込みのバインドポイントに関連付けられているポリシーバンクのいずれかでこのラベルの呼び出しを構成します。

GUI を使用してポリシーラベルの名前を変更するには、次の手順を実行します。

[最適化] > [統合キャッシュ] > [ポリシーラベル] に移動して、ポリシーラベルを選択し、名前を変更します。

統合キャッシュポリシーとポリシーラベルをバインド解除して削除する

ポリシーをポリシーバンクからバインド解除したり、削除したりできます。ポリシーを削除するには、まずそのポリシーをバインド解除する必要があります。ポリシーラベルの呼び出しを削除したり、ポリシーラベルを削除したりすることもできます。ポリシーラベルを削除するには、まずラベルに設定した呼び出しをすべて削除する必要があります。

組み込みのバインドポイント (リクエストデフォルト、リクエストオーバーライド、レスポンスデフォルト、レスポンスオーバーライド) のラベルをバインド解除したり削除したりすることはできません。

コマンドラインインターフェイスを使用してグローバルキャッシュポリシーをバインド解除するには

コマンドプロンプトで入力します。

```
unbind cache global <policy>
```

コマンドラインインターフェイスを使用して仮想サーバー固有のキャッシュポリシーをバインド解除するには

コマンドプロンプトで入力します。

```
(unbind lb vserver|unbind cs vserver)<vserverName> -policyName <
policyName> -type(REQUEST|RESPONSE)
```

コマンドラインインターフェイスを使用してキャッシュポリシーを削除するには

コマンドプロンプトで入力します。

```
rm cache policy <policyName>
```

GUI を使用してキャッシュポリシーをバインド解除するには、次の手順を実行します。

[最適化] > [統合キャッシュ] に移動し、[キャッシュポリシーマネージャ] をクリックし、関連するバインドポイントと接続タイプ (要求/応答) を指定してポリシーをバインド解除します。

GUI を使用してポリシーラベルの呼び出しを削除するには、次の手順を実行します。

1. [最適化] > [統合キャッシュ] に移動し、[キャッシュポリシーマネージャ] をクリックし、関連するバインドポイント (負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバー) と接続タイプを指定して、この仮想サーバーにバインドされたキャッシュポリシーのリストを表示します。
2. ポリシーの [呼び出し] 列で、エントリをクリアします。

データベースプロトコルのキャッシュサポート

August 15, 2023

統合キャッシュ機能は、キャッシュポリシーによって決定されたデータベース要求を監視し、キャッシュします。NetScaler アプライアンスにはデフォルトポリシーがないため、ユーザーは MYSQL および MSSQL プロトコルの

キャッシュポリシーを構成する必要があります。デフォルトプロトコルを設定する場合、リクエストベースのポリシーは CACHE と INVAL アクションのみをサポートするのに対し、応答ベースのポリシーは「NOCACHE」アクションのみをサポートしていることに注意してください。ポリシーを設定したら、それらを仮想サーバーにバインドする必要があります。MYSQL と MSSQL のポリシーは、要求と応答の両方で、仮想サーバーにのみバインドされます。

キャッシュポリシーを作成する前に、MYSQL または MSSQL タイプのキャッシュコンテンツグループを作成する必要があります。キャッシュコンテンツグループを作成するときは、少なくとも 1 つの選択セレクターを関連付けます。キャッシュコンテンツグループの設定については、「基本コンテンツグループの設定」を参照してください。

次の例では、SQL プロトコルのキャッシュサポートを構成および検証する方法について説明します。

```
> enable feature IC
> set cache parameter -memlimit 100
> add cache selector sel1 mssql.req.query.text

> add cache contentgroup cg1  -type "MSSQL" -hitselector "sel1" -
    invalselector "inval_sel" -relExpiry "500" -maxResSize
 "100"
> add cache policy cp1  -rule "mssql.req.query.command.contains("select
    ")" -action "CACHE" -storeInGroup "cg1"
> add cache policy cp2  -invalObjects "cg1" -rule "mssql.req.query.text
    .contains("insert")" -action "INVAL"
> add db user user1  -password "Pass1"
> add service svc_sql_1 10.102.147.70 mssql 64834 -healthMonitor "NO" -
    downstateflush "ENABLED"
> add lb vserver lb_mssql1 mssql 10.102.147.77 1433  -lbmethod "
    roundrobin"
> bind lb vserver lb_mssql1 svc_sql_1
> bind lb vserver lb_mssql1 -policyName cp1  -type "REQUEST" -priority
    "2"
> bind lb vserver lb_mssql1 -policyName cp2  -type "REQUEST" -priority
    "1"

> show cache selector sel1
     Name:sel1
                    Expressions:
                    1)mssql.req.query.text
> show cache policy cp1
                    Name:cp1
     Rule:mssql.req.query.command.contains("select")
                    CacheAction:CACHE
                    Stored in group: cg1
                    UndefAction:Use Global
                Hits:2
                    Undef Hits:0
                    Policy is bound to following entities
                    1) Bound to:
                                REQ VSERVER lb_mssql1
                                Priority:2
                                GotoPriorityExpression: END
<!--NeedCopy-->
```


> 注:
>
> フラッシュクラウドの削減で説明されているように、フラッシュクラウドを削減する方法は、MYSQL および MSSQL プロトコルではサポートされていません。

キャッシュポリシーとセレクタの式を構成する

December 8, 2023

リクエスト時間式は、リクエストタイムトランザクションのデータを検査し、応答時間式は応答時間トランザクション内のデータを検査します。キャッシュのポリシーで、式が要求または応答のデータと一致する場合、NetScaler アプライアンスはポリシーに関連付けられたアクションを実行します。セレクタでは、リクエスト時間式を使用して、コンテンツグループに格納されている一致する応答を検索します。

統合キャッシュのポリシーとセレクタを構成する前に、少なくとも HTTP 要求および応答 URL に表示されるホスト名、パス、および IP アドレスを知っておく必要があります。そして、おそらく HTTP リクエストとレスポンス全体の形式を知る必要があります。Live HTTP ヘッダーhttp://livehttpheaders.mozdev.org/)or HTTPFox https://addons.mozilla.org/en-US/firefox/addon/6647などのプログラムは、組織が操作する HTTP データの構造を調べるのに役立ちます。

株価プログラムに対する HTTP GET リクエストの例を次に示します。

```
GET /quote.dll?page=dynamic&mode=data&mode=stock&symbol=CTXS&page=multi
    &selected=CTXS&random=0.00792039478975548 HTTP/1.1

Host: quotes.mystockquotes.com

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9)
    Gecko/2008052906 Firefox/3.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q
    =0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate,compress,pack200-gzip

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referer: http://quotes.mystockquotes.com/quote.dll?mode=stock&symbol=
    CTXS&page=multi&selected=CTXS

```



```
Cookie: __qca=1210021679-72161677-10297606
<!--NeedCopy-->
```

式を設定するときは、次の制限事項に注意してください。

エクスプレッションタイプ	制限
リクエスト	CACHE アクションまたは NOCACHE アクションを使用して、ポリシーにリクエスト時間式を設定しないでください。代わりに MAY_CACHE または MAY_NOCACHE を使用してください。
応答	キャッシュポリシーでのみ応答時間式を構成します。セレクタはリクエスト時間式のみを使用できます。INVAL アクションを使用してポリシーで応答時間式を設定しないでください。注:CACHE アクションとパラメータ化されたコンテンツグループを使用して、ポリシーで応答時間式を設定しないでください。MAY_CACHE アクションを使用します。

> 注:
> 高度な式の包括的な説明については、「ポリシーと式」を参照してください。

式の構文

構文の基本的なコンポーネントは次のとおりです。

- 次のように、キーワードをピリオド (.) で区切ります。

```
http.req.url
```

- 次のように、文字列値を括弧と引用符で囲みます。

```
http.req.url.query.contains("this")
```

- コマンドラインから式を構成するときは、内部引用符 (式を区切る引用符ではなく、式の値を区切る引用符) をエスケープする必要があります。1 つの方法は、次のようにスラッシュを使用することです。

```
\"abc\"
```

セレクタ式は外観順に評価され、セレクタ定義内の複数の式は論理 AND で結合されます。セレクタ式とは異なり、ブール演算子を指定し、ポリシールールの高度な式の優先順位を変更できます。

キャッシュポリシーまたはセレクタで式を設定する

> 注:
> ポリシー式の構文は、セレクタ式とは異なります。高度な表現に関する包括的な説明については、「ポリシーと式」を参照してください。

コマンドラインインターフェイスを使用してポリシー式を設定するには

1. 「統合キャッシュポリシーのグローバルバインド」の説明に従って、ポリシー定義を開始します。
2. ポリシールールを設定するには、ルール全体を引用符で囲み、ルール内の文字列値をエスケープ引用符で囲みます。

以下はその例です:

“http.req.url.contains(“jpg”)”

ブール値を追加するには、&& を挿入し、

1.

以下の例を参照してください:

```
"http.req.url.contains(\"jpg\")|| http.req.url.contains(\"jpeg\")"
```

```
"http.req.url.query.contains(\"IssuePage\")"
```

```
"http.req.header(\"Host\")contains(\"my.company.com\")&& http.req.
method.eq(\"GET\")&& http.req.url.query.contains(\"v=7\")"
```

1. コンパウンドの構成部分の評価順序を設定するには

```
"http.req.url.contains(\"jpg\")|| (http.req.url.contains(\"jpeg\")&&
http.req.method.eq(\"GET\"))"
```

コマンドラインインターフェイスを使用してセレクタ式を設定するには、次の手順を実行します。

1. 「コンテンツ・グループについて」の説明に従って、セレクタ定義を開始します。
2. セレクタ式を設定するには、ルール全体を引用符で囲み、ルール内の文字列値をエスケープ引用符で囲みます。

以下はその例です:

```
"http.req.url.contains(\"jpg\")"
```

ブール値を追加したり、&& を挿入したりすることはできません。

1.

以下の例を参照してください:

```
"http.req.url.query.value("ProductId")" "http.req.url.query.value("
    BatchNum")" "http.req.url.query.value("depotLocation")"
<!--NeedCopy-->
```

GUI を使用してポリシーまたはセレクタ式を設定するには

1. 「構成ユーティリティを使用してキャッシュまたは無効化のポリシーを構成するには」または「構成ユーティリティを使用してセレクタを構成するには」の説明に従って、ポリシーまたはセレクタ定義を開始します。
2. [式] フィールドでは、[クラシック構文に切り替え] をクリックして詳細ポリシーを手動で入力するか、式エディタを使用して新しい式を作成できます。

複合式の 2 つの部分の間に演算子を挿入するには、[演算子] ボタンをクリックし、オペレータタイプを選択します。以下は、Boolean OR () (2 本の縦棒で示される) を使用して設定された式の例です。

3.
4. [頻繁に使用する式] ドロップダウンリストをクリックして、よく使用される式を挿入します。
5. 式をテストするには、[評価] をクリックします。[式エバリュエータ] ダイアログボックスで、式と一致するフロータイプを選択します。データフィールドに、式を使用して解析する HTTP リクエストまたはレスポンスを貼り付け、[評価] をクリックします。

キャッシュされたオブジェクトとキャッシュ統計を表示する

特定のキャッシュされたオブジェクトを表示したり、キャッシュリクエスト、ミス、メモリ使用量に関するサマリー統計を表示できます。この統計は、キャッシュから提供されるデータの量、最大のパフォーマンス上の利点の原因となる項目、およびキャッシュのパフォーマンスを向上させるために調整できる内容に関する洞察を提供します。

ここでは、次の詳細について説明します。

- キャッシュされたオブジェクトの表示
- 特定のキャッシュされたレスポンスの検索
- キャッシュ統計の表示

キャッシュされたオブジェクトの表示

キャッシュを有効にすると、キャッシュされたオブジェクトの詳細を表示できます。たとえば、次の項目を表示できます。

- レスポンスサイズとヘッダーサイズ
- ステータスコード

- コンテンツグループ
- ETag、最終変更ヘッダー、およびキャッシュ制御ヘッダー
- URL をリクエストする
- ヒットパラメータ
- 宛先 IP アドレス
- リクエストと応答時間

コマンドラインインターフェイスを使用してキャッシュされたオブジェクトのリストを表示するには

コマンドプロンプトで入力します:

```
show cache object
```

プロパティ	説明
応答サイズ (バイト)	レスポンスヘッダーとボディのサイズ。
レスポンスヘッダーのサイズ (バイト)	レスポンスのヘッダー部分のサイズ。
レスポンスステータスコード	レスポンスとともに送信されるステータスコード。
ETag	応答に挿入された ETag ヘッダー。通常、このヘッダーは、応答が最近変更されたかどうかを示します。
最終変更日	レスポンスに挿入された Last-Modified ヘッダー。このヘッダーは、レスポンスが最後に変更された日付を示します。
キャッシュコントロール	レスポンスに挿入された Cache-Control ヘッダー。
日付	応答がいつ送信されたかを示す Date ヘッダー。
コンテンツグループ	レスポンスが格納されるコンテンツグループ。
複合マッチ	このオブジェクトがパラメータ化された値に基づいてキャッシュされている場合、このフィールド値は YES です。
ホスト	このレスポンスを要求した URL で指定されたホスト。
ホストポート	このレスポンスを要求した URL で指定されたホストのリスニングポート
URL	格納されたレスポンスに対して発行された URL。
接続先 IP	このレスポンスが取得されたサーバーの IP アドレス。
Destination port	宛先サーバーのリスニングポート。
ヒットパラメータ	レスポンスを格納するコンテンツグループがヒットパラメータを使用している場合は、このフィールドに表示されます。

プロパティ	説明
ヒットセレクタ	このコンテンツグループがヒットセレクタを使用している場合は、このフィールドに表示されます。
無効なセレクタ	このコンテンツグループが無効化セレクタを使用している場合は、このフィールドに表示されます。
セレクタ式	このコンテンツグループがセレクタを使用している場合、このフィールドには選択ルールを定義する式が表示されます。
要求時間	リクエストが発行されてからの時間（ミリ秒）。
応答時間	キャッシュがレスポンスの受信を開始してから経過した時間（ミリ秒）。
年齢	オブジェクトがキャッシュ内にある時間。
有効期限	オブジェクトが期限切れとしてマークされるまでの時間。
フラッシュされた	有効期限が切れた後に応答がフラッシュされたかどうか。
プリフェッチ	このコンテンツグループに対してプリフェッチが設定されている場合、オブジェクトがオリジンからフェッチされる有効期限までの時間。プリフェッチはネガティブオブジェクトには適用されません (たとえば、404「オブジェクトが見つかりません」というレスポンス)。
現在の読者	現在処理されているリクエストの概数。 Content-Length ヘッダーオブジェクトを含むレスポンスがダウンロードされている場合、現在のミスおよび現在のリーダー値はそれぞれ 1 になります。チャンクレスポンスオブジェクトがダウンロードされている場合、現在のミス値は通常 1 になりますが、クライアントに対して提供されるチャンク応答は統合キャッシュバッファから送信されないため、現在のリーダー値は通常 0 になります。
現在のミス	キャッシュミスおよびオリジンサーバーからのフェッチの結果となったリクエストの現在の数。この値は、通常 0 または 1 です。コンテンツグループに対して [毎回ポーリング] が有効になっている場合、カウントは 1 より大きくなる可能性があります。
ヒット数	このオブジェクトのキャッシュヒット数。
ミス	このオブジェクトのキャッシュミス数

プロパティ	説明
圧縮形式	このオブジェクトに適用される圧縮のタイプ。圧縮形式としては、gzip、deflate、compress、pack200-gzip などがある。
応答中の HTTP バージョン	レスポンスの送信に使用された HTTP のバージョン。
応答に弱い etag が存在する	エンティティのビットが変更された場合、強力な etag ヘッダーが変わります。強力なヘッダーは、オブジェクトのオクテット値に基づいています。エンティティの意味が変わると、弱い etag ヘッダーが変わります。弱な etag 値は、セマンティックアイデンティティに基づいています。弱い etags の値は「W」で始まります。
ネガティブマーカーセル	マーカーオブジェクトはキャッシュ可能ですが、キャッシュされるための条件をすべて満たしているわけではありません。たとえば、オブジェクトがコンテンツグループの最大応答サイズを超えている可能性があります。このタイプのオブジェクトに対してマーカーセルが作成されます。ユーザーが次回このオブジェクトに対するリクエストを送信すると、キャッシュミスが提供されます。
理由マーカーが作成されました	マーカーセルが作成された理由 (たとえば、「minhit を待っている」、「コンテンツ長の応答データがグループサイズの制限にありません」など)。
毎回オートポーリング	統合キャッシュがバリデーター（Last-Modified または eTag レスポンスヘッダーのいずれか）で既に期限切れの 200 OK レスポンスを受け取った場合、レスポンスを保存し、Auto-PET（毎回自動的にポーリング）としてマークします。
NetScaler Etag がレスポンスで挿入されました	NetScaler アプライアンスによって生成される ETag ヘッダーのバリエーション。NetScaler が応答に Etag を挿入すると、値 YES が表示されます。
キャッシュに完全なレスポンスが存在	これが完全な応答であるかどうかを示します。
DNS によって検証された宛先 IP	オブジェクトの格納時に DNS 解決が実行されたかどうかを示します。
キャッシュフォワードプロキシを介して格納されたオブジェクト	統合キャッシュに設定されたフォワードプロキシが原因で、この応答が格納されたかどうかを示します。
オブジェクトはデルタベースファイルです	デルタ圧縮された応答。
minhits 待ち	レスポンスをキャッシュする前に、このコンテンツグループで最小数のオリジンサーバーヒットが必要かどうかを示します。

プロパティ	説明
ミニヒットカウント	オブジェクトをキャッシュする前に、このコンテンツグループが最小数のオリジンサーバー要求を必要とする場合、このフィールドにはこれまでに受信したリクエストの数が表示されます。
HTTP リクエストメソッド	このオブジェクトを取得したリクエストで使用されるメソッド GET または POST。
ポリシー別に格納	このオブジェクトが格納された原因となったキャッシュポリシーの名前。[利用不可] の値は、ポリシーが非アクティブ化または削除されたことを示します。NONE の値は、オブジェクトが可視ポリシーと一致せず、キャッシュの内部基準に従って格納されたことを示します。
アプリケーションファイアウォールのメタデータが存在します	このパラメータは、アプリケーションファイアウォールと統合キャッシュの両方が有効になっている場合に使用されます。アプリケーションファイアウォールは、レスポンスページのコンテンツを分析し、そのメタデータ（ページに含まれる URL やフォームなど）を保存し、レスポンスを含むメタデータをキャッシュにエクスポートします。キャッシュはページとメタデータを格納し、キャッシュがページを提供すると、メタデータをリクエストのセッションに送り返します。
HTTP コールアウトオブジェクト、名前、タイプ、応答	これらのセルは、HTTP コールアウト式の結果としてこのデータが格納されたかどうかを示し、コールアウトのさまざまな側面と対応する応答に関する情報を提供します。HTTP コールアウトの詳細については、「HTTP コールアウト」を参照してください。

GUI を使用してキャッシュされたオブジェクトを表示するには

[最適化] > [統合キャッシュ] > [キャッシュオブジェクト] に移動します。キャッシュされたすべてのオブジェクトを表示し、必要に応じて並べ替えることができます。

キャッシュされた特定のレスポンスを見つける

検索条件に基づいて、キャッシュ内の個々のアイテムを検索できます。キャッシュされたアイテムを検索するには、データを含むコンテンツグループがヒットセレクターと無効化セレクターを使用するかどうかに応じて、次のようにさまざまな方法があります。

- コンテンツグループがセレクターを使用している場合、キャッシュされたアイテムのロケータ ID を使用してのみ検索を実行できます。

- コンテンツグループがセレクタを使用しない場合は、URL、ホスト、コンテンツグループ名などの基準を使用して検索を実行します。

キャッシュされたレスポンスを検索するときに、URL とホストでいくつかの項目を見つけることができます。レスポンスがセレクタを使用するコンテンツグループ内にある場合は、ロケータ番号（たとえば、0x00000000ad7af00000050）を使用してのみ検索できます。後で使用するためにロケータ番号を保存するには、エントリを右クリックし、[コピー] を選択します。セレクタの詳細については、「セレクタと基本コンテンツグループの設定」を参照してください。

コマンドラインインターフェイスを使用して、セレクタを持たないコンテンツグループのキャッシュされたレスポンスを表示するには

コマンドプロンプトで入力します：

```
show cache object [-locator <positiveInteger>] | [(-url <URL> (-
host <hostName> [-port <port>] [-groupName <contentGroupName>]
[-httpMethod GET | POST ]))| [-httpStatus <positive integer>] |
 -group <contentGroupName> | -ignoreMarkerObjects (ON | OFF)| -
includeNotReadyObjects (ON | OFF)]
```

コマンドラインインターフェイスを使用してセレクタを持つコンテンツグループのキャッシュされたレスポンスを表示するには

コマンドプロンプトで入力します：

```
show cache object -locator <locatorString> MarkerObjects ( ON | OFF )
| -includeNotReadyObjects ( ON | OFF )| [-httpStatus<positive integer
>]
```

構成ユーティリティを使用して、セレクタを持たないコンテンツグループのキャッシュされた応答を表示するには

[最適化] **>** [統合キャッシュ] **>** [キャッシュオブジェクト] に移動し、[検索] をクリックして、必要なキャッシュされた応答を表示するための検索条件を設定します。

コンテンツグループをまだ構成していない場合は、すべてのオブジェクトが Default グループに属します。

キャッシュ統計の表示

次の表に、表示できる詳細なキャッシュ統計情報をまとめたものです。

カウンター	説明
ヒット数	統合キャッシュで見つかり、統合キャッシュから提供されるレスポンス。イメージファイルなどの静的オブジェクト、ステータスコード 200、203、300、301、302、304、307、403、404、410、および CACHE アクションを使用したユーザー定義のポリシーに一致する応答を含むページが含まれます。
ミス	レスポンスが最終的にオリジンサーバーからフェッチされた HTTP リクエストをインターセプトしました。

| リクエスト | キャッシュリクエストの合計とキャッシュミスの合計。|
|304 ヒット以外 | ユーザーがアイテムを複数回要求し、NetScaler アプライアンスが最後にアイテムを提供してからキャッシュ内のアイテムが変更されていない場合、NetScaler アプライアンスはキャッシュされたオブジェクトの代わりに 304 の応答を提供します。
この統計は、304 の応答を除いて、NetScaler アプライアンスがキャッシュから処理したアイテムの数を示します。|
304 ヒット	NetScaler アプライアンスがキャッシュから処理した 304（オブジェクトが変更されていない）応答の数。
304 ヒット率 (%)	NetScaler アプライアンスが提供した 304 の応答のうち、他の応答に対する割合。
ヒット率 (%)	キャッシュから提供できなかった応答に対する NetScaler アプライアンスがキャッシュ（キャッシュリクエスト）から処理したレスポンスの割合。
オリジン帯域幅の保存 (%)	キャッシュからの応答を提供するため、NetScaler アプライアンスがオリジンサーバーに保存した処理能力の推定。
NetScaler によって処理されるバイト数	NetScaler アプライアンスがオリジンサーバーとキャッシュから処理した合計バイト数。
キャッシュによって処理されるバイト数	NetScaler アプライアンスがキャッシュから処理した合計バイト数。
バイトヒット率（%）	NetScaler アプライアンスがキャッシュから提供したデータの割合。すべての応答に含まれるすべてのデータに対する相対値。
キャッシュから圧縮されたバイト	NetScaler アプライアンスが圧縮形式で処理したデータの量（バイト単位）。
ストレージ可能なミス	NetScaler アプライアンスがキャッシュ内で要求されたオブジェクトを見つけない場合、オリジンサーバーからオブジェクトを取得します。これは、キャッシュミスと呼ばれます。保存可能なキャッシュミスはキャッシュに格納できます。
格納不能なミス	格納不可能なキャッシュミスをキャッシュに保存できません。
Misses	すべてのキャッシュミス。
Revalidations	Cache-Control ヘッダーの Max-Age 設定により、ユーザーに提供する前に、中間キャッシュが統合キャッシュでコンテンツを再検証する必要があるタイミングを秒数で決定します。
詳しい情報については、「キャッシュコントロールヘッダーの挿入」を参照してください。
正常に再検証	実行された再検証の数。
詳しい情報については、「キャッシュコントロールヘッダーの挿入」を参照してください。
条件付き Req への変換	キャッシュされた PET オブジェクトに対するユーザーエージェントリクエストは、常に条件付きリクエストに変換され、オリジンサーバーに送信されます。
詳しい情報については、「リクエストを受信するたびにオリジンサーバーのポーリングを行うこと」を参照してください。
ストレージ可能なミス率 (%)	格納可能なキャッシュミスを格納できないキャッシュミスのパーセンテージで表す。
成功したリヴァール率 (%)	すべての再検証試行に対するパーセンテージで表した再検証の成功率。
詳しい情報については、「キャッシュコントロールヘッダーの挿入」を参照してください。
最後のバイトで期限切れ	最後のボディバイトを受け取った直後にキャッシュのコンテンツが期限切れになった回数。表の「キャッシュヒットとミス」で説明されているように、ポジティブレスポンスにのみ適用されます。
詳細については、「パフォーマンス最適化の例」を参照してください。

|Flashcache misses|Flash Cache を有効にした場合、キャッシュはサーバーへのリクエストを 1 つだけ許可し、フラッシュの群れを排除します。この統計は、キャッシュミスであったフラッシュキャッシュ要求の数を示します。
詳細は、「キャッシュへのリクエストのキューイング。」|
|Flashcache ヒット | キャッシュヒットしたフラッシュキャッシュリクエストの数。
詳細は、「キャッシュへのリクエストのキューイング」を参照してください。|
パラメータ化されたインバルリクエスト | 無効化 (INVAL) アクション、および無効化セレクタまたはパラメータを使用してグループ内のキャッシュされたオブジェクトを選択的に期限切れにするコンテンツグループを持つポリシーに一致するリクエスト。|
| 完全インバル要求 |invalGroups パラメータが設定され、1 つ以上のコンテンツグループが期限切れになる無効化ポリシーに一致するリクエスト。|
|Inval Request| 無効化ポリシーに一致し、特定のキャッシュされたレスポンスまたはコンテンツグループ全体の有効期限が切れるリクエスト。|
パラメータ化されたリクエスト	パラメータ化されたコンテンツグループのポリシーを使用して処理されたキャッシュリクエストの数。
パラメータ化された非 304 ヒット	パラメータ化されたコンテンツグループを持つポリシーを使用して処理されたキャッシュリクエストの数。フルキャッシュされたレスポンスが検出され、応答が 304（オブジェクトが更新されていない）応答ではありませんでした。
Parameterized 304 ヒット	パラメータ化されたコンテンツグループを持つポリシーを使用して処理されたキャッシュリクエストの数。キャッシュされたオブジェクトが見つかり、オブジェクトが 304（オブジェクトが更新されていない）応答でした。
パラメータ化されたヒット数の合計	パラメータ化されたコンテンツグループを持つポリシーを使用して処理されたキャッシュリクエストの数。キャッシュされたオブジェクトが見つかった。
パラメータ化された 304 ヒット率 (%)	パラメータ化されたポリシーを使用して検出された 304 (オブジェクトが更新されていない) 応答のうち、すべてのキャッシュヒットに対する割合。
リクエストのたびにポーリング	毎回ポーリングが有効になっている場合、NetScaler アプライアンスは、保存されたオブジェクトを提供する前に、常にオリジンサーバーを参照します。
詳しい情報については、「リクエストを受信するたびにオリジンサーバーのポーリングを行うこと」を参照してください。|
ヒットするたびにポーリング	毎回ポーリングメソッドを使用してキャッシュヒットが検出された回数。
詳しい情報については、「リクエストを受信するたびにオリジンサーバーのポーリングを行うこと」を参照してください。|
毎回ポーリングヒット率 (%)	Poll Every Time メソッドを使用したキャッシュヒットのパーセンテージ。[毎回ポーリング] を使用したキャッシュオブジェクトに対するすべての検索に対する相対値。詳しい情報については、「リクエストを受信するたびにオリジンサーバーのポーリングを行うこと」を参照してください。
最大メモリ（KB）	キャッシュに割り当てられている NetScaler アプライアンスの最大メモリ容量。詳細は、「キャッシュのグローバル属性の設定」を参照してください。
最大メモリアクティブ値 (KB)	メモリをキャッシュに割り当てた後に設定される最大メモリ量 (アクティブ値)。詳細については、「さまざまなシナリオで NetScaler アプライアンスの統合キャッシュ機能を構成する方法」を参照し

てください。|
使用済みメモリ (KB)	実際に使用されているメモリの量。
メモリ割り当て失敗	キャッシュにレスポンスを格納する目的でメモリの使用に失敗した回数。
これまでの最大の応答	キャッシュまたはオリジンサーバーのいずれかで見つかり、クライアントに送信された最大レスポンス (バイト単位)。
キャッシュされたオブジェクト	キャッシュ内のオブジェクトの数。まだ完全にダウンロードされていないレスポンスや、有効期限が切れているがまだフラッシュされていないレスポンスを含みます。
Marker オブジェクト	マーカーオブジェクトは、応答がコンテンツグループの最大または最小応答サイズを超えている場合、またはコンテンツグループの最小ヒット数をまだ受け取っていない場合に作成されます。
処理中のヒット	キャッシュから配信されたヒットの数。
処理ミス	オリジンサーバーからフェッチされ、キャッシュに保存され、処理された応答。保管可能なミスの数を概算する必要があります。保存不可能なミスは含まれません。

コマンドラインインターフェイスを使用してサマリーキャッシュ統計を表示するには、次の手順を実行します。

コマンドプロンプトで入力します:

```
stat cache
```

コマンドラインインターフェイスを使用して特定のキャッシュ統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します:

```
stat cache -detail
```

```
> stat cache -detail

Integrated Cache Statistics - Detail
Integrated Cache Statistics - Summary
                                          Rate (/s)
                                             Total


Hits                                              0
                        0

Misses                                            0
                        0

Requests                                          0
                        0

Hit ratio(%)                                     --
                        0

Origin bandwidth saved(%)                        --
                        0
Cached objects                                   --
                        0

```



```
Marker objects                                          --
                        0
                                                Rate (/s)
                                                   Total

Requests                                                 0
                        0


Hit Statistics

                                                Rate (/s)
                                                   Total


Non-304 hits                                             0
                        0

304 hits                                                 0
                        0


Sql hits                                                 0
                        0


Hits                                                     0
                        0

304 hit ratio(%)                                        --
                        0

Hit ratio(%)                                            --
                        0

Origin bandwidth saved(%)                               --
                        0
Byte Statistics
                                                Rate (/s)
                                                   Total


Bytes served by NetScaler                              648
    55379204

Bytes served by cache                                    0
                        0
Byte hit ratio(%)                                       --
                        0
Compressed bytes from cache                              0
                        0

Miss Statistics
```



```
56
57                                                  Rate (/s)
                                                       Total
58
59
60  Storable misses                                         0
                          0
61
62  Non-storable misses                                     0
                          0
63
64  Misses                                                  0
                          0
65
66  Revalidations                                           0
                          0
67
68  Successful revalidations                                0
                          0
69
70  Conversions to conditional req                          0
                          0
71
72
73  Storable miss ratio(%)                                 --
                          0
74  Successful reval ratio(%)                              --
                          0
75
76  Flashcache Statistics
77                                                  Rate (/s)
                                                       Total
78
79
80  Expire at last byte                                     0
                          0
81
82  Flashcache misses                                       0
                          0
83  Flashcache hits                                         0
                          0
84
85  Invalidation Statistics
86
87                                                  Rate (/s)
                                                       Total
88
89  Parameterized inval requests                            0
                          0
90
91
92  Full inval requests                                     0
                          0
```



```
 93
 94
 95
 96  Inval requests                                      0
                          0
 97
 98  Parameterized Caching Statistics
 99
100                                                 Rate (/s)
                                                       Total
101
102
103  Parameterized requests                              0
                          0
104
105  Parameterized non-304 hits                          0
                          0
106
107  Parameterized 304 hits                              0
                          0
108
109
110  Total parameterized hits                            0
                          0
111
112  Parameterized 304 hit ratio(%)                     --
                          0
113
114  Poll Every Time (PET) Statistics
115
116                                                 Rate (/s)
                                                       Total
117
118
119  Poll every time requests                            0
                          0
120
121  Poll every time hits                                0
                          0
122
123  Poll every time hit ratio(%)                       --
                          0
124
125  Memory Usage Statistics
126                                                     Total
127
128  Maximum memory(KB)                                  0
129
130  Maximum memory active value(KB)                     0
131
132  Utilized memory(KB)                                 0
133
134  Memory allocation failures                          0
```



```
135
136  Largest response so far(B)                          0
137
138  Cached objects                                      0
139
140  Marker objects                                      0
141
142  Hits being served                                   0
143  Misses being handled                                0
144   Done
145  <!--NeedCopy-->
```

GUI を使用してサマリーキャッシュ統計を表示するには

1. ページの上部にある [ダッシュボード] タブをクリックします。
2. ウィンドウの [統合キャッシュ] セクションまでスクロールします。
3. 詳細な統計を表示するには、表の下部にある [More…] リンクをクリックします。

GUI を使用して特定のキャッシュ統計を表示するには

1. ページの上部にある [レポート] タブをクリックします。
2. [組み込みレポート] で、[統合キャッシュ] を展開し、表示する統計情報を含むレポートをクリックします。
3. レポートをテンプレートとして保存するには、[名前を付けて保存] をクリックし、レポートの名前を指定します。保存したレポートが [カスタムレポート] の下に表示されます。

キャッシュされたオブジェクトとキャッシュ統計を表示する

August 15, 2023

キャッシュされた特定のオブジェクトを表示したり、キャッシュヒット、ミス、メモリ使用量に関する要約統計を表示したりできます。この統計は、キャッシュから提供されるデータの量、最大のパフォーマンス上の利点の原因となる項目、およびキャッシュのパフォーマンスを向上させるために調整できる内容に関する洞察を提供します。

ここでは、次の詳細について説明します。

- キャッシュされたオブジェクトの表示
- 特定のキャッシュされたレスポンスの検索
- キャッシュ統計の表示

キャッシュされたオブジェクトの表示

キャッシュを有効にすると、キャッシュされたオブジェクトの詳細を表示できます。たとえば、次の項目を表示できます。

- レスポンスサイズとヘッダーサイズ
- ステータスコード
- コンテンツグループ
- ETag、最終変更ヘッダー、およびキャッシュ制御ヘッダー
- URL をリクエストする
- ヒットパラメータ
- 宛先 IP アドレス
- リクエストと応答時間

コマンドラインインターフェイスを使用してキャッシュされたオブジェクトのリストを表示するには

コマンドプロンプトで入力します。

```
show cache object
```

プロパティ	仕様
応答サイズ (バイト)	レスポンスヘッダーとボディのサイズ。
レスポンスヘッダーのサイズ (バイト)	レスポンスのヘッダー部分のサイズ。
レスポンスステータスコード	レスポンスとともに送信されるステータスコード。
ETag	レスポンスに挿入されたETagヘッダー。通常、このヘッダーは、応答が最近変更されたかどうかを示します。
最終変更日	レスポンスに挿入された Last-Modified ヘッダー。このヘッダーは、レスポンスが最後に変更された日付を示します。
Cache-Control	レスポンスに挿入された Cache-Control ヘッダー。
日付	応答がいつ送信されたかを示す Date ヘッダー。
Contentgroup	レスポンスが格納されるコンテンツグループ。
複合マッチ	このオブジェクトがパラメータ化された値に基づいてキャッシュされている場合、このフィールド値は YES です。
ホスト	このレスポンスを要求した URL で指定されたホスト。
ホストポート	このレスポンスを要求した URL で指定されたホストのリスニングポート
URL	格納されたレスポンスに対して発行された URL。
接続先 IP	このレスポンスが取得されたサーバーの IP アドレス。
Destination port	宛先サーバーのリスニングポート。

プロパティ	仕様
ヒットパラメータ	レスポンスを格納するコンテンツグループがヒットパラメータを使用している場合は、このフィールドに表示されます。
ヒットセレクタ	このコンテンツグループがヒットセレクタを使用している場合は、このフィールドに表示されます。
無効なセレクタ	このコンテンツグループが無効化セレクタを使用している場合は、このフィールドに表示されます。
セレクタ式	このコンテンツグループがセレクタを使用している場合、このフィールドには選択ルールを定義する式が表示されます。
要求時間	リクエストが発行されてからの時間（ミリ秒）。
応答時間	キャッシュがレスポンスの受信を開始してから経過した時間（ミリ秒）。
年齢	オブジェクトがキャッシュ内にある時間。
有効期限	オブジェクトが期限切れとしてマークされるまでの時間。
フラッシュされた	有効期限が切れた後に応答がフラッシュされたかどうか。
プリフェッチ	このコンテンツグループに対してプリフェッチが設定されている場合、オブジェクトがオリジンからフェッチされる有効期限までの時間。プリフェッチはネガティブオブジェクトには適用されません (たとえば、404「オブジェクトが見つかりません」というレスポンス)。
現在の読者	現在処理されているヒット数とほぼ同じです。Content-Length ヘッダーオブジェクトを含むレスポンスがダウンロードされている場合、現在のミスおよび現在のリーダー値はそれぞれ 1 になります。チャンクレスポンスオブジェクトがダウンロードされている場合、現在のミス値は通常 1 になりますが、クライアントに対して提供されるチャンク応答は統合キャッシュバッファから送信されないため、現在のリーダー値は通常 0 になります。
現在のミス	キャッシュミスおよびオリジンサーバーからのフェッチの結果となったリクエストの現在の数。この値は、通常 0 または 1 です。コンテンツグループに対して [毎回ポーリング] が有効になっている場合、カウントは 1 より大きくなる可能性があります。
ヒット数	このオブジェクトのキャッシュヒット数。

プロパティ	仕様
ミス	このオブジェクトのキャッシュミスの数。
圧縮形式	このオブジェクトに適用される圧縮のタイプ。圧縮形式としては、gzip、deflate、compress、pack200-gzip などがある。
応答中の HTTP バージョン	レスポンスの送信に使用された HTTP のバージョン。
反応が弱いetag	エンティティのビットが変更されると、ストロングetagヘッダも変更されます。強力なヘッダーは、オブジェクトのオクテット値に基づいています。エンティティの意味が変わると、弱etagヘッダも変わります。弱いetag値はセマンティックアイデンティティに基づいています。弱いetags値は「W」で始まります。
ネガティブマーカーセル	マーカーオブジェクトはキャッシュ可能ですが、キャッシュされるための条件をすべて満たしているわけではありません。たとえば、オブジェクトがコンテンツグループの最大応答サイズを超えている可能性があります。このタイプのオブジェクトに対してマーカーセルが作成されます。ユーザーが次回このオブジェクトに対するリクエストを送信すると、キャッシュミスが提供されます。
理由マーカーが作成されました	マーカーセルが作成された理由 (たとえば、「minhit を待っている」、「コンテンツ長の応答データがグループサイズの制限にありません」など)。
毎回オートポーリング	統合キャッシュは、バリデーター（Last-Modified またはレスポンスヘッダーのいずれか）で既に期限切れの 200 OK 応答を受け取ると、ETag その応答を保存して Auto-PET（毎回自動的にポーリング）としてマークします。
NetScaler Etag がレスポンスで挿入されました	ETag NetScaler アプライアンスによって生成されるヘッダーのバリエーション。NetScaler が応答にEtagを挿入すると、「YES」という値が表示されます。
キャッシュに完全なレスポンスが存在	これが完全な応答であるかどうかを示します。
DNS によって検証された宛先 IP	オブジェクトの格納時に DNS 解決が実行されたかどうかを示します。
キャッシュフォワードプロキシを介して格納されたオブジェクト	統合キャッシュに設定されたフォワードプロキシが原因で、この応答が格納されたかどうかを示します。
オブジェクトはデルタベースファイルです	デルタ圧縮された応答。

プロパティ	仕様
minhits 待ち	レスポンスをキャッシュする前に、このコンテンツグループで最小数のオリジンサーバーヒットが必要かどうかを示します。
ミニヒットカウント	このコンテンツグループが、オブジェクトをキャッシュする前にオリジンサーバーのヒット数を最小限に抑える必要がある場合、このフィールドにはこれまでに受信されたヒット数が表示されます。
HTTP リクエストメソッド	このオブジェクトを取得したリクエストで使用されるメソッド GET または POST。
ポリシー別に格納	このオブジェクトが格納された原因となったキャッシュポリシーの名前。[利用不可] の値は、ポリシーが非アクティブ化または削除されたことを示します。NONE の値は、オブジェクトが可視ポリシーと一致せず、キャッシュの内部基準に従って格納されたことを示します。
アプリケーションファイアウォールのメタデータが存在します	このパラメータは、アプリケーションファイアウォールと統合キャッシュの両方が有効になっている場合に使用されます。アプリケーションファイアウォールは、レスポンスページのコンテンツを分析し、そのメタデータ（ページに含まれる URL やフォームなど）を保存し、レスポンスを含むメタデータをキャッシュにエクスポートします。キャッシュはページとメタデータを格納し、キャッシュがページを提供すると、メタデータをリクエストのセッションに送り返します。
HTTP コールアウトオブジェクト、名前、タイプ、応答	これらのセルは、HTTP コールアウト式の結果としてこのデータが格納されたかどうかを示し、コールアウトのさまざまな側面と対応する応答に関する情報を提供します。HTTP コールアウトの詳細については、「HTTP コールアウト」を参照してください。

キャッシュされた特定のレスポンスを見つける

検索条件に基づいて、キャッシュ内の個々のアイテムを検索できます。キャッシュされたアイテムを検索するには、データを含むコンテンツグループがヒットセレクターと無効化セレクターを使用するかどうかに応じて、次のようにさまざまな方法があります。

コンテンツグループがセレクターを使用している場合、キャッシュされたアイテムのロケータ ID を使用してのみ検索を実行できます。

コンテンツグループがセレクタを使用しない場合は、URL、ホスト、コンテンツグループ名などの基準を使用して検索を実行します。

キャッシュされたレスポンスを検索するときに、URL とホストでいくつかの項目を見つけることができます。レスポンスがセレクタを使用するコンテンツグループ内にある場合は、ロケータ番号（たとえば、0x00000000ad7af00000050）を使用してのみ検索できます。後で使用するためにロケータ番号を保存するには、エントリを右クリックし、[コピー] を選択します。セレクタの詳細は、「セレクタおよび基本コンテンツグループの構成」を参照してください。

コマンドラインインターフェイスを使用して、セレクタを持たないコンテンツグループのキャッシュされたレスポンスを表示するには

コマンドプロンプトで入力します。

```
show cache object [-locator <positiveInteger>] | [(-url <URL> (-
host <hostName> [-port <port>] [-groupName <contentGroupName>] [-
httpMethod GET | POST ]))| [-httpStatus<positive integer>] | -group <
contentGroupName> | -ignoreMarkerObjects (ON | OFF)| -includeNotReadyObjects
 (ON | OFF)]
```

コマンドラインインターフェイスを使用してセレクタを持つコンテンツグループのキャッシュされたレスポンスを表示するには

コマンドプロンプトで入力します。

```
show cache object -locator <locatorString> MarkerObjects ( ON | OFF )
| -includeNotReadyObjects ( ON | OFF )| [-httpStatus<positive integer
>]
```

GUI を使用してセレクターのないコンテンツグループにキャッシュされたレスポンスを表示するには

[最適化] > [統合キャッシュ] > [キャッシュオブジェクト] に移動し、検索条件を設定して必要なキャッシュレスポンスを表示します。

コンテンツグループをまだ構成していない場合は、すべてのオブジェクトが Default グループに属します。

GUI を使用してセレクターのあるコンテンツグループにキャッシュされたレスポンスを表示するには

[最適化] > [統合キャッシュ] > [キャッシュオブジェクト] に移動し、[検索] をクリックして、必要なキャッシュされたレスポンスを表示するためのセレクター検索条件を設定します。

キャッシュ統計の表示

次の表は、キャッシュ統計をまとめたものです。

カウンター

仕様

キャッシュ統計の表示

更新日:2013-10-28

次の表に、表示できる詳細なキャッシュ統計情報をまとめたものです。

カウンター	Specifies
ヒット数	統合キャッシュで見つかり、統合キャッシュから提供されるレスポンス。画像ファイルなどの静的オブジェクト、ステータスコードが 200、203、300、301、302、304、304、307、403、404、410 のページ、およびユーザー定義ポリシーと CACHE アクションに一致する応答が含まれます。
ミス	レスポンスが最終的にオリジンサーバーからフェッチされた HTTP リクエストをインターセプトしました。
要求	キャッシュヒット数の合計とキャッシュミスの合計。
304 ヒット以外	ユーザーがアイテムを複数回要求し、NetScaler ADC アプライアンスが最後にアイテムを提供してからキャッシュ内のアイテムが変更されていない場合、NetScaler ADC アプライアンスはキャッシュされたオブジェクトの代わりに 304 の応答を提供します。この統計は、NetScaler アプライアンスがキャッシュから処理したアイテムの数（304 レスポンスを除く）を示します。
304 ヒット	NetScaler アプライアンスがキャッシュから処理した 304（オブジェクトは変更されていない）応答の数。
304 ヒット率 (%)	NetScaler アプライアンスが処理した 304 件の回答のうち、他の回答と比較した割合。
ヒット率 (%)	キャッシュから処理できなかった応答に対する、NetScaler アプライアンスがキャッシュから処理した応答（キャッシュヒット）の割合。
配信元帯域幅の節約 (%)	キャッシュからの応答を処理することによって NetScaler アプライアンスが配信元サーバーで節約した処理能力の推定値。
NetScaler が処理するバイト数	NetScaler アプライアンスがオリジンサーバーとキャッシュから提供したバイトの総数。
キャッシュが処理するバイト数	NetScaler アプライアンスがキャッシュから処理したバイトの総数。
バイトヒット率 (%)	NetScaler アプライアンスがキャッシュから処理したデータの割合（処理されたすべての応答のすべてのデータに対する割合）。

カウンター	Specifies
キャッシュから圧縮されたバイト数	NetScaler アプライアンスが圧縮形式で提供したデータ量（バイト単位）。
保存可能なミス	NetScaler アプライアンスは、要求されたオブジェクトをキャッシュで見つけられない場合、オリジンサーバーからオブジェクトを取得します。これは、キャッシュミスと呼ばれます。保存可能なキャッシュミスをキャッシュに保存できます。
保存できないミス	保存できないキャッシュミスはキャッシュに保存できません。
ミス	すべてのキャッシュミス。
再検証	Cache-Control ヘッダーの Max-Age 設定により、中間キャッシュが統合キャッシュでコンテンツを再検証してからユーザーに提供するタイミングを秒単位で決定します。詳しい情報については、「キャッシュコントロールヘッダーの挿入」を参照してください。
再検証成功	実行された再検証の数。詳しい情報については、「キャッシュコントロールヘッダーの挿入」を参照してください。
条件付き要求への変換	キャッシュされた PET オブジェクトに対するユーザーエージェントリクエストは、常に条件付きリクエストに変換され、オリジンサーバーに送信されます。詳しい情報については、「リクエストを受信するたびにオリジンサーバーをポーリングする」を参照してください。
保存可能なミス率 (%)	保存可能なキャッシュミスの割合。保存できないキャッシュミスの割合。
成功率 (%)	すべての再検証試行に対する再検証の成功率。詳しい情報については、「キャッシュコントロールヘッダーの挿入」を参照してください。
最後のバイトで期限切れ	最後のボディバイトを受け取った直後にキャッシュがコンテンツを期限切れにした回数。「キャッシュのヒットとミスの数」の表で説明されているように、肯定的な回答にのみ適用されます。詳しい情報については、「パフォーマンス最適化の例」を参照してください。
フラッシュキャッシュミス	Flash Cache を有効にすると、キャッシュがサーバーに到達するリクエストは 1 つだけ許可されるため、フラッシュの混雑がなくなります。この統計は、キャッシュミスであったフラッシュキャッシュ要求の数を示します。詳細については、「リクエストをキャッシュにキューイングする」を参照してください。

カウンター	Specifies
Flashcache ヒット	キャッシュヒットしたフラッシュキャッシュリクエストの数。詳しい情報については、「リクエストをキャッシュにキューに入れる」を参照してください。
パラメータ化された無効なリクエスト	無効化（INVAL）アクションを含むポリシーと、無効化セレクターまたはパラメータを使用してグループ内のキャッシュオブジェクトを選択的に期限切れにするコンテンツグループに一致するリクエスト。
すべてのリクエストが無効です	InvalGroups パラメータが設定されている無効化ポリシーに一致し、1 つ以上のコンテンツグループが期限切れになるリクエスト。
リクエストが無効です	無効化ポリシーに一致し、特定のキャッシュされたレスポンスまたはコンテンツグループ全体が期限切れになるリクエスト。
パラメータ化されたリクエスト	パラメータ化されたコンテンツグループのポリシーを使用して処理されたキャッシュリクエストの数。
パラメータ化された非 304 ヒット	パラメータ化されたコンテンツグループを含むポリシーを使用して処理されたキャッシュリクエストのうち、キャッシュされたレスポンスがすべて見つかり、レスポンスが 304（オブジェクト未更新）レスポンスではなかったものの数。
パラメータ化された 304 ヒット	パラメータ化されたコンテンツグループを含むポリシーを使用して処理されたキャッシュリクエストの数。キャッシュされたオブジェクトが見つかり、そのオブジェクトは 304（オブジェクト未更新）のレスポンスでした。
パラメータ化されたヒット数の合計	パラメータ化されたコンテンツグループを含むポリシーを使用して処理された、キャッシュされたオブジェクトが見つかったキャッシュリクエストの数。
パラメータ化された 304 ヒット率 (%)	すべてのキャッシュヒットに対する、パラメータ化されたポリシーを使用して見つかった 304（オブジェクトは更新されていない）応答の割合。
リクエストのたびに投票	Poll Every Time が有効になっている場合、NetScaler アプライアンスは保存されたオブジェクトを提供する前に常にオリジンサーバーと照合します。詳しい情報については、「リクエストを受信するたびにオリジンサーバーをポーリングする」を参照してください。

カウンター	Specifies
ヒットするたびに投票	Poll Every Time メソッドを使用してキャッシュヒットが見つかった回数。詳しい情報については、「リクエストを受信するたびにオリジンサーバーをポーリングする」を参照してください。
毎回の投票ヒット率 (%)	Poll Every Time メソッドを使用したキャッシュ・ヒットの割合。Poll Every Time を使用したキャッシュ・オブジェクトのすべての検索を基準としています。詳しい情報については、「リクエストを受信するたびにオリジンサーバーをポーリングする」を参照してください。
最大メモリ (KB)	キャッシュに割り当てられる NetScaler アプライアンスの最大メモリ量。詳しい情報については、「キャッシュ用のグローバル属性の設定」を参照してください。
最大メモリーアクティブ値 (KB)	メモリが実際にキャッシュに割り当てられた後に設定される最大メモリ量 (アクティブ値)。詳しくは、「さまざまなシナリオに合わせて NetScaler アプライアンスの統合キャッシュ機能を構成する方法」を参照してください。
使用済みメモリ (KB)	実際に使用されているメモリの量。
メモリ割り当ての失敗	応答をキャッシュに保存する目的でメモリを利用しようとして失敗した回数。
これまでで最大の反響を呼んだ	キャッシュまたはオリジンサーバーで見つかり、クライアントに送信された最大レスポンス（バイト単位)。
キャッシュされたオブジェクト	まだ完全にダウンロードされていないレスポンス、有効期限が切れているがまだフラッシュされていないレスポンスを含む、キャッシュ内のオブジェクトの数。
マーカーオブジェクト	マーカーオブジェクトは、応答がコンテンツグループの最大または最小応答サイズを超えるか、コンテンツグループの最小ヒット数をまだ受け取っていない場合に作成されます。
配信中のヒット数	キャッシュから処理されたヒット数。
処理中のミス	オリジンサーバーから取得され、キャッシュに保存されてから処理されたレスポンス。保管可能なミスの数を概算する必要があります。保存できないミスは含まれません。

コマンドラインインターフェイスを使用してサマリーキャッシュ統計を表示するには

コマンドプロンプトで入力します。

stat cache

コマンドラインインターフェイスを使用して特定のキャッシュ統計を表示するには

コマンドプロンプトで入力します。

```
stat cache -detail

    > stat cache -detail
    Integrated Cache Statistics - Detail
    Integrated Cache Statistics - Summary
                                                    Rate (/s)
                                                       Total
    Hits                                                    0
                            0
    Misses                                                  0
                            0
    Requests                                                0
                            0
    Hit ratio(%)                                           --
                            0
    Origin bandwidth saved(%)                              --
                            0
    Cached objects                                         --
                            0
    Marker objects                                         --
                            0
                                                    Rate (/s)
                                                       Total
    Requests                                                0
                            0
    Hit Statistics
                                                    Rate (/s)
                                                       Total
    Non-304 hits                                            0
                            0
    304 hits                                                0
                            0
    Sql hits                                                0
                            0
    Hits                                                    0
                            0
    304 hit ratio(%)                                       --
                            0
    Hit ratio(%)                                           --
                            0
    Origin bandwidth saved(%)                              --
                            0

    Byte Statistics
                                                    Rate (/s)
                                                       Total
    Bytes served by NetScaler                           648
       55379204
```



```
29      Bytes served by cache                                  0
                            0
30      Byte hit ratio(%)                                     --
                            0
31      Compressed bytes from cache                            0
                            0
32      Miss Statistics
33                                                   Rate (/s)
                                                        Total
34      Storable misses                                        0
                            0
35      Non-storable misses                                    0
                            0
36      Misses                                                 0
                            0
37      Revalidations                                          0
                            0
38      Successful revalidations                               0
                            0
39      Conversions to conditional req                         0
                            0
40      Storable miss ratio(%)                                --
                            0
41      Successful reval ratio(%)                             --
                            0
42      Flashcache Statistics
43                                                   Rate (/s)
                                                        Total
44      Expire at last byte                                    0
                            0
45      Flashcache misses                                      0
                            0
46      Flashcache hits                                        0
                            0
47
48      Invalidation Statistics
49                                                   Rate (/s)
                                                        Total
50      Parameterized inval requests                           0
                            0
51      Full inval requests                                    0
                            0
52      Inval requests                                         0
                            0
53
54      Parameterized Caching Statistics
55                                                   Rate (/s)
                                                        Total
56      Parameterized requests                                 0
                            0
57      Parameterized non-304 hits                             0
                            0
58      Parameterized 304 hits                                 0
```



```
                              0
59      Total parameterized hits                              0
                              0
60      Parameterized 304 hit ratio(%)                       --
                              0
61
62      Poll Every Time (PET) Statistics
63                                                 Rate (/s)
                                                      Total
64      Poll every time requests                              0
                              0
65      Poll every time hits                                  0
                              0
66      Poll every time hit ratio(%)                         --
                              0
67      Memory Usage Statistics
68                                                        Total
69      Maximum memory(KB)                                    0
70      Maximum memory active value(KB)                       0
71      Utilized memory(KB)                                   0
72      Memory allocation failures                            0
73      Largest response so far(B)                            0
74      Cached objects                                        0
75      Marker objects                                        0
76      Hits being served                                     0
77      Misses being handled                                  0
78       Done
79  <!--NeedCopy-->
```

GUI を使用してサマリーキャッシュ統計を表示するには

1. ページの上部にある [ダッシュボード] タブをクリックします。
2. ウィンドウの [統合キャッシュ] セクションまでスクロールします。
3. 詳細な統計を表示するには、表の下部にある [More…] リンクをクリックします。

GUI を使用して特定のキャッシュ統計を表示するには

1. ページの上部にある [レポート] タブをクリックします。
2. [組み込みレポート] で、[統合キャッシュ] を展開し、表示する統計情報を含むレポートをクリックします。
3. レポートをテンプレートとして保存するには、[名前を付けて保存] をクリックし、レポートの名前を指定します。保存したレポートが [カスタムレポート] の下に表示されます。

キャッシュのパフォーマンスを向上させる

August 15, 2023

統合キャッシュのパフォーマンスを向上させることができます。たとえば、同じキャッシュデータに対する同時リク

エストの処理、オリジンサーバーからのキャッシュされたレスポンスの更新に伴う遅延の回避、キャッシュするだけの頻度でレスポンスがリクエストされるようにするなどです。

フラッシュの混雑を軽減

フラッシュクラウドは、多数のユーザーが同じデータを同時にリクエストしたときに発生します。オブジェクト全体がダウンロードされた後にのみヒットを処理するようにキャッシュを設定すると、フラッシュクラウド内のリクエストがキャッシュミスになる可能性があります。

次のテクニックを使うと、フラッシュの混雑を減らすか、なくすことができます。

- **PREFETCH**: 有効期限が切れる前に肯定的な回答を更新して、古くなったり非アクティブになったりしないようにします。詳細については、「有効期限が切れる前の応答の更新」セクションを参照してください。
- キャッシュバッファリング: 応答全体がダウンロードされるのを待つのではなく、オリジンサーバーから応答ヘッダーを受信すると、複数のクライアントに応答の提供を開始します。レスポンスを同時にダウンロードできるクライアント数の制限は、使用可能なシステムリソースだけです。NetScaler アプライアンスは、ダウンロードを開始したクライアントがダウンロードが完了する前に停止した場合でも、ダウンロードして応答を提供します。応答がキャッシュサイズを超えるか、応答がチャンク化されている場合、キャッシュは応答の保存を停止しますが、クライアントへのサービスは中断されません。
- **Flash Cache:** Flash Cache はリクエストをキャッシュにキューに入れ、一度に 1 つのリクエストのみがサーバーに到達できるようにします。

詳しい情報については、「キャッシュへのリクエストのキューイング」セクションを参照してください。

有効期限が切れる前に回答を更新する

キャッシュされた応答を必要なときにいつでも最新の状態に保つために、PREFETCH オプションは計算された有効期限が切れる前に応答を更新します。プリフェッチ間隔は、最初のクライアント要求を受信した後に計算されます。それ以降、NetScaler アプライアンスは、PREFETCH パラメーターで設定した時間間隔でキャッシュされた応答を更新します。

この設定は、リクエスト間で頻繁に更新されるデータに役立ちます。否定的な応答（404 メッセージなど）には適用されません。

コマンドラインインターフェイスを使用してコンテンツグループのプリフェッチを設定するには

コマンドプロンプトで入力します。

```
set cache contentgroup <name> -prefetch YES [-prefetchPeriod <seconds> | -prefetchPeriodMilliSec <milliseconds>] [-prefetchMaxPending <positiveInteger>]
```

* GUI を使用してコンテンツグループのプリフェッチを設定するには

最適化に移動 > 統合キャッシング > コンテンツグループ、および コンテンツグループを選択します。

「その他」タブの「Flash Crowd とプリフェッチ」グループで「プリフェッチ」オプションを選択し、「間隔」テキストボックスと「保留中のプリフェッチの最大数」テキストボックスに値を指定します。

リクエストをキャッシュにキューに入れる

Flash Cache オプションでは、同時に届く要求 (フラッシュクラウド) をキューに入れ、応答を取得して、その要求がキューに入っているすべてのクライアントに配信します。このプロセス中に応答がキャッシュできなくなった場合、NetScaler アプライアンスはキャッシュからの応答の処理を停止し、代わりに元のサーバーの応答をキューに入れたクライアントに提供します。応答がない場合、クライアントはエラーメッセージを受け取ります。

フラッシュキャッシュはデフォルトでは無効になっています。同じコンテンツグループで Poll Every Time（PET）とフラッシュキャッシュを有効にすることはできません。

Flash Cache のデメリットの 1 つは、サーバーがエラーを返した場合 (たとえば、すぐに修正される 404)、エラーが待機中のクライアントに拡散されることです。

> 注:
> Flash Cache が有効になっていると、NetScaler アプライアンスがクライアント要求の Accept-Encoding ヘッダーを応答内の Content-Encoding ヘッダーと正しく照合できない場合があります。NetScaler アプライアンスは、これらのヘッダーが一致すると想定して、誤ってヒットする可能性があります。回避策として、統合キャッシュポリシーを設定して、適切な Accept-Encoding ヘッダーを持たないクライアントにヒットを提供しないようにすることができます。

コマンドラインインターフェイスを使用して Flash Cache を有効にするには

コマンドプロンプトで入力します。

```
set cache contentgroup <contentGroupName> -flashcache yes
```

GUI を使用してフラッシュキャッシュを有効にするには

[最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。

「その他」タブの「Flash Crowd とプリフェッチ」グループで、「プリフェッチ」オプションを選択します。

クライアントがダウンロードを停止した後に応答をキャッシュする

Quick Abort パラメータを設定すると、応答がキャッシュに入る前にクライアントが要求を停止した場合でも、応答をキャッシュし続けることができます。

ダウンロードした応答サイズが Quick Abort のサイズ以下の場合、NetScaler アプライアンスは応答のダウンロードを停止します。Quick Abort パラメータを 0 に設定すると、すべてのダウンロードが中止されます。

コマンドラインインターフェイスを使用してクイックアボートサイズを設定するには

コマンドプロンプトで入力します。


```
set cache contentgroup <name> -quickAbortSize <integerInKBytes>
```

GUI を使用してクイックアボートサイズを設定するには

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. [メモリ] タブの [クイック中止: テキストボックスより多い場合はキャッシュを続行] で適切な値を設定します。

キャッシュする前に最低限のサーバーヒット数が必要です

レスポンスをキャッシュする前に、オリジンサーバーでレスポンスが見つかるまでの最小回数を設定できます。キャッシュメモリがすぐにいっぱいになり、ヒット率が予想よりも低い場合は、最小ヒット数を増やすことを検討する必要があります。

最小ヒット数のデフォルト値は 0 です。この値は、最初のリクエストの後の応答をキャッシュします。

コマンドラインインターフェイスを使用してキャッシュする前に必要な最小ヒット数を設定するには

コマンドプロンプトで入力します。

```
set cache contentgroup <name> -minhits <positiveInteger>
```

GUI を使用してキャッシュする前に必要な最小ヒット数を設定するには

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. ヒット数がテキストボックスより少ない場合は、[メモリ] タブの [キャッシュしない] で適切な値を設定します。

パフォーマンス最適化の例

この例では、クライアントが株価にアクセスします。株価は非常に動的です。統合キャッシュは、オリジンサーバーに複数のリクエストを送信せずに同じ株価を同時クライアントに提供するように構成します。株価はクライアントにダウンロードされた後に期限切れになり、次のリクエストはオリジンサーバーから取得されます。これにより、見積もりが常に最新の状態に保たれます。

次のタスク概要では、株価アプリケーションのキャッシュを設定する手順について説明します。

株価アプリケーションのキャッシュを設定

株価情報用のコンテンツグループの作成

詳しい情報については、「コンテンツグループについて」を参照してください。

このコンテンツグループには、次の設定を行います。

1.「有効期限方法」タブで、「回答の受信が完了したら期限切れにする」チェックボックスを選択します。
2.「その他」タブで「**Flash Cache** 」チェックボックスを選択し、「作成」をクリックします。

3. 株価をキャッシュするキャッシュポリシーを追加します。

詳しい情報については、「統合キャッシュのポリシーの設定」を参照してください。

ポリシーとして次の設定を行います

1. 「アクション」リストと「グループに保存」リストで、「キャッシュ」を選択し、前のステップで定義したグループを選択します。
2. 「追加」をクリックし、「式の追加」ダイアログボックスで、株価リクエストを識別する式を設定します。たとえば、http.req.url.contains (「cgi-bin/stock-quote.pl」)
3. ポリシーを有効にします。

詳しい情報については、「統合キャッシュポリシーをグローバルにバインドする」を参照してください。この例では、このポリシーを request-time 上書き処理にバインドし、プライオリティを低い値に設定します。

Cookie、ヘッダー、およびポーリングを構成する

December 8, 2023

このトピックでは、Cookie、HTTP ヘッダー、およびオリジンサーバーのポーリングを管理するキャッシュを構成する方法について説明します。これには、キャッシュが文書化された標準から外れるようなデフォルトの動作の変更、キャッシュ可能なコンテンツがキャッシュに保存されなくなる可能性がある HTTP ヘッダーのオーバーライド、更新されたコンテンツについてオリジンに常にポーリングするようにキャッシュを設定することが含まれます。

標準からのキャッシュ動作の相違について

デフォルトでは、統合キャッシュは次の RFC 標準に準拠しています。

- RFC 2616「HTTP HTTP/1.1」
- RFC 2617「HTTP 認証: 基本認証とダイジェストアクセス認証」で説明されているキャッシュ動作
- RFC 2965「HTTP 状態管理メカニズム」で説明されているキャッシュ動作

組み込みのポリシーと [Default] コンテンツグループ属性により、これらの標準のほとんどに準拠していることが保証されます。

デフォルトの統合キャッシュの動作は、以下のように仕様とは違います。

- ヴァリヘッダは限定的にサポートされています。デフォルトでは、ヴァリヘッダーを含むレスポンスは、圧縮されない限りキャッシュ不可とみなされます。圧縮された応答には、コンテンツエンコーディング:gzip、コンテンツエンコーディング:deflate、またはコンテンツエンコーディング:pack200-gzip が含まれており、Vary: Accept-encoding ヘッダーが含まれていてもキャッシュ可能です。

- 統合キャッシュは、ヘッダーの cache-control (no-cache および cache-control: private) の値を無視します。たとえば、cache-control: no-cache= “set-cookie” を含むレスポンスは、レスポンスに Cache-Control: no-cache が含まれているかのように扱われます。デフォルトでは、レスポンスはキャッシュされません。
- イメージレスポンスに set-cookie または set-cookie2 ヘッダーが含まれている場合や、イメージリクエストに Cookie ヘッダーが含まれている場合でも、イメージ（content-type = image/*）は常にキャッシュ可能と見なされます。統合キャッシュは、set-cookie と set-cookie2 ヘッダーをキャッシュする前にレスポンスから削除します。これは RFC 2965 とは違います。RFC 準拠の動作は次のように設定できます。

```
1 add cache policy rfc_compliant_images_policy -rule "http.res.header.set
     -cookie2.exists || http.res.header.set-cookie.exists" -action
     NOCACHE
2
3
4 bind cache global rfc_compliant_images_policy -priority 100 -type
     REQ_OVERRIDE
5 <!--NeedCopy-->
```

- リクエスト内の次のキャッシュ制御ヘッダーは、RFC 準拠のキャッシュにオリジンサーバーからキャッシュされたレスポンスをリロードするように強制します。

`Cache-control: max-age=0`

`Cache-control: no-cache`

サービス拒否攻撃を防ぐため、この動作はデフォルトではありません。

- デフォルトでは、キャッシュモジュールは、レスポンスヘッダーの状態がそうでない場合を除き、レスポンスはキャッシュ可能であるとみなします。この動作を RFC 2616 に準拠させるには、すべてのコンテンツグループに対して`-weakPosRelExpiry`と`-weakNegResExpiry`を 0 に設定します。

レスポンスからクッキーを削除する

クッキーはユーザーに合わせてカスタマイズされることが多く、通常はキャッシュすべきではありません。`Remove Response Cookies`パラメーターは、レスポンスをキャッシュする前に`Set-Cookie and Set-Cookie2`ヘッダーを削除します。既定では、コンテンツグループの`Remove Response Cookies`オプションでは、`Set-Cookie`ヘッダーまたは`Set-Cookie2`ヘッダーを含む応答のキャッシュは禁止されています。

> 注:
> イメージがキャッシュされる場合、組み込みの動作では、コンテンツグループがどのように設定されていても、キャッシュの前に`Set-Cookie`ヘッダーと`Set-Cookie2`ヘッダーが削除されます。

> 画像などの埋め込みレスポンスを保存するすべてのコンテンツグループに対して、デフォルトRemove Response Cookiesを使用することをお勧めします。

コマンドラインインターフェイスを使用してコンテンツグループに対してRemove Response Cookiesを設定するには、次の手順を実行します。

コマンドプロンプトで入力します:

```
set cache contentgroup <name> -removeCookies YES
```

NetScaler GUI を使用してコンテンツグループの応答 **Cookie** の削除を構成する

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. [その他] タブの [設定] グループで、[レスポンス Cookie の削除] オプションを選択します。

レスポンスタイムに **HTTP** ヘッダを挿入する

統合キャッシュは、キャッシュリクエストから返されるレスポンスに HTTP ヘッダーを挿入できます。NetScaler アプライアンスは、キャッシュミスによる応答のヘッダーを変更しません。

次の表に、レスポンスに挿入できるヘッダーについて説明します。

Header	仕様
年齢	オリジンサーバーでレスポンスが生成された時間から計算された、レスポンスの経過時間（秒）を提供します。デフォルトでは、キャッシュはキャッシュから提供されるすべてのレスポンスに Age ヘッダーを挿入します。
経由で	要求または応答の開始点と終了点の間にあるプロトコルと受信者を一覧表示します。NetScaler アプライアンスは、キャッシュから処理されるすべての応答に Via ヘッダーを挿入します。挿入されるヘッダーのデフォルト値は、NS-CACHE-10.0。NetScaler IP アドレスの最後のオクテットです。詳しい情報については、「キャッシュ用のグローバル属性の設定」を参照してください。

Header	仕様
Tag	コマンドラインインターフェイスを使用してコンテンツグループに対してTagを設定するには、次の手順を実行します。キャッシュは、レスポンスをキャッシュし、オリジンサーバーが独自のTagヘッダーを挿入していない場合にのみ、Tagをレスポンスに挿入します。Tag 値は任意の一意の数値です。レスポンスのTag値は、オリジンサーバーから更新されると変更されますが、サーバーが 304 (object Not updated) レスポンスを送信した場合は同じままです。動的コンテンツはキャッシュ不可と見なされるため、通常、オリジンサーバーは動的コンテンツのバリデータを生成しません。この動作はオーバーライドできます。Tag ヘッダーを挿入すると、キャッシュは完全なレスポンスを処理しないことが許可されます。代わりに、ユーザーエージェントは、統合キャッシュによって最初に送信された動的応答をキャッシュする必要があります。ユーザーエージェントにレスポンスを強制的にキャッシュさせるには、統合キャッシュを設定してTagヘッダーを挿入し、オリジンが提供する Cache-Control ヘッダーを置き換えます。
キャッシュコントロール	NetScaler アプライアンスは通常、オリジンサーバーから送信される応答のキャッシュ性ヘッダーを変更しません。オリジンサーバーがキャッシュ不可とラベル付けされた応答を送信した場合、NetScaler アプライアンスが応答をキャッシュしていても、クライアントはその応答をキャッシュ不可として扱います。動的レスポンスをユーザーエージェントにキャッシュするには、オリジンサーバーの Cache-Control ヘッダーを置き換えることができます。これは、ユーザーエージェントとその他の介在するキャッシュにのみ適用されます。統合キャッシュには影響しません。

Header	仕様
年齢	オリジンサーバーでレスポンスが生成された時間から計算された、レスポンスの経過時間（秒）を提供します。デフォルトでは、キャッシュはキャッシュから提供されるすべてのレスポンスに Age ヘッダーを挿入します。

Header	仕様
経由で	要求または応答の開始点と終了点の間にあるプロトコルと受信者を一覧表示します。NetScaler アプライアンスは、キャッシュから処理されるすべての応答に Via ヘッダーを挿入します。挿入されたヘッダーのデフォルト値は「NS-CACHE-9.2: NetScaler IP アドレスの最後のオクテット」です。詳しい情報については、「キャッシュ用のグローバル属性の設定」を参照してください。
Tag	キャッシュは、Last-Modified ヘッダーと Tag ヘッダーを使用した応答検証をサポートし、応答が古くなっているかどうかを判断します。キャッシュは、レスポンスをキャッシュし、オリジンサーバーが独自のTagヘッダーを挿入していない場合にのみ、Tagをレスポンスに挿入します。Tag 値は任意の一意の数値です。レスポンスのTag値は、オリジンサーバーから更新されると変更されますが、サーバーが 304 (object Not updated) レスポンスを送信した場合は同じままです。動的コンテンツはキャッシュ不可と見なされるため、通常、オリジンサーバーは動的コンテンツのバリデータを生成しません。この動作はオーバーライドできます。Tag ヘッダーを挿入すると、キャッシュは完全なレスポンスを処理しないことが許可されます。代わりに、ユーザーエージェントは、統合キャッシュによって最初に送信された動的応答をキャッシュする必要があります。ユーザーエージェントにレスポンスを強制的にキャッシュさせるには、統合キャッシュを設定してTagヘッダーを挿入し、オリジンが提供する Cache-Control ヘッダーを置き換えます。
キャッシュコントロール	NetScaler アプライアンスは通常、オリジンサーバーから送信される応答のキャッシュ性ヘッダーを変更しません。オリジンサーバーがキャッシュ不可とラベル付けされた応答を送信した場合、NetScaler アプライアンスが応答をキャッシュしていても、クライアントはその応答をキャッシュ不可として扱います。動的レスポンスをユーザーエージェントにキャッシュするには、オリジンサーバーの Cache-Control ヘッダーを置き換えることができます。これは、ユーザーエージェントとその他の介在するキャッシュにのみ適用されます。統合キャッシュには影響しません。

年齢、経由、またはタグヘッダーの挿入

次の手順では、Age、Via、および ETag ヘッダーを挿入する方法について説明します。

NetScaler コマンドインターフェイスを使用して、**Age**、**Via**、または **Etag** ヘッダーを挿入します。

コマンドプロンプトで入力します:

```
set cache contentgroup <name> -insertVia YES -insertAge YES -insertETag
 YES
```

NetScaler GUI を使用して **Age**、**Via**、または **Etag** ヘッダーを構成する

1. [最適化] > [統合キャッシュ] > [コンテンツグループ**]** に移動し、コンテンツグループを選択します。
2. [その他] タブの [HTTP ヘッダー挿入] グループで、必要に応じて [経由]、[経過日数]、または [**ETag**] オプションを選択します。
3. その他のヘッダータイプの値は自動的に計算されます。Via の値は、キャッシュのメイン設定で設定します。

キャッシュコントロールヘッダを挿入する

統合キャッシュが、オリジンサーバーが挿入した Cache-Control ヘッダーを置き換える場合、Expires ヘッダーも置き換えられます。新しい Expires ヘッダーには、過去の有効期限が含まれています。これにより、HTTP/1.0 クライアントと (Cache-Control ヘッダーを認識しない) キャッシュがコンテンツをキャッシュしないようにします。

NetScaler コマンドインターフェイスを使用してキャッシュ制御ヘッダーを挿入する

コマンドプロンプトで入力します:

```
set cache contentgroup <name> -cacheControl <value>
```

NetScaler GUI を使用してキャッシュ制御ヘッダーを挿入する

1. [最適化] > [統合キャッシュ] > [コンテンツグループ**]** に移動し、
    a) [有効期限方法] タブをクリックし、ヒューリスティックと既定の有効期限設定をクリアし、[コンテンツの有効期限が切れるまでの期間] テキストボックスで適切な値を設定します。
    b) [その他] タブをクリックし、[Cache-Control] テキストボックスに挿入するヘッダーを入力します。または、[Configure] をクリックして、キャッシュされたレスポンスに Cache-Control ディレクティブを設定します。

リクエスト内のキャッシュコントロールとプラグマヘッダーを無視する

デフォルトでは、キャッシュモジュールは Cache-Control ヘッダーと Pragma ヘッダーを処理します。Cache-Control ヘッダー内の次のトークンは、RFC 2616 の説明に従って処理されます。

- 最大年齢
- 最大失効しました
- キャッシュされた場合のみ有る
- キャッシュなし

リクエスト内の Pragma: no-cache ヘッダーは、Cache-Control: no-cache ヘッダーと同じように扱われます。

Cache-Control ヘッダーと Pragma ヘッダーを無視するようにキャッシュモジュールを構成すると、Cache-Control: No-Cache ヘッダーを含む要求により、NetScaler アプライアンスはオリジンサーバーから応答を取得しますが、キャッシュされた応答は更新されません。キャッシュモジュールが Cache-Control ヘッダーと Pragma ヘッダーを処理すると、キャッシュされたレスポンスが更新されます。

次の表に、これらのヘッダーのさまざまな設定と [ブラウザのリロード要求を無視] 設定の影響をまとめています。

キャッシュコントロールとプラグマヘッダーを無視するための設定	ブラウザのリロード要求を無視する設定	結果
はい	はいまたはいいえ	Cache-Control: no-cache ディレクティブを含め、クライアントからの Cache-Control ヘッダーと Pragma ヘッダーは無視してください。
番号	はい	Cache-Control: no-cache ヘッダーはキャッシュミスを生成しますが、すでにキャッシュにあるレスポンスは更新されません。
番号	番号	Cache-Control: no-cache ヘッダーを含むリクエストはキャッシュミスを引き起こし、保存されたレスポンスはリフレッシュされます。

コマンドラインインターフェイスを使用してリクエストの Cache-Control および Pragma ヘッダーを無視するには

コマンドプロンプトで入力します:

```
set cache contentgroup <name> -ignoreReqCachingHdrs YES
```

コマンドラインインターフェイスを使用してブラウザのリロード要求を無視するには

コマンドプロンプトで入力します:

```
set cache contentgroup <name> -ignoreReloadReq NO
```

> メモ:
> デフォルトでは、-ignoreReloadReq パラメータは YES に設定されています。

GUI を使用してリクエストの **Cache-Control** ヘッダーと **Pragma** ヘッダーを無視する

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. [その他] タブの [設定] グループで、[要求] オプションの [キャッシュコントロールとプラグマヘッダーを無視] を選択します。

Cache-Control ヘッダーを無視するポリシーの例:

次の例では、Content-type: image/* を含むレスポンスを、レスポンスの Cache-Control ヘッダーに関係なく、キャッシュするようにリクエスト時間オーバーライドポリシーを設定します。

image/* を含むすべてのレスポンスをキャッシュするようにリクエスト時間オーバーライドポリシーを設定するには

[すべて無効化] オプションを使用してキャッシュをフラッシュします。

新しいキャッシュポリシーを設定し、そのポリシーを特定のコンテンツグループに向けます。詳しい情報については、「統合キャッシュのポリシーの設定」を参照してください。

「リクエストでの Cache-Control ヘッダーと Pragma ヘッダーを無視する」の説明に従って、ポリシーが使用するコンテンツグループが Cache-Control ヘッダーを無視するように設定されていることを確認します。

ポリシーをリクエスト時間上書きポリシーバンクにバインドします。

詳細については、「統合キャッシュポリシーのグローバルバインド 」トピックを参照してください。

リクエストを受信するたびにオリジンサーバーをポーリングします

NetScaler アプライアンスは、保存された応答を処理する前に常に配信元サーバーを参照するように構成できます。これは、毎回ポーリング (PET) と呼ばれます。NetScaler アプライアンスが配信元サーバーを参照していて、PET 応答の有効期限が切れていない場合、配信元サーバーからの応答が完全になってもキャッシュされたコンテンツは上書きされません。このプロパティは、クライアント固有のコンテンツを提供する場合に便利です。

PET 応答の有効期限が切れると、NetScaler アプライアンスはオリジンサーバーから最初の完全応答が届いたときにそれを更新します。

毎回ポーリング (PET) 関数は次のように機能します。

Tag または Last-Modified ヘッダーの形式のバリデータを含むキャッシュされたレスポンスの場合、レスポンスの有効期限が切れると、自動的に PET とマークされ、キャッシュされます。

コンテンツグループに PET を設定できます。

コンテンツグループを PET として設定すると、コンテンツグループ内のすべての応答に PET とマークされます。PET コンテンツグループは、バリデータを持たないレスポンスを保存できます。自動的に PET とマークされた応答は、常に期限切れとなります。PET コンテンツグループに属する応答は、コンテンツグループの設定方法に基づいて、遅延後に期限切れになることがあります。

ポーリングは、次の 2 種類のリクエストに影響します。

- 条件付きリクエスト: クライアントは、応答が最新のコピーであることを確認するために、条件付きリクエストを発行します。キャッシュされた PET 応答に対するユーザーエージェントリクエストは、常に条件付きリクエストに変換され、オリジンサーバーに送信されます。条件付きリクエストの`If-Modified-Since`ヘッダーまたは`If-None-Match`ヘッダーにバリデータがあります。`If-Modified-Since`ヘッダーには`Last-Modified`ヘッダーの時刻が含まれます。If-None-Match ヘッダーには、レスポンスの Tag ヘッダー値が含まれます。クライアントのレスポンスのコピーが新鮮な場合、オリジンサーバーは 304 Not Modified と応答します。コピーが古くなっている場合、条件付き応答では 200 OK が生成され、応答全体が格納されます。
- 非条件付きリクエスト: 非条件リクエストでは、レスポンス全体を含む 200 OK しか生成できません。

オリジンサーバーのレスポンス	アクション
完全な回答を送信	オリジンサーバーはレスポンスをそのままクライアントに送信します。キャッシュされたレスポンスの有効期限が切れた場合、そのレスポンスはリフレッシュされます。
304 変更されていない	304 レスポンスのヘッダー値がキャッシュされたレスポンスとマージされ、キャッシュされたレスポンスがクライアントに提供されます。Date、Expires、Age、Cache-Control ヘッダー Max-Age、および S-Maxage トークン
401 無許可、400 不正な要求、405 メソッドは許可されない、406 は受け入れられない、407 プロキシ認証が必要	オリジンの応答は、そのままクライアントに提供されます。キャッシュされたレスポンスは変更されません。
その他のエラー応答 (404 Not Found など)	オリジンの応答は、そのままクライアントに提供されます。キャッシュされたレスポンスは削除されます。

> 注:
> Poll Every Time パラメータは、影響を受けるレスポンスを保存不可として扱います。

コマンドラインインターフェイスを使用してポーリングを毎回設定するには

コマンドプロンプトで入力します:

```
add cache contentgroup <contentGroupName> -pollEveryTime YES
```


GUI を使用してポーリングする

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. [その他] タブの [設定] グループで、[毎回ポーリング (リクエストごとにキャッシュされたコンテンツをオリジンで検証)] オプションを選択します。

PET およびクライアント固有のコンテンツ

PET 機能を使用すると、クライアント向けにコンテンツを確実にカスタマイズできます。たとえば、複数の言語でコンテンツを提供する Web サイトでは、Accept-Language リクエストヘッダーを調べて、配信するコンテンツの言語を選択します。英語が主言語である多言語の Web サイトでは、すべての英語コンテンツを PET コンテンツグループにキャッシュできます。これにより、すべてのリクエストがオリジンサーバーに送信され、レスポンスの言語が決定されます。応答が英語で、コンテンツが変更されていない場合、オリジンサーバーは 304 Not Modified をキャッシュに提供できます。

次に、英語による応答を PET コンテンツグループにキャッシュし、キャッシュ内の英語の応答を識別する名前付き式を設定し、このコンテンツグループと名前付き式を使用するポリシーを設定するコマンドの例を示します。太字は強調のために使われます。

```
1 add cache contentgroup EnglishLanguageGroup -pollEveryTime YES
2 add expression containsENExpression  - rule "http.res.header(\\"Content-
     Language\\").contains(\\"en\\")"
3 add cache policy englishPolicy -rule containsENExpression -action CACHE
      -storeInGroup englishLanguageGroup
4 bind cache policy englishPolicy -priority 100 -precedeDefRules NO
5 <!--NeedCopy-->
```

PET と認証、承認、監査

Outlook Web Access (OWA) は、PET の恩恵を受ける動的に生成されたコンテンツの好例です。すべてのメールレスポンス (*.EML オブジェクト) には、PET ETag レスポンスとして保存できるバリデータがあります。

メールレスポンスのリクエストはすべて、レスポンスがキャッシュされていても、オリジンサーバーに送信されます。オリジンサーバーは、リクエスタが認証され、承認されているかどうかを判断します。また、オリジンサーバーに応答が存在するかどうかも検証します。すべての結果が肯定的である場合、オリジンサーバーは 304 Not Modified レスポンスを送信します。

統合キャッシュをフォワードプロキシとして構成する

August 15, 2023

統合キャッシュは、リクエストを他の NetScaler アプライアンスまたは他のタイプのキャッシュサーバーに渡す転送プロキシデバイスとして機能します。統合キャッシュをフォワードプロキシとして設定するには、1 つまたは複数のキャッシュサーバーの IP アドレスを特定します。転送プロキシを構成すると、NetScaler アプライアンスは、統合キャッシュを使用せずに、構成済みの IP アドレスを含む要求をキャッシュサーバーに送信します。

コマンドラインインターフェイスを使用して NetScaler をフォワードキャッシュプロキシとして構成するには

コマンドプロンプトで入力します。

```
add cache forwardProxy <IPAddress> <port>
```

GUI を使用して NetScaler をフォワードキャッシュプロキシとして構成するには

1. [最適化] > [統合キャッシュ] > [転送プロキシ] に移動し、IP アドレスとポート番号を指定して転送プロキシを追加します。

統合キャッシュのデフォルト設定

August 15, 2023

NetScaler 統合キャッシュ機能には、デフォルトコンテンツグループのデフォルト設定と初期設定を含む組み込みポリシーが用意されています。このセクションの情報は、組み込みポリシーとデフォルトコンテンツグループのパラメータを定義します。

既定のキャッシュポリシー

統合キャッシュにはポリシーが組み込まれています。NetScaler アプライアンスは、次のセクションで説明するように、特定の順序でポリシーを評価します。

これらの組み込みポリシーを、要求時オーバーライドまたは応答時間オーバーライドポリシーバンクにバインドされたユーザー定義ポリシーでオーバーライドできます。

> 注:
> リリース 9.0 より前のポリシーを設定し、ポリシーをバインドするときに-precedeDefrules パラメータを指定した場合、それらのポリシーは移行時に自動的にオーバーライドタイムのバインドポイントに割り当てられます。

デフォルトポリシーを表示

組み込みポリシー名はアンダースコア (_) で始まります。組み込みポリシーは、show cache policy コマンドを使用してコマンドラインと管理コンソールから表示できます。

デフォルトのリクエストポリシー

新しいポリシーを設定してリクエスト時間オーバーライド処理ポイントにバインドすることで、以下の組み込みリクエスト時間ポリシーをオーバーライドできます。以下のポリシーでは、MAY_NOCACHE アクションは、応答時にユーザー設定または組み込みの CACHE ディレクティブがある場合にのみトランザクションをキャッシュするように規定していることに注意してください。

以下のポリシーは _ReqBuiltInDefaults ポリシーラベルにバインドされています。それらは優先度順にリストされています。

GET 以外のメソッドを使用するリクエストの応答をキャッシュしないでください。

ポリシー名は _NongEtreq です。次に、ポリシールールを示します。

```
!HTTP.REQ.METHOD.eq(GET)
```

If-Match または If-Unmodified-Since を含むヘッダー値を持つリクエストに NOCACHE アクションを設定します。

ポリシー名は _AdvancedConditionalReq です。次に、ポリシールールを示します。

```
HTTP.REQ.HEADER("If-Match").EXISTS || HTTP.REQ.HEADER("If-Unmodified-Since").EXISTS
```

クッキー、承認、プロキシ認証、または NTLM またはネゴシエートヘッダーを含むリクエストのヘッダー値を持つリクエストに MAY_NOCACHE アクションを設定します。

ポリシー名は _PersonalizedRq です。次に、ポリシールールを示します。

```
HTTP.REQ.HEADER("Cookie").EXISTS || HTTP.REQ.HEADER("Authorization").EXISTS || HTTP.REQ.HEADER("Proxy-Authorization").EXISTS || HTTP.REQ.IS_NTLM_OR_NEGOTIATE
```

デフォルト・レスポンス・ポリシー

新しいポリシーを設定して応答時間オーバーライド処理ポイントにバインドすることで、次のデフォルトの応答時間ポリシーをオーバーライドできます。

以下のポリシーは **_ResBuiltInDeFaults** ポリシーラベルにバインドされ、記載されている順序で評価されます。

1. HTTP 応答のタイプが 200、304、307、203 の場合、またはタイプが 400 から 499 または 300 から 302 の間でない限り、HTTP 応答をキャッシュしないでください。

   ポリシー名は **_uncacheableStatusRes** です。次に、ポリシールールを示します。

   ```
   !((HTTP.RES.STATUS.EQ(200))|| (HTTP.RES.STATUS.EQ(304))|| (HTTP.RES.STATUS.BETWEEN(400,499))|| (HTTP.RES.STATUS.BETWEEN(300, 302))|| (HTTP.RES.STATUS.EQ(307))|| (HTTP.RES.STATUS.EQ(203)))
   ```


2. HTTP レスポンスに Accept-Encoding 以外の値の Vary ヘッダーがある場合は、キャッシュしないでください。

   圧縮モジュールは Vary: Accept_Encoding ヘッダーを挿入します。このエクスプレッションの名前は **_uncacheableVaryRes** です。次に、ポリシールールを示します。

   ```
   ((HTTP.RES.HEADER("Vary").EXISTS)&& ((HTTP.RES.HEADER("Vary").
   INSTANCE(1).LENGTH > 0)|| (!HTTP.RES.HEADER("Vary").STRIP_END\\
   _WS.SET_TEXT_MODE(IGNORECASE).eq("Accept-Encoding"))))
   ```

3. Cache-Control ヘッダー値が No-Cache、No-Store、Private の場合、または Cache-Control ヘッダーが有効でない場合は、レスポンスをキャッシュしないでください。

   ポリシー名は **_uncacheableCacheControlRes** です。次に、ポリシールールを示します。

   ```
   ((HTTP.RES.CACHE\\_CONTROL.IS\\_PRIVATE)|| (HTTP.RES.CACHE\\
   _CONTROL.IS\\_NO\\_CACHE)|| (HTTP.RES.CACHE\\_CONTROL.IS\\_NO
   \\_STORE)|| (HTTP.RES.CACHE\\_CONTROL.IS\\_INVALID))
   ```

4. Cache-Control ヘッダーに Public、Must-Revalidate、Proxy Revalidate、Max-Age、S-Maxage のいずれかの値が含まれている場合、応答をキャッシュします。

   ポリシー名は **_CacheableCacheControllRes** です。次に、ポリシールールを示します。

   ```
   ((HTTP.RES.CACHE_CONTROL.IS_PUBLIC)|| (HTTP.RES.CACHE_CONTROL.
   IS_MAX_AGE)|| (HTTP.RES.CACHE_CONTROL.IS_MUST_REVALIDATE)|| (HTTP
   .RES.CACHE_CONTROL.IS_PROXY_REVALIDATE)|| (HTTP.RES.CACHE_CONTROL
   .IS_S_MAXAGE))
   ```

5. Pragma ヘッダーを含む応答はキャッシュしないでください。

   ポリシーの名前は **_UncacheablePragmares** です。次に、ポリシールールを示します。

   ```
   HTTP.RES.HEADER("Pragma").EXISTS
   ```

6. Expires ヘッダーを含むレスポンスをキャッシュします。

   ポリシーの名前は **_cacheableExpiryRes** です。次に、ポリシールールを示します。

   ```
   HTTP.RES.HEADER("Expires").EXISTS
   ```

7. レスポンスに Image という値の Content-Type ヘッダーが含まれている場合は、ヘッダー内の Cookie をすべて削除してキャッシュします。

   ポリシーの名前は **_ImageRes** です。次に、ポリシールールを示します。

   ```
   HTTP.RES.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).STARTSWITH
   ("image/")
   ```

   このポリシーと連携するように次のコンテンツグループを設定できます。

   ```
   add cache contentgroup nocookie -group -removeCookies YES
   ```


8. Set-Cookie ヘッダーを含む応答をキャッシュしないでください。

   ポリシーの名前は **_ PersonalizedRes** です。次に、ポリシールールを示します。

```
HTTP.RES.HEADER(“Set-Cookie”).EXISTS
```

デフォルトポリシーの制限

以下の組み込みリクエスト時間ポリシーをユーザー定義ポリシーで上書きすることはできません。

これらのポリシーは、優先度順にリストされています。

1. 対応する HTTP リクエストに GET または POST メソッドがない場合は、応答をキャッシュしないでください。
2. HTTP リクエスト URL の長さとホスト名の合計が 1744 バイトを超える場合は、リクエストに対する応答をキャッシュしないでください。
3. If-Match ヘッダーを含むリクエストの応答をキャッシュしないでください。
4. If-Unmodified-Since ヘッダーを含むリクエストをキャッシュしないでください。

> 注:
> これは If-Modified-Since ヘッダーとは異なります。

1. サーバーが有効期限ヘッダーを設定していない場合は、応答をキャッシュしないでください。

以下の組み込み応答時間ポリシーを上書きすることはできません。これらのポリシーは、記載されている順に評価されます。

1. HTTP 応答ステータスコードが 201、202、204、205、206 の応答をキャッシュしないでください。
2. ステータスコード 403、404、410 を除き、HTTP 応答ステータスコードが 4xx の応答はキャッシュしないでください。
3. レスポンスタイプが FIN 終了の場合、またはレスポンスに Content-Length または Transfer-Encoding: Chunked のいずれかの属性がない場合は、レスポンスをキャッシュしないでください。
4. キャッシュモジュールが Cache-Control ヘッダーを解析できない場合は、応答をキャッシュしないでください。

既定のコンテンツグループの初期設定

統合キャッシュを初めて有効にすると、NetScaler ADC アプライアンスは、デフォルトのコンテンツグループという名前の定義済みコンテンツグループを提供します。詳細については、「既定のコンテンツグループ設定テーブル」を参照してください。

トラブルシューティング

August 15, 2023

統合キャッシュ機能を構成しても期待どおりに機能しない場合は、いくつかの一般的なツールを使用して NetScaler リソースにアクセスし、問題を診断できます。

トラブルシューティングのリソース

トラブルシューティングに使用できるリソースと設定例の詳細については、「PDF ファイルのトラブルシューティング用リソース」を参照してください。

フロントエンドの最適化

April 15, 2024

> 注：フロントエンドの最適化は、高度なライセンスまたはプレミアム NetScaler ADC ライセンスがあり、NetScaler ADC リリース 12.1 以降を実行している場合に利用できます。

Web アプリケーションの基礎となる HTTP プロトコルは、元々、単純な Web ページの送信とレンダリングをサポートするために開発されました。JavaScript やカスケードスタイルシート（CSS）などの新しいテクノロジーや、Flash ビデオやグラフィックが豊富な画像などの新しいメディアタイプでは、フロントエンドのパフォーマンス、つまりブラウザーレベルでのパフォーマンスに大きな要求が課せられます。

NetScaler のフロントエンド最適化（FEO）機能はこのような問題を解決し、次の方法で Web ページの読み込み時間とレンダリング時間を短縮します：

- リクエスト数を減らす。
- 各ページのレンダリングに必要です。
- ページ応答のバイト数を減らします。

クライアントブラウザに提供されるコンテンツを簡素化および最適化します。

FEO 構成をカスタマイズして、ユーザーに最良の結果を提供できます。NetScaler は、デスクトップユーザーとモバイルユーザーの両方を対象に、さまざまな Web コンテンツの最適化をサポートしています。次の表では、FEO 機能によって提供されるフロントエンドの最適化と、さまざまなタイプのファイルに対して実行される操作について説明します。

FEO 機能によって実行される最適化

ウェブ最適化	問題	NetScaler FEO 機能は何をしますか	メリット
インライン化	クライアントブラウザは、多くの場合、Web ページに関連付けられた外部 CSS、画像、および JavaScript をロードするためにサーバーに複数の要求を送信します。	CSS インライン、JavaScript インライン、CSS の組み合わせ	外部 CSS、画像、および JavaScript を HTML ファイルにインラインでロードすると、ページレンダリング時間が短縮されます。この最適化は、一度しか表示されないコンテンツや、キャッシュサイズが限られているモバイルデバイスにとって有益です。
ミニフィケーション	サーバーから取得されたデータには、空白、コメント、改行文字などの不要な文字が含まれます。ブラウザがこのようなデータの処理に費やす時間により、ウェブサイトに遅延が生じます。	CSS ミニファイファイズ、ジャバスクリプトミニフィケーション、HTML コメントの削除	圧縮されたファイルは帯域幅の消費量が少なく、特別な処理による遅延を回避できます。
画像最適化	モバイルブラウザは、接続速度が遅く、キャッシュメモリが限られていることがよくあります。モバイルクライアントに画像をダウンロードすると、より多くの帯域幅、処理時間、およびキャッシュスペースが消費され、Web サイトの待ち時間が発生します。	JPEG 最適化, CSS 画像のインライン化, 画像の縮小属性, GIF から PNG への変換, HTML 画像のインライン化, WebP 画像変換, JPEG, GIF, PNG から JPEG-XR 画像変換	画像を NetScaler ADC によって画像タグに示されているサイズに縮小し、クライアントブラウザが画像をより高速にロードできるようにします。
再配置	外部 CSS、画像、および JavaScript の処理が効率的でないと、ページの読み込み時間が長くなります。	画像の遅延読み込み、CSS のヘッドへの移動、JavaScript の終了への移動	HTML 要素を再配置して、Web ページのレンダリング時間を短縮し、クライアントブラウザがオブジェクトをより高速にロードできるようにします。

ウェブ最適化	問題	NetScaler FEO 機能は何をしますか	メリット
接続管理	多くのブラウザは、1 つのドメインに対して確立できる同時接続の数に制限を設けています。これにより、ブラウザーが Web ページのリソースを 1 つずつダウンロードし、ブラウザーの時間が長くなる可能性があります。	ドメインシャーディング	接続制限を克服し、クライアントブラウザがより多くのリソースを並行してダウンロードできるようにすることで、ページのレンダリング時間を短縮します。

さまざまなファイルタイプでのウェブ最適化**:**

NetScaler ADC は、CSS、画像、Javascript、および HTML 上で Web 最適化を実行できます。詳細については、Web 最適化 PDF を参照してください。

> 注:
> フロントエンド最適化機能は ASCII 文字のみをサポートします。Unicode 文字セットはサポートしていません。

フロントエンド最適化の仕組み

NetScaler がサーバーからの応答を受信したら、

1. ページの内容を解析し、キャッシュにエントリを作成し (該当する場合)、FEO ポリシーを適用します。

   たとえば、NetScaler は次の最適化ルールを適用できます。

   - CSS または JavaScript 内に存在する空白やコメントを削除します。
   - 1 つ以上の CSS ファイルを 1 つのファイルに結合します。
   - GIF 画像形式を PNG 形式に変換します。

2. 埋め込まれたオブジェクトを書き換え、最適化されたコンテンツを、最初のキャッシュエントリに使用されたものとは異なる署名でキャッシュに保存します。
3. 後続のリクエストでは、最適化されたオブジェクトをサーバーからではなくキャッシュから取得し、その応答をクライアントに転送します。

**

空白やコメントなどの無関係な情報を削除します。

ブラウザがサーバー上に新しいコンテンツがあるかどうかを確認せずに、キャッシュされたリソースを使用できる期間。

フロントエンド最適化の設定

オプションで、フロントエンド最適化のグローバル設定の値を変更できます。それ以外の場合は、埋め込みオブジェクトに適用する最適化ルールを指定するアクションを作成することから始めてください。

アクションを設定したら、レスポンスを最適化するリクエストのタイプを指定するルールを含むポリシーを作成し、アクションをポリシーに関連付けます。

> 注: NetScaler ADC は、リクエスト時にのみフロントエンド最適化ポリシーを評価し、応答時間ではなく評価します。

ポリシーを有効にするには、ポリシーをバインドポイントにバインドします。ポリシーをグローバルにバインドして NetScaler を通過するすべてのトラフィックに適用することも、HTTP または SSL タイプの負荷分散またはコンテンツスイッチング仮想サーバーにポリシーをバインドすることもできます。ポリシーをバインドするときは、そのポリシーに優先度を割り当てます。優先度の数値が小さいほど、値が高いことを示します。NetScaler はポリシーを優先順位に従って適用します。

NetScaler でフロントエンド最適化を構成する方法について詳しくは、次のビデオを参照してください:

これは埋め込みビデオです。リンクをクリックしてビデオを見る

前提条件

フロントエンドを最適化するには、NetScaler 統合キャッシュ機能を有効にする必要があります。また、次の統合キャッシュ構成を実行する必要があります。

- キャッシュメモリを割り当てます。
- デフォルトのキャッシュコンテンツグループの最大応答サイズとメモリ制限を設定します。

統合キャッシュの構成の詳細については、統合キャッシュを参照してください。

注: 統合キャッシュという用語は、AppCache と同じ意味で使用できます。機能の観点から、両方の用語は同じ意味であることに注意してください。

NetScaler コマンドインターフェイスを使用してフロントエンド最適化を構成する

コマンドプロンプトで、次の操作を行います:

1. フロントエンド最適化機能を有効にします。

```
enable ns feature FEO
```

1. 1 つ以上のフロントエンド最適化アクションを作成します。


```
add feo action <name> [-imgShrinkToAttrib] [-imgGifToPng] ...
```

例: フロントエンド最適化アクションを追加して、GIF 形式の画像を PNG 形式に変換し、キャッシュの有効期限を延長するには:

```
add feo action allact -imgGifToPng -pageExtendCache
```

1. [オプション] フロントエンド最適化のグローバル設定にデフォルト以外の値を指定します。

```
set feo parameter [-cacheMaxage <integer>] [-JpegQualityPercent
<integer>] [-cssInlineThresSize <integer>] [-inlineJsThresSize <
integer> [-inlineImgThresSize <integer>]
```

例: キャッシュの最大有効期限を指定するには:

```
set feo parameter -cacheMaxage 10
```

1. 1 つ以上のフロントエンド最適化ポリシーを作成します。

```
add feo policy <name> <rule> <action>
```

例: フロントエンド最適化ポリシーを追加し、上記で指定した allact アクションに関連付けるには、次の手順を実行します。

```
1 >add feo policy pol1 TRUE all act
2 >add feo policy pol1 "(HTTP.REQ.URL.CONTAINS("testsite"))" allact1
3 <!--NeedCopy-->
```

1. ポリシーを負荷分散またはコンテンツスイッチング仮想サーバーにバインドするか、グローバルにバインドします。

```
bind lb vserver <name> -policyName <string> -priority <num>

bind cs vserver <name> -policyName <string> -priority <num>

bind feo global <policyName> <priority> -type <type> <gotoPriorityExpression
>
```

例:「abc」という名前の仮想サーバーにフロントエンド最適化ポリシーを適用するには:

```
> bind lb vserver abc -policyName pol1 -priority 1 -type NONE
```

例:ADC に到達するすべてのトラフィックにフロントエンド最適化ポリシーを適用するには:

```
> bind feo global pol1 100 -type REQ_DEFAULT
```

1. 構成を保存します。ns 構成を保存

GUI を使用したフロントエンドの最適化の構成

1. [最適化] > [フロントエンド最適化] > [アクション] に移動し、[追加] をクリックし、関連する詳細を指定してフロントエンド最適化アクションを作成します。

2. [オプション] フロントエンド最適化のグローバル設定を指定します。
3. [最適化] > [フロントエンド最適化] に移動し、右側のペインの [設定] で [フロントエンド最適化設定の変更] をクリックして、フロントエンド最適化のグローバル設定を指定します。
4. フロントエンド最適化ポリシーを作成します。
5. [最適化] > [フロントエンド最適化] > [ポリシー] に移動し、[追加] をクリックして、関連する詳細を指定してフロントエンド最適化ポリシーを作成します。
6. 負荷分散またはコンテンツスイッチング仮想サーバーにポリシーをバインドします。

    a) [最適化] > [フロントエンド最適化] > [ポリシー] に移動します。
    b) フロントエンド最適化ポリシーを選択し、[**Policy Manager**] をクリックします。
    c) フロントエンド最適化ポリシーマネージャーで、フロントエンド最適化ポリシーを負荷分散またはコンテンツスイッチング仮想サーバーにバインドします。

フロントエンド最適化構成の確認

ダッシュボードユーティリティは、要約と詳細な統計を表形式とグラフィック形式で表示します。FEO 統計を表示して FEO 構成を評価できます。

オプションで、ポリシーベースの FEO 中にポリシーカウンタがインクリメントするセレクトの数など、FEO ポリシーの統計を表示することもできます。

> 注:
> 統計とグラフの詳細については、NetScaler アプライアンスのダッシュボードヘルプを参照してください。

CLI を使用して **FEO** 統計を表示する

コマンドプロンプトで次のコマンドを入力して、FEO 統計の概要、FEO ポリシーの選択と詳細、および詳細な FEO 統計をそれぞれ表示します。

- `stat feo` 注:stat feo policy コマンドは、高度な FEO ポリシーの統計情報のみを表示します。
- `show feo policy name`
- `stat feo -detail`

NetScaler ダッシュボードで **FEO** の統計情報を表示する

ダッシュボード GUI では、次のことができます。

- 「フロントエンドの最適化」を選択すると、FEO 統計の概要が表示されます。
- 「グラフィカルビュー」タブをクリックすると、FEO 機能によって処理されたリクエストの割合が表示されます。

サンプルの最適化:

HTML コンテンツおよび HTML コンテンツ内の埋め込みオブジェクトに適用されるコンテンツ最適化アクションの例については、サンプル PDF を参照してください。

メディア分類

August 15, 2023

ネットワーク内のトラフィックの種類を理解することで、ネットワーク管理者は帯域幅消費量を管理して最適なネットワークパフォーマンスを実現できます。メディア分類モードは、NetScaler アプライアンスを経由するメディアトラフィックの統計を監視および表示します。

このモードを有効にすると、ネットワーク管理者はアクセスされたデータ量と、メディアファイルにアクセスしたデバイスの種類を示す統計情報を収集できます。NetScaler アプライアンスは、このモードでのバイト範囲リクエストもサポートします。

現在、NetScaler アプライアンスは以下のメディアファイルタイプの統計を監視および表示できます。

メディア	ファイルタイプ
Microsoft スムーズストリーミング	ビデオ
Apple ライブストリーミング	ビデオ
オーディオデータ転送ストリーム (ADTS)	オーディオ
アドバンスト・オーディオ・コーディング (AAC)	オーディオ
フラッシュビデオ (FLV)	オーディオとビデオ
3GP	オーディオとビデオ

アプライアンスは次のデバイスの統計情報を表示できます。

デバイスプラットフォーム	デバイスの種類
iOS	iPad と iPod
Android	モバイルとタブレット
ラップトップまたはデスクトップ	Windows ラップトップおよびデスクトップコンピューター
その他のトピック	その他のモバイルデバイス (モバイルおよびタブレット)

ネットワーク管理者は、次の統計カウンターを確認して、さまざまなメディアトラフィックタイプについて NetScaler アプライアンスを介してアクセスされたデータ量を把握できます。

メディアファイル名	統計カウンター
Microsoft スムーズストリーミング	mcmssmthstrmvid—このカウンターは、NetScaler アプライアンスによって提供される Microsoft スムーズストリーミングビデオの総数を記録します。Mcmssmthstrvidpl—このカウンターは、NetScaler アプライアンスによって提供される Microsoft スムーズストリーミングのビデオプレイリストの合計数を記録します。Mcmssmthstrmvidbytes—このカウンターは、NetScaler アプライアンス上で Microsoft スムーズストリーミングメディアトラフィックに提供されたデータバイトの総数を記録します。Mcmssmthstrmplvidbytespl—このカウンターは、NetScaler アプライアンスによって提供された Microsoft スムーズストリーミングプレイリストのバイト数の合計を記録します。
Apple ライブストリーミング	mccaplelivestrmngvid—このカウンターは、NetScaler アプライアンスによって配信された Apple Live ストリーミングビデオの総数を記録します。Mccapplelivestrmngvidpl—このカウンターは、NetScaler アプライアンスによって提供される Apple ライブストリーミングビデオプレイリストの総数を記録します。Mcapplelivestreamingvidbytes—このカウンターは、NetScaler アプライアンス上で Apple Live Streaming メディアトラフィックに対して処理されたデータバイトの総数を記録します。Mcapplelivestreamingplaylistvidbytespl—このカウンタは、NetScaler アプライアンスによって処理された Apple Live プレイリストのバイトの総数を記録します。

メディアファイル名	統計カウンター
オーディオデータ転送ストリーム (ADTS)	mcadtsaudio—このカウンターは、NetScaler アプライアンスによって提供される ADTS オーディオクリップの総数を記録します。Mcadtsaudiobytes—このカウンターは、NetScaler アプライアンス上で ADTS メディアトラフィックに対して処理されたデータバイトの総数を記録します。
アドバンスト・オーディオ・コーディング (AAC)	Mcaacaudio—このカウンターは、NetScaler アプライアンスによって提供される AAC オーディオクリップの総数を記録します。Mcaacaudiobytes—このカウンターは、NetScaler アプライアンスで AAC メディアトラフィックに対して処理されたデータバイトの総数を記録します。
フラッシュビデオ (FLV)	Mcflvvid—このカウンターは、NetScaler アプライアンスによって配信されたフラッシュビデオの総数を記録します。Mcflvvidbytes—このカウンターは、NetScaler アプライアンスでフラッシュビデオに提供されたデータバイトの総数を記録します。
3GP	mc3gpvidbytes—このカウンターは、NetScaler アプライアンスで 3GP メディアトラフィックに対して処理されたデータバイトの総数を記録します。

NetScaler アプライアンスは、レスポンスの最初の本文バイトに含まれる署名によってメディアファイルの種類を検出します。たとえば、mp4 ファイルの最初の本文バイトのレスポンスには次のシグネチャがあります。

```
**....ftypmp42** ....isommp42....moov...lmvhd.....c.\!.c.\!..
```

NetScaler アプライアンスは、クライアントデバイスが HTTP *GET* リクエストに含めるユーザーエージェント文字列によってクライアントデバイスタイプを検出します。たとえば、UC ブラウザを使用するウィンドウフォンでは、HTTP GET リクエストに次のユーザエージェント文字列が含まれます。

```
User-Agent: **UCWEB**/2.0 (**Windows**; U; wds 8.10; en-US; HTC; 8X
by HTC)U2/1.0.0
```

メディア分類を有効にする

デフォルトでは、NetScaler アプライアンスのメディア分類は無効になっています。使用する前にモードを有効にする必要があります。

コマンドラインインターフェイスを使用してメディア分類を有効にするには

コマンドプロンプトで入力します。

```
enable ns mode Mediaclassification
```

GUI を使用してメディア分類を有効にするには

NetScaler アプライアンスでメディア分類を有効にする

[システム] > [設定] > [モードの設定] に移動し、[メディア分類] を選択します。

NetScaler アプライアンスのメディアトラフィック統計を表示するには

[最適化] に移動し、[メディア分類] をクリックすると、メディアトラフィックの統計が表示されます。

メディア分類統計の検証

メディアトラフィックの統計は、ダッシュボードユーティリティまたはコマンドラインインターフェイスを使用して表示できます。ダッシュボードユーティリティは、サマリーおよび詳細な統計を表形式およびグラフィック形式で表示します。

> 注:
> 統計とグラフの詳細については、NetScaler アプライアンスのダッシュボードヘルプを参照してください。

コマンドラインインターフェイスを使用してメディア分類統計を表示するには

コマンドプロンプトで、次のいずれかのコマンドを入力して、メディア分類統計の概要を表示したり、詳細な統計情報を表示したり、表示をクリアしたりします。

```
stat Mediaclassification

stat Mediaclassification -detail

stat Mediaclassification -clearstats
```

メディア分類統計をダッシュボードに表示するには

ダッシュボードユーティリティでは、次の種類のメディア分類統計を表示できます。

1. メディア分類を選択すると、メディアトラフィック統計の概要が表示されます。
2. メディアトラフィックの詳細な統計情報を表示するには、「詳細」をクリックします。
3. メディアトラフィックの統計情報を消去するには、「クリア」をクリックします。

レピュテーション

August 15, 2023

NetScaler はレピュテーションベースのセキュリティを提供します。レピュテーション評価を使用してリクエストの処理に関するリスクを判断すると、特定のリクエストをブロックまたはドロップするなどのアクションを実行して、アプリケーションのパフォーマンスを向上させることができます。

NetScaler IP レピュテーション機能は、IP レピュテーションチェックを使用して、ゼロデイ攻撃を防ぎ、Web 攻撃、フィッシング活動、または Web スキャンに関連する悪意のあるソースからの保護を提供します。

詳しくは、「IP レピュテーション」を参照してください。

IP レピュテーション

December 8, 2023

IP レピュテーションは、不要な要求を送信する IP アドレスを識別するツールです。IP レピュテーションリストを使用すると、レピュテーションの悪い IP アドレスからのリクエストを拒否できます。処理しない要求をフィルタリングして、Web アプリケーションファイアウォールのパフォーマンスを最適化します。リセット、要求のドロップ、またはレスポンダーポリシーを構成して、特定のレスポンダーアクションを実行します。

IP レピュテーションを使用して防止できる攻撃には、次のようなものがあります:

- ウイルスに感染したパーソナルコンピュータ。(自宅の PC) は、インターネット上のスパムの唯一の最大のソースです。IP レピュテーションは、不要な要求を送信している IP アドレスを識別できます。IP レピュテーションは、既知の感染元からの大規模な DDoS 攻撃、DoS 攻撃、または異常な SYN フラッド攻撃をブロックする場合に特に役立ちます。
- 一元管理および自動化されたボットネット。数百台のコンピューターが連携してパスワードを解読するのに時間がかからないため、攻撃者はパスワードを盗むことで人気を博しています。ボットネット攻撃を開始して、よく使われる辞書の単語を使用するパスワードを見つけ出すのは簡単です。
- 侵害された **Web** サーバー。攻撃はそれほど一般的ではありません。なぜなら、意識とサーバーのセキュリティが高まっているため、ハッカーやスパマーはより簡単な標的を探します。ハッカーが妥協してスパム (ウイルスやポルノなど) を送信するために使用できる Web サーバーやオンラインフォームはまだあります。このようなアクティビティは、SpamRats などのレピュテーションリストを使用して検出してすばやくシャットダウンしたり、ブロックしたりするのが簡単です。
- **Windows** エクスプロイト。(マルウェア、シェルコード、ルートキット、ワーム、またはウイルスを提供または配布するアクティブ IP など)。
- 既知のスパマーとハッカー。
- マスメールマーケティングキャンペーン。
- フィッシングプロキシ (フィッシングサイトをホストしている IP アドレス、および広告クリック詐欺やゲーム詐欺などのその他の詐欺)。
- 匿名プロキシ **(**プロキシおよび匿名化サービスを提供する **IP** (オニオンルーター別名 TOR を含む)。

NetScaler アプライアンスは、動的に生成される悪意のある **IP** データベースとそれらの **IP** アドレスのメタデータのサービスプロバイダーとして **Webroot** を使用します。メタデータには、位置情報の詳細、脅威カテゴリ、脅威数

などが含まれます。Webroot 脅威インテリジェンスエンジンは、数百万のセンサーからリアルタイムデータを受信します。高度な機械学習と行動分析を使用して、データを自動的かつ継続的にキャプチャ、スキャン、分析、スコアリングします。脅威に関するインテリジェンスは継続的に更新されます。

NetScaler アプライアンスは、Webroot の usesIP レピュテーションデータベースを使用して、受信したリクエストの評判が悪いかどうかを検証します。データベースには、IP アドレス分類に基づく IP 脅威カテゴリの膨大なコレクションがあります。次に、IP 脅威のカテゴリとその説明を示します。

- スパムソース。スパム送信元には、プロキシを介したスパムメッセージのトンネリング、異常な SMTP アクティビティ、フォーラムスパムアクティビティが含まれます。
- Windows エクスプロイト。Windows Exploits カテゴリには、マルウェア、シェルコード、ルートキット、ワーム、またはウイルスを提供または配布するアクティブな IP アドレスが含まれます
- ウェブ攻撃。Web 攻撃カテゴリには、クロスサイトスクリプティング、iFrame インジェクション、SQL インジェクション、クロスドメインインジェクション、またはドメインパスワードブルートフォース攻撃が含まれます
- ボットネット。ボットネットカテゴリには、ボットネット、C&C チャネル、ボットマスターが制御する感染したゾンビマシンが含まれます
- スキャナー。スキャナーカテゴリには、プローブ、ホストスキャン、ドメインスキャン、パスワードブルートフォース攻撃など、すべての偵察が含まれます
- サービス拒否。サービス拒否カテゴリには、DOS、DDOS、異常同期フラッド、異常トラフィック検出が含まれます
- 評判。マルウェアに感染していることが現在わかっている IP アドレスからのアクセスを拒否します。このカテゴリには、Webroot レピュテーションインデックススコアが平均的に低い IP も含まれます。このカテゴリを有効にすると、マルウェアの配布ポイントに接触していると特定されたソースからのアクセスが防止されます。
- フィッシング。フィッシングカテゴリには、フィッシングサイトをホストしている IP アドレス、広告クリック詐欺、ゲーム詐欺などのその他の種類の詐欺行為が含まれます
- プロキシ。プロキシカテゴリには、プロキシサービスとデフサービスを提供する IP アドレスが含まれます。
- モバイルの脅威。モバイル脅威カテゴリには、悪意のあるモバイルアプリケーションや望ましくないモバイルアプリケーションの IP アドレスが含まれます。このカテゴリは、Webroot モバイル脅威調査チームのデータを活用します。
- Tor プロキシ。Tor Proxy カテゴリには、Tor ネットワークの出口ノードとして動作する IP アドレスが含まれます。終了ノードはプロキシチェーンに沿った最後のポイントで、オリジネータの意図したデスティネーションに直接コネクトします。

ネットワーク内の任意の場所で脅威が検出されると、IP アドレスに悪意のあるフラグが付けられ、ネットワークに接続されているすべてのアプライアンスが即座に保護されます。IP アドレスの動的な変更は、高度な機械学習を使用して高速かつ正確に処理されます。

Webroot のデータシートに記載されているように、Webroot のセンサーネットワークは、スパムソース、Windows エクスプロイト、ボットネット、スキャナーなど、多くの主要な IP 脅威の種類を特定しています。(データシートのフロー図を参照してください。)

NetScaler アプライアンスは、`iprep`クライアントプロセスを使用して Webroot からデータベースを取得します。

`iprep`クライアントは HTTP GET メソッドを使用して、Webroot から絶対 IP リストを初めて取得します。その後、デルタの変更を 5 分ごとに 1 回チェックします。

> 重要:
>
> - IP レピュテーション機能を使用する前に、NetScaler アプライアンスがインターネットにアクセスでき、DNS が設定されていることを確認してください。
> - ウェブルートデータベースにアクセスするには、Citrix ****ADC** アプライアンスがポート **443** で **api.bcti.brightcloud.com** に接続できる必要があります **。HA またはクラスタ展開の各ノードは、Webroot からデータベースを取得し、この完全修飾ドメイン名（FQDN）にアクセスできる必要があります。
> - Webroot は現在 AWS でレピュテーションデータベースをホストしています。そのため、NetScaler はレピュテーションデータベースをダウンロードする AWS ドメインを解決できなければなりません。また、ファイアウォールは AWS ドメインに対して開いている必要があります。
>
> 注:
>
> IP レピュテーション機能が有効の場合、各パケットエンジンが正しく機能するには、少なくとも 4 GB が必要です。

高度なポリシー式。Web アプリケーションファイアウォールやレスポンダなど、サポートされているモジュールにバインドされたポリシーで高度なポリシー式（高度なポリシー式）を使用して、IP レピュテーション機能を設定します。クライアントの IP アドレスが悪意のあるものかどうかを検出するために使用できる式を示す 2 つの例を次に示します。

1. **CLIENT.IP.SRC.IPREP_IS_MALICIALICE**: この式は、クライアントが悪意のある IP リストに含まれている場合に TRUE と評価されます。
2. **CLIENT.IP.SRC.IPREP_THREAT_CATEGORY (**カテゴリ**)**: この式は、クライアント IP が悪意のある IP であり、指定された脅威カテゴリにある場合に TRUE と評価されます。
3. **CLIENT.IPV6.SRC.IPREP_IS_MALICIOUS** および **CLIENT.IPV6.SRC.IPREP_THREAT_CATEGORY**: クライアント IP のタイプが IPv6 で、指定された脅威カテゴリの悪意のある IP アドレスである場合、この式は TRUE と評価されます。

脅威カテゴリに指定できる値は次のとおりです。

SPAM_SOURCES, WINDOWS_EXPLOITS, WEB_ATTACKS, ボットネット, スキャナ, DOS, レピュテーション, フィッシング, プロキシ, ネットワーク, CLOUD_PROVIDERS, MOBILE_ 脅威, TOR_PROXY。

> 注:
>
> IP レピュテーション機能は、送信元と宛先 IP アドレスの両方をチェックします。ヘッダー内の悪意のある IP を検出します。ポリシー内の PI 式が IP アドレスを識別できる場合、IP レピュテーションチェックはそれが悪意があるかどうかを判断します。

iPrep ログメッセージ。/var/log/iprep.logファイルには、Webroot データベースとの通信に関する情報をキャプチャする便利なメッセージが含まれています。この情報には、Webroot 通信中に使用される資格情報、Webroot との接続の失敗、更新に含まれる情報（データベース内の IP アドレス数など）が含まれます。

ポリシーデータセットを使用して **IP** のブロックリストまたは許可リストを作成する。許可リストを維持して、Webroot データベースでブロックリストに登録されている特定の IP アドレスへのアクセスを許可できます。また、Webroot レピュテーションチェックを補完するために、IP アドレスのカスタマイズされたブロックリストを作成することもできます。これらのリストは、ポリシーデータセットを使用して作成できます。データセットは、IPv4 または IPv6 アドレスのマッチングに最適な特殊なパターンセットです。データセットを使用するには、まずデータセットを作成し、IPv4 または IPv6 アドレスをそのデータセットにバインドします。パケット内の文字列を比較するポリシーを設定する場合は、適切な演算子を使用し、パターンセットまたはデータセットの名前を引数として渡します。

IP レピュテーション評価中に例外として処理するアドレスの許可リストを作成するには、次の手順を実行します。

- 許可リストのアドレスが Webroot (または任意のサービスプロバイダー) によって悪意のあるアドレスとしてリストされている場合でも、PI 式が False と評価されるようにポリシーを構成します。

IP レピュテーションの有効化または無効化。IP レピュテーションは、ライセンスベースの一般レピュテーション機能の一部です。レピュテーション機能を有効または無効にすると、IP レピュテーションが有効または無効になります。

一般的な手順。IP レピュテーションの展開には、次のタスクが含まれます。

- NetScaler アプライアンスにインストールされているライセンスに IP レピュテーションがサポートされていることを確認します。プレミアムおよびスタンドアロンのアプリケーションファイアウォールライセンスは、IP レピュテーション機能をサポートします。
- IP レピュテーションおよびアプリケーションファイアウォール機能を有効にします。
- アプリケーションファイアウォールプロファイルを追加します。
- PI 式を使用して、IP レピュテーションデータベース内の悪意のある IP アドレスを識別するアプリケーションファイアウォールポリシーを追加します。
- アプリケーションファイアウォールポリシーを適切なバインドポイントにバインドします。
- 悪意のあるアドレスから受信した要求がns.logファイルに記録され、要求がプロファイルの指定どおりに処理されたことを示します。

CLI を使用して **IP** レピュテーション機能を設定する

コマンドプロンプトで入力します：

- `enable feature reputation`
- `disable feature reputation`

次の例は、PI 式を使用して悪意のあるアドレスを識別するアプリケーションファイアウォールポリシーを追加する方法を示しています。組み込みプロファイルを使用するか、プロファイルを追加するか、既存のプロファイルを設定して、リクエストがポリシー一致と一致したときに目的のアクションを呼び出すことができます。

例 3 と 4 は、IP アドレスのブロックリストまたは許可リストを生成するポリシーデータセットを作成する方法を示しています。

例 **1**:

次のコマンドは、悪意のある IP アドレスを識別し、一致がトリガーされた場合に要求をブロックするポリシーを作成します。

```
add appfw policy pol1 CLIENT.IP.SRC.IPREP_IS_MALICIOUS APPFW_BLOCK
add appfw policy pol1 CLIENT.IPv6.SRC.IPREP_IS_MALICIOUS APPFW_BLOCK
add appfw policy pol1 "HTTP.REQ.HEADER(\"X-Forwarded-For\").TYPECAST_IPv6_ADDR
.IPREP_IS_MALICIOUS"APPFW_RESET
```

例 **2**:

次のコマンドは、レピュテーションサービスを使用してX-Forwarded-Forヘッダー内のクライアント IP アドレスをチェックし、一致がトリガーされた場合は接続をリセットするポリシーを作成します。

```
> add appfw policy pol1 "HTTP.REQ.HEADER(\"X-Forwarded-For\").TYPECAST_IP_ADDR
.IPREP_IS_MALICIOUS"APPFW_RESET**
```

例 **3:**

次に、リストを追加して、指定した IP アドレスを許可する例外を追加する例を示します。

```
> add policy dataset Allow_list1 ipv4
> bind policy dataset Allow_list1 10.217.25.17 -index 1
> bind policy dataset Allow_list1 10.217.25.18 -index 2
```

次に、指定した IPv6 アドレスを許可する例外を追加するリストを追加する例を示します。

```
add policy dataset Allow_list_ipv6 ipv6
bind policy dataset Allow_list_ipv6 fe80::98c7:d8ff:fe3a:b562 -index 1
bind policy dataset Allow_list_ipv6 fe80::98c7:d8ff:fe3a:b563  -index 2

<!--NeedCopy-->
```

例 **4:**

次に、カスタマイズリストを追加して、指定した IP アドレスに悪意のあるフラグを付ける例を示します。

```
> add policy dataset Block_list1 ipv4
> bind policy dataset Block_list1 10.217.31.48 -index 1
> bind policy dataset Block_list1 10.217.25.19 -index 2
```

次に、カスタマイズしたリストを追加して、指定した IPv6 アドレスに悪意のあるものとしてフラグを付ける例を示します。


```
add policy dataset Block_list_ipv6 ipv6
bind policy dataset Block_list_ipv6 fe80::98c7:d8ff:ff3b:b562 -index 1
bind policy dataset Block_list_ipv6 fe80::ffc7:d8ff:fe3a:b562 -index 2
<!--NeedCopy-->
```

例 **5:**

次の例は、次の条件でクライアント IP をブロックするポリシー式を示しています。

- カスタマイズされた block_list1 で設定された IP アドレスと一致します（例 4）
- allow_list1 に含めることによって緩和されない限り、Webroot データベースにリストされている IP アドレスと一致します（例 3）。

```
> add appfw policy "Ip_Rep_Policy" "((CLIENT.IP.SRC.IPREP_IS_MALICIOUS
    || CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY("Block_list1")) && ! (
    CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY("Allow_list1")))"
    APPFW_BLOCK
<!--NeedCopy-->
```

次の例は、次の条件下でクライアント IPv6 をブロックするポリシー式を示しています。

1. カスタマイズされた block_list_IPv6 に設定された IPv6 アドレスと一致します (例 4)
2. allow_List_IPv6 に含めることで緩和されない限り、Webroot データベースにリストされている Ipv6 アドレスと一致します (例 3)。

```
add appfw policy "Ip_Rep_v6_Policy" "((CLIENT.IPV6.SRC.
    IPREP_IS_MALICIOUS || CLIENT.IPV6.SRC.TYPECAST_TEXT_T.CONTAINS_ANY("
    Block_list_ipv6")) && ! (CLIENT.IPV6.SRC.TYPECAST_TEXT_T.
    CONTAINS_ANY("Allow_list_ipv6")))" APPFW_BLOCK
<!--NeedCopy-->
```

プロキシサーバーの使用**:**

NetScaler アプライアンスがインターネットに直接アクセスせず、プロキシに接続されている場合は、プロキシに要求を送信するように IP レピュテーションクライアントを構成します。

アプライアンスのセキュリティを強化するために、プロキシサーバでプロキシユーザ名とパスワードを設定します。

コマンドプロンプトで入力します:

```
set reputation settings -proxyServer <proxy server ip> -proxyPort <
proxy server port> -proxyUsername <username> -proxyPassword <password
>
```

例:

```
> set reputation settings proxyServer 10.102.30.112 proxyPort 3128 -
proxyUsername defaultusername -proxyPassword defaultpassword
```

```
> set reputation settings -proxyServer testproxy.citrite.net -proxyPort
 3128 -proxyUsername defaultusername -proxyPassword defaultpassword
```



```
> unset reputation settings -proxyserver -proxyport -proxyUsername -
proxyPassword
> sh reputation settings
```

> 注:
>
> プロキシサーバー IP には、IP アドレスまたは完全修飾ドメイン名 (FQDN) を指定できます。

NetScaler GUI を使用して **IP** レピュテーションを設定する

1. [システム**] > [**設定**]** に移動します。[モードと機能] セクションで、リンクをクリックして [拡張機能の設定] ペインにアクセスし、[レピュテーション] チェックボックスを有効にします。
2. [**OK**] をクリックします。

NetScaler GUI を使用してプロキシサーバーを構成するには

1. [設定] タブで、[セキュリティ**] > [**レピュテーション] に移動します。
2. [設定] で、[レピュテーション設定の変更] をクリックしてプロキシサーバーを構成します。
3. レピュテーション機能を有効または無効にします。
4. 次の詳細を入力してプロキシサーバーを構成します。
   a) プロキシサーバー -IP アドレスでも完全修飾ドメイン名 (FQDN) でもかまいません。
   b) プロキシポート -[1—65535] の間の値を受け入れます。
   c) プロキシユーザー名—プロキシサーバー認証用のユーザー名を指定します。
   d) プロキシパスワード -プロキシサーバー認証用のパスワードを指定します。

   > 注**:**
   > **proxyUsername** と **proxyPassword** f フィールドが設定されている場合、**proxyServer** と **proxyPort** フィールドが有効になります。

GUI を使用してクライアント **IP** アドレスの許可リストとブロックリストを作成する

1. [構成] タブで、[**AppExpert] > [**データセット] に移動します。
2. [追加] をクリックします。

- [データセットの作成] (または [データセットの構成]) ペインで、IP アドレスのリストに意味のある名前を入力します。名前は、リストの目的を反映している必要があります。
- [タイプ] を **IPv4** または **IPv6** として選択します。
- [**Insert**] をクリックしてエントリを追加します。
- [ポリシーデータセットバインドの構成] ウィンドウで、[値] 入力ボックスに IPv4 または IPv6 形式の IP アドレスを追加します。

- インデックスを指定します。
- リストの目的を説明するコメントを追加します。この手順はオプションですが、説明的なコメントがリストの管理に役立つため、推奨されています。

同様に、ブロックリストを作成し、悪意のあると見なされる IP アドレスを追加できます。

データセットの使用と高度なポリシー式の設定の詳細については、パターンセットとデータセットも参照してください。

NetScaler GUI を使用してアプリケーションファイアウォールポリシーを構成する

1. [構成] タブで、[セキュリティ] **>** [アプリケーションファイアウォール] **>** [ポリシー] **>** [ファイアウォール] に移動します。IP レピュテーションを使用する PI 式を使用するポリシーを追加するには、[追加] をクリックします。

式エディタを使用して、独自のポリシー式を作成することもできます。このリストには、脅威カテゴリを使用して式を設定するのに役立つ、構成済みのオプションが表示されます。

ハイライト

- さまざまな種類の脅威をもたらす既知の悪意のある IP アドレスから、ネットワークのエッジで悪質なトラフィックを迅速かつ正確に阻止します。本文を解析せずにリクエストをブロックできます。
- 複数のアプリケーションの IP レピュテーション機能を動的に設定します。
- パフォーマンスを犠牲にすることなくネットワークをデータ漏えいから守り、迅速かつ簡単な導入によって保護を単一のサービスファブリックに統合します。
- 送信元と宛先 IP で IP レピュテーションチェックを実行できます。
- ヘッダーを検査して、悪意のある IP を検出することもできます。
- IP レピュテーションチェックは、フォワードプロキシとリバースプロキシの両方の展開でサポートされています。
- IP レピュテーションプロセスは Webroot に接続し、5 分ごとにデータベースを更新します。
- 高可用性 (HA) またはクラスタ展開の各ノードは、Webroot からデータベースを取得します。
- IP レピュテーションデータは、管理パーティション展開のすべてのパーティションで共有されます。
- AppExpert データセットを使用して IP アドレスのリストを作成し、Webroot データベースでブロックリストに登録されている IP の例外を追加できます。独自にカスタマイズしたブロックリストを作成して、特定の IP を悪意のあるものとして指定することもできます。
- iprep.db ファイルが/var/nslog/iprepフォルダ内に作成されます。一度作成すると、フィーチャが無効になっていても削除されません。
- レピュテーション機能が有効になると、NetScaler Webroot データベースがダウンロードされます。その後、5 分ごとに更新されます。
- Webroot データベースのメジャーバージョンはバージョン:1 です。
- マイナーバージョンは毎日更新されます。更新バージョンは 5 分ごとにインクリメントされ、マイナーバージョンがインクリメントされると 1 にリセットされます。

- PI 式を使用すると、応答側や書き換えなどの他の機能で IP レピュテーションを使用できます。
- データベース内の IP アドレスは 10 進表記です。

デバッグに関するヒント

- GUI にレピュテーション機能が表示されない場合は、適切なライセンスがあることを確認します。
- デバッグのために、`var/log/iprep.log`のメッセージをモニタします。
- **Webroot** 接続: `ns iprep: Not able to connect/resolve WebRoot`メッセージが表示された場合は、アプライアンスにインターネットアクセスがあり、DNS が設定されていることを確認します。
- プロキシサーバー: `ns iprep: iprep_curl_download: 88 curl_easy_perform failed. Error code: 5 Err msg:couldnt resolve proxy name`メッセージが表示された場合は、プロキシサーバーの設定が正確であることを確認してください。
- **IP** レピュテーション機能が動作しない: レピュテーション機能を有効にすると、IP レピュテーションプロセスが開始するまでに約 5 分かかります。IP レピュテーション機能はその期間動作しない場合があります。
- データベースのダウンロード: IP レピュテーション機能を有効にした後に IP DB データのダウンロードが失敗した場合、ログに次のエラーが表示されます。

```
iprep: iprep_curl_download:86 curl_easy_perform failed. Error code:7
Err msg:Couldn't connect to server
```

解決方法: 次の URL へのアウトバウンドトラフィックを許可するか、プロキシを設定して問題を解決します。

```
1 localdb-ip-daily.brightcloud.com:443
2 localdb-ip-rtu.brightcloud.com:443
3 api.bcti.brightcloud.com:443
4 localdb-ipv6-daily.brightcloud.com:443
5 ipce-daily.brightcloud.com:443
6 ipce-rtu.brightcloud.com:443
7 <!--NeedCopy-->
```

SSL オフロードおよび SSL アクセラレーション

August 15, 2023

SSL アクセラレーション用に構成された NetScaler アプライアンスは、SSL 処理をサーバーからオフロードすることで、SSL トランザクションを透過的に高速化します。SSL オフロードを設定するには、SSL トランザクションをインターセプトして処理し、復号化されたトラフィックをサーバーに送信するように仮想サーバーを設定します (ただし、エンドツーエンド暗号化を設定した場合、トラフィックは再暗号化されます)。サーバーから応答を受信すると、アプライアンスはクライアントとの安全なトランザクションを完了します。クライアントから見ると、トランザクシ

ョンはサーバーと直接行われているようです。SSL アクセラレーション用に構成された NetScaler は、負荷分散などの他の構成済み機能も実行します。

SSL オフロードを設定するには、SSL 証明書とキーペアが必要です。SSL 証明書をまだ持っていない場合は、これらを取得する必要があります。実行する必要のあるその他の SSL 関連のタスクには、証明書の管理、証明書失効リストの管理、クライアント認証の設定、SSL アクションとポリシーの管理などがあります。

FIPS 以外の NetScaler アプライアンスは、サーバーの秘密鍵をハードディスクに保存します。FIPS アプライアンスでは、キーはハードウェアセキュリティモジュール (HSM) と呼ばれる暗号化モジュールに格納されます。

FIPS カードをサポートしていないすべての NetScaler アプライアンス（仮想アプライアンスを含む）は、Thales nShield® Connect と SafeNet 外部 HSM をサポートしています。（MPX 9700/10500/12500/15500 アプライアンスは外部 HSM をサポートしていません。）

注：このドキュメントで説明されている一部の SSL 構成手順の FIPS 関連オプションは、FIPS 対応の NetScaler ADC アプライアンスに固有のものです。

SSL オフロード構成

August 15, 2023

SSL オフロードを構成するには、NetScaler ADC アプライアンスで SSL 処理を有効にし、SSL ベースの仮想サーバーを構成する必要があります。仮想サーバは SSL トラフィックをインターセプトし、トラフィックを復号化し、仮想サーバにバインドされたサービスに転送します。メディアストリーミングなどの時間的制約のあるトラフィックを保護するために、DTLS 仮想サーバーを構成できます。SSL オフロードを有効にするには、有効な証明書とキーをインポートし、そのペアを仮想サーバーにバインドする必要があります。

> 注
>
> リリース 13.1 ビルド 17.x から、TLSv1.2 より前のプロトコルは SSL 内部サービスでは無効になっています。デフォルト (拡張) プロファイルが有効になっている場合、`ns_default_ssl_profile_internal_frontend_ser` プロファイルは SSL 内部サービスにバインドされ、SSLv3、TLSv1.0、および TLSv1.1 プロトコルはプロファイルで無効になります。

SSL を有効にする

SSL トラフィックを処理するには、SSL 処理を有効にする必要があります。SSL 処理を有効にしなくても、仮想サーバやサービスなどの SSL ベースのエンティティを設定できます。ただし、SSL 処理が有効になるまでは機能しません。

CLI を使用して **SSL** 処理を有効にする

コマンドプロンプトで入力します。

```
enable ns feature ssl

show ns feature
<!--NeedCopy-->
```

例**:**

```
enable ns feature SSL
 Done
show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 9)     SSL Offloading                  SSL                 ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

GUI を使用して **SSL** 処理を有効にする

[システム**] > [**設定**]** に移動し、[モードと機能] グループで [基本機能の構成] をクリックし、[**SSL** オフロード] をクリックします。

サービスを構成する

NetScaler ADC アプライアンスでは、サービスは物理サーバーまたは物理サーバー上のアプリケーションを表します。設定が完了すると、アプライアンスがネットワーク上の物理サーバに到達してそのステータスを監視できるようになるまで、サービスは無効状態になります。

CLI を使用してサービスを追加する

コマンドプロンプトで次のコマンドを入力してサービスを追加し、構成を確認します。


```
add service <name> (<IP> | <serverName>) <serviceType> <port>
show service <serviceName>
<!--NeedCopy-->
```

例:

```
add service sslsvc 198.51.100.225 SSL 443

Done

sh ssl service sslsvc

            Advanced SSL configuration for Back-end SSL Service sslsvc:
            DH: DISABLED
            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                RSA: DISABLED
            Session Reuse: ENABLED          Timeout: 300 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Server Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
            Send Close-Notify: YES
            Strict Sig-Digest Check: DISABLED
            Zero RTT Early Data: ???
            DHE Key Exchange With PSK: ???
            Tickets Per Authentication Context: ???

            ECC Curve: P_256, P_384, P_224, P_521

    1)      Cipher Name: DEFAULT_BACKEND
            Description: Default cipher list for Backend SSL session
     Done
<!--NeedCopy-->
```

CLI を使用してサービスを変更または削除する

サービスを変更するには、set service コマンドを使用します。これは add service コマンドと同様ですが、既存のサービスの名前を入力する点が異なります。

サービスを削除するには、rm service コマンドを使用します。このコマンドは <name> 引数のみを受け付けます。

```
rm service <servicename>
<!--NeedCopy-->
```


例:

```
rm service sslsvc
<!--NeedCopy-->
```

サービスを変更するには、set service コマンドを使用し、任意のパラメータを選択してその設定を変更します。

```
set service <name> (<IP> | <serverName>) <serviceType> <port>
<!--NeedCopy-->
```

例:

```
set service sslsvc 198.51.100.225 SSL 443
<!--NeedCopy-->
```

GUI を使用してサービスを構成する

トラフィック管理に移動します > 負荷分散 > サービス、サービスを作成し、プロトコルを SSL として指定します。

SSL 仮想サーバ構成

セキュアセッションでは、NetScaler ADC アプライアンス上のクライアントと SSL ベースの仮想サーバー間の接続を確立する必要があります。SSL 仮想サーバは、SSL トラフィックをインターセプトし、復号化して処理してから、仮想サーバにバインドされたサービスに送信します。

注：有効な証明書/キーのペアと少なくとも 1 つのサービスがバインドされるまで、NetScaler ADC アプライアンスでは SSL 仮想サーバーがダウンしているとマークされます。SSL ベースの仮想サーバーは、プロトコルタイプが SSL または SSL_TCP の負荷分散仮想サーバーです。NetScaler ADC アプライアンスで負荷分散機能を有効にする必要があります。

CLI を使用して **SSL** ベースの仮想サーバーを追加します

コマンドプロンプトで次のコマンドを入力し、SSL ベースの仮想サーバーを追加して構成を確認します。

```
add lb vserver <name> (serviceType) <IPAddress> <port>
show ssl vserver <name>
<!--NeedCopy-->
```

例:

```
add lb vserver sslvs SSL 192.0.2.240 443
Done

sh ssl vserver sslvs

            Advanced SSL configuration for VServer sslvs:
```



```
7               DH: DISABLED
8               DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                    RSA: ENABLED          Refresh Count: 0
9               Session Reuse: ENABLED          Timeout: 120 seconds
10              Cipher Redirect: DISABLED
11              SSLv2 Redirect: DISABLED
12              ClearText Port: 0
13              Client Auth: DISABLED
14              SSL Redirect: DISABLED
15              Non FIPS Ciphers: DISABLED
16              SNI: DISABLED
17              OCSP Stapling: DISABLED
18              HSTS: DISABLED
19              HSTS IncludeSubDomains: NO
20              HSTS Max-Age: 0
21              SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                    ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
22              Push Encryption Trigger: Always
23              Send Close-Notify: YES
24              Strict Sig-Digest Check: DISABLED
25              Zero RTT Early Data: DISABLED
26              DHE Key Exchange With PSK: NO
27              Tickets Per Authentication Context: 1
28              ECC Curve: P_256, P_384, P_224, P_521
29
30      1)      Cipher Name: DEFAULT
31              Description: Default cipher list with encryption strength
                    >= 128bit
32       Done
33  <!--NeedCopy-->
```

CLI を使用して **SSL** ベースの仮想サーバを変更または削除する

SSL 仮想サーバの負荷分散プロパティを変更するには、`set lb vserver` コマンドを使用します。set コマンドは`add lb vserver`コマンドと似ていますが、既存の仮想サーバの名前を入力する点が異なります。**SSL** ベースの仮想サーバの **SSL** プロパティを変更するには、`set ssl vserver` コマンドを使用します。詳細については、このページの「SSL 仮想サーバーパラメーター」を参照してください。

SSL 仮想サーバを削除するには、<name>引数のみを受け入れる`rm lb vserver`コマンドを使用します。

GUI を使用して **SSL** ベースの仮想サーバを構成する

[トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを作成し、プロトコルを SSL として指定します。

サービスの **SSL** 仮想サーバーへのバインド

ADC アプライアンスは、復号化された SSL データをネットワーク内のサーバに転送します。データを転送するには、これらの物理サーバを表すサービスを、SSL データを受信する仮想サーバにバインドする必要があります。

通常、ADC アプライアンスと物理サーバ間のリンクはセキュアです。したがって、アプライアンスと物理サーバ間のデータ転送を暗号化する必要はありません。ただし、アプライアンスとサーバー間のデータ転送を暗号化することで、エンドツーエンド暗号化を提供できます。詳細については、エンドツーエンド暗号化による SSL オフロードの設定を参照してください。

注: SSL ベースの仮想サーバーにサービスをバインドする前に、ADC アプライアンスで負荷分散機能を有効にしてください。

CLI を使用してサービスを仮想サーバーにバインドします

コマンドプロンプトで次のコマンドを入力して、サービスを仮想サーバーにバインドし、構成を確認します。

```
bind lb vserver <name> <serviceName>
show lb vserver <name>
<!--NeedCopy-->
```

例:

```
bind lb vserver sslvs sslsvc
     Done

sh lb vserver sslvs

            sslvs (192.0.2.240:443) - SSL       Type: ADDRESS
            State: DOWN[Certkey not bound]
            Last state change was at Wed May  2 11:43:04 2018
            Time since last state change: 0 days, 00:13:21.150
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Appflow logging: ENABLED
            No. of Bound Services :  1 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION      BackupMethod:
               ROUNDROBIN
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none
            L2Conn: OFF
            Skip Persistency: None
            Listen Policy: NONE
            IcmpResponse: PASSIVE
```



```
            RHIstate: PASSIVE
            New Service Startup Request Rate: 0 PER_SECOND, Increment
               Interval: 0
            Mac mode Retain Vlan: DISABLED
            DBS_LB: DISABLED
            Process Local: DISABLE
            Traffic Domain: 0
            TROFS Persistence honored: ENABLED
            Retain Connections on Cluster: NO
    1) sslsvc (198.51.100.225: 443) - SSL State: DOWN       Weight: 1
     Done
<!--NeedCopy-->
```

CLI を使用して仮想サーバからサービスをバインド解除する

コマンドプロンプトで、次のコマンドを入力します。

```
unbind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

```
unbind lb vserver sslvs sslsvc
     Done
<!--NeedCopy-->
```

GUI を使用してサービスを仮想サーバーにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、[サービスとサービスグループ] セクションの **[**** 負荷分散仮想サーバーサービスバインディング ****]** タイルをクリックします。
3. [負荷分散仮想サーバーサービスバインド] ページで、**[**バインドの追加] タブをクリックし、**[**** サービスの選択**]** の [クリックして選択 ****] をクリックし、バインドするサービスの横にあるチェックボックスをオンにします。
4. [選択] をクリックして [バインド] をクリックします

複数のサイトを安全にホストするためのサーバー名表示 **(SNI)** 仮想サーバーを構成する

仮想ホスティングは、同じ IP アドレスを持つ複数のドメイン名をホストするために Web サーバーによって使用されます。アプライアンスは、透過的な SSL サービスまたは仮想サーバベースの SSL オフロードを使用して Web サーバから SSL 処理をオフロードすることにより、複数のセキュアドメインのホスティングをサポートします。ただし、複数の Web サイトが同じ仮想サーバーでホストされている場合は、想定されるホスト名が仮想サーバーに送信される前に SSL ハンドシェイクが完了します。その結果、アプライアンスは、接続の確立後にクライアントに提示する証

明書を判断できません。この問題は、仮想サーバで SNI を有効にすることで解決します。SNI は、クライアントがハンドシェイクの開始時にホスト名を提供するために使用するトランスポート層セキュリティ (TLS) 拡張です。ADC アプライアンスはこのホスト名を共通名と比較し、一致しない場合はサブジェクト代替名 (SAN) と比較します。名前が一致すると、アプライアンスは対応する証明書をクライアントに提示します。

ワイルドカード SSL 証明書は、同じ組織がこれらのドメインを管理していて、第 2 レベルのドメイン名が同じ場合に、複数のサブドメインで SSL 暗号化を有効にするのに役立ちます。たとえば、一般名「*.sports.net」を使用してスポーツネットワークに発行されたワイルドカード証明書は、「login.sports.net」や「help.sports.net」などのドメインを保護するために使用できます。「login.ftp.sports.net」ドメインをセキュリティで保護することはできません。

注:
ADC アプライアンスでは、**SAN** フィールドのドメイン名、URL、電子メール ID DNS エントリのみが比較されます。

-snicert オプションを使用すると、複数のサーバー証明書を 1 つの SSL 仮想サーバーまたはトランスペアレントサービスにバインドできます。仮想サーバーまたはサービスで SNI が有効になっている場合、仮想サーバーまたはサービスはこれらの証明書を発行します。SNI はいつでも有効にできます。

CLI を使用して複数のサーバ証明書を **1** つの **SSL** 仮想サーバにバインドする

コマンドプロンプトで次のコマンドを入力して SNI を構成し、構成を確認します。

```
set ssl vserver <vServerName>@ [-SNIEnable ( ENABLED | DISABLED )]

bind ssl vserver <vServerName>@ -certkeyName <string> -SNICert

show ssl vserver <vServerName>
<!--NeedCopy-->
```

CLI を使用して複数のサーバ証明書を 1 つのトランスペアレントサービスにバインドするには、前述のコマンドで service を`vserver`に、service name を`vservername`に置き換えます。

注**:**`-clearTextPort 80` オプションを使用して SSL サービスを作成します。

GUI を使用して複数のサーバ証明書を **1** つの **SSL** 仮想サーバにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. SSL 仮想サーバーを開き、[証明書] で [サーバー証明書] を選択します。
3. 証明書を追加するか、一覧から証明書を選択して、[**SNI** のサーバー証明書] をクリックします。
4. **[**詳細設定**]** で [**SSL** パラメータ] を選択します。
5. [**SNI** 有効化] をクリックします。

バックエンドサービスでの **SNI** のサポート

> 注:SNI は DTLS バックエンドサービスではサポートされていません。

NetScaler アプライアンスは、バックエンドでサーバー名表示（SNI）をサポートしています。つまり、ハンドシェイクが正常に完了するために、共通名がクライアント hello のサーバ名としてバックエンドサーバに送信されます。このサポートは、連邦政府システムインテグレータのお客様のセキュリティ要件を満たすのに役立ちます。また、SNI には、ファイアウォールで数百もの異なる IP アドレスやポートを開くのではなく、1 つのポートしか使用できないという利点があります。

連邦システムインテグレーターのお客様のセキュリティ要件には、2012R2 および WAP サーバーにおける Active Directory フェデレーションサービス (ADFS) 3.0 のサポートが含まれます。この要件を満たすには、NetScaler ADC アプライアンスのバックエンドで SNI をサポートする必要があります。

> 注:
>
> SNI を機能させるには、クライアント hello のサーバ名が、SSL 仮想サーバにバインドされたバックエンドサービスに設定されたホスト名と一致する必要があります。たとえば、バックエンドサーバーのホスト名が www.mail.example.com の場合、SNI 対応バックエンドサービスはサーバー名をhttps://www.mail.example.comとして構成する必要があります。このホスト名は、クライアント hello のサーバ名と一致する必要があります。

バックエンドサービスでの動的 **SNI** のサポート

NetScaler アプライアンスは、バックエンド TLS 接続で動的 SNI をサポートします。つまり、アプライアンスはクライアント接続で SNI を学習し、サーバー側接続で SNI を使用します。SSL サービス、サービスグループ、またはプロファイルに共通名を指定する必要がなくなりました。Client Hello メッセージの SNI 拡張で受信した共通名は、バックエンド SSL 接続に転送されます。

以前は、SSL サービス、サービスグループ、および SSL プロファイルに静的 SNI を設定する必要がありました。その結果、設定された静的 SNI 拡張だけがサーバに送信されました。クライアントが同時に複数のドメインにアクセスする必要がある場合、ADC アプライアンスはクライアントから受け取った SNI をバックエンドサービスに送信できませんでした。代わりに、設定された静的な共通名が送信されました。バックエンドサーバが複数のドメインに対して設定されている場合、サーバは、アプライアンスからの Client Hello メッセージで受信した SNI に基づいて、正しい証明書で応答できます。

注意点**:**

- フロントエンドで SNI を有効にし、正しい SNI 証明書を SSL 仮想サーバーにバインドする必要があります。フロントエンドで SNI を有効にしないと、SNI 情報はバックエンドに渡されません。
- サーバ認証を有効にすると、サーバ証明書が CA 証明書によって検証され、サーバ証明書のコモンネーム/SAN エントリが SNI と照合されます。そのため、CA 証明書をサービスにバインドする必要があります。
- ダイナミック SNI が有効な場合、バックエンド接続と SSL セッションの再利用は SNI に基づきます。

動的 SNI が有効な場合、SSL モニターは SNI を送信しません。SNI ベースのプロービングでは、静的 SNI が設定されているバックエンドプロファイルを SSL モニターにアタッチします。モニターは SNI と同じカスタムヘッダーで構成する必要があります。

CLI を使用してバックエンドサービスで **SNI** を設定する

コマンドプロンプトで入力します。

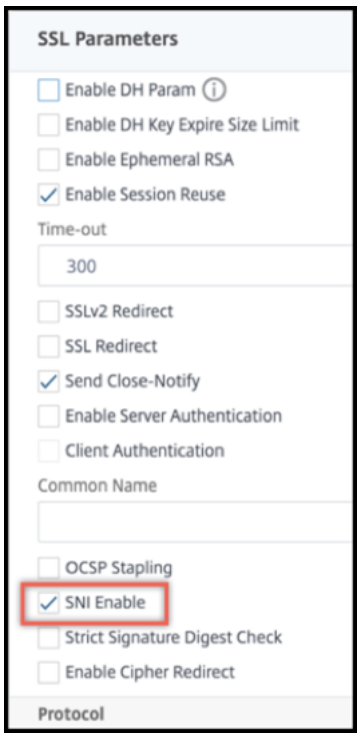
```
add service <name>  <IP>  <serviceType>  <port>

add lb vserver <name>  <IPAddress> <serviceType>  <port>

bind lb vserver <name> <serviceName>

set ssl service <serviceName> -SNIEnable ENABLED -commonName <string>

set ssl profile <name> -SNIEnable ENABLED
<!--NeedCopy-->
```

例**:**

```
    add service service_ssl 198.51.100.100 SSL 443

    add lb vserver ssl-vs 203.0.113.200 SSL 443

    bind lb vserver ssl-vs service_ssl

    set ssl service service_ssl -SNIEnable ENABLED  - commonName www.
       example.com

    set ssl profile sslprof -SNIEnable ENABLED
<!--NeedCopy-->
```

GUI を使用してバックエンドサービスで **SNI** を設定する

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
2. SSL サービスを選択し、[詳細設定] で [**SSL** パラメータ] をクリックします。
3. [**SNI** 有効化] をクリックします。



GUI を使用して **SSL** プロファイルで **SNI** を設定する

1. **[**システム**] > [**プロファイル**] > [SSL** プロファイル**]** に移動します。
2. ［追加］をクリックします。
3. **[**基本設定**]** で [**SNI** 有効] を選択します。

4. ［**OK**］をクリックします。

セキュアモニタを **SNI** 対応バックエンドサービスにバインドする

HTTP、HTTP-ECV、TCP、または TCP-ECV タイプのセキュアモニタを、SNI をサポートするバックエンドサービスおよびサービスグループにバインドできます。ただし、動的 SNI が有効な場合、モニタプローブは SNI 拡張を送信しません。SNI プローブを送信するには、バックエンド SSL プロファイルで静的 SNI を有効にし、プロファイルをモニタにバインドします。モニタのカスタムヘッダーを、モニタプローブの client hello で SNI 拡張として送信されるサーバ名に設定します。

CLI を使用して、セキュアモニタを設定して **SNI** 対応バックエンドサービスにバインドする　コマンドプロンプトで入力します。

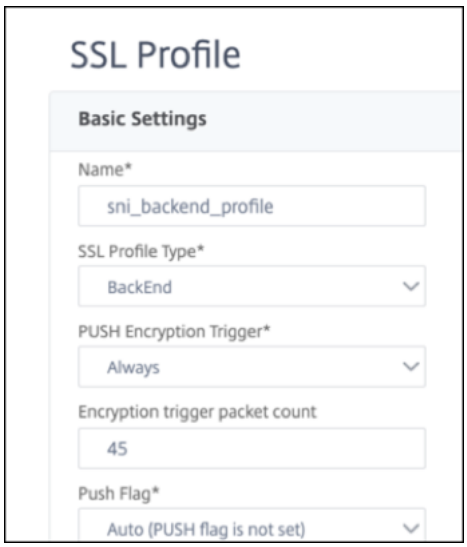
```
add lb monitor <monitorName> <type> -secure YES
add ssl profile <name> -sslProfileType BackEnd
set lb monitor <monitorName> <type> -customHeaders <string> -sslprofile
    <backend ssl profile>
set ssl profile <name> -sniEnable ENABLED -commonName <string>
bind service <name> -monitorName <string>
<!--NeedCopy-->
```

例**:**

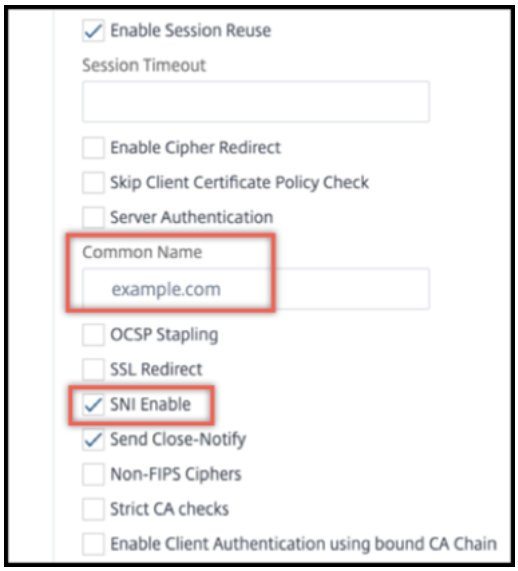
```
add ssl profile sni_backend_profile -sslProfileType BackEnd
set ssl profile sni_backend_profile -sniEnable ENABLED -commonName
   example.com
add lb monitor http-ecv-mon HTTP-ECV -secure YES
set monitor http-ecv-mon HTTP-ECV -customHeaders "Host: example.com\r\n
   " -sslprofile sni_backend_profile
bind service ssl_service  - monitorName http-ecv-mon
<!--NeedCopy-->
```

GUI を使用して、セキュアモニタを構成し、**SNI** 対応バックエンドサービスにバインドする

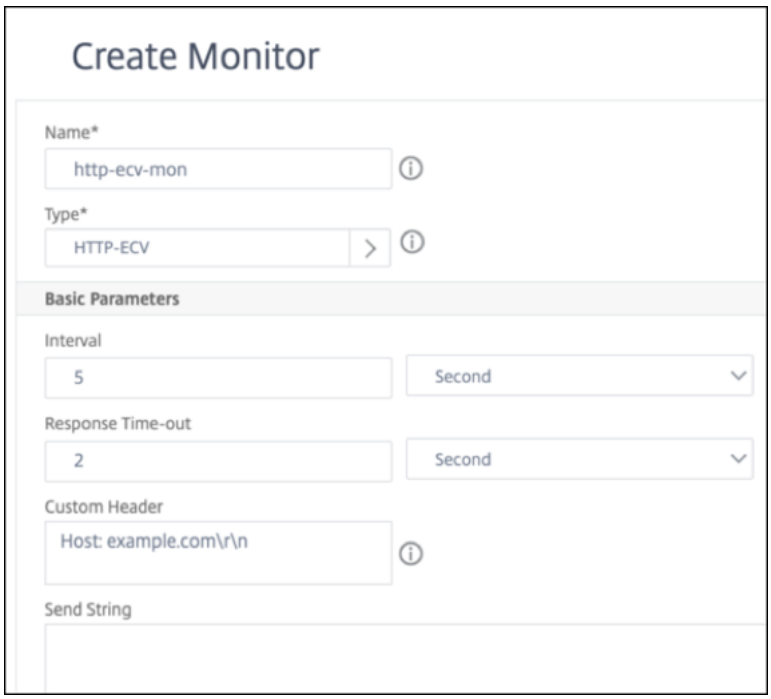
1. **[**システム**] > [**プロファイル**] > [SSL** プロファイル**]** に移動します。
2. ［追加］をクリックします。
3. プロファイルの名前を指定し、[**SSL** プロファイルタイプ] で [バックエンド] を選択します。



4. 共通名 (ホストヘッダーと同じ) を指定し、[**SNI Enable**] を選択します。



5. [**OK**] をクリックします。
6. [トラフィック管理**] > [**負荷分散**] > [**監視] に移動します。
7. [追加] をクリックします。
8. モニターの名前を指定します。[タイプ] で、[HTTP]、[HTTP-ECV]、[TCP]、または [TCP-ECV] を選択します。
9. カスタムヘッダーを指定します。



10. [安全] を選択します。
11. [**SSL** プロファイル] で、前の手順で作成したバックエンド SSL プロファイルを選択します。
12. ［作成］をクリックします。
13. ［**Traffic Management**］ **>** ［**Load Balancing**］ **>** ［**Services**］の順に移動します。
14. SSL サービスを選択し、[編集] をクリックします。
15. [モニター] で、[バインドの追加] をクリックし、前の手順で作成したモニターを選択して [バインド] をクリックします。

GUI を使用してセキュアモニタを構成し、**SNI** 対応バックエンドサービスにバインドする

1. **[**トラフィック管理**]** > **[**負荷分散**]** > **[**監視**]** に移動します。
2. **HTTP-ECV** または TCP-ECV****** タイプのモニターを追加し、**[**** カスタムヘッダー] を指定します。
3. ［**Create**］を選択します。
4. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
5. SSL サービスを選択し、[編集] をクリックします。
6. [モニター] で [バインドの追加] をクリックし、手順 3 で作成したモニターを選択して [バインド] をクリックします。

不明なサーバー名でもハンドシェイクの継続を許可

> 注:
>
> この機能はリリース 13.1 ビルド 45.x 以降で使用できます。

SNI が有効になっていて、NetScaler アプライアンスが不明なサーバー名のクライアント Hello を受信すると、SSL ハンドシェイクは終了します。リリース 13.1 ビルド 45.x 以降、アプライアンスは不明なサーバー名でも SSL ハンドシェイクを継続できるようにし、ハンドシェイクのドロップまたは完了の決定はクライアントに任せます。SNI がパラメーターを使用して **ENABLED** に設定されている場合、フロントエンド SSL プロファイルでこの設定を構成できます。`allowUnknownSNI`

SNI ベースのルールで転送アクションを使用する必要がある場合は、このパラメータを無効のままにしてください。たとえば、仮想サーバー v1 で SNI を有効にし、特定のドメイン（www.example.com）のすべての要求を仮想サーバー v2 に転送するポリシーを設定したとします。以前は、このドメインの v1 で受信したリクエストは自動的に v2 に転送されていました。ただし、`allowunknownSNI` パラメータが有効な場合、リクエストは v1 で処理されます。アプライアンスが v1 でリクエストを処理するには、パラメータを無効にする必要があります。

CLI を使用して未知の **SNI** を許可するように設定

コマンドプロンプトで入力します。

```
set ssl profile default_profile -SNIEnable Enabled -allowUnknownSNI <DISABLED/ENABLED>
```

‘AllowUnknownSNI パラメータはデフォルトでは無効になっています。その結果、アプライアンスは不明なサーバー名のハンドシェイクを中止します。この設定を有効にするには、次のように入力します。

```
set ssl profile default_profile -SNIEnable Enabled -allowUnknownSNI ENABLED
```

GUI を使用して未知の **SNI** を許可するように設定

1. [システム] > [プロファイル] > [**SSL** プロファイル] に移動します。
2. プロファイルを追加する場合は、**SSL** プロファイルタイプリストで **FrontEnd** を選択します。それ以外の場合は、既存のフロントエンドプロファイルを編集できます。
3. 「未知の **SNI** を許可」を選択します。
4. 「**OK**」をクリックし、「完了」をクリックします。

証明書とキーのペアを追加または更新する

> 注:
>
> 既存の証明書とキーがない場合は、「証明書の作成」を参照してください。
>
> ECDSA [証明書とキーのペアを作成するには、ECDSA 証明書とキーのペアの作成] をクリックします。
>
> ビルド 41.x から、63 文字までの証明書名がサポートされます。
>
> リリース 13.0 ビルド 79.x からは、パスワードで保護された証明書とキーのペアが常に正常に追加されます。以前は、NetScaler アプライアンスで強力なパスワードオプションが有効になっていると、パスワードで保護された証明書とキーのペアが追加されないことがありました。ただし、以前のビルドにダウングレードすると、証明書キーの設定は失われます。また、証明書とキーのペアの NITRO API レスポンスでは、`passplain`変数ではなく`passcrypt`変数が送信されます。

SSL トランザクションの場合、サーバーには有効な証明書と、対応する秘密鍵と公開鍵のペアが必要です。SSL データはサーバーの公開鍵で暗号化されます。公開鍵はサーバーの証明書から入手できます。復号化には、対応する秘密鍵が必要です。SSL 証明書とキーのペアを追加する際に使用される秘密鍵のパスワードは、NetScaler アプライアンスごとに固有の暗号化キーを使用して保存されます。

ADC アプライアンスは、SSL トランザクションをサーバからオフロードします。したがって、サーバの証明書と秘密キーがアプライアンスに存在し、証明書が対応する秘密キーとペアになっている必要があります。この証明書とキーのペアは、SSL トランザクションを処理する仮想サーバーにバインドする必要があります。

注: NetScaler アプライアンスのデフォルト証明書は 2048 ビットです。以前のビルドでは、デフォルトの証明書は 512 ビットまたは 1024 ビットでした。リリース 11.0 にアップグレードした後、`"ns-"`で始まる古い証明書とキーのペアをすべて削除し、アプライアンスを再起動して 2048 ビットのデフォルト証明書を自動的に生成する必要があります。

証明書とキーの両方をアプライアンスに追加するには、NetScaler ADC アプライアンスのローカルストレージに存在する必要があります。証明書またはキーファイルがアプライアンス上にない場合は、ペアを作成する前に証明書またはキーファイルをアプライアンスにアップロードします。

重要: 証明書とキーは、デフォルトで /nsconfig/ssl ディレクトリに保存されます。証明書またはキーが他の場所に保存されている場合は、NetScaler アプライアンス上のファイルへの絶対パスを指定する必要があります。NetScaler FIPS アプライアンスは外部キー（非 FIPS キー）をサポートしていません。FIPS アプライアンスでは、ハードディスクやフラッシュメモリなどのローカルストレージデバイスからキーをロードすることはできません。FIPS キーは、アプライアンスのハードウェアセキュリティモジュール (HSM) に存在する必要があります。

NetScaler アプライアンスでは RSA キーのみがサポートされています。

通知期間を設定し、有効期限モニターが証明書が期限切れになる前にプロンプトを発行できるようにします。

NetScaler アプライアンスは、証明書と秘密鍵ファイルの次の入力形式をサポートしています。

- PEM-プライバシー強化メール
- DER-識別符号化規則
- PFX-個人情報交換

ソフトウェアはフォーマットを自動的に検出します。そのため、inform パラメーターで形式を指定する必要はなくなりました。フォーマット (正しいか正しくない) を指定した場合、ソフトウェアはそのフォーマットを無視します。証明書と鍵ファイルの形式は同じである必要があります。

注**:** 証明書は、次のいずれかのハッシュアルゴリズムを使用して署名する必要があります。

- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

MPX アプライアンスは、次のサイズまで、512 ビット以上の証明書をサポートします。

- 仮想サーバー上の 4096 ビットサーバー証明書
- サービス上の 4096 ビットのクライアント証明書
- 4096 ビット CA 証明書（中間証明書とルート証明書を含む）
- バックエンドサーバー上の 4096 ビット証明書
- 4096 ビットのクライアント証明書 (仮想サーバーでクライアント認証が有効になっている場合)

VPX 仮想アプライアンスは、次のサイズまで、512 ビット以上の証明書をサポートします。

- 仮想サーバー上の 4096 ビットサーバー証明書
- サービス上の 4096 ビットのクライアント証明書
- 4096 ビット CA 証明書（中間証明書とルート証明書を含む）
- バックエンドサーバー上の 4096 ビット証明書
- 4096 ビットのクライアント証明書 (仮想サーバーでクライアント認証が有効になっている場合)

リリース 13.1 ビルド 17.x 以降、すべての NetScaler プラットフォームは、RSASSA-PSS アルゴリズムを使用して署名された証明書をサポートしています。
これらのアルゴリズムは X.509 証明書パスの検証でサポートされています。
次の表は、NetScaler アプライアンスでサポートされている RSASSA-PSS パラメーターセットを示しています。

パブリックキー OID	マスク生成関数 (MGF)	MGF ダイジェスト機能	シグネチャダイジェスト関数	塩の長さ
RSAEncryption	MGF1	SHA-256	SHA-256	32 バイト
RSAEncryption	MGF1	SHA-384	SHA-384	48 バイト
RSAEncryption	MGF1	SHA-512	SHA-512	64 バイト

> 注
>
> NetScaler SDX アプライアンスは、512 ビット以上の証明書をサポートします。アプライアンスでホストされている各 NetScaler ADC VPX インスタンスは、VPX 仮想アプライアンスの前述の証明書サイズをサポートします。ただし、SSL チップがインスタンスに割り当てられている場合、そのインスタンスは MPX アプライアンスでサポートされる証明書サイズをサポートします。

CLI を使用した証明書とキーのペアの追加

コマンドプロンプトで次のコマンドを入力して、証明書とキーのペアを追加し、構成を確認します。

```
add ssl certKey <certkeyName> -cert <string>[(-key <string> [-password
    ]) | -fipsKey <string>] [-inform ( DER | PEM )] [<passplain>] [-
    expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]]

show ssl certKey [<certkeyName>]
<!--NeedCopy-->
```

例**:**

```
add ssl certKey sslckey -cert server_cert.pem -key server_key.pem -
    password ssl -expiryMonitor ENABLED -notificationPeriod 30
 Done
Note: For FIPS appliances, replace -key with -fipskey

show ssl certKey sslckey
        Name: sslckey           Status: Valid,   Days to expiration
            :8418
        Version: 3
        Serial Number: 01
        Signature Algorithm: md5WithRSAEncryption
        Issuer:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.root.com
        Validity
                Not Before: Jul 15 02:25:01 2005 GMT
                Not After : Nov 30 02:25:01 2032 GMT
        Subject:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.server.com
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048
 Done
<!--NeedCopy-->
```

CLI を使用して証明書とキーのペアを更新または削除する

証明書とキーのペアの有効期限モニタまたは通知期間を変更するには、`set ssl certkey` コマンドを使用します。証明書とキーのペアの証明書またはキーを置き換えるには、`update ssl certkey` コマンドを使用します。`update ssl certkey`コマンドには、ドメインチェックをオーバーライドするための追加パラメータ

があります。どちらのコマンドでも、既存の証明書とキーのペアの名前を入力します。SSL 証明書とキーのペアを削除するには、`rm ssl certkey`コマンドを使用します。このコマンドでは<`certkeyName`>引数のみを受け付けます。

例**:**

```
set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED )
       [-notificationPeriod <positive_integer>]]

update ssl certKey <certkeyName> [-cert <string> [-password]] [-key
       <string> | -fipsKey <string>] [-inform <inform>] [-noDomainCheck
          ]
<!--NeedCopy-->
```

GUI を使用して証明書とキーのペアを追加または更新する

1. トラフィック管理 **> SSL >** 証明書 **>** サーバに移動します。
2. 次のパラメータの値を入力し、[**Install**] をクリックします。
   - 証明書とキーのペア名-証明書と秘密キーのペアの名前。
   - 証明書ファイル名-認証局から受け取った署名付き証明書。
   - [Key File Name]：証明書とキーのペアを形成するために使用される秘密キーファイルの名前と、オプションでパスです。

証明書とキーのペアを **SSL** 仮想サーバーにバインドする

重要: 証明書を SSL 仮想サーバーにバインドする前に、中間証明書をこの証明書にリンクします。証明書のリンクの詳細については、「証明書のチェーンを作成する」を参照してください。

SSL トランザクションの処理に使用される証明書は、SSL データを受信する仮想サーバーにバインドする必要があります。SSL データを受信する仮想サーバが複数ある場合は、有効な証明書とキーのペアをそれぞれにバインドする必要があります。

NetScaler アプライアンスにアップロードした有効な既存の SSL 証明書を使用してください。テスト目的の代わりに、アプライアンスに独自の SSL 証明書を作成します。アプライアンスで FIPS キーを使用して作成された中間証明書は、SSL 仮想サーバーにバインドできません。

SSL ハンドシェイク中、クライアント認証中の証明書要求メッセージに、サーバはサーバにバインドされているすべての認証局（CA）の識別名（DN）を一覧表示します。サーバーは、このリストからのみクライアント証明書を受け入れます。特定の CA 証明書の DN 名を SSL クライアントに送信したくない場合は、`skipCA` フラグを設定します。この設定は、特定の CA 証明書の識別名を SSL クライアントに送信してはならないことを示します。

独自の証明書を作成する方法の詳細については、「証明書の管理」を参照してください。

注：信頼できる認証局が発行する有効な SSL 証明書のみを使用することをお勧めします。

CLI を使用して **SSL** 証明書とキーのペアを仮想サーバにバインドする

コマンドプロンプトで次のコマンドを入力して、SSL 証明書とキーのペアを仮想サーバーにバインドし、構成を確認します。

```
-  bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName > -CA -skipCAName
-  show ssl vserver <vServerName>
<!--NeedCopy-->
```

例:

```
bind ssl vs vs1 -certkeyName cert2 -CA -skipCAName
 Done
sh ssl vs vs1

 Advanced SSL configuration for VServer vs1:

 DH: DISABLED

 Ephemeral RSA: ENABLED Refresh Count: 0

 Session Reuse: ENABLED Timeout: 120 seconds

 Cipher Redirect: DISABLED

 SSLv2 Redirect: DISABLED

 ClearText Port: 0

 Client Auth: DISABLED

 SSL Redirect: DISABLED

 Non FIPS Ciphers: DISABLED

 SNI: DISABLED

 OCSP Stapling: DISABLED

 HSTS: DISABLED

 IncludeSubDomains: NO

 HSTS Max-Age: 0

 SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED

 Push Encryption Trigger: Always

 Send Close-Notify: YES

```



```
41   Strict Sig-Digest Check: DISABLED
42
43 ECC Curve: P_256, P_384, P_224, P_521
44
45   1) CertKey Name: cert1 CA Certificate OCSPCheck: Optional CA_Name Sent
46   2) CertKey Name: cert2 CA Certificate OCSPCheck: Optional CA_Name
       Skipped
47   1) Cipher Name: DEFAULT
48
49 Description: Default cipher list with encryption strength >= 128bit
50 Done
51 <!--NeedCopy-->
```

CLI を使用して **SSL** 証明書とキーのペアを仮想サーバからバインド解除する

`unbind ssl certKey <certkeyName>` コマンドを使用して証明書とキーのペアを仮想サーバからバインド解除しようとすると、エラーメッセージが表示されます。このエラーは、コマンドの構文が変更されたために表示されます。コマンドプロンプトで、次のコマンドを入力します。

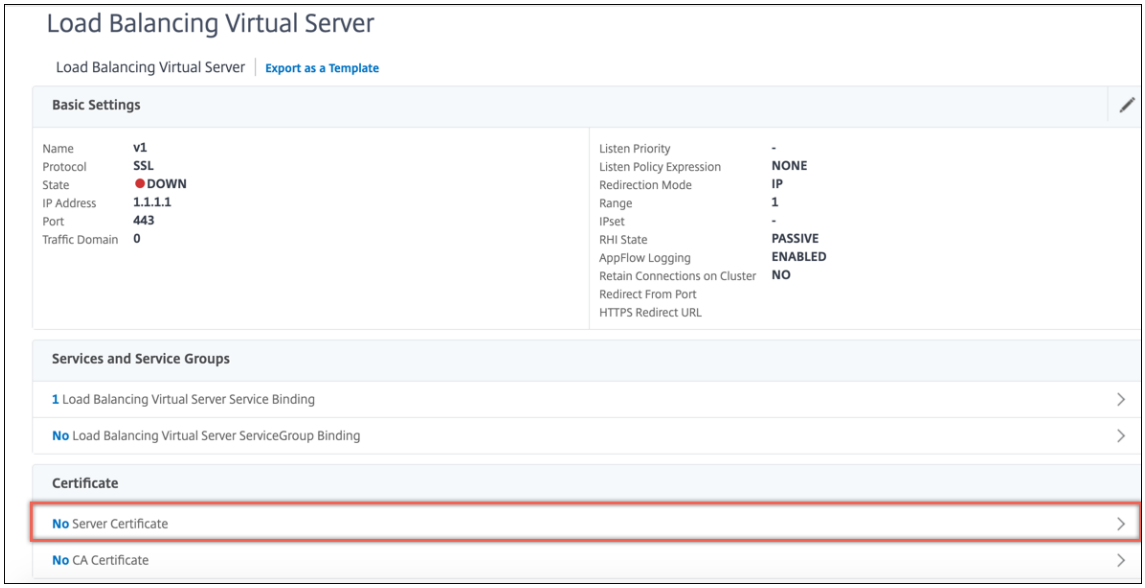
```
1 unbind ssl vserver <vServerName> -certkeyName <string>
2 <!--NeedCopy-->
```

例**:**

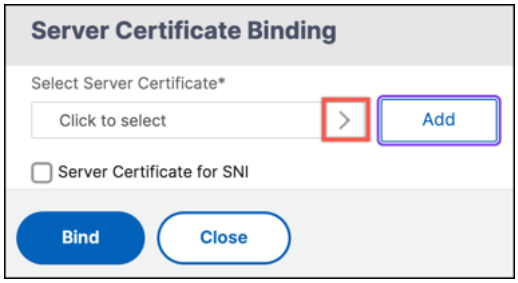
```
1 unbind ssl vserver vssl -certkeyName sslckey
2 <!--NeedCopy-->
```

GUI を使用して **SSL** 証明書とキーのペアを仮想サーバにバインドする

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、SSL 仮想サーバーを開きます。[証明書] セクション内をクリックします。



2. 矢印をクリックして、証明書とキーのペアを選択します。



3. リストから証明書とキーのペアを選択します。
4. 証明書とキーのペアを仮想サーバーにバインドします。サーバー証明書を SNI 証明書として追加するには、SNI の **[**サーバー証明書] を選択します。

SSL 仮想サーバパラメータ

SSL 仮想サーバの高度な SSL 構成を設定します。SSL プロファイルでは、これらのパラメータの多くを設定することもできます。SSL プロファイルで設定できるパラメータの詳細については、SSL プロファイルパラメータを参照してください。

CLI を使用した **SSL** 仮想サーバーのパラメータの設定

コマンドプロンプトで入力します。

```
set ssl vserver <vServerName>@ [-clearTextPort <port>] [-dh ( ENABLED |
    DISABLED ) -dhFile <string>] [-dhCount <positive_integer>][-
    dhKeyExpSizeLimit ( ENABLED | DISABLED )] [-eRSA ( ENABLED |
    DISABLED) [-eRSACount <positive_integer>]] [-sessReuse ( ENABLED |
```



```
      DISABLED )[-sessTimeout <positive_integer>]] [-cipherRedirect (
      ENABLED | DISABLED ) [-cipherURL <URL>]] [-sslv2Redirect ( ENABLED |
       DISABLED )[-sslv2URL <URL>]] [-clientAuth ( ENABLED | DISABLED ) [-
      clientCert ( Mandatory | Optional )]] [-sslRedirect ( ENABLED |
      DISABLED )][-redirectPortRewrite ( ENABLED | DISABLED )] [-ssl2 (
      ENABLED | DISABLED )] [-ssl3 ( ENABLED | DISABLED )] [-tls1 (
      ENABLED | DISABLED )] [-tls11 ( ENABLED | DISABLED )] [-tls12 (
      ENABLED | DISABLED )][-tls13 ( ENABLED | DISABLED )] [-SNIEnable (
      ENABLED | DISABLED )][-ocspStapling ( ENABLED | DISABLED )] [-
      pushEncTrigger <pushEncTrigger>] [-sendCloseNotify ( YES | NO )] [-
      dtlsProfileName <string>] [-sslProfile <string>] [-HSTS ( ENABLED |
      DISABLED )][-maxage <positive_integer>] [-IncludeSubdomains ( YES |
      NO )][-strictSigDigestCheck ( ENABLED | DISABLED )] [-
      zeroRttEarlyData (ENABLED | DISABLED )] [-
      tls13SessionTicketsPerAuthContext <positive_integer>] [-
      dheKeyExchangeWithPsk ( YES | NO )]
<!--NeedCopy-->
```

Diffie-Hellman (DH) パラメーター

SSL トランザクションを設定するために DH キー交換を必要とする暗号をアプライアンスで使用するには、アプライアンスで DH キーエクスチェンジを有効にします。ネットワークに基づいて他の設定を構成します。

CLI を使用して DH パラメータを設定する必要がある暗号の一覧を表示するには、sh cipher DH と入力します。

設定ユーティリティを使用して DH パラメータを設定する必要がある暗号を一覧表示するには、[トラフィック管理] **> [SSL] >** [暗号グループ] に移動し、[**DH**] をダブルクリックします。

DH キー交換を有効にする方法の詳細については、Diffie-Hellman (DH) キーの生成を参照してください。

CLI を使用した **DH** パラメータの設定　コマンドプロンプトで次のコマンドを入力して DH パラメータを構成し、構成を確認します。

```
-  `set ssl vserver <vserverName> -dh <Option> -dhCount <
    RefreshCountValue> -filepath <string>
-  show ssl vserver <vServerName>`
<!--NeedCopy-->
```

例**:**

```
set ssl vserver vs-server -dh ENABLED  -dhFile /nsconfig/ssl/ns-server.
   cert -dhCount 1000
Done

show ssl vserver vs-server

            Advanced SSL configuration for VServer vs-server:
            DH: ENABLED
            Ephemeral RSA: ENABLED          Refresh Count: 1000
            Session Reuse: ENABLED          Timeout: 120 seconds
```



```
10              Cipher Redirect: DISABLED
11              SSLv2 Redirect: DISABLED
12              ClearText Port: 0
13              Client Auth: DISABLED
14              SSL Redirect: DISABLED
15              Non FIPS Ciphers: DISABLED
16              SNI: DISABLED
17              OCSP Stapling: DISABLED
18              HSTS: DISABLED
19              HSTS IncludeSubDomains: NO
20              HSTS Max-Age: 0
21              SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
                   ENABLED  TLSv1.2: ENABLED
22
23      1)      Cipher Name: DEFAULT
24              Description: Predefined Cipher Alias
25  Done
26  <!--NeedCopy-->
```

GUI を使用して **DH** パラメータを設定する

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、[**DH Param** を有効にする] を選択し、更新回数とファイルパスを指定します。

エフェメラル **RSA**

Ephemeral RSA を使用すると、サーバー証明書がエクスポートクライアント (1024 ビット証明書) をサポートしていない場合でも、エクスポートクライアントはセキュアサーバーと通信できます。エクスポートクライアントがセキュア Web オブジェクトまたはリソースにアクセスできないようにするには、エフェメラル RSA キー交換を無効にする必要があります。

デフォルトでは、この機能は NetScaler ADC アプライアンスで有効になっており、更新カウントはゼロ（無限使用）に設定されています。

注:

エクスポート暗号を SSL または TCP ベースの SSL 仮想サーバーまたはサービスにバインドすると、エフェメラル RSA キーが自動的に生成されます。エクスポート暗号を削除しても、ErSA キーは削除されません。後で別のエクスポート暗号が SSL または TCP ベースの SSL 仮想サーバーまたはサービスにバインドされるときに再利用されます。ErSA キーは、システムの再起動時に削除されます。

CLI を使用してエフェメラル **RSA** を構成する　コマンドプロンプトで次のコマンドを入力して、エフェメラル RSA を構成し、構成を確認します。


```
set ssl vserver <vServerName> -eRSA (enabled | disabled) -eRSACount <
   positive_integer>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

例:

```
set ssl vserver vs-server -eRSA ENABLED -eRSACount 1000
 Done

show ssl vserver vs-server

        Advanced SSL configuration for VServer vs-server:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
           ENABLED  TLSv1.2: ENABLED

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
<!--NeedCopy-->
```

GUI を使用してエフェメラル **RSA** を構成する

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、[エフェメラル **RSA** を有効にする] を選択し、更新回数を指定します。

セッション再利用

SSL トランザクションの場合、初期 SSL ハンドシェイクを確立するには、CPU を集中的に使用する公開キー暗号化操作が必要です。ほとんどのハンドシェイク操作は、SSL セッションキー (クライアントキー交換メッセージ) の交換に関連付けられています。クライアントセッションがしばらくアイドル状態になってから再開されると、通常、SSL ハンドシェイクが最初からやり直されます。セッション再利用を有効にすると、クライアントから受け取ったセッション再開要求に対するセッション鍵の交換が回避されます。

NetScaler アプライアンスでは、セッションの再利用がデフォルトで有効になっています。この機能を有効にすると、サーバーの負荷が軽減され、応答時間が短縮され、サーバーがサポートできる 1 秒あたりの SSL トランザクション (TPS) の数が増加します。

CLI を使用してセッションの再利用を設定する　コマンドプロンプトで次のコマンドを入力して、セッションの再利用を構成し、構成を確認します。

```
set ssl vserver <vServerName> -sessReuse ( ENABLED | DISABLED ) -
   sessTimeout <positive_integer>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

例**:**

```
set ssl vserver vs-ssl -sessreuse enabled -sesstimeout 600
 Done

show ssl vserver vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 600 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
           ENABLED  TLSv1.2: ENABLED

1)      CertKey Name: Auth-Cert-1       Server Certificate

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
<!--NeedCopy-->
```

GUI を使用してセッションの再利用を構成する

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、[セッション再利用を有効にする] を選択し、セッションをアクティブにしておく時間を指定します。

SSL プロトコル設定

NetScaler アプライアンスは、SSLv3、TLSv1、TLSv1.1、および TLSv1.2 プロトコルをサポートしています。これらの各プロトコルは、導入環境およびアプライアンスに接続するクライアントのタイプに応じて、アプライアンス上で設定できます。

TLS プロトコルバージョン 1.0、1.1、および 1.2 は、古いバージョンの TLS/SSL プロトコルよりも安全性が高くなっています。ただし、レガシーシステムをサポートするために、多くの TLS 実装では SSLv3 プロトコルとの下位互換性が維持されています。SSL ハンドシェイクでは、クライアントと NetScaler アプライアンス上で構成されている SSL 仮想サーバーに共通する最も高いプロトコルバージョンが使用されます。

最初のハンドシェイク試行では、TLS クライアントはサポートしている最も高いプロトコルバージョンを提供します。ハンドシェイクが失敗した場合、クライアントはより低いプロトコルバージョンを提供します。たとえば、TLS バージョン 1.1 のハンドシェイクが成功しなかった場合、クライアントは TLSv1.0 プロトコルを提供して再ネゴシエーションを試みます。この試行が失敗した場合、クライアントは SSLv3 プロトコルを使用して再試行します。「Man in the Middle」（MITM）攻撃者は、最初のハンドシェイクを破り、SSLv3 プロトコルとの再ネゴシエーションをトリガーし、SSLv3 の脆弱性を悪用する可能性があります。このような攻撃を軽減するには、SSLv3 を無効にするか、ダウングレードされたプロトコルを使用した再ネゴシエーションを許可しないようにします。ただし、導入にレガシーシステムが含まれている場合、この方法は実用的ではない可能性があります。別の方法として、クライアント要求内のシグナリング暗号スイート値 (TLS_FALLBACK_SCSV) を認識する方法があります。

クライアントの hello メッセージの TLS_FALLBACK_SCSV 値は、クライアントが以前に上位のプロトコルバージョンで接続しようとしたことがあり、現在の要求がフォールバックであることを仮想サーバーに示します。仮想サーバがこの値を検出し、クライアントが指定したバージョンよりも新しいバージョンをサポートしている場合、接続は拒否され、致命的な警告が表示されます。ハンドシェイクは、次のいずれかの条件が満たされた場合に成功します。

- TLS_FALLBACK_SCSV 値は、クライアントの hello メッセージには含まれません。
- クライアント hello のプロトコルバージョンは、仮想サーバでサポートされる最も高いプロトコルバージョンです。

CLI を使用して **SSL** プロトコルサポートを設定する　コマンドプロンプトで次のコマンドを入力して SSL プロトコルサポートを構成し、構成を確認します。

```
set ssl vserver <vServerName> -ssl2 ( ENABLED | DISABLED ) -ssl3 (
    ENABLED | DISABLED ) -tls1 ( ENABLED | DISABLED ) -tls11 ( ENABLED |
     DISABLED ) -tls12 ( ENABLED | DISABLED )

show ssl vserver <vServerName>
<!--NeedCopy-->
```

例**:**

```
set ssl vserver vs-ssl -tls11 ENABLED -tls12 ENABLED
Done

```



```
sh ssl vs vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
            DH: DISABLED
            Ephemeral RSA: ENABLED                              Refresh
               Count: 0
            Session Reuse: ENABLED                                Timeout
               : 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            SSLv2: DISABLED        SSLv3: ENABLED    TLSv1.0: ENABLED
               TLSv1.1: ENABLED  TLSv1.2: ENABLED
            Push Encryption Trigger: Always
            Send Close-Notify: YES
            1 bound certificate:

    1)      CertKey Name: mycert  Server Certificate
            1 configured cipher:

    1)      Cipher Name: DEFAULT
            Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

GUI を使用して **SSL** プロトコルサポートを構成する

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、有効にするプロトコルを選択します。

閉じる通知

クローズ通知は、SSL データ伝送の終了を示す安全なメッセージです。クローズ通知設定はグローバルレベルで必要です。この設定は、すべての仮想サーバ、サービス、およびサービスグループに適用されます。グローバル設定の詳細については、このページの「グローバル SSL パラメータ」を参照してください。

グローバル設定に加えて、close-notify パラメータを仮想サーバ、サービス、またはサービスグループレベルで設定できます。したがって、1 つのエンティティに対してパラメータを設定し、別のエンティティに対してはパラメータを設定解除できる柔軟性があります。ただし、このパラメータは必ずグローバルレベルで設定してください。そうしないと、エンティティレベルの設定は適用されません。

CLI を使用して、エンティティレベルでクローズ通知を設定する　コマンドプロンプトで、次のいずれかのコマンドを入力して close-notify 機能を構成し、構成を確認します。

1. 仮想サーバレベルで構成するには、以下のように入力します。

```
set ssl vserver <vServerName> -sendCloseNotify ( YES | NO )
show ssl vserver <vServerName>
<!--NeedCopy-->
```

1. をサービスレベルで設定するには、次のように入力します。

```
set ssl service <serviceName> -sendCloseNotify ( YES | NO )
show ssl service <serviceName>
<!--NeedCopy-->
```

1. サービスグループレベルで設定するには、次のように入力します。

```
set ssl serviceGroup <serviceGroupName> -sendCloseNotify ( YES | NO )
show ssl serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

例**:**

```
set ssl vserver sslvsvr -sendCloseNotify YES

Done
<!--NeedCopy-->
```

GUI を使用してエンティティレベルでクローズ通知機能を設定する

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、[クローズ通知を送信] を選択します。

グローバル **SSL** パラメータ

SSL 設定を高度にカスタマイズすることで、特定の問題に対処できます。`set ssl parameter` コマンドまたは構成ユーティリティを使用して、次の項目を指定できます。

- SSL トランザクションに使用される量子サイズ。
- CRL メモリサイズ。
- OCSP キャッシュサイズ。
- SSL 再ネゴシエーションを拒否する
- 復号化されたレコード、暗号化されたレコード、またはすべてのレコードに PUSH フラグを設定します。
- クライアントが 1 つのドメインに対してハンドシェイクを開始し、別のドメインに対して HTTP 要求を送信した場合は、要求をドロップします。

- 暗号化がトリガーされるまでの時間を設定します。
  注: 指定した時刻は、
  `set ssl vserver` コマンドまたは構成ユーティリティを使用してタイマーベースの暗号化を設定する場合にのみ適用されます。
- NDCPP 準拠証明書チェック—アプライアンスがクライアント (バックエンド接続) として動作する場合に適用されます。SSL 証明書に SAN が存在する場合は、証明書の検証時にコモンネームを無視します。
- MPX 14000 などの Cavium チップベースのアプライアンスの異種クラスタと、パケットエンジンの数が異なる MPX 15000 アプライアンスなどの Intel Coleto チップベースのアプライアンスを有効にします。(リリース 13.0 ビルド 47.x でサポートが追加されました)。
- バックエンドで安全な再ネゴシエーションを有効にします（リリース 1.0 ビルド 58.x からサポートが追加されました)。
- 適応型 SSL トラフィック制御（リリース 13.0 ビルド 58.x で追加されたサポート)。

CLI を使用してグローバル **SSL** パラメータを設定する

コマンドプロンプトで次のコマンドを入力して、SSL の詳細設定を構成し、構成を確認します。

```
set ssl parameter [-quantumSize <quantumSize>] [-crlMemorySizeMB <
    positive_integer>] [-strictCAChecks (YES | NO)] [-sslTriggerTimeout
    <positive_integer>] [-sendCloseNotify (YES | NO)] [-
    encryptTriggerPktCount <positive_integer>] [-denySSLReneg <
    denySSLReneg>] [-insertionEncoding (Unicode|UTF-8)] [-ocspCacheSize
    <positive_integer>][- pushFlag <positive_integer>] [-
    dropReqWithNoHostHeader (YES | NO)] [-pushEncTriggerTimeout <
    positive_integer>] [-ndcppComplianceCertCheck ( YES | NO)] [-
    heterogeneousSSLHW (ENABLED | DISABLED )]
show ssl parameter
<!--NeedCopy-->
```

例**:**

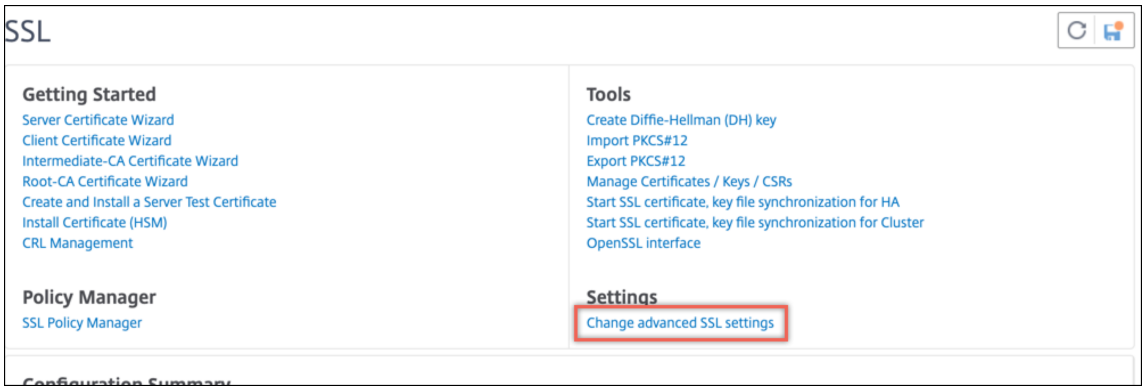
```
set ssl parameter -quantumSize 8 -crlMemorySizeMB 256 -strictCAChecks
    no -ssltriggerTimeout 100 -sendClosenotify no -
    encryptTriggerPktCount 45 -denySSLReneg NONSECURE -insertionEncoding
     unicode -ocspCacheSize 10 -pushFlag 3 -dropReqWithNoHostHeader YES
     -pushEncTriggerTimeout 100 ms -ndcppComplianceCertCheck YES
Done

show ssl parameter
Advanced SSL Parameters
-----------------------
    SSL quantum size                                    : 8 KB
    Max CRL memory size                                 : 256 MB
    Strict CA checks                                    : NO
    Encryption trigger timeout                          : 100 ms
    Send Close-Notify                                   : NO
    Encryption trigger packet count                     : 45
    Deny SSL Renegotiation                              : NONSECURE
```



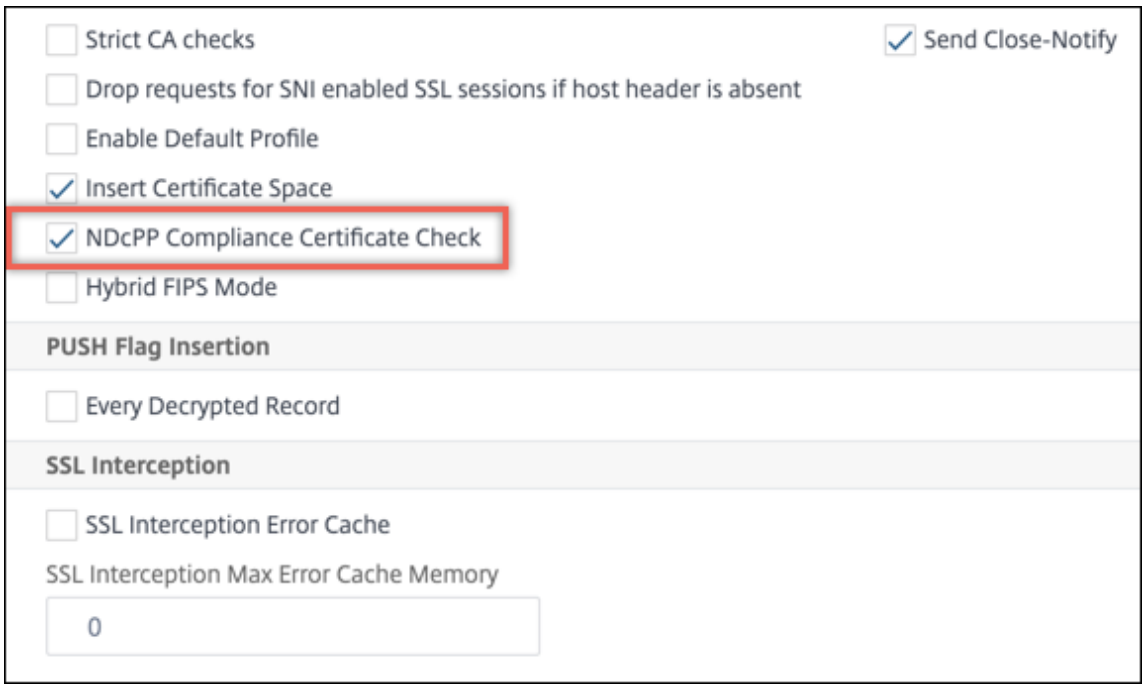
```
14      Subject/Issuer Name Insertion Format                    : Unicode
15      OCSP cache size                                         : 10 MB
16      Push flag                                               : 0x3 (On
           every decrypted and encrypted record)
17      Strict Host Header check for SNI enabled SSL sessions   : YES
18      PUSH encryption trigger timeout                         : 100 ms
19      Crypto Device Disable Limit                             : 0
20      Global undef action for control policies                : CLIENTAUTH
21      Global undef action for data policies                   : NOOP
22      Default profile                                         : DISABLED
23      SSL Insert Space in Certificate Header                  : YES
24      Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors      : NO
25      Disable TLS 1.1/1.2 for dynamic and VPN services        : NO
26      Software Crypto acceleration CPU Threshold              : 0
27      Hybrid FIPS Mode                                        : DISABLED
28      Signature and Hash Algorithms supported by TLS1.2       : ALL
29      SSL Interception Error Learning and Caching             : DISABLED
30      SSL Interception Maximum Error Cache Memory             : 0 Bytes
31      NDCPP Compliance Certificate Check                      : YES
32      Heterogeneous SSL HW (Cavium and Intel Based)           : ENABLED
33   Done
34  <!--NeedCopy-->
```

GUI を使用して **ndCPP** 準拠証明書チェックを構成する

1. [トラフィック管理**] > [SSL**] に移動し、[設定**]** グループで [**SSL** の詳細設定の変更] を選択します。



2. [**NDCPP** 準拠証明書チェック**]** を選択します。[**OK**] をクリックします。



NetScaler アプライアンスのバックエンドでの安全な再ネゴシエーションのサポート

注**:** この機能は、リリース 13.0 ビルド 58.x 以降でサポートされています。以前のリリースおよびビルドでは、非セキュアな再ネゴシエーションのみがバックエンドでサポートされていました。

この機能は、次のプラットフォームでサポートされています。
• VPX
• N2 または N3 チップを含む MPX プラットフォーム
• Intel Coleto SSL チップベースのプラットフォーム

この機能は FIPS プラットフォームではまだサポートされていません。

ADC アプライアンスのバックエンドでは、セキュアな再ネゴシエーションはデフォルトで拒否されます。つまり、denySSLReneg パラメータは ALL (デフォルト) に設定されます。

バックエンドでセキュアな再ネゴシエーションを許可するには、次のいずれかのdenySSLRenegパラメータ設定を選択します。

- いいえ
- FRONTEND_CLIENT
- FRONTEND_CLIENTSERVER
- NONSECURE

CLI を使用してセキュアな再ネゴシエーションを有効にする　コマンドプロンプトで入力します。

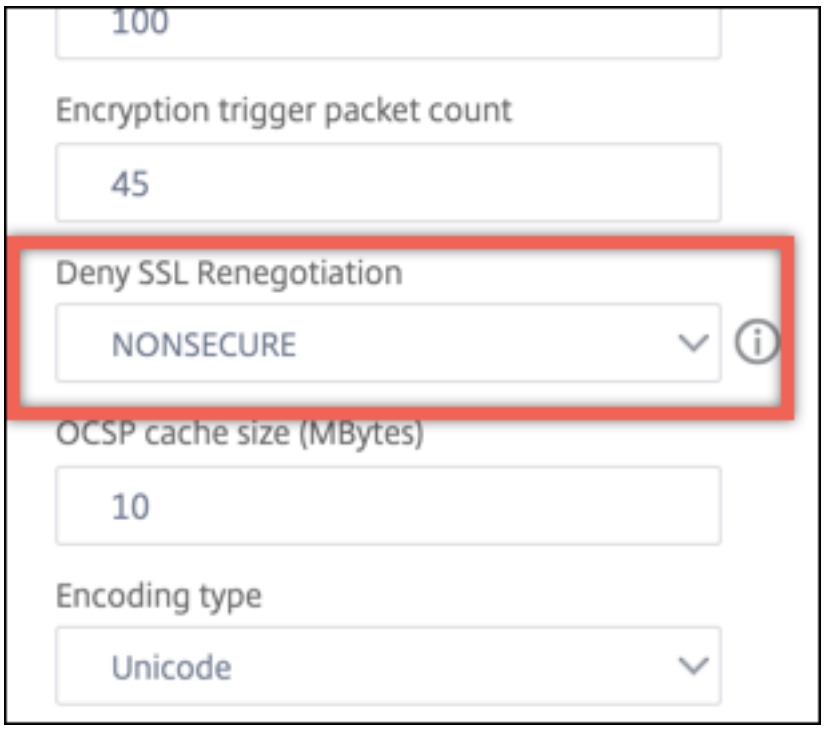
```
set ssl parameter -denySSLReneg <denySSLReneg>
```


例:

```
set ssl parameter -denySSLReneg NONSECURE
 Done

sh ssl parameter
Advanced SSL Parameters
-----------------------
    SSL quantum size                                    : 8 KB
    Max CRL memory size                                 : 256 MB
    Strict CA checks                                    : NO
    Encryption trigger timeout                          : 100 ms
    Send Close-Notify                                   : YES
    Encryption trigger packet count                     : 45
    Deny SSL Renegotiation                              : NONSECURE
    Subject/Issuer Name Insertion Format                : Unicode
    OCSP cache size                                     : 10 MB
    Push flag                                           : 0x0 (Auto)
    Strict Host Header check for SNI enabled SSL sessions : NO
    Match HTTP Host header with SNI                     : CERT
    PUSH encryption trigger timeout                     : 1 ms
    Crypto Device Disable Limit                         : 0
    Global undef action for control policies            : CLIENTAUTH
    Global undef action for data policies               : NOOP
    Default profile                                     : ENABLED
    SSL Insert Space in Certificate Header              : YES
    Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors  : NO
    Disable TLS 1.1/1.2 for dynamic and VPN services    : NO
    Software Crypto acceleration CPU Threshold          : 0
    Hybrid FIPS Mode                                    : DISABLED
    Signature and Hash Algorithms supported by TLS1.2   : ALL
    SSL Interception Error Learning and Caching         : DISABLED
    SSL Interception Maximum Error Cache Memory         : 0 Bytes
    NDCPP Compliance Certificate Check                  : NO
    Heterogeneous SSL HW (Cavium and Intel Based)       : DISABLED
    Crypto Operation Queue Limit                        : 150%
 Done
<!--NeedCopy-->
```

GUI を使用してセキュアな再ネゴシエーションを有効にする

1. [トラフィック管理] **> [SSL] > [SSL** の詳細設定の変更] に移動します。
2. [**SSL** 再ネゴシエーションの拒否] を ALL 以外の値に設定します。



アダプティブ **SSL** トラフィック制御

注**:** この機能は、リリース 13.0 ビルド 58.x 以降でサポートされています。

アプライアンスで大量のトラフィックが受信され、暗号アクセラレーションキャパシティがいっぱいになると、アプライアンスは後で処理するために接続のキューイングを開始します。現在、このキューのサイズは 64 K に固定されており、この値を超えるとアプライアンスは接続のドロップを開始します。

リリース 13.0 ビルド 58.x から、ユーザーは実際のキャパシティに対するパーセンテージの値を設定できます。この機能強化により、キュー内の要素の数が、適応的かつ動的に計算された制限を超えると、アプライアンスは新しい接続をドロップします。このアプローチは、着信 SSL 接続を制御し、アプライアンスでの過剰なリソース消費やその他の障害（負荷分散モニタリングの障害やセキュアなアプリケーションへの応答の遅延など）を防止します。

キューが空の場合、アプライアンスは引き続き接続を受け入れることができます。キューが空でない場合、暗号システムは容量に達しており、アプライアンスは接続のキューイングを開始します。

制限は次に基づいて計算されます。

- アプライアンスの実際の容量。
- 実際のキャパシティに対するパーセンテージとしてユーザーが設定した値。デフォルト値は 150% に設定されています。

たとえば、アプライアンスの実際のキャパシティが 1 秒あたり 1000 オペレーションで、デフォルトのパーセンテージが設定されている場合、アプライアンスが接続を切断するまでの制限は 1500（1000 の 150%）です。

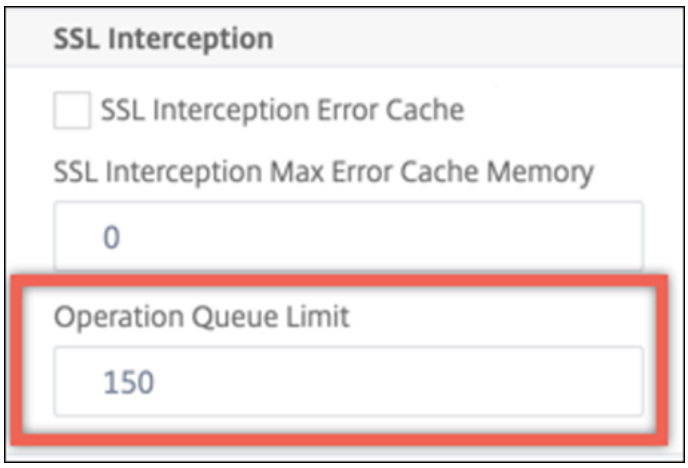
CLI を使用してオペレーションキュー制限を設定するには　コマンドプロンプトで入力します。

```
set ssl parameter -operationQueueLimit <positive_integer>
```

Operation Queue Limit：暗号化操作キューの容量に対するパーセンテージの制限。この値を超えると、キューが削減されるまで新しい SSL 接続が受け入れられません。デフォルト値:150。最小値:0。最大値は 10000 です。

GUI を使用して操作キュー制限を設定するには

1. **Traffic Management > SSL** に移動します。
2. [設定] で、[**SSL** の詳細設定の変更] をクリックします。
3. [操作キューの制限] に値を入力します。デフォルトは 150 です。
4. [**OK**] をクリックします。



異機種混在クラスタの展開

リリース 13.0 ビルド 47.x から、SSL パラメータ「異機種間 SSL HW」を「ENABLED」に設定することで、異なる数のパケットエンジンで NetScaler ADC MPX アプライアンスの異機種混在クラスタ展開を形成できます。たとえば、Cavium チップベースのアプライアンス（MPX 14000 または同様のもの）と Intel Coleto チップベースのアプライアンス（MPX 15000 または同様のもの）のクラスタを形成するには、SSL パラメータ「異種 SSL HW」を有効にします。同じチップを使用してプラットフォームのクラスタを形成するには、このパラメータのデフォルト値（DISABLED）を維持します。

> 注：
>
> 異機種混在クラスタでは、次の機能はサポートされていません。
>
> - NetScaler SDX アプライアンスでホストされている VPX インスタンス。
> - 仮想サーバー、サービス、サービスグループ、内部サービスなどの SSL エンティティに対する SSLv3 プロトコル。

> - ソフトウェア暗号加速 CPU しきい値（ハードウェアとソフトウェアを使用して ECDSA および ECDHE 暗号のパフォーマンスを向上させます）。

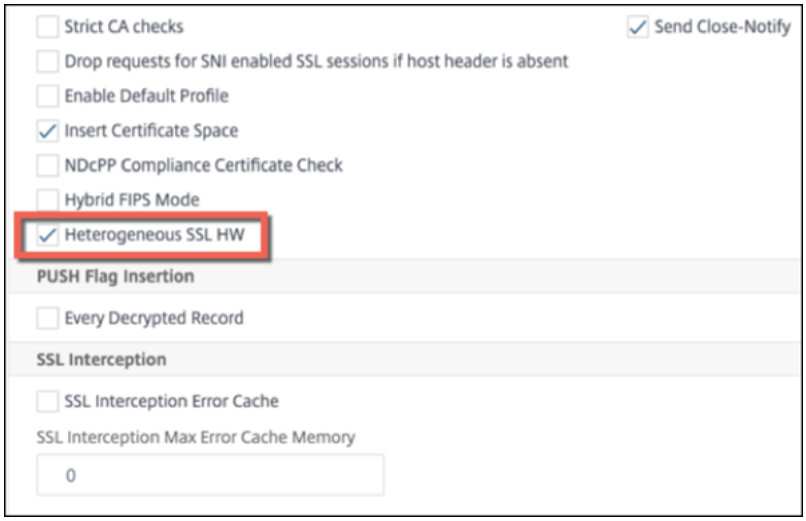
異機種混在クラスタでサポートされるプラットフォームの詳細については、「https://docs.citrix.com/en-us/citrix-adc/13-1/clustering/support-for-heterogeneous-cluster.html」を参照してください。

CLI を使用した異機種クラスターの有効化　コマンドプロンプトで入力します。

```
set ssl parameter -heterogeneousSSLHW ENABLED
```

GUI を使用した異機種混在クラスタの有効化

1. [トラフィック管理] **> [SSL**] に移動し、[設定] グループで [**SSL** の詳細設定の変更] を選択します。
2. [異機種混在 **SSL HW**] を選択します。[**OK**] をクリックします。



PUSH フラグベースの暗号化トリガーメカニズム

PSH TCP フラグに基づく暗号化トリガメカニズムにより、次のことが可能になりました。

- PSH フラグが設定された連続したパケットを 1 つの SSL レコードにマージするか、PSH フラグを無視します。
- `set ssl parameter -pushEncTriggerTimeout <positive_integer>` コマンドを使用してタイムアウト値をグローバルに設定する、タイマーベースの暗号化を実行します。

CLI を使用して **PUSH** フラグベースの暗号化を設定する　コマンドプロンプトで次のコマンドを入力して PUSH フラグベースの暗号化を構成し、構成を確認します。


```
set ssl vserver <vServerName> [-pushEncTrigger <pushEncTrigger>]

show ssl vserver
<!--NeedCopy-->
```

例:

```
set ssl vserver vserver1 -pushEncTrigger always

Done

sh ssl vserver vserver1

            Advanced SSL configuration for VServer vserver1:
            DH: DISABLED
            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                RSA: ENABLED

                Refresh Count: 0
            Session Reuse: ENABLED          Timeout: 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            HSTS: DISABLED
            HSTS IncludeSubDomains: NO
            HSTS Max-Age: 0
            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
            Push Encryption Trigger: Always
            Send Close-Notify: YES
            Strict Sig-Digest Check: DISABLED
            Zero RTT Early Data: DISABLED
            DHE Key Exchange With PSK: NO
            Tickets Per Authentication Context: 1
            ECC Curve: P_256, P_384, P_224, P_521

    1)      Cipher Name: DEFAULT
            Description: Default cipher list with encryption strength
                >= 128bit
Done
<!--NeedCopy-->
```

GUI を使用して **PUSH** フラグベースの暗号化を構成する

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、SSL 仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションの [**PUSH** 暗号化トリガー] リストから値を選択します。

TLS1.2 署名ハッシュアルゴリズムのサポート

NetScaler ADC アプライアンスは、TLS1.2 署名ハッシュ拡張に完全に準拠しています。

SSL ハンドシェイクでは、クライアントは、サポートされている署名ハッシュアルゴリズムのリストを送信します。クライアントは、「signature_algorithm」拡張を使用して、SSL ハンドシェイクメッセージ (SKE と CCV) でどのシグニチャハッシュアルゴリズムペアを使用できるかをサーバに指示します。この拡張モジュールの「拡張データ」フィールドには、クライアントの Hello メッセージに「サポートされた署名のアルゴリズム」値が含まれています。SSL ハンドシェイクは、サーバがこれらのシグニチャハッシュアルゴリズムのいずれかをサポートしている場合に処理されます。サーバーがこれらのアルゴリズムをサポートしていない場合、接続は切断されます。

同様に、サーバーがクライアント認証のためにクライアント証明書を要求した場合、証明書要求メッセージには「supported_signature_algorithms」の値が含まれます。クライアント証明書は、この署名ハッシュアルゴリズムに基づいて選択されます。

> 注:
>
> NetScaler ADC アプライアンスは、クライアントに対してはサーバーとして、バックエンドサーバーに対してはクライアントとして機能します。

このアプライアンスは、フロントエンドでは RSA-SHA1 と RSA-SHA256 のみをサポートし、バックエンドでは RSA-MD5、RSA-SHA1、RSA-SHA256 のみをサポートします。

MPX/SDX/VPX アプライアンスは、次の署名ハッシュの組み合わせをサポートしています。SDX アプライアンスでは、SSL チップが VPX インスタンスに割り当てられている場合、MPX アプライアンスの暗号サポートが適用されます。それ以外の場合は、VPX インスタンスの通常の暗号サポートが適用されます。

- VPX インスタンスおよび N3 チップのない MPX/SDX アプライアンスでは、次のようになります。
  - RSA-MD5
  - RSA-SHA1
  - RSA-SHA224
  - RSA-SHA256
  - RSA-SHA384
  - RSA-SHA512
- N3 チップを搭載した MPX/SDX アプライアンスでは、次の手順を実行します。
  - RSA-MD5
  - RSA-SHA1
  - RSA-SHA224
  - RSA-SHA256
  - RSA-SHA384
  - RSA-SHA512
  - ECDSA-SHA1

- ECDSA-SHA224
- ECDSA-SHA256
- ECDSA-SHA384
- ECDSA-SHA512

デフォルトでは、すべてのシグニチャハッシュアルゴリズムが有効になっています。ただし、次のコマンドを使用して有効にできるシグニチャハッシュアルゴリズムはごくわずかです。

```
set ssl parameter -sigDigestType <sigDigestType>

Parameters

sigDigestType

Signature digest algorithms supported by the appliance. The platform
   determines the list of algorithms supported by default.

              On VPX: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384
                   RSA-

              SHA512

              On MPX with N3 cards: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-

              SHA256 RSA-SHA384 RSA-SHA512 ECDSA-SHA1 ECDSA-SHA224
                  ECDSA-

              SHA256 ECDSA-SHA384 ECDSA-SHA512

              Other MPX Platforms: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-
                  SHA256 RSA-SHA384 RSA-

              SHA512.

    set ssl parameter -sigDigestType RSA-SHA224 RSA-SHA256 RSA-SHA384
       RSA-SHA512
<!--NeedCopy-->
```

ピア証明書を検証する　RFC 5246 によると、ピア証明書は Client Hello 拡張に含まれる署名ハッシュアルゴリズムのいずれかを使用して署名される必要があります。strictSigDigestCheck パラメータを使用できます。クライアントから送信されたシグニチャハッシュリストに応じて、strictSigDigestCheckを有効にすると、アプライアンスは Client Hello 拡張機能に記載されているシグニチャハッシュアルゴリズムのいずれかによって署名された証明書を返します。ピアに適切な証明書がない場合、接続は切断されます。このパラメータを無効にすると、ピア証明書でシグニチャハッシュはチェックされません。

SSL 仮想サーバおよびサービスでは、厳密なシグニチャダイジェストチェックを設定できます。SSL 仮想サーバーでこのパラメーターを有効にする場合、サーバーから送信されるサーバー証明書は、Client Hello 拡張機能にリストされている署名ハッシュアルゴリズムのいずれかによって署名されている必要があります。クライアント認証が有効な

場合、サーバーが受信したクライアント証明書は、サーバーから送信された証明書要求にリストされている署名ハッシュアルゴリズムのいずれかを使用して署名されている必要があります。

SSL サービスでこのパラメーターを有効にする場合、クライアントが受信するサーバー証明書は、Client Hello 拡張機能にリストされている署名ハッシュアルゴリズムのいずれかによって署名されている必要があります。クライアント証明書は、証明書要求メッセージに記載されている署名ハッシュアルゴリズムのいずれかを使用して署名されている必要があります。

デフォルトプロファイルが有効な場合は、このプロファイルを使用して、SSL 仮想サーバ、SSL サービス、および SSL プロファイルに対して厳密なシグニチャダイジェストチェックを設定できます。

CLI を使用して **SSL** 仮想サーバ、サービス、またはプロファイルに厳密なシグニチャダイジェストチェックを設定する　コマンドプロンプトで入力します。

```
set ssl vserver <vServerName> -strictSigDigestCheck ( ENABLED |
   DISABLED )

set ssl service <serviceName> -strictSigDigestCheck ( ENABLED |
   DISABLED )

set ssl profile <name>-strictSigDigestCheck ( ENABLED | DISABLED )

Parameters

strictSigDigestCheck

              Check whether peer entity certificate is signed using one
                  of the signature-hash algorithms supported by the
                 NetScaler appliance.

              Possible values: ENABLED, DISABLED

              Default: DISABLED
<!--NeedCopy-->
```

例:

```
set ssl vserver v1  -strictSigDigestCheck Enabled
set ssl service s1  -strictSigDigestCheck Enabled
set ssl profile p1  -strictSigDigestCheck Enabled
<!--NeedCopy-->
```

> 重要:
>
> DH、ECDHE、または ECDSA 暗号がアプライアンスに設定されている場合、SKE メッセージは、クライアントリストとアプライアンスに設定されたリストに共通するシグニチャハッシュの 1 つを使用して署名する必要があります。共通の署名ハッシュがない場合、接続は切断されます。

ADC 管理者 UI アクセス用に SSL を構成する

構成ユーティリティへの HTTPS アクセスおよびセキュアなリモートプロシージャコールには、証明書とキーのペアが必要です。NetScaler MPX アプライアンスまたは VPX 仮想アプライアンスでは、証明書とキーのペアが自動的に内部サービスにバインドされます。ただし、この証明書はブラウザによって信頼されていない可能性があります。エラーなしで認証を完了するには、ブラウザに有効な CA 証明書をアップロードする必要があります。

CLI を使用してセキュアな HTTPS の構成

CLI を使用してセキュア HTTPS を設定するには、次の手順に従います。

1. 証明書とキーのペアを追加します。

```
add certkey server -cert servercert -key serverkey
<!--NeedCopy-->
```

2. この証明書とキーのペアを次の内部サービスにバインドします。

```
bind ssl service nshttps-127.0.0.1-443 -certkeyname server

bind ssl service nshttps-::11-443 -certkeyname server
<!--NeedCopy-->
```

GUI を使用して安全な HTTPS を構成する

GUI を使用して安全な HTTPS を構成するには、次の手順に従います。

1. [**Traffic Management**] **>** [**SSL**] **>** [**Certificates**] に移動します。
2. 詳細ペインで、[インストール] をクリックします。
3. [証明書のインストール] ダイアログボックスで、証明書の詳細を入力します。
4. [インストール] をクリックし、[閉じる] をクリックします。
5. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
6. 詳細ウィンドウの [操作] タブで、[内部サービス] をクリックします。
7. リストからnshttps-127.0.0.1-443を選択し、[開く] をクリックします。
8. [**SSL** 設定] タブの [使用可能] ペインで、手順 4 で作成した証明書を選択し、[バインド] をクリックして、[**OK**] をクリックします。
9. リストからnshttps-::11-443を選択し、[開く] をクリックします。
10. [**SSL** 設定] タブの [使用可能] ペインで、手順 4 で作成した証明書を選択し、[バインド] をクリックして、[**OK**] をクリックします。
11. [**OK**] をクリックします。

RFC 8446 で定義されている TLSv1.3 プロトコルのサポート

August 15, 2023

NetScaler VPX および NetScaler ADC MPX アプライアンスは、RFC 8446 で指定されている TLSv1.3 プロトコルをサポートするようになりました。

> 注:
>
> - リリース 13.0 ビルド 71.x 以降では、TLS1.3 ハードウェアアクセラレーションが次のプラットフォームでサポートされています。
>   - MPX 5900
>   - MPX/SDX 8900
>   - MPX/SDX 9100
>   - MPX/SDX 15000
>   - MPX/SDX 15000-50G
>   - MPX/SDX 16000
>   - MPX/SDX 26000
>   - MPX/SDX 26000-50S
>   - MPX/SDX 26000-100G
> - TLSv1.3 プロトコルのソフトウェアのみのサポートは、NetScaler FIPS アプライアンスを除く他のすべての NetScaler ADC MPX および SDX アプライアンスで利用できます。
> - TLSv1.3 は、拡張プロファイルでのみサポートされます。拡張プロファイルを有効にするには、「デフォルトプロファイルを有効にする」を参照してください。
> - TLS1.3 を使用するには、RFC 8446 仕様に準拠したクライアントを使用する必要があります。

サポートされている NetScaler ADC 機能

次の SSL 機能がサポートされています。

1. TLSv1.3 暗号スイート:
   - TLS1.3-AES256-GCM-SHA384 (0x1302)
   - TLS1.3_CHACHA20_POLY1305_SHA256 (0x1303)
   - TLS1.3-AES128_GCM-SHA256 (0x1301)
2. エフェメラルな Diffie-Hellman キー交換のための ECC カーブ:
   - P_256
   - P_384

- P_521

3. チケットベースのセッション再開が有効な場合のハンドシェイクの簡略化
4. 0-RTT アーリーアプリケーションデータ
5. クライアント証明書の OCSP および CRL 検証をサポートする、オプションまたは必須の証明書ベースのクライアント認証
6. サーバー名拡張:SNI を使用したサーバー証明書の選択
7. application_level_protocol_negotiation 拡張機能を使用した、アプリケーションプロトコルネゴシエーション (ALPN)。
8. OCSP ステープリング
9. TLSv1.3 ハンドシェイク用のログメッセージと AppFlow レコードが生成されます。
10. `nstrace` パケットキャプチャユーティリティによる TLS 1.3 トラフィックシークレットのロギング（オプション）。
11. RFC 8446 を実装する TLS クライアントとの相互運用性。たとえば、Mozilla Firefox、Google Chrome、OpenSSL などです。

サポートされているブラウザー

次のブラウザバージョンがサポートされ、TLS 1.3 プロトコルの NetScaler ADC 実装と互換性があります。

- Google Chrome-バージョン 72.0.3626.121 (公式ビルド) (64 ビット)
- Mozilla Firefox-65.0.2 (64 ビット)
- Opera -バージョン:58.0.3135.79

構成

TLSv1.3 は SSL プロファイルではデフォルトで無効になっています。

CLI を使用して **SSL** プロファイルを追加する

コマンドプロンプトで入力します。

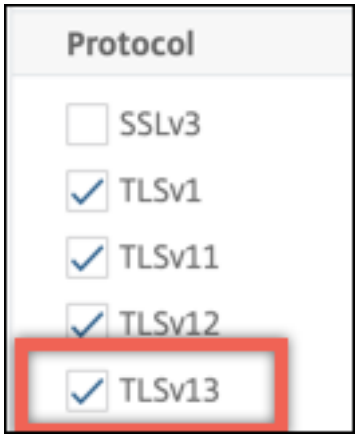
```
add ssl profile <tls13-profile-name>
<!--NeedCopy-->
```

例:


```
add ssl profile tls13profile

sh ssl profile tls13profile
1)  Name: tls13profile           (Front-End)
    SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1.1: ENABLED
       TLSv1.2: ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:   NO
    Session Reuse: ENABLED                        Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED   Ephemeral RSA:
       ENABLED                                Refresh Count: 0
    Deny SSL Renegotiation                                  ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:              1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions: NO
    Push flag:            0x0 (Auto)
    SSL quantum size:                             8 kB
    Encryption trigger timeout           100 mS
    Encryption trigger packet count:                45
    Subject/Issuer Name Insertion Format: Unicode

    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server:  10
    Session Ticket: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0

    ECC Curve: P_256, P_384, P_224, P_521

1)  Cipher Name: DEFAULT Priority :1
    Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```


GUI を使用して **SSL** プロファイルを追加する

1. [システム**] > [**プロファイル] に移動します。[**SSL** プロファイル] を選択します。
2. [追加] をクリックし、プロファイルの名前を指定します。
3. [プロトコル] で **TLSv13** を選択します。



4. [**OK**] をクリックします。

CLI を使用して **SSL** プロファイルを **SSL** 仮想サーバにバインドする

コマンドプロンプトで入力します。

```
set ssl vserver <vServerName> -sslProfile <tls13-profile-name>
<!--NeedCopy-->
```

例:

```
set ssl vserver ssl-vs -sslProfile tls13profile
```

GUI を使用して **SSL** プロファイルを **SSL** 仮想サーバにバインドする

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、SSL 仮想サーバーを選択します。
2. [詳細設定**]** で [**SSL** プロファイル] をクリックします。
3. 以前に作成した TLSv1.3 プロファイルを選択します。
4. [**OK**] をクリックします。
5. [完了] をクリックします。

TLSv1.3 プロトコルの **SSL** プロファイルパラメータ

1. SSL プロファイルで TLS1.3 パラメータを有効または無効にします。

tls13: SSL プロファイルに対する TLSv1.3 プロトコルサポートの状態。

設定可能な値: ENABLED, DISABLED

デフォルト値: 無効

```
set ssl profile tls13profile -tls13 enable
<!--NeedCopy-->
```

```
set ssl profile tls13profile -tls13 disable
<!--NeedCopy-->
```

2. 発行されたセッションチケットの数を設定します。

TLS13SessionTicketsPerAuthContext: TLS1.3 がネゴシエートされ、チケットベースの再開が有効になっていて、(1) ハンドシェイクが完了した、または (2) ハンドシェイク後にクライアント認証が完了したときに、SSL 仮想サーバーが発行するチケットの数。

この値を増やすと、クライアントは接続ごとに新しいチケットを使用して複数のパラレル接続を開くことができます。

再開が無効になっている場合、チケットは送信されません。

デフォルト値:1

最小値:1

最大値:10

```
set ssl profile tls13profile -tls13sessionTicketsPerAuthContext 1

set ssl profile tls13profile -tls13sessionTicketsPerAuthContext 10
<!--NeedCopy-->
```

3. DH キー交換の設定

dheKeyExchangeWithPsk: TLS 1.3 セッション再開ハンドシェイク中に事前共有キーが受け入れられたときに、SSL 仮想サーバで DHE キー交換が必要かどうかを指定します。DHE キー交換は、チケットキーが侵害された場合でも、**DHE** キー交換の実行に必要な追加リソースを犠牲にして、前方秘匿性を確保します。

セッションチケットが有効な場合、使用可能な設定は次のように機能します。

はい:DHE キー交換は、クライアントがキー交換をサポートしているかどうかにかかわらず、事前共有キーが受け入れられる場合に必要です。事前共有キーを提供するときにクライアントが DHE キー交換をサポートしていない場合、ハンドシェイクは致命的なアラートで中止されます。

NO: DHE キー交換は、事前共有キーが受け入れられたときに、クライアントから要求された場合にのみ実行されます。

可能な値: はい、いいえ

デフォルト値:NO


```
set ssl profile tls13profile dheKeyExchangeWithPsk yes

set ssl profile tls13profile dheKeyExchangeWithPsk no
<!--NeedCopy-->
```

4. 0-RTT アーリーデータ受け入れの有効化または無効化

   zeroRttEarlyData: TLS 1.3 の初期アプリケーションデータの状態。適用可能な設定は次のように機能します。

   ENABLED: ハンドシェイクが完了する前に早期アプリケーションデータが処理されることがあります。
   DISABLED: 初期のアプリケーションデータは無視されます。

   設定可能な値: ENABLED, DISABLED

   デフォルト値: 無効

```
set ssl profile tls13profile -zeroRttEarlyData ENABLED

set ssl profile tls13profile -zeroRttEarlyData DISABLED
<!--NeedCopy-->
```

既定の暗号グループ

デフォルトの暗号グループには TLS1.3 暗号が含まれています。

```
sh cipher DEFAULT
1) Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035

2)  Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
...
...
27) Cipher Name: TLS1.3-AES256-GCM-SHA384         Priority : 27
    Description: TLSv1.3 Kx=any      Au=any  Enc=AES-GCM(256) Mac=AEAD
         HexCode=0x1302

28) Cipher Name: TLS1.3_CHACHA20_POLY1305_SHA256     Priority : 28
    Description: TLSv1.3 Kx=any      Au=any  Enc=CHACHA20/POLY1305(256)
        Mac=AEAD   HexCode=0x1303

29) Cipher Name: TLS1.3-AES128_GCM-SHA256        Priority : 29
    Description: TLSv1.3 Kx=any      Au=any  Enc=AES-GCM(128) Mac=AEAD
         HexCode=0x1301
Done
<!--NeedCopy-->
```


制限事項

- TLSv1.3 はバックエンドではサポートされていません。
- TLSv1.3 は、Citrix Secure Web Gateway アプライアンスおよび NetScaler ADC FIPS アプライアンスではサポートされていません。
- TLSv1.3 ハンドシェイクでは、1024 ビット以上のキーを含む RSA 証明書のみがサポートされます。

セキュリティ制限

TLSv1.3 サーバオペレータは、RFC 8446 で概説されている下位互換性のために、次のセキュリティ制限に留意する必要があります。NetScaler アプライアンスのデフォルト構成は、これらの制限に準拠しています。ただし、NetScaler アプライアンスでは、これらのルールの遵守は強制されません。

- RC4 暗号スイートのセキュリティは、RFC7465 で説明されているように不十分であると考えられています。実装では、TLS のどのバージョンでも RC4 暗号スイートを提供したり、ネゴシエートしたりしてはなりません。
- 古いバージョンの TLS では、低強度の暗号を使用できました。強度が 112 ビット未満の暗号は、どのバージョンの TLS でも提供またはネゴシエートしてはなりません。
- SSL 3.0 [SSLv3] のセキュリティは、RFC7568 で説明されているように不十分であると見なされ、ネゴシエートしてはなりません。TLSv1.3 が有効になっている場合は、SSLv3 を無効にします（SSLv3 はデフォルトで無効になっています）。
- SSL 2.0 [SSLv2] のセキュリティは、RFC6176 で説明されているように不十分であると見なされ、ネゴシエートしてはなりません。TLS 1.3 が有効な場合は、SSLv2 を無効にします (SSLv2 はデフォルトで無効になっています)。

> 注:
>
> TLS1.3 で実行されるプロトコルのトラブルシューティングについては、パケットトレースからの TLS1.3 トラフィックの復号化を参照してください。

ハウツー記事

August 15, 2023

ハウツー記事はシンプルで使いやすい記事で、一般的な導入環境の設定手順が記載されています。リンクをクリックすると記事が表示されます。

NetScaler ADC アプライアンスで証明書署名リクエストを作成して SSL 証明書を使用します

クライアントトラフィックを転送するように SSL アクションを設定

ADC で暗号がサポートされていない場合にクライアントトラフィックを転送するように SSL アクションを設定します

ディレクトリごとのクライアント認証の設定

Outlook Web Access に対するサポートの構成

SSL ベースのヘッダー挿入の設定

エンドツーエンド暗号化による SSL オフロードの設定

トランスペアレント SSL アクセラレーションの設定

フロントエンドに HTTP、バックエンドに SSL を使用して SSL アクセラレーションを設定します。

他の TCP プロトコルで SSL オフロードを設定

SSL ブリッジの設定

バックエンドサービスでクライアント認証が有効になっている場合の SSL モニタリングの設定

安全なコンテンツスイッチングサーバーの設定

HTTP トラフィックを受け入れるように HTTPS 仮想サーバーを構成する

SSL セッションのグレースフルクリーンアップの設定

HTTP 厳密な転送セキュリティ (HSTS) のサポートを構成する

SSLv2 リダイレクトの設定

高可用性セットアップでのファイル同期の設定

NSIP で TLS 1.0 と TLS 1.1 を無効化

NetScaler アプライアンスで使用されている証明書を PFX ファイルとしてエクスポートする

SSL 証明書

February 15, 2024

SSL 証明書は、SSL トランザクションの一部であり、企業（ドメイン）または個人を識別するデジタルデータフォーム（X509）です。この証明書には、サーバーとの安全なトランザクションを開始しようとするすべてのクライアントが確認できる公開キーコンポーネントが含まれます。対応する秘密キーは、NetScaler ADC アプライアンスに安全に存在し、非対称キー（または公開キー）の暗号化と復号化を完了するために使用されます。

SSL 証明書およびキーは、次のいずれかの方法で入手できます。

- Verisign などの承認された認証局 (CA) から
- NetScaler アプライアンス上で新しい SSL 証明書とキーを生成する

または、アプライアンスで既存の SSL 証明書を使用することもできます。

証明書は、NetScaler ADC アプライアンスによって次の 4 つのタイプに分類されます。

- サーバー証明書**:** サーバー証明書は、サーバーの ID をクライアントに対して認証します。フロントエンドでは、ADC アプライアンスはサーバーとして機能します。サーバー証明書と秘密鍵を ADC アプライアンスの SSL 仮想サーバーにバインドします。
- クライアント証明書**:** クライアント証明書は、クライアントの ID をサーバーに対して認証します。バックエンドでは、ADC アプライアンスはクライアントとして機能します。クライアント証明書と秘密キーを ADC アプライアンスの SSL サービスまたはサービスグループにバインドします。
- **CA** 証明書**:** CA 証明書は、エンドユーザー証明書 (クライアント証明書とサーバー証明書) を発行します。CA 証明書は、信頼されたルート CA（認証局によって自己署名）または中間 CA（信頼されたルート CA によって署名された）にすることができます。通常、CA 証明書には秘密キーは必要ありません。
- 不明な証明書**:** 他のすべての証明書はこのカテゴリに分類されます。

重要：すべての SSL トランザクションには、Verisign などの承認された CA から取得した証明書を使用することをお勧めします。NetScaler ADC アプライアンスで生成された証明書は、テスト目的でのみ使用し、ライブ展開では使用しないでください。

- 証明書とキーのペアの追加時に、既存の証明書ファイルと同じ名前の証明書ファイルを追加すると、元の証明書ファイルは警告なしで上書きされます。この操作により、`/nsconfig/ssl` 元の証明書ファイルがディレクトリで使用できなくなったため、アプライアンスの再起動後に問題が発生する可能性があります。
- クラスタ環境で証明書またはキーファイルを削除すると、ADC アプライアンスのさらなる設定が制限されます。構成を変更する場合は、ファイルを同じ場所に戻します。

注： SSL 証明書の管理を容易にするために ADM SSL ダッシュボードを使用し、未使用または間もなく期限切れになる証明書の通知を設定できます。詳細については、「SSL 証明書管理」を参照してください。

証明書を作成する

August 15, 2023

認証局 (CA) は、公開キー暗号化で使用するデジタル証明書を発行するエンティティです。SSL トランザクションを実行するアプリケーション（Web ブラウザなど）は、認証局が発行または署名した証明書を信頼します。これらのアプリケーションでは、信頼する CA のリストが保持されます。信頼できる CA のいずれかが、セキュアなトランザクションに使用されている証明書に署名すると、アプリケーションはトランザクションを続行します。

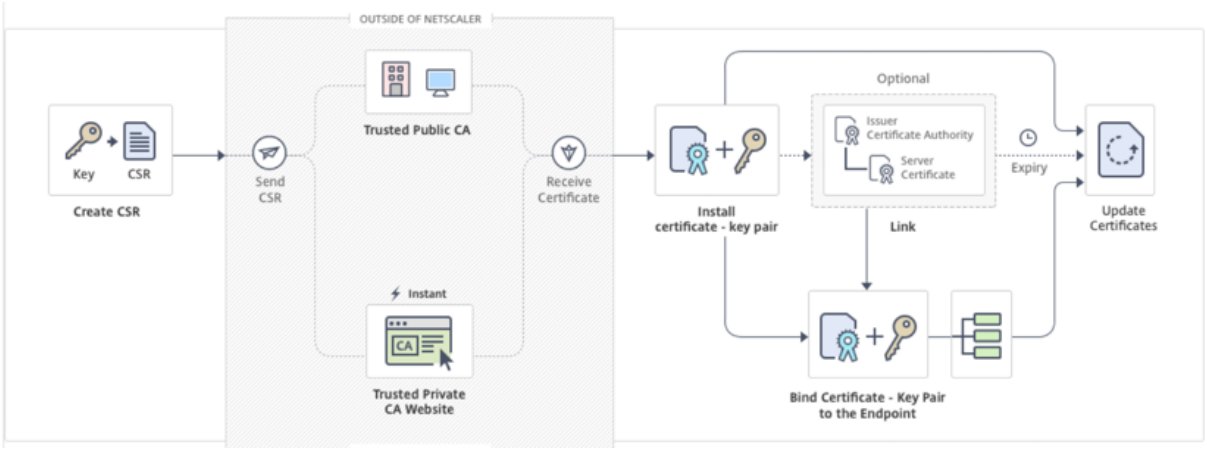
注意：すべての SSL トランザクションには、Verisign などの承認された CA から取得した証明書を使用することをお勧めします。NetScaler ADC アプライアンスで生成された証明書は、テスト目的でのみ使用し、ライブ展開では使用しないでください。

既存の証明書とキーをインポートするには 、証明書のインポートを参照してください。

証明書を作成し、SSL 仮想サーバーにバインドするには、次の手順を実行します。ファイル名に使用できる特殊文字は、アンダースコアとドットだけです。ファイル名の最初の文字には特殊文字を使用できません。

- プライベートキーを作成します。
- 証明書署名要求 (CSR) を作成します。
- CSR を認証局に送信します。
- 証明書とキーのペアを作成します。
- 証明書とキーのペアを SSL 仮想サーバーにバインドする

次の図は、このワークフローを示しています。



新しい証明書を作成してインストールする方法

これは埋め込みビデオです。リンクをクリックしてビデオを見る

秘密キーの作成

> メモ:
>
> - リリース 12.1 ビルド 49.x から、PEM キー形式の AES256 アルゴリズムを使用して、アプライアンスの秘密キーを暗号化できます。256 ビットキーの AES は、データ暗号化規格 (DES) の 56 ビットキーに比べて、数学的に効率的で安全です。
> - リリース 12.1 ビルド 50.x から、PKCS #8 形式で RSA キーを作成できます。

秘密キーはデジタル証明書で最も重要な部分です。定義上、このキーは誰とも共有されるべきではなく、NetScaler ADC アプライアンス上で安全に保管する必要があります。公開鍵で暗号化されたデータは、秘密鍵を使用することによってのみ復号化できます。

CA から受け取った証明書は、CSR の作成に使用された秘密キーでのみ有効です。このキーは、証明書を NetScaler ADC アプライアンスに追加するために必要です。

アプライアンスは、秘密キーの作成に RSA 暗号化アルゴリズムのみをサポートします。どちらのタイプの秘密キーも認証局 (CA) に送信できます。CA から受け取った証明書は、CSR の作成に使用された秘密キーでのみ有効です。このキーは、証明書を NetScaler ADC アプライアンスに追加するために必要です。

重要:

> - 秘密鍵へのアクセスは必ず制限してください。プライベートキーにアクセスできる人なら誰でも、SSL データを復号できます。
> - 許可される SSL キー名の長さには、パスがキー名に含まれている場合、絶対パス名の長さも含まれます。

SSL 証明書とキーはすべて、アプライアンス上の/nsconfig/sslフォルダに保存されます。セキュリティを強化するために、DES アルゴリズムまたはトリプル DES（3DES）アルゴリズムを使用して、アプライアンスに保存されている秘密キーを暗号化できます。

CLI を使用して **RSA** 秘密キーを作成する

コマンドプロンプトで入力します。

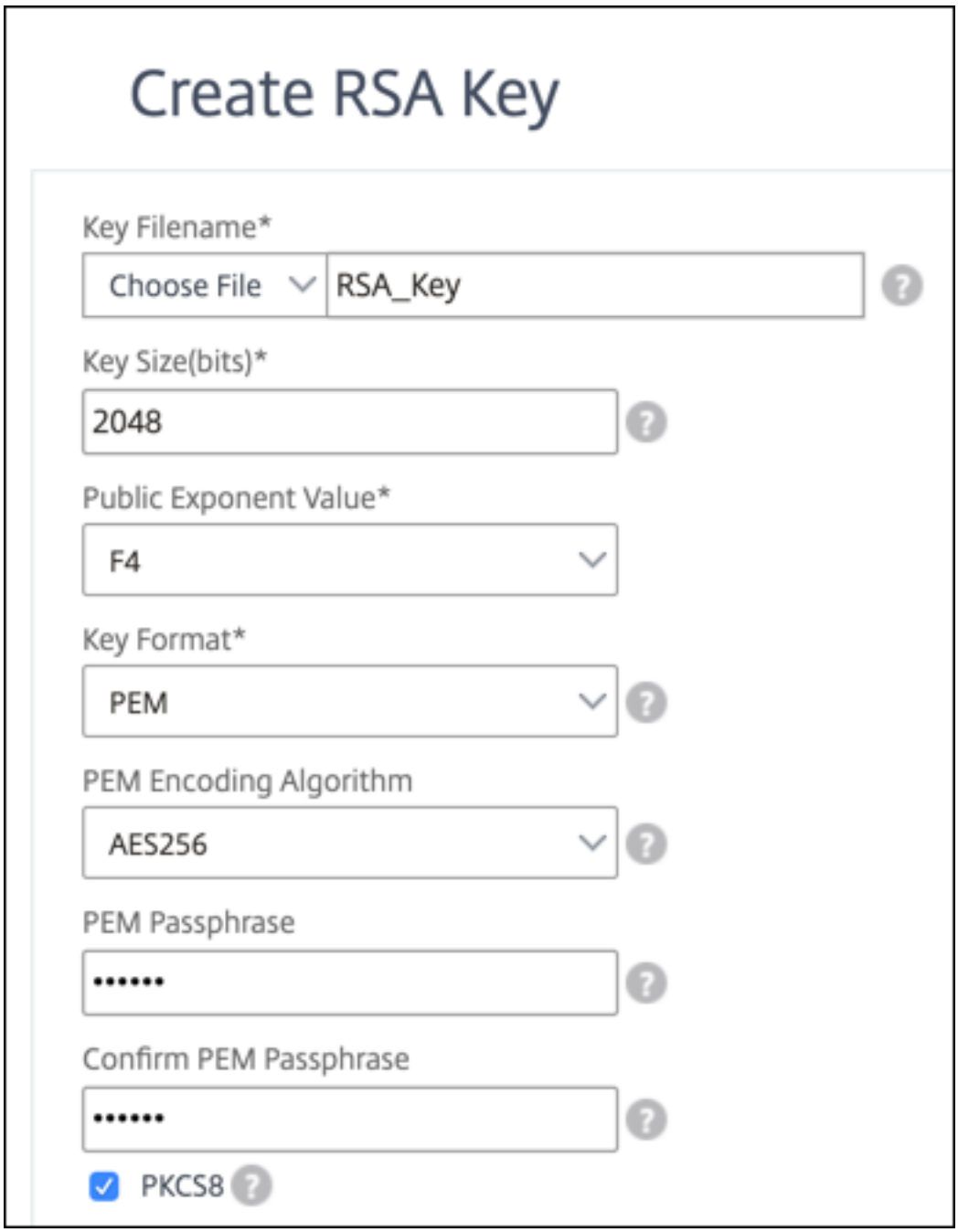
```
create ssl rsakey <keyFile> <bits> [-exponent ( 3 | F4 )] [-keyform (
   DER | PEM )] [-des | -des3 | -aes256] {
 -password  }
  [-pkcs8]
<!--NeedCopy-->
```

例**:**

```
create rsakey testkey 2048 -aes256 -password 123456 -pkcs8
<!--NeedCopy-->
```

GUI を使用して **RSA** 秘密キーを作成する

1. [トラフィック管理] **> [SSL] > [SSL** ファイル] に移動します。
2. [キー] タブで、[**RSA** キーの作成] を選択します。
3. 次のパラメータに値を入力し、[**Create**] をクリックします。
   - **Key Filename** : RSA キーファイルの名前と、オプションで RSA キーファイルのパス。/nsconfig/ssl/ がデフォルトパスです。
   - キーサイズ: RSA キーのサイズ（ビット単位）。範囲は 512 ビットから 4096 ビットです。
   - 公開指数値 -RSA キーの公開指数。指数は暗号アルゴリズムの一部であり、RSA キーの作成に必要です。
   - **Key Format** : RSA キー・ファイルがアプライアンスに保存される形式。
   - **PEM** エンコードアルゴリズム -AES 256、DES、またはトリプル DES (DES3) アルゴリズムを使用して、生成された RSA キーを暗号化します。デフォルトでは、秘密鍵は暗号化されません。
   - **PEM Passphrase** : 秘密キーが暗号化されている場合は、キーのパスフレーズを入力します。



GUI を使用して **RSA** キーで **AES256** エンコードアルゴリズムを選択する

1. トラフィック管理 **> SSL > SSL** ファイル **> RSA** キーの作成に移動します。
2. [キー形式] で [**PEM**] を選択します。
3. [**PEM** エンコードアルゴリズム] で [**AES256**] を選択します。
4. **PKCS8** を選択します。

CLI を使用して証明書署名要求を作成する

コマンドプロンプトで入力します。

```
create ssl certreq <reqFile> -keyFile <input_filename> | -fipsKeyName <
   string>) [-keyForm (DER | PEM) {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName <string>
     -organizationUnitName <string> -localityName <string> -commonName
    <string> -emailAddress <string> {
 -challengePassword  }
  -companyName <string> -digestMethod ( SHA1 | SHA256 )
<!--NeedCopy-->
```

例**:**

```
create ssl certreq priv_csr_sha256 -keyfile priv_2048_2 -keyform PEM -
   countryName IN -stateName Karnataka -localityName Bangalore -
   organizationName Citrix -organizationUnitName NS -digestMethod
   SHA256
<!--NeedCopy-->
```

GUI を使用して証明書署名要求を作成する

1. [トラフィック管理**] > [SSL**] に移動します。
2. [**SSL** 証明書] で、[証明書署名要求 **(CSR)** の作成] をクリックします。
3. 「ダイジェスト 方式」で「**SHA256** 」を選択します。

詳細については 、CSR の作成を参照してください 。

証明書署名要求でのサブジェクト代替名のサポート

証明書のサブジェクト代替名 (SAN) フィールドを使用すると、ドメイン名や IP アドレスなどの複数の値を 1 つの証明書に関連付けることができます。つまり、www.example.com、www.example1.com、www.example2.com などの複数のドメインを 1 つの証明書で保護できます。

Google Chrome などの一部のブラウザでは、証明書署名要求 (CSR) で共通名がサポートされなくなりました。公的に信頼されたすべての証明書に SAN を適用します。

NetScaler ADC アプライアンスは、CSR の作成時に SAN 値を追加することをサポートしています。SAN エントリを含む CSR を認証局に送信して、その SAN エントリを含む署名付き証明書を取得できます。アプライアンスは要求を受信すると、サーバ証明書の SAN エントリで一致するドメイン名があるかどうかを確認します。一致が見つかると、証明書がクライアントに送信され、SSL ハンドシェイクが完了します。CLI または GUI を使用して、SAN 値を持つ CSR を作成できます。

> 注： NetScaler ADC アプライアンスは、DNS ベースの SAN 値のみを処理します。

CLI を使用してサブジェクトの別名で **CSR** を作成する

```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-subjectAltName <string>] [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName <string>
      [-organizationUnitName <string>] [-localityName <string>] [-
     commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

パラメーター:

subjectAltName: サブジェクト代替名 (SAN) は X.509 の拡張であり、subjectAltName フィールドを使用してさまざまな値をセキュリティ証明書に関連付けることができます。これらの値は「サブジェクトの別名」(SAN) と呼ばれます。名前には以下が含まれます。

1. IP アドレス（「IP:」が付いたプレフィックス例:IP: 198.51.10.5 IP: 192.0.2.100）
2. DNS 名 (プレフィックスに「DNS:」が付く例:DNS: www.example.com dns: www.example.org dns: www.example.net)

コマンドラインで、引用符で囲んで値を入力します。2 つの値はスペースで区切ります。GUI では引用符は必要ありません。
最大長：127

例**:**

```
create certReq test1.csr -keyFile test1.ky -countryName IN -stateName
    Kar -organizationName citrix -commonName ctx.com -subjectAltName "
    DNS:*.example.com DNS:www.example.org DNS:www.example.net"
<!--NeedCopy-->
```

> 注:
>
> FIPS アプライアンスで、アプライアンスで FIPS キーを直接作成する場合は、キーファイル名を FIPS キー名に置き換える必要があります。
>
> ```
> create certReq <csrname> -fipsKeyName fipskey.ky -countryName IN -
>     stateName Kar -organizationName citrix -commonName ctx.com -
>     subjectAltName "DNS:www.example.com DNS:www.example.org DNS:www.
>     example.net"
> <!--NeedCopy-->
> ```


GUI を使用して **CSR** を作成する

1. **Traffic Management** > **SSL** > **SSL Files** に移動します。
2. [**CSR**] タブで、[証明書署名要求 **(CSR)** の作成] をクリックします。
3. 値を入力し、[**Create**] をクリックします。

制限事項

SSL 証明書の作成時に SAN を使用するには、SAN 値を明示的に指定する必要があります。値は CSR ファイルから自動的には読み込まれません。

CSR を認証局に提出する

ほとんどの認証局 (CA) は、電子メールによる証明書の提出を受け付けています。CA は、CSR の送信元の電子メールアドレスに有効な証明書を返します。

CSR は/`nsconfig`/`ssl`フォルダに保存されます。

テスト証明書の生成

> 注:
>
> サーバーテスト証明書を生成するには、サーバーテスト証明書の生成を参照してください。

NetScaler ADC アプライアンスには、テスト用に自己署名証明書を作成するために使用できる CA ツールスイートが組み込まれています。

注意： NetScaler ADC アプライアンスは実際の CA ではなくこれらの証明書に署名するため、実稼働環境では使用しないでください。実稼働環境で自己署名証明書を使用しようとすると、仮想サーバーにアクセスするたびに「証明書が無効です」という警告がユーザーに表示されます。

アプライアンスでは、次のタイプの証明書の作成がサポートされています。

- ルート CA 証明書
- 中間 CA 証明書
- エンドユーザー証明書
  - サーバー証明書
  - クライアント証明書

証明書を生成する前に、秘密キーを作成し、その秘密キーを使用してアプライアンスに証明書署名要求 (CSR) を作成します。次に、CSR を CA に送信する代わりに、NetScaler CA Tools を使用して証明書を生成します。

ウィザードを使用して証明書を作成する

1. [トラフィック管理] **> [SSL**] に移動します。
2. 詳細ウィンドウの [はじめに] で、作成する証明書の種類に応じたウィザードを選択します。
3. 画面の指示に従って操作します。

CLI を使用してルート **CA** 証明書を作成する

コマンドプロンプトで、次のコマンドを入力します。

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days <positive_integer>]
<!--NeedCopy-->
```

次の例では、csreq1 が CSR で、rsa1 が以前に作成された秘密キーです。

例**:**

```
create ssl cert cert1 csreq1 ROOT_CERT -keyFile rsa1 -keyForm PEM -days
     365

  Done
<!--NeedCopy-->
```

CLI を使用して中間 **CA** 証明書を作成する

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days <positive_integer>]
     [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm (
     DER | PEM )] [-CAkey <input_filename>] [-CAkeyForm ( DER | PEM )]
    [-CAserial <output_filename>]
<!--NeedCopy-->
```

次の例では、csr1 は以前に作成された CSR です。cert1 と rsakey1 は、自己署名 (ルート CA) 証明書の証明書および対応するキーであり、pvtkey1 は中間 CA 証明書の秘密キーです。

例**:**

```
create ssl cert certsy csr1 INTM_CERT -CAcert cert1 -CAkey rsakey1 -
    CAserial 23
Done

create ssl rsakey pvtkey1 2048 -exponent F4 -keyform PEM
Done
<!--NeedCopy-->
```


GUI を使用してルート **CA** 証明書を作成する

[トラフィック管理**] > [SSL**] に移動し、[はじめに] グループで [ルート **CA** 証明書ウィザード] を選択して、ルート CA 証明書を構成します。

GUI を使用して中間 **CA** 証明書を作成する

[トラフィック管理**] > [SSL**] に移動し、[はじめに] グループで [中間 **CA** 証明書ウィザード] を選択して、中間 CA 証明書を設定します。

エンドユーザー証明書を作成する

エンドユーザー証明書は、クライアント証明書でもサーバー証明書でもかまいません。テストエンドユーザ証明書を作成するには、中間 CA 証明書または自己署名ルート CA 証明書を指定します。

> 注**:** 本番環境で使用するエンドユーザー証明書を作成するには、信頼できる CA 証明書を指定し、その CSR を認証局 (CA) に送信します。

コマンドラインインターフェイスを使用してテスト用エンドユーザー証明書を作成する

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days<positive_integer>]
    [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm (
    DER | PEM )] [-CAkey<input_filename>] [-CAkeyForm ( DER | PEM )] [-
    CAserial <output_filename>]
<!--NeedCopy-->
```

中間証明書がない場合は、CAcertおよびCAkeyのルート CA 証明書の証明書（cert1）と秘密キー（rsakey1）の値を使用します。

例**:**

```
create ssl cert cert12 csr1 SRVR_CERT -CAcert cert1 -CAkey rsakey1 -
    CAserial 23

Done
<!--NeedCopy-->
```

中間証明書がある場合は、CAcertとCAkeyの中間証明書の certificate (certsy) と秘密キー (pvtkey1) の値を使用します。

例**:**

```
create ssl cert cert12 csr1 SRVR_CERT -CAcert certsy -CAkey pvtkey1 -
    CAserial 23

```



```
Done
<!--NeedCopy-->
```

OpenSSL を使用して自己署名 SAN 証明書を作成する

複数のサブジェクト代替名を持つ自己署名 SAN 証明書を作成するには、次の手順を実行します。

1. 会社の要件に従って関連するフィールドを編集して、ローカルコンピューターに OpenSSL 設定ファイルを作成します。

   注**:** 次の例では、設定ファイルは「**req.conf**」です。

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no
[req_distinguished_name]
C = US
ST = VA
L = SomeCity
O = MyCompany
OU = MyDivision
CN = www.company.com
[v3_req]
keyUsage = keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names
[alt_names]
DNS.1 = www.company.net
DNS.2 = company.com
DNS.3 = company.net
<!--NeedCopy-->
```

2. NetScaler ADC アプライアンスの /nsconfig/ssl ディレクトリにファイルをアップロードします。
3. NetScaler CLI にユーザーnsrootとしてログオンし、シェルプロンプトに切り替えます。
4. 次のコマンドを実行して証明書を作成します。

```
cd /nsconfig/ssl
openssl req -x509 -nodes -days 730 -newkey rsa:2048 -keyout cert.
   pem -out cert.pem -config req.conf -extensions 'v3_req'
<!--NeedCopy-->
```

5. 次のコマンドを実行して証明書を検証します。

```
openssl x509 -in cert.pem -noout  - text
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
```



```
ed:90:c5:f0:61:78:25:ab
Signature Algorithm: md5WithRSAEncryption
Issuer: C=US, ST=VA, L=SomeCity, O=MyCompany, OU=MyDivision, CN=
   www.company.com
Validity
Not Before: Nov 6 22:21:38 2012 GMT
Not After : Nov 6 22:21:38 2014 GMT
Subject: C=US, ST=VA, L=SomeCity, O=MyCompany, OU=MyDivision, CN=
   www.company.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (2048 bit)
Modulus (2048 bit):
…
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Key Usage:
Key Encipherment, Data Encipherment
X509v3 Extended Key Usage:
TLS Web Server Authentication
X509v3 Subject Alternative Name:
DNS:www.company.net, DNS:company.com, DNS:company.net
Signature Algorithm: md5WithRSAEncryption …
<!--NeedCopy-->
```

証明書のインストール、リンク、および更新

March 20, 2024

証明書をインストールするには、「証明書とキーのペアの追加または更新」を参照してください。

> 注:
>
> NetScaler PEM、DER、および PFX (PKCS12) の証明書形式をサポートしています。
>
> - PEM には、.crt、.pem、または.cer 拡張子を付けることができます。
> - DER には.der という拡張子が付いています。
> - PFX には、.p12 または.pfx の拡張子を付けることができます。
>
> 通常、ファイル名拡張子は関係ありません。ファイルの内容は異なる方法でエンコードされ、それによって形式が決まり、それに応じて証明書が解析されます。

リンク証明書

多くのサーバー証明書は、複数の階層的な認証局 (CA) によって署名されています。つまり、証明書は次のようなチェーンを形成します。


```
Root CA
|
+------------ Intermediate CA
|
+----------------------- End-Entity (Server) Certificate
```

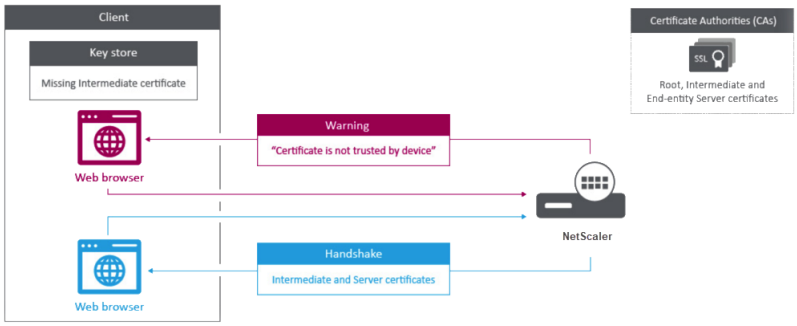
中間 CA がプライマリとセカンダリの中間 CA 証明書に分割されることがあります。証明書は次のようなチェーンを形成します。

```
Root CA
|
+------ Primary Intermediate CA
|+--------------- Secondary Intermediate CA
|
+----------------------- End-Entity (Server) Certificate
```

通常、クライアントマシンのローカル証明書ストアにはルート CA 証明書が含まれていますが、1 つ以上の中間 CA 証明書は含まれていません。ADC アプライアンスは、1 つ以上の中間 CA 証明書をクライアントに送信する必要があります。

注**:** アプライアンスはルート CA 証明書をクライアントに送信してはなりません。公開キー基盤 (PKI) の信頼関係モデルでは、ルート CA 証明書をアウトオブバンド方式でクライアントにインストールする必要があります。たとえば、証明書はオペレーティングシステムや Web ブラウザに含まれています。クライアントは、アプライアンスから送信されたルート CA 証明書を無視します。

標準的な Web ブラウザが信頼できる CA として認識しない中間 CA が、サーバ証明書を発行することがあります。この場合、1 つ以上の CA 証明書をサーバ独自の証明書とともにクライアントに送信する必要があります。そうしないと、サーバ証明書の認証に失敗するため、ブラウザは SSL セッションを終了します。



サーバ証明書と中間証明書を追加するには、次のセクションを参照してください。

- 手動による証明書リンク
- 証明書リンクの自動化
- 証明書のチェーンを作成する

中間機関証明書をリンクする方法

これは埋め込みビデオです。リンクをクリックしてビデオを見る

手動による証明書リンク

注：この機能は、NetScaler FIPS プラットフォームおよびクラスターセットアップではサポートされていません。

個々の証明書を追加してリンクする代わりに、サーバ証明書と最大 9 個の中間証明書を 1 つのファイルにグループ化できるようになりました。証明書とキーのペアを追加するときに、ファイル名を指定できます。その前に、次の前提条件が満たされていることを確認してください。

- ファイル内の証明書は次の順序になっています。
  - サーバー証明書 (ファイル内の最初の証明書であることが必要)
  - オプションで、サーバーキー
  - 中間証明書 1 (ic1)
  - 中間証明書 2 (ic2)
  - 中間証明書 3 (ic3) など
    注: 中間証明書ファイルは、中間証明書ごとに「<certificatebundlename>.pem_ic<*n*>」という名前で作成されます。n は 1 ～9 です。たとえば、bundle.pem_ic1 と指定します。**bundle** は証明書セットの名前で、ic1 は証明書セット内の最初の中間証明書です。
- [バンドル] オプションが選択されている。
- このファイルには中間証明書が 9 つまで存在しません。

ファイルが解析され、サーバ証明書、中間証明書、およびサーバキー (存在する場合) が識別されます。まず、サーバ証明書とキーが追加されます。次に、中間証明書がファイルに追加された順序で追加され、それに応じてリンクされます。

次の条件のいずれかに当てはまる場合、エラーが報告されます。

- アプライアンスには、中間証明書の 1 つの証明書ファイルがあります。
- この鍵は、ファイル内のサーバ証明書の前に置かれます。
- 中間証明書はサーバ証明書の前に置かれます。
- 中間証明書は、作成時と同じ順序でファイル内に配置されません。
- このファイルには証明書がありません。
- 証明書が適切な PEM 形式ではありません。
- ファイル内の中間証明書の数が 9 個を超えています。

CLI を使用して証明書セットを追加する

コマンドプロンプトで次のコマンドを入力して証明書セットを作成し、構成を確認します。

```
add ssl certKey <certkeyName> -cert <string> -key <string> -bundle (YES
    | NO)

show ssl

```



```
show ssl certlink
<!--NeedCopy-->
```

次の例では、証明書セット (bundle.pem) に次のファイルが含まれています。

bundle_ic1 にリンクされたサーバー証明書 (バンドル)

bundle_ic2 にリンクされた最初の中間証明書 (bundle_ic1)

bundle_ic3 にリンクされた 2 つ目の中間証明書 (bundle_ic2)

3 番目の中間証明書 (bundle_ic3)

```
add ssl certKey bundletest -cert bundle9.pem -key bundle9.pem -bundle
   yes

sh ssl certkey

1)      Name: ns-server-certificate
        Cert Path: ns-server.cert
        Key Path: ns-server.key
        Format: PEM
        Status: Valid,   Days to expiration:5733
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS
           Internal,CN=default OULLFT
        Validity
                Not Before: Apr 21 15:56:16 2016 GMT
                Not After : Mar  3 06:30:56 2032 GMT
        Subject:  C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS
           Internal,CN=default OULLFT
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

2)      Name: servercert
        Cert Path: complete/server/server_rsa_1024.pem
        Key Path: complete/server/server_rsa_1024.ky
        Format: PEM
        Status: Valid,   Days to expiration:7150
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 1F
        Signature Algorithm: sha1WithRSAEncryption
        Issuer:  C=IN,ST=KAR,O=Citrix R&D Pvt Ltd,CN=Citrix
        Validity
                Not Before: Sep  2 09:54:07 2008 GMT
                Not After : Jan 19 09:54:07 2036 GMT
```



```
        Subject:  C=IN,ST=KAR,O=Citrix Pvt Ltd,CN=Citrix
        Public Key Algorithm: rsaEncryption
        Public Key size: 1024

3)      Name: bundletest
        Cert Path: bundle9.pem
        Key Path: bundle9.pem
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA9
        Validity
                Not Before: Nov 28 06:43:11 2014 GMT
                Not After : Nov 25 06:43:11 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=Server9
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

4)      Name: bundletest_ic1
        Cert Path: bundle9.pem_ic1
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA8
        Validity
                Not Before: Nov 28 06:42:56 2014 GMT
                Not After : Nov 25 06:42:56 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA9
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

5)      Name: bundletest_ic2
        Cert Path: bundle9.pem_ic2
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA7
        Validity
```



```
                Not Before: Nov 28 06:42:55 2014 GMT
                Not After : Nov 25 06:42:55 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA8
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

6)      Name: bundletest_ic3
        Cert Path: bundle9.pem_ic3
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA6
        Validity
                Not Before: Nov 28 06:42:53 2014 GMT
                Not After : Nov 25 06:42:53 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA7
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

7)      Name: bundletest_ic4
        Cert Path: bundle9.pem_ic4
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA5
        Validity
                Not Before: Nov 28 06:42:51 2014 GMT
                Not After : Nov 25 06:42:51 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA6
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

8)      Name: bundletest_ic5
        Cert Path: bundle9.pem_ic5
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA4
```



```
        Validity
                Not Before: Nov 28 06:42:50 2014 GMT
                Not After : Nov 25 06:42:50 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA5
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

9)      Name: bundletest_ic6
        Cert Path: bundle9.pem_ic6
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA3
        Validity
                Not Before: Nov 28 06:42:48 2014 GMT
                Not After : Nov 25 06:42:48 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA4
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

10)     Name: bundletest_ic7
        Cert Path: bundle9.pem_ic7
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA2
        Validity
                Not Before: Nov 28 06:42:46 2014 GMT
                Not After : Nov 25 06:42:46 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA3
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

11)     Name: bundletest_ic8
        Cert Path: bundle9.pem_ic8
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
```



```
198          Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA1
199          Validity
200                  Not Before: Nov 28 06:42:45 2014 GMT
201                  Not After : Nov 25 06:42:45 2024 GMT
202          Subject:  C=IN,ST=ka,O=sslteam,CN=ICA2
203          Public Key Algorithm: rsaEncryption
204          Public Key size: 2048
205
206  12)     Name: bundletest_ic9
207          Cert Path: bundle9.pem_ic9
208          Format: PEM
209          Status: Valid,   Days to expiration:3078
210          Certificate Expiry Monitor: ENABLED
211          Expiry Notification period: 30 days
212          Certificate Type: Intermediate CA
213          Version: 3
214          Serial Number: 01
215          Signature Algorithm: sha256WithRSAEncryption
216          Issuer:  C=IN,ST=ka,O=sslteam,CN=RootCA4096
217          Validity
218                  Not Before: Nov 28 06:42:43 2014 GMT
219                  Not After : Nov 25 06:42:43 2024 GMT
220          Subject:  C=IN,ST=ka,O=sslteam,CN=ICA1
221          Public Key Algorithm: rsaEncryption
222          Public Key size: 2048
223  Done
224
225  sh ssl certlink
226
227  1)      Cert Name: bundletest    CA Cert Name: bundletest_ic1
228  2)      Cert Name: bundletest_ic1        CA Cert Name: bundletest_ic2
229  3)      Cert Name: bundletest_ic2        CA Cert Name: bundletest_ic3
230  4)      Cert Name: bundletest_ic3        CA Cert Name: bundletest_ic4
231  5)      Cert Name: bundletest_ic4        CA Cert Name: bundletest_ic5
232  6)      Cert Name: bundletest_ic5        CA Cert Name: bundletest_ic6
233  7)      Cert Name: bundletest_ic6        CA Cert Name: bundletest_ic7
234  8)      Cert Name: bundletest_ic7        CA Cert Name: bundletest_ic8
235  9)      Cert Name: bundletest_ic8        CA Cert Name: bundletest_ic9
236  Done
237  <!--NeedCopy-->
```

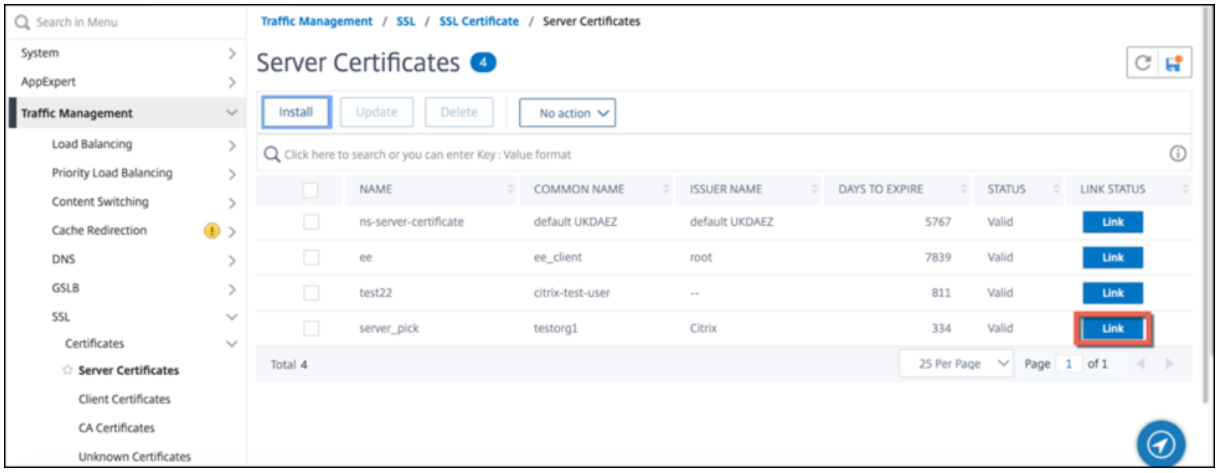
GUI を使用して証明書セットを追加する

1. [トラフィック管理] > [**SSL**] > [証明書] > [**CA** 証明書] に移動します。
2. 詳細ペインで、[インストール] をクリックします。
3. [証明書のインストール] ダイアログボックスで、証明書やキーファイル名などの詳細を入力し、[証明書バンドル] を選択します。
4. [インストール] をクリックし、[閉じる] をクリックします。

証明書リンクの自動化

注**:** この機能は、リリース 13.0 ビルド 47.x から利用できます。

ルート証明書に至るまで、証明書をその発行者に手動でリンクする必要がなくなりました。中間 CA 証明書とルート証明書がアプライアンスに存在する場合は、エンドユーザ証明書の [リンク（Link）] ボタンをクリックできます。



ポテンシャルチェーンが表示されます。

[証明書をリンク] をクリックして、すべての証明書をリンクします。

証明書のチェーンを作成する

証明書のセット (1 つのファイル) を使用する代わりに、証明書のチェーンを作成できます。チェーンはサーバー証明書をその発行者 (中間 CA) にリンクします。この方法では、中間 CA 証明書ファイルが ADC アプライアンスにインストールされ、クライアントアプリケーションがチェーン内の証明書の 1 つを信頼する必要があります。たとえば、Cert-Intermediate-A を Cert-Intermediate-B にリンクします。ここで、Cert-Intermediate-B は、クライアントアプリケーションによって信頼される証明書である Cert-Intermediate-C にリンクされます。

注：アプライアンスは、クライアントに送信される証明書のチェーン内で最大 10 個の証明書（1 つのサーバ証明書と 9 つの CA 証明書）の送信をサポートします。

CLI を使用して証明書チェーンを作成する

コマンドプロンプトで次のコマンドを入力して証明書チェーンを作成し、構成を確認します。(チェーン内の新しいリンクごとに 1 つ目のコマンドを繰り返します)。

```
link ssl certkey <certKeyName> <linkCertKeyName>
show ssl certlink
<!--NeedCopy-->
```

例:


```
link ssl certkey siteAcertkey CAcertkey
Done

show ssl certlink

linked certificate:
       1) Cert Name: siteAcertkey CA Cert Name: CAcertkey
Done
<!--NeedCopy-->
```

GUI を使用して証明書チェーンを作成する

1. [トラフィック管理] > [**SSL**] > [証明書] に移動します。
2. サーバー証明書を選択し、［アクション］リストで［リンク］を選択し、CA 証明書名を指定します。

SSL 証明書バンドルのサポート

> 注:
>
> この機能は、リリース 13.1 ビルド 12.x から利用できます。

証明書バンドルの現在の設計には、次のような欠点があります。

- 証明書バンドルを追加すると、設定に複数のコマンドが追加されます。したがって、2 つのバンドルが共通の中間証明書を共有している場合は、別の証明書バンドルを追加することはできません。
- 証明書バンドルの削除は手動のプロセスです。ファイルは特定の順序で手動で削除する必要があります。
- 証明書バンドルの更新はサポートされていません。
- クラスタはサポートされていません。

証明書バンドルの新しい設計により、これらの問題がすべて解決されます。新しいエンティティは証明書バンドルファイルで動作します。したがって、中間証明書ごとにファイルを作成する必要はありません。この新しいエンティティを使用すると、削除も簡単です。

2 つの証明書バンドルで、中間証明書チェーンの一部を共有できます。証明書バンドルの一部でもある同じサーバ証明書とキーを使用して、証明書とキーのペアを追加することもできます。

次の例では、次のようになります。

1. 証明書バンドル bundle1.pem には、サーバ証明書 (S1) と中間証明書 (IC1 および IC2) が含まれています。
2. サーバー証明書は server_cert.pem (S1) です。
3. 中間証明書は ic1.pem (IC1) と ic2.pem (IC2) です。

S1、IC1、IC2 を含む証明書バンドルを追加できます。

```
add ssl certkeybundle b1 -bundlefile bundle1.pem
```


S1 と IC1 を使用して証明書とキーのペアを追加することもできます。

```
add ssl certkey server-cert -cert server_cert.pem

add ssl certkey ic1 -cert ic1.pem
```

重要:

- 次の順序が満たされない場合、バンドルの作成は失敗します。
  - サーバー証明書 (SC) はバンドルファイルの先頭に配置する必要があります。
  - IC[1-9]は中間証明書です。IC[i]はIC[i+1]によって発行されます。証明書は順番に配置し、すべての中間証明書がバンドル内に存在する必要があります。
- 証明書は PEM 形式のみである必要があります。
- サーバー証明書キー (SCK) はバンドルの任意の場所に配置できます。
- 最大 9 個の中間証明書がサポートされます。

CLI を使用して証明書バンドルを追加するには

コマンドプロンプトで入力します:

```
add ssl certKeyBundle <bundle_name> -bundlefile <bundle_file_name> -
passplain <>
```

例:

```
add ssl certkeyBundle cert_bundle -bundlefile bundle_4096.pem
```

GUI を使用して証明書バンドルを追加するには

1. [トラフィック管理] **> [SSL] > [**証明書キーバンドル] に移動します。
2. [**Install**] をクリックします。
3. バンドルの名前を指定し、アプライアンスまたはローカルコンピュータ上のバンドルの場所を参照して、ファイルを選択します。
4. [**Create**] をクリックします。

CLI を使用して証明書バンドルを更新するには

> 注**:**
>
> 更新コマンドは、リリース 13.1 ビルド 52.x 以降で使用できます。

コマンドプロンプトで入力します:

```
update ssl certKeyBundle [-bundlefile <input_filename>]
```


例:

```
update ssl certKeyBundle cert_bundle -bundlefile bundle_4096_updated.
pem
```

GUI を使用して証明書バンドルを更新するには

1. [トラフィック管理] **> [SSL] > [**証明書キーバンドル] に移動します。
2. バンドルを選択し、「更新」をクリックします。
3. アプライアンスまたはローカルコンピュータ上のバンドルの場所を参照し、ファイルを選択します。
4. [**OK**] をクリックします。

CLI を使用して証明書バンドルを削除するには

コマンドプロンプトで入力します:

```
rm ssl certKeyBundle <bundle_name>
```

例:

```
rm ssl certkeybundle cert_bundle
```

GUI を使用して証明書バンドルを削除するには

1. [トラフィック管理] **> [SSL] > [**証明書キーバンドル] に移動します。
2. バンドルを選択し、「削除」をクリックします。
3. 確認ダイアログボックスで、「はい」をクリックします。

証明書バンドルを **SSL** 仮想サーバーにバインドするには

コマンドプロンプトで入力します:

```
bind ssl vserver <vip-name> -certkeybundleName <certkeybundle_name> [
 -SNICertkeybundle]
```

例:

```
1  bind ssl vserver v_server -certkeyBundleName cert_bundle
2
3  show ssl certkeyBundle cert_bundle
4
5  1) Name: cert_bundle
6          Bundle path: bundle_4096.pem
7          Certificate:
8                  Status: Valid,   Days to expiration:278
```



```
                Serial Number: 83
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=4096.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=ia24096.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:17:57 2021 GMT
                        Not After : Jul 13 10:17:57 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 4096
                SAN ENTRIES: None


        CA Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 82
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=ia24096.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=ia14098.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:15:37 2021 GMT
                        Not After : Jul 13 10:15:37 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 4096
                SAN ENTRIES: None

        CA Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 81
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=ia14098.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:13:20 2021 GMT
                        Not After : Jul 13 10:13:20 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 4096
                SAN ENTRIES: None

        CA Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 00
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:10:23 2021 GMT
                        Not After : Jul 13 10:10:23 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 2048
                SAN ENTRIES: None

1)      Vserver Name: v_server
<!--NeedCopy-->
```


証明書バンドルを **SNI** 証明書バンドルとして **SSL** 仮想サーバーにバインドするには

コマンドプロンプトで入力します:

```
bind ssl vserver <vip-name> -certkeybundleName b2 -SNICertkeybundle
```

例:

```
bind ssl vserver v_server -certkeyBundleName cert_bundle -
    sniCertkeybundle

sh ssl certkeybundle cert_bundle

1) Name: cert_bundle
        Bundle path: bundle_4096.pem
        Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 83
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=4096.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=ia24096.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:17:57 2021 GMT
                        Not After : Jul 13 10:17:57 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 4096
                SAN ENTRIES: None


        CA Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 82
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=ia24096.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=ia14098.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:15:37 2021 GMT
                        Not After : Jul 13 10:15:37 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 4096
                SAN ENTRIES: None

        CA Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 81
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=ia14098.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:13:20 2021 GMT
                        Not After : Jul 13 10:13:20 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 4096
                SAN ENTRIES: None
```



```
46
47          CA Certificate:
48                  Status: Valid,   Days to expiration:278
49                  Serial Number: 00
50                  Subject:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
51                  Issuer:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
52                  Signature Algorithm: sha256WithRSAEncryption
53                  Validity
54                          Not Before: Jul 13 10:10:23 2021 GMT
55                          Not After : Jul 13 10:10:23 2022 GMT
56                  Public Key Algorithm: rsaEncryption
57                  Public Key size: 2048
58                  SAN ENTRIES: None
59
60  1)    Vserver Name: v_server
61  2)    Vserver Name: v_server
62  <!--NeedCopy-->
```

SSL 仮想サーバーから証明書バンドルをバインド解除するには

コマンドプロンプトで入力します:

```
unbind ssl vserver <vip-name> -certkeybundleName <certkeybundle_name>
 [ -SNICertkeybundle]
```

例:

```
unbind ssl vserver v_server -certkeybundleName cert_bundle
```

証明書バンドルバインドのユーザーシナリオ

次のシナリオでは、ADC アプライアンスが証明書バンドルに関連する要求を処理する方法を説明します。

シナリオ **1:** 証明書とキーのペアと、同じサーバー証明書を含む証明書バンドルが同じ **SSL** 仮想サーバーにバインドされている

証明書とキーのペアと、同じサーバ証明書を含む証明書バンドルを同じ SSL 仮想サーバにバインドする場合、コマンドの順序によって最終的なバインドが決まります。

例:

- 証明書バンドル bundle1.pem には、サーバ証明書 S1 と中間証明書 IC1 および IC2 が含まれています。
- 証明書ファイル server_cert.pem には S1 が含まれています。

bundle1.pem と server_cert.pem はどちらも同じサーバー証明書 S1 を持っています。

次のコマンドを指定された順序で実行すると、SSL 仮想サーバーにバインドされているサーバー証明書が、その仮想サーバーにバインドされている証明書バンドルに置き換えられます。

1. `add ssl certkeybundle b1 -bundlefile bundle1.pem`
2. `add ssl certkey server_cert -cert server_cert.pem`
3. `bind ssl vserver v1 -certkeybundle b1`
4. `bind ssl vserver v1 -cert server_cert`

シナリオ **2:2** つの証明書バンドルに同じ中間証明書チェーンが含まれている

同じ中間証明書チェーンを持つ 2 つの証明書バンドルを追加できます。2 つのバンドルは独立したエンティティとして機能します。

次の例では、証明書バンドル 1 にはサーバ証明書 S1 と中間証明書 IC1 と IC2 がこの順序で含まれています。証明書バンドル-2 には、サーバ証明書 S2 と、中間証明書 IC1 および IC2 がこの順序で含まれています。

- 証明書バンドル bundle1.pem (S1、IC1、IC2)
- 証明書バンドル bundle2.pem (S2、IC1、IC2)

SSL ハンドシェイクプロセスで bundle-1 の S1 が選択されると、bundle-1 の中間証明書チェーンがクライアントに送信されます。

```
add ssl certkeybundle bundle-1 -bundlefile bundle1.pem
add ssl certkeybundle bundle-2 -bundlefile bundle2.pem
```

シナリオ **2:2** つの証明書バンドルに、チェーン内に共通の中間証明書がいくつか含まれている

いくつかの一般的な中間証明書を含む 2 つの証明書バンドルをチェーンに追加できます。

次の例では、bundle-1 にはサーバ証明書 S1 と中間証明書 IC1 と IC2 が含まれています。証明書バンドル 2 には、サーバ証明書 S2 と中間証明書 IC1、IC2、および IC3 が含まれています。

証明書バンドル bundle1.pem (S1、IC1、IC2)
証明書バンドル 2.pem (S2、IC1、IC2、IC3)

```
add ssl certkeybundle bundle-1 -bundlefile bundle1.pem
add ssl certkeybundle bundle-2 -bundlefile bundle2.pem
```

SSL ハンドシェイクプロセスで bundle-1 の S1 が選択されると、bundle-1 の中間証明書チェーンがクライアントに送信されます。つまり、(S1→IC1→IC2) がクライアントに送信されます。IC3 は追加されません。

SSL ハンドシェイクプロセスで bundle-2 の S2 を選択すると、bundle-2 の中間証明書チェーンはクライアントにのみ送信されます。つまり、(S1→IC1→IC2→IC3) がクライアントに送信されます。

証明書バンドルの制限事項

- 証明書バンドル内の証明書のステータスの監視はサポートされていません。

- 証明書バンドルは SSL 仮想サーバにのみバインドできます。
- OCSP ホチキス止めはサポートされていません。

> 注
>
> 更新操作は、リリース 13.1 ビルド 52.x の証明書バンドルでサポートされています。証明書バンドルを直接更新できるようになりました。以前は、最初にバンドルをバインド解除して削除し、次に証明書バンドルを追加してバインドする必要がありました。

既存のサーバー証明書を更新する

既存のサーバ証明書を手動で変更するには、次の手順を実行する必要があります。

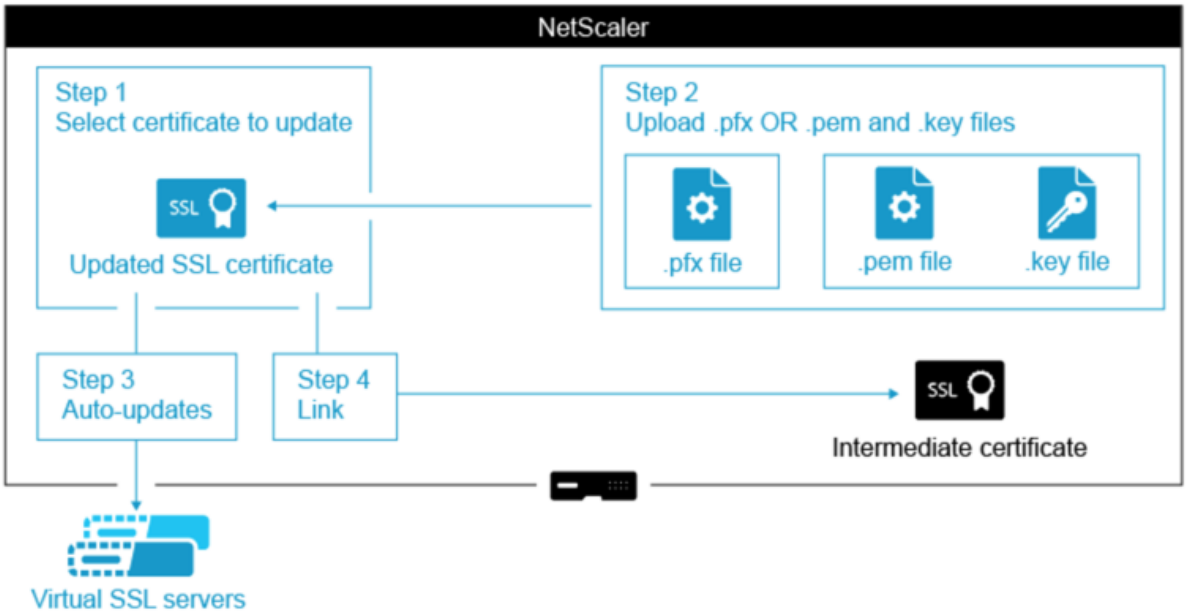
1. 仮想サーバから古い証明書のバインドを解除します。
2. 証明書をアプライアンスから削除します。
3. 新しい証明書をアプライアンスに追加します。
4. 新しい証明書を仮想サーバにバインドします。

証明書とキーのペアを置き換える際のダウンタイムを短縮するために、既存の証明書を更新できます。証明書を別のドメインに発行された証明書に置き換える場合は、証明書を更新する前にドメインチェックを無効にする必要があります。

有効期限が切れる証明書に関する通知を受け取るには、有効期限モニターを有効にします。

構成済みの SSL 仮想サーバーまたはサービスから証明書を削除またはバインド解除すると、その仮想サーバーまたはサービスは非アクティブになります。新しい有効な証明書がバインドされると、アクティブになります。ダウンタイムを減らすために、更新機能を使用して、SSL 仮想サーバーまたは SSL サービスにバインドされている証明書とキーのペアを置き換えることができます。

NetScaler アプライアンスで SSL 証明書を更新する方法の概要図。



既存の証明書を更新する方法

これは埋め込みビデオです。リンクをクリックしてビデオを見る

CLI を使用して既存の証明書とキーペアを更新する

コマンドプロンプトで次のコマンドを入力して、既存の証明書とキーのペアを更新し、構成を確認します。

```
update ssl certkey <certkeyName> -cert <string> -key <string>

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

例:

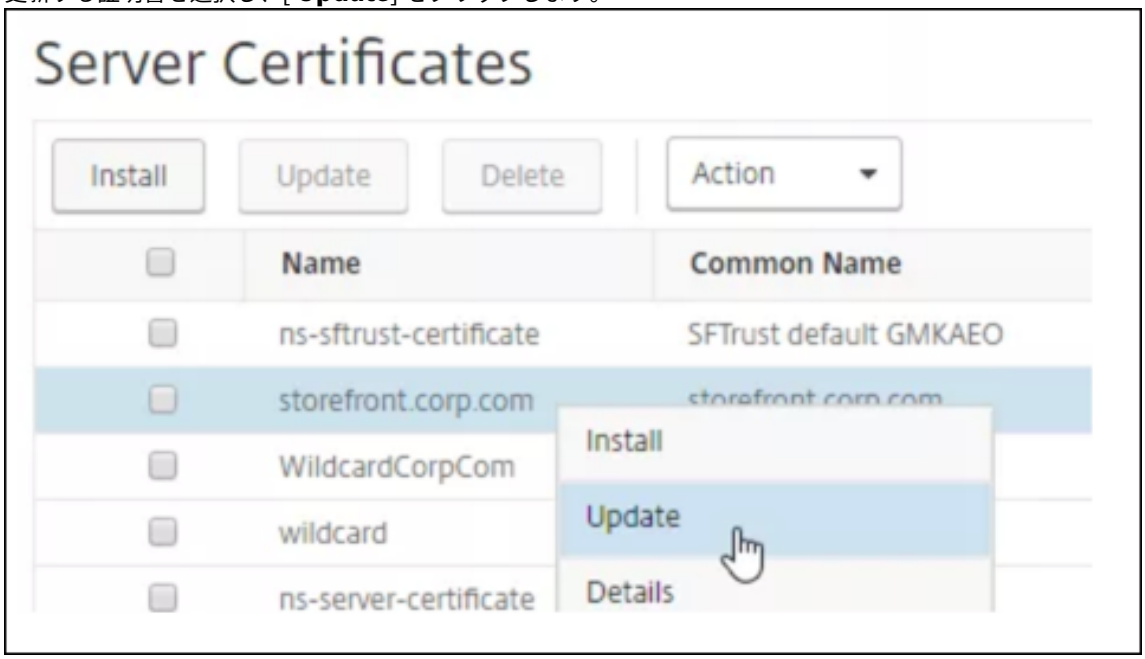
```
update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem -key /
   nsconfig/ssl/pkey.pem

Done

show ssl certkey siteAcertkey

Name: siteAcertkey       Status: Valid
          Version: 3
          Serial Number: 02
          Signature Algorithm: md5WithRSAEncryption
          Issuer: /C=US/ST=CA/L=Santa Clara/O=siteA/OU=Tech
          Validity
               Not Before: Nov 11 14:58:18 2001 GMT
```



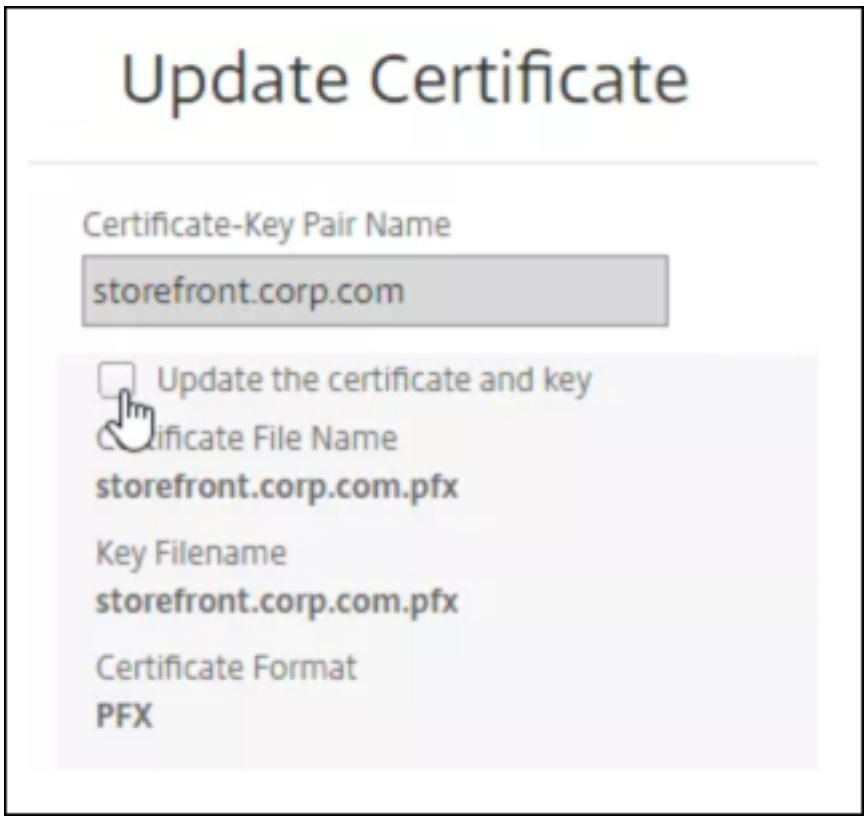
```
                Not After: Aug 7 14:58:18 2004 GMT
            Subject: /C=US/ST-CA/L=San Jose/O=CA/OU=Security
            Public Key Algorithm: rsaEncryption
            Public Key size: 2048
Done
<!--NeedCopy-->
```

GUI を使用して既存の証明書とキーのペアを更新する

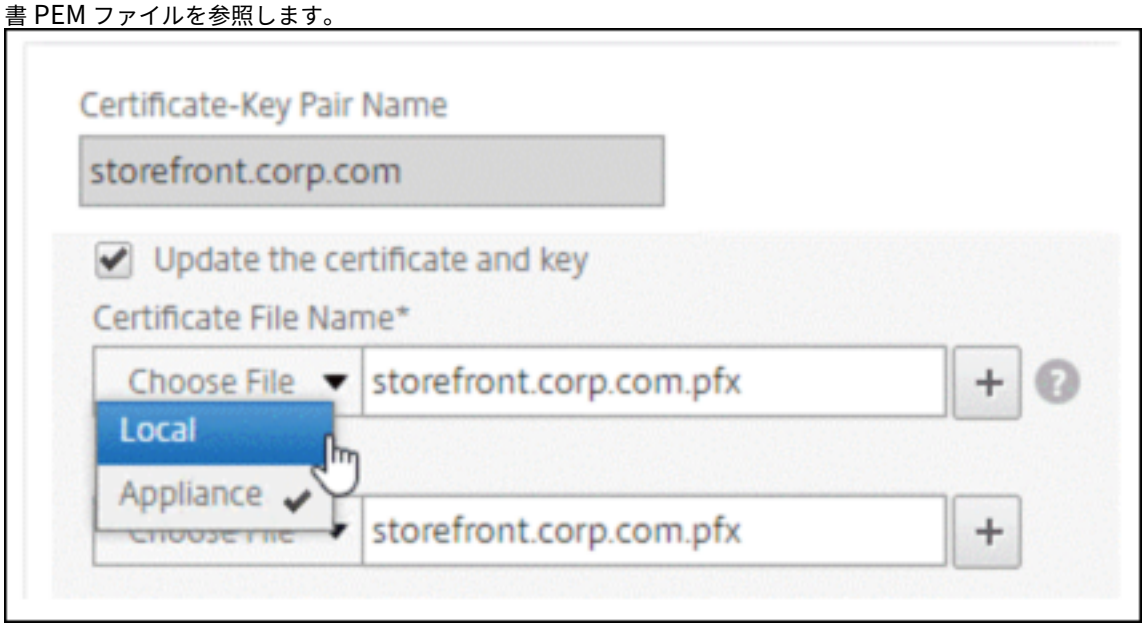
1. [トラフィック管理] > [**SSL**] > [証明書] > [サーバー証明書] に移動します。
2. 更新する証明書を選択し、[**Update**] をクリックします。



3. [証明書とキーを更新] を選択します。

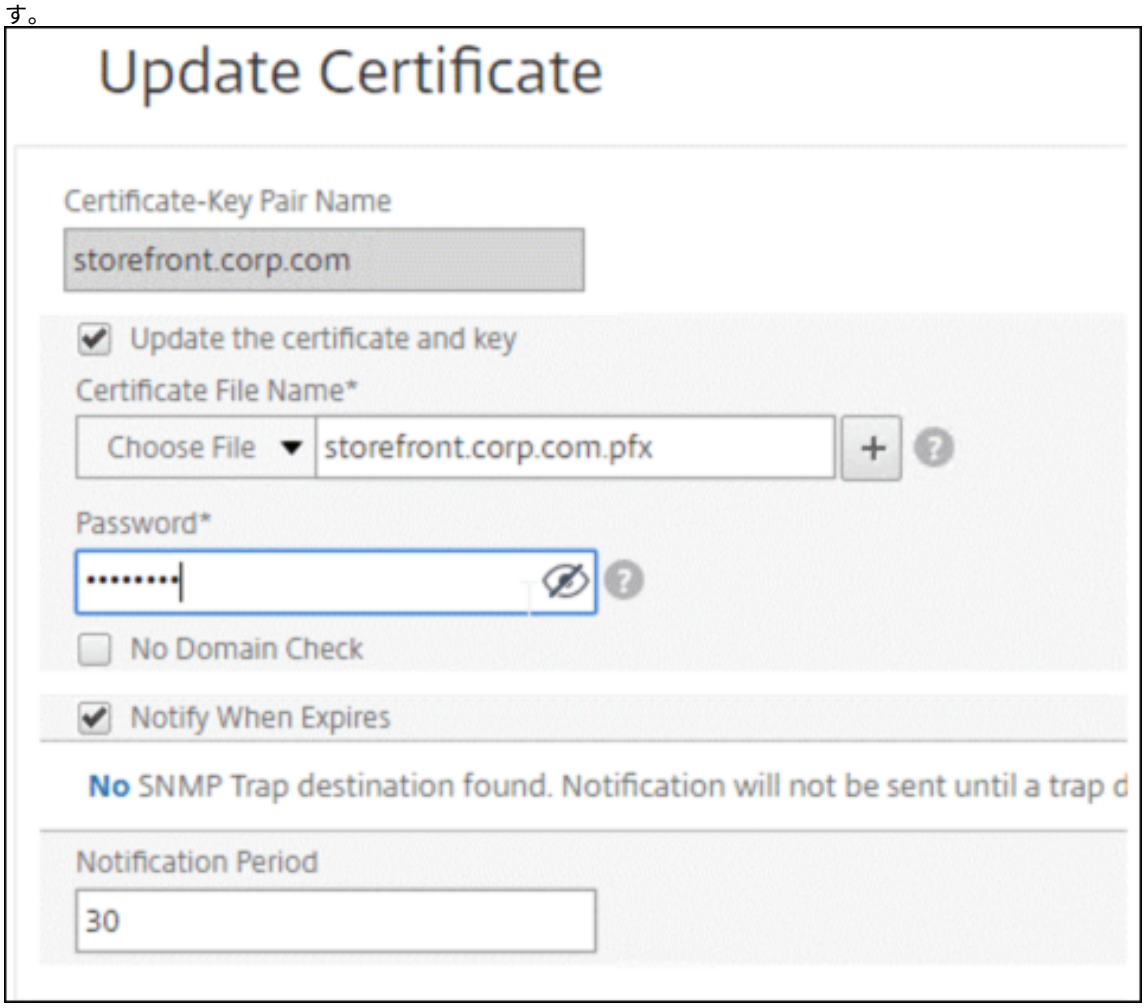


4. [証明書ファイル名] で、[ファイルの選択] > [ローカル] をクリックし、更新された.pfx ファイルまたは証明書 PEM ファイルを参照します。



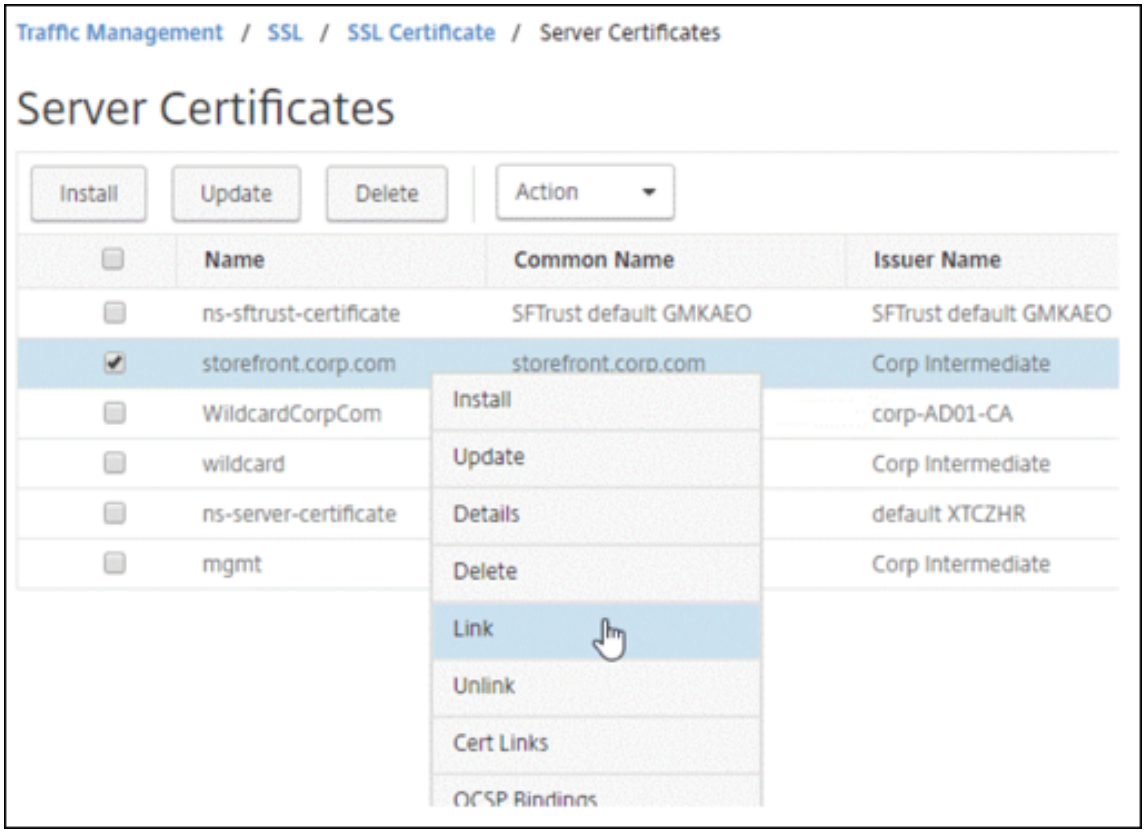
- .pfx ファイルをアップロードすると、.pfx ファイルのパスワードを指定するよう求められます。
- 証明書 pem ファイルをアップロードする場合は、証明書キーファイルもアップロードする必要があります。キーが暗号化されている場合は、暗号化パスワードを指定する必要があります。

5. 新しい証明書の共通名が古い証明書と一致しない場合は、[**No Domain Check**] を選択します。
6. [**OK**] をクリックします。この証明書がバインドされているすべての SSL 仮想サーバーは自動的に更新されます。



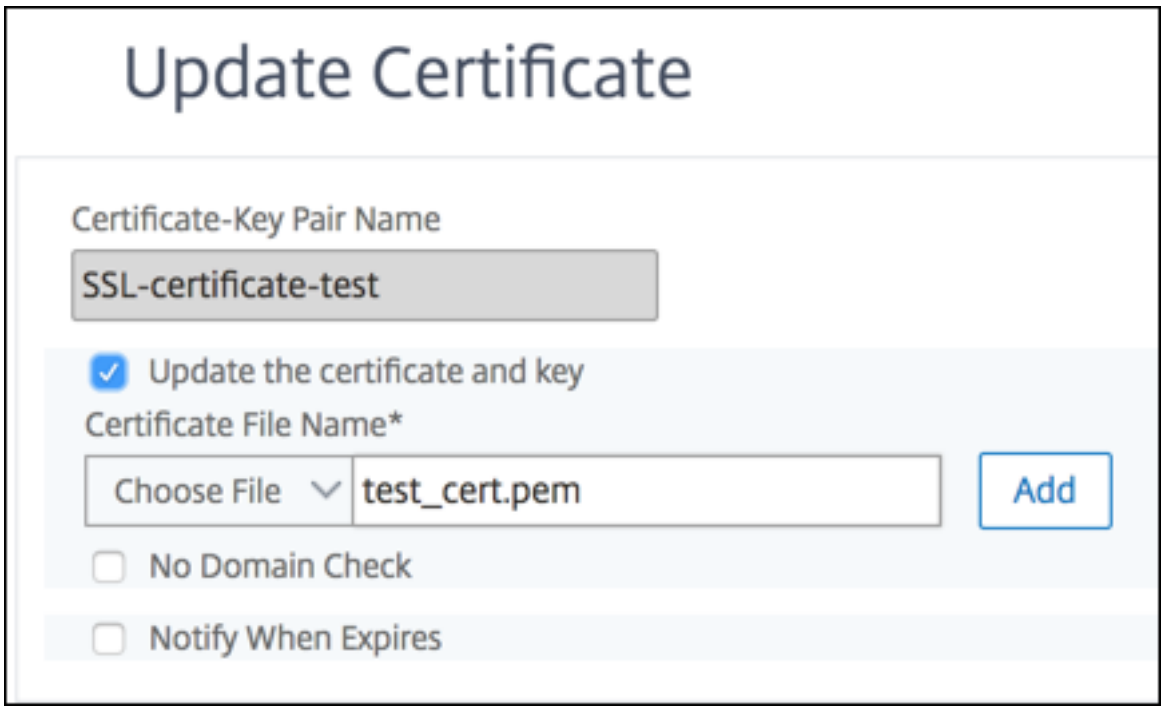
7. 証明書を置き換えた後、証明書リンクを新しい中間証明書に更新しなければならない場合があります。リンクを壊さずに中間証明書を更新する方法の詳細については、「リンクを壊さずに中間証明書を更新する」を参照してください。

   - 更新された証明書を右クリックし、[証明書リンク] をクリックして、中間証明書にリンクされているかどうかを確認します。
   - 証明書がリンクされていない場合は、更新された証明書を右クリックし、[リンク] をクリックして中間証明書にリンクします。リンクするオプションが表示されない場合は、まず、新しい中間証明書をアプライアンスの [**CA Certificates**] ノードにインストールする必要があります。



既存の **CA** 証明書を更新する

既存の CA 証明書を更新する手順は、既存のサーバ証明書を更新する手順と同じです。唯一の違いは、CA 証明書の場合、キーは必要ないということです。



ドメインチェックを無効にする

アプライアンスで SSL 証明書が置き換えられる場合、新しい証明書に記載されているドメイン名は、置き換えられる証明書のドメイン名と一致する必要があります。たとえば、abc.com に発行された証明書があり、def.com に発行された証明書で更新すると、証明書の更新は失敗します。

ただし、特定のドメインをホストしていたサーバーで新しいドメインをホストする場合は、証明書を更新する前にドメインチェックを無効にします。

CLI を使用して証明書のドメインチェックを無効にする

コマンドプロンプトで次のコマンドを入力して、ドメインチェックを無効にし、構成を確認します。

```
update ssl certKey <certkeyName> -noDomainCheck

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

例:

```
update ssl certKey sv -noDomainCheck

Done

show ssl certkey sv

```



```
Name: sv
Cert Path: /nsconfig/ssl/complete/server/server_rsa_512.pem
Key Path: /nsconfig/ssl/complete/server/server_rsa_512.ky
Format: PEM
Status: Valid,   Days to expiration:9349
Certificate Expiry Monitor: DISABLED
Done
<!--NeedCopy-->
```

GUI を使用して証明書のドメインチェックを無効にする

1. [トラフィック管理] **> [SSL] > [**証明書] に移動し、証明書を選択して [更新] をクリックします。
2. [ドメインチェックなし] を選択します。

ADC アプライアンスのデフォルト証明書を、アプライアンスのホスト名と一致する信頼できる **CA** 証明書に置き換えます

次の手順では、デフォルトの証明書 (ns-server-certificate) が内部サービスにバインドされていることを前提としています。

1. トラフィック管理 **> SSL > SSL** 証明書 **>** 証明書要求の作成に移動します。
2. [共通名] に、test.citrixadc.comと入力します。
3. CSR を信頼できる認証局に送信します。
4. 信頼できる CA から証明書を受け取ったら、このファイルを/nsconfig/sslディレクトリにコピーします。
5. [トラフィック管理] **> [SSL] > [**証明書] **> [**サーバー証明書] に移動します。
6. デフォルトのサーバ証明書 (ns-server-certificate) を選択し、[**Update**] をクリックします。
7. [証明書の更新] ダイアログボックスの [証明書ファイル名] で、署名後に CA から受け取った証明書を参照します。
8. [**Key File Name**] フィールドで、デフォルトのプライベートキーファイル名 (ns-server.key) を指定します。
9. [ドメインチェックなし] を選択します。
10. [**OK**] をクリックします。

有効期限モニターを有効にする

SSL 証明書は一定期間有効です。一般的な展開には、SSL トランザクションを処理する複数の仮想サーバーが含まれ、それらにバインドされた証明書は異なる時間に期限切れになる可能性があります。アプライアンスに設定された有効期限モニタは、設定された証明書の有効期限が切れると、アプライアンスの syslog および ns 監査ログにエントリを作成します。

証明書の期限切れに関する SNMP アラートを作成する場合は、別途設定する必要があります。

CLI を使用して証明書の有効期限モニターを有効にする

コマンドプロンプトで次のコマンドを入力して、証明書の有効期限モニターを有効にし、構成を確認します。

```
set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED ) [-
    notificationPeriod <positive_integer>]]

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

例:

```
set ssl certKey sv -expiryMonitor ENABLED  - notificationPeriod 60
Done
<!--NeedCopy-->
```

GUI を使用して証明書の有効期限モニターを有効にする

1. [トラフィック管理] **> [SSL] > [**証明書] に移動し、証明書を選択して [更新] をクリックします。
2. [有効期限が切れたら通知する] を選択し、必要に応じて通知期間を指定します。

リンクを解除せずに中間証明書を更新する

既存のリンクを壊すことなく中間証明書を更新できます。置き換えられる証明書によって発行されたリンクされた証明書の AuthorityKeyIdentifier 拡張には、認証局証明書のシリアル番号 (‘authorityCertSerialNumber’) フィールドを含めないでください。「AuthorityKeyIdentifier」拡張にシリアル番号フィールドが含まれる場合、古い証明書と新しい証明書のシリアル番号が同じである必要があります。上記の条件が満たされれば、リンク内の証明書をいくつでも 1 つずつ更新できます。以前は、中間証明書が更新されるとリンクが壊れました。

たとえばCertA、CertB、CertC、CertDの 4 つの証明書があります。証明書CertAはCertBの発行者、CertBはCertCの発行者、などです。リンクを解除せずに中間証明書CertBをCertB_newに置き換える場合は、次の条件を満たす必要があります。

次の両方の条件が満たされる場合、CertBの証明書シリアル番号はCertB_newの証明書シリアル番号と一致する必要があります。

- AuthorityKeyIdentifier拡張子はCertCにあります。
- この拡張機能にはシリアル番号フィールドが含まれます。

証明書内の共通名が変更された場合は、証明書の更新中にnodomainchecを指定します。

前述の例では、CertDの「www.example.com」を「*.example.com」に変更するには、「ドメインチェックなし」パラメータを選択します。

CLI を使用して証明書を更新する

コマンドプロンプトで入力します:

```
update ssl certKey <certkeyName> -cert <string> [-password] -key <
    string>  [-noDomainCheck]
<!--NeedCopy-->
```

例:

```
update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem -key /
    nsconfig/ssl/pkey.pem -noDomainCheck
<!--NeedCopy-->
```

証明書チェーンを表示する

証明書には、発行機関の名前と、証明書が発行されたサブジェクトが含まれます。証明書を検証するには、その証明書の発行者を調べて、その発行者を信頼しているかどうかを確認する必要があります。発行者を信頼できない場合は、発行者証明書の発行者を確認する必要があります。ルート CA 証明書または信頼できる発行者に到達するまで、チェーンを上に移動します。

SSL ハンドシェイクの一環として、クライアントが証明書を要求すると、アプライアンスは証明書と、アプライアンスに存在する発行者証明書のチェーンを提示します。管理者は、アプライアンスに存在する証明書の証明書チェーンを表示し、不足している証明書をインストールできます。

CLI を使用して、アプライアンスに存在する証明書の証明書チェーンを表示する

コマンドプロンプトで入力します:

```
show ssl certchain <cert_name>
<!--NeedCopy-->
```

例

c1、c2、および c3 の 3 つの証明書があります。証明書 c3 はルート CA 証明書で、c2 に署名し、c2 は c1 に署名します。次の例は、さまざまなシナリオでの
`show ssl certchain c1` コマンドの出力を示しています。

シナリオ **1:**

証明書 c2 は c1 にリンクされ、c3 は c2 にリンクされています。

証明書 c3 はルート CA 証明書です。

次のコマンドを実行すると、ルート CA 証明書までの証明書リンクが表示されます。


```
show ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate name: c2               linked; not a root
       certificate
    2) Certificate name: c3               linked; root certificate
Done
<!--NeedCopy-->
```

シナリオ **2:**

証明書 c2 は c1 にリンクされています。

証明書 c2 はルート CA 証明書ではありません。

次のコマンドを実行すると、証明書 c3 はルート CA 証明書であるが、c2 にリンクされていないという情報が表示されます。

```
show ssl certchain c1

Certificate chain  details of certificate name c1 are:
    1) Certificate Name: c2               linked; not a root
       certificate
    2) Certificate Name: c3               not linked; root certificate
    Done
<!--NeedCopy-->
```

シナリオ **3:**

証明書 c1、c2、および c3 はリンクされていませんが、アプライアンス上に存在します。

次のコマンドを実行すると、証明書 c1 の発行元から始まるすべての証明書に関する情報が表示されます。また、証明書がリンクされていないことも指定されています。

```
show ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate Name: c2               not linked; not a root
       certificate
    2) Certificate Name: c3               not linked; root certificate
    Done
<!--NeedCopy-->
```

シナリオ **4:**

証明書 c2 は c1 にリンクされています。

証明書 c3 はアプライアンスに存在しません。

次のコマンドを実行すると、c1 にリンクされた証明書に関する情報が表示されます。c2 で指定したサブジェクト名の証明書を追加するよう求められます。この場合、ユーザーはルート CA 証明書 c3 を追加するように求められます。


```
show ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate Name: c2               linked; not a root
       certificate
    2) Certificate Name: /C=IN/ST=ka/O=netscaler/CN=test
       Action: Add a certificate with this subject name.
Done
<!--NeedCopy-->
```

シナリオ **5:**

証明書が証明書 c1 にリンクされておらず、c1 の発行者証明書がアプライアンスに存在しません。

次のコマンドを実行すると、証明書 c1 にサブジェクト名を持つ証明書を追加するよう求められます。

```
sh ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate Name: /ST=KA/C=IN
       Action: Add a certificate with this subject name.
<!--NeedCopy-->
```

サーバーテスト証明書を生成する

August 15, 2023

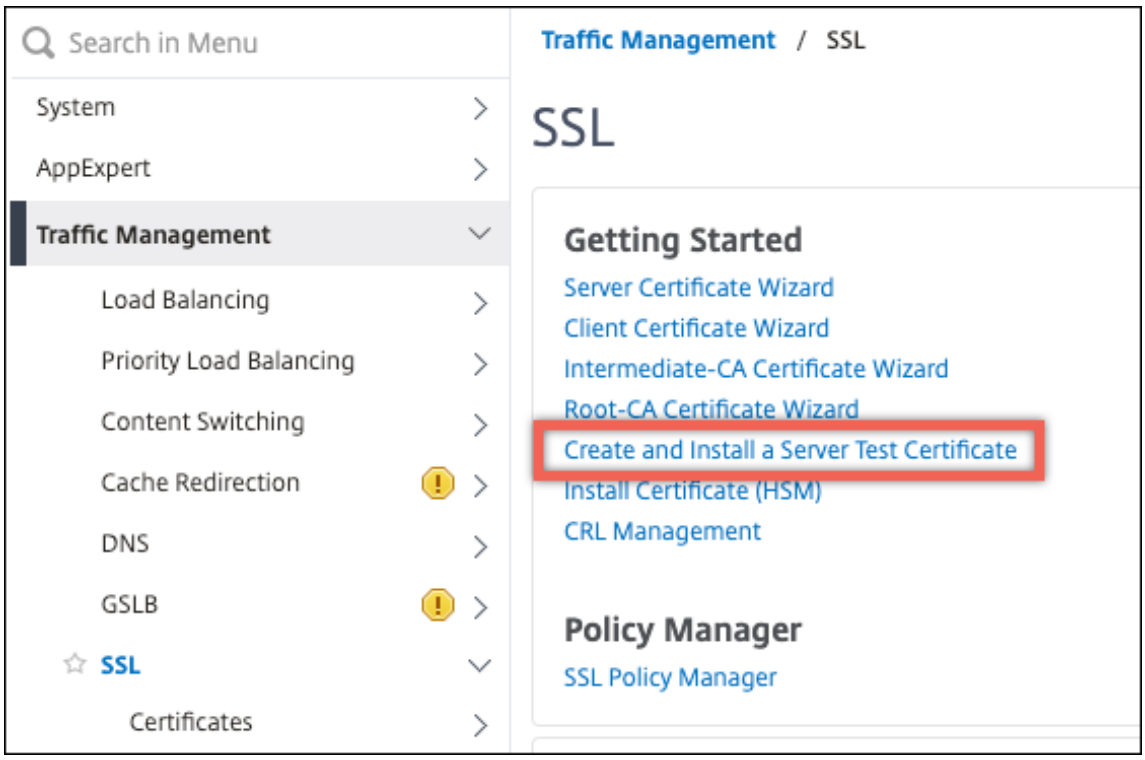
NetScaler アプライアンスでは、構成ユーティリティの GUI ウィザードを使用してサーバー認証用のテスト証明書を作成できます。サーバー証明書は、SSL ハンドシェイクでサーバーを認証および識別するために使用されます。通常、信頼できる CA がサーバー証明書を発行します。サーバーは証明書をクライアントに送信し、クライアントはそれを使用してサーバーを認証します。

サーバーテスト証明書を発行する場合、アプライアンスは認証局として機能します。この証明書を SSL 仮想サーバーにバインドして、クライアントとの SSL ハンドシェイクで認証できます。この証明書はテストのみを目的としています。実稼働環境では使用しないでください。

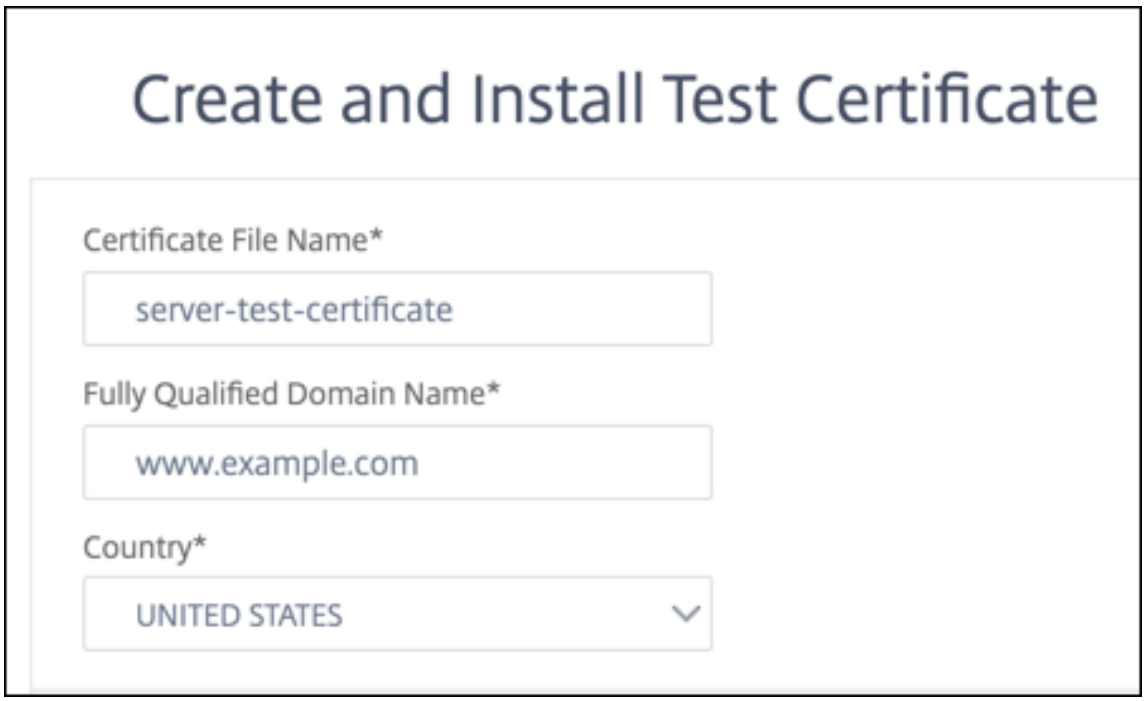
サーバーテスト証明書は、SSL または SSL_TCP プロトコルを使用する任意の仮想サーバーにインストールできます。

GUI を使用してサーバーテスト証明書を生成する

1. [トラフィック管理**] > [SSL**] に移動し、[**SSL** 証明書**]** グループで [サーバーテスト証明書の作成とインストール] を選択します。



2. パラメータの詳細を入力して、[作成] をクリックします。



SSL ファイルのインポートと変換

August 15, 2023

リモートホストへの FTP アクセスが利用できない場合でも、証明書、秘密キー、CRL、DH キーなどの SSL リソースをリモートホストからインポートできるようになりました。この機能は、リモートホストへのシェルアクセスが制限されている環境で特に役立ちます。デフォルトフォルダは /nsconfig/ssl に次のように作成されます。

- 証明書ファイルの場合:/nsconfig/ssl/certfile
- プライベートキーの場合:/nsconfig/ssl/キーファイル
- CRL の場合:/var/netscaler/ssl/crlfile
- DH キーの場合:/nsconfig/ssl/dhfile

HTTP サーバーと HTTPS サーバーの両方からのインポートがサポートされます。ただし、ファイルがアクセスにクライアント証明書認証を必要とする HTTPS サーバー上にある場合、インポートは失敗します。

> 注:
>
> 再起動後にファイルを再インポートするとエラーが発生する可能性があるため、import コマンドは構成 (ns.conf) ファイルに保存されません。

証明書ファイルをインポートする

CLI と GUI を使用して、リモートホストからファイル (リソース) をインポートできます。

CLI を使用してリモートホストから証明書ファイルをインポートする

コマンドプロンプトで入力します。

```
import ssl certFile [<name>] [<src>]
<!--NeedCopy-->
```

例:

```
import ssl certfile my-certfile http://www.example.com/file_1
<!--NeedCopy-->
```

```
show ssl certfile
      Name : my-certfile
      URL : http://www.example.com/file_1
<!--NeedCopy-->
```

証明書ファイルを削除するには、'name' `rm ssl certFile` 引数のみを受け入れるコマンドを使用します。

CLI を使用してリモートホストからキーファイルをインポートする

コマンドプロンプトで入力します。

```
import ssl keyFile [<name>] [<src>]
<!--NeedCopy-->
```

例:

```
import ssl keyfile my-keyfile http://www.example.com/key_file
<!--NeedCopy-->
```

```
show ssl keyfile
     Name : my-keyfile
     URL : http://www.example.com/key_file
<!--NeedCopy-->
```

キーファイルを削除するには、’name’ `rm ssl keyFile` 引数のみを受け入れるコマンドを使用します。

CLI を使用してリモートホストから **CRL** ファイルをインポートする

コマンドプロンプトで入力します。

```
import ssl crlFile [<name>] [<src>]
<!--NeedCopy-->
```

<name\ >CRL ファイルを削除するには、\ 引数のみを受け入れる`rm ssl crlFile`コマンドを使用します。

例:

```
import ssl crlfile my-crlfile http://www.example.com/crl_file

show ssl crlfile

    Name : my-crlfile
    URL : http://www.example.com/crl_file
<!--NeedCopy-->
```

CLI を使用してリモートホストから **DH** ファイルをインポートする

コマンドプロンプトで入力します。

```
import ssl dhFile [<name>] [<src>]
<!--NeedCopy-->
```

例:

```
import ssl dhfile my-dhfile http://www.example.com/dh_file
show ssl dhfile
```



```
      Name : my-dhfile
      URL : http://www.example.com/dh_file
<!--NeedCopy-->
```

<name\ >DH ファイルを削除するには、\ 引数のみを受け入れる`rm ssl dhFile`コマンドを使用します。

GUI を使用して **SSL** リソースをインポートする

[トラフィック管理**] > [SSL] > [**インポート] に移動し、適切なタブを選択します。

PKCS #8 証明書と **PKCS #12** 証明書をインポートする

ネットワーク内の他の安全なサーバーやアプリケーションで既に持っている証明書やキーを使用する場合は、それらをエクスポートしてから NetScaler アプライアンスにインポートできます。エクスポートした証明書とキーを NetScaler アプライアンスにインポートする前に、それらを変換する必要がある場合があります。

ネットワーク内の安全なサーバーまたはアプリケーションから証明書をエクスポートする方法の詳細については、エクスポート元のサーバーまたはアプリケーションのマニュアルを参照してください。

> 注:
>
> NetScaler アプライアンスにインストールする場合、キーと証明書の名前には、UNIX ファイルシステムでサポートされている文字以外のスペースや特殊文字を含めることはできません。エクスポートしたキーと証明書を保存するときは、適切な命名規則に従ってください。

証明書と秘密鍵のペアは、通常 PKCS #12 形式で送信されます。アプライアンスは、証明書とキーの PEM 形式と DER 形式をサポートしています。PKCS #12 を PEM または DER に、または PEM または DER を PKCS #12 に変換するには、このページの後半にある「インポートまたはエクスポート用の SSL 証明書の変換」セクションを参照してください。

NetScaler アプライアンスは、PKCS #8 形式の PEM キーをサポートしていません。ただし、CLI または設定ユーティリティからアクセスできる OpenSSL インターフェイスを使用して、これらのキーをサポートされている形式に変換できます。キーを変換する前に、プライベートキーが PKCS #8 形式であることを確認する必要があります。PKCS #8 形式のキーは通常、次のテキストで始まります。

```
-----BEGIN ENCRYPTED PRIVATE KEY-----


leuSSZQZKgrgUQ==


-----END ENCRYPTED PRIVATE KEY-----
<!--NeedCopy-->
```


CLI から **OpenSSL** インターフェイスを開きます

1. PuTTY などの SSH クライアントを使用して、アプライアンスへの SSH 接続を開きます。
2. 管理者の資格情報を使用して、アプライアンスにログオンします。
3. コマンドプロンプトで、shell と入力します。
4. シェルプロンプトで、「`openssl`」と入力します。

GUI から **OpenSSL** インターフェイスを開きます

[トラフィック管理**] > [SSL**] に移動し、[ツール] グループで [**OpenSSL** インターフェイス] を選択します。

OpenSSL インターフェイスを使用して、サポートされていない **PKCS #8** キー形式を暗号化されたサポートされているキー形式に変換する

OpenSSL プロンプトで、サポートされていないキー形式が RSA または ECDSA のどちらであるかに応じて、次のいずれかのコマンドを入力します。

```
OpenSSL>rsa- in <PKCS#8 Key Filename> -des3 -out <encrypted Key
    Filename>

OpenSSL>ec -in <PKCS#8 Key Filename> -des3 -out <encrypted Key Filename
    >
<!--NeedCopy-->
```

サポートされていないキー形式をサポートされているキー形式に変換するためのパラメータ

- **PKCS #8** キーファイル名**:** 互換性のない PKCS #8 プライベートキーの入力ファイル名。
- 暗号化鍵ファイル名**:** 互換性のある暗号化された秘密鍵の PEM 形式の出力ファイル名。
- 暗号化されていない鍵ファイル名**:** 互換性のある暗号化されていない秘密鍵の PEM 形式の出力ファイル名。

SSL 証明書をインポートまたはエクスポート用に変換

NetScaler アプライアンスは、SSL 証明書の PEM 形式と DER 形式をサポートしています。クライアントブラウザや一部の外部セキュアサーバーなどの他のアプリケーションでは、さまざまな公開鍵暗号化標準 (PKCS) 形式が必要です。アプライアンスは、証明書をアプライアンスにインポートするために PKCS #12 形式を PEM または DER 形式に変換し、証明書をエクスポートするために PEM または DER を PKCS #12 に変換できます。セキュリティを高めるために、インポート用のファイルの変換には、DES または DES3 アルゴリズムによる秘密鍵の暗号化を含めることができます。

> 注:
>
> GUI を使用して PKCS #12 証明書をインポートし、パスワードにドル記号 ($)、逆引用符 (‘)、またはエスケープ () 文字が含まれていると、インポートが失敗する可能性があります。その場合、「エラー: パスワードが無効です」というメッセージが表示されます。パスワードに特殊文字を使用する必要がある場合は、CLI を使用してすべてのインポートを実行しない限り、必ずエスケープ文字 () の前にエスケープ文字 () を付けてください。

CLI を使用して証明書の形式を変換する

コマンドプロンプトで、次のコマンドを入力します。

```
convert ssl pkcs12 <outfile> [-import [-pkcs12File <inputFilename>] [-des | -des3] [-export [-certFile <inputFilename>] [-keyFile <inputFilename>]]
<!--NeedCopy-->
```

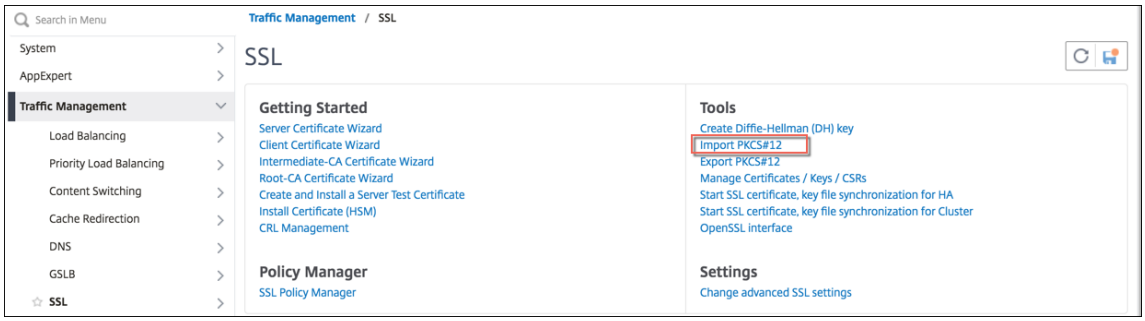
操作中に、インポートパスワードまたはエクスポートパスワードの入力を求められます。暗号化されたファイルの場合は、パスフレーズの入力も求められます。

例**:**

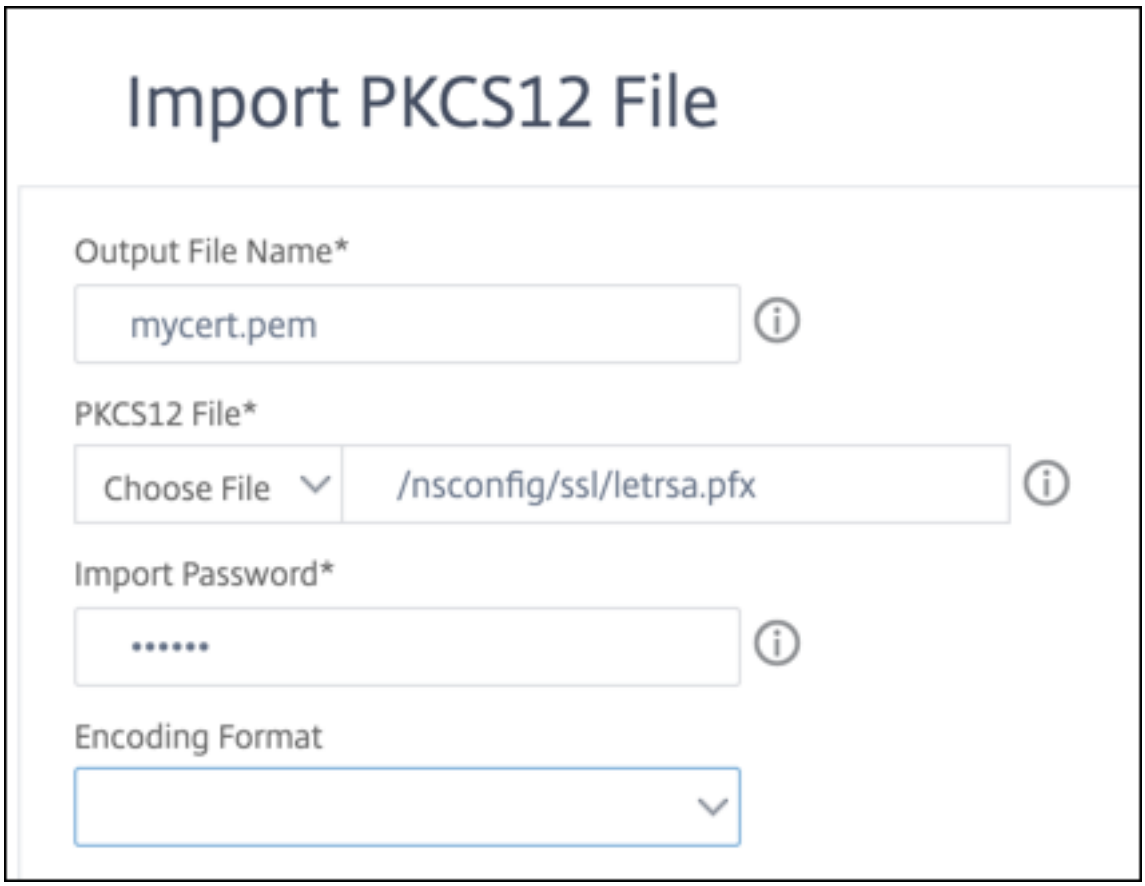
```
convert ssl pkcs12 Cert-Import-1.pem -import -pkcs12File Cert-Import-1.pfx -des

convert ssl pkcs12 Cert-Client-1.pfx -export -certFile Cert-Client-1 -keyFile Key-Client-1
<!--NeedCopy-->
```

GUI を使用して証明書の形式を変換する

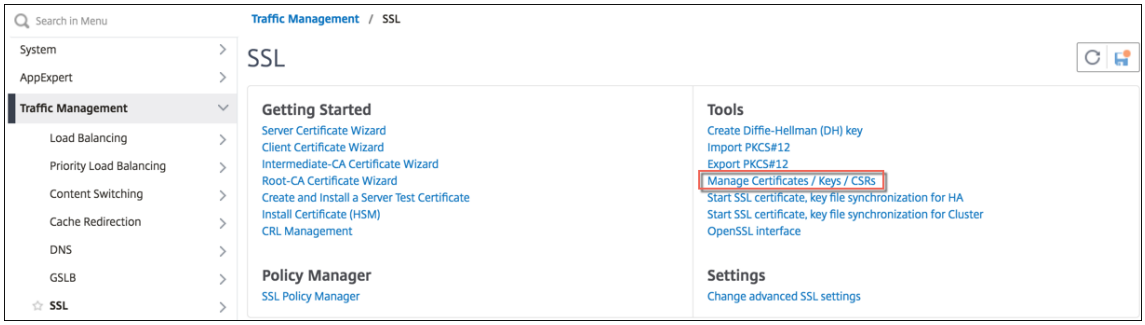
1. [トラフィック管理**] > [SSL**] に移動し、[ツール] グループで [**PKCS #12** のインポート] を選択します。



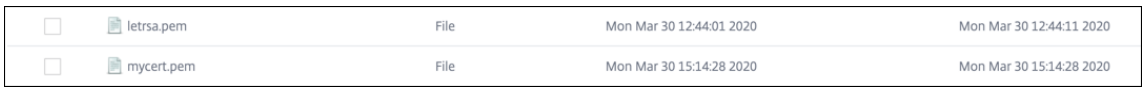
2. 「出力ファイル名」フィールドに **PEM** 証明書名を指定します。
3. ローカルコンピュータまたはアプライアンス上の PFX 証明書の場所を参照します。



4. [**OK**] をクリックします。
5. [証明書の管理]/[キー]/[**CSR**] をクリックして、変換された PEM ファイルを表示します。

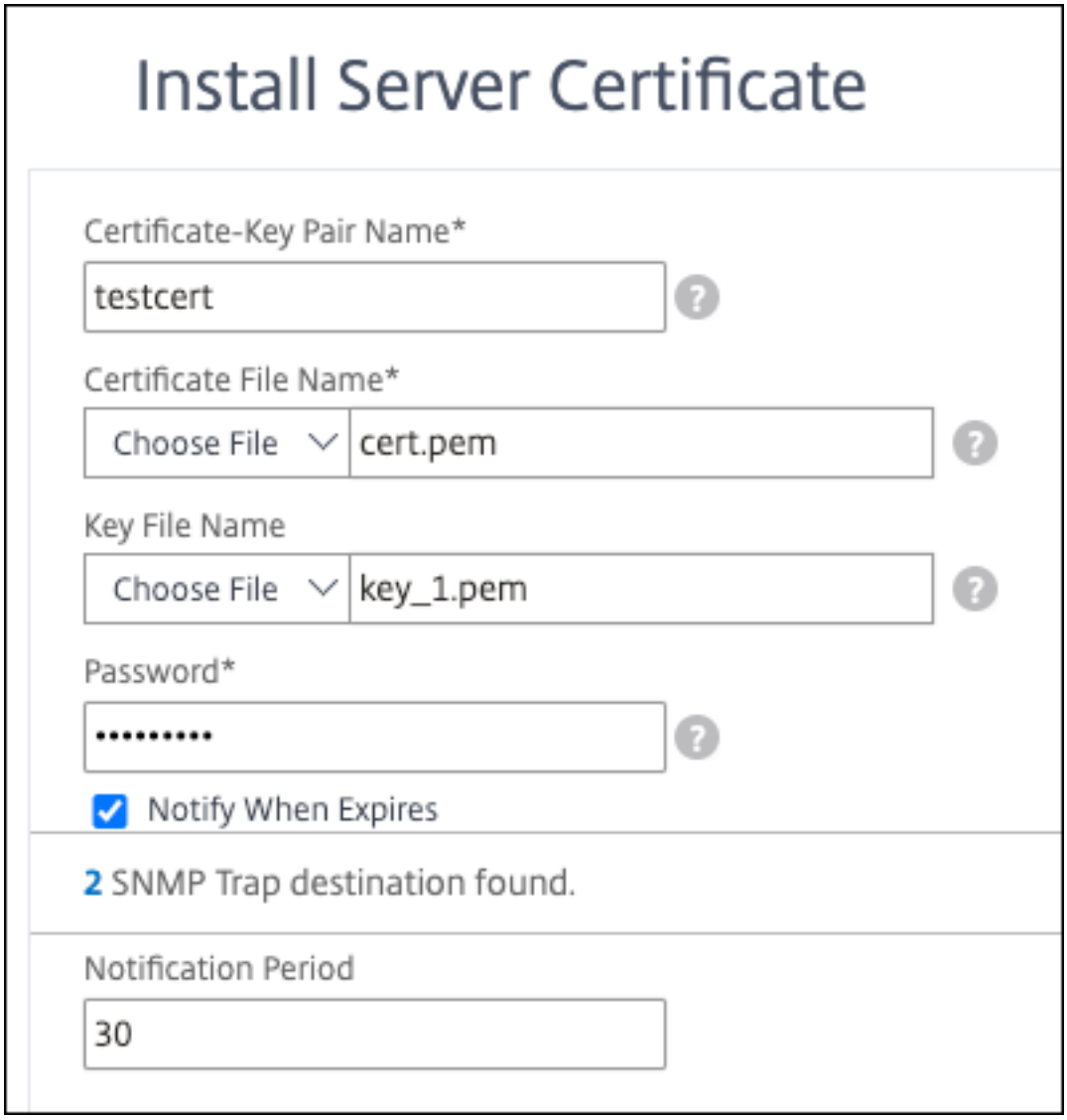


6. アップロードされた PFX ファイルと変換された PEM ファイルを表示できます。



7. [**SSL**] > [証明書] > [サーバー証明書] に移動し、[インストール] をクリックします。
8. 証明書とキーのペア名を指定します。
9. PEM ファイルの場所を参照します。
10. プロンプトが表示されたら、パスワードを指定します。

11. [**Install**] をクリックします。



12. 証明書とキーのペアを SSL 仮想サーバーにバインドします。

SSL 証明書を NetScaler ADC アプライアンスの仮想サーバーにバインドする

August 15, 2023

SSL 証明書は、SSL 暗号化および復号化プロセスの重要な部分です。証明書は、SSL ハンドシェイク中に使用され、SSL サーバー（クライアントの SSL ターミネーションポイントとして機能する NetScaler ADC アプライアンス）の ID を確立します。

SSL トランザクションの処理に使用される証明書は、SSL データを受信する仮想サーバー (SSL) にバインドする必

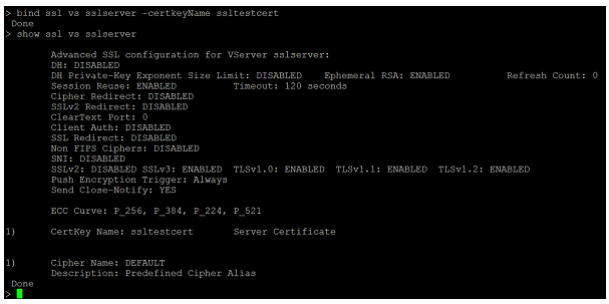
要があります。

コマンドラインインターフェイスを使用して **SSL** 証明書を **SSL** 仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind ssl vs <vServerName> -certkeyName <certificate-KeyPairName>
show ssl vs <vServerName>
<!--NeedCopy-->
```

例**:**



GUI を使用して **SSL** 証明書を **SSL** 仮想サーバーにバインドするには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. SSL タイプの仮想サーバを選択し、[編集（Edit）] をクリックします。
3. [負荷分散仮想サーバー] ページの [証明書] セクションで、[サーバー証明書なし] をクリックします。

4. [サーバー証明書のバインド] ページで、[クリックして選択] をクリックします。
5. SSL 証明書を選択し、[選択] をクリックします。
6. [バインド] をクリックして、SSL 証明書を仮想サーバーにバインドします。
7. [完了] をクリックします。

   SSL 証明書を仮想サーバにバインドする作業が完了しました。

> 注:
>
> すでに証明書とキーのペアがバインドされている仮想サーバーに証明書とキーのペアをバインドしようとすると、NetScaler は古い証明書キーをバインド解除して新しい証明書キーをバインドします。次のメッセージが表示されます。

> `Warning: Current certificate replaces the previous binding`
>
> ハンドシェイクが完了した既存の接続は影響を受けません。他の接続は終了します。

SSL プロファイル

August 15, 2023

SSL プロファイルを使用して、NetScaler ADC アプライアンスが SSL トラフィックを処理する方法を指定できます。プロファイルは、仮想サーバー、サービス、サービスグループなどの SSL エンティティの SSL パラメーター設定の集まりであり、構成が簡単で柔軟性があります。設定できるグローバルパラメータは 1 セットだけに限定されません。

グローバルパラメータの複数のセット (プロファイル) を作成し、異なる SSL エンティティに異なるセットを割り当てることができます。SSL プロファイルは次の 2 つのカテゴリに分類されます。

- フロントエンドプロファイル: フロントエンドエンティティ (クライアントからリクエストを受け取るエンティティ) に適用されるパラメーターが含まれます。
- バックエンドプロファイル: バックエンドエンティティ (クライアントリクエストをサーバーに送信するエンティティ) に適用されるパラメーターが含まれます。

TCP や HTTP プロファイルとは異なり、SSL プロファイルはオプションです。SSL プロファイルを有効にすると、すべての SSL エンドポイントがデフォルトプロファイルを継承します。同じプロファイルを複数のエンティティで再利用できます。エンティティにプロファイルが添付されていない場合は、グローバルレベルで設定された値が適用されます。動的に学習されるサービスには、現在のグローバル値が適用されます。

個々の SSL エンドポイントで SSL パラメータ、暗号、および ECC 曲線を構成する必要がある代替方法と比較して、NetScaler ADC アプライアンスの SSL プロファイルは、関連するすべてのエンドポイントの SSL 構成の単一ポイントとして機能するため、構成管理が簡素化されます。SSL プロファイルを使用すると、暗号の順序変更や暗号の順序変更時のダウンタイムに関連する設定上の問題を解決できます。

SSL プロファイルは、従来これらのパラメータやバインディングを設定できない SSL エンドポイントに必要な SSL パラメータと暗号バインディングを設定するのに役立ちます。SSL プロファイルは安全なモニターにも設定できます。

SSL プロファイルインフラストラクチャは、最新の暗号とプロトコルを使用するように強化されました。レガシープロファイル (古いプロファイル) と拡張 SSL プロファイル (新しいプロファイル) の違いが強調表示されます。

古い **SSL** プロファイルインフラストラクチャと新しい **SSL** プロファイルインフラストラクチャの違い

相違点	古いプロフィール	新しいプロフィール
プロファイルに含まれる暗号と ECC カーブ	いいえ	はい
既存のリストの中間への暗号または暗号グループの挿入	すべての暗号をバインド解除し、必要な優先度の順序で再度バインドします。	暗号を追加して優先順位を割り当てます。優先度が指定されていない場合、暗号にはリストで最も低い優先度が割り当てられます。
すべての暗号のバインド解除	`unbind ssl vserver <name> ciphername -ALL`	`unbind ssl profile -cipherName FlushAllCiphers`（リリース 12.1 以降では、ALL は暗号グループと同様に扱われるため、すべての暗号または暗号グループをプロファイルからバインド解除するための flushAllCiphers パラメータが含まれています。）
SSLv3 の状態	なし	デフォルトのフロントエンドプロファイル (ns_default_ssl_profile_frontend) では無効になっています。注: このプロファイルを有効にする前は、SSLv3 がグローバルに有効になっています。プロファイルを有効にすると、フロントエンドのデフォルトプロファイルで SSLv3 が無効になります。

SSL プロファイルインフラストラクチャ

August 15, 2023

SSLv3 と RC4 実装の脆弱性により、ネットワーク接続のセキュリティ設定をネゴシエートするために最新の暗号とプロトコルを使用する必要性が強調されています。数千の SSL エンドポイントで SSLv3 を無効にするなど、設定の変更を実装するのは面倒なプロセスです。そのため、SSL エンドポイント設定の一部であった設定は、デフォルトの暗号とともに SSL プロファイルに移動されました。暗号のサポートを含む設定の変更を実装するには、エンティティにバインドされているプロファイルを変更するだけで済みます。

デフォルトのフロントエンドおよびデフォルトのバックエンド SSL プロファイルには、古いプロファイルに含まれていた設定に加えて、デフォルトの暗号と ECC 曲線がすべて含まれています。デフォルトプロファイルの出力例は付録に記載されています。[Enable Default Profile] 操作により、デフォルトのフロントエンドプロファイルがすべてのフロントエンドエンティティに自動的にバインドされ、デフォルトのバックエンドプロファイルがすべてのバックエンドエンティティに自動的にバインドされます。デフォルトのプロファイルは、導入環境に合わせて変更できます。また、カスタムプロファイルを作成して SSL エンティティにバインドすることもできます。

フロントエンドプロファイルには、フロントエンドエンティティ (クライアントからリクエストを受け取るエンティティ) に適用されるパラメーターが含まれています。通常、このエンティティは SSL 仮想サーバー、透過 SSL サービス、または NetScaler ADC アプライアンス上の内部サービスです。バックエンドプロファイルには、バックエンドエンティティ（クライアント要求をバックエンドサーバーに送信する ADC アプライアンス上のエンティティ）に適用されるパラメーターが含まれています。通常、このエンティティは NetScaler ADC アプライアンス上の SSL サービスまたはサービスグループです。サポートされていないパラメータを設定しようとすると、エラー`ERROR: Specified parameters are not applicable for this type of SSL profile`が表示されます。CRL メモリサイズ、OCSP キャッシュサイズ、デファクション解除制御、およびデファクション解除データなど、一部の SSL パラメータはエンティティに依存しないため、どのプロファイルにも含まれません。これらのパラメータは、[トラフィック管理] **> [SSL] >** [高度な **SSL** 設定] にあります。セキュア・モニターでサポートされている SSL パラメーターの詳細については、「セキュア・モニターでの SSL パラメーターの設定」を参照してください。

SSL プロファイルは次の操作をサポートします。

- 追加:NetScaler アプライアンスに SSL プロファイルを作成します。プロファイルがフロントエンドかバックエンドかを指定します。デフォルトはフロントエンドです。
- 設定:—既存のプロファイルの設定を変更します。
- **Unset**: 指定されたパラメータをデフォルト値に設定します。パラメータを何も指定しない場合、エラーメッセージが表示されます。エンティティのプロファイルを設定解除すると、プロファイルはエンティティからバインド解除されます。
- 削除: プロファイルを削除します。どのエンティティでも使用されているプロファイルは削除できません。設定をクリアすると、すべてのエンティティが削除されます。その結果、プロファイルも削除されます。
- バインド**:** プロファイルを **SSL** エンティティにバインドします。
- バインド解除:SSL エンティティからプロファイルをバインド解除します。
- 表示:NetScaler アプライアンスで使用可能なすべてのプロファイルが表示されます。プロファイル名を指定すると、そのプロファイルの詳細が表示されます。エンティティを指定すると、そのエンティティに関連付けられているプロファイルが表示されます。

> 重要:
>
> - SSL プロファイルは SSL パラメータよりも優先されます。つまり、`set ssl parameter` コマンドを使用して SSL パラメータを設定し、後でプロファイルを SSL エンティティにバインドすると、プロファイル内の設定が優先されます。

> - アップグレード後、デフォルトプロファイルを有効にすると、変更を元に戻すことはできません。つまり、プロファイルは無効にできません。プロファイルを有効にする前に、構成を保存し、構成ファイル (ns.conf) のコピーを作成します。ただし、デフォルトプロファイルの機能を使用しない場合は、引き続き古い SSL プロファイルを使用できます。これらのプロファイルの詳細については、レガシー SSL プロファイルを参照してください。
> - リリース 11.1 51.x 以降、GUI および CLI で、デフォルトプロファイルを有効にすると、誤って有効にならないように確認プロンプトが追加されます。

リリース 13.1 ビルド 17.x から、TLSv1.2 より前のプロトコルは SSL 内部サービスでは無効になっています。デフォルト (拡張) プロファイルが有効になっている場合、`ns_default_ssl_profile_internal_frontend_service` プロファイルは SSL 内部サービスにバインドされ、SSLv3、TLSv1.0、および TLSv1.1 プロトコルはプロファイルで無効になります。

コマンド:

```
set ssl parameter -defaultProfile ENABLED
    Save your configuration before enabling the Default profile. You
       cannot undo the changes. Are you sure you want to enable the
       Default profile? [Y/N]Y
Done
<!--NeedCopy-->
```

デフォルトでは、グローバルパラメータと呼ばれる一部の *SSL* パラメータがすべての SSL エンドポイントに適用されます。ただし、プロファイルが SSL エンドポイントにバインドされている場合、グローバルパラメータは適用されません。プロファイルで指定されている設定が代わりに適用されます。

注意事項

1. プロファイルは複数の仮想サーバにバインドできますが、1 つの仮想サーバにバインドできるプロファイルは 1 つだけです。
2. 仮想サーバにバインドされているプロファイルを削除するには、まずプロファイルのバインドを解除します。
3. 1 つの暗号または暗号グループは、異なる優先順位で複数のプロファイルにバインドできます。
4. プロファイルには、異なる優先順位でバインドされた複数の暗号および暗号グループを含めることができます。
5. 暗号グループへの変更は、すべてのプロファイルと、プロファイルの 1 つがバインドされているすべての仮想サーバに即座に反映されます。
6. 暗号スイートが暗号グループの一部である場合は、プロファイルから暗号スイートを削除する前に、暗号グループを編集してその暗号スイートを削除します。
7. プロファイルにアタッチされた暗号スイートまたは暗号グループにプライオリティを割り当てなかった場合、プロファイル内で最も低いプライオリティが割り当てられます。
8. 既存の暗号グループと暗号スイートから、カスタム暗号グループ (ユーザー定義暗号グループとも呼ばれる) を作成できます。暗号グループ A を作成し、既存の暗号グループ X と Y をこの順序で追加すると、Y は X よりも低い優先度で割り当てられます。つまり、最初に追加されたグループの方が優先度が高くなります。

9. 暗号スイートが同じプロファイルにアタッチされた 2 つの暗号グループの一部である場合、その暗号スイートは 2 番目の暗号グループの一部として追加されません。プライオリティの高い暗号スイートは、トラフィックが処理されるときに有効になります。
10. 暗号グループはプロファイル内で展開されません。その結果、設定ファイル (ns.conf) の行数が大幅に削減されます。たとえば、それぞれ 15 個の暗号を含む 2 つの暗号グループが 1,000 台の SSL 仮想サーバーにバインドされている場合、拡張により、構成ファイルに 30*1000 個の暗号関連エントリが追加されます。新しいプロファイルでは、プロファイルにバインドされた暗号グループごとに 1 つずつ、エントリが 2 つだけになります。
11. 既存の暗号と暗号グループからユーザー定義の暗号グループを作成するのは、コピーアンドペースト操作です。元のグループに加えた変更は、新しいグループには反映されません。
12. ユーザー定義の暗号グループには、そのグループが属するすべてのプロファイルが一覧表示されます。
13. プロファイルには、バインドされているすべての SSL 仮想サーバ、サービス、およびサービスグループが一覧表示されます。
14. デフォルトの SSL プロファイル機能が有効になっている場合は、このプロファイルを使用して SSL エンティティの任意の属性を設定または変更します。たとえば、仮想サーバ、サービス、サービスグループ、内部サービスなどです。

CLI を使用して設定を保存する

コマンドプロンプトで入力します。

```
save config

shell

root@ns# cd /nsconfig

root@ns# cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber
    >
<!--NeedCopy-->
```

例:

```
save config
shell
root@ns# cd /nsconfig
root@ns# cp ns.conf ns.conf.NS.11.0.jun.16
<!--NeedCopy-->
```

既定のプロファイルを有効にする

> 重要:
>
> - ソフトウェアをアップグレードする前に設定を保存し、デフォルトプロファイルを有効にします。

> - リリース 11.1 ビルド 51.x から、GUI および CLI では、デフォルトプロファイルを有効にすると、誤って有効にしないように、確認プロンプトが表示されます。

コマンド**:** 次のコマンドは、デフォルトプロファイルを有効にし、プロファイルが既にバインドされている SSL エンティティにこのプロファイルをバインドします。つまり、プロファイル (P1 など) がすでに SSL エンティティにバインドされている場合、P1 は既定のフロントエンドプロファイルまたは既定のバックエンドプロファイルに置き換えられます。古いプロファイル (P1) は削除されません。このプロファイルは拡張 SSL プロファイルになり、以前の設定、暗号、ECC 曲線が含まれています。デフォルトのプロファイルを使用しない場合は、P1 を SSL エンティティに明示的にバインドできます。

```
set ssl parameter -defaultProfile ENABLED
    Save your configuration before enabling the Default profile. You
       cannot undo the changes. Are you sure you want to enable the
       Default profile? [Y/N]Y
Done
<!--NeedCopy-->
```

拡張プロファイルインフラストラクチャをサポートするビルドにソフトウェアをアップグレードし、既定のプロファイルを有効にします。

> メモ:
>
> - レガシープロファイル (P1) がすでに SSL エンティティにバインドされている場合に、デフォルトプロファイルを有効にすると、以前のバインドがデフォルトプロファイルによって上書きされます。つまり、デフォルトプロファイルは SSL エンティティにバインドされます。デフォルトプロファイルをバインドしない場合は、P1 を SSL エンティティに再度バインドする必要があります。
> - 1 回の操作 ([Enable Default Profile or `set ssl parameter -defaultProfile ENABLED`]) で、既定のフロントエンドプロファイルと既定のバックエンドプロファイルの両方を有効化 (バインド) できます。

デフォルトプロファイルの一部であるパラメータ

次のコマンドを実行して、デフォルトのフロントエンドプロファイルとバックエンドプロファイルに含まれるパラメーターを一覧表示します。

```
sh ssl profile ns_default_ssl_profile_frontend
sh ssl profile ns_default_ssl_profile_backend
<!--NeedCopy-->
```

例**:**

```
> sh ssl profile ns_default_ssl_profile_frontend
1)  Name: ns_default_ssl_profile_frontend   (Front-End)
    SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
```



```
    Use only bound CA certificates: DISABLED
    Strict CA checks:   NO
    Session Reuse: ENABLED  Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED  Refresh Count: 0
    Deny SSL Renegotiation  ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:    1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:  NO
    Match HTTP Host header with SNI:    CERT
    Push flag:  0x0 (Auto)
    SSL quantum size:   8 kB
    Encryption trigger timeout  100 mS
    Encryption trigger packet count:    45
    Subject/Issuer Name Insertion Format:   Unicode

    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server: 10
    Session Ticket: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0
    HSTS Preload: NO
    Allow Extended Master Secret: NO
    Send ALPN Protocol: NONE


    ECC Curve: P_256, P_384, P_224, P_521

1)  Cipher Name: DEFAULT    Priority :1
    Description: Predefined Cipher Alias


> sh ssl profile ns_default_ssl_profile_backend
1)  Name: ns_default_ssl_profile_backend    (Back-End)
    SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Server Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:   NO
```

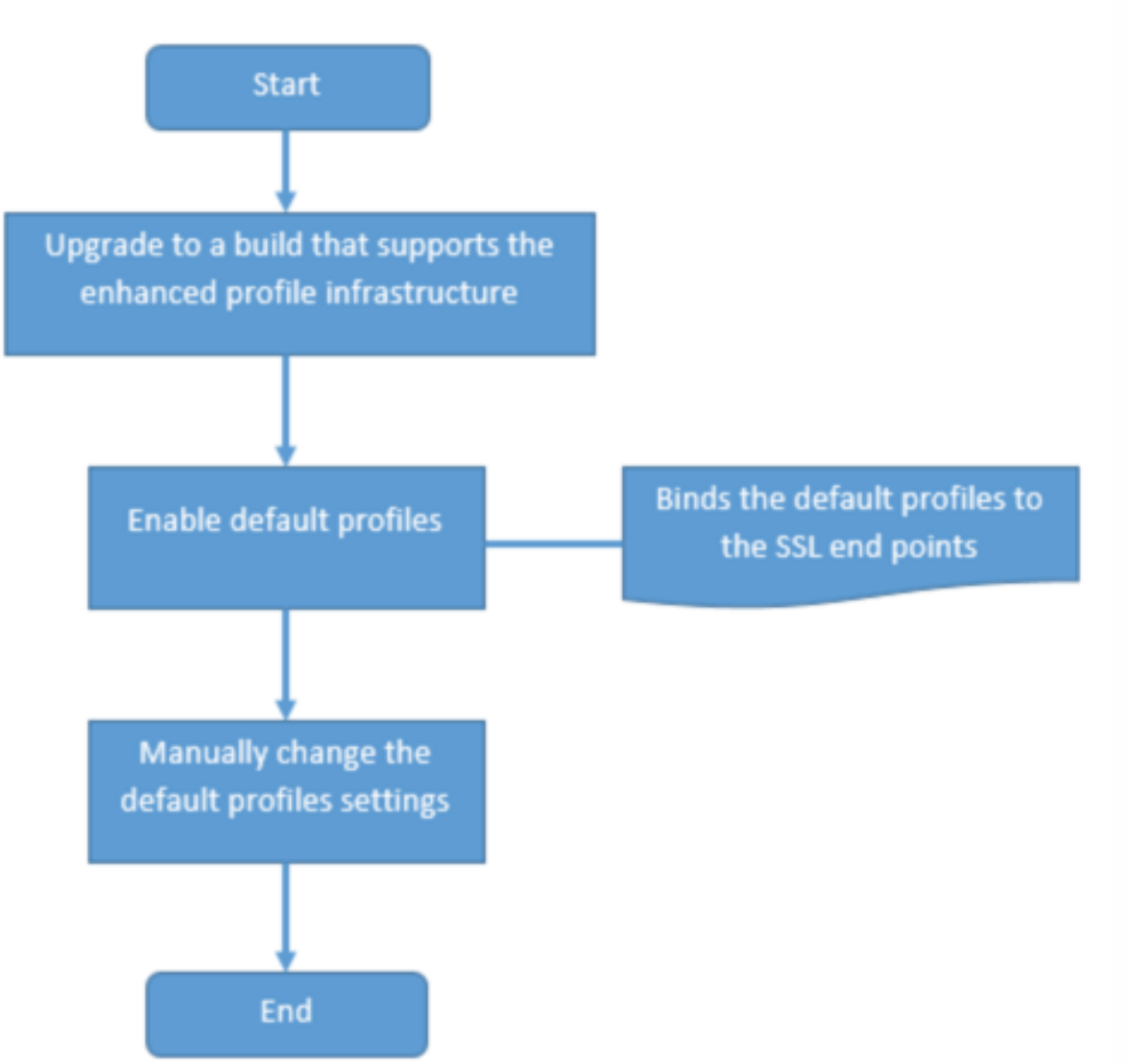


```
    Session Reuse: ENABLED  Timeout: 300 seconds
    DH: DISABLED
    Ephemeral RSA: DISABLED
    Deny SSL Renegotiation  ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:    1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:  NO
    Push flag:  0x0 (Auto)
    SSL quantum size:   8 kB
    Encryption trigger timeout  100 mS
    Encryption trigger packet count:    45

    Allow Extended Master Secret: NO

    ECC Curve: P_256, P_384, P_224, P_521

1)  Cipher Name: DEFAULT_BACKEND    Priority :1
    Description: Predefined Cipher Alias
 Done
 <!--NeedCopy-->
```

使用例

デフォルトプロファイルを有効にすると、すべての SSL エンドポイントにバインドされます。既定のプロファイルは編集可能です。デプロイメントでほとんどのデフォルト設定が使用され、一部のパラメータしか変更されない場合は、デフォルトプロファイルを編集できます。変更内容は、すべてのエンドポイントに即座に反映されます。また、カスタムパラメータとデフォルトパラメータを使用してカスタム SSL プロファイルを作成し、SSL エンティティにバインドすることもできます。

次のフローチャートは、実行する必要のある手順を説明しています。



1. ソフトウェアのアップグレードについては、「システムソフトウェアのアップグレード」を参照してください。
2. CLI または GUI を使用して、デフォルトプロファイルを有効にします。
   - コマンドラインで、次を入力します: `set ssl parameter -defaultProfile ENABLED`。
   - GUI を使用する場合は、[トラフィック管理] > [SSL] > [**SSL** の詳細設定の変更] に移動し、下にスクロールして [デフォルトプロファイルを有効にする] を選択します。

アップグレード前にプロファイルがエンドポイントにバインドされていなかった場合、デフォルトのプロファイルが SSL エンドポイントにバインドされます。アップグレード前にプロファイルがエンドポイントにバインドされていた場合、アップグレード後に同じプロファイルがバインドされ、デフォルトの暗号がプロファイルに追加されます。

1. (オプション) デフォルトプロファイルの設定を手動で変更します。

- コマンドラインで、: `set ssl profile <name>` に続けて変更するパラメータを入力します。
- GUI を使用したい場合は、[システム] > [プロファイル] に移動します。**SSL** プロファイルで、プロファイルを選択して [編集] をクリックします。

SSL プロファイルパラメータ

SSL プロファイルでは、次の SSL パラメータを設定できます。これらのパラメータの一部は、SSL 仮想サーバーで設定できます。SSL 仮想サーバーパラメーターの詳細については、「SSL 仮想サーバーパラメータ」を参照してください。

NetScaler アプライアンスのバックエンドでの安全な再ネゴシエーションのサポート

注: このパラメータは、リリース 13.0 ビルド 58.x 以降で導入されました。以前のリリースおよびビルドでは、非セキュアな再ネゴシエーションのみがバックエンドでサポートされていました。

この機能は、次のプラットフォームでサポートされています。

- VPX
- N2 または N3 チップを含む MPX プラットフォーム
- Intel Coleto SSL チップベースのプラットフォーム

この機能は FIPS プラットフォームではまだサポートされていません。

ADC アプライアンスのバックエンドでは、セキュアな再ネゴシエーションはデフォルトで拒否されます。つまり、denySSLReneg パラメータは ALL (デフォルト) に設定されます。

バックエンドでセキュアな再ネゴシエーションを許可するには、次のいずれかのdenySSLRenegパラメータ設定を選択します。

- いいえ
- FRONTEND_CLIENT
- FRONTEND_CLIENTSERVER
- NONSECURE

CLI を使用してセキュアな再ネゴシエーションを有効にする

コマンドプロンプトで入力します。

```
set ssl profile <name> -denySSLReneg <denySSLReneg>
```

例:

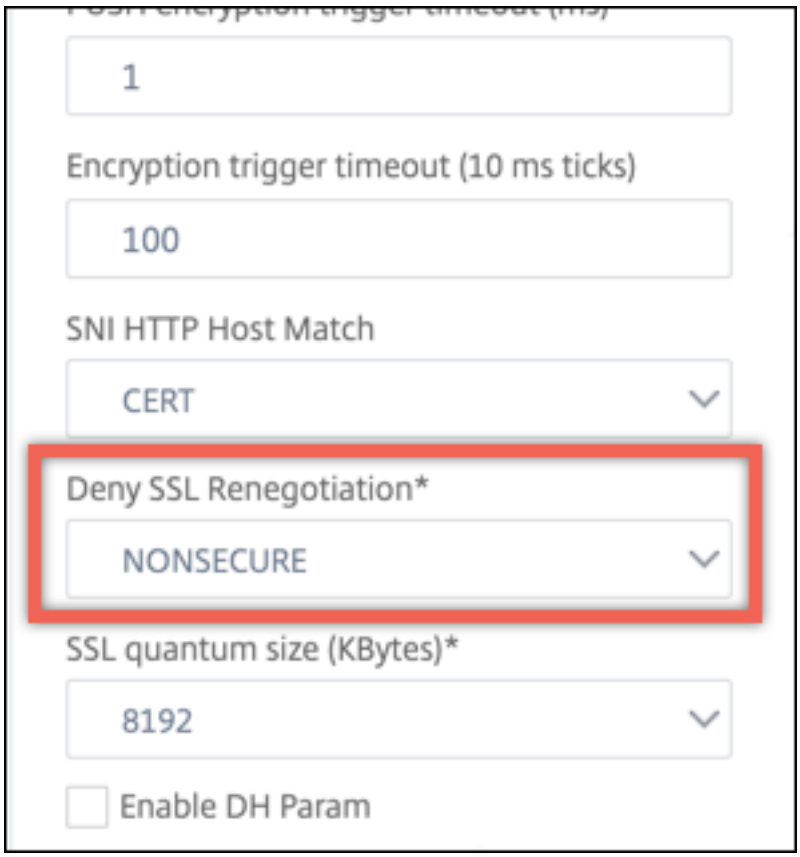
```
1 set ssl profile ns_default_ssl_profile_backend -denySSLReneg NONSECURE
2  Done
```



```
3
4  sh ssl profile ns_default_ssl_profile_backend
5  1)  Name: ns_default_ssl_profile_backend    (Back-End)
6      SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
           ENABLED  TLSv1.3: DISABLED
7      Server Auth: DISABLED
8      Use only bound CA certificates: DISABLED
9      Strict CA checks:       NO
10     Session Reuse: ENABLED      Timeout: 300 seconds
11     DH: DISABLED
12     Ephemeral RSA: DISABLED
13     Deny SSL Renegotiation      NONSECURE
14     Non FIPS Ciphers: DISABLED
15     Cipher Redirect: DISABLED
16     SSL Redirect: DISABLED
17     Send Close-Notify: YES
18     Strict Sig-Digest Check: DISABLED
19     Push Encryption Trigger: Always
20     PUSH encryption trigger timeout:    1 ms
21     SNI: DISABLED
22     OCSP Stapling: DISABLED
23     Strict Host Header check for SNI enabled SSL sessions:      NO
24     Push flag:  0x0 (Auto)
25     SSL quantum size:       8 kB
26     Encryption trigger timeout  100 mS
27     Encryption trigger packet count:    45
28
29     ECC Curve: P_256, P_384, P_224, P_521
30
31 1)  Cipher Name: DEFAULT_BACKEND        Priority :2
32     Description: Predefined Cipher Alias
33
34 1)  Service Name: s187
35  Done
36 <!--NeedCopy-->
```

GUI を使用してセキュアな再ネゴシエーションを有効にする

1. **[**システム**] > [**プロファイル**] > [SSL** プロファイル**]** に移動します。
2. プロファイルを追加または編集します。
3. [**SSL** 再ネゴシエーションの拒否] を ALL 以外の値に設定します。



ホストヘッダ検証

注**:** このパラメーターは、リリース 13.0 ビルド 52.x で導入されました。

HTTP/1.1 では、クライアントは複数のリクエストを処理するために複数の接続を使用する必要がありました。HTTP/2 を使用すると、クライアントは同じ証明書の対象となるドメイン間で接続を再利用できます。SNI 対応セッションの場合、ADC アプライアンスは、この変更に対応するために HTTP ホストヘッダーの検証方法を制御できる必要があります。以前のビルドでは、パラメータが有効 (「Yes」に設定) され、リクエストに SNI 対応セッションのホストヘッダーが含まれていない場合、リクエストはドロップされていました。パラメータが無効になっている (「No」に設定されている) 場合、アプライアンスは検証を実行しませんでした。この検証をより適切に制御できるように、SSL プロファイルと SSL グローバルパラメータに新しいパラメータ`SNIHTTPHostMatch`が追加されました。このパラメータには、CERT、STRICT、NONE の 3 つの値を指定できます。これらの値は、SNI が有効なセッションでのみ次のように機能します。SSL 仮想サーバまたは仮想サーバにバインドされたプロファイルで SNI を有効にする必要があり、HTTP 要求にはホストヘッダーが含まれている必要があります。

- CERT-要求内のホストヘッダー値が、この SSL セッションの確立に使用された証明書によってカバーされている場合、接続が転送されます。
- STRICT-要求内のホストヘッダー値が、SSL 接続の Client Hello メッセージで渡されたサーバ名の値と一致する場合にのみ、接続が転送されます。
- NO-ホストヘッダー値は検証されません。

指定可能な値: いいえ、CERT、STRICT
デフォルト値:CERT

新しいパラメーターSNIHTTPHostMatchの導入により、dropReqWithNoHostHeaderパラメーターの動作が変更されました。dropReqWithNoHostHeaderパラメータの設定は、SNI 証明書に対するホストヘッダーの検証方法に影響しなくなりました。

CLI を使用して SSL プロファイルパラメータを設定する

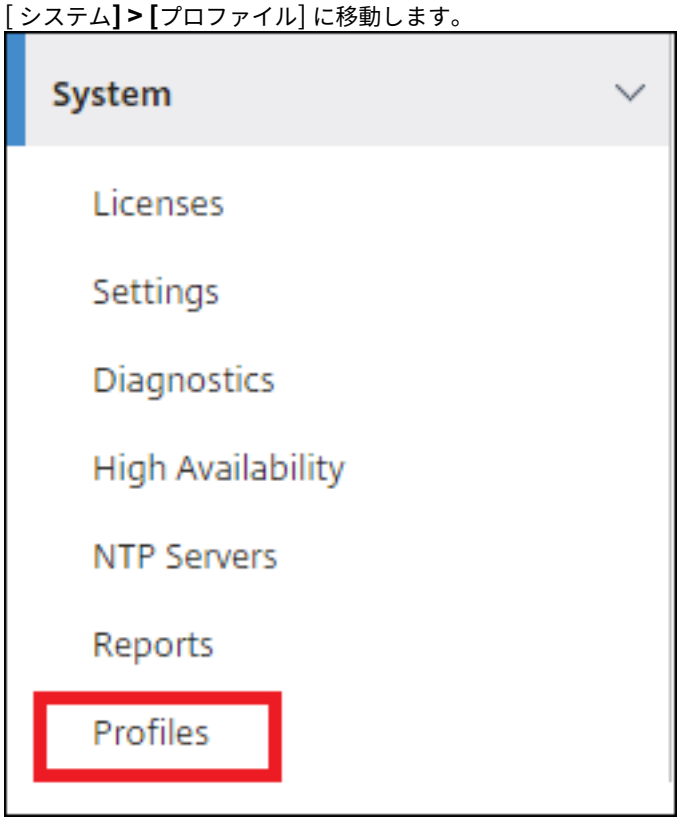
コマンドプロンプトで入力します。

```
set ssl profile <name> [-ssllogProfile <string>] [-dh ( ENABLED | DISABLED ) -dhFile <string>] [-dhCount <positive_integer>][-dhKeyExpSizeLimit ( ENABLED | DISABLED )] [-eRSA ( ENABLED | DISABLED) [-eRSACount <positive_integer>]] [-sessReuse ( ENABLED | DISABLED )
[-sessTimeout <positive_integer>]] [-cipherRedirect ( ENABLED | DISABLED ) [-cipherURL <URL>]] [-clientAuth ( ENABLED | DISABLED )[-clientCert ( Mandatory | Optional )]] [-sslRedirect ( ENABLED |
DISABLED )] [-redirectPortRewrite ( ENABLED | DISABLED )] [-ssl3 ( ENABLED | DISABLED )] [-tls1 ( ENABLED | DISABLED )] [-tls11 ( ENABLED| DISABLED )] [-tls12 ( ENABLED | DISABLED )] [-tls13 ( ENABLED |DISABLED )] [-SNIEnable ( ENABLED | DISABLED )] [-ocspStapling (ENABLED | DISABLED )] [-serverAuth ( ENABLED | DISABLED )] [-commonName <string>] [-pushEncTrigger <pushEncTrigger>] [-sendCloseNotify ( YES |
NO )] [-clearTextPort <port|*>] [-insertionEncoding ( Unicode | UTF-8)]
 [-denySSLReneg <denySSLReneg>] [-quantumSize <quantumSize>]
[-strictCAChecks ( YES | NO )] [-encryptTriggerPktCount <positive_integer>] [-pushFlag <positive_integer>][-dropReqWithNoHostHeader ( YES | NO )] [-SNIHTTPHostMatch <SNIHTTPHostMatch>] [-pushEncTriggerTimeout <positive_integer>]
[-sslTriggerTimeout <positive_integer>] [-clientAuthUseBoundCAChain ( ENABLED | DISABLED )] [-sslInterception ( ENABLED | DISABLED )][-ssliReneg ( ENABLED | DISABLED )] [-ssliOCSPCheck ( ENABLED | DISABLED )] [-ssliMaxSessPerServer <positive_integer>] [-HSTS ( ENABLED| DISABLED )] [-maxage <positive_integer>] [-IncludeSubdomains ( YES | NO )] [-preload ( YES | NO )] [-sessionTicket ( ENABLED | DISABLED )][-sessionTicketLifeTime <positive_integer>] [-sessionTicketKeyRefresh (ENABLED | DISABLED )] {
 -sessionTicketKeyData  }
  [-sessionKeyLifeTime <positive_integer>] [-prevSessionKeyLifeTime <positive_integer>]
[-cipherName <string> -cipherPriority <positive_integer>][-strictSigDigestCheck ( ENABLED | DISABLED )]
[-skipClientCertPolicyCheck ( ENABLED | DISABLED )] [-zeroRttEarlyData ( ENABLED | DISABLED )] [-tls13SessionTicketsPerAuthContext
<positive_integer>] [-dheKeyExchangeWithPsk ( YES | NO )]
<!--NeedCopy-->
```

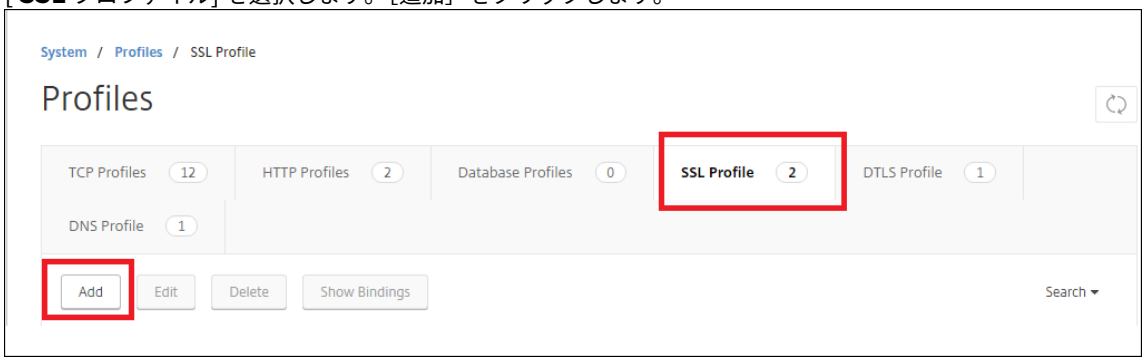

GUI を使用して SSL プロファイルパラメータを設定する

プロファイルを追加するには:

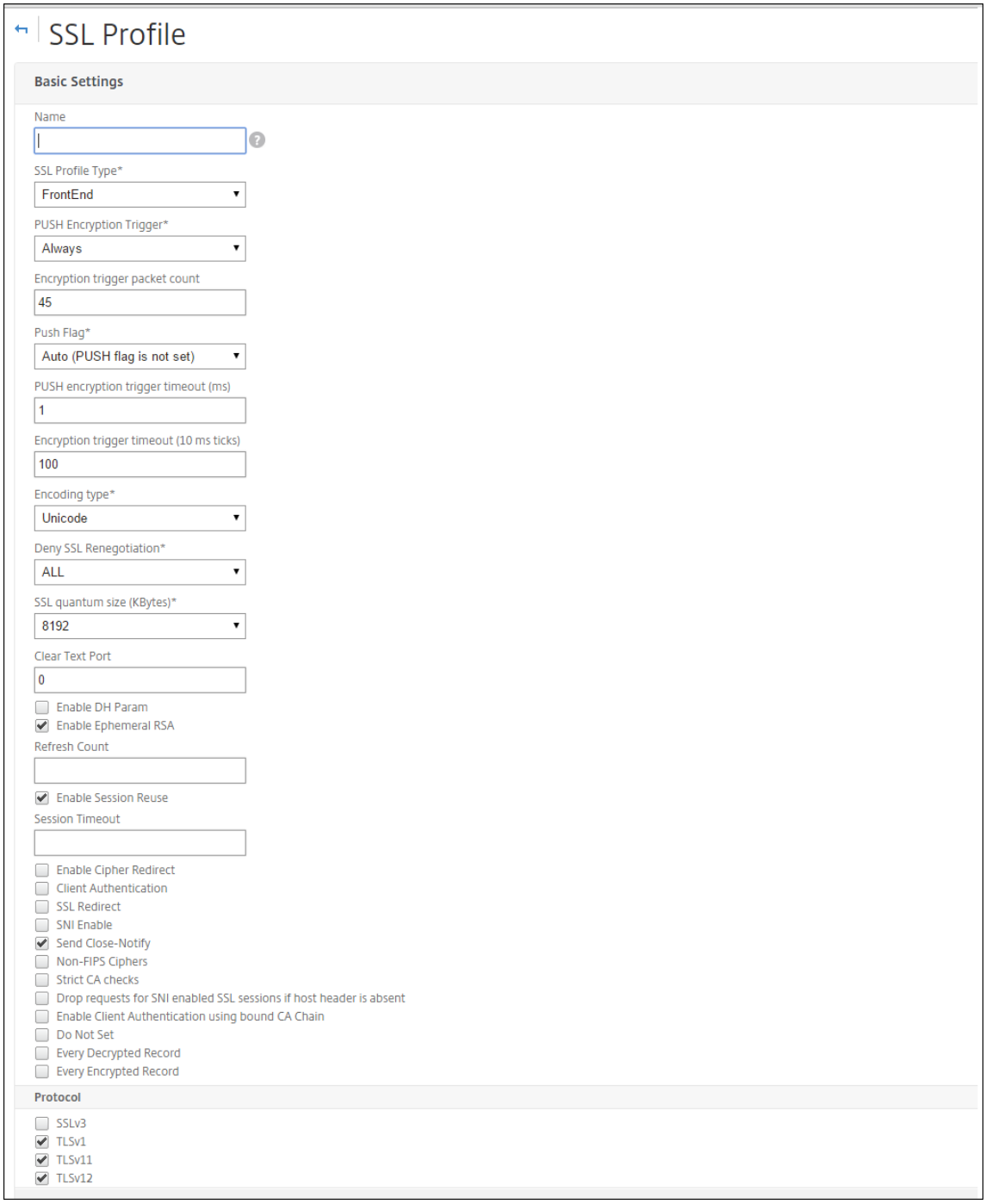
1. [システム] > [プロファイル] に移動します。



2. [SSL プロファイル] を選択します。[追加] をクリックします。



3. さまざまなパラメーターの値を指定します。



4. [**OK**] をクリックします。
5. [完了] をクリックします。

既存の SSL プロファイルを再利用するには、次の手順を実行します。

1. [システム**] > [**プロファイル] に移動します。
2. 既存のプロファイルを選択し、[**Add**] をクリックします。
3. 別の名前を指定し、パラメータを変更して「**OK**」をクリックします。

4. ［完了］をクリックします。

TLS セッションチケット拡張

SSL ハンドシェイクは CPU に負荷がかかる操作です。セッション再利用が有効な場合、既存のクライアントに対するサーバー/クライアントキー交換操作はスキップされます。ユーザーはセッションを再開できます。このアクションにより、応答時間が短縮され、サーバーがサポートできる 1 秒あたりの SSL トランザクション数が増加します。ただし、サーバーは各セッション状態の詳細を格納する必要があるため、メモリを消費し、要求がサーバー間で負荷分散されている場合は複数のサーバー間で共有することが困難になります。

NetScaler アプライアンスはセッションチケット TLS 拡張をサポートしています。この拡張を使用すると、セッションの詳細がサーバーではなくクライアントに保存されます。クライアントは、クライアントの Hello メッセージにセッションチケット TLS 拡張を含めることによって、このメカニズムをサポートしていることを示す必要があります。新規クライアントの場合、この拡張機能は空です。サーバーは NewsessionTicket ハンドシェイクメッセージで新しいセッションチケットを送信します。セッションチケットは、サーバーだけが認識しているキーペアを使用して暗号化されます。サーバが今新しいチケットを発行できない場合は、通常のハンドシェイクが完了します。

この機能は、フロントエンド SSL プロファイルでのみ使用でき、アプライアンスがサーバとして機能してセッションチケットを生成する通信のフロントエンドでのみ使用できます。

制限事項

- この機能は FIPS プラットフォームではサポートされていません。
- この機能は TLS バージョン 1.1 および 1.2 でのみサポートされます。
- SSL セッション ID の永続性は、セッションチケットではサポートされていません。

CLI を使用して TLS セッションチケット拡張を有効にする

コマンドプロンプトで入力します。

```
set ssl profile <name> -sessionTicket (ENABLED | DISABLED ) [-
    sessionTicketLifeTime <positive_integer>
<!--NeedCopy-->
```

引数**:**

SessionTicket: TLS セッションチケット拡張の状態。この拡張を使用すると、RFC 5077 で定義されているように、セッションの詳細がサーバではなくクライアントに保存されます。

設定可能な値：ENABLED, DISABLED

デフォルト値: 無効

SessionTicketLifetime: セッションチケットが期限切れになり、新しい SSL ハンドシェイクが開始されるまでの時間を秒単位で指定します。

デフォルト値:300

最小値:0

最大値:172800

例**:**

```
1 add ssl profile profile1 -sessionTicket ENABLED -sessionTicketlifeTime
     300
2 Done
3 <!--NeedCopy-->
```

GUI を使用して **TLS** セッションチケット拡張を有効にする

1. **System** > **Profiles** に移動します。[**SSL** プロファイル] を選択します。
2. [追加] をクリックし、プロファイルの名前を指定します。
3. [セッションチケット] を選択します。
4. 必要に応じて、[セッションチケットの有効期間 **(**秒**)**] を指定します。

セッションチケットの安全な実装

TLS セッションチケットを使用することで、クライアントは短縮されたハンドシェイクを使用してサーバーへの再接続を高速化できます。ただし、セッションチケットが長期間暗号化または変更されないと、セキュリティリスクが生じる可能性があります。セッションチケットを対称鍵で暗号化することで、セッションチケットを保護できます。前方秘匿性を実現するために、セッションチケットキーが更新される時間間隔を指定できます。

アプライアンスはデフォルトでセッションチケットキーを生成します。ただし、展開内の複数のアプライアンスが互いのセッションチケットを復号化する必要がある場合は、すべて同じセッションチケットキーを使用する必要があります。したがって、すべてのアプライアンスで、同じセッションチケットキーデータを手動で設定（追加またはロード）する必要があります。セッションチケットキーデータには次の情報が含まれます。

- セッションチケット名。
- チケットの暗号化または復号化に使用されるセッション AES キー。
- チケットのダイジェストを計算するために使用されるセッション HMAC キー。

RFC 5077 で推奨されているように、長さ 64 バイトのセッションチケットキーデータを 256 ビット HMAC キーをサポートするように構成できるようになりました。下位互換性を保つため、キーの長さ 48 バイトもサポートされています。

> 注:
>
> セッションチケットキーデータを手動で入力するときは、HA セットアップまたはクラスターセットアップのすべての NetScaler アプライアンスの構成が同じであることを確認してください。

sessionTicketKeyLifeTime このパラメータは、セッションチケットキーが更新される頻度を指定します。prevSessionTicketKeyLifeTime パラメータを設定して、新しいキーが生成された後、そのキーを使用してチケットを復号化するために、前のセッションチケットキーを保持する期間を指定できます。prevSessionTicketKeyLifeTime設定により、クライアントが短縮ハンドシェイクを使用して再接続できる時間が長くなります。たとえば、sessionTicketKeyLifeTimeが 10 分、prevSessionTicketKeyLifeTimeが 5 分に設定されている場合、10 分後に新しいキーが生成され、すべての新しいセッションに使用されます。ただし、以前に接続したクライアントにはさらに 5 分かかり、その間は以前に発行されたチケットが短縮されたハンドシェイクで優先されます。

CLI を使用して **SSL** セッションチケットデータを構成する

コマンドプロンプトで入力します。

```
set ssl profile <name> -sessionTicket ENABLED -sessionTicketLifeTime <
    positive_integer> -sessionTicketKeyRefresh ( ENABLED | DISABLED )] -
    sessionTicketKeyLifeTime <positive_integer> [-
    prevSessionTicketKeyLifeTime <positive_integer>]
<!--NeedCopy-->
```

引数**:**

SessionTicket: RFC 5077 で説明されているようにセッションチケットを使用します。初期ハンドシェイクを確立するには、CPU を集中的に使用する公開キー暗号化操作が必要です。**ENABLED** を設定すると、サーバーはクライアントにセッションチケットを発行し、クライアントはこれを使用して短縮ハンドシェイクを実行できます。

指定可能な値: 有効、無効。既定: 無効

SessionTicketLifetime: セッションチケットの有効期間 (秒単位)。この時間が経過すると、クライアントはこのチケットを使用してセッションを再開できなくなります。

最大値は 172800 です。最小値:0。デフォルト:300。

SessionTicketKeyRefresh: セッションチケットキーの有効期間パラメータで指定された時間が経過したら、セッションチケットの暗号化または復号化に使用されるセッションチケットキーを再生成します。SessionTicket が有効な場合、自動的に有効になります。管理者がセッションチケットデータを入力すると無効になります。

指定可能な値: 有効、無効。既定: 有効

SessionKeyLifeTime: NetScaler アプライアンスによって発行されたセッションチケットの暗号化に使用される対称キーの有効期間（秒単位)。

最大値は 86400 です。最小値:600。デフォルト:3000

prevSessionKeyLifetime: セッションチケットキーの有効期間が終了した後も、セッションチケットの暗号化に使用された以前の対称キーが既存のクライアントに対して有効である時間（秒）。この時間内に、既存のクライアントは前のセッションチケットキーを使用してセッションを再開できます。新しいクライアントのセッションチケットは、新しい鍵を使用して暗号化されます。

最大値は 172800 です。最小値:0。デフォルト:0

例**:**

```
set ssl profile ns_default_ssl_profile_frontend -sessionTicket ENABLED
    -sessionTicketlifeTime 120 -sessionTicketKeyRefresh ENABLED -
    sessionTicketKeyLifeTime 100 -prevSessionTicketKeyLifeTime 60

Done

show ssl profile ns_default_ssl_profile_frontend

     Session Ticket: ENABLED
     Session Ticket Lifetime: 120 (secs)
     Session Key Auto Refresh: ENABLED
     Session Key Lifetime: 100 (secs)
     Previous Session Key Lifetime: 60 (secs)
<!--NeedCopy-->
```

GUI を使用して **SSL** セッションチケットデータを構成する

1. [システム] > [プロファイル] に移動し、[**SSL** プロファイル] を選択します。
2. **ns_default_ssl_profile_frontend** を選択し、[編集] をクリックします。
3. 「基本設定 **(Basic Settings**)」セクションで、鉛筆アイコンをクリックし、次のパラメータを設定します。
   - セッションチケット
   - セッションチケットの有効期間 (秒)
   - セッションチケットキーの自動更新
   - セッションチケットキーの有効期間 (秒)
   - 前のセッションチケットキーの有効期間 (秒)
4. [**OK**] をクリックします。

CLI を使用して **SSL** セッションチケットデータを手動で入力する

コマンドプロンプトで入力します。

```
set ssl profile <name> -sessionTicket ENABLED

set ssl profile <name> -sessionTicketKeyData

```



```
show ssl profile ns_default_ssl_profile_frontend
<!--NeedCopy-->
```

引数**:**

SessionTicket: RFC 5077 で説明されているセッションチケットの使用。初期ハンドシェイクを確立するには、CPU を集中的に使用する公開キー暗号化操作が必要です。**ENABLED** を設定すると、サーバーはクライアントにセッションチケットを発行し、クライアントはこれを使用して短縮ハンドシェイクを実行できます。

指定可能な値: 有効、無効。既定: 無効

セッションチケットキーデータ: Contains the session ticket name (0-15 bytes), the session AES key used to encrypt or decrypt the session ticket (16-31 bytes), and the session HMAC key used to compute the digest of the ticket (32-63 bytes). Externally generated by an administrator and added to a NetScaler appliance.

最大長:64 バイト

例**:**

```
set ssl profile ns_default_ssl_profile_frontend -sessionTicket ENABLED

Done

set ssl profile ns_default_ssl_profile_frontend -sessionTicketKeyData
    1111111111111111111111111111111111111111111111111111

Done

show ssl profile ns_default_ssl_profile_frontend

    1) Name: ns_default_ssl_profile_frontend (Front-End)
    SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: ENABLED TLSv1.2: ENABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks: NO
    Session Reuse: ENABLED Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED Ephemeral RSA: ENABLED
        Refresh Count: 0
    Deny SSL Renegotiation ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout: 1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions: NO
    Push flag: 0x0 (Auto)
```



```
30      SSL quantum size: 8 kB
31      Encryption trigger timeout 100 mS
32      Encryption trigger packet count: 45
33      Subject/Issuer Name Insertion Format: Unicode
34      Session Ticket: ENABLED
35      Session Ticket Lifetime: 300 (secs)
36      Session Key Auto Refresh: DISABLED
37      Session Key Lifetime: 3000 (secs)
38      Previous Session Key Lifetime: 0 (secs)
39      Session Key Data: 84
           dad1afc6d56b0deeb0a7fd7f299a207e8d8c15cdd087a5684a11a329fd732e87a0535d9088

40      47
           e8c181ba266f5c8838ae472cb3ab9255b683bf922fad32cee816c329989ef7cdeb278e93ac

41
42      ECC Curve: P_256, P_384, P_224, P_521
43
44      1) Cipher Name: DEFAULT Priority :4
45      Description: Predefined Cipher Alias
46
47      1) Internal Service Name (Front-End): nsrnatsip-127.0.0.1-5061
48      2) Internal Service Name (Front-End): nskrpcs-127.0.0.1-3009
49      3) Internal Service Name (Front-End): nshttps-::1l-443
50      4) Internal Service Name (Front-End): nsrpcs-::1l-3008
51      5) Internal Service Name (Front-End): nshttps-127.0.0.1-443
52      6) Internal Service Name (Front-End): nsrpcs-127.0.0.1-3008
53      7) Vserver Name: v1
54
55  Done
56  <!--NeedCopy-->
```

GUI を使用して **SSL** セッションチケットデータを手動で入力する

1. [システム**] > [**プロファイル] に移動し、[**SSL** プロファイル] を選択します。
2. **ns_default_ssl_profile_frontend** を選択し、[編集] をクリックします。
3. 「基本設定 **(Basic Settings**)」セクションで、鉛筆アイコンをクリックし、次のパラメータを設定します。
   - セッションチケット
   - セッションチケットキーデータ
   - セッションチケットキーデータの確認
4. [**OK**] をクリックします。

NetScaler 非 **FIPS** プラットフォームでの **SSL** ハンドシェイクの拡張マスターシークレットのサポート

注**:** このパラメーターは、リリース 13.0 ビルド 61.x で導入されました。

拡張マスターシークレット (EMS) は、トランスポート層セキュリティ (TLS) プロトコルのオプションの拡張です。NetScaler アプライアンス上の EMS をサポートするために、フロントエンドとバックエンドの両方の SSL プロファイルに適用される新しいパラメーターが追加されました。パラメータが有効で、ピアが EMS をサポートしている場合、ADC アプライアンスは EMS 計算を使用します。ピアが EMS をサポートしていない場合、アプライアンスでパラメータが有効になっていても、EMS 計算は接続に使用されません。EMS の詳細については、RFC 7627 を参照してください。

注**:** EMS は、TLS プロトコルバージョン 1.0、1.1、または 1.2 を使用するハンドシェイクにのみ適用されます。

EMS のプラットフォームサポート

- Cavium N3 チップまたは Intel Coleto Creek 暗号カードのいずれかを搭載した MPX および SDX プラットフォーム Intel Coleto チップには、次のプラットフォームが同梱されています。
- MPX 5900
- MPX/SDX 8900
- MPX/SDX 26000
- MPX/SDX 26000-50S
- MPS/SDX 26000-100G
- MPX/SDX 15000-50G

また、`show hardware`コマンドを使用して、アプライアンスに Coleto (COL) チップと N3 チップのどちらが搭載されているかを識別することもできます。

- 暗号カードを使用しない MPX および SDX プラットフォーム (ソフトウェアのみ)
- ソフトウェアのみのプラットフォーム:VPX、CPX、BLX

EMS は次のプラットフォームでは有効にできません。

- MPX 9700 FIPS および MPX 14000 FIPS プラットフォーム。
- Cavium N2 暗号チップを含む MPX および SDX プラットフォーム

パラメータが有効な場合、ADC アプライアンスは TLS 1.2、TLS 1.1、および TLS 1.0 接続で EMS を使用しようとします。この設定は TLS 1.3 または SSLv3 接続には影響しません。

EMS がピアとネゴシエートされるようにするには、仮想サーバー (フロントエンド) またはサービス (バックエンド) にバインドされた SSL プロファイルで設定を有効にします。

CLI を使用して **EMS** を有効にする

コマンドプロンプトで入力します。

```
set ssl profile <profile name> [-allowExtendedMasterSecret (YES | NO)
]
```


例

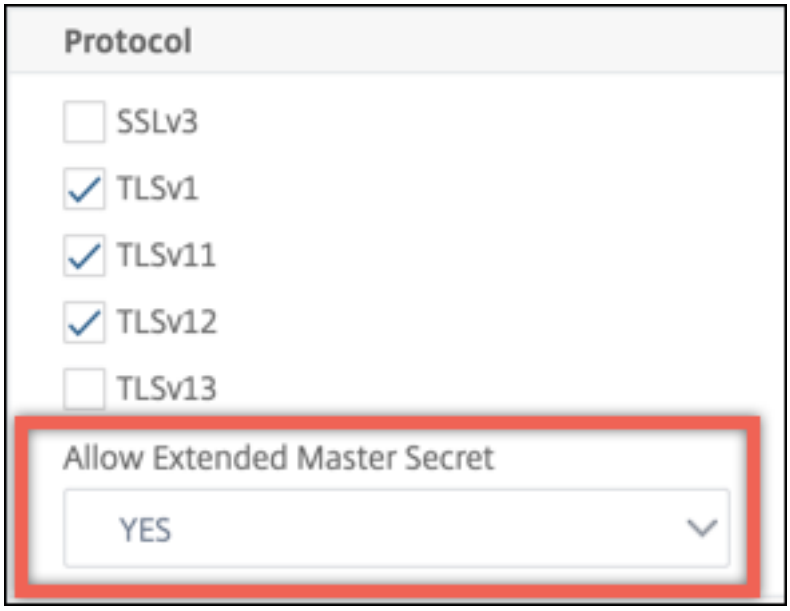
```
set ssl profile ns_default_ssl_profile_frontend  -
    allowExtendedMasterSecret YES

set ssl profile ns_default_ssl_profile_backend  -
    allowExtendedMasterSecret YES
<!--NeedCopy-->
```

次の表は、`allowExtendedMasterSecret` さまざまなデフォルトプロファイルとユーザー定義プロファイルのパラメーターのデフォルト値を示しています。

Profile	デフォルト設定
既定のフロントエンドプロファイル	いいえ
デフォルトのフロントエンドセキュアプロファイル	はい
デフォルトのバックエンドプロファイル	いいえ
ユーザー定義プロファイル	いいえ

GUI を使用して **EMS** を有効にする

1. **[**システム**] > [**プロファイル**] > [SSL** プロファイル**]** に移動します。
2. プロファイルを追加するか、プロファイルを編集します。
3. [拡張マスターシークレットを許可] を [はい]



クライアントの **hello** メッセージでの **ALPN** 拡張機能の処理のサポート

注: この機能は、リリース 11.0 ビルド 64.x 以降でサポートされています。

SSL_TCP 仮想サーバーによって処理される接続の ALPN 拡張でアプリケーションプロトコルをネゴシエートするために、フロントエンド SSL プロファイルにパラメーター`alpnProtocol`が追加されます。クライアント hello メッセージの ALPN 拡張で同じプロトコルが受信された場合、SSL プロファイルで指定されたプロトコルだけがネゴシエートされます。

注:`alpnProtocol`パラメーターは、フロントエンド SSL プロファイルでのみサポートされ、SSL_TCP タイプの仮想サーバーによって処理される SSL 接続に適用されます。

CLI を使用してフロントエンド **SSL** プロファイルにプロトコルを設定する

コマンドプロンプトで入力します。

```
set ssl profile ns_default_ssl_profile_frontend -alpnProtocol <
protocol_name>
```

`alpnProtocol`パラメータには 3 つの値を指定できます。最大長:4096 バイト。

- **NONE:** アプリケーションプロトコルのネゴシエーションは行われません。この設定がデフォルトです。
- **HTTP1:** HTTP1 はアプリケーションプロトコルとしてネゴシエートできます。
- **HTTP2:** HTTP2 はアプリケーションプロトコルとしてネゴシエートできます。

例:

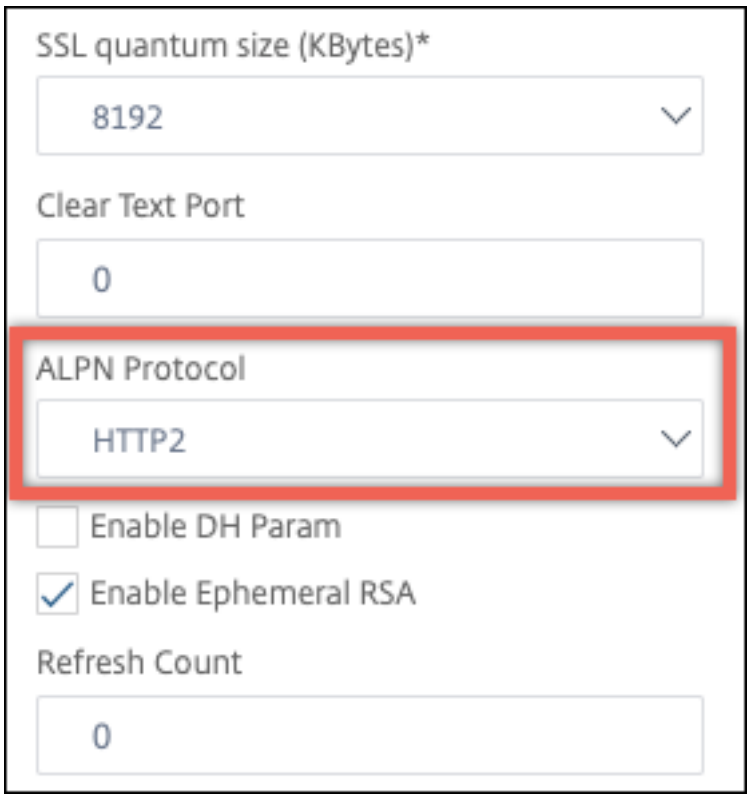
```
1  set ssl profile ns_default_ssl_profile_frontend -ALPNProtocol HTTP2
2  > sh ssl profile ns_default_ssl_profile_frontend
3  1)  Name: ns_default_ssl_profile_frontend   (Front-End)
4      SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
           ENABLED  TLSv1.3: DISABLED
5      Client Auth: DISABLED
6      Use only bound CA certificates: DISABLED
7      Strict CA checks:   NO
8      Session Reuse: ENABLED  Timeout: 120 seconds
9      DH: DISABLED
10     DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
           ENABLED  Refresh Count: 0
11     Deny SSL Renegotiation  ALL
12     Non FIPS Ciphers: DISABLED
13     Cipher Redirect: DISABLED
14     SSL Redirect: DISABLED
15     Send Close-Notify: YES
16     Strict Sig-Digest Check: DISABLED
17     Zero RTT Early Data: DISABLED
18     DHE Key Exchange With PSK: NO
19     Tickets Per Authentication Context: 1
20     Push Encryption Trigger: Always
```



```
21      PUSH encryption trigger timeout:    1 ms
22      SNI: DISABLED
23      OCSP Stapling: DISABLED
24      Strict Host Header check for SNI enabled SSL sessions:  NO
25      Match HTTP Host header with SNI:    CERT
26      Push flag:  0x0 (Auto)
27      SSL quantum size:   8 kB
28      Encryption trigger timeout  100 mS
29      Encryption trigger packet count:    45
30      Subject/Issuer Name Insertion Format:   Unicode
31
32      SSL Interception: DISABLED
33      SSL Interception OCSP Check: ENABLED
34      SSL Interception End to End Renegotiation: ENABLED
35      SSL Interception Maximum Reuse Sessions per Server: 10
36      Session Ticket: DISABLED
37      HSTS: DISABLED
38      HSTS IncludeSubDomains: NO
39      HSTS Max-Age: 0
40      HSTS Preload: NO
41      Allow Extended Master Secret: NO
42      Send ALPN Protocol: HTTP2
43
44   Done
45   <!--NeedCopy-->
```

GUI を使用してフロントエンド **SSL** プロファイルにプロトコルを設定する

1. [システム**] > [**プロファイル] に移動し、[**SSL** プロファイル] を選択します。
2. **ns_default_ssl_profile_frontend** を選択し、[編集] をクリックします。
3. [**ALPN** プロトコル] リストで [**HTTP2**] を選択します。



古い設定をロードする

デフォルトプロファイルの有効化は元に戻せません。ただし、展開にデフォルトプロファイルが必要ないと判断した場合は、デフォルトプロファイルを有効にする前に保存した古い設定を読み込むことができます。この変更は、アプライアンスの再起動後に有効になります。

CLI を使用して古い設定をロードする

コマンドプロンプトで入力します。

```
shell

root@ns# clear config

root@ns# cd /nsconfig

root@ns# cp ns.conf.NS.11.0.jun.16 ns.conf

root@ns# reboot
<!--NeedCopy-->
```


安全なフロントエンドプロファイル

August 15, 2023

デフォルトのフロントエンドとデフォルトのバックエンド・プロファイルに加えて、新しいデフォルトのセキュア・フロントエンド・プロファイルがリリース 12.1 から利用できるようになりました。Qualys SSL Labs の A+ 評価 (2018 年 5 月現在) に必要な設定は、このプロファイルにプリロードされています。以前は、SSL フロントエンドプロファイルまたは SSL 仮想サーバーで A+ 評価に必要な各パラメーターを明示的に設定する必要がありました。これで ns_default_ssl_profile_secure_frontend プロファイルを SSL 仮想サーバーにバインドでき、必要なパラメーターが SSL 仮想サーバーに自動的に設定されます。

> 注:
>
> セキュアフロントエンドプロファイルは編集できません。

デフォルトプロファイルを有効にすると、デフォルトのフロントエンドプロファイルは自動的にすべての SSL 仮想サーバーにバインドされます。A+ 評価を取得するには、ns_default_ssl_profile_secure_frontend プロファイルを明示的にバインドする必要があります。また、SHA2/SHA256 サーバー証明書を SSL 仮想サーバーにバインドする必要があります。

安全なフロントエンドプロファイルパラメータ

パラメータとそのデフォルト設定は次のとおりです。

```
SSLv3: DISABLED  TLSv1.0: DISABLED  TLSv1.1: DISABLED  TLSv1.2: ENABLED
      TLSv1.3: DISABLED

Deny SSL Renegotiation: NONSECURE

HSTS: ENABLED

HSTS IncludeSubDomains: YES

HSTS Max-Age: 15552000

Cipher Name: SECURE    Priority :1
<!--NeedCopy-->
```

安全な暗号エイリアス

新しいセキュア暗号エイリアスが追加され、セキュアフロントエンドプロファイルにバインドされます。このエイリアスに含まれる暗号を一覧表示するには、コマンドプロンプトで show cipher SECURE と入力します。

```
show cipher SECURE
```



```
2
3       1) Cipher Name: TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 Priority : 1
4               Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(256)
                   Mac=AEAD   HexCode=0xc030
5       2) Cipher Name: TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 Priority : 2
6               Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(128)
                   Mac=AEAD   HexCode=0xc02f
7       3) Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
           Priority : 3
8               Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(256)
                   Mac=AEAD   HexCode=0xc02c
9       4) Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
           Priority : 4
10              Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(128)
                   Mac=AEAD   HexCode=0xc02b
11  Done
12  <!--NeedCopy-->
```

構成

次の手順を実行します:

1. SSL タイプの負荷分散仮想サーバーを追加します。
2. SHA2/SHA256 証明書をバインドします。
3. デフォルトプロファイルを有効にします。
4. セキュアフロントエンドプロファイルを SSL 仮想サーバーにバインドします。

CLI を使用して **SSL** 仮想サーバーの **A+** 評価を取得する

コマンドプロンプトで入力します。

```
1  add lb vserver <name> <serviceType> <IPAddress> <port>
2  bind ssl vserver <vServerName> -certkeyName <string>
3  set ssl parameter -defaultProfile ENABLED
4  set ssl vserver <vServerName> -sslProfile
       ns_default_ssl_profile_secure_frontend
5  show ssl vserver [<vServerName>]
6  <!--NeedCopy-->
```

例**:**

```
1  add lb vserver ssl-vsvr SSL 192.0.2.240 443
2
3  bind ssl vserver ssl-vsvr -certkeyName letrsa
4
5  set ssl parameter -defaultProfile ENABLED
6
```



```
Save your configuration before enabling the Default profile. You cannot undo the changes. Are you sure you want to enable the Default profile? [Y/N]y

set ssl vserver ssl-vsvr -sslProfile ns_default_ssl_profile_secure_frontend
<!--NeedCopy-->
```

```
sh ssl vserver ssl-vsvr

    Advanced SSL configuration for VServer ssl-vsvr:
    Profile Name :ns_default_ssl_profile_secure_frontend
    1) CertKey Name: letrsa     Server Certificate
Done
<!--NeedCopy-->
```

```
sh ssl profile ns_default_ssl_profile_secure_frontend

    1) Name: ns_default_ssl_profile_secure_frontend (Front-End)
    SSLv3: DISABLED  TLSv1.0: DISABLED  TLSv1.1: DISABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:                   NO
    Session Reuse: ENABLED   Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA: ENABLED     Refresh Count: 0
    Deny SSL Renegotiation                NONSECURE
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:  1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:
                           NO
    Push flag:             0x0 (Auto)
    SSL quantum size:                8 kB
    Encryption trigger timeout       100 mS
    Encryption trigger packet count: 45
    Subject/Issuer Name Insertion Format:  Unicode
    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server:   10
    Session Ticket: DISABLED
```



```
35      HSTS: ENABLED
36      HSTS IncludeSubDomains: YES
37      HSTS Max-Age: 15552000
38      ECC Curve: P_256, P_384, P_224, P_521
39      1) Cipher Name: SECURE    Priority :1
40      Description: Predefined Cipher Alias
41      1) Vserver Name: v2
42  Done
43  <!--NeedCopy-->
```

GUI を使用して **SSL** 仮想サーバーの **A+** 評価を取得

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、SSL 仮想サーバーを選択します。
2. [詳細設定] で [SSL プロファイル] をクリックします。
3. ns_default_ssl_profile_secure_frontend を選択します。
4. 「OK」をクリックします。
5. [完了] をクリックします。

付録 **A:** アップグレード後の **SSL** 設定の移行例

August 15, 2023

注**:** 新しいデフォルトプロファイルの SSL 移行スクリプトがサポートされなくなったため、このコンテンツは削除されました。

付録 **B**: デフォルトのフロントエンドおよびバックエンド **SSL** プロファイル設定

August 15, 2023

デフォルトのフロントエンドプロファイルには以下の設定があります。

```
1  sh ssl profile ns_default_ssl_profile_frontend
2
3  1)Name: ns_default_ssl_profile_frontend
4
5       Configuration for Front-End SSL profile
6       DH: DISABLED
7       Ephemeral RSA: ENABLED          Refresh Count: 0
8       Session Reuse: ENABLED          Timeout: 120 seconds
9       Non FIPS Ciphers: DISABLED
10      Cipher Redirect: ENABLED   Redirect URL: http://10.102.28.212/
           redirect.html
```



```
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      SNI: DISABLED
      SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
         ENABLED
      Push Encryption Trigger: Always
      PUSH encryption trigger timeout:     1 ms
      Send Close-Notify: YES
      Push flag: 0x0 (Auto)
      Deny SSL Renegotiation          NO
      SSL quantum size:          8 kB
      Strict CA checks:          NO
      Encryption trigger timeout 100 mS
      Encryption trigger packet count:     45
      Use only bound CA certificates: DISABLED
      Subject/Issuer Name Insertion Format: Unicode
      Strict Host Header check for SNI enabled SSL sessions:          NO

      ECC Curve: P_256, P_384, P_521

1)    Cipher Name: AES     Priority :2
      Description: Predefined Cipher Alias

1)    Vserver Name: v1
2)    Vserver Name: nshttps-::1l-443
3)    Vserver Name: nsrpcs-::1l-3008
4)    Vserver Name: nskrpcs-127.0.0.1-3009
5)    Vserver Name: nshttps-127.0.0.1-443
6)    Vserver Name: nsrpcs-127.0.0.1-3008
Done
<!--NeedCopy-->
```

デフォルトのバックエンドプロファイルには、次の設定があります。

```
sh ssl profile ns_default_ssl_profile_backend

1)Name: ns_default_ssl_profile_backend

      Configuration for Back-End SSL profile
      Session Reuse: ENABLED          Timeout: 300 seconds
      Non FIPS Ciphers: DISABLED
      Server Auth: DISABLED
      SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2:
         DISABLED
      Push Encryption Trigger: Always
      PUSH encryption trigger timeout:     1 ms
      Send Close-Notify: YES
      Push flag: 0x0 (Auto)
      Deny SSL Renegotiation          ALL
      SSL quantum size:          8 kB
      Strict CA checks:          NO
      Encryption trigger timeout 100 mS
      Encryption trigger packet count:     45
```



```
19       Use only bound CA certificates: DISABLED
20
21       ECC Curve: P_256, P_224, P_521
22
23  1)   Cipher Name: AES      Priority :1
24       Description: Predefined Cipher Alias
25
26  2)   Cipher Name: RC4      Priority :2
27       Description: Predefined Cipher Alias
28
29  1)   Service Name: s2
30  2)   Service Name: s1
31  Done
32  <!--NeedCopy-->
```

レガシー **SSL** プロファイル

August 15, 2023

> 注:
>
> 従来のプロファイルではなく、拡張プロファイルの使用をお勧めします。拡張プロファイルインフラストラクチャの詳細については、「SSL プロファイルインフラストラクチャ」を参照してください。

重要:

> SSL プロファイルを SSL 仮想サーバーにバインドします。DTLS プロファイルを SSL 仮想サーバーにバインドしないでください。DTLS プロファイルの詳細については、DTLS プロファイルを参照してください。

SSL プロファイルを使用して、NetScaler ADC が SSL トラフィックを処理する方法を指定できます。プロファイルは、仮想サーバー、サービス、サービスグループなどの SSL エンティティの SSL パラメータ設定の集合であり、構成が容易で柔軟性があります。設定できるグローバルパラメータは 1 セットだけに限定されません。グローバルパラメータの複数のセット (プロファイル) を作成し、異なる SSL エンティティに異なるセットを割り当てることができます。SSL プロファイルは次の 2 つのカテゴリに分類されます。

- フロントエンドエンティティに適用されるパラメータを含むフロントエンドプロファイル。つまり、クライアントからリクエストを受け取るエンティティに適用されます。
- バックエンド・エンティティに適用可能なパラメータを含むバックエンド・プロファイル。つまり、クライアント要求をサーバーに送信するエンティティに適用されます。

TCP や HTTP プロファイルとは異なり、SSL プロファイルはオプションです。したがって、デフォルトの SSL プロファイルはありません。同じプロファイルを複数のエンティティで再利用できます。エンティティにプロファイルが添付されていない場合は、グローバルレベルで設定された値が適用されます。動的に学習されるサービスには、現在のグローバル値が適用されます。

次の表は、各プロファイルに含まれるパラメータを示しています。

フロントエンドプロファイル	バックエンドプロファイル
cipherRedirect, cipherURL	denySSLReneg
clearTextPort*	encryptTriggerPktCount
clientAuth, clientCert	nonFipsCiphers
denySSLReneg	pushEncTrigger
dh, dhFile, dhCount	pushEncTriggerTimeout
dropReqWithNoHostHeader	pushFlag
encryptTriggerPktCount	quantumSize
eRSA, eRSACount	serverAuth
insertionEncoding	commonName
nonFipsCiphers	sessReuse, sessTimeout
pushEncTrigger	SNIEnable
pushEncTriggerTimeout	ssl3
pushFlag	sslTriggerTimeout
quantumSize	strictCAChecks
redirectPortRewrite	tls1
sendCloseNotify	-
sessReuse, sessTimeout	-
SNIEnable	-
ssl3	-
sslRedirect	-
sslTriggerTimeout	-
strictCAChecks	-
tls1, tls11, tls12	-

* ClearTextPort パラメータは SSL 仮想サーバーにのみ適用されます。

プロファイルの一部ではないパラメータを設定しようとすると、エラーメッセージが表示されます。たとえば、バックエンドプロファイルで ClientAuth パラメーターを設定しようとしたとします。

CRL メモリサイズ、OCSP キャッシュサイズ、UnDefaction コントロール、UnDefaction Data などの一部の SSL

パラメータは、エンティティから独立しているため、前述のどのプロファイルにも含まれていません。

SSL プロファイルは次の操作をサポートします。

- 追加—NetScaler に SSL プロファイルを作成します。プロファイルがフロントエンドかバックエンドかを指定します。フロントエンドがデフォルトです。
- 設定-既存のプロファイルの設定を変更します。
- 「設定解除」(Unset)-指定されたパラメータをデフォルト値に設定します。パラメータを何も指定しない場合、エラーメッセージが表示されます。エンティティのプロファイルを設定解除すると、プロファイルはエンティティからバインド解除されます。
- 削除-プロファイルを削除します。どのエンティティでも使用されているプロファイルは削除できません。設定をクリアすると、すべてのエンティティが削除されます。その結果、プロファイルも削除されます。
- 表示-NetScaler で使用可能なすべてのプロファイルを表示します。プロファイル名を指定すると、そのプロファイルの詳細が表示されます。エンティティを指定すると、そのエンティティに関連付けられているプロファイルが表示されます。

CLI を使用して SSL プロファイルを作成する

- SSL プロファイルを追加するには、次のように入力します。

```
add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )]
<!--NeedCopy-->
```

- 既存のプロファイルを変更するには、次のように入力します。

```
set ssl profile <name>
<!--NeedCopy-->
```

- 既存のプロファイルを設定解除するには、次のように入力します。

```
unset ssl profile <name> [-dh] [-dhFile] [-dhCount] [-eRSA]…
<!--NeedCopy-->
```

- エンティティから既存のプロファイルを設定解除するには、次のように入力します。

```
unset ssl vserver <vServerName>  -sslProfile
<!--NeedCopy-->
```

- 既存のプロファイルを削除するには、次のように入力します。

```
rm ssl profile <name>
<!--NeedCopy-->
```

- 既存のプロファイルを表示するには、次のように入力します。


```
sh ssl profile <name>
<!--NeedCopy-->
```

GUI を使用して **SSL** プロファイルを作成する

[システム] > [プロファイル] に移動し、[SSL プロファイル] タブを選択し、SSL プロファイルを作成します。

クライアント証明書の検証をより厳密に制御できるようにする

NetScaler アプライアンスは、1 つのルート CA が発行した有効な中間 CA 証明書を受け入れます。つまり、ルート CA 証明書のみが仮想サーバーにバインドされ、そのルート CA がクライアント証明書とともに送信された中間証明書のいずれかを検証する場合、アプライアンスは証明書チェーンを信頼し、ハンドシェイクは成功します。

ただし、クライアントがハンドシェイクで一連の証明書を送信する場合、その証明書が SSL 仮想サーバーにバインドされている場合にのみ、CRL または OCSP レスポンダーを使用して中間証明書を検証できます。したがって、中間証明書の 1 つが失効しても、ハンドシェイクは成功します。ハンドシェイクの一部として、SSL 仮想サーバはバインドされている CA 証明書のリストを送信します。より厳密に制御するには、その仮想サーバーにバインドされた CA 証明書のいずれかが署名した証明書のみを受け入れるように SSL 仮想サーバーを構成できます。そのためには、`ClientAuthUseBoundCAChain` 仮想サーバーにバインドされた SSL プロファイルの設定を有効にする必要があります。仮想サーバーにバインドされている CA 証明書の 1 つがクライアント証明書に署名していない場合、ハンドシェイクは失敗します。

たとえば、clientcert1 と clientcert2 の 2 つのクライアント証明書が、それぞれ中間証明書 int-CA-A と int-CA-B によって署名されているとします。中間証明書は、ルート証明書 root-CA によって署名されます。int-CA-A とルート CA は SSL 仮想サーバにバインドされます。デフォルトの場合 (ClientAuthuseBoundCachain は無効)、clientcert1 と clientcert2 の両方が受け入れられます。ただし、ClientAuthUseBoundCachain が有効な場合、NetScaler ADC アプライアンスは clientcert1 のみを受け入れます。

CLI を使用して、クライアント証明書の検証をより厳密に制御できるようにする

コマンドプロンプトで､次のように入力します｡`set ssl profile <name> -ClientAuthUseBoundCAChain Enabled`

GUI を使用してクライアント証明書の検証をより厳密に制御できるようにする

1. [システム] > [プロファイル] に移動し、[**SSL** プロファイル] タブを選択して SSL プロファイルを作成するか、既存のプロファイルを選択します。
2. [バインドされた **CA** チェーンを使用したクライアント認証を有効にする] を選択します。

証明書失効リスト

August 15, 2023

CA が発行した証明書は、通常、有効期限が切れるまで有効です。ただし、状況によっては、CA が有効期限が切れる前に発行された証明書を取り消すことがあります。たとえば、所有者の秘密鍵が侵害されたり、会社名や個人の名前が変更されたり、サブジェクトと CA との関係が変わったりします。

証明書失効リスト (CRL) は、無効な証明書をシリアル番号と発行者で識別します。

認証局は定期的に CRL を発行します。CRL を使用して無効な証明書を提示するクライアント要求をブロックするように NetScaler アプライアンスを構成できます。

CA からの CRL ファイルが既にある場合は、それを NetScaler アプライアンスに追加します。更新オプションを設定できます。また、Web ロケーションまたは LDAP ロケーションのいずれかから、指定した間隔で CRL ファイルを自動的に同期するように NetScaler を構成することもできます。アプライアンスは PEM または DER ファイル形式の CRL をサポートします。NetScaler アプライアンスに追加する CRL ファイルのファイル形式を必ず指定してください。

ADC を CA として使用して SSL 展開で使用される証明書を作成したことがある場合は、CRL を作成して特定の証明書を取り消すこともできます。この機能を使用すると、たとえば、NetScaler で作成された自己署名証明書が本番環境で使用されたり、特定の日付以降に使用されたりしないようにすることができます。

注:

デフォルトでは、CRL は NetScaler アプライアンスの/var/netscaler/ssl ディレクトリに保存されます。

ADC アプライアンスで **CRL** を作成します

ADC アプライアンスを使用して CA の役割を果たし、自己署名証明書を作成できるため、次の証明書を取り消すこともできます。

- 作成した証明書。
- CA 証明書を所有している証明書。

アプライアンスは、それらの証明書の CRL を作成する前に、無効な証明書を取り消す必要があります。アプライアンスは、失効した証明書のシリアル番号をインデックスファイルに保存し、証明書を失効させるたびにファイルを更新します。インデックスファイルは、証明書が初めて失効したときに自動的に作成されます。

CLI を使用して証明書を取り消すか、**CRL** を作成する

コマンドプロンプトで、次のコマンドを入力します。


```
create ssl crl <CAcertFile> <CAkeyFile> <indexFile> (-revoke <
   input_filename> | -genCRL <output_filename>)
<!--NeedCopy-->
```

例:

```
create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -revoke Invalid-1

create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -genCRL CRL-1
<!--NeedCopy-->
```

GUI を使用して証明書を取り消すか、CRL を作成する

1. [トラフィック管理] > [SSL] に移動し、[はじめに] グループで [CRL 管理] を選択します。
2. 証明書の詳細を入力し、「操作の選択」リストで「証明書の取り消し」または「CRL の生成」を選択します。

ADC に既存の CRL を追加します

NetScaler アプライアンスで CRL を構成する前に、CRL ファイルが NetScaler アプライアンスにローカルに保存されていることを確認してください。HA セットアップでは、CRL ファイルが両方の ADC アプライアンスに存在し、ファイルへのディレクトリパスが両方のアプライアンスで同じである必要があります。

CLI を使用して NetScaler に CRL を追加します

コマンドプロンプトで次のコマンドを入力して NetScaler に CRL を追加し、構成を確認します。

```
add ssl crl <crlName> <crlPath> [-inform (DER | PEM)]

show ssl crl [<crlName>]
<!--NeedCopy-->
```

例:

```
> add ssl crl crl-one /var/netscaler/ssl/CRL-one -inform PEM

Done

> show ssl crl crl-one

            Name: crl-one   Status: Valid,  Days to expiration: 29
            CRL Path: /var/netscaler/ssl/CRL-one
            Format: PEM     CAcert: samplecertkey
            Refresh: DISABLED
            Version: 1
            Signature Algorithm: sha1WithRSAEncryption
```



```
13              Issuer:  C=US,ST=California,L=Santa Clara,O=NetScaler Inc.,
                   OU=SSL Acceleration,CN=www.ns.com/emailAddress=
                   support@NetScaler appliance.com
14              Last_update:Jun 15 10:53:53 2010 GMT
15              Next_update:Jul 15 10:53:53 2010 GMT
16
17      1)      Serial Number: 00
18              Revocation Date:Jun 15 10:51:16 2010 GMT
19       Done
20  <!--NeedCopy-->
```

GUI を使用して **NetScaler** に **CRL** を追加します

[トラフィック管理] > [**SSL**] > [**CRL**] に移動し、CRL を追加します。

CRL 更新パラメータの設定

CRL は、認証局によって定期的に、または特定の証明書が取り消された直後に生成および公開されます。Citrix では、クライアントが不正な証明書で接続しようとするのを防ぐために、NetScaler アプライアンスの CRL を定期的に更新することをお勧めします。

NetScaler アプライアンスは、Web ロケーションまたは LDAP ディレクトリから CRL を更新できます。更新パラメータと Web ロケーションまたは LDAP サーバを指定する場合、コマンドの実行時に CRL がローカルハードディスクドライブに存在している必要はありません。最初の更新では、CRL File パラメーターで指定されたパスのローカルハードディスクドライブにコピーが格納されます。CRL を格納するためのデフォルトのパスは /var/netscaler/SSL です。

注: リリース 10.0 以降では、CRL を更新する方法はデフォルトでは含まれていません。HTTP または LDAP メソッドを指定します。以前のリリースからリリース 10.0 以降にアップグレードする場合は、メソッドを追加してコマンドを再実行する必要があります。

CLI を使用して **CRL** 自動更新を設定する

コマンドプロンプトで次のコマンドを入力して CRL 自動更新を設定し、構成を確認します。

```
1  set ssl crl <crlName> [-refresh ( ENABLED | DISABLED )] [-CAcert <
       string>] [-server <ip_addr|ipv6_addr|*> | -url <URL>] [-method (
       HTTP | LDAP )] [-port <port>] [-baseDN <string>] [-scope ( Base |
       One )] [-interval <interval>] [-day <positive_integer>] [-time <HH:
       MM>][-bindDN <string>] {
2   -password  }
3    [-binary ( YES | NO )]
4
5  show ssl crl [<crlName>]
6  <!--NeedCopy-->
```


例:

```
1      set CRL crl1  -refresh enabled -method ldap -inform DER -CAcert ca1
           -server 10.102.192.192 -port 389 -scope base -baseDN "cn=
       clnt_rsa4_multicert_der,ou=eng,o=ns,c=in" -time 00:01
2
3      set ssl crl crl1 -refresh enabled -method http -cacert ca1 -port 80
           -time 00:10 -url http://10.102.192.192/crl/ca1.crl
4
5
6      > sh crl
7
8      1)          Name: crl1        Status: Valid,     Days to expiration:
           355
9                   CRL Path: /var/netscaler/ssl/crl1
10                  Format: PEM       CAcert: ca1
11                  Refresh: ENABLED          Method: HTTP
12                  URL: http://10.102.192.192/crl/ca1.crl
                        Port:80
13                  Refresh Time: 00:10
14                  Last Update: Successful, Date:Tue Jul  6 14:38:13 2010
15     Done
16 <!--NeedCopy-->
```

GUI を使用して **LDAP** または **HTTP** を使用して **CRL** 自動更新を設定する

1. [トラフィック管理] **> [SSL] > [CRL]** に移動します。
2. CRL を開き、「**CRL** 自動更新を有効にする」を選択します。

> 注
>
> : CRL の［最終更新日時］フィールドで指定された実際の更新時刻よりも前に新しい **CRL** が外部リポジトリで更新された場合は、次の操作を行う必要があります。
>
> NetScaler アプライアンスの CRL を直ちに更新してください。

最終更新時刻を表示するには、CRL を選択し、[詳細] をクリックします。

CRL を同期する

NetScaler アプライアンスは、最近配布された CRL を使用して、証明書が失効したクライアントが安全なリソースにアクセスできないようにします。

CRL が頻繁に更新される場合、NetScaler アプライアンスはリポジトリから最新の CRL を取得する自動メカニズムを必要とします。指定した更新間隔で CRL を自動的に更新するようにアプライアンスを設定できます。

アプライアンスは、定期的に更新する必要がある CRL の内部リストを管理しています。指定された間隔で、アプライアンスはリストをスキャンして、更新する必要のある CRL を探します。次に、リモートの LDAP サーバーまたは HTTP サーバーに接続し、最新の CRL を取得して、ローカル CRL リストを新しい CRL で更新します。

> 注:
>
> CA 証明書が仮想サーバーにバインドされているときに CRL チェックが必須に設定されていて、最初の CRL 更新が失敗した場合、接続に対して次のアクションが実行されます。CRL
>
> と同じ発行者によるすべてのクライアント認証接続は、CRL が正常に更新されるまで REVOKED として拒否されます。

CRL 更新を実行する間隔を指定できます。正確な時間を指定することもできます。

CLI を使用して **CRL** 自動更新を同期する

コマンドプロンプトで、次のコマンドを入力します。

```
set ssl crl <crlName> [-interval <interval>] [-day <integer>] [-time <
    HH:MM>]
<!--NeedCopy-->
```

例:

```
set ssl crl CRL-1 -refresh ENABLE -interval MONTHLY -days 10 -time
    12:00
<!--NeedCopy-->
```

GUI を使用して **CRL** 更新を同期する

1. [トラフィック管理] **> [SSL] > [CRL]** に移動します。
2. CRL を開き、「**CRL** 自動更新を有効にする」を選択し、間隔を指定します。

証明書失効リストを使用してクライアント認証を実行する

NetScaler アプライアンスに証明書失効リスト（CRL）が存在する場合、CRL チェックの実行が必須に設定されているかオプションに設定されているかにかかわらず、CRL チェックが実行されます。

ハンドシェイクの成否は、次の要因の組み合わせによって決まります。

- CRL チェックのルール
- クライアント証明書チェックのルール
- CA 証明書に設定された CRL の状態

次の表は、失効した証明書を含むハンドシェイクの可能な組み合わせの結果を示しています。

表 1. 失効した証明書を使用したクライアントとのハンドシェイクの結果

CRL チェックのルール	クライアント証明書チェックのルール	CA 証明書に設定された CRL の状態	証明書が取り消された場合のハンドシェイクの結果
オプション	オプション	行方不明です	成功
オプション	固定	行方不明です	成功
オプション	固定	プレゼント	失敗
固定	オプション	行方不明です	成功
固定	固定	行方不明です	失敗
固定	オプション	プレゼント	成功
固定	固定	プレゼント	失敗
オプション/必須	オプション	期限切れ	成功
オプション/必須	固定	期限切れ	失敗

> 注:
>
> - CRL チェックはデフォルトではオプションです。オプションから必須に、またはその逆に変更するには、まず証明書を SSL 仮想サーバーからバインド解除し、オプションを変更した後に証明書を再バインドする必要があります。
> - `sh ssl vserver` コマンドの出力で、「OCSP チェック: オプション」は、CRL チェックもオプションであることを意味します。CRL チェック設定は、CRL チェックが必須に設定されている場合にのみ、`sh ssl vserver` コマンドの出力に表示されます。CRL チェックがオプションに設定されている場合、CRL チェックの詳細は表示されません。

CLI を使用して CRL チェックを設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
bind ssl vserver <vServerName> -certkeyName <string> [(-CA -crlCheck (
    Mandatory | Optional ))]
sh ssl vserver
<!--NeedCopy-->
```

例:

```
bind ssl vs v1 -certkeyName ca -CA -crlCheck mandatory
> sh ssl vs v1

Advanced SSL configuration for VServer v1:

DH: DISABLED
DH Private-Key Exponent Size Limit: DISABLED
```



```
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: ENABLED TLSv1
    .2: ENABLED
Push Encryption Trigger: Always
Send Close-Notify: YES

ECC Curve: P_256, P_384, P_224, P_521

1) CertKey Name: ca CA Certificate CRLCheck: Mandatory CA_Name Sent

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用して **CRL** チェックを設定

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、SSL 仮想サーバーを開きます。
2. 「証明書」セクションをクリックします。
3. 証明書を選択し、OCSP と CRL チェックリストで「CRL 必須」を選択します。

失効した証明書または有効な証明書とのハンドシェイクの結果

CRL チェックのルール	クライアント証明書チェックのルール	CA 証明書に設定された CRL の状態	証明書が失効した状態でのハンドシェイクの結果	有効な証明書を使用したハンドシェイクの結果
固定	固定	プレゼント	失敗	成功
固定	固定	期限切れ	失敗	失敗
固定	固定	行方不明です	失敗	失敗
固定	固定	未定義	失敗	失敗
オプション	固定	プレゼント	失敗	成功

CRL チェックのルール	クライアント証明書チェックのルール	CA 証明書に設定された CRL の状態	証明書が失効した状態でのハンドシェイクの結果	有効な証明書を使用したハンドシェイクの結果
オプション	固定	期限切れ	成功	成功
オプション	固定	行方不明です	成功	成功
オプション	固定	未定義	成功	成功
固定	オプション	プレゼント	成功	成功
固定	オプション	期限切れ	成功	成功
固定	オプション	行方不明です	成功	成功
固定	オプション	未定義	成功	成功
オプション	オプション	プレゼント	成功	成功
オプション	オプション	期限切れ	成功	成功
オプション	オプション	行方不明です	成功	成功
オプション	オプション	未定義	成功	成功

OCSP で証明書のステータスを監視する

August 15, 2023

Online Certificate Status Protocol (OCSP) は、クライアント SSL 証明書の状態を判断するために使用されるインターネットプロトコルです。NetScaler アプライアンスは、RFC 2560 で定義されている OCSP をサポートしています。OCSP には、タイムリーな情報の点で、証明書失効リスト (CRL) よりも大きな利点があります。クライアント証明書の最新の失効ステータスは、多額の資金や高額株式取引を含む取引で特に役立ちます。また、使用するシステムリソースとネットワークリソースも少なくなります。NetScaler の OCSP の実装には、リクエストのバッチ処理と応答キャッシュが含まれます。

OCSP の実装

NetScaler アプライアンスの OCSP 検証は、SSL ハンドシェイク中にアプライアンスがクライアント証明書を受け取ったときに開始されます。証明書を検証するために、アプライアンスは OCSP 要求を作成し、それを OCSP レスポンダに転送します。そのために、アプライアンスはローカルに設定された URL を使用します。アプライアンスがサーバーからの応答を評価し、トランザクションを許可するか拒否するかを決定するまで、トランザクションは中断状態になります。サーバからの応答が設定された時間を超えて遅延し、他の応答側が設定されていない場合、アプライ

アンスは OCSP チェックがそれぞれオプションに設定されているか必須に設定されているかに応じて、トランザクションを許可するか、エラーを表示します。

アプライアンスは、OCSP 要求のバッチ処理と OCSP 応答のキャッシュをサポートしているため、OCSP 応答側の負荷が軽減され、応答が速くなります。

OCSP リクエストのバッチング

アプライアンスはクライアント証明書を受信するたびに、OCSP レスポンダに要求を送信します。OCSP レスポンダーの過負荷を避けるため、アプライアンスは同じリクエストで複数のクライアント証明書のステータスを照会できます。この機能を効率的に機能させるには、バッチの作成を待っている間に 1 つの証明書の処理が過度に遅延しないように、タイムアウトを定義する必要があります。

OCSP 応答キャッシュ

OCSP 応答側から受信した応答をキャッシュすると、クライアントへの応答が速くなり、OCSP 応答側の負荷が軽減されます。OCSP レスポンダからクライアント証明書の失効ステータスを受信すると、アプライアンスは事前に定義された時間だけ応答をローカルにキャッシュします。SSL ハンドシェイク中にクライアント証明書を受信すると、アプライアンスはまずローカルキャッシュにこの証明書のエントリがないか確認します。（キャッシュのタイムアウト制限内の）まだ有効なエントリが見つかると、そのエントリが評価され、クライアント証明書が承認または拒否されます。証明書が見つからない場合、アプライアンスは OCSP 応答側に要求を送信し、その応答を設定された期間ローカルキャッシュに保存します。

> 注: リリース 12.1 ビルド 49.x から、キャッシュのタイムアウト制限が最大 43200 分 (30 日) に延長されました。以前の制限は 1440 分 (1 日) でした。制限を増やすと、OCSP サーバーでの検索回数が減り、ネットワークやその他の問題が原因で OCSP サーバーにアクセスできなくなった場合に SSL/TLS 接続が失敗するのを防ぐことができます。

OCSP レスポンダ設定

OCSP の設定には、OCSP レスポンダーを追加し、OCSP レスポンダーを認証局 (CA) 証明書にバインドし、証明書を SSL 仮想サーバーにバインドする必要があります。すでに設定されている OCSP レスポンダに別の証明書をバインドする必要がある場合は、まずレスポンダをバインド解除してから、レスポンダを別の証明書にバインドする必要があります。

CLI を使用して **OCSP** レスポンダーを追加する

コマンドプロンプトで、次のコマンドを入力して OCSP を構成し、構成を確認します。


```
add ssl ocspResponder <name> -url <URL> [-cache ( ENABLED | DISABLED )
    [-cacheTimeout <positive_integer>]] [ -batchingDepth <
    positive_integer>][-batchingDelay <positive_integer>] [-resptimeout
    <positive_integer>] [-responderCert <string> | -trustResponder] [-
    producedAtTimeSkew <positive_integer>][-signingCert <string>][-
    useNonce ( YES | NO )][ -insertClientCert( YES | NO )]
<!--NeedCopy-->
```

```
bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]
<!--NeedCopy-->
```

```
bind ssl vserver <vServerName>@ (-certkeyName <string> ( CA [-ocspCheck
     ( Mandatory | Optional )]))
<!--NeedCopy-->
```

```
show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

例:

```
add ssl ocspResponder ocsp_responder1 -url "http:// www.myCA.org:80/
    ocsp/" -cache ENABLED -cacheTimeout 30 -batchingDepth 8 -
    batchingDelay 100 -resptimeout 100 -responderCert responder_cert -
    producedAtTimeSkew 300 -signingCert sign_cert  -insertClientCert YES
<!--NeedCopy-->
```

```
bind ssl certKey ca_cert -ocspResponder ocsp_responder1 -priority 1
<!--NeedCopy-->
```

```
bind ssl vserver vs1 -certkeyName ca_cert -CA -ocspCheck Mandatory
<!--NeedCopy-->
```

```
sh ocspResponder ocsp_responder1

    1)Name: ocsp_responder1
    URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
    Caching: Enabled        Timeout: 30 minutes
    Batching: 8 Timeout: 100 mS
    HTTP Request Timeout: 100mS
    Request Signing Certificate: sign_cert
    Response Verification: Full, Certificate: responder_cert
    ProducedAt Time Skew: 300 s
    Nonce Extension: Enabled
     Client Cert Insertion: Enabled
    Done
<!--NeedCopy-->
```

```
show certkey ca_cert

    Name: ca_cert     Status: Valid,   Days to expiration:8907
```



```
    Version: 3
    …

    1)  VServer name: vs1      CA Certificate
    1)  OCSP Responder name: ocsp_responder1     Priority: 1
Done
<!--NeedCopy-->
```

```
sh ssl vs vs1

    Advanced SSL configuration for VServer vs1:
    DH: DISABLED
    …

    1) CertKey Name: ca_cert CA Certificate OCSPCheck: Mandatory
    1) Cipher Name: DEFAULT
      Description: Predefined Cipher Alias
    Done
<!--NeedCopy-->
```

CLI を使用して **OCSP** レスポンダーを変更する

レスポンダー名は変更できません。他のすべてのパラメータは、`set ssl ocspResponder` コマンドを使用して変更できます。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
set ssl ocspResponder <name> [-url <URL>] [-cache ( ENABLED | DISABLED)
    ] [-cacheTimeout <positive_integer>] [-batchingDepth <
    positive_integer>] [-batchingDelay <positive_integer>] [-resptimeout
     <positive_integer>] [ -responderCert <string> | -trustResponder][-
    producedAtTimeSkew <positive_integer>][-signingCert <string>] [-
    useNonce ( YES | NO )]

unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

GUI を使用して **OCSP** レスポンダを設定する

1. [トラフィック管理] > [**SSL**] > [**OCSP** レスポンダー] に移動し、OCSP レスポンダーを構成します。
2. [トラフィック管理] > [**SSL**] > [証明書] に移動し、証明書を選択し、[アクション] リストで [**OCSP** バインディング] を選択します。OCSP レスポンダーをバインドします。

3. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開き、[証明書] セクションをクリックして CA 証明書をバインドします。
4. オプションで、「**OCSP** 必須」を選択します。

OCSP ステープリング

August 15, 2023

CRL および OCSP の NetScaler ADC 実装では、クライアント証明書の失効ステータスのみが報告されます。SSL ハンドシェイク中に受信したサーバー証明書の失効ステータスを確認するには、クライアントが認証局に要求を送信する必要があります。

トラフィックが多いウェブサイトでは、多くのクライアントが同じサーバー証明書を受け取ります。各クライアントがサーバー証明書の失効ステータスのクエリを送信すると、認証局は証明書の有効性をチェックする OCSP 要求で溢れかえってしまいます。

OCSP ホチキス止めソリューション

不要な輻輳を回避するために、NetScaler ADC アプライアンスは OCSP ホチキス止めをサポートするようになりました。つまり、SSL ハンドシェイク時に、アプライアンスは OCSP レスポンダからの応答を検証した後、サーバ証明書のステータスをクライアントに送信できるようになりました。サーバ証明書のステータスは、SSL ハンドシェイクの一環としてアプライアンスがクライアントに送信する証明書に「ホチキス止め」されます。OCSP ホチキス止め機能を使用するには、SSL 仮想サーバーで有効にし、アプライアンスに OCSP レスポンダーを追加する必要があります。

> メモ
>
> - リリース 13.1-30.x から、次の条件が満たされると、すべての中間証明書に OCSP 応答拡張が含まれるようになりました。
>
>   - TLS 1.3 protocol is used
>   - Client sends a status request
>
>   以前は、サーバー証明書のみが、クライアントからのステータス要求への応答にこの拡張を含んでいました。
>
> - 他のプロトコル (TLS 1.2 を含む) では、サーバーはサーバー証明書のみの OCSP 応答を送信します。つまり、RFC 6961 は TLS 1.2 プロトコルではサポートされていません。
> - NetScaler ADC アプライアンスは、RFC 6066 で定義されている OCSP ホチキス止めをサポートします。

- OCSP ホチキス止めは、NetScaler ADC アプライアンスのフロントエンドでのみサポートされています。
- TLS 1.3 プロトコルが使用されている場合、ADC アプライアンスは次のように動作します。キャッシュされた OCSP 応答が無効（空または期限切れ）の場合、要求は OCSP レスポンダに送信されますが、SSL ハンドシェイクは応答を待たずに完了します。応答が受信されると、キャッシュされ、クライアントからの今後のステータス要求に使用できます。
- OCSP ホチキス止めに対する CCitrix ADC サポートは、TLS プロトコルバージョン 1.0 以上を使用するハンドシェイクに制限されています。

サーバー証明書の **OCSP** 応答キャッシング

> 注
>
> リリース 13.1-30.x 以降、TLS 1.3 プロトコルが使用されている場合、OCSP 応答はサーバー証明書とすべての中間証明書に対してキャッシュされます。

SSL ハンドシェイク中に、クライアントがサーバ証明書の失効ステータスを要求すると、アプライアンスはまずローカルキャッシュでこの証明書のエントリを確認します。有効なエントリが見つかると、そのエントリが評価され、サーバ証明書とそのステータスがクライアントに提示されます。失効ステータスエントリが見つからない場合、アプライアンスはサーバー証明書の失効ステータスの要求を OCSP レスポンダに送信します。応答を受信すると、証明書と失効ステータスをクライアントに送信します。次の更新フィールドが OCSP 応答に存在する場合、応答は設定された期間（タイムアウトフィールドに指定された値）だけキャッシュされます。

> 注: リリース 12.1 ビルド 49.x から、タイムアウトが切れる前でも、OCSP レスポンダからサーバ証明書のキャッシュされたレスポンスをクリアできます。以前は、設定されたタイムアウトが終了するまで、証明書とキーのペアのキャッシュされたステータスを破棄することはできませんでした。

CLI を使用してキャッシュされたステータスをクリアするには、コマンドプロンプトで次のように入力します。

```
clear ssl certKey <certkey name> -ocspstaplingCache
<!--NeedCopy-->
```

例:

```
clear ssl certKey s1 -ocspstaplingCache
<!--NeedCopy-->
```

GUI を使用してキャッシュされたステータスをクリアするには

1. GUI で、[トラフィック管理] > [**SSL**] > [証明書] > [**CA** 証明書] に移動します。
2. 詳細ペインで、証明書を選択します。
3. **[**アクションを選択] リストで、[クリア] を選択します。確認を求められたら、[はい] をクリックします。

OCSP ホチキス止め構成

OCSP ホチキス止めの設定には、機能の有効化と OCSP の設定が含まれます。OCSP を構成するには、OCSP レスポンダーを追加し、OCSP レスポンダーを CA 証明書にバインドし、証明書を SSL 仮想サーバーにバインドする必要があります。

> 注:
>
> HTTP ベースの URL のみを持つ OCSP レスポンダがサポートされています。

CLI を使用して OCSP ホチキス止めを有効にする

コマンドプロンプトで入力します。

```
set ssl vserver <name> -ocspstapling [ENABLED | DISABLED]
<!--NeedCopy-->
```

例:

```
set ssl vserver vip1 -ocspStapling ENABLED
Done

sh ssl vserver vip1

     Advanced SSL configuration for VServer vip1:
     DH: DISABLED
     DH Private-Key Exponent Size Limit: DISABLED Ephemeral RSA:
        ENABLED Refresh Count: 0
     Session Reuse: ENABLED Timeout: 120 seconds
     Cipher Redirect: DISABLED
     SSLv2 Redirect: DISABLED
     ClearText Port: 0
     Client Auth: DISABLED
     SSL Redirect: DISABLED
     Non FIPS Ciphers: DISABLED
     SNI: ENABLED
     OCSP Stapling: ENABLED
     SSLv2: DISABLED SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED
          TLSv1.2: ENABLED
     Push Encryption Trigger: Always
     Send Close-Notify: YES

     ECC Curve: P_256, P_384, P_224, P_521

    1) CertKey Name: server_certificate1 Server Certificate

    1) Cipher Name: DEFAULT
     Description: Default cipher list with encryption strength >= 128
        bit
Done
<!--NeedCopy-->
```


注**:** デフォルト（拡張）プロファイルが有効になっている場合は、`set ssl profile <profile name> -ocspStapling [ENABLED | DISABLED]` コマンドを使用して OCSP を有効または無効にします。

GUI を使用して **OCSP** ホチキス止めを有効にする

1. トラフィック管理 > **SSL** > 仮想サーバーに移動します。
2. 仮想サーバーを開き、[**SSL** パラメータ] で [**OCSP Stapling**] を選択します。

OCSP 構成

OCSP レスポンダは、OCSP ホチキス止め要求を送信するために動的または手動で追加されます。サーバー証明書の OCSP URL に基づいてサーバー証明書とその発行者証明書を追加すると、内部レスポンダーが動的に追加されます。手動 OCSP レスポンダは CLI または GUI から追加されます。サーバー証明書の OCSP 要求を送信するために、NetScaler ADC アプライアンスは、発行者証明書にバインドするときに割り当てられた優先順位に基づいて OCSP レスポンダーを選択します。応答側が OCSP ホチキス止め要求の送信に失敗した場合、次に高い優先度を持つ応答側が要求の送信対象として選択されます。たとえば、手動で構成されたレスポンダーが 1 つだけ失敗し、動的にバインドされたレスポンダーが存在する場合、そのレスポンダーが OCSP 要求の送信用に選択されます。

OCSP URL が HTTP 以外の場合、内部 OCSP レスポンダは作成されません。

> 注
>
> 手動で追加された OCSP レスポンダーは、動的に追加されたレスポンダーよりも優先されます。

手動で作成された **OCSP** レスポンダーと内部で作成された **OCSP** レスポンダーの違い

手動で作成された **OCSP** レスポンダー	内部 **(**動的**)** に作成された **OCSP** レスポンダ
手動で作成され、優先度付きで発行者証明書に明示的にバインドされます。	サーバー証明書とその発行者証明書 (CA 証明書) を追加しながら、デフォルトで作成およびバインドされます。名前は「ns_internal_」で始まります。
1 ～127 の優先度は、設定された応答側用に予約されています。	優先順位は 128 から自動的に割り当てられます。
URL とバッチ深度は変更できます。	URL とバッチ深度は変更できません。
直接削除されました。	サーバー証明書または CA 証明書を削除した場合にのみ削除されます。
任意の CA 証明書にバインドできます。	デフォルトでは、1 つの CA 証明書にバインドされています。他の CA 証明書にはバインドできません。

構成 (ns.conf) に保存されます。	追加コマンドは設定に保存されません。set コマンドだけが保存されます。
3 つの OCSP レスポンダを、それぞれ 1、2、および 3 の同じ発行者証明書にバインドし、後で優先度 2 のバインドを解除しても、他の優先度は影響を受けません。	3 つの OCSP レスポンダが、それぞれ 128、129、および 130 の優先度を持つ発行者証明書に自動的にバインドされます。優先度 129 でバインドされたレスポンダーの作成に使用されたサーバー証明書を削除すると、そのレスポンダーは削除されます。また、次の応答者の優先度（優先度 130）は自動的に 129 に変更されます。

> リクエスト処理の例:

1. 仮想サーバー (VIP1) を追加します。
2. 発行者証明書 (CA1) を追加し、VIP1 にバインドします。
3. 3 つの証明書 S1、S2、および S3 を追加します。内部レスポンダー resp1、resp2、および resp3 はそれぞれデフォルトで作成されます。
4. S3 を VIP1 にバインドします。
5. VIP1 にリクエストが来る。レスポンダー resp3 が選択されています。

内部 OCSP レスポンダを動的に作成するには、アプライアンスに次のものが必要です。

- サーバー証明書の発行者の証明書 (通常は CA 証明書)。
- サーバー証明書の証明書とキーのペア。この証明書には、認証局から提供された OCSP URL が含まれている必要があります。この URL は、動的に追加された内部レスポンダーの名前として使用されます。

内部 OCSP レスポンダには、手動で設定されたレスポンダと同じデフォルト値があります。

> 注:
>
> 内部レスポンダーでは、キャッシュはデフォルトで無効になっています。`set ssl ocspResponder` コマンドを使用してキャッシュを有効にします。

CLI を使用して **OCSP** を構成する

コマンドプロンプトで、次のコマンドを入力して OCSP を構成し、構成を確認します。

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
    string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>]
    [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]] [-bundle ( YES | NO )]

add ssl ocspResponder <name> -url <URL> [-cache ( ENABLED | DISABLED )
    [-cacheTimeout <positive_integer>]] [-resptimeout <positive_integer
    >] [-responderCert <string> | -trustResponder] [-producedAtTimeSkew
```



```
    <positive_integer>][-signingCert <string>][-useNonce ( YES | NO )][
    -insertClientCert ( YES | NO )]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

パラメーター:

HttpMethod:

OCSP 要求の送信に使用される HTTP メソッド。255 バイト未満のリクエストでは、OCSP サーバーへのクエリに HTTP GET メソッドを設定できます。GET メソッドを指定したが、長さが 255 バイトを超える場合、アプライアンスはデフォルトのメソッド（POST）を使用します。

可能な値:GET、POST

デフォルト値:POST

OCSPurlResolveTimeout:

OCSP URL の解決を待機する時間 (ミリ秒)。この時間が経過すると、次に高い優先度を持つ応答者が選択されます。すべてのレスポンダーが失敗した場合、仮想サーバーの設定に応じて、エラーメッセージが表示されるか、接続が切断されます。

最小値:100

最大値:2000

例**:**

```
add ssl certkey root_ca1  - cert root_cacert.pem
add ssl ocspResponder ocsp_responder1 -url "http:// www.myCA.org:80/
    ocsp/" -cache ENABLED -cacheTimeout 30  -resptimeout 100 -
    responderCert responder_cert -producedAtTimeSkew 300 -signingCert
    sign_cert  -insertClientCert YES
bind ssl certKey root_ca1 -ocspResponder ocsp_responder1 -priority 1
sh ocspResponder ocsp_responder1
    1)Name: ocsp_responder1
    URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
    Caching: Enabled        Timeout: 30 minutes
    Batching: 8 Timeout: 100 mS
    HTTP Request Timeout: 100mS
    Request Signing Certificate: sign_cert
    Response Verification: Full, Certificate: responder_cert
    ProducedAt Time Skew: 300 s
    Nonce Extension: Enabled
     Client Cert Insertion: Enabled
    Done

show certkey root_ca1
```



```
    Name: root_ca1     Status: Valid,   Days to expiration:8907
    Version: 3
    …
    1)  OCSP Responder name: ocsp_responder1     Priority: 1
    Done
<!--NeedCopy-->
```

CLI を使用して OCSP を変更する

OCSP レスポンダの名前は変更できませんが、`set ssl ocspResponder` コマンドを使用して他のパラメータを変更することはできます。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
set ssl ocspResponder <name> [-url <URL>] [-cache ( ENABLED | DISABLED)
    ] [-cacheTimeout <positive_integer>]  [-resptimeout <
    positive_integer>] [ -responderCert <string> | -trustResponder][-
    producedAtTimeSkew <positive_integer>][-signingCert <string>] [-
    useNonce ( YES | NO )]

unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

GUI を使用して OCSP を構成する

1. [トラフィック管理] > [**SSL**] > [**OCSP** レスポンダー] に移動し、OCSP レスポンダーを構成します。
2. [トラフィック管理] > [**SSL**] > [証明書] に移動し、証明書を選択し、[アクション] リストで [**OCSP** バインディング] を選択します。**OCSP** レスポンダーをバインドします。
3. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開き、[証明書] セクションをクリックして CA 証明書をバインドします。
4. 必要に応じて、[**OCSP** 必須] を選択します。

> 注:
>
> `add ssl ocspResponder`コマンドおよび`set ssl ocspResponder`コマンドの insert クライアント証明書パラメータは無効になりました。つまり、パラメータは設定時に無視されます。

NetScaler ADC アプライアンスで利用可能な暗号

January 11, 2024

NetScaler ADC アプライアンスには、事前定義された一連の暗号グループが付属しています。DEFAULT 暗号グループに含まれない暗号を使用するには、SSL 仮想サーバーに明示的にバインドする必要があります。また、ユーザー定義の暗号グループを作成して、SSL 仮想サーバーにバインドすることもできます。ユーザー定義の暗号グループの作成の詳細については、ADC アプライアンスでのユーザー定義の暗号グループの構成を参照してください。

> メモ
>
> - リリース 13.0 ビルド 71.x 以降では、TLS1.3 ハードウェアアクセラレーションが次のプラットフォームでサポートされています。
>
>   - MPX 5900
>   - MPX/SDX 8900
>   - MPX/SDX 9100
>   - MPX/SDX 15000
>   - MPX/SDX 15000-50G
>   - MPX/SDX 16000
>   - MPX/SDX 26000
>   - MPX/SDX 26000-50S
>   - MPX/SDX 26000-100G
>
>   - TLSv1.3 プロトコルのソフトウェアのみのサポートは、NetScaler FIPS アプライアンスを除く他のすべての NetScaler ADC MPX および SDX アプライアンスで利用できます。
> - TLSv1.3 は、拡張プロファイルでのみサポートされます。拡張プロファイルを有効にするには、拡張プロファイルの有効化を参照してください。
> - TLS1.3 を使用するには、RFC 8446 仕様に準拠したクライアントを使用する必要があります。
> - RC4 暗号は、NetScaler ADC アプライアンスのデフォルトの暗号グループには含まれません。ただし、N3 ベースのアプライアンス上のソフトウェアではサポートされています。ハンドシェイクを含む RC4 暗号化はソフトウェアで行われます。
> - この暗号は安全ではないと見なされ、RFC 7465 で非推奨になっているため、使用しないことをお勧めします。
> - アプライアンスに N3 チップが搭載されているかどうかを確認するには、「show hardware」コマンドを使用します。

```
1 sh hardware
2
3 Platform: NSMPX-22000 16\*CPU+24\*IX+12\*E1K+2\*E1K+4*CVM N3 2200100
4
```



```
Manufactured on: 8/19/2013

CPU: 2900MHZ

Host Id: 1006665862

Serial no: ENUK6298FT

Encoded serial no: ENUK6298FT
<!--NeedCopy-->
```

- フロントエンド (仮想サーバー) でデフォルトでバインドされている暗号スイートに関する情報を表示するには、次のように入力します。`sh cipher DEFAULT`
- バックエンドで (サービスに) デフォルトでバインドされている暗号スイートに関する情報を表示するには、以下のように入力します。`sh cipher DEFAULT_BACKEND`
- アプライアンスに定義されているすべての暗号グループ (エイリアス) に関する情報を表示するには、次のように入力します。`sh cipher`
- 特定の暗号グループに含まれるすべての暗号スイートに関する情報を表示するには、`sh cipher <alias name>`と入力します。たとえば、sh 暗号 ECDHE です。

以下のリンクは、さまざまな NetScaler ADC プラットフォームおよび外部ハードウェアセキュリティモジュール（HSM）でサポートされている暗号スイートの一覧です。

- **NetScaler MPX/SDX Intel** ルイスバーグアプライアンス：NetScaler MPX/SDX Intel ルイスバーグ SSL チップベースのアプライアンスでの暗号サポート
- **NetScaler MPX/SDX**（**N3**）アプライアンス：NetScaler MPX/SDX（N3）アプライアンスでの暗号サポート
- **NetScaler MPX/SDX Intel Coleto** アプライアンス: NetScaler MPX/SDX Intel Coleto SSL チップベースのアプライアンスでの暗号サポート
- **NetScaler VPX** アプライアンス：NetScaler VPX アプライアンスでの暗号サポート
- **NetScaler MPX/SDX 14000 FIPS** アプライアンス：NetScaler MPX/SDX 14000 FIPS アプライアンスでの暗号サポート
- 外部 **HSM (Thales/**セーフネット**)**: 外部 HSM でサポートされる暗号 (Thales/Safenet)
- **NetScaler MPX/SDX**（**N2**）アプライアンス：NetScaler MPX/SDX（N2）アプライアンスでの暗号サポート
- **NetScaler MPX 9700 FIPS** アプライアンス：ファームウェア 2.2 を搭載した NetScaler ADC MPX 9700 FIPS での暗号サポート
- **NetScaler VPX FIPS** および **MPX FIPS** アプライアンス：**NetScaler VPX FIPS** および **MPX FIPS** アプライアンスでの暗号サポート

> 注:
>
> DTLS 暗号のサポートについては、NetScaler VPX、MPX、および SDX アプライアンスでの DTLS 暗号のサポートを参照してください。

表 **1-**仮想サーバー**/**フロントエンドサービス**/**内部サービスのサポート:

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
–	–	–	–	–	–
TLS 1.3	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	未サポート	13.1 全ビルド
	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	未サポート	13.0 すべてのビルド
	12.1—50.x (TLS1.3-CHACHA20-POLY1305-SHA256 を除く)	12.1—50.x (TLS1.3-CHACHA20-POLY1305-SHA256 を除く)	12.1–50.x	未サポート	12.1–50.x
TLS 1.1/1.2	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド
	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 〜50.x
	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x
	11.1 すべてのビルド	11.1 すべてのビルド	11.1 すべてのビルド	11.1—56.x MPX 5900/8900 と MPX 15000-50G、11.1-60.x MPX 26000-100G
		11.0 すべてのビルド	11.0 すべてのビルド	11.0 すべてのビルド	11.0—70.x (MPX 5900/8900 のみ)
	10.5 すべてのビルド	10.5 すべてのビルド	10.5—57.x	10.5—59.1359.e	10.5—67.x、10.5-63.47 (MPX 5900/8900 のみ)
ECDHE/DHE (例 TLS1-ECDHE-RSA-AES128-SHA)	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド
13.1 すべてのビルド	13.1	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてビルド
12.1 すべてのビルド	12.1 すべてのビルド	12.1 すべてのビルド	12.1 すべてのビルド	MPX 5900/8900 の 12.1 すべてのビルド、MPX 15000-50G および MPX 26000-100G の 12.1-50.x
	12.0 すべてのビルド	12.0 すべてのビルド	12.0 すべてのビルド	12.0 MPX 5900/8900 のすべてのビルド	12.0 MPX 5900/8900 のすべてのビルド	12.0 すべてのビルド	12.0 MPX 5900/8900 のすべてのビルド	12.0 すべてのビルド	12.0 MPX 5900/8900 0-57.xMPX 15000-50G、12.0-60.x MPX 26000-100G
	11.1 すべてのビルド	11.1 すべてのビルド	11.1 すべてのビルド	11.1—51.x	11.1—56.x MPX 5900/8900 と MPX 15000-50G、11.1-60.x MPX 26000-100G			11.0 すべてのビルド	11.0 すべてのビルド	11.0 すべてのビルド	11.0 すべてのビルド	11.0 すべてのビルド
11.0 すべてのビルド	11.0 すべてのビルド	11.0 すべてのビルド	11.0 すべてのビルド	11.0 すべてのビルド 0 全ビルド		11.0—70.114 (MPX 5900/8900 のみ)
	10.5—53.x	10.5—53.x	10.5 全ビルド		10.5—67.x、10.5-63.47 (MPX 5900/8900 のみ)
AES-GCM (例 TLS1.2-AES128-GCM-SHA256)	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド
13.|13.0 すべてのビルド | 13.0 すべてのビルド | 13.0 すべてのビルド | 13.0 すべてのビルド | 13.0 すべてのビルド

|
|| 12.1 すべてのビルド | 12.1 すべてのビルド | 12.1 すべてのビルド | 12.1 すべてのビルド | 12.1 MPX 5900/8900 のすべてのビルド | 12.1 MPX 15000-50G と MPX 26000-100G | || 12.0 すべてのビルド | 12.0 すべてのビルド | 12.0 すべてのビルド | 12.0 すべてのビルド | 12.0 すべてのビルド | 12.0 すべてのビルド
| 12.0 すべてのビルド | 12.0 すべてのビルド | 12.0 すべてのビルド | 12.0 すべてのビルド | 12.0 すべてのビルド | | 12.0 すべて MPX 5900/8900 用のビルド、MPX 15000-50G 用の 12.0-57.x、MPX 26000-100G 用の 12.0-60.x | 11.1 すべてのビルド
11.1 すべてのビルド	11.1 すべてのビルド	11.1—51.x（注を参照）	MPX 5900/8900 および MPX 15000-50G の場合は 11.1—56.x、MPX 用 11.1-60.x、MPX 用 11.1-60.x 26000-100G
11.0 すべてのビルド	11.0 すべてのビルド	11.0—66.x		11.0—70.114 (MPX 5900/8900 のみ)
	10.5—53.x	10.5—53.x			10.5—67.x、10.5-63.47 (MPX 5900/8900 のみ)
SHA-2 暗号 (例 TLS1.2-AES-128-SHA256)	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド	13.1 すべてのビルド
	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	12.1 すべてのビルド	12.1
すべてのビルド	12.1 すべてのビルド	12.1 すべてのビルド	12.1 MPX 5900/8900 のすべてのビルド、12.1-50.x MPX 15000-50G と MPX 26000-100G
	12.0 すべてのビルド	12.0 すべてのビルド	12.0 すべてのビルド	12.0 すべてのビルド	12.0 すべてのビルド	12.0 すべてのビルド	12.0 すべてのビルド	12.0 すべてのビルド	12.0 すべてのビルド	12.0 すべてのビルド	12.0 すべてのビルドビルド	12.0MPX 5900/8900 用のすべてのビルド、MPX 15000-50G 用の 12.0-57.x、MPX 26000-100G 用の 12.0-60.x
	11.1 すべてのビルド	11.1 すべてのビルド	11.1 すべてのビルド	11.1—52.x	MPX 5900/8900 および MPX 15000-50G 用の 11.1—56.x、MPX 26000-50G 用の 11.1-60.x、MPX 26000-50G の場合は 11.1-60.x、MPX 26000-100G 用の 11.1-60.x 100G
		11.0 すべてのビルド	11.0 すべてのビルド	11.0—66.x		11.0—72.x、11.0-70.114 (MPX 5900/8900 のみ)
	10.5—53.x	10.5—53.x		10.5—67.x、10.5-63.47 (MPX 5900/8900 のみ)
ECDSA (例 TLS1-ECDHE-ECDSA-AES256-SHA)	サポートされていない	13.3 1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド
	サポートされていない	13.0 全ビルド	13.0 全ビルド	13.0 全ビルド
	サポートされていない	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 用全ビルド、12.1-50.x MPX 15000-50G および MPX 26000-100G
	サポートされていない	12.0 全ビルド	12.0-57.x	サポートされていません	12.0 MPX 5900/8900 用のすべてのビルド、MPX 15000-50G 用の 12.0-57.x、MPX 26000-100G 用の 12.0-60.x
	11.1 すべてのビルド			11.1—56.x、11.1-54.126 (ECC カーブ P_256 と P_384 のみがサポートされています)。
CHACHA20	サポートされていない	13.1 全ビルド	13.1 全ビルド	サポートされていない	13.1 全ビルド
	サポートされていない	13.0 全ビルド	13.0 全ビルド	サポートされていない	サポートされていない
12.1 全ビルド	サポートされていない	12.1 全ビルド	サポートされていない	12.1—49.x (のみ MPX 5900/8900)
|| サポートされていない	12.0—56.x	サポートされていない	サポートされていない

表 **2-**バックエンドサービスのサポート**:**

TLS 1.3 はバックエンドではサポートされていません。

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
TLS 1.1/1.2	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド
		13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
		12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド
		12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド
		11.1 全ビルド	11.1 全ビルド	11.1 全ビルド	11.1 全ビルド
		11.0–50.x	11.0–50.x	11.0–66.x	
		10.5–59.x	10.5–59.x		10.5—59.1359.e
ECDHE/DHE (例 TLS1-ECDHE-RSA-AES128-SHA)	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド
		13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
		12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド
		12.0 全ビルド	12.0 全ビルド	12.0–56.x	12.0 全ビルド
		11.1 全ビルド	11.1 全ビルド		11.1–51.x
		11.0–50.x	11.0–50.x		
		10.5–58.x	10.5–58.x		
AES-GCM (例 TLS1.2-AES128-GCM-SHA256)	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド
		13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
		12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド
		12.0 全ビルド	12.0 全ビルド	未サポート	12.0 全ビルド

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
		11.1 全ビルド	11.1 全ビルド		11.1–51.x
SHA-2 暗号 (例 TLS1.2-AES-128-SHA256)	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド
		13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
		12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド
		12.0 全ビルド	12.0 全ビルド	未サポート	12.0 全ビルド
		11.1 全ビルド	11.1 全ビルド		11.1–52.x
ECDSA (例 TLS1-ECDHE-ECDSA-AES256-SHA)	未サポート	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド	13.1 全ビルド
		未サポート	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
		未サポート	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド
		未サポート	12.0 全ビルド	12.0–57.x	未サポート
			11.1–51.x		
CHACHA20	未サポート	13.1 全ビルド	13.1 全ビルド	未サポート	13.1 全ビルド
		未サポート	13.0 すべてのビルド	13.0 すべてのビルド	未サポート
		未サポート	未サポート	12.1 全ビルド	未サポート
		未サポート	未サポート	12.0–56.x	未サポート

サポートされる ECDSA 暗号の詳細なリストについては、ECDSA 暗号スイートのサポートを参照してください。

> メモ
>
> - TLS-fallback_SCSV 暗号スイートは、リリース 10.5 ビルド 57.x 以降のすべてのアプライアンスでサポートされています。

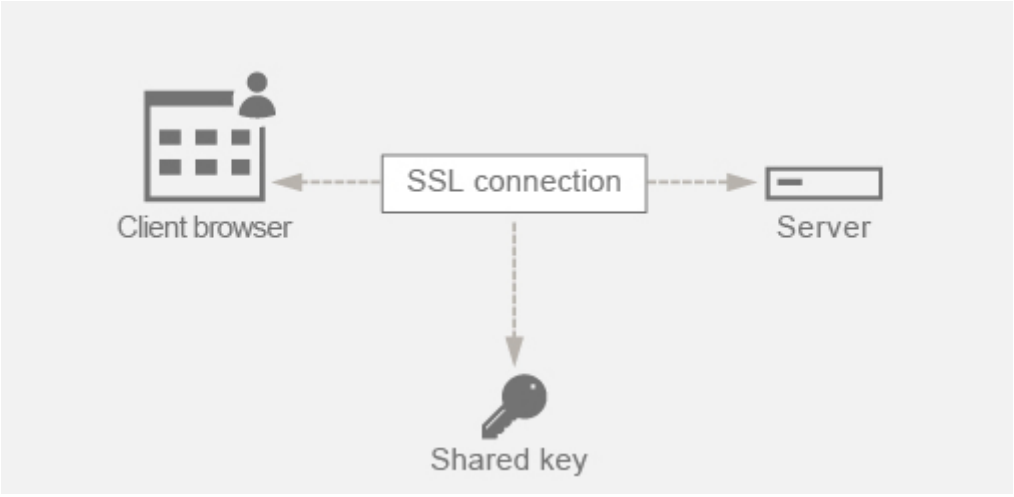
- HTTP 厳密なトランスポートセキュリティ (HSTS) のサポートはポリシーベースです。
- すべての SHA-2 署名付き証明書（SHA256、SHA384、SHA512）は、すべてのアプライアンスのフロントエンドでサポートされています。リリース 11.1 ビルド 54.x 以降では、これらの証明書はすべてのアプライアンスのバックエンドでもサポートされています。リリース 11.0 以前では、すべてのアプライアンスのバックエンドで SHA256 署名付き証明書のみがサポートされています。
- リリース 11.1 ビルド 52.x 以前では、次の暗号は MPX9700 のフロントエンドでのみサポートされています。MPX/SDX 14000 FIPS アプライアンス:
  - TLS1.2-ECDHE-RSA-AES-256-SHA384
  - TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 From release 11.1 build 53.x, and in release 12.0, these ciphers are also supported on the back end.
- すべての ChaCha20-Poly1035 暗号は、SHA-256 ハッシュ関数と TLS 擬似ランダム関数（PSF）を使用します。

Perfect Forward Secrecy (PFS)

Perfect Forward Secrecy は、ウェブサーバーのセッションキーが後の時点で侵害された場合でも、現在の SSL 通信を確実に保護します。

なぜ **Perfect Forward Secrecy (PFS)** が必要なのですか

SSL 接続は、クライアントとサーバー間で渡されるデータを保護するために使用されます。この接続は、クライアントのブラウザとアクセスした Web サーバの間で行われる SSL ハンドシェイクから始まります。このハンドシェイク中に、ブラウザとサーバーが特定の情報を交換してセッションキーに到達します。セッションキーは、通信の残りの部分を通してデータを暗号化する手段として機能します。



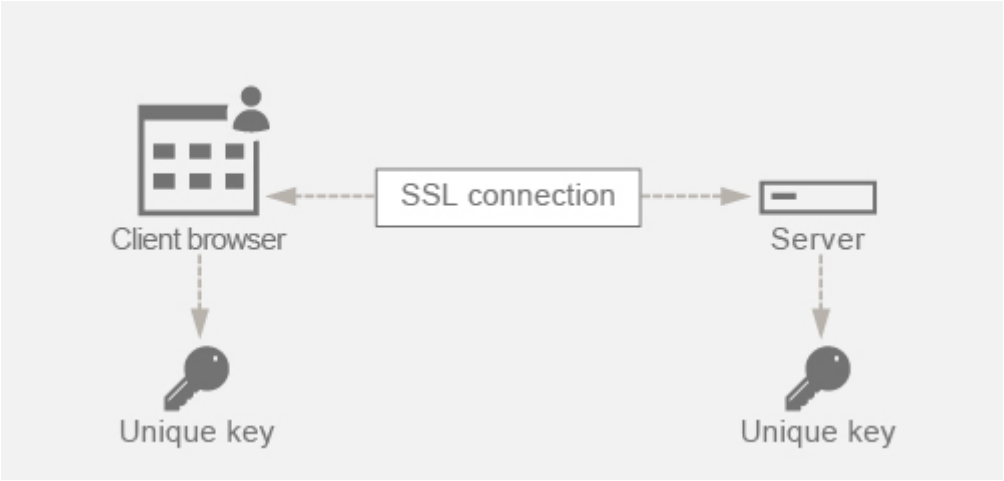
RSA は、鍵交換で最も一般的に使用されるアルゴリズムです。ブラウザはサーバーの公開鍵を使用して暗号化し、プレマスターシークレットを介してサーバーに送信します。この事前マスターシークレットは、セッションキーに到達

するために使用されます。RSA 鍵交換アプローチの問題点は、攻撃者が将来の任意の時点でサーバーの秘密鍵をなんとか入手できた場合、攻撃者はセッション鍵を取得できるプレマスターシークレットを入手できることです。攻撃者がこのセッションキーを使用して、すべての SSL カンバセーションを復号化できるようになりました。その結果、サーバーの盗まれた秘密キーを使用してセッションキーに到達し、保存された履歴カンバセーションも復号化される可能性があるため、以前にセキュリティで保護されていた履歴 SSL 通信はセキュリティで保護されなくなります。

サーバーの秘密鍵が侵害された場合でも、過去の SSL 通信を保護できることが必要です。Perfect Forward Secrecy (PFS) を設定すると、この問題に対処できます。

PFS はどのように役立つのですか

PFS は、クライアントとサーバがセッションごとに新しいキーについて合意し、このセッションキーの計算を秘密にしておくことで、過去の SSL 通信を保護します。これは、サーバーキーの侵害によってセッションキーが侵害されてはならないという基準で機能します。セッションキーは両端で別々に派生し、ワイヤを介して転送されることはありません。セッションキーは、通信が完了すると破棄されます。これらの事実により、誰かがサーバーの秘密鍵にアクセスできたとしても、セッション鍵に到達できないことが保証されます。したがって、過去のデータを復号化することはできません。



例を挙げた説明

PFS の達成に DHE を使用していると仮定します。DH アルゴリズムは、ハッカーがサーバーの秘密キーを入手しても、ハッカーがセッションキーに到達できないようにします。その理由は、セッションキーと乱数 (セッションキーに到達するために使用) は両端で秘密にされ、ネットワーク上で交換されることはないからです。
PFS は、SSL セッションごとに新しい一時キーを作成するエフェメラル Diffie-Hellman キー交換を使用することで実現できます。

セッションごとにキーを作成する反対に、追加の計算が必要になるという点があります。ただし、この問題は、キーサイズが小さい楕円曲線を使用することで解決できます。

NetScaler ADC アプライアンスで **PFS** を構成する

PFS は、DHE または ECDHE 暗号を構成することで、NetScaler ADC 上で構成できます。これらの暗号により、作成されたシークレットセッションキーがワイヤ上で共有されず（DH アルゴリズム）、セッションキーが短時間だけ存続することが保証されます（エフェメラル）。次のセクションでは、両方の構成について説明します。

注**:** DHE の代わりに ECDHE 暗号を使用すると、小さなキーサイズで通信がより安全になります。

GUI を使用して **DHE** を構成する

1. DH キーを生成します。

   a. [トラフィック管理**] > [SSL] > [**ツール**]** に移動します。

   b. 「**Diffie Helman (DH)** キーを作成」をクリックします。

   注**:** 2048 ビット DH キーの生成には最大 30 分かかることがあります。

2. SSL 仮想サーバーの DH Param を有効にし、SSL 仮想サーバーに DH キーを接続します。

   a. [設定**] > [**トラフィック管理**] > [**仮想サーバ] に移動します。

   b. DH を有効にする仮想サーバーを選択します。

   c. [編集] をクリックし、[**SSL** パラメータ] をクリックし、[**DH** パラメータを有効にする] をクリックします。

3. DHE 暗号を仮想サーバーにバインドします。

   a. [設定**] > [**トラフィック管理**] > [**仮想サーバ] に移動します。

   b. DH を有効にする仮想サーバーを選択し、鉛筆アイコンをクリックして編集します。

   c. [詳細設定] で、[**SSL Ciphers**] の横にあるプラスアイコンをクリックし、DHE 暗号グループを選択して [**OK**] をクリックしてバインドします。

   注**:** DHE 暗号が仮想サーバにバインドされた暗号リストの一番上に配置されていることを確認してください。

GUI を使用した **ECDHE** の設定

1. ECC カーブを SSL 仮想サーバーにバインドします。

   a.**[**設定**] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動します。

   b. 編集する SSL 仮想サーバーを選択し、[**ECC Curve**] をクリックして、[バインドの追加] をクリックします。

   c. 必要な ECC カーブを仮想サーバにバインドします。

2. ECDHE 暗号を仮想サーバにバインドします。

   a. [構成**] > [**トラフィック管理**] > [**仮想サーバー] に移動し、DH を有効にする仮想サーバーを選択します。

   b. [編集**] > [SSL** 暗号] をクリックし、ECDHE 暗号グループを選択して [バインド] をクリックします。

   注**:** ECDHE 暗号が、仮想サーバーにバインドされた暗号リストの一番上にあることを確認してください。

注：それぞれのケースで、NetScaler ADC アプライアンスが通信に使用する暗号をサポートしていることを確認します。

SSL プロファイルを使用した **PFS** の設定

注**:** SSL プロファイルを使用して PFS（暗号または ECC）を設定するオプションは、11.0 64.x リリース以降から導入されます。古いバージョンの場合は、次のセクションを無視してください。

SSL プロファイルを使用して PFS を有効にするには、仮想サーバ上で直接設定するのではなく、SSL プロファイルで同様の設定（前の設定セクションで説明した）を実行する必要があります。

GUI を使用して **SSL** プロファイルを使用した **PFS** の設定

1. SSL プロファイルで ECC カーブと ECDHE 暗号をバインドします。

   注**:** ECC カーブは、デフォルトですべての SSL プロファイルに既にバインドされています。

   a. [システム**] > [**プロファイル**] > [SSL** プロファイル] に移動し、PFS を有効にするプロファイルを選択します。

   b. ECDHE 暗号をバインドします。

2. SSL プロファイルを仮想サーバーにバインドします。

   a. [構成**] > [**トラフィック管理**] > [**仮想サーバー] に移動し、仮想サーバーを選択します。

   b. 鉛筆アイコンをクリックして SSL プロファイルを編集します。

   c. 「**OK**」をクリックし、「完了」をクリックします。

CLI を使用した **SSL** を使用した **PFS** の設定　コマンドプロンプトで入力します：

1. ECC カーブを SSL プロファイルにバインドします。

```
bind sslprofile <SSLProfileName> -eccCurveName <Name_of_curve>
<!--NeedCopy-->
```

2. ECDHE 暗号グループをバインドします。

```
bind sslprofile <SSLProfileName> cipherName <ciphergroupName>
<!--NeedCopy-->
```


3. ECDHE 暗号のプライオリティを 1 に設定します。

```
set sslprofile <SSLProfileName> cipherName <ciphergroupName>
   cipherPriority <positive_integer>
<!--NeedCopy-->
```

4. SSL プロファイルを仮想サーバーにバインドします。

```
set SSL vserver <vservername> sslProfile <SSLProfileName>
<!--NeedCopy-->
```

ECDHE 暗号

January 11, 2024

すべての NetScaler アプライアンスは、フロントエンドとバックエンドで ECDHE 暗号グループをサポートしています。SDX アプライアンスでは、SSL チップが VPX インスタンスに割り当てられている場合、MPX アプライアンスの暗号サポートが適用されます。それ以外の場合は、VPX インスタンスの通常の暗号サポートが適用されます。

これらの暗号をサポートするビルドとプラットフォームの詳細については、NetScaler アプライアンスで使用可能な暗号を参照してください。

ECDHE 暗号スイートは、楕円曲線暗号（ECC）を使用します。キーサイズが小さいため、ECC は、ミリ秒単位が重要なモバイル（ワイヤレス）環境やインタラクティブな音声応答環境で特に役立ちます。キーサイズを小さくすると、電力、メモリ、帯域幅、および計算コストを節約できます。

NetScaler アプライアンスは以下の ECC カーブをサポートしています。

- P_256
- P_384
- P_224
- P_521

注: リリース 10.1 ビルド 121.10 より前のビルドからアップグレードする場合は、ECC カーブを既存の SSL 仮想サーバーとサービスに明示的にバインドする必要があります。カーブはデフォルトで、アップグレード後に作成したすべての仮想サーバーとサービスにバインドされます。

ECC カーブを SSL フロントエンドとバックエンドのエンティティにバインドできます。デフォルトでは、4 つのカーブはすべて P_256、P_384、P_224、P_521 の順序でバインドされます。順序を変更するには、最初にすべてのカーブをバインド解除してから、目的の順序でバインドする必要があります。

CLI を使用して ECC カーブを SSL 仮想サーバーにバインドする

コマンドプロンプトで入力します:

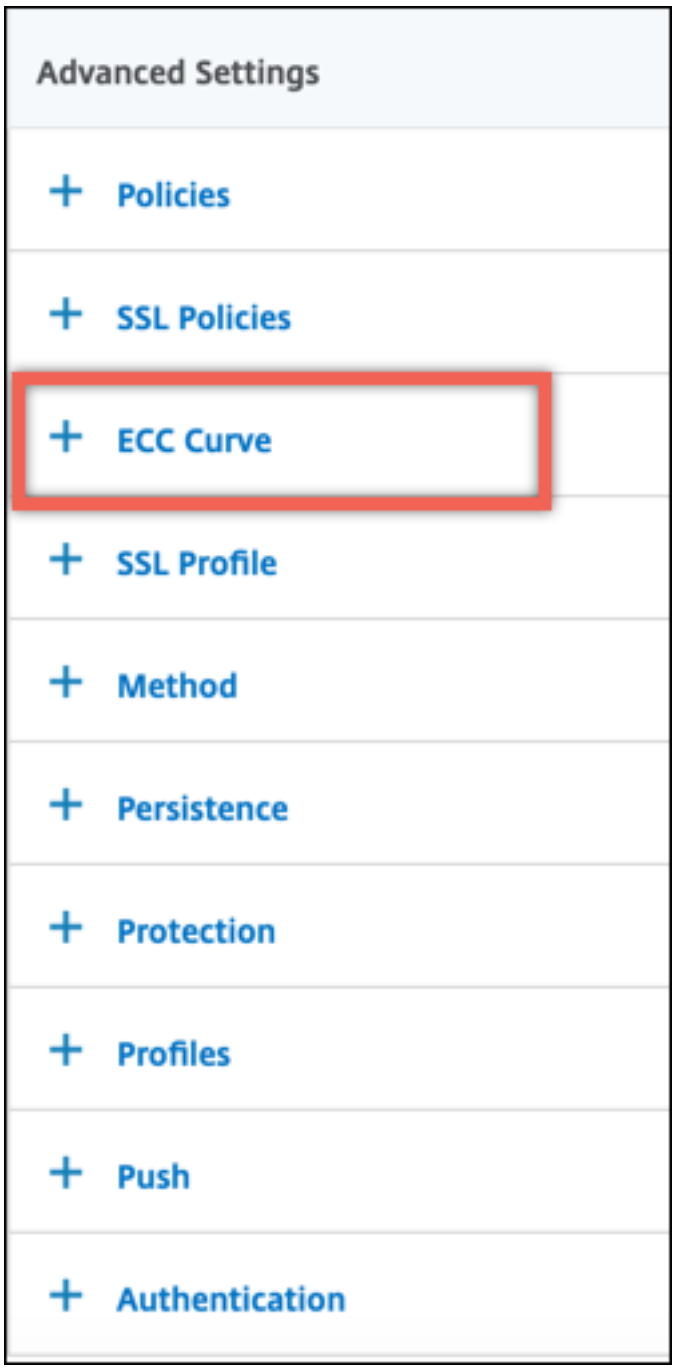

```
bind ssl vserver <vServerName > -eccCurveName <eccCurveName >
```

例:

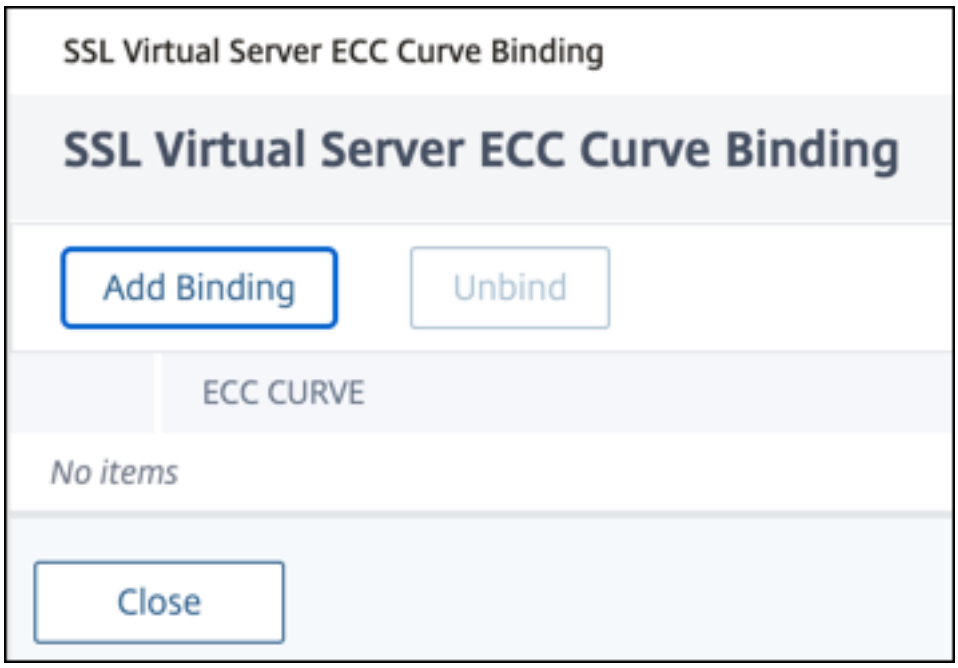
```
bind ssl vserver  v1 -eccCurveName P_224

sh ssl vserver v1

Advanced SSL configuration for VServer v1:
DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED
   TLSv1.2: DISABLED
Push Encryption Trigger: Always
Send Close-Notify: YES
ECC Curve: P_224

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用して **ECC** カーブを **SSL** 仮想サーバーにバインドする

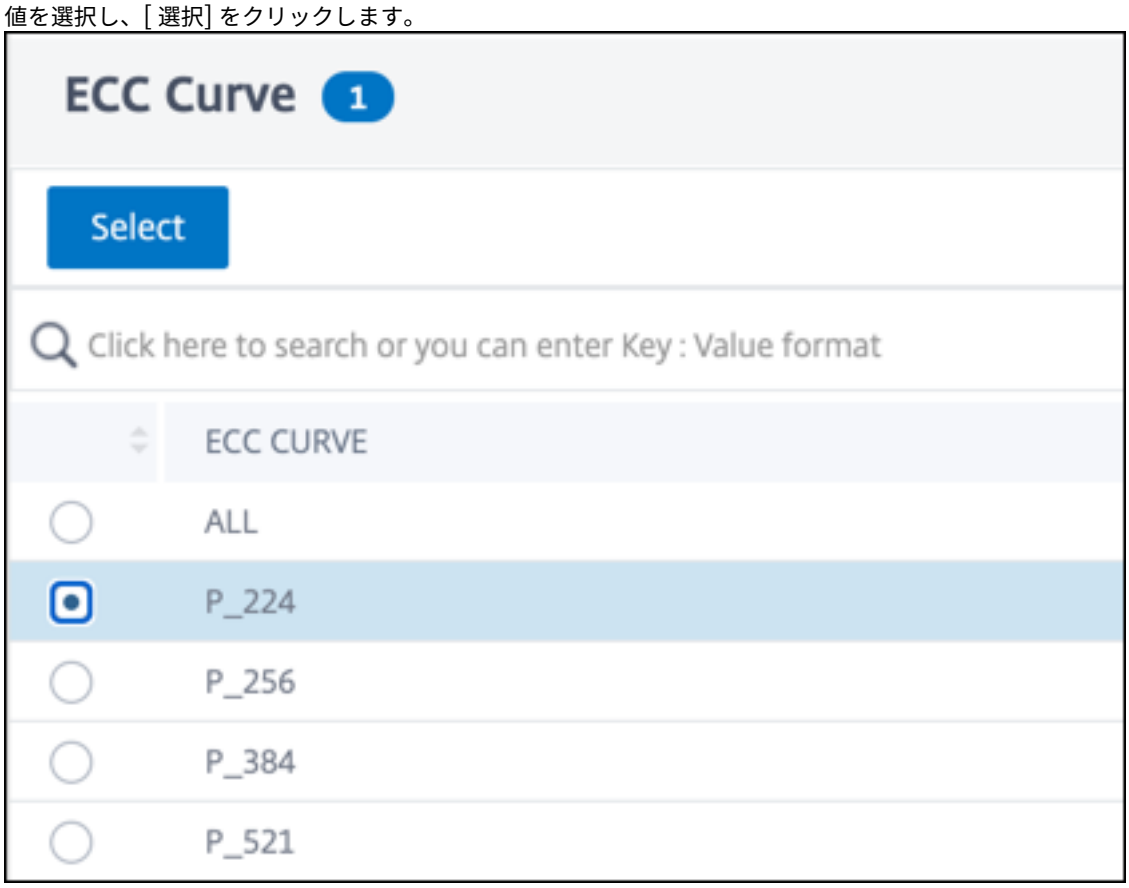
1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. SSL 仮想サーバーを選択し、[編集] をクリックします。
3. [詳細設定] で [**ECC** カーブ] をクリックします。



4. ECC カーブセクションの内側をクリックします。
5. **SSL** 仮想サーバー **ECC** カーブバインディングページで、バインディングの追加をクリックします **。



6. 「**ECC** カーブバインディング」で、「**ECC** カーブを選択」をクリックします。
7. 値を選択し、[選択] をクリックします。



8. [**Bind**] をクリックします。
9. [閉じる] をクリックします。

10. ［完了］をクリックします。

CLI を使用して ECC カーブを SSL サービスにバインドする

コマンドプロンプトで入力します:

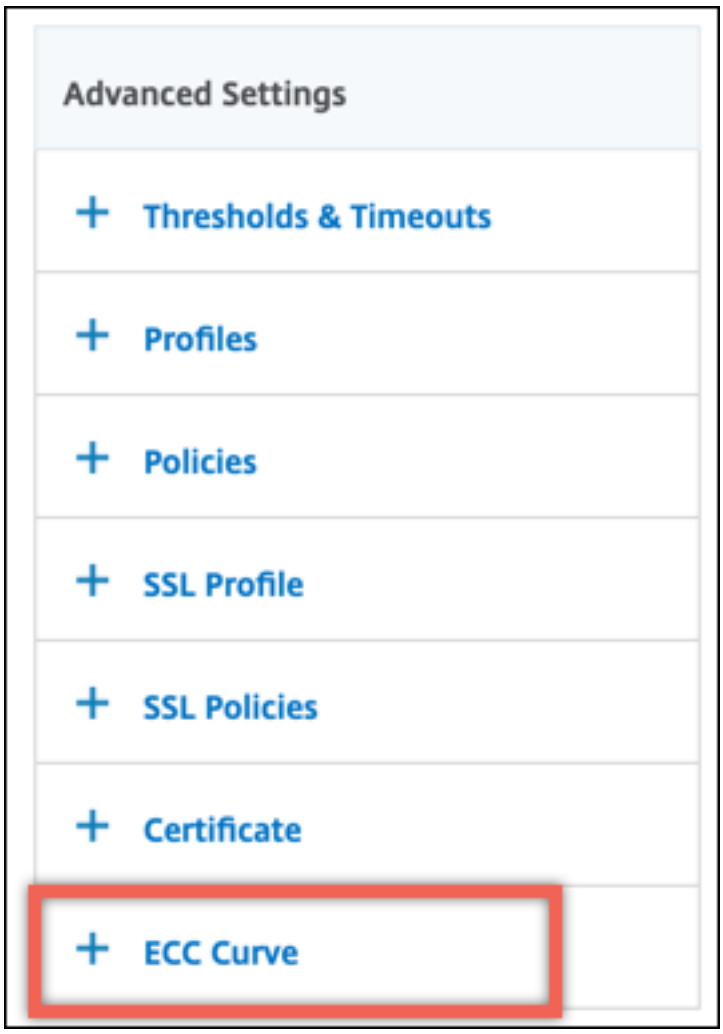
```
bind ssl service <vServerName > -eccCurveName <eccCurveName >
```

例:

```
> bind ssl service sslsvc -eccCurveName P_224
 Done
> sh ssl service sslsvc

    Advanced SSL configuration for Back-end SSL Service sslsvc:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       DISABLED
    Session Reuse: ENABLED      Timeout: 300 seconds
    Cipher Redirect: DISABLED
    ClearText Port: 0
    Server Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: DISABLED
    OCSP Stapling: DISABLED
    SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: ???
    DHE Key Exchange With PSK: ???
    Tickets Per Authentication Context: ???

    ECC Curve: P_224


1)  Cipher Name: DEFAULT_BACKEND
    Description: Default cipher list for Backend SSL session
 Done
<!--NeedCopy-->
```

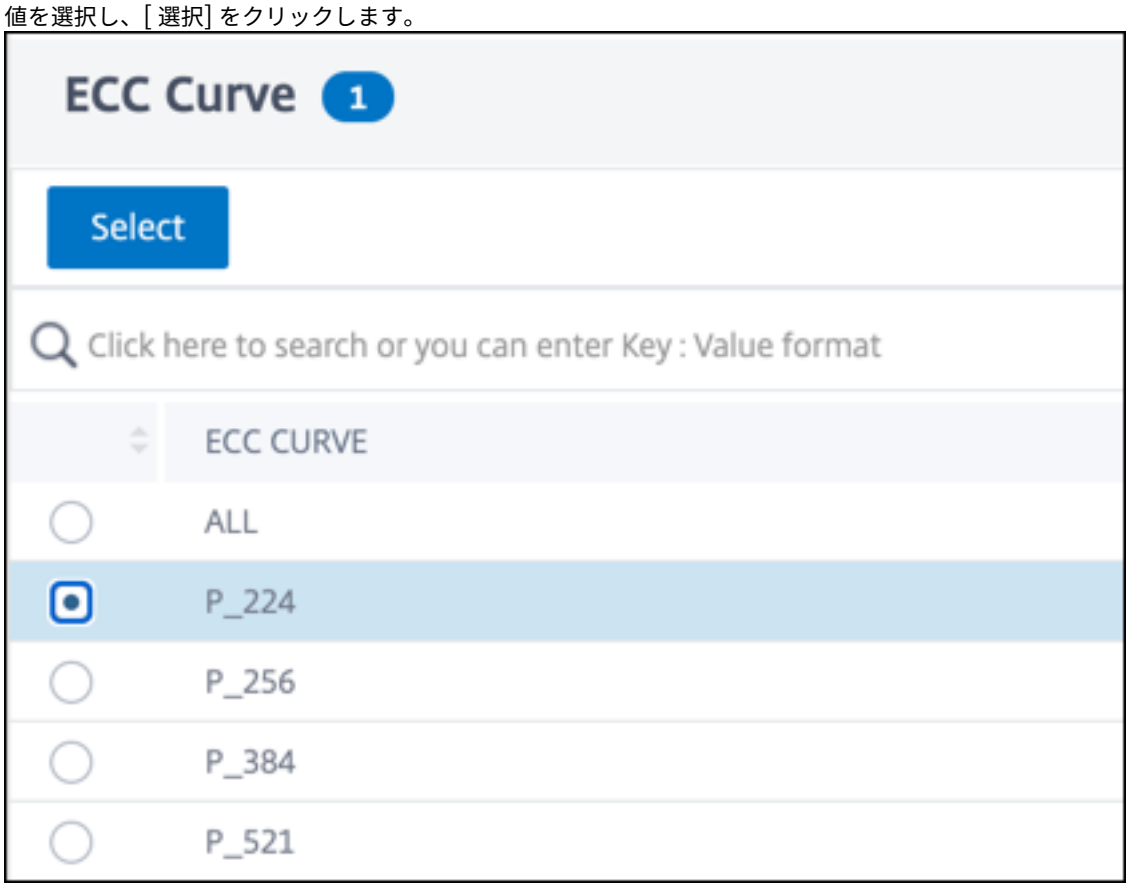
GUI を使用して ECC カーブを SSL サービスにバインドする

1. ［**Traffic Management**］ **>** ［**Load Balancing**］ **>** ［**Services**］の順に移動します。
2. SSL サービスを選択し、[編集] をクリックします。
3. [詳細設定] で [**ECC** カーブ] をクリックします。



4. ECC カーブセクションの内側をクリックします。
5. **SSL** サービス **ECC** カーブバインディングページで、バインディングの追加をクリックします **。

6.「**ECC** カーブバインディング」で、「**ECC** カーブを選択」をクリックします。
7. 値を選択し、[選択] をクリックします。



8.［**Bind**］をクリックします。

9. ［閉じる］をクリックします。
10. ［完了］をクリックします。

Diffie-Hellman のパラメータ生成と DHE による PFS の実現

August 15, 2023

Diffie-Hellman (DH) キー交換は、SSL トランザクションに関与する 2 つの当事者が、安全でないチャネルを介して共有シークレットについて合意する方法です。これらの当事者は、お互いについて事前に知ることはありません。このシークレットは、このような鍵交換を必要とする対称鍵暗号アルゴリズム用の暗号鍵資料に変換できます。

この機能はデフォルトでは無効になっています。キー交換アルゴリズムとして DH を使用する暗号をサポートするように機能を設定しました。

> 注記:
>
> 2048 ビットの DH パラメータの生成には、長い時間 (最大 30 分) かかる場合があります。

CLI を使用して DH パラメータを生成

コマンドプロンプトで、次のコマンドを入力します。

```
create ssl dhparam <dhFile> [<bits>] [-gen (2 | 5)]
<!--NeedCopy-->
```

例:

```
create ssl dhparam Key-DH-1 512 -gen 2
<!--NeedCopy-->
```

GUI を使用して DH パラメータを生成

[** トラフィック管理] > [**SSL**] に移動し、[ツール] グループで [**Diffie-Hellman (DH)** キーの作成] を選択し、**[SSL DH** パラメータの設定] を選択します。*****

> 注:
>
> DH パラメーターの詳細については、Diffie-Hellman パラメーターを参照してください。

DHE で完全な前方秘密を達成する

DH パラメータの生成は CPU を大量に消費する操作です。以前のリリースでは、VPX アプライアンスでのパラメータ生成はソフトウェアで行われていたため、長い時間がかかりました。パラメータ生成は、dhKeyExpSizeLimit

パラメータを設定することによって最適化されます。このパラメータを SSL 仮想サーバーまたは SSL プロファイルに設定し、プロファイルを仮想サーバーにバインドできます。

DH カウントをゼロに設定することで、NetScaler MPX アプライアンスで Perfect Forward Secrecy（PFS）を維持できます。その結果、NetScaler MPX アプライアンスのトランザクションごとに DH パラメーターが生成されます（最小値DHcountは 0）。これらのパラメータは、操作が最適化されるため、パフォーマンスを大幅に低下させることなく生成されます。以前は、許容される最小 DH カウントは 500 でした。つまり、最大 500 件のトランザクションでキーを再生成することはできません。

> 制限事項:
>
> NetScaler VPX アプライアンスでは、DH 数をゼロに設定した場合、DH パラメーターは再生成されません。そのため、PFS を維持するには DH 数を 500 に設定する必要があります。DH パラメータは 500 回のトランザクション後に再生成されます。

CLI を使用して **DH** パラメータ生成を最適化する

コマンドプロンプトで、コマンド 1 と 2 を入力するか、コマンド 3 を入力します。

```
1.  add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )] [-dhCount <positive_integer>] [-dh ( ENABLED | DISABLED) -dhFile <string>] [-dhKeyExpSizeLimit ( ENABLED | DISABLED)]
2.  set ssl vserver <vServerName> [-sslProfile <string>]
<!--NeedCopy-->
```

```
3.  set ssl vserver <vServerName> [-dh ( ENABLED | DISABLED) -dhFile <string>] [-dhCount <positive_integer>] [-dhKeyExpSizeLimit ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

GUI を使用して **DH** パラメータ生成を最適化

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2.「**SSL** パラメータ」セクションで、「**DH** キーの有効期限サイズ制限を有効にする」を選択します。

暗号リダイレクト

August 15, 2023

SSL ハンドシェイク中、SSL クライアント（通常は Web ブラウザ）は、サポートしている一連の暗号を、設定された暗号プリファレンスの順序でアナウンスします。SSL サーバは、そのリストから、設定された暗号の独自のリストと一致する暗号を選択します。

クライアントがアナウンスした暗号が SSL サーバーに設定されている暗号と一致しない場合、SSL ハンドシェイクは失敗します。この失敗は、ブラウザにわかりにくいエラーメッセージが表示されることで通知されます。これらのメッセージには、エラーの正確な原因が記載されることはほとんどありません。

暗号リダイレクトを使用すると、SSL ハンドシェイクが失敗したときに正確で意味のあるエラーメッセージを配信するように SSL 仮想サーバーを構成できます。SSL ハンドシェイクが失敗すると、ADC アプライアンスはユーザーを以前に設定された URL にリダイレクトするか、URL が設定されていない場合は、内部で生成されたエラーページを表示します。

CLI を使用して暗号リダイレクションを設定する

コマンドプロンプトで次のコマンドを入力して暗号リダイレクションを構成し、構成を確認します。

```
1 -  set ssl vserver <vServerName> -cipherRedirect < ENABLED | DISABLED>
     -cipherURL < URL>
2 -  show ssl vserver <vServerName>
3 <!--NeedCopy-->
```

例:

```
1  set ssl vserver vs-ssl -cipherRedirect ENABLED -cipherURL http://
       redirectURl
2
3  Done
4
5  show ssl vserver vs-ssl
6
7  Advanced SSL configuration for VServer vs-ssl:
8  DH: DISABLED
9  Ephemeral RSA: ENABLED          Refresh Count: 1000
10 Session Reuse: ENABLED          Timeout: 600 seconds
11 Cipher Redirect: ENABLED        Redirect URL: http://redirectURl
12 SSLv2 Redirect: DISABLED
13 ClearText Port: 0
14 Client Auth: DISABLED
15 SSL Redirect: DISABLED
16 Non FIPS Ciphers: DISABLED
17 SNI: DISABLED
18 OCSP Stapling: DISABLED
19 HSTS: DISABLED
20 HSTS IncludeSubDomains: NO
21 HSTS Max-Age: 0
22 SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2: ENABLED
       TLSv1.2: ENABLED
23     1)      CertKey Name: Auth-Cert-1       Server Certificate
24     1)      Cipher Name: DEFAULT
25             Description: Predefined Cipher Alias
26 Done
27 <!--NeedCopy-->
```


GUI を使用して暗号リダイレクトを設定する

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2.「**SSL** パラメータ」セクションで、「暗号リダイレクトを有効にする」を選択し、リダイレクト URL を指定します。

ハードウェアとソフトウェアを使用して **ECDHE** と **ECDSA** 暗号のパフォーマンスを向上させましょう

August 15, 2023

> 注:
>
> この拡張機能は次のプラットフォームにのみ適用されます。
>
> - MPX/SDX 11000
> - MPX/SDX 14000
> - MPX 22000、MPX 24000、MPX 25000
> - MPX/SDX 14000 FIPS

以前は、NetScaler アプライアンスでの ECDHE と ECDSA の計算はハードウェア（Cavium チップ）上でのみ実行されていたため、いつでも SSL セッションの数が制限されていました。この拡張機能により、一部の操作もソフトウェアで実行されるようになりました。つまり、Cavium チップと CPU コアの両方で処理が行われ、ECDHE と ECDSA の暗号性能が向上します。

処理はまず、設定されたソフトウェア暗号しきい値までソフトウェアで実行されます。このしきい値に達すると、操作はハードウェアにオフロードされます。そのため、このハイブリッドモデルでは、ハードウェアとソフトウェアの両方を使用して SSL のパフォーマンスを向上させます。要件に合わせて「SoftwareCryptoThreshold」パラメーターを設定することで、ハイブリッドモデルを有効にできます。ハイブリッドモデルを無効にするには、このパラメーターを 0 に設定します。

CPU のしきい値は ECDHE と ECDSA の計算に限定されないため、現在の CPU 使用率がそれほど高くない場合に最大のメリットが得られます。たとえば、アプライアンスの現在のワークロードが CPU サイクルの 50% を消費し、しきい値が 80% に設定されている場合、ECDHE と ECDSA の計算では 30% しか使用できません。設定したソフトウェア暗号化しきい値の 80% に達すると、ECDHE と ECDSA の計算がさらにハードウェアにオフロードされます。その場合、ハードウェアで ECDHE と ECDSA の計算を実行すると一部の CPU サイクルが消費されるため、実際の CPU 使用率は 80% を超える可能性があります。

CLI を使用してハイブリッドモデルを有効にする

コマンドプロンプトで入力します。


```
set ssl parameter -softwareCryptoThreshold <positive_integer>

Synopsis:

softwareCryptoThreshold:

NetScaler CPU utilization threshold (as a percentage) beyond which
   crypto operations are not done in software. A value of zero implies
   that CPU is not utilized for doing crypto in software.

Default = 0

Min = 0

Max = 100
<!--NeedCopy-->
```

例:

```
set ssl parameter - softwareCryptoThreshold 80
Done

show ssl parameter
Advanced SSL Parameters

SSL quantum size                  : 8 KB
Max CRL memory size               : 256 MB
Strict CA checks                  : NO
Encryption trigger timeout        : 100 ms
Send Close-Notify                 : YES
Encryption trigger packet c       : 45
Deny SSL Renegotiation            : ALL
Subject/Issuer Name Insertion Format : Unicode
OCSP cache size                   : 10 MB
Push flag                         : 0x0 (Auto)
Strict Host Header check for SNI enabled SSL sessions : NO
PUSH encryption trigger timeout   : 1 ms
Crypto Device Disable Limit       : 0
Global undef action for control policies : CLIENTAUTH
Global undef action for data policies : NOOP
Default profile                   : DISABLED
Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors    : NO
Disable TLS 1.1/1.2 for dynamic and VPN services : NO
Software Crypto acceleration CPU Threshold : 80
Signature and Hash Algorithms supported by TLS1.2 : ALL
<!--NeedCopy-->
```

GUI を使用してハイブリッドモデルを有効にする

1. [トラフィック管理] > [**SSL**] > **[SSL** の詳細設定の変更] に移動します。

2.「ソフトウェア暗号しきい値 **(%)**」に値を入力します。

ECDHE 為替レートの **SNMP** アラームを設定します

ECDHE ベースのキー交換により、アプライアンスの 1 秒あたりのトランザクション数が減少する可能性があります。リリース 13.0 ビルド 52.x から、ECDHE ベースのトランザクションに SNMP アラームを設定できます。このアラームでは、ECDHE 為替レートのしきい値と通常の制限を設定できます。`nsssl_tot_sslInfo_ECDHE_Tx` 新しいカウンターが追加されます。このカウンタは、アプライアンスのフロントエンドとバックエンドにあるすべての ECDHE ベースのトランザクションカウンタの合計です。ECDHE ベースのキー交換が設定された制限を超えると、SNMP トラップが送信されます。値が設定された標準値に戻ると、別のトラップが送信されます。

CLI を使用して **ECDHE** 為替レートの **SNMP** アラームを設定します

コマンドプロンプトで入力します。

```
set snmp alarm ECDHE-EXCHANGE-RATE -logging ( ENABLED | DISABLED ) -
    severity <severity>
 -state ( ENABLED | DISABLED ) -thresholdValue <positive_integer>  [-
    normalValue <positive_integer>] -time <secs>
<!--NeedCopy-->
```

例**:**

```
set snmp alarm ECDHE-EXCHANGE-RATE -logging eNABLED -severity critical
    -state eNABLED -thresholdValue 100 -normalValue 50
<!--NeedCopy-->
```

ECDSA 暗号の組み合わせのサポート

August 15, 2023

ECDSA 暗号スイートは楕円曲線暗号 (ECC) を使用します。サイズが小さいため、処理能力、ストレージ領域、帯域幅、消費電力に制約がある環境で役立ちます。

ECDHE_ECDSA 暗号グループを使用する場合、サーバーの証明書に ECDSA 対応の公開鍵が含まれている必要があります。

次の表は、N3 チップ、NetScaler VPX アプライアンス、MPX 5900/26000、MPX/SDX 8900/15000 アプライアンスを搭載したアプライアンスでサポートされる ECDSA 暗号の一覧です。

暗号名	優先度	説明	鍵交換アルゴリズム	認証アルゴリズム	暗号化アルゴリズム (キーサイズ)	メッセージ認証コード (MAC) アルゴリズム	16 進コード
TLS1-ECDHE-ECDSA-AES128-SHA	1	SSLv3	ECC-DHE	ECDSA	AES (128)	SHA1	0xc009
TLS1-ECDHE-ECDSA-AES256-SHA	2	SSLv3	ECC-DHE	ECDSA	AES (256)	SHA1	0xc00a
TLS1.2-ECDHE-ECDSA-AES128-SHA256	3	TLSv1.2	ECC-DHE	ECDSA	AES (128)	SHA-256	023
TLS1.2-ECDHE-ECDSA-AES256-SHA384	4	TLSv1.2	ECC-DHE	ECDSA	AES (256)	SHA-384	024
TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256	5	TLSv1.2	ECC-DHE	ECDSA	AES-GCM(128)	SHA-256	0xc
TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384	6	TLSv1.2	ECC-DHE	ECDSA	AES-GCM(256)	SHA-384	0xc
TLS1-ECDHE-ECDSA-RC4-SHA	7	SSLv3	ECC-DHE	ECDSA	RC4(128)	SHA1	0xc007

暗号名	優先度	説明	鍵交換アルゴリズム	認証アルゴリズム	暗号化アルゴリズム (キーサイズ)	メッセージ認証コード (MAC) アルゴリズム	16 進コード
TLS1-ECDHE-ECDSA-DES-CBC3-SHA	8	SSLv3	ECC-DHE	ECDSA	3DES(168)	SHA1	0xc008
TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305	9	TLSv1.2	ECC-DHE	ECDSA	CHACHA20/POLY1305(256)	AEAD	0xcca9

ECDSA/RSA 暗号と証明書の選択

ECDSA と RSA のサーバー証明書の両方を SSL 仮想サーバーに同時にバインドできます。ECDSA 証明書と RSA 証明書の両方が仮想サーバーにバインドされると、クライアントに提示する適切なサーバー証明書が自動的に選択されます。クライアント暗号リストに RSA 暗号が含まれているが ECDSA 暗号が含まれていない場合、仮想サーバーは RSA サーバー証明書を提示します。クライアントのリストに両方の暗号が存在する場合、表示されるサーバー証明書は仮想サーバーに設定されている暗号優先度によって異なります。つまり、RSA の優先順位が高い場合は、RSA 証明書が表示されます。ECDSA の優先順位が高い場合は、ECDSA 証明書がクライアントに提示されます。

ECDSA または **RSA** 証明書を使用したクライアント認証

クライアント認証では、仮想サーバーにバインドされた CA 証明書に ECDSA または RSA 署名を付けることができます。アプライアンスは混合証明書チェーンをサポートします。たとえば、次の証明書チェーンがサポートされています。

クライアント証明書（ECDSA）<-> CA 証明書（RSA）<-> 中間証明書（RSA）<-> ルート証明書（RSA）

次の表に、ECDSA 暗号グループおよび ECDSA 証明書を使用するさまざまな NetScaler ADC アプライアンスでサポートされている楕円曲線を示します。

楕円曲線	サポートされているプラットフォーム
prime256v1	FIPS を含むすべてのプラットフォーム。

楕円曲線	サポートされているプラットフォーム
secp384r1	FIPS を含むすべてのプラットフォーム。
secp521r1	MPX 5900、MPX/SDX 8900、MPX/SDX 15000、MPX/SDX 26000、VPX
secp224r1	MPX 5900、MPX/SDX 8900。MPX/SDX 15000、MPX/SDX 26000、VPX

ECDSA 証明書とキーペアの作成

CLI または GUI を使用して、ECDSA 証明書とキーのペアを NetScaler アプライアンス上で直接作成できます。以前は、アプライアンスに ECC 証明書とキーのペアをインストールしてバインドすることはできましたが、証明書とキーのペアを作成するには OpenSSL を使用する必要がありました。

P_256 カーブと P_384 カーブのみがサポートされています。

> 注
>
> このサポートは、MPX 9700/1050/12500/15500 を除くすべてのプラットフォームで利用できます。

CLI を使用して **ECDSA** 証明書とキーのペアを作成するには:

コマンドプロンプトで入力します。

```
create ssl ecdsaKey <keyFile> -curve ( P_256 | P_384 ) [-keyform ( DER | PEM )] [-des | -des3] {
 -password  }
  [-pkcs8]
<!--NeedCopy-->
```

例:

```
create ecdsaKey ec_p256.ky -curve P_256 -pkcs8
Done
create ecdsaKey ec_p384.ky -curve P_384
Done
<!--NeedCopy-->
```

GUI を使用して **ECDSA** 証明書とキーのペアを作成するには:

1. [トラフィック管理] **>** **[SSL]** **>** **[SSL** ファイル] **>** **[**キー] に移動し、[**ECDSA** キーの作成] をクリックします。
2. **PKCS #8** 形式でキーを作成するには、**PKCS8** を選択します。

ADC アプライアンスでユーザー定義の暗号グループを構成する

August 15, 2023

暗号グループは、NetScaler アプライアンス上の SSL 仮想サーバー、サービス、またはサービスグループにバインドする暗号スイートのセットです。暗号スイートは、プロトコル、キー交換 (Kx) アルゴリズム、認証 () アルゴリズム、暗号化 (AuEnc) アルゴリズム、およびメッセージ認証コード (Mac) アルゴリズムで構成されています。アプライアンスには、事前に定義された暗号グループのセットが付属しています。SSL サービスまたは SSL サービスグループを作成すると、ALL 暗号グループが自動的にそれにバインドされます。ただし、SSL 仮想サーバーまたはトランスペアレント SSL サービスを作成すると、DEFAULT 暗号グループが自動的にそれにバインドされます。さらに、ユーザー定義の暗号グループを作成して、SSL 仮想サーバー、サービス、またはサービスグループにバインドできます。

注**:** MPX アプライアンスにライセンスがない場合は、EXPORT 暗号のみが SSL 仮想サーバー、サービス、またはサービスグループにバインドされます。

ユーザー定義の暗号グループを作成するには、まず暗号グループを作成し、次に暗号または暗号グループをこのグループにバインドします。暗号エイリアスまたは暗号グループを指定すると、その暗号エイリアスまたはグループ内のすべての暗号がユーザー定義の暗号グループに追加されます。個々の暗号 (暗号スイート) をユーザー定義のグループに追加することもできます。ただし、定義済みの暗号グループを変更することはできません。暗号グループを削除する前に、グループ内のすべての暗号スイートをバインド解除します。

暗号グループを SSL 仮想サーバー、サービス、またはサービスグループにバインドすると、エンティティにバインドされている既存の暗号に暗号が追加されます。特定の暗号グループをエンティティにバインドするには、まずエンティティにバインドされている暗号または暗号グループのバインドを解除する必要があります。次に、特定の暗号グループをエンティティにバインドします。たとえば、AES 暗号グループのみを SSL サービスにバインドするには、次の手順を実行します。

1. サービスの作成時にデフォルトでサービスにバインドされているデフォルトの暗号グループ ALL をバインド解除します。

   ```
   unbind ssl service <service name> -cipherName ALL
   <!--NeedCopy-->
   ```

2. AES 暗号グループをサービスにバインドする

   ```
   bind ssl service <Service name> -cipherName AE
   <!--NeedCopy-->
   ```

   AES に加えて暗号グループ DES をバインドする場合は、コマンドプロンプトで次のように入力します。

   ```
   bind ssl service <service name> -cipherName DES
   <!--NeedCopy-->
   ```

注：無料の NetScaler 仮想アプライアンスは、DH 暗号グループのみをサポートします。

CLI を使用してユーザー定義の暗号グループを設定します

コマンドプロンプトで次のコマンドを入力して、暗号グループを追加するか、以前に作成したグループに暗号を追加し、設定を確認します。

```
add ssl cipher <cipherGroupName>
bind ssl cipher <cipherGroupName> -cipherName <cipherGroup/cipherName>
show ssl cipher <cipherGroupName>
<!--NeedCopy-->
```

例:

```
add ssl cipher test

Done

bind ssl cipher test -cipherName ECDHE

Done

sh ssl cipher test

1)      Cipher Name: TLS1-ECDHE-RSA-AES256-SHA  Priority : 1
Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1   HexCode
    =0xc014
2)      Cipher Name: TLS1-ECDHE-RSA-AES128-SHA  Priority : 2
Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1   HexCode
    =0xc013
3)      Cipher Name: TLS1.2-ECDHE-RSA-AES-256-SHA384    Priority : 3
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA-384
    HexCode=0xc028
4)      Cipher Name: TLS1.2-ECDHE-RSA-AES-128-SHA256    Priority : 4
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA-256
    HexCode=0xc027
5)      Cipher Name: TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 Priority : 5
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(256) Mac=AEAD
    HexCode=0xc030
6)      Cipher Name: TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 Priority : 6
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(128) Mac=AEAD
    HexCode=0xc02f
7)      Cipher Name: TLS1-ECDHE-ECDSA-AES256-SHA        Priority : 7
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA1
    HexCode=0xc00a
8)      Cipher Name: TLS1-ECDHE-ECDSA-AES128-SHA        Priority : 8
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA1
    HexCode=0xc009
9)      Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-SHA384   Priority : 9
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA-384
    HexCode=0xc024
10)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-SHA256   Priority : 10
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA-256
    HexCode=0xc023
11)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
```



```
      Priority : 11
32 Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(256) Mac=AEAD
      HexCode=0xc02c
33 12)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
      Priority : 12
34 Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(128) Mac=AEAD
      HexCode=0xc02b
35 13)     Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA        Priority : 13
36 Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1   HexCode
      =0xc012
37 14)     Cipher Name: TLS1-ECDHE-ECDSA-DES-CBC3-SHA      Priority : 14
38 Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=3DES(168) Mac=SHA1
      HexCode=0xc008
39 15)     Cipher Name: TLS1-ECDHE-RSA-RC4-SHA     Priority : 15
40 Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=RC4(128)  Mac=SHA1   HexCode
      =0xc011
41 16)     Cipher Name: TLS1-ECDHE-ECDSA-RC4-SHA   Priority : 16
42 Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=RC4(128)  Mac=SHA1
      HexCode=0xc007
43 17)     Cipher Name: TLS1.2-ECDHE-RSA-CHACHA20-POLY1305 Priority : 17
44 Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=CHACHA20/POLY1305(256) Mac
      =AEAD   HexCode=0xcca8
45 18)     Cipher Name: TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305
      Priority : 18
46 Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=CHACHA20/POLY1305(256)
      Mac=AEAD   HexCode=0xcca9
47 Done
48
49 bind ssl cipher test -cipherName TLS1-ECDHE-RSA-DES-CBC3-SHA
50 <!--NeedCopy-->
```

CLI を使用して暗号グループから暗号をバインド解除する

コマンドプロンプトで次のコマンドを入力して、ユーザー定義の暗号グループから暗号をバインド解除し、設定を確認します。

```
1 show ssl cipher <cipherGroupName>
2
3 unbind ssl cipher <cipherGroupName> -cipherName <string>
4
5 show ssl cipher <cipherGroupName>
6 <!--NeedCopy-->
```

CLI を使用して暗号グループを削除する

注: 組み込みの暗号グループは削除できません。ユーザー定義の暗号グループを削除する前に、暗号グループが空であることを確認してください。

コマンドプロンプトで次のコマンドを入力してユーザー定義の暗号グループを削除し、構成を確認します。


```
rm ssl cipher <userDefCipherGroupName> [<cipherName> ...]
show ssl cipher <cipherGroupName>

<!--NeedCopy-->
```

例:

```
rm ssl cipher test Done

sh ssl cipher test ERROR: No such resource [cipherGroupName, test]
<!--NeedCopy-->
```

GUI を使用してユーザー定義の暗号グループを設定します

1. [トラフィック管理] > [SSL] > [暗号グループ] に移動します。
2. [追加] をクリックします。
3. 暗号グループの名前を指定します。
4. [追加] をクリックすると、使用可能な暗号と暗号グループが表示されます。
5. 暗号または暗号グループを選択し、矢印ボタンをクリックして追加します。
6. [作成] をクリックします。
7. [閉じる] をクリックします。

CLI を使用して暗号グループを **SSL** 仮想サーバー、サービス、またはサービスグループにバインドするには:

コマンドプロンプトで、次のいずれかを入力します。

```
bind ssl vserver <vServerName> -cipherName <string>

bind ssl service <serviceName> -cipherName <string>

bind ssl serviceGroup <serviceGroupName> -cipherName <string>

<!--NeedCopy-->
```

例:

```
bind ssl vserver ssl_vserver_test -cipherName test
Done

bind ssl service  nshttps -cipherName test
Done

bind ssl servicegroup  ssl_svc  -cipherName test
Done
<!--NeedCopy-->
```

GUI を使用して暗号グループを **SSL** 仮想サーバー、サービス、またはサービスグループにバインドするには:

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。

   サービスについては、仮想サーバーをサービスに置き換えます。サービスグループの場合は、仮想サーバーをサービスグループに置き換えます。

   仮想サーバー、サービス、またはサービスグループを開きます。

2.「詳細設定」で、「**SSL** 暗号」を選択します。

3. 暗号グループを仮想サーバー、サービス、またはサービスグループにバインドします。

個々の暗号を **SSL** 仮想サーバーまたはサービスにバインドする

暗号グループの代わりに個々の暗号を仮想サーバーまたはサービスにバインドすることもできます。

CLI を使用して暗号をバインドするには、コマンドプロンプトで次のように入力します。

```
bind ssl vserver <vServerName> -cipherName <string>
bind ssl service <serviceName> -cipherName <string>
<!--NeedCopy-->
```

例**:**

```
bind ssl vserver v1 -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
Done

bind ssl service sslsvc -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
Done
<!--NeedCopy-->
```

GUI を使用して暗号を **SSL** 仮想サーバーにバインドするには**:**

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. SSL 仮想サーバーを選択し、[編集] をクリックします。
3.「詳細設定」で、「**SSL** 暗号」を選択します。
4.「暗号スイート」で、「追加」を選択します。
5. 使用可能なリストで暗号を検索し、矢印をクリックして設定済みリストに追加します。
6. [**OK**] をクリックします。
7. [完了] をクリックします。

暗号を SSL サービスにバインドするには、仮想サーバをサービスに置き換えた後、上記の手順を繰り返します。

ADC アプライアンスのサーバー証明書サポートマトリックス

August 15, 2023

リリース 13.0 ビルド 41.x から、ADC アプライアンスは、合計サイズが 32 KB 以内の場合に複数のレコードにフラグメント化されるサーバ証明書メッセージをサポートします。以前は、サポートされる最大サイズは 16 KB で、フラグメンテーションはサポートされていませんでした。

NetScaler ADC アプライアンスは、次のサーバー証明書をサポートしています。

表 1: フロントエンド (FE) サービスとバックエンド (BE) サービスのサポート

サーバー証明書/プラットフォーム	MPX/SDX (N2 CHIPS) FE	MPX/SDX (N2 CHIPS) BE	MPX/SDX (N3 CHIPS) FE	MPX/SDX (N3 CHIPS) BE	VPX FE	VPX BE
MD5	Y	Y	Y	Y	Y	Y
SHA1	Y	Y	Y	Y	Y	Y
SHA224	Y	Y	Y	Y	Y	Y
SHA256	Y	Y	Y	Y	Y	Y
SHA384	Y	Y	Y	Y	Y	Y
SHA512	Y	Y	Y	Y	Y	Y
RSA キー	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット
DH キー	1024 ビットと 2048 ビット	1024 ビットと 2048 ビット	1024 ビットと 2048 ビット	1024 ビットと 2048 ビット	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット

サーバー証明書/プラットフォーム	MPX/SDX 14030/14060/14080 FIPS FE	MPX/SDX 14030/14060/14080 FIPS BE
MD5	Y	Y
SHA1	Y	Y
SHA224	Y	Y
SHA256	Y	Y
SHA384	Y	Y
SHA512	Y	Y
RSA キー	2048 ビットと 3072 ビット	2048 ビットと 3072 ビット
DH キー	N	N

サーバー証明書/プラットフォーム	MPX 5900、MPX/SDX 8900、MPX/SDX 9100、MPX/SDX 15000、MPX/SDX 15000-50G、MPX/SDX 16000、MPX/SDX 26000、MPX/SDX 26000-50G、MPX/SDX 26000-50G、MPX/SDX 26000-100G (フロントエンド)	MPX 5900、MPX/SDX 8900、MPX/SDX 9100 MPX/SDX 15000、MPX/SDX 15000-50G、MPX/SDX 16000、MPX/SDX 26000、MPX/SDX 26000-50G、MPX/SDX 26000-50G、MPX/SDX 26000-100G (バックエンド)
MD5	Y	Y
SHA1	Y	Y
SHA224	Y	Y
SHA256	Y	Y
SHA384	Y	Y
SHA512	Y	Y
RSA キー	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット
DH キー	1024 ビットと 2048 ビット	1024 ビットと 2048 ビット

メモ

- 4k 証明書はより高い CPU サイクルを必要とし、ローエンドアプライアンスのパフォーマンスに影響を与える可能性があります。
- リリース 11.1 以前では、NetScaler ADC アプライアンスは、バックエンドクライアントのハローメッセージで、RSA-MD5、RSA-SHA1、および RSA-SHA256 の「署名アルゴリズム」拡張をサポートします。NetScaler ADC アプライアンスは、SHA 384 および SHA 512 署名アルゴリズム拡張をサポートしていません。そのため、Windows IIS サーバーなどの一部のサーバーでは、接続がリセットされます。
- リリース 12.0 以降、NetScaler ADC アプライアンスはすべての signature_algorithms 拡張をサポートします。

クライアント認証または相互 **TLS (mTLS)**

August 15, 2023

通常の SSL トランザクションでは、セキュリティで保護された接続を介してサーバーに接続しているクライアントが、サーバーの有効性を確認します。そのためには、SSL トランザクションを開始する前にサーバーの証明書をチェックします。ただし、サーバーに接続しているクライアントを認証するようにサーバーを構成したい場合があります。

注：リリース 13.0 ビルド 41.x 以降、NetScaler ADC アプライアンスは、合計サイズが 32 KB 以内の場合に複数のレコードにフラグメント化される証明書要求メッセージをサポートします。以前は、サポートされる最大サイズは 16 KB で、フラグメンテーションはサポートされていませんでした。

SSL 仮想サーバーでクライアント認証を有効にすると、NetScaler ADC アプライアンスは SSL ハンドシェイク中にクライアント証明書を要求します。アプライアンスは、発行者の署名や有効期限などの通常の制約について、クライアントから提示された証明書をチェックします。

リリース 13.1 ビルド 42.x 以降、NetScaler ADC アプライアンスはクロス署名証明書検証をサポートしています。つまり、証明書が複数の発行者によって署名されている場合、ルート証明書への有効なパスが少なくとも 1 つあれば検証は成功します。以前は、証明書チェーン内の証明書の 1 つがクロス署名されていて、ルート証明書へのパスが複数ある場合、ADC アプライアンスはパスを 1 つだけチェックしていました。そのパスが有効でない場合、検証は失敗しました。

> （注）
>
> アプライアンスが発行者の署名を検証するには、クライアント証明書を発行した CA の証明書が次の条件を満たしている必要があります。
>
> - アプライアンスにインストールされている。
> - クライアントがトランザクションを行っている仮想サーバにバインドされます。

証明書が有効な場合、アプライアンスはクライアントにすべてのセキュアなリソースへのアクセスを許可します。ただし、証明書が無効な場合、アプライアンスは SSL ハンドシェイク中にクライアント要求を破棄します。

アプライアンスは、まずクライアント証明書から始まり、クライアントのルート CA 証明書（Verisign など）で終わる証明書のチェーンを作成して、クライアント証明書を検証します。ルート CA 証明書には、1 つまたは複数の中間 CA 証明書が含まれる場合があります (ルート CA がクライアント証明書を直接発行しない場合)。

NetScaler ADC アプライアンスでクライアント認証を有効にする前に、有効なクライアント証明書がクライアントにインストールされていることを確認してください。次に、トランザクションを処理する仮想サーバのクライアント認証を有効にします。最後に、クライアント証明書を発行した CA の証明書をアプライアンス上の仮想サーバーにバインドします。

注： NetScaler MPX アプライアンスは、512 ビットから 4096 ビットの証明書とキーのペアのサイズをサポートしています。証明書は、次のいずれかのハッシュアルゴリズムを使用して署名する必要があります。

- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

SDX アプライアンスでは、SSL チップが VPX インスタンスに割り当てられている場合、MPX アプライアンスの証明書とキーのペアサイズのサポートが適用されます。それ以外の場合は、VPX インスタンスの通常の証明書とキーのペ

アサイズのサポートが適用されます。

NetScaler ADC 仮想アプライアンス（VPX インスタンス）は、次のサイズまで、512 ビット以上の証明書をサポートします。

- 仮想サーバー上の 4096 ビットサーバー証明書
- サービス上の 4096 ビットのクライアント証明書
- 4096 ビット CA 証明書
- 物理サーバー上の 4096 ビット証明書

リリース 13.1 ビルド 17.x 以降、すべての NetScaler プラットフォームは、RSASSA-PSS アルゴリズムを使用して署名された証明書をサポートしています。
これらのアルゴリズムは X.509 証明書パスの検証でサポートされています。
次の表は、NetScaler アプライアンスでサポートされている RSASSA-PSS パラメーターセットを示しています。

パブリックキー OID	マスク生成関数 (MGF)	MGF ダイジェスト機能	シグネチャダイジェスト関数	塩の長さ
RSAEncryption	MGF1	SHA-256	SHA-256	32 バイト
RSAEncryption	MGF1	SHA-384	SHA-384	48 バイト
RSAEncryption	MGF1	SHA-512	SHA-512	64 バイト

注：リリース 13.0 ビルド 79.x 以降、VPX プラットフォームでの SSL ハンドシェイク中、4096 ビット RSA クライアント証明書によるクライアント認証がサポートされています。

注：

- MPX FIPS の制限については、MPX FIPS の制限を参照してください。
- SDX FIPS の制限については、SDX FIPS の制限を参照してください。

クライアント証明書の提供

クライアント認証を構成する前に、有効なクライアント証明書をクライアントにインストールしておく必要があります。クライアント証明書には、NetScaler ADC アプライアンスとの安全なセッションを作成する特定のクライアントシステムに関する詳細が含まれています。各クライアント証明書は一意であり、1 つのクライアントシステムでのみ使用する必要があります。

CA からクライアント証明書を取得するか、既存のクライアント証明書を使用するか、NetScaler ADC アプライアンスでクライアント証明書を生成するかにかかわらず、証明書を正しい形式に変換する必要があります。NetScaler ADC アプライアンスでは、証明書は PEM または DER 形式で保存され、クライアントシステムにインストールする前に PKCS #12 形式に変換する必要があります。証明書を変換してクライアントシステムに転送したら、証明書がそのシステムにインストールされ、クライアントアプリケーション用に構成されていることを確認します。Web ブラウザーなどのアプリケーションは、SSL トランザクションの一部である必要があります。

証明書を PEM または DER 形式から PKCS #12 形式に変換する方法については、「SSL ファイルのインポートと変換」を参照してください。

クライアント証明書を生成する方法については、「証明書の作成」を参照してください。

クライアント証明書ベースの認証を有効にする

デフォルトでは、NetScaler ADC アプライアンスではクライアント認証が無効になっており、すべての SSL トランザクションはクライアントを認証せずに続行されます。SSL ハンドシェイクの一部として、クライアント認証をオプションまたは必須のいずれかに設定できます。

クライアント認証がオプションの場合、アプライアンスはクライアント証明書を要求しますが、クライアントが無効な証明書を提示した場合でも SSL トランザクションは続行されます。クライアント認証が必須の場合、SSL クライアントが有効な証明書を提供しない場合、アプライアンスは SSL ハンドシェイクを終了します。

注意：クライアント証明書ベースの認証チェックをオプションに変更する前に、適切なアクセス制御ポリシーを定義することをお勧めします。

注**:** クライアント認証は、グローバルではなく個々の SSL 仮想サーバーに対して構成されます。

CLI を使用してクライアント証明書ベースの認証を有効にする

コマンドプロンプトで次のコマンドを入力して、クライアント証明書ベースの認証を有効にし、構成を確認します。

```
set ssl vserver <vServerName> [-clientAuth (ENABLED | DISABLED)] [-
   clientCert (MANDATORY | OPTIONAL)]
show ssl vserver <vServerName>
<!--NeedCopy-->
```

例**:**

```
set ssl vserver vssl -clientAuth ENABLED -clientCert Mandatory
Done
show ssl vserver vssl

Advanced SSL configuration for VServer vssl:
DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
```



```
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.2: ENABLED TLSv1
    .2: ENABLED

1) CertKey Name: sslckey Server Certificate

1) Policy Name: client_cert_policy Priority: 0

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用してクライアント証明書ベースの認証を有効にする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで [クライアント認証] を選択し、[クライアント証明書] リストで [必須] を選択します。

> 注:
>
> クライアント認証が必須に設定されていて、クライアント証明書にポリシー拡張が含まれていると、証明書の検証は失敗します。リリース 12.0-56.x からは、フロントエンド SSL プロファイルにパラメータを設定して、このチェックをスキップできます。このパラメーターはデフォルトでは無効になっています。つまり、このチェックはデフォルトで実行されます。

CLI を使用してクライアント認証中にポリシー拡張チェックをスキップする

コマンドプロンプトで入力します。

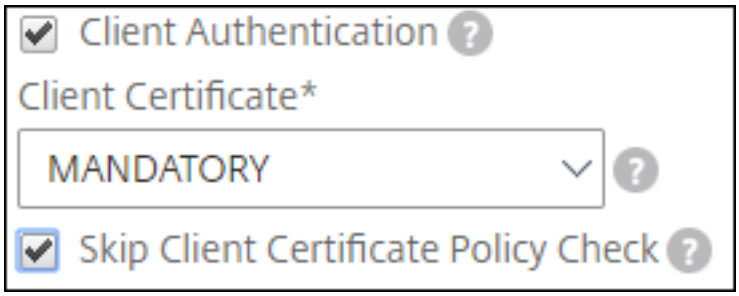
```
set ssl profile ns_default_ssl_profile_frontend -clientauth ENABLED -
    skipClientCertPolicyCheck ENABLED

Parameter

skipClientCertPolicyCheck

              Control policy extension check, if present inside the
                 X509 certificate chain. Applicable only if client
                 authentication is enabled and client certificate is
                 set to mandatory. Possible values functions as follows
                 :

- ENABLED: Skip the policy check during client authentication.

- DISABLED: Perform policy check during client authentication.

```



```
Possible values: ENABLED, DISABLED

Default: DISABLED
<!--NeedCopy-->
```

GUI を使用してクライアント認証中にポリシー拡張チェックをスキップする

1. [システム] > [プロファイル] > [**SSL** プロファイル] に移動します。
2. 新しいフロントエンドプロファイルを作成するか、既存のフロントエンドプロファイルを編集します。
3. クライアント認証が有効になっていて、クライアント証明書が必須に設定されていることを確認します。
4. [クライアント証明書ポリシーチェックをスキップ] を選択します。



CA 証明書を仮想サーバにバインドする

NetScaler ADC アプライアンスに証明書が存在する CA は、クライアント認証に使用されるクライアント証明書を発行する必要があります。この証明書を、クライアント認証を実行する NetScaler ADC 仮想サーバーにバインドします。

アプライアンスがクライアント証明書を検証するときに完全な証明書チェーンを形成できるように、CA 証明書を SSL 仮想サーバーにバインドします。そうしないと、証明書チェーンの形成に失敗し、証明書が有効であってもクライアントはアクセスを拒否されます。

CA 証明書は SSL 仮想サーバに任意の順序でバインドできます。アプライアンスは、クライアント証明書の検証中に正しい順序を形成します。

たとえば、クライアントが **CA_A** によって発行された証明書を提示する場合、**CA_A** は証明書が **CA_B** によって発行される中間 CA であり、証明書は信頼されたルート CA **Root_CA** によって発行されます。この証明書は、これらの 3 つの証明書はすべて、NetScaler ADC アプライアンス上の仮想サーバーにバインドする必要があります。

1 つ以上の証明書を仮想サーバーにバインドする手順については、SSL 仮想サーバーへの証明書とキーのペアのバインドを参照してください。

証明書のチェーンを作成する手順については、「証明書のチェーンを作成する」を参照してください。

クライアント証明書の検証の厳密な制御

NetScaler ADC アプライアンスは、単一のルート CA によって発行された場合、有効な中間 CA 証明書を受け入れます。つまり、ルート CA 証明書のみが仮想サーバーにバインドされ、そのルート CA がクライアント証明書とともに送信された中間証明書を検証すると、アプライアンスは証明書チェーンを信頼し、ハンドシェイクは成功します。

ただし、クライアントがハンドシェイクで一連の証明書を送信する場合、証明書が SSL 仮想サーバーにバインドされていない限り、CRL または OCSP レスポンダーを使用して中間証明書を検証することはできません。したがって、中間証明書の 1 つが失効しても、ハンドシェイクは成功します。ハンドシェイクの一部として、SSL 仮想サーバはバインドされている CA 証明書のリストを送信します。より厳密に制御するために、その仮想サーバーにバインドされた CA 証明書の 1 つによって署名された証明書のみを受け入れるように SSL 仮想サーバーを構成できます。これを行うには、仮想サーバーにバインドされている SSL プロファイルで **ClientAuthUseBoundCAChain** 設定を有効にする必要があります。仮想サーバーにバインドされている CA 証明書の 1 つがクライアント証明書に署名していない場合、ハンドシェイクは失敗します。

たとえば、clientcert1 と clientcert2 の 2 つのクライアント証明書が、それぞれ中間証明書 int-CA-A と int-CA-B によって署名されているとします。中間証明書は、ルート証明書 root-CA によって署名されます。int-CA-A とルート CA は SSL 仮想サーバにバインドされます。デフォルトの場合 (ClientAuthuseBoundCachain は無効)、clientcert1 と clientcert2 の両方が受け入れられます。ただし、ClientAuthUseBoundCachain が有効な場合、NetScaler ADC アプライアンスは clientcert1 のみを受け入れます。

CLI を使用して、クライアント証明書の検証をより厳密に制御できるようにする

コマンドプロンプトで入力します。

```
set ssl profile <name> -ClientAuthUseBoundCAChain Enabled
<!--NeedCopy-->
```

GUI を使用してクライアント証明書の検証をより厳密に制御できるようにする

1. [システム] > [プロファイル] に移動し、[**SSL** プロファイル] タブを選択して SSL プロファイルを作成するか、既存のプロファイルを選択します。
2. [バインドされた **CA** チェーンを使用したクライアント認証を有効にする] を選択します。

サーバー認証

August 15, 2023

NetScaler ADC アプライアンスは Web サーバーに代わって SSL オフロードとアクセラレーションを実行するため、アプライアンスは通常、Web サーバーの証明書を認証しません。ただし、エンドツーエンド SSL 暗号化を必要とする展開環境では、サーバを認証できます。

このような状況では、アプライアンスが SSL クライアントになり、SSL サーバーとの安全なトランザクションを実行します。証明書が SSL サービスにバインドされている CA がサーバー証明書に署名していることを確認し、サーバー証明書の有効性を確認します。

サーバーを認証するには、サーバー認証を有効にし、サーバーの証明書に署名した CA の証明書を ADC アプライアンスの SSL サービスにバインドします。証明書をバインドする場合、CA オプションとしてバインドを指定する必要があります。

リリース 13.1 ビルド 42.x 以降、NetScaler ADC アプライアンスはクロス署名証明書検証をサポートしています。つまり、証明書が複数の発行者によって署名されている場合、ルート証明書への有効なパスが少なくとも 1 つあれば検証は成功します。以前は、証明書チェーン内の証明書の 1 つがクロス署名されていて、ルート証明書へのパスが複数ある場合、ADC アプライアンスはパスを 1 つだけチェックしていました。そのパスが有効でない場合、検証は失敗しました。

サーバー証明書認証の有効化（または無効化）

CLI と GUI を使用して、サーバー証明書認証を有効または無効にできます。

CLI を使用してサーバー証明書認証を有効化 **(**または無効化**)** する

コマンドプロンプトで次のコマンドを入力して、サーバー証明書認証を有効にし、構成を確認します。

```
set ssl service <serviceName> -serverAuth ( ENABLED | DISABLED )
show ssl service <serviceName>
<!--NeedCopy-->
```

例**:**

```
set ssl service ssl-service-1 -serverAuth ENABLED

show ssl service ssl-service-1

            Advanced SSL configuration for Back-end SSL Service ssl-
               service-1:`
            DH: DISABLED
            Ephemeral RSA: DISABLED
            Session Reuse: ENABLED          Timeout: 300 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            Server Auth: ENABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED
```



```
15      1)      Cipher Name: ALL
16              Description: Predefined Cipher Alias
17  Done
18  <!--NeedCopy-->
```

GUI を使用してサーバー証明書認証を有効化 **(**または無効化**)** する

1. [トラフィック管理**] > [**負荷分散**] > [**サービス] に移動し、SSL サービスを開きます。
2. 「SSL パラメータ」セクションで、「サーバー認証を有効にする」を選択し、共通名を指定します。
3. [詳細設定] で [証明書] を選択し、CA 証明書をサービスにバインドします。

CLI を使用して **CA** 証明書をサービスにバインドします

コマンドプロンプトで次のコマンドを入力して、CA 証明書をサービスにバインドし、構成を確認します。

```
1  bind ssl service <serviceName> -certkeyName <string> -CA
2
3  show ssl service <serviceName>
4  <!--NeedCopy-->
```

例**:**

```
1  bind ssl service ssl-service-1 -certkeyName samplecertkey -CA
2
3  show ssl service ssl-service-1
4
5              Advanced SSL configuration for Back-end SSL Service ssl-
                  service-1:
6              DH: DISABLED
7              Ephemeral RSA: DISABLED
8              Session Reuse: ENABLED          Timeout: 300 seconds
9              Cipher Redirect: DISABLED
10             SSLv2 Redirect: DISABLED
11             Server Auth: ENABLED
12             SSL Redirect: DISABLED
13             Non FIPS Ciphers: DISABLED
14             SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED
15      1)     CertKey Name: samplecertkey     CA Certificate
           CRLCheck: Optional
16      1)     Cipher Name: ALL
17             Description: Predefined Cipher Alias
18 Done
19 <!--NeedCopy-->
```


サーバー証明書認証の共通名の設定

サーバー認証を有効にしたエンドツーエンド暗号化では、SSL サービスまたはサービスグループの設定に共通名を含めることができます。指定した名前は、SSL ハンドシェイク中にサーバー証明書の共通名と比較されます。2 つの名前が一致すれば、ハンドシェイクは成功します。
共通名が一致しない場合、サービスまたはサービスグループに指定された共通名が、証明書のサブジェクト代替名 (SAN) フィールドの値と比較されます。これらの値のいずれかに一致すれば、ハンドシェイクは成功します。この構成は、たとえば、ファイアウォールの内側に 2 つのサーバーがあり、一方のサーバーが他方のサーバーの ID を偽装している場合に特に役立ちます。共通名がチェックされていない場合、IP アドレスが一致すれば、どちらかのサーバから提示された証明書が受け入れられます。

注**:** SAN フィールドのドメイン名、URL、電子メール ID の DNS エントリのみが比較されます。

CLI を使用して **SSL** サービスまたはサービスグループの共通名検証を設定する

コマンドプロンプトで次のコマンドを入力して、共通名検証によるサーバー認証を指定し、構成を確認します。

1. サービスの共通名を設定するには、次のように入力します。

```
set ssl service <serviceName> -commonName <string> -serverAuth
   ENABLED
show ssl service <serviceName>
<!--NeedCopy-->
```

2. サービスグループの共通名を設定するには、次のように入力します。

```
set ssl serviceGroup <serviceGroupName> -commonName <string> -
   serverAuth ENABLED
show ssl serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

例**:**

```
set ssl service svc1 -commonName xyz.com -serverAuth ENABLED

show ssl service svc

     Advanced SSL configuration for Back-end SSL Service svc1:
     DH: DISABLED
     Ephemeral RSA: DISABLED
     Session Reuse: ENABLED Timeout: 300 seconds
     Cipher Redirect: DISABLED
     SSLv2 Redirect: DISABLED
     Server Auth: ENABLED Common Name: www.xyz.com
     SSL Redirect: DISABLED
     Non FIPS Ciphers: DISABLED
     SNI: DISABLED
     SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
```



```
    1) CertKey Name: cacert CA Certificate OCSPCheck: Optional
    1) Cipher Name: ALL
     Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用して **SSL** サービスまたはサービスグループの共通名検証を設定する

1. [トラフィック管理] **>** [負荷分散] **>** [サービス] に移動するか、[トラフィック管理] **>** [負荷分散] **>** [サービスグループ] に移動し、サービスまたはサービスグループを開きます。
2. 「**SSL** パラメータ」セクションで、「サーバー認証を有効にする」を選択し、共通名を指定します。

SSL アクションとポリシー

August 15, 2023

SSL ポリシーは、着信トラフィックを評価し、ルール（式）に一致する要求に定義済みのアクションを適用します。ポリシーの作成時にアクションを指定できるように、ポリシーを作成する前にアクションを設定します。ポリシーを有効にするには、次のいずれかを実行します。

- ポリシーをアプライアンス上の仮想サーバにバインドして、その仮想サーバを通過するトラフィックにのみ適用されるようにします。
- ポリシーをグローバルにバインドして、アプライアンスを通過するすべてのトラフィックに適用します。

SSL アクションは、選択したリクエストに適用できる SSL 設定を定義します。アクションを 1 つ以上のポリシーに関連付けます。クライアント接続要求またはクライアント接続応答のデータは、ポリシーで指定された規則と比較され、その規則 (式) に一致する接続にアクションが適用されます。

クラシックポリシーをクラシック式で構成し、高度なポリシーポリシーを SSL の高度なポリシー式で構成できます。

注**:** CLI でのポリシーの設定に慣れていないユーザは、通常、設定ユーティリティを使用する方がはるかに簡単です。

ユーザー定義アクションまたは組み込みアクションを詳細ポリシーに関連付けることができます。クラシックポリシーでは、ユーザー定義のアクションのみが許可されます。[詳細ポリシー] では、ポリシーラベルの下にポリシーをグループ化することもできます。この場合、ポリシーは、別のポリシーから呼び出された場合にのみ適用されます。

SSL アクションとポリシーの一般的な用途には、ディレクトリごとのクライアント認証、Outlook Web アクセスのサポート、SSL ベースのヘッダー挿入などがあります。SSL ベースのヘッダー挿入には、SSL 処理が NetScaler ADC アプライアンスにオフロードされているサーバーに必要な SSL 設定が含まれています。

SSL ポリシー

August 15, 2023

NetScaler ADC アプライアンスのポリシーは、処理する特定の接続を識別するのに役立ちます。処理は、その特定のポリシーに対して設定されたアクションに基づいて行われます。ポリシーを作成し、そのアクションを設定したら、次のいずれかを実行する必要があります。

- ポリシーをアプライアンス上の仮想サーバにバインドして、その仮想サーバを通過するトラフィックにのみ適用されるようにします。
- ポリシーをグローバルにバインドして、NetScaler ADC アプライアンスで構成された仮想サーバーを通過するすべてのトラフィックに適用します。

NetScaler ADC アプライアンスの SSL 機能は、高度なポリシー（詳細）ポリシーをサポートします。高度なポリシー式の詳細、その仕組み、および手動での設定方法の詳細については、「ポリシーと式」を参照してください。SSL 式の詳細については、「高度なポリシー式:SSL の解析」を参照してください。

> 注:
>
> CLI でのポリシーの設定に慣れていないユーザは、通常、設定ユーティリティの使用がかなり簡単になります。

SSL ポリシーでは、ポリシーの作成時にアクションを指定できるように、ポリシーを作成する前にアクションを作成する必要があります。
SSL Advanced ポリシーでは、組み込みアクションを使用することもできます。組み込みアクションの詳細については、SSL 組み込みアクションとユーザー定義アクションを参照してください。

SSL 詳細ポリシー

SSL アドバンスドポリシーは、高度なポリシーとも呼ばれ、要求に対して実行されるコントロールまたはデータアクションを定義します。したがって、SSL ポリシーは、制御ポリシーとデータポリシーに分類できます。

- 制御ポリシー。制御ポリシーは、クライアント認証の強制などの制御アクションを使用します。
  注: リリース 10.5 以降では、SSL 再ネゴシエーションの拒否（denySSLreneg）はデフォルトで ALL に設定されます。ただし、CLIENTEAUTH などの制御ポリシーは、再ネゴシエーションハンドシェイクをトリガーします。このようなポリシーを使用する場合は、denySSLreneg を NO に設定する必要があります。
- データポリシー。データポリシーは、リクエストへのデータの挿入などのデータアクションを使用します。

ポリシーの重要なコンポーネントは、式とアクションです。式は、アクションが実行されるリクエストを識別します。

高度なポリシーは、組み込みアクションまたはユーザー定義アクションで構成できます。別のアクションを作成しなくても、組み込みアクションを使用してポリシーを設定できます。ただし、ユーザ定義アクションを使用してポリシーを設定するには、まずアクションを設定してから、ポリシーを設定します。

リクエストに式を適用した結果が未定義である場合に実行される、UNDEF アクションと呼ばれる追加のアクションを指定できます。

SSL ポリシー設定

SSL 詳細ポリシーは、CLI および GUI を使用して設定できます。

CLI を使用して SSL ポリシーを設定する

コマンドプロンプトで入力します。

```
1 add ssl policy <name> -rule <expression> -Action <string> [-undefAction
      <string>] [-comment <string>]
2 <!--NeedCopy-->
```

GUI を使用した SSL ポリシーの設定

[トラフィック管理] **>** **[SSL]** **>** **[**ポリシー] に移動し、[ポリシー] タブで [追加] をクリックします。

TLS1.3 プロトコルでの SSL ポリシーのサポート

リリース 13.0 ビルド 71.x 以降から、TLS1.3 プロトコルを使用した SSL ポリシーのサポートが追加されました。TLSv1.3 プロトコルが接続に対してネゴシエートされると、クライアントから受信した TLS データを検査するポリシールールが、設定されたアクションをトリガーするようになりました。

たとえば、次のポリシールールが true を返した場合、トラフィックはアクションで定義された仮想サーバーに転送されます。

```
1 add ssl action action1 -forward vserver2
2 add ssl policy pol1 -rule client.ssl.client_hello.sni.contains( “xyz” )
      -action action1
3 <!--NeedCopy-->
```

制限事項

- 制御ポリシーはサポートされていません。
- 次のアクションはサポートされていません。
  - DOCLIENTAUTH
  - NOCLIENTAUTH
  - caCertGrpName

- clientCertVerification
- ssllogProfile

SSL 組み込みアクションとユーザー定義アクション

August 15, 2023

ポリシーに組み込みアクションのみが必要な場合を除き、ポリシーを作成する前にアクションを作成する必要があります。次に、ポリシーを作成するときにアクションを指定できます。組み込みアクションには、制御アクションとデータアクションの 2 種類があります。コントロールポリシーではコントロールアクションを使用し、データポリシーではデータアクションを使用します。

組み込みのコントロールアクションは以下のとおりです。

- doClientAuth —クライアント証明書認証を実行します。(TLS1.3 ではサポートされていません)
- noClientAuth —クライアント証明書認証を実行しないでください。(TLS1.3 ではサポートされていません)

組み込みデータアクションは以下のとおりです。

- リセット-RST パケットをクライアントに送信して接続を閉じます。
- ドロップ：クライアントからすべてのパケットをドロップします。接続は、クライアントが閉じるまで開いたままになります。
- noop：何も処理せずにパケットを転送します。

注**:** ClientCertVerification や SSLogProfile などのクライアント認証に依存するアクションは、TLS 1.3 プロトコルではサポートされていません。

ユーザー定義のデータアクションを作成できます。クライアント認証を有効にすると、リクエストを Web サーバーに転送する前に、クライアント証明書データをリクエストヘッダーに挿入する SSL アクションを作成できます。

ポリシー評価の結果が未定義の状態になった場合、UNDEF アクションが実行されます。データポリシーまたは制御ポリシーのどちらでも、UNDEF アクションとして RESET、DROP、または NOOP を指定できます。コントロールポリシーには、DOCLIENTAUTH または NOCLIENTAUTH を指定するオプションもあります。

ポリシーに組み込まれたアクションの例

次の例では、クライアントが EXPORT カテゴリの暗号以外の暗号を送信すると、NetScaler アプライアンスはクライアント認証を要求します。トランザクションを成功させるには、クライアントは有効な証明書を提供する必要があります。

```
1 add ssl policy pol1 -rule CLIENT.SSL.CIPHER_EXPORTABLE.NOT -reqAction
     DOCLIENTAUTH
2 <!--NeedCopy-->
```


次の例では、クライアント認証が有効になっていることを前提としています。

ユーザーから提供された証明書のバージョンがポリシーのバージョンと一致する場合、アクションは実行されず、パケットが転送されます。

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
    reqAction NOOP
<!--NeedCopy-->
```

ユーザーから提供された証明書のバージョンがポリシーのバージョンと一致する場合、接続は切断されます。

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
    reqAction DROP
<!--NeedCopy-->
```

ユーザーから提供された証明書のバージョンがポリシーのバージョンと一致する場合、接続はリセットされます。

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
    reqAction RESET
<!--NeedCopy-->
```

ポリシーベースのクライアント認証によるクライアント証明書の検証

ポリシーベースのクライアント認証を設定している場合、クライアント証明書の検証を必須またはオプションに設定できます。デフォルトは必須です。

CLI を使用してクライアント証明書の検証をオプションに設定する

コマンドプロンプトで入力します。

```
add ssl action <name> ((-clientAuth ( DOCLIENTAUTH | NOCLIENTAUTH ) [-
    clientCertVerification ( Mandatory | Optional )]
<!--NeedCopy-->
```

例**:**

```
add ssl action sslact -clientauth DOCLIENTAUTH -clientcertverification
    OPTIONAL
<!--NeedCopy-->
```

GUI を使用してクライアント証明書の検証をオプションに設定します

1. [**Traffic Management**] **>** [**SSL**] **>** [**Policies**] に移動します。
2. 「**SSL** アクション」タブで、「追加」をクリックします。
3. 名前を指定し、「クライアント証明書の検証」リストで「オプション」を選択します。

ユーザー定義の **SSL** アクション

組み込みアクションに加えて、デプロイメントに応じて他の SSL アクションを設定することもできます。これらのアクションはユーザー定義アクションと呼ばれます。

CLI を使用してユーザー定義の **SSL** アクションを設定します

コマンドプロンプトで次のコマンドを入力してアクションを構成し、構成を確認します。

```
add SSL action <name> -clientAuth(DOCLIENTAUTH | NOCLIENTAUTH) -
    clientCert (ENABLED | DISABLED) certHeader <string> -clientHeader <
    string> -clientCertSerialNumber (ENABLED | DISABLED) -
    certSerialHeader <string> -clientCertSubject (ENABLED | DISABLED) -
    certSubjectHeader <string> -clientCertHash (ENABLED | DISABLED) -
    certHashHeader <string> -clientCertIssuer (ENABLED | DISABLED) -
    certIssuerHeader <string> -sessionID (ENABLED | DISABLED) -
    sessionIDheader <string> -cipher (ENABLED | DISABLED) -cipherHeader
    <string> -clientCertNotBefore (ENABLED | DISABLED) -
    certNotBeforeHeader <string> -clientCertNotAfter (ENABLED | DISABLED
    ) -certNotAfterHeader <string> -OWASupport (ENABLED | DISABLED)
<!--NeedCopy-->
```

```
show ssl action [<name>]
<!--NeedCopy-->
```

例**:**

```
add ssl action Action-SSL-ClientCert -clientCert ENABLED -certHeader "X
    -Client-Cert"
<!--NeedCopy-->
```

```
show ssl action Action-SSL-ClientCert

1)      Name: Action-SSL-ClientCert
        Data Insertion Action:
        Cert Header: ENABLED            Cert Tag: X-Client-Cert
Done
<!--NeedCopy-->
```

GUI を使用してユーザー定義の **SSL** アクションを設定する

[トラフィック管理**] > [SSL] > [**ポリシー] に移動し、[アクション] タブで [追加] をクリックします。

クライアントトラフィックを別の仮想サーバーに転送する **SSL** アクションの設定

管理者は、SSL オフロードを回避するために、SSL 仮想サーバーで受信したクライアントトラフィックを別の仮想サーバーに転送する SSL アクションを構成できます。または、ADC アプライアンスの接続を終了する場合にも使用で

きます。この仮想サーバーには、SSL、TCP、または SSL_BRIDGE のいずれかのタイプがあります。たとえば、管理者は、次のいずれかの場合に接続を終了する代わりに、リクエストを別の仮想サーバーに転送してさらにアクションを起こすことを選択できます。

- アプライアンスには証明書がありません。
- アプライアンスは特定の暗号をサポートしていません。

上記を実現するために、クライアント hello を受信したときにクライアントトラフィックを評価する新しいバインドポイント‘CLIENTHELLO_REQ’が追加されました。クライアントトラフィックを受信する仮想サーバーにバインドされたポリシーが、client hello を解析した後に true と評価された場合、トラフィックは別の仮想サーバーに転送されます。この仮想サーバーが SSL タイプの場合、ハンドシェイクを実行します。この仮想サーバーのタイプが TCP または SSL_BRIDGE の場合、バックエンドサーバーがハンドシェイクを実行します。

リリース 12.1-49.x では、CLIENTHELLO_REQ バインドポイントではフォワードアクションとリセットアクションのみがサポートされています。次のエクスプレッションプレフィックスを使用できます。

- CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXCODE
- CLIENT.SSL.CLIENT_HELLO.CLIENT_VERSION
- CLIENT.SSL.CLIENT_HELLO.IS_RENEGOTIATE
- CLIENT.SSL.CLIENT_HELLO.IS_REUSE
- CLIENT.SSL.CLIENT_HELLO.IS_SCSV
- CLIENT.SSL.CLIENT_HELLO.IS_SESSION_TICKET
- CLIENT.SSL.CLIENT_HELLO.LENGTH
- CLIENT.SSL.CLIENT_HELLO.SNI
- CLIENT.SSL.CLIENT_HELLO.ALPN.HAS_NEXTPROTOCOL (13.0 ビルド 61.x から)

これらのプレフィクスの説明については、「高度なポリシー式:SSL の解析」を参照してください。

`add SSL action`コマンドにパラメータ`forward`が追加され、新しいバインドポイント`CLIENTHELLO_REQ`が`bind ssl vserver`コマンドに追加されます。

CLI を使用した設定

コマンドプロンプトで入力します。

```
add ssl action <name> -forward <virtual server name>

add ssl policy <name> -rule <expression> -action <string>

bind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type <type>
<!--NeedCopy-->
```

例:


```
add ssl action act1 -forward v2

add ssl policy pol1 -rule client.ssl.client_hello.ciphers.has_hexcode(0
   x002f) -action act1

bind ssl vserver v1 -policyName pol1 -priority 1 -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

GUI を使用した設定

[**Traffic Management**] > [**SSL**] > [**Policies**] に移動します。

SSL アクションの作成:

1. 「**SSL** アクション」で、「追加」をクリックします。
2. 「**SSL** アクションの作成」で、アクションの名前を指定します。
3. 「転送アクション仮想サーバー」で、既存の仮想サーバーを選択するか、トラフィックを転送する新しい仮想サーバーを追加します。
4. オプションで、他のパラメータを設定します。
5. [作成] をクリックします。

SSL ポリシーの作成:

1. **SSL** ポリシーで、「追加」をクリックします。
2. 「**SSL** ポリシーの作成」で、ポリシーの名前を指定します。
3. 「アクション」で、前に作成したアクションを選択します。
4. エクスプレッションエディタで、評価するルールを入力します。
5. [作成] をクリックします。

仮想サーバーとバインドポリシーを作成または追加します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバーを追加または選択します。
3. [詳細設定] で、[**SSL** ポリシー] をクリックします。
4. SSL ポリシーセクションをクリックします。
5. 「ポリシーの選択」で、以前に作成したポリシーを選択します。
6. 「ポリシー・バインディング」で、ポリシーの優先度を指定します。
7. 「タイプ」で「**CLIENTHELLO_REQ**」を選択します。
8. [**Bind**] をクリックします。
9. [完了] をクリックします。

最も一般的なユースケースのエンドツーエンド構成については、次のトピックを参照してください。

- アプライアンスにドメイン固有（SNI）証明書がない場合は、クライアントトラフィックを転送するように SSL アクションを設定します。

- クライアント hello メッセージの ALPN 拡張のプロトコルに基づいてクライアントトラフィックを転送するように SSL アクションを設定します。
- ADC で暗号がサポートされていない場合に、クライアントトラフィックを転送するように SSL アクションを設定します。

クライアント認証用に **SNI** に基づいて **CA** を選択的に選択する **SSL** アクション

SSL 仮想サーバーにバインドされたすべての CA のリストではなく、クライアント証明書要求の SNI (ドメイン) に基づく CA のリストのみを送信できます。たとえば、クライアント hello を受信すると、SSL ポリシー表現 (SNI など) に基づく CA 証明書のみが送信されます。特定の証明書セットを送信するには、CA 証明書グループを作成する必要があります。次に、このグループを SSL アクションにバインドし、アクションを SSL ポリシーにバインドします。クライアントトラフィックを受信する仮想サーバーにバインドされたポリシーが、client hello を解析した後に true と評価された場合、特定の CA 証明書グループのみがクライアント要求証明書で送信されます。

以前は、CA 証明書を SSL 仮想サーバーにバインドする必要がありました。この拡張機能により、CA 証明書グループを追加して SSL アクションに関連付けるだけで済みます。

注**:** SSL 仮想サーバーでクライアント認証と SNI を有効にします。正しい SNI 証明書を仮想サーバーにバインドします。

次の手順を実行します:

1. CA 証明書グループを追加します。
2. 証明書とキーのペアを追加します。
3. 証明書とキーのペアをこのグループにバインドします。
4. SSL アクションを追加します。
5. SSL ポリシーを追加します。ポリシーにアクションを指定してください。
6. ポリシーを SSL 仮想サーバーにバインドします。バインドポイントを CLIENTHELLO_REQ として指定します。

CLI を使用した設定

コマンドプロンプトで、次のコマンドを順番に入力します。

```
add ssl caCertGroup <caCertGroupName>
add ssl certkey <certkey_name> -cert <cert> -key <key>
bind ssl caCertGroup <caCertGroupName> <certkey_name>
add ssl action <name> -caCertGrpName <string>
add ssl policy <name> -rule <expression> -action <string>
bind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type CLIENTHELLO_REQ
<!--NeedCopy-->
```


例：

```
add ssl cacertGroup ca_cert_group

add ssl certkey ca_certkey1 -cert cacert1 -key cakey1
add ssl certkey ca_certkey2 -cert cacert2 -key cakey2
add ssl certkey snicert -cert snicert -key snikey

bind ssl cacertGroup ca_cert_group ca_certkey1
bind ssl caCertGroup ca_cert_group ca_certkey2
<!--NeedCopy-->
```

```
sh ssl caCertGroup ca_cert_group

CA GROUP NAME:     ca_cert_group
ACTIONS REFERRING: 1

1) CertKey Name: ca_certkey1   CA Certificate   CRLCheck: Optional
   CA_Name Sent
2) CertKey Name: ca_certkey2   CA Certificate   CRLCheck: Optional
   CA_Name Sent
<!--NeedCopy-->
```

```
add ssl action pick_ca_group -cacertGrpName ca_cert_group
<!--NeedCopy-->
```

```
sh ssl action pick_ca_group
1) Name: pick_ca_group
   Type: Data Insertion
   PickCaCertGroup: ca_cert_group
   Hits: 0
   Undef Hits: 0
   Action Reference Count: 1
<!--NeedCopy-->
```

```
add ssl policy snipolicy -rule client.ssl.client_hello.sni.contains("
   abc") -action pick_ca_group
bind ssl vserver v_SSL -policyName snipolicy -type CLIENTHELLO_REQ -
   priority 10
<!--NeedCopy-->
```

```
sh ssl policy snipolicy
    Name: snipolicy
    Rule: client.ssl.client_hello.sni.contains("abc")
    Action: pick_ca_group
    UndefAction: Use Global
    Hits: 0
    Undef Hits: 0


    Policy is bound to following entities
1)  Bound to: CLIENTHELLO_REQ VSERVER v_SSL
```



```
    Priority: 10
<!--NeedCopy-->
```

```
set ssl vserver v_SSL -clientauth ENABLED -SNIEnable ENABLED
bind ssl vserver v_SSL -certkeyName snicert -sniCert
<!--NeedCopy-->
```

```
sh ssl vserver v_SSL

    Advanced SSL configuration for VServer v_SSL:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED   Ephemeral RSA:
       ENABLED   Refresh Count: 0
    Session Reuse: ENABLED   Timeout: 120 seconds
    Cipher Redirect: DISABLED
    SSLv2 Redirect: DISABLED
    ClearText Port: 0
    Client Auth: ENABLED   Client Cert Required: Mandatory
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: ENABLED
    OCSP Stapling: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0
    SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED
         TLSv1.2: ENABLED  TLSv1.3: DISABLED
    Push Encryption Trigger: Always
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1

    ECC Curve: P_256, P_384, P_224, P_521

1)  CertKey Name: snicert   Server Certificate for SNI


    Data policy
1)  Policy Name: snipolicy  Priority: 10


1)  Cipher Name: DEFAULT
    Description: Default cipher list with encryption strength >= 128bit
<!--NeedCopy-->
```

GUI を使用した設定

CA 証明書グループを作成し、証明書をグループにバインドします。

1. トラフィック管理 **> SSL > CA** 証明書グループに移動します。
2. [追加] をクリックし、グループの名前を指定します。
3. ［作成］をクリックします。
4. **CA** 証明書グループを選択し、「バインディングを表示」をクリックします。
5. ［**Bind**］をクリックします。
6. **CA** 証明書バインディングページで、既存の証明書を選択するか、[追加] をクリックして新しい証明書を追加します。
7. [選択] をクリックし、[バインド] をクリックします。
8. 別の証明書をバインドするには、手順 5 ～7 を繰り返します。
9. ［閉じる］をクリックします。

［**Traffic Management**］**>** ［**SSL**］**>** ［**Policies**］に移動します。

SSL アクションの作成:

1.「**SSL** アクション」で、「追加」をクリックします。
2.「**SSL** アクションの作成」で、アクションの名前を指定します。
3.「転送アクション仮想サーバー」で、既存の仮想サーバーを選択するか、トラフィックを転送する仮想サーバーを追加します。
4. オプションで、他のパラメータを設定します。
5. ［作成］をクリックします。

SSL ポリシーの作成:

1. **SSL** ポリシーで、「追加」をクリックします。
2.「**SSL** ポリシーの作成」で、ポリシーの名前を指定します。
3.「アクション」で、以前に作成したアクションを選択します。
4. エクスプレッションエディタで、評価するルールを入力します。
5. ［作成］をクリックします。

仮想サーバーとバインドポリシーを作成または追加します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバーを追加または選択します。
3. [詳細設定] で、[**SSL** ポリシー] をクリックします。
4. SSL ポリシーセクションをクリックします。
5.「ポリシーの選択」で、以前に作成したポリシーを選択します。
6.「ポリシー・バインディング」で、ポリシーの優先度を指定します。
7.「タイプ」で「**CLIENTHELLO_REQ**」を選択します。
8. ［**Bind**］をクリックします。
9. ［完了］をクリックします。

GUI を使用して **CA** 証明書グループをバインド解除する

1. トラフィック管理 **> SSL > CA** 証明書グループに移動します。
2. 証明書グループを選択し、「バインディングを表示」をクリックします。
3. グループから削除する証明書を選択し、[バインド解除] をクリックします。
4. 確認を求められたら、「** はい」をクリックします••.
5. [閉じる] をクリックします。

GUI を使用して **CA** 証明書グループを削除する

1. トラフィック管理 **> SSL > CA** 証明書グループに移動します。
2. 証明書グループを選択し、[削除] をクリックします。
3. 確認を求められたら、「はい」をクリックします。

SSL ポリシーのバインディング

August 15, 2023

SSL ポリシーは、グローバルにバインドすることも、SSL タイプの仮想サーバーのみにバインドすることもできます。グローバルにバインドされたポリシーは、サービス、仮想サーバー、またはその他の NetScaler バインドポイントにバインドされたすべてのポリシーが評価された後に評価されます。受信データが SSL ポリシーで設定されたルールのいずれかに一致すると、ポリシーがトリガーされ、それに関連するアクションが実行されます。

SSL ポリシーを仮想サーバーにバインドする場合、次のバインドポイントのいずれかを選択する必要があります。

- REQUEST (デフォルトのバインドポイント)。ポリシー評価は SSL ハンドシェイクの完了後に HTTP レイヤーで行われます。)
- INTERCEPT_REQ_REQ（このオプションは、Citrix Secure Web Gateway のセットアップに適用されます。詳細については、「SSL インターセプション用の SSL ポリシーインフラストラクチャ」を参照してください。
- CLIENTHELLO.REQ

同様に、仮想サーバーからポリシーをバインド解除する場合は、バインドポイントを指定する必要があります。

CLIENTHELLO_REQ をバインドポイントとして指定すると、クライアントの hello メッセージを受信したときにポリシーが評価されます。実行できるアクションは、リセット、転送、`caCertGrpName`およびです。リセットアクションは接続を終了します。転送アクションは、要求を負荷分散仮想サーバーに転送して処理します。`caCertGrpName` アクションは、クライアント認証のために SNI に基づいて CA を選択的に選択します。SSL アクションの詳細については、SSL 組み込みアクションとユーザー定義アクションを参照してください。

注：アクション caCertGrpName は、TLS1.3 プロトコルではサポートされていません。

CLI を使用して SSL ポリシーをグローバルにバインドする

コマンドプロンプトで次のコマンドを入力してグローバル SSL ポリシーをバインドし、構成を確認します。

```
bind ssl global - policyName <string> [- priority <positive_integer>]
show ssl global
<!--NeedCopy-->
```

例:

```
bind ssl global -policyName Policy-SSL-2 -priority 90
Done

sh ssl global

      1) Name: Policy-SSL-2 Priority: 90
      2) Name: Policy-SSL-1 Priority: 100
      Done
<!--NeedCopy-->
```

GUI を使用して SSL ポリシーをグローバルにバインドする

1. **Traffic Management** > **SSL** > **Policies** に移動します。
2. 詳細ウィンドウで、[グローバルバインド] をクリックします。
3. 「SSL ポリシーをグローバルにバインド/バインド解除」ダイアログボックスで、「ポリシーを挿入」をクリックします。
4. 「ポリシー名」リストで、ポリシーを選択します。
5. オプションとして、エントリをポリシーバンクの新しい位置にドラッグして、優先度レベルを自動的に更新することもできます。
6. [**OK**] をクリックします。ステータスバーに、ポリシーが正常にバインドされたことを示すメッセージが表示されます。

CLI を使用して SSL ポリシーを仮想サーバーにバインドまたはバインド解除する

コマンドプロンプトで次のコマンドを入力して SSL ポリシーを仮想サーバーにバインドし、構成を確認します。

```
bind ssl vserver <vServerName> -policyName <string> -priority <
    positive_integer> -type <type>

unbind ssl vserver <vServerName> -policyName <string> -priority <
    positive_integer> -type <type>

<!--NeedCopy-->
```

例:


```
bind ssl vserver v1 -policyName pol1 -priority 1 -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

```
unbind ssl vserver v1 -policyName pol1 -priority 1 -type
   CLIENTHELLO_REQ
<!--NeedCopy-->
```

```
show ssl vserver vs-server

Advanced SSL configuration for VServer vs-server:

DH: DISABLED

Ephemeral RSA: ENABLED          Refresh Count: 1000

Session Reuse: ENABLED          Timeout: 120 seconds

Cipher Redirect: DISABLED

SSLv2 Redirect: DISABLED

ClearText Port: 80

Client Auth: DISABLED

SSL Redirect: ENABLED

SSL-REDIRECT Port Rewrite: ENABLED

Non FIPS Ciphers: DISABLED

SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      Policy Name: ssl-policy-1        Priority: 10

1)      Cipher Name: DEFAULT

            Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

GUI を使用して **SSL** ポリシーを仮想サーバーにバインドする

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、SSL 仮想サーバーを開きます。
2. [詳細設定] で [**SSL** ポリシー] を選択します。**SSL** ポリシーセクションをクリックして、ポリシーを仮想サーバーにバインドします。
3. ポリシーバインディングページで、既存のポリシーを選択するか、新しいポリシーを追加します。
4. ポリシーの優先度とタイプ (バインドポイント) を指定します。

5.「バインド」を選択します。
6.［完了］を選択します。

SSL ポリシーのラベル

August 15, 2023

ポリシーラベルはポリシーのホルダーです。ポリシーラベルは、別のポリシーから呼び出すことができるポリシーバンクと呼ばれるポリシーのグループを管理するのに役立ちます。SSL ポリシーラベルは、ポリシーラベルに含まれるポリシーの種類に応じて、コントロールラベルでもデータラベルでもかまいません。データポリシーラベルにはデータポリシーのみを追加でき、制御ポリシーラベルには制御ポリシーのみを追加できます。ポリシーバンクを作成するには、ポリシーをラベルにバインドし、ポリシーラベルのポリシーバンク内の他のポリシーと比較して各ポリシーの評価順序を指定します。CLI では、2 つのコマンドを入力してポリシーラベルを作成し、ポリシーをポリシーラベルにバインドします。設定ユーティリティでは、ダイアログボックスからオプションを選択します。

注**:** タイプコントロールのポリシーラベルは TLS 1.3 プロトコルではサポートされていません。

CLI を使用して SSL ポリシーラベルを作成し、ポリシーをラベルにバインドします

コマンドプロンプトで入力します。

```
add ssl policylabel <labelName> -type ( CONTROL | DATA )

bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>) ]
<!--NeedCopy-->
```

例**:**

```
add ssl policylabel cpl1 -type CONTROL
add ssl policylabel dpl1 -type DATA

add ssl action act1 -clientauth DOCLIENTAUTH
add ssl policy ctrlpol -rule HTTP.REQ.METHOD.EQ("GET") -action act1

add ssl action act2 -clientCert ENABLED -certHeader "X-Client-Cert"
add ssl policy datapol -rule CLIENT.SSL.CLIENT_CERT.EXISTS -action act2

bind ssl policylabel cpl1 ctrlpol 1
bind ssl policylabel dpl1 datapol 1

> sh ssl policylabel
Control policylabels
1)    Label Name: cpl1
    Type: CONTROL
```



```
     Number of bound policies: 1
     Number of times invoked: 0

Data policylabels
1)  Label Name: dpl1
     Type: DATA
     Number of bound policies: 1
     Number of times invoked: 0
Done
>
<!--NeedCopy-->
```

GUI を使用して **SSL** ポリシーラベルを設定し、ポリシーをラベルにバインドします

[トラフィック管理] **> [SSL] > [**ポリシーラベル] に移動し、SSL ポリシーラベルを設定します。

選択的な **SSL** ログ

August 15, 2023

数千台の仮想サーバーで構成される大規模な導入環境では、SSL に関連するすべての情報がログに記録されます。以前は、いくつかの重要な仮想サーバーについて、クライアント認証と SSL ハンドシェイクの成功と失敗をフィルタリングすることは容易ではありませんでした。この情報を取得するためにログ全体を調べるのは時間がかかり、面倒な作業でした。インフラストラクチャではログをフィルタリングする制御が提供されていなかったためです。これで、`ns.log`の特定の仮想サーバまたは仮想サーバのグループの SSL 関連情報を記録できます。この情報は、障害をデバッグする際に特に役立ちます。

DEBUG 設定では、SSL 関連のすべての情報が`ns.log`にログインされます。ただし、SSL ログプロファイルを設定すると、クライアント認証と SSL ハンドシェイクに関連する情報のみが記録されます。この情報を記録するには、以下の手順を実行します。

1. syslog パラメータに DEBUG を設定します。
2. SSL ログプロファイルを設定します。クライアント認証と SSL ハンドシェイクの失敗/成功と失敗のみのロギングを有効にします。SSL ログプロファイルを SSL プロファイルに添付すると、4 つすべてが記録されます。SSL アクションで SSL ログプロファイルを添付すると、クライアント認証の失敗/成功と失敗のみが記録されます。
3. SSL ログプロファイルを SSL プロファイルまたは SSL アクションに添付します。

このページの最後にある、クライアント認証が成功した場合の ns.log 出力例を参照してください。

デバッグレベルの設定

Syslog ログレベルを DEBUG に設定します。コマンドプロンプトで入力します。

```
set audit syslogParams -logLevel DEBUG
```

デバッグを設定すると、フロントエンド (仮想サーバー) とバックエンド (サービスとサービスグループ) の両方の SSL ログが含まれます。ただし、選択的 SSL ロギングでは、フロントエンドのみを制御できます。

SSL ログプロファイル

SSL ログプロファイルは、仮想サーバーまたは仮想サーバーのグループに関する次のイベントのログ記録を制御します。

- クライアント認証の成功と失敗、または失敗のみ。
- SSL ハンドシェイクの成功と失敗、または失敗のみ。

デフォルトでは、すべてのパラメータは無効になっています。

SSL ログプロファイルは、SSL プロファイルまたは SSL アクションに設定できます。SSL プロファイルに設定すると、クライアント認証と SSL ハンドシェイクの成功と失敗の両方の情報をログに記録できます。SSL アクションに設定すると、ポリシーが評価される前にハンドシェイクが完了するため、クライアント認証の成功と失敗の情報のみをログに記録できます。

SSL ログプロファイルを設定しなくても、クライアント認証と SSL ハンドシェイクの成功と失敗が記録されます。ただし、選択的ロギングは SSL ログプロファイルが使用されている場合にのみ可能です。

> 注:
>
> SSL ログプロファイルは、高可用性とクラスタセットアップでサポートされています。

CLI を使用した SSL ログプロファイルの追加

コマンドプロンプトで入力します。

```
1 add ssl logprofile <name> [-sslLogClAuth ( ENABLED | DISABLED )] [-
      ssllogClAuthFailures ( ENABLED | DISABLED )] [-sslLogHS ( ENABLED |
      DISABLED )] [-sslLogHSfailures ( ENABLED | DISABLED )]
2 <!--NeedCopy-->
```

パラメーター:

名前:

SSL ログプロファイルの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。プロファイルの作成後は変更できません。

Name は必須の引数です。最大長：127

sslLogClAuth:

すべてのクライアント認証イベントをログに記録します。成功イベントと失敗イベントの両方が含まれます。

設定可能な値：ENABLED, DISABLED

デフォルト値: 無効

ssllogClAuthFailures:

すべてのクライアント認証失敗イベントをログに記録します。

設定可能な値：ENABLED, DISABLED

デフォルト値: 無効

sslLogHS:

SSL ハンドシェイクに関連するすべてのイベントをログに記録します。成功イベントと失敗イベントの両方が含まれます。

設定可能な値：ENABLED, DISABLED

デフォルト値: 無効

sslLogHSfailures:

SSL ハンドシェイクに関連するすべての障害イベントをログに記録します。

設定可能な値：ENABLED, DISABLED

デフォルト値: 無効

例**:**

```
> add ssl logprofile ssllog10 -sslLogClAuth ENABLED -sslLogHS ENABLED

 Done

sh ssllogprofile ssllog10

1)      Name: ssllog10

        SSL log ClientAuth [Success/Failures] : ENABLED

        SSL log ClientAuth [Failures] : DISABLED

        SSL log Handshake [Success/Failures] : ENABLED

        SSL log Handshake [Failures] : DISABLED

 Done
<!--NeedCopy-->
```


GUI を使用した **SSL** ログプロファイルの追加

[システム] **>** [プロファイル] **>** [**SSL** ログプロファイル] に移動し、プロファイルを追加します。

CLI を使用して **SSL** ログプロファイルを変更する

コマンドプロンプトで次のように入力します。

```
set ssl logprofile <name> [-sslLogClAuth ( ENABLED | DISABLED )][-
    ssllogClAuthFailures ( ENABLED | DISABLED )] [-sslLogHS ( ENABLED |
    DISABLED )] [-sslLogHSfailures ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

例**:**

```
set ssllogprofile ssllog10 -ssllogClAuth en -ssllogClAuthFailures en -
    ssllogHS en -ssllogHSfailures en

Done

sh ssllogprofile ssllog10

    1)            Name: ssllog10

                    SSL log ClientAuth [Success/Failures] : ENABLED
                    SSL log ClientAuth [Failures] : ENABLED
                    SSL log Handshake [Success/Failures] : ENABLED
                    SSL log Handshake [Failures] : ENABLED
     Done
<!--NeedCopy-->
```

GUI を使用して **SSL** ログプロファイルを変更する

1. [システム] **>** [プロファイル] **>** [**SSL** ログプロファイル] に移動し、プロファイルを選択して [編集] をクリックします。
2. 変更を加えて「**OK**」をクリックします。

CLI を使用してすべての **SSL** ログプロファイルを表示する

コマンドプロンプトで入力します。

```
sh ssl logprofile
<!--NeedCopy-->
```

例**:**


```
sh ssl logprofile

    1)            Name: ssllogp1
                    SSL log ClientAuth [Success/Failures] : ENABLED
                    SSL log ClientAuth [Failures] : ENABLED
                    SSL log Handshake [Success/Failures] : DISABLED
                    SSL log Handshake [Failures] : ENABLED

    2)            Name: ssllogp2
                    SSL log ClientAuth [Success/Failures] : DISABLED
                    SSL log ClientAuth [Failures] : DISABLED
                    SSL log Handshake [Success/Failures] : DISABLED
                    SSL log Handshake [Failures] : DISABLED

    3)            Name: ssllogp3
                    SSL log ClientAuth [Success/Failures] : DISABLED
                    SSL log ClientAuth [Failures] : DISABLED
                    SSL log Handshake [Success/Failures] : DISABLED
                    SSL log Handshake [Failures] : DISABLED

    4)            Name: ssllog10
                    SSL log ClientAuth [Success/Failures] : ENABLED
                    SSL log ClientAuth [Failures] : ENABLED
                    SSL log Handshake [Success/Failures] : ENABLED
                    SSL log Handshake [Failures] : ENABLED
Done
<!--NeedCopy-->
```

GUI を使用してすべての **SSL** ログプロファイルを表示する

システム **>** プロファイル **> SSL** ログプロファイルに移動します。すべてのプロファイルが一覧表示されます。

SSL ログプロファイルを **SSL** プロファイルにアタッチする

SSL プロファイルの作成時に SSL ログプロファイルを SSL プロファイルに添付 (設定) したり、後で SSL プロファイルを編集したりすることができます。クライアント認証とハンドシェイクの成功と失敗の両方を記録できます。

> 重要:
>
> SSL ログプロファイルをアタッチする前に、デフォルトの SSL プロファイルを有効にする必要があります。デフォルトの SSL プロファイルの有効化について詳しくは、「デフォルトプロファイルの有効化」を参照してください。

CLI を使用して **SSL** ログプロファイルを **SSL** プロファイルにアタッチする

コマンドプロンプトで入力します。


```
set ssl profile <name> [-ssllogProfile <string>]
<!--NeedCopy-->
```

例:

```
set ssl profile fron_1 -ssllogProfile ssllog10
<!--NeedCopy-->
```

GUI を使用して **SSL** ログプロファイルを **SSL** プロファイルにアタッチする

1. **[**システム**] > [**プロファイル**] > [SSL** プロファイル**]** に移動します。
2. [編集] をクリックし、[**SSL** ログプロファイル] でプロファイルを指定します。

SSL ログプロファイルを SSL アクションにアタッチする

SSL ログプロファイルを設定できるのは、SSL アクションの作成時だけです。SSL アクションを変更してログプロファイルを設定することはできません。アクションをポリシーに関連付けます。ログに記録できるのは、クライアント認証の成功と失敗だけです。

CLI を使用して **SSL** ログプロファイルを **SSL** アクションにアタッチする

コマンドプロンプトで入力します。

```
add ssl action <name> -clientAuth ( DOCLIENTAUTH | NOCLIENTAUTH ) -
    ssllogProfile <string>
<!--NeedCopy-->
```

例:

```
> add ssl action act1 -clientAuth DoCLIENTAUTH -ssllogProfile ssllog10

Done

> sh ssl action act1

    1)            Name: act1
                    Type: Client Authentication (DOCLIENTAUTH)
                    Hits: 0
                    Undef Hits: 0
                    Action Reference Count: 0
                    SSLlogProfile: ssllog10
Done
<!--NeedCopy-->
```


GUI を使用して **SSL** ログプロファイルを **SSL** アクションにアタッチする

1. トラフィック管理 **> SSL >** ポリシーに移動し、[**SSL** アクション] をクリックします。
2. [追加] をクリックします。
3. [クライアント認証] で [有効] を選択します
4. [SSL ログプロファイル] で、リストからプロファイルを選択するか、[+] をクリックしてプロファイルを作成します。
5. [作成] をクリックします。

ログファイルからの出力例

ns.log クライアント認証が成功した場合のからのログ出力例を次に示します。

```
1 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
     PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 158 0 :  SPCBId 671 -
     ClientIP 10.102.1.98 - ClientPort 49451 - VserverServiceIP
     10.102.57.82 - VserverServicePort 443 - ClientVersion TLSv1.2 -
     CipherSuite "AES-256-CBC-SHA TLSv1.2 Non-Export 256-bit" - Session
     New - CLIENT_AUTHENTICATED -SerialNumber "2A" - SignatureAlgorithm "
     sha1WithRSAEncryption" - ValidFrom "Sep 22 09:15:20 2008 GMT" -
     ValidTo "Feb  8 09:15:20 2036 GMT" - HandshakeTime 10 ms
2 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
     PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 159 0 :  SPCBId 671
     - IssuerName " C=IN,ST=KAR,O=Citrix R&D Pvt Ltd,CN=Citrix"
3 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
     PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 160 0 :  SPCBId 671
      - SubjectName " C=IN,ST=KAR,O=Citrix Pvt Ltd,OU=A,CN=B"
4 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
     PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 161 0 :  Backend SPCBId
      674 - ServerIP 10.102.57.85 - ServerPort 443 - ProtocolVersion
     TLSv1.2 - CipherSuite "AES-256-CBC-SHA TLSv1.2 Non-Export 256-bit" -
      Session Reuse - SERVER_AUTHENTICATED -SerialNumber "3E" -
     SignatureAlgorithm "sha1WithRSAEncryption" - ValidFrom "Sep 24
     06:40:37 2008 GMT" - ValidTo "Feb 10 06:40:37 2036 GMT" -
     HandshakeTime 1 ms
5 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
     PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 162 0 :  SPCBId 674
     - IssuerName " C=IN,ST=KAR,O=Citrix Pvt Ltd"
6 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
     PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 163 0 :  SPCBId 674
      - SubjectName " C=IN,ST=P,L=Q,O=R"
7 <!--NeedCopy-->
```


DTLS プロトコルのサポート

April 15, 2024

> メモ
>
> - DTLS 1.2 プロトコルは次のアプライアンスでサポートされています。
>   - NetScaler MPX/SDX (N2 and N3 based) and VPX appliances. It is not supported on external HSMs.
>   - NetScaler appliances containing Intel Coleto and Intel Lewisburg SSL chips.
>   - Front-end of NetScaler VPX appliances.
>   - Front-end of NetScaler appliances containing Intel Coleto SSL chips. For more information about the platforms containing Intel Coleto SSL chips, see Support for Intel Coleto SSL chip-based platforms.
>   - Front-end of NetScaler MPX (N3 based) appliances except the MPX 14000 FIPS appliances.
> - DTLS タイプのサービスグループはサポートされません。
> - NetScaler Gateway の Enlightened Data Transport（EDT）サポートの詳細については、「HDX 啓発データトランスポートのサポート」を参照してください。
> - サポートされているプラットフォームとビルドについては、「NetScaler MPX ハードウェアとソフトウェアの互換性マトリックス」を参照してください。

SSL プロトコルと TLS プロトコルは、従来、ストリーミングトラフィックの保護に使用されてきました。これらのプロトコルはどちらも TCP をベースにしており、これは低速です。また、TLS は、損失または並べ替えられたパケットを処理できません。

UDP は、Lync、Skype、iTunes、YouTube、トレーニングビデオ、フラッシュなどのオーディオおよびビデオアプリケーションに適したプロトコルです。ただし、UDP は安全でも信頼性もありません。DTLS プロトコルは、UDP を介してデータを保護するように設計されており、メディアストリーミング、VOIP、通信用のオンラインゲームなどのアプリケーションに使用されます。DTLS では、各ハンドシェイクメッセージには、そのハンドシェイク内の特定のシーケンス番号が割り当てられます。ピアは、ハンドシェイクメッセージを受信すると、そのメッセージが次に予想されるメッセージであるかどうかを迅速に判断できます。そうであれば、ピアはメッセージを処理します。そうでない場合、メッセージは以前のすべてのメッセージを受信した後に処理のためにキューに入れられます。

DTLS 仮想サーバーと UDP タイプのサービスを作成します。デフォルトでは、DTLS プロファイル (ns-dtls_default_profile) は仮想サーバーにバインドされます。オプションで、ユーザー定義の DTLS プロファイルを作成し、仮想サーバーにバインドできます。

注:RC4 暗号は DTLS 仮想サーバーではサポートされていません。

DTLS 構成

コマンドライン（CLI）または構成ユーティリティ（GUI）を使用して、ADC アプライアンスで DTLS を設定できます。

注：DTLS 1.2 プロトコルは NetScaler VPX アプライアンスのフロントエンドでサポートされています。DTLSv1.2 仮想サーバーを構成するときに、DTLS12 を指定します。デフォルトは DTLS1 です。

コマンドプロンプトで入力します。

```
set ssl vserver DTLS [-dtls1 ( ENABLED | DISABLED )] [-dtls12 (
ENABLED | DISABLED )]
```

CLI を使用して DTLS 設定を作成する

コマンドプロンプトで入力します。

```
1 add lb vserver <vserver_name> DTLS <IPAddress>  <port>
2 add service  <service_name>  <IPAddress> UDP 443
3 bind lb vserver  <vserver_name>  <udp_service_name>
4 <!--NeedCopy-->
```

次の手順はオプションです。

```
1 add dtlsProfile dtls-profile -maxretryTime <positive_integer>
2 set ssl vserver <vserver_name> -dtlsProfileName <dtls_profile_name>
3 <!--NeedCopy-->
```

GUI を使用して DTLS 設定を作成する

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. DTLS タイプの仮想サーバーを作成し、UDP サービスを仮想サーバーにバインドします。
3. デフォルトの DTLS プロファイルは DTLS 仮想サーバーにバインドされます。別のプロファイルをバインドするには、[SSL パラメータ] で別の DTLS プロファイルを選択します。プロファイルを作成するには、[DTLS プロファイル] の横のプラス (+) をクリックします。

DTLS 仮想サーバでの SNI のサポート

SNI の詳細については、「複数のサイトのセキュアなホスティングのための SNI 仮想サーバーを構成する」を参照してください。

CLI を使用して DTLS 仮想サーバで SNI を構成する

コマンドプロンプトで入力します。


```
set ssl vserver <vServerName> -SNIEnable ENABLED
bind ssl vserver <vServerName> -certkeyName <string> -SNICert
show ssl vserver <vServerName>
<!--NeedCopy-->
```

例:

```
set ssl vserver v1 -sniEnable ENABLED
bind ssl vserver  v1 -certkeyName san2 -sniCert
bind ssl vserver  v1 -certkeyName san13  - sniCert
bind ssl vserver  v1 -certkeyName san17  - sniCert
<!--NeedCopy-->
```

```
sh ssl vserver v1

Advanced SSL configuration for VServer v1:
DH: DISABLED
DH Private-Key Exponent Size Limit: DISABLED
Ephemeral RSA: ENABLED
Refresh Count: 0
Session Reuse: ENABLED
Timeout: 1800 seconds
Cipher Redirect: DISABLED

ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: ENABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
DTLSv1: ENABLED
Send Close-Notify: YES
Strict Sig-Digest Check: DISABLED
Zero RTT Early Data: DISABLED
DHE Key Exchange With PSK: NO
Tickets Per Authentication Context: 1

DTLS profile name: nsdtls_default_profile

ECC Curve: P_256, P_384, P_224, P_521

1) CertKey Name: ca
CA Certificate  OCSPCheck: OptionalCA_Name Sent
2) CertKey Name: san2 Server Certificate for SNI
3) CertKey Name: san17 Server Certificate for SNI
4) CertKey Name: san13 Server Certificate for SNI


1) Cipher Name: DEFAULT
```



```
Description: Default cipher list with encryption strength >= 128bit
Done
<!--NeedCopy-->
```

GUI を使用した **DTLS** 仮想サーバでの **SNI** の設定

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. DTLS 仮想サーバーを開き、[証明書] で [サーバー証明書] をクリックします。
3. 証明書を追加するか、リストから証明書を選択して、[**SNI** のサーバー証明書] を選択します。
4. [詳細設定] で、[**SSL** パラメータ] をクリックします。
5. [**SNI** 有効] を選択します。

DTLS 仮想サーバーでサポートされていない機能

次のオプションは、DTLS 仮想サーバーでは有効にできません。

- SSLv2
- SSLv3
- TLSv1
- TLSv1.1
- TLSv1.2
- プッシュ暗号化トリガー
- SSLv2Redirect
- sslv2URL

DTLS 仮想サーバーによって使用されないパラメータ

DTLS 仮想サーバーは、設定されている場合でも、次の SSL パラメータを無視します。

- 暗号化トリガーパケット数
- プッシュ暗号化トリガータイムアウト
- SSL 量子サイズ
- 暗号化トリガータイムアウト
- 件名/発行者名の挿入形式

DTLS サービスでの再ネゴシエーションの設定

非セキュア再ネゴシエーションは、DTLS サービスでサポートされています。CLI または GUI を使用して、この設定を構成できます。

CLI を使用した **DTLS** サービスでの再ネゴシエーションの設定

コマンドプロンプトで入力します。

```
set ssl parameter -denysslreneg NONSECURE
<!--NeedCopy-->
```

例**:**

```
set ssl parameter -denysslreneg NONSECURE


sh ssl parameter
Advanced SSL Parameters
-----------------------
SSL quantum size                                    : 8 KB
Max CRL memory size                                 : 256 MB
Strict CA checks                                    : NO
Encryption trigger timeout                          : 100 ms
Send Close-Notify                                   : YES
Encryption trigger packet count                     : 45
Deny SSL Renegotiation                              : NONSECURE
Subject/Issuer Name Insertion Format                : Unicode
OCSP cache size                                     : 10 MB
Push flag                                           : 0x0 (Auto)
Strict Host Header check for SNI enabled SSL sessions : NO
PUSH encryption trigger timeout                     : 1 ms
Crypto Device Disable Limit                         : 0
Global undef action for control policies            : CLIENTAUTH
Global undef action for data policies               : NOOP
Default profile                                     : DISABLED
SSL Insert Space in Certificate Header              : YES
Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors  : NO
Disable TLS 1.1/1.2 for dynamic and VPN services    : NO
Software Crypto acceleration CPU Threshold          : 0
Hybrid FIPS Mode                                    : DISABLED
Signature and Hash Algorithms supported by TLS1.2   : ALL
SSL Interception Error Learning and Caching         : DISABLED
SSL Interception Maximum Error Cache Memory         : 0 Bytes
Done
<!--NeedCopy-->
```

GUI を使用した **DTLS** サービスでの再ネゴシエーションの設定

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
2. DTLS サービスを選択し、[編集（Edit）] をクリックします。
3. [**SSL**] > [詳細設定**]** に移動します。
4. [**SSL** 再ネゴシエーションを拒否する] を選択します。

DTLS サービスでサポートされていない機能

次のオプションは、DTLS サービスでは有効にできません。

- SSLv2
- SSLv3
- TLSv1
- TLSv1.1
- TLSv1.2
- プッシュ暗号化トリガー
- SSLv2Redirect
- sslv2URL
- SNI
- 安全な再ネゴシエーション

DTLS サービスによって使用されないパラメータ

DTLS サービスは、設定されている場合でも、次の SSL パラメータを無視します。

- 暗号化トリガーパケット数
- プッシュ暗号化トリガータイムアウト
- SSL 量子サイズ
- 暗号化トリガータイムアウト
- 件名/発行者名の挿入形式

> 注:
>
> DTLS サービスでセッションの再利用が現在サポートされていないため、SSL セッション再利用ハンドシェイクは DTLS サービスで失敗します。
>
> 回避策: DTLS サービスでセッションの再利用を手動で無効にします。CLI で、次のように入力します。
>
> ```
> set ssl service <dtls-service-name> -sessReuse DISABLED
> ```

DTLS プロファイル

デフォルト設定の DTLS プロファイルは、自動的に DTLS 仮想サーバーにバインドされます。ただし、要件に合わせて特定の設定で DTLS プロファイルを作成できます。

DTLS 仮想サーバーまたは VPN DTLS 仮想サーバーで DTLS プロファイルを使用します。DTLS 仮想サーバでは SSL プロファイルを使用できません。

> 注:
>
> MTU とパケットサイズの変更に基づいて、DTLS プロファイルの最大レコードサイズ設定を変更してください。たとえば、デフォルトの最大レコードサイズ 1459 バイトは、IPv4 アドレスヘッダーサイズに基づいて計算されます。IPv6 レコードでは、ヘッダーサイズが大きくなります。したがって、次の基準を満たすには、最大レコードサイズを小さくする必要があります。

```
max record size + UDP header(8bytes)+ IP header size < MTU
```

例:

```
Default DTLS profile
    1) Name: nsdtls_default_profile
    PMTU Discovery: DISABLED
    Max Record Size: 1459 bytes
    Max Retry Time: 3 sec
    Hello Verify Request: ENABLED
    Terminate Session: DISABLED
    Max Packet Count: 120 bytes

Custom DTLS profile
    1) Name: ns_dtls_profile_ipv6_1
    PMTU Discovery: DISABLED
    Max Record Size: 1450 bytes
    Max Retry Time: 3 sec
    Hello Verify Request: ENABLED
    Terminate Session: DISABLED
    Max Packet Count: 120 bytes
<!--NeedCopy-->
```

CLI を使用した **DTLS** プロファイルの作成

> 注:
>
> - helloverifyrequestこのパラメーターはデフォルトで有効になっています。このパラメータを有効にすると、攻撃者またはボットがネットワークのスループットを圧倒するリスクを軽減し、アウトバウンド帯域幅の枯渇につながる可能性があります。つまり、DTLS DDoS 増幅攻撃を軽減するのに役立ちます。
> - maxHoldQlenパラメータが追加されます。このパラメーターは、DTLS レイヤーで処理するためにキューに入れられるデータグラムの数を定義します。UDP 多重化が高い UDP トラフィックを送信している場合、maxHoldQlen パラメータの値が大きいと、DTLS レイヤでメモリが蓄積する可能性があります。したがって、低い値を設定することをお勧めします。最小値は 32、最大値は 65535、デフォルト値は 32 です。

DTLS セッションで受信した不正な MAC レコードを無視するために、DTLS プロファイルに新しいmaxBadmacIgnorecountパラメータが導入されました。このパラメータを使用すると、パラメータに設定

された値までの不良レコードは無視されます。アプライアンスは、制限に達した後にのみ、セッションを終了し、アラートを送信します。

このパラメーター設定は、terminateSession パラメーターが有効になっている場合にのみ有効です。

```
ssl dtlsProfile <name> -maxRetryTime <positive_integer> -
    helloVerifyRequest ( ENABLED | DISABLED ) -terminateSession (ENABLED
     | DISABLED ) -maxHoldQLen <positive_integer> -maxBadmacIgnorecount
    <positive_integer>

helloVerifyRequest
               Send a Hello Verify request to validate the client.
               Possible values: ENABLED, DISABLED
               Default value: ENABLED

terminateSession
               Terminate the session if the message authentication code
                   (MAC)
               of the client and server do not match.
               Possible values: ENABLED, DISABLED
               Default value: DISABLED

maxHoldQLen
               Maximum number of datagrams that can be queued at DTLS
                   layer for
               processing
               Default value: 32
               Minimum value: 32
               Maximum value: 65535

maxBadmacIgnorecount
               Maximum number of bad MAC errors to ignore for a
                   connection prior disconnect. Disabling parameter
                   terminateSession
terminates session immediately when bad MAC is detected in the
    connection.
               Default value: 100
               Minimum value: 1
               Maximum value: 65535
<!--NeedCopy-->
```

例:

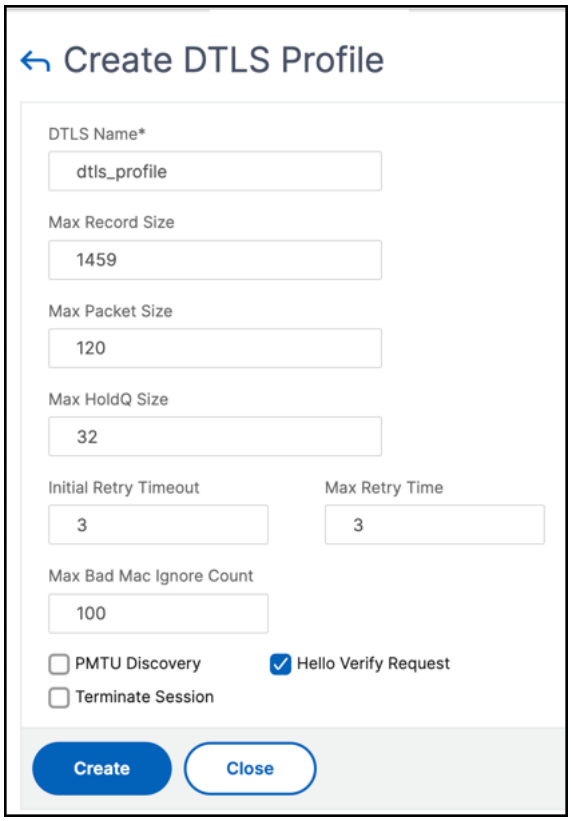
```
> add ssl dtlsprofile dtls_profile -maxRetryTime 4 -helloVerifyRequest
    ENABLED -terminateSession ENABLED -maxHoldQLen 40 -
    maxBadmacIgnorecount 150
 Done
> sh dtlsprofile dtls_profile
1)  Name: dtls_profile
    PMTU Discovery: DISABLED
    Max Record Size: 1459 bytes
    Max Retry Time: 4 sec
    Hello Verify Request: ENABLED
```



```
    Terminate Session: ENABLED
    Max Packet Count: 120 bytes
    Max HoldQ Size: 40 datagrams
    Max bad-MAC Ignore Count: 150

 Done
<!--NeedCopy-->
```

GUI を使用した **DTLS** プロファイルの作成

1. [システム] **>** [プロファイル] **>** [**DTLS** プロファイル] に移動し、[追加] をクリックします。
2. [**DTLS** プロファイルの作成] ページで、さまざまなパラメータの値を入力します。



3. [作成] をクリックします。

エンドツーエンド **DTLS** 設定の例

```
enable ns feature SSL LB

add server s1 198.51.100.2

en ns mode usnip

```



```
add service svc_dtls s1 DTLS 443

add lb vserver v1 DTLS 10.102.59.244 443

bind ssl vserver v1 -ciphername ALL

add ssl certkey servercert -cert servercert_aia_valid.pem -key
   serverkey_aia.pem

bind ssl vserver v1 -certkeyname servercert

bind lb vserver lb1 svc_dtls

sh lb vserver v1

                    v1 (10.102.59.244:4433) - DTLS     Type: ADDRESS
                    State: UP
                    Last state change was at Fri Apr 27 07:00:27 2018
                    Time since last state change: 0 days, 00:00:04.810
                    Effective State: UP
                    Client Idle Timeout: 120 sec
                    Down state flush: ENABLED
                    Disable Primary Vserver On Down : DISABLED
                    Appflow logging: ENABLED
                    No. of Bound Services :  1 (Total) 0 (Active)
                    Configured Method: LEASTCONNECTION
                    Current Method: Round Robin, Reason: A new service
                       is bound         BackupMethod: ROUNDROBIN
                    Mode: IP
                    Persistence: NONE
                    L2Conn: OFF
                    Skip Persistency: None
                    Listen Policy: NONE
                    IcmpResponse: PASSIVE
                    RHIstate: PASSIVE
                    New Service Startup Request Rate: 0 PER_SECOND,
                       Increment Interval: 0
                    Mac mode Retain Vlan: DISABLED
                    DBS_LB: DISABLED
                    Process Local: DISABLED
                    Traffic Domain: 0
                    TROFS Persistence honored: ENABLED
                    Retain Connections on Cluster: NO

    1) svc_dtls (10.102.59.190: 4433) - DTLS State: UP  Weight: 1
Done


sh ssl vserver v1

                    Advanced SSL configuration for VServer v1:
                    DH: DISABLED
                    DH Private-Key Exponent Size Limit: DISABLED
```



```
                                    Ephemeral RSA: ENABLED
                                                          Refresh Count: 0
                      Session Reuse: ENABLED                  Timeout:
                         1800 seconds
                      Cipher Redirect: DISABLED
                      ClearText Port: 0
                      Client Auth: DISABLED
                      SSL Redirect: DISABLED
                      Non FIPS Ciphers: DISABLED
                      SNI: DISABLED
                      OCSP Stapling: DISABLED
                      HSTS: DISABLED
                      HSTS IncludeSubDomains: NO
                      HSTS Max-Age: 0
                      DTLSv1: ENABLED
                      Send Close-Notify: YES
                      Strict Sig-Digest Check: DISABLED
                      Zero RTT Early Data: DISABLED
                      DHE Key Exchange With PSK: NO
                      Tickets Per Authentication Context: 1
                      DTLS profile name: nsdtls_default_profile

                      ECC Curve: P_256, P_384, P_224, P_521

    1)            CertKey Name: servercert               Server
       Certificate

    1)            Cipher Name: DEFAULT
                    Description: Default cipher list with encryption
                       strength >= 128bit

    2)            Cipher Name: ALL
                    Description: All ciphers supported by NetScaler,
                       excluding NULL ciphers
     Done

sh service svc_dtls

                      svc_dtls (10.102.59.190:4433) - DTLS
                      State: UP
                      Last state change was at Fri Apr 27 07:00:26 2018
                      Time since last state change: 0 days, 00:00:22.790
                      Server Name: s1
                      Server ID : None                  Monitor Threshold
                         : 0
                      Max Conn: 0        Max Req: 0           Max
                         Bandwidth: 0 kbits
                      Use Source IP: NO
                      Client Keepalive(CKA): NO
                      Access Down Service: NO
                      TCP Buffering(TCPB): NO
                      HTTP Compression(CMP): NO
                      Idle timeout: Client: 120 sec           Server: 120
```



```
                            sec
102                       Client IP: DISABLED
103                       Cacheable: NO
104                       SC: OFF
105                       SP: OFF
106                       Down state flush: ENABLED
107                       Monitor Connection Close : NONE
108                       Appflow logging: ENABLED
109                       Process Local: DISABLED
110                       Traffic Domain: 0
111
112      1)             Monitor Name: ping-default
113                                    State: UP                Weight: 1
                                                          Passive: 0
114                                  Probes: 5                Failed [Total
                                        : 0 Current: 0]
115                                  Last response: Success - ICMP echo
                                        reply received.
116                           Response Time: 2.77 millisec
117       Done
118
119  sh ssl service svc_dtls
120
121                       Advanced SSL configuration for Back-end SSL Service
                              svc_dtls:
122                       DH: DISABLED
123                       DH Private-Key Exponent Size Limit: DISABLED
                                    Ephemeral RSA: DISABLED
124                       Session Reuse: ENABLED                  Timeout:
                             1800 seconds
125                       Cipher Redirect: DISABLED
126                       ClearText Port: 0
127                       Server Auth: DISABLED
128                       SSL Redirect: DISABLED
129                       Non FIPS Ciphers: DISABLED
130                       SNI: DISABLED
131                       OCSP Stapling: DISABLED
132                       DTLSv1: ENABLED
133                       Send Close-Notify: YES
134                       Strict Sig-Digest Check: DISABLED
135                       Zero RTT Early Data: ???
136                       DHE Key Exchange With PSK: ???
137                       Tickets Per Authentication Context: ???
138                       DTLS profile name: nsdtls_default_profile
139                       ECC Curve: P_256, P_384, P_224, P_521
140      1)             Cipher Name: DEFAULT_BACKEND
141                       Description: Default cipher list for Backend SSL
                             session
142       Done
143
144
145  > sh dtlsProfile nsdtls_default_profile
146  1)  Name: nsdtls_default_profile
```



```
147      PMTU Discovery: DISABLED
148      Max Record Size: 1459 bytes
149      Max Retry Time: 3 sec
150      Hello Verify Request: DISABLED
151      Terminate Session: ENABLED
152      Max Packet Count: 120 bytes
153      Max HoldQ Size: 32 datagrams
154      Max bad-MAC Ignore Count: 10
155
156   Done
157  <!--NeedCopy-->
```

IPv6 アドレスの DTLS サポート

DTLS は IPv6 アドレスでもサポートされています。ただし、IPv6 アドレスで DTLS を使用するには、DTLS プロファイルで最大レコードサイズを調整する必要があります。

最大レコードサイズにデフォルト値を使用すると、初期 DTLS 接続が失敗することがあります。DTLS プロファイルを使用して最大レコードサイズを調整します。

DTLS 暗号サポート

デフォルトでは、DTLS 仮想サーバーまたはサービスの作成時に DTLS 暗号グループがバインドされます。DEFAULT_DTLS には、フロントエンド DTLS エンティティがサポートする暗号が含まれています。このグループは、DTLS 仮想サーバーの作成時にデフォルトでバインドされます。DEFAULT_DTLS_BACKEND には、バックエンド DTLS エンティティでサポートされる暗号が含まれています。このグループは、デフォルトでは DTLS バックエンドサービスにバインドされています。DTLS_FIPS には、NetScaler FIPS プラットフォームでサポートされている暗号が含まれています。このグループは、デフォルトでは、FIPS プラットフォームで作成された DTLS 仮想サーバーまたはサービスにバインドされます。

NetScaler VPX、MPX/SDX（N2 および N3 ベース）アプライアンスでの DTLS 暗号サポート

テーブルの読み方**:**

ビルド番号が指定されていない限り、暗号スイートはリリースのすべてのビルドでサポートされます。

例**:**

- **11.1**、**12.1**、**13.0**、**13.1**: 11.1、12.1、13.0、13.1 リリースのすべてのビルド。
- **-NA-**: 該当なし。

NetScaler VPX、MPX/SDX（N2、N3、および Coleto ベース）アプライアンスでの DTLS 暗号サポート

暗号スイート名	16 進コード	Wireshark 暗号スイート名	サポートされているビルド (フロントエンド)	サポートされているビルド (バックエンド)
TLS1-AES-256-CBC-SHA	0x0035	TLS_RSA_WITH_AES_256_CBC_SHA	12.1, 13.0, 13.1	12.1, 13.0, 13.1
TLS1-AES-128-CBC-SHA	0x002f	TLS_RSA_WITH_AES_128_CBC_SHA	12.1, 13.0, 13.1	12.1, 13.0, 13.1
SSL3-DES-CBC-SHA	0x0009	TLS_RSA_WITH_DES_CBC_SHA	13.0, 13.1	-なし-
SSL3-DES-CBC3-SHA	0x000a	TLS_RSA_WITH_3DES_EDE_CBC_SHA	12.1, 13.0, 13.1	12.1, 13.0, 13.1
SSL3-EDH-RSA-DES-CBC3-SHA	0x0016	TLS_DHE_RSA_WITH_3DES_EDE_CBC_S	13.0, 13.1	-なし-
SSL3-EDH-RSA-DES-CBC-SHA	0x0015	TLS_DHE_RSA_WITH_DES_CBC_SHA	13.0, 13.1	-なし-
TLS1-ECDHE-RSA-AES256-SHA	0xc014	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	12.1, 13.0, 13.1	12.1, 13.0, 13.1
TLS1-ECDHE-RSA-AES128-SHA	0xc013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	12.1, 13.0, 13.1	12.1, 13.0, 13.1
TLS1-ECDHE-RSA-DES-CBC3-SHA	0xc012	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	13.0, 13.1	-なし-
TLS1-DHE-RSA-AES-128-CBC-SHA	0x0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	12.1, 13.0, 13.1	12.1, 13.0, 13.1
TLS1-DHE-RSA-AES-256-CBC-SHA	0x0039	TLS_DHE_RSA_WITH_AES_256_CBC_SHA	12.1, 13.0, 13.1	12.1, 13.0, 13.1

フロントエンドでサポートされているデフォルトの暗号の一覧を表示するには、コマンドプロンプトで次のように入力します。

```
show ssl cipher DEFAULT_DTLS
1)    Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035
2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0xc014
4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
```



```
            HexCode=0xc013
10 5)    Cipher Name: TLS1-DHE-RSA-AES-256-CBC-SHA    Priority : 5
11      Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
            HexCode=0x0039
12 6)    Cipher Name: TLS1-DHE-RSA-AES-128-CBC-SHA    Priority : 6
13      Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
            HexCode=0x0033
14 7)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA Priority : 7
15      Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
            HexCode=0xc012
16 8)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 8
17      Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
            HexCode=0x000a
18 <!--NeedCopy-->
```

バックエンドでサポートされているデフォルトの暗号の一覧を表示するには、コマンドプロンプトで次のように入力します。

```
1  show ssl cipher DEFAULT_DTLS_BACKEND
2  1)  Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
3      Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
           HexCode=0x0035
4  2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
5      Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
           HexCode=0x002f
6  3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
7      Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
           HexCode=0xc014
8  4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
9      Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
           HexCode=0xc013
10 5)    Cipher Name: TLS1-DHE-RSA-AES-256-CBC-SHA    Priority : 5
11     Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
           HexCode=0x0039
12 6)    Cipher Name: TLS1-DHE-RSA-AES-128-CBC-SHA    Priority : 6
13     Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
           HexCode=0x0033
14 7)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA Priority : 7
15     Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
           HexCode=0xc012
16 8)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 8
17     Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
           HexCode=0x000a
18 <!--NeedCopy-->
```

NetScaler MPX 14000 FIPS プラットフォームでの DTLS 暗号サポート

注: 次の条件が満たされる場合、FIPS プラットフォームでエンライトンドデータサポート（EDT）がサポートされます。

- StoreFront で設定された UDT MSS の値は 900 です。

- Windows クライアントのバージョンは 4.12 以降です。
- DTLS が有効な VDA のバージョンは 7.17 以降です。
- 非 DTLS VDA のバージョンは 7.15 LTSR CU3 以降である。

テーブルの読み方**:**

ビルド番号が指定されていない限り、暗号スイートはリリースのすべてのビルドでサポートされます。

例**:**

- **11.1**、**12.1**、**13.0**、**13.1**: 11.1、12.1、13.0、13.1 リリースのすべてのビルド。
- **-NA-**: 該当なし。

暗号スイート名	16 進コード	Wireshark 暗号スイート名	サポートされているビルド (フロントエンド)	サポートされているビルド (バックエンド)
TLS1-AES-256-CBC-SHA	0x0035	TLS_RSA_WITH_AES_256_CBC_SHA	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
TLS1-AES-128-CBC-SHA	0x002f	TLS_RSA_WITH_AES_128_CBC_SHA	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
SSL3-DES-CBC-SHA	0x0009	TLS_RSA_WITH_DES_CBC_SHA	12.1–49.x, 13.0, 13.1	-なし-
SSL3-DES-CBC3-SHA	0x000a	TLS_RSA_WITH_3DES_EDE_CBC_SHA	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
SSL3-EDH-RSA-DES-CBC3-SHA	0x0016	TLS_DHE_RSA_WITH_3DES_EDE_CBC_S	12.1–49.x, 13.0, 13.1	-なし-
SSL3-EDH-RSA-DES-CBC-SHA	0x0015	TLS_DHE_RSA_WITH_DES_CBC_SHA	12.1–49.x, 13.0, 13.1	-なし-
TLS1-ECDHE-RSA-AES256-SHA	0xc014	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
TLS1-ECDHE-RSA-AES128-SHA	0xc013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
TLS1-ECDHE-RSA-DES-CBC3-SHA	0xc012	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	12.1–49.x, 13.0, 13.1	-なし-
TLS1-DHE-RSA-AES-128-CBC-SHA	0x0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
TLS1-DHE-RSA-AES-256-CBC-SHA	0x0039	TLS_DHE_RSA_WITH_AES_256_CBC_SHA	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1

暗号スイート名	16 進コード	Wireshark 暗号スイート名	サポートされているビルド (フロントエンド)	サポートされているビルド (バックエンド)
TLS1-ECDHE-ECDSA-AES128-SHA	0xc009	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	13.1–21.x	13.1–21.x
TLS1-ECDHE-ECDSA-AES256-SHA	0xc00a	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	13.1–21.x	13.1–21.x
TLS1-ECDHE-ECDSA-DES-CBC3-SHA	0xc008	TLS_ECDHE_ECDSA_WITH_3DES_CBC_SHA	13.1–21.x	13.1–21.x

NetScaler FIPS アプライアンスでサポートされているデフォルト暗号のリストを表示するには、コマンドプロンプトで次のように入力します。

```
show ssl cipher DTLS_FIPS
1)    Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035
2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0xc014
4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0xc013
5)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA  Priority : 5
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0xc012
6)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 6
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0x000a
<!--NeedCopy-->
```

フロントエンド **VPX** アプライアンス、**MPX/SDX**（**Coleto** および **N3** ベース）アプライアンスでの **DTLSv1.2** 暗号サポート

次の表に、DTLSv1.2 プロトコルでサポートされる追加の暗号を示します。

暗号スイート名	16 進コード	Wireshark 暗号スイート名	サポートされているビルド (VPX フロントエンド)	サポートされているビルド (Coleto ベース)	ビルドサポート (N3 ベース)

TLS1.2-AES256-GCM-SHA384	0x009D	TLS_RSA_WITH_AES_256_GCM_SHA384	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-AES128-GCM-SHA256	0x009c	TLS_RSA_WITH_AES_128_GCM_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-ECDHE-RSA-AES256-GCM-SHA384	0xc030	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-ECDHE-RSA-AES128-GCM-SHA256	0xc02f	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-DHE-RSA-AES256-GCM-SHA384	0x009f	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-DHE-RSA-AES128-GCM-SHA256	0x009e	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-AES-256-SHA256	0x003d	TLS_RSA_WITH_AES_256_CBC_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-AES-128-SHA256	0x003c	TLS_RSA_WITH_AES_128_CBC_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-ECDHE-RSA-AES-256-SHA384	0xc028	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-ECDHE-RSA-AES-128-SHA256	0xc027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-DHE-RSA-AES-256-SHA256	0x006b	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-DHE-RSA-AES-128-SHA256	0x0067	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1-ECDHE-ECDSA-AES128-SHA	0xc009	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	13.1–21.x	NA
TLS1-ECDHE-ECDSA-AES256-SHA|0xc00a|TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA|13.1–21.x|NA|
TLS1-ECDHE-ECDSA-DES-CBC3-SHA|0xc008|TLS_ECDHE_ECDSA_WITH_3DES_CBC_SHA|13.1–21.x|NA|
TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256	0xc02b	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	13.1–21.x	NA
TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384	0xc02c	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	13.1–21.x	NA
TLS1.2-ECDHE-ECDSA-AES128-SHA256	0xc023	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	13.1–21.x	NA
TLS1.2-ECDHE-ECDSA-AES256-SHA384	0xc024	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	13.1–21.x	NA

Intel Coleto および Intel Lewisburg SSL チップベースのプラットフォームのサポート

August 15, 2023

以下のアプライアンスには、Intel Coleto チップが搭載されています。

- MPX 5900
- MPX/SDX 8900
- MPX/SDX 15000
- MPX/SDX 15000-50G
- MPX/SDX 26000
- MPX/SDX 26000-50S
- MPX/SDX 26000-100G

次のアプライアンスには Intel Lewisburg チップが同梱されています。

- MPX/SDX 9100
- MPX/SDX 16000

アプライアンスに Coleto（COL）チップとルイスバーグ（LBG）チップのどちらが搭載されているかを識別するには、「show hardware」コマンドを使用します。

```
> sh hardware

    Platform: NSMPX-8900 8\*CPU+4\*F1X+6\*E1K+1\*E1K+1*COL 8955 30010
    Manufactured on: 10/18/2016
    CPU: 2100MHZ
    Host Id: 0
    Serial no: CRAC5CR8UA
    Encoded serial no: CRAC5CR8UA
 Done
<!--NeedCopy-->
```

```
> sh hardware
        Platform: NSMPX-9100 10\*CPU+64GB+8\*F2X+E1K+1*LBG C627 35000
        Manufactured on: 10/1/2021
        CPU: 2300MHZ
        Host Id: 161644678
        Serial no: N2Z3ZD9S21
        Encoded serial no: N2Z3ZD9S21
        Netscaler UUID: 41a26261-227e-11ec-b4db-3cecef56f86b
        BMC Revision: 1.00
Done
<!--NeedCopy-->
```


制限事項

次の暗号、プロトコル、および機能はサポートされていません。

- DH 512 暗号
- SSLv3 プロトコル
- Azure Key Vault
- GnuTLS
- ECC カーブ P_224 および P521 を含む ECDSA 証明書
- DNSSEC オフロード

> 注:Thales Luna Network ハードウェアセキュリティモジュール (HSM)
>
> のサポートは、リリース 13.1 ビルド 33.x 以降で利用できます。

NetScaler MPX および **SDX** プラットフォームでのソフトウェアベースの **SSL** チップ使用率をご覧ください

リリース 13.1 ビルド 21.x から、次のプラットフォームでのソフトウェアベースの SSL チップの使用状況に関する詳細を表示するカウンタが追加されました。

- Intel Coleto チップに同梱されている MPX および SDX プラットフォーム
- Intel Lewisburg チップに同梱されている MPX プラットフォーム。

> 注
>
> : この機能は次のプラットフォームではサポートされていません。
>
> - SDX 9100
> - MPX/SDX 16000

コマンドプロンプトで入力します。

```
> stat ssl

SSL Summary

1.  SSL cards present 4
2.  SSL cards UP 4
    SSL engine status 1
    SSL sessions (Rate) 19849
    SSL Crypto Utilization Asym (%) 88
    SSL Crypto Utilization Symm (%) 1

Crypto Utilization(%)
Asymmetric Crypto Utilization 86.30
Symmetric Crypto Utilization 0.97
```



```
15
16 System
17 Transactions Rate (/s) Total
18 SSL transactions 19849 45900312
19 SSLv2 transactions 0 0
20 SSLv3 transactions 0 0
21 TLSv1 transactions 0 0
22 TLSv1.1 transactions 0 0
23 TLSv1.2 transactions 19849 45900312
24 TLSv1.3 transactions 0 0
25 DTLSv1 transactions 0 0
26 DTLSv1.2 transactions 0 0
27
28 Front End
29 Sessions Rate (/s) Total
30 SSL sessions 19849 45937019
31 SSLv2 sessions 0 0
32 SSLv3 sessions 0 0
33 TLSv1 sessions 0 0
34 TLSv1.1 sessions 0 0
35 TLSv1.2 sessions 19849 45937019
36 TLSv1.3 sessions 0 0
37 DTLSv1 sessions 0 0
38 DTLSv1.2 sessions 0 0
39 New SSL sessions 19881 50722628
40 SSL session misses 0 0
41 SSL session hits 0 0
42
43 Back End
44 Sessions Rate (/s) Total
45 SSL sessions 0 137
46 SSLv3 sessions 0 0
47 TLSv1 sessions 0 0
48 TLSv1.1 sessions 0 0
49 TLSv1.2 sessions 0 137
50 DTLSv1 sessions 0 0
51 Session multiplex attempts 0 0
52 Session multiplex successes 0 0
53 Session multiplex failures 0 0
54
55 Encryption/Decryption statistics
56 Crypto Operation Rate (bytes/s) Total Bytes
57 Bytes encrypted 24338213 27705995030
58 Bytes decrypted 24664169 27942280990
59 Done
60 <!--NeedCopy-->
```

ハードウェアをポーリングすると、次のカウンタの値が得られます。

```
1 -  SSL Crypto Utilization Asym (%) 88
2 -  SSL Crypto Utilization Symm (%) 1
3 <!--NeedCopy-->
```


次のカウンタの値は、ソフトウェアを使用して取得されます。この値は、ハードウェアでポーリングされた値と若干異なる場合があります。

- 暗号使用率 (%)
- 非対称暗号使用率 85.92
- RSA 暗号使用率 11.43
  RSA_4K 0.00
  RSA_2K 11.43
  RSA_1K 0.00
  RSA_Others 0.00
- DH 暗号使用率 74.50
  ECDH 暗号使用率 0.00
  ECDH_P224 0.00
  ECDH_P256 0.00
  ECDH_P384 0.00
  ECDH_P521 0.00
- ECDSA 暗号使用率 0.00
  ECDSA_P224 0.00
  ECDSA_P256 0.00
  ECDSA_P384 0.00
  ECDSA_P521 0.00
- 対称暗号使用率 0.72

暗号ごとに細かく使用するには、次のコマンドを実行します。

```
> stat ssl -detail

SSL Offloading

1.  SSL cards present 4
2.  SSL cards UP 4
    SSL engine status 1
    SSL sessions (Rate) 19862
    SSL Crypto Utilization Asym (%) 88
    SSL Crypto Utilization Symm (%) 1

Crypto Utilization(%)

Asymmetric Crypto Utilization 85.92

RSA Crypto Utilization 11.43
RSA_4K 0.00
RSA_2K 11.43
RSA_1K 0.00
RSA_Others 0.00

DH Crypto Utilization 74.50
```



```
23
24 ECDH Crypto Utilization 0.00
25 ECDH_P224 0.00
26 ECDH_P256 0.00
27 ECDH_P384 0.00
28 ECDH_P521 0.00
29
30 ECDSA Crypto Utilization 0.00
31 ECDSA_P224 0.00
32 ECDSA_P256 0.00
33 ECDSA_P384 0.00
34 ECDSA_P521 0.00
35
36 Symmetric Crypto Utilization 0.72
37 System
38 Transactions Rate (/s) Total
39 SSL transactions 19861 46039342
40 SSLv2 transactions 0 0
41 SSLv3 transactions 0 0
42 TLSv1 transactions 0 0
43 TLSv1.1 transactions 0 0
44 TLSv1.2 transactions 19861 46039342
45 TLSv1.3 transactions 0 0
46 DTLSv1 transactions 0 0
47 DTLSv1.2 transactions 0 0
48 Server in record 117437 277622634
49 Front End
50 Sessions Rate (/s) Total
51 SSL sessions 19862 46076050
52 SSLv2 sessions 0 0
53 SSLv3 sessions 0 0
54 TLSv1 sessions 0 0
55 TLSv1.1 sessions 0 0
56 TLSv1.2 sessions 19862 46076050
57 TLSv1.3 sessions 0 0
58 DTLSv1 sessions 0 0
59 DTLSv1.2 sessions 0 0
60 New SSL sessions 19801 50861234
61 SSL session misses 0 0
62 SSL session hits 0 0
63 Session Renegotiation
64 SSL session renegotiations 0 0
65 SSLv3 session renegotiations 0 0
66 TLSv1 session renegotiations 0 0
67 TLSv1.1 session renegotiations 0 0
68 TLSv1.2 session renegotiations 0 0
69 DTLSv1 session renegotiations 0 0
70 DTLSv1.2 session renegotiations 0 0
71 Key Exchanges
72 RSA 512-bit key exchanges 0 0
73 RSA 1024-bit key exchanges 0 2032658
74 RSA 2048-bit key exchanges 0 143
75 RSA 3072-bit key exchanges 0 7757028
```



```
RSA 4096-bit key exchanges 0 2238698
DH 512-bit key exchanges 0 0
DH 1024-bit key exchanges 0 0
DH 2048-bit key exchanges 19862 5477702
DH 4096-bit key exchanges 0 0
ECDHE 521 curve key exchanges 0 0
ECDHE 384 curve key exchanges 0 0
ECDHE 256 curve key exchanges 0 28569821
ECDHE 224 curve key exchanges 0 0
Total ECDHE key exchanges 0 28569821
Ciphers Negotiated
RC4 40-bit encryptions 0 0
RC4 56-bit encryptions 0 0
RC4 64-bit encryptions 0 0
RC4 128-bit encryptions 0 0
DES 40-bit encryptions 0 0
DES 56-bit encryptions 0 0
3DES 168-bit encryptions 0 0
AES 128-bit encryptions 0 0
AES 256-bit encryptions 19862 17506229
RC2 40-bit encryptions 0 0
RC2 56-bit encryptions 0 0
RC2 128-bit encryptions 0 0
AES-GCM 128-bit encryptions 0 0
AES-GCM 256-bit encryptions 0 28569821
Null cipher encryptions 0 0
Hashes
MD5 hashes 0 0
SHA hashes 0 12028527
SHA256 hashes 19862 5477702
SHA384 hashes 0 0
Handshakes
SSLv2 SSL handshakes 0 0
SSLv3 SSL handshakes 0 0
TLSv1 SSL handshakes 0 0
TLSv1.1 SSL handshakes 0 0
TLSv1.2 SSL handshakes 19862 46076050
TLSv1.3 SSL handshakes 0 0
DTLSv1 SSL handshakes 0 0
DTLSv1.2 SSL handshakes 0 0
Client Authentications
SSLv2 client authentications 0 0
SSLv3 client authentications 0 0
TLSv1 client authentications 0 0
TLSv1.1 client authentications 0 0
TLSv1.2 client authentications 0 0
TLSv1.3 client authentications 0 0
DTLSv1 client authentications 0 0
DTLSv1.2 client authentications 0 0
Authentications
RSA authentications 19862 17506229
DH authentications 0 0
DSS (DSA) authentications 0 0
```



```
ECDSA authentications 0 28569821
Null authentications 0 0
Back End
Sessions Rate (/s) Total
SSL sessions 0 137
SSLv3 sessions 0 0
TLSv1 sessions 0 0
TLSv1.1 sessions 0 0
TLSv1.2 sessions 0 137
DTLSv1 sessions 0 0
Session multiplex attempts 0 0
Session multiplex successes 0 0
Session multiplex failures 0 0
Session Renegotiation
SSL session renegotiations 0 0
SSLv3 session renegotiations 0 0
TLSv1 session renegotiations 0 0
TLSv1.1 back-end session renegot 0 0
TLSv1.2 back-end session renegot 0 0
DTLSv1 session renegotiations 0 0
Key Exchanges
RSA 512-bit key exchanges 0 0
RSA 1024-bit key exchanges 0 0
RSA 2048-bit key exchanges 0 137
RSA 3072-bit key exchanges 0 0
RSA 4096-bit key exchanges 0 0
DH 512-bit key exchanges 0 0
DH 1024-bit key exchanges 0 0
DH 2048-bit key exchanges 0 0
DH 4096-bit key exchanges 0 0
ECDHE 521 curve key exchanges 0 0
ECDHE 384 curve key exchanges 0 0
ECDHE 256 curve key exchanges 0 0
ECDHE 224 curve key exchanges 0 0
Ciphers Negotiated
RC4 40-bit encryptions 0 0
RC4 56-bit encryptions 0 0
RC4 64-bit encryptions 0 0
RC4 128-bit encryptions 0 0
DES 40-bit encryptions 0 0
DES 56-bit encryptions 0 0
3DES 168-bit encryptions 0 0
AES 128-bit encryptions 0 0
AES 256-bit encryptions 0 137
RC2 40-bit encryptions 0 0
RC2 56-bit encryptions 0 0
RC2 128-bit encryptions 0 0
AES-GCM 128-bit encryptions 0 0
AES-GCM 256-bit encryptions 0 0
Null encryptions 0 0
Hashes
MD5 hashes 0 0
SHA hashes 0 137
```



```
SHA256 hashes 0 0
SHA384 hashes 0 0
Handshakes
SSLv3 handshakes 0 0
TLSv1 handshakes 0 0
TLSv1.1 handshakes 0 0
TLSv1.2 handshakes 0 137
DTLSv1 handshakes 0 0
Client Authentications
SSLv3 client authentications 0 0
TLSv1 client authentications 0 0
TLSv1.1 client authentications 0 0
TLSv1.2 client authentications 0 0
DTLSv1 client authentications 0 0
Authentications
RSA authentications 0 137
DH authentications 0 0
DSS authentications 0 0
ECDSA authentications 0 0
Null authentications 0 0
System Total
RSA key exchanges offloaded 0 0
RSA sign operations offloaded 0 0
DH key exchanges offloaded 19841 5481037
RC4 encryptions offloaded 0 0
DES encryptions offloaded 0 0
AES encryptions offloaded 0 0
AES-GCM 128-bit encryptions offl 0 0
AES-GCM 256-bit encryptions offl 0 0
Encryption/Decryption statistics
Crypto Operation Rate (bytes/s) Total Bytes
Bytes encrypted 12129801 27790903638
Bytes encrypted in hardware 12129801 27790903638
Bytes encrypted in software 0 0
Bytes encrypted on the front-end 5450907 13430410630
Bytes encrypted in hardware on t 5450907 13430410630
Bytes encrypted in software on t 0 0
Bytes encrypted on the back-end 6678894 14360493008
Bytes encrypted in hardware on t 6678894 14360493008
Bytes encrypted in software on t 0 0
Bytes decrypted 12449504 28029427518
Bytes decrypted in hardware 12449504 28029427518
Bytes decrypted in software 0 0
Bytes decrypted on the front-end 8190208 19876552670
Bytes decrypted in hardware on t 8190208 19876552670
Bytes decrypted in software on t 0 0
Bytes decrypted on the back-end 4259296 8152874848
Bytes decrypted in hardware on t 4259296 8152874848
Bytes decrypted in software on t 0 0
SSL
Rate (/s) Total
Total SPCB in use -87 84656
Active SSL sessions -30309 5615559
```



```
Current queue size -1 4153
CardQ
Rate (/s) Total
In Q count for current card -1 4153
In BulkQ count for current card 0 0
In KeyQ count for current card -1 4153
Done
<!--NeedCopy-->
```

メモ

- 管理パーティションはサポートされていますが、すべてのパーティションの使用率はデフォルトパーティションに表示されます。デフォルト以外のパーティションでは、これらの値は 0 と表示されます。
- クラスタセットアップでは、CLIP アドレスには、クラスタ内のすべてのノードの平均使用率が表示されます。ノード固有の使用方法については、各ノードの CLI でコマンドを実行します。クラスタのノードが同じハードウェアでホストされている場合、SDX プラットフォームではこのデータが正しくない可能性があります。
- SDX プラットフォーム上の VPX インスタンスの場合、各 VPX インスタンスの使用率が表示されます。

VPX FIPS アプライアンス

February 15, 2024

NetScaler VPX FIPS アプライアンスは、米国国立標準技術研究所（NIST）による FIPS 140-3 レベル 1 の検証中です（現在 IUT https://csrc.nist.gov/projects/cryptographic-module-validation-program/modules-in-process/iut-list中です）。FIPS 140-3 標準および検証プログラムの詳細については、NIST およびカナダサイバーセキュリティセンター（CCCS）の暗号モジュール検証プログラム（CMVP）の Web サイトhttps://csrc.nist.gov/projects/cryptographic-module-validation-programを参照してください。

注

- MPX 8900 FIPS、MPX 9100 FIPS、MPX 15000-50G FIPS、および VPX FIPS プラットフォームでサポートされているのは、NetScaler ダウンロードページの「NetScaler Release 13.1-FIPS」に記載されているファームウェアバージョンのみです。
- 12.1-FIPS ソフトウェアバージョンを実行する NetScaler FIPS アプライアンスでクラシックポリシーを構成している場合は、13.1-FIPS へのアップグレードを実行する前にhttps://support.citrix.com/article/CTX234821/citrix-adc-deprecated-classic-policy-based-features-and-functionalities-faqsを参照してください。
- 13.1-FIPS 上の TLS 1.3 は、拡張 SSL プロファイルを使用してのみ設定できます。プロファイルを使用して TLS 1.3 を設定する方法の詳細については、RFC 8446 で定義されている TLS 1.3 プロトコルのサポートを参照してください。

前提条件

- オンプレミスのハイパーバイザーの場合は、Citrix の Web サイトから特別なビルドをダウンロードしてください。それぞれのハイパーバイザー用の完全な VPX FIPS パッケージをダウンロードしてください。
- NetScaler VPX FIPS アプライアンスがプールライセンスモデルで期待どおりに機能するには、FIPS インスタンスライセンスと帯域幅プールが必要です。プールされていないライセンスの場合、必要な帯域幅容量の VPX FIPS ライセンスが 1 つ必要です。

構成

このモジュールは、アプリケーションソフトウェアとオペレーティングシステムの両方を含むソフトウェアパッケージとして利用できます。NetScaler VPX FIPS ライセンスを購入したら、Citrix の Web サイトから最新の NetScaler VPX FIPS イメージを入手してください。

次の手順を実行します:

1. 最新の NetScaler VPX FIPS イメージを、ESXi、Citrix Hypervisor、Hyper-V、KVM、AWS、Azure、または GCP のいずれかのハイパーバイザーにアップロードします。

   > 注:
   >
   > VPX FIPS は ESXi 7.0.3 で認定される予定です。

2. NetScaler VPX FIPS プラットフォームライセンスと NetScaler VPX 帯域幅ライセンスを適用し、アプライアンスをウォームリブートします。
3. アプライアンスが起動したら、CLI で次のコマンドを実行します。

   ```
   1 > show system fipsStatus
   2 <!--NeedCopy-->
   ```

   次の出力を取得する必要があります。

   ```
   1 FipsStatus: System is operating in FIPS mode
   2 Done
   3 <!--NeedCopy-->
   ```

   次の出力が表示される場合は、トラブルシューティングのセクションを参照して解決手順を確認してください。

   ```
   1 FipsStatus: "System is operating in non FIPS mode"
   2 Done
   3 >
   4 <!--NeedCopy-->
   ```

4. 『セキュア・デプロイメント・ガイド』の設定ガイドラインに従ってください。

RADIUS を使用したリモート認証の詳細については、「RADIUS を使用したリモート認証の設定」を参照してください。

VPX FIPS アプライアンスでサポートされている暗号

3DES 暗号を除く NetScaler MPX/SDX 14000 FIPS アプライアンスでサポートされているすべての暗号は、VPX FIPS アプライアンスでサポートされています。NetScaler VPX FIPS アプライアンスでサポートされている暗号の全リストについては、次のトピックを参照してください。

- NetScaler VPX FIPS および MPX FIPS アプライアンスでの暗号サポート。

VPX FIPS アプライアンスのアップグレード

「NetScaler スタンドアロンアプライアンスのアップグレード」の手順に従って、VPX FIPS アプライアンスをアップグレードします 。

重要:. /installns コマンドは . /installns -Fに置き換えてください。

> 注:
>
> リリース 13.1 FIPS ビルド 37.159 以降にアップグレードする場合、pfx ファイルを使用して証明書とキーのペアを追加すると失敗します。
>
> 回避策: アップグレードの前に、AES256 などの FIPS 認定の暗号を使用して pfx ファイルを作成します。
>
> 例:

```
root@ns# cd /nsconfig/ssl/
root@ns# openssl pkcs12 -export -out example.name.pfx -inkey
    example.key -in example.pem -certpbe AES-256-CBC -keypbe AES
    -256-CBC
<!--NeedCopy-->
```

制限事項

- VPX FIPS アプライアンスでは TACACS 認証はサポートされていません。
- VPX FIPS は別のイメージです。VPX バージョンから VPX FIPS バージョンへのソフトウェアバージョンアップグレードはサポートされていません。また、VPX FIPS ソフトウェアバージョンを VPX ソフトウェアバージョンにダウングレードまたはアップグレードすることはできません。
- VPX FIPS イメージは、NetScaler SDX および NetScaler SDX FIPS アプライアンスではサポートされていません。

トラブルシューティング

show system fipsStatus コマンドを実行すると、出力は次のようになります。


```
FipsStatus: "System is operating in non FIPS mode"
Done
>
<!--NeedCopy-->
```

理由は次のいずれかである可能性があります。

1. ライセンスの有効期限が切れているか、正しくありません。
2. システムが FIPS モードで起動できない。このエラーは、管理コアまたはパケットエンジンの POST 障害が原因である可能性があります。

解決するには:

1. 正しい NetScaler VPX FIPS ライセンスがインストールされていて、ライセンスの有効期限が切れていないことを確認してください。
2. 管理コアまたはパケットエンジンでパワーオンセルフテスト (POST) 障害が発生していないかどうかを確認します。次のコマンドを実行します:

   ```
   >shell
   #nsconmsg -g drbg -g ssl_err -g fips -d statswt0
   <!--NeedCopy-->
   ```

   パケットエンジンのブートアップ中に POST `nsssl_err_fips_post_failed counter` が失敗した場合にインクリメントされます。つまり、データプレーンに障害が発生しています。

   カウンタが増えない場合は、(`/var/log/FIPS-post.log`) ログファイルでアルゴリズムテストに失敗したエントリがないか確認してください。つまり、管理コアの POST 障害（コントロールプレーンの障害）を確認します。

   いずれの場合も、NetScaler サポートにお問い合わせください。

MPX FIPS アプライアンス

December 8, 2023

NetScaler MPX 8900 FIPS、MPX 9100 FIPS、および MPX 15000-50G FIPS アプライアンスは、FIPS 140-3 レベル 1 のセキュリティ要件について、サードパーティの研究所によって（現在 IUT で https://csrc.nist.gov/projects/cryptographic-module-validation-program/modules-in-process/iut-list）検証中です。FIPS 140-3 標準および検証プログラムの詳細については、米国標準技術研究所（NIST）およびカナダサイバーセキュリティセンター（CCCS）の暗号モジュール検証プログラム（CMVP）の Web サイトを参照してください。https://csrc.nist.gov/projects/cryptographic-module-validation-program

> メモ
>
> - MPX 8900 FIPS、MPX 9100 FIPS、および MPX 15000-50G FIPS アプライアンスは、サードパーティ製のハードウェアセキュリティモジュールを使用しなくなりました。FIPS 検証の要件はシステムに組み込まれています。
> - MPX 8900 FIPS、MPX 9100 FIPS、MPX 15000-50G FIPS、および VPX FIPS プラットフォームでサポートされているのは、NetScaler ダウンロードページの「NetScaler Release 13.1-FIPS」に記載されているファームウェアバージョンのみです。
> - 12.1-FIPS ソフトウェアバージョンを実行する NetScaler FIPS アプライアンスでクラシックポリシーを構成している場合は、13.1-FIPS へのアップグレードを実行する前にhttps://support.citrix.com/article/CTX234821/citrix-adc-deprecated-classic-policy-based-features-and-functionalities-faqsを参照してください。
> - 13.1-FIPS 上の TLS 1.3 は、拡張 SSL プロファイルを使用してのみ設定できます。プロファイルを使用して TLS 1.3 を設定する方法の詳細については、RFC 8446 で定義されている TLS 1.3 プロトコルのサポートを参照してください。

前提条件

- 帯域幅ライセンスに加えて FIPS プラットフォームライセンス。

MPX 8900 FIPS、**MPX 9100 FIPS**、および **MPX 15000-50G FIPS** アプライアンスでサポートされている暗号

3DES 暗号を除く NetScaler MPX/SDX 14000 FIPS アプライアンスでサポートされているすべての暗号は、MPX 8900、MPX 9100 FIPS、および MPX 15000-50G FIPS アプライアンスでサポートされています。これらのアプライアンスでサポートされている暗号の全リストについては、「NetScaler VPX FIPS および MPX FIPS アプライアンスでの暗号サポート」を参照してください。

MPX FIPS アプライアンスのアップグレード

「NetScaler スタンドアロンアプライアンスのアップグレード」の手順に従って、MPX FIPS アプライアンスをアップグレードします 。

> 注**:**
>
> リリース 13.1 FIPS ビルド 37.159 以降にアップグレードする場合、pfx ファイルを使用して証明書とキーのペアを追加すると失敗します。
>
> 回避策: アップグレードの前に、AES256 などの FIPS 認定の暗号を使用して pfx ファイルを作成します。
>
> 例:


```
root@ns# cd /nsconfig/ssl/
root@ns# openssl pkcs12 -export -out example.name.pfx -inkey
    example.key -in example.pem -certpbe AES-256-CBC -keypbe AES
    -256-CBC
<!--NeedCopy-->
```

制限事項

TACACS 認証は MPX FIPS アプライアンスではサポートされていません。

構成

1. アプライアンスが起動したら、CLI で次のコマンドを実行します。

```
> show system fipsStatus
<!--NeedCopy-->
```

2. 次の出力を取得する必要があります。

```
FipsStatus: "System is operating in FIPS mode"
Done
>
<!--NeedCopy-->
```

3. 次のような出力が出た場合は、ライセンスを確認してください。

```
FipsStatus: "System is operating in non FIPS mode"
Done
>
<!--NeedCopy-->
```

MPX アプライアンスを FIPS モードで動作するように初期化するには、次の手順を実行します。

1. 強力なパスフレーズ要件を強制してください。
2. デフォルトの TLS 証明書を置き換えてください。
3. ウェブ GUI への HTTP アクセスを無効にします。
4. 初期設定後、ローカル認証を無効にし、LDAP を使用してリモート認証を設定します。

GUI を使用して強固なパスフレーズ要件を適用

パスフレーズは、PBKDF2 を使用してキーを生成するために使用されます。管理者は、GUI を使用して強力なパスフレーズ要件を設定してください。

1. [**System**] **>** [**Settings**] に移動します。
2. 「設定」セクションで、「グローバルシステム設定の変更」をクリックします。

3.「強力なパスワード」フィールドで、「すべて有効にする」を選択します。
4.「パスワードの最小文字数」フィールドに「8」と入力します。
5. [**OK**] をクリックします。

デフォルトの **TLS** 証明書を置き換える

デフォルトでは、MPX FIPS アプライアンスには TLS 接続（`ns-server.cert`および`ns-server.key`）用の工場出荷時にプロビジョニングされた RSA 証明書が含まれています。この証明書は本番環境での使用を目的としたものではないため、交換する必要があります。初期インストール後に、デフォルトの証明書を新しい証明書に置き換えます。

デフォルトの **TLS** 証明書を置き換えるには**:**

1. コマンドプロンプトで次のコマンドを入力して、アプライアンスのホスト名を設定します。

```
set ns hostName <hostname>
```

GUI を使用して証明書署名リクエスト **(CSR)** を作成する

1. [トラフィック管理**] > [SSL] > [SSL** ファイル] に移動します。
2. [**CSR**] タブで、[証明書署名要求 **(CSR)** の作成] をクリックします。
3. 値を入力し、[**Create**] をクリックします。

> 注**:**
>
> 共通名フィールドには、ADC CLI を使用して設定されたホスト名の値が含まれます。

4. CSR ファイルを信頼できる認証局 (CA) に送信します。CSR ファイルは`/nsconfig/ssl`ディレクトリにあります。
5. CA から証明書を受け取ったら、ファイルを`/nsconfig/ssl`ディレクトリにコピーします。
6. [トラフィック管理**] > [SSL] > [**証明書**] > [**サーバー証明書] に移動します。
7. **ns-server-certificate** を選択します。
8. [**Update**] をクリックします。
9.「証明書とキーを更新」をクリックします。
10.「証明書ファイル名」フィールドで、認証局 (CA) から受け取った証明書ファイルを選択します。ファイルがローカルコンピューターにある場合は、「ローカル」を選択します。それ以外の場合は、「アプライアンス」を選択します。
11. [**Key File Name**] フィールドで、デフォルトのプライベートキーファイル名 (`ns-server.key`) を指定します。

12.「ドメインチェックなし」オプションを選択します。

13.［**OK**］をクリックします。

ウェブ **GUI** への **HTTP** アクセスを無効にする

管理インターフェイスと Web GUI へのトラフィックを保護するには、アプライアンスを HTTPS を使用するように設定する必要があります。新しい証明書を追加したら、CLI を使用して GUI 管理インターフェイスへの HTTP アクセスを無効にします。

コマンドプロンプトで入力します:

```
set ns ip <NSIP> -gui SECUREONLY
```

ローカル認証を無効にし、**LDAP** を使用してリモート認証を設定します

スーパーユーザーアカウントは、初期設定に必要なルート CLI アクセス権限を持つデフォルトアカウントです。初期設定時には、ローカルシステム認証を無効にして、すべてのローカルアカウント (スーパーユーザーアカウントを含む) へのアクセスをブロックし、スーパーユーザー権限がどのユーザーアカウントにも割り当てられないようにします。

CLI を使用してローカルシステム認証を無効にし、外部システム認証を有効にするには:

コマンドプロンプトで入力します:

```
set system parameter -localauth disabled
```

LDAP 認証の設定の指示に従って、LDAP を使用するように外部システム認証を設定します。

RADIUS によるリモート認証の設定

FIPS 環境では RADIUS 認証を設定できます。

> 注**:**
>
> 「**RADIUS** 到達可能性のテスト」オプションは RADIUS ではサポートされていません。

CLI を使用して **TLS** 経由の **RADIUS** を構成する

コマンドプロンプトで次のように入力します:

```
1 add authentication radiusAction <name> [-serverIP] [-serverPort ] [-
     transport <transport>] [-targetLBVserver <string>]
2 <!--NeedCopy-->
```

例


```
add authentication radiusAction RadAction -serverIP 1.1.1.1 -radkey 123
    -transport TLS -targetLBVserver rad-lb
<!--NeedCopy-->
```

注:

- TLS トランスポートタイプの場合は、TCP タイプのターゲット負荷分散仮想サーバーを構成し、タイプ SSL_TCP のサービスをこの仮想サーバーにバインドします。
- サーバー名はサポートされていません。
- RADIUS アクション用に構成された IP アドレスとポート番号は、構成されたターゲット負荷分散仮想サーバーの IP アドレスとポート番号と一致する必要があります。

GUI を使用して **TLS** 経由の **RADIUS** を構成する

1. セキュリティ **>** **AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** アクション **>** サーバに移動します。
2. 既存のサーバーを選択するか、サーバーを作成します。

   サーバーの作成の詳細については、「GUI を使用して RADIUS サーバーを構成するには」を参照してください。
3. [トランスポート] で [**TLS**] を選択します。
4. [ターゲット負荷分散仮想サーバー] で、仮想サーバーを選択します。負荷分散仮想サーバーの作成について詳しくは、「仮想サーバーの作成」を参照してください。

   注:

   - TLS トランスポートタイプの場合は、TCP タイプのターゲット負荷分散仮想サーバーを構成し、タイプ SSL_TCP のサービスをこの仮想サーバーにバインドします。
   - サーバー名はサポートされていません。
   - RADIUS アクション用に構成された IP アドレスとポート番号は、構成されたターゲット負荷分散仮想サーバーの IP アドレスとポート番号と一致する必要があります。
5. 「作成」をクリックします。

MPX 14000 FIPS アプライアンス

August 15, 2023

重要:

- MPX 9700/10500/12500/15500 FIPS プラットフォームは終焉を迎えている。

> - NetScaler MPX 14000FIPS および NetScalerMPX の構成手順 9700/10500/12500/15500 FIPS アプライアンスは異なります。MPX 14000FIPS アプライアンスはファームウェア v2.2 を使用しません。MPX 9700 プラットフォームのハードウェアセキュリティモジュール (HSM) で作成された FIPS キーは、MPX 14000 プラットフォームの HSM に転送できません。もう一方のラウンドもサポートされていません。ただし、RSA キーを FIPS キーとしてインポートした場合は、RSA キーを MPX 14000 プラットフォームにコピーできます。次に、FIPS キーとしてインポートします。2048 ビットと 3072 ビットのキーのみがサポートされています。
> - NetScaler ダウンロードページの「NetScaler リリース 12.1-FIPS」および「NetScaler リリース 12.1-ndCPP」にリストされているファームウェアのバージョンは、MPX 14000 FIPS または SDX 14000 FIPS プラットフォームではサポートされていません。これらのプラットフォームでは、ダウンロードページで入手できる他の最新の NetScaler ADC ファームウェアバージョンを使用できます。

FIPS アプライアンスには、改ざん防止（改ざん防止）暗号化モジュール（Cavium CNN3560-NFBE-G）が装備されており、FIPS 140-2 レベル 3 仕様（リリース 12.0 ビルド 56.x 以降）に準拠するように設計されています。クリティカルセキュリティパラメータ（CSP）は、主にサーバーの秘密鍵であり、HSM とも呼ばれる暗号化モジュール内に安全に格納され、生成されます。CSP は、HSM の境界外でアクセスされることはありません。スーパーユーザー (`nsroot`) だけが、HSM 内に格納されているキーに対して操作を実行できます。

FIPS アプライアンスを設定する前に、FIPS カードの状態を確認し、カードを初期化する必要があります。FIPS キーとサーバー証明書を作成し、追加の SSL 構成を追加します。

サポートされている FIPS 暗号の詳細については、「FIPS 承認アルゴリズムと暗号」を参照してください。

HA セットアップでの FIPS アプライアンスの構成の詳細については、HA セットアップのアプライアンスでの FIPS の構成を参照してください。

制限事項

1. SSLv3 プロトコルを使用した SSL 再ネゴシエーションは、MPX FIPS アプライアンスのバックエンドではサポートされません。
2. 1024 ビットおよび 4096 ビットのキーと指数値 3 はサポートされていません。
3. 4096 ビットサーバー証明書はサポートされていません。
4. 4096 ビットのクライアント証明書はサポートされていません (バックエンドサーバーでクライアント認証が有効になっている場合)。

HSM の構成

MPX 14000 FIPS アプライアンスで HSM を構成する前に、FIPS カードの状態をチェックして、ドライバが正しくロードされていることを確認します。次に、カードを初期化します。

コマンドプロンプトで入力します。


```
show fips

FIPS Card is not configured
<!--NeedCopy-->
```

ドライバが正しくロードされていない場合は、「エラー：操作が許可されていません-システムに FIPS カードがありません」というメッセージが表示されます。

FIPS カードの初期化

FIPS カードを適切に初期化するには、アプライアンスを 3 回再起動する必要があります。

> 重要
>
> - アプライアンスで/nsconfig/fipsディレクトリが正常に作成されたことを確認します。
> - アプライアンスを 3 回目に再起動する前に、設定を保存しないでください。

FIPS カードを初期化するには、次の手順に従います。

1. FIPS カードをリセットします (reset fips)。
2. アプライアンスを再起動します (reboot)。
3. パーティション 0 と 1 にはセキュリティ担当者のパスワードを設定し、partition (set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -hsmLabel NSFIPS) にはユーザーパスワードを設定します。

   注:set または reset コマンドの実行には 60 秒以上かかります。
4. 設定を保存します (saveconfig)。
5. メインパーティション (master_pek.key) のパスワードで暗号化されたキーが/nsconfig/fips/ディレクトリに作成されていることを確認します。
6. アプライアンスを再起動します (reboot)。
7. デフォルトパーティション (default_pek.key) のパスワードで暗号化されたキーが/nsconfig/fips/ディレクトリに作成されていることを確認します。
8. アプライアンスを再起動します (reboot)。
9. FIPS カードが稼働していることを確認します（show fips)。

CLI を使用して **FIPS** カードを初期化します　set fipsコマンドは、FIPS カードのハードウェアセキュリティモジュール (HSM) を初期化し、新しいセキュリティ担当者のパスワードとユーザーパスワードを設定します。

注意：このコマンドは、FIPS カード上のすべてのデータを消去します。コマンドの実行を続行する前に、プロンプトが表示されます。変更を適用するには、このコマンドの実行前と実行後に再起動が必要です。このコマンドを実行した後、アプライアンスを再起動する前に、設定を保存します。

コマンドプロンプトで、次のコマンドを入力します。

```
reset fips

reboot

set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS

This command will erase all data on the FIPS card. You must save the
   configuration (saveconfig) after executing this command. Do you want
    to continue?(Y/N)y

<!--NeedCopy-->
```

注:`set fips` コマンドを実行すると、次のメッセージが表示されます。

```
This command will erase all data on the FIPS card. You must save the
   configuration (saveconfig) after executing this command. [Note: On
   MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by
   default, and the -initHSM Level-2 option is internally converted to
   Level-3]  Do you want to continue?(Y/N)y

saveconfig

reboot

reboot

show fips

        FIPS HSM Info:
                 HSM Label              : NetScaler FIPS
                 Initialization         : FIPS-140-2 Level-3
                 HSM Serial Number      : 3.1G1836-ICM000136
                 HSM State              : 2
                 HSM Model              : NITROX-III CNN35XX-NFBE
                 Hardware Version       : 0.0-G
                 Firmware Version       : 1.0
                 Firmware Build         : NFBE-FW-1.0-48
                 Max FIPS Key Memory    : 102235
                 Free FIPS Key Memory   : 102231
                 Total SRAM Memory      : 557396
                 Free SRAM Memory       : 262780
                 Total Crypto Cores     : 63
                 Enabled Crypto Cores   : 63

<!--NeedCopy-->
```


FIPS キーを作成する

MPX 14000 FIPS アプライアンスで FIPS キーを作成するか、既存の FIPS キーをアプライアンスにインポートできます。MPX 14000 FIPS アプライアンスは、2048 ビットと 3072 ビットのキーと、指数値 F4 (その値は 65537) のみをサポートします。PEM キーの場合、指数は不要です。FIPS キーが正しく作成されていることを確認します。証明書署名要求とサーバー証明書を作成します。最後に、証明書とキーのペアをアプライアンスに追加します。

キータイプ (RSA または ECDSA) を指定します。ECDSA キーの場合は、カーブだけを指定します。カーブ P_256 および P_384 の ECDSA キーの作成がサポートされています。

> 注:
>
> 1024 ビットおよび 4096 ビットのキー、および指数値 3 はサポートされていません。

CLI を使用して FIPS キーを作成する

コマンドプロンプトで入力します。

```
create ssl fipsKey <fipsKeyName> -keytype ( RSA | ECDSA ) [-exponent ( 3 | F4 )] [-modulus <positive_integer>] [-curve ( P_256 | P_384 )]
<!--NeedCopy-->
```

Example1:

```
create fipsKey f1 -keytype RSA -modulus 2048 -exponent F4


show ssl fipskey f1

FIPS Key Name: f1  Key Type: RSA Modulus: 2048   Public Exponent: F4 (Hex: 0x10001)

<!--NeedCopy-->
```

Example2:

```
> create fipskey f2 -keytype ECDSA -curve P_256


> sh fipskey f2
    FIPS Key Name: f2   Key Type: ECDSA Curve: P_256

<!--NeedCopy-->
```

GUI を使用して FIPS キーを作成する

1. [トラフィック管理] **> [SSL] > [FIPS**] に移動します。

2. 詳細ウィンドウの [FIPS キー] タブで、[追加] をクリックします。
3. [FIPS キーの作成] ダイアログボックスで、次のパラメータの値を指定します。

   - FIPS キー名 *—fipsKeyName
   - モジュラス *-モジュラス
   - 指数 *-指数

   * 必須パラメータ

4. [作成] をクリックし、[閉じる] をクリックします。
5. [FIPS キー] タブで、作成した FIPS キーに対して表示された設定が正しいことを確認します。

FIPS キーをインポートする

既存の FIPS キーを FIPS アプライアンスで使用するには、アプライアンスのハードディスクから HSM に FIPS キーを転送する必要があります。

注**:** FIPS キーのインポート時のエラーを回避するには、インポートされたキーの名前が、作成時の元のキー名と同じであることを確認してください。

CLI を使用して FIPS キーをインポートする

コマンドプロンプトで入力します。

```
import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-
   wrapKeyName <string>] [-iv<string>] -exponent  F4 ]
<!--NeedCopy-->
```

例**:**

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4


import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM

<!--NeedCopy-->
```

show fipskeyコマンドを実行して、FIPS キーが正しく作成またはインポートされていることを確認します。

```
show fipskey
1)      FIPS Key Name: Key-FIPS-2

<!--NeedCopy-->
```


GUI を使用して **FIPS** キーをインポートする

1. [トラフィック管理] **> [SSL] > [FIPS]** に移動します。
2. 詳細ウィンドウの [FIPS キー] タブで、[インポート] をクリックします。
3. [FIPS キーとしてインポート] ダイアログボックスで、[FIPS キーファイル] を選択し、次のパラメータの値を設定します。

   - FIPS キー名 *
   - Key File Name* —デフォルト以外の場所にファイルを配置するには、完全なパスを指定するか、[参照] をクリックして場所に移動します。
   - 指数 *

   * 必須パラメータ
4. [インポート] をクリックし、[閉じる] をクリックします。
5. [FIPS キー] タブで、インポートした FIPS キーに対して表示される設定が正しいことを確認します。

FIPS キーをエクスポートする

FIPS HSM で作成されたキーのバックアップを作成することをお勧めします。HSM 内のキーが削除された場合、同じキーを再度作成することはできず、それに関連付けられているすべての証明書は役に立たなくなります。

キーをバックアップとしてエクスポートするだけでなく、別のアプライアンスに転送するためにキーをエクスポートする必要がある場合があります。

次の手順では、FIPS キーをアプライアンスの CompactFlash 上の/nsconfig/sslフォルダにエクスポートし、強力な非対称キー暗号化方式を使用してエクスポートされたキーを保護する方法について説明します。

CLI を使用して **FIPS** キーをエクスポートする

コマンドプロンプトで入力します。

```
export ssl fipsKey <fipsKeyName> -key <string>
<!--NeedCopy-->
```

例**:**

```
export fipskey Key-FIPS-1 -key Key-FIPS-1.key
<!--NeedCopy-->
```

GUI を使用して **FIPS** キーをエクスポートする

1. [トラフィック管理] **> [SSL] > [FIPS]** に移動します。

2. 詳細ウィンドウの [FIPS キー] タブで、[エクスポート] をクリックします。
3. [FIPS キーをファイルにエクスポート] ダイアログボックスで、次のパラメータの値を指定します。
   - FIPS キー名 *—fipsKeyName
   - ファイル名 *-キー (デフォルト以外の場所にファイルを配置するには、完全パスを指定するか、[参照] ボタンをクリックして場所を指定します)。

   * 必須パラメータ
4. **[エクスポート]** をクリックし、**[閉じる]** をクリックします。

外部キーをインポートする

NetScaler ADC アプライアンスの HSM 内に作成された FIPS キーを転送できます。外部秘密鍵（標準の NetScaler ADC、Apache、または IIS で作成された鍵など）を NetScaler ADC FIPS アプライアンスに転送することもできます。外部キーは、OpenSSL などのツールを使用して HSM の外部で作成されます。外部キーを HSM にインポートする前に、/nsconfig/sslの下にあるアプライアンスのフラッシュドライブにコピーします。

MPX 14000 FIPS アプライアンスでは、外部キーのインポート時に、**import** ssl fipskeyコマンドの-exponent パラメータは不要です。キーのインポート時に正しい公開指数が自動的に検出され、-exponent パラメータの値は無視されます。

NetScaler FIPS アプライアンスは、3 または F4 以外の公開指数を持つ外部キーをサポートしていません。

MPX 14000 FIPS アプライアンスにはラップキーは必要ありません。

外部の暗号化された FIPS キーを MPX 14000 FIPS アプライアンスに直接インポートすることはできません。キーをインポートするには、まずキーを復号化してからインポートする必要があります。キーを復号化するには、シェルプロンプトで次のように入力します。

```
openssl rsa -in <EncryptedKey.key> > <DecryptedKey.out>
<!--NeedCopy-->
```

注： RSA キーを FIPS キーとしてインポートする場合は、セキュリティ上の理由から RSA キーをアプライアンスから削除することをお勧めします。

CLI を使用して外部キーを **FIPS** キーとしてインポートする

1. 外部キーをアプライアンスのフラッシュドライブにコピーします。
2. キーが.pfx 形式の場合は、まず PEM 形式に変換する必要があります。コマンドプロンプトで入力します。

```
convert ssl pkcs12 <output file> -import -pkcs12File <input .pfx
    file name> -password <password>
<!--NeedCopy-->
```


3. コマンドプロンプトで次のコマンドを入力して、外部キーを FIPS キーとしてインポートし、設定を確認します。

```
import ssl fipsKey <fipsKeyName> -key <string> -informPEM
show ssl fipskey<fipsKeyName>
<!--NeedCopy-->
```

例:

```
convert ssl pkcs12 iis.pem -password 123456 -import -pkcs12File iis.pfx

import fipskey Key-FIPS-2 -key iis.pem -inform PEM

show ssl fipskey key-FIPS-2

FIPS Key Name: Key-FIPS-2 Modulus: 0   Public Exponent: F4 (Hex value 0
   x10001)
<!--NeedCopy-->
```

GUI を使用して外部キーを **FIPS** キーとしてインポートする

1. キーが.pfx 形式の場合は、まず PEM 形式に変換する必要があります。

   a) [トラフィック管理**] > [SSL**] に移動します。
   b) 詳細ウィンドウの [ツール] で、[**PKCS #12** のインポート] をクリックします。
   c) [PKCS12 ファイルのインポート] ダイアログボックスで、次のパラメータを設定します。

   - 出力ファイル名 *
   - PKCS12 ファイル名 *-.pfx ファイル名を指定します。
   - パスワードのインポート *
   - エンコード形式

   * 必須パラメータ

2. [トラフィック管理**] > [SSL] > [FIPS**] に移動します。
3. 詳細ウィンドウの [FIPS キー] タブで、[インポート] をクリックします。
4. [FIPS キーとしてインポート] ダイアログボックスで、[PEM ファイル] を選択し、次のパラメータの値を設定します。

   - FIPS キー名 *
   - Key File Name*-デフォルト以外の場所にファイルを配置するには、完全パスを指定するか、「参照」(Browse) をクリックして場所に移動します。

   * 必須パラメータ

5. [インポート] をクリックし、[閉じる] をクリックします。
6. [FIPS キー] タブで、インポートした FIPS キーに対して表示される設定が正しいことを確認します。

HA セットアップのアプライアンスで FIPS を構成する

1 つの HA ペアに 2 つのアプライアンスを FIPS アプライアンスとして設定できます。

前提条件

- 両方のアプライアンスで Hardware Security Module (HSM) を構成する必要があります。詳細については、「HSM の設定」を参照してください。
- GUI を使用する場合、アプライアンスで HA セットアップ済みであることを確認します。HA セットアップの構成の詳細については、「高可用性」を参照してください。

> 注:
>
> この手順には、構成ユーティリティ（GUI）を使用することをお勧めします。コマンドライン（CLI）を使用する場合は、手順に記載されている手順に注意深く従ってください。ステップの順序を変更したり、誤った入力ファイルを指定したりすると、アプライアンスの再起動を必要とする不整合が発生する可能性があります。また、CLI を使用する場合、`create ssl fipskey`コマンドはセカンダリノードに伝播されません。2 つの異なる FIPS アプライアンスでモジュラスサイズと指数に同じ入力値を指定してコマンドを実行すると、生成されるキーは同じではありません。いずれかのノードで FIPS キーを作成し、もう一方のノードに転送します。ただし、構成ユーティリティを使用して HA セットアップで FIPS アプライアンスを構成すると、作成した FIPS キーが自動的にセカンダリノードに転送されます。FIPS キーを管理および転送するプロセスは、セキュア情報管理 (SIM) と呼ばれます。

重要： HA セットアップは 6 分以内に完了する必要があります。いずれかの手順で手順が失敗した場合は、次の操作を行います。

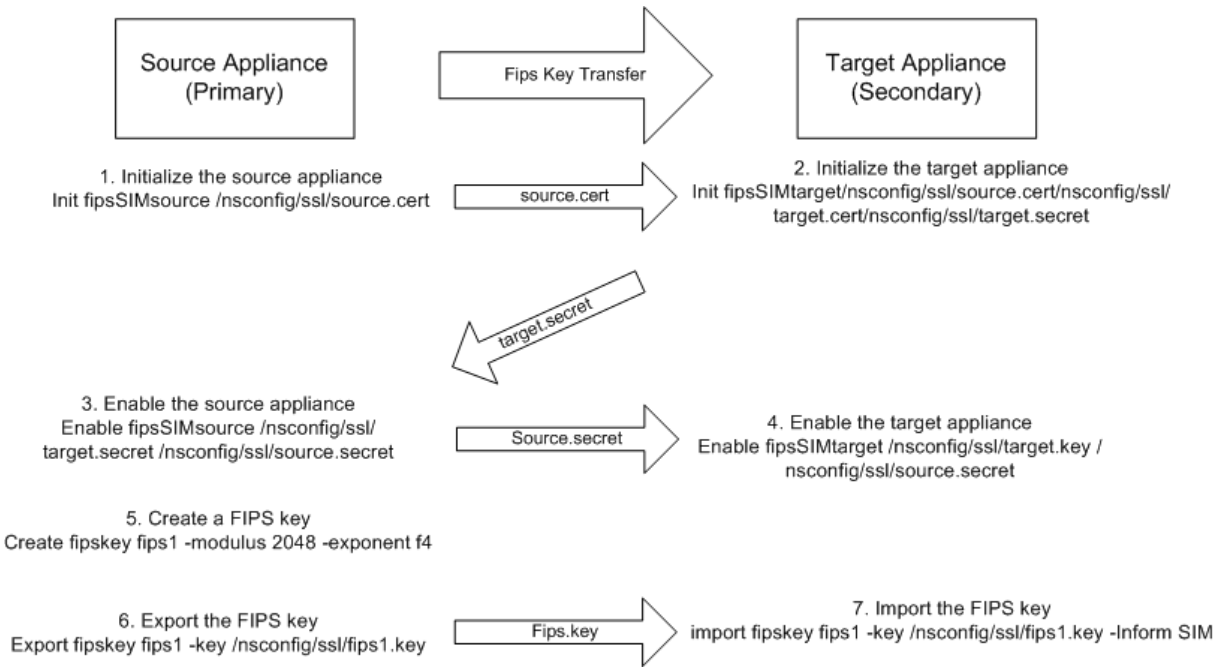
> 1. アプライアンスを再起動するか、10 分間待ちます。
> 2. プロシージャで作成されたすべてのファイルを削除します。
> 3. HA セットアップ手順を繰り返します。
>
> 既存のファイル名を再利用しないでください。

次の手順では、アプライアンス A がプライマリノード、アプライアンス B がセカンダリノードです。

CLI を使用して HA セットアップのアプライアンスで FIPS を構成する

次の図は、CLI で転送プロセスをまとめたものです。

図 1：FIPS キーサマリーを転送する



1. アプライアンス **A** で、PuTTY などの SSH クライアントを使用してアプライアンスへの SSH 接続を開きます。
2. 管理者の資格情報を使用して、アプライアンスにログオンします。
3. アプライアンス A をソースアプライアンスとして初期化します。コマンドプロンプトで入力します。

```
1 init ssl fipsSIMsource <certFile>
2 <!--NeedCopy-->
```

例:

```
init fipsSIMsource /nsconfig/ssl/nodeA.cert
```

4. この<certFile>ファイルをアプライアンス B の /nconfig/ssl フォルダにコピーします。

例:

```
scp /nsconfig/ssl/nodeA.cert nsroot@198.51.100.10:/nsconfig/ssl
```

5. アプライアンス **B** で、PuTTY などの SSH クライアントを使用してアプライアンスへの SSH 接続を開きます。
6. 管理者の資格情報を使用して、アプライアンスにログオンします。
7. アプライアンス B をターゲットアプライアンスとして初期化します。コマンドプロンプトで入力します。

```
1 init ssl fipsSIMtarget <certFile> <keyVector> <targetSecret>
2 <!--NeedCopy-->
```

例:

```
init fipsSIMtarget /nsconfig/ssl/nodeA.cert /nsconfig/ssl/nodeB.
key /nsconfig/ssl/nodeB.secret
```


8. この<targetSecret>ファイルをアプライアンス A にコピーします。

   例:

   ```
   scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.20:/
   nsconfig/ssl
   ```

9. アプライアンス **A** で、アプライアンス A をソースアプライアンスとして有効にします。コマンドプロンプトで入力します。

   ```
   1 enable ssl fipsSIMSource <targetSecret> <sourceSecret>
   2 <!--NeedCopy-->
   ```

   例:

   ```
   enable fipsSIMsource /nsconfig/ssl/nodeB.secret /nsconfig/ssl/
   nodeA.secret
   ```

10. この<sourceSecret>ファイルをアプライアンス B にコピーします。

    例:

    ```
    scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.10:/
    nsconfig/ssl
    ```

11. アプライアンス **B** で、アプライアンス B をターゲットアプライアンスとして有効にします。コマンドプロンプトで入力します。

    ```
    1 enable ssl fipsSIMtarget <keyVector> <sourceSecret>
    2 <!--NeedCopy-->
    ```

    例:

    ```
    enable fipsSIMtarget /nsconfig/ssl/nodeB.key /nsconfig/ssl/nodeA.
    secret
    ```

12. アプライアンス **A** で、FIPS キーの作成の説明に従って FIPS キーを作成します。
13. FIPS キーのエクスポートの説明に従って、FIPS キーをアプライアンスのハードディスクにエクスポートします。
14. SCP などの安全なファイル転送ユーティリティを使用して、セカンダリアプライアンスのハードディスクに FIPS キーをコピーします。
15. アプライアンス **B** で、FIPS キーのインポートの説明に従って、FIPS キーをハードディスクからアプライアンスの HSM にインポートします。

GUI を使用して **HA** セットアップのアプライアンスで **FIPS** を構成する

1. ソースアプライアンスとして設定するアプライアンスで、トラフィック管理 **> SSL > FIPS** に移動します。
2. 詳細ウィンドウの [FIPS 情報] タブで、[**SIM** を有効にする] をクリックします。

3. [**HA** ペアの **SIM** を有効にする] ダイアログボックスの [証明書ファイル名] テキストボックスに、ファイル名を入力します。ファイル名には、ソースアプライアンス上の FIPS 証明書を保存する必要がある場所へのパスが含まれている必要があります。
4. [キーベクトルファイル名] テキストボックスに、ファイル名を入力します。ファイル名には、ソースアプライアンス上の FIPS キーベクトルを格納する必要がある場所へのパスが含まれている必要があります。
5. [ターゲットシークレットファイル名] テキストボックスに、ターゲットアプライアンス上のシークレットデータを格納する場所を入力します。
6. [ソースシークレットファイル名（Source Secret File Name）] テキストボックスに、ソースアプライアンス上のシークレットデータを格納する場所を入力します。
7. [セカンダリシステムのログイン資格情報] で、[ユーザー名] と [パスワード] の値を入力します。
8. [**OK**] をクリックします。これで、FIPS アプライアンスが HA モードに設定されます。

注**:** HA でアプライアンスを設定したら、FIPS キーの作成の説明に従って FIPS キーを作成します。FIPS キーは、プライマリからセカンダリアプライアンスに自動的に転送されます。

CLI を使用して証明書署名要求を作成する

コマンドプロンプトで入力します。

```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName<string>
     [-organizationUnitName <string>] [-localityName <string>] [-
     commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

例**:**

```
>create certreq f1.req  - fipsKeyName  f1 -countryName US  -stateName CA
     -organizationName Citrix -companyName Citrix -commonName ctx -
    emailAddress test@example.com
Done
<!--NeedCopy-->
```

CLI を使用してサーバー証明書を作成する

コマンドプロンプトで入力します。

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM ) {
 -PEMPassPhrase  }
```



```
3   ] [-days <positive_integer>] [-certForm ( DER | PEM )] [-CAcert <
       input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey <
       input_filename>][-CAkeyForm ( DER | PEM )] [-CAserial <
       output_filename>]
4 <!--NeedCopy-->
```

例:

```
1 create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-
     root.key -CAserial ns-root.srl -days 1000
2 Done
3 <!--NeedCopy-->
```

前の例では、アプライアンスのローカルルート CA を使用してサーバ証明書を作成します。

CLI を使用した証明書とキーのペアの追加

コマンドプロンプトで入力します。

```
1 add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
     string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>][-
     expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
     positive_integer>]] [-bundle ( YES | NO )]
2 <!--NeedCopy-->
```

例:

```
1 add certkey cert1 -cert f1.cert -fipsKey f1
2
3 <!--NeedCopy-->
```

FIPS キーとサーバー証明書を作成したら、汎用 SSL 設定を追加できます。展開に必要な機能を有効にします。サーバー、サービス、および SSL 仮想サーバーを追加します。証明書とキーのペアとサービスを SSL 仮想サーバーにバインドします。構成を保存します。

```
1  enable ns feature SSL LB
2
3  add server s1 10.217.2.5
4
5  add service sr1 s1 HTTP 80
6
7  add lb vserver v1 SSL 10.217.2.172 443
8
9  bind ssl vserver v1  - certkeyName cert1
10
11 bind lb vserver v1 sr1
12
13 saveconfig
14
15 <!--NeedCopy-->
```


これで、MPX 14000 FIPS アプライアンスの基本構成が完了しました。

セキュア HTTPS の構成の詳細については、「FIPS の構成」をクリックしてください。

セキュア RPC の構成の詳細については、[FIPS の構成] を初めてクリックしてください。

MPX 14000 FIPS アプライアンスのライセンスを更新する

このプラットフォームでライセンスを更新するには、2 回再起動する必要があります。

1. /nsconfig/licenseフォルダ内のライセンスを更新します。
2. アプライアンスを再起動します。
3. アプライアンスにログオンします。
4. アプライアンスを再起動します。
   注**:** 2 回目の再起動の前に、新しいコマンドを追加したり、設定を保存したり、システム状態をチェックしたりしないでください。
5. アプライアンスにログオンし、`show ssl fips` コマンドを実行して FIPS が初期化されていることを確認します。

MPX 14000 FIPS および SDX 14000 FIPS プラットフォームでのハイブリッド FIPS モードのサポート

> 注:
>
> この機能は、1 つのプライマリ FIPS カードと 1 つ以上のセカンダリカードを含む新しい MPX/SDX 14000 FIPS プラットフォームでのみサポートされます。VPX プラットフォームや、1 種類のハードウェアカードのみを含むプラットフォームではサポートされていません。

FIPS プラットフォームでは、セキュリティ上の理由から、非対称および対称の暗号化と復号化が FIPS カードで実行されます。ただし、FIPS カードでこのアクティビティの一部（非対称）を実行し、キーのセキュリティを損なうことなく、バルク暗号化と復号化（対称）を別のカードにオフロードできます。

新しい MPX/SDX 14000 FIPS プラットフォームには、1 枚のプライマリカードと 1 つ以上のセカンダリカードが含まれています。ハイブリッド FIPS モードを有効にすると、秘密キーがこのカードに保存されるため、プリマスターシークレット復号化コマンドがプライマリカードで実行されます。ただし、バルク暗号化と復号化はセカンダリカードにオフロードされます。このオフロードにより、非ハイブリッド FIPS モードおよび既存の MPX 9700/10500/12500/15000 FIPS プラットフォームと比較して、MPX/SDX 14000 FIPS プラットフォームでのバルク暗号化スループットが大幅に向上します。ハイブリッド FIPS モードを有効にすると、このプラットフォームでの 1 秒あたりの SSL トランザクションも向上します。

> 注:
>
> - ハイブリッド FIPS モードは、すべての暗号計算を FIPS 認定モジュール内で行う必要がある厳格な認証要件を満たすために、デフォルトで無効になっています。ハイブリッドモードを有効にして、バルク暗号化と復号化をセカンダリカードにオフロードします。

> - SDX 14000 FIPS プラットフォームでは、ハイブリッドモードを有効にする前に、まず VPX インスタンスに SSL チップを割り当てる必要があります。

CLI を使用してハイブリッド **FIPS** モードを有効にする

コマンドプロンプトで入力します。

```
set SSL parameter -hybridFIPSMode {
 ENABLED|DISABLED }


Arguments

hybridFIPSMode

When this mode is enabled, system will use additional crypto hardware
   to accelerate symmetric crypto operations.

Possible values: ENABLED, DISABLED

Default value: DISABLED
<!--NeedCopy-->
```

例**:**

```
    set SSL parameter -hybridFIPSMode ENABLED
    show SSL parameter
    Advanced SSL Parameters
    -----------------------
    . . . . . .
    Hybrid FIPS Mode    : ENABLED
    . . . . . .

<!--NeedCopy-->
```

GUI を使用してハイブリッド **FIPS** モードを有効にする

1. [トラフィック管理] **> [SSL**] に移動します。
2. 詳細ウィンドウの [設定] で、[**SSL** の詳細設定の変更] をクリックします。
3. [**SSL** の詳細設定の変更] ダイアログボックスで、[ハイブリッド **FIPS** モード] を選択します。

制限事項:

1. 再ネゴシエーションはサポートされていません。
2. SDX 14000 プラットフォームの`stat ssl parameter`コマンドは、正しいセカンダリカードの使用率を表示しません。常に 0.00% の使用率が表示されます。


```
stat ssl

SSL Summary
# SSL cards present 1
# SSL cards UP  1
# Secondary SSL cards present   4
# Secondary SSL cards UP    4
SSL engine status      1
SSL sessions (Rate)     963
Secondary card utilization (%)     0.00
<!--NeedCopy-->
```

SDX 14000 FIPS アプライアンス

August 15, 2023

> 注:
>
> NetScaler のダウンロードページの「NetScaler リリース 12.1-FIPS」および「NetScaler リリース 12.1-ndCPP」にリストされているファームウェアのバージョンは、MPX 14000 FIPS または SDX 14000 FIPS プラットフォームではサポートされていません。これらのプラットフォームでは、ダウンロードページで入手できる他の最新の NetScaler ADC ファームウェアバージョンを使用できます。

NetScaler SDX アプライアンスは、複数の仮想 NetScaler ADC インスタンスをプロビジョニングおよび管理できるマルチテナントプラットフォームです。SDX アプライアンスは、単一の管理者がアプライアンスを構成および管理し、各ホストされたインスタンスの管理をテナントに委任できるようにすることで、クラウドコンピューティングおよびマルチテナンシーの要件に対応します。

NetScaler SDX 14030/14060/14080 FIPS アプライアンスは、FIPS 機能を備えた SDX アプライアンスの機能を提供します。FIPS 140-2 レベル 3 仕様（リリース 12.0 ビルド 56.x 以降）に準拠するように設計された、改ざん防止（改ざん防止）暗号化モジュール Cavium CNN3560-NFBE-G が装備されています。クリティカルセキュリティパラメータ (CSP) は、主にサーバの秘密キーで、暗号化モジュール内で安全に保存および生成されます。このモジュールは、ハードウェアセキュリティモジュール (HSM) とも呼ばれます。CSP は、HSM の境界外でアクセスされることはありません。スーパーユーザー (`nsroot`) だけが、HSM 内に格納されているキーに対して操作を実行できます。

NetScaler SDX 14030/14060/14080 FIPS アプライアンスには、63 コアの FIPS HSM モジュールが 1 つ含まれています。FIPS HSM モジュールは、最大 32 個のパーティションまでパーティション化できます。SDX 管理者は、専用のキーストレージ、暗号化リソース、および暗号化 SSL FIPS コアの数を各パーティションに割り当てることができます。パーティションに割り当てられたキーとリソースは専用で安全であり、他のパーティションはそれらにアクセスしたり共有したりすることはできません。

作成した FIPS HSM パーティションは、インスタンスのプロビジョニング時、または後でインスタンスを編集して VPX インスタンスに割り当てたり、アタッチしたりできます。作成され、インスタンスにアタッチされた FIPS パー

ティションは、そのインスタンスの仮想 HSM モジュールのように動作します。

SDX 14030/14060/14080 FIPS アプライアンスの VPX インスタンスには FIPS 仮想機能 (VF) パーティションが割り当てられ、分離された FIPS 仮想カードまたは HSM として扱われます。したがって、VPX インスタンス内で FIPS パーティションを構成する手順は、MPX FIPS アプライアンスを構成する手順と似ています。コンプライアンスの詳細については、米国国立標準技術研究所 (NIST) の Web サイトにあるセキュリティポリシーの詳細を参照してください。

高可用性セットアップでの FIPS アプライアンスの構成については、「HA セットアップでの FIPS アプライアンスの設定」を参照してください。

> 重要
>
> 各キーには、プライベートキーとパブリックキーが含まれます。その結果、2 つのキースペースを占有します。したがって、キーの最大数は、キーストアサイズの半分未満の 1 つに制限されます。

SDX 14000 FIPS プラットフォームは、ハイブリッド FIPS モードをサポートしています。このモードでは、暗号化および復号化アクティビティの一部を FIPS 以外のカードにオフロードできます。詳細については、「ハイブリッド FIPS モード」を参照してください。

制限事項

August 15, 2023

1. SSLv3 プロトコルを使用した SSL 再ネゴシエーションは、SDX FIPS アプライアンスのバックエンドではサポートされていません。
2. 1024 ビットおよび 4096 ビットのキー、および指数値 3 はサポートされていません。
3. バックアップと復元はサポートされていません。
4. クラスターと管理ドメインはサポートされていません。
5. インスタンスにアタッチできる FIPS パーティションは 1 つだけです。
6. FIPS パーティションを持つインスタンスには、1 つの CPU コアのみを割り当てることができます。
7. FIPS パーティションまたは SSL コアのいずれかをインスタンスに割り当てることができますが、両方を割り当てることはできません。
8. 4096 ビットのサーバー証明書はサポートされていません。
9. 4096 ビットのクライアント証明書はサポートされていません (バックエンドサーバーでクライアント認証が有効になっている場合)。

用語

August 15, 2023

ゼロ化**:HSM** をリセットします。HSM のすべてのデータが削除されます。このステップは、HSM を初期化する前には必須です。

初期化:HSM 機能を設定します。NetScaler SDX FIPS アプライアンスは、FIPS-140-2 レベル 2 に準拠しています。チップを初期化した後でパーティションを作成できます。

キーストアサイズ: パーティションに保存できるキーの数。最大 102235 個のキーを指定できます。保存できるキーの最大数は、指定した数の半分未満です。たとえば、100 を指定した場合、作成できるキーは 49 個だけです。これは、キーの 1 つが 2 つのキーストアを消費する RSA キーペアであるためです。

暗号コア容量: パーティションに割り当てられた暗号コアの数。最大 63 コアまで使用できます。

SSL コンテキスト: パーティション上に作成できる同時 SSL 接続の数。

HSM を初期化する

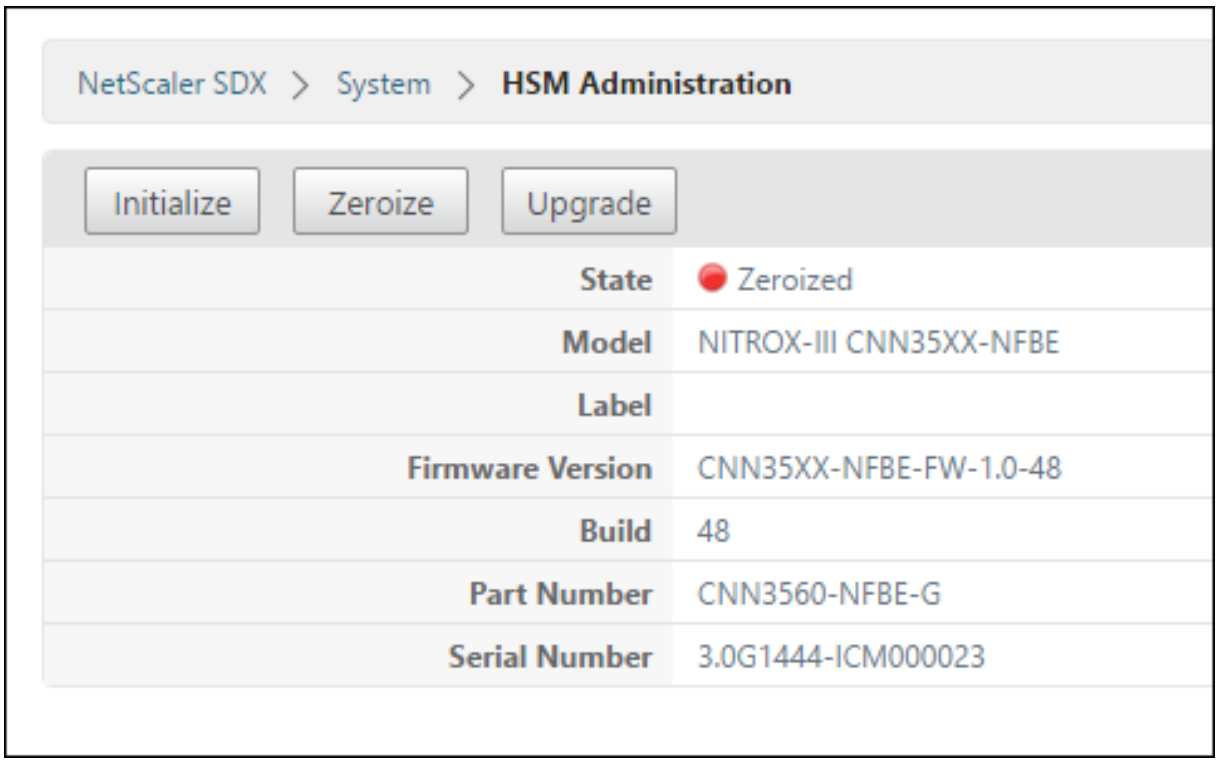
August 15, 2023

HSM を初期化する前に、まず HSM をゼロ化する必要があります。

管理サービスを使用して **HSM** をゼロ化

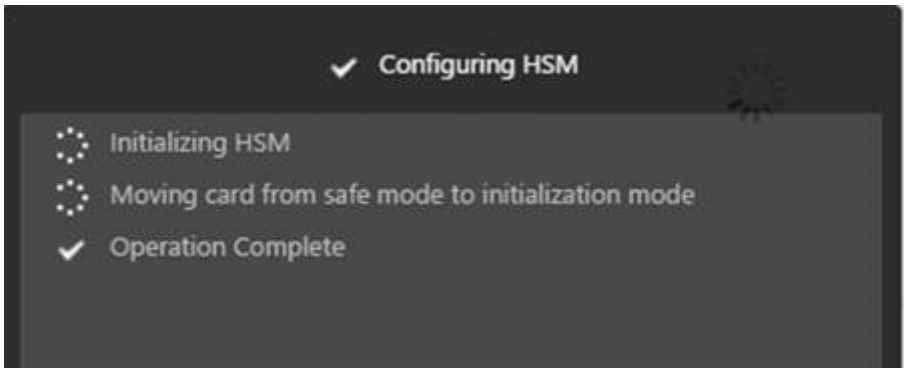
1. ブラウザを開き、アプライアンスにログオンします。
2. **[**構成**]** タブで、**[**システム**] > [HSM** 管理**]** に移動し、詳細プレーンで **[**ゼロ化**]** をクリックします。

すべてのデータは FIPS チップから消去され、状態は「ゼロ化」と表示されます。以前に作成された HSM パーティションはすべて削除されます。

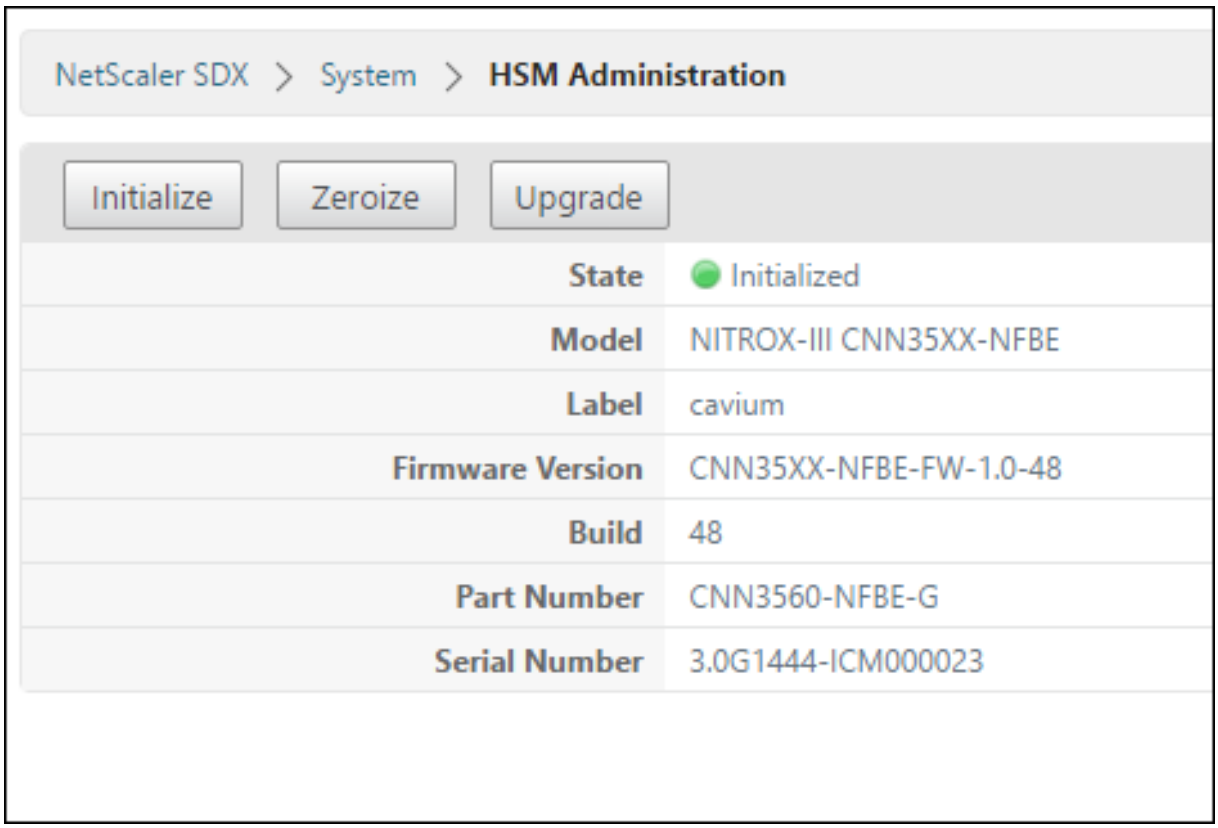


管理サービスを使用して **HSM** を初期化します

1. **[**構成**]** タブで、**[**システム**] > [HSM** 管理**]** に移動し、詳細プレーンで **[**初期化**]** をクリックします。
2. 新しいユーザー名を入力し、パスワードを指定して、「**OK**」をクリックします。



カードの状態は「初期化済み」と表示されます。



パーティションを作成する

January 11, 2024

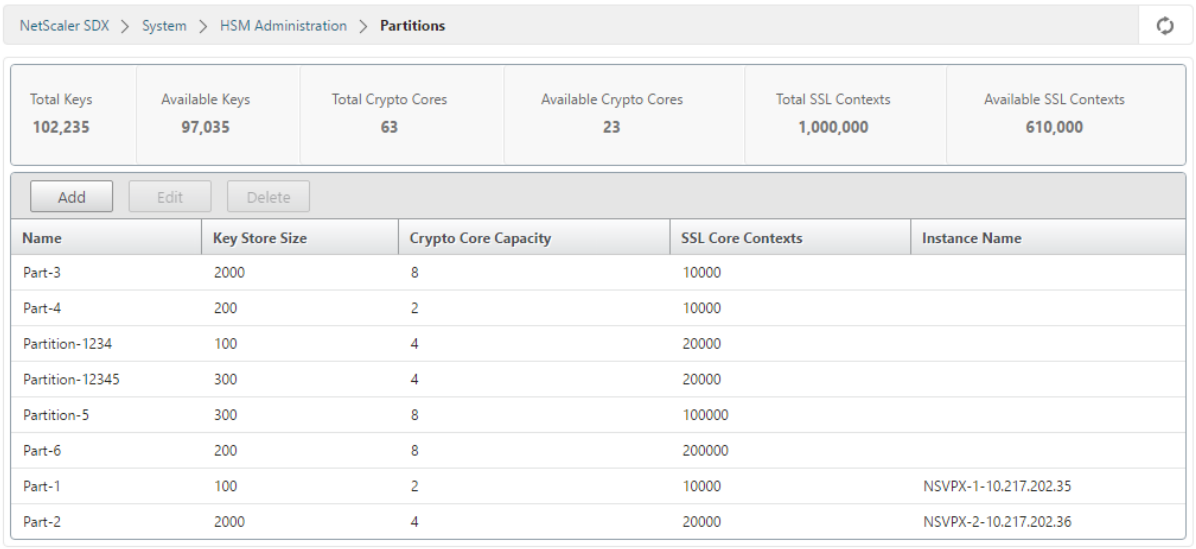
さまざまなテナント用のパーティションを作成し、各パーティションの暗号化リソースを指定します。各インスタンスには 1 つのパーティションが割り当てられ、1 つのパーティションは 1 つのインスタンスにのみ割り当てることができます。インスタンスを削除すると、そのインスタンスに割り当てられているパーティションも削除されます。その結果、パーティションデータも削除され、保護されていないままになったり、後でアクセス可能になったりすることはありません。キーの数と SSL コンテキストの割り当ては、アプリケーションによって異なります。割り当てるコアの数については、NetScaler データシートを参照してください。

> 重要
>
> HSM パーティションにキーストアサイズとコアを割り当てた後は、実行時に変更することはできません。まず、インスタンスからパーティションをデタッチします。

管理サービスを使用してパーティションを作成する

1. **[設定]** タブで **[** システム] > [HSM 管理] > [パーティション] に移動し、詳細ペインで [追加] をクリックします。**
2. パーティションの名前と、このパーティションに割り当てるリソースを指定します。
3. [**OK**] をクリックします。

概要ページには、作成されたすべてのパーティションが表示されます。一部のパーティションにはインスタンスが割り当てられていますが、一部は空きパーティションです。



新しいインスタンスのプロビジョニングまたは既存のインスタンスの変更とパーティションの割り当て

January 11, 2024

パーティションを作成したら、それらをインスタンスに割り当てる必要があります。

> 重要:
>
> - インスタンスにアタッチできる FIPS パーティションは 1 つだけです。
> - FIPS パーティションを持つインスタンスには、1 つの CPU コアのみを割り当てることができます。

新しいインスタンスをプロビジョニングするか、既存のインスタンスを変更します

1. [構成] タブで、[**NetScaler**] > [インスタンス] に移動し、インスタンスを追加または変更します。

2. [**Enable FIPS**] を選択し、[**Partitions**] リストから、このインスタンスにアタッチするパーティションを選択します。

GUI または CLI を使用して、パーティションがインスタンスにアタッチされていることを確認できます。

GUI で、[システム] > [**HSM** 管理] > [パーティション] に移動します。パーティションにアタッチされたインスタンス名が表示されます。

FIPS パーティションの割り当てを解除するには、[**NetScaler**] > [インスタンス] に移動します。インスタンスを編集し、[**Enable FIPS**] チェックボックスをオフにします。

CLI のコマンドプロンプトで、次のコマンドを入力します。

```
show fips

FIPS Card is not configured
 Done
<!--NeedCopy-->
```

次の出力が表示された場合は、デバッグのトラブルシューティングのセクションを参照してください。

エラー: 操作は許可されていません。システムに FIPS カードがありません。

> 注:
>
> パーティションが既存の VPX インスタンスのいずれかからデタッチされると、そのパーティションのデータはクリアされます。その結果、現在の設定 (FIPS キーなど) はすべて失われます。新しいまたは以前にバインドされた VPX インスタンスにパーティションをデタッチまたは再アタッチした後、安全な接続にパーティションを使用する前に、HSM の構成の手順に従ってパーティションを初期化する必要があります 。
>
> この間（パーティションがデタッチまたは再アタッチされた後)、対応する VPX インスタンスには、HTTP を使用して GUI を介して、SSH を使用して CLI を介してアクセスできます。

SDX 14030/14060/14080 FIPS アプライアンスでインスタンスの HSM を構成する

August 15, 2023

まず FIPS カードの状態をチェックして、ドライバが正しくロードされたことを確認してから、カードを初期化します。

コマンドプロンプトで入力します。

```
show fips

FIPS Card is not configured

Done
<!--NeedCopy-->
```


ドライバが正しくロードされていない場合は、「エラー: 操作は許可されていません-システムに FIPS カードがありません」というメッセージが表示されます。

FIPS カードの初期化

> 重要:
>
> /nsconfig/fips ディレクトリがアプライアンスに正常に作成されたことを確認します。

アプライアンスを 3 回目に再起動する前に、設定を保存しないでください。

FIPS カードを初期化するには、次の手順に従います。

1. FIPS カードをリセットします (reset fips)。
2. アプライアンスを再起動します (reboot)。
3. パーティション 0 と 1 にはセキュリティ担当者のパスワードを設定し、partition (set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -hsmLabel NSFIPS) にはユーザーパスワードを設定します。

   注:set または reset コマンドの実行には 60 秒以上かかります。
4. 設定を保存します (saveconfig)。
5. メインパーティション (master_pek.key) のパスワード暗号化キーが /nsconfig/fips/ ディレクトリに作成されたことを確認します。
6. アプライアンスを再起動します (reboot)。
7. FIPS カードが稼働していることを確認します（show fips)。

CLI を使用して FIPS カードを初期化します

コマンドプロンプトで、次のコマンドを入力します。

```
reset fips

reboot

set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -
    hsmLabel <string>
<!--NeedCopy-->
```

> 注: **set fips** コマンドを実行すると、次のメッセージが表示されます。


```
This command will erase all data on the FIPS card. You must save the configuration (saveconfig) after executing this command. [Note: On MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by default, and the -initHSM Level-2 option is internally converted to Level-3]  Do you want to continue?(Y/N)y

saveconfig

reboot

show fips
<!--NeedCopy-->
```

例:

```
reset fips

Done

reboot

set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS

This command will erase all data on the FIPS card. You must save the configuration (saveconfig) after executing this command. [Note: On MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by default, and the -initHSM Level-2 option is internally converted to Level-3]  Do you want to continue?(Y/N)y

Done

saveconfig

Done

reboot

show fips

    FIPS HSM Info:
    HSM Label : NSFIPS
    Initialization : FIPS-140-2 Level-2
    HSM Serial Number : 3.0G1532-ICM000228
    HSM State : 2
    HSM Model : NITROX-III CNN35XX-NFBE
    Hardware Version : 0.0-G
    Firmware Version : 1.0
    Firmware Build : NFBE-FW-1.0-48
    Max FIPS Key Memory : 1000
    Free FIPS Key Memory : 1000
    Total SRAM Memory : 557396
    Free SRAM Memory : 238088
    Total Crypto Cores : 4
```



```
35      Enabled Crypto Cores : 4
36  Done
37  <!--NeedCopy-->
```

SDX 14030/14060/14080 FIPS アプライアンスでインスタンスの FIPS キーを作成する

August 15, 2023

インスタンスに FIPS キーを作成するか、既存の FIPS キーをインスタンスにインポートできます。SDX 14030/14060/14080 FIPS アプライアンスは、2048 ビットと 3072 ビットのキーと F4 の指数値のみをサポートします。PEM キーの場合、指数は不要です。FIPS キーが正しく作成されていることを確認します。証明書署名要求とサーバー証明書を作成します。最後に、証明書とキーのペアをインスタンスに追加します。

> 注:
>
> 1024 ビットおよび 4096 ビットのキー、および指数値 3 はサポートされていません。

CLI を使用して FIPS キーを作成する

コマンドプロンプトで入力します。

```
1  create ssl fipsKey <fipsKeyName> -keytype ( RSA | ECDSA ) [-exponent (3
       | F4 )] [-modulus <positive_integer>] [-curve ( P_256 | P_384 )]
2  <!--NeedCopy-->
```

例:

```
1  create fipsKey f1 -keytype RSA -modulus 2048 -exponent F4
2
3  Done
4
5  show ssl fipskey ddvws
6
7  FIPS Key Name: f1  Key Type: RSA Modulus: 2048   Public Exponent: F4 (
       Hex: 0x10001)
8
9  Done
10 <!--NeedCopy-->
```

CLI を使用して FIPS キーをインポートする

コマンドプロンプトで入力します。


```
import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-
    wrapKeyName <string>] [-iv<string>] [-exponent F4 ]
<!--NeedCopy-->
```

例:

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4
Done
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM
Done
<!--NeedCopy-->
```

show fipskey コマンドを実行して、**FIPS** キーが正しく作成またはインポートされていることを確認します。

```
show fipskey
1)      FIPS Key Name: Key-FIPS-2
Done
<!--NeedCopy-->
```

CLI を使用して証明書署名要求を作成する

コマンドプロンプトで入力します。

```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName<string>
     [-organizationUnitName <string>] [-localityName <string>] [-
     commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

例:

```
create certreq f1.req  - fipsKeyName f1 -countryName US -stateName CA -
    organizationName Citrix -companyName Citrix -commonName ctx -
    emailAddress test@example.com`
`Done
<!--NeedCopy-->
```

CLI を使用してサーバー証明書を作成する

コマンドプロンプトで入力します。

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM ) {
 -PEMPassPhrase  }
```



```
3   ] [-days <positive_integer>] [-certForm ( DER | PEM )] [-CAcert <
       input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey <
       input_filename>] [-CAkeyForm ( DER | PEM )] [-CAserial <
       output_filename>]
4 <!--NeedCopy-->
```

例:

```
1 create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-
     root.key -CAserial ns-root.srl -days 1000
2 Done
3 <!--NeedCopy-->
```

前の例では、アプライアンスのローカルルート CA を使用してサーバ証明書を作成します。

CLI を使用した証明書とキーのペアの追加

コマンドプロンプトで入力します。

```
1 add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
     string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>]
     [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod   <
     positive_integer>]] [-bundle ( YES | NO )]
2 <!--NeedCopy-->
```

例:

```
1 add certkey cert1 -cert f1.cert -fipsKey f1
2 Done
3 <!--NeedCopy-->
```

FIPS キーとサーバー証明書を作成したら、汎用 SSL 設定を追加できます。展開に必要な機能を有効にします。サーバー、サービス、および SSL 仮想サーバーを追加します。証明書とキーのペアとサービスを SSL 仮想サーバにバインドし、設定を保存します。

```
1  enable ns feature SSL LB
2  Done
3  add server s1 10.217.2.5
4  Done
5  add service sr1 s1 HTTP 80
6  Done
7  add lb vserver v1 SSL 10.217.2.172 443
8  Done
9  bind ssl vserver v1  - certkeyName cert1
10 Done
11 bind lb vserver v1 sr1
12 Done
13 saveconfig
14 Done
15 <!--NeedCopy-->
```


セキュア HTTPS とセキュア RPC の設定については、ここをクリックしてください。

VPX インスタンスの FIPS HSM ファームウェアのアップグレード

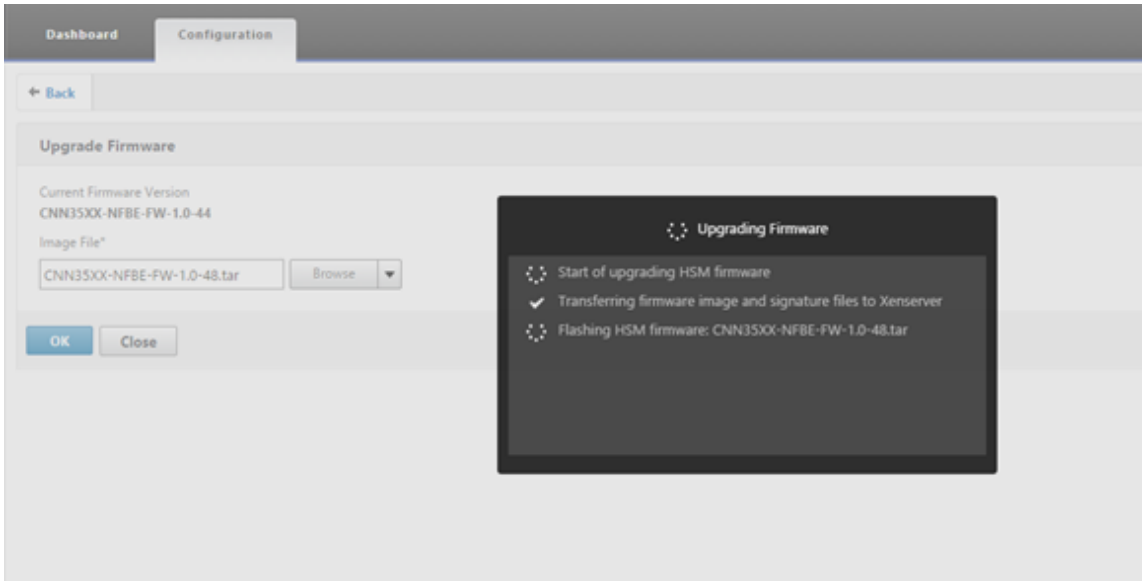
January 11, 2024

> 注:
>
> このアップグレードは SDX 14000 アプライアンスの FIPS カードに適用されます。

FIPS HSM ファームウェアのアップデートは随時リリースされます。NetScaler のダウンロードページから最新のファームウェアをダウンロードし、アプライアンスにアップロードします。アップグレードプロセスが完了するまでに最大 10 分かかる場合があります。インスタンスはアップグレード後に再起動されます。

FIPS HSM ファームウェアのアップグレード

1. [システム] > [HSM 管理] > [ファームウェアイメージ] に移動します。
2. [アップロード] を選択します。
3. ファームウェアイメージを含むフォルダに移動し、ファイルを選択します。
4. [システム] > [HSM 管理] に移動し、[ファームウェアのアップグレード] を選択します。
5. アップグレードするファームウェアイメージを選択し、**OK** をクリックします。



Thales Luna Network ハードウェアセキュリティモジュールのサポート

August 15, 2023

FIPS 以外の NetScaler ADC アプライアンスは、サーバーの秘密キーをハードディスクに保存します。FIPS アプライアンスでは、キーはハードウェアセキュリティモジュール (HSM) と呼ばれる暗号化モジュールに格納されます。HSM にキーを保存すると、物理攻撃やソフトウェア攻撃からキーを保護できます。さらに、キーは FIPS 承認の特殊な暗号で暗号化されます。

NetScaler MPX/SDX 14000 FIPS アプライアンスのみが FIPS カードをサポートします。FIPS のサポートは、他の MPX/SDX アプライアンスまたは NetScaler ADC VPX アプライアンスでは使用できません。この制限は、MPX/SDX 14000 FIPS アプライアンスを除くすべての NetScaler ADC MPX、SDX、VPX アプライアンスで Thales Luna ネットワーク HSM をサポートすることで解決されます。

> 注:
>
> Intel Coleto および Intel Lewisburg SSL チップベースのプラットフォームのサポートに記載されているアプライアンスのサポートは 、リリース 13.1 ビルド 33.x 以降でご利用いただけます。

Thales Luna ネットワーク HSM は、重要な暗号化キーを保護し、幅広いセキュリティアプリケーションにわたって機密性の高い暗号化操作を高速化するように設計されています。

サポートされているバージョンのマトリックス

NetScaler バージョン	ソフトウェアアプライアンスバージョン	ファームウェアバージョン	クライアントのバージョン
11.1, 12.0, 12.1	5.2.3-1	6.2.1	6.0.0
11.1, 12.0, 12.1	6.2.2-5	6.10.9	6.2.2
13.0	7.2.0-220	7.0.3	7.2.2 (7.2.0-220)
13.1	7.2.0-220	7.0.3	10.3.0

前提条件

August 15, 2023

NetScaler ADC で Thales Luna ネットワーク HSM を使用する前に、次の前提条件が満たされていることを確認してください。

- Thales Luna ネットワーク HSM がネットワークにインストールされ、すぐに使用でき、NetScaler ADC からアクセスできます。つまり、NSIP アドレスまたは SNIP アドレスが、HSM で許可されたクライアントとして追加されます。
- ライセンスは、HSM 上で必要な数のパーティションをサポートするために利用できます。
- Thales Luna ネットワーク HSM と NetScaler ADC は、ポート 1792 を介して相互に接続を開始できます。
- NetScaler リリース 11.1 以降を使用しています。
- NetScaler ADC アプライアンスには、FIPS キャビウムカードは含まれていません。

> 重要
>
> Thales Luna ネットワーク HSM は MPX 9700/10500/12500/15500 FIPS アプライアンスではサポートされていません。

ADC で Thales Luna クライアントを構成する

August 15, 2023

Thales Luna HSM を構成し、必要なパーティションを作成したら、クライアントを作成してパーティションに割り当てる必要があります。まず、NetScaler ADC で Thales Luna クライアントを構成し、Thales Luna クライアントと Thales Luna HSM の間のネットワークトラストリンク（NTL）を設定します。設定例については、付録に記載されています。

> 注:
>
> ソフトウェアバージョン 13.1 にアップグレードする場合は、Thales Luna クライアントバージョン 10.3.0 をインストールし、次の手順を実行する必要があります。

1. ディレクトリを /var/safenet に変更し、Thales Luna クライアントをインストールします。シェルプロンプトで、次のように入力します。

```
cd /var/safenet
<!--NeedCopy-->
```

Thales Luna クライアントバージョン 6.0.0 をインストールするには、次のように入力します。

```
install_client.sh -v 600
<!--NeedCopy-->
```

Thales Luna クライアントバージョン 6.2.2 をインストールするには、次のように入力します。

```
install_client.sh -v 622
<!--NeedCopy-->
```

Thales Luna クライアントバージョン 7.2.2 をインストールするには、次のように入力します。


```
install_client.sh -v 722
<!--NeedCopy-->
```

Thales Luna クライアントバージョン 10.3.0 をインストールするには、次のように入力します。

```
install_client.sh -v 1030
<!--NeedCopy-->
```

2. Thales Luna クライアント（ADC）と HSM の間の NTL を設定します。

   ‘/var/safenet/’ ディレクトリを作成したら、ADC で次のタスクを実行します。

   a) ディレクトリを ‘/var/safenet/config/’ に変更し、’safenet_config’ スクリプトを実行します。シェルプロンプトで、次のように入力します。

```
cd /var/safenet/config

sh safenet_config
<!--NeedCopy-->
```

   このスクリプトは「chrystoki.conf」ファイルを /etc/ ディレクトリにコピーします。また、’/usr/lib/’ ディレクトリにシンボリックリンク ‘libCryptoki2_64.so’ を生成します。

   b) ADC と Thales Luna HSM の間で証明書とキーを作成し、転送します。

   安全に通信するには、ADC と HSM が証明書を交換する必要があります。ADC で証明書とキーを作成し、HSM に転送します。HSM 証明書を ADC にコピーします。

   i) ディレクトリを /var/safenet/safenet/lunaclient/bin に変更します。

   ii) ADC で証明書を作成します。シェルプロンプトで、次のように入力します。

```
./vtl createCert -n <ip address of NetScaler>
<!--NeedCopy-->
```

   このコマンドはまた、証明書とキーパスを「/etc/chrystoki.conf」ファイルに追加します。

   iii) この証明書を HSM にコピーします。シェルプロンプトで、次のように入力します。

```
scp /var/safenet/safenet/lunaclient/cert/client/<ip address of NS
   >.pem <LunaSA_HSM account>@<IP address of Luna SA>
<!--NeedCopy-->
```

   iv) HSM 証明書を NetScaler にコピーします。シェルプロンプトで、次のように入力します。

```
scp <HSM account>@<HSM IP>:server.pem  /var/safenet/safenet/
   lunaclient/server_<HSM ip>.pem
<!--NeedCopy-->
```

3. NetScaler ADC をクライアントとして登録し、Thales Luna HSM 上のパーティションを割り当てます。

HSM にログオンし、クライアントを作成します。クライアント IP として NSIP を入力します。このアドレスは、HSM への証明書の転送元の ADC の IP アドレスである必要があります。クライアントが正常に登録されたら、パーティションを割り当てます。HSM で次のコマンドを実行します。

a) SSH を使用して Thales Luna HSM に接続し、パスワードを入力します。

b) Thales Luna HSM に NetScaler ADC を登録します。クライアントは HSM 上に作成されます。IP アドレスはクライアントの IP アドレスです。つまり、NSIP アドレスです。

プロンプトで、次のように入力します。

```
client register  -client <client name> -ip <NetScaler ip>
<!--NeedCopy-->
```

c) パーティションリストからクライアントにパーティションを割り当てます。使用可能なパーティションを表示するには、次のように入力します。

```
<luna_sh> partition list
<!--NeedCopy-->
```

このリストからパーティションを割り当てます。タイプ:

```
<lunash:> client assignPartition -client <Client Name> -par <
   Partition Name>
<!--NeedCopy-->
```

4. HSM とその証明書を NetScaler ADC に登録します。

   ADC で、ディレクトリを「/var/safenet/safenet/lunaclient/bin」に変更し、シェルプロンプトで次のように入力します。

   ```
   ./vtl addserver -n <IP addr of HSM> -c /var/safenet/safenet/
      lunaclient/server_<HSM_IP>.pem
   <!--NeedCopy-->
   ```

   ADC に登録されている HSM を削除するには、次のように入力します。

   ```
   ./vtl deleteServer -n <HSM IP> -c <cert path>
   <!--NeedCopy-->
   ```

   ADC に設定されている HSM サーバーを一覧表示するには、次のように入力します。

   ```
   ./vtl listServer
   <!--NeedCopy-->
   ```

   > 注:
   >
   > vtlを使用して HSM を削除する前に、その HSM のすべてのキーがアプライアンスから手動で削除されていることを確認してください。HSM サーバーを削除すると、HSM キーは削除できません。

5. ADC と HSM の間のネットワーク信頼リンク (NTL) 接続を確認します。シェルプロンプトで、次のように入力します。


```
./vtl verify
<!--NeedCopy-->
```

検証に失敗した場合は、すべての手順を確認します。エラーは、クライアント証明書の IP アドレスが正しくないことが原因です。

6. 構成を保存します。

   前の手順では、「/etc/chrystoki.conf」設定ファイルを更新します。このファイルは、ADC が起動すると削除されます。構成をデフォルト構成ファイルにコピーします。このファイルは、ADC の再起動時に使用されます。

   シェルプロンプトで、次のように入力します。

```
root@ns# cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

   推奨される方法は、Thales Luna 関連の設定が変更されるたびにこのコマンドを実行することです。

7. Thales Luna ゲートウェイプロセスを開始します。

   シェルプロンプトで、次のように入力します。

```
sh /var/safenet/gateway/start_safenet_gw
<!--NeedCopy-->
```

8. 起動時に Gateway デーモンの自動起動を設定します。

   この ADC で Thales Luna HSM が設定されていることを示す「safenet_is_reglarbed」ファイルを作成します。ADC が再起動し、このファイルが見つかるたびに、Gateway が自動的に起動します。

   シェルプロンプトで、次のように入力します。

```
touch /var/safenet/safenet_is_enrolled
<!--NeedCopy-->
```

9. NetScaler アプライアンスを再起動します。コマンドプロンプトで入力します。

```
reboot
<!--NeedCopy-->
```

ADC の高可用性セットアップで Thales Luna HSM を構成する

August 15, 2023

Thales Luna HSM をハイアベイラビリティ（HA）に設定することで、いずれかのデバイスをすべて使用できない場合でも、サービスが中断されないようにします。HA 設定では、各 HSM はアクティブ-アクティブモードで HA グループに加入します。HA セットアップの Thales Luna HSM は、すべてのグループメンバーのロードバランシングを

提供し、パフォーマンスと応答時間を向上させながら、高可用性サービスを保証します。詳細については、Thales Luna のセールスおよびサポートにお問い合わせください。

前提条件**:**

- 最低 2 台の Thales Luna HSM デバイス。HA グループ内のすべてのデバイスには、PED（信頼できるパス）認証またはパスワード認証が必要です。HA グループでのトラステッドパス認証とパスワード認証の組み合わせはサポートされていません。
- 各 HSM デバイスのパーティションには、ラベル (名前) が異なっていても同じパスワードが必要です。
- HA 内のすべてのパーティションをクライアント（NetScaler ADC アプライアンス）に割り当てる必要があります。

ADC での Thales Luna クライアントの設定の説明に従って、ADC で Thales Luna クライアントを構成した後、次の手順を実行して HA で Thales Luna HSM を設定します。

1. NetScaler のシェルプロンプトで、`lunacm`（/usr/safenet/lunaclient/bin）を起動します。

   例**:**

   ```
   root@ns# cd /var/safenet/safenet/lunaclient/bin/

   root@ns# ./lunacm
   <!--NeedCopy-->
   ```

2. パーティションのスロット ID を識別します。使用可能なスロット (パーティション) を一覧表示するには、次のように入力します。

   ```
   lunacm:> slot list
   <!--NeedCopy-->
   ```

   例**:**

   ```
      Slot Id ->              0
      HSM Label ->            trinity-p1
      HSM Serial Number ->    481681014
      HSM Model ->            LunaSA 6.2.1
      HSM Firmware Version -> 6.10.9
      HSM Configuration ->    Luna SA Slot (PED) Signing With
         Cloning Mode
      HSM Status ->           OK

      Slot Id ->              1
      HSM Label ->            trinity-p2
      HSM Serial Number ->    481681018
      HSM Model ->            LunaSA 6.2.1
      HSM Firmware Version -> 6.10.9
      HSM Configuration ->    Luna SA Slot (PED) Signing With
         Cloning Mode
      HSM Status ->           OK

       Slot Id ->              2
   ```



```
18       HSM Label ->             neo-p1
19       HSM Serial Number ->     487298014
20       HSM Model ->             LunaSA 6.2.1
21       HSM Firmware Version -> 6.10.9
22       HSM Configuration ->     Luna SA Slot (PED) Signing With
             Cloning Mode
23      HSM Status ->            OK
24
25      Slot Id ->               3
26      HSM Label ->             neo-p2
27      HSM Serial Number ->     487298018
28      HSM Model ->             LunaSA 6.2.1
29      HSM Firmware Version -> 6.10.9
30      HSM Configuration ->     Luna SA Slot (PED) Signing With
            Cloning Mode
31      HSM Status ->            OK
32
33      Slot Id ->               7
34      HSM Label ->             hsmha
35      HSM Serial Number ->     1481681014
36      HSM Model ->             LunaVirtual
37      HSM Firmware Version -> 6.10.9
38      HSM Configuration ->     Luna Virtual HSM (PED) Signing With
            Cloning Mode
39      HSM Status ->            N/A - HA Group
40
41      Slot Id ->               8
42      HSM Label ->             newha
43      HSM Serial Number ->     1481681018
44      HSM Model ->             LunaVirtual
45      HSM Firmware Version -> 6.10.9
46      HSM Configuration ->     Luna Virtual HSM (PED) Signing With
            Cloning Mode
47      HSM Status ->            N/A - HA Group
48
49      Current Slot Id: 0
50  <!--NeedCopy-->
```

3. HA グループを作成します。最初のパーティションはプライマリパーティションと呼ばれます。複数のセカンダリパーティションを追加できます。

```
1  lunacm:> hagroup createGroup -slot <slot number of primary
       partition> -label <group name> -password <partition password >
2
3  lunacm:> hagroup createGroup -slot 1 -label gp12 -password ******
4  <!--NeedCopy-->
```

4. セカンダリメンバー (HSM パーティション) を追加します。HA グループに追加するすべてのパーティションに対してこの手順を繰り返します。

```
1  lunacm:> hagroup addMember -slot <slot number of secondary
       partition to be added> -group <group name> -password <partition
        password>
```



```
2 <!--NeedCopy-->
```

コード:

```
1 lunacm:> hagroup addMember -slot 2 -group gp12 -password ******
2 <!--NeedCopy-->
```

5. HA 専用モードを有効にします。

```
1 lunacm:> hagroup HAOnly  - enable
2 <!--NeedCopy-->
```

6. アクティブリカバリモードを有効にします。

```
1 lunacm:.>hagroup recoveryMode  - mode active
2 <!--NeedCopy-->
```

7. 自動回復の間隔時間 (秒単位) を設定します。デフォルト値は 60 秒です。

```
1 lunacm:.>hagroup interval  - interval <value in seconds>
2 <!--NeedCopy-->
```

例:

```
1 lunacm:.>hagroup interval  - interval 120
2 <!--NeedCopy-->
```

8. リカバリの再試行回数を設定します。値が -1 の場合、再試行回数は無限です。

```
1 lunacm:> hagroup retry -count <xxx>
2 <!--NeedCopy-->
```

例:

```
1 lunacm:> hagroup retry -count 2
2 <!--NeedCopy-->
```

9. 設定を SafeNet `Chrystoki.conf` 設定ディレクトリにコピーします。

```
1 cp /etc/Chrystoki.conf /var/safenet/config/
2 <!--NeedCopy-->
```

10. ADC アプライアンスを再起動します。

```
1 reboot
2 <!--NeedCopy-->
```

HA で Thales Luna HSM を設定した後、ADC での詳細設定については、その他の ADC の設定を参照してください。

Other ADC configuration

August 15, 2023

1. HSM でキーを生成します。

   サードパーティのツールを使用して HSM にキーを作成します。

2. ADC に HSM キーを追加します。

   重要**:** # 文字は、キー名ではサポートされていません。キー名にこの文字が含まれている場合、キーのロード操作は失敗します。

   CLI を使用して **Thales Luna HSM** キーを追加するには、次の手順を実行します。

   コマンドプロンプトで入力します。

   ```
   1 add ssl hsmkey <KeyName> -hsmType SAFENET -serialNum <serial #> -
       password
   2 <!--NeedCopy-->
   ```

   各項目の意味は次のとおりです:

   -KeyName は、サードパーティのツールを使用して HSM で作成されたキーです。

   -SerialNum は、キーが生成される HSM 上のパーティションのシリアル番号です。

   注**:** 高可用性セットアップの HSM では、高可用性グループのシリアル番号を使用してください。

   -password は、キーが存在するパーティションのパスワードです。

   GUI を使用して **Thales Luna HSM** キーを追加するには、次の手順を実行します。

   [トラフィック管理] > [**SSL**] > [**HSM**] に移動し、HSM キーを追加します。HSM タイプを **SAFENET** として指定する必要があります。

3. ADC に証明書とキーのペアを追加します。まず、サードパーティのツールを使用して、キーに関連する証明書を生成します。次に、証明書を ADC の /nsconfig/ssl/ ディレクトリにコピーします。

   注**:** キーは HSM キーでなければなりません。

   CLI を使用して **ADC** に証明書キーペアを追加するには:

   コマンドプロンプトで入力します。

   ```
   1 add ssl certkey <CertkeyName> -cert <cert name> -hsmkey <KeyName>
   2 <!--NeedCopy-->
   ```

   GUI を使用して **ADC** に証明書キーペアを追加するには:

   a) [トラフィック管理] > [**SSL**] に移動します。

b)「はじめに」で、「証明書 **(HSM)** をインストール」を選択し、HSM キーを使用して証明書とキーのペアを作成します。

4. 仮想サーバーを作成し、証明書とキーのペアをこの仮想サーバーにバインドします。

仮想サーバーの作成の詳細については、「SSL 仮想サーバー構成」をクリックしてください。

証明書とキーのペアの追加の詳細については、[証明書とキーのペアの追加または更新] をクリックします。

証明書とキーのペアを SSL 仮想サーバーにバインドする方法については、「証明書とキーのペアを SSL 仮想サーバーにバインドする」をクリックします。

高可用性セットアップでの **NetScaler ADC** アプライアンス

August 15, 2023

Thales Luna HSM 構成を使用して NetScaler ADC アプライアンスで高可用性（HA）セットアップを構成するには、次のいずれかの方法があります。

- まず、同じ HSM とパーティションを使用して、2 つのノードで Thales Luna HSM を構成します。次に、HA ペアを作成します。最後に、キー、証明書とキーのペア、仮想サーバーなどの NetScaler ADC 構成をプライマリノードに追加します。
- Thales Luna HSM が NetScaler ADC 構成のあるノードで既に構成されている場合は、他のノードに同様の構成を追加します。最初のノードから別のノードに「/var/safenet/sfgw_ident_file」をコピーし、safenet_gw バイナリを再起動します。Gateway が起動して実行したら、HA セットアップでノードを追加します。

制限事項

August 15, 2023

1. HSM の追加や削除、HA セットアップの作成など、既存のセットアップで HSM 関連の設定を変更する場合は、「/etc/chrystoki.conf」を「/var/safenet/config」にコピーします。
2. HSM を追加、削除、または再起動したら、’/var/safenet/gateway/safenet_gw’ バイナリを再起動する必要があります。ゲートウェイバイナリを再起動しないと、HSM は追加後または再起動後にトラフィックを処理しません。
3. 現在の ‘/var/safenet/gateway/safenet_gw’ バイナリを再起動または停止するには、


```
kill  - SIGTERM <PID>
kill  - SIGINT <PID>
<!--NeedCopy-->
```

重要!`kill -9 <PID>`や`kill -6 <PID>`を使わない

4. ADC から既存の HSM を削除する前に、その HSM に関連付けられているすべてのキーと証明書とキーのペアを ADC から削除します。HSM を削除した後、ADC からこれらのファイルを削除することはできません。
5. スタンドアロンの NetScaler ADC アプライアンスでは、HA の Thales Luna HSM が Luna バージョン 6.2 以降でサポートされています。
6. EXPORT 暗号はサポートされていません。
7. 証明書とキーのペアの更新操作はサポートされていません。
8. サードパーティツールで HSM キーを生成する場合、プライベートキーとパブリックキーの名前は同じである必要があります。アプライアンスに HSM キーを追加するときは、この名前をキー名として入力します。
9. #文字は、キー名とパーティションパスワードではサポートされていません。
10. クラスタパーティションと管理パーティションはサポートされていません。

付録

August 15, 2023

サンプルコマンドとその出力:

スクリプトの実行

```
     root@ns# pwd
    /var/safenet/config
    root@ns# sh safenet_config
<!--NeedCopy-->
```

証明書を作成する

```
    root@ns# cd /var/safenet/safenet/lunaclient/bin
    root@ns# ./vtl createcert -n 10.102.59.175
    Private Key created and written to: /var/safenet/safenet/lunaclient
       /cert/client/10.102.59.175Key.pem
    Certificate created and written to: /var/safenet/safenet/lunaclient
       /cert/client/10.102.59.175.pem
<!--NeedCopy-->
```


証明書を **HSM** にコピー

```
    root@ns# scp /var/safenet/safenet/lunaclient/cert/client
       /10.102.59.175.pem admin@10.217.2.7:
    admin@10.217.2.7's password:

    10.102.59.175.pem          100%  818     0.8KB/s   00:00
<!--NeedCopy-->
```

証明書とキーを **HSM** から **NetScaler ADC** アプライアンスにコピーします

```
    root@ns# scp admin@10.217.2.7:server.pem /var/Thales Luna/safenet/
       lunaclient/server.2.7.pem
    admin@10.217.2.7's password:

    server.pem            100% 1164     1.1KB/s   00:01
<!--NeedCopy-->
```

SSH を使用して **Thales Luna HSM** に接続する

```
    ssh admin@10.217.2.7
    Connecting to 10.217.2.7:22...
    Connection established.
    To escape to local shell, press 'Ctrl+Alt+]'.

    Last login: Thu Jun 23 02:20:29 2016 from 10.252.243.11

    Luna SA 5.2.3-1 Command Line Shell - Copyright (c) 2001-2014
       SafeNet, Inc. All rights reserved.

    [Safenet1] lunash:>hsm login


      Please enter the HSM Administrators' password:
      > *******

    'hsm login' successful.


    Command Result : 0 (Success)
    [Safenet1] lunash:>
<!--NeedCopy-->
```

Thales Luna HSM に **NetScaler ADC** を登録する

```
    [Safenet1] lunash:>client register -client ns175 -ip 10.102.59.175
```



```
2
3      'client register' successful.
4
5
6      Command Result : 0 (Success)
7      [Safenet1] lunash:>
8  <!--NeedCopy-->
```

パーティション・リストからクライアントにパーティションを割り当てる

```
1      [Safenet1] lunash:>client assignPartition -client ns175 -partition
           p2
2
3      'client assignPartition' successful.
4
5
6      Command Result : 0 (Success)
7      [Safenet1] lunash:>
8  <!--NeedCopy-->
```

HSM とその証明書を **NetScaler** に登録します

```
1      root@ns# ./vtl addserver -n 10.217.2.7 -c /var/safenet/safenet/
           lunaclient/server.2.7.pem
2
3      New server 10.217.2.7 successfully added to server list.
4  <!--NeedCopy-->
```

ADC と **HSM** 間のネットワークトラストリンク **(NTL)** 接続を確認します

```
1      root@ns# ./vtl verify
2
3      The following Luna SA Slots/Partitions were found:
4
5      Slot        Serial #                  Label
6      ====      ================          =====
7         0              477877010          p2
8  <!--NeedCopy-->
```

構成を保存します

```
1      root@ns# cp /etc/Chrystoki.conf /var/safenet/config/
2  <!--NeedCopy-->
```


起動時にゲートウェイデーモンの自動起動を設定

```
    touch /var/safenet/safenet_is_enrolled
<!--NeedCopy-->
```

よくある質問

August 15, 2023

- **Thales Luna** プロセスが実行されていることを確認するにはどうすればよいですか?

  NetScaler ADC シェルプロンプトで、次のように入力します。

  ```
  ps  - aux | grep safenet_gw
  <!--NeedCopy-->
  ```

- **ADC** と **HSM** の間のネットワーク信頼リンク（**NTL**）接続を確認するにはどうすればよいですか**?**

  Thales Luna を設定したら、ディレクトリを「/var/safenet/safenet/lunaclient/bin」に変更し、次のように入力します。

  ```
  ./vtl verify
  <!--NeedCopy-->
  ```

Azure Key Vault のサポート

January 11, 2024

NetScaler ADC アプライアンスは、オンプレミス展開用に外部 HSM（SafeNet および Thales）と統合されます。クラウドデプロイの場合、ADC アプライアンスは Azure Key Vault と統合されます。アプライアンスは、パブリッククラウドドメインでの秘密キーの管理とセキュリティを容易にするために、秘密キーを Key Vault に保存します。複数のデータセンターやクラウドプロバイダーに展開されている ADC アプライアンスのキーを異なる場所に保管して管理する必要がなくなりました。

HSM でバックアップされたキーが提供された Azure Key Vault Premium の料金範囲で ADC を使用すると、FIPS 140-2 レベル 2 のコンプライアンスが提供されます。

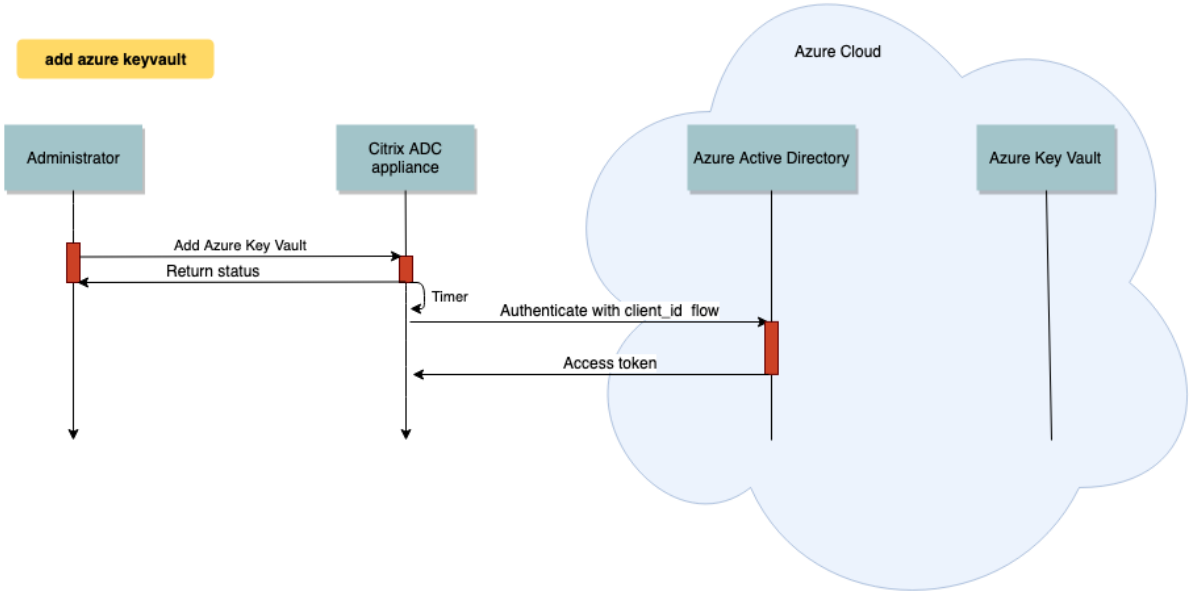
Azure Key Vault は、Microsoft が提供する標準サービスです。Azure Key Vault の詳細については、Microsoft Azure のドキュメントを参照してください。

> 注:
>
> NetScaler と Azure Key Vault の統合は、TLS 1.3 プロトコルでサポートされています。

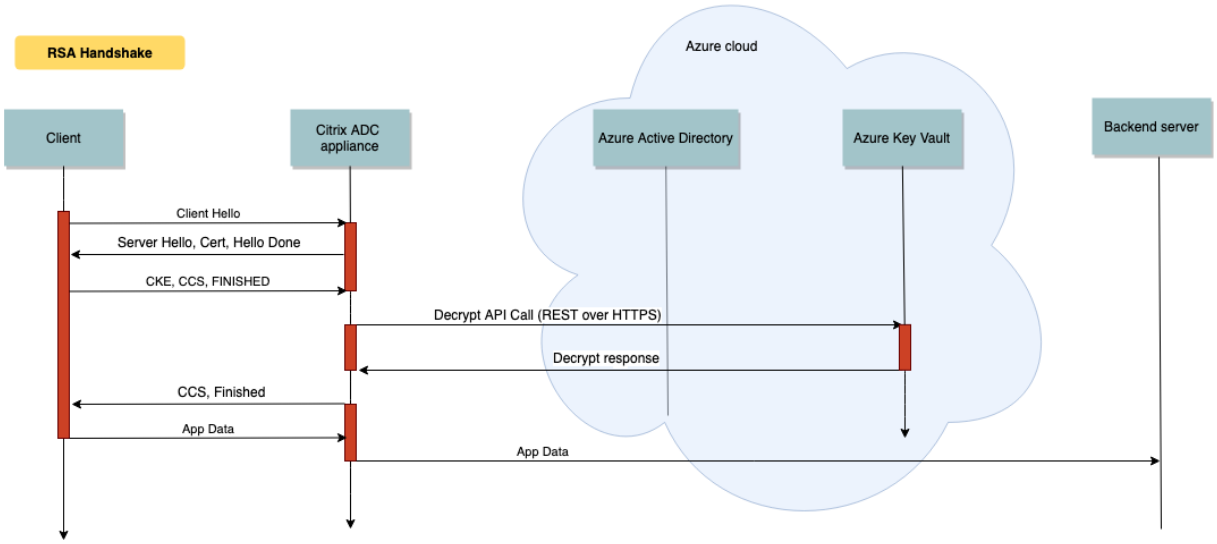
アーキテクチャの概要

Azure Key Vault は、Azure クラウドにシークレットを安全に保存するためのサービスです。キーを Azure Key Vault に保存することで、キーが盗まれる可能性を減らすことができます。Key Vault の設定が完了したら、キーを保管できます。Key Vault で秘密鍵操作を実行するように ADC アプライアンス上の仮想サーバーを構成します。ADC アプライアンスは、SSL ハンドシェイクごとにキーにアクセスします。

次の図は、認証後に Azure Active Directory からアクセストークンを取得するプロセスを示しています。このトークンは、秘密鍵を使用する暗号操作の REST API 呼び出しで使用されます。



次の図は、一般的な RSA ハンドシェイクを示しています。公開キーを使用して暗号化されたクライアントキー交換 (CKE) メッセージは、Key Vault に保存されている秘密キーを使用して復号されます。



ECDHE ハンドシェイクでは、NetScaler アプライアンスから送信されるサーバーキー交換（SKE）メッセージは、Key Vault に保存されている秘密鍵を使用して署名されます。

前提条件

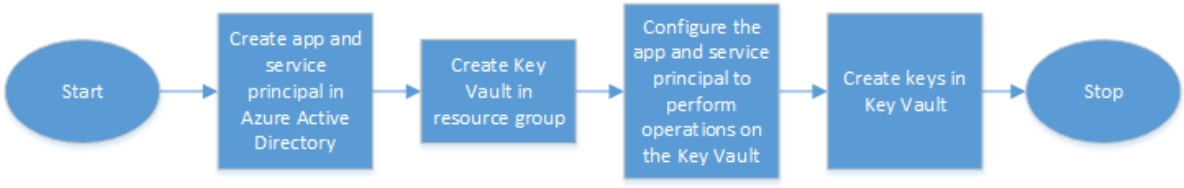
1. Azure サブスクリプションが必要です。
2. (オプション) Linux マシンに Azure CLI をインストールします。手順については、Azure のドキュメントhttps://docs.microsoft.com/en-us/cli/azure/install-azure-cli-apt?view=azure-cli-latestを参照してください。
3. ADC アプライアンスでエンティティを構成する前に、Azure Portal で構成を完了します。

ADC Azure Key Vault 統合を構成する

最初に Azure Portal で構成を実行し、続いて ADC アプライアンスで構成を実行します。

Azure ポータルで次の手順を実行します

次のフローチャートは、Azure Portal で必要な構成の概要を示しています。

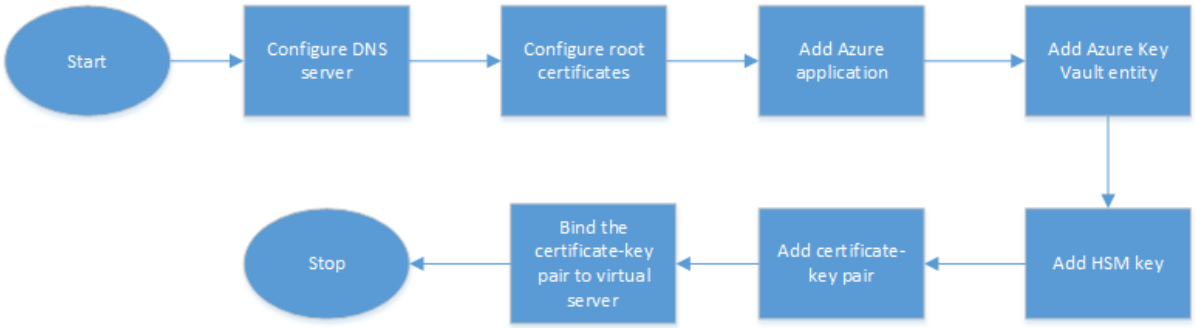


1. Azure Active Directory にアプリとサービスのプリンシパルを作成します
2. リソースグループに Key Vault を作成します。
3. Key Vault で署名および復号化操作を実行するようにアプリとサービスプリンシパルを設定します。
4. 次のいずれかの方法で Key Vault にキーを作成します。
    a) キーファイルをインポートする。
    b) 証明書を生成する。

上記の手順を構成するコマンドについては、https://docs.microsoft.com/en-us/azure/active-directory/develop/app-objects-and-service-principalsの Azure のドキュメントを参照してください。

ADC アプライアンスで次の手順を実行します

次のフローチャートは、ADC アプライアンスで必要な設定の概要を示しています。



1. DNS サーバーを構成します。
2. ルート証明書を構成して、Azure によって提示された証明書を検証します。
3. Azure アプリケーションを作成します。
4. Azure Key Vault エンティティを作成します。
5. HSM キーを作成します。
6. 証明書とキーのペアを作成します。
7. 証明書とキーのペアを仮想サーバーにバインドします。

DNS サーバーを構成する　Key Vault ホストと Azure Active Directory エンドポイントの名前解決には、DNS サーバーが必要です。

CLI を使用して DNS サーバーを設定するには

コマンドプロンプトで入力します:

```
add dns nameserver <IP address>
<!--NeedCopy-->
```

例**:**


```
add dns nameserver 192.0.2.150
<!--NeedCopy-->
```

GUI を使用して DNS サーバーを構成するには

1. [トラフィック管理] **> [DNS] > [**ネームサーバー] に移動します。［追加］をクリックします。
2. 次のパラメーターの値を入力します。
   - IP アドレス-外部ネームサーバの IP アドレス。Local パラメータが設定されている場合は、ローカル DNS サーバ (LDNS) の IP アドレス。
   - Protocol: ネームサーバーが使用するプロトコル。UDP_TCP は、ネームサーバーがアプライアンスに構成されている DNS 仮想サーバーである場合は無効です。
3. ［作成］をクリックします。

ルート証明書を追加してバインドする　AzureKey Vault`https://<vault_name>.vault.azure.net`と Azure Active Directory (AAD) https://login.microsoftonline.com によって提示された証明書のルート証明書をダウンロードし、ADC アプライアンスにロードします。これらの証明書は、Azure Key Vault と AAD によって提示された証明書を検証するために必要です。1 つ以上の証明書を CA 証明書グループ`ns_callout_certs`にバインドします。

CLI を使用してルート証明書を追加するには

コマンドプロンプトで入力します:

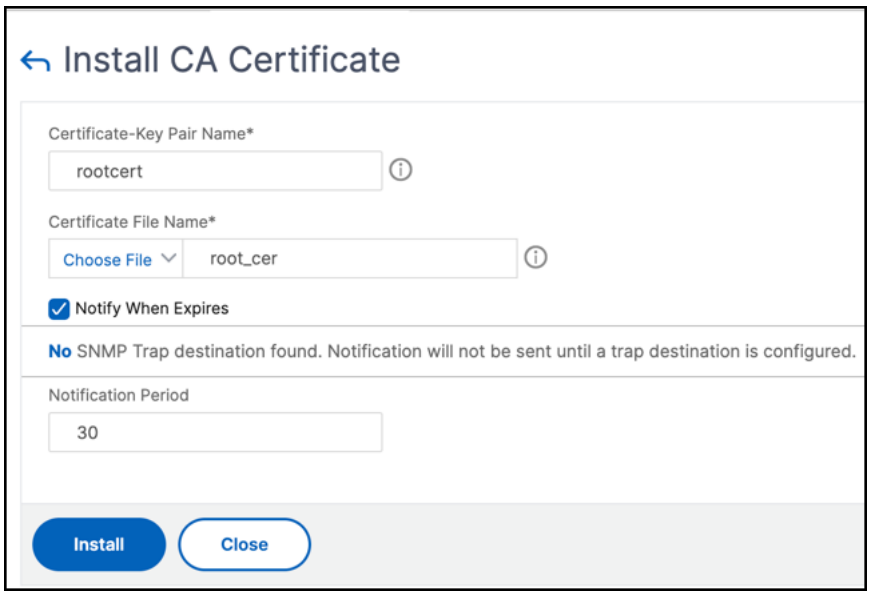
```
add ssl certkey <certkeyname> -cert <certname>
bind ssl caCertGroup <caCertGroupName> <certkeyName>
<!--NeedCopy-->
```

例:
次の例では、Azure Key Vault と AAD によって提示されるルート証明書は同じです。

```
add ssl certKey rootcert -cert RootCyberTrustRoot.crt
bind ssl cacertGroup ns_callout_certs  rootcert
<!--NeedCopy-->
```

GUI を使用してルート証明書を追加するには

1. [トラフィック管理] **> [SSL] > [**証明書] **> [CA** 証明書] に移動します。
2. 次のパラメーターの値を入力します。
   - 証明書とキーのペア名
   - 証明書ファイル名



3. [**Install**] をクリックします。
4. トラフィック管理 **> SSL > CA** 証明書グループに移動します。
5. **ns_callout_certs** を選択し、[バインディングの表示] をクリックします。
6. [**Bind**] をクリックします。
7. 前に作成した CA 証明書を選択し、[**Select**] をクリックします。
8. [バインド] をクリックし、[閉じる] をクリックします。

Azure アプリケーションの構成

Azure アプリケーションエンティティには、Azure Active Directory への認証とアクセストークンの取得に必要な資格情報が含まれています。つまり、Key Vault リソースと API への認証アクセスを取得するには、Azure アプリケーション ID、シークレット (パスワード)、テナント ID を ADC アプライアンスに追加します。
CLI を使用して Azure Application エンティティを構成する場合は、パスワードを入力する必要があります。GUI を使用する場合、Azure アプリケーションエンティティには、Azure Active Directory への認証とアクセストークンの取得に必要な資格情報が含まれます。

CLI を使用して Azure アプリケーションを構成するには

リリース 13.0 ～61.x では、アクセストークンがアプリケーションに付与される前にリソースグループのドメインを取得するためのパラメータ VaultResource が`add azure application`コマンドに追加されました。ドメイン名は地域によって異なる可能性があるため、このパラメータが追加されます。たとえば、ドメインは`vault.azure.net`または`vault.usgov.net`です。

コマンドプロンプトで入力します:

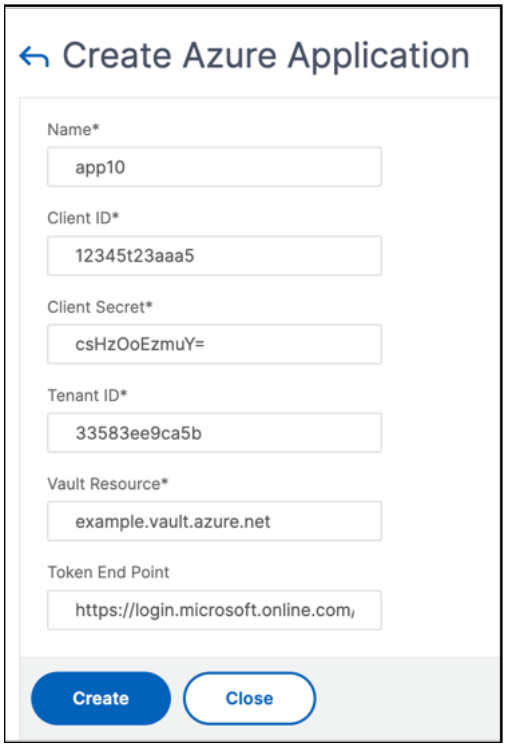

```
add azure application <name> -clientID <string> -clientSecret -tenantID
    <string> -vaultResource <string> [-tokenEndpoint <URL>]
show azure application
<!--NeedCopy-->
```

例:

```
add azure application app10 -clientiD 12345t23aaa5 -clientsecret
   csHzOoEzmuY= -vaultResource example.vault.azure.net -tenantID 33583
   ee9ca5b
 Done
> sh azure application app10
1)  Name:  app10            ClientID:  12345t23aaa5
    TokenEndpoint:  "https://login.microsoftonline.com/33583ee9ca5b/"
    TenantID:  33583ee9ca5b  VaultResource:  example.vault.azure.net
 Done

<!--NeedCopy-->
```

GUI を使用して Azure アプリケーションを構成するには

1. トラフィック管理 **> SSL > Azure >** アプリケーションに移動します。
2. 詳細ペインで、[追加] をクリックします。
3. 次のパラメーターの値を入力します。
   - 名前—NetScaler アプライアンス上のアプリケーションオブジェクトの名前。
   - クライアント ID —アプリケーションが Azure CLI または Azure ポータル (GUI) を使用して Azure Active Directory に作成されたときに生成されるアプリケーション ID。
   - クライアントシークレット—Azure Active Directory で構成されたアプリケーションのパスワード。パスワードは Azure CLI で指定されるか、Azure ポータル (GUI) で生成されます。
   - テナント ID —アプリケーションが作成された Azure Active Directory 内のディレクトリの ID。
   - Vault リソース-アクセストークンが付与される Vault リソース。例 `vault.azure.net`。
   - トークンエンドポイント—アクセストークンを取得できる URL。トークンエンドポイントが指定されていない場合、デフォルト値は`https://login.microsoftonline.com/<tenant id>`です。



Azure Key Vault を構成する　ADC アプライアンスに Azure Key Vault オブジェクトを作成します。

CLI を使用して Azure Key Vault を構成するには

コマンドプロンプトで入力します:

```
add azure keyVault <name> -azureVaultName <string> -azureApplication
       <string>
show azure keyvault
<!--NeedCopy-->
```

例:

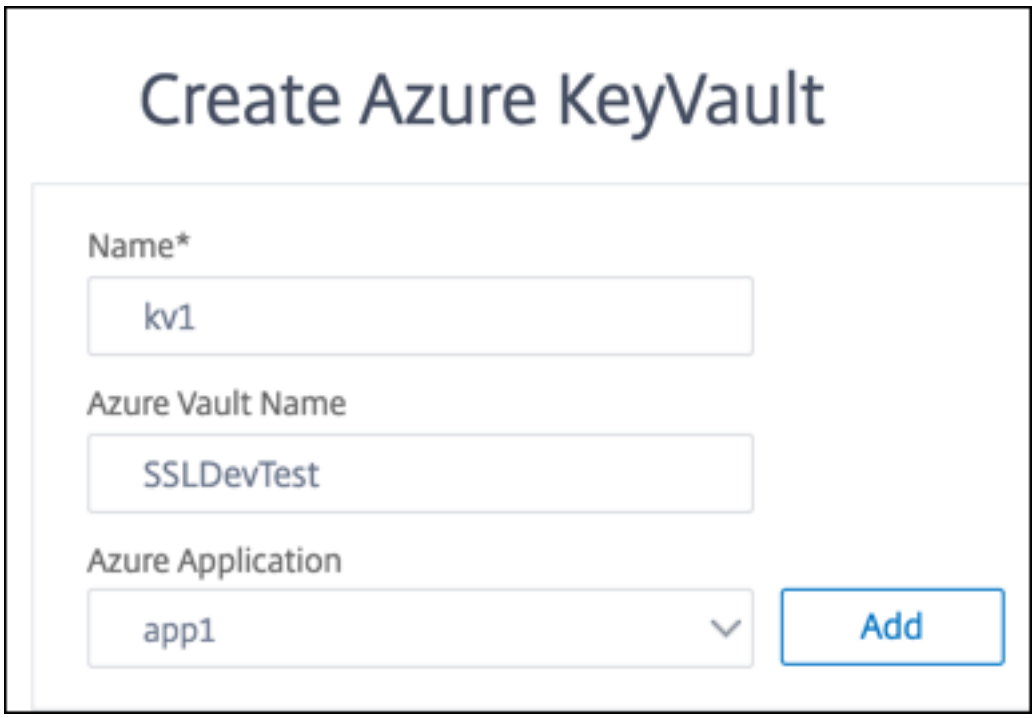
```
add azure keyvault kv1 -azureapplication app10 -azurevaultName pctest.
   vault.azure.net
> sh azure keyVault
1)  Name:  kv1              AzureVaultName:  pctest.vault.azure.net
  AzureApplication:  app10  State:  "Access token obtained"
 Done
<!--NeedCopy-->
```

次の表に、Azure Key Vault の状態で使用できるさまざまな値と、各状態に関する簡単な説明を示します。

状態	説明
Created	Key Vault オブジェクトの初期状態。認証は試行されていません。
Could not reach token end point	DNS サーバーが構成されていない、発行者証明書が CA 証明書グループにバインドされていない、またはネットワークの問題のいずれかを示します。
Authorization failed	アプリケーション認証情報が正しくありません。
Token parse error	Azure Active Directory からの応答が期待される形式ではありません。
Access token obtained	Azure Active Directory によって正常に認証されました。

GUI を使用して Azure Key Vault を構成するには

1. トラフィック管理 **> SSL > Azure > Key Vault** に移動します。
2. 次のパラメーターの値を入力します。
    - Name：Key Vault の名前。
    - Azure Key Vault 名-Azure CLI または Azure ポータル (GUI) を使用してドメイン名を使用して Azure クラウドで構成された Key Vault の名前。
    - Azure アプリケーション名-ADC アプライアンスで作成された Azure アプリケーションオブジェクトの名前。この名前の Azure アプリケーションオブジェクトは、Azure Active Directory での認証に使用されます。



HSM キーの追加　プライベートキーを HSM に保存すると、FIPS 140-2 レベル 2 に準拠できます。

CLI を使用して HSM キーを追加するには

コマンドプロンプトで入力します:

```
add ssl hsmKey <hsmKeyName> [-hsmType <hsmType>] [-key <string> |
       -serialNum <string>] {
 -password  }
  [-keystore <string>]
<!--NeedCopy-->
```

例**:**

```
add ssl hsmKey h1 -keystore kv1 -key san15key -hsmType KEYVAULT


> sh ssl hsmKey h1
    HSM Key Name: h1       Type: KEYVAULT
    Key: san15key
    Key store: kv1
    State:  “Created”
 Done
<!--NeedCopy-->
```

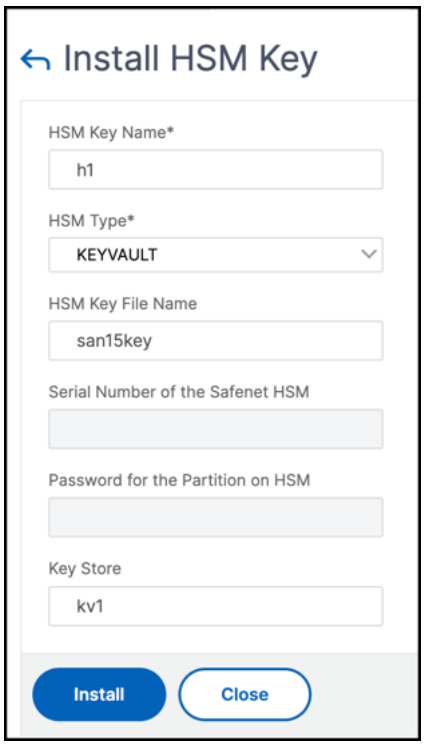
次の表に、HSM キーの状態で使用できるさまざまな値と、各状態に関する簡単な説明を示します。

状態	説明
作成日時	HSM キーが ADC アプライアンスに追加されます。キー操作はまだ試行されていません。
アクセストークンが使用	キー操作が試行されたときには、アクセストークンを使用できません。
無許可	構成された Azure アプリケーションには、キー操作を実行する権限がありません。
存在しない	キーは Azure Key ボールトに存在しません。
到達不能	ネットワーク上の Key Vault ホストにアクセスできません。
マークダウン	キーの操作中にしきい値エラーが発生したため、ADC アプライアンスで HSM キーが DOWN とマークされます。
主要なオペレーションが成功しました	キー操作について Key Vault から成功レスポンスが届きました。
キー操作に失敗しました	キー操作のために Key Vault からエラー応答を受信しました。
キー操作が調整された	キー操作のリクエストは、Key Vault によって調整されます。

GUI を使用して HSM キーを追加するには

1. トラフィック管理**] > [SSL] > [HSM]** に移動します。
2. 次のパラメータの値を入力します。

    - HSM キー名-キーの名前。
    - HSM タイプ-HSM のタイプ。
    - Key store-キーが格納されている HSM を表すキーストアオブジェクトの名前。たとえば、キー Vault オブジェクトや Azure Key Vault 認証オブジェクトの名前などです。KEYVAULTタイプ HSM にのみ適用されます。



3. [追加] をクリックします

証明書とキーのペアを追加します　前に作成した HSM キーを使用して、証明書とキーのペアを追加します。

CLI を使用して証明書とキーのペアを追加するには

コマンドプロンプトで入力します:

```
add ssl certKey <certkeyName> (-cert <string> [-password]) -hsmKey <
    string>]
show ssl certkey
<!--NeedCopy-->
```

例:

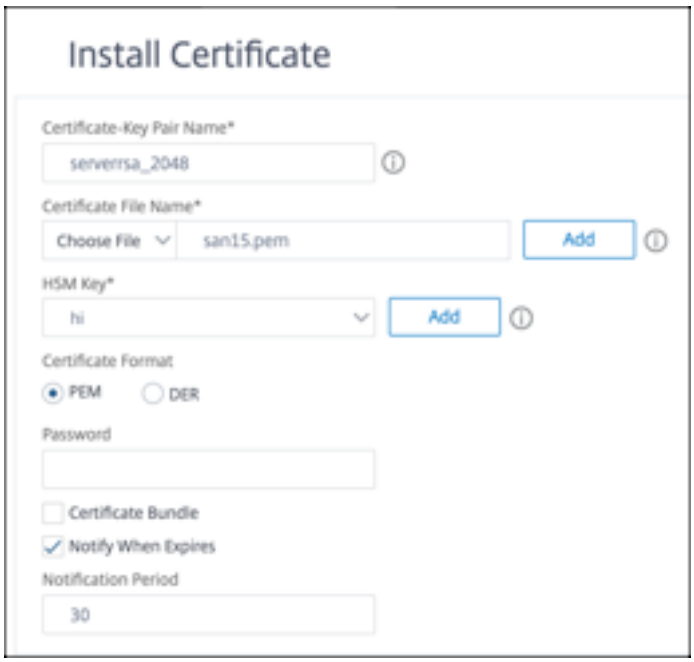
```
add ssl certKey serverrsa_2048 -cert /nsconfig/ssl/san_certs/san15.pem
    -hsmKey h1
> sh ssl certkey serverrsa_2048
    Name: serverrsa_2048        Status: Valid,   Days to expiration
        :9483
    Version: 3
    Serial Number: F5CFF9EF1E246022
    Signature Algorithm: sha256WithRSAEncryption
    Issuer:  C=in,O=citrix,CN=ca
    Validity
        Not Before: Mar 20 05:42:57 2015 GMT
        Not After : Mar 12 05:42:57 2045 GMT
    Certificate Type:   "Server Certificate"
    Subject:  C=in,O=citrix
```



```
    Public Key Algorithm: rsaEncryption
    Public Key size: 2048
    Ocsp Response Status: NONE
 Done
<!--NeedCopy-->
```

GUI を使用して証明書とキーのペアを追加するには

1. [トラフィック管理] **> [SSL] > [**証明書のインストール **(HSM)**] に移動します。
2. 次のパラメーターの値を入力します。
   - 証明書とキーのペア名
   - 証明書ファイル名
   - HSM キー



3. [**Install**] をクリックします。

証明書とキーのペアを仮想サーバーにバインドする　SSL トランザクションの処理に使用される証明書は、SSL データを受信する仮想サーバーにバインドする必要があります。

CLI を使用して SSL 証明書とキーのペアを仮想サーバーにバインドするには

コマンドプロンプトで入力します:

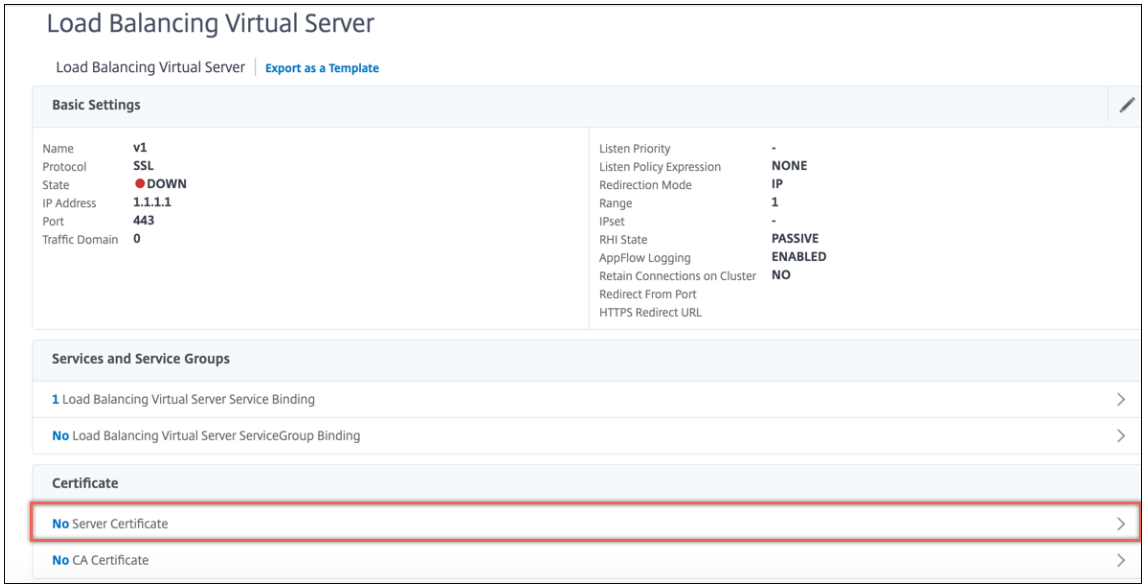
```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

例**:**


```
bind ssl vserver v1 -certkeyName serverrsa_2048

sh ssl vserver v1

    Advanced SSL configuration for VServer v1:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED      Refresh Count: 0
    Session Reuse: ENABLED      Timeout: 120 seconds
    Cipher Redirect: DISABLED
    ClearText Port: 0
    Client Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: DISABLED
    OCSP Stapling: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0
    HSTS Preload: NO
    SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Push Encryption Trigger: Always
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1

    ECC Curve: P_256, P_384, P_224, P_521

1)  CertKey Name: serverrsa_2048    Server Certificate



1)  Cipher Name: DEFAULT
    Description: Default cipher list with encryption strength >= 128bit
 Done
<!--NeedCopy-->
```

GUI を使用して SSL 証明書とキーのペアを仮想サーバーにバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、SSL 仮想サーバーを開きます。[証明書] セクション内をクリックします。



2. 矢印をクリックして、証明書とキーのペアを選択します。
3. リストから証明書とキーのペアを選択します。
4. 証明書とキーのペアを仮想サーバーにバインドします。

制限事項

- キー操作のための Azure Key Vault への同時呼び出しの数には制限があります。ADC アプライアンスのパフォーマンスは、Key Vault の制限によって異なります。詳細については、Microsoft Azure Key Vault のドキュメントを参照してください。
- EC キーはサポートされていません。
- EDT プロトコルと DTLS プロトコルはサポートされていません。
- Intel Coleto SSL チップを搭載した ADC アプライアンスはサポートされていません。
- クラスタリングおよび管理パーティションはサポートされていません。
- Azure アプリケーションエンティティ、Azure Key Vault オブジェクト、および HSM 証明書とキーのペアを ADC アプライアンスに追加した後は、更新できません。
- HSM キーを含む証明書バンドルはサポートされていません。
- HSM キーと証明書が一致しない場合、エラーは表示されません。証明書とキーのペアを追加する際は、HSM キーと証明書が一致していることを確認します。
- HSM キーを DTLS 仮想サーバーにバインドすることはできません。
- HSM キーを使用して作成された証明書とキーのペアを使用して OCSP リクエストに署名することはできません。
- HSM キーを使用して証明書とキーのペアを作成した場合、証明書とキーのペアを SSL サービスにバインドすることはできません。

よくある質問

Azure Key Vault と統合した場合、秘密キーは **ADC** アプライアンスのメモリに保存されますか

いいえ。秘密鍵は ADC アプライアンスのメモリには保存されません。SSL トランザクションごとに、アプライアンスは Key Vault にリクエストを送信します。

統合 **FIPS 140-2** レベル **2** は準拠していますか

はい。統合ソリューションは FIPS 140-2 レベル 2 のサポートを提供します。

どのキータイプがサポートされていますか

RSA キータイプのみがサポートされています。

どのキーサイズがサポートされていますか

1024 ビット、2048 ビット、および 4096 ビットの RSA キーがサポートされています。

どの暗号がサポートされていますか

ECDHE と SHA256 を使用した TLSv1.3 暗号を含む、ADC アプライアンスでサポートされるすべての暗号がサポートされます。

トランザクションはログに記録されますか

ADC アプライアンスは、Key Vault との各トランザクションをログに記録します。時間、Vault IP アドレス、ポート、接続の成功または失敗、エラーなどの詳細がログに記録されます。
次に、SSL ログ出力の例を示します。

```
1 Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
       0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 896 0 :
     Backend SPCBId 30894 - ServerIP 104.211.224.186 - ServerPort 443
      - ProtocolVersion TLSv1.2 - CipherSuite "ECDHE-RSA-AES256-GCM-
     SHA384 TLSv1.2 Non-Export 256-bit" - Session New -
     SERVER_AUTHENTICATED -SerialNumber "200005
     A75B04365827852D630000000005A75B" - SignatureAlgorithm "
     sha256WithRSAEncryption" - ValidFrom "Mar 17 03:28:42 2019 GMT"
     - ValidTo "Mar 17 03:28:42 2021 GMT" - HandshakeTime 40 ms
2 Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
       0-PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 897 0 :
     SPCBId 30894 - IssuerName " C=US,ST=Washington,L=Redmond,O=
```



```
      Microsoft Corporation,OU=Microsoft IT,CN=Microsoft IT TLS CA 2"
3 Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
       0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 898 0 :
      SPCBId 30894 - SubjectName " CN=vault.azure.net"
4 <!--NeedCopy-->
```

トラブルシューティング

August 15, 2023

構成後に SSL 機能が期待どおりに機能しない場合は、いくつかの一般的なツールを使用して NetScaler リソースにアクセスし、問題を診断できます。

トラブルシューティングのリソース

最良の結果を得るには、次のリソースを使用して NetScaler アプライアンスの SSL 問題をトラブルシューティングしてください。

- 関連する ns.log ファイル
- 最新の ns.conf ファイル
- メッセージファイル
- `newnslog` 関連ファイル
- トレースファイル
- 証明書ファイルのコピー (可能な場合)
- キーファイルのコピー (可能な場合)
- エラーメッセージ (ある場合)

これらのリソースに加えて、NetScaler トレースファイル用にカスタマイズされた Wireshark アプリケーションを使用して、トラブルシューティングを迅速に行うことができます。

SSL 問題のトラブルシューティング

SSL の問題をトラブルシューティングするには、以下の手順に従ってください。

- NetScaler アプライアンスに SSL オフロードと負荷分散のライセンスがあることを確認します。
- SSL オフロードおよび負荷分散機能がアプライアンスで有効になっていることを確認します。
- SSL 仮想サーバーのステータスが DOWN と表示されていないことを確認します。
- 仮想サーバーにバインドされているサービスのステータスが DOWN と表示されていないことを確認します。
- 有効な証明書が仮想サーバーにバインドされていることを確認します。
- サービスが適切なポート (できればポート 443) を使用していることを確認します。

パケットトレースからの **TLS1.3** トラフィックの復号化

TLS1.3 上で動作するプロトコルをトラブルシューティングするには、まず TLS1.3 トラフィックを復号化する必要があります。Wireshark で TLS 1.3 を復号化するには、秘密をNSS キーログ形式でエクスポートする必要があります。キーログ形式の詳細については、「NSS キーログ形式」を参照してください。

パケットトレースのキャプチャ方法については、トレース中の SSL セッションキーのキャプチャを参照してください。

注：NetScaler ADC は、各接続の秘密を、使用中の TLS/SSL プロトコルのバージョンに適した形式で自動的に記録します。

HA セットアップのセカンダリノードでは **CRL** 更新は行われません

CRL サーバにはプライベートネットワーク経由でプライマリノードにのみアクセスできるため、更新は行われません。

回避策**:** CRL サーバーの IP アドレスを使用してプライマリノードにサービスを追加します。このサービスは CRL サーバーのプロキシとして機能します。ノード間で構成が同期されると、CRL の更新は、プライマリノードで構成されたサービスを通じて、プライマリノードとセカンダリノードの両方に対して機能します。

SSL に関するよくある質問

August 15, 2023

基本的な質問

VPX インスタンスで **GUI** への **HTTPS** アクセスが失敗する。アクセスするにはどうしたらいいですか

GUI への HTTPS アクセスには、証明書とキーのペアが必要です。NetScaler ADC アプライアンスでは、証明書とキーのペアが自動的に内部サービスにバインドされます。デフォルトのキーサイズは、MPX または SDX アプライアンスで 1024 バイト、VPX インスタンスで 512 バイトです。ただし、最新の Web ブラウザーの多くは 1024 バイト未満のキーを受け入れません。このため、VPX 構成ユーティリティへの HTTPS アクセスがブロックされてしまいます。

構成ユーティリティへの HTTPS アクセスのために、少なくとも 1024 バイトの証明書とキーのペアをインストールし、内部サービスにバインドすることをお勧めします。または、`ns-server-certificate` を 1024 バイトに更新します。設定ユーティリティまたは CLI への HTTP アクセスを使用して、証明書をインストールできます。

MPX アプライアンスにライセンスを追加すると、証明書とキーのペアバインディングが失われます。この問題を解決するにはどうしたらいいですか

MPX アプライアンスの起動時にライセンスが存在せず、後でライセンスを追加してアプライアンスを再起動すると、証明書のバインドが失われる可能性があります。証明書を再インストールし、内部サービスにバインドします。

アプライアンスを起動する前に、適切なライセンスをインストールすることをお勧めします。

SSL トランザクションのセキュリティで保護されたチャネルの設定に関連するさまざまな手順は何ですか

SSL トランザクション用のセキュアチャネルを設定するには、次の手順を実行します。

1. クライアントは、セキュアチャネルに対する HTTPS 要求をサーバーに送信します。
2. プロトコルと暗号を選択すると、サーバーは証明書をクライアントに送信します。
3. クライアントはサーバー証明書の信頼性をチェックします。
4. いずれかのチェックが失敗した場合、クライアントは対応するフィードバックを表示します。
5. チェックが合格した場合、またはチェックが失敗してもクライアントが継続することを決定した場合、クライアントは一時的な使い捨てキーを作成します。このキーは事前マスターシークレットと呼ばれ、クライアントはサーバー証明書の公開キーを使用してこのキーを暗号化します。
6. サーバーは、プリマスターシークレットを受信すると、サーバーの秘密キーを使用してそれを復号化し、セッションキーを生成します。クライアントは、プリマスターシークレットからセッションキーも生成します。したがって、クライアントとサーバーの両方に、アプリケーションデータの暗号化と復号化に使用される共通のセッションキーがあります。

SSL は **CPU** 集約型のプロセスであることを理解しています。**SSL** プロセスに関連付けられた **CPU** コストはいくらですか

SSL プロセスには、次の 2 つのステージが関連付けられています。

- 公開鍵と秘密鍵技術を使用した最初のハンドシェイクとセキュアチャネル設定。
- 対称キー技術を使用した一括データ暗号化。

前述のステージはどちらもサーバーのパフォーマンスに影響を与える可能性があり、次の理由から集中的な CPU 処理が必要になります。

1. 最初のハンドシェイクには、公開秘密キーの暗号化が含まれます。これは、キーサイズ（1024 ビット、2048 ビット、4096 ビット）が大きいため、CPU が非常に集中します。
2. データの暗号化/復号化は、暗号化または復号化する必要があるデータの量に応じて、計算コストも高くなります。

SSL 設定のさまざまなエンティティは何ですか

SSL 設定には、次のエンティティがあります。

- サーバー証明書
- 認証局 (CA) 証明書
- 次のタスクのプロトコルを指定する暗号スイート。
  - 初期キー交換
  - サーバーとクライアントの認証
  - 一括暗号化アルゴリズム
  - メッセージ認証
- クライアント認証
- CRL
- SSL 証明書キー生成ツールを使用すると、次のファイルを作成できます。
  - 証明書リクエスト
  - 自己署名証明書
  - RSA キー
  - DH パラメータ

NetScaler ADC アプライアンスの **SSL** オフロード機能を使いたい。**SSL** 証明書を受信するためのさまざまなオプションは何ですか

NetScaler ADC アプライアンスで SSL セットアップを構成する前に、SSL 証明書を受け取る必要があります。SSL 証明書を受信するには、次のいずれかの方法を使用できます。

- 承認された認証局 (CA) に証明書を要求します。
- 既存のサーバー証明書を使用します。
- NetScaler ADC アプライアンスで証明書とキーのペアを作成します。

注：この証明書は、NetScaler ADC アプライアンスによって生成されたテストルート CA によって署名されたテスト証明書です。テスト root-CA によって署名されたテスト証明書は、ブラウザでは受け入れられません。ブラウザは、サーバーの証明書を認証できないことを示す警告メッセージがスローされます。

- テスト目的以外には、サーバー証明書に署名するための有効な CA 証明書と CA キーを指定する必要があります。

SSL セットアップの最小要件は何ですか

SSL セットアップを構成するための最小要件は次のとおりです。

- 証明書とキーを取得します。
- 負荷分散 SSL 仮想サーバーを作成します。
- HTTP または SSL サービスを SSL 仮想サーバーにバインドします。
- 証明書とキーのペアを SSL 仮想サーバーにバインドします。

SSL のさまざまなコンポーネントの制限は何ですか

SSL コンポーネントには、次の制限があります。

- SSL 証明書のビットサイズ:4096。
- SSL 証明書の数：アプライアンスの使用可能なメモリによって異なります。
- リンクされた中間 CA SSL 証明書の最大数: チェーンあたり 9
- CRL 失効：アプライアンスの使用可能なメモリによって異なります。

NetScaler ADC アプライアンスのエンドツーエンドのデータ暗号化に関連するさまざまな手順は何ですか

NetScaler ADC アプライアンスのサーバー側の暗号化プロセスに関連する手順は次のとおりです。

1. クライアントは、セキュリティで保護されたサイトの NetScaler ADC アプライアンスで構成された SSL VIP に接続します。
2. セキュアな要求を受信すると、アプライアンスは要求を復号化し、レイヤ 4 ～7 のコンテンツスイッチング技術とロードバランシングポリシーを適用します。次に、リクエストに使用可能な最適なバックエンド Web サーバーを選択します。
3. NetScaler ADC アプライアンスは、選択したサーバーとの SSL セッションを作成します。
4. SSL セッションを確立した後、アプライアンスはクライアント要求を暗号化し、セキュア SSL セッションを使用して Web サーバに送信します。
5. アプライアンスは、サーバから暗号化された応答を受信すると、データを復号化して再暗号化します。次に、クライアント側の SSL セッションを使用してデータをクライアントに送信します。

NetScaler ADC アプライアンスの多重化技術により、アプライアンスは Web サーバーで確立された SSL セッションを再利用できます。したがって、アプライアンスは、フルハンドシェイクと呼ばれる CPU を大量に消費するキー交換を回避します。このプロセスにより、サーバー上の SSL セッションの総数が削減され、エンドツーエンドのセキュリティが維持されます。

証明書とキー

証明書とキーファイルは任意の場所に配置できますか。これらのファイルを保存する推奨場所はありますか

証明書とキーファイルは、NetScaler ADC アプライアンスまたはローカルコンピュータに格納できます。ただし、証明書とキーファイルは Citrix ADC アプライアンスの/nsconfig/sslディレクトリに保存することをお勧めし

ます。/etc ディレクトリは、NetScaler ADC アプライアンスのフラッシュメモリに存在します。この操作により、移植性が提供され、アプライアンス上の証明書ファイルのバックアップと復元が容易になります。

注**:** 証明書とキーファイルが同じディレクトリに保存されていることを確認してください。

NetScaler ADC アプライアンスでサポートされる証明書キーの最大サイズはどれくらいですか

リリース 9.0 より前のソフトウェアリリースを実行している NetScaler ADC アプライアンスは、2048 ビットの最大証明書キーサイズをサポートします。リリース 9.0 以降では、4096 ビットの最大証明書キーサイズがサポートされています。この制限は RSA 証明書に適用されます。

MPX アプライアンスは、512 ビットから次のサイズまでの証明書をサポートします。

- 仮想サーバー上の 4096 ビットサーバー証明書
- サービス上の 4096 ビットのクライアント証明書
- 4096 ビット CA 証明書（中間証明書とルート証明書を含む）
- バックエンドサーバー上の 4096 ビット証明書
- 4096 ビットのクライアント証明書 (仮想サーバーでクライアント認証が有効になっている場合)

仮想アプライアンスは、512 ビットから次のサイズまでの証明書をサポートします。

- 仮想サーバー上の 4096 ビットサーバー証明書
- サービス上の 4096 ビットのクライアント証明書
- 4096 ビット CA 証明書（中間証明書とルート証明書を含む）
- リリース 12.0-56.x のバックエンドサーバー上の 4096 ビット証明書。古いリリースでは 2048 ビット証明書がサポートされています。
- リリース 12.0-56.x からの 2048 ビットのクライアント証明書（仮想サーバでクライアント認証が有効になっている場合）。

NetScaler ADC アプライアンスでサポートされる **DH** パラメータの最大サイズはどれくらいですか

NetScaler ADC アプライアンスは、最大 2048 ビットの DH パラメータをサポートしています。

NetScaler ADC アプライアンスでサポートされる証明書チェーンの最大長、つまりチェーン内の最大証明書数はどれくらいですか

NetScaler ADC アプライアンスは、サーバー証明書メッセージを送信するときに、チェーン内で最大 10 個の証明書を送信できます。最大長のチェーンには、サーバ証明書と 9 つの中間 CA 証明書が含まれます。

NetScaler ADC アプライアンスでサポートされるさまざまな証明書とキーの形式は何ですか

NetScaler ADC アプライアンスは、次の証明書とキーの形式をサポートしています。

- プライバシー強化メール (PEM)
- 識別エンコーディングルール (DER)

NetScaler ADC アプライアンスにインストールできる証明書とキーの数に制限はありますか

なしインストールできる証明書とキーの数は、NetScaler ADC アプライアンスの使用可能なメモリによってのみ制限されます。

証明書とキーファイルをローカルコンピューターに保存しました。**FTP** プロトコルを使用してこれらのファイルを **NetScaler ADC** アプライアンスに転送したい。これらのファイルを **NetScaler ADC** アプライアンスに転送するための優先モードはありますか

はい。FTP プロトコルを使用している場合は、バイナリモードを使用して証明書とキーファイルを NetScaler ADC アプライアンスに転送する必要があります。

注**:** デフォルトでは、FTP は無効になっています。証明書およびキーファイルを転送するには、SCP プロトコルをお勧めします。構成ユーティリティは、暗黙的に SCP を使用してアプライアンスに接続します。

証明書とキーのデフォルトのディレクトリパスは何ですか

証明書とキーのデフォルトのディレクトリパスは‘/nsconfig/ssl’です。

証明書とキーペアを追加するときに、証明書とキーファイルへの絶対パスを指定しないとどうなりますか

証明書とキーのペアを追加するときは、証明書とキーファイルへの絶対パスを指定します。指定しない場合、ADC アプライアンスはこれらのファイルをデフォルトディレクトリで検索し、カーネルにロードしようとします。デフォルトのディレクトリは/nsconfig/sslです。たとえば、アプライアンスの/nsconfig/sslディレクトリで cert1024.pem ファイルと rsa1024.pem ファイルが使用可能な場合、次のコマンドは両方とも正常に実行できます。

```
1 add ssl certKey cert1 -cert cert1204.pem -key rsa1024.pem
2 <!--NeedCopy-->
```

```
1 add ssl certKey cert1 -cert /nsconfig/ssl/cert1204.pem -key /nsconfig/
     ssl/rsa1024.pem
2 <!--NeedCopy-->
```


高可用性セットアップを構成しました。セットアップに **SSL** 機能を実装したい。高可用性セットアップで証明書とキーファイルをどのように処理する必要がありますか

高可用性設定では、プライマリとセカンダリの NetScaler ADC アプライアンスの両方に証明書とキーファイルを保存する必要があります。プライマリアプライアンスに SSL 証明書とキーのペアを追加する前に、証明書とキーファイルのディレクトリパスが両方のアプライアンスで同じである必要があります。

nCipher nShield® HSM

nCipher nShield® HSM と統合する場合、**NetScaler ADC** アプライアンスを **HA** に追加する際に特定の構成に留意する必要がありますか

HA の両方のノードで同じ nCipher デバイスを構成します。nCipher 構成コマンドは HA で同期しません。nCipher nShield® HSM の前提条件については、「前提条件」を参照してください。

nCipher nShield® HSM と **RFS** の両方を個別に統合する必要がありますか**? HA** セットアップの前後にこのアクションを完了する必要がありますか

統合は、HA セットアップの前後に完了できます。HA セットアップ後に統合が行われると、セカンダリノードを設定する前にプライマリノードにインポートされたキーは、セカンダリノードに同期されません。したがって、高可用性セットアップの前に nCipher 統合をお勧めします。

プライマリとセカンダリの両方の **NetScaler ADC** アプライアンスにキーをインポートする必要がありますか、それともプライマリノードからセカンダリノードにキーが同期されていますか

nCipher が HA を形成する前に両方のデバイスに統合されている場合、キーは統合プロセスで RFS から自動的に同期されます。

HSM が **NetScaler ADC** アプライアンスではなく **nCipher** 上にある場合、ノードに障害が発生して交換されたときのキーと証明書はどうなりますか

ノードに障害が発生した場合、新しいノードに nCipher を統合することで、キーと証明書を新しいノードに同期できます。次に、次のコマンドを実行します。

```
sync ha files ssl
force ha sync
<!--NeedCopy-->
```

nCipher の統合プロセスでキーが同期されている場合、証明書は同期され、追加されます。

暗号

ヌル暗号って何ですか

暗号化のない暗号は、NULL-Ciphers と呼ばれます。たとえば、NULL-MD5 はヌル暗号です。

SSL VIP または **SSL** サービスの場合、**NULL** 暗号はデフォルトで有効になっていますか

なし SSL VIP または SSL サービスでは、NULL 暗号はデフォルトで有効になりません。

ヌル暗号を削除する手順は何ですか

SSL VIP から NULL 暗号を削除するには、次のコマンドを実行します。

```
bind ssl cipher <SSL_VIP> REM NULL
<!--NeedCopy-->
```

SSL サービスから NULL 暗号を削除するには、次のコマンドを実行します。

```
bind ssl cipher <SSL_Service> REM NULL -service
<!--NeedCopy-->
```

NetScaler ADC アプライアンスでサポートされるさまざまな暗号エイリアスは何ですか

アプライアンスでサポートされている暗号エイリアスを一覧表示するには、コマンドプロンプトで次のように入力します。

```
sh cipher
<!--NeedCopy-->
```

NetScaler ADC アプライアンスの事前定義された暗号をすべて表示するコマンドは何ですか

NetScaler ADC アプライアンスの事前定義された暗号をすべて表示するには、CLI で次のように入力します。

```
show ssl cipher
<!--NeedCopy-->
```

NetScaler ADC アプライアンスの個々の暗号の詳細を表示するコマンドは何ですか

NetScaler ADC アプライアンスの個々の暗号の詳細を表示するには、CLI で次のように入力します。

```
show ssl cipher <Cipher_Name/Cipher_Alias_Name/Cipher_Group_Name>
<!--NeedCopy-->
```


例:

```
show cipher SSL3-RC4-SHA
      1) Cipher Name: SSL3-RC4-SHA
      Description: SSLv3 Kx=RSA Au=RSA Enc=RC4(128)
     Mac=SHA1
      Done
<!--NeedCopy-->
```

NetScaler ADC アプライアンスの事前定義された暗号を追加することの意義は何ですか

NetScaler ADC アプライアンスの事前定義された暗号を追加すると、NULL 暗号が SSL VIP または SSL サービスに追加されます。

NetScaler ADC アプライアンスの暗号グループからバインド解除せずに暗号の順序を変更することは可能ですか

はい。カスタム暗号グループから暗号をバインド解除することなく、暗号の順序を変更できます。ただし、組み込みの暗号グループの優先順位を変更することはできません。SSL エンティティにバインドされた暗号の優先順位を変更するには、まず仮想サーバー、サービス、またはサービスグループから暗号をバインド解除します。

注: SSL エンティティにバインドされた暗号グループが空の場合、ネゴシエートされた暗号がないため、SSL ハンドシェイクは失敗します。暗号グループには、少なくとも 1 つの暗号が含まれている必要があります。

ECDSA は **NetScaler ADC** アプライアンスでサポートされていますか

ECDSA は、次の NetScaler ADC プラットフォームでサポートされています。サポートされているビルドの詳細については、NetScaler ADC アプライアンスで使用可能な暗号の表 1 と表 2 を参照してください。

- N3 チップを搭載した NetScaler ADC MPX および SDX アプライアンス
- NetScaler MPX 5900/8900/15000/26000
- NetScaler SDX 8900/15000
- NetScaler VPX アプライアンス

NetScaler VPX アプライアンスは、フロントエンドで **AES-GCM/SHA2** 暗号をサポートしていますか

はい、AES-GCM/SHA2 暗号は NetScaler ADC VPX アプライアンスでサポートされています。サポートされているビルドの詳細については、NetScaler ADC アプライアンスで使用可能な暗号を参照してください。

証明書

クライアント証明書の識別名は、ユーザーセッションの長さで使用できますか

はい。ユーザーセッションの間は、後続のリクエストでクライアント証明書の識別名にアクセスできます。つまり、SSL ハンドシェイクが完了し、証明書がブラウザによって再度送信されない後でも同様です。次の設定例で説明されているように、変数と代入を使用します。

例:

```
add ns variable v2 -type "text(100)"

add ns assignment a1 -variable "$v2" -set       "CLIENT.SSL.CLIENT_CERT
    .SUBJECT.TYPECAST_NVLIST_T('=','/').VALUE("CN")"

add rewrite action act1 insert_http_header subject "$v2"  // example:
    to insert the distinguished name in the header

add rewrite policy pol1 true a1

add rewrite policy pol2 true act1

bind rewrite global pol1 1 next -type RES_DEFAULT

bind rewrite global pol2 2 next -type RES_DEFAULT

set rewrite param -undefAction RESET
<!--NeedCopy-->
```

サーバー証明書をバインドする必要があるのはなぜですか

サーバー証明書のバインドは、SSL 設定で SSL トランザクションを処理するための基本的な要件です。

サーバー証明書を SSL VIP にバインドするには、CLI で次のように入力します。

```
bind ssl vserver <vServerName> -certkeyName <cert_name>
<!--NeedCopy-->
```

サーバー証明書を SSL サービスにバインドするには、CLI で次のように入力します。

```
bind ssl service <serviceName> -certkeyName <cert_name>
<!--NeedCopy-->
```

SSL VIP または **SSL** サービスにバインドできる証明書はいくつありますか

NetScaler VPX、MPX/SDX（N3）、および MPX/SDX 14000 FIPS アプライアンスでは、2 つの証明書を SSL 仮想サーバーまたは SSL サービスにバインドできます（SNI が無効の場合）。証明書は RSA および ECDSA の各タイプの

1 つである必要があります。SNI が有効の場合、RSA または ECDSA タイプの複数のサーバ証明書をバインドできます。NetScaler MPX（N2）または MPX 9700 FIPS アプライアンスで、SNI が無効になっている場合は、RSA タイプの証明書を 1 つだけバインドできます。SNI が有効な場合、RSA タイプの複数のサーバ証明書のみをバインドできます。

サーバー証明書のバインドを解除または上書きするとどうなりますか

サーバー証明書のバインドを解除または上書きすると、既存の証明書を使用して作成されたすべての接続と SSL セッションが終了します。既存の証明書を上書きすると、次のメッセージが表示されます。

```
ERROR:

Warning: Current certificate replaces the previous binding.
<!--NeedCopy-->
```

NetScaler ADC アプライアンスに中間証明書をインストールし、サーバー証明書にリンクするにはどうすればよいですか

中間証明書のインストールについては、「http://support.citrix.com/article/ctx114146」の記事を参照してください。

NetScaler ADC に証明書をインストールしようとすると、「リソースはすでに存在します」というエラーが発生するのはなぜですか

「リソースはすでに存在します」エラーの解決方法については、「http://support.citrix.com/article/CTX117284」の記事を参照してください。

NetScaler ADC アプライアンスでサーバー証明書を作成して、製品をテストおよび評価したい。サーバー証明書を作成する手順は何ですか

次の手順を実行して、テスト証明書を作成します。

注: この手順で作成された証明書を使用して、すべてのユーザおよびブラウザを認証することはできません。テストに証明書を使用した後は、承認されたルート認証局によって署名されたサーバー証明書を取得する必要があります。

自己署名サーバー証明書を作成するには、次の手順を実行します。

1. ルート CA 証明書を作成するには、CLI で次のように入力します。

```
create ssl rsakey /nsconfig/ssl/test-ca.key 1024

create ssl certreq /nsconfig/ssl/test-ca.csr -keyfile /nsconfig/
   ssl/test-ca.key
```



```
4
5 Enter the required information when prompted, and then type the
     following command:
6
7 create ssl cert /nsconfig/ssl/test-ca.cer /nsconfig/ssl/test-ca.
     csr ROOT_CERT -keyfile /nsconfig/ssl/test-ca.key
8 <!--NeedCopy-->
```

2. 次の手順を実行して、サーバー証明書を作成し、作成したルート CA 証明書で署名します。

   a) リクエストとキーを作成するには、CLI で次のように入力します。

```
1 create ssl rsakey /nsconfig/ssl/test-server.key 1024
2
3     create ssl certreq /nsconfig/ssl/test-server.csr -keyfile
         /nsconfig/ssl/test-server.key
4 <!--NeedCopy-->
```

   b) プロンプトが表示されたら、必要な情報を入力します。

   c) シリアル番号ファイルを作成するには、CLI で次のように入力します。

```
1 shell
2  # echo '01' >
3 /nsconfig/ssl/serial.txt
4  # exit
5 <!--NeedCopy-->
```

   d) ステップ 1 で作成したルート CA 証明書によって署名されたサーバ証明書を作成するには、CLI で次のように入力します。

```
1 create ssl cert /nsconfig/ssl/test-server.cer /nsconfig/ssl/
     test-server.csr SRVR_CERT -CAcert /nsconfig/ssl/test-ca.cer
      -CAkey /nsconfig/ssl/test-ca.key -CAserial /nsconfig/ssl/
     serial.txt
2 <!--NeedCopy-->
```

   e) SSL ハンドシェイクとバルク暗号化のサーバー証明書情報を保持するメモリ内オブジェクトである NetScaler ADC 証明書とキーのペアを作成するには、CLI で次のように入力します。

```
1 add ssl certkey test-certkey -cert /nsconfig/ssl/test-server.
     cer -key /nsconfig/ssl/test-server.key
2 <!--NeedCopy-->
```

   f) 証明書とキーのペアを SSL 仮想サーバーにバインドするには、CLI で次のように入力します。

```
1 bind ssl vserver <vServerName> -certkeyName <cert_name>
2 <!--NeedCopy-->
```


NetScaler ソフトウェアリリース **9.0** がインストールされている **NetScaler ADC** アプライアンスを受け取りました。アプライアンスに追加のライセンスファイルがあることに気付きました。**NetScaler** ソフトウェアリリース **9.0** 以降のライセンスポリシーに変更はありますか

はい。NetScaler ソフトウェアリリース 9.0 以降では、アプライアンスにライセンスファイルが 1 つもない場合があります。ライセンスファイルの数は、NetScaler ADC ソフトウェアリリースエディションによって異なります。たとえば、Advanced Edition をインストールしている場合、さまざまな機能の完全な機能のために追加のライセンスファイルが必要になる場合があります。ただし、Premium エディションをインストールした場合、アプライアンスにはライセンスファイルが 1 つしかありません。

インターネットインフォメーションサービス **(IIS)** から証明書をエクスポートするにはどうすればよいですか

さまざまな方法がありますが、次の方法を使用して、Web サイトの適切な証明書と秘密キーがエクスポートされます。この手順は、実際の IIS サーバーで実行する必要があります。

1. インターネットインフォメーションサービス (IIS) マネージャー管理ツールを開きます。
2. Web サイトノードを展開し、NetScaler ADC アプライアンスを介して提供する SSL 対応の Web サイトを見つけます。
3. この Web サイトを右クリックし、[プロパティ] をクリックします。
4. [ディレクトリセキュリティ] タブをクリックし、ウィンドウの [セキュリティで保護された通信] セクションで、[証明書の表示] ボックスを選択します。
5. [詳細] タブをクリックし、[ファイルにコピー] をクリックします。
6. [証明書のエクスポートウィザードへようこそ] ページで、[次へ] をクリックします。
7. [はい、秘密キーをエクスポートする] を選択し、[次へ] をクリックします。

   注： SSL オフロードが NetScaler ADC で動作するには、秘密キーをエクスポートする必要があります。
8. *[*個人情報交換*-PKCS #12]* ラジオボタンが選択されていることを確認し、*[*可能な場合はすべての証明書を証明パスに含める*]* チェックボックスをオンにします。［次へ］をクリックします。
9. パスワードを入力し、[次へ] をクリックします。
10. ファイル名と場所を入力し、[次へ] をクリックします。ファイルの拡張子を.PFX にします。
11. ［完了］をクリックします。

PKCS #12 証明書を変換して **NetScaler ADC** にインストールするにはどうすればよいですか

1. エクスポートされた.PFX 証明書ファイルを、NetScaler ADC アプライアンスにコピーできる場所に移動します。つまり、NetScaler ADC アプライアンスの管理インターフェイスへの SSH アクセスを許可するマシンです。SCP などのセキュアコピーユーティリティを使用して、証明書をアプライアンスにコピーします。

2. BSD シェルにアクセスし、証明書（Cert.pfx など）を.PEM 形式に変換します。

```
root@ns# openssl pkcs12 -in cert.PFX -out cert.PEM
<!--NeedCopy-->
```

3. 変換された証明書が正しい x509 形式であることを確認するには、次のコマンドでエラーが生成されないことを確認します。

```
root@ns# openssl x509 -in cert.PEM -text
<!--NeedCopy-->
```

4. 証明書ファイルに秘密キーが含まれていることを確認します。次のコマンドを発行することから始めます。

```
root@ns# cat cert.PEM

Verify that the output file includes an RSA PRIVATE KEY section.

-----BEGIN RSA PRIVATE KEY-----
Mkm^s9KMs9023pz/s...
-----END RSA PRIVATE KEY-----
<!--NeedCopy-->
```

次に、RSA 秘密キーセクションの別の例を示します。

```
    Bag Attributes
    1.3.6.1.4.1.311.17.2: <No Values>
    localKeyID: 01 00 00 00
    Microsoft CSP Name: Microsoft RSA SChannel Cryptographic
    Provider
    friendlyName:
    4b9cef4cc8c9b849ff5c662fd3e0ef7e_76267e3e-6183-4d45-886e-6
       e067297b38f

    Key Attributes
    X509v3 Key Usage: 10
    -----BEGIN RSA PRIVATE KEY-----
    Proc-Type: 4,ENCRYPTED
    DEK-Info: DES-EDE3-CBC,43E7ACA5F4423968
    pZJ2SfsSVqMbRRf6ug37Clua5gY0Wld4frPIxFXyJquUHr31dilW5ta3hbIaQ+
       Rg

    ... (more random characters)
    v8dMugeRplkaH2Uwt/mWBk4t71Yv7GeHmcmjafK8H8iW80ooPO3D/ENV8X4U/
       tlh

    5eU6ky3WYZ1BTy6thxxLlwAullynVXZEflNLxq1oX+ZYl6djgjE3qg==
    -----END RSA PRIVATE KEY-----
<!--NeedCopy-->
```

以下は、サーバー証明書のセクションです。

```
    Bag Attributes
    localKeyID: 01 00 00 00
```



```
    friendlyName: AG Certificate
    subject=/C=AU/ST=NSW/L=Wanniassa/O=Dave Mother
    Asiapacific/OU=Support/CN=davemother.food.lan
    issuer=/DC=lan/DC=food/CN=hotdog
    -----BEGIN CERTIFICATE-----
    MIIFiTCCBHGgAwIBAgIKCGryDgAAAAAHzANBgkqhkiG9w0BAQUFADA8MRMwEQYK


    ... (more random characters) 5
       pLDWYVHhLkA1pSxvFjNJHRSIydWHc5ltGyKqIUcBezVaXyel94pNSUYx07NpPV
       /

    MY2ovQyQZM8gGe3+lGFum0VHbv/y/gB9HhFesog=
    -----END CERTIFICATE-----
<!--NeedCopy-->
```

以下は、中級 CA 証明書のセクションです。

```
    Bag Attributes: <Empty Attributes>
    subject=/DC=lan/DC=food/CN=hotdog
    issuer=/DC=lan/DC=food/CN=hotdog
    -----BEGIN CERTIFICATE-----
    MIIESDCCAzCgAwIBAgIQah20fCRYTY9LRXYMIRaKGjANBgkqhkiG9w0BAQUFADA8


    ... (more random characters)
       Nt0nksawDnbKo86rQcNnY5xUs7c7pj2zxj/IOsgNHUp5W6dDI9pQoqFFaDk
       =

    -----END CERTIFICATE-----
<!--NeedCopy-->
```

エクスポートされた証明書の証明書パスによっては、さらに中間 CA 証明書が続く場合があります。

5. テキストエディタで.PEM ファイルを開きます
6. .PEM ファイルの最初の行と次の行の最初のインスタンスを見つけ、それらの 2 行とその間のすべての行をコピーします。

```
-----END CERTIFICATE-----

Note: Make sure that last copied line is the first
-----END CERTIFICATE----- line in the .PEM file.

<!--NeedCopy-->
```

7. コピーした行を新しいファイルに貼り付けます。cert-key.pem など、新しいファイルを直観的に呼び出します。この証明書とキーのペアは、HTTPS サービスをホストするサーバー用です。このファイルには、RSA PRIVATE KEY というラベルの付いたセクションと、前述の例の SERVER CERTIFICATE というラベルの付いたセクションの両方が含まれている必要があります。

   注**:** 証明書とキーのペアファイルには、秘密キーが含まれているため、安全に保つ必要があります。

8. —BEGIN CERTIFICATE—で始まり—END CERTIFICATE—で終わる後続のセクションを探し、そのようなセクションをそれぞれ別の新しいファイルにコピーします。

   これらのセクションは、証明書パスに含まれている信頼できる CA の証明書に対応しています。これらのセクションは、これらの証明書の新しい個別のファイルにコピーして貼り付ける必要があります。たとえば、前の例の中間 CA 証明書セクションをコピーして新しいファイルに貼り付ける必要があります)。

   元のファイルに複数の中間 CA 証明書がある場合は、各中間 CA 証明書のファイルを、ファイル内の順番に作成します。後の手順で正しい順序で証明書をリンクする必要があるため、証明書が表示される順序を（適切なファイル名を使用して）追跡します。

9. 証明書キーファイル（cert-key.pem）および追加の CA 証明書ファイルを NetScaler ADC アプライアンス上の/nsconfig/ssl ディレクトリにコピーします。
10. BSD シェルを終了し、NetScaler ADC プロンプトにアクセスします。
11. 「アプライアンスに証明書キーファイルをインストールする」の手順に従って、デバイスにアップロードされたキー/証明書をインストールします。

PKCS #7 証明書を変換して **NetScaler ADC** アプライアンスにインストールするにはどうすればよいですか

OpenSSL を使用して、PKCS #7 証明書を NetScaler ADC アプライアンスで認識できる形式に変換できます。この手順は PKCS #12 証明書の手順と同じです。ただし、異なるパラメータで OpenSSL を呼び出す点が異なります。PKCS #7 証明書を変換する手順は次のとおりです。

1. SCP などのセキュアコピーユーティリティを使用して、証明書をアプライアンスにコピーします。
2. 証明書（Cert.p7b など）を PEM 形式に変換します。

```
1 openssl pkcs7 -inform DER -in cert.p7b -print_certs -text -out
     cert.pem
2 <!--NeedCopy-->
```

3. PKCS #12 証明書の回答で説明されている手順 3 ～7 に従います。
   注：変換された PKCS #7 証明書をアプライアンスにロードする前に、PKCS #12 手順のステップ 3 で説明したとおりに秘密キーが含まれていることを確認します。PKCS #7 証明書、特に IIS からエクスポートされた証明書には、通常、秘密キーは含まれません。

bind cipher コマンドを使用して暗号を仮想サーバーまたはサービスにバインドすると、「**Command deprecated**?」

暗号を仮想サーバーまたはサービスにバインドするコマンドが変更されました。

SSL 暗号を SSL 仮想サーバにバインドするには、`bind ssl vserver <vsername> -ciphername <ciphername>` コマンドを使用します。

SSL 暗号を SSL サービスにバインドするには、`bind ssl service <serviceName> -ciphername <ciphername>` コマンドを使用します。

注**:** 新しい暗号および暗号グループは既存のリストに追加され、置き換えられません。

add cipher コマンドを使用して暗号グループを作成し、それに暗号をバインドできないのはなぜですか

add cipher コマンドの機能がリリース 10 で変更されました。このコマンドは、暗号グループのみを作成します。グループに暗号を追加するには、bind cipher コマンドを使用します。

openSSL

OpenSSL を使用して **PEM** と **DER** の間で証明書を変換するにはどうすればよいですか**?**

OpenSSL を使用するには、OpenSSL ソフトウェアを正常にインストールし、コマンドラインから OpenSSL を実行できる必要があります。

x509 証明書と RSA キーは、いくつかの異なる形式で保存できます。

一般的な形式は次のとおりです。

- DER (主に Java および Macintosh プラットフォームで使用されるバイナリ形式)
- PEM (ヘッダーとフッター情報を持つ DER の base64 表現。主に UNIX および Linux プラットフォームで使用されます)。

ルート証明書および中間証明書に加えて、キーと対応する証明書は、単一の PKCS #12 (.P12、.PFX) ファイルに格納することもできます。

手順

OpenSSL コマンドを使用して、次の形式を変換します。

1. 証明書を PEM から DER に変換するには、次の手順を実行します。

```
x509 -in input.crt -inform PEM -out output.crt -outform DER
<!--NeedCopy-->
```

2. 証明書を DER から PEM に変換するには、次の手順を実行します。

```
x509 -in input.crt -inform DER -out output.crt -outform PEM
<!--NeedCopy-->
```

3. キーを PEM から DER に変換するには:

```
rsa -in input.key -inform PEM -out output.key -outform DER
<!--NeedCopy-->
```

4. キーを DER から PEM に変換するには、次の手順を実行します。


```
rsa -in input.key -inform DER -out output.key -outform PEM
<!--NeedCopy-->
```

注**:** インポートするキーがサポートされている対称暗号で暗号化されている場合、パスフレーズの入力を求められます。

注: キーを旧式の NET（Netscape サーバ）形式に変換するには、必要に応じて PEM または DER を NET に置き換えます。保存された鍵は、脆弱な無塩 RC4 対称暗号で暗号化されているため、パスフレーズが要求されます。パスフレーズは空白でもかまいません。

システム制限

覚えておくべき重要な数字は何ですか

1. 証明書リクエストの作成:
   - リクエストファイル名: 最大 63 文字
   - キーファイル名: 最大 63 文字
   - PEM パスフレーズ (暗号化キー用): 最大 31 文字
   - 一般名: 最大 63 文字
   - 市区町村: 最大 127 文字
   - 組織名: 最大 63 文字
   - 州/州名: 最大 63 文字
   - メールアドレス: 最大 255 文字
   - 組織単位: 最大 63 文字
   - チャレンジパスワード: 最大 20 文字
   - 会社名: 最大 127 文字
2. 証明書の作成:
   - 証明書ファイル名: 最大 63 文字
   - 証明書要求ファイル名: 最大 63 文字
   - キーファイル名: 最大 63 文字
   - PEM パスフレーズ: 最大 31 文字
   - 有効期間: 最大 3650 日
   - CA 証明書ファイル名: 最大 63 文字
   - CA キーファイル名: 最大 63 文字
   - PEM パスフレーズ: 最大 31 文字
   - CA シリアル番号ファイル: 最大 63 文字
3. サーバーテスト証明書を作成してインストールします。
   - 証明書ファイル名: 最大 31 文字

- 完全修飾ドメイン名: 最大 63 文字

4. Diffie-Hellman (DH) キーを作成します。

   - DH ファイル名 (パスあり): 最大 63 文字
   - DH パラメータサイズ: 最大 2048 ビット

5. PKCS12 キーをインポート:

   - 出力ファイル名: 最大 63 文字
   - PKCS12 ファイル名: 最大 63 文字
   - パスワードのインポート: 最大 31 文字
   - PEM パスフレーズ: 最大 31 文字
   - PEM パスフレーズの確認: 最大 31 文字

6. PKCS12 をエクスポート

   - PKCS12 ファイル名: 最大 63 文字
   - 証明書ファイル名: 最大 63 文字
   - キーファイル名: 最大 63 文字
   - エクスポートパスワード: 最大 31 文字
   - PEM パスフレーズ: 最大 31 文字

7. CRL 管理:

   - CA 証明書ファイル名: 最大 63 文字
   - CA キーファイル名: 最大 63 文字
   - CA キーファイルパスワード: 最大 31 文字
   - インデックスファイル名: 最大 63 文字
   - 証明書ファイル名: 最大 63 文字

8. RSA キーの作成:

   - キーファイル名: 最大 63 文字
   - キーサイズ: 最大 4096 ビット
   - PEM パスフレーズ: 最大 31 文字
   - パスフレーズの確認: 最大 31 文字

9. SSL の詳細設定を変更します。

   - 最大 CRL メモリサイズ: 最大 1024 M バイト
   - 暗号化トリガタイムアウト (10 mS ティック): 最大 200
   - 暗号化トリガーパケット数: 最大 50
   - OCSP キャッシュサイズ: 最大 512 MB

10. 証明書のインストール:

- 証明書とキーのペア名: 最大 31 文字
- 証明書ファイル名: 最大 63 文字
- 秘密キーファイル名: 最大 63 文字
- パスワード: 最大 31 文字
- 通知期間: 最大 100

11. 暗号グループの作成:

    - 暗号グループ名: 最大 39 文字

12. CRL の作成:

    - CRL 名: 最大 31 文字
    - CRL ファイル: 最大 63 文字
    - URL: 最大 127 文字
    - ベース DN: 最大 127 文字
    - バインド DN: 最大 127 文字
    - パスワード: 最大 31 文字
    - 日数: 最大 31

13. SSL ポリシーの作成:

    - 名前: 最大 127 文字

14. SSL アクションの作成:

    - 名前: 最大 127 文字

15. OCSP レスポンダーの作成:

    - 名前: 最大 32 文字
    - URL: 最大 128 文字
    - バッチの深さ: 最大 8
    - バッチ処理遅延: 最大 10000
    - タイムスキューで生産: 最大 86400
    - リクエストタイムアウト: 最大 120000

16. 仮想サーバーの作成:

    - 名前: 最大 127 文字
    - リダイレクト URL: 最大 127 文字
    - クライアントのタイムアウト: 最大 31536000 秒

17. サービスの作成:

    - 名前: 最大 127 文字

- アイドルタイムアウト (秒):
  クライアント: 最大 31536000
  サーバー: 最大 31536000

18. サービスグループの作成:

    - サービスグループ名: 最大 127 文字
    - サーバー ID: 最大 4294967295
    - アイドルタイムアウト (秒):
      クライアント: 最大値 31536000
      サーバー: 最大 31536000

19. モニターの作成:

    - 名前: 最大 31 文字

20. サーバーを作成:

    - サーバー名: 最大 127 文字
    - ドメイン名: 最大 255 文字
    - 解決再試行: 最大 20939 秒

コンテンツ検査

August 15, 2023

最近では、さまざまなマルチメディアコンテンツを表示するためにデバイスの種類も増えています。デバイスの種類は、携帯電話からタブレット、デスクトップまでさまざまです。中間インフラストラクチャプロバイダーは、Web サーバーからコンテンツを要求するデバイスに適した形式へと、元のコンテンツを変換する必要があります。外部デバイスは、トランスコードするコンテンツを検査してから、クライアントに送り返します。このために一般的に使用されるプロトコルは、ICAP です。ICAP を使用すると、NetScaler アプライアンスをさまざまな展開に配置できます。ICAP は、マルウェアやセキュリティの問題についてデータを検査する、コンテンツ検査技術を使用します。

> 注
>
> HTTP/2 は、コンテンツ検査とは互換性がありません。を使用するアプリケーション HTTP/2 トラフィックがコンテンツ検査を介して送信される場合、正しく機能しない可能性があります。

リモートコンテンツ検査用の **ICAP**

August 15, 2023

インターネットコンテンツ適応プロトコル (ICAP) は、HTTP メッセージで付加価値変換サービスを実行するためのシンプルで軽量なプロトコルです。一般的なシナリオでは、ICAP クライアントは HTTP 要求と応答を 1 つ以上の ICAP サーバーに転送して処理します。ICAP サーバーは、要求に対してコンテンツ変換を実行し、要求または応答に対して実行する適切なアクションを含む応答を送り返します。

NetScaler ADC アプライアンス上の ICAP

NetScaler ADC セットアップでは、アプライアンスはサードパーティの ICAP サーバー（マルウェア対策やデータ損失保護（DLP）など）と相互運用する ICAP クライアントとして機能します。アプライアンスが着信 Web トラフィックを受信すると、アプライアンスはトラフィックをインターセプトし、コンテンツ検査ポリシーを使用して HTTP 要求に ICAP 処理が必要かどうかを評価します。「はい」の場合、アプライアンスはメッセージを復号化し、プレーンテキストとして ICAP サーバーに送信します。ICAP サーバーは、要求メッセージに対してコンテンツ変換サービスを実行し、アプライアンスに応答を送り返します。適応されたメッセージは、HTTP リクエストまたは HTTP レスポンスのいずれかです。アプライアンスが複数の ICAP サーバーと相互運用する場合、アプライアンスは ICAP サーバーの負荷分散を実行します。このシナリオは、1 つの ICAP サーバーではすべてのトラフィック負荷を処理できない場合に発生します。ICAP サーバーが変更されたメッセージを返すと、アプライアンスは変更されたメッセージをバックエンドのオリジンサーバーに転送します。

NetScaler ADC アプライアンスは、着信トラフィックが HTTPS タイプの場合、安全な ICAP サービスも提供します。アプライアンスは SSL ベースの TCP サービスを使用して、アプライアンスと ICAP サーバー間の安全な接続を確立します。

ICAP リクエスト変更 (REQMOD) の仕組み

要求変更（REQMOD）モードでは、NetScaler ADC アプライアンスはクライアントから受信した HTTP 要求を ICAP サーバーに転送します。次に、ICAP サーバーは次のいずれかを実行します。

1. リクエストの変更バージョンを送り返し、アプライアンスは変更されたリクエストをバックエンドオリジンサーバーに送信するか、変更されたリクエストを別の ICAP サーバーにパイプライン化します。
2. 適応が必要ないことを示すメッセージで応答します。
3. エラーを返し、アプライアンスはエラーメッセージをユーザーに送信します。

ICAP 応答修正 (RESPMOD) の仕組み

応答変更（RESPMOD）モードでは、NetScaler ADC アプライアンスは HTTP 応答を ICAP サーバーに送信します（アプライアンスによって送信される応答は通常、オリジンサーバーから送信された応答です）。次に、ICAP サーバーは次のいずれかを実行します。

1. 応答の修正バージョンを送信し、アプライアンスは応答をユーザーに送信するか、応答を別の ICAP サーバーにパイプライン化します。

2. 適応が必要ないことを示すメッセージで応答します。
3. エラーを返し、アプライアンスはエラーメッセージをユーザーに送信します。

ICAP ライセンス

ICAP 機能は、NetScaler ADC スタンドアロンまたは NetScaler ADC Premium または Advanced ライセンスエディションの高可用性セットアップで動作します。

コンテンツ変換サービスの **ICAP** を構成する

コンテンツ変換サービスに ICAP を使用するには、まずコンテンツ検査と負荷分散機能を有効にする必要があります。機能を有効にすると、次のタスクを完了できます

コンテンツ検査を有効にするには

NetScaler ADC アプライアンスを ICAP クライアントとして機能させる場合は、まずコンテンツ検査と負荷分散機能を有効にする必要があります。

コマンドプロンプトで入力します。

```
enable ns feature contentInspection LoadBalancing
<!--NeedCopy-->
```

ICAP プロファイルを追加

NetScaler ADC アプライアンスの ICAP 構成は、ICAP プロファイルと呼ばれるエンティティで指定されます。プロファイルには ICAP 設定のコレクションがあります。設定には、ICAP 要求を動的に生成し、ICAP 応答を受信し、コンテンツ検査データをログに記録するパラメータが含まれます。

ICAP サーバーへの ICAP 要求を動的に生成するために、新しいパラメーター「insertHttpRequest」が ICAP プロファイルに追加されます。このパラメータが設定されている場合、アプライアンスは構成された値をポリシー式として受け取り、式を評価し、その結果をカプセル化された HTTP 要求または応答として含めて、ICAP サーバーに送信します。また、新しいパラメータ「insertiCapHeaders」は、ICAP ヘッダーを動的に評価して含めるように構成できます。

アプライアンスが ICAP 要求を送信し、ICAP サーバーへの応答を受信しない場合、接続は応答しなくなります。ICAP サーバーが応答を送信するか、セッションが解放されるまで発生します。この動作は、ICAP 応答タイムアウトオプションを設定することで処理できます。ICAP 応答が遅れている場合は、アクションのリクエストタイムアウトパラメータを設定できます。NetScaler ADC アプライアンスが構成された要求タイムアウト内に応答を受信しない場合、要求タイムアウトアクションが実行されます。

reqTimeoutAction: 可能な値は、バイパス、リセット、ドロップです。
BYPASS: リモート ICAP サーバーの応答を無視し、要求/応答をクライアント/サーバーに送信します。
RESET (デフォルト): クライアント接続を閉じてリセットします。
DROP: ユーザーに応答を送信せずにリクエストをドロップします

ICAP 応答を評価するために、コンテンツ検査コールアウト戻り値式で新しいポリシー式ICAP.RESが使用されます。この式は、HTTP_CALLOUTのHTTP.RES式に似た ICAP 応答を評価します。

たとえば、NetScaler ADC アプライアンスが NetScaler ADC 仮想 IP アドレスの背後でホストされているサービスの HTTP 要求を受信すると、アプライアンスは外部サーバーとのクライアントの認証を確認し、アクションを実行する必要があります。

コマンドプロンプトで入力します。

```
add ns icapProfile <name> [-preview ( ENABLED | DISABLED )][-previewLength
 <positive_integer>] -uri <string> [-hostHeader <string>] [-userAgent
 <string>] -Mode ( REQMOD | RESPMOD )[-queryParams <string>] [-
connectionKeepAlive ( ENABLED | DISABLED )][-allow204 ( ENABLED
 | DISABLED )] [-insertICAPHeaders <string>][-insertHTTPRequest
 <string>] [-reqTimeout <positive_integer>][-reqTimeoutAction <
reqTimeoutAction>] [-logAction <string>]
```

例:

```
add icapprofile reqmod-profile -mode RESPMOD -uri “/req_scan” -
hostHeader “Webroot.reqsca” -useragent “NS_SWG-Proxy”
```

```
add ns icapProfile icap_prof1 -uri "/example"-Mode REQMOD -reqtimeout
 4 -reqtimeoutaction BYPASS
```

```
> add icapProfile reqmode-profile -uri '/example'-mode REQMOD -
insertHTTPRequest q{ HTTP.REQ.METHOD + ""+ HTTP.REQ.URL + "HTTP/1.1\r
\n"+ "Host: "+ HTTP.REQ.HOSTNAME + "\r\n\r\n"}
```

ICAP コンテンツ検査アクションを記録する　コンテンツ検査ログストリームレコードまたは SYSLOG ログを動的に生成するには、ICAP 応答で ICAP.RES ベースのポリシー式を使用できます。このパラメーターは、動的ログレコードを生成するポリシー式を構成するために ICAP プロファイルで設定できます。

コマンドプロンプトで入力します。

```
add audit messageaction icap_log_expr INFORMATIONAL icap.res.full_header
```

```
set icapProfile reqmode-profile -logAction messageaction
```


ICAP サービスを **TCP** または **SSL_TCP** サービスとして追加する

コンテンツ検査機能を有効にしたら、負荷分散設定の一部となる ICAP サーバー用の ICAP サービスを追加する必要があります。追加するサービスは、NetScaler ADC アプライアンスと負荷分散仮想サーバー間の ICAP 接続を提供します。

注: 管理者は、コンテンツ検査アクションで ICAP サービスを追加し、ICAP サーバーの IP アドレスを直接設定できます。

コマンドプロンプトで、次のように入力します。

```
add service <name> <IP> <serviceType> <port>
<!--NeedCopy-->
```

例:

```
add service icapsv1 10.10.10.10 SSL_TCP 1345
add service icapsv2 10.10.10.11 SSL_TCP 1345
```

TCP または **SSL_TCP** ベースの負荷分散仮想サーバーを追加する

ICAP サービスを作成したら、ICAP トラフィックを受け入れ、ICAP サーバーの負荷を分散する仮想サーバーを作成する必要があります。

> 注:
>
> SSL ベースの TCP サービスは、セキュリティで保護されたチャネルで使用することもできます。SSL_TCP サービスを使用し、コンテンツ検査アクションにバインドします。

コマンドプロンプトで、次のように入力します。

```
add lb vserver <name> <serviceType> <port>
<!--NeedCopy-->
```

例:

```
add lb vserver vicap TCP 0.0.0.0 - persistenceType NONE -cltTimeout
    9000

add lb vserver vicap SSL_TCP 0.0.0.0 0 - persistenceType NONE -
    cltTimeout 9000
<!--NeedCopy-->
```

ICAP サービスを負荷分散仮想サーバーにバインドする

ICAP サービスと仮想サーバーを作成したら、ICAP サービスを仮想サーバーにバインドする必要があります。

コマンドプロンプトで、次のように入力します。


```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:

```
bind lb vserver vicap icapsv1
<!--NeedCopy-->
```

コンテンツ検査アクションを追加

コンテンツ検査機能を有効にしたら、ICAP 要求情報を処理するための ICAP アクションを追加する必要があります。作成された ICAP プロファイルとサービス、または負荷分散仮想サーバーは、ICAP アクションにバインドされます。ICAP サーバーがダウンしている場合は、次のアクションのいずれかを実行するようにアプライアンスの`ifserverdown`パラメータを設定できます。

CONTINUE：リモートサーバーがダウンしているときにユーザーがコンテンツ検査をバイパスしたい場合は、デフォルトで「CONTINUE」アクションを選択できます。
RESET（デフォルト）：このアクションは、RST との接続を閉じることによってクライアントに応答します。
DROP: このアクションは、ユーザーに応答を送信せずにパケットをサイレントにドロップします。

コマンドプロンプトで、次のように入力します。

```
add contentInspection action <name> -type ICAP -serverName <string> -
    icapProfileName <string>

add ContentInspection action <name> -type ICAP -serverip <ip> -
    serverport <port> -icapProfileName <string>
<!--NeedCopy-->
```

> 注:
>
> 負荷分散仮想サーバーの代わりに ICAP サービスを構成できる場合は、`\<-serverip>` オプションにサービス名を指定できます。コンテンツ検査アクションを追加すると、ポート 1344 の指定された IP アドレスに対して TCP サービスが自動的に作成され、ICAP 通信に使用されます。

例:

```
add ContentInspection action ci_act_lb -type ICAP -serverName vicap -
    icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverName icapsv1
    -icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverip 1.1.1.1 -
    serverport 1344 -icapProfileName icap_reqmod
<!--NeedCopy-->
```


コンテンツ検査ポリシーを追加する

コンテンツ検査アクションを作成したら、コンテンツ検査ポリシーを作成して、ICAP 処理と監査ログの要求を評価する必要があります。このポリシーは、1 つ以上の式で構成されるルールに基づいています。ルールは、リクエストがルールと一致する場合に関連付けられるコンテンツ検査アクションに関連付けられます。

コマンドプロンプトで、次のように入力します。

```
add contentInspection policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

例:

```
add ContentInspection policy ci_pol_basic  - rule true  - action
    ci_act_svc

add ContentInspection policy ci_pol_HTTP  - rule HTTP.REQ.URL.CONTAINS(
    “html” )  - action ci_act_svc
<!--NeedCopy-->
```

コンテンツ検査ポリシーをコンテンツスイッチまたは負荷分散仮想サーバーにバインドする

ICAP ポリシーを有効にするには、そのポリシーをグローバルにバインドするか、アプリケーションのフロントエンドであるコンテンツスイッチングまたは負荷分散仮想サーバーにバインドする必要があります。ポリシーをバインドするときは、優先度を割り当てる必要があります。プライオリティによって、定義したポリシーが評価される順序が決まります。

> 注:
>
> アプリケーション仮想サーバーのタイプは、HTTP/SSL/CS-PROXY である必要があります。

コンテンツ変換後にトラフィックをバックエンドオリジンサーバーに転送するための負荷分散設定の構成については、「負荷分散」を参照してください。

セキュリティで保護された **ICAP** サービスの構成

NetScaler ADC アプライアンスと ICAP Web サーバー間の安全な接続を確立するために、アプライアンスは ICAP アクションにバインドされた SSL ベースの TCP サービスまたは負荷分散仮想サーバーを使用します。

セキュリティで保護された ICAP 接続を確立するには、次のタスクを実行します。

1. SSL ベースの TCP サービスを追加します。
2. SSL ベースの TCP サービスを、TCP または SSL_TCP タイプの負荷分散仮想サーバーにバインドします。
3. SSL ベースの TCP サービスまたは負荷分散仮想サーバーをコンテンツ検査アクションにバインドします。

SSL ベースの **TCP** サービスを負荷分散仮想サーバーに追加する

NetScaler ADC アプライアンスと ICAP Web サーバー間の安全な接続を確立するために、アプライアンスは ICAP アクションにバインドされた SSL ベースの TCP サービスまたは負荷分散仮想サーバーを使用します。

セキュリティで保護された ICAP 接続を確立するには、次のタスクを実行します。

1. SSL ベースの TCP サービスを追加します。
2. SSL ベースの TCP サービスを、TCP または SSL_TCP タイプの負荷分散仮想サーバーにバインドします。

SSL ベースの TCP サービスまたは負荷分散仮想サーバーをコンテンツ検査アクションにバインドする

SSL ベースの **TCP** サービスを負荷分散仮想サーバーに追加する

コンテンツ検査機能を有効にしたら、負荷分散設定の一部となるセキュリティで保護された ICAP サービスを追加する必要があります。追加するサービスは、NetScaler ADC アプライアンスと負荷分散仮想サーバー間の安全な ICAP 接続を提供します。

コマンドプロンプトで、次のように入力します。

```
add service <name> <IP> <serviceType> <port>
<!--NeedCopy-->
```

例**:**

```
add service icapsv2 10.102.29.200 SSL_TCP 1344 -gslb NONE -maxclient
    0 -maxReq 0 -cip DISABLED -usip NO -useproxport YES -sp ON -
    cltTimeout 9000 -svrTimeout 9000 -CKA NO -TCPB NO -CMP NO
<!--NeedCopy-->
```

SSL ベースの **TCP** サービスを **SSL_TCP** または **TCP** 負荷分散仮想サーバーにバインドする

セキュリティで保護された ICAP サービスを作成したら、サービスを負荷分散仮想サーバーにバインドする必要があります。負荷分散仮想サーバーを使用して ICAP サーバーの負荷を分散する場合に必要です。

コマンドプロンプトで、次のように入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

```
bind lb vserver vicap icapsv2
<!--NeedCopy-->
```


SSL ベースの **TCP** サービスまたは負荷分散仮想サーバーをコンテンツ検査アクションにバインドする

ICAP 要求情報を処理する ICAP アクションを追加し、SSL ベースの TCP サービスをアクションにバインドします。

コマンドプロンプトで、次のように入力します。

```
add contentInspection action <name> -type ICAP -serverName <string> -
    icapProfileName <string>
<!--NeedCopy-->
```

例**:**

```
add ContentInspection action ci_act_svc -type ICAP -serverName icapsv2
    -icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverName vicap -
    icapProfileName icap_reqmod
<!--NeedCopy-->
```

GUI を使用して **ICAP** プロトコルを構成する

1. 負荷分散 > サービスに移動し、追加をクリックします。
2. [サービス] ページで、サービスの詳細を入力します。
3. [負荷分散] > [仮想サーバー] に移動します。HTTP/SSL タイプの負荷分散仮想サーバーを追加します。または、仮想サーバーを選択して [編集**]** をクリックすることもできます。
4. サーバーの基本情報を入力したら、[続行] をクリックします。
5. **[**詳細設定**]** セクションで、[ポリシー] をクリックします。
6. [ポリシー] セクションに移動し、鉛筆アイコンをクリックして、コンテンツ検査ポリシーを設定します。
7. 「ポリシーの選択」ページで、「コンテンツ検査」を選択します。［続行］をクリックします。
8. [ポリシーバインディング] セクションで、[**+**] をクリックしてコンテンツ検査ポリシーを追加します。
9. **[ICAP** ポリシーの作成] ページで、ポリシーの名前を入力します。
10. [アクション] フィールドで、[+] 記号をクリックして ICAP アクションを追加します。
11. **[ICAP** アクションの作成] ページで、アクションの名前を入力します。
12. アクションの名前を入力します。
13. [**Server Name**] フィールドに、すでに作成されている TCP サービスの名前を入力します。
14. [**ICAP** プロファイル] フィールドで、[+] 記号をクリックして ICAP プロファイルを追加します。
15. [**ICAP** プロファイルの作成] ページで、プロファイル名、URI、および MODE を入力します。
16. ［作成］をクリックします。
17. [**ICAP** アクションの作成] ページで、[作成] をクリックします。
18. [**ICAP** ポリシーの作成] ページで、式エディタに「true」と入力し、[作成] をクリックします。
19. ［バインド］をクリックします。
20. コンテンツ検査機能を有効にするかどうかを確認するメッセージが表示されたら、[はい] をクリックします。

21. ［完了］をクリックします。

コンテンツ変換後にトラフィックをバックエンドオリジンサーバーに負荷分散および転送するための NetScaler ADC GUI 構成の詳細については、「負荷分散」を参照してください。

GUI を使用してセキュリティで保護された **ICAP** プロトコルを構成する

1. 負荷分散 > サービスに移動し、追加をクリックします。
2. [サービス] ページで、サービスの詳細を入力します。
3. [負荷分散] > [仮想サーバー] に移動します。HTTP/SSL タイプの仮想サーバーを追加します。または、仮想サーバーを選択して [編集**]** をクリックすることもできます。
4. サーバーの基本情報を入力したら、[続行] をクリックします。
5. **[**詳細設定**]** セクションで、[ポリシー] をクリックします。
6. [ポリシー] セクションに移動し、鉛筆アイコンをクリックして、コンテンツ検査ポリシーを設定します。
7. 「ポリシーの選択」ページで、「コンテンツ検査」を選択します。［続行］をクリックします。
8. [ポリシーバインディング] セクションで、[**+**] をクリックしてコンテンツ検査ポリシーを追加します。
9. **[ICAP** ポリシーの作成] ページで、ポリシーの名前を入力します。
10. [アクション] フィールドで、[+] 記号をクリックして ICAP アクションを追加します。
11. **[ICAP** アクションの作成] ページで、アクションの名前を入力します。
12. アクションの名前を入力します。
13. 「サーバー名」フィールドに、作成済みの TCP_SSL サービスの名前を入力します。
14. [**ICAP** プロファイル] フィールドで、[+] 記号をクリックして ICAP プロファイルを追加します。
15. [**ICAP** プロファイルの作成] ページで、プロファイル名、URI、および MODE を入力します。
16. ［作成］をクリックします。
17. [**ICAP** アクションの作成] ページで、[作成] をクリックします。
18. [**ICAP** ポリシーの作成] ページで、式エディタに「true」と入力し、[作成] をクリックします。
19. ［バインド］をクリックします。
20. コンテンツ検査機能を有効にするかどうかを確認するメッセージが表示されたら、[はい] をクリックします。
21. ［完了］をクリックします。

リモートコンテンツ検査の監査ログサポート

着信要求または送信応答がコンテンツ検査された場合、NetScaler ADC アプライアンスは ICAP の詳細を記録します。アプライアンスは、詳細をログ・メッセージとして ns.log ファイルに保存します。

各ログメッセージには通常、次の詳細が含まれます。

```
1 <Source IP> <Destination IP> <Domain> <ICAP server IP><ICAP Mode> <
     Service URI> <ICAP response> <Policy action>
2 <!--NeedCopy-->
```

制限:App Firewall のストリーミングモードは、コンテンツ検査機能ではサポートされていません。

コンテンツ検査要求ログメッセージの例**:**

```
Apr 18 14:45:41 <local0.info> 10.106.97.104 04/18/2018:14:45:41 GMT 0-PPE-0 : default CI ICAP_LOG 788 0 : Source 10.102.1.98:39048 - Destination 10.106.97.89:8011 - Domain 10.106.97.89 - Content-Type application/x-www-form-urlencoded - ICAP Server 10.106.97.99:1344 - Mode REQMOD - Service /example - Response 204 - Action FORWARD
<!--NeedCopy-->
```

コンテンツ検査された応答ログメッセージの例:

```
Apr 18 12:34:08 <local0.info> 10.106.97.104 04/18/2018:12:34:08 GMT 0-PPE-0 : default CI ICAP_LOG 71 0 : Source 10.106.97.105:18552 - Destination 10.106.97.99:80 - Domain NA - Content-Type NA - ICAP Server 10.106.97.99:1344 - Mode RESPMOD - Service /example - Response 400 - Action Internal Error
<!--NeedCopy-->
```

NetScaler とのインラインデバイス統合

December 8, 2023

侵入防止システム（IPS）や次世代ファイアウォール（NGFW）などのセキュリティデバイスは、ネットワーク攻撃からサーバーを保護します。これらのデバイスはレイヤー 2 インラインモードで導入され、その主な機能はサーバーをネットワーク攻撃から保護し、ネットワーク上のセキュリティ脅威を報告することです。

脆弱な脅威を防ぎ、高度なセキュリティ保護を提供するために、NetScaler アプライアンスは 1 つ以上のインラインデバイスと統合されています。インラインデバイスは、IPS、NGFW など、どのセキュリティデバイスでもかまいません。

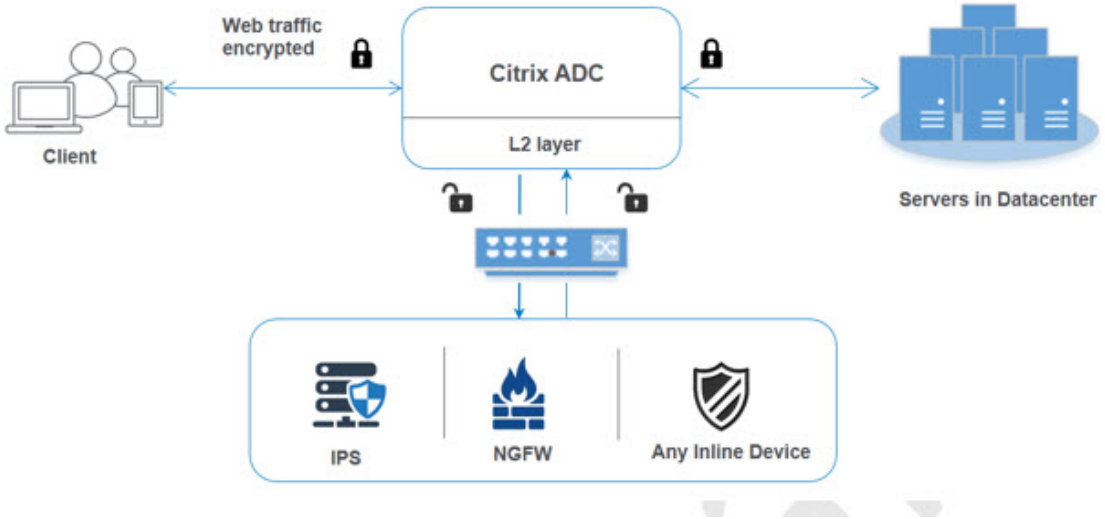
NetScaler アプライアンスとインラインデバイス統合を使用することでメリットが得られるユースケースのいくつかを以下に示します。

- 暗号化されたトラフィックを検査する。ほとんどの IPS および NGFW アプライアンスは暗号化されたトラフィックをバイパスするため、サーバーは攻撃に対して脆弱なままになります。NetScaler アプライアンスはトラフィックを復号化し、インラインデバイスに送信して検査することができます。これにより、お客様のネットワークセキュリティが強化されます。
- インラインデバイスを **TLS/SSL** 処理からオフロードする。TLS/SSL 処理にはコストがかかるため、IPS または NGFW アプライアンスがトラフィックを復号化すると、システム CPU の使用率が高くなる可能性があります。暗号化されたトラフィックが急速に増加するにつれて、これらのシステムは暗号化されたトラフィックの復号化と検査に失敗します。NetScaler は、インラインデバイスを TLS/SSL 処理からオフロードするのに役立ちます。その結果、インラインデバイスは大量のトラフィック検査をサポートすることになります。
- インラインデバイスの負荷分散。NetScaler アプライアンスは、大量のトラフィックがある場合に複数のインラインデバイスの負荷分散を行います。

- トラフィックのスマートな選択。アプライアンスに流れ込むすべてのパケットは、テキストファイルのダウンロードなど、内容が検査される場合があります。ユーザーは、NetScaler アプライアンスを構成して、検査対象として特定のトラフィック（.exe ファイルなど）を選択し、そのトラフィックをインラインデバイスに送信してデータを処理できます。

NetScaler をインラインデバイスと統合する方法

次の図は、NetScaler がインラインセキュリティデバイスとどのように統合されるかを示しています。



インラインデバイスを NetScaler アプライアンスと統合すると、コンポーネントは次のように相互作用します。

1. クライアントが NetScaler アプライアンスにリクエストを送信します。
2. アプライアンスは要求を受信し、ポリシー評価に基づいてインラインデバイスに送信します。
   注: インラインデバイスが 2 つ以上ある場合、アプライアンスはデバイスの負荷分散を行い、トラフィックを送信します。
   着信トラフィックが暗号化されたものである場合、アプライアンスはデータを復号化し、コンテンツを検査するためにプレーンテキストとしてインラインデバイスに送信します。
3. インラインデバイスは、データの脅威を検査し、データをドロップ、リセット、またはアプライアンスに戻すかどうかを決定します。
4. セキュリティ上の脅威がある場合、デバイスはデータを修正してアプライアンスに送信します。
5. NetScaler はデータを再暗号化し、リクエストをバックエンドサーバーに転送します。
6. バックエンドサーバーは、NetScaler アプライアンスに応答を送信します。
7. アプライアンスは再びデータを復号化し、検査のためにインラインデバイスに送信します。
8. アプライアンスはデータを再暗号化し、応答をクライアントに送信します

ソフトウェアライセンス

インラインデバイス統合を展開するには、NetScaler アプライアンスに次のいずれかのライセンスをプロビジョニングする必要があります。

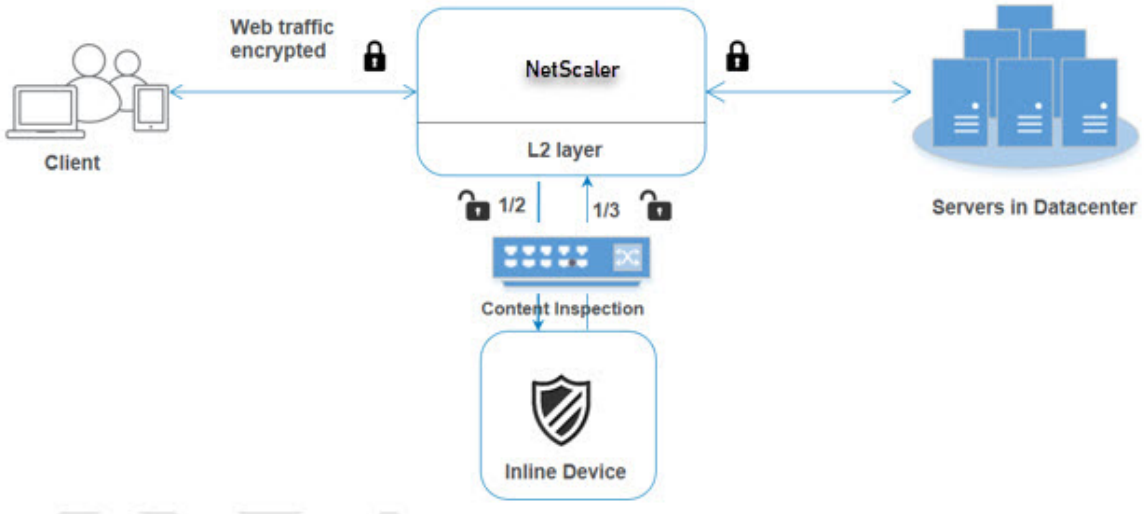
1. ADC Premium
2. ADC Advanced
3. Telco Advanced
4. Telco プレミアム
5. SWG ライセンス

インラインデバイス統合の設定

NetScaler アプライアンスをインラインデバイスで構成するには、3 つの方法があります。設定シナリオは次のとおりです。

1 つのインラインデバイスを使用する場合のシナリオ **1**

セキュリティデバイス（IPS または NGFW）をインラインモードで統合する場合は、まずコンテンツ検査機能を有効にし、NetScaler in MBF（MAC ベースの転送）をグローバルモードで有効にする必要があります。機能を有効にしたら、コンテンツ検査プロファイルを追加し、インラインデバイスで検査に基づいてトラフィックをリセット、ブロック、またはドロップするためのコンテンツ検査アクションを追加する必要があります。次に、アプライアンスにコンテンツ検査ポリシーを追加して、インラインデバイスに送信するトラフィックのサブセットを決定します。次に、サーバー上でレイヤー 2 接続が有効になっている負荷分散仮想サーバーを構成します。最後に、コンテンツ検査ポリシーを負荷分散仮想サーバーにバインドします。



MBF (MAC ベース転送**)** モードを有効にする　NetScaler アプライアンスを IPS やファイアウォールなどのインラインデバイスに統合する場合は、このモードを有効にする必要があります。MBF の詳細については、「MAC ベースの転送の設定」トピックを参照してください。

コマンドプロンプトで入力します:

```
enable ns mode mbf
```

コンテンツ検査を有効にする　NetScaler アプライアンスにコンテンツを復号化して検査用にインラインデバイスに送信する場合は、コンテンツ検査と負荷分散機能を有効にする必要があります。

```
enable ns feature contentInspection LoadBalancing
```

レイヤ **2** 接続方法の追加　インラインデバイスから生成された応答を処理するために、アプライアンスはインラインデバイスとの通信のレイヤー 2 方式（L2ConnMethod）として VLAN チャネルを使用します。

コマンドプロンプトで入力します:

```
set l4param -l2ConnMethod <l2ConnMethod>
```

例

```
set l4param -l2ConnMethod VlanChannel
```

サービスのコンテンツ検査プロファイルの追加　NetScaler アプライアンスのインラインデバイス構成は、コンテンツ検査プロファイルと呼ばれるエンティティで指定できます。プロファイルには、インラインデバイスとの統合方法を説明する設定のコレクションがあります。

コマンドプロンプトで入力します:

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <
positive_integer>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile Inline_profile1 -type InlineInspection
-ingressinterface “1/2” -egressInterface “1/3”
```

IPS-TCP モニターの追加　モニターを設定する場合は、ユーザー定義のモニターを追加します。
注: モニターを設定する場合は、カスタムモニターを使用する必要があります。モニターを追加するときは、transparent パラメーターを有効にする必要があります。

コマンドプロンプトで入力します:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr>] [-
destPort <port>] [-transparent ( YES | NO )]
```


例:

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
 YES
```

サービスを追加する　サービスを追加します。インラインデバイスを含め、どのデバイスも所有していないダミー IP アドレスを指定します。use source IP address (USIP) を「はい」に設定します。useproxyportを NO に設定します。デフォルトでは、ヘルスモニタリングはオンになっており、サービスをヘルスモニターにバインドし、モニターの TRANSPARENT オプションもオンに設定します。コマンドプロンプトで入力します:

```
add service <Service_name> <IP> TCP * - contentinspectionProfileName
<Name> -healthMonitor YES -usip ON -useproxyport OFF
```

例:

```
add service ips_service 192.168.10.2 TCP * -healthMonitor YES -usip
YES -useproxyport NO -contentInspectionProfileName ipsprof
```

ヘルスモニターの追加　デフォルトでは、ヘルスモニターはオンになっており、必要に応じて無効にすることもできます。コマンドプロンプトで入力します:

```
add lb monitor <name> TCP -destIP <ip address> -destPort 80 -transparent
 <YES, NO>
```

例:

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
 YES
```

サービスをヘルスモニターにバインドする　ヘルスモニターを設定したら、サービスをヘルスモニターにバインドする必要があります。コマンドプロンプトで入力します:

```
bind service <name> -monitorName <name>
```

例:

```
bind service ips_svc -monitorName ips_tcp
```

サービスのコンテンツ検査アクションを追加　コンテンツ検査機能を有効にし、インラインプロファイルとサービスを追加したら、要求を処理するためのコンテンツ検査アクションを追加する必要があります。コンテンツ検査アクションに基づいて、インラインデバイスはデータを検査した後にドロップ、リセット、またはブロックアクションを行うことができます。

インラインサーバーまたはサービスが停止している場合は、ifserverdown 次のアクションのいずれかを実行するようにアプライアンスのパラメータを設定できます。

CONTINUE: リモートサーバーがダウンしているときにユーザーがコンテンツ検査をバイパスしたい場合は、デフォルトで「CONTINUE」アクションを選択できます。
RESET (デフォルト): このアクションは、RST との接続を閉じることによってクライアントに応答します。
DROP: このアクションは、ユーザーに応答を送信せずにパケットをサイレントにドロップします。

コマンドプロンプトで入力します:

```
add contentInspection action <name> -type <type> (-serverName <string> [-ifserverdown <ifserverdown>] [-reqTimeout <positive_integer>] [-reqTimeoutAction <reqTimeoutAction>]
```

```
add ContentInspection action <action_name> -type InlineINSPECTION -serverName Service_name/Vserver_name>
```

例:

```
add ContentInspection action <Inline_action> -type InlineSPECTION -serverName Inline_service1
```

検査用のコンテンツ検査ポリシーを追加する　コンテンツ検査アクションを作成したら、コンテンツ検査ポリシーを追加して検査の要求を評価する必要があります。このポリシーは、1 つ以上の式で構成されるルールに基づいています。ポリシーは、ルールに基づいてインスペクション対象のトラフィックを評価し、選択します。

コマンドプロンプトで、次のように入力します:

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name>
```

例

```
add contentInspection policy Inline_pol1 -rule true -action Inline_action
```

HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーを追加する　Web トラフィックを受信するには、負荷分散仮想サーバーを追加する必要があります。また、仮想サーバ上で layer2 接続を有効にする必要があります。

コマンドプロンプトで入力します:

```
add lb vserver <name> <vserver name> -l2Conn ON
```

例:

```
add lb vserver HTTP_vserver HTTP 10.102.29.200 8080 -l2Conn ON
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプのコンテンツスイッチング仮想サーバーまたは負荷分散仮想サーバーにバインドする　負荷分散仮想サーバーまたは HTTP/SSL タイプのコンテンツスイッチング仮想サーバーをコンテ

ンツ検査ポリシーにバインドします。
コマンドプロンプトで、次のように入力します:

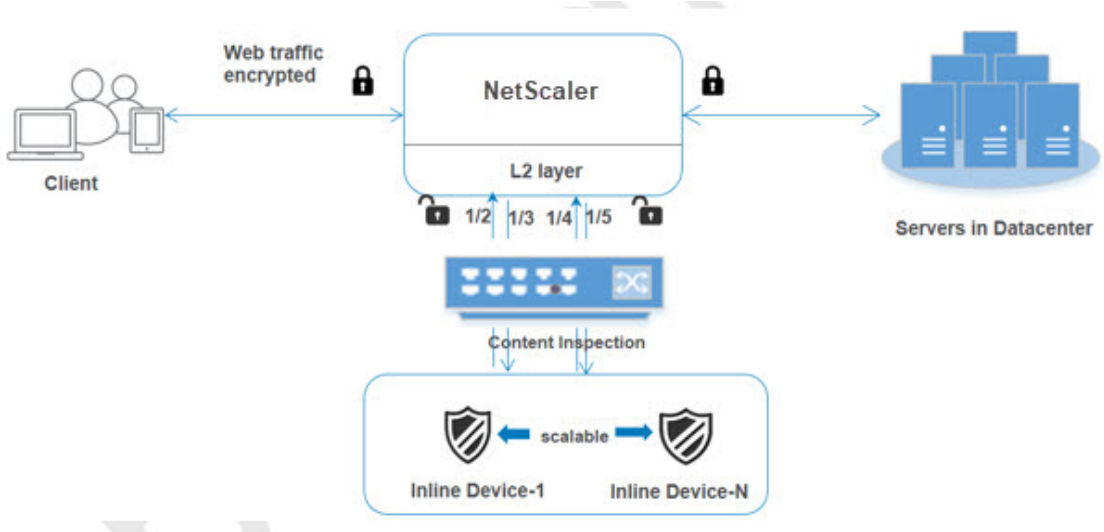
```
bind lb vserver <vserver name> -policyName < policy_name > -priority
< priority > -type <REQUEST>
```

例:

```
bind lb vserver HTTP_vserver -policyName Inline_pol1 -priority 100 -
type REQUEST
```

シナリオ **2:** 専用インターフェイスを使用した複数のインラインデバイスの負荷分散

2 つ以上のインラインデバイスを使用している場合は、専用の VLAN 設定で異なるコンテンツ検査サービスを使用してデバイスのロードバランシングを行う必要があります。この場合、NetScaler アプライアンスは、専用のインターフェイスを介して各デバイスにトラフィックのサブセットを送信することに加えて、デバイスの負荷分散を行います。基本的な設定手順については、シナリオ 1 を参照してください。



service1 のコンテンツ検査プロファイル **1** を追加　NetScaler アプライアンスのインライン構成は、コンテンツ検査プロファイルと呼ばれるエンティティで指定できます。プロファイルにはデバイス設定のコレクションがあります。コンテンツ検査プロファイル 1 はインラインサービス 1 用に作成され、通信は 1/2 および 1/3 専用インターフェイスを介して行われます。

コマンドプロンプトで入力します:

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <
positive_integer>] [-ingressVlan <positive_integer>]
```

例:


```
add contentInspection profile Inline_profile1 -type InlineInspection
-ingressinterface “1/2” -egressInterface “1/3”
```

service2 のコンテンツ検査プロファイル **2** を追加　コンテンツ検査プロファイル 2 が service2 に追加され、インラインデバイスは専用インターフェイスを介してアプライアンスと通信します。1/41/5
コマンドプロンプトで入力します:

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <
positive_integer>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile Inline_profile2 -type InlineInspection
-ingressinterface “1/4” -egressInterface “1/5”
```

インラインデバイス **1** にサービス **1** を追加　コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 1 のインラインサービス 1 を負荷分散設定の一部として追加する必要があります。追加したサービスによって、インライン構成の詳細がすべて提供されます。

コマンドプロンプトで入力します:

```
add service <Service_name_1> <Pvt_IP1> TCP * -contentInspectionProfileName
 <Inline_Profile_1> -healthmonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service Inline_service1 10.102.29.200 TCP 80 -contentInspectionProfileName
 Inline_profile1 -healthmonitor OFF -usip ON -useproxyport OFF
```

インラインデバイス **2** にサービス **2** を追加　コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 2 にインラインサービス 2 を追加する必要があります。追加したサービスによって、インライン構成の詳細がすべて提供されます。

コマンドプロンプトで入力します:

```
add service <Service_name_1> <Pvt_IP1> TCP * -contentInspectionProfileName
 <Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service Inline_service1 10.29.20.205 TCP 80 -contentInspectionProfileName
 Inline_profile2 -healthmonitor OFF -usip ON -useproxyport OFF
```


負荷分散仮想サーバの追加　インラインプロファイルとサービスを追加したら、サービスの負荷分散用の負荷分散仮想サーバを追加する必要があります。

コマンドプロンプトで入力します:

```
add lb vserver <vserver_name> TCP <Pvt_IP3> <port>
```

例:

```
add lb vserver lb-Inline_vserver TCP *
```

サービス **1** を負荷分散仮想サーバーにバインドします　負荷分散仮想サーバーを追加したら、負荷分散仮想サーバーを最初のサービスにバインドします。

コマンドプロンプトで入力します:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-Inline_vserver Inline_service1
```

サービス **2** を負荷分散仮想サーバーにバインドします　負荷分散仮想サーバーを追加したら、そのサーバーを 2 番目のサービスにバインドします。

コマンドプロンプトで入力します:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-Inline_vserver Inline_service2
```

サービスのコンテンツ検査アクションを追加　コンテンツ検査機能を有効にしたら、インラインリクエスト情報を処理するための「コンテンツ検査」アクションを追加する必要があります。選択したアクションに基づいて、インラインデバイスは特定のトラフィックのサブセットを検査した後、ドロップ、リセット、またはブロックします。

コマンドプロンプトで入力します:

```
add contentInspection action <name> -type <type> (-serverName <string
> [-ifserverdown <ifserverdown>] [-reqTimeout <positive_integer>] [-
reqTimeoutAction <reqTimeoutAction>]
```

```
add ContentInspection action < action_name > -type InlineINSPECTION -
serverName Service_name/Vserver_name>
```

例:

```
add ContentInspection action Inline_action -type InlineINSPECTION -
serverName lb-Inline_vserver
```


検査用のコンテンツ検査ポリシーを追加する　コンテンツ検査アクションを作成したら、サービス要求を評価するためのコンテンツ検査ポリシーを追加する必要があります。このポリシーは、1 つ以上の式で構成されるルールに基づいています。ルールは、リクエストがルールに一致する場合に関連するコンテンツ検査アクションに関連付けられます。

コマンドプロンプトで、次のように入力します:

```
add contentInspection policy <policy_name> -rule <Rule> -action <
action_name>
```

例:

```
add contentInspection policy Inline_pol1 -rule true -action Inline_action
```

HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーを追加する　Web トラフィックを受け入れるために、コンテンツスイッチングまたは負荷分散仮想サーバーを追加します。また、仮想サーバ上で layer2 接続を有効にする必要があります。
ロードバランシングの詳細については、「負荷分散の仕組み」トピックを参照してください。

コマンドプロンプトで入力します:

```
add lb vserver <name> <vserver name> -l2Conn ON
```

例:

```
add lb vserver http_vserver HTTP 10.102.29.200 8080 -l2Conn ON
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプの負荷分散仮想サーバーにバインドする　HTTP/SSL タイプのコンテンツスイッチング仮想サーバーまたは負荷分散仮想サーバーをコンテンツ検査ポリシーにバインドする必要があります。

コマンドプロンプトで、次のように入力します:

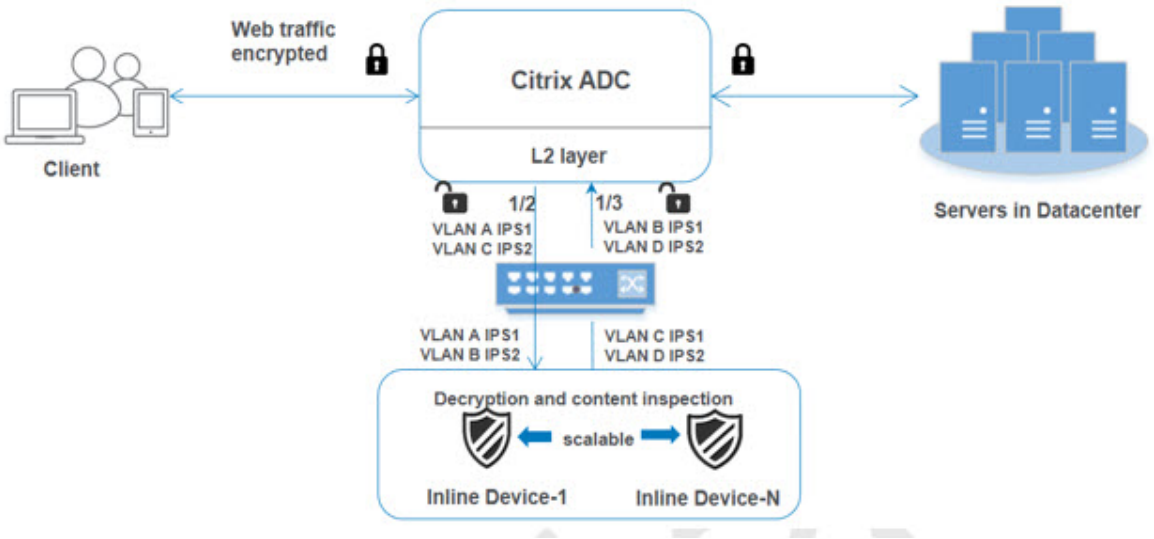
```
bind lb vserver <vserver name> -policyName < policy_name > -priority
<> -type <L7InlineREQUEST | L4Inline-REQUEST>
```

例:

```
bind lb vserver http_vserver -policyName Inline_pol1 -priority 100 -
type REQUEST
```

シナリオ **3:** 共有インターフェイスを使用した複数のインラインデバイスの負荷分散

複数のインラインデバイスを使用していて、共有 VLAN インターフェイスで異なるサービスを使用するデバイスのロードバランシングを行う場合は、この設定を参照できます。共有 VLAN インターフェイスを使用するこの設定は、ユースケース 2 と似ています。基本的な設定については、シナリオ 2 を参照してください。



共有オプションを有効にして **VLAN A** をバインドする　コマンドプロンプトで、次のように入力します:

```
bind vlan <id> -ifnum <interface> -tagged
```

例:

```
bind vlan 100 -ifnum 1/2 tagged
```

共有オプションを有効にして **VLAN B** をバインド　コマンドプロンプトで、次のように入力します:

```
bind vlan <id> -ifnum <interface> -tagged
```

例:

```
bind vlan 200 -ifnum 1/3 tagged
```

共有オプションが有効な状態で **VLAN C** をバインド　コマンドプロンプトで、次のように入力します:

```
bind vlan <id> -ifnum <interface> -tagged
```

例:

```
bind vlan 300 -ifnum 1/2 tagged
```

共有オプションを有効にして **VLAN D** をバインド　コマンドプロンプトで、次のように入力します:

```
bind vlan <id> -ifnum <interface> -tagged
```

例:

```
bind vlan 400 -ifnum 1/3 tagged
```


service1 のコンテンツ検査プロファイル **1** を追加　NetScaler アプライアンスのインライン構成は、コンテンツ検査プロファイルと呼ばれるエンティティで指定できます。プロファイルにはデバイス設定のコレクションがあります。コンテンツ検査プロファイルはインラインサービス 1 用に作成され、通信は 1/2 と 1/3 の専用インターフェイスを介して行われます。

コマンドプロンプトで入力します:

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <
positive_integer>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile Inline_profile1 -type InlineInspection
 -ingressinterface “1/2” -egressInterface “1/3” -egressVlan 100 -
ingressVlan 300
```

service2 のコンテンツ検査プロファイル **2** を追加　コンテンツ検査プロファイル 2 が service2 に追加され、インラインデバイスは専用インターフェイスを介してアプライアンスと通信します。1/21/3

コマンドプロンプトで入力します:

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <
positive_integer>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile Inline_profile2 -type InlineInspection
 -ingressinterface “1/2” -egressInterface “1/3” -egressVlan 200 -
ingressVlan 400
```

NetScaler GUI を使用してインラインサービス統合を構成する

1. NetScaler アプライアンスにログオンし、「構成」タブページに移動します。
2. [システム] > [設定**] > [**モードの設定] に移動します。
3. 「モードの設定」ページで、「**Mac** ベースの転送」を選択します。
4. 「**OK**」をクリックして「閉じる」をクリックします。
5. [システム] > [設定**] > [**拡張機能の設定] に移動します。
6. 「拡張機能の設定」ページで、「コンテンツ検査」を選択します。
7. 「**OK**」をクリックして「閉じる」をクリックします。
8. [セキュリティ] > [コンテンツ検査] > [コンテンツ検査プロファイル] に移動します。

9. 「コンテンツ検査プロファイル」ページで、「追加」をクリックします。
10. 「コンテンツ検査プロファイルの作成」ページで、次のパラメータを設定します。

    a) プロファイル名。コンテンツ検査プロファイルの名前。
    b) タイプ。プロファイルタイプをインライン検査として選択します。
    c) 出力インターフェイス。アプライアンスが NetScaler からインラインデバイスにトラフィックを送信するインターフェイス。
    d) 入力インターフェース。アプライアンスがインラインデバイスから NetScaler へのトラフィックを受信するインターフェイス。
    e) 出力 VLAN。トラフィックがインラインデバイスに送信されるインターフェイス VLAN ID。
    f) 入力側の VLAN。アプライアンスがインラインから NetScaler へのトラフィックを受信するインターフェイス VLAN ID（構成されている場合）。

11. **[**作成**]** して **[**閉じる**]** をクリックします。
12. トラフィック管理 > 負荷分散 > サービスに移動し、追加をクリックします。
13. 「サービス」ページで、次のパラメータを設定します。

    a) サービス名。負荷分散サービスの名前。
    b) IP アドレス。ダミー IP アドレスを使用してください。注: どのデバイスも IP アドレスを所有している必要はありません。
    c) プロトコル。プロトコルタイプを TCP として選択します。
    d) ポート。* を入力してください
    e) ヘルスモニタリング。サービスを TCP タイプのモニターにバインドする場合にのみ、このオプションをオフにして有効にしてください。モニターをサービスにバインドする場合は、モニターのTRANSPARENTオプションをオンにする必要があります。モニタの追加方法とサービスにバインドする方法については、手順 14 を参照してください。
    f) [**OK**] をクリックします。

14. 「設定」セクションで以下を編集し、「**OK**」をクリックします。

    a) プロキシポートを使用: オフにする
    b) 送信元 IP アドレスを使用: 有効にする

15. **[**詳細設定**]** セクションで、[プロファイル] をクリックします。
16. 「プロファイル」セクションに移動し、インラインコンテンツ検査プロファイルを追加して「**OK**」をクリックします。
17. **[** モニター**]** セクションに移動し、**[** バインディングを追加**]>[**モニター**]** > **[**追加**]** を選択します。

    a) 名前: モニターの名前
    b) タイプ:TCP タイプを選択
    c) 送信先 IP、ポート: 送信先 IP アドレスとポート。

d) 透明: オンにする

注: インラインデバイスのステータスを監視するには、監視パケットがインラインデバイスを経由する必要があります。

18. ［作成］をクリックします。
19. ［完了］をクリックします。
20. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。HTTP または SSL タイプの仮想サーバを追加します。
21. サーバーの詳細を入力したら、「**OK**」をクリックし、もう一度「**OK**」をクリックします。
22. 負荷分散仮想サーバーのトラフィック設定セクションで、レイヤー 2 パラメーターをオンにします。
23. **[**詳細設定**]** セクションで、[ポリシー] をクリックします。
24. 「ポリシー」セクションに移動し、「+」アイコンをクリックしてコンテンツ検査ポリシーを設定します。
25. 「ポリシーの選択」ページで、「コンテンツ検査」を選択します。［続行］をクリックします。
26. 「ポリシーバインディング」セクションで、「追加」をクリックしてコンテンツ検査ポリシーを追加します。
27. コンテンツ検査ポリシーの作成ページで、インラインコンテンツ検査ポリシーの名前を入力します。
28. 「アクション」フィールドで、「追加」をクリックしてインラインコンテンツ検査アクションを作成します。
29. 「**CI** アクションの作成」ページで、次のパラメータを設定します。

    a) Name：コンテンツ検査インラインポリシーの名前。
    b) タイプ。タイプをインライン検査として選択します。
    c) サーバー。サーバー/サービスをインラインデバイスとして選択します。
    d) サーバーがダウンした場合。サーバがダウンした場合のオペレーションを選択します。
    e) リクエストのタイムアウト。タイムアウト値を選択します。デフォルト値を使用できます。
    f) タイムアウトアクションの要求。タイムアウトアクションを選択します。デフォルト値を使用できます。

30. ［作成］をクリックします。
31. 「**CI** ポリシーの作成」ページで、その他の詳細を入力します。
32. 「**OK**」をクリックして「閉じる」をクリックします。

SSL フォワードプロキシを使用して **IPS** または **NGFW** をインラインデバイスとして統合

December 8, 2023

侵入防止システム（IPS）や次世代ファイアウォール（NGFW）などのセキュリティデバイスは、ネットワーク攻撃からサーバーを保護します。これらのデバイスはライブトラフィックを検査でき、通常はレイヤ 2 インラインモードで展開されます。SSL 転送プロキシアプライアンスは、インターネット上のリソースにアクセスする際に、ユーザーと企業ネットワークのセキュリティを確保します。

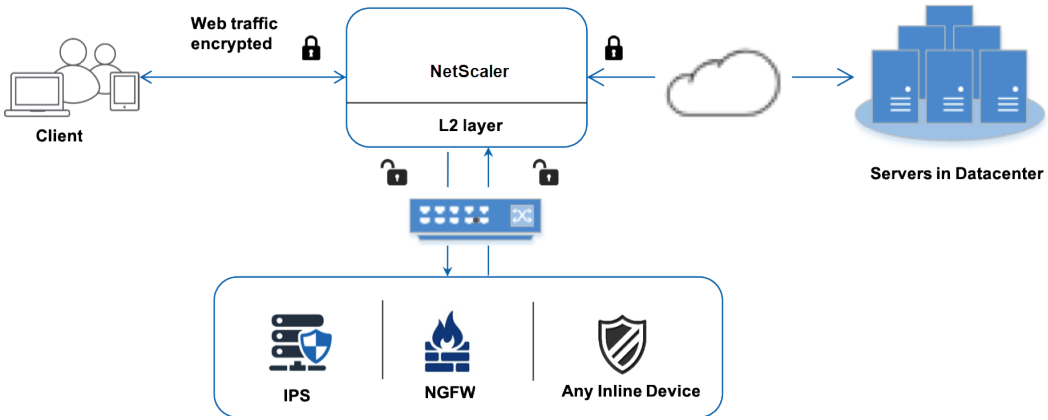
SSL 転送プロキシ・アプライアンスを 1 つ以上のインラインデバイスと統合することで、脅威を防ぎ、高度なセキュリティ保護を実現できます。インラインデバイスには、IPS や NGFW などの任意のセキュリティデバイスを使用できます。

SSL 転送プロキシアプライアンスとインラインデバイス統合を使用することでメリットが得られるユースケースには、次のようなものがあります。

- 暗号化されたトラフィックの検査**:** ほとんどの IPS および NGFW アプライアンスは暗号化されたトラフィックをバイパスするため、サーバーが攻撃に対して脆弱になる可能性があります。SSL 転送プロキシアプライアンスは、トラフィックを復号化してインラインデバイスに送信して検査できます。この統合により、お客様のネットワークセキュリティが強化されます。
- **TLS/SSL** 処理からのインラインデバイスのオフロード：TLS/SSL 処理はコストがかかり、IPS または NGFW アプライアンスがトラフィックを復号化すると CPU 使用率が高くなる可能性があります。SSL 転送プロキシアプライアンスは、TLS/SSL 処理をインラインデバイスからオフロードするのに役立ちます。その結果、インラインデバイスは大量のトラフィックを検査できます。
- インラインデバイスの負荷分散**:** 大量のトラフィックを管理するように複数のインラインデバイスを設定している場合、SSL Forward Proxy アプライアンスは負荷分散を行い、トラフィックをこれらのデバイスに均等に分散できます。
- トラフィックのスマート選択：アプライアンスは、検査のためにすべてのトラフィックをインラインデバイスに送信する代わりに、トラフィックのスマートな選択を行います。たとえば、インラインデバイスへの検査用のテキストファイルの送信はスキップされます。

インラインデバイスとの **SSL** フォワードプロキシ統合

次の図は、SSL 転送プロキシがインラインセキュリティデバイスとどのように統合されるかを示しています。



インラインデバイスを SSL 転送プロキシアプライアンスと統合すると、コンポーネントは次のように相互作用します。

1. クライアントは SSL 転送プロキシアプライアンスに要求を送信します。
2. アプライアンスは、ポリシー評価に基づいてコンテンツ検査のためにデータをインラインデバイスに送信します。HTTPS トラフィックの場合、アプライアンスはデータを復号化し、コンテンツ検査のためにプレーンテキストでインラインデバイスに送信します。

   > 注
   >
   > インラインデバイスが 2 台以上ある場合、アプライアンスはデバイスの負荷分散を行い、トラフィックを送信します。

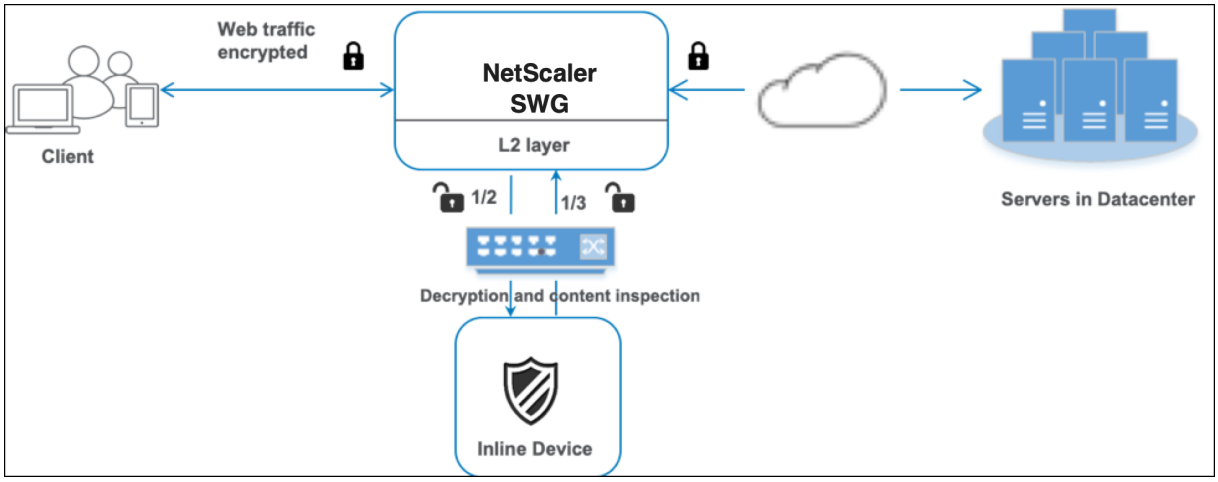
3. コンテンツスイッチングまたは HTTP/HTTPS 負荷分散仮想サーバーを追加します。
4. インラインデバイスは、データの脅威を検査し、データをドロップ、リセット、またはアプライアンスに戻すかどうかを決定します。
5. セキュリティ上の脅威がある場合、デバイスはデータを修正してアプライアンスに送信します。
6. HTTPS トラフィックの場合、アプライアンスはデータを再暗号化し、要求をバックエンドサーバーに転送します。
7. バックエンドサーバーは応答をアプライアンスに送信します。
8. アプライアンスは再びデータを復号化し、検査のためにインラインデバイスに送信します。
9. インラインデバイスはデータを検査します。セキュリティ上の脅威がある場合、デバイスはデータを修正してアプライアンスに送信します。
10. アプライアンスはデータを再暗号化し、応答をクライアントに送信します。

インラインデバイス統合の設定

SSL 転送プロキシアプライアンスをインラインデバイスで構成するには、次の 3 つの方法があります。

シナリオ **1**：単一のインラインデバイスを使用する

セキュリティデバイス（IPS または NGFW）をインラインモードで統合するには、SSL 転送プロキシアプライアンスでグローバルモードでコンテンツ検査と MAC ベース転送（MBF）を有効にする必要があります。次に、コンテンツインスペクションプロファイル、TCP サービス、インラインデバイスのコンテンツインスペクションアクションを追加して、インスペクションに基づいてトラフィックをリセット、ブロック、またはドロップします。また、アプライアンスがインラインデバイスに送信するトラフィックのサブセットを決定するために使用するコンテンツ検査ポリシーを追加します。最後に、サーバ上でレイヤ 2 接続を有効にしてプロキシ仮想サーバを設定し、コンテンツ検査ポリシーをこのプロキシ仮想サーバにバインドします。



次の手順を実行します:

1. MAC ベース転送 (MPF) モードを有効にします。
2. コンテンツ検査機能を有効にします。
3. サービスのコンテンツ検査プロファイルを追加します。コンテンツ検査プロファイルには、SSL 転送プロキシアプライアンスをインラインデバイスと統合するインラインデバイス設定が含まれています。
4. (任意) TCP モニタを追加します。

   > 注:
   >
   > トランスペアレントデバイスには IP アドレスがありません。したがって、ヘルスチェックを実行するには、モニターを明示的にバインドする必要があります。

5. サービスを追加します。サービスはインラインデバイスを表します。
6. (オプション) サービスを TCP モニターにバインドします。
7. サービスのコンテンツ検査アクションを追加します。
8. コンテンツ検査ポリシーを追加し、アクションを指定します。
9. HTTP または HTTPS プロキシ (コンテンツスイッチング) 仮想サーバーを追加します。
10. コンテンツ検査ポリシーを仮想サーバーにバインドします。

CLI を使用して設定する　コマンドプロンプトで次のコマンドを入力します。例はほとんどのコマンドの後に記載されています。

1. MBF を有効にします。

```
enable ns mode mbf
```

1. 本機能を有効にします。


```
enable ns feature contentInspection
```

1. コンテンツ検査プロファイルを追加します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <
positive_integer>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile ipsprof -type InlineInspection -ingressinterface
 “1/2” -egressInterface “1/3”
```

1. サービスを追加します。インラインデバイスを含め、どのデバイスも所有していないダミー IP アドレスを指定します。use source IP address (USIP) を「はい」に設定します。useproxyportを NO に設定します。デフォルトでは、ヘルスモニタリングはオンになっており、サービスをヘルスモニターにバインドし、モニターの TRANSPARENT オプションもオンに設定します。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName
<Name> -healthMonitor YES -usip YES -useproxyport NO
```

例:

```
add service ips_service 198.51.100.2 TCP * -healthMonitor YES -usip
YES -useproxyport NO -contentInspectionProfileName ipsprof
```

1. ヘルスモニターを追加します。デフォルトでは、ヘルスモニターはオンになっており、必要に応じて無効にすることもできます。コマンドプロンプトで入力します:

   ```
   add lb monitor <name> TCP -destIP <ip address> -destPort 80 -
   transparent <YES, NO>
   ```

例:

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
 YES
```

1. サービスをヘルスモニターにバインドする

ヘルスモニターを設定したら、サービスをヘルスモニターにバインドする必要があります。コマンドプロンプトで入力します:

```
bind service <name> -monitorName <name>
```

例:

```
bind service ips_svc -monitorName ips_tcp
```

1. コンテンツ検査アクションを追加します。


```
add contentInspection action <name> -type INLINEINSPECTION -serverName
 <string>
```

例:

```
add contentInspection action ips_action -type INLINEINSPECTION -
serverName ips_service
```

1. コンテンツ検査ポリシーを追加します。

```
add contentInspection policy <name> -rule <expression> -action <
string>
```

例:

```
add contentInspection policy ips_pol -rule "HTTP.REQ.METHOD.NE(\"
CONNECT\")"-action ips_action
```

1. プロキシ仮想サーバーを追加します。

```
add cs vserver <name> PROXY <IPAddress> <port> -cltTimeout <secs> -
Listenpolicy <expression> -authn401 ( ON | OFF )-authnVsName <string>
 -l2Conn ON
```

> 注:
>
> HTTP/SSL タイプの負荷分散仮想サーバーもサポートされています。

例:

```
add cs vserver transparentcs PROXY * * -cltTimeout 180 -Listenpolicy
exp1 -authn401 on -authnVsName swg-auth-vs-trans-http -l2Conn ON
```

1. ポリシーを仮想サーバーにバインドします。

```
bind cs vserver <name> -policyName <string> -priority <positive_integer
> -gotoPriorityExpression <expression> -type REQUEST
```

例:

```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpres
 END -type REQUEST
```

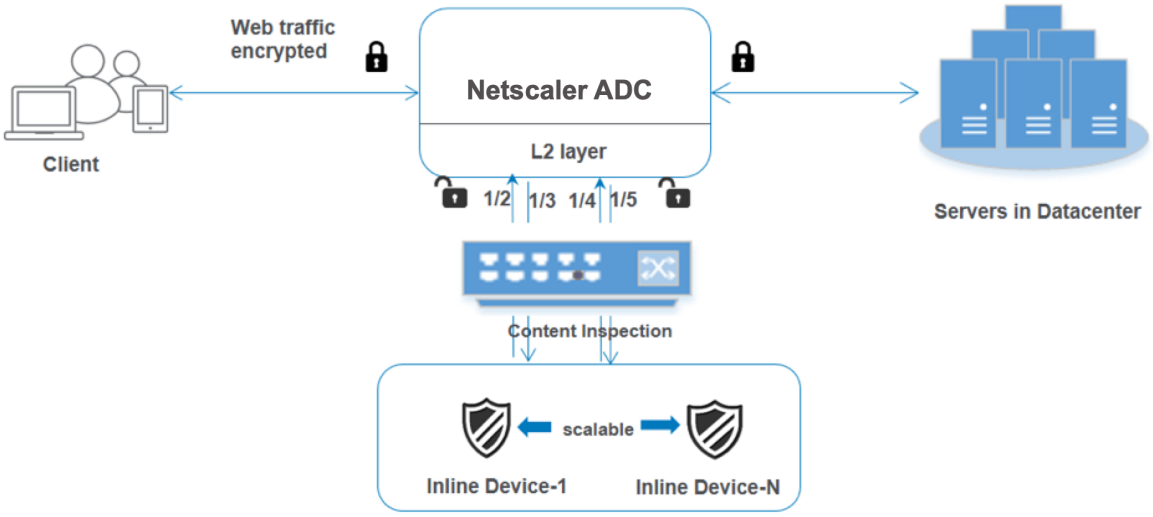
GUI を使用して設定する

1. [**System**] **>** [**Settings**] に移動します。「モードと機能」で、「モードの構成」をクリックします。
2. 「モードの設定」ページで、「**MAC** ベースの転送」オプションを選択します。
3. [**System**] **>** [**Settings**] に移動します。「モードと機能」で、「拡張機能の設定」をクリックします。

4. 「拡張機能の設定」ページで、「コンテンツ検査」オプションを選択します。
5. [セキュリティ**] > [**コンテンツ検査**] > [**コンテンツ検査プロファイル] に移動します。［追加］をクリックします。
6. [負荷分散**] > [**サービス**] > [**サービスの追加と追加] に移動します。
   a) 「詳細設定」で、「プロフィール」をクリックします。**CI** プロファイル名リストで、以前に作成したコンテンツ検査プロファイルを選択します。
   b) **[**サービス設定**]** で、**[**送信元 **IP** アドレスを使用] を [はい] に、[プロキシポートを使用する] を [いいえ] に設定します。「基本設定」で、「ヘルスモニタリング」を「いいえ」に設定します。
   c) ヘルスモニタリングは、このサービスを TCP モニターにバインドする場合にのみ有効にします。モニターをサービスにバインドする場合は、モニターの TRANSPARENT オプションを ON に設定します。
7. [セキュリティ**] > [**プロキシ仮想サーバー**] **[**追加**]** に移動します。名前、**IP** アドレス、およびポートを指定します。**[**** 詳細設定**]** で [ポリシー] を選択します。「+」記号をクリックします。
8. 「ポリシーの選択」で「コンテンツ検査」を選択します。［続行］をクリックします。
9. ［追加］をクリックします。名前を指定してください。[アクション] で [追加] をクリックします。
10. 名前を指定してください。「タイプ」で、「インライン検査」を選択します。「サーバー名」で、以前に作成した TCP サービスを選択します。
11. ［作成］をクリックします。ルールを指定して [作成] をクリックします。
12. ［**Bind**］をクリックします。
13. ［完了］をクリックします。

シナリオ **2:** 専用インターフェイスを備えた複数のインラインデバイスの負荷分散

2 つ以上のインラインデバイスを使用している場合は、専用のインターフェイスで異なるコンテンツインスペクションサービスを使用して、デバイスを負荷分散できます。この場合、SSL 転送プロキシアプライアンスは、専用インターフェイスを介して各デバイスに送信されるトラフィックのサブセットを負荷分散します。サブセットは、設定されたポリシーに基づいて決定されます。たとえば、TXT ファイルやイメージファイルは、検査のためにインラインデバイスに送信されない場合があります。



基本設定はシナリオ 1 と同じです。ただし、インラインデバイスごとにコンテンツ検査プロファイルを作成し、各プロファイルで入力インターフェイスと出力インターフェイスを指定する必要があります。各インラインデバイスにサービスを追加します。負荷分散仮想サーバーを追加し、コンテンツ検査アクションで指定します。次の追加手順を実行してください。

1. 各サービスのコンテンツ検査プロファイルを追加します。
2. デバイスごとにサービスを追加します。
3. 負荷分散仮想サーバーを追加します。
4. コンテンツ検査アクションで負荷分散仮想サーバーを指定します。

CLI を使用して設定する　コマンドプロンプトで次のコマンドを入力します。各コマンドの後に例が示されています。

1. MBF を有効にします。

```
enable ns mode mbf
```

1. 本機能を有効にします。

```
enable ns feature contentInspection
```

1. サービス 1 のプロファイル 1 を追加します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <
positive_integer>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile ipsprof1 -type InlineInspection -
ingressInterface "1/2"-egressInterface "1/3"
```


1. サービス 2 のプロファイル 2 を追加します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <
positive_integer>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile ipsprof2 -type InlineInspection -
ingressInterface "1/4"-egressInterface "1/5"
```

1. サービス 1 を追加します。インラインデバイスを含め、どのデバイスも所有していないダミー IP アドレスを指定します。use source IP address (USIP) を「はい」に設定します。useproxyportを NO に設定します。Transparent オプションを ON に設定して TCP モニターでヘルスモニタリングを有効にします。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName
<Name> -healthMonitor NO -usip YES -useproxyport NO
```

例:

```
add service ips_service1 192.168.10.2 TCP * -healthMonitor NO -usip
YES -useproxyport NO -contentInspectionProfileName ipsprof1
```

1. サービス 2 を追加します。インラインデバイスを含め、どのデバイスも所有していないダミー IP アドレスを指定します。use source IP address (USIP) を「はい」に設定します。useproxyportを NO に設定します。トランスペアレントオプションをオンに設定してヘルスモニタリングをオンにします。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName
<Name> -healthMonitor NO -usip YES -useproxyport NO
```

例:

```
add service ips_service2 192.168.10.3 TCP * -healthMonitor NO -usip
YES -useproxyport NO -contentInspectionProfileName ipsprof2
```

1. 負荷分散仮想サーバーを追加します。

```
add lb vserver <LB_VSERVER_NAME> TCP <IP> <port>
```

例:

```
add lb vserver lb_inline_vserver TCP 192.0.2.100 *
```

1. サービスを負荷分散仮想サーバーにバインドします。

```
bind lb vserver <LB_VSERVER_NAME> <service_name>
bind lb vserver <LB_VSERVER_NAME> <service_name>
```

例:


```
bind lb vserver lb_inline_vserver ips_service1
bind lb vserver lb_inline_vserver ips_service2
```

1. コンテンツ検査アクションで負荷分散仮想サーバーを指定します。

```
add contentInspection action <name> -type INLINEINSPECTION -serverName
 <string>
```

例:

```
add contentInspection action ips_action -type INLINEINSPECTION -
serverName lb_inline_vserver
```

1. コンテンツ検査ポリシーを追加します。ポリシーでコンテンツ検査アクションを指定します。

```
add contentInspection policy <name> -rule <expression> -action <
string>
```

例:

```
add contentInspection policy ips_pol -rule "HTTP.REQ.METHOD.NE(\"
CONNECT\")"-action ips_action
```

1. プロキシ仮想サーバーを追加します。

```
add cs vserver <name> PROXY <IPAddress> <port> -l2Conn ON
```

例:

```
add cs vserver transparentcs PROXY * * -l2Conn ON
```

1. コンテンツ検査ポリシーを仮想サーバーにバインドします。

```
bind cs vserver <name> -policyName <string> -priority <positive_integer
> -gotoPriorityExpression <expression> -type REQUEST
```

例:

```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpres
 END -type REQUEST
```

GUI を使用した設定

1. [**System**] **>** [**Settings**] に移動します。「モードと機能」で、「モードの構成」をクリックします。
2. 「モードの設定」ページで、「**MAC** ベースの転送」オプションを選択します。
3. [**System**] **>** [**Settings**] に移動します。「モードと機能」で、「拡張機能の設定」をクリックします。
4. 「拡張機能の設定」ページで、「コンテンツ検査」オプションを選択します。

5. [セキュリティ**] > [**コンテンツ検査**] > [**コンテンツ検査プロファイル] に移動します。［追加］をクリックします。
6. 入力インターフェイスと出力インターフェイスを指定します。
7. 2 つのプロファイルを作成します。2 番目のプロファイルには、異なる入力インターフェイスと出力インターフェイスを指定します。
8. [負荷分散**] > [**サービス**] > [**サービスの追加と追加] に移動します。

   a)「詳細設定」で、「プロフィール」をクリックします。**CI** プロファイル名リストで、以前に作成したコンテンツ検査プロファイルを選択します。

   b) **[**サービス設定**]** で、**[**送信元 **IP** アドレスを使用] を [はい] に、[プロキシポートを使用する] を [いいえ] に設定します。「基本設定」で、「ヘルスモニタリング」を「いいえ」に設定します。

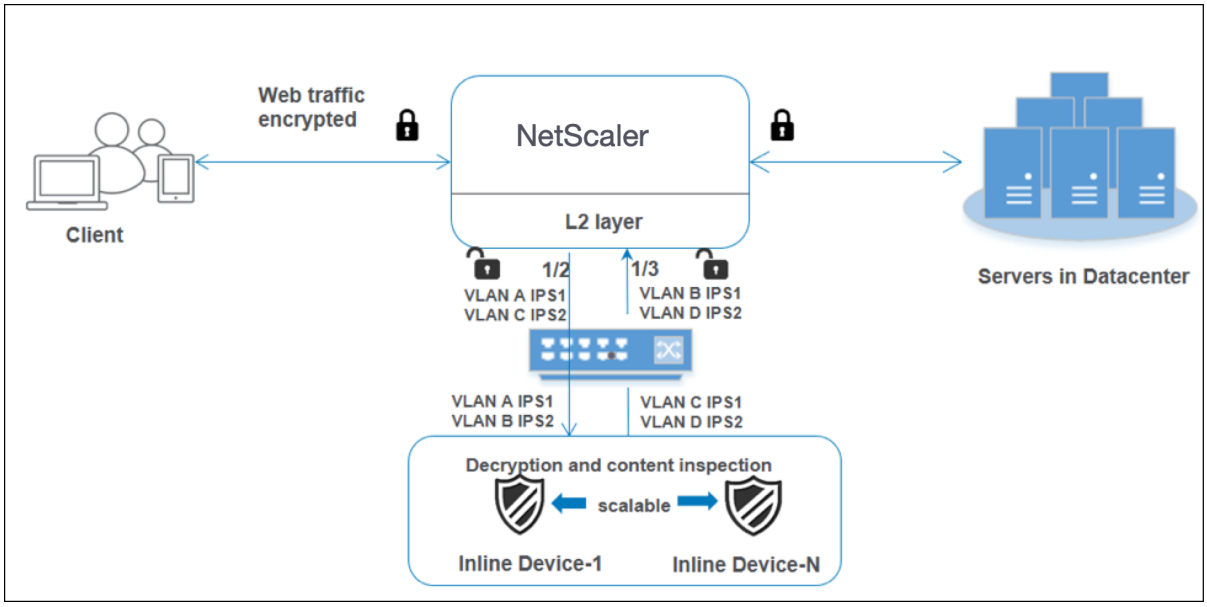
   c) ヘルスモニタリングは、このサービスを TCP モニターにバインドする場合にのみ有効にします。モニターをサービスにバインドする場合は、モニターの TRANSPARENT オプションを ON に設定します。

   2 つのサービスを作成します。インラインデバイスを含め、どのデバイスにも所有されていないダミー IP アドレスを指定します。
9. [負荷分散**] > [**仮想サーバー**] > [**追加**]** に移動します。TCP 負荷分散仮想サーバーを作成し、**OK** をクリックします。
10. 負荷分散仮想サーバーサービスバインディングセクションをクリックします。「サービス・バインディング」で、「サービスを選択」の矢印をクリックします。前に作成した 2 つのサービスを選択し、[選択] をクリックします。［**Bind**］をクリックします。
11. [セキュリティ**] > [**プロキシ仮想サーバー**] **[**追加**]** に移動します。名前、**IP** アドレス、およびポートを指定します。**[**** 詳細設定**]** で [ポリシー] を選択します。「+」記号をクリックします。
12.「ポリシーの選択」で「コンテンツ検査」を選択します。［続行］をクリックします。
13.［追加］をクリックします。名前を指定してください。[アクション] で [追加] をクリックします。
14. 名前を指定してください。「タイプ」で、「インライン検査」を選択します。[サーバー名] で、先ほど作成した負荷分散仮想サーバーを選択します。
15.［作成］をクリックします。ルールを指定して [作成] をクリックします。
16.［**Bind**］をクリックします。
17.［完了］をクリックします。

シナリオ **3:** 共有インターフェイスを持つ複数のインラインデバイスの負荷分散

2 つ以上のインラインデバイスを使用している場合は、共有インターフェイスで異なるコンテンツインスペクションサービスを使用して、デバイスを負荷分散できます。この場合、SSL 転送プロキシアプライアンスは、共有インター

フェイスを介して各デバイスに送信されるトラフィックのサブセットを負荷分散します。サブセットは、設定されたポリシーに基づいて決定されます。たとえば、TXT ファイルやイメージファイルは、検査のためにインラインデバイスに送信されない場合があります。



基本構成はシナリオ 2 と同じです。このシナリオでは、インターフェイスを異なる VLAN にバインドして、各インラインデバイスのトラフィックを分離します。コンテンツ検査プロファイルで VLAN を指定します。次の追加手順を実行してください。

1. 共有インターフェイスを異なる VLAN にバインドします。
2. コンテンツ検査プロファイルで入力 VLAN と出力 VLAN を指定します。

CLI を使用した設定　コマンドプロンプトで次のコマンドを入力します。各コマンドの後に例が示されています。

1. MBF を有効にします。

```
enable ns mode mbf
```

1. 本機能を有効にします。

```
enable ns feature contentInspection
```

1. 共有インターフェイスを異なる VLAN にバインドします。

```
bind vlan <id> -ifnum <interface> -tagged
```

例:

```
bind vlan 100  - ifnum 1/2 tagged
bind vlan 200  - ifnum 1/3 tagged
bind vlan 300  - ifnum 1/2 tagged
```



```
4 bind vlan 400  -ifnum 1/3 tagged
5 <!--NeedCopy-->
```

1. サービス 1 のプロファイル 1 を追加します。プロファイルの入力 VLAN と出力 VLAN を指定します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <
positive_integer>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile ipsprof1 -type InlineInspection -
egressInterface “1/3” -ingressinterface “1/2” -egressVlan 100 -
ingressVlan 300
```

1. サービス 2 のプロファイル 2 を追加します。プロファイルの入力 VLAN と出力 VLAN を指定します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <
positive_integer>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile ipsprof2 -type InlineInspection -
egressInterface “1/3” -ingressinterface “1/2” -egressVlan 200 -
ingressVlan 400
```

1. サービス 1 を追加します。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName
<Name> -healthMonitor NO -usip YES -useproxyport NO
```

例:

```
add service ips_service1 192.168.10.2 TCP * -healthMonitor NO -usip
YES -useproxyport NO -contentInspectionProfileName ipsprof1
```

1. サービス 2 を追加します。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName
<Name> -healthMonitor NO -usip YES -useproxyport NO
```

例:

```
add service ips_service2 192.168.10.3 TCP * -healthMonitor NO -usip
YES -useproxyport NO -contentInspectionProfileName ipsprof2
```

1. 負荷分散仮想サーバーを追加します。


```
add lb vserver <LB_VSERVER_NAME> TCP <IP> <port>
```

例:

```
add lb vserver lb_inline_vserver TCP 192.0.2.100 *
```

1. サービスを負荷分散仮想サーバーにバインドします。

```
bind lb vserver <LB_VSERVER_NAME> <service_name>
bind lb vserver <LB_VSERVER_NAME> <service_name>
```

例:

```
bind lb vserver lb_inline_vserver ips_service1
bind lb vserver lb_inline_vserver ips_service2
```

1. コンテンツ検査アクションで負荷分散仮想サーバーを指定します。

```
add contentInspection action <name> -type INLINEINSPECTION -serverName
 <string>
```

例:

```
add contentInspection action ips_action -type INLINEINSPECTION -
serverName lb_inline_vserver
```

1. コンテンツ検査ポリシーを追加します。ポリシーでコンテンツ検査アクションを指定します。

```
add contentInspection policy <name> -rule <expression> -action <
string>
```

例:

```
add contentInspection policy ips_pol -rule "HTTP.REQ.METHOD.NE(\"
CONNECT\")"-action ips_action
```

1. プロキシ仮想サーバーを追加します。

```
add cs vserver <name> PROXY <IPAddress> <port> -l2Conn ON
```

例:

```
add cs vserver transparentcs PROXY * * -l2Conn ON
```

1. コンテンツ検査ポリシーを仮想サーバーにバインドします。

```
bind cs vserver <name> -policyName <string> -priority <positive_integer
> -gotoPriorityExpression <expression> -type REQUEST
```

例:

```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpres
 END -type REQUEST
```


GUI を使用した設定

1. [**System**] **>** [**Settings**] に移動します。「モードと機能」で、「モードの構成」をクリックします。
2. 「モードの設定」ページで、「**MAC** ベースの転送」オプションを選択します。
3. [**System**] **>** [**Settings**] に移動します。「モードと機能」で、「拡張機能の設定」をクリックします。
4. 「拡張機能の設定」ページで、「コンテンツ検査」オプションを選択します。
5. [システム**] > [**ネットワーク**] > [VLAN] > [**追加**]** に移動します。4 つの VLAN を追加し、インターフェイスにタグを付けます。
6. [セキュリティ**] > [**コンテンツ検査**] > [**コンテンツ検査プロファイル] に移動します。[追加] をクリックします。
7. 入力側と出力側の VLAN を指定します。
8. 別のプロファイルを作成します。2 番目のプロファイルには、異なる入力 VLAN と出力 VLAN を指定します。
9. [負荷分散**] > [**サービス**] > [**サービスの追加と追加] に移動します。

   a) 「詳細設定」で、「プロフィール」をクリックします。**CI** プロファイル名リストで、以前に作成したコンテンツ検査プロファイルを選択します。

   b) **[**サービス設定**]** で、**[**送信元 **IP** アドレスを使用] を [はい] に、[プロキシポートを使用する] を [いいえ] に設定します。「基本設定」で、「ヘルスモニタリング」を「いいえ」に設定します。

   c) ヘルスモニタリングは、このサービスを TCP モニターにバインドする場合にのみ有効にします。モニターをサービスにバインドする場合は、モニターの TRANSPARENT オプションを ON に設定します。

   2 つのサービスを作成します。インラインデバイスを含め、どのデバイスにも所有されていないダミー IP アドレスを指定します。サービス 1 でプロファイル 1 を指定し、サービス 2 でプロファイル 2 を指定します。
10. [負荷分散**] > [**仮想サーバー**] > [**追加**]** に移動します。TCP 負荷分散仮想サーバーを作成し、**OK** をクリックします。
11. 負荷分散仮想サーバーサービスバインディングセクションをクリックします。「サービス・バインディング」で、「サービスを選択」の矢印をクリックします。前に作成した 2 つのサービスを選択し、[選択] をクリックします。[**Bind**] をクリックします。
12. [セキュリティ**] > [**プロキシ仮想サーバー**] **[**追加**]** に移動します。名前、**IP** アドレス、およびポートを指定します。**[**** 詳細設定**]** で [ポリシー] を選択します。「+」記号をクリックします。
13. 「ポリシーの選択」で「コンテンツ検査」を選択します。[続行] をクリックします。
14. [追加] をクリックします。名前を指定してください。[アクション] で [追加] をクリックします。
15. 名前を指定してください。「タイプ」で、「インライン検査」を選択します。[サーバー名] で、先ほど作成した負荷分散仮想サーバーを選択します。
16. [作成] をクリックします。ルールを指定して [作成] をクリックします。

17. [**Bind**] をクリックします。
18. [完了] をクリックします。

NetScaler とパッシブセキュリティデバイスの統合（侵入検知システム）

December 8, 2023

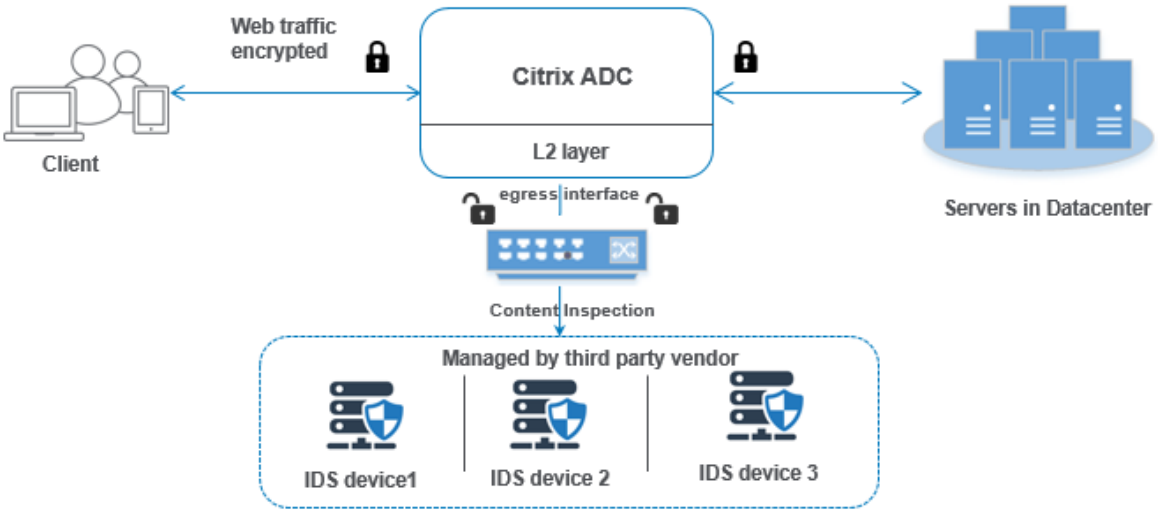
NetScaler アプライアンスは、侵入検知システム（IDS）などのパッシブセキュリティデバイスと統合されました。これらのパッシブデバイスはログを保存し、不良または非準拠のトラフィックを検出するとアラートをトリガーします。また、コンプライアンスのためのレポートも生成します。NetScaler アプライアンスが 2 つ以上の IDS デバイスに統合されており、トラフィックが多い場合、アプライアンスは仮想サーバーレベルでトラフィックのクローンを作成することでデバイスの負荷を分散できます。

高度なセキュリティ保護のために、NetScaler アプライアンスは、検出専用モードで展開された IDS などのパッシブセキュリティデバイスと統合されています。これらのデバイスはログを保存し、不良または非準拠のトラフィックを検出するとアラートをトリガーします。また、コンプライアンスのためのレポートも生成します。NetScaler を IDS デバイスと統合する利点の一部を次に示します。

- 暗号化されたトラフィックを検査する。ほとんどのセキュリティデバイスは暗号化されたトラフィックをバイパスするため、サーバは攻撃に対して脆弱になります。NetScaler アプライアンスは、トラフィックを復号化して IDS デバイスに送信し、顧客のネットワークセキュリティを強化できます。
- インラインデバイスを **TLS/SSL** 処理からオフロードする。TLS/SSL の処理にはコストがかかり、侵入検知デバイスがトラフィックを復号化すると、システム CPU が高くなります。暗号化されたトラフィックが急速に増加するにつれて、これらのシステムは暗号化されたトラフィックの復号化と検査に失敗します。NetScaler は、TLS/SSL 処理から IDS デバイスへのトラフィックをオフロードするのに役立ちます。この方法でデータをオフロードすると、IDS デバイスは大量のトラフィックインスペクションをサポートすることになります。
- **IDS** デバイスの負荷分散 NetScaler アプライアンスは、トラフィックが多い場合、仮想サーバーレベルでトラフィックのクローンを作成することにより、複数の IDS デバイスの負荷を分散します。
- トラフィックをパッシブデバイスに複製する。アプライアンスに流入するトラフィックは、コンプライアンスレポートを生成するために、他のパッシブデバイスに複製できます。たとえば、一部のパッシブデバイスにすべてのトランザクションを記録するよう義務付けている政府機関はほとんどありません。
- トラフィックを複数のパッシブデバイスにファンニングする。一部のお客様は、着信トラフィックを複数のパッシブデバイスにファンアウトまたは複製することを好みます。
- トラフィックのスマートな選択。アプライアンスに流入するすべてのパケットは、テキストファイルのダウンロードなど、内容を検査する必要がない場合があります。ユーザーは、NetScaler アプライアンスを構成して、検査する特定のトラフィック（.exe ファイルなど）を選択し、そのトラフィックを IDS デバイスに送信してデータを処理することができます。

NetScaler が **L2** 接続を備えた **IDS** デバイスとどのように統合されるか

次の図は、IDS が NetScaler アプライアンスとどのように統合されるかを示しています。



コンポーネントの相互作用は次のように与えられます。

1. クライアントは、HTTP/HTTPS リクエストを NetScaler アプライアンスに送信します。
2. アプライアンスはトラフィックをインターセプトし、コンテンツインスペクションポリシーの評価に基づいて IDS デバイスに複製します。
3. トラフィックが暗号化されたものである場合、アプライアンスはデータを復号化し、プレーンテキストとして送信します。
4. ポリシー評価に基づいて、アプライアンスは「MIRROR」タイプのコンテンツ検査アクションを適用します。
5. アクションには、IDS サービスまたは負荷分散サービス (複数の IDS デバイス統合用) が設定されています。
6. IDS デバイスは、アプライアンス上でコンテンツ検査サービスタイプ「Any」として設定されています。コンテンツ検査サービスは、IDS デバイスにデータを転送する必要がある出力インターフェイスを指定する「MIRROR」タイプのコンテンツ検査プロファイルに関連付けられます。オプションで、コンテンツ検査プロファイルに VLAN タグを設定することもできます。

   > 注:
   >
   > - IDS サービスまたはサーバに使用される IP アドレスはダミーアドレスです。
   > - NetScaler アプライアンスは、出力インターフェイスで LA チャネルをサポートしていません。

7. その後、アプライアンスは出力インターフェイスを介して 1 つ以上の IDS デバイスにデータを複製します。
8. 同様に、バックエンドサーバーが NetScaler に応答を送信すると、アプライアンスはデータを複製して IDS デバイスに転送します。

9. アプライアンスが 1 つ以上の IDS デバイスに統合されていて、デバイスの負荷分散を希望する場合は、負荷分散仮想サーバを使用できます。

ソフトウェアライセンス

インラインデバイス統合を展開するには、NetScaler アプライアンスに次のいずれかのライセンスをプロビジョニングする必要があります。

1. ADC Premium
2. ADC Advanced
3. Telco Advanced
4. Telco プレミアム

侵入検知システム統合の設定

IDS デバイスを NetScaler と統合するには、2 つの方法があります。

シナリオ **1:** 単一の **IDS** デバイスとの統合

コマンドラインインターフェイスを使用して設定する必要がある手順を次に示します。

1. コンテンツ検査を有効にする
2. IDS デバイスを表すサービス用に、MIRROR タイプのコンテンツ検査プロファイルを追加します。
3. タイプ「ANY」の IDS サービスを追加する
4. タイプ「MIRROR」のコンテンツ検査アクションを追加する
5. IDS 検査のコンテンツ検査ポリシーを追加する
6. コンテンツ検査ポリシーを HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サービスにバインドする

コンテンツ検査を有効にする　NetScaler アプライアンスから IDS デバイスに検査用のコンテンツを送信する場合は、復号化の実行とは関係なく、コンテンツ検査と負荷分散機能を有効にする必要があります。

コマンドプロンプトで入力します:

```
enable ns feature contentInspection LoadBalancing
```

タイプ「**MIRROR**」のコンテンツ検査プロファイルの追加　「MIRROR」タイプのコンテンツ検査プロファイルは、IDS デバイスへの接続方法を説明しています。
コマンドプロンプトで、「」と入力します。


```
add contentInspection profile <name> -type MIRROR -egressInterface <
interface_name> [-egressVlan <positive_integer>]
```

例:

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
 1/1 -egressVLAN 10
```

IDS サービスの追加　アプライアンスと統合されている IDS デバイスごとに、「ANY」タイプのサービスを設定する必要があります。このサービスには IDS デバイス設定の詳細が含まれています。このサービスは IDS デバイスを表します。

コマンドプロンプトで入力します:

```
add service <Service_name> <IP> ANY <Port> - contentinspectionProfileName
 <Name> -healthMonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service IDS_service 1.1.1.1 ANY 8080 -contentInspectionProfileName
 IDS_profile1 -healthMonitor OFF
```

IDS サービスの **MIRROR** タイプのコンテンツ検査アクションを追加する　コンテンツ検査機能を有効にして IDS プロファイルとサービスを追加したら、要求を処理するための Content Inspection アクションを追加する必要があります。コンテンツ検査アクションに基づいて、アプライアンスは IDS デバイスにデータをドロップ、リセット、ブロック、または送信できます。

コマンドプロンプトで入力します:

```
add ContentInspection action < action_name > -type MIRROR -serverName
 Service_name/Vserver_name>
```

例:

```
add ContentInspection action IDS_action -type MIRROR -serverName
IDS_service
```

IDS 検査のコンテンツ検査ポリシーを追加する　コンテンツ検査アクションを作成したら、コンテンツ検査ポリシーを追加して検査の要求を評価する必要があります。このポリシーは、1 つ以上の式で構成されるルールに基づいています。ポリシーは、ルールに基づいてインスペクション対象のトラフィックを評価し、選択します。

コマンドプロンプトで、次のように入力します:

```
add contentInspection policy < policy_name > -rule <Rule> -action <
action_name>
```

例:


```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプのコンテンツスイッチングまたは負荷分散仮想サービスにバインドする　Web トラフィックを受信するには、負荷分散仮想サーバーを追加する必要があります。
コマンドプロンプトで入力します:

```
add lb vserver <name> <vserver name>
```

例:

```
add lb vserver HTTP_vserver HTTP 1.1.1.3 8080
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプのコンテンツスイッチング仮想サーバーまたは負荷分散仮想サーバーにバインドする　HTTP/SSL タイプの負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーをコンテンツ検査ポリシーにバインドする必要があります。

コマンドプロンプトで、次のように入力します:

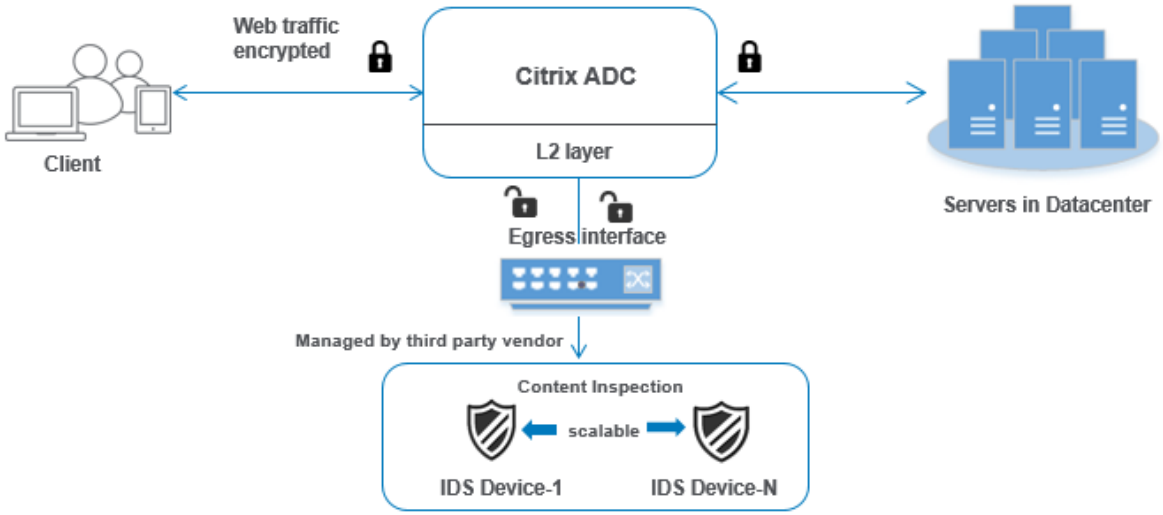
```
bind lb vserver <vserver name> -policyName < policy_name > -priority
< priority > -type <REQUEST>
```

例:

```
bind lb vserver HTTP_vserver -policyName IDS_pol1 -priority 100 -type
 REQUEST
```

シナリオ **2:** 複数の **IDS** デバイスの負荷分散

2 つ以上の IDS デバイスを使用している場合は、異なるコンテンツ検査サービスを使用してデバイスの負荷分散を行う必要があります。この場合、NetScaler アプライアンスは、トラフィックのサブセットを各デバイスに送信することに加えて、デバイスの負荷分散を行います。
基本的な設定手順については、シナリオ 1 を参照してください。



コマンドラインインターフェイスを使用して設定する必要がある手順を次に示します。

1. IDS サービス 1 の MIRROR タイプのコンテンツ検査プロファイル 1 を追加します。
2. IDS サービス 2 の MIRROR タイプのコンテンツ検査プロファイル 2 を追加する
3. IDS デバイス 1 にタイプ ANY の IDS サービス 1 を追加する
4. IDS デバイス 2 にタイプ ANY の IDS サービス 2 を追加する
5. ANY タイプの負荷分散仮想サーバを追加する
6. IDS サービス 1 を負荷分散仮想サーバーにバインドする
7. IDS サービス 2 を負荷分散仮想サーバーにバインドする
8. IDS デバイスの負荷分散のためのコンテンツ検査アクションを追加します。
9. 検査用のコンテンツ検査ポリシーを追加する
10. HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーを追加する
11. コンテンツ検査ポリシーを HTTP/SSL タイプの負荷分散仮想サーバーにバインドする

IDS サービス **1** の **MIRROR** タイプのコンテンツ検査プロファイル **1** を追加します　IDS 設定は、コンテンツ検査プロファイルと呼ばれるエンティティで指定できます。プロファイルにはデバイス設定のコレクションがあります。コンテンツ検査プロファイル 1 が IDS サービス 1 用に作成されます。

コマンドプロンプトで入力します:

```
add contentInspection profile <name> -type ANY -egressInterface <
interface_name> [-egressVlan <positive_integer>]
```

例:

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
 1/1 -egressVLAN 1
```


IDS サービス **2** のタイプ **MIRROR** のコンテンツ検査プロファイル **2** を追加する　サービス 2 にはコンテンツ検査プロファイル 2 が追加され、インラインデバイスは出力 1/1 インターフェイスを介してアプライアンスと通信します。

コマンドプロンプトで入力します:

```
add contentInspection profile <name> -type MIRROR -egressInterface -
egressVlan <positive_integer>]
```

例:

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
 1/1 -egressVLAN 1
```

IDS デバイス **1** にタイプ **ANY** の **IDS** サービス **1** を追加する　コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 1 のインラインサービス 1 を負荷分散設定の一部として追加する必要があります。追加したサービスによって、インライン構成の詳細がすべて提供されます。

コマンドプロンプトで入力します:

```
add service <Service_name_1> <Pvt_IP1> ANY <Port> -contentInspectionProfileNam
 <IDS_Profile_1> -usip ON -useproxyport OFF
```

例:

```
add service IDS_service1 1.1.1.1 ANY 80 -contentInspectionProfileName
 IDS_profile1 -usip ON -useproxyport OFF
```

> 注
>
> この例で示されている IP アドレスはダミーの IP アドレスです。

IDS デバイス **2** にタイプ **ANY** の **IDS** サービス **2** を追加する　コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 2 にインラインサービス 2 を追加する必要があります。追加したサービスによって、インライン構成の詳細がすべて提供されます。

コマンドプロンプトで入力します:

```
add service <Service_name_1> <Pvt_IP1> ANY -contentInspectionProfileName
 <Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service IDS_service 1 1.1.2 ANY 80 -contentInspectionProfileName
IDS_profile2
```


> 注
>
> この例で示されている IP アドレスはダミーの IP アドレスです。

負荷分散仮想サーバの追加　インラインプロファイルとサービスを追加したら、サービスの負荷分散用の負荷分散仮想サーバを追加する必要があります。

コマンドプロンプトで入力します:

```
add lb vserver <vserver_name> ANY <Pvt_IP3> <port>
```

例:

```
add lb vserver lb-IDS_vserver ANY 1.1.1.2
```

IDS サービス **1** を負荷分散仮想サーバーにバインドする　負荷分散仮想サーバーを追加したら、負荷分散仮想サーバーを最初のサービスにバインドします。

コマンドプロンプトで入力します:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-IDS_vserver IDS_service1
```

IDS サービス **2** を負荷分散仮想サーバーにバインドする　負荷分散仮想サーバーを追加したら、そのサーバーを 2 番目のサービスにバインドします。

コマンドプロンプトで入力します:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-IDS_vserver IDS_service2
```

IDS サービスのコンテンツ検査アクションを追加する　コンテンツ検査機能を有効にしたら、インラインリクエスト情報を処理するための「コンテンツ検査」アクションを追加する必要があります。選択したアクションに基づいて、アプライアンスは IDS デバイスへのトラフィックをドロップ、リセット、ブロック、または送信します。

コマンドプロンプトで入力します:

```
add contentInspection action <name> -type <type> (-serverName <string
> [-ifserverdown <ifserverdown>]
```

例:

```
add ContentInspection action IDS_action -type MIRROR -serverName lb-
IDS_vserver
```


検査用のコンテンツ検査ポリシーを追加する　コンテンツ検査アクションを作成したら、サービス要求を評価するコンテンツ検査ポリシーを追加する必要があります。

コマンドプロンプトで、次のように入力します:

```
add contentInspection policy <policy_name> -rule <Rule> -action <
action_name>
```

例:

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーを追加する　Web トラフィックを受け入れるために、コンテンツスイッチングまたは負荷分散仮想サーバーを追加します。また、仮想サーバ上で layer2 接続を有効にする必要があります。

負荷分散の詳細については、「負荷分散の仕組み」トピックを参照してください。

コマンドプロンプトで入力します:

```
add lb vserver <name> <vserver name>
```

例:

```
add lb vserver http_vserver HTTP 1.1.1.1 8080
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプの負荷分散仮想サーバーにバインドする　HTTP/SSL タイプのコンテンツスイッチング仮想サーバーまたは負荷分散仮想サーバーをコンテンツ検査ポリシーにバインドする必要があります。

コマンドプロンプトで、次のように入力します:

```
bind lb vserver <vserver name> -policyName < policy_name > -priority
<> -type <REQUEST>
```

例:

```
bind lb vserver http_vserver -policyName IDS_pol1 -priority 100 -type
 REQUEST
```

NetScaler GUI を使用してインラインサービス統合を構成する

1. [セキュリティ] > [コンテンツ検査] > [コンテンツ検査プロファイル] に移動します。
2. [コンテンツ検査プロファイル] ページで、[追加] をクリックします。
3. **[**コンテンツ検査プロファイルの作成] ページで、次のパラメータを設定します。

    a) プロファイル名。IDS のコンテンツ検査プロファイルの名前。

b) タイプ。プロファイルタイプを MIRROR として選択します。
c) 出力インターフェイス。NetScaler から IDS デバイスにトラフィックが送信されるインターフェイス。
d) 出力 VLAN（任意）トラフィックが IDS デバイスに送信される際のインターフェイス VLAN ID。

4. ［作成］をクリックします。
5. トラフィック管理 > 負荷分散 > サービスに移動し、追加をクリックします。
6. [負荷分散サービス] ページで、コンテンツ検査サービスの詳細を入力します。
7. **[**詳細設定**]** セクションで、[プロファイル] をクリックします。
8. [プロファイル] セクションに移動し、[鉛筆] アイコンをクリックしてコンテンツ検査プロファイルを追加します。
9. ［**OK**］をクリックします。
10. [負荷分散] > [サーバー] に移動します。HTTP または SSL タイプの仮想サーバを追加します。
11. サーバーの詳細を入力したら、「**OK**」をクリックし、もう一度「**OK**」をクリックします。
12. **[**詳細設定**]** セクションで、[ポリシー] をクリックします。
13. [ポリシー] セクションに移動し、[鉛筆] アイコンをクリックしてコンテンツ検査ポリシーを設定します。
14. 「ポリシーの選択」ページで、「コンテンツ検査」を選択します。［続行］をクリックします。
15. [ポリシーバインド] セクションで、[+] をクリックしてコンテンツ検査ポリシーを追加します。
16. [**CI** ポリシーの作成] ページで、インラインコンテンツ検査ポリシーの名前を入力します。
17. [**Action**] フィールドで [+] 記号をクリックし、MIRROR タイプの IDS コンテンツ検査アクションを作成します。
18. **[CI** アクションの作成] ページで、次のパラメータを設定します。

    a) Name：コンテンツ検査インラインポリシーの名前。
    b) タイプ。タイプとして MIRROR を選択します。
    c) サーバー名：サーバ/サービス名を [インラインデバイス] として選択します。
    d) サーバーがダウンした場合。サーバがダウンした場合のオペレーションを選択します。
    e) リクエストのタイムアウト。タイムアウト値を選択します。デフォルト値を使用できます。
    f) タイムアウトアクションの要求。タイムアウトアクションを選択します。デフォルト値を使用できます。

19. ［作成］をクリックします。
20. **[CI** ポリシーの作成] ページで、その他の詳細を入力します。
21. 「**OK**」をクリックして「閉じる」をクリックします。

負荷分散と IDS デバイスへのトラフィックの複製のための NetScaler GUI 構成については、「負荷分散」を参照してください。

コンテンツ変換後にトラフィックをバックエンドオリジンサーバーに負荷分散および転送するための NetScaler GUI 構成の詳細については、「負荷分散」を参照してください。

NetScaler レイヤー 3 とパッシブセキュリティデバイス（侵入検知システム）の統合

December 8, 2023

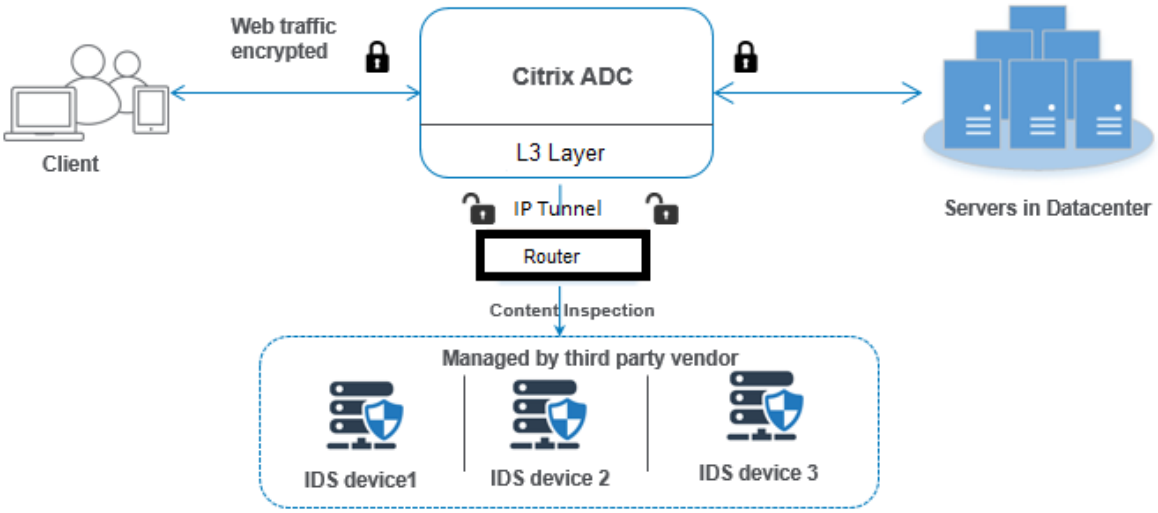
NetScaler アプライアンスは、侵入検知システム（IDS）などのパッシブセキュリティデバイスと統合されました。この設定では、アプライアンスは元のトラフィックのコピーをリモート IDS デバイスに安全に送信します。これらのパッシブデバイスはログを保存し、不良または非準拠のトラフィックを検出するとアラートをトリガーします。また、コンプライアンスのためのレポートも生成します。NetScaler アプライアンスが 2 つ以上の IDS デバイスと統合されていて、トラフィック量が多い場合、アプライアンスは仮想サーバーレベルでトラフィックをクローニングすることでデバイスの負荷分散を行うことができます。

高度なセキュリティ保護のために、NetScaler アプライアンスは、検出専用モードで展開された IDS などのパッシブセキュリティデバイスと統合されています。これらのデバイスはログを保存し、不良または非準拠のトラフィックを検出するとアラートをトリガーします。また、コンプライアンスのためのレポートも生成します。NetScaler を IDS デバイスと統合する利点の一部を次に示します。

- 暗号化されたトラフィックを検査する。ほとんどのセキュリティデバイスは暗号化されたトラフィックをバイパスするため、サーバは攻撃に対して脆弱になります。NetScaler アプライアンスは、トラフィックを復号化して IDS デバイスに送信し、顧客のネットワークセキュリティを強化できます。
- インラインデバイスを **TLS/SSL** 処理からオフロードする。TLS/SSL の処理にはコストがかかり、侵入検知デバイスがトラフィックを復号化すると、システム CPU が高くなります。暗号化されたトラフィックが急速に増加するにつれて、これらのシステムは暗号化されたトラフィックの復号化と検査に失敗します。NetScaler は、TLS/SSL 処理から IDS デバイスへのトラフィックをオフロードするのに役立ちます。この方法でデータをオフロードすると、IDS デバイスは大量のトラフィックインスペクションをサポートすることになります。
- **IDS** デバイスの負荷分散 NetScaler アプライアンスは、トラフィックが多い場合、仮想サーバーレベルでトラフィックのクローンを作成することにより、複数の IDS デバイスの負荷を分散します。
- トラフィックをパッシブデバイスに複製する。アプライアンスに流入するトラフィックは、コンプライアンスレポートを生成するために、他のパッシブデバイスに複製できます。たとえば、一部のパッシブデバイスにすべてのトランザクションを記録するよう義務付けている政府機関はほとんどありません。
- トラフィックを複数のパッシブデバイスにファンニングする。一部のお客様は、着信トラフィックを複数のパッシブデバイスにファンアウトまたは複製することを好みます。
- トラフィックのスマートな選択。アプライアンスに流入するすべてのパケットは、テキストファイルのダウンロードなど、内容を検査する必要がない場合があります。ユーザーは、NetScaler アプライアンスを構成して、検査する特定のトラフィック（.exe ファイルなど）を選択し、そのトラフィックを IDS デバイスに送信してデータを処理することができます。

NetScaler を L3 接続を備えた IDS デバイスと統合する方法

次の図は、IDS が NetScaler アプライアンスとどのように統合されているかを示しています。



コンポーネントの相互作用は次のように与えられます。

1. クライアントは、HTTP/HTTPS リクエストを NetScaler アプライアンスに送信します。
2. アプライアンスはトラフィックを傍受し、さまざまなデータセンターやクラウドにあるリモート IDS デバイスにデータを送信します。この統合は、IP トンネリングされたレイヤ 3 を介して行われます。NetScaler アプライアンスの IP トンネリングについて詳しくは、「IP トンネル」トピックを参照してください。
3. トラフィックが暗号化されたものである場合、アプライアンスはデータを復号化し、プレーンテキストとして送信します。
4. ポリシー評価に基づいて、アプライアンスは「MIRROR」タイプのコンテンツ検査アクションを適用します。
5. アクションには IDS サービスまたは負荷分散サービス (複数の IDS デバイス統合用) が設定されています。
6. IDS デバイスは、アプライアンス上でコンテンツ検査サービスタイプ「Any」として設定されています。次に、コンテンツ検査サービスは「MIRROR」タイプのコンテンツ検査プロファイルと、データが IDS デバイスに転送される IP トンネリングレイヤ 3 インターフェイスを指定するトンネルパラメータに関連付けられます。

   > 注:
   >
   > オプションで、コンテンツ検査プロファイルに VLAN タグを設定することもできます。

7. 同様に、バックエンドサーバーが NetScaler に応答を送信すると、アプライアンスはデータを複製して IDS デバイスに転送します。
8. アプライアンスが 1 つ以上の IDS デバイスに統合されていて、デバイスの負荷分散を希望する場合は、負荷分散仮想サーバを使用できます。

ソフトウェアライセンス

IDS 統合を導入するには、NetScaler アプライアンスに次のいずれかのライセンスをプロビジョニングする必要があります。

1. ADC Premium
2. ADC Advanced

侵入検知システム統合の設定

IDS デバイスを NetScaler と統合するには、2 つの方法があります。

シナリオ **1:** 単一の **IDS** デバイスとの統合

コマンドラインインターフェイスを使用して設定する必要がある手順を次に示します。

1. コンテンツ検査を有効にする
2. IDS デバイスを表すサービス用に、MIRROR タイプのコンテンツ検査プロファイルを追加します。
3. タイプ「ANY」の IDS サービスを追加する
4. タイプ「MIRROR」のコンテンツ検査アクションを追加する
5. IDS 検査のコンテンツ検査ポリシーを追加する
6. コンテンツ検査ポリシーを HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サービスにバインドする

コンテンツ検査を有効にする　NetScaler アプライアンスから IDS デバイスに検査用のコンテンツを送信する場合は、復号化の実行とは関係なく、コンテンツ検査と負荷分散機能を有効にする必要があります。

コマンドプロンプトで入力します:

```
enable ns feature contentInspection LoadBalancing
```

「**MIRROR**」タイプのコンテンツ検査プロファイルを追加　「MIRROR」タイプのコンテンツ検査プロファイルは、IDS デバイスへの接続方法を説明しています。
コマンドプロンプトで、「」と入力します。

> 注:
>
> IP トンネルパラメータは、レイヤ 3 IDS トポロジにのみ使用する必要があります。それ以外の場合は、出力インターフェイスに egress VLAN オプションを指定して使用する必要があります。GRE/IPIP トンネルタイプは、レイヤ 3 IDS トポロジでサポートされています。


```
add contentInspection profile <name> -type MIRROR -ipTunnel <iptunnel_name
>
```

例:

```
add contentInspection profile IDS_profile1 -type MIRROR -ipTunnel
ipsect-tunnel1
```

IDS サービスの追加　アプライアンスと統合されている IDS デバイスごとに、「ANY」タイプのサービスを設定する必要があります。このサービスには IDS デバイス設定の詳細が含まれています。このサービスは IDS デバイスを表します。

コマンドプロンプトで入力します:

```
add service <Service_name> <IP> ANY <Port> - contentinspectionProfileName
 <Name> -healthMonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service IDS_service 1.1.1.1 ANY 8080 -contentInspectionProfileName
 IDS_profile1 -healthMonitor OFF
```

IDS サービスの **MIRROR** タイプのコンテンツ検査アクションを追加する　コンテンツ検査機能を有効にして IDS プロファイルとサービスを追加したら、要求を処理するための Content Inspection アクションを追加する必要があります。コンテンツ検査アクションに基づいて、アプライアンスは IDS デバイスにデータをドロップ、リセット、ブロック、または送信できます。

コマンドプロンプトで入力します:

```
add ContentInspection action < action_name > -type MIRROR -serverName
 Service_name/Vserver_name>
```

例:

```
add ContentInspection action IDS_action -type MIRROR -serverName
IDS_service
```

IDS 検査のコンテンツ検査ポリシーを追加する　コンテンツ検査アクションを作成したら、コンテンツ検査ポリシーを追加して検査の要求を評価する必要があります。このポリシーは、1 つ以上の式で構成されるルールに基づいています。ポリシーは、ルールに基づいてインスペクション対象のトラフィックを評価し、選択します。

コマンドプロンプトで、次のように入力します:

```
add contentInspection policy < policy_name > -rule <Rule> -action <
action_name>
```

例:


```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプのコンテンツスイッチングまたは負荷分散仮想サービスにバインドする　Web トラフィックを受信するには、負荷分散仮想サーバーを追加する必要があります。
コマンドプロンプトで入力します:

```
add lb vserver <name> <vserver name>
```

例:

```
add lb vserver HTTP_vserver HTTP 1.1.1.3 8080
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプのコンテンツスイッチング仮想サーバーまたは負荷分散仮想サーバーにバインドする　HTTP/SSL タイプの負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーをコンテンツ検査ポリシーにバインドする必要があります。

コマンドプロンプトで、次のように入力します:

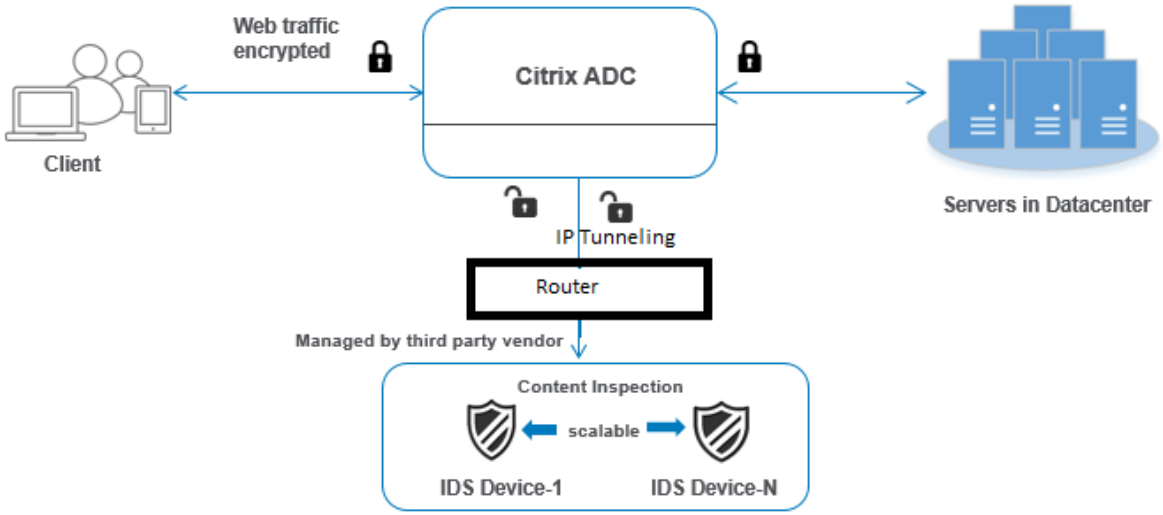
```
bind lb vserver <vserver name> -policyName < policy_name > -priority
< priority > -type <REQUEST>
```

例:

```
bind lb vserver HTTP_vserver -policyName IDS_pol1 -priority 100 -type
 REQUEST
```

シナリオ **2:** 複数の **IDS** デバイスの負荷分散

2 つ以上の IDS デバイスを使用している場合は、異なるコンテンツ検査サービスを使用して IDS デバイスの負荷を分散する必要があります。この場合、NetScaler アプライアンスは、トラフィックのサブセットを各デバイスに送信することに加えて、デバイスの負荷分散を行います。
基本的な設定手順については、シナリオ 1 を参照してください。



コマンドラインインターフェイスを使用して設定する必要がある手順を次に示します。

1. IDS サービス 1 の MIRROR タイプのコンテンツ検査プロファイル 1 を追加します。
2. IDS サービス 2 の MIRROR タイプのコンテンツ検査プロファイル 2 を追加する
3. IDS デバイス 1 にタイプ ANY の IDS サービス 1 を追加する
4. IDS デバイス 2 にタイプ ANY の IDS サービス 2 を追加する
5. ANY タイプの負荷分散仮想サーバを追加する
6. IDS サービス 1 を負荷分散仮想サーバーにバインドする
7. IDS サービス 2 を負荷分散仮想サーバーにバインドする
8. IDS デバイスの負荷分散のためのコンテンツ検査アクションを追加します。
9. 検査用のコンテンツ検査ポリシーを追加する
10. HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーを追加する
11. コンテンツ検査ポリシーを HTTP/SSL タイプの負荷分散仮想サーバーにバインドする

IDS サービス **1** の **MIRROR** タイプのコンテンツ検査プロファイル **1** を追加します　IDS 設定は、コンテンツ検査プロファイルと呼ばれるエンティティで指定できます。プロファイルにはデバイス設定のコレクションがあります。コンテンツ検査プロファイル 1 が IDS サービス 1 用に作成されます。

> 注:
> IP トンネルパラメータは、レイヤ 3 IDS トポロジにのみ使用する必要があります。それ以外の場合は、出力インターフェイスに egress VLAN オプションを指定して使用する必要があります。GRE/IPIP トンネルタイプは、レイヤ 3 IDS トポロジでサポートされています。

コマンドプロンプトで入力します:

```
add contentInspection profile <name> -type ANY - ipTunnel <iptunnel_name
>
```


例:

```
add contentInspection profile IDS_profile1 -type MIRROR - ipTunnel
ipsect_tunnel1
```

IDS サービス **2** のタイプ **MIRROR** のコンテンツ検査プロファイル **2** を追加する　サービス 2 にはコンテンツ検査プロファイル 2 が追加され、インラインデバイスは出力 1/1 インターフェイスを介してアプライアンスと通信します。

コマンドプロンプトで入力します:

```
add contentInspection profile <name> -type ANY - ipTunnel <iptunnel_name
>
```

例:

```
add contentInspection profile IDS_profile2 -type ANY - ipTunnel
ipsect_tunnel2
```

IDS デバイス **1** にタイプ **ANY** の **IDS** サービス **1** を追加する　コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 1 のインラインサービス 1 を負荷分散設定の一部として追加する必要があります。追加したサービスによって、インライン構成の詳細がすべて提供されます。

コマンドプロンプトで入力します:

```
add service <Service_name_1> <Pvt_IP1> ANY <Port> -contentInspectionProfileNam
 <IDS_Profile_1> -usip ON -useproxyport OFF
```

例:

```
add service IDS_service1 1.1.1.1 ANY 80 -contentInspectionProfileName
 IDS_profile1 -usip ON -useproxyport OFF
```

> 注意:
>
> この例で示されている IP アドレスはダミーの IP アドレスです。

IDS デバイス **2** にタイプ **ANY** の **IDS** サービス **2** を追加する　コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 2 にインラインサービス 2 を追加する必要があります。追加したサービスによって、インライン構成の詳細がすべて提供されます。

コマンドプロンプトで入力します:

```
add service <Service_name_1> <Pvt_IP1> ANY -contentInspectionProfileName
 <Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

例:


```
add service IDS_service 1 1.1.2 ANY 80 -contentInspectionProfileName
IDS_profile2
```

> 注意:
>
> この例で示されている IP アドレスはダミーの IP アドレスです。

負荷分散仮想サーバの追加　インラインプロファイルとサービスを追加したら、サービスの負荷分散用の負荷分散仮想サーバを追加する必要があります。

コマンドプロンプトで入力します:

```
add lb vserver <vserver_name> ANY <Pvt_IP3> <port>
```

例:

```
add lb vserver lb-IDS_vserver ANY 1.1.1.2
```

IDS サービス **1** を負荷分散仮想サーバーにバインドする　負荷分散仮想サーバーを追加したら、負荷分散仮想サーバーを最初のサービスにバインドします。

コマンドプロンプトで入力します:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-IDS_vserver IDS_service1
```

IDS サービス **2** を負荷分散仮想サーバーにバインドする　負荷分散仮想サーバーを追加したら、そのサーバーを 2 番目のサービスにバインドします。

コマンドプロンプトで入力します:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-IDS_vserver IDS_service2
```

IDS サービスのコンテンツ検査アクションを追加する　コンテンツ検査機能を有効にしたら、インラインリクエスト情報を処理するための「コンテンツ検査」アクションを追加する必要があります。選択したアクションに基づいて、アプライアンスは IDS デバイスへのトラフィックをドロップ、リセット、ブロック、または送信します。

コマンドプロンプトで入力します:

```
add contentInspection action <name> -type <type> (-serverName <string
> [-ifserverdown <ifserverdown>]
```


例:

```
add ContentInspection action IDS_action -type MIRROR -serverName lb-
IDS_vserver
```

検査用のコンテンツ検査ポリシーを追加する　コンテンツ検査アクションを作成したら、サービス要求を評価するためのコンテンツ検査ポリシーを追加する必要があります。

コマンドプロンプトで、次のように入力します:

```
add contentInspection policy <policy_name> -rule <Rule> -action <
action_name>
```

例:

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーを追加する　Web トラフィックを受け入れるために、コンテンツスイッチングまたは負荷分散仮想サーバーを追加します。また、仮想サーバ上で layer2 接続を有効にする必要があります。

負荷分散の詳細については、「負荷分散の仕組み」トピックを参照してください。

コマンドプロンプトで入力します:

```
add lb vserver <name> <vserver name>
```

例:

```
add lb vserver http_vserver HTTP 1.1.1.1 8080
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプの負荷分散仮想サーバーにバインドする　HTTP/SSL タイプのコンテンツスイッチング仮想サーバーまたは負荷分散仮想サーバーをコンテンツ検査ポリシーにバインドする必要があります。

コマンドプロンプトで、次のように入力します:

```
bind lb vserver <vserver name> -policyName < policy_name > -priority
<> -type <REQUEST>
```

例:

```
bind lb vserver http_vserver -policyName IDS_pol1 -priority 100 -type
 REQUEST
```


NetScaler GUI を使用してインラインサービス統合を構成する

1. [セキュリティ] > [コンテンツ検査] > [コンテンツ検査プロファイル] に移動します。
2. コンテンツインスペクションプロファイルページで、「追加」をクリックします。
3. 「コンテンツインスペクションプロファイルの作成」ページで、次のパラメータを設定します。

    a) プロファイル名。IDS のコンテンツ検査プロファイルの名前。
    b) タイプ。プロファイルタイプを MIRROR として選択します。
    c) 接続性。レイヤ 2 またはレイヤ 3 インターフェイス。
    d) IP トンネル。2 つのネットワーク間のネットワーク通信チャネルを選択します。

4. [作成] をクリックします。
5. トラフィック管理 > 負荷分散 > サービスに移動し、追加をクリックします。
6. [負荷分散サービス] ページで、コンテンツ検査サービスの詳細を入力します。
7. **[**詳細設定**]** セクションで、[プロファイル] をクリックします。
8. [プロファイル] セクションに移動し、[鉛筆] アイコンをクリックしてコンテンツ検査プロファイルを追加します。
9. [**OK**] をクリックします。
10. [負荷分散] > [サーバー] に移動します。HTTP または SSL タイプの仮想サーバを追加します。
11. サーバーの詳細を入力したら、「**OK**」をクリックし、もう一度「**OK**」をクリックします。
12. **[**詳細設定**]** セクションで、[ポリシー] をクリックします。
13. [ポリシー] セクションに移動し、[鉛筆] アイコンをクリックしてコンテンツ検査ポリシーを設定します。
14. 「ポリシーの選択」ページで、「コンテンツ検査」を選択します。[続行] をクリックします。
15. [ポリシーバインド] セクションで、[+] をクリックしてコンテンツ検査ポリシーを追加します。
16. [**CI** ポリシーの作成] ページで、インラインコンテンツ検査ポリシーの名前を入力します。
17. [**Action**] フィールドで [+] 記号をクリックし、MIRROR タイプの IDS コンテンツ検査アクションを作成します。
18. **[CI** アクションの作成] ページで、次のパラメータを設定します。

    a) Name: コンテンツ検査インラインポリシーの名前。
    b) タイプ。タイプとして MIRROR を選択します。
    c) サーバー名: サーバ/サービス名を [インラインデバイス] として選択します。
    d) サーバーがダウンした場合。サーバがダウンした場合のオペレーションを選択します。
    e) リクエストのタイムアウト。タイムアウト値を選択します。デフォルト値を使用できます。
    f) タイムアウトアクションの要求。タイムアウトアクションを選択します。デフォルト値を使用できます。

19. [作成] をクリックします。
20. **[CI** ポリシーの作成] ページで、その他の詳細を入力します。
21. 「**OK**」をクリックして「閉じる」をクリックします。

負荷分散と IDS デバイスへのトラフィックの複製のための NetScaler GUI 構成については、「負荷分散」を参照してください。

コンテンツ変換後にトラフィックをバックエンドオリジンサーバーに負荷分散および転送するための NetScaler GUI 構成の詳細については、「負荷分散」を参照してください。

ICAP、IPS、および IDS のコンテンツインスペクション統計情報

August 15, 2023

ICAP、インラインデバイスインテグレーション（IDS）、および侵入防止システム（IPS）デバイスのコンテンツ検査統計情報は、要求、応答、およびサーバアクションの詳細の詳細な出力（要約）です。

コンテンツ検査統計は、コンテンツ検査のために送信された HTTP/HTTPS 要求を含む統計データの集まりです。IPS、IDS、ICAP デバイスから受信した HTTP/HTTPS 応答とバックエンドサーバーのアクション。

CLI を使用してコンテンツ検査統計情報を表示するには：

コマンドプロンプトで入力します。

```
stat contentInspection
```

```
ContentInspection Stats

Inline Statistics
                                  Total
Requests                           10
Responses                           6
Request Bytes Sent                 3235
Request Bytes Received             2977
Response Bytes Sent                17302
Response Bytes Received            19681
Serverdown Reset Action taken       1
Serverdown Drop Action taken        0
Serverdown BYPASS Action taken      0
Inline device Generated Response    3

Mirror Statistics
                                  Total
Requests                           4
Responses                          4
Requests Bytes Sent               2763
Responses Bytes Sent              16732
Serverdown Reset Action taken      0
Serverdown Drop Action taken       0
Serverdown BYPASS Action taken     1

ICAP Statistics
                                  Total
REQMOD requests Sent                6
RESPMOD requests Sent               4
Preview requests                    1
```



```
204 Responses Received                6
100 Continue Responses Received       1
204 NO content Received               5
Adaptive Requests                     0
Adaptive Responses                    4
Callout requests Initiated            1
Callout requests completed            1
ICAP Req/Resp Errors handled          1
Serverdown Reset Action taken         1
Serverdown Drop Action taken          0
Serverdown BYPASS Action taken        1

Done
<!--NeedCopy-->
```

SSL フォワードプロキシ

March 20, 2024

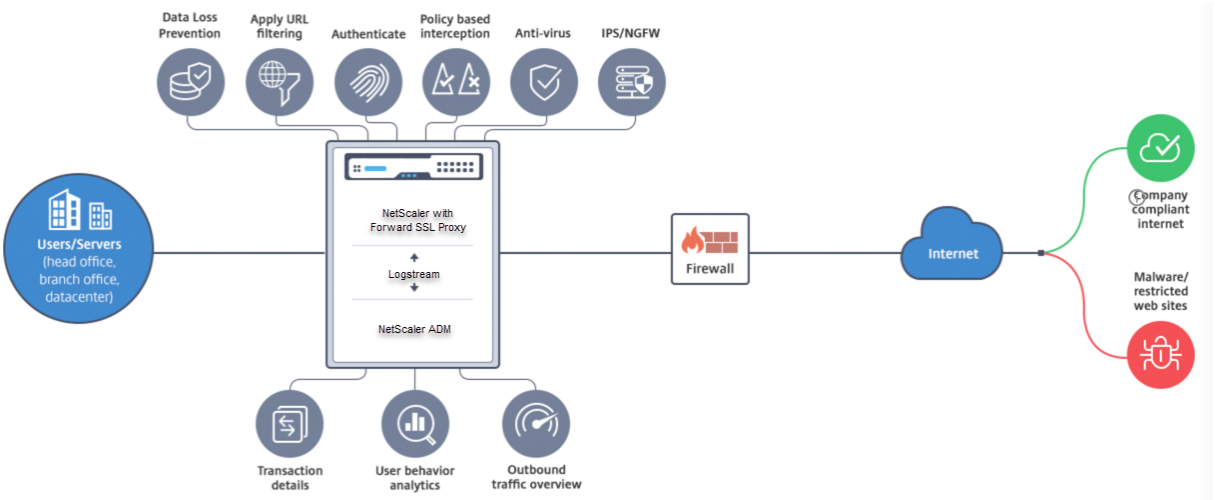
注**:** SSL 転送プロキシ機能は ADC Premium ライセンスで利用できます。

近年、ウェブトラフィックは指数関数的に増加しており、企業は日常業務にインターネットに依存する傾向が高まっています。これに加えて、エンドポイント、モビリティ、BYOD の多様化、攻撃者ベースの増加が相まって、ユーザーは現代のマルウェアの標的になりやすくなっています。個人情報の盗難やデータの漏洩に対する脆弱性がますます高まっています。従来、企業は HTTP トラフィックにマルウェアやウイルスがないか検査してきました。HTTPS/TLS トラフィックはそれほど目立たないため、バイパスしました。機密性が高く信頼できるコンテンツには控えめに使用されていました。しかし、今ではほとんどの公共のインターネットウェブサイトがユーザーのプライバシーを保護するために HTTPS を使用することを好むようになったため、状況は急速に変化しています。その結果、暗号化されたパケットを検査できないため、マルウェアや企業ネットワークへの侵入が許されます。SSL 転送プロキシソリューションは、企業がインターネットの脅威から保護するために使用できるツールを提供します。

プロキシは、ユーザーとインターネットまたは SaaS アプリケーション間のすべてのトラフィックを制御するサーバーです。すべてのトラフィックはこのプロキシを通過するため、ユーザー認証や URL 分類などのセキュリティ関連の機能を実行します。

次の図は、SSL 転送プロキシの実装の概要です。トラフィックは、本社、支社、データセンター、およびリモート従業員から企業ネットワークを経由して流れます。ネットワークの端にある NetScaler アプライアンスはプロキシの役割を果たします。このアプライアンスは透過プロキシモードまたは明示的プロキシモードで動作し、HTTPS を含むインターネットトラフィックの傍受を制御できるようにします。アプライアンスに設定されたポリシーによって、特定のリクエストをインターセプト、バイパス、またはブロックするかどうかが決まります。制限付きサイトへのアクセスは、URL フィルタリングを使用してブロックできます。ユーザーは社内ネットワークにログオンする前に認証されます。すべてのリクエストとレスポンスにはユーザーを識別するためのタグが付けられ、インターネットサイトへのアクセスは分類されます。ユーザーアクティビティはログに記録され、レポートの生成に使用されます。侵害

が発生した場合、管理者は感染したシステムを隔離し、その Web サイトにアクセスした他のユーザーのデバイスが侵害されていないかどうかを判断し、適切な措置を講じることができます。NetScaler アプリケーション配信管理（ADM）を SSL フォワードプロキシと統合すると、アプライアンス内のログに記録されたユーザーアクティビティとその後のレコードが、`logstream`を使用して NetScaler Console にエクスポートされます。NetScaler Console は、訪問した Web サイトからオンラインで過ごした時間まで、ユーザーのアクティビティに関する情報を照合して表示します。また、帯域幅の使用やマルウェアやフィッシングサイトなどの検出された脅威に関する情報も提供します。これらの主要なメトリックを使用してネットワークを監視し、SSL 転送プロキシ機能を使用して是正措置を講じることができます。



SSL 転送プロキシにより、IT ディレクターは次のことを実行できます。

- バイパスされたセキュアトラフィックを可視化します。
- 悪意のあるサイトや未知のサイトへのアクセスをブロックし、企業内のユーザーへの感染を防ぎます。
- 個人用メール、ソーシャルネットワーキング、求人検索の Web サイトなど、一部の Web サイトへのアクセスを企業ネットワークから制御します。
- インテリジェントなコンテンツ管理ポリシーを適用して、ユーザーの生産性を最大化します。

SSL フォワードプロキシ機能の開始

August 15, 2023

重要:

- OCSP チェックでは、証明書の有効性をチェックするためにインターネット接続が必要です。NSIP アドレスを使用してインターネットからアプライアンスにアクセスできない場合は、アクセス制御リスト（ACL）を追加して、NSIP アドレスからサブネット IP（SNIP）アドレスに NAT を実行します。SNIP はインターネットにアクセスできる必要があります。たとえば、次のようにします。


```
add ns acl a1 ALLOW -srcIP = <NSIP> -destIP  “!=”
    10.0.0.0-10.255.255.255

add rnat RNAT-1 a1

bind rnat RNAT-1 <SNIP>

apply acls
<!--NeedCopy-->
```

- ドメイン名を解決する DNS ネームサーバーを指定します。
- アプライアンスの日付が NTP サーバと同期していることを確認します。日付が同期されていない場合、アプライアンスはオリジンサーバー証明書の有効期限が切れているかどうかを効果的に検証できません。

SSL フォワードプロキシ機能を使用するには、次のタスクを実行する必要があります。

- エクスプリシットモードまたはトランスペアレントモードでプロキシサーバーを追加します。
- SSL インターセプトを有効にします。
  - SSL プロファイルを構成します。
  - SSL ポリシーをプロキシサーバーに追加してバインドします。
  - SSL インターセプト用の CA 証明書とキーのペアを追加してバインドします。

注:

透過プロキシモードで構成された ADC アプライアンスは、HTTP プロトコルと HTTPS プロトコルのみをインターセプトできます。telnet などの他のプロトコルをバイパスするには、プロキシ仮想サーバーに次のリッスンポリシーを追加する必要があります。

仮想サーバーは、HTTP と HTTPS の着信トラフィックのみを受け入れるようになりました。

```
set cs vserver transparent-pxy1 PROXY * * -cltTimeout 180 -Listenpolicy
    "CLIENT.TCP.DSTPORT.EQ(80) || CLIENT.TCP.DSTPORT.EQ(443)"`
<!--NeedCopy-->
```

展開によっては、次の機能を設定する必要がある場合があります。

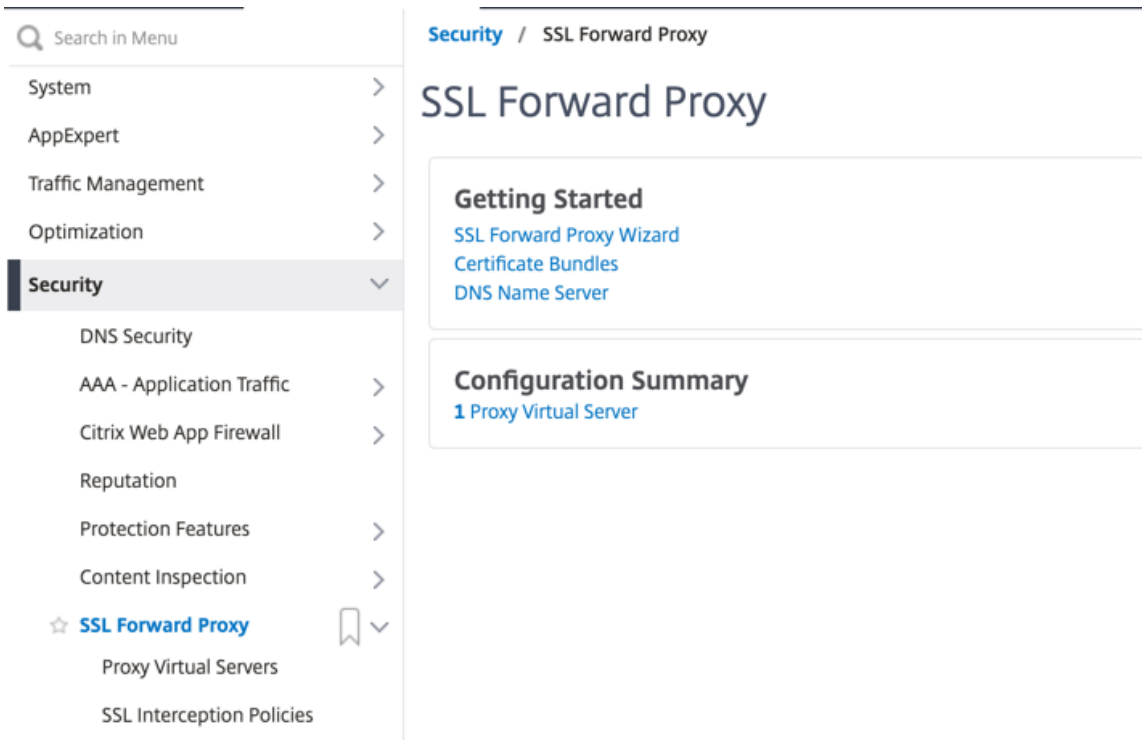
- 認証サービス (推奨) —ユーザーを認証します。認証サービスがない場合、ユーザーアクティビティはクライアント IP アドレスに基づきます。
- URL フィルタリング—カテゴリ、レピュテーションスコア、および URL リストによって URL をフィルタリングします。
- 分析-NetScaler Application Delivery Management（ADM）でのユーザーアクティビティ、ユーザーリスクインジケータ、帯域幅消費、トランザクションの分類を表示します。

注**:** SSL フォワードプロキシは、最も一般的な HTTP および HTTPS 標準を実装し、それに続く類似の製品を実装この実装は、特定のブラウザを念頭に置いて行われ、最も一般的なブラウザと互換性があります。SSL フォワードプロキシは、一般的なブラウザと Google Chrome、Internet Explorer、および Mozilla Firefox の最新バージョンでテストされています。

SSL 転送プロキシウィザード

SSL フォワードプロキシウィザードは、Web ブラウザを使用して SSL フォワードプロキシの展開全体を管理するためのツールを管理者に提供します。これは、お客様が SSL フォワードプロキシサービスを迅速に開始できるようにガイドし、明確に定義された一連の手順に従うことで構成を簡素化するのに役立ちます。

1. セキュリティ **> SSL** フォワードプロキシに移動します。[はじめに] で [**SSL** 転送プロキシウィザード**]** をクリックします



2. ウィザードの手順に従って配置を構成します。

透過プロキシサーバーへのリッスンポリシーの追加

1. セキュリティ **> SSL** フォワードプロキシ > プロキシ仮想サーバーに移動します。透過プロキシサーバを選択し、[編集] をクリックします。
2. [基本設定**]** を編集し、[その他] をクリックします。
3. [リスン優先度] に 1 と入力します。
4. [リッスンポリシー式] に、次の式を入力します。

```
(CLIENT.TCP.DSTPORT.EQ(80)||CLIENT.TCP.DSTPORT.EQ(443))
<!--NeedCopy-->
```

この式は、HTTP および HTTPS トラフィックの標準ポートを想定しています。HTTP には 8080、HTTPS には 8443 など、異なるポートを設定した場合は、それらのポートを反映するように式を変更します。

制限事項

SSL 転送プロキシは、クラスタ設定、管理パーティション、および NetScaler ADC FIPS アプライアンスではサポートされません。

プロキシモード

December 8, 2023

NetScaler アプライアンスは、インターネットや SaaS アプリケーションに接続するクライアントのプロキシとして機能します。プロキシとして、すべてのトラフィックを受け入れ、トラフィックのプロトコルを決定します。トラフィックが HTTP または SSL でない限り、そのまま宛先に転送されます。アプライアンスはクライアントから要求を受け取ると、要求をインターセプトし、ユーザー認証、サイト分類、リダイレクトなどのアクションを実行します。ポリシーを使用して、許可するトラフィックとブロックするトラフィックを決定します。

アプライアンスは 2 つの異なるセッションを維持します。1 つはクライアントとプロキシの間、もう 1 つはプロキシとオリジンサーバーの間です。プロキシは、ユーザーが定義したポリシーに基づいて、HTTP および HTTPS トラフィックを許可またはブロックします。そのため、財務情報などの機密データをバイパスするポリシーを定義することが重要です。アプライアンスには、トラフィック管理ポリシーを作成するためのレイヤー 4〜レイヤー 7 のトラフィック属性とユーザー ID 属性の豊富なセットが用意されています。

SSL トラフィックの場合、プロキシはオリジンサーバーの証明書を確認し、サーバーとの正当な接続を確立します。次に、サーバー証明書をエミュレートし、NetScaler にインストールされた CA 証明書を使用して署名し、作成したサーバー証明書をクライアントに提示します。SSL セッションを正常に確立するには、CA 証明書を信頼できる証明書としてクライアントのブラウザに追加する必要があります。

アプライアンスは、トランスペアレントプロキシモードとエクスプリシットプロキシモードをサポートしています。明示的なプロキシモードでは、組織が設定をクライアントのデバイスにプッシュしない限り、クライアントはブラウザで IP アドレスを指定する必要があります。このアドレスは、ADC アプライアンスで設定されているプロキシサーバーの IP アドレスです。すべてのクライアント要求は、この IP アドレスに送信されます。明示的プロキシの場合は、PROXY タイプのコンテンツスイッチング仮想サーバーを構成し、IP アドレスと有効なポート番号を指定する必要があります。また、デフォルトの HTTP `markconnReqInval` プロファイルでパラメータがグローバルに ON に設定されている場合は、OFF `markconnReqInval` に設定された別の HTTP プロファイルをコンテンツスイッチング仮想サーバーにバインドする必要があります。

カスタム **HTTP** プロファイルをプロキシコンテンツスイッチング仮想サーバーにバインドする例**:**

```
1 add ns httpprofile custom_http_profile1 -markconnReqInval OFF
2 set cs vserver swgVS -httpprofileName custom_http_profile1
3 <!--NeedCopy-->
```

透過プロキシは、名前が示すように、クライアントに対して透過的です。つまり、クライアントは、プロキシサーバーが要求を仲介していることを認識していない可能性があります。ADC アプライアンスはインライン展開で構成さ

れ、すべての HTTP および HTTPS トラフィックを透過的に受け入れます。トランスペアレントプロキシの場合、IP アドレスおよびポートとしてアスタリスク（*）を使用して、タイプ PROXY のコンテンツスイッチング仮想サーバを設定する必要があります。GUI で **SSL** 転送プロキシウィザードを使用する場合、IP アドレスとポートを指定する必要はありません。

> 注
>
> 透過プロキシモードで HTTP および HTTPS 以外のプロトコルを代行受信するには、リッスンポリシーを追加し、プロキシサーバにバインドする必要があります。

CLI を使用して **SSL** 転送プロキシを設定する

コマンドプロンプトで入力します:

```
add cs vserver <name> PROXY <ipaddress> <port>
<!--NeedCopy-->
```

引数:

名前:

プロキシサーバーの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。CS 仮想サーバーの作成後は変更できません。

次の要件は、CLI だけに適用されます。

名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲みます (たとえば、「my server」や「my server」)。

この引数は必須です。最大長：127

IP アドレス**:**

プロキシサーバの IP アドレス。

ポート:

プロキシサーバーのポート番号。最小値:1

明示的なプロキシの例:

```
add cs vserver swgVS PROXY 192.0.2.100 80
<!--NeedCopy-->
```

透過プロキシの例:

```
add cs vserver swgVS PROXY * *
<!--NeedCopy-->
```


GUI を使用してトランスペアレントプロキシサーバにリッスンポリシーを追加します

1. [セキュリティ] > [**SSL** フォワードプロキシ] > [プロキシ仮想サーバー] に移動します。透過プロキシサーバを選択し、[編集] をクリックします。
2. [基本設定**]** を編集し、[その他] をクリックします。
3. [リスン優先度] に 1 と入力します。
4. [リッスンポリシー式] に、次の式を入力します。

```
(CLIENT.TCP.DSTPORT.EQ(80)||CLIENT.TCP.DSTPORT.EQ(443))
<!--NeedCopy-->
```

> 注
>
> この式は、HTTP および HTTPS トラフィックの標準ポートを想定しています。HTTP には 8080、HTTPS には 8443 など、異なるポートを設定した場合は、前述の式を変更してそれらのポートを指定します。

SSL インターセプト

August 15, 2023

SSL インターセプト用に構成された NetScaler アプライアンスはプロキシとして機能します。SSL/TLS トラフィックを傍受および復号化し、暗号化されていない要求を検査し、管理者がコンプライアンスルールとセキュリティチェックを適用できるようにします。SSL インターセプションでは、代行受信、ブロック、または許可するトラフィックを指定するポリシーを使用します。たとえば、銀行などの金融ウェブサイトとの間のトラフィックは傍受してはいけませんが、他のトラフィックは傍受でき、ブラックリストに載っているサイトを特定してブロックすることはできます。トラフィックを傍受する一般的なポリシーを 1 つ構成し、一部のトラフィックをバイパスするより具体的なポリシーを構成することをお勧めします。

クライアントとプロキシは HTTPS/TLS ハンドシェイクを確立します。プロキシはサーバーと別の HTTPS/TLS ハンドシェイクを確立し、サーバー証明書を受け取ります。プロキシはクライアントに代わってサーバー証明書を検証し、オンライン証明書ステータスプロトコル (OCSP) を使用してサーバー証明書の有効性も確認します。サーバー証明書を再生成し、アプライアンスにインストールされている CA 証明書のキーを使用して署名し、クライアントに提示します。そのため、クライアントと NetScaler アプライアンスの間では 1 つの証明書が使用され、アプライアンスとバックエンドサーバーの間では別の証明書が使用されます。

> 重要
>
> 再生成されたサーバー証明書がクライアントによって信頼されるように、サーバー証明書の署名に使用される CA 証明書は、すべてのクライアントデバイスにプレインストールする必要があります。

傍受された HTTPS トラフィックの場合、プロキシサーバは送信トラフィックを復号化し、クリアテキストの HTTP 要求にアクセスし、任意のレイヤ 7 アプリケーションを使用してトラフィックを処理できます。たとえば、プレーンテキスト URL を調べたり、企業ポリシーや URL レピュテーションに基づいてアクセスを許可またはブロックしたりできます。ポリシーがオリジンサーバーへのアクセスを許可することである場合、プロキシサーバーは再暗号化されたリクエストを（オリジンサーバー上の）宛先サービスに転送します。プロキシは、オリジンサーバーからの応答を復号化し、クリアテキストの HTTP 応答にアクセスし、オプションで任意のポリシーを応答に適用します。次に、プロキシは応答を再暗号化してクライアントに転送します。ポリシーがオリジンサーバーへの要求をブロックする場合、プロキシは HTTP 403 などのエラー応答をクライアントに送信できます。

SSL インターセプトを実行するには、以前に設定したプロキシサーバーに加えて、ADC アプライアンスで次の設定を行う必要があります。

- SSL プロファイル
- SSL ポリシー
- CA 証明書ストア
- SSL エラーの自動学習とキャッシュ

> 注:
>
> HTTP/2 トラフィックは SSL インターセプト機能によって傍受されません。

SSL インターセプト証明書ストア

SSL 証明書は、SSL トランザクションの一部であり、企業（ドメイン）または個人を識別するデジタルデータフォーム（X509）です。SSL 証明書は認証局 (CA) によって発行されます。CA はプライベートでもパブリックでもかまいません。Verisign などのパブリック CA によって発行された証明書は、SSL トランザクションを実行するアプリケーションによって信頼されます。これらのアプリケーションは、信頼する CA の一覧を維持します。

ADC アプライアンスはフォワードプロキシとして、クライアントとサーバー間のトラフィックの暗号化と復号化を実行します。これは、クライアント（ユーザー）のサーバーとして、およびサーバーのクライアントとして機能します。アプライアンスが HTTPS トラフィックを処理する前に、不正なトランザクションを防ぐためにサーバーの ID を検証する必要があります。したがって、オリジンサーバーのクライアントとして、アプライアンスはオリジンサーバー証明書を受け入れる前にオリジンサーバー証明書を確認する必要があります。サーバ証明書を検証するには、サーバ証明書の署名および発行に使用されるすべての証明書（ルート証明書および中間証明書など）がアプライアンス上に存在する必要があります。デフォルトの CA 証明書セットは、アプライアンス上にプレインストールされています。アプライアンスは、これらの証明書を使用して、ほぼすべての一般的なオリジンサーバ証明書を検証できます。この既定のセットは変更できません。ただし、導入環境でさらに多くの CA 証明書が必要な場合は、そのような証明書のバンドルを作成して、そのバンドルをアプライアンスにインポートできます。バンドルには 1 つの証明書を含めることもできます。

証明書バンドルをアプライアンスにインポートすると、アプライアンスはリモートの場所からバンドルをダウンロードし、バンドルに証明書のみが含まれていることを確認した後、アプライアンスにインストールします。証明書バン

ドルを使用してサーバー証明書を検証する前に、証明書バンドルを適用する必要があります。また、証明書バンドルをエクスポートして編集したり、オフラインの場所にバックアップとして保存したりすることもできます。

CLI を使用して **CA** 証明書バンドルをアプライアンスにインポートして適用します

コマンドプロンプトで入力します。

```
import ssl certBundle <name> <src>
apply ssl certBundle <name>
<!--NeedCopy-->
```

```
show ssl certBundle
<!--NeedCopy-->
```

引数:

名前:

インポートした証明書バンドルに割り当てる名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。次の要件は、CLI だけに適用されます。

名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲みます (たとえば、「my file」や「my file」)。

最大長: 31

アーク**:**

インポートまたはエクスポートする証明書バンドルのプロトコル、ホスト、およびパス (ファイル名を含む) を指定する URL。例: `http://www.example.com/cert_bundle_file`。

注: インポートするオブジェクトが、アクセスにクライアント証明書認証を必要とする HTTPS サーバー上にある場合、インポートは失敗します。

最大長: 2047

例:

```
import ssl certbundle swg-certbundle http://www.example.com/cert_bundle
apply ssl certBundle swg-certbundle
<!--NeedCopy-->
```

```
show ssl certbundle

            Name : swg-certbundle(Inuse)

            URL : http://www.example.com/cert_bundle

    Done
<!--NeedCopy-->
```


GUI を使用して **CA** 証明書バンドルをアプライアンスにインポートして適用する

1. セキュリティ **>** **SSL** 転送プロキシ **>** はじめに **>** 証明書バンドルに移動します。
2. 次のいずれかを行います:
   - リストから証明書バンドルを選択します。
   - 証明書バンドルを追加するには、「+」をクリックし、名前とソース URL を指定します。[**OK**] をクリックします。
3. [**OK**] をクリックします。

CLI を使用して **CA** 証明書バンドルをアプライアンスから削除する

コマンドプロンプトで入力します。

```
remove certBundle <cert bundle name>
<!--NeedCopy-->
```

例:

```
remove certBundle mytest-cacert
<!--NeedCopy-->
```

CLI を使用して **CA** 証明書バンドルをアプライアンスからエクスポートする

コマンドプロンプトで入力します。

```
export certBundle <cert bundle name> <Path to export>
<!--NeedCopy-->
```

引数:

名前:

インポートした証明書バンドルに割り当てる名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) 文字のみを含める必要があります。次の要件は、CLI だけに適用されます。

名前にスペースが 1 つ以上含まれる場合は、名前を二重引用符または一重引用符で囲みます (たとえば、「my file」や「my file」)。

最大長: 31

アーク**:**

インポートまたはエクスポートする証明書バンドルのプロトコル、ホスト、およびパス (ファイル名を含む) を指定する URL。例: `http://www.example.com/cert_bundle_file`。

注: インポートするオブジェクトが、アクセスにクライアント証明書認証を必要とする HTTPS サーバー上にある場合、インポートは失敗します。

最大長: 2047

例:

```
export certBundle mytest-cacert http://192.0.2.20/
<!--NeedCopy-->
```

Mozilla CA 証明書ストアから **CA** 証明書バンドルをインポート、適用、検証する

コマンドプロンプトで入力します。

```
> import certbundle mozilla_public_ca https://curl.haxx.se/ca/cacert.
   pem
Done
<!--NeedCopy-->
```

バンドルを適用するには、次のように入力します。

```
> apply certbundle mozilla_public_ca
Done
<!--NeedCopy-->
```

使用中の証明書バンドルを確認するには、次のように入力します。

```
> sh certbundle | grep mozilla
                Name : mozilla_public_ca (Inuse)
<!--NeedCopy-->
```

制限事項

- 証明書バンドルは、クラスタ設定またはパーティション化されたアプライアンスではサポートされません。
- TLSv1.3 プロトコルは SSL 転送プロキシではサポートされていません。

SSL インターセプト用の **SSL** ポリシーインフラストラクチャ

ポリシーは、着信トラフィックに対するフィルタのように動作します。ADC アプライアンスのポリシーは、プロキシ接続と要求の管理方法を定義するのに役立ちます。処理は、そのポリシーに対して設定されているアクションに基づきます。つまり、接続要求のデータはポリシーで指定された規則と比較され、規則に一致する接続にアクションが適用されます (式)。ポリシーに割り当てるアクションを定義してポリシーを作成したら、そのアクションをプロキシサーバにバインドして、そのプロキシサーバを流れるトラフィックに適用する必要があります。

SSL インターセプトの SSL ポリシーは、受信トラフィックを評価し、ルール (表現) に一致するリクエストに定義済みのアクションを適用します。接続をインターセプト、バイパス、またはリセットするかどうかは、定義された SSL

ポリシーに基づいて決定されます。ポリシーに対して、INTERCEPT、BYPASS、または RESET の 3 つのアクションのいずれかを設定できます。ポリシーを作成するときは、アクションを指定する必要があります。ポリシーを有効にするには、アプライアンスのプロキシサーバーにポリシーをバインドする必要があります。ポリシーが SSL インターセプトを対象としていることを指定するには、ポリシーをプロキシサーバーにバインドするときに、タイプ (バインドポイント) を INTERCEPT_REQ として指定する必要があります。ポリシーをバインド解除するときは、タイプを INTERCEPT_REQ として指定する必要があります。

> 注記:
>
> ポリシーを指定しない限り、プロキシサーバーはインターセプトの決定を下すことができません。

トラフィックインターセプトは、任意の SSL ハンドシェイク属性に基づいて行うことができます。最も一般的に使用されるのは SSL ドメインです。SSL ドメインは通常、SSL ハンドシェイクの属性によって示されます。これは、SSL Client Hello メッセージから抽出されたサーバー名インジケータ値（存在する場合）、または元のサーバー証明書から抽出されたサーバー別名（SAN）値になります。SSL 代行受信ポリシーは、特別な属性 DETECTED_DOMAIN を提示します。この属性により、顧客はオリジンサーバー証明書からの SSL ドメインに基づいて代行受信ポリシーを簡単に作成できるようになります。顧客は、ドメイン名を文字列、URL リスト（URL セットまたは`patset`）、またはドメインから派生した URL カテゴリと照合できます。

CLI を使用して **SSL** ポリシーを作成します

コマンドプロンプトで入力します。

```
add ssl policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

例:

次の例は、`detected_domain` 属性を使用してドメイン名を確認する式を含むポリシーのものです。

XYZBANK などの金融機関へのトラフィックを傍受しない

```
add ssl policy pol1 -rule client.ssl.detected_domain.contains("XYZBANK"
    ) -action BYPASS
<!--NeedCopy-->
```

ユーザーが企業ネットワークから YouTube に接続することを許可しないでください

```
add ssl policy pol2 -rule client.ssl.client.ssl.detected_domain.
    url_categorize(0,0).category.eq ("YouTube") -action RESET
<!--NeedCopy-->
```

すべてのユーザートラフィックを傍受する

```
add ssl policy pol3  - rule true  - action INTERCEPT
<!--NeedCopy-->
```


お客様が detected_domain を使用したくない場合は、任意の SSL ハンドシェイク属性を使用してドメインを抽出および推測できます。

たとえば、クライアントの hello メッセージの SNI 拡張にドメイン名が見つかりません。ドメイン名はオリジンサーバー証明書から取得する必要があります。次の例は、オリジンサーバー証明書のサブジェクト名でドメイン名をチェックする式を持つポリシーの例です。

任意の Yahoo ドメインへのすべてのユーザートラフィックを傍受する

```
add ssl policy pol4 -rule client.ssl.origin_server_cert.subject.
    contains("yahoo")  - action INTERCEPT
<!--NeedCopy-->
```

「ショッピング/小売」カテゴリのすべてのユーザートラフィックを傍受する

```
add ssl policy pol_url_category -rule client.ssl.origin_server_cert.
    subject.URL_CATEGORIZE(0,0).CATEGORY.eq("Shopping/Retail") -action
    INTERCEPT
<!--NeedCopy-->
```

未分類の URL へのすべてのユーザトラフィックを代行受信する

```
add ssl policy pol_url_category -rule client.ssl.origin_server_cert.
    subject.url_categorize(0,0).category.eq("Uncategorized") -action
    INTERCEPT
<!--NeedCopy-->
```

次の例は、URL セットのエントリに対してドメインを照合するポリシーの例です。

SNI のドメイン名が URL セット「top100」のエントリと一致する場合、すべてのユーザートラフィックを代行受信します。

```
add ssl policy pol_url_set  -rule client.ssl.client_hello.SNI.
    URLSET_MATCHES_ANY("top100") -action INTERCEPT
<!--NeedCopy-->
```

オリジンサーバー証明書が URL セット「top100」のエントリと一致する場合、ドメイン名のすべてのユーザートラフィックを代行受信します。

```
add ssl policy pol_url_set  -rule client.ssl.origin_server_cert.subject
    .URLSET_MATCHES_ANY("top100") -action INTERCEPT
<!--NeedCopy-->
```

GUI を使用したプロキシサーバーへの **SSL** ポリシーの作成

1. [**Traffic Management**] > [**SSL**] > [**Policies**] に移動します。
2. 「**SSL** ポリシー」タブで、「追加」をクリックし、次のパラメータを指定します。

    - ポリシー名

- ポリシーアクション—インターセプト、バイパス、またはリセットから選択します。
- 式

3. ［作成］をクリックします。

CLI を使用して **SSL** ポリシーをプロキシサーバーにバインドする

コマンドプロンプトで入力します。

```
bind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type  INTERCEPT_REQ
<!--NeedCopy-->
```

例:

```
bind ssl vserver <name> -policyName pol1 -priority 10 -type
   INTERCEPT_REQ
<!--NeedCopy-->
```

GUI を使用して **SSL** ポリシーをプロキシサーバにバインドする

1. [セキュリティ**] > [SSL** 転送プロキシ**] > [**プロキシ仮想サーバー] に移動します。
2. 仮想サーバを選択し、[編集（Edit）] をクリックします。
3. [詳細設定] で、[**SSL** ポリシー] をクリックします。
4. **SSL** ポリシーボックス内をクリックします。
5.「ポリシーの選択」で、バインドするポリシーを選択します。
6.「タイプ」で「インターセプト **_REQ**」を選択します。
7. [バインド] をクリックし、[**OK**] をクリックします。

CLI を使用して **SSL** ポリシーをプロキシサーバーにバインド解除します

コマンドプロンプトで入力します。

```
unbind ssl vserver <vServerName> -policyName <string> -type
   INTERCEPT_REQ
<!--NeedCopy-->
```

SSL ポリシーで使用される **SSL** 表現

式	説明
CLIENT.SSL.CLIENT_HELLO.SNI.*	SNI 拡張子を文字列形式で返します。文字列を評価して、指定されたテキストが含まれているかどうかを確認します。例:client.ssl.client_hello.sni.contains () “xyz.com”
CLIENT.SSL.ORIGIN_SERVER_CERT.*	バックエンドサーバーから受け取った証明書を文字列形式で返します。文字列を評価して、指定されたテキストが含まれているかどうかを確認します。例:client.ssl.origin_server_cert.subject.contains () “xyz.com”
CLIENT.SSL.DETECTED_DOMAIN.*	SNI 拡張またはオリジンサーバー証明書のドメインを文字列形式で返します。文字列を評価して、指定されたテキストが含まれているかどうかを確認します。例: クライアント.ssl.detected_domain.contains () “xyz.com”

SSL エラーの自動学習中

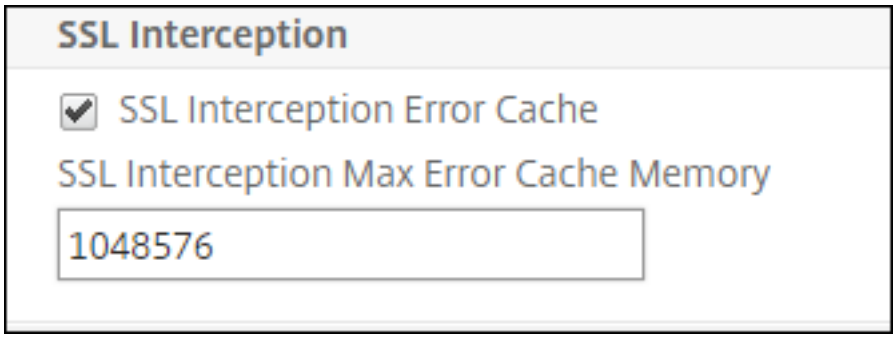
ラーニングモードがオンの場合、アプライアンスは SSL バイパスリストにドメインを追加します。ラーニングモードは、クライアントまたはオリジナルサーバから受信した SSL アラートメッセージに基づいています。つまり、学習はアラートメッセージを送信するクライアントまたはサーバーによって異なります。アラートメッセージが送信されない場合、ラーニングは行われません。アプライアンスは、次の条件のいずれかが満たされているかどうかを学習します。

1. クライアント証明書の要求がサーバーから受信されます。
2. ハンドシェイクの一部として、次のアラートのいずれかが受信されます。
    - BAD_CERTIFICATE
    - UNSUPPORTED_CERTIFICATE
    - CERTIFICATE_REVOKED
    - CERTIFICATE_EXPIRED
    - CERTIFICATE_UNKNOWN
    - UNKNOWN_CA (クライアントがピニングを使用している場合、サーバー証明書を受け取ると、このアラートメッセージが送信されます。)
    - HANDSHAKE_FAILURE

学習を有効にするには、エラーキャッシュを有効にし、学習用に予約したメモリを指定する必要があります。

GUI を使用して学習を有効にする

1. [トラフィック管理] **> [SSL**] に移動します。
2. [設定] で、[**SSL** の詳細設定の変更] をクリックします。
3.「**SSL** インターセプション」で、「**SSL** インターセプションエラーキャッシュ」を選択します。
4.「**SSL** インターセプション最大エラーキャッシュメモリ」で、予約するメモリ (バイト単位) を指定します。



5. [**OK**] をクリックします。

CLI を使用して学習を有効にする

コマンドプロンプトで次のように入力します。

```
set ssl parameter -ssliErrorCache ( ENABLED | DISABLED ) -
    ssliMaxErrorCacheMem <positive_integer>
<!--NeedCopy-->
```

引数:

ssliErrorCache:

動的学習を有効または無効にし、学習した情報をキャッシュして、後でリクエストをインターセプトするかバイパスするかを決定します。有効にすると、アプライアンスはキャッシュ検索を実行して、要求をバイパスするかどうかを決定します。

設定可能な値: ENABLED, DISABLED

デフォルト値: 無効

ssliMaxErrorCacheMem:

学習したデータのキャッシュに使用できる最大メモリをバイト単位で指定します。このメモリは LRU キャッシュとして使用され、設定したメモリ制限がなくなると、古いエントリが新しいエントリに置き換えられます。値 0 を指定すると、自動的に制限が決定されます。

デフォルト値:0

最小値:0

最大値:4294967294

SSL プロファイル

SSL プロファイルは、暗号やプロトコルなどの SSL 設定の集まりです。プロファイルは、さまざまなサーバーに共通の設定がある場合に役立ちます。各サーバーに同じ設定を指定する代わりに、プロファイルを作成してプロファイルで設定を指定し、プロファイルを別のサーバーにバインドできます。カスタムフロントエンド SSL プロファイルが作成されていない場合、デフォルトのフロントエンドプロファイルはクライアント側のエンティティにバインドされます。このプロファイルでは、クライアント側の接続を管理するための設定を構成できます。

SSL インターセプトの場合は、SSL プロファイルを作成し、そのプロファイルで SSL インターセプトを有効にする必要があります。このプロファイルにはデフォルトの暗号グループがバインドされていますが、展開に合わせてさらに多くの暗号を設定できます。SSL インターセプション CA 証明書をこのプロファイルにバインドしてから、プロファイルをプロキシサーバーにバインドします。SSL インターセプトの場合、プロファイル内の重要なパラメータは以下のアクションに使用されるパラメータです。

- オリジンサーバー証明書の OCSP ステータスを確認します。
- オリジンサーバーが再ネゴシエーションを要求した場合、クライアントの再ネゴシエーションをトリガーします。
- フロントエンド SSL セッションを再利用する前に、オリジンサーバーの証明書を確認してください。

オリジンサーバーと通信するときは、デフォルトのバックエンドプロファイルを使用してください。デフォルトのバックエンドプロファイルで、暗号スイートなどのサーバー側パラメーターを設定します。カスタムバックエンドプロファイルはサポートされません。

最も一般的に使用される SSL 設定の例については、このセクションの最後にある「サンプルプロファイル」を参照してください。

暗号/プロトコルのサポートは、内部ネットワークと外部ネットワークによって異なります。次の表では、ユーザーと ADC アプライアンス間の接続は内部ネットワークです。外部ネットワークは、アプライアンスとインターネットの間にあります。

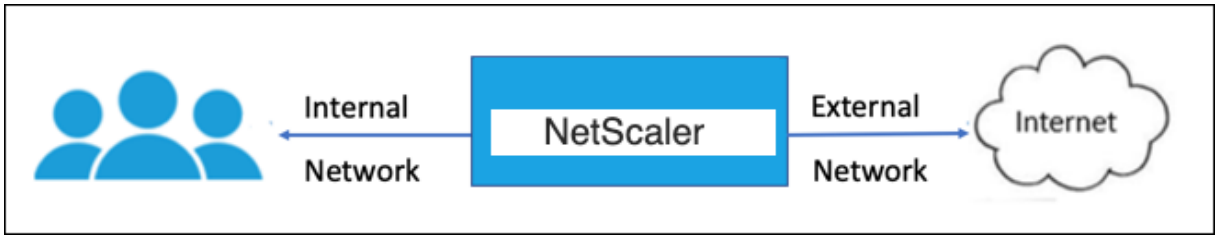


表 1: 内部ネットワークの暗号/プロトコルサポートマトリックス

表 1-NetScaler ADC アプライアンスで使用可能な Ciphers での仮想サーバー/フロントエンドサービス/内部サービスのサポートを参照してください。

表 2: 外部ネットワークの暗号/プロトコルサポートマトリックス

表 2-NetScaler ADC アプライアンスで使用可能な Ciphers でのバックエンドサービスのサポートを参照してください。

CLI を使用して **SSL** プロファイルを追加し、**SSL** インターセプションを有効にする

コマンドプロンプトで入力します。

```
add ssl profile <name> -sslinterception ENABLED -ssliReneg ( ENABLED | DISABLED )-ssliOCSPCheck ( ENABLED | DISABLED )-ssliMaxSessPerServer <positive_integer>
```

引数:

sslInterception:

SSL セッションのインターセプトを有効または無効にします。

設定可能な値: ENABLED, DISABLED

デフォルト値: 無効

ssliReneg:

オリジンサーバーから再ネゴシエーション要求を受信したときのクライアント再ネゴシエーションのトリガーを有効または無効にします。

設定可能な値: ENABLED, DISABLED

デフォルト値: 有効

ssliOCSPCheck:

オリジンサーバー証明書の OCSP チェックを有効または無効にします。

設定可能な値: ENABLED, DISABLED

デフォルト値: 有効

サーバあたりの **SSLI** の最大数**:**

ダイナミック・オリジン・サーバーごとにキャッシュされる SSL セッションの最大数。クライアント hello メッセージでクライアントから受信した SNI 拡張ごとに、固有の SSL セッションが作成されます。一致するセッションは、サーバーセッションの再利用に使用されます。

デフォルト値:10

最小値:1

最大値:1000

例:


```
add ssl profile swg_ssl_profile  -sslinterception ENABLED

Done

sh ssl profile swg_ssl_profile

1)    Name: swg_ssl_profile (Front-End)

                SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1
                   .1: ENABLED  TLSv1.2: ENABLED

                Client Auth: DISABLED

                Use only bound CA certificates: DISABLED

                Strict CA checks:                                NO

                Session Reuse: ENABLED
                   Timeout: 120 seconds

                DH: DISABLED

                DH Private-Key Exponent Size Limit: DISABLED
                   Ephemeral RSA: ENABLED
                   Refresh Count: 0

                Deny SSL Renegotiation
                   ALL

                Non FIPS Ciphers: DISABLED

                Cipher Redirect: DISABLED

                SSL Redirect: DISABLED

                Send Close-Notify: YES

                Strict Sig-Digest Check: DISABLED

                Push Encryption Trigger: Always

                PUSH encryption trigger timeout:             1 ms

                SNI: DISABLED

                OCSP Stapling: DISABLED

                Strict Host Header check for SNI enabled SSL sessions:
                                        NO

                Push flag:            0x0 (Auto)

                SSL quantum size:                            8 kB
```



```
48
49                  Encryption trigger timeout           100 mS
50
51                  Encryption trigger packet count:               45
52
53                  Subject/Issuer Name Insertion Format: Unicode
54
55                  SSL Interception: ENABLED
56
57                  SSL Interception OCSP Check: ENABLED
58
59                  SSL Interception End to End Renegotiation: ENABLED
60
61                  SSL Interception Server Cert Verification for Client
                       Reuse: ENABLED
62
63                  SSL Interception Maximum Reuse Sessions per Server:  10
64
65                  Session Ticket: DISABLED              Session Ticket
                       Lifetime: 300 (secs)
66
67                  HSTS: DISABLED
68
69                  HSTS IncludeSubDomains: NO
70
71                  HSTS Max-Age: 0
72
73                  ECC Curve: P_256, P_384, P_224, P_521
74
75  1)             Cipher Name: DEFAULT Priority :1
76
77                  Description: Predefined Cipher Alias
78
79  Done
80  <!--NeedCopy-->
```

CLI を使用して **SSL** インターセプション **CA** 証明書を **SSL** プロファイルにバインドする

コマンドプロンプトで入力します。

```
bind ssl profile <name> -ssliCACertkey <ssli-ca-cert>
```

例:

```
1  bind ssl profile swg_ssl_profile -ssliCACertkey swg_ca_cert
2
3  Done
4
5  sh ssl profile swg_ssl_profile
6
7  1)             Name: swg_ssl_profile (Front-End)
8
```



```
 9                  SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1
                        .1: ENABLED  TLSv1.2: ENABLED
10
11                  Client Auth: DISABLED
12
13                  Use only bound CA certificates: DISABLED
14
15                  Strict CA checks:                                NO
16
17                  Session Reuse: ENABLED
                        Timeout: 120 seconds
18
19                  DH: DISABLED
20
21                  DH Private-Key Exponent Size Limit: DISABLED
                        Ephemeral RSA: ENABLED
                        Refresh Count: 0
22
23                  Deny SSL Renegotiation
                        ALL
24
25                  Non FIPS Ciphers: DISABLED
26
27                  Cipher Redirect: DISABLED
28
29                  SSL Redirect: DISABLED
30
31                  Send Close-Notify: YES
32
33                  Strict Sig-Digest Check: DISABLED
34
35                  Push Encryption Trigger: Always
36
37                  PUSH encryption trigger timeout:             1 ms
38
39                  SNI: DISABLED
40
41                  OCSP Stapling: DISABLED
42
43                  Strict Host Header check for SNI enabled SSL sessions:
                                          NO
44
45                  Push flag:            0x0 (Auto)
46
47                  SSL quantum size:                            8 kB
48
49                  Encryption trigger timeout           100 mS
50
51                  Encryption trigger packet count:               45
52
53                  Subject/Issuer Name Insertion Format: Unicode
54
55                  SSL Interception: ENABLED
```



```
56
57                  SSL Interception OCSP Check: ENABLED
58
59                  SSL Interception End to End Renegotiation: ENABLED
60
61                  SSL Interception Server Cert Verification for Client
                       Reuse: ENABLED
62
63                  SSL Interception Maximum Reuse Sessions per Server:  10
64
65                  Session Ticket: DISABLED              Session Ticket
                       Lifetime: 300 (secs)
66
67                  HSTS: DISABLED
68
69                  HSTS IncludeSubDomains: NO
70
71                  HSTS Max-Age: 0
72
73                  ECC Curve: P_256, P_384, P_224, P_521
74
75  1)             Cipher Name: DEFAULT Priority :1
76
77                  Description: Predefined Cipher Alias
78
79  1)             SSL Interception CA CertKey Name: swg_ca_cert
80
81  Done
82  <!--NeedCopy-->
```

GUI を使用して **SSL** インターセプション **CA** 証明書を **SSL** プロファイルにバインドする

1. [システム] > [プロファイル] > [**SSL** プロファイル] に移動します。
2. [追加] をクリックします。
3. プロファイルの名前を指定します。
4. **SSL** セッションインターセプトを有効にします。
5. [**OK**] をクリックします。
6. [詳細設定] で、[証明書キー] をクリックします。
7. プロファイルにバインドする SSL インターセプト CA 証明書キーを指定します。
8. [選択] をクリックし、[バインド] をクリックします。
9. オプションで、導入環境に合わせて暗号を設定します。
   - 編集アイコンをクリックし、「追加」をクリックします。
   - 1 つまたは複数の暗号グループを選択し、右矢印をクリックします。

- [**OK**] をクリックします。

10. [完了] をクリックします。

GUI を使用して **SSL** プロファイルをプロキシサーバーにバインドする

1. [セキュリティ] > [**SSL** 転送プロキシ] > [プロキシ仮想サーバー] に移動し、サーバーを追加するか、変更するサーバーを選択します。
2. **SSL** プロファイルで、編集アイコンをクリックします。
3. **SSL** プロファイルリストで、以前に作成した SSL プロファイルを選択します。
4. [**OK**] をクリックします。
5. [完了] をクリックします。

サンプルプロファイル:

```
Name: swg_ssl_profile (Front-End)

                SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1
                    .1: ENABLED  TLSv1.2: ENABLED

                Client Auth: DISABLED

                Use only bound CA certificates: DISABLED

                Strict CA checks:                                 NO

                Session Reuse: ENABLED
                    Timeout: 120 seconds

                DH: DISABLED

                DH Private-Key Exponent Size Limit: DISABLED
                    Ephemeral RSA: ENABLED
                    Refresh Count: 0

                Deny SSL Renegotiation
                    ALL

                Non FIPS Ciphers: DISABLED

                Cipher Redirect: DISABLED

                SSL Redirect: DISABLED

                Send Close-Notify: YES

                Strict Sig-Digest Check: DISABLED

                Push Encryption Trigger: Always

```



```
31                  PUSH encryption trigger timeout:              1 ms
32
33                  SNI: DISABLED
34
35                  OCSP Stapling: DISABLED
36
37                  Strict Host Header check for SNI enabled SSL sessions:
                                NO
38
39                  Push flag:             0x0 (Auto)
40
41                  SSL quantum size:                             8 kB
42
43                  Encryption trigger timeout           100 mS
44
45                  Encryption trigger packet count:                45
46
47                  Subject/Issuer Name Insertion Format: Unicode
48
49                  SSL Interception: ENABLED
50
51                  SSL Interception OCSP Check: ENABLED
52
53                  SSL Interception End to End Renegotiation: ENABLED
54
55                  SSL Interception Maximum Reuse Sessions per Server:  10
56
57                  Session Ticket: DISABLED              Session Ticket
                       Lifetime: 300 (secs)
58
59                  HSTS: DISABLED
60
61                  HSTS IncludeSubDomains: NO
62
63                  HSTS Max-Age: 0
64
65                  ECC Curve: P_256, P_384, P_224, P_521
66
67  1)            Cipher Name: DEFAULT Priority :1
68
69                  Description: Predefined Cipher Alias
70
71  1)            SSL Interception CA CertKey Name: swg_ca_cert
72  <!--NeedCopy-->
```

ユーザー ID 管理

March 20, 2024

セキュリティ侵害の増加とモバイルデバイスの人気の高まりにより、外部インターネットの使用が企業ポリシーに準拠していることを確認する必要性が強調されています。企業担当者がプロビジョニングした外部リソースへのアクセスを許可されるのは、承認されたユーザーのみです。ID 管理は、個人またはデバイスの ID を検証することによって可能にします。これは、個人が取ることができるタスクや個人が参照できるファイルを決定するものではありません。

SSL フォワードプロキシ展開は、インターネットへのアクセスを許可する前にユーザを識別します。ユーザーからのすべての要求と応答が検査されます。ユーザーアクティビティはログに記録され、レコードはレポート用に NetScaler Application Delivery Management（ADM）にエクスポートされます。NetScaler Console では、ユーザーアクティビティ、トランザクション、および帯域幅消費に関する統計を表示できます。

デフォルトでは、ユーザの IP アドレスのみが保存されますが、ユーザの詳細を記録するように機能を設定できます。この ID 情報を使用して、特定のユーザーに対してより豊富なインターネット使用ポリシーを作成できます。

NetScaler アプライアンスは、明示的なプロキシ構成に対して次の認証モードをサポートしています。

- **Lightweight Directory Access Protocol (LDAP)**. 外部 LDAP 認証サーバーを介してユーザーを認証します。詳細については、「LDAP 認証ポリシー」を参照してください。
- **RADIUS**。外部 RADIUS サーバーを介してユーザを認証します。詳細については、「RADIUS 認証」を参照してください。
- **TACACS+**. 外部ターミナルアクセスコントローラアクセスコントロールシステム (TACACS) 認証サーバを介してユーザを認証します。詳細については、「TACACS 認証ポリシー」を参照してください。
- **Negotiate**. Kerberos 認証サーバを介してユーザを認証します。Kerberos 認証でエラーが発生した場合、アプライアンスは NTLM 認証を使用します。詳細については、「認証ポリシーの交渉」を参照してください。

透過プロキシの場合、IP ベースの LDAP 認証のみがサポートされます。クライアント要求を受信すると、プロキシは Active Directory 内のクライアント IP アドレスのエントリをチェックして、ユーザを認証します。次に、ユーザ IP アドレスに基づいてセッションを作成します。ただし、LDAP アクションで ssonameAttribute を設定すると、IP アドレスの代わりにユーザー名を使用してセッションが作成されます。従来のポリシーは、トランスペアレントプロキシセットアップでの認証ではサポートされていません。

> 注
>
> 明示的なプロキシの場合は、LDAP ログイン名を **samAccountName** に設定する必要があります。トランスペアレントプロキシの場合は、LDAP ログイン名を **networkAddress** に設定し、属性 1 を **sAMAccountName** に設定する必要があります。

明示的なプロキシの例:

```
1 add authentication ldapAction swg-auth-action-explicit -serverIP
      10.105.157.116 -ldapBase "CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDn "
      CN=Administrator,CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDnPassword
      freebsd123$ -ldapLoginName sAMAccountName
2 <!--NeedCopy-->
```

透過プロキシの例:


```
add authentication ldapAction swg-auth-action-explicit -serverIP
    10.105.157.116 -ldapBase "CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDn "
    CN=Administrator,CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDnPassword
    freebsd123$ -ldapLoginName networkAddress -authentication disable -
    Attribute1 sAMAccountName
<!--NeedCopy-->
```

CLI を使用してユーザ認証をセットアップする

コマンドプロンプトで次のように入力します:

```
add authentication vserver <vserver name> SSL

bind ssl vserver <vserver name> -certkeyName <certkey name>

add authentication ldapAction <action name> -serverIP <ip_addr> -
    ldapBase <string> -ldapBindDn <string> -ldapBindDnPassword -
    ldapLoginName <string>

add authentication Policy <policy name> -rule <expression> -action <
    string>

bind authentication vserver <vserver name> -policy <string> -priority <
    positive_integer>

set cs vserver <name> -authn401 ON -authnVsName <string>
<!--NeedCopy-->
```

引数:

Vserver name:

ポリシーをバインドする認証仮想サーバの名前。

最大長: 127

serviceType:

認証仮想サーバーのプロトコルタイプ。常に SSL。

可能な値:SSL

デフォルト値:SSL

Action name:

新しい LDAP アクションの名前。文字、数字、またはアンダースコア文字（_）で始まり、文字、数字、ハイフン（-）、ピリオド（.）ポンド（#）、スペース（）、アットマーク（@）、等号（=）、コロン（:）、およびアンダースコア文字のみを含める必要があります。LDAP アクションが追加された後は変更できません。次の要件は、CLI だけに適用されます。

名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます (「認証アクション」や「認証アクション」など)。

最大長：127

serverIP:

LDAP サーバに割り当てられた IP アドレス。

ldapBase:

LDAP 検索を開始するベース (ノード)。LDAP サーバーがローカルで実行されている場合、base のデフォルト値は dc =netscaler、dc=com. 最大長：127

ldapBindDn:

LDAP サーバーへのバインドに使用される完全識別名 (DN)。

デフォルト：cn=manager、dc=netscaler、dc=com

最大長：127

ldapBindDnPassword:

LDAP サーバーへのバインドに使用するパスワード。

最大長：127

ldapLoginName:

LDAP ログイン名属性。NetScaler アプライアンスは、LDAP ログイン名を使用して、外部 LDAP サーバーまたはアクティブディレクトリを照会します。最大長：127

Policy name:

高度な認証ポリシーの名前。文字、数字、またはアンダースコア文字（_）で始まり、文字、数字、ハイフン（-)、ピリオド（.）ポンド（#)、スペース ()、アットマーク（@)、等号（=)、コロン（:)、およびアンダースコア文字のみを含める必要があります。認証ポリシーの作成後は変更できません。次の要件は、CLI だけに適用されます。

名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます (「認証ポリシー」や「認証ポリシー」など)。

最大長：127

rule:

AUTHENTICATION サーバーでユーザーを認証するかどうかを決定するためにポリシーが使用するルールの名前、または高度なポリシー式。

最大長：1499

action:

ポリシーが一致した場合に実行される認証アクションの名前。

最大長：127

priority:

ポリシーのプライオリティを指定する正の整数。数字が小さいほど、プライオリティが高くなります。ポリシーは、優先順位の順に評価され、リクエストに一致する最初のポリシーが適用されます。認証仮想サーバにバインドされたポリシーのリスト内で一意である必要があります。

最小値:0

最大値:4294967295

例:

```
add authentication vserver swg-auth-vs SSL

Done

bind ssl vserver explicit-auth-vs -certkeyName ns-swg-ca-certkey

Done

add authentication ldapAction swg-auth-action-explicit -serverIP
    192.0.2.116 -ldapBase "CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDn "CN=
    Administrator,CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDnPassword zzzzz
     -ldapLoginName sAMAccountName

Done

add authenticationpolicy swg-auth-policy -rule true -action swg-auth-
    action-explicit
Done

bind authentication vserver swg-auth-vs -policy swg-auth-policy -
    priority 1

Done

set cs vserver testswg -authn401 ON -authnVsName swg-auth-vs

Done
<!--NeedCopy-->
```

CLI を使用してユーザー名ロギングを有効にする

コマンドプロンプトで入力します:

```
set appflow param -AAAUserName ENABLED
<!--NeedCopy-->
```

引数:

AAAAUserName

AppFlow 認証、承認、および監査ユーザー名ロギングを有効にします。

設定可能な値：ENABLED, DISABLED

デフォルト値: 無効

例:

```
set appflow param -AAAUserName ENABLED
<!--NeedCopy-->
```

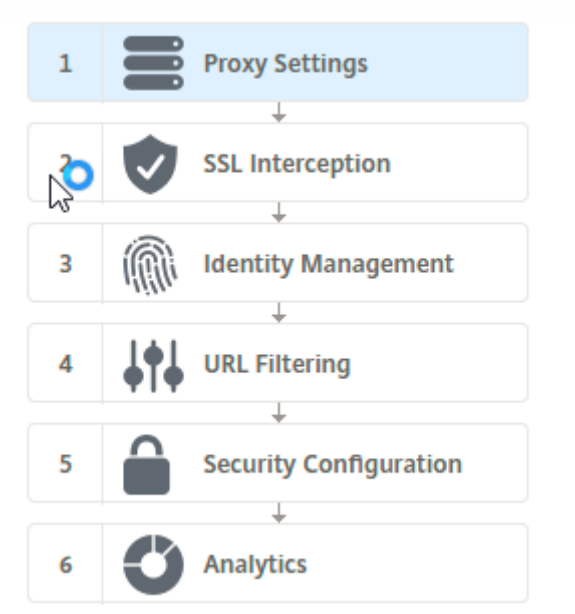
URL フィルタリング

August 15, 2023

URL フィルタリングは、URL に含まれる情報を使用して Web サイトをポリシーベースで制御します。この機能は、ネットワーク管理者がネットワーク上の悪意のある Web サイトへのユーザーアクセスを監視および制御するのに役立ちます。

開始

新規ユーザーで URL フィルタリングを設定する場合は、SSL 転送プロキシの初期設定を完了する必要があります。URL フィルタリングを開始するには、まず SSL 転送プロキシウィザードにログオンする必要があります。ウィザードでは、URL フィルタリングポリシーを適用する前に、一連の構成手順を実行します。



> 注
>
> 開始する前に、有効な URL 脅威インテリジェンス機能ライセンスがアプライアンスにインストールされていることを確認してください。試用版を使用している場合は、ADC アプライアンスでこの機能を引き続き使用するには、必ず有効なライセンスを購入してください。

SSL 転送プロキシウィザードへのログオン

SSL 転送プロキシウィザードの指示に従って一連の簡単な設定タスクを実行すると、右側のペインには対応するフローシーケンスが表示されます。このウィザードを使用して、URL フィルタリングポリシーを URL リストまたは定義済みのカテゴリの一覧に適用できます。

手順 **1:** プロキシ設定を構成する

まず、クライアントがゲートウェイにアクセスするためのプロキシサーバーを設定します。このサーバは SSL タイプであり、明示的または透過的モードで動作します。プロキシサーバー構成の詳細については、「プロキシモード」を参照してください。

ステップ **2: SSL** インターセプションを設定する

プロキシサーバーを構成したら、NetScaler アプライアンスで暗号化されたトラフィックをインターセプトするように SSL インターセプトプロキシを構成する必要があります。URL フィルタリングの場合、SSL プロキシはトラフィックをインターセプトし、ブロックされた URL を許可しませんが、他のすべてのトラフィックはバイパスできます。SSL 代行受信の設定の詳細については、「SSL インターセプション」を参照してください。

ステップ **3: ID** 管理を構成する

ユーザーは、企業ネットワークへのログオンを許可される前に認証されます。認証では、役割に基づいて、ユーザーまたはユーザーのグループに対して特定のポリシーを柔軟に定義できます。ユーザー認証の詳細については、「ユーザー識別管理」を参照してください。

手順 **4: URL** フィルタリングを構成する

管理者は、URL 分類機能または URL リスト機能を使用して、URL フィルタリングポリシーを適用できます。

URL 分類。事前定義されたカテゴリのリストに基づいてトラフィックをフィルタリングすることにより、Web サイトおよび Web ページへのアクセスを制御します。

URL リスト。アプライアンスにインポートされた URL セットに含まれる URL へのアクセスを拒否することにより、ブラックリストに登録された Web サイトおよび Web ページへのアクセスを制御します。

ステップ **5:** セキュリティー構成を設定する

このステップでは、レピュテーションスコアを設定し、スコアが低すぎる場合はアクセスを拒否してユーザーがウェブサイトへのアクセスを制御できるようにします。レピュテーションスコアの範囲は 1 ～4 で、スコアが許容範囲外になるしきい値を設定できます。しきい値を超えるスコアの場合は、トラフィックを許可する、ブロック、またはリダイレクトするポリシーアクションを選択できます。詳細については、URL レピュテーションスコアを参照してください。

ステップ **6: SSL** 転送プロキシ分析を構成する

この手順では、Web トラフィックの分類、ユーザトランザクションログの URL カテゴリのロギング、トラフィック分析の表示のために SSL 転送プロキシ分析を有効にできます。SSL フォワードプロキシ分析の詳細については、「Analytics」を参照してください。

ステップ **7: [**完了**]** をクリックして初期構成を完了し、**URL** フィルタリング構成の管理を続行します

URL リスト

December 8, 2023

URL リスト機能により、企業のお客様は特定の Web サイトや Web サイトカテゴリへのアクセスを制御できます。この機能は、URL マッチングアルゴリズムにバインドされたレスポンダーポリシーを適用して Web サイトをフィルタリングします。このアルゴリズムは、受信 URL を最大 100 万（1,000,000）のエントリで構成される URL セットと照合します。受信 URL リクエストがセット内のエントリと一致する場合、アプライアンスはレスポンダーポリシーを使用してリクエスト（HTTP/HTTPS）を評価し、そのリクエストへのアクセスを制御します。

URL セットタイプ

URL セットの各エントリには、URL と、オプションでそのメタデータ (URL カテゴリ、カテゴリグループ、またはその他の関連データ) を含めることができます。メタデータを含む URL の場合、アプライアンスはメタデータを評価するポリシー式を使用します。詳細については、「URL セット」を参照してください。

SSL フォワードプロキシはカスタム URL セットをサポートします。パターンセットを使用して URL をフィルタリングすることもできます。

カスタム **URL** セット。最大 1,000,000 の URL エントリを含むカスタマイズされた URL セットを作成し、それをテキストファイルとしてアプライアンスにインポートできます。

パターンセット。ADC アプライアンスは、Web サイトへのアクセスを許可する前に、パターンセットを使用して URL をフィルタリングできます。パターンセットは、着信 URL と最大 5000 エントリの間で完全に一致する文字列を検索する文字列マッチングアルゴリズムです。詳細については、パターンセットを参照してください。

読み込んだ URL セットの各 URL には、URL メタデータの形式でカスタムカテゴリを設定できます。組織はセットをホストし、手動で操作しなくてもセットを定期的に更新するように ADC アプライアンスを構成できます。

セットが更新されると、NetScaler アプライアンスはメタデータを自動的に検出します。このカテゴリは、URL を評価したり、許可、ブロック、リダイレクト、ユーザーへの通知などのアクションを適用したりするためのポリシー表現として使用できるようになりました。

URL セットで使用される高度なポリシー表現

次の表に、着信トラフィックの評価に使用できる基本的な式を示します。

1. .URLSET_MATCHES_ANY-URL が URL セット内のいずれかのエントリと完全に一致する場合、TRUE と評価されます。
2. .GET_URLSET_METADATA ()-GET_URLSET_METADATA () 式は、URL が URL セット内のいずれかのパターンと完全に一致する場合、関連するメタデータを返します。一致しない場合は、空の文字列が返されます。
3. `.GET_ URLSET_METADATA().EQ(<METADATA)- .GET_ URLSET_METADATA().EQ(<METADATA)`
4. .GET_URLSET_METADATA () .TYPECAST_LIST_T (‘,’) .GET (0) .EQ ()-一致するメタデータがカテゴリの先頭にある場合に TRUE と評価されます。このパターンは、メタデータ内の個別のフィールドをエンコードするために使用できますが、最初のフィールドのみに一致します。
5. HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL)-ホストパラメータと URL パラメータを結合して、マッチングに使用できるようにします。

レスポンダーのアクションの種類

> 注**:** この表では、HTTP.REQ.URL は次のように一般化されています。`<URL expression>`

次の表に、着信インターネットトラフィックに適用できるアクションを示します。

レスポンダーアクション	説明
許可	リクエストがターゲット URL にアクセスすることを許可します。
リダイレクト	ターゲットとして指定された URL にリクエストをリダイレクトします。
ブロック	リクエストを拒否します。

前提条件

ホスト名 URL から URL セットをインポートする場合は、DNS サーバーを設定します。IP アドレスを使用する場合、この設定は不要です。

コマンドプロンプトで入力します:

```
add dns nameServer ((<IP> [-local])| <dnsVserverName>)[-state (
ENABLED | DISABLED )] [-type <type>] [-dnsProfileName <string>]
```

例:

```
add dns nameServer 10.140.50.5
```

URL リストを構成する

URL リストを構成するには、Citrix SSL 転送プロキシウィザードまたは NetScaler コマンドラインインターフェイス（CLI）を使用できます。NetScaler アプライアンスでは、まずレスポンダーポリシーを構成してから、ポリシーを URL セットにバインドする必要があります。

Citrix では、URL リストを構成する推奨オプションとして Citrix SSL 転送プロキシウィザードを使用することをお勧めします。ウィザードを使用して、レスポンダーポリシーを URL セットにバインドします。または、ポリシーをパターンセットにバインドすることもできます。

SSL 転送プロキシウィザードを使用して **URL** リストを設定する

GUI を使用して HTTPS トラフィックの URL リストを設定するには:

1. [セキュリティ] > [**SSL** 転送プロキシ] ページに移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います:
   a)「**SSL** 転送プロキシウィザード」をクリックします。
   b) 既存の設定を選択し、[**Edit**] をクリックします。
3. [**URL** フィルタ] セクションで、[編集] をクリックします。
4.「**URL** リスト」チェックボックスを選択して機能を有効にします。
5. **URL** リストポリシーを選択し、「バインド」をクリックします。
6. [続行] をクリックし、[完了] をクリックします。

詳細については、「URL リストポリシーを作成する方法」を参照してください。

CLI を使用した **URL** リストの設定

URL リストを設定するには、次の手順を実行します。

1. HTTP および HTTPS トラフィック用のプロキシ仮想サーバーを構成します。
2. HTTPS トラフィックを傍受するための SSL インターセプトを設定します。
3. HTTP トラフィック用の URL セットを含む URL リストを設定します。
4. HTTPS トラフィック用に設定された URL を含む URL リストを設定します。

5. プライベート URL セットを設定します。

> 注
>
> ADC アプライアンスをすでに構成している場合は、手順 1 と 2 をスキップして、手順 3 で設定できます。

インターネットトラフィック用のプロキシ仮想サーバーの構成　NetScaler アプライアンスは、透過的で明示的なプロキシ仮想サーバーをサポートします。エクスプリシットモードでインターネットトラフィック用のプロキシ仮想サーバーを構成するには、次の手順を実行します。

1. プロキシ SSL 仮想サーバーを追加します。
2. レスポンダーポリシーをプロキシ仮想サーバーにバインドします。

CLI を使用してプロキシ仮想サーバーを追加するには:

コマンドプロンプトで入力します:

```
add cs vserver <name> <serviceType> <IPAddress> <port>
<!--NeedCopy-->
```

例:

```
add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180
<!--NeedCopy-->
```

CLI を使用してレスポンダーポリシーをプロキシ仮想サーバーにバインドするには:

```
bind ssl vserver <vServerName> -policyName <string> [-priority <
    positive_integer>]
<!--NeedCopy-->
```

> 注
>
> NetScaler 構成の一部として SSL インターセプターをすでに構成している場合は、次の手順をスキップできます。

HTTPS トラフィックの **SSL** インターセプトの設定　HTTPS トラフィックの SSL インターセプトを設定するには、次の手順を実行します。

1. CA 証明書とキーのペアをプロキシ仮想サーバにバインドします。
2. デフォルトの SSL プロファイルを有効にします。
3. フロントエンド SSL プロファイルを作成してプロキシ仮想サーバーにバインドし、フロントエンド SSL プロファイルで SSL インターセプトを有効にします。

CLI を使用して CA 証明書とキーのペアをプロキシ仮想サーバーにバインドするには:

コマンドプロンプトで入力します:


```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
<!--NeedCopy-->
```

CLI を使用してフロントエンド SSL プロファイルを設定するには:

コマンドプロンプトで入力します:

```
set ssl parameter -defaultProfile ENABLED

add ssl profile <name> -sslInterception ENABLED -ssliMaxSessPerServer <
   positive_integer>
<!--NeedCopy-->
```

CLI を使用してフロントエンド SSL プロファイルをプロキシ仮想サーバーにバインドするには

コマンドプロンプトで入力します:

```
set ssl vserver <vServer name>  -sslProfile <name>
<!--NeedCopy-->
```

HTTP トラフィックの **URL** セットをインポートして **URL** リストを構成する　HTTP トラフィックの URL セットを設定する方法については、「URL セット」を参照してください。

明示的なサブドメイン一致の実行　インポートされた URL セットに対して明示的なサブドメイン一致を実行できるようになりました。新しいパラメーター「SubdomainExactMatch」がコマンドに追加されました。

```
import policy URLset
```

パラメーターを有効にすると、URL フィルタリングアルゴリズムは明示的なサブドメイン一致を実行します。たとえば、受信 URL が`news.example.com`で、URL セットのエントリが`example.com`の場合、アルゴリズムは URL と一致しません。

コマンドプロンプトで入力します:

```
import policy urlset <name> [-overwrite] [-delimiter <character>][-
rowSeparator <character>] -url [-interval <secs>] [-privateSet][-
subdomainExactMatch] [-canaryUrl <URL>]
```

例

```
import policy urlset test -url http://10.78.79.80/top-1k.csv -privateSet
 -subdomainExactMatch -interval 900
```

HTTPS トラフィック用の **URL** セットを構成する　CLI を使用して HTTPS トラフィック用の URL セットを設定するには

コマンドプロンプトで次のように入力します:


```
add ssl policy <name> -rule <expression> -action <string> [-undefAction
    <string>] [-comment <string>]
<!--NeedCopy-->
```

例:

```
add ssl policy pol1 -rule client.ssl.client_hello.SNI.
   URLSET_MATCHES_ANY("top1m") -action INTERCEPT
<!--NeedCopy-->
```

SSL 転送プロキシウィザードを使用して **HTTPS** トラフィックの **URL** セットを構成するには　Citrix では、URL リストを構成する推奨オプションとして SSL 転送プロキシウィザードを使用することをお勧めします。ウィザードを使用して、カスタム URL セットをインポートし、レスポンダーポリシーにバインドします。

1. [セキュリティ] **>** [**SSL** 転送プロキシ] **>** [**URL** フィルタ] **>** [**URL** リスト] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [**URL** リストポリシー] ページで、ポリシー名を指定します。
4. URL セットをインポートするオプションを選択します。
5.「**URL** リストポリシー」タブページで、「**URL** セットのインポート」チェックボックスを選択し、次の URL セットパラメータを指定します。

   a) URL セット名-カスタム URL セットの名前。
   b) URL —URL セットにアクセスする場所の Web アドレス。
   c) [上書き]-以前にインポートした URL セットを上書きします。
   d) 区切り文字—CSV ファイルレコードを区切る文字シーケンス。
   e) 行セパレーター—CSV ファイルで使用される行セパレータ。
   f) 間隔：URL セットが更新されるまでの間隔（秒単位）。15 分に相当する最も近い秒数に四捨五入されます。
   g) プライベートセット-URL セットをエクスポートしないようにするオプション。
   h) Canary URL —URL セットのコンテンツを秘密にしておくべきかどうかをテストするための内部 URL。URL の最大長は 2047 文字です。

6. ドロップダウンリストからレスポンダーアクションを選択します。
7. **[**作成**]** して **[**閉じる**]** をクリックします。

プライベート **URL** セットの設定　プライベート URL セットを設定し、その内容を秘密にしておくと、ネットワーク管理者はセット内のブラックリストに登録されている URL を知らない可能性があります。このような場合は、Canary URL を設定して URL セットに追加できます。管理者は Canary URL を使用して、すべての検索リクエストに使用するプライベート URL セットをリクエストできます。各パラメータの説明については、ウィザードのセクションを参照してください。

CLI を使用して URL セットをインポートするには:

コマンドプロンプトで入力します:

```
import policy urlset <name> [-overwrite] [-delimiter <character>] [-
   rowSeparator <character>] -url <URL> [-interval <secs>] [-privateSet
   ] [-canaryUrl <URL>]
<!--NeedCopy-->
```

例:

```
import policy urlset test1 -url http://10.78.79.80/alytra/top-1k.csv -
   private -canaryUrl http://www.in.gr
<!--NeedCopy-->
```

インポートした **URL** セットを表示

追加された URL セットに加えて、読み込んだ URL セットも表示できるようになりました。新しいパラメーター「import」`show urlset` がコマンドに追加されます。このオプションを有効にすると、アプライアンスはインポートされたすべての URL セットを表示し、インポートされた URL セットが追加された URL セットと区別されます。

コマンドプロンプトで入力します:

```
show policy urlset [<name>] [-imported]
```

例

```
show policy urlset -imported
```

監査ログメッセージを設定する

監査ログを使用すると、URL リストプロセスのどの段階でも条件や状況を確認できます。NetScaler アプライアンスが受信 URL を受信すると、レスポンダーポリシーに URL セットの高度なポリシー表現が含まれている場合、監査ログ機能は URL の URL セット情報を収集します。監査ログで許可されている任意のターゲットの詳細をログメッセージとして保存します。

ログメッセージには次の情報が含まれます。

1. タイムスタンプ。
2. ログメッセージタイプ。
3. 定義済みのログレベル (重大、エラー、通知、警告、情報、デバッグ、アラート、緊急)。
4. URL セット名、ポリシーアクション、URL などのメッセージ情報をログに記録します。

URL リスト機能の監査ログを設定するには、次の作業を完了する必要があります。

1. 監査ログを有効化。
2. 「監査ログの作成」メッセージアクション。
3. [監査ログメッセージ] アクションで URL リストレスポンダポリシーを設定します。

詳細については、「監査ログ」トピックを参照してください。

URL パターンのセマンティクス

August 15, 2023

次の表に、フィルタリングするページのリストを指定するために使用する URL パターンを示します。たとえば、www.example.com/bar というパターンは www.example.com/bar の 1 ページのみに一致します。URL が 'www.example.com/bar' で始まるすべてのページに一致させるには、URL の末尾にアスタリスク (*) を追加します。

メタデータマッピングと一致する URL パターンのセマンティクス

パターンマッチングセマンティクスは、テーブル形式で使用できます。詳細については、パターンセマンティクス PDF ページを参照してください。

URL カテゴリのマッピング

August 15, 2023

サードパーティのカテゴリとカテゴリグループのリスト。詳細については、「URL カテゴリマッピング」ページを参照してください。

ユースケース: カスタム URL セットを使用した URL フィルタリング

August 15, 2023

特定の Web サイトや Web サイトカテゴリへのアクセスを制御したいと考えている企業のお客様は、レスポンダーポリシーにバインドされたカスタム URL セットを使用します。組織のネットワークインフラストラクチャでは、URL フィルタを使用して、悪意のある Web サイトや危険な Web サイトへのアクセスをブロックできます。たとえば、成人向け、暴力、ゲーム、麻薬、政治、求人ポータルを取り上げた Web サイトなどです。URL のフィルタリングに加えて、カスタマイズされた URL のリストを作成して ADC アプライアンスにインポートできます。たとえば、組織のポリシーで、ソーシャルネットワーキング、ショッピングポータル、求人ポータルなどの特定の Web サイトへのアクセスをブロックするよう求められる場合があります。

リスト内の各 URL は、メタデータの形式でカスタムカテゴリを持つことができます。組織は、NetScaler ADC アプライアンス上で URL セットとして URL のリストをホストできます。手動で操作しなくても、セットを定期的に更新するようにアプライアンスを設定します。

セットが更新されると、NetScaler ADC アプライアンスはメタデータを自動的に検出します。レスポンダーポリシーは URL メタデータ (カテゴリの詳細) を使用して受信 URL を評価し、許可、ブロック、リダイレクト、ユーザーへの通知などのアクションを適用します。

そのためには、ネットワークで設定し、次のタスクを実行できます。

1. カスタム URL セットをインポートする
2. カスタム URL セットの追加
3. SSL Forward Proxy ウィザードでカスタム URL リストを設定します。

CLI を使用してカスタム URL セットをインポートする

コマンドプロンプトで入力します。

```
import policy urlset <name> [-overwrite] [-delimiter <character>] [-
   rowSeparator <character>] -url <URL> [-interval <secs>] [-privateSet
   ] [-canaryUrl <URL>]

import policy urlset test1  - url http://10.78.79.80/alytra/top-1k.csv
<!--NeedCopy-->
```

CLI を使用してカスタム URL セットを追加する

コマンドプロンプトで入力します。

```
add urlset <urlset_name>
```

例:

```
add urlset test1
```

SSL Forward Proxy ウィザードを使用して URL リストを構成する

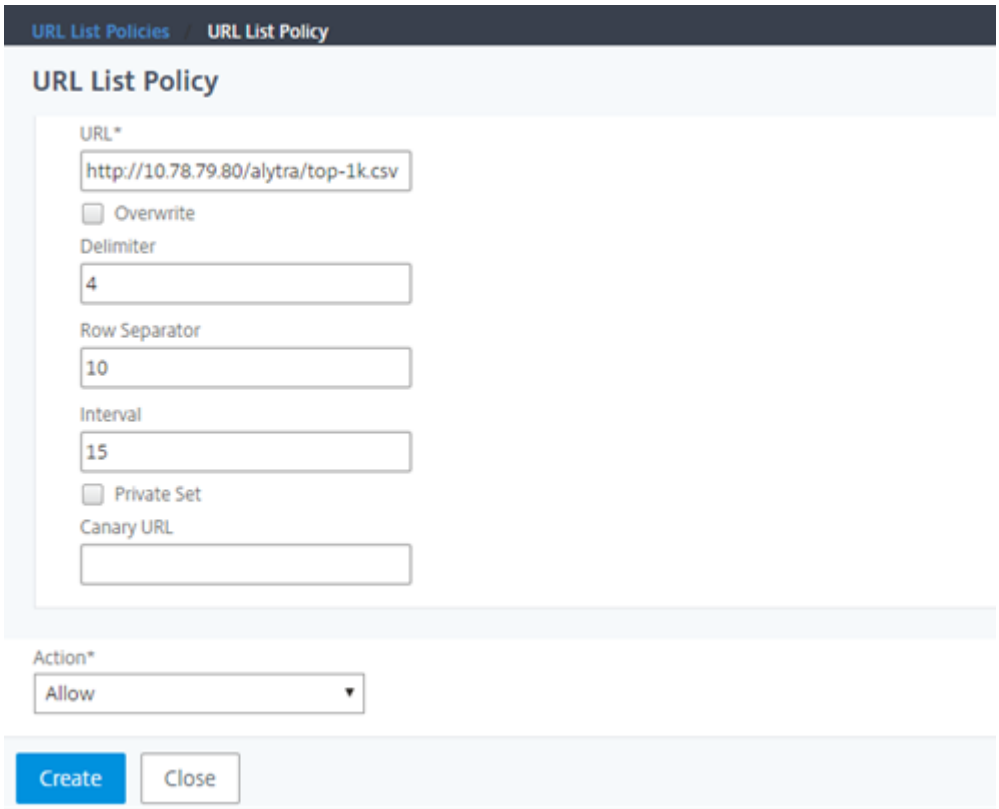
URL リストを構成するには、SSL 転送プロキシウィザードを優先オプションとして使用することをお勧めします。ウィザードを使用して、カスタム URL セットをインポートし、レスポンダーポリシーにバインドします。

1. セキュリティ **> SSL** 転送プロキシ **> URL** フィルタリング **> URL** リストに移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [**URL** リストポリシー] ページで、ポリシー名を指定します。
4. URL セットをインポートするオプションを選択します。

5. [**URL** リストポリシー] タブページで、[**URL** セットのインポート] チェックボックスをオンにし、次の URL セットパラメータを指定します。

   a) URL セット名-カスタム URL セットの名前。
   b) URL —URL セットにアクセスする場所の Web アドレス。
   c) [上書き]-以前にインポートした URL セットを上書きします。
   d) 区切り文字—CSV ファイルレコードを区切る文字シーケンス。
   e) 行セパレーター—CSV ファイルで使用される行セパレータ。
   f) 間隔：URL セットが更新される間隔（秒単位）。15 分未満を四捨五入します。
   g) プライベートセット-URL セットをエクスポートしないようにするオプション。
   h) Canary URL—URL セットの内容を秘密にするかどうかをテストするための内部 URL。URL の最大長は 2047 文字です。

6. ドロップダウンリストからレスポンダーアクションを選択します。
7. **[**作成**]** して **[**閉じる**]** をクリックします。



カスタム **URL** セットのメタデータセマンティクス

カスタム URL セットをインポートするには、URL をテキストファイルに追加し、ソーシャルネットワーキング URL をブロックするレスポンダーポリシーにバインドします。

テキストファイルに追加できる URL の例を次に示します。

cnn.com, ニュース

bbc.com, ニュース

google.com、検索エンジン

yahoo.com, 検索エンジン

facebook.com, ソーシャルメディア

twitter.com, ソーシャルメディア

CLI を使用してソーシャルメディアの **URL** をブロックするレスポンダーポリシーを設定する

```
add responder action act_url_unauthorized respondwith '"HTTP/1.1 451
    Unavailable For Legal Reasons\r\n\r\nURL is NOT authorized\n"'

add responder policy pol_url_meta_match 'HTTP.REQ.HOSTNAME.APPEND(HTTP.
    REQ.URL).GET_URLSET_METADATA("u1").EQ("Social Media")'
    act_url_unauthorized
<!--NeedCopy-->
```

URL の分類

March 20, 2024

URL の分類は、特定の Web サイトおよび Web サイトのカテゴリへのユーザーアクセスを制限します。NetSTARと連携したサブスクライブサービスであるこの機能により、企業のお客様は、市販の分類データベースを使用して Web トラフィックをフィルタリングできます。NetSTARデータベースには、ソーシャルネットワーキング、ギャンブル、アダルトコンテンツ、ニューメディア、ショッピングなど、さまざまなカテゴリに分類された膨大な数（数十億）の URL があります。分類に加えて、各 URL には、サイトの履歴リスクプロファイルに基づいて最新のレピュテーションスコアが保持されます。カテゴリ、カテゴリグループ（テロ、違法薬物など）、またはサイトレピュテーションスコアに基づいて高度なポリシーを設定することで、NetSTARデータを使用してトラフィックをフィルタリングできます。

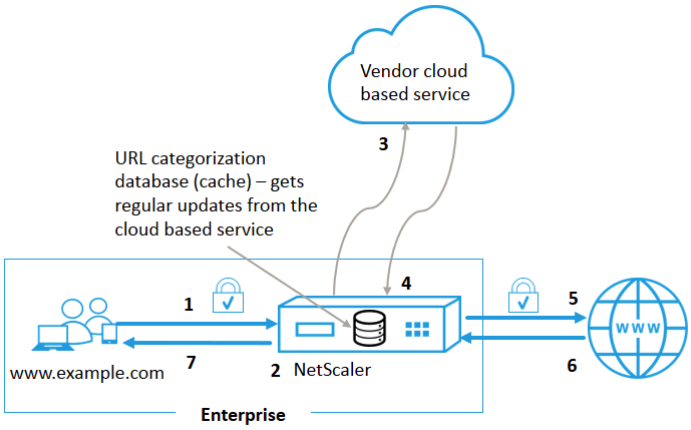
たとえば、マルウェアに感染していることがわかっているサイトなど、危険なサイトへのアクセスをブロックできます。また、エンタープライズユーザーに対して、アダルトコンテンツやエンターテイメントストリーミングメディアなどのコンテンツへのアクセスを選択的に制限することもできます。また、NetScaler Console サーバー上の Web トラフィック分析を監視するために、ユーザーのトランザクションの詳細とアウトバウンドトラフィックの詳細をキャプチャすることもできます。

NetScalerは、事前構成されたNetSTARデバイスnsv10.netstar-inc.comおよびincompasshybridpc.netstar-inc.comからデータをアップロードまたはダウンロードし、クラウド分類要求のデフォルトでクラウドホストとして使用されます。URL フィルタリングが正しく機能するには、これらの URL にファイアウォールを

介してアクセス可能である必要があります。アプライアンスは、NSIP アドレスを送信元 IP アドレスとして使用し、443 を通信の宛先ポートとして使用します。

URL 分類の仕組み

次の図は、NetScaler URL 分類サービスが商用 URL 分類データベースおよびクラウドサービスと統合され、頻繁に更新される様子を示しています。



コンポーネントは次のように相互作用します。

1. クライアントは、インターネットにバインドされた URL 要求を送信します。
2. SSL フォワードプロキシは、カテゴリ、カテゴリグループ、サイトレピュテーションスコアなどのカテゴリの詳細に基づいて、要求にポリシー適用を適用します。カテゴリの詳細は URL 分類データベースから取得されます。データベースがカテゴリの詳細を返す場合、プロセスは手順 5 にジャンプします。
3. データベースで分類の詳細が見つからない場合、URL 分類ベンダーが管理するクラウドベースの検索サービスにリクエストが送信されます。ただし、アプライアンスは応答を待たずに、URL が未分類としてマークされ、ポリシーの強制が実行されます（ステップ 5 にジャンプします）。アプライアンスは、クラウドクエリーフィードバックを監視し続け、キャッシュを更新して、今後の要求がクラウドルックアップの恩恵を受けることができるようにします。
4. ADC アプライアンスは、クラウドベースのサービスから URL カテゴリの詳細（カテゴリ、カテゴリグループ、レピュテーションスコア）を受信し、分類データベースに保存します。
5. ポリシーでは URL が許可され、リクエストはオリジンサーバーに送信されます。それ以外の場合、アプライアンスはカスタム HTML ページを使用してドロップ、リダイレクト、または応答します。
6. オリジンサーバーは、要求されたデータで ADC アプライアンスに応答します。
7. アプライアンスは応答をクライアントに送信します。

ユースケース**:** 企業コンプライアンス下での企業向けインターネット利用

URL フィルタリング機能を使用して、コンプライアンスポリシーを検出して実装し、企業のコンプライアンスに違反するサイトをブロックできます。たとえば、アダルトサイト、ストリーミングメディア、ソーシャルネットワーキングなど、非生産的と見なされるサイトや、企業ネットワークで過剰なインターネット帯域幅を消費するサイト。これらの Web サイトへのアクセスをブロックすると、従業員の生産性が向上し、帯域幅使用の運用コストが削減され、ネットワーク消費のオーバーヘッドが削減されます。

前提条件

URL 分類機能は、URL フィルタリング機能と SSL フォワードプロキシの脅威インテリジェンスを備えたオプションのサブスクリプションサービスがある場合にのみ、NetScaler プラットフォームで機能します。サブスクリプションにより、顧客は Web サイトの最新の脅威分類をダウンロードし、そのカテゴリを SSL フォワードプロキシに適用できます。この機能を有効にして設定する前に、次のライセンスをインストールする必要があります。

- CNS_WEBF_SSERVER_Retail.lic
- CNS_XXXX_SERVER_PLT_Retail.lic

ここで、XXXXX はプラットフォーム・タイプです（例：V25000）。

レスポンダーポリシー式

次の表に、着信 URL を許可、リダイレクト、またはブロックする必要があるかどうかを確認するために使用できるさまざまなポリシー式を示します。

1. <text>. URL_CATEGORIZE (<min_reputation>, <max_reputation>) - URL_CATEGORY オブジェクトを返します。<min_reputation>が 0 より大きい場合、返されるオブジェクトには<min_reputation>以下の評価を持つカテゴリは含まれません。<max_reputation>が 0 より大きい場合、返されるオブジェクトには、<max_reputation>より高い評価を持つカテゴリは含まれません。カテゴリがタイムリーに解決に失敗した場合は、undef 値が返されます。
2. <url_category>. CATEGORY() -このオブジェクトのカテゴリ文字列を返します。URL にカテゴリがない場合、または URL の形式が正しくない場合、戻り値は「Unknown」になります。
3. <url_category>. CATEGORY_GROUP() -オブジェクトのカテゴリグループを識別する文字列を返します。このグループ化は、カテゴリの上位レベルのグループ化であり、URL カテゴリに関する詳細情報が少ない操作で役立ちます。URL にカテゴリがない場合、または URL の形式が正しくない場合、戻り値は「Unknown」になります。
4. <url_category>. REPUTATION() -レピュテーションスコアを 0 ～5 の数値として返します。5 は最もリスクの高いレピュテーションを示します。カテゴリ「不明」がある場合、評価値は 1 です。

ポリシータイプ**:**

1. 検索エンジンのカテゴリに含まれる URL のリクエストを選択するポリシー- `add responder policy p1 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY().EQ( "Search Engine" )`
2. Adult カテゴリグループに含まれる URL のリクエストを選択するポリシー- `add responder policy p1 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY_GROUP().EQ( "Adult" )'`
3. レピュテーションスコアが 4 未満の検索エンジン URL のリクエストを選択するポリシー- `add responder policy p2 'HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL). URL_CATEGORIZE(4,0).HAS_CATEGORY( "Search Engine" )`
4. 検索エンジンとショッピング URL のリクエストを選択するポリシー- `add responder policy p3 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY().EQ ( "good_categories" )`
5. レピュテーションスコアが 4 以上の検索エンジン URL のリクエストを選択するポリシー- `add responder policy p5 'CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE (4,0). CATEGORY().EQ( "Search Engines" )`
6. 検索エンジンのカテゴリにある URL のリクエストを選択し、それらを URL セットと比較するポリシー- `'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY().EQ( "Search Engine" )&& HTTP.REQ.HOSTNAME.APPEND(HTTP. REQ.URL).URLSET_MATCHES_ANY( "u1" )'`

レスポンダーポリシーの種類

URL 分類機能で使用されるポリシーには 2 種類あり、次の表ではこれらのポリシータイプについて説明します。

ポリシーの種類	説明
URL カテゴリ	Web トラフィックを分類し、評価結果に基づいてトラフィックをブロック、許可、またはリダイレクトします。
URL レピュテーションスコア	Web サイトのレピュテーションスコアを決定し、管理者が設定したレピュテーションスコアのしきい値レベルに基づいてアクセスを制御できるようにします。

URL 分類の構成

NetScaler アプライアンスで URL 分類を構成するには、次の手順を実行します。

1. URL フィルタリングを有効にします。
2. Web トラフィック用にプロキシサーバーを構成します。
3. 明示モードで Web トラフィックの SSL インターセプトを設定します。
4. 共有メモリを設定してキャッシュメモリを制限します。

5. URL 分類パラメータを設定します。
6. Citrix SSL フォワードプロキシウィザードを使用して URL 分類を構成します。
7. SSL フォワードプロキシウィザードを使用して URL 分類パラメータを設定します。
8. シードデータベースパスとクラウドサーバー名の設定

ステップ **1: URL** フィルタリングの有効化

URL 分類を有効にするには、URL フィルタリング機能を有効にし、URL 分類のモードを有効にします。

CLI を使用して URL 分類を有効にするには

コマンドプロンプトで入力します:

```
enable ns feature URLFiltering
```

```
disable ns feature URLFiltering
```

手順 **2:** 明示モードで **Web** トラフィック用のプロキシサーバーを構成する

NetScaler アプライアンスは、透過的で明示的なプロキシ仮想サーバーをサポートします。明示モードで SSL トラフィック用のプロキシ仮想サーバを設定するには、次の手順を実行します。

1. プロキシサーバを追加します。
2. SSL ポリシーをプロキシサーバーにバインドします。

CLI を使用してプロキシサーバを追加するには

コマンドプロンプトで入力します:

```
add cs vserver <name> [-td <positive_integer>] <serviceType> [-
cltTimeout <secs>]
```

例:

```
add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180
```

CLI を使用して **SSL** ポリシーをプロキシ仮想サーバーにバインドする

```
bind ssl vserver <vServerName> -policyName <string> [-priority <
positive_integer>]
```

ステップ **3: HTTPS** トラフィックの **SSL** インターセプトを構成する

HTTPS トラフィックの SSL インターセプトを設定するには、次の手順を実行します。

1. CA 証明書とキーのペアをプロキシ仮想サーバにバインドします。
2. SSL パラメータを使用して、デフォルトの SSL プロファイルを設定します。
3. フロントエンド SSL プロファイルをプロキシ仮想サーバーにバインドし、フロントエンド SSL プロファイルで SSL インターセプトを有効にします。

CLI を使用して CA 証明書とキーのペアをプロキシ仮想サーバーにバインドするには

コマンドプロンプトで入力します:

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
 -CA -skipCAName
```

CLI を使用してデフォルト SSL プロファイルを設定するには

コマンドプロンプトで入力します:

```
set ssl profile <name> -denySSLReneg <denySSLReneg> -sslInterception
(ENABLED | DISABLED)-ssliMaxSessPerServer positive_integer>
```

CLI を使用してフロントエンド **SSL** プロファイルをプロキシ仮想サーバーにバインドする

コマンドプロンプトで入力します:

```
set ssl vserver <vServer name> -sslProfile ssl_profile_interception
```

手順 **4:** キャッシュメモリを制限するように共有メモリを構成する

CLI を使用してキャッシュメモリを制限するように共有メモリを設定するには

コマンドプロンプトで入力します:

```
set cache parameter [-memLimit <megaBytes>]
```

ここで、キャッシュ用に構成されたメモリ制限は 10 MB に設定されます。

ステップ **5: URL** 分類パラメーターの設定

CLI を使用して URL 分類パラメータを設定するには

コマンドプロンプトで入力します:

```
set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer
>] [-TimeOfDayToUpdateDB <HH:MM>]
```

例:

```
set urlfiltering parameter -urlfilt_hours_betweenDB_updates 20
```


手順 **6**: **Citrix SSL** フォワードプロキシウィザードを使用して **URL** 分類を構成する

1. NetScaler アプライアンスにログオンし、[セキュリティ] > [**SSL** フォワードプロキシ] ページに移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います:

   a) [**SSL** フォワードプロキシウィザード] をクリックして、新しい構成を作成します。
   b) 既存の設定を選択し、[**Edit**] をクリックします。

3. [**URL** フィルタ] セクションで、[編集] をクリックします。
4. この機能を有効にするには、[**URL** 分類] チェックボックスをオンにします。
5. **URL** 分類ポリシーを選択し、[バインド] をクリックします。
6. [続行] をクリックし、[完了] をクリックします。

URL 分類ポリシーの詳細については、「URL 分類ポリシーを作成する方法」を参照してください。

ステップ **7: SSL** フォワードプロキシウィザードを使用した **URL** 分類パラメータの設定

1. **NetScaler** アプライアンスにログオンし、[セキュリティ **] > [URL** フィルタリング] に移動します。
2. [**URL** フィルタリング] ページで、[**URL** フィルタリング設定の変更] リンクをクリックします。
3. [**URL** フィルタリングパラメータの設定] ページで、次のパラメータを指定します。

   a) DB 更新間隔の時間。URL データベース更新の間隔をフィルタリングする時間。最小値:0、最大値:720。
   b) DB を更新する時刻。URL フィルタリングでデータベースを更新する時刻。
   c) クラウドホスト。クラウドサーバーの URL パス。
   d) シード DB パス。シードデータベース検索サーバーの URL パス。

4. 「**OK**」をクリックして「閉じる」をクリックします。

サンプル構成**:**

```
enable ns feature LB CS SSL IC RESPONDER AppFlow URLFiltering

enable ns mode FR L3 Edge USNIP PMTUD

set ssl profile ns_default_ssl_profile_frontend -denySSLReneg NONSECURE
    -sslInterception ENABLED -ssliMaxSessPerServer 100

add ssl certKey swg_ca_cert -cert ns_swg_ca.crt -key ns_swg_ca.key

set cache parameter -memLimit 100

add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180

add responder action act1 respondwith ""HTTP/1.1 200 OK\r\n\r\n" + http
    .req.url.url_categorize(0,0).reputation + "\n""

```



```
15 add responder policy p1 "HTTP.REQ.URL.URL_CATEGORIZE(0,0).CATEGORY.eq("
      Shopping/Retail") || HTTP.REQ.URL.URL_CATEGORIZE(0,0).CATEGORY.eq("
      Search Engines & Portals
16
17 ")" act1
18
19 bind cs vserver starcs_PROXY -policyName p1 -priority 10 -
      gotoPriorityExpression END -type REQUEST
20
21 add dns nameServer 10.140.50.5
22
23 set ssl parameter -denySSLReneg NONSECURE -defaultProfile ENABLED -
      sigDigestType RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384 RSA-
      SHA512 -ssliErrorCache ENABLED
24
25 -ssliMaxErrorCacheMem 100000000
26
27 add ssl policy pol1 -rule "client.ssl.origin_server_cert.subject.
      URL_CATEGORIZE(0,0).CATEGORY.eq("Search Engines & Portals")"" -
      action INTERCEPT
28
29 add ssl policy pol3 -rule "client.ssl.origin_server_cert.subject.ne("
      citrix")" -action INTERCEPT
30
31 add ssl policy swg_pol -rule "client.ssl.client_hello.SNI.
      URL_CATEGORIZE(0,0).CATEGORY.ne("Uncategorized")" -action INTERCEPT
32
33 set urlfiltering parameter -HoursBetweenDBUpdates 3 -
      TimeOfDayToUpdateDB 03:00
34 <!--NeedCopy-->
```

シードデータベースパスとクラウドサーバー名の設定

クラウド検索サーバー名とシードデータベースパスを手動で設定するために、シードデータベースパスとクラウドルックアップサーバー名を構成できるようになりました。そのために、「CloudHost」と「seedDBPath」という 2 つの新しいパラメータが URL フィルタリングパラメータに追加されます。

コマンドプロンプトで入力します:

```
set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer
>] [-TimeOfDayToUpdateDB <HH:MM>] [-LocalDatabaseThreads <positive_integer
>] [-CloudHost <string>] [-SeedDBPath <string>]
```

例:

```
set urlfiltering parameter -HoursBetweenDBUpdates 3 -TimeOfDayToUpdateDB
 03:00 -CloudHost localhost -SeedDBPath /mypath
```

NetScaler アプライアンスと NetSTAR の間の通信には、ドメインネームサーバーが必要な場合があります。アプライアンスからの単純なコンソール接続または Telnet 接続を使用してテストできます。

例:

```
root@ns# telnet nsv10.netstar-inc.com 443
Trying 1.1.1.1...
Connected to nsv10.netstar-inc.com.
Escape character is '^]'.

root@ns# telnet incompasshybridpc.netstar-inc.com 443
Trying 10.10.10.10...
Connected to incompasshybridpc.netstar-inc.com.
Escape character is '^]'.
<!--NeedCopy-->
```

監査ログメッセージを設定する

監査ログを使用すると、URL 分類プロセスのどの段階でも条件や状況を確認できます。NetScaler アプライアンスが着信 URL を受信すると、レスポンダーポリシーに URL フィルタリング式がある場合、監査ログ機能は URL に URL セット情報を収集します。この情報は、監査ログで許可されるすべてのターゲットのログメッセージとして格納されます。

- 送信元 IP アドレス (要求を行ったクライアントの IP アドレス)。
- 宛先 IP アドレス（要求されたサーバの IP アドレス)。
- スキーマ、ホスト、およびドメイン名 (http://www.example.com) を含むリクエストされた URL。
- URL フィルタリングフレームワークが返す URL カテゴリ。
- URL フィルタリングフレームワークが返した URL カテゴリグループ。
- URL フィルタリングフレームワークが返した URL レピュテーション番号。
- ポリシーによって実行された監査ログアクション。

URL リスト機能の監査ロギングを設定するには、次のタスクを完了する必要があります。

1. 監査ログを有効化。
2. 「監査ログの作成」メッセージアクション。
3. [監査ログメッセージ] アクションで URL リストレスポンダポリシーを設定します。

詳細については、「監査ログ 」トピックを参照してください。

SYSLOG メッセージングを使用した障害エラーの保存

URL フィルタリングプロセスのどの段階でも、システムレベルの障害が発生した場合、ADC アプライアンスは監査ログメカニズムを使用して ns.log ファイルにログを保存します。エラーは SYSLOG 形式でテキストメッセージとして保存されるため、管理者は後でイベント発生の時系列順に表示できます。これらのログは、アーカイブのために外部 SYSLOG サーバにも送信されます。詳細については、CTX229399 を参照してください。

たとえば、URL フィルタリング SDK を初期化するときにエラーが発生した場合、エラーメッセージは次のメッセージ形式で格納されます。

```
Oct 3 15:43:40 <local0.err> ns URLFiltering[1349]: Error initializing
 NetStar SDK (SDK error=-1). (status=1).
```

NetScaler アプライアンスは、4 つの異なる障害カテゴリにエラーメッセージを格納します。

- ダウンロードに失敗しました。分類データベースをダウンロードしようとしたときにエラーが発生した場合。
- 統合に失敗しました。更新を既存の分類データベースに統合するときにエラーが発生した場合。
- 初期化に失敗しました。URL 分類機能の初期化時にエラーが発生した場合は、分類パラメータを設定するか、分類サービスを終了してください。
- 検索に失敗しました。アプライアンスがリクエストの分類の詳細を取得するときにエラーが発生した場合。

NetStar イベント用の **SNMP** トラップの設定

URL フィルタリング機能では、次の条件が発生すると SNMP トラップが生成されます。

- NetStar データベースの更新が失敗するか成功する。
- NetStar SDK の初期化が失敗するか成功する。

アプライアンスには、SNMP アラームと呼ばれる条件付きエンティティのセットがあります。SNMP アラームの条件が満たされると、アプライアンスはトラップを生成し、指定されたトラップ宛先に送信します。たとえば、NetStar SDK の初期化に失敗すると、SNMP OID 1.3.6.1.4.1.5951.1.1.0.183 が生成され、トラップの送信先に送信されます。

アプライアンスがトラップを生成するには、まず SNMP アラームを有効にして設定する必要があります。次に、生成されたトラップメッセージをアプライアンスが送信するトラップ宛先を指定します。

SNMP アラームを有効にする

NetScaler アプライアンスは、有効になっている SNMP アラームに対してのみトラップを生成します。一部のアラームはデフォルトで有効になっていますが、無効にすることもできます。

SNMP アラームを有効にすると、成功または失敗のイベントが発生すると、URL フィルタリング機能によってトラップメッセージが生成されます。一部のアラームは、デフォルトで有効になっています。

コマンドラインインターフェイスを使用して SNMP アラームを有効にするには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します:

```
enable snmp alarm <trapName>
show snmp alarm <trapName>
```

NetScaler GUI を使用して SNMP アラームを有効にするには

1. [システム] > [**SNMP**] > [アラーム] に移動し、アラームを選択します。
2. [アクション] をクリックし、[有効] を選択します。

CLI を使用して SNMP アラームを設定する

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します:

```
set snmp alarm <trapName> [-thresholdValue <positive_integer> [-
normalValue <positive_integer>]] [-time <secs>] [-state ( ENABLED
| DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )
]
```

例:

```
set snmp alarm URL-FIL-DB-UPDATE-STATUS -state ENABLED
set snmp alarm URL-FIL-INIT-SDK -state ENABLED
```

GUI を使用して SNMP アラームを設定する

[システム] > [**SNMP**] > [アラーム] に移動し、アラームを選択し、アラームパラメータを設定します。

SNMP トラップの詳細については、SNMP のトピックを参照してください。

URL レピュテーションスコア

February 15, 2024

URL 分類機能は、ブラックリストに登録された URL を制限するポリシーベースの制御を提供します。URL カテゴリ、レピュテーションスコア、または URL カテゴリとレピュテーションスコアに基づいて、Web サイトへのアクセスを制御できます。ネットワーク管理者は、危険度の高い Web サイトにアクセスするユーザを監視する場合、URL レピュテーションスコアにバインドされたレスポンダーポリシーを使用して、そのような危険な Web サイトをブロックできます。

着信 URL 要求を受信すると、アプライアンスは URL 分類データベースからカテゴリおよびレピュテーションスコアを取得します。データベースから返されたレピュテーションスコアに基づいて、アプライアンスは Web サイトにレピュテーションレーティングを割り当てます。値の範囲は 1 ～4 です。次の表に示すように、4 は最もリスクのある Web サイトのタイプです。

URL レピュテーション評価	レピュテーションコメント
1	クリーンサイト
2	不明なサイト
3	危険の可能性がある、または危険なサイトに所属している

URL レピュテーション評価	レピュテーションコメント
4	悪意のあるサイト

ユースケース**:URL** レピュテーションスコアによるフィルタリング

ユーザーのトランザクションとネットワーク帯域幅の消費を監視するネットワーク管理者を持つ企業組織を考えてみましょう。マルウェアがネットワークに侵入した場合、管理者はデータセキュリティを強化し、ネットワークにアクセスする悪意のある危険な Web サイトへのアクセスを制御する必要があります。このような脅威からネットワークを保護するために、管理者は URL レピュテーションスコアによるアクセスを許可または拒否するように URL フィルタリング機能を設定できます。

ネットワーク上のアウトバウンドトラフィックとユーザーアクティビティの監視の詳細については、「Analytics」を参照してください。

組織の従業員がソーシャルネットワーキング Web サイトにアクセスしようとすると、ADC アプライアンスは URL 要求を受け取ります。URL 分類データベースを照会して、URL カテゴリをソーシャルネットワーキングとして取得し、レピュテーションスコア 3 を取得します。これは、潜在的に危険な Web サイトを示します。次に、アプライアンスは、管理者が設定したセキュリティポリシー（レピュテーションレーティングが 3 以上のサイトへのアクセスをブロックするなど）を確認します。次に、ポリシーアクションを適用して、Web サイトへのアクセスを制御します。

この機能を実装するには、SSL Forward Proxy ウィザードを使用して、URL レピュテーションスコアとセキュリティしきい値レベルを設定する必要があります。

GUI を使用したレピュテーションスコアの設定

SSL フォワードプロキシウィザードを使用して、レピュテーションスコアとセキュリティレベルを構成することをお勧めします。設定されたしきい値に基づいて、トラフィックを許可、ブロック、またはリダイレクトするポリシーアクションを選択できます。

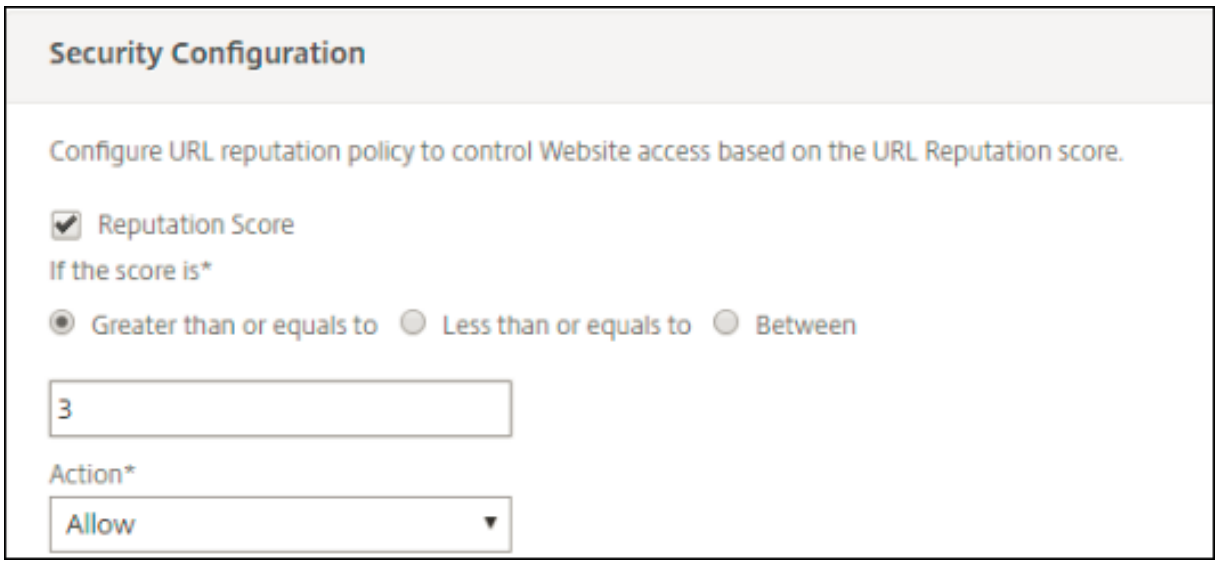
1. [セキュリティ] > [**SSL** 転送プロキシ] に移動します。
2. 詳細ウィンドウで、[**SSL** 転送プロキシウィザード] をクリックします。
3. 詳細ページで、プロキシサーバーの設定を指定します。
4. [**Continue**] をクリックして、SSL インターセプトなどの他の設定を指定し、管理を識別します。
5. [続行] をクリックして、[セキュリティの設定] セクションにアクセスします。
6. [セキュリティの設定] セクションで、[レピュテーションスコア] チェックボックスをオンにして、URL レピュテーションスコアに基づいてアクセスを制御します。
7. セキュリティレベルを選択し、レピュテーションスコアのしきい値を指定します。

    a) 次の値より大きい-しきい値が N 以上の場合に Web サイトを許可またはブロックします。N の範囲は 1 ～4 です。

b) 以下-しきい値が N 以下の場合に Web サイトを許可またはブロックします。N の範囲は 1 ～4 です。
c) [間隔（in between）]：しきい値が N1 ～N2 の間で、範囲が 1 ～4 の場合に、Web サイトを許可またはブロックします。

8. ドロップダウンリストからレスポンダーアクションを選択します。
9. [**** 続行して閉じる ****] をクリックします。

次の図は、SSL フォワードプロキシウィザードの [セキュリティの設定] セクションを示しています。[URL レピュテーションスコア] オプションを有効にして、ポリシー設定を構成します。



分析

March 20, 2024

NetScaler アプライアンスでは、すべてのユーザーレコードと後続のレコードがログに記録されます。NetScaler アプリケーション配信管理（ADM）を NetScaler アプライアンスと統合すると、アプライアンス内のログに記録されたユーザーアクティビティとその後のレコードが、`logstream`機能を使用して NetScaler Console にエクスポートされます。

NetScaler Console は、訪問した Web サイトや使用した帯域幅など、ユーザーのアクティビティに関する情報を照合して表示します。また、帯域幅の使用量と検出された脅威（マルウェアやフィッシングサイトなど）をレポートします。これらの主要なメトリックを使用して、ネットワークを監視し、Citrix SWG アプライアンスで修正アクションを実行できます。詳細については、「Citrix SSL フォワードプロキシ分析」を参照してください。

NetScaler アプライアンスを NetScaler コンソールと統合するには：

1. NetScaler アプライアンスで、SSL フォワードプロキシ機能を構成する際に、アナリティクスを有効にして、分析に使用する NetScaler Console インスタンスの詳細を指定します。

2. NetScaler コンソールで、NetScaler アプライアンスをインスタンスとして NetScaler コンソールに追加します。詳しくは、「NetScaler コンソールへのインスタンスの追加」を参照してください。

使用事例**:** リモートマルウェア検査に **ICAP** を使用して企業ネットワークを安全にする

December 8, 2023

NetScaler アプライアンスはプロキシとして機能し、すべてのクライアントトラフィックをインターセプトします。アプライアンスは、ポリシーを使用してトラフィックを評価し、リソースが存在するオリジンサーバーにクライアント要求を転送します。アプライアンスはオリジンサーバーからの応答を復号化し、プレーンテキストのコンテンツを ICAP サーバーに転送してマルウェア対策チェックを行います。ICAP サーバーは、「調整不要」、エラー、または変更された要求を示すメッセージで応答します。ICAP サーバーからの応答に応じて、要求されたコンテンツがクライアントに転送されるか、適切なメッセージが送信されます。

このユースケースでは、NetScaler アプライアンスで一般的な構成、プロキシと SSL インターセプトに関連する構成、および ICAP 構成を実行する必要があります。

一般的な構成

次のエンティティを構成します。

- NSIP アドレス
- サブネット IP（SNIP）アドレス
- DNS ネームサーバー
- SSL インターセプト用のサーバー証明書に署名するための CA 証明書とキーのペア

プロキシサーバーと **SSL** インターセプトの設定

次のエンティティを構成します。

- すべてのアウトバウンド HTTP および HTTPS トラフィックをインターセプトする明示モードのプロキシサーバー。
- SSL プロファイルは、接続の暗号やパラメータなどの SSL 設定を定義します。
- トラフィックを傍受するためのルールを定義する SSL ポリシー。true に設定すると、すべてのクライアント要求がインターセプトされます。

詳細については、以下のトピックを参照してください。

- プロキシモード
- SSL インターセプト

次の設定例では、マルウェア対策検出サービスはwww.example.comにあります。

一般的な設定の例:

```
add dns nameServer 203.0.113.2

add ssl certKey ns-swg-ca-certkey -cert ns_swg_ca.crt -key ns_swg_ca.
    key
<!--NeedCopy-->
```

プロキシサーバーと **SSL** インターセプト設定の例:

```
add cs vserver explicitswg PROXY 192.0.2.100 80  -Authn401 ENABLED  -
    authnVsName explicit-auth-vs

set ssl parameter -defaultProfile ENABLED

add ssl profile swg_profile -sslInterception ENABLED

bind ssl profile swg_profile -ssliCACertkey ns-swg-ca-certkey

set ssl vserver explicitswg -sslProfile swg_profile

add ssl policy ssli-pol_ssli -rule true -action INTERCEPT

bind ssl vserver explicitswg -policyName ssli-pol_ssli -priority 100 -
    type INTERCEPT_REQ
<!--NeedCopy-->
```

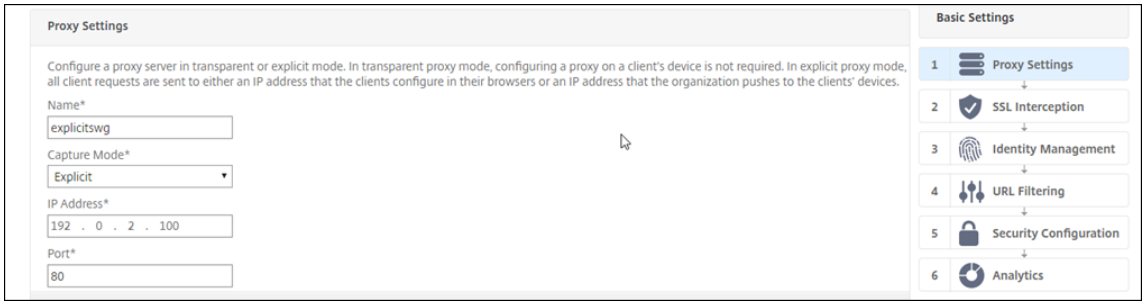
ICAP 設定の例:

```
add service icap_svc 203.0.113.225 TCP 1344

enable ns feature contentinspection

add icapprofile icapprofile1 -uri /example.com -Mode RESMOD

add contentInspection action CiRemoteAction -type ICAP -serverName
    icap_svc -icapProfileName icapprofile1

add contentInspection policy CiPolicy -rule "HTTP.REQ.METHOD.NE("
    CONNECT")" -action CiRemoteAction

bind cs vserver explicitswg -policyName  CiPolicy -priority 200 -type
    response
<!--NeedCopy-->
```

プロキシ設定を構成する

1. セキュリティ > **SSL** フォワードプロキシ > **SSL** フォワードプロキシウィザードに移動します。
2. [始める] をクリックし、[続行] をクリックします。

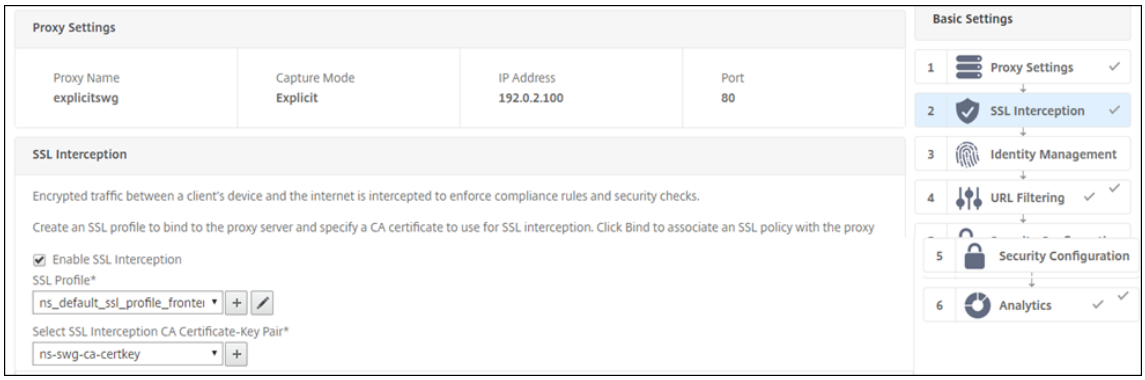
3. [プロキシ設定] ダイアログボックスで、明示的なプロキシサーバーの名前を入力します。
4. [キャプチャモード] で [明示的] を選択します。
5. IP アドレスとポート番号を入力します。



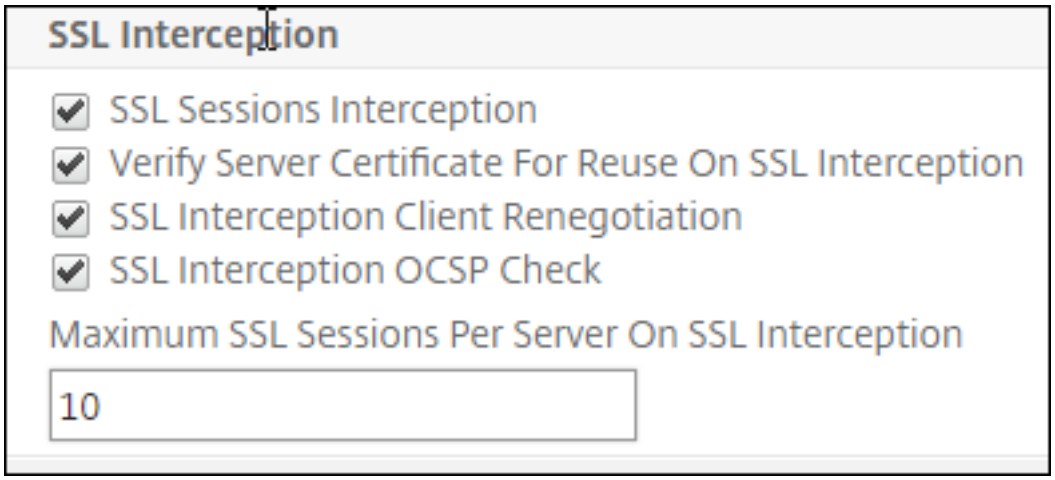
6. [続行] をクリックします。

SSL インターセプト設定を構成します

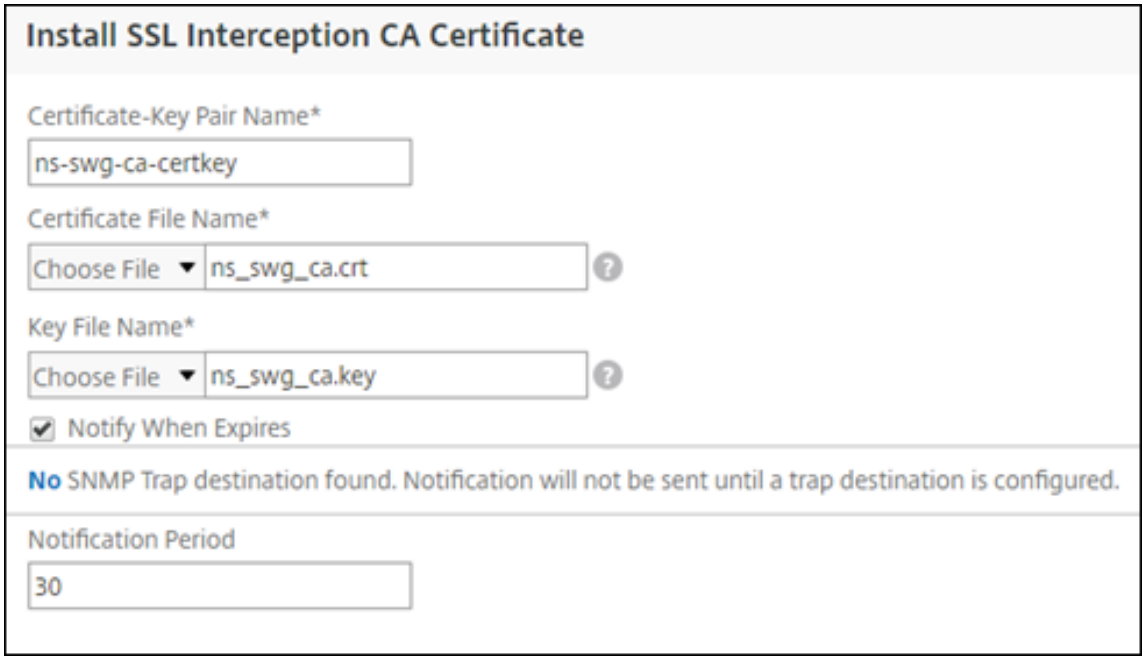
1. [**SSL** インターセプトを有効にする] を選択します。



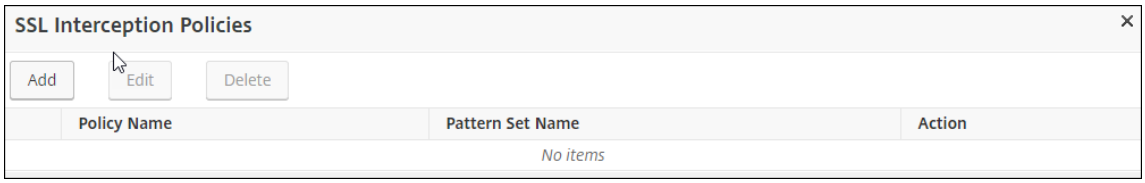
2.「**SSL** プロファイル」で、既存のプロファイルを選択するか、「+」をクリックして新しいフロントエンド SSL プロファイルを追加します。このプロファイルで **SSL** セッションインターセプト を有効にします。既存のプロファイルを選択する場合は、次の手順をスキップしてください。



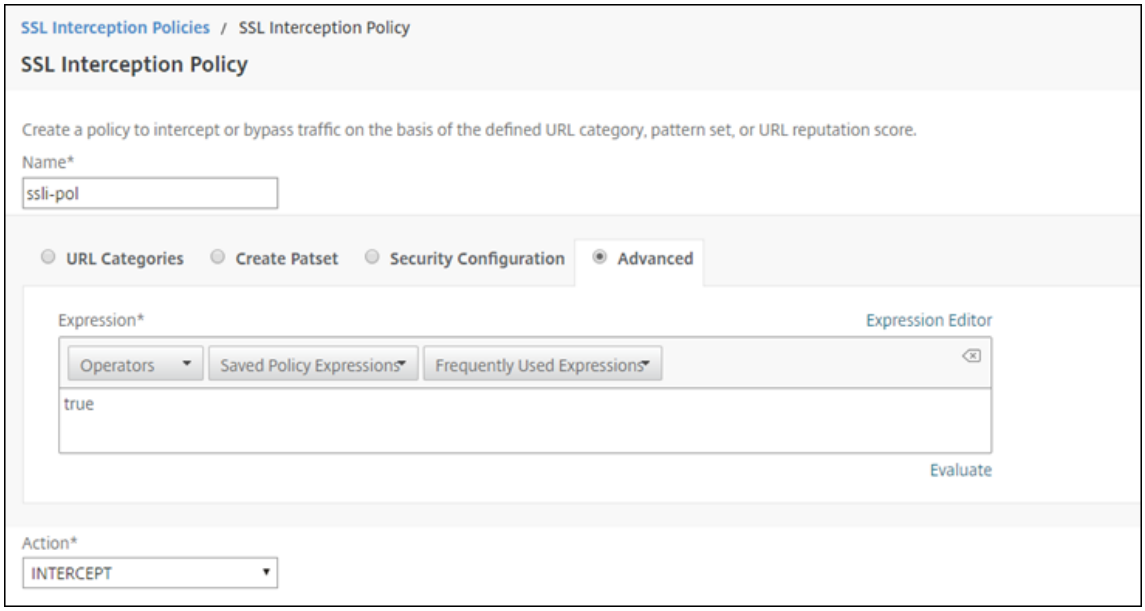
3. 「**OK**」をクリックし、「完了」をクリックします。
4. 「**SSL** インターセプション **CA** 証明書とキーのペアの選択」で、既存の証明書を選択するか、「**+**」をクリックして **SSL** インターセプト用の **CA** 証明書とキーのペアをインストールします。既存の証明書を選択した場合は、次の手順をスキップします。



5. [インストール] をクリックし、[閉じる] をクリックします。
6. すべてのトラフィックを代行受信するポリシーを追加します。[**Bind**] をクリックします。[追加] をクリックして新しいポリシーを追加するか、既存のポリシーを選択します。既存のポリシーを選択した場合は、[挿入] をクリックし、次の 3 つの手順をスキップしてください。



7. ポリシーの名前を入力し、[**Advanced**] を選択します。エクスプレッションエディタで、true と入力します。
8. [アクション] で [インターセプト] を選択します。

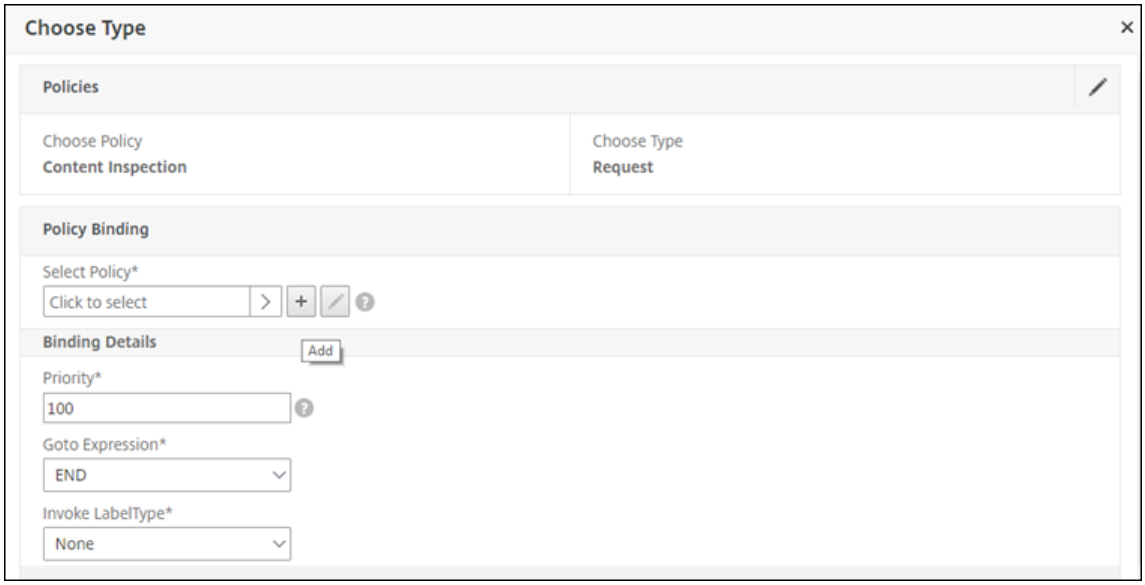


9. [作成] をクリックします。
10. [続行] を 4 回クリックし、[完了] をクリックします。

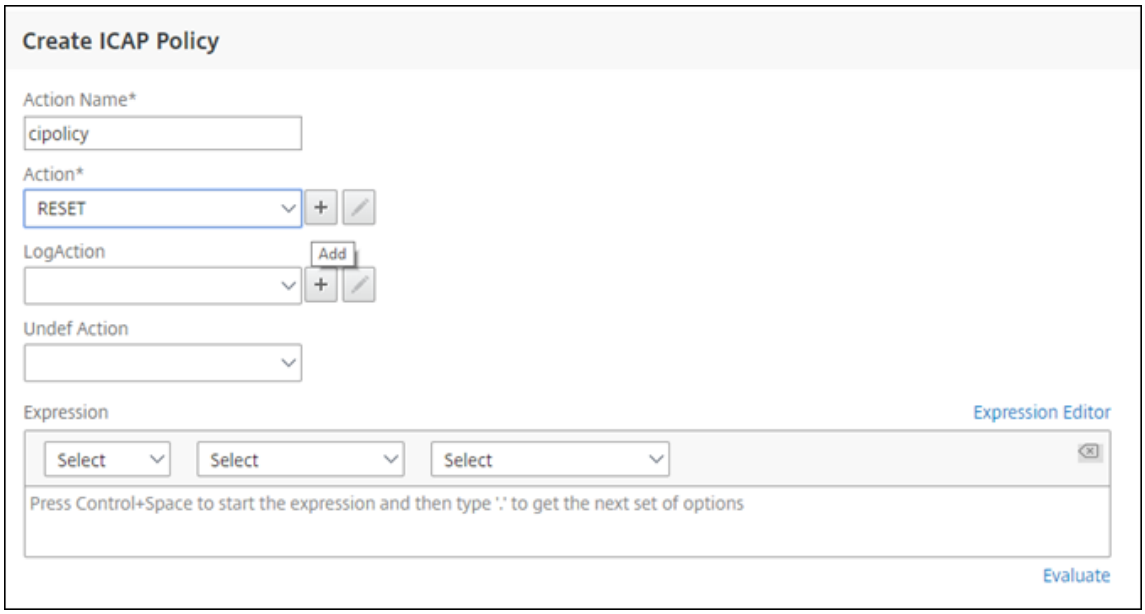
ICAP 設定を構成します

1. 負荷分散 > サービスに移動し、追加をクリックします。
2. 名前と IP アドレスを入力します。「プロトコル」で「**TCP**」を選択します。[ポート] に「**1344**」と入力します。[**OK**] をクリックします。
3. [**SSL** 転送プロキシ] > [プロキシ仮想サーバー] に移動します。プロキシ仮想サーバーを追加するか、仮想サーバーを選択して「編集」をクリックします。詳細を入力したら、「**OK**」をクリックします。

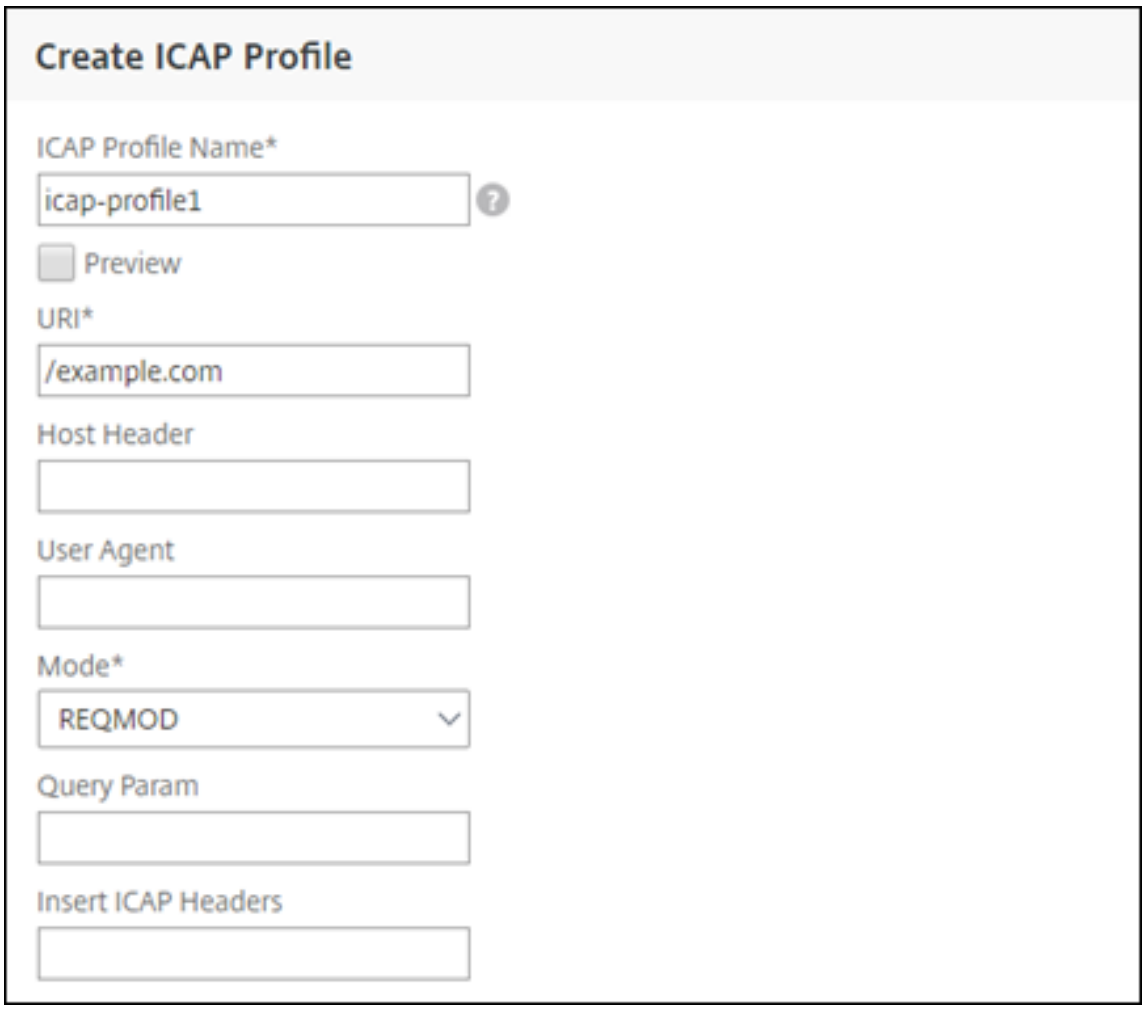
   もう一度「**OK**」をクリックします。
4. [詳細設定] で、[ポリシー] をクリックします。
5. 「ポリシーの選択」で、「コンテンツ検査」を選択します。[続行] をクリックします。
6. 「ポリシーの選択」で、「+」記号をクリックしてポリシーを追加します。



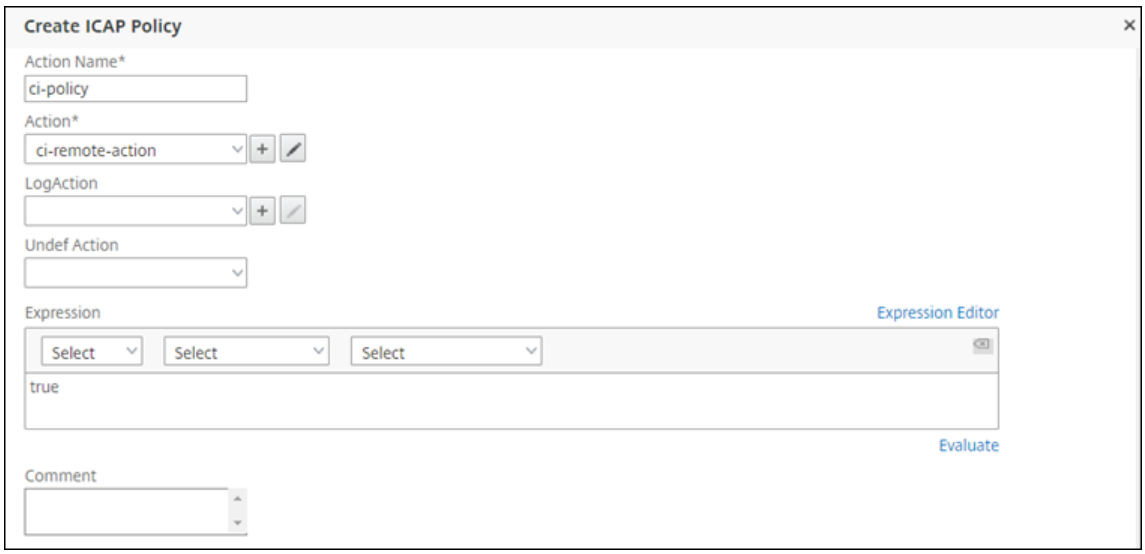
7. ポリシーの名前を入力します。「アクション」で、「+」記号をクリックしてアクションを追加します。



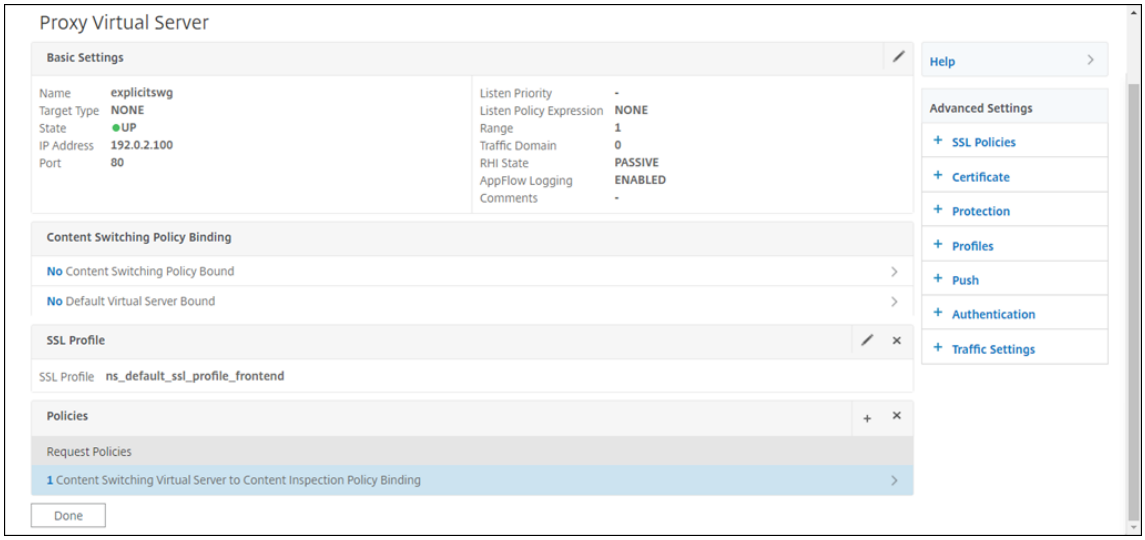
8. アクションの名前を入力します。**[**サーバー名**]** に、以前に作成した TCP サービスの名前を入力します。**ICAP** プロファイルで、「+」記号をクリックして ICAP プロファイルを追加します。
9. プロファイル名、URI を入力します。「モード」で「**REQMOD**」を選択します。



10. ［作成］をクリックします。
11. [**ICAP** アクションの作成] ページで、[作成] をクリックします。
12. **ICAP** ポリシーの作成ページで、エクスプレッションエディタに true と入力します。次に、「作成」をクリックします。



13. [**Bind**] をクリックします。
14. コンテンツ検査機能を有効にするように求められたら、[はい] を選択します。
15. [完了] をクリックします。



NetScaler アプライアンスと **RESPMOD** の **ICAP** サーバー間の **ICAP** トランザクションの例

NetScaler アプライアンスから **ICAP** サーバーへのリクエスト:

```
RESPMOD icap://10.106.137.15:1344/resp ICAP/1.0

Host: 10.106.137.15

Connection: Keep-Alive

Encapsulated: res-hdr=0, res-body=282
```



```
8
9  HTTP/1.1 200 OK
10
11 Date: Fri, 01 Dec 2017 11:55:18 GMT
12
13 Server: Apache/2.2.21 (Fedora)
14
15 Last-Modified: Fri, 01 Dec 2017 11:16:16 GMT
16
17 ETag: "20169-45-55f457f42aee4"
18
19 Accept-Ranges: bytes
20
21 Content-Length: 69
22
23 Keep-Alive: timeout=15, max=100
24
25 Content-Type: text/plain; charset=UTF-8
26
27 X5O!P%@AP[4PZX54(P^)7CC)7 }
28  $EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
29 <!--NeedCopy-->
```

ICAP サーバーから **NetScaler** アプライアンスへの応答:

```
1  ICAP/1.0 200 OK
2
3  Connection: keep-alive
4
5  Date: Fri, 01 Dec, 2017 11:40:42 GMT
6
7  Encapsulated: res-hdr=0, res-body=224
8
9  Server: IWSVA 6.5-SP1_Build_Linux_1080 $Date: 04/09/2015 01:19:26 AM$
10
11 ISTag: "9.8-13.815.00-3.100.1027-1.0"
12
13 X-Virus-ID: Eicar_test_file
14
15 X-Infection-Found: Type=0; Resolution=2; Threat=Eicar_test_file;
16
17 HTTP/1.1 403 Forbidden
18
19 Date: Fri, 01 Dec, 2017 11:40:42 GMT
20
21 Cache-Control: no-cache
22
23 Content-Type: text/html; charset=UTF-8
24
25 Server: IWSVA 6.5-SP1_Build_Linux_1080 $Date: 04/09/2015 01:19:26 AM$
26
27 Content-Length: 5688
28
```



```
<html><head><META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset
    =UTF-8"/>

...

...

</body></html>
<!--NeedCopy-->
```

ハウツー記事

August 15, 2023

以下に、SSL フォワードプロキシ導入の管理に役立つ「ハウツー」記事として用意されている設定手順や機能的なユースケースをいくつか紹介します。

URL フィルタリング

URL 分類ポリシーの作成方法

URL リストポリシーの作成方法

例外的な URL を許可する方法

アダルトカテゴリのウェブサイトをブロックする方法

セキュリティ

August 15, 2023

次のトピックでは、NetScaler セキュリティ機能の構成とインストールについて説明します。これらの機能のほとんどはポリシーベースです。

コンテンツフィルタリング	不適切な HTML 要求をブロックし、要求が Web サーバーに到達するのを防ぎます。
サージ保護	急激に増加した接続の試行を検出し、接続がサーバーに進むことができる速度を調整して、サーバーの過負荷を防ぎます。

DNS セキュリティオプション	DNS 攻撃から保護するためのポリシーを作成する、簡略化された UI ウィザード。

サージ保護

August 15, 2023

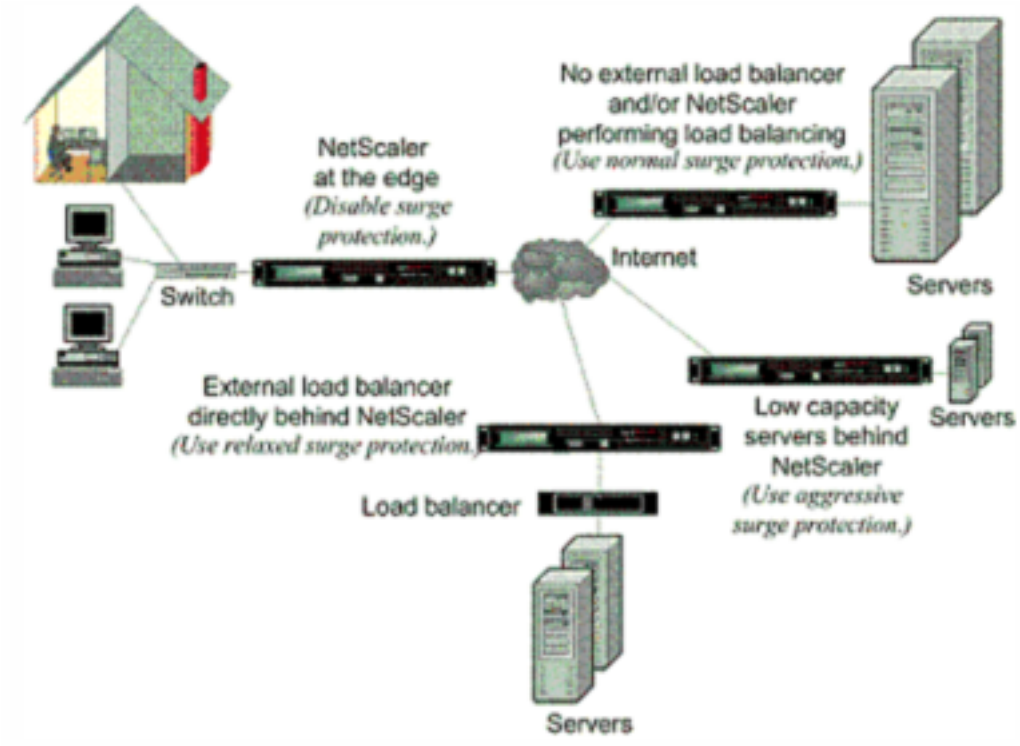
クライアント要求の急増によってサーバが過負荷になると、サーバの応答が遅くなり、サーバは新しい要求に応答できなくなります。サージ保護機能により、サーバが処理できる速度でサーバへの接続が確実に行われます。応答率は、サージ保護の設定方法によって異なります。NetScaler ADC アプライアンスはサーバーへの接続数も追跡し、その情報を使用して新しいサーバー接続を開く速度を調整します。

サージ保護はデフォルトで有効になっています。いくつかの特別な構成の場合のように、サージ保護を使用しない場合は、それを無効にする必要があります。

ほとんどの用途にはデフォルトのサージ保護設定で十分ですが、サージ保護を構成して必要に応じて調整できます。まず、スロットル値を設定して、接続試行をどれだけ積極的に管理するかを知ることができます。次に、基本しきい値を設定して、NetScaler ADC アプライアンスがサージ保護をトリガーする前に許可する同時接続の最大数を制御できます。(デフォルトのベースしきい値はスロットル値によって設定されますが、スロットル値を設定した後は任意の数値に変更できます)。

次の図は、Web サイトへのトラフィックを処理するようにサージ保護がどのように構成されているかを示しています。

図 1: NetScaler サージ保護の機能図



> 注
>
> NetScaler ADC アプライアンスがインターネットのクライアント側のネットワークデバイスとやり取りするネットワークのエッジにインストールされている場合は、サージ保護機能を無効にする必要があります。アプライアンスで USIP (ソース IP を使用) モードを有効にする場合は、サージ保護も無効にする必要があります。

サージ保護を無効にして要求が急増すると、サーバーは同時に処理できる限り多くの要求を受け入れ、要求の破棄を開始します。サーバの過負荷が増えるにつれて、サーバはダウンし、応答速度はゼロに低下します。数分後にサーバーがクラッシュから回復すると、異常な動作であるすべての保留中の要求のリセットが送信され、リセットで新しい要求にも応答します。このプロセスは、要求の急増ごとに繰り返されます。したがって、DDoS 攻撃を受けて複数の要求が急増するサーバーは、正当なユーザーが使用できなくなる可能性があります。

サージ保護が有効になっていて要求の急増が発生した場合、サージ保護はサーバーへの要求のレートを管理し、サーバーが要求を処理できる速さでサーバーに要求を送信します。これにより、サーバーは各要求に受信順に正しく応答できます。急増が終わると、バックログされたリクエストは、リクエストレートがレスポンスレートと一致するまで、サーバーが処理できる限り早くクリアされます。

サージ保護を無効にしてから再度有効にする

August 15, 2023

サージ保護機能は、デフォルトで有効になっています。サージ保護を有効にすると、追加するすべてのサービスに対してアクティブになります。

CLI を使用したサージ保護の無効化または再有効化

コマンドプロンプトで、次のいずれかのコマンドセットを入力して、サージ保護を無効または再度有効にし、構成を確認します。

```
-  disable ns feature SurgeProtection
-  show ns feature
-  enable ns feature SurgeProtection
-  show ns feature
<!--NeedCopy-->
```

例:

```
disable ns feature SurgeProtection
 Done show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   OFF
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

```
 enable ns feature SurgeProtection
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 .
 .
 .

 24)    NetScaler Push                  push                 OFF
 Done
>
<!--NeedCopy-->
```


GUI を使用したサージ保護の無効化または再有効化

1. ナビゲーションウィンドウで、[システム] を展開し、[設定] を選択します。
2. 詳細ウィンドウで、[高度な機能の変更] をクリックします。
3. [高度な機能の構成] ダイアログボックスで、[サージ保護] チェックボックスをオフにしてサージ保護機能を無効にするか、チェックボックスをオンにして機能を有効にします。
4. [**OK**] をクリックします。
5. [機能の有効化/無効化] ダイアログボックスで、[はい] をクリックします。ステータスバーに、機能が有効または無効になったことを示すメッセージが表示されます。

GUI を使用して特定のサービスのサージ保護を無効または再度有効にする

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。構成されたサービスのリストが詳細ペインに表示されます。
2. 詳細ウィンドウで、サージ保護機能を無効または再度有効にするサービスを選択し、[開く] をクリックします。
3. [サービスの構成] ダイアログボックスで、[詳細設定**]** タブをクリックし、下にスクロールします。
4. [その他（Others）] フレームで、[サージ保護（**Surge Protection** ）] チェックボックスの選択を解除してサージ保護機能を無効にするか、またはチェックボックスを選択してサージ保護機能を有効にします。
5. [**OK**] をクリックします。ステータスバーに、機能が有効または無効になったことを示すメッセージが表示されます。
   注**:** サージ保護は、機能とサービス設定の両方が有効になっている場合にのみ機能します。

サージ保護のしきい値を設定する

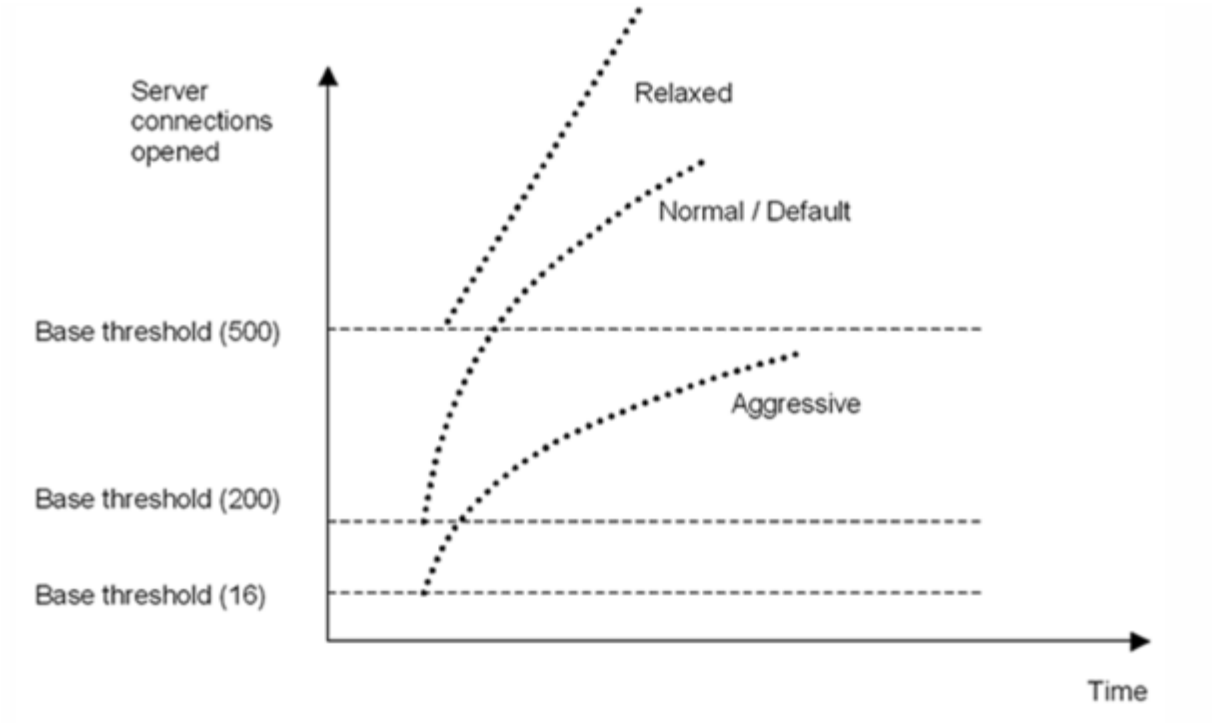
August 15, 2023

NetScaler ADC アプライアンスがサーバーへの接続を開くレートを設定するには、サージ保護のしきい値とスロットル値を構成する必要があります。

> 注
>
> しきい値はグローバルに設定されますが、個々の負荷分散サーバーまたはサービスごとに適用されます。

次の図は、スロットルレートをリラックス、ノーマル、またはアグレッシブに設定した結果のサージ保護曲線を示しています。サーバ容量の構成に応じて、適切なサージ保護曲線を生成するための基本しきい値を設定できます。

図 1: サージ保護曲線

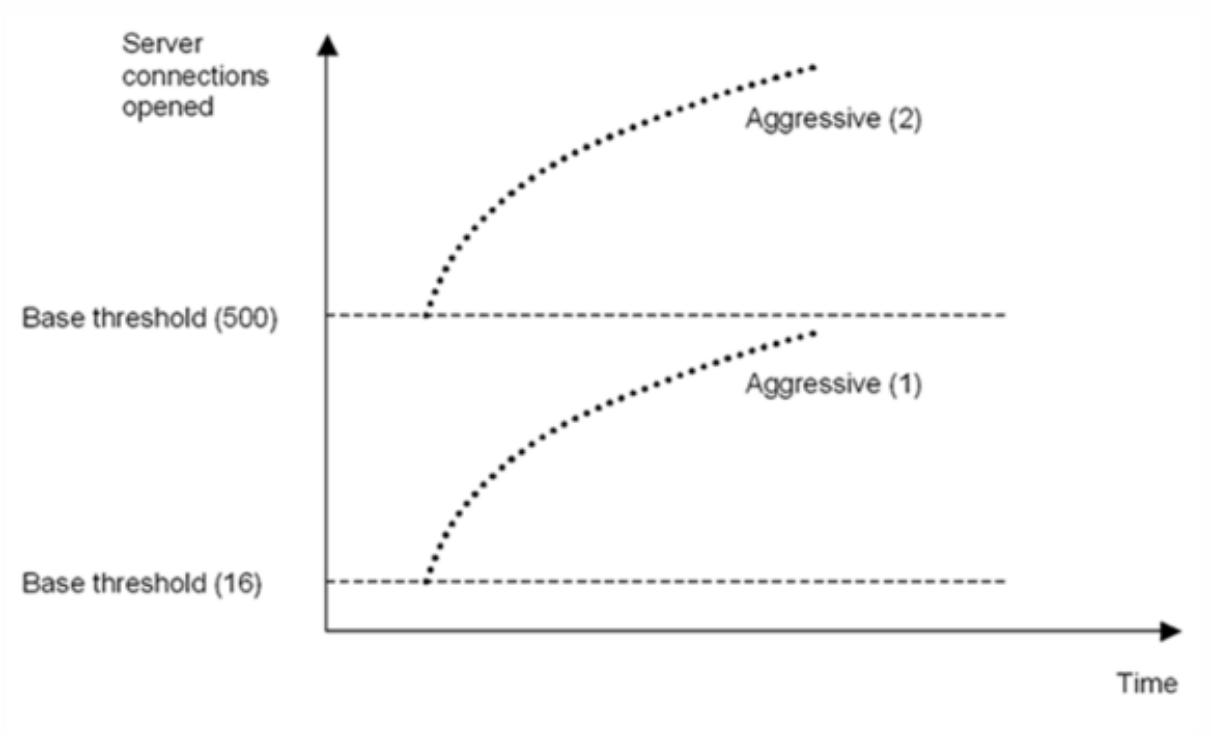


構成設定は、サージ保護の動作に次のように影響します。

- スロットルレートを指定しない場合、前の図に示すように、スロットルレートは標準（デフォルト値）に設定され、基本しきい値は 200 に設定されます。
- 基本しきい値を指定せずにスロットルレート（アグレッシブ、ノーマル、リラックス）を指定した場合、カーブにはそのスロットルレートの基本しきい値のデフォルト値が反映されます。たとえば、スロットルレートをリラックスに設定すると、作成されるカーブのベースしきい値は 500 になります。
- ベースしきい値のみを指定した場合、次の図に示すように、指定した値に応じて、サージ保護曲線全体が上下にシフトします。
- ベースしきい値とスロットルレートの両方を指定した場合、結果として得られるサージ保護曲線は、設定されたスロットルレートに基づいており、ベースしきい値に設定された値に従って調整されます。

次の図では、スロットルレートがアグレッシブに設定されているが、ベースしきい値は設定されていない場合に、下のカーブ（アグレッシブ 1）になります。上側の曲線（Aggressive 2）は、ベースしきい値が 500 に設定されているが、スロットルレートが設定されていない場合の結果になります。2 番目の上側の曲線 (アグレッシブ 2) は、ベースしきい値が 500 に設定され、スロットルレートがアグレッシブに設定されている場合にも生じます。

図 2：デフォルトまたは設定されたベースしきい値を使用したアグレッシブレート



GUI を使用してサージ保護のしきい値を設定する

1. ナビゲーションウィンドウで、[システム] を展開し、[設定] を選択します。
2. 詳細ペインで、[グローバルシステム設定] をクリックします。
3. スロットルレートのデフォルトとは異なる基本しきい値を設定する場合は、[グローバル設定の構成] ダイアログボックスの [基本しきい値] テキストボックスに、サージ保護がトリガーされるまでに許可される同時サーバー接続の最大数を入力します。基本しきい値は、サージ保護が有効になる前に開くことができるサーバー接続の最大数です。この設定の最大値は 32,767 サーバー接続です。この値のデフォルト設定は、次の手順で選択するスロットルレートによって制御されます。

   注**:** ここで明示的な値を設定しない場合、デフォルト値が使用されます。
4. [Throttle] ドロップダウンリストで、スロットルレートを選択します。スロットルは、NetScaler ADC アプライアンスがサーバーへの接続を開くことを許可するレートです。スロットルは次の値に設定できます。

   - アグレッシブ**:** サーバーの接続処理能力とサージ処理能力が低く、接続を慎重に管理する必要がある場合に、このオプションを選択します。スロットルをアグレッシブに設定すると、基本しきい値はデフォルト値の 16 に設定されます。これは、サーバーへの同時接続が 17 以上ある場合は常にサージ保護がトリガーされることを意味します。
   - 通常： NetScaler ADC アプライアンスの背後または下流に外部ロードバランサがない場合は、このオプションを選択します。基本しきい値は 200 に設定されています。これは、サーバーへの同時接続数が

201 以上ある場合は常にサージ保護がトリガーされることを意味します。Normal はデフォルトのスロットルオプションです。

- リラックス： NetScaler ADC アプライアンスが多数の Web サーバー間で負荷分散を実行しているため、多数の同時接続を処理できる場合は、このオプションを選択します。基本しきい値は 500 に設定されています。これは、サーバーへの同時接続が 501 以上ある場合にのみサージ保護がトリガーされることを意味します。

5.「OK」をクリックします。ステータスバーに、グローバル設定が構成されていることを示すメッセージが表示されます。

サージキューをフラッシュする

August 15, 2023

物理サーバーが要求を大量に受信すると、現在接続しているクライアントへの応答が遅くなり、ユーザーは不満を抱き、不満を抱きます。多くの場合、オーバーロードにより、クライアントはエラーページを受信します。このような過負荷を回避するために、NetScaler アプライアンスには、サービスへの新しい接続を確立する速度を制御するサージ保護などの機能が備わっています。

アプライアンスは、クライアントと物理サーバ間の接続の多重化を行います。サーバー上のサービスにアクセスするクライアント要求を受信すると、アプライアンスはサーバーへの接続がすでに確立されていて、空いているかを探します。空き接続が見つかった場合、その接続を使用してクライアントとサーバー間の仮想リンクを確立します。既存の空き接続が見つからない場合、アプライアンスはサーバとの新しい接続を確立し、クライアントとサーバ間の仮想リンクを確立します。ただし、アプライアンスがサーバーとの新しい接続を確立できない場合、クライアント要求をサージキューに送信します。負荷分散またはコンテンツスイッチング仮想サーバーにバインドされているすべての物理サーバーがクライアント接続の上限（最大クライアント値、サージ保護しきい値、またはサービスの最大容量）に達すると、アプライアンスはどのサーバーとも接続を確立できなくなります。サージ保護機能は、サージキューを使用して物理サーバーとの接続を開く速度を調整します。アプライアンスは、仮想サーバにバインドされたサービスごとに異なるサージキューを維持します。

サージキューの長さは、アプライアンスが接続を確立できない要求が来るたびに増加し、キュー内の要求がサーバーに送信されるか、要求がタイムアウトしてキューから削除されるたびに長くなります。

サービスまたはサービスグループのサージキューが長すぎる場合は、そのサージキューをフラッシュする必要があります。特定のサービスまたはサービスグループ、または負荷分散仮想サーバーにバインドされているすべてのサービスとサービスグループのサージキューをフラッシュできます。サージキューをフラッシュしても、既存の接続には影響しません。サージキューに存在するリクエストのみが削除されます。これらのリクエストについては、クライアントは新たなリクエストを行う必要があります。

コンテンツスイッチング仮想サーバーのサージキューをフラッシュすることもできます。コンテンツスイッチング仮想サーバーが特定の負荷分散仮想サーバーにいくつかの要求を転送し、負荷分散仮想サーバーも他の要求を受信した

場合、コンテンツスイッチング仮想サーバーのサージキューをフラッシュすると、このコンテンツスイッチング仮想サーバーから受信した要求のみがフラッシュされます。負荷分散仮想サーバーのサージキュー内の他の要求はフラッシュされません。

注:

> - キャッシュリダイレクト、認証、VPN、または GSLB 仮想サーバーまたは GSLB サービスのサージキューをフラッシュすることはできません。
> - [ソース IP の使用（USIP）] が有効になっている場合は、サージ保護機能を使用しないでください。

CLI を使用してサージキューをフラッシュする

flush ns SurgeQ コマンドは次のように機能します。

- サージキューをフラッシュする必要があるサービス、サービスグループ、または仮想サーバーの名前を指定できます。
- コマンドの実行中に名前を指定すると、指定したエンティティのサージキューがフラッシュされます。複数のエンティティが同じ名前を持つ場合、アプライアンスはこれらすべてのエンティティのサージキューをフラッシュします。
- コマンドの実行中にサービスグループの名前、サーバー名、およびポートを指定すると、アプライアンスは指定されたサービスグループメンバーのみのサージキューをフラッシュします。
- サービスグループ<name>の名前を指定せずにサービスグループメンバー<serverName> and <port>を直接指定することはできません。また、<serverName>なしで<port>を指定することはできません。特定のサービスグループメンバーのサージキューをフラッシュする場合は、<serverName>および<port>を指定します。
- 名前を指定せずにコマンドを実行すると、アプライアンスはアプライアンスに存在するすべてのエンティティのサージキューをフラッシュします。
- サービスグループメンバーがサーバ名で識別される場合は、このコマンドでサーバ名を指定する必要があります。IP アドレスは指定できません。

コマンドプロンプトで入力します。

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
```

例

1. `flush ns surgeQ -name SVC1ANZGB -serverName 10.10.10.1 80`

   前述のコマンドは、SVC1ANZGB という名前で IP アドレスが 10.10.10 のサービスまたは仮想サーバのサージキューをフラッシュします。

2. `flush ns surgeQ`

   前述のコマンドは、アプライアンス上のすべてのサージキューをフラッシュします。

GUI を使用したサージキューのフラッシュ

[トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバーを選択し、[アクション] リストで [フラッシュサージキュー] を選択します。

DNS セキュリティオプション

August 15, 2023

NetScaler GUI の［DNS セキュリティプロファイルの追加］ページから DNS セキュリティオプションを構成できるようになりました。NetScaler CLI または NITRO API から DNS セキュリティオプションを構成するには、AppExpert コンポーネントを使用します。手順については、NITRO API ドキュメントと NetScaler コマンドリファレンスガイドを参照してください。

オプションの 1 つであるキャッシュポイズニング保護はデフォルトで有効になっており、無効にすることはできません。次の表に示すように、他のオプションをすべての DNS エンドポイントに適用することも、導入環境内の特定の DNS 仮想サーバーに適用することもできます。

セキュリティオプション	すべての DNS エンドポイントに適用できますか?	特定の DNS 仮想サーバーに適用できますか?
DNS DDoS 保護	はい	はい
例外の管理—ホワイトリスト/ブラックリストサーバー	はい	はい
ランダムなサブドメイン攻撃を防ぐ	はい	はい
キャッシュをバイパスする	はい	いいえ
TCP 経由の DNS トランザクションを強制	はい	はい
DNS 応答にルートの詳細を提供する	はい	いいえ

キャッシュポイズニング保護

キャッシュポイズニング攻撃は、ユーザーを正規のサイトから悪意のあるウェブサイトにリダイレクトします。

たとえば、攻撃者は DNS キャッシュ内の正規の IP アドレスを制御する偽の IP アドレスに置き換えます。サーバーがこれらの IP アドレスからの要求に応答すると、キャッシュはポイズニングされます。ドメインのアドレスに対する後続のリクエストは、攻撃者のサイトにリダイレクトされます。

[キャッシュポイズニング保護] オプションを使用すると、DNS サーバーの要求と応答をキャッシュするデータベースに破損したデータが挿入されるのを防ぐことができます。この機能は NetScaler ADC アプライアンスに組み込まれており、常に有効になっています。

DNS DDoS 保護

DDoS 攻撃で使用される可能性があると思われる要求の種類ごとに、DNS DDoS 保護オプションを設定できます。アプライアンスは、タイプごとに、指定した期間（タイムスライス）で受信した要求数のしきい値を超えた後に受信したすべての要求をドロップします。このオプションを設定して、SYSLOG サーバに警告を記録することもできます。次に例を示します:

- **DROP:** -ログなしでリクエストをドロップするには、このオプションを選択します。しきい値 15、タイムスライス 1 秒で A レコード保護を有効にし、[DROP] を選択したと仮定します。着信要求が 1 秒で 15 クエリを超えると、パケットがドロップし始めます。
- 警告**:** -リクエストをログに記録およびドロップするには、このオプションを選択します。しきい値 15、タイムスライス 1 秒で A レコード保護を有効にし、[WARN] を選択したと仮定します。着信要求が 1 秒で 15 クエリを超えると、脅威を示す警告メッセージがログに記録され、パケットがドロップされます。WARN のしきい値を、レコードタイプの DROP のしきい値よりも小さく設定することをお勧めします。このような設定は、実際の攻撃が発生して NetScaler ADC が着信要求のドロップを開始する前に警告メッセージを記録することにより、管理者が攻撃を識別するのに役立ちます。

GUI を使用して着信トラフィックのしきい値を設定する

1. [設定] > [セキュリティ] > [**DNS** セキュリティ] に移動します。
2. 「**DNS** セキュリティプロファイル」ページで、「追加」をクリックします。
3. 「**DNS** セキュリティプロファイルの追加」ページで、次の操作を行います。
4. **DNS DDoS** 保護を拡張してください。
    a) レコードタイプを選択し、しきい値制限とタイムスライス値を入力します。
    b) [ドロップ] または [警告] を選択します。
    c) 保護対象とする他のレコードタイプについて、手順 a と b を繰り返します。
5. [**Submit**] をクリックします。

例外の管理—許可リスト/ブロックリストサーバー

例外の管理を使用すると、ブロックリストまたは許可リストのドメイン名と IP アドレスに例外を追加できます。次に例を示します:

- 攻撃をポストする特定の IP アドレスが特定されると、そのような IP アドレスをブロックリストに追加できます。

- 管理者が特定のドメイン名に対する要求数が予期せず多いことがわかった場合、そのドメイン名をブロックリストに追加できます。
- サーバーリソースを消費する可能性がある`NXDomains`および既存のドメインの一部はブラックリストに登録できます。
- 管理者がリストドメイン名または IP アドレスを許可すると、これらのドメインまたは IP アドレスからのクエリまたは要求のみが応答され、その他はすべて削除されます

GUI を使用して許可リストまたはブロックリストを作成する

1. [構成**] > [**セキュリティ**] > [DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3.「**DNS** セキュリティプロファイルの追加」ページで、次の操作を行います。

   a)「例外管理—ホワイトリスト**/**ブラックリストサーバー」を拡張してください。
   b) ブラックリストに登録されているドメイン**/**アドレスからのクエリをブロックするには [ブロック] を選択し、ホワイトリストに登録されているドメイン**/**アドレスからのクエリを許可するには [許可のみ] を選択します。
   c)「ドメイン名**/IP** アドレス」ボックスに、ドメイン名、IP アドレス、または IP アドレス範囲を入力します。エントリはカンマで区切ります。
   注記:「詳細オプション」**(Advanced Options**) を選択した場合は、「次で始まる」、「次を含む」(contains)、および「次で終わる」(ends with) オプションを使用して条件を設定できます。
   たとえば、「image」で始まる、または「.co.ru」で終わる、または「モバイルサイト」を含む DNS クエリをブロックする条件を設定できます。

4. [**Submit**] をクリックします。

ランダムなサブドメイン攻撃を防ぐ

ランダムなサブドメイン攻撃では、正当なドメインのランダムで存在しないサブドメインにクエリが送信されます。この操作により、DNS リゾルバとサーバーの負荷が増加します。その結果、過負荷になり、減速する可能性があります。

ランダムサブドメイン攻撃を防止するオプションは、指定された長さを超える DNS クエリをドロップするように DNS レスポンダーに指示します。

example.com は自分が所有するドメイン名であるため、解決リクエストはお客様の DNS サーバーに届くと仮定します。攻撃者は example.com にランダムなサブドメインを追加して、リクエストを送信することができます。指定したクエリの長さと FQDN に基づいて、ランダムクエリがドロップされます。

たとえば、クエリが www.image987trending.example.com の場合、クエリの長さが 20 に設定されている場合、クエリは削除されます。

GUI を使用した **DNS** クエリの長さの指定

1. [構成**] > [**セキュリティ**] > [DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3.「**DNS** セキュリティプロファイルの追加」ページで、次の操作を行います。

   a)「ランダムサブドメイン攻撃の防止」を拡張。
   b) クエリの長さの数値を入力します。

4. [**Submit**] をクリックします。

キャッシュをバイパスする

攻撃時には、すでにキャッシュされているデータを保護する必要があります。キャッシュを保護するために、特定のドメイン、レコードタイプ、またはレスポンスコードに対する新しいリクエストは、キャッシュされる代わりにオリジンサーバーに送信できます。

キャッシュのバイパスオプションは、攻撃が検出されたときに、NetScaler アプライアンスに指定されたドメイン、レコードタイプ、またはレスポンスコードのキャッシュをバイパスするように指示します。

GUI を使用して、指定したドメイン、レコードタイプ、またはレスポンスタイプのキャッシュをバイパスする

1. [構成**] > [**セキュリティ**] > [DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3. **[DNS** セキュリティプロファイルの追加] ページで、[キャッシュのバイパス] を展開し、ドメイン名を入力します。必要に応じて、キャッシュをバイパスする必要があるレコードタイプまたはレスポンスタイプを選択します。

   - [ドメイン] をクリックし、ドメイン名を入力します。エントリはカンマで区切ります。
   - [レコードタイプ] をクリックし、レコードタイプを選択します。
   - [レスポンスタイプ] をクリックし、レスポンスタイプを選択します。

4. [**Submit**] をクリックします。

TCP 経由の **DNS** トランザクションを強制

一部の DNS 攻撃は、トランザクションが UDP の代わりに TCP を使用することを強制することで防ぐことができます。たとえば、ボット攻撃中、クライアントは大量のクエリを送信しますが、応答を処理できません。これらのトランザクションに TCP の使用が強制されている場合、ボットは応答を理解できないため、TCP 経由でリクエストを送信できません。

GUI を使用してドメインまたはレコードタイプを **TCP** レベルで強制的に動作させる

1. [構成**] > [**セキュリティ**] > [DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3.「**DNS** セキュリティプロファイルの追加」ページで、「**TCP** 経由での **DNS** トランザクションの適用」を展開し、ドメイン名と/を入力するか、DNS トランザクションを TCP 経由で適用する必要があるレコードタイプを選択します。
   - [ドメイン] をクリックし、ドメイン名を入力します。エントリはカンマで区切ります。
   - [** レコードタイプ **] をクリックし、レコードタイプを選択します。
4. [**Submit**] をクリックします。

DNS 応答にルートの詳細を提供する

一部の攻撃では、NetScaler ADC アプライアンスに構成またはキャッシュされていない無関係なドメインに対して、攻撃者が大量のクエリを送信します。`dnsRootReferral`パラメータが ENABLED の場合、すべてのルートサーバーを公開します。

[DNS レスポンスでルート詳細を提供する] オプションを選択すると、NetScaler ADC アプライアンスは、構成またはキャッシュされていないクエリのルート参照へのアクセスを制限するように指示します。アプライアンスは空白の応答を送信します。

[DNS 応答にルートの詳細を提供する] オプションでは、増幅攻撃を軽減またはブロックすることもできます。dnsrootReferral パラメータが無効の場合、NetScaler ADC 応答にルート紹介が存在しないため、それらは増幅されません。

GUI を使用してルートサーバーへのアクセスを有効または無効にする

1. [構成**] > [**セキュリティ**] > [DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3.「**DNS** セキュリティプロファイルの追加」ページで、次の操作を行います。
   a)「****DNS** レスポンスの「ルートの詳細を入力」を拡張してください **。
   b) [オン] または [オフ] をクリックして、ルートサーバーへのアクセスを許可または制限します。
4. [**Submit**] をクリックします。

システム

August 15, 2023

このセクションでは、NetScaler のシステムレベルの情報を提供します。これには、システムレベルの機能の詳細な説明、機能を使用できるシナリオ、構成手順、および機能をさらに理解するために役立つ例が含まれます。

- 基本操作
- 認証と承認
- TCP 構成
- HTTP 構成
- SNMP
- 監査ログ
- Web サーバーログ
- Call Home
- レポートツール
- CloudBridge Connector
- 高可用性
- TCP 最適化

システムベースのオペレーション

March 20, 2024

次の構成では、NetScaler アプライアンスでシステムベースの操作を実行できます。

NetScaler 構成を表示、保存、およびクリアする方法

NetScaler 構成は、`/nsconfig/ns.conf directory`に格納されます。セッション間で構成を使用できるようにするには、構成を変更するたびに構成を保存する必要があります。

コマンドインターフェイスを使用して実行構成を表示する

コマンドプロンプトで入力します:

```
1 show ns runningConfig
2 <!--NeedCopy-->
```

GUI を使用した実行構成の表示

1. [システム] > [診断] に移動し、[構成の表示] グループで [実行構成] をクリックします。

コマンドインターフェイスを使用して、**2** つの構成ファイルの違いを表示します

コマンドプロンプトで入力します:

```
diff ns config <configfile> <configfile2>
<!--NeedCopy-->
```

GUI を使用して **2** つの設定ファイルの違いを表示します

1. [システム] > [診断] に移動し、[構成の表示] グループ で [構成の違い] をクリックします。

コマンドインターフェイスを使用して **NetScaler** 構成を保存します

コマンドプロンプトで入力します:

```
save ns config
<!--NeedCopy-->
```

GUI を使用して **NetScaler** 構成を保存する

1. [構成] タブの右上隅にある [保存] アイコンをクリックします。

コマンドインターフェイスを使用した保存済み設定の表示

コマンドプロンプトで入力します:

```
show ns ns.conf
<!--NeedCopy-->
```

GUI を使用した保存済み設定の表示

[システム] > [診断] に移動し、[構成の表示] グループで [保存された構成] をクリックします。

コマンドインターフェイスを使用して **NetScaler** 構成をクリアします

NetScaler 構成をクリアするには、次の 3 つのオプションがあります。

基本レベル。基本レベルで設定をクリアすると、次の設定を除くすべての設定がクリアされます。

- `Nsroot`: パスワード
- タイムゾーン

- NTP サーバー
- ADM サーバー接続
- ライセンスファイル情報
- NSIP、MIP、および SNIP
- ネットワーク設定（デフォルトゲートウェイ、VLAN、RHI、NTP、および DNS 設定）
- HA ノード定義
- 機能およびモードの設定
- デフォルトの管理者パスワード (nsroot)

拡張レベル。拡張レベルで設定をクリアすると、次の設定を除くすべての設定がクリアされます。

- NSIP と SNIP
- ネットワーク設定（デフォルトゲートウェイ、VLAN、RHI、NTP、および DNS 設定）
- HA ノード定義

機能およびモードの設定は、デフォルト値に戻ります。

フルレベル。設定をフルレベルでクリアすると、すべての設定が工場出荷時のデフォルト値に戻ります。ただし、NSIP とデフォルトゲートウェイを変更すると、アプライアンスのネットワーク接続が失われる可能性があるため、変更されません。

コマンドプロンプトで入力します：

```
clear ns config -force
<!--NeedCopy-->
```

例：アプライアンスの基本設定を強制的にクリアする場合。

```
clear ns config -force basic
<!--NeedCopy-->
```

GUI を使用して **NetScaler** 構成をクリアします

[システム] > [診断] に移動し、[メンテナンス] グループで [設定のクリア] をクリックし、アプライアンスからクリアする設定レベルを選択します。

保存されていない **NetScaler** 構成のアプライアンスを再起動またはシャットダウンする方法

NetScaler アプライアンスは、使用可能なユーザーインターフェイスからリモートで再起動またはシャットダウンできます。スタンドアロンの NetScaler アプライアンスを再起動またはシャットダウンすると、保存されていない構成 (
最後の save ns config コマンドが発行されてから実行された構成）は失われます。

高可用性設定では、プライマリアプライアンスがリブートまたはシャットダウンされると、セカンダリアプライアンスが引き継ぎ、プライマリになります。古いプライマリの未保存の設定は、新しいプライマリアプライアンスで使用できます。

また、NetScaler ソフトウェアを再起動するだけで、基盤となるオペレーティングシステムを再起動しないで、アプライアンスを再起動することもできます。これはウォームリブートと呼ばれます。たとえば、新しいライセンスを追加したり、IP アドレスを変更したりすると、NetScaler アプライアンスをウォームリブートしてこれらの変更を行うことができます。

> 注:
>
> ウォームリブートは、スタンドアロンの NetScaler アプライアンスでのみ実行できます。

コマンドインターフェイスを使用してアプライアンスを再起動します

コマンドプロンプトで入力します:

```
reboot [-warm]
<!--NeedCopy-->
```

GUI を使用して **NetScaler** アプライアンスを再起動します

1. 設定ページで、[**Reboot**] をクリックします。
2. 再起動を促すメッセージが表示されたら、[設定の保存（**Save configuration** ）] を選択して、設定が失われないようにします。

> 注:
>
> ウォームリブートを選択すると、ウォームリブートを実行できます。

コマンドインターフェイスを使用してアプライアンスをシャットダウンする

シェルプロンプトで、次のように入力します:

- `shutdown -p now`: ソフトウェアをシャットダウンし、NetScaler をオフにします。NetScaler MPX を再起動するには、AC 電源スイッチを押します。NetScaler VPX を再起動するには、VPX インスタンスを再起動します。
- `shutdown -h now`: ソフトウェアをシャットダウンし、NetScaler スイッチを入れたままにします。任意のキーを押して NetScaler を再起動します。このコマンドは、NetScaler をオフにしません。したがって、AC 電源をオフにしたり、AC 電源ケーブルを取り外したりしないでください。

> 注:

> NetScaler GUI を使用してアプライアンスをシャットダウンすることはできません。

システムクロックをネットワーク上のサーバーと同期させる方法

NetScaler アプライアンスを設定して、ローカルの時刻を、NTP（Network Time Protocol：ネットワークタイムプロトコル）サーバーの時刻と同期することができます。これにより、NetScaler のクロックの設定は、ネットワーク上のほかのサーバーと同じ日付と時刻になります。

アプライアンスでクロック同期を設定するには、GUI またはコマンドラインインターフェイスから NTP サーバエントリを ntp.conf ファイルに追加するか、または ntp.conf ファイルを手動で変更してから NTP デーモン（NTPD）を起動します。アプライアンスが再起動、アップグレード、またはダウングレードされても、クロック同期の設定は変更されません。ただし、高可用性セットアップでは、構成はセカンダリ NetScaler に伝播されません。

NetScaler GUI を使用すると、初回ユーザー（FTU）画面でクロック同期に必要なタイムゾーンと NTP サーバーの IP アドレスを構成できます。

> 注：
>
> ローカル NTP サーバを持っていない場合は、公式 NTP サイト<http://www.ntp.org>の Public Time Servers List の下に、パブリック、オープンアクセス、NTP サーバのリストがあります。パブリック NTP サーバーを使用するように NetScaler を構成する前に、「契約ルール」ページ（すべてのパブリックタイムサーバーページに含まれるリンク）を必ずお読みください。
>
> NetScaler リリース 11 では、NTP バージョンが 4.2.6p3 から 4.2.8p2 に更新されました。

前提要件

クロック同期を設定するには、次のエンティティを設定する必要があります。

1. NTP サーバ
2. NTP 同期。

コマンドインターフェイスを使用して **NTP** サーバを追加する

コマンドプロンプトで次のコマンドを入力して、NTP サーバを追加し、構成を確認します。

- `add ntp server (<serverIP> | <serverName>)[-minpoll <positive_integer>] [-maxpoll <positive_integer>]`
- `show ntp server`

例：

```
add ntp server 10.102.29.30 -minpoll 6 -maxpoll 11
<!--NeedCopy-->
```


GUI を使用して **NTP** サーバを追加する

[システム] > [**NTP** サーバ] に移動し、NTP サーバを作成します。

コマンドインターフェイスを使用した **NTP** 同期の有効化

NTP 同期を有効にすると、NetScaler は NTP デーモンを起動し、ntp.conf ファイルの NTP サーバーエントリを使用してローカル時刻設定を同期します。アプライアンスの時刻をネットワーク内の他のサーバと同期させたくない場合は、NTP 同期を無効にして、NTP デーモン（NTPD）を停止できます。

コマンドプロンプトで、次のコマンドのいずれかを入力します:

```
enable ntp sync
<!--NeedCopy-->
```

GUI を使用した **NTP** 同期の有効化

[システム] > [**NTP** サーバ] に移動し、[アクション] をクリックして [**NTP** 同期] を選択します。

GUI を使用して **ntp.conf** ファイルを編集するようにクロック同期を構成する

1. コマンドラインインターフェイスにログオンします。
2. シェルプロンプトに切り替えます。
3. /nsconfig directoryにすでにntp.confファイルが含まれている場合を除き、/etc/ntp.confファイルを/nsconfig/ntp.confにコピーします。
4. 追加する NTP サーバごとに、次の 2 行を/nsconfig/ntp.conf ファイルに追加する必要があります。

   ```
   server <IP address for NTP server> iburst

   restrict <IP address for NTP server> mask <netmask> nomodify
      notrap nopeer noquery
   <!--NeedCopy-->
   ```

   > 注:
   >
   > セキュリティ上の理由から、サーバエントリごとに対応する restrict エントリがあるはずです。

   例

   次の例では、管理者が既存の NTP エントリを「コメントアウト」するために # 文字を挿入し、エントリを追加しています。


```
#server 1.2.3.4 iburst

#restrict 1.2.3.4 mask 55.255.255.255 nomodify notrap nopeer
    noquery

server 10.102.29.160 iburst

restrict 10.102.29.160 mask 255.255.255.255 nomodify notrap nopeer
    noquery
<!--NeedCopy-->
```

5. /nsconfigディレクトリにrc.netscalerという名前のファイルがない場合は、ファイルを作成します。
6. 次のエントリを/nsconfig/rc.netscaler: /bin/sh /etc/ntpd_ctl full_start に追加

   このエントリは、ntpd サービスを開始し、ntp.conf ファイルをチェックし、メッセージを /var/log ディレクトリに記録します。

   このプロセスは、NetScaler が再起動されるたびに実行されます。
7. NetScaler アプライアンスを再起動して、クロック同期を有効にします。または、アプライアンスを再起動せずに時刻同期プロセスを開始するには、シェルプロンプトで次のコマンドを入力します。

```
rm /etc/ntp.conf
ln -s /nsconfig/ntp.conf /etc/ntp.conf
/bin/sh /etc/ntpd_ctl full_start
<!--NeedCopy-->
```

アイドル状態のクライアント接続のセッションタイムアウトを構成する方法

セッションタイムアウト間隔は、セッション（GUI、CLI、または API）が使用されていないときにアクティブのままになる時間を制限するために提供されます。NetScaler の場合、システムセッションタイムアウトは次のレベルで構成できます。

- ユーザーレベルのタイムアウト。特定のユーザーに適用されます。

インターフェースタイプ	タイムアウト設定
GUI	[システム] **>** [ユーザ管理] **>** [ユーザ] に移動し、ユーザを選択して、ユーザのタイムアウト設定を編集します。
CLI	コマンドプロンプトで、次のコマンドを入力します。 `set system user <name> -timeout <secs>`

- ユーザーグループレベルのタイムアウト。グループ内のすべてのユーザーに適用されます。

インターフェースタイプ	タイムアウト設定
GUI	[システム] **>** [ユーザー管理] **>** [グループ] に移動し、グループを選択して、グループのタイムアウト設定を編集します。
CLI	コマンドプロンプトで、次のコマンドを入力します。 `set system group <groupName> -timeout <secs>`

- グローバルシステムタイムアウト。タイムアウトが設定されていないグループのすべてのユーザおよびユーザに適用されます。

インターフェースタイプ	タイムアウト設定
GUI	[システム] **>** [設定] に移動し、[グローバルシステム設定の変更] をクリックして、必要に応じてタイムアウト値を更新します。
CLI	コマンドプロンプトで、次のコマンドを入力します。 `set system parameter -timeout <secs>`

ユーザに指定されたタイムアウト値は、最も優先度が高くなります。ユーザに対してタイムアウトが設定されていない場合は、メンバグループに設定されたタイムアウトが考慮されます。グループに対してタイムアウトが指定されていない（またはユーザがグループに属していない）場合は、グローバルに設定されたタイムアウト値が考慮されます。timeout がどのレベルでも設定されていない場合、デフォルト値の 900 秒がシステムセッションタイムアウトとして設定されます。

また、アクセスするインターフェイスごとにタイムアウト期間を指定することもできます。ただし、特定のインターフェイスに指定されたタイムアウト値は、インターフェイスにアクセスしているユーザに設定されたタイムアウト値に制限されます。たとえば、タイムアウト値が 20 分のユーザー「publicadmin」について考えてみましょう。インターフェイスにアクセスする場合、ユーザは 20 分以内のタイムアウト値を指定する必要があります。

> 注:
>
> タイムアウトを制限として指定することで（CLI で *restrictedTimeout* パラメータを指定して）タイムアウトの最小値と最大値を常にチェックするように選択できます。このパラメーターは、タイムアウト値が制限されなかった以前の NetScaler バージョンを考慮して提供されます。

- 有効の場合、設定可能な最小タイムアウト値は 5 分（300 秒）で、最大値は 1 日（86400 秒）です。タイムアウト値がすでに 1 日より大きい値に設定されている場合、このパラメータを有効にすると、変更を求めるプロ

ンプトが表示されます。値を変更しない場合、タイムアウト値は次回の再起動時にデフォルトのタイムアウト期間である 15 分（900 秒）に自動的に再構成されます。設定したタイムアウト値が 5 分未満の場合も同じことが起こります。

- 無効にすると、設定されたタイムアウト期間が考慮されます。
- 各インターフェイスのタイムアウト時間**:**

インターフェースタイプ	タイムアウト設定
CLI	次のコマンドを使用して、コマンドプロンプトでタイムアウト値を指定します。 `set cli mode -timeout <secs>`
API	ログインペイロードにタイムアウト値を指定します。

システムの日付と時刻を設定して時計をタイムサーバと同期させる方法

システムの日付と時刻を変更するには、基盤となる FreeBSD OS へのシェルインターフェースを使用する必要があります。ただし、システムの日付と時刻を表示するには、コマンドラインインターフェイスまたは GUI を使用できます。

コマンドインターフェイスを使用してシステムの日付と時刻を表示する

コマンドプロンプトで入力します:

```
1 show ns config
2 <!--NeedCopy-->
```

GUI を使用してシステムの日付と時刻を表示する

[システム] に移動し、[システム情報] タブを選択してシステム日付を表示します。

内部サービス用の **HTTP** および **HTTPS** 管理ポートの設定方法

NetScaler アプライアンスのシングル IP モードの展開では、単一の IP アドレスが NSIP、SNIP、および VIP アドレスとして使用されます。この単一の IP アドレスは、異なるポート番号を使用して、NSIP、SNIP、および VIP アドレスとして機能します。

ポート番号 80 と 443 は、HTTP および HTTPS サービスの既知のポートです。以前は、NetScaler IP アドレス（NSIP）のポート 80 および 443 は、内部 HTTP および HTTPS 管理サービスの専用ポートでした。これらのポートは内部サービス用に予約されているため、VIP アドレスから HTTP および HTTPS データサービスを提供するため

に、これらの既知のポートを使用することはできません。VIP アドレスは、シングル IP モード展開の NSIP アドレスと同じアドレスを持っています。

この要件に対処するために、ポート 80 と 443 以外の（NSIP アドレスの）内部 HTTP および HTTPS 管理サービスのポートを構成できるようになりました。

NetScaler MPX、VPX、および CPX アプライアンスの内部 HTTP および HTTPS 管理サービスのデフォルトのポート番号を以下に示します。

- NetScaler MPX および VPX アプライアンス：80（HTTP）および 443（HTTPS）アプライアンス
- NetScaler CPX アプライアンス：9080（HTTP）および 9443（HTTPS）

コマンドインターフェイスを使用して **HTTP** および **HTTPS** 管理ポートを設定します

HTTP および HTTPS 管理サービスをサポートするために、NetScaler アプライアンスで HTTP および HTTPS ポートを任意の値に構成できます。ただし、デフォルトでは、NetScaler アプライアンスは HTTP および HTTPS 接続に 80 ポートと 443 ポートを使用します。

コマンドプロンプトで入力します：

```
set ns param  - mgmtHttpPort<port>
<!--NeedCopy-->
```

例：

```
set ns param -mgmtHttpPort 2000
<!--NeedCopy-->
```

コマンドインターフェイスを使用して HTTPS ポートを設定するには

コマンドプロンプトで入力します：

```
set ns param  - mgmtHttpsPort<port>
<!--NeedCopy-->
```

例：

```
set ns param -mgmtHttpsPort 3000
<!--NeedCopy-->
```

GUI を使用して **HTTP** および **HTTPS** 管理ポートを設定します

HTTP および HTTPS ポート値を設定するには、以下の手順に従います。

1. [システム] > [設定**] > [**グローバルシステム設定の変更] に移動します。
2. グローバルシステム設定の構成パラメータページの [その他の設定] セクションで、次のパラメータを設定します。

a) 管理 HTTP ポート。ポート値を 2000 に設定します。デフォルト = 80、最小 = 1、最大 = 65534。
b) 管理 HTTPS ポート。ポート値を 3000 に設定します。デフォルト = 443、最小 = 1、最大 = 65534。

NetScaler GUI、**NetScaler CLI**、または **NetScaler NITRO API** を使用して、内部 **HTTP GUI** サービスを構成します

NetScaler アプライアンスでは、/etc/httpd.conf は NetScaler GUI への接続を管理する内部 HTTP GUI サービスの構成ファイルです。

httpd.confファイルを使用して内部 HTTP GUI サービスを構成する代わりに、NetScaler GUI、NetScaler CLI、または NetScaler NITRO API を使用できるようになりました。たとえば、NetScaler CLI を使用して、内部 HTTP GUI サービスに一度に接続できるクライアントの最大数を変更できます。

内部 HTTP GUI サービスの名前形式は次のとおりです。**nshttpd-gui-80**<loop back IP address>

NetScaler サービスコマンド操作を使用して、内部 HTTP GUI サービスを構成します。

CLI を使用して内部 **HTTP GUI** サービスを変更するには、次の手順を実行します。

- set service コマンドを使用します。詳細については、「set service」を参照してください。
- show service コマンドを使用して、設定を確認します。詳細については、show serviceを参照してください。

設定例:

次の設定例では、内部 HTTP GUI サービスのmaxClientパラメータは 300 に設定されています。

```
> sh service nshttpd-gui-127.0.0.1-80
                nshttpd-gui-127.0.0.1-80 (127.0.0.1:80) - HTTP
                State: UP
                Last state change was at Wed Mar 16 20:16:16 2022
                Time since last state change: 0 days, 22:31:00.970
                Server Name: #ns-internal-127.0.0.1#
                Server ID : None                Monitor Threshold : 0
                Max Conn: 0       Max Req: 0          Max Bandwidth: 0
                   kbits
                Use Source IP: NO
                Client Keepalive(CKA): NO
                Monitoring Owner: 0
                Access Down Service: NO
                TCP Buffering(TCPB): NO
                HTTP Compression(CMP): NO
                Idle timeout: Client: 180 sec        Server: 360 sec
                Client IP: ENABLED cip-header
                Cacheable: NO
                SC: ???
                SP: OFF
                Down state flush: DISABLED
                Monitor Connection Close : NONE
                Appflow logging: DISABLED
```



```
                TCP profile name: nstcp_internal_apps
                HTTP profile name: nshttp_default_internal_apps
                Process Local: DISABLED
                Traffic Domain: 0

Done

> set service nshttpd-gui-127.0.0.1-80 -maxclient 300
Done

> sh service nshttpd-gui-127.0.0.1-80
                nshttpd-gui-127.0.0.1-80 (127.0.0.1:80) - HTTP
                State: UP

                ...

                Max Conn: 300  Max Req: 0          Max Bandwidth: 0
                   kbits

                ...

Done

<!--NeedCopy-->
```

コマンドインターフェイスを使用してメモリ回復をトリガーする

コマンドラインインターフェイスからメモリ回復をトリガーできます。

コマンドプロンプトで、次のコマンドを入力します:

```
start ns memrecovery [-percentage <positive_integer>]
```

例:

```
start nsmemrecovery -percentage 30
```

実際にリカバリされたメモリ量を確認するには、コマンドプロンプトで次のコマンドを使用します。

```
stat system memory
```

データ処理と監視のために追加の管理 **CPU** を割り当てる方法

NetScaler MPX アプライアンスの構成と監視のパフォーマンスを向上させる必要がある場合は、アプライアンスのパケットエンジンプールから追加の管理 CPU を割り当てることができます。この機能は、特定の NetScaler MPX モデルと、NetScaler SDX アプライアンスで実行される VPX インスタンスを除くすべての VPX モデルでサポートされています。これは、統計システム CPU および stat システムコマンドの出力に影響します。

サポートされている NetScaler MPX モデル:

- 25xxx
- 22xxx
- 14xxx
- 115xx
- 15xxx
- 26xxx

> 注:
>
> 20 コアを超える NetScaler MPX 26xxx モデルの場合、必須の追加管理 CPU 機能がデフォルトで有効になっています。NetScaler VPX モデルの場合、この機能を有効にするには、少なくとも 12 の vCPU をサポートするライセンスが必要です。

コマンドインターフェイスを使用して追加の管理 **CPU** を割り当てる

コマンドプロンプトで、次のコマンドのいずれかを入力します:

- `enable extramgmtcpu`
- `disable extramgmtcpu`

> 注:
>
> この機能を有効または無効にすると、NetScaler アプライアンスは、変更を有効にするためにアプライアンスを再起動するように警告を表示します。

追加の管理 CPU の設定済みおよび有効な状態を表示します。

コマンドプロンプトで入力します:

```
1 show extramgmtcpu
2 <!--NeedCopy-->
```

例:

```
1 > show extramgmtcpu
2 ConfiguredState:  ENABLED EffectiveState:  ENABLED
3 <!--NeedCopy-->
```

> 注:
>
> この例では、アプライアンスを再起動する前に show コマンドを入力します。

GUI を使用して追加の管理 **CPU** を割り当てる

GUI を使用して追加の管理 CPU を割り当てるには、[システム] > [設定] に移動し、[追加管理 **CPU** の設定] をクリックします。［構成済みの状態］ドロップダウンメニューから［有効］を選択し、［**OK**］を選択します。

CPU 使用率を確認するには、[システム] **>** [設定] **>** [ダッシュボード] に移動します。

NITRO API を使用して追加の管理 **CPU** を構成する

次の NITRO 方式と形式を使用して、追加の管理 CPU を有効化、無効化、および表示します。

追加の管理 **CPU** を有効にするには、次の手順を実行します。

```
HTTP Method: POST

URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu?action=enable

Payload: {
 "systemextramgmtcpu":{
  }
  }


curl -v -X POST -H "Content-Type: application/json" -u nsroot:nsroot
    http://10.102.201.92/nitro/v1/config/systemextramgmtcpu?action=
    enable -d '{
 "systemextramgmtcpu":{
  }
  }
 '
<!--NeedCopy-->
```

追加の管理 CPU を無効にするには

```
HTTP Method: POST
URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu?action=disable
Payload: {
 "systemextramgmtcpu":{
  }
  }

curl -v -X POST -H "Content-Type: application/json" -u nsroot:nsroot
    http://10.102.201.92/nitro/v1/config/systemextramgmtcpu?action=
    disable -d '{
 "systemextramgmtcpu":{
  }
  }
 '
<!--NeedCopy-->
```

追加の管理 CPU を表示するには

```
HTTP Method: GET
URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu
<!--NeedCopy-->
```

例:

```
curl -v -X GET -H "Content-Type: application/json" -u nsroot:nsroot
    http://10.102.201.92/nitro/v1/config/systemextramgmtcpu
```



```
<!--NeedCopy-->
```

管理 **CPU** の追加前と後の統計とモニタリング

次の例は、追加の管理 CPU を追加する前と後の stat system CPU コマンドと stat system コマンドの出力の違いを示しています。

```
stat system cpu
<!--NeedCopy-->
```

このコマンドは、CPU の統計情報を表示します。

次に、サポートされているモデルのいずれかで追加の管理 CPU を追加する前の出力例を示します。

例

```
> stat system cpu

    CPU statistics

    ID          Usage

    8               1

    7               1

    11              2

    1               1

    6               1

    9               1

    3               1

    5               1

    4               1

    10              1

    2               1
<!--NeedCopy-->
```

次に、同じ MPX アプライアンスで管理用 CPU を追加した後の出力を示します。

```
    > stat system cpu

    CPU statistics

```



```
    ID          Usage

    9               1

    7               1

    5               1

    8               1

    11              2

    10              1

    6               1

    4               1

    3               1

    2               1
<!--NeedCopy-->
```

```
stat system
<!--NeedCopy-->
```

このコマンドは CPU 使用率を表示します。次の例では、サポートされているモデルのいずれかで追加の管理 CPU を追加する前の出力は次のようになります。

Mgmt Additional-CPU usage (%) 0.00

例

```
    > stat system

    NetScaler Executive View

    System Information:

    Up since         Wed Oct 11 11:17:54 2017

    /flash Used (%)                          0

    Packet CPU usage (%)                 1.30

    Management CPU usage (%)             4.00

    Mgmt CPU0 usage (%)                  4.00

    Mgmt Additional-CPU  usage (%)       0.00

    Memory usage (MB)                    2167

```



```
    InUse Memory (%)                    5.76

    /var Used (%)                          0
<!--NeedCopy-->
```

次の例では、同じ MPX アプライアンスで管理用 CPU を追加した後の出力は次のようになります。

Mgmt Additional-CPU usage (%) 0.80

```
> stat system


NetScaler Executive View

System Information:

Up since        Wed Oct 11 11:55:56 2017

/flash Used (%)                        0

Packet CPU usage (%)                1.20

Management CPU usage (%)            5.70

Mgmt CPU0 usage (%)                10.60

Mgmt Additional-CPU  usage (%)      0.80

Memory usage (MB)                   1970

InUse Memory (%)                    5.75

/var Used (%)                          0

<!--NeedCopy-->
```

失われた構成を回復するためにアプライアンスをバックアップおよび復元する方法

アプライアンスが破損した場合やアップグレードが必要な場合は、システム設定をバックアップできます。バックアップ手順は、CLI または GUI インターフェイスのいずれかを使用して実行されます。アプライアンスでは、外部ソースからバックアップファイルをインポートすることもできます。ただし、これは GUI インターフェイスを介してのみ実行でき、CLI インターフェイスによるサポートはありません。

確認事項

アプライアンスをバックアップおよび復元するときは、次の点を覚えておく必要があります。

- 新しいプラットフォームでのネットワーク構成のサポートが必要です。

- 新しいプラットフォームビルドは、バックアップファイルまたはそれ以降のバージョンと同じである必要があります。

NetScaler アプライアンスをバックアップする

データとバックアップの要件に応じて、「基本」バックアップまたは「フル」バックアップを作成できます。

- 基本バックアップ。この種類のバックアップは、常に変化するファイルをバックアップする場合に行うことができます。バックアップできるファイルを次の表に示します。

基本的なバックアップの詳細については、「表 」のトピックを参照してください。

- フルバックアップ基本バックアップでバックアップされるファイルに加えて、完全バックアップではファイルの更新頻度は低くなります。「フル」バックアップオプションを使用したときにバックアップされるファイルは次のとおりです。

ディレクトリ	サブディレクトリまたはファイル
nsconfig	ssl, *license*, fips*
/var/	netscaler/ssl/、 *wi/java_home/jre/lib/security/cacerts/*、 wi/java_home/lib/security/cacerts/*

バックアップされたデータは、圧縮された TAR ファイルとして/var/ns_sys_backup/ディレクトリに格納されます。ディスク領域が利用できないことによる問題を回避するために、このディレクトリには最大 50 個のバックアップファイルを保存できます。`rm system backup`コマンドを使用して、既存のバックアップファイルを削除し、さらにバックアップを作成できます。

> 注:
>
> バックアップ操作が進行中の場合は、設定に影響を与えるコマンドを実行しないでください。
>
> バックアップが必要なファイルが使用できない場合、そのファイルはスキップされます。

コマンドインターフェイスを使用して NetScaler アプライアンスをバックアップする

NetScaler コマンドインターフェイスを使用して NetScaler アプライアンスをバックアップするには、以下の手順に従います。

コマンドプロンプトで、次の操作を行います:

1. NetScaler 構成を保存します。


```
save ns config
<!--NeedCopy-->
```

1. バックアップファイルを作成します。

```
create system backup [<fileName>] -level <basic | full> -comment <
    string>
<!--NeedCopy-->
```

> 注:
>
> ファイル名が指定されていない場合、アプライアンスは次の命名規則で TAR ファイルを作成します。backup_<level>_<nsip_address>_<date-timestamp>.tgz。

例**:** バックアップファイルのデフォルトの命名規則を使用して、完全なアプライアンスをバックアップする場合。

```
> create system backup -level full
<!--NeedCopy-->
```

1. バックアップファイルが作成されたことを確認します。

```
show system backup
<!--NeedCopy-->
```

fileName パラメータを使用すると、特定のバックアップファイルのプロパティを表示できます。

コマンドインターフェイスを使用して **NetScaler** アプライアンスを復元する

> 重要:
>
> バックアップファイルの名前を変更または変更すると、アプライアンスを正常に復元できません。

アプライアンスを復元すると、復元操作によって/var/ns_sys_backup/ディレクトリからバックアップファイルが解凍されます。ファイルが解凍されると、ファイルはそれぞれのディレクトリにコピーされます。

コマンドインターフェイスを使用して、ローカルバックアップファイルから **NetScaler** を復元します

> 注:
>
> 以前の構成を復元する前に、現在の構成をバックアップすることをお勧めします。ただし、restore コマンドで現在の設定のバックアップを自動的に作成しない場合は、 - skipBackup パラメータを使用します。

コマンドプロンプトで、次の操作を行います:

1. アプライアンスで使用可能なバックアップファイルのリストを取得します。


```
show system backup
<!--NeedCopy-->
```

2. バックアップファイルの 1 つを指定して、アプライアンスを復元します。

   ```
   restore system backup <filename> [-skipBackup]
   ```

   例**:** アプライアンスの完全バックアップを使用して復元するには

   ```
   > restore system backup backup_full_<nsip_address>_<date-timestamp>.tgz
   ```

3. アプライアンスを再起動します。

   ```
   reboot
   ```

GUI を使用して **NetScaler** アプライアンスをバックアップおよび復元する

1. [システム] > [バックアップと復元] に移動します。
2. [バックアップ**/**インポート**]** をクリックして、プロセスを開始します。
3. [バックアップ**/**インポート] ページで、[作成] を選択し、次のパラメータを設定します。
   a) ファイル名。アプライアンスのバックアップファイルの名前。
   b) [レベル]。バックアップレベルとして「基本」または「フル」を選択します。
   c) [コメント]。バックアップの簡単な説明を入力します。
4. [バックアップ] をクリックします。
5. バックアップをインポートする場合は、[インポート] を選択する必要があります。
6. バックアップが完了したら、ファイルを選択して [ダウンロード] をクリックします。
7. 復元するには、バックアップファイルを選択して [復元] をクリックします。
8. [復元] ページで、バックアップファイルの詳細を確認し、[復元] をクリックします。
9. 復元した後、アプライアンスを再起動する必要があります。

NetScaler インスタンスをバックアップおよび復元する方法について詳しくは、「NetScaler Console を使用したバックアップと復元」トピックを参照してください 。

SDX アプライアンスのバックアップと復元方法の詳細については、SDX アプライアンスのバックアップと復元を参照してください

システムバックアップで実行される操作の詳細については、「システムバックアップ 」トピックを参照してください。

アプライアンスの問題を解決するためのテクニカルサポートバンドルを生成する方法

NetScaler アプライアンスに関する問題の分析と解決に役立つように、アプライアンスでテクニカルサポートバンドルを生成し、そのバンドルを Citrix のテクニカルサポートに送信できます。NetScaler テクニカルサポートバンドルは、システム構成データと統計情報の圧縮された tar アーカイブです。バンドルを生成する NetScaler アプライアンスから次のデータを収集します。

- 設定ファイル。/flash/nsconfig ディレクトリ内のすべてのファイル。
- **Newnslog** ファイル。現在実行中の newnslog といくつかの以前のファイル。アーカイブファイルのサイズを最小化するために、newnslogコレクションは 500 MB、6 ファイル、7 日のいずれか早い方に制限されます。古いデータが必要な場合は、手動で収集する必要があります。
- ログファイル。/var/log/messages 、*/var/log/ns.log* 内のファイル、および /var/log および /var/nslog の下のその他のファイル。
- アプリケーションコアファイル。先週以内に /var/core ディレクトリに作成されたファイル (存在する場合)。
- いくつかの **CLI** の **show** コマンドの出力。
- いくつかの **CLI** 統計コマンドの出力。
- **BSD** シェルコマンドの出力。

テクニカルサポートバンドルは、NetScaler アプライアンスの次の場所にアーカイブに保存されます。

```
1 /var/tmp/support/support.tgz
2 <!--NeedCopy-->
```

このパスは、簡単にアクセスできるように最新のコレクターへのシンボリックリンクです。完全なファイル名は、デプロイメントトポロジによって異なりますが、一般的に次のような形式になります。

```
1 collector_<P/S>_<NS IP>_<DateTime>.tgz.
2 <!--NeedCopy-->
```

1 つのコマンドでテクニカルサポートバンドルを生成し、それを Citrix テクニカルサポートサーバーに安全にアップロードできます。NetScaler 13.1-50.x リリース以降、テクニカルサポートバンドルをアップロードする前に認証トークンを生成する必要があります。以前は、Citrix のユーザー名とパスワードを使用してテクニカルサポートバンドルをアップロードしていました。

認証トークンを生成するには:

1. ブラウザを起動し、次の URL (https://cis.citrix.com/auth/api/create_identity_v2/?expiration=3600) を入力します。
2. 多要素認証を使用してログインします。

   > 注:
   >
   > 多要素認証への登録方法については、https://support.citrix.com/article/CTX461297/how-to-enroll-into-multi-factor-authentication-mfa を参照してください。

3. 「コピー」をクリックして、画面に表示されている認証トークンをコピーします。トークンは 3600 秒 (1 時間) 有効です。トークンの最大許容長は 1023 文字です。
4. 認証トークンをコピーしたら、CLI または GUI を使用してファイルをアップロードします。

CLI を使用してテクニカルサポートバンドルをアップロードするには、次のコマンドを実行します。

```
showtechsupport -upload -file <string> -description <string> -authtoken
    <string>
<!--NeedCopy-->
```

例:

```
showtechsupport -upload -file /var/tmp/support/callhome/
   collector_callhome_P_10.102.168.52_14Mar2022_12_50.tar.gz -
   description "test upload" -authtoken
   eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.
   eyJzdWIiOiJDSVMiLCJpc3MiOiJidHMuY2lzLmNpdHJpeC5jb20iLCJqdGkiOiJkNDI4N
   *******
<!--NeedCopy-->
```

GUI を使用してテクニカルサポートバンドルをアップロードするには:

1. [設定] **>** [システム] **>** [診断] に移動します。
2. 「テクニカルサポートツール」セクションで、「サポートファイルを生成」をクリックします。
3. **Scope** オプションを使用して、現在のノード、すべてのクラスターノード、または指定されたパーティションのデータを収集するかどうかを指定します。
4. 「コレクターアーカイブをアップロード」を選択します。
5. [マイ **Citrix** アカウント] セクションの [**Citrix** 認証トークン] フィールドに認証トークンを入力します。

NetScaler アプライアンスに直接インターネット接続がない場合は、プロキシサーバーを使用して、テクニカルサポートバンドルを Citrix テクニカルサポートサーバーに直接アップロードできます。プロキシ文字列の基本的な形式は次のとおりです。

```
proxy_IP:<proxy_port>
<!--NeedCopy-->
```

プロキシサーバーが認証を必要とする場合、形式は次のとおりです。

```
username:password@proxsy_IP:<proxy_port>
<!--NeedCopy-->
```

> 注:
>
> 高可用性ペアの NetScaler アプライアンスの場合は、2 つのノードのそれぞれでテクニカルサポートバンドルを生成する必要があります。
>
> クラスタセットアップの NetScaler アプライアンスの場合、各ノードで個別にテクニカルサポートバンドルを生成することも、クラスタ IP アドレスを使用してすべてのノードに対してより小さな省略アーカイブを生成す

> ることもできます。
>
> NetScaler 管理パーティションの場合は、デフォルトの管理パーティションからテクニカルサポートバンドルを生成する必要があります。特定のパーティションのテクニカルサポートバンドルを取得するには、テクニカルサポートバンドルを生成するパーティションの名前を指定する必要があります。パーティションの名前を指定しない場合、データはすべての管理パーティションから収集されます。

コマンドインターフェイスを使用して **NetScaler** テクニカルサポートバンドルを生成する

コマンドプロンプトで入力します:

```
1 show techsupport [-scope <scope> <partitionName>] [-upload [-proxy <
     string>] [-casenumber <string>] [-file <string>] [-description <
     string>] [-userName <string> -password ]]
2 <!--NeedCopy-->
```

シニアいいえ	タスク	コマンド
1	テクニカルサポートバンドルを生成して、Citrix のテクニカルサポートサーバーにアップロードします。	show techsupport –upload –userName account1 –password xxxxxxx
2	テクニカルサポートバンドルを生成し、プロキシサーバー経由で Citrix のテクニカルサポートサーバーにアップロードします	show techsupport –upload –proxy 1.1.1.1:80 –userName account1 –password xxxxxxx
3	既存のテクニカルサポートバンドルを Citrix のテクニカルサポートサーバーにアップロードします。	show techsupport –upload -file,/var/tmp/support/collector_P_10.102.29 –userName account1 –password xxxxxxx
4	クラスタ設定のすべてのノードに対して、小さくて省略されたアーカイブを生成します。クラスター IP アドレスを使用してこのコマンドを実行します。	show techsupport –scope CLUSTER
5	管理者パーティションに固有のテクニカルサポートバンドルを生成します。このコマンドをデフォルトの admin パーティションで実行します。	show techsupport –scope PARTITION partition1

洞察分析のために **SDX** および **VPX** アプライアンスからテクニカルサポートバンドルを収集する方法

NetScaler アプライアンスには、ログファイルを収集するメカニズムが組み込まれています。ログファイルは、分析のために Citrix Insight Services に送信されます。

> 注:
>
> すべての手順は、ソフトウェアリリース 9.2 以降に適用されます。

NetScaler MPX および **VPX** アプライアンスからテクニカルサポートバンドルをダウンロードする

NetScaler GUI を使用してコレクタファイルを実行するには、次の手順を完了する必要があります。

> 注**:**
>
> この手順は、ソフトウェアリリース 9.2 以降に適用されます。

1. [システム] > [診断] に移動します。
2. [テクニカルサポートツール] セクションで、[サポートファイルを生成] リンクをクリックします。
3. [テクニカルサポート（**Tech** Support）] ページで、次のパラメータを設定します。
    a) スコープ。1 つ以上のノードからデータを収集します。
    b) パーティション。パーティションの名前。
    c) Citrix のテクニカルサポートロードオプション。プロキシサーバ、サービスケース番号、コレクタアーカイブファイル名、テクニカルサポートバンドルをアップロードするためのアーカイブファイルの簡単な説明など、すべてのオプションを設定します。
    d) Citrix アカウント。Citrix 資格情報を入力します。
4. [実行] をクリックします。
5. テクニカルサポートバンドルが生成されます。
6. [はい] をクリックして、テクニカルサポートバンドルをローカルデスクトップにダウンロードします。

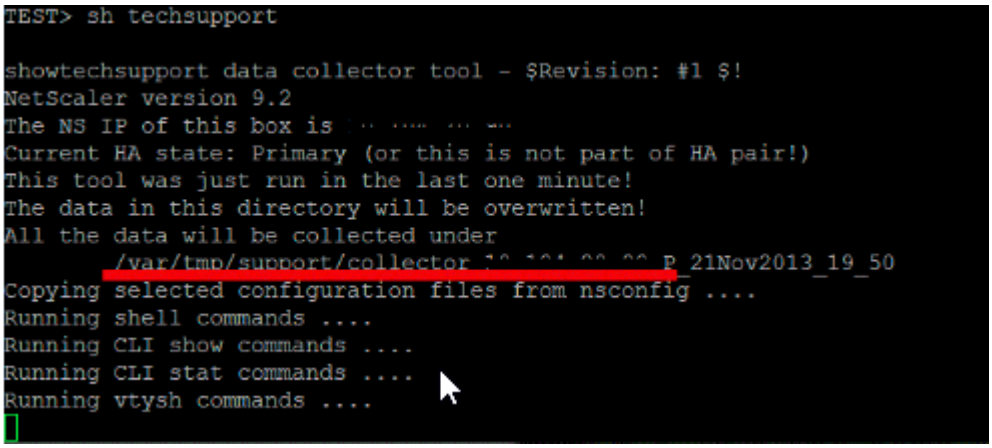
コマンドインターフェイスを使用してテクニカルサポートバンドルを入手する

1. `WinSCP`などのセキュア FTP（SFTP）またはセキュアコピー（SCP）ユーティリティを使用してアプライアンスからファイルをダウンロードし、分析のために Citrix Insight Services にアップロードします。

> 注:
>
> 9.0 より前の NetScaler ソフトウェアリリースでは、コレクタースクリプトを個別にダウンロードして実行する必要があります。


```
> show techsupport -scope CLUSTER
<!--NeedCopy-->
```

1. これにより、クラスター内のすべてのノードから表示テクニカルサポート情報が収集され、ファイルが 1 つのアーカイブに圧縮されます。
2. アプライアンスがコレクタアーカイブを生成すると、次のスクリーンショットのようにファイルの場所が表示されます。



ファイルは/var/tmp/supportに保存され、NetScaler アプライアンスにログインし、シェルプロンプトから次のコマンドを実行することで確認できます。

```
root@NS# cd /var/tmp/support/
root@NS# ls -l
<!--NeedCopy-->
```

GUI を使用して **NetScaler SDX** から診断バンドルを取得します

1. NetScaler SDX GUI を開きます。
2. [診断] ノードを展開します。
3. [テクニカルサポート] ノードを選択します。
4. 「テクニカルサポートファイルを生成」をクリックします。
5. ドロップダウンメニューから [アプライアンス (インスタンスを含む)] を選択します。
6. [追加] をクリックします。
7. 追加するインスタンスを 1 つ以上選択します。
8. [**OK**] をクリックします。プロセスが完了するのを待ちます。
9. 生成されたバンドル名を選択し、[ダウンロード] をクリックします。
10. バンドルファイルを Citrix Insight Servicesアップロードします。

その他のリソース

ビデオを見る

別のトピックを読む

コマンドリファレンスドキュメント

システムユーザーの認証と承認

August 15, 2023

NetScaler ユーザーの認証と承認を構成するには、まず NetScaler アプライアンスにアクセスできるユーザーを定義する必要があります。次に、これらのユーザーをグループに整理できます。ユーザーとグループを設定したら、コマンドポリシーを設定してアクセスのタイプを定義し、そのポリシーをユーザーまたはグループに割り当てる必要があります。

ユーザー、グループ、およびコマンドポリシーを設定するには、管理者としてログオンする必要があります。デフォルトの *NetScaler* 管理者のユーザー名は *nsroot* です。デフォルト管理者としてログオンしたら、nsroot アカウントのパスワードを変更する必要があります。パスワードを変更すると、そのユーザーのアカウントを作成するまで、ユーザーは NetScaler ADC アプライアンスにアクセスできなくなります。デフォルトから変更した後で管理者パスワードを忘れた場合は、nsroot にリセットできます。

> 注:
>
> - ローカルユーザーは、外部認証サーバーが構成されている場合でも、NetScaler ADC に対して認証できます。これを制限するには、set system parameter コマンドの LocalAuth パラメーターを無効にします。
> - セキュリティを強化するため、Citrix では nsroot パスワードを変更することをお勧めします。パスワードを頻繁に変更することをお勧めします。nsroot パスワードを変更する方法については、「デフォルトの管理者 (nsroot) パスワードのリセット」トピックを参照してください。

ユーザー、ユーザーグループ、コマンドポリシー

January 11, 2024

最初にアカウントを持つユーザーを定義し、次にすべてのユーザーをグループにまとめる必要があります。コマンドポリシーを作成するか、組み込みコマンドポリシーを使用してコマンドへのユーザーアクセスを規制できます。

> 注:
>
> トラフィック管理のための NetScaler 認証および承認設定の一部としてユーザーとユーザーグループを構成する方法について詳しく知りたい場合は、「ユーザーとグループの構成」トピックを参照してください。

また、ユーザーのコマンドラインプロンプトをカスタマイズすることもできます。プロンプトは、ユーザー設定、ユーザーグループ設定、およびグローバルシステム構成設定で定義できます。ユーザーに表示されるプロンプトの優先順位は次のとおりです。

1. ユーザーの設定で定義されているプロンプトを表示します。
2. ユーザーのグループのグループ設定で定義されているプロンプトを表示します。
3. システムグローバル設定で定義されているプロンプトを表示します。

システムユーザーの非アクティブな CLI セッションのタイムアウト値を指定できるようになりました。ユーザーの CLI セッションがタイムアウト値を超える時間アイドル状態になると、NetScaler アプライアンスは接続を終了します。タイムアウトは、ユーザー構成、ユーザーグループ構成、またはグローバルシステム構成設定で定義できます。ユーザの非アクティブな CLI セッションのタイムアウトは、次の優先順位で決定されます。

1. [ユーザー設定]:
2. ユーザーのグループのグループ構成。
3. グローバルシステム構成設定。

NetScaler のルート管理者は、システムユーザーの最大同時セッション制限を構成できます。制限を制限することで、開いている接続の数を減らし、サーバーのパフォーマンスを向上させることができます。CLI の数が設定された制限内であれば、同時ユーザーは GUI に何度でもログオンできます。ただし、CLI セッションの数が設定された制限に達すると、ユーザは GUI にログオンできなくなります。たとえば、同時セッション数が 20 に設定されている場合、同時ユーザーは 19 の CLI セッションにログオンできます。ただし、ユーザーが 20th CLI セッションにログオンしている場合、GUI、CLI、または NITRO にログオンしようとするとエラー・メッセージが表示されます (エラー:CFE への接続制限を超えました)。

> 注:
>
> デフォルトでは、同時セッション数は 20 に設定され、最大同時セッション数は 40 に設定されています。

ユーザーアカウントの構成

ユーザーアカウントを設定するには、ユーザー名とパスワードを指定するだけです。パスワードの変更やユーザーアカウントの削除はいつでもできます。

> 注:
>
> パスワードのすべての文字は使用できません。ただし、文字を引用符で囲んで入力すると機能します。
>
> また、文字列は最大長の 127 文字を超えてはなりません。

コマンドラインインターフェイスを使用してユーザーアカウントを作成するには

コマンドプロンプトで、次のコマンドを入力してユーザーアカウントを作成し、構成を確認します。

- `add system user <username> [-externalAuth ( ENABLED | DISABLED )] [-promptString <string>] [-timeout \<secs>] [-logging ( ENABLED | DISABLED )] [-maxsession <positive_integer>]`
- `show system user <userName>`

外部ユーザーは、「logging」パラメーターを設定して、Web ロギングまたは監査ロギングメカニズムを使用して外部ログを収集できます。このパラメータを有効にすると、監査クライアントは NetScaler ADC アプライアンスで自分自身を認証してログを収集します。

例**:**

```
> add system user johnd -promptString user-%u-at-%T
```

```
1 Enter password:
2 Confirm password:
3 > show system user johnd
4 user name: john
5       Timeout:900 Timeout Inherited From: Global
6       External Authentication: ENABLED
7       Logging: DISABLED
8       Maximum Client Sessions: 20
9 <!--NeedCopy-->
```

パラメータの説明については、「認証および認可ユーザコマンドリファレンス」を参照してください。

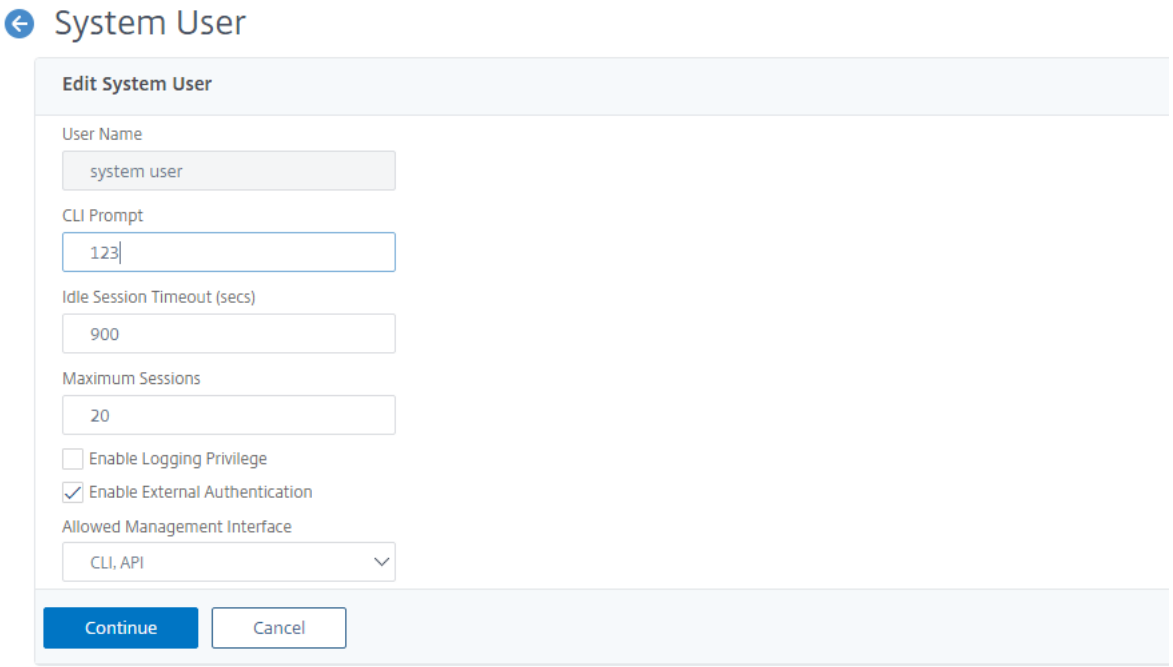
NetScaler GUI を使用してユーザーアカウントを構成します

1. [システム] > [ユーザー管理] > [ユーザー] に移動し、ユーザーを作成します。
2. 詳細ペインで、[追加] をクリックしてシステムユーザーを作成します。
3. **[**システムグループの作成] ページで、次のパラメータを設定します。

    a) ユーザー名。ユーザーグループの名前。
    b) CLI プロンプト。CLI インターフェイスアクセス用に設定したいプロンプト。
    c) アイドルセッションタイムアウト (秒)。セッションがタイムアウトして終了する前に、ユーザーが非アクティブでいられる時間を設定します。
    d) 最大セッション数。ユーザーが試すことができる最大セッション数を設定します。nsroot ユーザーの場合、最大セッションパラメーターは変更できません。
    e) ロギング権限を有効にします。ユーザーのログ権限を有効にします。
    f) 外部認証を有効にします。外部認証サーバーを使用してユーザーを認証する場合は、このオプションを選択します。

g) 許可された管理インターフェイス。ユーザーグループにアクセス権限が付与されている NetScaler インターフェイスを選択します。
h) コマンドポリシー。コマンドポリシーをユーザーグループにバインドします。
i) パーティション。パーティションをユーザーグループにバインドします。

4. **[作成]** して **[閉じる]** をクリックします。



ユーザーグループの設定

ユーザーグループを設定したら、グループ内の全員に同じアクセス権を簡単に付与できます。グループを設定するには、グループを作成し、ユーザーをそのグループにバインドします。各ユーザーアカウントを複数のグループにバインドできます。ユーザーアカウントを複数のグループにバインドすると、コマンドポリシーをより柔軟に適用できる場合があります。

コマンドラインインターフェイスを使用してユーザーグループを作成するには

コマンドプロンプトで、次のコマンドを入力してユーザーグループを作成し、構成を確認します。

- `add system group <groupName> [-promptString <string>] [-timeout <secs>]`
- `show system group <groupName>`

例:

```
> add system group Managers -promptString Group-Managers-at-%h
```


CLI を使用してユーザーアカウントをグループにバインドする

コマンドプロンプトで次のコマンドを入力して、ユーザーアカウントをグループにバインドし、構成を確認します。

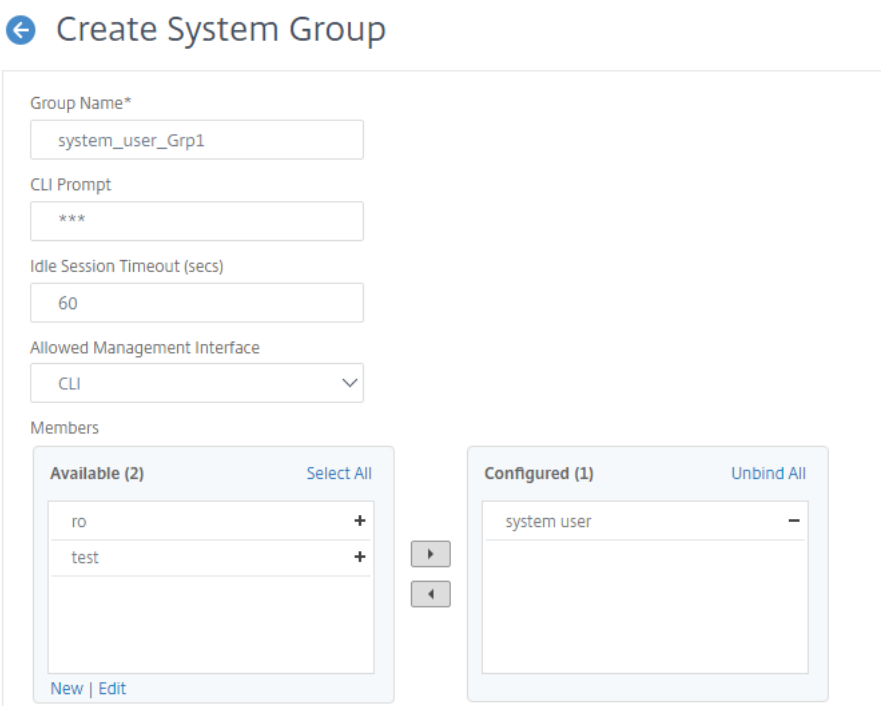
- `bind system group <groupName> -userName <userName>`
- `show system group <groupName>`

例**:**

```
> bind system group Managers -userName user1
```

NetScaler GUI を使用してユーザーグループを構成する

1. [システム] > [ユーザー管理] > [グループ] に移動し、ユーザーグループを作成します。
2. 詳細ペインで、[追加] をクリックしてシステムユーザーグループを作成します。
3. **[**システムグループの作成] ページで、次のパラメータを設定します。
   a) グループ名。ユーザーグループの名前。
   b) CLI プロンプト。CLI インターフェイスアクセス用に設定したいプロンプト。
   c) アイドルセッションタイムアウト (秒)。セッションがタイムアウトして終了する前に、ユーザーが非アクティブでいられる時間を設定します。
   d) 許可された管理インターフェイス。ユーザーグループにアクセス権限が付与されている NetScaler インターフェイスを選択します。
   e) メンバー。ユーザーアカウントをグループに追加します。
   f) コマンドポリシー。コマンドポリシーをユーザーグループにバインドします。
   g) パーティション。パーティションをユーザーグループにバインドします。
4. **[**作成**]** して **[**閉じる**]** をクリックします。



> 注:
>
> グループにメンバーを追加するには、「メンバー」セクションで「追加」をクリックします。「使用可能」リストからユーザーを選択し、「構成済み」リストに追加します。

コマンドポリシーの設定

コマンドポリシーは、ユーザーとユーザーグループが使用できるコマンド、コマンドグループ、仮想サーバー、およびその他のエンティティを規制します。

アプライアンスには一連の組み込みコマンドポリシーがあり、カスタムポリシーを設定できます。ポリシーを適用するには、ポリシーをユーザーまたはグループにバインドします。

コマンドポリシーを定義して適用する際に留意すべき重要なポイントは次のとおりです。

- グローバルコマンドポリシーを作成することはできません。コマンドポリシーは、アプライアンス上のユーザーとグループに直接バインドする必要があります。
- コマンドポリシーが関連付けられていないユーザーまたはグループには、デフォルト（DENY-ALL）コマンドポリシーが適用されるため、適切なコマンドポリシーがアカウントにバインドされるまで、設定コマンドを実行できません。
- すべてのユーザーは、所属するグループのポリシーを継承します。
- コマンドポリシーをユーザーアカウントまたはグループアカウントにバインドするときは、コマンドポリシーに優先順位を割り当てる必要があります。これにより、アプライアンスは、複数の競合するポリシーが同じユーザーまたはグループに適用される場合に、どのポリシーが優先されるかを判断できます。

- 同じ優先度の 2 つの異なるコマンドポリシーがユーザーアカウントまたはグループアカウントにバインドされている場合、最初にバインドされたポリシーが最も優先されます。
- 次のコマンドは、デフォルトですべてのユーザーが使用でき、指定したコマンドの影響を受けません:
- ヘルプ、CLI 属性の表示、CLI プロンプトの設定、CLI プロンプトのクリア、CLI プロンプトの表示、エイリアス、ユニエイリアス、履歴、終了、whoami、config、CLI モードの設定、CLI モードの設定解除、CLI モードの設定解除、CLI モードの表示

次の表では、組み込みポリシーについて説明します。

ポリシー名	許可する
read-only	show ns RunningConfig、show ns ns.conf、および NetScaler コマンドグループの show コマンドを除くすべての show コマンドへの読み取り専用アクセス。
operator	読み取り専用アクセスと、サービスとサーバーを有効または無効にするコマンドへのアクセス。
network	SSL コマンドの設定と設定解除、show ns ns.conf、show ns RunningConfig、show gslb RunningConfig コマンドを除くフルアクセス。
sysadmin	[NetScaler 12.0 以降に搭載] システム管理者は、アプライアンスで許可されるアクセス権の点ではスーパーユーザーよりも劣ります。sysadmin ユーザーは、NetScaler シェルにアクセスできない、ユーザー構成を実行できない、パーティション構成を実行できない、および sysadmin コマンドポリシーに記載されているその他の構成を実行できないという例外を除いて、すべての NetScaler 操作を実行できます。
スーパーユーザー	フルアクセス。nsroot ユーザーと同じ権限です。

カスタムコマンドポリシーの作成

正規表現のサポートは、よりカスタマイズされた表現を維持するためのリソースがあるユーザーや、正規表現が提供する柔軟性を必要とする導入環境向けに提供されています。ほとんどのユーザーには、組み込みのコマンドポリシーで十分です。より高度な制御が必要だが、正規表現に慣れていないユーザーは、ポリシーを読みやすくするために、このセクションで紹介する例のような単純な表現だけを使用するとよいでしょう。

正規表現を使用してコマンドポリシーを作成するときは、次の点に注意してください。

- 正規表現を使用してコマンドポリシーの影響を受けるコマンドを定義する場合は、コマンドを二重引用符で囲む必要があります。たとえば、**show** で始まるすべてのコマンドを含むコマンドポリシーを作成するには、次のように入力します。

- “^show .*$”
- rm で始まるすべてのコマンドを含むコマンドポリシーを作成するには、次のように入力します。
- “^rm .*$”
- コマンドポリシーで使用される正規表現では、大文字と小文字は区別されません。

次の表は、コマンドポリシーの正規表現の例を示しています。

コマンド仕様	以下のコマンドにマッチします
“^rm\s+.*$”	すべての削除アクションは rm 文字列で始まり、その後にスペースと、コマンドグループ、コマンドオブジェクトタイプ、引数などのパラメータが続くためです。
“^show\s+.*$”	すべての show コマンド。すべての show アクションは show 文字列で始まり、その後にスペースと、コマンドグループ、コマンドオブジェクトタイプ、引数などのパラメータが続くためです。
“^shell$”	シェルコマンドだけですが、コマンドグループ、コマンドオブジェクトタイプ、引数などの追加パラメーターと組み合わせることはできません。
“^add\s+vserver\s+.*$”	すべて仮想サーバーアクションを作成します。仮想サーバーアクションは、add virtual server コマンドの後にスペースが続き、コマンドグループ、コマンドオブジェクトタイプ、引数などのパラメーターが続きます。
“^add\s+(lb\s+vserver)\s+.*”	すべてが lb 仮想サーバーアクションを作成します。これは、add lb 仮想サーバーコマンドとそれに続くスペース、およびコマンドグループ、コマンドオブジェクトタイプ、引数などのパラメーターで構成されます。

組み込みコマンドポリシーの詳細については、「組み込みコマンドポリシーテーブル」を参照してください。

コマンドラインインターフェイスを使用してコマンドポリシーを作成するには

コマンドプロンプトで、次のコマンドを入力してコマンドポリシーを作成し、構成を確認します。

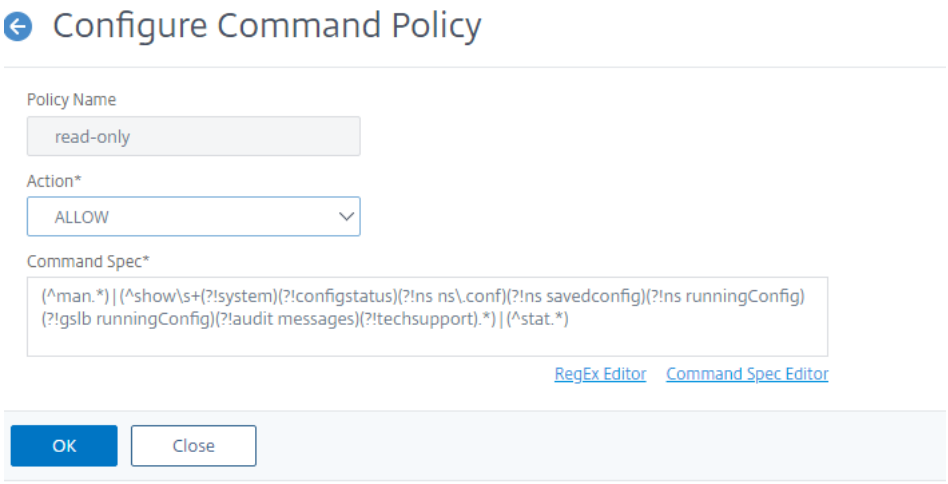
- `add system cmdPolicy <policyname> <action> <cmdspec>`
- `show system cmdPolicy <policyName>`

例:

```
add system cmdPolicy USER-POLICY ALLOW (\ server\ )|(\ service(Group)
*\ )|(\ vserver\ )|(\ policy\ )|(\ policylabel\ )|(\ limitIdentifier\
 )|(^show\ (?!(system|ns\ (ns.conf|runningConfig))))|(save)|(stat\ .*
serv)
```


NetScaler GUI を使用してコマンドポリシーを構成する

1. [システム] > [ユーザー管理] > [コマンドポリシー] に移動します。
2. 詳細ペインで、[追加] をクリックしてコマンドポリシーを作成します。
3. 「コマンドポリシーの設定」ページで、次のパラメータを設定します。
   a) ポリシー名
   b) アクション
   c) コマンド仕様
4. [**OK**] をクリックします。



コマンドポリシーをユーザーアカウントとユーザーグループにバインドする

コマンドポリシーを定義したら、それらを適切なユーザーアカウントとグループにバインドする必要があります。ポリシーをバインドするときは、ポリシーに優先順位を割り当てる必要があります。これにより、適用可能な複数のコマンドポリシーが競合した場合に、アプライアンスがどのコマンドポリシーに従うべきかを判断できます。

コマンドポリシーは次の順序で評価されます。

- ユーザーと対応するグループに直接バインドされたコマンドポリシーは、優先度番号に従って評価されます。プライオリティ番号が小さいコマンドポリシーは、プライオリティ番号が高いコマンドポリシーよりも先に評価されます。したがって、番号の小さいコマンドポリシーが明示的に付与または拒否する権限が、番号の大きいコマンドポリシーによって上書きされることはありません。
- ユーザーアカウントとグループにバインドされた 2 つのコマンドポリシーの優先順位番号が同じ場合、ユーザーアカウントに直接バインドされたコマンドポリシーが最初に評価されます。

コマンドラインインターフェイスを使用してコマンドポリシーをユーザーにバインドするには

コマンドプロンプトで次のコマンドを入力して、コマンドポリシーをユーザーにバインドし、構成を確認します。

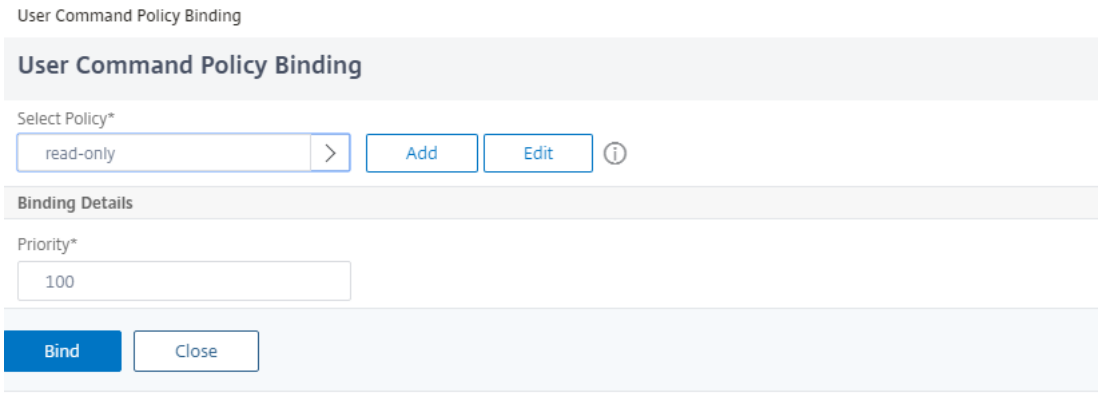
- `bind system user <userName> -policyName <policyName> <priority>`
- `show system user <userName>`

例:

```
> bind system user user1 -policyName read_all 1
```

NetScaler GUI を使用してコマンドポリシーをユーザーアカウントにバインドする

[システム] > [ユーザー管理] > [ユーザー] に移動し、ユーザーを選択してコマンドポリシーをバインドします。



オプションで、ポリシーが正しい順序で評価されるように、デフォルトの優先度を変更できます。

コマンドラインインターフェイスを使用してコマンドポリシーをグループにバインドするには

コマンドプロンプトで次のコマンドを入力して、コマンドポリシーをユーザーグループにバインドし、構成を確認します。

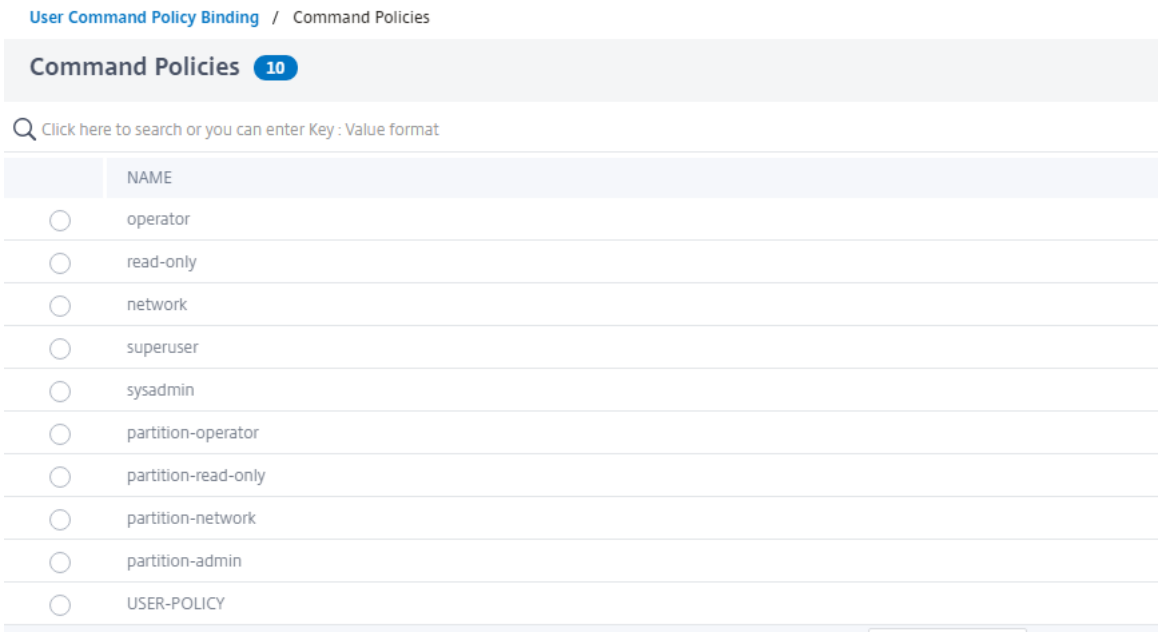
- `bind system group <groupName> -policyName <policyName> <priority>`
- `show system group <groupName>`

例:

```
> bind system group Managers -policyName read_all 1
```

NetScaler GUI を使用してコマンドポリシーをユーザーグループにバインドする

[システム] > [ユーザー管理] > [グループ] に移動し、グループを選択してコマンドポリシーをバインドします。



オプションで、ポリシーが正しい順序で評価されるように、デフォルトの優先度を変更できます。

ユースケースの例**:** 製造組織のユーザーアカウント、ユーザーグループ、およびコマンドポリシーの管理

次の例は、ユーザーアカウント、グループ、およびコマンドポリシーの完全なセットを作成し、各ポリシーを適切なグループとユーザーにバインドする方法を示しています。Example Manufacturing, Inc. という会社には、NetScaler アプライアンスにアクセスできるユーザーが 3 人います。

- **John Doe.** IT マネージャー。John は NetScaler 構成のすべての部分を確認できる必要がありますが、何も変更する必要はありません。
- **Maria Ramiez.** 主任の IT 管理者。Maria は、NetScaler コマンド（ローカルポリシーにより、nsroot としてログオンした状態で実行する必要がある）を除く NetScaler 構成のすべての部分を表示および変更できる必要があります。
- **Michael Baldrock.** 負荷分散を担当する IT 管理者。Michael は NetScaler 構成のすべての部分を確認できる必要がありますが、変更する必要があるのは負荷分散機能だけです。

次の表は、サンプル会社のネットワーク情報、ユーザーアカウント名、グループ名、およびコマンドポリシーの内訳を示しています。

フィールド	**Value**	注
Citrix ADC ホスト名	ns01.example.net	-

フィールド	**Value**	注
ユーザーアカウント	johnd, mariar, and michaelb	IT マネージャーのジョン・ドウ、IT 管理者のマリア・ラミレス、IT 管理者のマイケル・バルドロック。
グループ	マネージャーとシステムオペス	すべてのマネージャーとすべての IT 管理者。
コマンドポリシー	read_all, modify_lb, and modify_all	[完全な読み取り専用アクセスを許可]、[負荷分散への変更アクセスを許可]、[完全な変更アクセスを許可]。

次の説明では、ns01.example.net という名前の NetScaler アプライアンスでユーザーアカウント、グループ、およびコマンドポリシー一式を作成するプロセスについて説明します。

説明には、適切なユーザーアカウントとグループを相互にバインドする手順と、適切なコマンドポリシーをユーザーアカウントとグループにバインドする手順が含まれています。

この例は、優先順位付けを使用して IT 部門の各ユーザーに正確なアクセス権と権限を付与する方法を示しています。

この例では、NetScaler で初期インストールと構成がすでに実行されていることを前提としています。

サンプル組織のユーザーアカウント、グループ、コマンドポリシーの設定

1. 「ユーザーアカウントの設定」セクションで説明されている手順を使用して、**johnd**、**mariar**、および **michaelb** のユーザーアカウントを作成します。
2. 「ユーザーグループの設定」で説明されている手順を使用して、ユーザーグループ **Manager** と **SysOps** を作成し、ユーザー **mariar** と **michaelb** を **SysOps** グループにバインドし、ユーザー **johnd** を **Manager** グループにバインドします。
3. 「カスタムコマンドポリシーの作成」で説明されている手順を使用して、以下のコマンドポリシーを作成します:
   - **read_all** アクション付き [許可] と [コマンド仕様] `"(^show\s+(?!system)(?!ns ns.conf)(?!ns runningConfig).*)|(^stat.*)"`
   - **modify_lb (** アクションが [許可] で、コマンドスペック) `"^set\s+lb\s+.*$"`
   - modify_all とアクション Allow およびコマンド仕様 `"^\S+\s+(?!system).*"`
4. **read_all** コマンドポリシーをプライオリティ値 **1** で **SysOps** グループにバインドするには、「コマンドポリシーをユーザおよびグループへのバインド」で説明する手順を使用します。
5. 「ユーザーおよびグループへのコマンドポリシーのバインド」で説明されている手順を使用して、**modify_lb** コマンドポリシーをユーザー **michaelb** にバインドし、優先度値 **5** を指定します。

作成した構成は、次のようになります:

- IT マネージャーの John Doe は、NetScaler 構成全体に対する読み取り専用アクセス権を持っていますが、変更を加えることはできません。
- IT リーダーのマリア・ラミレスは、NetScaler 構成のすべての領域にほぼ完全にアクセスでき、ログオンするだけで NetScaler レベルのコマンドを実行できます。
- 負荷分散を担当する IT 管理者の Michael Baldrock は、NetScaler 構成への読み取り専用アクセス権を持ち、負荷分散の構成オプションを変更できます。

特定のユーザーに適用されるコマンドポリシーのセットは、ユーザーのアカウントに直接適用されるコマンドポリシーと、ユーザーがメンバーになっている 1 つ以上のグループに適用されるコマンドポリシーを組み合わせたものです。

ユーザーがコマンドを入力するたびに、オペレーティングシステムは、コマンドと一致する ALLOW または DENY アクションを含むポリシーが見つかるまで、そのユーザーのコマンドポリシーを検索します。一致するものが見つかると、オペレーティングシステムはコマンドポリシー検索を停止し、コマンドへのアクセスを許可または拒否します。

オペレーティングシステムが一致するコマンドポリシーを見つけられない場合、NetScaler ADC アプライアンスのデフォルトの拒否ポリシーに従って、コマンドへのユーザーアクセスを拒否します。

> 注:
>
> ユーザーを複数のグループに分ける場合は、意図しないユーザーコマンドの制限や権限を与えないように注意してください。このような競合を避けるため、ユーザーをグループにまとめる際には、NetScaler のコマンドポリシー検索手順とポリシー順序規則を念頭に置いてください。

ユーザーアカウントとパスワード管理

January 11, 2024

NetScaler では、ユーザーアカウントとパスワード構成を管理できます。以下は、システムユーザーアカウントまたは管理者ユーザーアカウントを`nsroot`使用して実行できるアクティビティの一部です。

- システムユーザーアカウントのロックアウト
- システムユーザーアカウントをロックして管理アクセスできるようにする
- ロックされたシステムユーザーアカウントのロックを解除して管理アクセスできるようにする
- システムユーザーアカウントの管理アクセスを無効にする
- `nsroot`管理ユーザーのパスワード変更を強制
- システムユーザーアカウントの機密ファイルを削除する
- システムユーザー向けの強力なパスワード設定

システムユーザーアカウントのロックアウト

ブルートフォースセキュリティ攻撃を防ぐために、ユーザーロックアウト設定を構成できます。この構成により、ネットワーク管理者はシステムユーザーが NetScaler にログオンできないようにすることができます。また、ロック期間が終了する前にユーザーアカウントのロックを解除してください。

再起動後に失敗したユーザーログインの詳細を取得するには`persistentLoginAttempts`、パラメーターを有効にします。

コマンドプロンプトで入力します:

```
set aaa parameter -maxloginAttempts <value> -failedLoginTimeout <
value> -persistentLoginAttempts (ENABLED | DISABLED)
```

例:

```
set aaa parameter -maxloginAttempts 3 -failedLoginTimeout 10 -persistentLoginA
 ENABLED
```

> 注:
>
> この`aaa.user.login_attempts`式を有効にするには、「永続的なログイン試行」パラメータを無効にする必要があります。
>
> コマンドを`unset aaa parameter -persistentLoginAttempts`実行して、永続的なログイン試行を無効にします (有効になっている場合)。
>
> ログイン試行機能の詳細については、「ユーザーの現在のログイン試行を取得するためのサポート」を参照してください。

次の show コマンド出力には、認証、許可、および監査パラメータの設定ステータスが表示されます。

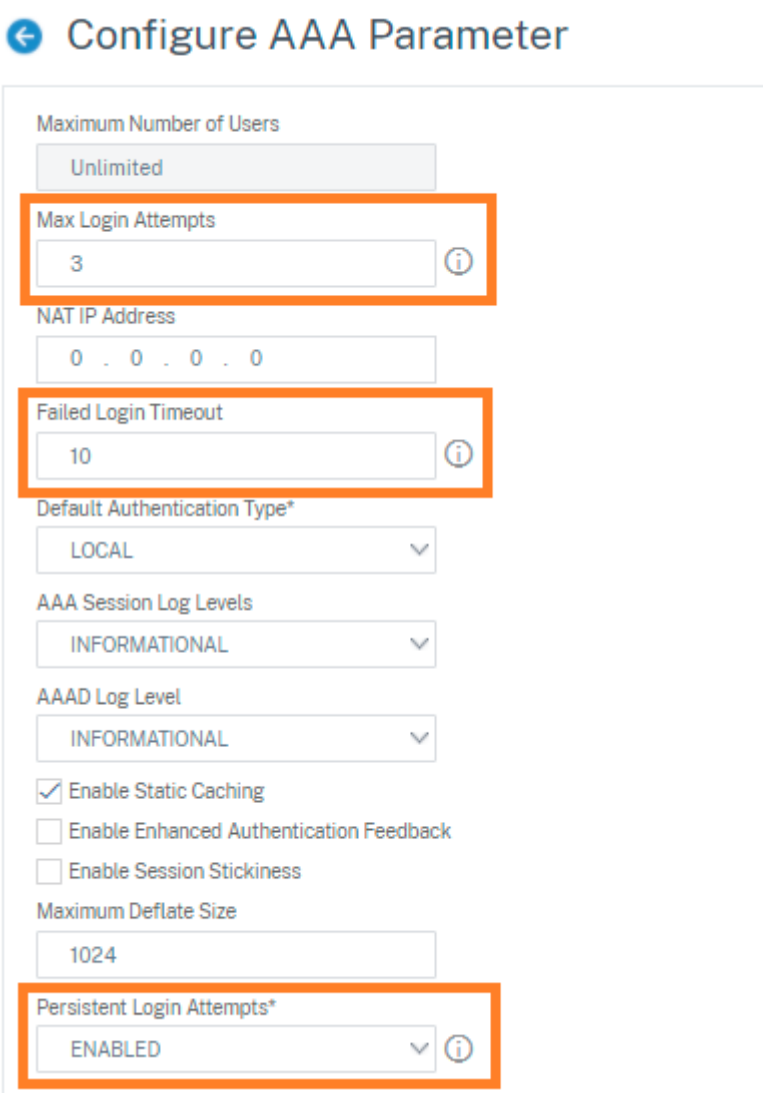
```
1  show aaaparameter
2
3  Configured AAA parameters
4
5  EnableStaticPageCaching: YES
6
7  EnableEnhancedAuthFeedback: NO
8
9  DefaultAuthType: LOCAL MaxAAAUsers: Unlimited
10
11 AAAD nat ip: None
12
13 EnableSessionStickiness : NO
14
15 aaaSessionLoglevel: INFORMATIONAL
16
17 AAAD Log Level: INFORMATIONAL
18
19 ...
```



```
20
21 Persistent Login Attempts: DISABLED
22
23 <!--NeedCopy-->
```

GUI を使用してシステムユーザーアカウントのロックアウトを設定します

1. [設定] **>** [セキュリティ] **>** [**AAA** アプリケーショントラフィック] **>** [認証設定] **>** [認証 **AAA** 設定の変更] に移動します。
2. **AAA** パラメータの設定ページで、次のパラメータを設定します。
   a) 最大ログイン試行回数。ユーザーが試すことができる最大ログオン試行回数。
   b) ログイン失敗タイムアウト。ユーザーによる無効なログオン試行の最大回数。
   c) 永続的なログインの試行。リブート後も失敗したユーザーのログイン試行を永続的に保存できます。
3. [**OK**] をクリックします。



パラメータを設定すると、3 回以上ログインを試みると、ユーザーアカウントは 10 分間ロックされます。また、ユーザーは有効な認証情報を使用しても 10 分間ログオンできません。

> 注
>
> ロックされたユーザーが NetScaler にログオンしようとすると、エラーメッセージが表示RBA `Authentication Failure: maxlogin attempt reached for test.`されます。

システムユーザーアカウントをロックして管理アクセスできるようにする

NetScaler では、システムユーザーを 24 時間ロックし、ユーザーへのアクセスを拒否できます。

NetScaler は、システムユーザーと外部ユーザーの両方の構成をサポートします。

> 注
>
> この機能は、aaaパラメータのpersistentLoginAttemptsオプションを無効にした場合にのみサポートされます。

コマンドプロンプトで次のように入力します:

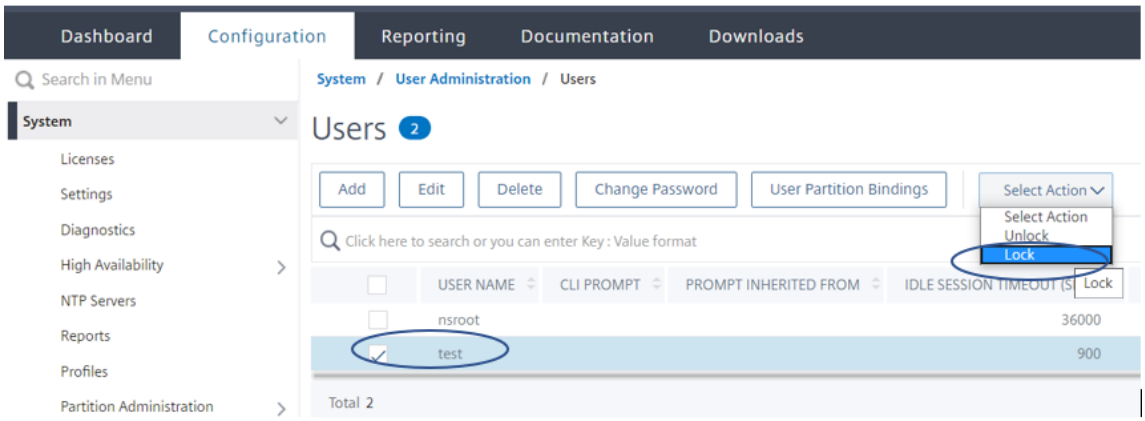
```
set aaa parameter -persistentLoginAttempts DISABLED
```

ここで、ユーザーアカウントをロックするには、コマンドプロンプトで次のように入力します。

```
lock aaa user test
```

GUI を使用してシステムユーザーアカウントをロックする

1. [設定] **>** [セキュリティ] **>** [**AAA** アプリケーショントラフィック] **>** [認証設定] **>** [認証 **AAA** 設定の変更] に移動します。
2. 「**AAA** パラメータの設定」の「持続的ログイン試行回数」リストで、「無効」を選択します。
3. [**System**] **>** [**User Administration**] **>** [**Users**] の順に選択します。
4. ユーザーを選択します。
5. 「アクションの選択」リストで、「ロック」を選択します。



> 注
>
> NetScaler GUI には、外部ユーザーをロックするオプションはありません。外部ユーザーをロックするには、ADC 管理者は CLI を使用する必要があります。
> ロックされたシステムユーザー（ロック認証、承認、および監査ユーザーコマンドでロックされている）が NetScaler にログインしようとすると、「RBA 認証失敗: ユーザーテストは 24 時間ロックされています。」というエラーメッセージが表示されます。

> ユーザーが管理アクセスにログオンするようにロックされている場合、コンソールアクセスは免除されます。ロックされたユーザーはコンソールにログオンできます。

ロックされたシステムユーザーアカウントのロックを解除して管理アクセスできるようにする

システムユーザーと外部ユーザーは、ロック認証、承認、および監査ユーザーコマンドを使用して 24 時間ロックできます。

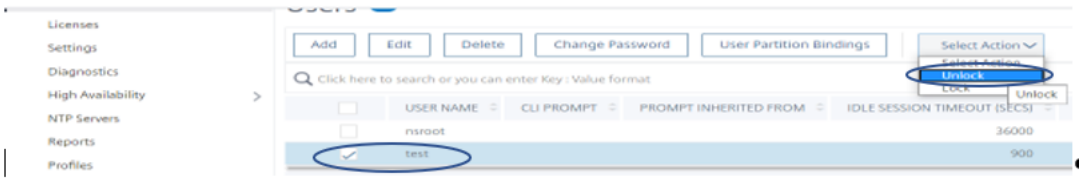
> 注
>
> NetScaler では、管理者はロックされたユーザーのロックを解除できます。この機能では、「PersistentLoginAttempts」コマンドで設定する必要はありません。

コマンドプロンプトで入力します:

```
unlock aaa user test
```

GUI を使用してシステムユーザーのロック解除を設定する

1. [**System**] > [**User Administration**] > [**Users**] の順に選択します。
2. ユーザーを選択します。
3. ロック解除] をクリックします。



NetScaler GUI には、ADC で作成されたシステムユーザーのみが表示されるため、GUI には外部ユーザーのロックを解除するオプションはありません。外部ユーザーのロックを解除するには、`nsroot` 管理者は CLI を使用する必要があります。

システムユーザーアカウントの管理アクセスを無効にする

外部認証を設定して管理者として、システムユーザーが管理アクセスにログオンすることを拒否したい場合は、システムパラメーターの LocalAuth オプションを無効にする必要があります。

コマンドプロンプトで、次のように入力します:

```
set system parameter localAuth <ENABLED|DISABLED>
```

例:

```
set system parameter localAuth DISABLED
```


GUI を使用してシステムユーザーへの管理アクセスを無効にする

1. [構成] **>** [システム] **>** [設定] **>** [グローバルシステム設定の変更] に移動します。
2. コマンドラインインターフェイス **(CLI)** セクションで、ローカル認証チェックボックスを選択解除します。

このオプションを無効にすると、ローカルシステムユーザーは ADC 管理アクセスにログオンできなくなります。

> 注
>
> システムパラメータでローカルシステムユーザー認証を許可しないように、外部認証サーバーを設定し、アクセスできるようにする必要があります。ADC で管理アクセス用に構成された外部サーバーにアクセスできない場合、ローカルシステムユーザーは NetScaler にログオンできます。この動作は回復を目的として設定されています。

管理ユーザーのパスワード変更を強制する

セキュリティでnsroot保護された認証では、システムパラメーターでオプションが有効になっている場合、NetScaler はユーザーにforcePasswordChangeデフォルトパスワードを新しいパスワードに変更するように求めます。デフォルトのクレデンシャルを使用して、初回ログイン時に CLI または GUI からnsrootパスワードを変更できます。

コマンドプロンプトで入力します:

```
set system parameter -forcePasswordChange ( ENABLED | DISABLED )
```

NSIP の **SSH** セッションの例**:**

```
ssh nsroot@1.1.1.1
Connecting to 1.1.1.1:22...
Connection established.
To escape to local shell, press Ctrl+Alt+].
###############################################################################

WARNING: Access to this system is for authorized users only #
Disconnect IMMEDIATELY if you are not an authorized user! #

###############################################################################

Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
Done
<!--NeedCopy-->
```

システムユーザーアカウントの機密ファイルを削除する

システムユーザーアカウントの認証キーや公開キーなどの機密データを管理するには、removeSensitiveFiles オプションを有効にする必要があります。システムパラメータが有効になっているときに機密ファイルを削除するコ

マンドは次のとおりです。

- rm cluster instance
- rm cluster node
- rm 高可用性ノード
- コンフィグをクリア
- join cluster
- add cluster instance

コマンドプロンプトで入力します:

```
set system parameter removeSensitiveFiles ( ENABLED | DISABLED )
```

例:

```
set system parameter -removeSensitiveFiles ENABLED
```

システムユーザー向けの強力なパスワード設定

安全な認証のため、NetScaler はシステムユーザーと管理者に、コンソールにログオンするための強力なパスワードを設定するよう求めます。パスワードは長くなければならず、次の組み合わせでなければなりません。

- 小文字を 1 つ
- 大文字 1 文字
- 1 つの数字
- 1 つの特殊文字

コマンドプロンプトで入力します:

```
set system parameter -strongpassword <value> -minpasswordlen <value>
```

各項目の意味は次のとおりです。

`Strongpassword`。強力なパスワード (`enable all` / `enablelocal`) を有効にした後は、すべてのパスワードまたは機密情報に次のものが必要です。

- 1 文字以上の小文字
- 少なくとも 1 文字の大文字
- 1 文字以上の数字
- 少なくとも 1 つの特殊文字

`enablelocal`の除外リストは - `NS_FIPS`, `NS_CRL`, `NS_RSAKEY`, `NS_PKCS12`, NS_PKCS8, NS_LDAP, `NS_TACACS`, `NS_TACACSACTION`, `NS_RADIUS`, `NS_RADIUSACTION`, `NS_ENCRYPTION_PARAMS`。そのため、システムユーザーに対してこれらの ObjectType コマンドに対して強固なパスワードチェックは行われません。

指定できる値: enableall、enablelocal、
無効デフォルト値: 無効

minpasswordlen。システムユーザーパスワードの最小文字数。強力なパスワードがデフォルトで有効になっている場合、最小文字数は 4 です。ユーザーが入力した値は 4 以上でもかまいません。強力なパスワードが無効になっている場合、デフォルトの最小値は 1 です。どちらの場合も、最大値は 127 です。

最小値:1
最大値:127

例:

```
set system parameter -strongpassword enablelocal -minpasswordlen 6
```

既定のユーザーアカウント

nsrecover 管理者はユーザーアカウントを使用して NetScaler アプライアンスをリカバリできます。予期しない問題が発生してデフォルトのシステムユーザー (nsroot) がログインできない場合は、を使用して NetScaler にnsrecoverログオンできます。nsrecover ログインはユーザー設定とは無関係で、シェルプロンプトに直接アクセスできます。設定の上限に達していても、nsrecoverからいつでもログインできます。

ルート管理者 (nsroot) パスワードをリセットする方法

April 15, 2024

NetScaler ルート管理者（nsroot）アカウントは、すべての ADC 機能への完全なアクセスを提供します。したがって、セキュリティを維持するために、管理者アカウントは必要な場合にのみ使用する必要があります。

管理者として、パスワードを変更することをお勧めします。パスワードを忘れた場合は、最初にデフォルトのパスワードにリセットしてから、新しいパスワードに変更する必要があります。

nsroot 管理者は、パスワードをリセットするには、アプライアンスにログオンしてパスワードを変更する必要があります。ただし、パスワードを覚えていない場合は、アプライアンスをシングルユーザーモードで再起動できます。ファイルシステムを読み取り/書き込みモードでマウントし、**ns.conf** ファイルから **NetScaler** エントリを削除します。最後の手順として、再起動してデフォルトのパスワードでアプライアンスにログオンし、新しいパスワードを設定します。

root 管理者パスワードをリセットするには、以下のステップを実行します。

1. コンピューターを NetScaler コンソールポートに接続してログオンします。

   > 注:
   >
   > この手順を実行するために SSH を使用してログオンすることはできません。アプライアンスに直接接続

> する必要があります。

2. NetScaler を再起動します。
3. 次のメッセージが表示されたら、Ctrl キーを押しながら C キーを押します。

```
Press [Ctrl-C] for command prompt, or any other key to boot
immediately.
Booting [kernel] in # seconds.
```

4. シングルユーザーモードで NetScaler を起動するには、次のコマンドを実行します。

```
boot -s
```

アプライアンスの起動後、次のメッセージが表示されます。

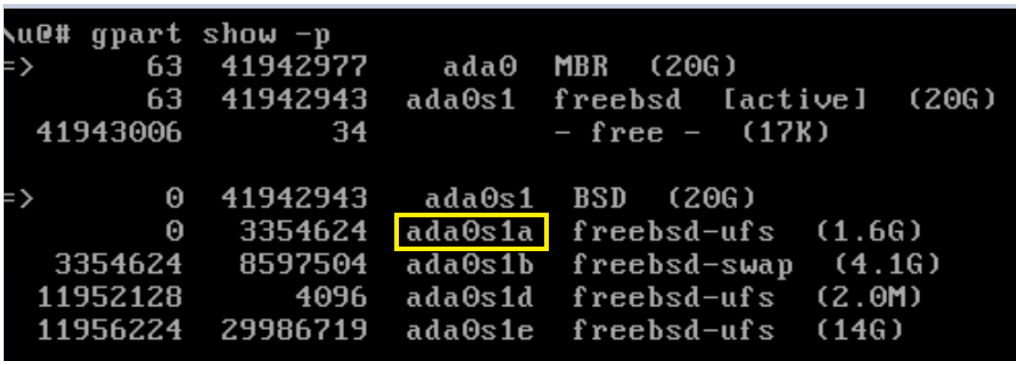
```
シェルのフルパス名またはRETURN for /bin/sh:を入力する
```

5. Enter キーを押して # プロンプトを表示し、次のコマンドを入力してファイルシステムをマウントします。

a) 次のコマンドを実行して、ディスクの整合性を確認します。

```
fsck_ufs /dev/ada0s1a
```

> 注
>
> フラッシュドライブには、NetScaler に応じて特定のデバイス名があります。ADC CLI で次のコマンドを実行し、「1a」で終わる名前をコピーします。
>
> ```
> gpart show -p
> ```
>
> たとえば、



b) dev ディレクトリにアクセスし、’ls’ と入力してドライブの詳細を確認します。

```
cd /dev/ada0s1a

ls
<!--NeedCopy-->
```

c) 次のコマンドを実行して、マウントされたパーティションを表示します。

```
df
```


> （注
>
> ）フラッシュパーティションがリストにない場合は、手動でマウントする必要があります。

d) 次のコマンドを実行して、フラッシュドライブをマウントします。

```
mount /dev/ada0s1a /flash
```

6. 以下のコマンドを実行して、nsconfig ディレクトリに移動します。

```
cd /flash/nsconfig
```

7. 次のコマンドを実行して ns.conf ファイルを書き換え、デフォルトで admin に設定されている一連のシステムコマンドを削除します。

a) 次のコマンドを実行して、管理者にデフォルト設定されるコマンドを含まない設定ファイルを作成します。

```
grep -v “set system user nsroot” ns.conf > new.conf
```

b) 次のコマンドを実行して、既存の設定ファイルのバックアップを作成します。

```
mv ns.conf old.ns.conf
```

c) 次のコマンドを実行して、新しい.conf ファイルの名前を ns.conf に変更します：

```
mv new.conf ns.conf
```

8. 次のコマンドを実行して NetScaler を再起動します：

```
reboot
```

9. デフォルトの管理者認証情報を使用してログオンします。NetScaler では、デフォルトのnsrootパスワードを新しいパスワードに置き換えるように求められます。

> 注
>
> “?”パスワード文字列内の文字。この文字の前に\文字を付けます。

たとえば、次の操作を実行した後、yourexamplepasswd?が管理者アカウントに設定されています：

```
> set system user nsroot yourexamplepasswd\?
```

10. 新しいパスワードを設定したら、save ns config コマンドを使用して設定を保存します。

> 注
>
> 高可用性セットアップで忘れた (nsroot) パスワードをリセットするには、ピアノードをシャットダウンすることをお勧めします。ピアノードがアクティブな場合、再起動後にノードが起動すると config sync がトリガーされるため、パスワードは上書きされます。

また、記事「CTX224027 」を読んで、NetScaler アプライアンスへの安全な SSH アクセスがどのように機能するかを確認してください。

外部ユーザー認証

January 11, 2024

NetScaler アプライアンスの認証サービスは、ローカルでも外部でもかまいません。外部ユーザ認証では、アプライアンスは LDAP、RADIUS、または TACACS+ などの外部サーバを使用してユーザを認証します。外部ユーザーを認証し、そのユーザーにアプライアンスへのアクセスを許可するには、認証ポリシーを適用する必要があります。NetScaler システム認証では、高度な認証ポリシーと高度なポリシー表現を使用します。詳細認証ポリシーは、パーティション化された NetScaler アプライアンスのシステムユーザー管理にも使用されます。

> 注
>
> アプライアンスがまだクラシックポリシーとその式を使用している場合は、アプライアンスの使用を停止し、クラシックポリシーの使用を高度なポリシーインフラストラクチャに移行する必要があります。

認証ポリシーを作成したら、それをシステムグローバルエンティティにバインドする必要があります。単一の認証ポリシーをシステムグローバルエンティティにバインドすることにより、外部認証サーバ（TACACS など）を設定できます。また、複数のポリシーをシステムグローバルエンティティにバインドすることによって、認証サーバのカスケードを構成することもできます。

> 注
>
> 外部ユーザーがアプライアンスにログインすると、システムは `ns.log` ファイルに「ユーザーが存在しません」というエラーメッセージを生成します。これは、システムが systemuser_systemcmdpolicy_binding コマンドを実行してユーザーの GUI を初期化するためです。

LDAP 認証 **(**外部 **LDAP** サーバーを使用**)**

NetScaler アプライアンスは、1 つ以上の LDAP サーバーでユーザーアクセスを認証するように構成できます。LDAP 認証には、Active Directory、LDAP サーバ、およびアプライアンスに同一のグループ名が必要です。文字と大文字小文字も同じである必要があります。

LDAP 認証ポリシーの詳細については、「LDAP 認証ポリシー 」トピックを参照してください。

デフォルトでは、LDAP 認証は SSL/TLS プロトコルを使用して保護されています。セキュア LDAP 接続には 2 つのタイプがあります。最初のタイプでは、LDAP サーバーは、クリア LDAP 接続を受け入れるために使用されるポートとは別のポートで SSL/TLS 接続を受け入れます。ユーザーが SSL/TLS 接続を確立すると、LDAP トラフィックは接続を介して送信できます。2 番目のタイプでは、非セキュアな LDAP 接続とセキュアな LDAP 接続の両方を許可し、単一のポートがサーバ上でそれを処理します。このシナリオでは、セキュリティで保護された接続を作成するには、クライアントはまず、クリア LDAP 接続を確立します。次に、**LDAP** コマンド **StartTLS** が接続を介してサーバーに送信されます。LDAP サーバーが StartTLS をサポートしている場合、接続は TLS を使用してセキュリティで保護された LDAP 接続に変換されます。

LDAP 接続のポート番号は次のとおりです。

- セキュリティで保護されていない LDAP 接続の場合は 389
- 636 セキュアな LDAP 接続の場合
- Microsoft のセキュアでない LDAP 接続の場合は 3268
- Microsoft のセキュア LDAP 接続の場合は 3269

StartTLS コマンドを使用する LDAP 接続では、ポート番号 389 を使用します。ポート番号 389 または 3268 がアプライアンスに構成されている場合、StartTLS を使用して接続を試みます。他のポート番号が使用されている場合、接続には SSL/TLS が使用されます。StartTLS または SSL/TLS を使用できない場合、接続は失敗します。

LDAP サーバーを構成する場合、英文字の大文字と小文字はサーバーとアプライアンスの大文字小文字と一致する必要があります。LDAP サーバーのルートディレクトリが指定されている場合、すべてのサブディレクトリも検索され、ユーザー属性が検索されます。大きなディレクトリでは、パフォーマンスに影響する可能性があります。このため、特定の組織単位（OU）を使用することをお勧めします。

次の表に、基本識別名 (DN) の例を示します。

LDAP サーバ	ベース **DN**
Microsoft Active Directory	DC=Citrix、DC= ローカル
Novell eDirectory	DC=Citrix、dc=net
IBM Directory Server	cn=users
Lotus Domino	OU= 都市、O=Citrix、C= 米国
Sun ONE ディレクトリ (旧 iPlanet)	ou= 人、DC=Citrix、dc=com

次の表に、バインド識別名 (DN) の例を示します。

LDAP サーバ	バインド **DN**
Microsoft Active Directory	CN= 管理者、CN= ユーザー、DC=Citrix、DC= ローカル
Novell eDirectory	cn= 管理者、DC=Citrix、dc=net
IBM Directory Server	LDAP_dn
Lotus Domino	cn=Notes 管理者、O=Citrix、C= 米国
Sun ONE ディレクトリ (旧 iPlanet)	uid=admin、OU= 管理者、OU= トポロジ管理、o=NetscapeRoot

LDAP サーバ	バインド **DN**
Microsoft Active Directory	CN= 管理者、CN= ユーザー、DC=Citrix、DC= ローカル

LDAP サーバ	バインド **DN**
Novell eDirectory	cn= 管理者、DC=Citrix、dc=net
IBM Directory Server	LDAP_dn
Lotus Domino	cn=Notes 管理者、O=Citrix、C= 米国
Sun ONE ディレクトリ (旧 iPlanet)	uid=admin、OU= 管理者、OU= トポロジ管理、o=NetscapeRoot

CLI を使用して **LDAP** ユーザー認証を構成する

外部ユーザーの LDAP 認証を構成するには、次の手順を実行します。

LDAP ポリシーを構成する

コマンドプロンプトで、次の操作を行います:

ステップ 1: LDAP アクションを作成します。

```
add authentication ldapAction <name> { -serverIP <ip_addr|ipv6_addr
|*> | { -serverName <string> } } >] [-authTimeout <positive_integer>]
 [-ldapBase <string>] [-ldapBindDn <string>] { -ldapBindDnPassword }
[-ldapLoginName <string>] [-groupAttrName <string>] [-subAttributeName
 <string>]
```

例:

```
add authentication ldapAction ldap_act -serverIP <IP> -authTimeout 30
 -ldapBase "CN=xxxxx,DC=xxxx,DC=xxx"-ldapBindDn "CN=xxxxx,CN=xxxxx,DC
=xxxx,DC=xxx"-ldapBindDnPassword abcd -ldapLoginName sAMAccountName -
groupattrName memberOf -subAttributeName CN
```

パラメータの説明については、「認証および認可コマンドのリファレンス」を参照してください。

ステップ 2: 従来の LDAP ポリシーを作成します。

```
add authentication ldapPolicy <name> <rule> [<reqAction>]
```

例:

```
add authentication ldappolicy ldap_pol_classic ns_true ldap_act
```

> 注

> クラシックまたは高度な LDAP ポリシーを使用して構成できますが、クラシックポリシーは NetScaler ADC 13.0 リリース以降では非推奨であるため、高度な認証ポリシーを使用することをお勧めします。

ステップ 3: 高度な LDAP ポリシーを作成する

```
add authentication Policy <name> <rule> [<reqAction>]
```

例:

```
add authentication policy ldap_pol_advance -rule true -action ldap_act
```

ステップ 4: LDAP ポリシーをシステムグローバルにバインドする

コマンドラインプロンプトで、次の操作を行います。

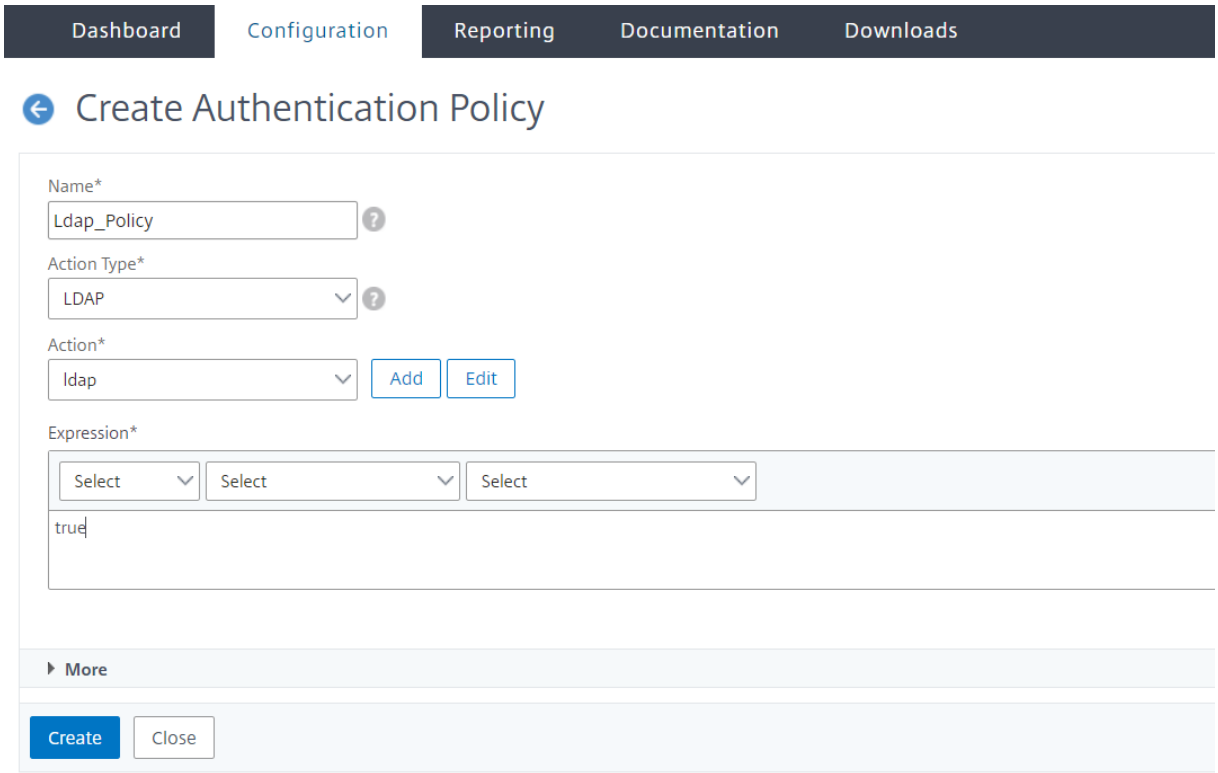
```
bind system global <policyName> [-priority <positive_integer]
```

例:

```
bind system global ldap_pol_advanced -priority 10
```

NetScaler GUI を使用して **LDAP** ユーザー認証を構成する

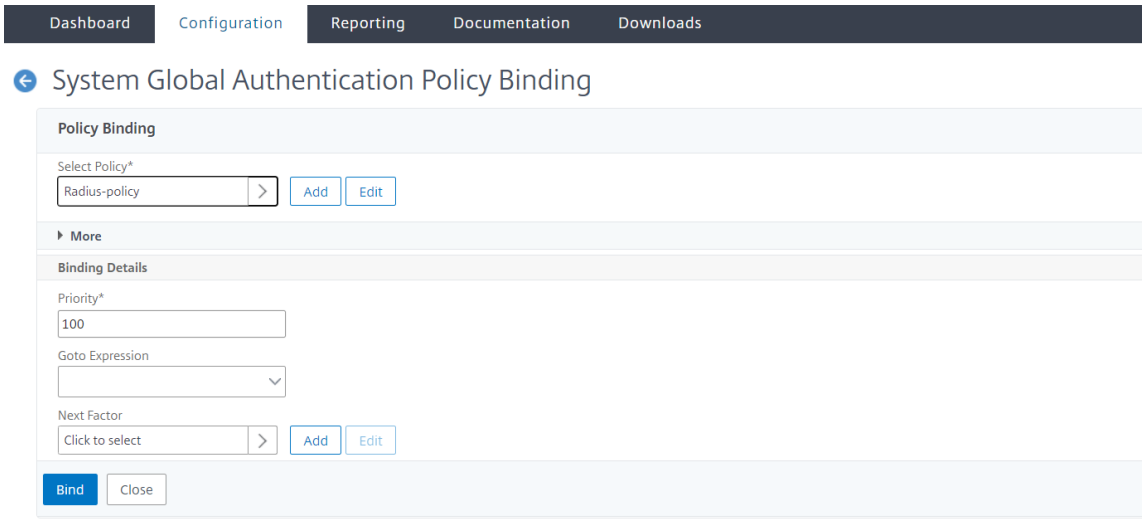
1. [システム] > [認証] > [詳細ポリシー] > [ポリシー] に移動します。
2. LDAP タイプの認証ポリシーを作成するには、[追加] をクリックします。
3. **[作成]** して **[閉じる]** をクリックします。



NetScaler GUI を使用して **LDAP** 認証用の認証ポリシーをシステムグローバルにバインドします

1. [システム] > [認証] > [詳細ポリシー] > [認証ポリシー] に移動します。
2. 詳細ペインで、[グローバルバインディング] をクリックして、システムグローバル認証ポリシーバインディングを作成します。
3. 「グローバルバインディング」をクリックします。
4. 認証プロファイルを選択します。
5. LDAP ポリシーを選択します。
6. [システムグローバル認証ポリシーのバインド] ページで、次のパラメータを設定します。

   a) 「ポリシー」を選択します。
   b) バインディングの詳細



7. [バインド] と [完了] をクリックします。
8. [グローバルバインディング] をクリックして、ポリシーがシステムグローバルにバインドされていることを確認します。

LDAP ディレクトリの属性を決定する

LDAP ディレクトリ属性の決定に助けが必要な場合は、Softerra の無料 LDAP ブラウザで簡単に検索できます。

LDAP ブラウザは、Softerra LDAP 管理者のウェブサイト<http://www.ldapbrowser.com>からダウンロードできます。ブラウザをインストールしたら、次の属性を設定します。

- LDAP サーバーのホスト名または IP アドレス。
- LDAP サーバーのポート。デフォルトは 389 です。

- ベース DN フィールドは空白のままにすることができます。
- LDAP ブラウザによって提供される情報は、[認証] タブに必要なベース DN を判断するのに役立ちます。
- Anonymous Bind チェックは、LDAP サーバがブラウザに接続するためにユーザクレデンシャルを必要とするかどうかを判断します。LDAP サーバがクレデンシャルを必要とする場合は、チェックボックスをオフのままにします。

設定が完了すると、LDAP ブラウザは左ペインにプロファイル名を表示し、LDAP サーバに接続します。

詳細については、LDAP のトピックを参照してください。

LDAP ユーザーに対するキーベース認証のサポート

キーベース認証では、SSH を使用して LDAP サーバー内のユーザーオブジェクトに格納されている公開キーのリストを取得できるようになりました。NetScaler アプライアンスは、ロールベース認証（RBA）プロセス中に LDAP サーバーからパブリック SSH キーを抽出する必要があります。取得した公開キーは SSH と互換性があり、RBA メソッドを使用してログインできる必要があります。

「認証の追加 ldapAction」コマンドと「認証の設定 ldapAction」コマンドに新しい属性「sshPublicKey」が導入されました。この属性を使用することで、次のようなメリットが得られます。

- 取得した公開キーを格納でき、LDAP アクションはこの属性を使用して LDAP サーバーから SSH キー情報を取得します。
- 最大 24 KB の属性名を抽出できます。

> 注
>
> LDAP などの外部認証サーバは、SSH キー情報の取得にのみ使用されます。認証目的では使用されません。

次に、SSH を介したイベントのフローの例を示します。

- SSH デーモンは、パスワードフィールドを空にして AAA_AUTHENTICATE 要求を認証、承認、および監査デーモンポートに送信します。
- LDAP が SSH 公開鍵を格納するように設定されている場合、認証、承認、および監査は、他の属性とともに`sshPublicKey`属性を使って応答します。
- SSH デーモンは、これらのキーをクライアントキーで検証します。
- SSH デーモンはリクエストペイロードでユーザー名を渡し、認証、承認、および監査は、このユーザーに固有のキーと汎用キーを返します。

sshPublicKey 属性を構成するには、コマンドプロンプトで次のコマンドを入力します。

- 追加操作では、`ldapAction` コマンドの設定中に「sshPublicKey」属性を追加できます。

```
add authentication ldapAction <name> ( -serverIP <ip_addr|ipv6_addr
|*> | ( -serverName <string> ) ) [-serverPort <port>] … [-Attribute1
 <string>] … [-Attribute16 <string>][-sshPublicKey <string>][-
authentication off]<!--NeedCopy-->
```


- 設定操作では、すでに追加されている ldapAction コマンドに「sshPublicKey」属性を設定できます。

```
set authentication ldapAction <name> [-sshPublicKey <string>][-
authentication off]<!--NeedCopy-->
```

RADIUS 認証（外部 RADIUS サーバーを使用）

NetScaler アプライアンスは、1 つ以上の RADIUS サーバーでユーザーアクセスを認証するように構成できます。RSA SecurID、SafeWord、または Gemalto Protiva 製品を使用している場合は、RADIUS サーバを使用してください。

RADIUS 認証ポリシーの詳細については、「RADIUS 認証」を参照してください。

構成によっては、ネットワークアクセスサーバの IP アドレス (NAS IP) またはネットワークアクセスサーバ識別子 (NAS ID) の使用が必要になる場合があります。RADIUS 認証サーバを使用するようにアプライアンスを設定する場合は、次のガイドラインに従ってください。

- NAS IP の使用を有効にすると、アプライアンスは、RADIUS 接続の確立に使用されるソース IP アドレスではなく、構成済みの IP アドレスを RADIUS サーバに送信します。
- NAS ID を構成すると、アプライアンスは RADIUS サーバーにこの識別子を送信します。NAS ID を構成しないと、アプライアンスは RADIUS サーバーにホスト名を送信します。
- NAS IP アドレスが有効になっている場合、アプライアンスは RADIUS サーバーとの通信に使用された NAS ID を無視します。

CLI を使用して RADIUS ユーザー認証を構成する

コマンドプロンプトで、次の操作を行います：

ステップ 1: RADIUS アクションを作成する

```
add authentication radiusaction <name> -serverip <ip> -radkey <key> -
radVendorID <id> -radattributetype <value>
```

ここで、
radVendorID RADIUS ベンダー ID 属性は、RADIUS グループの抽出に使用されます。
radAttributeType RADIUS グループ抽出に使用される RADIUS 属性タイプ。

例:

```
add authentication radiusaction RADserver531 rad_action -serverip
1.1.1.1 -radkey key123 -radVendorID 66 -radattributetype 6
```

ステップ 2: 従来の RADIUS ポリシーを作成します。

```
add authentication radiusPolicy <name> <rule> [<reqAction>]
```

例:


```
add authentication radiuspolicy radius_pol_classic ns_true radius_act
```

> 注
>
> クラシック RADIUS ポリシーまたは高度な RADIUS ポリシーを使用して設定できます。従来のポリシーは NetScaler 13.0 リリース以降では廃止されるため、Citrix では高度な認証ポリシーを使用することをお勧めします。

ステップ 3: 高度な RADIUS ポリシーを作成する

```
add authentication policy <policyname> -rule true -action <radius
action name>
```

例:

```
add authentication policy rad_pol_advanced -rule true -action radserver531rad_
```

ステップ 4: RADIUS ポリシーをシステムグローバルにバインドします。

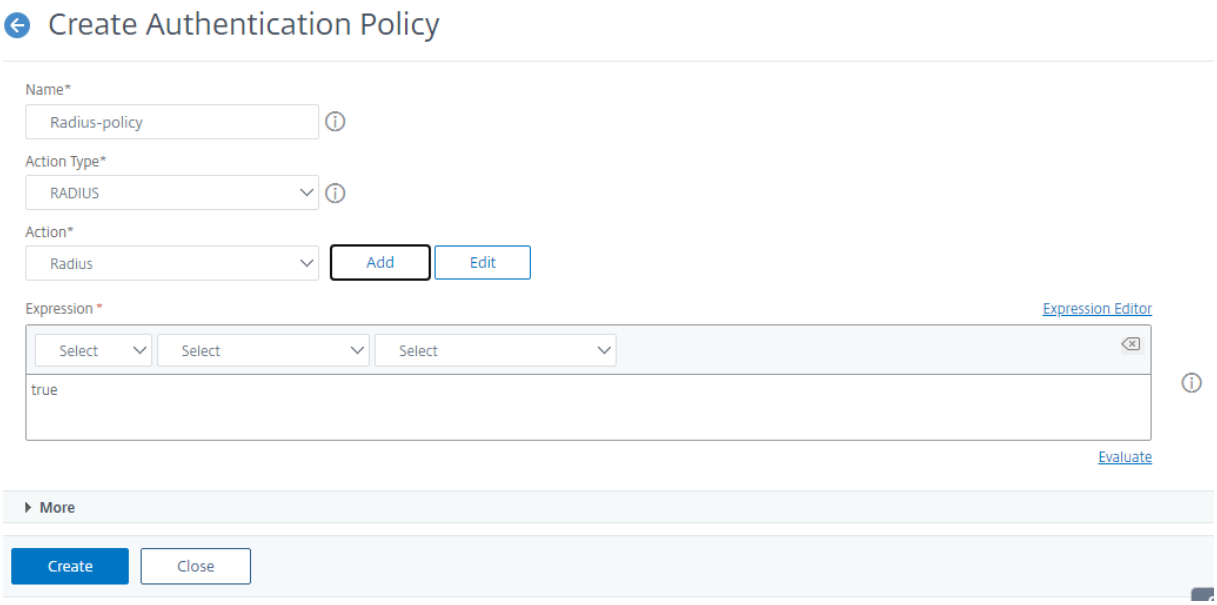
```
bind system global <policyName> -priority <positive_integer
```

例:

```
bind system global radius_pol_advanced -priority 10
```

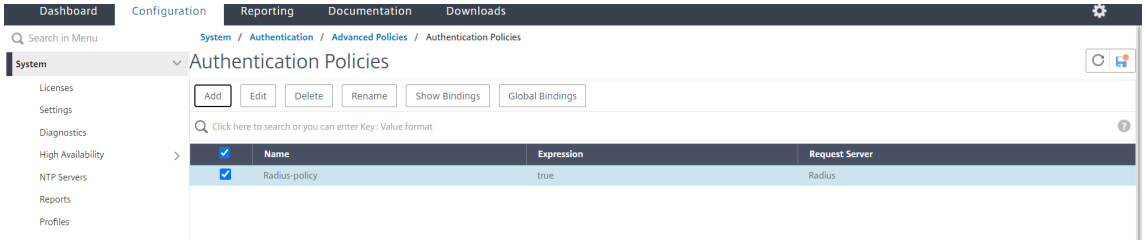
GUI を使用して **RADIUS** ユーザー認証を構成する

1. [システム] > [認証] > [詳細ポリシー] > [ポリシー] に移動します。
2. [追加] をクリックして、RADIUS タイプの認証ポリシーを作成します。
3. **[**作成**]** して **[**閉じる**]** をクリックします。

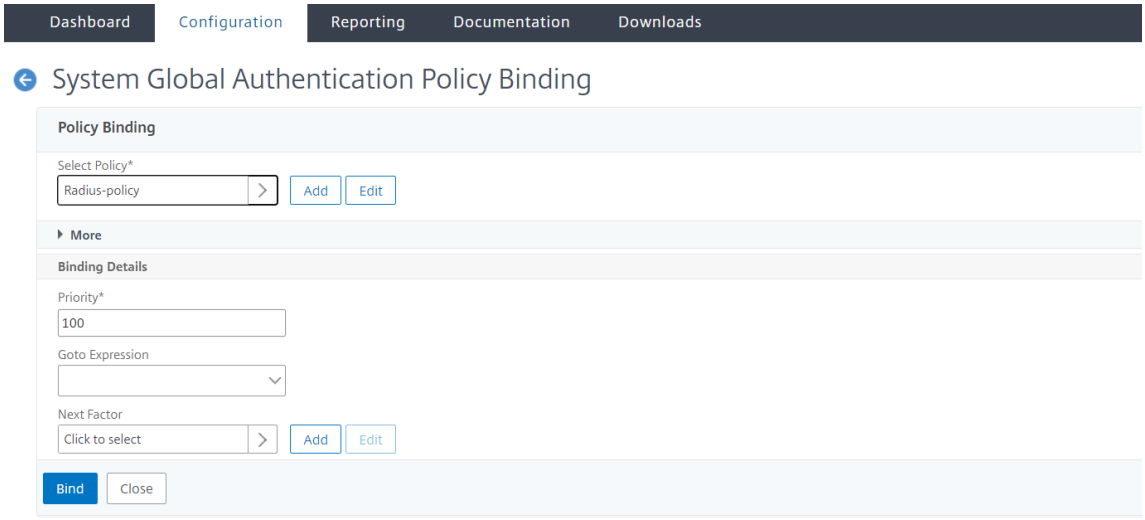


GUI を使用して、**RADIUS** 認証用に認証ポリシーをシステムグローバルにバインドします

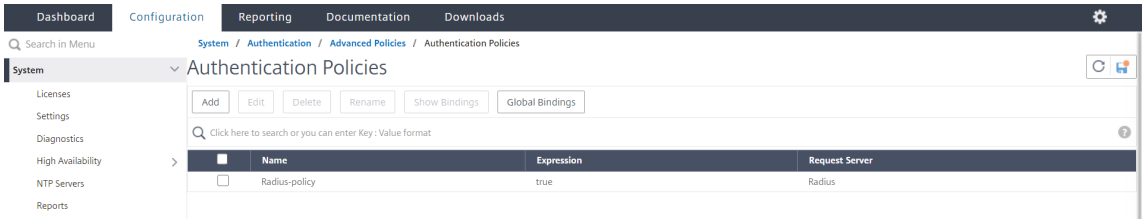
1. [システム] > [認証] > [詳細ポリシー] > [ポリシー] に移動します。
2. 詳細ペインで、[グローバルバインディング] をクリックして、システムグローバル認証ポリシーバインディングを作成します。
3. 「グローバルバインディング」をクリックします。



4. RADIUS を選択します。
5. [システムグローバル認証ポリシーのバインド] ページで、次のパラメータを設定します。
   a) ポリシーを選択します。
   b) バインディングの詳細。



6. [****** バインドして閉じる ******] をクリックします。
7. [グローバルバインディング] をクリックして、ポリシーがシステムグローバルにバインドされていることを確認します。



RADIUS ユーザー認証プロトコルを選択

NetScaler ADC アプライアンスは、ユーザー認証に次のようないくつかのプロトコルのいずれかを使用するように構成された RADIUS の実装をサポートします。

- パスワード認証プロトコル
- チャレンジハンドシェイク認証プロトコル (CHAP)
- Microsoft のチャレンジハンドシェイク認証プロトコル (MS-CHAP バージョン 1 およびバージョン 2)

展開が RADIUS 認証を使用するように構成され、RADIUS サーバにパスワード認証プロトコルが設定されている場合。RADIUS サーバに強力な共有秘密を割り当てることで、ユーザ認証を強化できます。強力な RADIUS 共有シークレットは、大文字と小文字、数字、句読点のランダムなシーケンスで構成され、長さは 22 文字以上です。可能であれば、ランダムな文字生成プログラムを使用して RADIUS 共有シークレットを判別します。

RADIUS トラフィックをさらに保護するには、各アプライアンスまたは仮想サーバーに異なる共有秘密を割り当てます。RADIUS サーバでクライアントを定義する場合、各クライアントに個別の共有秘密を割り当てることもできます。また、RADIUS 認証を使用する各ポリシーを個別に設定する必要があります。

IP アドレス抽出を構成する

RADIUS サーバから IP アドレスを抽出するようにアプライアンスを設定できます。ユーザが RADIUS サーバで認証されると、サーバは、ユーザに割り当てられているフレーム付き IP アドレスを返します。IP アドレス抽出の属性は次のとおりです。

- リモート RADIUS サーバが、アプライアンスにログオンしたユーザーの内部ネットワークからの IP アドレスを提供できるようにします。
- IP アドレスタイプを使用する任意の RADIUS 属性の設定（ベンダーエンコードを含む）を許可します。

IP アドレス抽出用に RADIUS サーバを設定する場合は、ベンダー ID と属性タイプを設定します。

ベンダー識別子により、RADIUS サーバは、RADIUS サーバ上に設定されている IP アドレスのプールから IP アドレスをクライアントに割り当てることができます。ベンダー ID と属性は、RADIUS クライアントと RADIUS サーバ間の関連付けを作成するために使用されます。ベンダー ID は、内部ネットワークの IP アドレスを提供する RADIUS 応答の属性です。ゼロの値は、属性がベンダーエンコードされていないことを示します。属性タイプは、RADIUS 応答のリモート IP アドレス属性です。最小値は 1 で、最大値は 255 です。

一般的な設定は、**RADIUS** 属性のフレーム *IP* アドレスを抽出することです。ベンダー ID がゼロに設定されているか、指定されていません。属性タイプは 8 に設定されています。

GUI を使用した **RADIUS** のグループ抽出

1. [システム] > [認証] > [詳細ポリシー] > [**Radius**] に移動し、ポリシーを選択します。
2. RADIUS ポリシーを選択または作成します。
3. **[**認証 **RADIUS** サーバの構成] ページで、次のパラメータを設定します。
   a) グループベンダー識別子
   b) グループ属性タイプ
4. 「**OK**」をクリックして「閉じる」をクリックします。

TACACS+ 認証（外部 **TACACS+** サーバを使用）

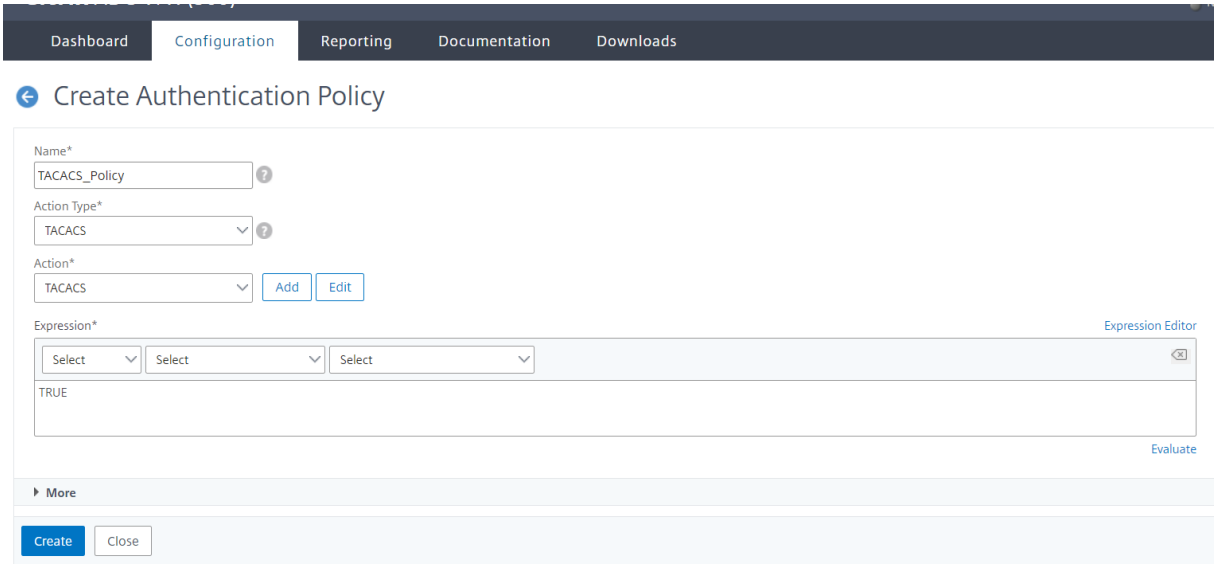
> 重要
>
> - 「clear ns config」コマンドを実行するときは、TACACS 関連の設定を変更しないことをお勧めします。
> - 詳細ポリシーに関する TACACS 関連の設定は、詳細ポリシーの「clear ns config」コマンドで`RBAconfig`パラメータが NO に設定されると、クリアされ、再適用されます。
> - 「構成のクリア」操作の一部として`RBAconfig`パラメータを「いいえ」に設定すると、NetScaler ADC は RBA 構成と TACACS ポリシーを保持するだけでなく、管理アクセスセッションも保持します。

TACACS+ サーバを認証用に設定できます。RADIUS 認証と同様に、TACACS+ は秘密キー、IP アドレス、およびポート番号を使用します。デフォルトのポート番号は 49 です。TACACS+ サーバを使用するようにアプライアンスを設定するには、サーバの IP アドレスと TACACS+ シークレットを指定します。使用中のサーバーのポート番号がデフォルトのポート番号 49 以外の場合にのみ、ポートを指定する必要があります。

詳細については、TACACS 認証を参照してください。

GUI を使用して TACACS + 認証を構成します

1. [システム] > [認証] > [詳細ポリシー] > [ポリシー] に移動します。
2. [**Add**] をクリックして、タイプ TACACS の認証ポリシーを作成します。
3. **[**作成**]** して **[**閉じる**]** をクリックします。



アプライアンスで TACACS+ サーバ設定を構成したら、ポリシーをシステムグローバルエンティティにバインドします。

CLI を使用して認証ポリシーをシステムグローバルエンティティにバインドする

認証ポリシーが設定されたら、ポリシーをシステムグローバルエンティティにバインドします。

コマンドラインプロンプトで、次の操作を行います。

```
bind system global <policyName> [-priority <positive_integer>]
```

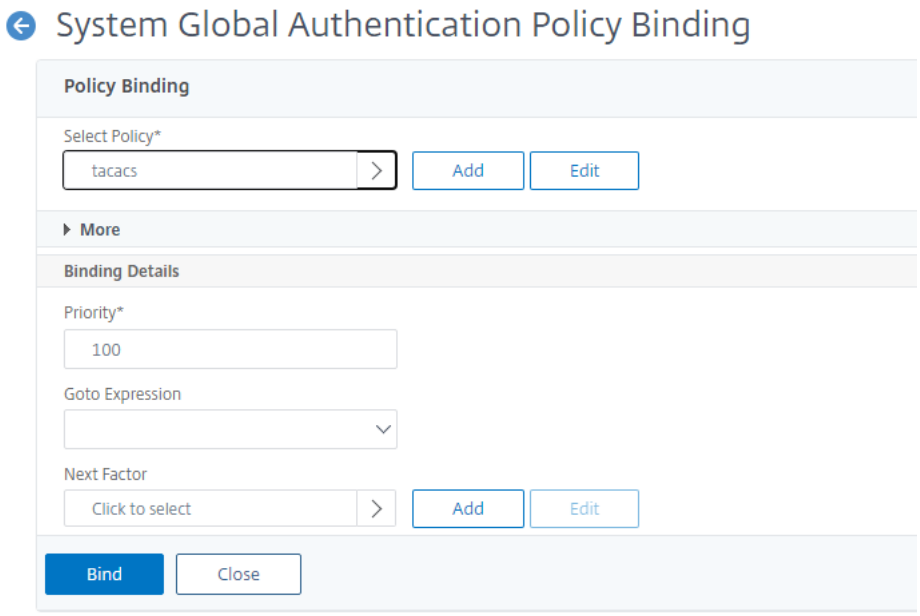
例**:**

```
bind system global pol_classic -priority 10
```

また、TACACS を使用した外部認証については、Citrix x の記事 CTX113820 を参照してください。

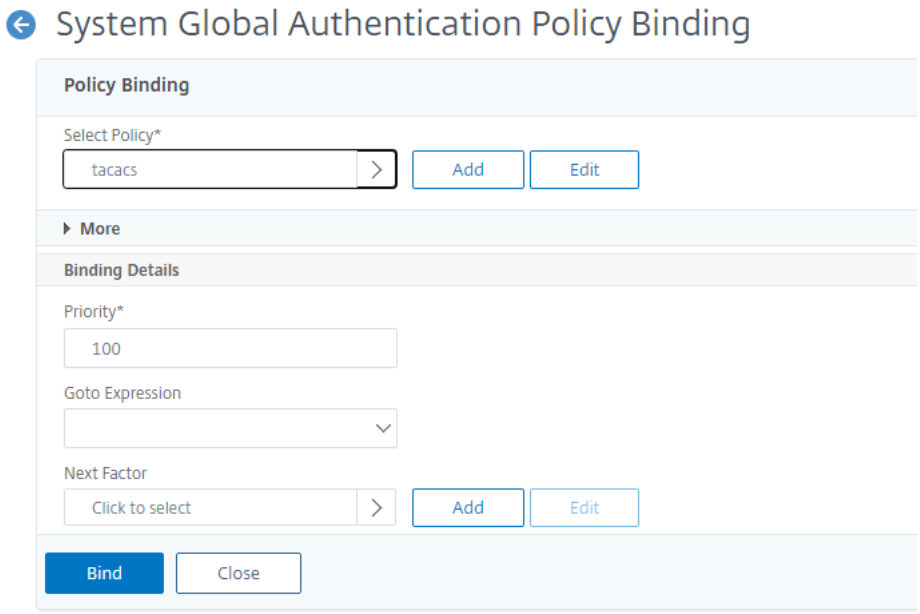
GUI を使用して、**RADIUS** 認証用に認証ポリシーをシステムグローバルにバインドします

1. システム > 認証 > 詳細ポリシー > 認証ポリシー **>** ポリシーに移動します。
2. 詳細ペインで、[グローバルバインディング] をクリックして、システムグローバル認証ポリシーバインディングを作成します。
3. 「グローバルバインディング」をクリックします。

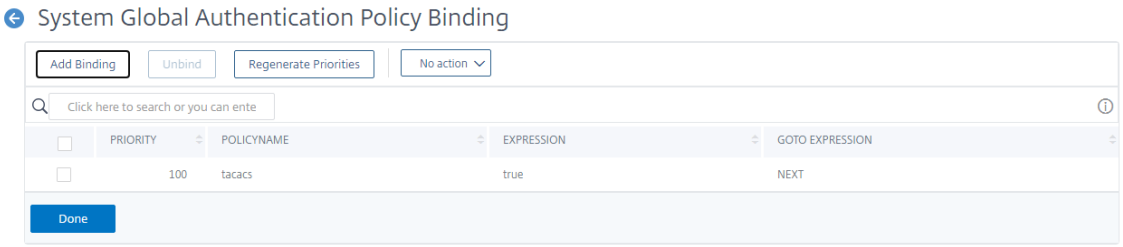


4. TACACS ポリシーを選択します。
5. [システムグローバル認証ポリシーのバインド] ページで、次のパラメータを設定します。

   a) 「ポリシー」を選択します。
   b) バインディングの詳細



6. [** バインドして閉じる **] をクリックします。
7. [グローバルバインディング] をクリックして、システムグローバルにバインドされたポリシーを確認します。



TACACS グループの抽出の詳細については、Citrix の記事 CTX220024を参照してください。

外部ユーザーのログオン試行の失敗回数を表示する

NetScaler アプライアンスは、NetScaler 管理コンソールに正常にログオンする前に少なくとも 1 回ログインに失敗すると、外部ユーザーに無効なログイン試行回数を表示します。

> 注
>
> 現在、NetScaler は、システムパラメーターで「PersistentLoginAttempts」パラメーターが有効になっている外部ユーザーのキーボードインタラクティブ認証のみをサポートしています。

コマンドプロンプトで入力します:

```
set aaa parameter -maxloginAttempts <value> -failedLoginTimeout <
value> -persistentLoginAttempts (ENABLED | DISABLED )]
```


例:

```
set aaa parameter -maxloginAttempts 5 -failedLoginTimeout 4 -persistentLoginAtt
 ENABLED
```

```
Following msg will be seen to external user when he tries 1 invalid
    login attempt before successfully login to the ADC management access
    .

Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.
##############################################################################

#

    #
#         WARNING: Access to this system is for authorized users only
             #
#          Disconnect IMMEDIATELY if you are not an authorized user!
              #
#

    #
##############################################################################


WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Mon Aug 24 17:09:00 2020 from 10.10.10.10

The number of unsuccessful login attempts since the last successful
    login : 1
Done
>
The number of unsuccessful login attempts since the last successful
    login : 1
Done
>
<!--NeedCopy-->
```

ローカルシステムユーザーの SSH キーベース認証

August 15, 2023

NetScaler アプライアンスへの安全なユーザーアクセスを確保するには、SSH サーバーの公開鍵認証を使用できます。SSH キーベースの認証は、次の理由から、従来のユーザー名またはパスワードベースの認証よりも優先されます。

- ユーザーパスワードよりも高い暗号化強度を提供します。

- 複雑なパスワードを覚える必要がなくなり、パスワードを使用すると起こり得るショルダーサーフィン攻撃を防ぐことができます。
- パスワードなしのログインが可能なため、自動化シナリオをより安全に行うことができます。

NetScaler は、公開鍵と秘密鍵の概念を適用することにより、SSH キーベースの認証をサポートしています。NetScaler の SSH キーベースの認証は、特定のユーザーまたはすべてのローカルユーザーに対して有効にできます。

> 注
>
> この機能は NetScaler ローカルユーザーでのみサポートされ、外部ユーザーにはサポートされません。

ローカルシステムユーザーの **SSH** キーベース認証

NetScaler アプライアンスでは、管理者は SSH キーベースの認証を設定して、安全なシステムアクセスを実現できます。ユーザーが秘密鍵を使用して NetScaler にログインすると、システムはアプライアンスで構成された公開鍵を使用してユーザーを認証します。

CLI を使用して **NetScaler** ローカルシステムユーザーの **SSH** キーベースの認証を構成します

次の構成は、NetScaler ローカルシステムユーザーのキーベース認証を構成するのに役立ちます。

1. 管理者の資格情報を使用して NetScaler アプライアンスにログオンします。
2. デフォルトでは、ファイルは `sshd_config` authorizedKeysFile **/nsconfig/ssh/authorized_keys** というパスにアクセスします。
3. **authorized_keys** ファイルに **/nsconfig/ssh/authorized_keys** というパブリックキーを追加します。`sshd_config` のファイルパスは/`etc`/`sshd_config`です。
4. `sshd_config` ファイルを/`nsconfig`にコピーして、アプライアンスを再起動しても変更が保持されるようにします。
5. `sshd` 次のコマンドを使用してプロセスを再起動できます。

```
    kill -HUP `cat /var/run/sshd.pid`
<!--NeedCopy-->
```

> 注
>
> authorized_keys ファイルがない場合は、まずファイルを作成してから公開鍵を追加する必要があります。ファイルに **authorized_keys** に対する以下の権限があることを確認してください。
>
> ```
> root@NetScaler# chmod 0644 authorized_keys
> ```

```
> shell
Copyright (c) 1992-2013 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993,
    1994
```



```
4 The Regents of the University of California. All rights reserved.
5 root@ns# cd /nsconfig/ssh
6 root@ns# vi authorized_keys
7 ### Add public keys in authorized_keys file
8 <!--NeedCopy-->
```

ローカルシステムユーザー向けのユーザー固有の **SSH** キーベースの認証

NetScaler アプライアンスでは、管理者がユーザー固有の SSH キーベースの認証を設定して、安全なシステムアクセスを実現できるようになりました。管理者は最初にsshd_configファイルでAuthorizedkeysfileオプションを設定し、次にシステムユーザーの公開鍵をauthorized_keysファイルに追加する必要があります。

> 注
>
> authorized_keys ファイルをユーザーが利用できない場合、管理者はまずファイルを作成してから公開鍵を追加する必要があります。

CLI を使用してユーザー固有の **SSH** キーベースの認証を設定する

以下の手順は、NetScaler ローカルシステムユーザーに対してユーザー固有の SSH キーベースの認証を構成するのに役立ちます。

1. 管理者の資格情報を使用して NetScaler アプライアンスにログオンします。
2. シェルプロンプトで、sshd_config ファイルにアクセスし、次の構成行を追加します。

   AuthorizedKeysFile ~/.ssh/authorized_keys

   > 注
   >
   > ~ はホームディレクトリで、ユーザーによって異なります。別のホームディレクトリに展開されます。

3. ディレクトリをシステムユーザーフォルダに変更し、authorized_keys ファイルに公開鍵を追加します。

   /var/pubkey/<username>/.ssh/authorized_keys

前の手順を完了したら、sshd 次のコマンドを使用してアプライアンス上のプロセスを再起動します。

```
1     kill -HUP `cat /var/run/sshd.pid`
2
3 <!--NeedCopy-->
```

> 注
>
> authorized_keys ファイルが利用できない場合は、まず作成してから公開鍵を追加する必要があります。


```
> shell
Copyright (c) 1992-2013 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993,
    1994
The Regents of the University of California. All rights reserved.
root@ns# cd /var/pubkey/<username>/
root@ns# ls
.ssh
root@ns# cd .ssh
root@ns# vi authorized_keys
### Add public keys in authorized_keys file

<!--NeedCopy-->
```

また、NetScaler ADC アプライアンスへのセキュアな SSH アクセスの仕組みについては、Citrix の記事 CTX109011 を読んでください 。

システムユーザーと外部ユーザーの二要素認証

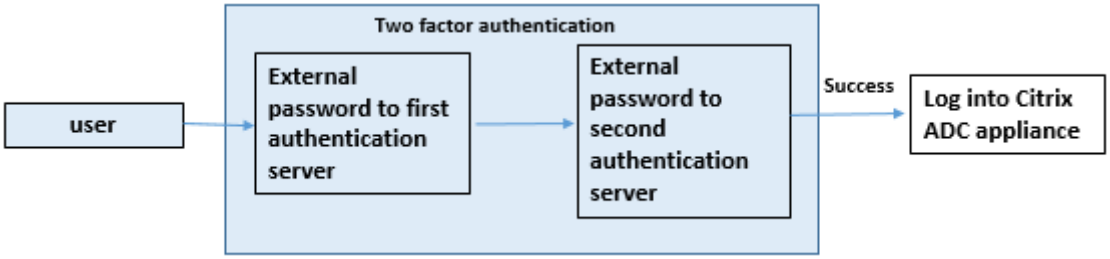
January 11, 2024

二要素認証は、NetScaler ADC アプライアンスがシステムユーザーを 2 つの認証レベルで認証するセキュリティメカニズムです。アプライアンスは、両方の認証レベルによるパスワードの検証が成功した後にのみ、ユーザーへのアクセスを許可します。ユーザーがローカルで認証される場合、ユーザープロファイルは NetScaler ADC データベースに作成する必要があります。ユーザーが外部で認証される場合、ユーザー名とパスワードは、外部認証サーバーに登録されているユーザー ID と一致する必要があります。

> 注:
>
> 二要素認証機能は、NetScaler 12.1 ビルド 51.16 以降でのみ機能します。

2 段階認証の仕組み

ユーザーが NetScaler ADC アプライアンスにログオンしようとしているとします。要求されたアプリケーションサーバーは、ユーザー名とパスワードを最初の外部認証サーバー (RADIUS、TACACS、LDAP、または AD) に送信します。ユーザー名とパスワードが検証されると、ユーザーは 2 段階目の認証を求められます。これで、ユーザーは 2 番目のパスワードを入力できます。両方のパスワードが正しい場合にのみ、ユーザーは NetScaler ADC アプライアンスにアクセスできます。次の図は、NetScaler ADC アプライアンスでの二要素認証の仕組みを示しています。



外部ユーザーとシステムユーザーの 2 要素認証を設定するさまざまなユースケースを以下に示します。

NetScaler ADC アプライアンスでは、さまざまな方法で二要素認証を構成できます。以下は、NetScaler ADC アプライアンスでの 2 要素認証のさまざまな構成シナリオです。

1. NetScaler、GUI、CLI、API、SSH にわたる二要素認証（2FA）。
2. システムユーザーの外部認証は有効で、ローカル認証は無効になっています。
3. システムユーザー向けのポリシーベースのローカル認証による外部認証が有効になっています。
4. ローカル認証が有効になっているシステムユーザーの外部認証は無効になっています。
5. システムユーザーの外部認証が有効で、ローカル認証が有効になっています。
6. 特定の LDAP ユーザーに対して外部認証が有効になっています

ユースケース **1: Citrix** の **ADC**、**GUI**、**CLI**、**API**、**SSH** インターフェイスにわたる二要素認証 **(2FA)**

二要素認証は有効になっており、GUI、API、SSH のすべての NetScaler ADC 管理アクセスで使用できます。

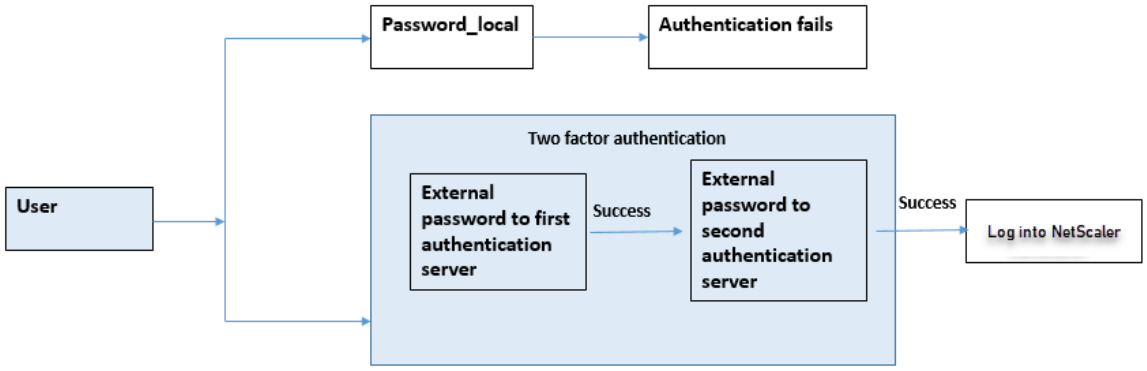
ユースケース **2: LDAP**、**RADIUS**、**Active Directory**、**TACACS** などの外部認証サーバーで **2** 要素認証がサポートされています

次の外部認証サーバーで、第 1 レベルおよび第 2 レベルのユーザー認証用に 2 要素認証を設定できます。

- RADIUS
- LDAP
- Active Directory
- TACACS

ユースケース **3:** システムユーザーの外部認証を有効にし、ローカル認証を無効にする

認証プロセスを開始するには、外部認証オプションを有効にし、システムユーザーのローカル認証を無効にします。



コマンドラインインターフェイスを使用して、次の手順を実行します。

1. LDAP ポリシーの認証アクションを追加
2. LDAP ポリシーの認証ポリシーを追加
3. RADIUS ポリシーの認証アクションを追加
4. RADIUS ポリシーの認証ポリシーを追加
5. 認証ログインスキーマの追加
6. 認証ポリシーラベルを RADIUS サーバに追加してバインドする
7. LDAP ポリシー用バインドシステムグローバル認証
8. システムパラメータのローカル認証を無効にする

LDAP サーバーの認証アクションを追加 **(**第 **1** レベルの認証**)**

コマンドプロンプトで入力します:

```
add authentication ldapaction <ldap action name> -serverip <IP> -
ldapbase <> -ldapbinddn <binddn name> -ldapbinddnpassword <password
>-ldaploginname <loginname> -groupattrname <grp attribute name> -
subAttributename <string>-ssoNameAttribute <string>
```

例**:**

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase
 base -ldapbindDn name -ldapbindDNpassword password -ldapLoginName
 name -groupAttrName name -subAttributeName name -ssoNameAttribute
name
```

LDAP サーバーの認証ポリシーの追加 **(**第 **1** レベルの認証**)**

コマンドプロンプトで入力します:

```
add authentication policy <ldap policy name> -rule true -action <ldap
 action name>
```


例:

```
add authentication policy pol1 -rule true -action ldapact1
```

RADIUS サーバの認証アクションの追加 **(**第 **2** レベル認証**)**

コマンドプロンプトで入力します:

```
add authentication radiusaction <rad action name> -serverip <rad
server ip> -radkey <key> -radVendorID <ID >-radattributetype <rad
attribute type>
```

例:

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123
 -radVendorID 1234 -radAttributeType 2
```

RADIUS サーバの認証ポリシーを追加 **(**第 **2** レベル認証**)**

コマンドプロンプトで入力します:

```
add authentication policy <radius policy name> -rule true -action <
rad action name>
```

例:

```
add authentication policy radpol11 -rule true -action radact1
```

認証ログインスキーマの追加

システムユーザーの「SingleAuth.xml」ログインスキーマを使用して、NetScaler ADC アプライアンスの 2 番目のパスワードを指定できます。コマンドプロンプトで入力します:

```
add authentication loginSchema <login schema name> -authenticationSchema
 LoginSchema/SingleAuth.xml
```

例:

```
add authentication loginSchema radschema -authenticationSchema
LoginSchema/SingleAuth.xml
```

認証ポリシーラベルを **RADIUS** サーバに追加してバインドする

コマンドプロンプトで入力します:


```
add authentication policylabel <labelName> [-type ( AAATM_REQ |
RBA_REQ )] [-comment <string>][-loginSchema <string>]

bind authentication policylabel <labelName> -policyName <string> -
priority <positive_integer> [-gotoPriorityExpression <expression>][-
nextFactor <string>]
```

例:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema
radschema

bind authentication policylabel label1 -policyName radpol11 -priority
 1
```

LDAP ポリシー用バインド認証システムグローバル

コマンドプロンプトで入力します:

```
bind system global ldappolicy -priority <priority> -nextFactor <
policy label name>
```

例:

```
bind system global pol11 -priority 1 -nextFactor label1
```

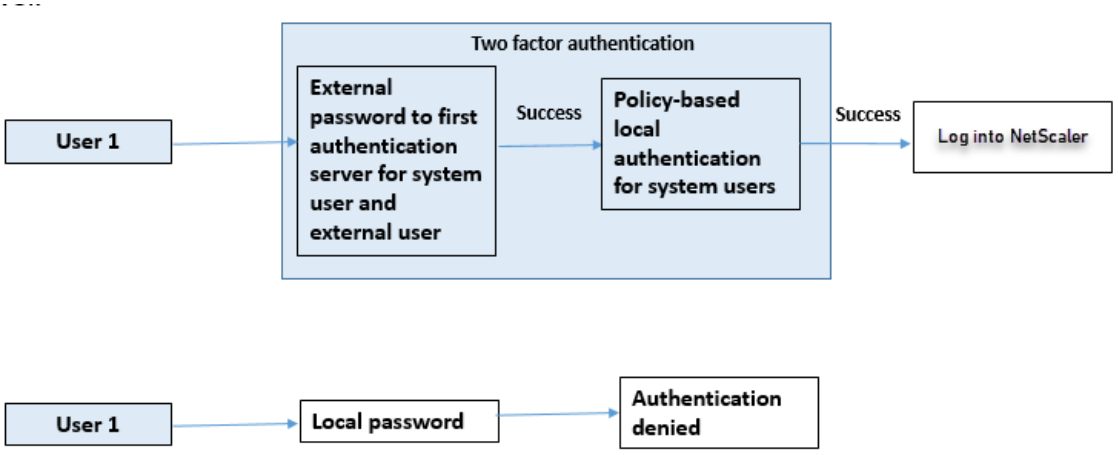
システムパラメータのローカル認証を無効にする

コマンドプロンプトで入力します:

```
set system parameter -localauth disabled
```

ユースケース **4:** ローカル認証ポリシーがアタッチされたシステムユーザーの外部認証を有効にする

このシナリオでは、ユーザーは 2 段階目のユーザー識別レベルでのローカル認証ポリシー評価による二要素認証を使用してアプライアンスにログオンできます。



コマンドラインインターフェイスを使用して、次の手順を完了します。

1. LDAP サーバーの認証アクションを追加
2. LDAP ポリシーの認証ポリシーを追加
3. ローカル認証ポリシーを追加
4. 認証ポリシーラベルを追加
5. LDAP ポリシーをシステムグローバルとしてバインドする
6. システムパラメータのローカル認証を無効にする

LDAP サーバーの認証アクションを追加 **(**第 **1** レベルの認証**)**

コマンドプロンプトで入力します:

```
add authentication ldapaction <ldap action name> -serverip <IP> -
ldapbase <> -ldapbinddn <binddn name> -ldapbinddnpassword <password
>-ldaploginname <loginname> -groupattrname <grp attribute name> -
subAttributename <string>-ssoNameAttribute <string>
```

例**:**

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase
 base -ldapbindDn name -ldapbindDNpassword password -ldapLoginName
 name -groupAttrName name -subAttributeName name -ssoNameAttribute
name -ssoNameAttribute name
```

LDAP サーバーの認証ポリシーの追加 **(**第 **1** レベルの認証**)**

コマンドプロンプトで入力します:

```
add authentication policy <ldap policy name> -rule true -action <ldap
 action name>
```


例:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase
 base -ldapbindDn name -ldapbindDNpassword password -ldapLoginName
 name -groupAttrName name -subAttributeName name -ssoNameAttribute
name
```

システムユーザー用のローカル認証ポリシーの追加 (第 2 レベル認証)

コマンドプロンプトで入力します:

```
add authentication policy <policy> -rule <rule> -action <action name>
```

例:

```
add authentication policy local_policy -rule true -action LOCAL
```

認証ポリシーラベルを追加してバインドする

コマンドプロンプトで入力します:

```
add authentication policylabel <labelName> [-type ( AAATM_REQ |
RBA_REQ )] [-comment <string>][-loginSchema <string>]
bind authentication policylabel <labelName> -policyName <string> -
priority <positive_integer> [-gotoPriorityExpression <expression>][-
nextFactor <string>]
```

> 注
>
> 管理アクセスの場合、ポリシータイプは RBA_REQ でなければなりません。

例:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema
radschema
bind authentication policylabel label1 -policyName radpol11 -priority
 1 -gotoPriorityExpression NEXT
```

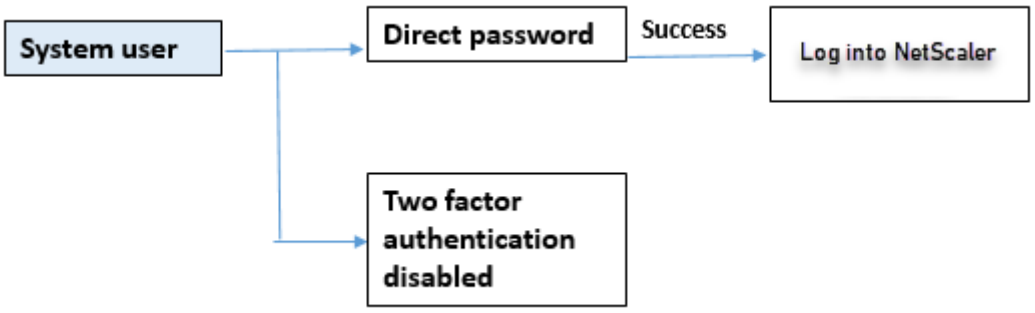
システムパラメータのローカル認証を無効にする

コマンドプロンプトで入力します:

```
set system parameter -localauth disabled
```


ユースケース **5:** システムユーザーの外部認証を無効にし、ローカル認証を有効にする

ユーザーの「ExternalAuth」が無効になっている場合は、そのユーザーが認証サーバーに存在しないことを示します。同じユーザー名のユーザーが外部認証サーバーに存在していても、ユーザーは外部認証サーバーで認証されません。ユーザーはローカルで認証されます。



システムユーザーパスワードを有効にし、外部認証を無効にするには

コマンドプロンプトで、次のように入力します:

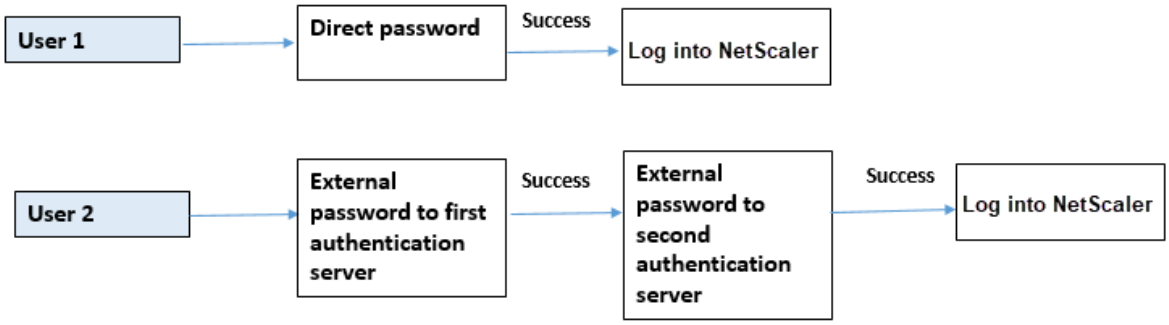
```
add system user <name> <password> -externalAuth DISABLED
```

例**:**

```
add system user user1 password1 -externalAuth DISABLED
```

ユースケース **6:** システムユーザーの外部認証が有効で、ローカル認証が有効になっている

ローカルパスワードを使用してシステムユーザーを認証するようにアプライアンスを構成すること。この認証に失敗すると、ユーザーは外部認証サーバーの外部認証パスワードを使用して 2 つのレベルで認証されます。



CLI を使用して次の手順を設定します。

1. LDAP サーバーの認証アクションを追加

2. LDAP ポリシーの認証ポリシーを追加
3. RADIUS ポリシーの認証アクションを追加
4. RADIUS ポリシーの認証ポリシーを追加
5. 認証ログインスキーマの追加
6. 認証ポリシーラベルを追加
7. ログインスキーマのバインド認証ポリシーラベル
8. RADIUS ポリシー用グローバル認証システムをバインドする
9. LDAP ポリシー用バインド認証システムグローバル

LDAP サーバーの認証アクションを追加

コマンドプロンプトで入力します:

```
add authentication ldapaction <ldap action name> -serverip <IP> -
ldapbase <> -ldapbinddn <binddn name> -ldapbinddnpassword <password
>-ldaploginname <loginname> -groupattrname <grp attribute name> -
subAttributename <>-ssoNameAttribute <>
```

例**:**

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase
 base -ldapbindDn name -ldapbindDNpassword password -ldapLoginName
 name -groupAttrName name -subAttributeName name -ssoNameAttribute
name
```

LDAP ポリシーの認証ポリシーを追加

コマンドプロンプトで入力します:

```
add authentication policy <policy name> --rule true -action <ldap
action name>
```

例**:**

```
add authentication policy pol1 -rule true -action ldapact1
```

RADIUS サーバの認証アクションを追加

コマンドプロンプトで入力します:

```
add authentication radiusaction <rad action name> -serverip <rad
server ip> -radkey <key> -radVendorID <ID >-radattributetype <rad
attribute type>
```


例:

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123
 -radVendorID 1234 -radAttributeType 2
```

RADIUS サーバ用の高度な認証ポリシーの追加

コマンドプロンプトで入力します:

```
add authentication policy <policy name> -rule true -action <rad
action name>
```

例:

```
add authentication policy radpol11 -rule true -action radact1
```

認証ログインスキーマの追加

SingleAuth.xml ログインスキーマを使用してログインページを表示し、第 2 レベルの認証でシステムユーザーを認証できます。

コマンドプロンプトで入力します:

```
add authentication loginSchema <name> -authenticationSchema <string>
```

例:

```
add authentication loginSchema radschema -authenticationSchema
LoginSchema/SingleAuth.xml
```

認証ポリシーラベルをユーザーログイン用の **RADIUS** 認証ポリシーに追加してバインドする

コマンドプロンプトで入力します:

```
add authentication policylabel <labelName> [-type ( AAATM_REQ |
RBA_REQ )] [-comment <string>][-loginSchema <string>]
```

例:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema
radschema
bind authentication policylabel <labelName> -policyName <string> -
priority <positive_integer> [-gotoPriorityExpression <expression>][-
nextFactor <string>]
```

例:


```
bind authentication policylabel label1 -policyName rad pol11 -priority
 1
```

バインド認証ポリシーグローバル

コマンドプロンプトで入力します:

```
bind system global [<policyName> [-priority <positive_integer>] [-
nextFactor <string>] [-gotoPriorityExpression <expression>]]
```

例:

```
bind system global radpol11 -priority 1 -nextFactor label11
```

ユースケース **7:** 特定の外部ユーザーに対してのみ外部認証を有効にする

LDAP アクションで設定した検索フィルターに従って選択的外部ユーザーを 2 要素認証で設定し、他のシステムユーザーは単一要素認証で認証します。

CLI を使用して次の手順を設定します。

1. LDAP サーバーの認証アクションを追加
2. LDAP ポリシーの認証ポリシーを追加
3. RADIUS ポリシーの認証アクションを追加
4. RADIUS ポリシーの認証ポリシーを追加
5. 認証ログインスキーマの追加
6. 認証ポリシーラベルを追加
7. ログインスキーマのバインド認証ポリシーラベル
8. RADIUS ポリシー用グローバル認証システムをバインドする

LDAP サーバーの認証アクションを追加

コマンドプロンプトで入力します:

```
add authentication ldapaction <ldap action name> -serverip <IP> -
ldapbase <> -ldapbinddn <binddn name> -ldapbinddnpassword <password
>-ldaploginname <loginname> -groupattrname <grp attribute name> -
subAttributename <>-ssoNameAttribute <>
```

例:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase
 base -ldapbindDn name -ldapbindDNpassword password -ldapLoginName
 name -groupAttrName name -subAttributeName name -ssoNameAttribute
name
```


LDAP ポリシーの認証ポリシーを追加

コマンドプロンプトで入力します:

```
add authentication policy <policy name> --rule true -action <ldap
action name>
```

例:

```
add authentication policy pol1 -rule true -action ldapact1
```

RADIUS サーバの認証アクションを追加

コマンドプロンプトで入力します:

```
add authentication radiusaction <rad action name> -serverip <rad
server ip> -radkey <key> -radVendorID <ID >-radattributetype <rad
attribute type>
```

例:

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123
 -radVendorID 1234 -radAttributeType 2
```

RADIUS サーバ用の高度な認証ポリシーの追加

コマンドプロンプトで入力します:

```
add authentication policy <policy name> -rule true -action <rad
action name>
```

例:

```
add authentication policy radpol11 -rule true -action radact1
```

認証ログインスキーマの追加

SingleAuth.xml ログインスキーマを使用して、アプライアンスがシステムユーザーを第 2 レベルの認証で認証するためのログインページを提供できます。

コマンドプロンプトで入力します:

```
add authentication loginSchema <name> -authenticationSchema <string>
```

例:


```
add authentication loginSchema radschema -authenticationSchema
LoginSchema/SingleAuth.xml
```

認証ポリシーラベルをユーザーログイン用の **RADIUS** 認証ポリシーに追加してバインドする

コマンドプロンプトで入力します:

```
add authentication policylabel <labelName> [-type ( AAATM_REQ |
RBA_REQ )] [-comment <string>][-loginSchema <string>]
```

例:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema
radschema
bind authentication policylabel <labelName> -policyName <string> -
priority <positive_integer> [-gotoPriorityExpression <expression>][-
nextFactor <string>]
```

例:

```
bind authentication policylabel label1 -policyName radpol11 -priority
```

バインド認証ポリシーグローバル

コマンドプロンプトで入力します:

```
bind system global [<policyName> [-priority <positive_integer>] [-
nextFactor <string>] [-gotoPriorityExpression <expression>]]
```

例:

```
bind system global radpol11 -priority 1 -nextFactor label11
```

検索フィルターを使用してグループユーザーに 2 要素認証なしで構成するには:

1. LDAP サーバーの認証アクションを追加
2. LDAP サーバーの認証ポリシーを追加
3. LDAP サーバー用グローバル認証システム

LDAP サーバーの認証アクションを追加

コマンドプロンプトで入力します:


```
add authentication ldapaction <ldap action name> -serverip <IP> -
ldapbase <> -ldapbinddn <binddn name> -ldapbinddnpassword <password
>-ldaploginname <loginname> -groupattrname <grp attribute name> -
subAttributename <>-searchFilter<>
```

例:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase
 base -ldapbindDn name -ldapbindDNpassword password -ldapLoginName
 name -groupAttrName name -subAttributeName name - searchFilter "
memberOf=CN=grp4,CN=Users,DC=aaatm-test,DC=com"
```

LDAP サーバーの認証ポリシーを追加

コマンドプロンプトで入力します:

```
add authentication policy <policy name> --rule true -action <ldap
action name>
```

例:

```
add authentication policy pol1 -rule true -action ldapact1
```

LDAP ポリシー用バインド認証システムグローバル

コマンドプロンプトで入力します:

```
bind system global ldappolicy -priority <priority> -nextFactor <
policy label name>
```

例:

```
bind system global pol11 -priority 1 -nextFactor label11
```

2 要素認証用にカスタマイズされたプロンプトメッセージを表示

/flash/nsconfig/loginschema/LoginSchemaの SingleAuth.xml ファイルで 2 要素パスワードフィールドを設定する場合

以下は SingleAuth.xml ファイルの抜粋です。’SecondPassword: ‘は 2 つ目のパスワードフィールド名で、ユーザーに 2 つ目のパスワードの入力を求められます。

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <AuthenticateResponse xmlns="http://citrix.com/authentication/response
     /1">
3 <Status>success</Status>
```



```
<Result>more-info</Result>
<StateContext/>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement><Credential><ID>login</ID><SaveID>ExplicitForms-Username</
    SaveID><Type>username</Type></Credential><Label><Text>
    singleauth_user_name</Text><Type>nsg-login-label</Type></Label><
    Input><AssistiveText>singleauth_please_supply_either_domain\
    username_or_user@fully.qualified.domain</AssistiveText><Text><Secret
    >false</Secret><ReadOnly>false</ReadOnly><InitialValue/><Constraint
    >.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password
    </SaveID><Type>password</Type></Credential><Label><Text>
    SecondPassword:</Text><Type>nsg-login-label</Type></Label><Input><
    Text><Secret>true</Secret><ReadOnly>false</ReadOnly><InitialValue/><
    Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>
    singleauth_first_factor</Text><Type>nsg_confirmation</Type></Label><
    Input/></Requirement>
<Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials
    </Type></Credential><Label><Text>singleauth_remember_my_password</
    Text><Type>nsg-login-label</Type></Label><Input><CheckBox><
    InitialValue>false</InitialValue></CheckBox></Input></Requirement>
<Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
    ><Label><Type>none</Type></Label><Input><Button>singleauth_log_on</
    Button></Input></Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

NetScaler GUI を使用した二要素認証の設定

1. NetScaler ADC アプライアンスにログオンします。
2. [システム] > [認証] > [詳細ポリシー**] > [**** ポリシー ******] に移動します。
3. [追加] をクリックして、第 1 レベルの認証ポリシーを作成します。
4. 「認証ポリシーの作成」ページで、次のパラメータを設定します。
    a) Name：ポリシーの名前
    b) アクションタイプ。LDAP、Active Directory、RADIUS、TACACS などとしてアクションタイプを選択してください
    c) 操作。ポリシーに関連付ける認証アクション (プロファイル)。既存の認証アクションを選択するか、プラス記号をクリックして適切なタイプのアクションを作成できます。
    d) 式。詳細なポリシー表現を提供してください。

5. [作成] をクリックし、[閉じる] をクリックします。

    a) 式。詳細なポリシー表現を提供してください。

6. ［作成］をクリックします。

7. [追加] をクリックして、第 2 レベルの認証ポリシーを作成します。

8. 「認証ポリシーの作成」ページで、次のパラメータを設定します。

    a) Name：ポリシーの名前
    b) アクションタイプ。LDAP、Active Directory、RADIUS、TACACS などとしてアクションタイプを選択してください
    c) 操作。ポリシーに関連付ける認証アクション (プロファイル)。既存の認証アクションを選択するか、+ アイコンをクリックして適切なタイプのアクションを作成できます。
    d) 式。詳細なポリシー表現を提供

9. [作成] をクリックし、[閉じる] をクリックします。

    a) 式。詳細なポリシー表現を提供してください。

10. ［作成］をクリックします。

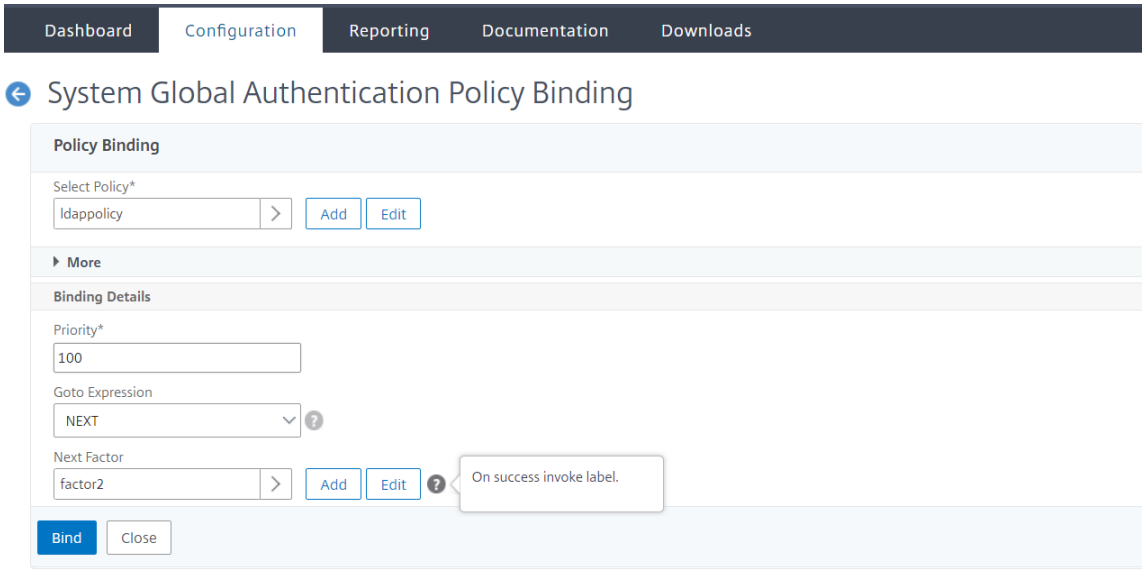
11. 「認証ポリシー」ページで、「グローバルバインディング」をクリックします。

12. 「グローバル認証ポリシーバインディングの作成」ページで、第 1 レベルの認証ポリシーを選択し、「バインドの追加」をクリックします。

13. ポリシーバインディングページで、認証ポリシーを選択し、次のポリシーバインディングパラメータを設定します。

    a) 次の要因。第 2 レベルの認証ポリシーラベルを選択します。

14. [****** バインドして閉じる ******] をクリックします。



15. ［完了］をクリックします。
16. NetScaler ADC アプライアンスにログオンして、第 2 レベルの認証を行います。これで、ユーザーは 2 番目のパスワードを入力できます。両方のパスワードが正しい場合にのみ、ユーザーは NetScaler ADC アプライアンスにアクセスできます。

> （注
>
> ）2 要素認証用に設定された TACACS は、「TACACSaction」コマンドで有効にしても、認可とアカウンティングをサポートしません。2 番目の要素は、認証目的でのみ使用されます。

また、NetScaler nFactor 認証のトピックの「二要素認証」を参照してください。

NetScaler 管理インターフェイスへのシステムユーザー認証の制限

August 15, 2023

CLI や API などの特定の NetScaler 管理インターフェイスへのシステムユーザーアクセスを制限できます。`allowedManagementInterface` パラメータは、許可される管理インターフェースのリストを定義します。たとえば、ユーザーまたはグループの管理インターフェイスが API に設定されている場合、グループ内のすべてのユーザーは、CLI ではなく API を介して NetScaler にアクセスできます。ただし、NetScaler GUI は API インターフェイスの一部であり、API 権限を持つユーザーも GUI インターフェイスにアクセスできます。

> 注:
>
> デフォルトでは、ユーザーとグループはすべてのインターフェイス（CLI、API、GUI）にアクセスできます。

パラメータは、ユーザーレベルまたはユーザーグループレベルのいずれかで設定できます。グループレベルで設定すると、その設定はグループ内のすべてのユーザーアカウントに適用されます。ユーザーが複数のグループにバインドされている場合、アプライアンスは集約された管理インターフェイスのセットへのアクセスを許可します。ユーザーレベルでパラメータを設定することにより、グループ内のユーザーの設定を指定できます。この場合、グループのユーザーレベル設定が設定されます。
特定のシナリオでは、顧客がユーザーアカウントの管理に外部認証サーバーを使用している場合、サーバーの詳細がアプライアンス上で設定されます。この場合、管理者は NetScaler アプライアンスでユーザーグループを作成し、(外部サーバーにグループ化された) すべてのユーザーをグループに追加できます。たとえば、外部サーバーで管理されているすべてのユーザーが api_users グループに追加され、管理者はアプライアンス上でグループをローカルに設定できます。

> 注:
>
> NetScaler アプライアンスでは、`nsroot` 管理者 (スーパーユーザー) のみがパラメーターを構成でき、システムユーザーはパラメーター設定を変更できません。

CLI を使用して NetScaler 管理インターフェイスへのユーザーアクセスを構成する

特定の管理インターフェースへのユーザーアクセスを許可するには、許可する管理インターフェースパラメーターを設定する必要があります。コマンドプロンプトで入力します。

```
set system group <groupName> [-allowedManagementInterface ( CLI | API )]
```

例:

```
set system group network_usergroup -allowedManagementInterface CLI
```

パラメータの説明については、「認証および認可コマンドのリファレンス」を参照してください。

GUI および CLI インターフェイスについて詳しくは、「NetScaler へのアクセス」トピックを参照してください。

TCP 構成

January 11, 2024

NetScaler ADC アプライアンスの TCP 構成は、TCP 設定の集合である TCP プロファイルと呼ばれるエンティティで指定できます。TCP プロファイルは、これらの TCP 設定を使用するサービスまたは仮想サーバに関連付けることができます。

デフォルトの TCP プロファイルを設定して、すべてのサービスおよび仮想サーバにグローバルに適用される TCP 設定を設定できます。

> 注:
>
> TCP パラメータがサービス、仮想サーバ、およびグローバルに異なる値を持つ場合、最も具体的なエンティティ（サービス）の値が最も優先されます。NetScaler ADC アプライアンスは、TCP を構成するための他のアプローチも提供します。詳細については読んでください。

サポートされる **TCP** 設定

NetScaler ADC アプライアンスは、次の TCP 機能をサポートしています。

なりすまし攻撃に対する **TCP** の防御

NetScaler ADC によるウィンドウ減衰の実装は、RFC 4953 に準拠しています。

明示的な輻輳通知（**ECN**）

アプライアンスは、ネットワークの輻輳ステータスの通知をデータの送信者に送信し、データの輻輳またはデータ破損に対する是正措置を講じます。ECN の NetScaler ADC の実装は、RFC 3168 に準拠しています。

タイムスタンプオプションを使用したラウンドトリップ時間測定 **(RTTM)**

TimeStamp オプションが機能するためには、接続の少なくとも片側（クライアントまたはサーバ）がそれをサポートしている必要があります。`TimeStamp`オプションの NetScaler ADC の実装は、RFC 1323 に準拠しています。

不適切な再送信の検出

この検出は、TCP 重複選択確認応答 (D-SACK) とフォワード RTO リカバリ (F-RTO) を使用して実行できます。スプリアス再送信がある場合、輻輳制御設定は元の状態に戻ります。D-SACK の NetScaler ADC 実装は RFC 2883 に準拠しており、F-RTO は RFC 5682 に準拠しています。

輻輳制御

この機能は、New-Reno、BIC、CUBIC、Nile、TCP ウエストウッドのアルゴリズムを使用します。

ウィンドウスケーリング

これにより、**TCP** 受信ウィンドウのサイズが最大値 65,535 バイトを超えて増加します。

ウィンドウスケーリングを構成する前に考慮すべきポイント

- スケールファクターには高い値を設定しないでください。これは、アプライアンスとネットワークに悪影響を及ぼす可能性があるためです。
- ウィンドウサイズを変更する理由が明確にわかっていない限り、ウィンドウのスケーリングは構成しません。
- TCP 接続内の両方のホストは、接続の確立時にウィンドウスケールオプションを送信します。接続の片側のみでこのオプションが設定されている場合、ウィンドウの拡大/縮小は接続に使用されません。
- 同じセッションの各接続は、独立したウィンドウスケーリングセッションです。たとえば、クライアントの要求とサーバーの応答がアプライアンスを通過する場合、アプライアンスとサーバーの間でウィンドウスケーリングを行わずに、クライアントとアプライアンスの間でウィンドウスケーリングを行うことができます。

TCP 最大輻輳ウィンドウ

ウィンドウサイズは、ユーザーが設定可能なサイズです。デフォルト値は 8190 バイトです。

選択的確認応答 **(SACK)**

これは、データレシーバー（NetScaler ADC アプライアンスまたはクライアント）を使用して、正常に受信されたすべてのセグメントについて送信者に通知します。

転送確認 **(FACK)**

この機能により、ネットワーク内の未処理のデータバイトの総数を明示的に測定し、送信者（NetScaler ADC またはクライアント）が再送信タイムアウト時にネットワークに注入されるデータ量を制御できるようにすることで、TCP の輻輳を回避できます。

TCP 接続多重化

この機能により、既存の TCP 接続の再利用が可能になります。NetScaler ADC アプライアンスは、確立された TCP 接続を再利用プールに保存します。クライアント要求が受信されるたびに、アプライアンスは再使用プール内の使用可能な接続をチェックし、接続が使用可能な場合は新しいクライアントに処理します。使用できない場合、アプライアンスはクライアント要求の接続を作成し、その接続を再利用プールに保存します。NetScaler ADC は、HTTP、SSL、および DataStream 接続タイプの接続多重化をサポートしています。

動的受信バッファリング

これにより、メモリとネットワークの状態に基づいて受信バッファを動的に調整できます。

MPTCP コネクション

クライアントと NetScaler ADC 間の MPTCP 接続。NetScaler ADC とバックエンドサーバー間では、MPTCP 接続はサポートされていません。MPTCP の NetScaler ADC 実装は、RFC 6824 に準拠しています。

コマンドラインインターフェイスを使用して、アクティブ MPTCP 接続やアクティブサブフロー接続などの MPTCP 統計情報を表示できます。

コマンドプロンプトで、次のコマンドのいずれかを入力して、MPTCP 統計情報の概要または詳細な概要を表示するか、統計情報の表示をクリアします。

1. `Stat MPTCP`
2. `Stat mptcp -detail`
3. `Clearstats basic`

> 注:
>
> MPTCP 接続を確立するには、クライアントと NetScaler ADC アプライアンスの両方が同じ MPTCP バージョンをサポートしている必要があります。NetScaler ADC アプライアンスをサーバーの MPTCP ゲートウェイとして使用する場合、サーバーは MPTCP をサポートする必要はありません。クライアントが新しい MPTCP 接続を開始すると、アプライアンスは SYN パケットの MP_CAPABALE オプションからクライアントの MPTPC バージョンを識別します。クライアントのバージョンがアプライアンスでサポートされているバージョンよりも高い場合、アプライアンスは SYN-ACK パケットの MP_CAPABALE オプションで最上位のバージョンを示します。その後、クライアントは下位バージョンにフォールバックし、ACK パケットの MP_CAPABALE オプションでバージョン番号を送信します。そのバージョンがサポート可能な場合、アプライアンスは MPTCP 接続を続行します。それ以外の場合、アプライアンスは通常の TCP にフォールバックします。NetScaler ADC アプライアンスはサブフロー（MP_JOIN）を開始しません。アプライアンスは、クライアントがサブフローを開始することを想定しています。

MPTCP での追加アドレスアドバタイズ **(ADD_ADDR)** のサポート

MPTCP 展開では、追加の仮想サーバの IP アドレスを持つ IP セットにバインドされた仮想サーバがある場合、追加のアドレスアドバタイズメント（ADD_ADDR）機能によって、IP セットにバインドされた仮想サーバの IP アドレスがアドバタイズされます。クライアントは、アドバタイズされた IP アドレスへの追加の`MP-JOIN`サブフローを開始できます。

MPTCP ADD_ADDR 機能について覚えておくべきポイント

- ADD_ADDRオプションの一部として最大 10 個の IP アドレスを送信できます。mptcpAdvertise パラメータが有効で 10 個以上の IP アドレスがある場合、10 の IP アドレスをアドバタイズした後、アプライアンスは残りの IP アドレスを無視します。
- MP-CAPABLE サブフローがプライマリ仮想サーバの IP アドレスではなく、IP セット内のいずれかの IP アドレスに対して行われる場合、仮想サーバの IP アドレスに対してmptcpAdvertiseパラメータが有効になっている場合、仮想サーバの IP アドレスがアドバタイズされます。

CLI を使用して追加の **VIP** アドレスをアドバタイズするアドレスアドバタイズメント（**ADD_ADDR**）機能を設定する

IPv4 と IPv6 の両方のアドレスタイプに対してMPTCP ADD_ADDR機能を設定できます。一般に、複数の IPv4 および IPv6 IP を単一の IP セットに接続でき、このパラメータは任意の IP アドレスのサブセットで有効にできます。ADD_ADDR 機能では、「mptcapAdvertise」オプションが有効になっている IP アドレスのみがアドバタイズされ、IP セットの残りの IP アドレスは無視されます。
ADD_ADDR機能を設定するには、次の手順を実行します。

1. IP セットを追加します。
2. MPTCP アドバタイズを有効にして、タイプ仮想サーバ IP（VIP）の IP アドレスを追加します。
3. IP アドレスを IP セットにバインドします。
4. 負荷分散仮想サーバーで IP セットを構成します。

IP セットの追加　コマンドプロンプトで入力します:

```
add ipset <name> [-td <positive_integer>]
<!--NeedCopy-->
```

例**:**

```
add ipset ipset_1
<!--NeedCopy-->
```

MPTCP アドバタイズが有効になっているタイプの仮想サーバ **IP**（**VIP**）の **IP** アドレスを追加します　コマンドで次のように入力します。

```
add ns ip <IPAddress>@ <netmask> [-mptcpAdvertise ( YES | NO )] -type <
    type>
<!--NeedCopy-->
```

例**:**

```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
```


IP アドレスを **IP** セットにバインドする　コマンドプロンプトで入力します:

```
bind ipset <name> <IPAddress>
<!--NeedCopy-->
```

例**:**

```
bind ipset ipset_1 10.10.10.10
```

IP セットを負荷分散仮想サーバーに設定する　コマンドプロンプトで入力します:

```
set lb vserver <name> [-ipset  <string>]
<!--NeedCopy-->
```

例**:**

```
set lb vserver lb1 -ipset ipset_1
<!--NeedCopy-->
```

サンプル構成**:**

```
Add ipset ipset_1
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
bind ipset ipset_1 10.10.10.10
set lb vserver lb1 -ipset ipset_1
<!--NeedCopy-->
```

ADD_ADDR 機能を使用してアドバタイズする外部 **IP** アドレスを構成する

アドバタイズされた IP アドレスが外部エンティティによって所有され、NetScaler ADC アプライアンスが IP アドレスをアドバタイズする必要がある場合は、「mptcpAdvertise」パラメータを state および ARP パラメータを無効にして有効にする必要があります。

外部 IP アドレスをアドバタイズするようにADD_ADDRを設定するには、次の手順を実行します。

1. MPTCP アドバタイズを有効にして、タイプ仮想サーバ IP（VIP）の IP アドレスを追加します。
2. IP アドレスを IP セットにバインドします。
3. IP セットを負荷分散仮想サーバーとバインドする

MPTCP アドバタイズを有効にして、仮想サーバー **IP (VIP)** タイプの外部 **IP** アドレスを追加する　コマンドプロンプトで入力します:

```
add ns ip <IPAddress>@ <External-IP-mask -type VIP> [-mptcpAdvertise (
    YES | NO )] -type <type> -state DISABLED -arp DISABLED
<!--NeedCopy-->
```


例:

```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP -
state DISABLED -arp DISABLED
```

IP アドレスを **IP** セットにバインドする　コマンドプロンプトで入力します:

```
bind ipset <name> <IPAddress>
<!--NeedCopy-->
```

例:

```
bind ipset ipset_1 10.10.10.10
```

IP セットを負荷分散仮想サーバーに設定する　コマンドプロンプトで入力します:

```
set lb vserver <name> [-ipset  <string>]
<!--NeedCopy-->
```

例:

```
set lb vserver lb1 -ipset ipset_1
```

サンプル構成:

```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
   state DISABLED -arp DISABLED
bind ipset ipset_1 10.10.10.10
set lb vserver lb1 -ipset ipset_1
<!--NeedCopy-->
```

NetScaler GUI を使用して、**IP** アドレスを **MPTCP** 対応クライアントにアドバタイズする

MPTCP 対応クライアントに IP アドレスをアドバタイズするには、次の手順を実行します。

1. [システム] **>** [ネットワーク] **>** [**IP**] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [**IP** アドレスの作成] ページで、[**MPTCP** アドバタイズ] チェックボックスをオンにしてパラメータを設定します。デフォルトでは、無効になっています。

TCP/IP パスオーバーレイオプションの抽出とクライアント **IP HTTP** ヘッダーの挿入

TCP/IP パスオーバーレイを抽出し、クライアント IP HTTP ヘッダーを挿入します。オーバーレイネットワークを介したデータ転送では、接続終了または送信元クライアントの IP アドレスが失われるネットワークアドレス変換 (NAT) を使用することがよくあります。これを回避するために、NetScaler ADC アプライアンスは TCP/IP パスオーバーレ

イオプションを抽出し、ソースクライアントの IP アドレスを HTTP ヘッダーに挿入します。ヘッダーに IP アドレスが含まれていると、Web サーバーは接続を確立したソースクライアントを識別できます。抽出されたデータは TCP 接続の存続期間にわたって有効であるため、ネクストホップホストがオプションを再度解釈する必要がなくなります。このオプションは、client-IP 挿入オプションが有効になっている Web サービスに対してのみ適用できます。

TCP セグメンテーションオフロード

TCP セグメンテーションを NIC にオフロードします。オプションを「自動」に設定すると、NIC がサポートされている場合、TCP セグメンテーションは NIC にオフロードされます。

TCP ハンドシェイクの **Cookie** をクライアントと同期させる

これは、SYN フラッド攻撃に抵抗するために使用されます。クライアントとの TCP `SYNCOOKIE` ハンドシェイクのメカニズムを有効または無効にできます。`SYNCOOKIE`を無効にすると、`SYN`が NetScaler ADC アプライアンスでの攻撃保護を防止します。

MSS の学習により、アプライアンスに設定されているすべての仮想サーバーで **MSS** ラーニングが有効になります

サポート可能な **TCP** パラメータ

次の表に、NetScaler ADC アプライアンスで構成された TCP パラメータとそのデフォルト値のリストを示します。

| パラメーター | デフォルト値 | 説明 |
|–|–|–|
| ウィンドウ管理 |
TCP 遅延-ACK タイマー	100 ミリ秒	TCP 遅延 ACK のタイムアウト（ミリ秒単位）。
TCP 最小再送信タイムアウト（RTO）（ミリ秒）	1000 ミリ秒	10 ミリ秒単位で指定された最小再送信タイムアウト（ミリ秒単位）（10 で割った場合は整数である必要があります）
キープアライブプローブを開始する前の接続のアイドル時間	900 秒	アイドルタイムアウト時に TCP 確立された接続をサイレントにドロップアイドルタイムアウト時に確立した接続
TCP タイムスタンプオプション	無効	タイムスタンプオプションを使用すると、正確な RTT 測定が可能になります。TCP タイムスタンプオプションを有効または無効にします。
マルチパス TCP セッションタイムアウト	0 秒	MPTCP セッションタイムアウト（秒単位）。この値が設定されていない場合は、アイドル状態になります。MPTCP セッションは、仮想サーバーのクライアントのアイドルタイムアウト後にフラッシュされます。
アイドルタイムアウト時に半閉じた接続をサイレントにドロップ	0 秒	アイドルタイムアウトで TCP ハーフクローズ接続をサイレントにドロップします。
アイドルタイムアウト時に確立された接続をサイレントにドロップする	無効	アイドルタイムアウト時に TCP 確立した接続をサイレントにドロップする

| メモリ管理 |
TCP バッファサイズ	131072 バイト	TCP バッファサイズは、NetScaler 受信バッファサイズです。このバッファサイズは、NetScaler ADC からクライアントとサーバーに通知され、NetScaler ADC にデータを送信する機能を制御します。デフォルトのバッファサイズは 8K で、通常、内部サーバファームと通信するときにこの値を増やしても安全です。バッファサイズは、SSL エンドポイントの場合は 40 K に設定され、圧縮の場合は 96 K に設定されるなど、NetScaler 実際のアプリケーション層の影響を受けます。** 注：** 動的調整を行うには、バッファサイズ引数を設定する必要があります。
TCP 送信バッファサイズ	8190 バイト	TCP 送信バッファサイズ
TCP ダイナミック受信バッファリング	無効	ダイナミック受信バッファリングを有効または無効にします。有効にすると、メモリやネットワークの状態に基づいて受信バッファを動的に調整できます。** 注:** 動的調整を行うには、buffer size 引数を設定する必要があります。
TCP 最大輻輳ウィンドウ (CWND)	524288 バイト	TCP 最大輻輳ウィンドウ
ウィンドウのスケーリングステータス	有効に	ウィンドウのスケーリングを有効または無効にします。
ウィンドウのスケーリング係数	8	新しいウィンドウサイズの計算に使用される係数。この引数は、ウィンドウのスケーリングが有効になっている場合にのみ必要です。
接続セットアップ
キープアライブプローブ	無効	定期的な TCP キープアライブ（KA）プローブを送信して、ピアがまだアップしているかどうかを確認します。
キープアライブプローブを開始する前の接続のアイドル時間	900 秒	キープアライブ（KA）プローブを送信する前の接続がアイドル状態になるまでの時間（秒）。
キープアライブプローブ間隔	75 秒	ピアが応答しない場合、次のキープアライブ（KA）プローブの前の時間間隔（秒）。
接続をドロップする前に逃すキープアライブプローブの最大数。	3	ピアがダウンしていると仮定する前に、確認応答がないときに送信されるキープアライブ（KA）プローブの数。
RST ウィンドウ減衰 (なりすまし保護)。	無効	スプーフィングから保護するために、RST ウィンドウ減衰を有効または無効にします。有効にすると、シーケンス番号が無効の場合、応答は是正 ACK となります。
最後に確認応答されたシーケンス番号で RST を受け入れます。	ENABLED
データ転送
PUSH パケットの即時 ACK	ENABLED	PUSH フラグを使用して TCP パケットの受信時に即時肯定応答（ACK）を送信します。
MSS あたりの最大パケット数	0	TCP データセグメントで許可するオクテットの最大数
ナーグルのアルゴリズム	無効	Nagle のアルゴリズムは、TCP 伝送における小さなパケットの問題と戦っています。Telnet などのリアルタイムエンジンなどのアプリケーションは、すべてのキーストロークを反対側に渡す必要があるため、小さなパケットを作成することがよくあります。Nagle のアルゴリズムで NetScaler は、このような小さなパケットをバッファリングし、接続効率を高めるために一緒に送信することができます。このアルゴリズムは、NetScaler 他の TCP 最適化手法と一緒に機能する必要があります。
バーストで許可される TCP セグメントの最大数	10 MSS	バーストで許可される TCP セグメントの最大数
キューに入れる順序外パケットの最大数	300	順不同パケットキューの最大サイズ。値 0 は制限がないことを意味します。

| 輻輳制御 |
TCP フレーバー	キュービック
初期輻輳ウィンドウ (cwnd) 設定	4 MSS	サーバへの TCP リンクで未処理になることができる TCP パケット数の初期の上限
TCP 明示的輻輳通知 (ECN)| 無効	明示的な輻輳通知（ECN）は、パケットをドロップすることなく、ネットワーク輻輳のエンドツーエンド通知を提供します。
TCP 最大輻輳ウィンドウ (CWND)	524288 バイト	TCP は輻輳ウィンドウ（CWND）を維持し、エンドツーエンドで送信される可能性のある未確認パケットの総数を制限します。TCP では、輻輳ウィンドウは、いつでも未処理になる可能性のあるバイト数を決定する要素の 1 つです。輻輳ウィンドウは、送信者と受信者の間のリンクが多すぎるトラフィックで過負荷にならないようにする手段です。これは、リンク上に存在する輻輳の量を推定することによって計算されます。
TCP ハイブリッドスタート (HyStart)	8 バイト
TCP 最小再送信タイムアウト（RTO）（ミリ秒）	1000	最小再送信タイムアウト（ミリ秒単位）。10 ミリ秒単位で指定します（10 で割った場合は、値が整数になる必要があります）。
TCP デュパックのしきい値	無効
バーストレート制御	3	TCP バーストレート制御無効/固定/ダイナミック。FIXED には TCP レートを設定する必要があります
TCP レート	無効 |TCP 接続ペイロード送信レート（KB/秒）
TCP レート最大キュー	0	BurstrateControl が使用されている場合の最大接続キューサイズ（バイト単位）。
MPTCP
マルチパス TCP	無効	Multipath TCP（MPTCP）は、マルチパス TCP サービスを提供するための通常の TCP に対する拡張のセットです。これにより、トランスポート接続を複数のパスで同時に動作させることができます。
事前に確立されたサブフロー上のマルチパス TCP ドロップデータ	無効	事前確立されたサブフローにデータをサイレントドロップするのを有効または無効にします。有効の場合、DSS データパケットは、事前に確立されたサブフローでデータが受信されたときに、接続をドロップするのではなく、サイレントにドロップされます。
マルチパス TCP ファストオープン	無効	マルチパス TCP ファストオープンを有効または無効にします。有効の場合、DSS データパケットは SYN ハンドシェイクの 3 番目の ACK を受信する前に受け入れられます。
マルチパス TCP セッションタイムアウト	0 秒	MPTCP セッションタイムアウト（秒単位）。この値が設定されていない場合、仮想サーバーのクライアントのアイドルタイムアウト後にアイドル状態の MPTCP セッションがフラッシュされます。
セキュリティ
SYN スプーフィング保護	無効	スプーフィングから保護するために、無効な SYN パケットのドロップを有効または無効にします。無効にすると、SYN パケットが受信されたときに確立された接続がリセットされます。
TCP Syncookie| 無効	これは、SYN フラッド攻撃に抵抗するために使用されます。クライアントとの TCP ハンドシェイクの SYNCOOKIE メカニズムを有効または無効にします。SYNCOOKIE を無効にすると、NetScaler ADC アプライアンスでの SYN 攻撃保護が防止されます。
損失検出とリカバリ
重複選択的謝辞 (DSACK)	ENABLED	NetScaler ADC アプライアンスは、重複選択確認（DSACK）を使用して、再送信がエラーで送信されたかどうかを判断します。

フォワード RTO リカバリ (FRTO)	ENABLED	スプリアス TCP 再送信タイムアウトを検出します。タイムアウトによってトリガーされた最初の未確認のセグメントを再送信した後、TCP 送信者のアルゴリズムは、着信確認を監視して、タイムアウトがスプリアスであるかどうかを判断します。次に、新しいセグメントを送信するか、未確認のセグメントを再送信するかを決定します。このアルゴリズムは、別の不要な再送信を効果的に回避するのに役立ち、スプリアスタイムアウトの場合の TCP パフォーマンスが向上します。
TCP 転送確認応答 (FACK)	ENABLED	FACK（フォワード ACK）を有効または無効にします。
選択的確認応答 (SACK) ステータス	ENABLED	TCP SACK は、全体的なスループット容量を低下させる複数のパケット損失の問題に対処します。選択的確認応答を使用すると、受信者は正常に受信されたすべてのセグメントについて送信者に通知できるため、送信者は失われたセグメントのみを再送信できるようになります。この手法は、NetScaler ADC が全体的なスループットを向上させ、接続待ち時間を短縮するのに役立ちます。
再送信あたりの最大パケット数	1	NetScaler ADC が、1 回の試行で再送信するパケット数を制御できるようにします。NetScaler ADC が部分的な ACK を受信し、再送信を行う必要がある場合、この設定が考慮されます。これは RTO ベースの再送信には影響しません。
TCP 遅延-ACK タイマー	100 ミリ秒	TCP 遅延 ACK のタイムアウト（ミリ秒単位）
TCO 最適化
TCP 最適化モード	トランスペアレント	TCP 最適化モードトランスペアレント/エンドポイント
適応型 TCP 最適化の適用	無効	アダプティブ TCP 最適化の適用
TCP セグメンテーションオフロード	自動	TCP セグメンテーションを NIC にオフロードします。AUTOMATIC に設定すると、NIC がサポートしている場合、TCP セグメンテーションは NIC にオフロードされます。
ACK 集約	無効	ACK 集約を有効または無効にする
TCP タイム待機 (または time_Wait)	40 秒	閉じた TCP 接続を解放するまでの経過時間
RST 上のクライアントとサーバーのリンクを解消する	無効	クライアントとサーバーの接続をリンク解除する (存在する場合)
相手側に送信される未処理のデータ。|

> 注:
> HTTP/2 が有効な場合は、TCP プロファイルで TCP 動的受信バッファリングパラメータを無効にすることをお勧めします。

グローバル **TCP** パラメータの設定

NetScaler ADC アプライアンスでは、すべての NetScaler ADC サービスと仮想サーバーに適用される TCP パラメータの値を指定できます。これは、以下を使用して実行できます。

- デフォルトの TCP プロファイル
- グローバル TCP コマンド
- TCP バッファリング機能

> 注:
>
> - set ns tcppParam コマンドの`recvBuffSize`パラメータは、リリース 9.2 以降から廃止されまし

> た。以降のリリースでは、set ns tcpProfile コマンドの`bufferSize`パラメータを使用してバッファサイズを設定します。`recvBuffSize`パラメータが廃止されるリリースにアップグレードすると、`bufferSize`パラメータはデフォルト値に設定されます。
>
> - TCP プロファイルを設定する際は、TCP `buffersize`パラメータが`httppipelinebuffersize`パラメータと同じかそれ以下であることを確認してください。
> TCP プロファイルの`buffersize`パラメーターが HTTP プロファイルの`httppipelinebuffersize`パラメーターよりも大きい場合、TCP ペイロードが蓄積され、HTTP パイプラインのバッファーサイズを超える可能性があります。その結果、NetScaler ADC アプライアンスは TCP 接続をリセットします。

デフォルトの **TCP** プロファイル

`nstcp_default_profile`という TCP プロファイルは、サービスレベルまたは仮想サーバレベルで TCP 設定が提供されない場合に使用される TCP 設定を指定するために使用されます。

> 注:
>
> - すべての TCP パラメータをデフォルトの TCP プロファイルで設定できるわけではありません。一部の設定は、グローバル TCP コマンドを使用して実行する必要があります (以下のセクションを参照)。
> - デフォルトプロファイルは、サービスまたは仮想サーバーに明示的にバインドする必要はありません。

デフォルトの TCP プロファイルを設定するには

- コマンド・ライン・インタフェースを使用して、コマンド・プロンプトに次のように入力します。

```
set ns tcpProfile nstcp_default_profile...
<!--NeedCopy-->
```

- GUI で、[システム] > [プロファイル] に移動し、[**TCP** プロファイル] をクリックして、nstcp_default_profile を更新します。

グローバル **TCP** コマンド

グローバル TCP パラメータを設定するために使用できるもう 1 つの方法は、global TCP コマンドです。このコマンドでは、一意のパラメータに加えて、TCP プロファイルを使用して設定できるいくつかのパラメータが複製されます。これらの重複パラメータに対する更新は、デフォルトの TCP プロファイルの対応するパラメータに反映されます。

たとえば、この方法を使用して SACK パラメータを更新すると、デフォルトの TCP プロファイル（nstcp_default_profile）の SACK パラメータにその値が反映されます。

> 注:
>
> このアプローチは、デフォルトの TCP プロファイルで使用できない TCP パラメータにのみ使用することをお勧めします。

グローバル TCP コマンドを設定するには

- コマンド・ライン・インタフェースを使用して、コマンド・プロンプトに次のように入力します。

```
set ns tcpParam …
<!--NeedCopy-->
```

- GUI で、[システム] **>** [設定] に移動し、[**TCP** パラメータの変更] をクリックし、必要な TCP パラメータを更新します。

TCP バッファリング機能

NetScaler ADC は、TCP バッファサイズを指定するために使用できる、TCP バッファリングと呼ばれる機能を提供します。この機能は、グローバルに有効にすることも、サービスレベルで有効にすることもできます。

> 注:
>
> バッファサイズは、デフォルトの TCP プロファイルで設定することもできます。TCP バッファリング機能とデフォルトの TCP プロファイルでバッファサイズの値が異なる場合は、大きい値が適用されます。

TCP バッファリング機能をグローバルに構成する

- コマンドプロンプトで、次のように入力します。

  enable ns mode TCPB

  `set ns tcpbufParam -size <positiveInteger> -memLimit <positiveInteger>`

- GUI で、[システム] > [設定] に移動し、[モードの設定] をクリックし、[**TCP** バッファリング] を選択します。

  そして、[システム] > [設定] に移動し、[**TCP** パラメータの変更] をクリックし、[バッファサイズ] と [メモリ使用制限] の値を指定します。

サービスまたは仮想サーバ固有の **TCP** パラメータの設定

TCP プロファイルを使用して、サービスおよび仮想サーバの TCP パラメータを指定できます。TCP プロファイルを定義し（または組み込みの TCP プロファイルを使用して）、プロファイルを適切なサービスおよび仮想サーバに関連付ける必要があります。

> 注:
>
> 要件に従って、デフォルトプロファイルの TCP パラメータを変更することもできます。

TCP バッファリング機能で指定されたパラメータを使用して、サービスレベルで TCP バッファサイズを指定できます。

コマンドラインインターフェイスを使用してサービスレベルまたは仮想サーバーレベルの TCP 構成を指定するには

コマンドプロンプトで、次の操作を実行します。

1. TCP プロファイルを設定します。

```
set ns tcpProfile <profile-name>...
<!--NeedCopy-->
```

2. TCP プロファイルをサービスまたは仮想サーバーにバインドします。

```
    set service <name> ....
    <!--NeedCopy-->
```

例**:**

```
> set service service1 -tcpProfileName profile1
```

TCP プロファイルを仮想サーバにバインドするには、次の手順を実行します。

```
set lb vserver <name> ....
<!--NeedCopy-->
```

例:

```
> set lb vserver lbvserver1 -tcpProfileName profile1
<!--NeedCopy-->
```

GUI を使用してサービスレベルまたは仮想サーバレベルの TCP 構成を指定するには

GUI で、次の手順を実行します。

1. TCP プロファイルを設定します。

   [システム] > [プロファイル] > [**TCP** プロファイル] に移動し、TCP プロファイルを作成します。

2. TCP プロファイルをサービスまたは仮想サーバーにバインドします。

   [トラフィック管理] > [負荷分散] > [サービス**/**仮想サーバー] に移動し、サービスまたは仮想サーバーにバインドする TCP プロファイルを作成します。

組み込み **TCP** プロファイル

構成の便宜上、NetScaler ADC には TCP プロファイルがいくつか組み込まれています。次の組み込みプロファイルを確認し、プロファイルを選択してそのまま使用するか、要件に合わせて変更します。これらのプロファイルを、必要なサービスまたは仮想サーバーにバインドできます。

組み込みのプロファイル	説明
nstcp_default_profile	アプライアンスのデフォルトのグローバル TCP 設定を表します。
nstcp_default_tcp_lan	バックエンドサーバー接続で、これらのサーバーがアプライアンスと同じ LAN 上に存在する場合に役立ちます。
nstcp_default_WAN	WAN 展開に便利です。
nstcp_default_tcp_lan_thin_stream	nstcp_default_tcp_lan プロファイルに似ています。ただし、設定は小さなサイズのパケットフローに調整されます。
nstcp_default_tcp_interactive_stream	nstcp_default_tcp_lan プロファイルに似ています。ただし、遅延 ACK タイマーと ACK ON **PUSH** パケットの設定が減少します。
nstcp_default_tcp_lfp	クライアント側の長太パイプネットワーク (WAN) に便利です。長いファットパイプネットワークには、遅延が長く、パケットドロップが最小限に抑えられ、帯域幅の回線が長くなります。
nstcp_default_tcp_lfp_thin_stream	nstcp_default_tcp_lfp プロファイルに似ています。ただし、この設定は小さいサイズのパケットフローに合わせて調整されます。
nstcp_default_tcp_lnp	クライアント側の細長いパイプネットワーク (WAN) に便利です。細長いパイプネットワークでは、かなりのパケット損失が発生することがあります。
nstcp_default_tcp_lnp_thin_stream	nstcp_default_tcp_lnp プロファイルに似ています。ただし、この設定は小さいサイズのパケットフローに合わせて調整されます。
nstcp_internal_apps	アプライアンス上の内部アプリケーション（GSLB サイトの同期など）に便利です。これには、目的のアプリケーションに合わせて調整されたウィンドウスケーリングと SACK オプションが含まれています。このプロファイルは、内部アプリケーション以外のアプリケーションにバインドしないでください。
nstcp_default_mobile_profile	モバイルデバイスに便利です。
nstcp_default_XA_XD_profile	Citrix Virtual Apps and Desktops の展開に役立ちます。

TCP 設定の例

次の設定に使用するコマンドラインインターフェイスの例の例。

なりすまし攻撃に対する **TCP** の防御

NetScaler ADC がなりすまし攻撃から TCP を防御できるようにします。デフォルトでは、「rstWindowWattenuation」パラメータは無効になっています。このパラメータは、スプーフィングからアプライアンスを保護するために有効になっています。有効にすると、無効なシーケンス番号に対する修正確認応答（ACK）で応答します。有効な値は、[有効]、[無効] です。

RST ウィンドウ減衰パラメータは、スプーフィングからアプライアンスを保護する場所です。有効にすると、シーケンス番号が無効な場合に是正 ACK で返信します。

```
    > set ns tcpProfile profile1 -rstWindowAttenuate ENABLED -
       spoofSynDrop ENABLED
    Done
    > set lb vserver lbvserver1 -tcpProfileName profile1
    Done
<!--NeedCopy-->
```

明示的な輻輳通知（**ECN**）

Enable ECN on the required TCP profile

```
    > set ns tcpProfile profile1 -ECN ENABLED
    Done
    > set lb vserver lbvserver1 -tcpProfileName profile1
    Done
<!--NeedCopy-->
```

選択的謝辞 **(SACK)**

必要な TCP プロファイルで SACK を有効にします。

```
    > set ns tcpProfile profile1 -SACK ENABLED
    Done
    > set lb vserver lbvserver1 -tcpProfileName profile1
    Done
<!--NeedCopy-->
```

転送確認応答 **(FACK)**

必要な TCP プロファイルで FACK を有効にします。

```
> set ns tcpProfile profile1 -FACK ENABLED
> set lb vserver lbvserver1 -tcpProfileName profile1
<!--NeedCopy-->
```


ウィンドウスケーリング **(WS)**

ウィンドウスケーリングを有効にし、必要な TCP プロファイルでウィンドウのスケーリング係数を設定します。

```
set ns tcpProfile profile1  -WS ENABLED  -WSVal 9
Done
set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy-->
```

最大セグメントサイズ **(MSS)**

MSS 関連の設定を更新します。

```
> set ns tcpProfile profile1  -mss 1460 - maxPktPerMss 512
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy-->
```

NetScaler で仮想サーバーの **MSS** を学習する

NetScaler ADC が VSS を学習し、その他の関連構成を更新できるようにします。

```
> set ns tcpParam -learnVsvrMSS ENABLED  -mssLearnInterval 180 -
    mssLearnDelay 3600
Done
<!--NeedCopy-->
```

TCP キープアライブ

TCP キープアライブを有効にし、その他の関連設定を更新します。

```
> set ns tcpProfile profile1 -KA ENABLED -KaprobeUpdateLastactivity
ENABLED -KAconnIdleTime 900 -KAmaxProbes 3 -KaprobeInterval 75
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

バッファサイズ**-TCP** プロファイルの使用

バッファサイズを指定します。


```
> set ns tcpProfile profile1 -bufferSize 8190
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

バッファサイズ**-TCP** バッファリング機能を使用

TCP バッファリング機能（グローバルまたはサービス）を有効にし、バッファサイズとメモリ制限を指定します。

```
> enable ns feature TCPB
Done
> set ns tcpbufParam -size 64 -memLimit 64
Done
```

MPTCP

MPTCP を有効にし、オプションの MPTCP 設定を設定します。

```
> set ns tcpProfile profile1 -mptcp ENABLED
Done
> set ns tcpProfile profile1 -mptcpDropDataOnPreEstSF ENABLED -
mptcpFastOpen ENABLED -mptcpSessionTimeout 7200
Done
> set ns tcpparam -mptcpConCloseOnPassiveSF ENABLED -mptcpChecksum
ENABLED -mptcpSFtimeout 0 -mptcpSFReplaceTimeout 10
-mptcpMaxSF 4 -mptcpMaxPendingSF 4 -mptcpPendingJoinThreshold 0 -
mptcpRTOsToSwitchSF 2 -mptcpUseBackupOnDSS ENABLED
Done
```

輻輳制御

必要な TCP 輻輳制御アルゴリズムを設定します。

```
set ns tcpProfile profile1 -flavor Westwood
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

動的受信バッファリング

必要な TCP プロファイルでダイナミック受信バッファリングを有効にします。


```
> set ns tcpProfile profile1 -dynamicReceiveBuffering ENABLED
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

マルチパス **TCP (MPTCP)** での **TCP** ファストオープン **(TFO)** のサポート

NetScaler ADC アプライアンスは、マルチパス TCP（MPTCP）接続を確立し、データ転送を高速化するための TCP 高速オープン（TFO）メカニズムをサポートするようになりました。このメカニズムにより、SYN および SYN-ACK パケットでの初期 MPTCP 接続ハンドシェイク中にサブフローデータを伝送できます。また、MPTCP 接続の確立中に受信ノードによってデータが消費されることも可能になります。

詳細については、「TCP 高速オープン 」のトピックを参照してください。

MPTCP の可変 **TFO** クッキーサイズのサポート

NetScaler ADC アプライアンスでは、TCP プロファイルで最小サイズが 4 バイト、最大サイズが 16 バイトの可変長 TCP ファストオープン（TFO）Cookie を構成できるようになりました。これにより、アプライアンスは SYN-ACK パケットで設定された TFO cookie サイズでクライアントに応答できます。

コマンドラインインターフェイスを使用して TCP プロファイルで TCP Fast Open（TFO）Cookie を設定するには

コマンドプロンプトで入力します：

```
set tcpProfile nstcp_default_profile -tcpFastOpenCookieSize <positive_integer
>
```

例

```
set tcpProfile nstcp_default_profile -tcpFastOpenCookieSize 8
```

GUI を使用して TCP プロファイルで TCP ファストオープン（TFO）クッキーを設定するには

1. [設定] > [システム] > [プロファイル] に移動します。
2. 詳細ペインで、[**TCP** プロファイル] タブに移動し、TCP プロファイルを選択します。
3. [**TCP** プロファイルの設定] ページで、[**TCP** 高速オープン Cookie サイズ] を設定します。
4. [**OK**] をクリックし、[完了] をクリックします。

SYN-Cookie timeout interval

TCPSyncookieパラメータは、SYN 攻撃に対する堅牢な（RFC 4987）ベースの保護を提供するために、TCP プロファイルでデフォルトで有効になっています。この保護と互換性がないカスタム TCP クライアントに対応する必要があるが、攻撃の場合にフォールバックを確保したい場合、synAttackDetectionは、

autosyncookietimeoutパラメータで指定された期間、内部的にSYNCookie動作を自動的にアクティブ化することによってこれを処理します。

コマンドラインインターフェイスを使用して SYN ACK 再送信の最大しきい値を設定するには、次の手順を実行します。

コマンドプロンプトで入力します:

```
    set ns tcpparam [-maxSynAckRetx <positive_integer>]

    Set ns tcpparam [-maxSynAckRetx 150]
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して自動 SYN Cookie タイムアウト間隔を設定するには

コマンドプロンプトで入力します:

```
set ns tcpparam [-autosyncookietimeout <positive_integer>]

Set ns tcpparam [-autosyncookietimeout 90]
```

クライアントとサーバーの接続をリンク解除する

有効にすると、相手側に送信する未処理のデータがある場合、このパラメータはクライアントとサーバーの接続を切断します。デフォルトでは、パラメータは無効になっています。

```
set ns tcpparam -delinkClientServerOnRST ENABLED
Done

<!--NeedCopy-->
```

スロースタートしきい値パラメータを構成する　スロースタートしきい値slowStartthresholdパラメーターを使用して、輻輳制御アルゴリズムのNileバリアントのtcp-slowstartthreshold値を構成できます。パラメータに使用できる値は、min = 8190とmax = 524288です。デフォルトの値は524288です。TCP プロファイルの下の TCP バリアントNileは、maxcwndパラメータに依存しなくなりました。NileバリアントのslowStartthresholdパラメータを設定する必要があります。

コマンドプロンプトで次のように入力します:

```
set tcpprofile nstcp_default_profile -slowstartthreshold 8190
Done

<!--NeedCopy-->
```


HTTP 構成

October 25, 2023

> 重要:
>
> NetScaler リリース 13.0 ビルド 71.x 以降、NetScaler アプライアンスは L7 アプリケーションリクエストに対応するために大きなヘッダーサイズの HTTP リクエストを処理できるようになりました。ヘッダーサイズは最大 120 KB まで設定できます。

NetScaler ADC アプライアンスの HTTP 構成は、HTTP 設定の集合である HTTP プロファイルと呼ばれるエンティティで指定できます。HTTP プロファイルは、これらの HTTP 設定を使用するサービスまたは仮想サーバに関連付けることができます。

デフォルトの HTTP プロファイルは、デフォルトですべてのサービスと仮想サーバーにグローバルに適用される HTTP 構成を設定するように構成できます。

> 注:
>
> HTTP パラメータにサービス、仮想サーバー、およびグローバルに異なる値がある場合、最も固有のエンティティ（サービス）の値が最も高い優先順位が与えられます。

NetScaler アプライアンスには、HTTP を構成するための他の方法も用意されています。詳細については読んでください。

NetScaler は WebSocket プロトコルをサポートしているため、ブラウザやその他のクライアントはサーバーへの双方向の全二重 TCP 接続を作成できます。WebSocket の NetScaler 実装は RFC 6455 に準拠しています。

> 注:
>
> NetScaler アプライアンスは、HTTP/1.1 プロトコルと HTTP/2 プロトコルの両方でユーザーソース IP（USIP）アドレス構成をサポートしています。

グローバル HTTP パラメータの設定

NetScaler アプライアンスでは、すべての NetScaler サービスと仮想サーバーに適用される HTTP パラメーターの値を指定できます。これは、以下を使用して実行できます。

- デフォルトの HTTP プロファイル
- グローバル HTTP コマンド

デフォルトの HTTP プロファイル

nshttp_default_profile という名前の HTTP プロファイルを使用して、サービスレベルまたは仮想サーバーレベルで HTTP 構成が提供されていない場合に使用される HTTP 構成を指定します。

> 注:
>
> - すべての HTTP パラメータをデフォルトの HTTP プロファイルで設定できるわけではありません。一部の設定は、グローバル HTTP コマンドを使用して実行します (次のセクションを参照)。
> - デフォルトプロファイルは、サービスまたは仮想サーバーに明示的にバインドする必要はありません。

デフォルトの HTTP プロファイルを設定するには

- コマンド・ライン・インタフェースを使用して、コマンド・プロンプトに次のように入力します。

  ```
  set ns httpProfile nshttp_default_profile …
  ```

- GUI で、[システム] > [プロファイル] に移動し、[**HTTP** プロファイル] をクリックし、nshttp_default_profile を更新します。

グローバル **HTTP** コマンド

グローバル HTTP パラメータを設定するために使用できるもう 1 つの方法は、global HTTP コマンドです。このコマンドは一意のパラメータに加えて、HTTP プロファイルを使用して設定できるいくつかのパラメータを複製します。これらの重複パラメータに対して行われた更新は、デフォルトの HTTP プロファイルの対応するパラメータに反映されます。

たとえば、このアプローチを使用して maxReusePool パラメータを更新すると、値はデフォルトの HTTP プロファイル (nshttp_default_profile) の maxReusePool パラメータに反映されます。

> 注:
>
> この方法は、デフォルトの HTTP プロファイルにはない HTTP パラメータにのみ使用することをお勧めします。

グローバル HTTP コマンドを設定するには

- コマンド・ライン・インタフェースを使用して、コマンド・プロンプトに次のように入力します。

  ```
  set ns httpParam …
  ```

- GUI で、[システム**] > [**設定] に移動し、[**HTTP** パラメータの変更] をクリックし、必要な HTTP パラメータを更新します。

接続要求に対して無視コーディングスキームを設定するには

HTTP/2 を有効にし、接続リクエストのコーディングスキームを無視するように HTTP/2 パラメータを設定するには、コマンドプロンプトで次のように入力します。

```
set ns httpParam [-ignoreConnectCodingScheme ( ENABLED | DISABLED )]
```


例:

```
set ns httpParam -ignoreConnectCodingScheme ENABLED
```

NetScaler ADC コマンドラインを使用して HTTP プロファイルを仮想サーバーにバインドするには

HTTP プロファイルを設定して TRACE または TRACK 無効な要求をドロップする

marktraceReqInval パラメーターを有効にして、TRACK リクエストと TRACK リクエストを無効としてマークできます。このオプションを仮想 IP アドレスで DropInvalidReqs オプションとともに有効にすると、NetScaler アプライアンスに TRACE または TRACK リクエストを送信するクライアントをリセットできます。

CLI を使用して HTTP プロファイルを設定するには

コマンドプロンプトで入力します。

```
set ns httpProfile <profile name> [-markTraceReqInval ENABLED |
DISABLED ]
```

例:

```
set ns httpProfile profile1 -markTraceReqInval ENABLED
```

サービスグループの HTTP プロファイルの設定

コマンドプロンプトで入力します。

```
1 add serviceGroup <serviceGroupName>@ <serviceType> [-cacheType <
     cacheType>] [-td <positive_integer>] [-maxClient <positive_integer>]
      [-maxReq <positive_integer>] [-cacheable ( YES | NO )] [-cip (
     ENABLED | DISABLED ) [<cipHeader>]] [-usip ( YES | NO )] [-
     pathMonitor ( YES | NO )] [-pathMonitorIndv ( YES | NO )] [-
     useproxyport ( YES | NO )] [-healthMonitor ( YES | NO )] [-sp ( ON |
      OFF )] [-rtspSessionidRemap ( ON | OFF )] [-cltTimeout <secs>] [-
     svrTimeout <secs>] [-CKA ( YES | NO )] [-TCPB ( YES | NO )] [-CMP (
     YES | NO )] [-maxBandwidth
2 <positive_integer>] [-monThreshold <positive_integer>] [-state ENABLED
     DISABLED )][-downStateFlush ( ENABLED | DISABLED )] [-tcpProfileName
      <string>] [-httpProfileName <string>] [-comment <string>] [-
     appflowLog ( ENABLED | DISABLED )] [-netProfile <string>] [-
     autoScale <autoScale> -memberPort <port> [-autoDisablegraceful ( YES
      | NO )] [-autoDisabledelay <secs>] ] [-monConnectionClose ( RESET |
      FIN )]
3
4 <!--NeedCopy-->
```

例:


```
add serviceGroup Service-Group-1 HTTP -maxClient 0 -maxReq 0 -cip
ENABLED -usip NO -useproxyport YES -cltTimeout 200 -svrTimeout 300 -
CKA NO -TCPB NO -CMP NO -httpProfileName profile1
```

NetScaler GUI を使用して **HTTP** プロファイルを構成する

TRACE または TRACK の無効なリクエストをマークするには、次の手順を実行します。

1. NetScaler ADC アプライアンスにサインインし、［構成］>［システム］>［プロファイル］に移動します。
2. [**HTTP** プロファイル] タブページで、[追加] をクリックします。
3. [**HTTP** プロファイルの作成] ページで、[トレース要求を無効としてマーク] オプションを選択します。
4. ［作成］をクリックします。

サービスまたは仮想サーバー固有の **HTTP** パラメータの設定

HTTP プロファイルを使用して、サービスおよび仮想サーバの HTTP パラメータを指定できます。HTTP プロファイルを定義し（または組み込みの HTTP プロファイルを使用して）、プロファイルを適切なサービスおよび仮想サーバに関連付ける必要があります。

> 注:
>
> 要件に従って、デフォルトプロファイルの HTTP パラメータを変更することもできます。

コマンドラインインターフェイスを使用してサービスレベルまたは仮想サーバーレベルの **HTTP** 構成を指定するには

コマンドプロンプトで、次の操作を実行します。

1. HTTP プロファイルを設定します。

   ```
   set ns httpProfile <profile-name>...
   ```

2. HTTP プロファイルをサービスまたは仮想サーバーにバインドします。

   HTTP プロファイルをサービスにバインドするには、次の手順を実行します。

```
set service <name> .....
```

例:

```
1 > set service service1 -httpProfileName profile1
2 <!--NeedCopy-->
```

HTTP プロファイルを仮想サーバーにバインドするには、次の手順を実行します。

```
set lb vserver <name> .....
```


例:

```
> set lb vserver lbvserver1 -httpProfileName profile1
<!--NeedCopy-->
```

GUI を使用してサービスレベルまたは仮想サーバレベルの HTTP 構成を指定するには

GUI で、次の手順を実行します。

1. HTTP プロファイルを設定します。

   [システム] > [プロファイル] > [**HTTP** プロファイル] に移動し、HTTP プロファイルを作成します。

2. HTTP プロファイルをサービスまたは仮想サーバーにバインドします。

   [トラフィック管理] > [負荷分散] > [サービス/仮想サーバー] に移動し、サービス/仮想サーバーにバインドする必要がある HTTP プロファイルを作成します。

組み込みの HTTP プロファイル

NetScaler には、構成の便宜上、いくつかの組み込み HTTP プロファイルが用意されています。リストされているプロファイルを確認し、そのまま使用するか、要件に合わせて変更します。これらのプロファイルは、必要なサービスまたは仮想サーバーにバインドできます。

組み込みのプロファイル	説明
nshttp_default_profile	アプライアンスのデフォルトのグローバル HTTP 設定を表します。
nshttp_default_strict_validation	HTTP リクエストとレスポンスの厳密な検証が必要なデプロイ用の設定。

HTTP 設定の例

次の設定に使用するコマンドラインインターフェイスのサンプル例。

- HTTP バンド統計情報
- WebSocket 接続

HTTP バンド統計情報

HTTP リクエストとレスポンスのバンドサイズを指定します。


```
> set protocol httpBand reqBandSize 300 respBandSize 2048
Done
> show protocol httpband -type REQUEST
<!--NeedCopy-->
```

WebSocket 接続

必要な HTTP プロファイルで WebSocket を有効にします。

```
> set ns httpProfile http_profile1 -webSocket ENABLED
Done
> set lb vserver lbvserver1 -httpProfileName profile1
Done
<!--NeedCopy-->
```

アップグレードヘッダーを削除するか、バックエンドサーバーに渡すように **NetScaler** アプライアンスを構成します

HTTP プロファイルの PassProtocol Upgrade パラメーターは、バックエンドサーバーへの攻撃を防ぎます。このパラメータの状態に応じて、アップグレードヘッダーはバックエンドサーバーに送信されるリクエストで渡されるか、リクエストを送信する前に削除されます。

- PassProtocol Upgrade パラメーターが有効になっている場合は、アップグレードヘッダーがバックエンドサーバーに渡されます。サーバーはアップグレード要求を受け入れ、応答で通知します。
- パラメータが無効になっている場合、アップグレードヘッダーは削除され、残りのリクエストはバックエンドサーバーに送信されます。

PassProtocolUpgrade パラメータが次のプロファイルに追加されます。

- nshttp_default_profile-デフォルトで有効になっています
- nshttp_default_strict_validation-デフォルトでは無効になっています
- nshttp_default_internal_apps-デフォルトでは無効になっています
- nshttp_default_http_quic_profile-デフォルトで有効になっています

PassProtocol Upgrade パラメーターをデフォルトで無効に設定することをお勧めします。

CLI を使用して **PassProtocol** アップグレードパラメータを設定します

コマンドプロンプトで、次のように入力します。

```
set ns httpProfile <name> [-passProtocolUpgrade ( ENABLED | DISABLED
)]
```


例**:**

```
set ns httpProfile profile1 -passProtocolUpgrade ENABLED
```

GUI を使用して **PassProtocol** アップグレードパラメータを設定します

1. [システム**] > [**プロファイル**] > [HTTP** プロファイル] に移動します。
2. HTTP プロファイルを作成または編集します。
3.「プロトコルアップグレードをパスする」を選択します。

HTTP/2 構成

February 15, 2024

> 注:
> HTTP/2 機能は、NetScaler MPX、VPX、および SDX モデルでサポートされています。NetScaler VPX アプライアンスでは、NetScaler ADC バージョン 11.0 以降から HTTP/2 機能がサポートされています。

Web アプリケーションのパフォーマンスに関する問題は、ページサイズと Web ページ上のオブジェクト数の増加傾向に直接関係しています。HTTP/1.1 は、現在一般的なものよりも小さな Web ページ、低速のインターネット接続、および制限されたサーバーハードウェアをサポートするために開発されました。JavaScript やカスケードスタイルシート (CSS) などの新しいテクノロジーや、Flash ビデオやグラフィックが豊富な画像などの新しいメディアタイプには適していません。これは、サーバーへの接続ごとに 1 つのリソースしか要求できないためです。この制限により、ラウンドトリップの数が大幅に増加し、ページレンダリングが長くなり、ネットワークパフォーマンスが低下します。

HTTP/2 プロトコルは、ネットワーク上で送信されるデータを減らして通信を可能にし、1 つの接続で複数の要求と応答を送信する機能を提供することで、これらの制限に対処します。HTTP/2 では、基盤となるネットワーク接続をより効率的に使用することで、HTTP/1.1 の主な制限に対処しています。これは、要求と応答がネットワーク上で移動する方法を変更します。

HTTP/2 はバイナリプロトコルです。HTTP/1.1 のようなテキストプロトコルと比較して、解析がより効率的で、ネットワーク上でよりコンパクトになり、最も重要なこととして、エラーが発生しにくくなります。HTTP/2 プロトコルは、フレームタイプと、クライアントとサーバ間で HTTP メッセージのカプセル化および転送方法を定義するバイナリフレーミング層を使用します。HTTP/2 機能は、CONNECT メソッドを使用して、単一の HTTP/2 ストリームを介してリモートホストへのトンネル接続を確立します。

HTTP/2 プロトコルには、特にモバイルネットワーク経由で接続するクライアントに対して、パフォーマンスを大幅に向上させるパフォーマンス向上の変更が多数含まれています。

次の表に、HTTP/1.1 から HTTP/2 の主な改善点を示します。

HTTP/2 の機能	説明
ヘッダー圧縮	HTTP ヘッダーには多くの繰り返し情報があるため、データ転送中に不要な帯域幅を消費します。HTTP/2 ヘッダーを圧縮し、要求と応答ごとに HTTP ヘッダーを転送する要件を最小限に抑えることで、帯域幅の要件を削減します。
接続多重化	レイテンシーは、ページの読み込み時間とエンドユーザーエクスペリエンスに大きな影響を与える可能性があります。接続の多重化は、単一の接続で複数の要求と応答を送信することにより、この問題を克服します。
サーバープッシュ	サーバープッシュにより、サーバーはクライアントブラウザーにコンテンツをプロアクティブにプッシュできるため、ラウンドトリップの遅延を回避できます。この機能は、クライアントが必要と考える応答をキャッシュし、ラウンドトリップ回数を減らし、ページのレンダリング時間を改善します。重要：NetScaler ADC アプライアンスは、サーバープッシュ機能をサポートしていません。
ヘッドオブラインブロッキングなし	HTTP 1.1 では、ブラウザは接続ごとに一度に 1 つのリソースをダウンロードできます。ブラウザが大きなリソースをダウンロードする必要がある場合、最初のダウンロードが完了するまで、他のすべてのリソースのダウンロードをブロックします。HTTP/2 は、多重化アプローチでこの問題を克服します。これにより、クライアントブラウザーは、同じ接続を介して他の Web コンポーネントを並行してダウンロードし、利用可能になったときにそれらを表示できます。

HTTP/2 の機能	説明
リクエストの優先順位付け	ブラウザーが Web ページをレンダリングするとき、すべてのリソースが同等の優先度を持つわけではありません。ロード時間を短縮するために、最新のブラウザーはすべて、アセットのタイプ、ページ上の位置、および以前の訪問から学習した優先度によってリクエストを優先します。HTTP/1.1 では、このプロトコルは多重化をサポートしておらず、サーバーによる要求の優先順位付けを通信する方法がないため、ブラウザは優先度データを使用する能力が制限される。その結果、不要なネットワーク遅延が生じます。HTTP/2 は、ブラウザがすべてのリクエストをディスパッチできるようにすることで、この問題を克服します。ブラウザーは、ストリームの依存関係と重みによってストリームの優先順位付けの優先度を伝えることができ、サーバーが応答配信を最適化できるようにします。重要：NetScaler ADC アプライアンスは、要求の優先順位付け機能をサポートしていません。

HTTP/2 のしくみ

NetScaler ADC アプライアンスはクライアント側でもサーバー側でも HTTP/2 をサポートします。クライアント側では、NetScaler ADC アプライアンスは HTTP/2 の HTTP/HTTPS 仮想サーバー をホストするサーバーとして動作します。バックエンド側では、NetScaler ADC は仮想サーバーにバインドされているサーバーのクライアントとして機能します。

したがって、NetScaler ADC アプライアンスは、クライアント側とサーバー側で別々の接続を維持します。NetScaler ADC アプライアンスには、クライアント側とサーバー側に別々の HTTP/2 構成があります。

HTTPS (SSL) 負荷分散構成の場合は **HTTP/2**

HTTPS 負荷分散構成の場合、NetScaler ADC アプライアンスは TLS ALPN 拡張（RFC 7301）を使用して、クライアント/サーバーが HTTP/2 をサポートするかどうかを判断します。その場合、アプライアンスは、クライアント/サーバー側でデータを送信するためのアプリケーション層プロトコルとして HTTP/2 を選択します（RFC 7540-セクション 3.3 を参照）。
アプライアンスは、TLS ALPN 拡張を使用してアプリケーション層プロトコルを選択するときに、次の優先順位を使用します。

- HTTP/2（HTTP プロファイルで有効になっている場合）
- HTTP/1.1

HTTP 負荷分散設定用の HTTP/2

HTTP 負荷分散構成の場合、NetScaler ADC アプライアンスは次のいずれかの方法を使用して、HTTP/2 を使用してクライアント/サーバーとの通信を開始します。

> 注
>
> 次のメソッドの説明では、クライアントとサーバーは HTTP/2 接続の一般的な用語です。たとえば、を使用した NetScaler ADC アプライアンスの負荷分散セットアップの場合 HTTP/2, NetScaler ADC アプライアンスは、クライアント側でサーバーとして機能し、サーバー側でクライアントとして機能します。

- **HTTP/2** アップグレード。クライアントは HTTP/1.1 リクエストをサーバーに送信します。リクエストにはアップグレードヘッダーが含まれており、HTTP/2 への接続をサーバーにアップグレードするように要求します。サーバーが HTTP/2 をサポートしている場合、サーバーはアップグレード要求を受け入れ、応答で通知します。クライアントとサーバは、クライアントがアップグレード確認応答を受信した後、HTTP/2 を使用して通信を開始します。
- 直接 **HTTP/2**。クライアントは、HTTP/2 アップグレード方式を使用する代わりに、HTTP/2 でサーバーとの通信を直接開始します。サーバーが HTTP/2 をサポートしていない場合、または HTTP/2 要求を直接受け付けるように構成されていない場合、クライアントからの HTTP/2 パケットはドロップされます。この方法は、クライアントデバイスの管理者が、サーバーが HTTP/2 をサポートしていることをすでに知っている場合に役立ちます。
- 代替サービス **(ALT-SVC)** を使用して直接 **HTTP/2**。サーバーは、HTTP/1.1 応答に代替サービス (ALT-SVC) フィールドを含めることで、HTTP/2 をサポートしていることをクライアントにアドバタイズします。クライアントが ALT-SVC フィールドを認識するように設定されている場合、クライアントとサーバは、クライアントが応答を受信した後、HTTP/2 を使用して直接通信を開始します。

NetScaler ADC アプライアンスは、HTTP/2 メソッドの HTTP プロファイルで構成可能なオプションを提供します。これら HTTP/2 オプションは、HTTPS または HTTP 負荷分散セットアップのサーバー側だけでなくクライアント側にも適用できます。HTTP/2 メソッドとオプションの詳細については、HTTP/2 オプション PDF を参照してください。

はじめに

NetScaler ADC アプライアンスで HTTP/2 の構成を開始する前に、次の点に注意してください。

- NetScaler ADC アプライアンスは、クライアント側とサーバー側で HTTP/2 をサポートします。
- NetScaler ADC アプライアンスは、HTTP/2 サーバープッシュ機能をサポートしていません。
- NetScaler ADC アプライアンスは、HTTP/2 リクエストの優先順位付け機能をサポートしていません。
- NetScaler ADC アプライアンスは、HTTPS 負荷分散セットアップの HTTP/2 SSL 再ネゴシエーションをサポートしていません。
- NetScaler ADC アプライアンスは HTTP/2 NTLM 認証をサポートしていません。

- HTTP/2 が有効で、接続多重化が無効で（USIP が有効の場合など）、クライアントとサーバの TCP 接続を 1 対 1 でマッピングすると、FIN、リセット（RST）などのクローズイベントがクライアントまたはサーバ接続からリンクされたピア接続に転送されます。

HTTP/2 を構成する

負荷分散設定（HTTPS または HTTP）の HTTP/2 の設定は、次のタスクで構成されます。

- **HTTP/2** を有効にし、**HTTP** プロファイルでオプションの **HTTP/2** パラメータを設定します。HTTP プロファイルで HTTP/2 を有効にします。HTTP プロファイルで HTTP/2 のみを有効にすると、NetScaler ADC アプライアンスは HTTP/2 での通信にアップグレード方法（HTTP の場合）または TLS ALPN 方式（HTTPS）のみを使用します。

  NetScaler ADC アプライアンスで直接 HTTP/2 方式を使用するには、**HTTP** プロファイルでダイレクト **HTTP/2** オプションを有効にする必要があります。NetScaler ADC アプライアンスが代替サービス方法を使用して直接 HTTP/2 を使用するには、**HTTP** プロファイルで代替サービス（**altsvc**）オプションを有効にする必要があります。

- **HTTP** プロファイルを仮想サーバーまたはサービスにバインドします。HTTP プロファイルを仮想サーバーにバインドして、負荷分散セットアップのクライアント側の HTTP/2 を構成します。HTTP プロファイルをサービスにバインドして、負荷分散設定のサーバー側の HTTP2 を設定します。

> 注
>
> クライアント側とサーバー側で別々の HTTP プロファイルをバインド Citrix。

- **HTTP/2** サーバー側のサポートのグローバルパラメータを有効にする。**HTTP/2** サービス側（**http2ServerSide**）グローバル HTTP パラメータを有効にして、HTTP/2 が構成されているすべての負荷分散セットアップのサーバー側で HTTP/2 サポートを有効にします。

  HTTP/2 サービス側が無効な場合、関連する負荷分散サービスにバインドされた **HTTP** プロファイルで **HTTP/2** が有効になっている場合でも、HTTP/2 は負荷分散設定のサーバー側では機能しません。

NetScaler ADC コマンドライン手順:

NetScaler ADC コマンドラインを使用して HTTP/2 を有効にして、HTTP/2 パラメーターを設定するには

- HTTP プロファイルの追加中に HTTP/2 を有効にして HTTP/2 パラメータを設定するには、コマンドプロンプトで次のように入力します。

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct
 ( ENABLED | DISABLED )] [-altsvc ( ENABLED | DISABLED )]
show ns httpProfile <name>
```

- HTTP プロファイルの変更中に HTTP/2 を有効にして HTTP/2 パラメータを設定するには、コマンドプロンプトで次のように入力します。


```
set ns httpProfile <name> -http2 ( ENABLED | DISABLED )[-http2Direct
( ENABLED | DISABLED)] [-altsvc (ENABLED | DISABLED )]
show ns httpProfile <name>
```

NetScaler ADC コマンドラインを使用して HTTP プロファイルを仮想サーバーにバインドするには

コマンドプロンプトで入力します:

```
set lb vserver <name> - httpProfileName <string>
show lb vserver <name>
```

NetScaler ADC コマンドラインを使用して HTTP プロファイルを負荷分散サービスにバインドするには

コマンドプロンプトで入力します:

```
set service <name> -httpProfileName <string>
show service <name>
```

NetScaler ADC コマンドラインを使用してサーバー側で HTTP/2 サポートをグローバルに有効にするには

コマンドプロンプトで入力します:

```
set ns httpParam -HTTP2Serverside( ENABLED | DISABLED )
show ns httpParam
```

NetScaler GUI を使用して HTTP/2 を有効にし、HTTP/2 パラメーターを設定するには

1. [システム] > [プロファイル] に移動し、[**HTTP** プロファイル] タブをクリックします。
2. HTTP プロファイルを追加するとき、または既存の HTTP プロファイルを変更するときに、**HTTP/2** を有効にします。

NetScaler GUI を使用して HTTP プロファイルを仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[**+ HTTP** プロファイル] をクリックして、作成した HTTP プロファイルを仮想サーバーにバインドします。

NetScaler GUI を使用して HTTP プロファイルを負荷分散サービスにバインドするには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを開きます。
2. [詳細設定] で、[**+ HTTP** プロファイル] をクリックして、作成した HTTP プロファイルをサービスにバインドします。

GUI を使用してサーバー側で HTTP/2 サポートをグローバルに有効にするには

[システム] > [設定] に移動し、[**HTTP** パラメーターの変更] をクリックし、[**HTTP/2** サーバー側] を有効にします。

設定例

次の設定例では、HTTP プロファイル HTTP-PROFILE-HTTP2-クライアント側で HTTP/2 およびダイレクト HTTP/2 が有効になっています。プロファイルは仮想サーバー LB-VS-1 にバインドされています。

```
set ns httpProfile HTTP-PROFILE-HTTP2-CLIENT-SIDE -http2 enabled -
    http2Direct enabled
Done

set lb vserver LB-VS-1 -httpProfileName HTTP-PROFILE-HTTP2-CLIENT-SIDE

Done
<!--NeedCopy-->
```

次の設定例では、HTTP プロファイル HTTP-PROFILE-HTTP2-サーバー側で HTTP/2 および代替サービス（ALT-SVC）が有効になっています。プロファイルはサービス LB-SERVICE-1 にバインドされます。

```
set ns httpparam -HTTP2Serverside ENABLED
Done

set ns httpProfile HTTP-PROFILE-HTTP2-SERVER-SIDE -http2 ENABLED -
    altsvc ENABLED
Done

set service LB-SERVICE-1 -httpProfileName HTTP-PROFILE-HTTP2-SERVER-
    SIDE
Done
<!--NeedCopy-->
```

HTTP/2 の初期接続ウィンドウサイズを構成する

RFC 7540 に従って、HTTP2 ストリームおよび接続のフロー制御ウィンドウを 64 K（65535）オクテットに設定する必要があり、この値に加えられた変更はピアに伝達する必要があります。ADC アプライアンスは、フロー制御ウィンドウサイズの変化を次のように伝えます。

- ストリームにSETTINGSフレームを使用する。
- 接続にWINDOW_UPDATEフレームを使用する。

HTTP プロファイルでは､ストリームレベルで初期ウィンドウサイズを設定するようにhttp2InitialWindowSizeパラメータを設定する必要があります。内部システムエラーのため、ADC アプライアンスは接続のフロー制御ウィンドウも初期化します。ストリームに設定されたフロー制御ウィンドウに変更がある場合、ADC アプライアンスは設定フレームを使用してピアと通信します。しかし、ADC アプライアンスは、WINDOW_UPDATEフレームを使用して接続のフロー制御ウィンドウの変更を伝達しません。これにより、接続がフリーズします。

この問題を克服するために、接続のフロー制御ウィンドウを制御するhttp2InitialConnWindowSizeパラメータ（バイト単位）が追加されました。個別の設定可能なパラメータを使用することで、ストリームレベルと接

続レベルの両方で、変更されたウィンドウサイズの更新をアプライアンスが送信できるようにできるようになりました。

CLI を使用して **HTTP/2** 初期接続ウィンドウサイズパラメータを設定します

コマンドプロンプトで入力します:

```
set http profile p1 -http2InitialConnWindowSize 8290
Initial window size for stream level flow control, in bytes.
Default value: 65535
Minimum value: 8192
Maximum value: 20971520
<!--NeedCopy-->
```

> 注:
> HTTP/2 が有効な場合は、TCP プロファイルで TCP 動的受信バッファリングパラメータを無効にすることをお勧めします。

HTTP/2 経由のウェブソケット設定

NetScaler ADC アプライアンスは、HTTP/2 経由の WebSocket 接続をサポートしています。CLI または GUI インターフェイスを使用して WebSocket 接続を有効にできます。WebSocket HTTP/2 接続は多重化できます。

CLI を使用して **HTTP/2** 経由の **WebSocket** 接続を設定します

デフォルトでは、**WebSocket** 接続パラメータは無効になっています。CLI インターフェイスを使用して WebSocket 接続を有効にできます。

フロントエンド **HTTP/2 Web** ソケット接続を有効にする**:**

コマンドプロンプトで入力します:

SSL 設定の場合**:**

```
add httpprofile <http_profile_name> -http2 enabled -websocket enabled

<!--NeedCopy-->
```

プレーンテキスト設定の場合**:**

```
add httpprofile <http_profile_name> -http2 enabled -http2direct enabled
    -websocket enabled

<!--NeedCopy-->
```


バックエンド **HTTP/2** ウェブソケット接続を有効にする**:**

コマンドプロンプトで入力します:

SSL 設定の場合**:**

```
add httpprofile <http_profile_name> -http2 enabled
set httpparam -http2serverside ON
<!--NeedCopy-->
```

プレーンテキスト設定の場合**:**

```
add httpprofile <http_profile_name> -http2 enabled -http2direct enabled
set httpparam -http2serverside ON
<!--NeedCopy-->
```

GUI を使用して **HTTP/2** 経由の **WebSocket** 接続を設定します

次の手順を実行すると、GUI インターフェイスを使用して WebSocket 接続を有効にできます。

既存のプロファイルの編集**:**

1. [システム**] > [**プロファイル**] > [HTTP** プロファイル] に移動します。
2.「プロファイル」から必要なプロファイルを選択し、「編集」をクリックします。
3.「**HTTP** プロファイルの設定」で、「**HTTP2** 」または「ダイレクト **HTTP2** 」チェックボックスを有効にします。
4.「WebSocket 接続を有効にする」チェックボックスを選択して、**WebSocket** 接続を有効にします。

新しいプロファイルの追加**:**

1. [システム**] > [**プロファイル**] > [HTTP** プロファイル] に移動します。
2. 新しい HTTP2 プロファイルを追加するには、[追加] をクリックします。
3.「**HTTP** プロファイルの作成」で、「**HTTP2** 」または「ダイレクト **HTTP2** 」チェックボックスを有効にします。
4.「**WebSocket** 接続を有効にする」チェックボックスを選択します。

次の表は、バックエンドの多重化が無効な場合の WebSocket 接続の動作を示しています。

HTTP パケットバージョン	HTTP プロファイル内のウェブソケット	アクションをリクエストする	バックエンド HTTP/1.1	バックエンド HTTP/2
HTTP/1.1	無効	落下しました	-	-
HTTP/1.1	有効	HTTP/1.1	各 HTTP/1.1 接続は、バックエンドの専用の HTTP/1.1 接続にマップされます	各 HTTP/1.1 接続のバックエンドでの専用の HTTP/2 接続

HTTP パケットバージョン	HTTP プロファイル内のウェブソケット	アクションをリクエストする	バックエンド HTTP/1.1	バックエンド HTTP/2
HTTP/2	有効	HTTP/2	フロントエンドの各ストリームは専用の HTTP/1.1 接続にマップされます	すべてのフロントエンドストリームは、バックエンドの 1 つの HTTP/2 接続または最大 3 つの HTTP/2 接続にマップできます。
HTTP/2	無効	落下しました	-	-

次の表は、バックエンド多重化が有効な場合の WebSocket 接続の動作を示しています。

HTTP パケットバージョン	HTTP プロファイル内のウェブソケット	アクションをリクエストする	バックエンド HTTP/1.1	バックエンド HTTP/2
HTTP/1.1	無効	落下しました	-	-
HTTP/1.1	有効	HTTP/1.1	各 HTTP/1.1 接続は、バックエンドの専用の HTTP/1.1 接続にマップされます	複数の HTTP/1.1 クライアントを 1 つの HTTP/2 接続または複数の HTTP/2 接続に多重化できます
HTTP/2	有効	HTTP/2	フロントエンドの各ストリームは専用の HTTP/1.1 接続にマップされます	すべてのフロントエンドストリームは、バックエンド上の単一の HTTP/2 接続または複数の HTTP/2 接続にマップできます
HTTP/2	無効	落下しました	-	-

HTTP/2 DoS の軽減

August 15, 2023

Http/2 サービス拒否（DoS）攻撃は、NetScaler ADC アプライアンスに影響を与えなくなりました。アプライアンスが最大制限を超えるフレームを受信すると、アプライアンスはサイレントに接続を閉じます。

攻撃を軽減するために、HTTP プロファイルを使用すると、HTTP/2 接続で受信したフレームのデフォルト構成を変更できます。

HTTP/2 DoS 緩和の表には 、HTTP/2 DoS 攻撃とその緩和策のリストが示されています。

コマンドラインインターフェイスを使用して **DoS** 攻撃を軽減する **HTTP/2** フレームの上限を設定します

コマンドプロンプトで、次のように入力します。

```
set ns httpprofile <profile_name> - http2MaxEmptyFramesPerMin <
positive_integer> -http2MaxPingFramesPerMin <positive_integer> -
http2MaxSettingsFramesPerMin <positive_integer> -http2MaxResetFramesPerMin
 <positive_integer>
```

例**:**

```
set ns httpprofile profile1 -http2MaxEmptyFramesPerMin 20 -http2MaxPingFramesP
 20 -http2MaxSettingsFramesPerMin 20 -http2MaxResetFramesPerMin 20
```

NetScaler GUI を使用して、**HTTP/2** 接続で受信するフレームの上限を設定します

以下の手順に従って、HTTP/2 接続で受信するフレームの上限を設定します。

1. ナビゲーションペインで [システム] を展開し、[プロファイル] をクリックします。
2. 「プロファイル」ページで、「**HTTP** プロファイル」タブを選択します。
3. [**HTTP** プロファイル] タブページで、[追加] をクリックします。
4. 「**HTTP** プロファイルの設定」ページで、次のパラメータを設定します。

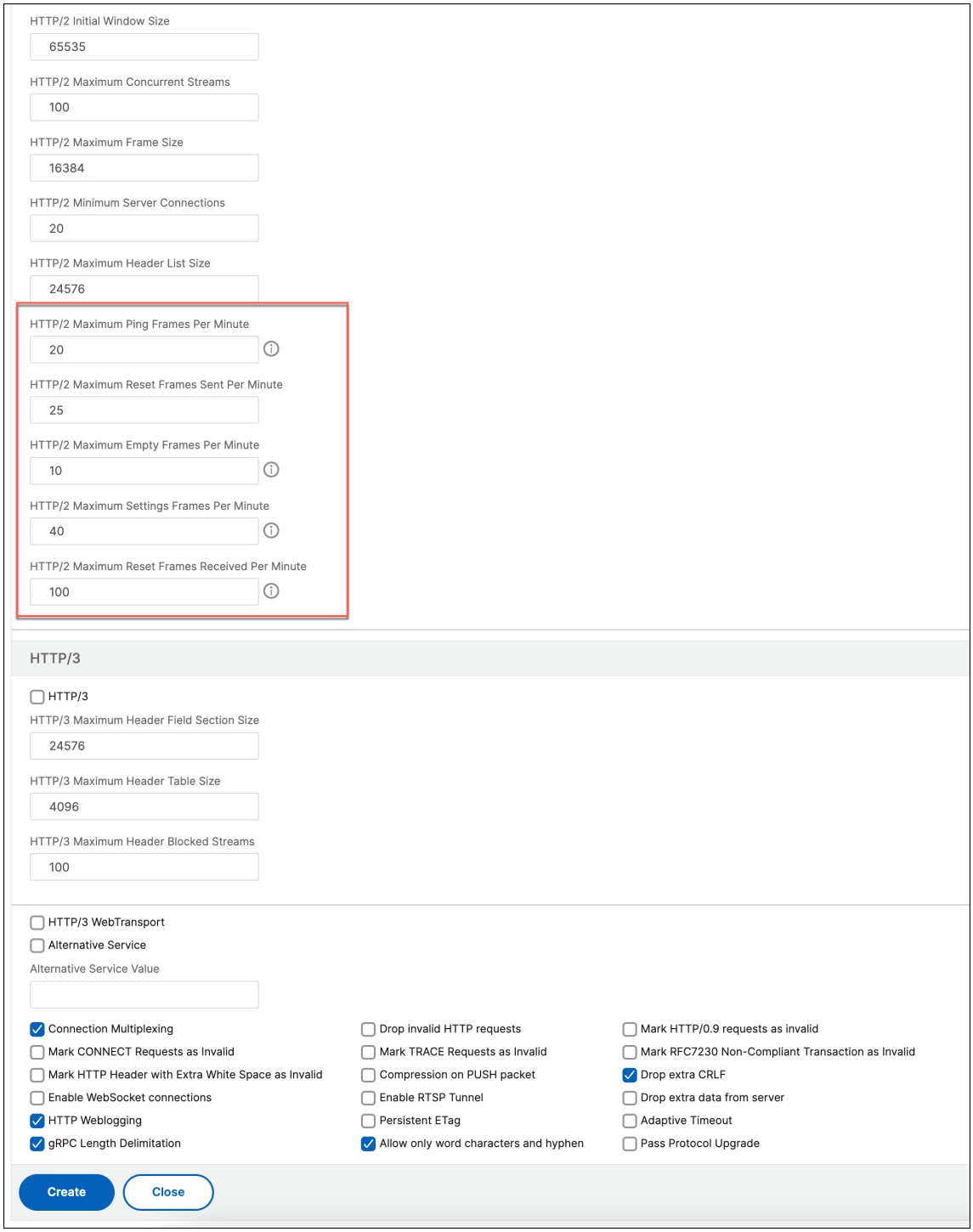
    a) http2MaxPingFramesPerMin. 1 分間に接続ごとに受信する PING フレームの最大数を設定します。PING フレームの数が設定制限を超えると、アプライアンスは接続上のパケットをサイレントにドロップします。

    b) http2MaxSettingsFramesPerMin. 1 分間に接続ごとに受信する SETTINGS フレームの最大数を設定します。SETTINGS フレームの数が設定制限を超えると、ADC は接続上のパケットをサイレントにドロップします。

    c) http2MaxResetFramesPerMin. 1 分間に接続ごとに送信される RESET フレームの最大数を設定します。RESET フレームの数が設定制限を超えると、ADC は接続上のパケットをサイレントにドロップします。

    d) http2MaxEmptyFramesPerMin. 1 回の接続で 1 分間に送信される空フレームの最大数を設定します。空のフレームの数が設定制限を超えると、ADC は接続上のパケットをサイレントにドロップします。

5.「**OK**」をクリックして「閉じる」をクリックします。



HTTP3 over QUIC プロトコル

August 15, 2023

HTTP/2 over TCP は、単一の接続で複数の HTTP リクエストストリームを送信する場合に推奨される標準です。ただし、TCP トランスポートメカニズムでは、Web サイトや Web アプリケーションへのアクセスに一定の制限とレイテンシーの問題があります。同じ接続で複数のリクエストを多重化すると、同じ接続の信頼性の影響を受けます。1 つの要求のパケットが失われた場合、他のすべての多重化要求は、失われたパケットが検出されて再送信されるまで遅延します。これにより、ヘッドオブラインブロッキングの遅延とレイテンシーの問題が発生します。

接続および転送の遅延については、HTTP/3 は TCP プロトコルの代わりに QUIC を使用します。QUIC は、TCP の代わりに UDP を基本トランスポートとして使用する新しいプロトコルです。HTTP-over-quic では、単一の TCP 接続に依存することなく、複数の独立した要求を多重化することができます。QUIC は、複数の HTTP リクエストをストリーミングできる信頼性の高い接続を実装しています。QUIC は、HTTP/1.1 や HTTP/2 のように追加のレイヤーとしてではなく、TLS を統合コンポーネントとして組み込んでいます。

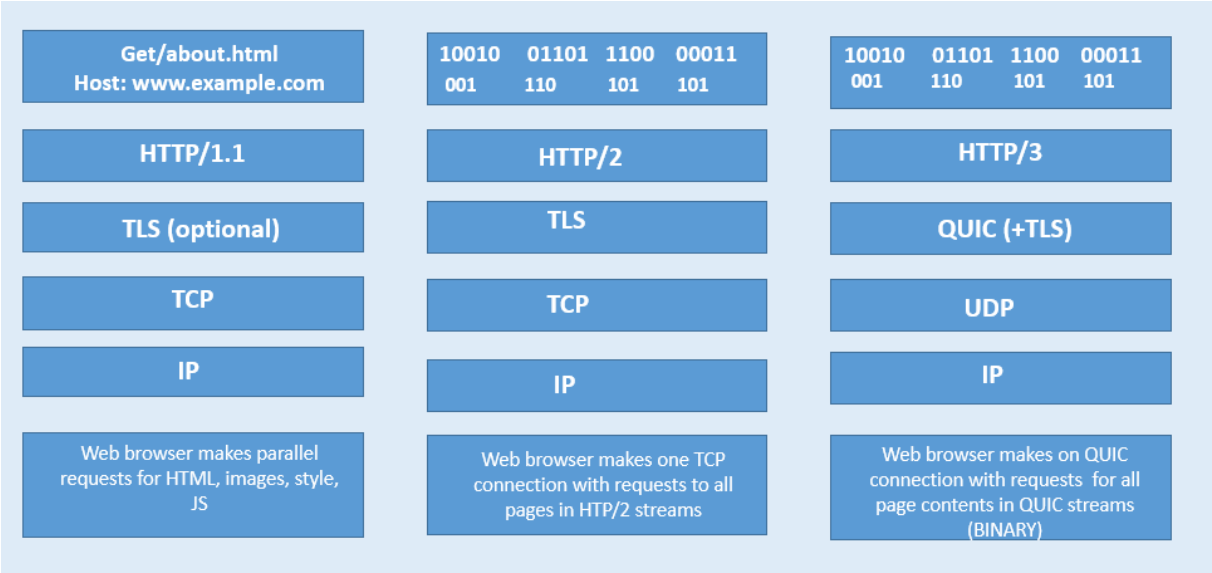
HTTP/3 プロトコルを使用する利点

HTTP/3 データ転送に QUIC プロトコルを使用する重要な利点のいくつかを以下に示します。

- ストリーム多重化
- ストリームおよび接続レベルのフロー制御
- 低レイテンシーの接続確立
- NAT 再バインドへの接続の移行と復元力
- 認証され、暗号化されたヘッダーとペイロード

HTTP プロトコルのトランスポートスタック

以下の図は、HTTP/1.1、HTTP/2、および HTTP/3 プロトコルのトランスポートスタックを示しています。



NetScaler ADC での **QUIC** および **HTTP/3** 接続管理の仕組み

次の図は、NetScaler ADC アプライアンスでの QUIC および HTTP/3 接続管理と、コンポーネントがどのように相互作用するかを示しています。

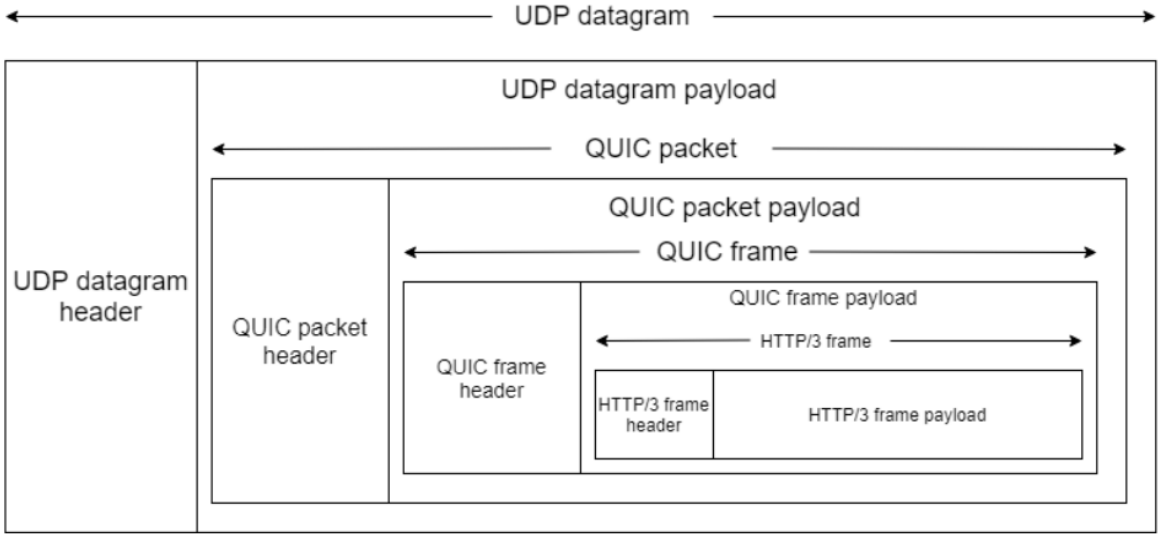
ステップ 1: NetScaler ADC アプライアンスへの QUIC プロトコル経由のクライアント側の HTTP/3 リクエスト。
ステップ 2: バックエンドサーバーのサポートに応じて、NetScaler AS HTTP/1.1 または HTTP/2 によって転送されるリクエスト。
ステップ 3: バックエンドサーバーから NetScaler ADC への HTTP/2 または HTTP/1.1 を介して応答します。
ステップ 4: ADC は HTTP/3 応答として応答をクライアントに転送します。

HTTP/3 プロトコルのしくみ

HTTP/3 では、クライアントが特定のエンドポイントに HTTP/3 サーバーが存在することを認識すると、QUIC 接続を開きます。QUIC プロトコルは、多重化およびフロー制御を提供します。各ストリーム内では、HTTP/3 通信の基

本単位はフレームです。フレームタイプごとに異なる目的を果たします。たとえば、HEADERS フレームと DATA フレームは HTTP リクエストとレスポンスの基礎を形成します。

要求の多重化は、QUIC ストリーム抽象化を使用して実行されます。各リクエストとレスポンスのペアは、単一の QUIC ストリームを消費します。ストリームは互いに独立しているため、ブロックされている、またはパケット損失が発生する 1 つのストリームは、他のストリームでの進行を妨げません。サーバープッシュは HTTP/2 で導入されたインタラクションモードで、指定された要求を行うクライアントを予測して、サーバーが要求と応答交換をクライアントにプッシュできるようにします。これにより、潜在的なレイテンシーゲインとネットワーク使用率がトレードオフされます。PUSH_PROMISE、MAX_PUSH_ID、CANCEL_PUSH など、サーバープッシュの管理にはいくつかの HTTP/3 フレームが使用されます。HTTP/2 と同様に、リクエストフィールドとレスポンスフィールドは送信用に圧縮されます。HPACK は圧縮フィールドセクションの順序通りの送信（QUIC によって提供されていない保証）に依存しているため、HTTP/3 は HPACK を QPACK に置き換えます。QPACK は、個別の単方向ストリームを使用してフィールドテーブルの状態を変更および追跡します。一方、エンコードされたフィールドセクションは、テーブルの状態を変更せずに表の状態を参照します。



HTTP/3 の設定と統計の概要

January 11, 2024

QUIC を使用して HTTP/3 データの複数のストリームを送信するための HTTP/3 プロトコルを構成するには、次の手順を実行する必要があります。

1. SSL および負荷分散機能を有効にします。
2. HTTP_QUIC タイプの負荷分散とコンテンツスイッチング（オプション）仮想サーバーを追加します。
3. QUIC プロトコルパラメータを HTTP_QUIC 仮想サーバに関連付けます。
4. HTTP_QUIC 仮想サーバーで HTTP/3 を有効にします。

5. SSL 証明書とキーのペアを HTTP_QUIC 仮想サーバーとバインドします。
6. SSL/TLS プロトコルパラメータを HTTP_QUIC 仮想サーバーに関連付けます。

SSL と負荷分散を有効にする

開始する前に、アプライアンスで SSL 機能と負荷分散機能が有効になっていることを確認します。コマンドプロンプトで次のように入力します:

```
enable ns feature ssl lb
<!--NeedCopy-->
```

HTTP/3 サービスのタイプ HTTP_QUIC の負荷分散とコンテンツスイッチング (オプション) 仮想サーバーを追加します

QUIC 経由で HTTP/3 トラフィックを受け入れるように、負荷分散仮想サーバーを追加します。
注: タイプ HTTP_QUIC の負荷分散仮想サーバーには、QUIC、SSL、および HTTP3 プロファイルが組み込まれています。ユーザー定義プロファイルを作成する場合は、新しいプロファイルを追加し、負荷分散仮想サーバーとバインドできます。

```
add lb vserver <vserver-name> HTTP_QUIC <IP-address> <UDP-listening-
    port>

add cs vserver <vserver-name> HTTP_QUIC <IP-address> <UDP-listening-
    port>
<!--NeedCopy-->
```

例:

```
add lb vserver lb-http3 HTTP_QUIC 1.1.1.1 443
add cs vserver cs-http3 HTTP_QUIC 10.10.10.10 443
```

QUIC プロトコルパラメータを HTTP_QUIC 仮想サーバに関連付ける

QUIC プロファイルを作成し、QUIC サービスの QUIC パラメータを指定し、それを負荷分散仮想サーバに関連付けることができます。ユーザー定義プロファイルを作成するか、組み込みの QUIC プロファイルを使用して、プロファイルを負荷分散仮想サーバーにバインドする必要があります。

ステップ 1: ユーザ定義の QUIC
プロファイルを設定するコマンドプロンプトで、次のように入力します。

```
set quic profile <profile_name> -transport_param <value>
<!--NeedCopy-->
```


例:

```
set quic profile quic_http3 -ackDelayExponent 10 -activeConnectionIDlimit
 4
```

異なる QUIC トランスポートパラメータは次のとおりです。

-ackDelayExponent. NetScaler ADC がリモート QUIC エンドポイントにアドバタイズする整数値。リモート QUIC エンドポイントが NetScaler ADC によって送信される QUIC ACK フレームの ACK 遅延フィールドをデコードするために使用する必要のある指数を示します。

-activeConnectionIDlimit. NetScaler ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値です。NetScaler ADC が保存するリモート QUIC エンドポイントからの QUIC 接続 ID の最大数を指定します。

-activeConnectionMigration. NetScaler ADC がリモート QUIC エンドポイントでアクティブな QUIC 接続移行を実行できるようにする必要があるかどうかを指定します。

-congestionCtrlAlgorithm. QUIC 接続に使用する輻輳制御アルゴリズムを指定します。

-initialMaxData. NetScaler ADC がリモート QUIC エンドポイントにアドバタイズする整数値で、QUIC 接続で送信できる最大データ量の初期値をバイト単位で指定します。

-initialMaxStreamDataBidiLocal. NetScaler ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値。NetScaler ADC によって開始される双方向 QUIC ストリームの初期フロー制御制限（バイト単位）を指定します。

-initialMaxStreamDataBidiRemote. NetScaler ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値。リモート QUIC エンドポイントによって開始される双方向 QUIC ストリームの初期フロー制御制限（バイト単位）を指定します。

-initialMaxStreamDataUni. NetScaler ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値。リモート QUIC エンドポイントによって開始される単方向ストリームの初期フロー制御制限（バイト単位）を指定します。

-initialMaxStreamsBidi. NetScaler ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値。リモート QUIC エンドポイントが開始する必要がある双方向ストリームの初期最大数を指定します。

-initialMaxStreamsUni. NetScaler ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値。リモート QUIC エンドポイントが開始する必要のある単一方向ストリームの初期最大数を指定します。

-maxAckDelay. NetScaler ADC がリモート QUIC エンドポイントにアドバタイズする整数値。NetScaler ADC が確認応答の送信を遅延する最大時間をミリ秒単位で指定します。

-maxIdleTimeout. NetScaler ADC がリモート QUIC エンドポイントにアドバタイズする整数値。QUIC 接続の最大アイドルタイムアウトを秒単位で指定します。NetScaler ADC とリモート QUIC エンドポイントによってアドバタイズされるアイドルタイムアウト値の最小値よりも長くアイドル状態のままであり、現在のプローブタイムアウト（PTO）の 3 倍の QUIC 接続は、NetScaler ADC によってサイレントに破棄されます。

-maxUDPPayloadSize. NetScaler ADC がリモート QUIC エンドポイントにアドバタイズする整数値。NetScaler ADC が QUIC 接続で受信する最大の UDP データグラムペイロードのサイズをバイト単位で指定します。

-newTokenValidityPeriod. NetScaler ADC によって送信される QUIC NEW_TOKEN フレームを介して発行されたアドレス検証トークンの有効期間を秒単位で指定する整数値です。

-retryTokenValidityPeriod. NetScaler ADC によって送信された QUIC 再試行パケットを介して発行されるアドレス検証トークンの有効期間を秒単位で指定する整数値です。

-statelessAddressValidation. NetScaler ADC が QUIC クライアントのステートレスアドレス検証を実行する必要があるかどうかを指定します。QUIC 接続の確立中に QUIC 再試行パケットでトークンを送信し、QUIC 接続の確立後に QUIC NEW_TOKEN フレームでトークンを送信する必要があります。

ステップ 2: ユーザ定義の QUIC プロファイルを http_quic タイプの負荷分散仮想サーバーに関連付けます。

コマンドプロンプトで入力します:

```
1 set lb vserver <name>@ [-IPAddress <ip_addr|ipv6_addr|*>@]  <
     serviceName>@] [-persistenceType <persistenceType>] [-
     quicProfileName <string>]
2 <!--NeedCopy-->
```

例:

```
set lb vserver lb-http3 -quicProfileName quic_http3
```

HTTP_QUIC 仮想サーバーで HTTP/3 を有効にしてバインドする

HTTP_QUIC 仮想サーバーで HTTP/3 を有効にするには、設定パラメータのセットが HTTP プロファイル設定に追加されます。設定を容易にするため、HTTP_QUIC 仮想サーバーを追加すると、新しいデフォルト/組み込みの HTTP プロファイルがアプライアンスで使用可能になります。プロファイルの HTTP/3 プロトコルサポートパラメータが ENABLED に設定され、HTTP_QUIC 仮想サーバにも制限されます（HTTP_QUIC 仮想サーバをユーザが追加した HTTP プロファイルに関連付けない場合に適用可能）。HTTP プロファイルの HTTP/3 パラメータの値は、QUIC プロトコルハンドシェイク中に TLS ALPN（アプリケーション層プロトコルネゴシエーション）拡張を処理するときに HTTP/3 プロトコルを選択し、アドバタイズするかどうかを決定します。

HTTP/3 プロファイルを作成し、HTTP/3 サービスと負荷分散仮想サーバーの HTTP パラメータを指定できます。ユーザー定義プロファイルを作成するか、組み込みの HTTP/3 プロファイルを使用して、プロファイルを負荷分散仮想サーバーにバインドする必要があります。

ステップ 1: ユーザ定義の HTTP/3
プロファイルを設定するコマンドプロンプトで、次のように入力します。

```
1 Add ns httpProfile <profile_name> -http3 ENABLED
2 <!--NeedCopy-->
```

例:

```
add ns httpProfile http3_quic -http3 ENABLED
```

ステップ 2: ユーザー定義の HTTP/3 プロファイルを http_quic
タイプの負荷分散仮想サーバーにバインドするコマンドプロンプトで、次のように入力します。


```
set lb vserver <name>@ [-IPAddress <ip_addr|ipv6_addr|*>@]  <
    serviceName>@] [-persistenceType <persistenceType>] [-
    httpProfileName <string>]
<!--NeedCopy-->
```

例:

```
set lb vserver lb-http3 -httpProfileName http3_quic
```

SSL 証明書とキーのペアを HTTP_QUIC 仮想サーバーでバインドする

暗号化されたトラフィックを処理するには、SSL 証明書とキーのペアを追加し、HTTP_QUIC 仮想サーバーにバインドする必要があります。

コマンドプロンプトで入力します:

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>

<!--NeedCopy-->
```

例:

```
bind ssl vserver lb-http3 -certkeyName rsa_certkeypair
```

詳細については、「SSL 証明書のバインド」トピックを参照してください。

SSL/TLS プロトコルパラメーターを HTTP_QUIC 仮想サーバーでバインドする

HTTP_QUIC タイプの仮想サーバーには、QUIC プロトコルが必須のセキュリティコンポーネントとして TLS 1.3 を使用するため、組み込みの TLS 1.3 サーバー機能があります。HTTP_QUIC 仮想サーバーを追加する際の設定を容易にするため、タイプ Quic-Frontend の新しいデフォルトまたは組み込みの SSL プロファイルが追加されます。SSL プロファイルでは、TLS 1.3 暗号化スイート（および楕円曲線）が設定された TLS 1.3 バージョンが有効になっています。SSL プロファイルは、新しく追加された HTTP_QUIC 仮想サーバーにバインドする必要があります。
SSL プロファイルを作成し、TLP 1.1 サービスと負荷分散仮想サーバーの SSL 暗号化パラメーターを指定できます。ユーザー定義プロファイルを作成するか、組み込みの SSL プロファイルを使用して、プロファイルを負荷分散仮想サーバーにバインドする必要があります。

ステップ 1: ユーザ定義の SSL
プロファイルを構成するコマンドプロンプトで、次のように入力します。

```
add ssl profile <name> -sslprofileType QUIC-FrontEnd
<!--NeedCopy-->
```

例:

```
add ssl profile ssl_profile1 -sslprofileType QUIC-FrontEnd -tls13
ENABLED -tls12 DISABLED -tls11 DISABLED -tls1 DISABLED
```


ステップ 2: ユーザー定義の SSL プロファイルを HTTP_QUIC
タイプの負荷分散仮想サーバーにバインドするコマンドプロンプトで、次のように入力します。

```
set ssl vserver <name>@ [-sslProfile <string>]
<!--NeedCopy-->
```

例**:**

```
set ssl vserver lb-http3 -sslprofile ssl_profile1
```

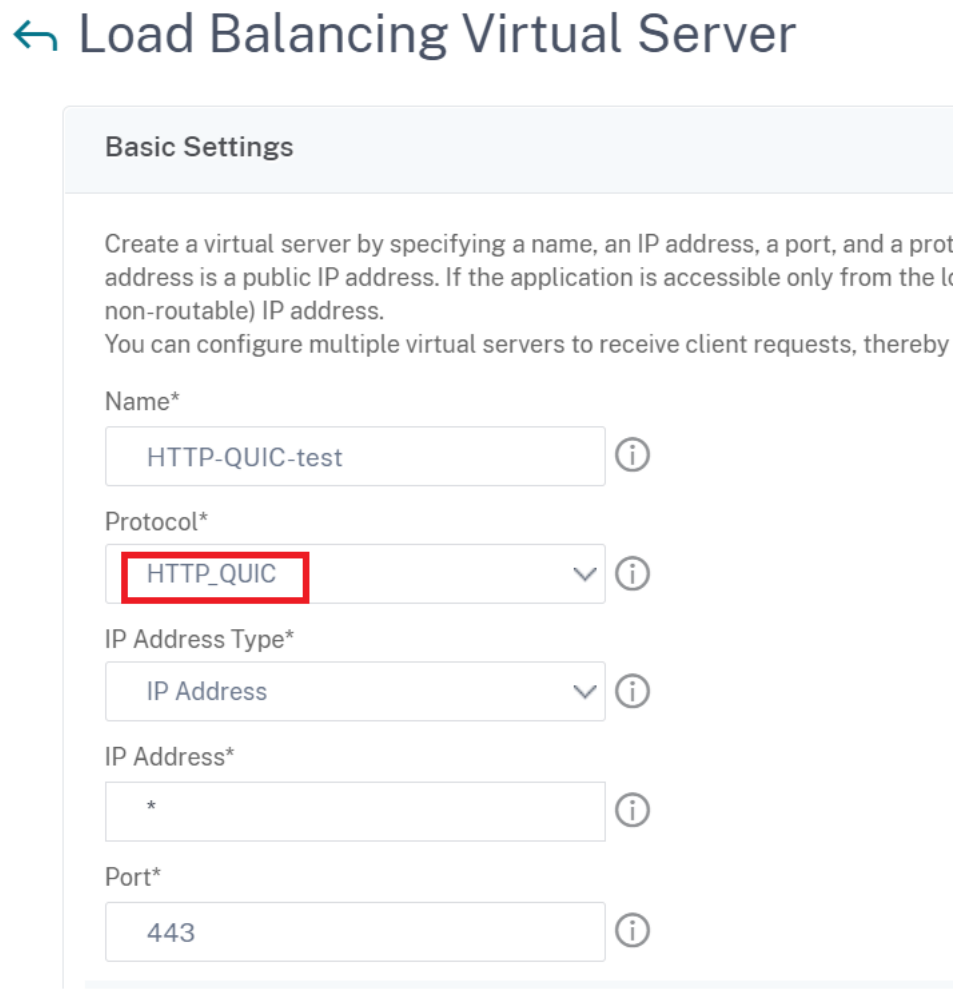
GUI を使用して **SSL** および負荷分散機能を有効にする

SSL および負荷分散機能を有効にするには、次の手順を実行します。

1. ナビゲーションウィンドウで、[システム] を展開し、[設定] をクリックします。
2. [基本機能の構成] ページで、[**SSL** と負荷分散] を選択します。
3. 「**OK**」をクリックし、「閉じる」をクリックします。

GUI を使用して **HTTP_QUIC** タイプの負荷分散とコンテンツスイッチング（オプション）仮想サーバーを追加します

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. [追加] をクリックして、タイプ HTTP_QUIC の負荷分散仮想サーバーを作成します。
3. **Load Balancing Virtual Server** ページで **Profiles** をクリックします。
4. [プロファイル] セクションで、プロファイルタイプを [QUIC] として選択します。注:QUIC、HTTP/3、および SSL プロファイルは組み込みのプロファイルです。
5. [**OK]**、[完了**]** の順にクリックします。



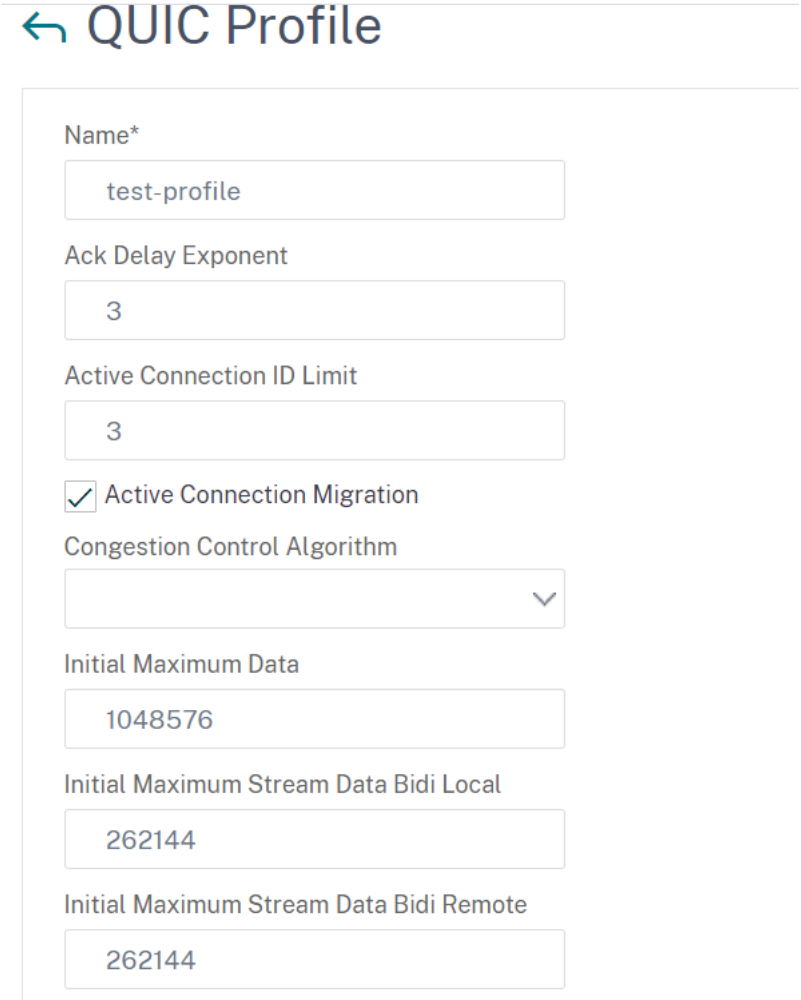
GUI を使用して **QUIC** プロトコルパラメータを **HTTP_QUIC** 仮想サーバに関連付けます

ステップ 1: QUIC プロファイルを追加する

1. [システム] **>** [プロファイル] **>** [**QUIC** プロファイル] に移動します。
2. [追加] をクリックします。
3. [QUIC プロファイル] ページで、次のパラメータを設定します。各パラメータの詳細については、「QUIC プロトコル CLI の関連付け」セクションを参照してください。

   a) `Ack Delay` 指数
   b) アクティブ接続 ID 制限
   c) アクティブな接続の移行
   d) 輻輳制御アルゴリズム
   e) 初期最大データ
   f) 初期最大ストリームデータ Bidi ローカル

g) 初期最大ストリームデータ Bidi リモート
h) 初期最大ストリームデータ単位
i) 初期最大ストリーム bidi
j) 初期最大ストリーム単位
k) 最大確認応答遅延
l) 最大アイドルタイムアウト
m) バーストあたりの最大 UDP データグラム数
n) 新しいトークンの有効期間
o) トークンの有効期間を再試行
p) ステートレスアドレス検証



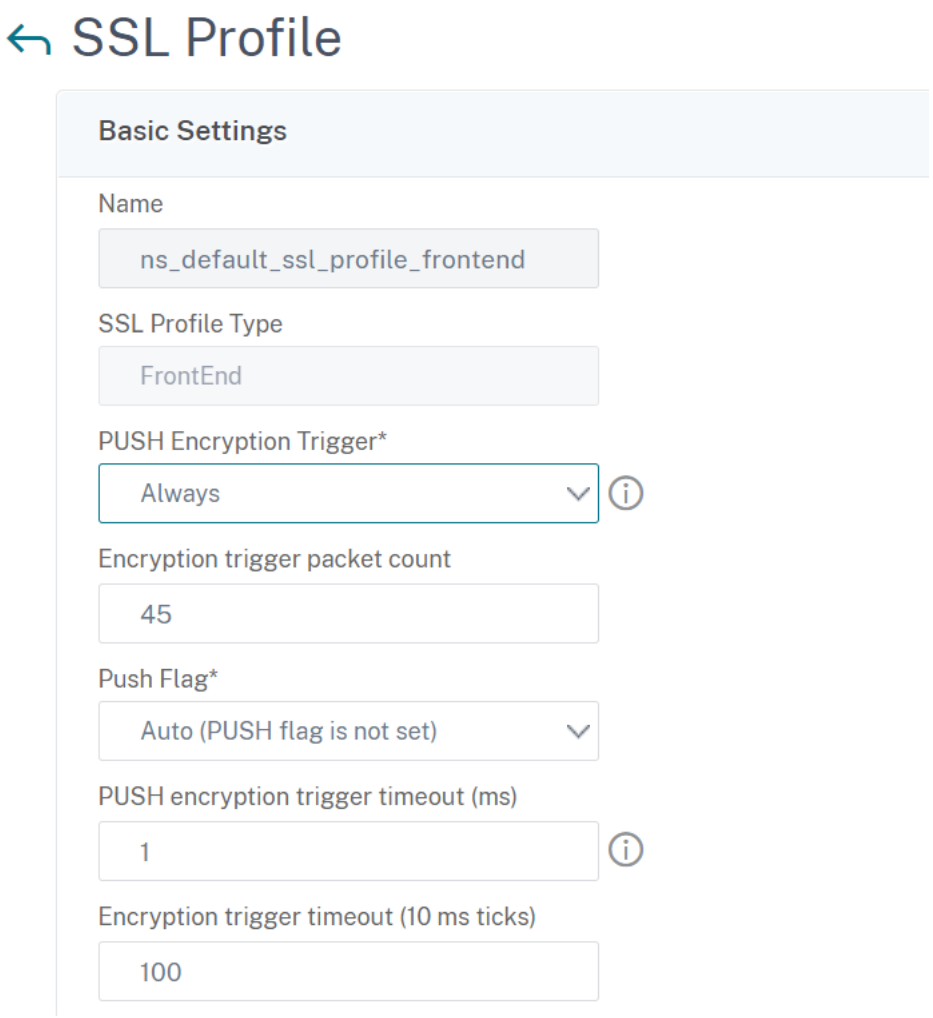
ステップ 2: QUIC プロファイルを HTTP_QUIC タイプの負荷分散仮想サーバーに関連付ける

1. [プロファイル] セクションで、QUIC プロファイルを選択します。注:QUIC、HTTP/3、および SSL プロファイルは組み込みのプロファイルです。
2. [**OK**]、[完了] の順にクリックします。

GUI を使用して、**SSL/TLS** プロトコルパラメータを **SSL** タイプの仮想サーバーに関連付けます

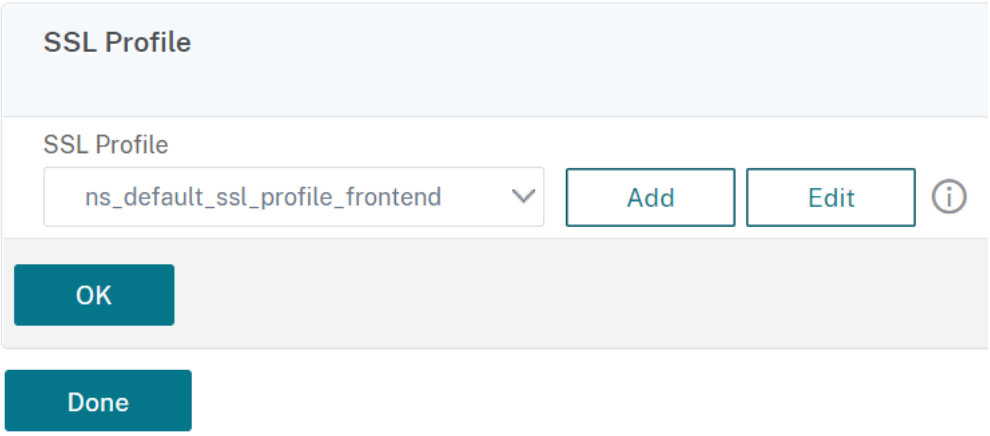
ステップ 1: SSL プロファイルを追加する

1. [システム] **>** [プロファイル] **>** [**SSL** プロファイル] に移動します。
2. [追加] をクリックします。
3. [**QUIC** プロファイル] ページで、SSL パラメータを設定します。詳細な説明については、SSL プロファイルの構成トピックを参照してください。
4. 「**OK**」をクリックして「閉じる」をクリックします。



ステップ 2: SSL プロファイルを SSL タイプの負荷分散仮想サーバーに関連付けます。

1. [プロファイル] セクションで、SSL プロファイルを選択します。
2. [**OK**]、[完了] の順にクリックします。



QUIC、および HTTP/3 の統計情報を表示する

次のコマンドは、QUIC および HTTP3 統計の詳細なサマリーを表示します。コマンドプロンプトで、次のように入力します:

```
> stat quic
> stat quic  - detail
<!--NeedCopy-->
```

統計情報の表示をクリアするには、次のいずれかを入力します。

```
> stat quic -clearstats basic
> stat quic -clearstats full

<!--NeedCopy-->
```

HTTP/3 統計の詳細なサマリーを表示するには、次の手順を実行します。

```
> stat http3
> stat http3  - detail
<!--NeedCopy-->
```

統計情報の表示をクリアするには、次のいずれかを入力します。

```
> stat http3 -clearstats basic
> stat http3 -clearstats full
<!--NeedCopy-->
```

HTTP/3 トラフィックのポリシー設定

August 15, 2023

HTTP/3 は UDP に基づく QUIC トランスポートを使用します。TCP ポリシー式を含む HTTP または SSL 仮想サーバに対して定義されたポリシー式がある場合、HTTP_QUIC 仮想サーバでは使用できなくなります。TCP またはクラシック式を持たない他のすべてのポリシーは、HTTP_QUIC 仮想サーバーでバインドできます。ポリシーを有効にするには、次のように、機能ポリシーが新しく追加されたグローバルバインドポイントにバインドされていることを確認する必要があります。

- HTTPQUIC_REQ_DEFAULT
- HTTPQUIC_REQ_OVERRIDE
- HTTPQUIC_RES_DEFAULT
- HTTPQUIC_RES_OVERRIDE

または、ポリシーを特定の仮想サーバーのバインドポイントにバインドできます。

- REQUEST
- RESPONSE

詳細については、「高度なポリシーインフラストラクチャを使用したポリシーのバインド」トピックを参照してください。

HTTP over QUIC 設定でサポートされているポリシーは次のとおりです。

- レスポンダー
- 書き換え
- HTTP 圧縮
- 統合キャッシング
- Web アプリケーションファイアウォール
- URL 変換
- SSL
- フロントエンド最適化 (FEO)
- AppQoE

HTTP/3 トラフィックのレスポンダポリシー設定

HTTP over QUIC タイプの仮想サーバには、レスポンダポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。

TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 QUIC 仮想サーバーまたは QUIC グローバルバインドポイント経由の HTTP にバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

URL をリダイレクトするためのレスポンダーアクションを追加する

レスポンダーアクションを追加するには、コマンドプロンプトで次のように入力します。


```
add responder action <name> <type> (<target> | <htmlpage>) [-comment <string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <expression>] [-headers <name(value)> ...]
<!--NeedCopy-->
```

例:

```
add responder action redirectURL redirect "\"https://www.citrix.com/\""
```

レスポンダーポリシーの追加

レスポンダーポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
add responder policy <name> <rule> <action> [<undefAction>] [-comment <string>] [-logAction <string>] [-appflowAction <string>]
<!--NeedCopy-->
```

例:

```
add responder policy res-pol "CLIENT.IP.SRC.IN_SUBNET(10.10.10.10/32)"redirectURL
```

レスポンダーポリシーベースの **UDP** 式を追加する

レスポンダーポリシーベースの UDP 式を追加するには、コマンドプロンプトで次のように入力します。

```
add responder policy <name> <rule> <action> [<undefAction>] [-comment <string>] [-logAction <string>] [-appflowAction <string>]
<!--NeedCopy-->
```

例:

```
add responder policy redirectCitrixUdp "CLIENT.UDP.DSTPORT.EQ(443)" redirectURL
```

HTTP/3 QUIC ベースの負荷分散仮想サーバーでレスポンダーポリシーベースの **UDP** 式をバインドする

レスポンダーポリシーベースの UDP 式を負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
bind lb vserver <name>@ ((<serviceName>@ [-weight <positive_integer>] ) | <serviceGroupName>@ | (-policyName <string>@ [-priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type <type>] [-invoke (<labelType> <labelName>) ] ) | -analyticsProfile <string>@)
<!--NeedCopy-->
```


例:

```
bind lb vserver lb-http3 -policyName redirectCitrixUdp -priority 9 -
gotoPriorityExpression END -type REQUEST
```

HTTP/3 QUIC ベースの負荷分散仮想サーバーでレスポンダーポリシーをバインドする

レスポンダーポリシーを負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
     ) | <serviceGroupName>@ |          (-policyName <string>@  [-
    priority <positive_integer>]  [-gotoPriorityExpression <expression>]
      [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
    analyticsProfile <string>@)
<!--NeedCopy-->
```

例:

```
bind lb vserver lb-http3 -policyName redirectCitrixUdp -priority 10 -
gotoPriorityExpression END -type REQUEST
```

HTTP/3 グローバルバインドポイントにレスポンダーポリシーをバインドする

HTTP/3 グローバルバインドポイントでレスポンダーポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
bind responder global <policyName> <priority>  [<gotoPriorityExpression
    >] [-type <type>] [-invoke (<labelType>  <labelName>) ] bind
    responder global redirectCitrixUdp 3 -type HTTPQUIC_REQ_DEFAULT
<!--NeedCopy-->
```

例:

```
bind responder global redirectCitrixUdp 3 -type HTTPQUIC_REQ_DEFAULT
```

> 注:
>
> 詳細については、レスポンダーポリシーのドキュメントを参照してください。

HTTP/3 トラフィックのポリシー設定を書き換え

HTTP over QUIC タイプの仮想サーバには、書き換えポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。

TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

次に、QUIC 上の HTTP3 の書き換えポリシーを設定する設定手順を示します。

QUIC 上の **HTTP** の書き換えアクションを追加

書き換えアクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-
     search <expression>] [-refineSearch <expression>] [-comment <string
     >]
2 <!--NeedCopy-->
```

例:

```
add rewrite action http3-altsvc-action insert_http_header Alt-Svc q/"
h3-29=\":443\"; ma=3600; persist=1"/
```

HTTP over QUIC の書き換えポリシーを追加する

書き込みアクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add rewrite policy <name> <rule> <action> [<undefAction>] [-comment  <
     string>] [-logAction <string>]
2 <!--NeedCopy-->
```

例:

```
add rewrite policy http3-altsvc-policy true http3-altsvc-action
```

HTTP/3_QUIC タイプの負荷分散仮想サーバーに書き換えポリシーをバインドする

書き換えポリシーを負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind lb vserver <name>@ ((<serviceName>@ [-weight <positive_integer>] )
      | <serviceGroupName>@ | (-policyName <string>@ [-priority <
     positive_integer>] [-gotoPriorityExpression <expression>] [-type <
     type>] [-invoke (<labelType> <labelName>) ] ) | -analyticsProfile  <
     string>@)
2 <!--NeedCopy-->
```

例:

```
bind lb vserver lb-http3 -policyName http3-altsvc-policy -priority 10
 -type RESPONSE
```


HTTP/3 グローバルバインドポイントに書き換えポリシーをバインドする

```
To bind a responder policy with HTTP/3 global bind point, at the
    command prompt, type:
bind rewrite global <policyName> <priority> [<gotoPriorityExpression>]
     [-type <type>] [-invoke (<labelType> <labelName>)]
<!--NeedCopy-->
```

例:

```
bind rewrite global http3-altsvc-policy 3 -type HTTPQUIC_RES_DEFAULT
```

> 注:
>
> 詳細については、「ポリシーの書き換え」を参照してください。

HTTP/3 トラフィックの圧縮ポリシー設定

NetScaler ADC はサーバーから HTTP 応答を受信すると、組み込みの圧縮ポリシーとカスタム圧縮ポリシーを評価して、応答を圧縮するかどうか、圧縮する場合は適用する圧縮の種類を決定します。ポリシーに割り当てられた優先順位によって、ポリシーがリクエストに対して照合される順序が決まります。
HTTP over QUIC タイプの仮想サーバには、圧縮ポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。
TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

圧縮ポリシーの追加

圧縮ポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
add cmp policy <name> -rule <expression> -resAction <string>
<!--NeedCopy-->
```

例:

```
add cmp policy udp_port_cmp_policy -rule "CLIENT.UDP.DSTPORT.EQ(443)"
-resAction COMPRESS
```

HTTP/3_QUIC タイプの負荷分散仮想サーバーで圧縮ポリシーをバインドする

HTTP/3_QUIC タイプの負荷分散仮想サーバーで URL 変換ポリシーをバインドするには、コマンドプロンプトで次のように入力します。


```
bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
    ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
    positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
     REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
    -analyticsProfile <string>@)
<!--NeedCopy-->
```

例:

```
bind lb vserver lb-http3 -policyName udp_port_cmp_policy -priority 10
 -type RESPONSE
```

HTTP/3 グローバルバインドポイントにグローバル圧縮をバインドする

HTTP/3 グローバルバインドポイントで圧縮ポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
bind compression global <policyName> <priority>  [<
    gotoPriorityExpression>] [-type <type>] [-invoke (<labelType>  <
    labelName>) ] bind responder global redirectCitrixUdp 3 -type
    HTTPQUIC_REQ_DEFAULT
<!--NeedCopy-->
```

例:

```
bind cmp global udp_port_cmp_policy -priority 100 -type HTTPQUIC_RES_DEFAULT
 Global built-in compression policies
```

アプライアンスを NetScaler ADC リリース 13.0 ビルド 82.x にアップグレードすると、次の圧縮ポリシーが HTTP/3 のデフォルトバインドポイントに自動的にバインドされます。

```
> sho cmp global -type HTTPQUIC_RES_DEFAULT
        Policy Name: ns_adv_nocmp_xml_ie
        Priority: 8700
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_nocmp_mozilla_47
        Priority: 8800
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_cmp_mscss
        Priority: 8900
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_cmp_msapp
        Priority: 9000
        GotoPriorityExpression: END
```



```
20          Type: HTTPQUIC_RES_DEFAULT
21
22          Policy Name: ns_adv_cmp_content_type
23          Priority: 10000
24          GotoPriorityExpression: END
25          Type: HTTPQUIC_RES_DEFAULT
26  <!--NeedCopy-->
```

バインドされていない場合、次のコマンドはコマンドプロンプトで設定でき、アプライアンスで設定できます。

```
bind cmp global ns_adv_nocmp_xml_ie -priority 8700 -gotoPriorityExpression
 END -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_nocmp_mozilla_47 -priority 8800 -gotoPriorityExpression
 END -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_mscss -priority 8900 -gotoPriorityExpression
 END -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_msapp -priority 9000 -gotoPriorityExpression
 END -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_content_type -priority 10000 -gotoPriorityExpressio
 END -type HTTPQUIC_RES_DEFAULT
```

詳細については、「圧縮ポリシーの設定」を参照してください。

HTTP/3 トラフィックのキャッシュポリシー設定

統合キャッシュは、NetScaler アプライアンスのメモリ内ストレージを提供し、オリジンサーバーへの往復を必要とせずにユーザーに Web コンテンツを提供します。静的コンテンツの場合、統合キャッシュの初期設定はほとんど必要ありません。統合キャッシュ機能を有効にし、基本的なセットアップ（たとえば、キャッシュが使用できる NetScaler ADC アプライアンスのメモリの量の決定）を実行すると、統合キャッシュは組み込みポリシーを使用して、単純な Web ページやイメージファイルなどの特定のタイプの静的コンテンツを格納および提供します。統合キャッシュを構成して、Web サーバーおよびアプリケーションサーバーによってキャッシュ不可としてマークされた動的コンテンツ（データベースレコードや株価など）を保存および提供することもできます。
HTTP over QUIC タイプの仮想サーバには、キャッシュポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。

TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

キャッシュコンテンツグループの追加

キャッシュコンテンツグループを追加するには、コマンドプロンプトで次のように入力します。


```
add cache contentGroup <name> [-weakPosRelExpiry <secs> | -relExpiry <
    secs> | -relExpiryMilliSec <msecs> | -absExpiry <HH:MM> ... | -
    absExpiryGMT <HH:MM> ...] [-heurExpiryParam <positive_integer>] [-
    weakNegRelExpiry <secs>] [-maxResSize <KBytes>] [-memLimit <MBytes>]
    ...
<!--NeedCopy-->
```

例:

```
add cache contentGroup DEFAULT -maxResSize 500
```

キャッシュポリシーの追加

キャッシュポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
add cache policy <policyName> -rule <expression> -action <action>  [-
    storeInGroup <string>] [-invalGroups <string> ...] [-invalObjects <
    string> ...] [-undefAction ( NOCACHE | RESET )] add cache policy <
    name> <rule> <profileName> [-comment <string>]  [-logAction <string
    >]
<!--NeedCopy-->
```

例:

```
add cache policy ctx_doc_pdf -rule "HTTP.REQ.URL.ENDSWITH(\".pdf\")"-
action CACHE -storeInGroup DEFAULT
```

HTTP/3_QUIC タイプの負荷分散仮想サーバーでキャッシュポリシーをバインドする

HTTP/3_QUIC タイプの負荷分散仮想サーバーでキャッシュポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
    ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
    positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
     REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
    -analyticsProfile <string>@)
<!--NeedCopy-->
```

例:

```
bind lb vserver lb-http3 -policyName ctx_doc_pdf -priority 100 -type
REQUEST
```

HTTP/3 グローバルバインドポイントにグローバルキャッシュポリシーをバインドする

キャッシュポリシー HTTP/3 グローバルバインドポイントをバインドするには、次の手順を実行します。


```
bind cache global <policy> -priority <positive_integer>  [-
    gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
    labelType> <labelName>) ]
<!--NeedCopy-->
```

例:

```
bind cache global ctx_doc_pdf -priority 3 -type HTTPQUIC_REQ_DEFAULT
```

詳細については、統合キャッシュポリシーの設定を参照してください。

グローバル組み込みキャッシュポリシー

アプライアンスを NetScaler ADC リリース 13.0 ビルド 82.x にアップグレードすると、次のキャッシュポリシーが自動的に HTTP/3 のデフォルトバインドポイントにバインドされます。

13.0 82.x リリースにアップグレードすると、次のキャッシュポリシーが HTTP/3 デフォルトのバインドポイントに自動的にバインドされます。

```
> sho cache global -type HTTPQUIC_REQ_DEFAULT
1)      Policy Name: NOPOLICY
        Priority: 185883
        GotoPriorityExpression: USE_INVOCATION_RESULT
        Invoke type: policylabel        Invoke name:
            _httpquicReqBuiltinDefaults
        Global bindpoint: HTTPQUIC_REQ_DEFAULT

 Done
> sho cache global -type HTTPQUIC_RES_DEFAULT
1)      Policy Name: NOPOLICY
        Priority: 185883
        GotoPriorityExpression: USE_INVOCATION_RESULT
        Invoke type: policylabel        Invoke name:
            _httpquicResBuiltinDefaults
        Global bindpoint: HTTPQUIC_RES_DEFAULT

<!--NeedCopy-->
```

アップグレード後、ポリシーがバインドされていない場合は、次のコマンドを使用して、手動で設定をバインドして保存できます。

```
add cache policylabel _httpquicReqBuiltinDefaults -evaluates
    HTTPQUIC_REQ

add cache policylabel _httpquicResBuiltinDefaults -evaluates
    HTTPQUIC_RES

bind cache policylabel _httpquicReqBuiltinDefaults -policyName
    _nonGetReq -priority 100
```



```
6
7  bind cache policylabel _httpquicReqBuiltinDefaults -policyName
       _advancedConditionalReq -priority 200
8
9  bind cache policylabel _httpquicReqBuiltinDefaults -policyName
       _personalizedReq -priority 300
10
11 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _uncacheableStatusRes -priority 100
12
13 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _uncacheableVaryRes -priority 200
14
15 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _uncacheableCacheControlRes -priority 300
16
17 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _cacheableCacheControlRes -priority 400
18
19 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _uncacheablePragmaRes -priority 500
20
21 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _cacheableExpiryRes -priority 600
22
23 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _imageRes -priority 700
24
25 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _personalizedRes -priority 800
26
27 bind cache global NOPOLICY -priority 185883 -gotoPriorityExpression
       USE_INVOCATION_RESULT -type HTTPQUIC_REQ_DEFAULT -invoke policylabel
        _httpquicReqBuiltinDefaults
28
29 bind cache global NOPOLICY -priority 185883 -gotoPriorityExpression
       USE_INVOCATION_RESULT -type HTTPQUIC_RES_DEFAULT -invoke policylabel
        _httpquicResBuiltinDefaults
30
31 <!--NeedCopy-->
```

> 注:
>
> コマンドリスト内の最初の 2 つのコマンドと、同じリストの最後の 2 つのコマンドは完全性のために含まれています。アプライアンスの再起動時にコマンドがすでに実行されているため、4 つのコマンドの実行時にエラーが発生することがあります。しかし、これらのエラーは無視してもかまいません。

HTTP/3 トラフィックの URL 変換ポリシー設定

URL 変換は、外部ユーザーが参照する外部バージョンからの指定されたリクエストに含まれるすべての URL を、Web サーバーおよび管理者だけが参照する内部 URL に変更します。ネットワーク構造をユーザーに公開することな

く、ユーザー要求をシームレスにリダイレクトできます。また、ユーザーが覚えにくい複雑な内部 URL を、よりシンプルで覚えやすい外部 URL に変更することもできます。
HTTP over QUIC タイプの仮想サーバには、キャッシュポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。
TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

URL 変換プロファイルを追加する

URL 変換プロファイルを追加するには、コマンドプロンプトで次のように入力します。

```
1 add transform profile <name> [-type URL]
2 <!--NeedCopy-->
```

例:

```
add transform profile msapps
```

URL 変換アクションを追加

URL 変換アクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add transform action <name> <profileName> <priority> [-state ( ENABLED
      | DISABLED )]
2 <!--NeedCopy-->
```

例:

```
add transform action docx2doc msapps 2
```

URL 変換アクションを追加

URL を置き換える URL 変換アクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add transform action <name> <profileName> <priority> [-state ( ENABLED
      | DISABLED )]
2 <!--NeedCopy-->
```

例:

```
add transform action docx2doc msapps 1
```


URL トランスフォームポリシーの追加

URL 変換ポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
1 add transform policy <name> <rule> <profileName> [-comment <string>]
     [-logAction <string>]
2 <!--NeedCopy-->
```

例:

```
add transform policy urltrans_udp "CLIENT.UDP.DSTPORT.EQ(443)"msapps
```

HTTP/3_QUIC タイプの負荷分散仮想サーバーで **URL** 変換ポリシーをバインドする

HTTP/3_QUIC タイプの負荷分散仮想サーバーで URL 変換ポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
     ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
     positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
      REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
     -analyticsProfile <string>@)
2 <!--NeedCopy-->
```

例:

```
bind lb vs lb-http3 -policyName urltrans_udp -type REQUEST -priority
8
```

HTTP/3 QUIC ベースの負荷分散仮想サーバーで **URL** 変換ポリシーをグローバルにバインドする

URL 変換ポリシー HTTP/3 グローバルバインドポイントをバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind transform global <policyName> <priority>  [<gotoPriorityExpression
     >] [-type <type>] [-invoke (<labelType>  <labelName>) ]
2 <!--NeedCopy-->
```

例:

```
bind transform global urltrans_udp 100 -type HTTPQUIC_REQ_DEFAULT
```

詳細については、URL 変換ポリシーの設定を参照してください。

HTTP/3 トラフィックに対するフロントエンド最適化 (FEO) ポリシー設定

Web アプリケーションの基礎となる HTTP プロトコルは、元々、単純な Web ページの送信とレンダリングをサポートするために開発されました。JavaScript やカスケードスタイルシート (CSS) などの新しいテクノロジや、Flash ビデオやグラフィックが豊富な画像などの新しいメディアタイプでは、フロントエンドのパフォーマンス、つまりブラウザーレベルでのパフォーマンスに多大な要求が課せられます。NetScaler ADC フロントエンド最適化（FEO）機能は、このような問題に対処し、ウェブページの読み込み時間とレンダリング時間を短縮します。

> 注:
>
> `HTTP_QUIC _Override/Default_Request` タイプは FEO ポリシーグローバルバインディングではサポートされていません。

フロントエンド最適化 (FEO) アクションを追加

FEO アクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add feo action <name> [-pageExtendCache] [<cacheMaxage>][-
      imgShrinkToAttrib] [-imgGifToPng] [-imgToWebp] [-imgToJpegXR] [-
      imgInline] [-cssImgInline] [-jpgOptimize] [-imgLazyLoad] [-cssMinify
      ] [-cssInline] [-cssCombine] [-convertImportToLink] [-jsMinify] [-
      jsInline] [-htmlMinify] [-cssMoveToHead] [-jsMoveToEND][-
      domainSharding <string> <dnsShards> ...] [-clientSideMeasurements]
2
3 <!--NeedCopy-->
```

例:

```
add feo action feoact -imgGifToPng -pageExtendCache
```

フロントエンド最適化 (FEO) ポリシーの追加

FEO ポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
add feo policy <name> <rule> <action>
```

例:

```
add feo policy udp_feo_img "CLIENT.UDP.DSTPORT.EQ(443)"IMG_OPTIMIZE
```

HTTP/3_QUIC タイプの負荷分散仮想サーバーと FEO ポリシーをバインドします

HTTP/3_QUIC タイプの負荷分散仮想サーバーと FEO ポリシーをバインドするには、コマンドプロンプトで次のように入力します。


```
bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
     ) | <serviceGroupName>@ |          (-policyName <string>@  [-
    priority <positive_integer>]  [-gotoPriorityExpression <expression>]
      [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
    analyticsProfile <string>@)
<!--NeedCopy-->
```

例:

```
bind lb vserver lb-http3 -policyName udp_feo_img -priority 4 -gotoPriorityExpr
 END -type REQUEST
```

FEO ポリシーを HTTP/3 グローバルバインドポイントにバインドする

HTTP/3 グローバルバインドポイントにキャッシュポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
bind cache global <policy> -priority <positive_integer>  [-
    gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
    labelType> <labelName>) ]
<!--NeedCopy-->
```

例:

```
bind cache global ctx_doc_pdf -priority 3 -type HTTPQUIC_REQ_DEFAULT
```

詳細については、「フロントエンド最適化ポリシーの設定」を参照してください。

HTTP/3 トラフィックの SSL ポリシー設定

HTTP over QUIC タイプの仮想サーバには SSL ポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。
TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。
TLSv1.3 でサポートされているアクションを持つ SSL ポリシーは、HTTP/3 バインドポイントまたは仮想サーバーにのみ適用されます。

SSL ポリシーの追加

FEO ポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
add ssl policy <name> -rule <expression> [-action <string>] [-
    undefAction <string>] [-comment <string>]
```



```
<!--NeedCopy-->
```

例:

```
add ssl policy ssl-pol -rule CLIENT.SSL.IS_SSL -action NOOP
```

SSL ポリシーを HTTP/3 仮想サーバーにバインドする

SSL ポリシーを HTTP/3 仮想サーバーにバインドするには、コマンドプロンプトで次の手順を実行します。

```
bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

例:

```
bind ssl vserver lb-http3 -policyName ssl-pol -priority 4 -type
REQUEST
```

SSL ポリシーの UDP 式で SSL ポリシーを追加する

UDP 式で SSL ポリシーを追加するには、コマンドプロンプトで次の操作を行います。

```
add ssl policy <name> -rule <expression> [-action <string>] [-
    undefAction <string>] [-comment <string>]
<!--NeedCopy-->
```

例:

```
add ssl policy ssl_udp_clnt -rule "CLIENT.UDP.DSTPORT.EQ(443)"-action
 NOOP
```

SSL ポリシーを UDP 式で HTTP/3 仮想サーバーにバインドする

HTTP/3 仮想サーバーに UDP 式を持つ SSL ポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

例:

```
bind ssl vs lb-http3 -policyName ssl_udp_clnt -priority 8 -type
REQUEST
```


HTTP/3 トラフィックの **CLIENTHELLO** バインドポイントの **SSL** ポリシーを追加する

HTTP/3 トラフィックの CLIENTHELLO バインドポイントの SSL ポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind ssl policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```

例:

```
add ssl policy ssl-pol-ch -rule "CLIENT.SSL.CLIENT_HELLO.CIPHERS.
HAS_HEXCODE(0x1301)"-action RESET
```

SSL ポリシーを **CLIENTHello** バインドポイントにバインドする

SSL ポリシーを CLIENTHELLO バインドポイントにバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind ssl policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```

例:

```
bind ssl vs lb-http3 -policyName ssl-pol-ch -type CLIENTHELLO_REQ -
priority 100
```

SSL ポリシーを **HTTP/3** グローバルバインドポイントにバインドする

SSL ポリシーを HTTP/3 グローバルバインドポイントにバインドするには、コマンドプロンプトで次のように入力します。

```
bind cache global <policy> -priority <positive_integer> [-gotoPriorityExpressi
 <expression>] [-type <type>] [-invoke (<labelType> <labelName>)]
```

例:

HTTP/3 グローバルバインドポイントにバインドされている DATA ポリシーの例を次に示します。

```
Bind ssl global -policyName ssl-pol-ch -priority 7 -type HTTPQUIC_DATA_DEFAULT
```

> 注:

> SSL 仮想サーバーの CLIENTHELLO バインドポイントに設定できる転送アクションは、HTTP_QUIC タイプの仮想サーバーでは現在サポートされていません。

HTTP/3 トラフィック用のアプリケーションファイアウォールポリシーの設定

HTTP over QUIC タイプの仮想サーバには、Web アプリケーションファイアウォールポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。
TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

UDP 式で **Web** アプリケーションファイアウォールポリシーを追加する

UDP 式を使用して Web アプリケーションファイアウォールポリシーを追加するには、コマンドプロンプトで次の操作を行います。

```
add appfw policy <name> <rule> <profileName> [-comment <string>] [-
    logAction <string>]
<!--NeedCopy-->
```

例:

```
add appfw policy appfw_udp "CLIENT.UDP.DSTPORT.EQ(443)"APPFW_BYPASS
```

Web アプリケーションファイアウォールプロファイルの **UDP** ベースの式を使用してログ式をバインドする

UDP for Web アプリケーションファイアウォールプロファイルでログ式をバインドするには、コマンドプロンプトで次の操作を行います。

例:

```
bind appfw profile APPFW_BLOCK -logExpression logexp-1 "CLIENT.UDP.
DSTPORT.EQ(443)"
```

HTTP/3 仮想サーバーでアプリケーションファイアウォールポリシーをバインドする

Web アプリケーションファイアウォールポリシーを HTTP/3 仮想サーバーでバインドするには、コマンドプロンプトで次の手順を実行します。

```
bind appfw policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```


例:

```
bind lb vs lb-http3 -policyName appfw_udp -priority 3 -type REQUEST
```

Web アプリケーションファイアウォールポリシーを HTTP/3 グローバルバインドポイントにバインドする

Web アプリケーションファイアウォールポリシーを HTTP/3 グローバルバインドポイントにバインドするには、コマンドプロンプトで次のように入力します。

```
bind appfw global <policy> -priority <positive_integer>  [-
    gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
    labelType> <labelName>) ]
<!--NeedCopy-->
```

例:

```
bind appfw global appfw_udp 100 -type HTTPQUIC_REQ_DEFAULT
```

HTTP/3 トラフィックの AppQoE ポリシー設定

HTTP over QUIC タイプの仮想サーバーには AppQoE ポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。
TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

UDP ベースの式で AppQoE ポリシーを追加する

UDP 式で AppQoE ポリシーを追加するには、コマンドプロンプトで次の操作を行います。

```
add AppQoE policy <name> <rule> <profileName> [-comment <string>] [-
    logAction <string>]
<!--NeedCopy-->
```

例:

```
add appqoe policy appqoe-pol-udp -rule "CLIENT.UDP.DSTPORT.EQ(443)"-
action appqoe-act-basic-prhigh
```

AppQoE ポリシーを HTTP/3 仮想サーバーでバインドする

AppQoE ポリシーを HTTP/3 仮想サーバーとバインドするには、コマンドプロンプトで次のように入力します。


```
bind appqoe policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

例:

```
bind lb vs lb-http3 -policyName appqoe-pol-udp -type REQUEST -priority
 3
```

AppQoE ポリシーを HTTP_QUIC 仮想サーバーにバインドする

AppQoE HTTP_QUIC ポリシーを仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
bind appqoe <policy> -priority <positive_integer>  [-
    gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
    labelType> <labelName>) ]
<!--NeedCopy-->
```

例:

```
bind lb vs lb-http3 -policyName appqoe-pol-primd -priority 8 -type
REQUEST
```

HTTP/3 サービスの検出

August 15, 2023

HTTP プロトコルは、同等のサービスの可用性をアドバタイズするために、オリジンサーバーに HTTP 代替サービスを使用することに依存しています。HTTP/3 サービスディスカバリも同じ原理を使用します。代替の HTTP/3 エンドポイントは、次のいずれかの方法を使用してアドバタイズできます。

- HTTP Alt-Svc レスポンスヘッダー
- HTTP/2 レスポンス内の Alt-Svc フレーム
- アプリケーション層プロトコルネゴシエーション (ALPN)

代替サービスは、HTTP Alt-Svc 応答ヘッダーと HTTP/2 Alt-Svc フレームの使用を HTTP/3 エンドポイントとしてアドバタイズします。サーバーは任意の UDP ポートで HTTP/3 を提供できます。代替サービスアドバタイズメントには明示的なポートが含まれ、URL にはスキームに関連付けられた明示的なポートまたはデフォルトポートのいずれかが含まれます。

代替サービスヘッダーまたはフレームを受信するクライアントは、それらを使用するようにバインドされません。クライアントは、代替サービスを認識し、代替サービスメカニズムをサポートしている場合は、アドバタイズされた適

切な代替サービスを使用する必要があります。つまり、HTTP/1.1 サービスまたは HTTP/2 サービスは、HTTP/3 プロトコルをサポートする同等のエンドポイントをアドバタイズする可能性があります。この代替サービス情報を受信するクライアントは、指定された代替サービスとの QUIC 接続を確立することを選択できます。使用可能になると、この接続を後続の要求に使用できます。選択した代替サービスとの接続の確立に失敗した場合、クライアントは元のエンドポイントにフォールバックできます。クライアントがアドバタイズされた代替サービスの使用を開始すると、は Alt-Used ヘッダーを含めることでこれを示します。

NetScaler ADC は、HTTP および SSL タイプの仮想サーバー上の同等の HTTP/3 エンドポイントの広告をサポートしています。

HTTP/3 サービスディスカバリを構成する

HTTP/3 サービスディスカバリを設定するには、次の手順を実行します。

1. HTTP Alt-Svc ヘッダーを使用して HTTP/3 代替サービスエンドポイントを構成する
2. HTTP/2 Alt-Svc フレームを使用した HTTP/3 代替サービスエンドポイントの設定 HTTP Alt-Svc ヘッダーを使用して HTTP/3
   代替サービスエンドポイントを構成する HTTP Alt-Svc ヘッダーを使用して HTTP/3
   エンドポイントをアドバタイズするには、次のコマンドを入力します。

注: 代替サービスを宣伝する主な目的は、HTTP/1.1 または.b.cd: 443 の HTTP/2 サービスでも HTTP/3 機能にアクセスできることをユーザーに知らせることです。

```
add ns httpProfile <name> -custom -altsvc [ ENABLED | DISABLED ]
<!--NeedCopy-->
```

例:

```
add ns httpProfile http-profile -altsvc ENABLED -altSvcValue  "h3-29="
    :443"; ma=3600; persist=1"
<!--NeedCopy-->
```

または

```
set ns httpProfile http-custom -altsvc ENABLED -altSvcValue  "h3-29="
    :443"; ma=3600; persist=1"
<!--NeedCopy-->
```

HTTP/2 Alt-Svc フレームを使用して **HTTP/3** 代替サービスエンドポイントを構成する

HTTP/2 Alt-svc フレームを使用して HTTP/3 エンドポイントをアドバタイズするには、次のコマンドを入力します。

```
add ns httpProfile <name> -custom -altsvc [ ENABLED | DISABLED ] -
    http2AltSvcFrame [ ENABLED | DISABLED ]
```



```
<!--NeedCopy-->
```

例:

```
add ns httpProfile http-custom -http2 ENABLED -http2Direct ENABLED -
http2AltSvcFrame ENABLED -altsvc ENABLED -altSvcValue "h3-29=\":443\";
 ma=3600; persist=1"
```

または

```
set ns httpProfile http-custom -http2 ENABLED -http2Direct ENABLED -
http2AltSvcFrame ENABLED -altsvc ENABLED -altSvcValue "h3-29=\":443\";
 ma=3600; persist=1"
```

GUI を使用して **HTTP Alt-Svc** ヘッダー値を使用した **HTTP/3** 代替サービスを構成する

1. [システム] **>** [プロファイル] **>** [**HTTP** プロファイル] に移動します。
2. [追加] をクリックします。
3. [**HTTP** プロファイルの作成] ページで、[HTTP/3] セクションに移動し、[代替サービス] チェックボックスをオンにします。
4. http2 セクションに [代替サービス値] テキストボックスが表示されます。
5. 代替サービス値を”h3-29=”：443”、ma=3600、永続 =1」と入力します。
6. 「**OK**」をクリックして「閉じる」をクリックします。

gRPC

August 15, 2023

NetScaler アプライアンスの gRPC は、軽量で高性能なオープンソースのユニバーサルリモートプロシージャコール（RPC）フレームワークです。このフレームワークは、どのオペレーティングシステムでも実行される複数の言語

で動作するのに最適です。また、他のプロトコルと比較すると、gRPC はパフォーマンスとセキュリティが優れています。

NetScaler 用 gRPC が推奨される理由は次のとおりです。

- データセンターとパブリック/プライベートクラウドインフラストラクチャ用の分散アプリケーションを構築します。
- モバイル、ウェブ、またはクラウド用のクライアント/サーバー通信を提供します。
- クラウドサービスとアプリケーションへのアクセス
- マイクロサービスデプロイメント

NetScaler に gRPC が選ばれる理由

NetScaler の gRPC は、高性能でスケーラブルな API をサポートするために、HTTP/2 を介して実装されています。テキストよりもバイナリを使用すると、ペイロードがコンパクトで効率的になります。NetScaler では、HTTP/2 リクエストは単一の TCP 接続で多重化されるため、ネットワークリソースの使用量を損なうことなく、複数のメッセージを同時に送信できます。また、ヘッダー圧縮を使用してリクエストとレスポンスのサイズを小さくします。

gRPC は、クライアントがパラメータや戻り値の型をリモートで呼び出すための以下のタイプのサービスメソッドをサポートしています。

1. 単項 **RPC**。クライアントは gRPC サーバに 1 つのリクエストを送信し、1 つの応答を返します。

   例:

   ```
   rpc SayHello(HelloRequest)returns (HelloResponse);
   ```

2. サーバーストリーミング **RPC**。クライアントは gRPC サーバーに単一のリクエストを送信し、ストリームレスポンスを取得します。

   例:

   ```
   rpc StreamingResponse(HelloRequest)returns (HelloResponse);
   ```

3. クライアントストリーミング **RPC**。クライアントは一連のメッセージを送信し、サーバーが応答を読み取って返すのを待ちます。

   例:

   ```
   rpc IntroduceYourself(stream HelloRequest)returns (HelloResponse)
   ```

4. 双方向ストリーミング **RPC**。クライアントとサーバーの両方が、読み取り/書き込みストリームを使用してメッセージのストリームを送信します。2 つのストリームは独立して動作します。

   例:

   ```
   rpc ChatSession (stream HelloRequest)returns (stream HelloResponse
   )
   ```

NetScaler は、gRPC エンドポイントを使用するサービスで次の機能をサポートしています。

- 負荷分散
- コンテンツの切り替え
- Web アプリケーションファイアウォール、認証などの安全なエンドポイントサービス。
- ポリシー設定
- 統計とロギング
- コンテンツの書き換え、コンテンツのフィルタリング
- レイヤー 4 とレイヤー 7 の最適化、TLS オファリング
- プロトコル変換用ゲートウェイソリューション

gRPC エンドツーエンド構成

January 11, 2024

gRPC のエンドツーエンド構成は、HTTP/2 プロトコルでクライアントから gRPC リクエストを送信し、gRPC サーバーが応答した gRPC メッセージを再度転送することで機能します。

エンドツーエンドの gRPC 設定の仕組み

次の図は、NetScaler ADC アプライアンスで機能する gRPC 構成を示しています。

1. gRPC 構成をデプロイするには、まず HTTP プロファイルで HTTP/2 を有効にし、サーバー側で HTTP/2 サポートをグローバルに有効にする必要があります。
2. クライアントが gRPC リクエストを送信すると、負荷分散仮想サーバーはポリシーを使用して gRPC トラフィックを評価します。
3. ポリシー評価に基づいて、負荷分散仮想サーバー（gRPC サービスがバインドされている）は要求を終了し、gRPC 要求としてバックエンドの gRPC サーバーに転送します。
4. 同様に、gRPC サーバーがクライアントに応答すると、アプライアンスは応答を終了し、gRPC 応答としてクライアントに転送します。

gRPC サーバーに送信された **gRPC** リクエストの例

リクエストヘッダーは次のように送信されます HTTP/2 のヘッダー HEADERS+CONTINUATION フレーム。

```
```
HEADERS (flags = END_HEADERS)
: method = POST
: scheme = http
: path = /helloworld.citrix-adc/SayHello
: authority = 10.10.10.10.:80
grpc-timeout = 15
content-type = application/grpc+proto
grpc-encoding = gzip
DATA (flags = END_STREAM)
<Length-Prefixed Message>
<!--NeedCopy--> ```
```

### **gRPC** サーバーから **NetScaler ADC** アプライアンスへの **gRPC** 応答ヘッダーの例

レスポンスヘッダーとトレーラーのみは、単一の HTTP/2 HEADERS フレームブロックで配信されます。ほとんどの応答にはヘッダーとトレーラーの両方が含まれることが予想されますが、Trailers-Only は、すぐにエラーが発生する呼び出しでは許可されます。HTTP ステータスコードが OK の場合でも、ステータスは Trailers で送信する必要があります。

```
```
HEADERS (flags = END_HEADERS)
: status = 200
Grpc-encoding= gzip
Content-type = application/grpc+proto
DATA
<Length-Prefixed Message>
HEADERS (flags = END_STREAM, END_HEADERS)
grpc-status = 0 # OK

<!--NeedCopy--> ```
```

CLI を使用して **gRPC** を構成する

エンドツーエンドの gRPC デプロイメントを設定するには、以下を完了する必要があります。

- HTTP/2 および HTTP/2 ダイレクトが有効になっている HTTP プロファイルを追加します。
- HTTP パラメータでグローバルバックエンド HTTP/2 サポートを有効にする
- SSL/HTTP タイプの負荷分散仮想サーバーを追加し、HTTP プロファイルを設定します
- gRPC エンドポイント用サービスの追加と HTTP プロファイルの設定
- gRPC エンドポイントサービスを負荷分散仮想サーバーにバインド

HTTP/2 および **HTTP/2** ダイレクトが有効な状態で **HTTP** プロファイルを追加

HTTP プロファイルで HTTP/2 と HTTP/2 のダイレクトパラメータを有効にする必要があります。また、gRPC over HTTP/2 クリアテキストが必要な場合は、HTTP/2 ダイレクトパラメータを有効にする必要があります。

コマンドプロンプトで入力します:

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct
 ( ENABLED | DISABLED )]
```

例:

```
add ns httpProfile http2gRPC -http2Direct ENABLED -http2 ENABLED
```

HTTP パラメータによるグローバルバックエンド **HTTP/2** サポートを有効にする

NetScaler コマンドラインを使用して、サーバー側で HTTP/2 サポートをグローバルに有効にします。

コマンドプロンプトで入力します:

```
set ns httpParam -http2ServerSide( ON | OFF )
```

例:

```
set ns httpParam -http2ServerSide ON
```

SSL/HTTP タイプの負荷分散仮想サーバーを追加し、**HTTP** プロファイルを設定します

NetScaler コマンドインターフェイスを使用して負荷分散仮想サーバーを追加するには:

コマンドプロンプトで入力します:

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-
httpProfileName <string>]
```

例:

```
add lb vserver lb-grpc HTTP 10.10.10.11 80 -httpProfileName http2gRPC
```

> 注:
>
> SSL タイプの負荷分散仮想サーバーを使用している場合は、サーバー証明書をバインドする必要があります。詳細については、「サーバー証明書のバインド」のトピックを参照してください。

gRPC エンドポイント用サービスの追加と **HTTP** プロファイルの設定

NetScaler コマンドインターフェイスを使用して **HTTP** プロファイル付きの **gRPC** サービスを追加するには、コマンドプロンプトで次のように入力します。

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-
httpProfileName <string>]
```

例**:**

```
add service svc-grpc 10.10.10.10 HTTP 80 -httpProfileName http2gRPC
```

gRPC エンドポイントサービスを負荷分散仮想サーバーにバインド

NetScaler コマンドインターフェイスを使用して **gRPC** サービスを負荷分散仮想サーバーにバインドするには:

コマンドインターフェイスで、次のように入力します。

```
bind lb vserver <name> <serviceName>
```

例**:**

```
bind lb vserver lb-grpc svc-grpc
```

GUI を使用してエンドツーエンドの **gRPC** デプロイメントを設定

GUI を使用して gRPC を設定するには、次の手順を実行します。

HTTP/2 および **HTTP/2** ダイレクトが有効な状態で **HTTP** プロファイルを追加

1. **System > Profiles** に移動して **HTTP Profiles** をクリックします。
2. 新しい **HTTP** プロファイルまたは既存の **HTTP** プロファイルの **HTTP/2** チェックボックスを選択します。

HTTP パラメータでグローバルバックエンド **HTTP/2** サポートを有効にする

1. [システム**] > [**設定**] > [HTTP** パラメータ] に移動します。
2.「**HTTP** パラメーターの設定」ページで、「サーバー側の **HTTP/2** 」チェックボックスを選択します。
3. [**OK**] をクリックします。

SSL/HTTP タイプの負荷分散仮想サーバーを追加し、**HTTP** プロファイルを設定します

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. [追加] をクリックして gRPC トラフィック用の負荷分散仮想サーバーを作成します。

3. Load Balancing Virtual Server ページで Profiles をクリックします。
4.「プロファイル」セクションで、プロファイルの種類を「HTTP」として選択します。
5. [OK] 、[完了] の順にクリックします。

gRPC エンドポイント用サービスの追加と **HTTP** プロファイルの設定

1. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。
2. [追加] をクリックして gRPC トラフィック用のアプリケーションサーバーを作成します。
3. 負荷分散サービスのページで、プロファイルセクションに移動します。
4. [プロファイル] に gRPC エンドポイントの HTTP プロファイルを追加します。
5. [OK] 、[完了] の順にクリックします。

ロードバランシングに関連する GUI 手順の詳細については、「負荷分散 」のトピックを参照してください。

gRPC ブリッジング

January 11, 2024

クライアントがリクエストを送信するとき HTTP/1.1 プロトコルでは、NetScaler ADC アプライアンスは gRPC リクエストのブリッジングをサポートします HTTP/1.1 gRPC サーバーに準拠しているプロトコル HTTP/2 プロトコル。同様に、リバースブリッジでは、アプライアンスは HTTP/2 プロトコルでクライアントの gRPC 要求を受信し、HTTP/1.1 プロトコルの gRPC サーバーに準拠して gRPC 要求に対してリバースブリッジを実行します。

gRPC ブリッジングの仕組み

このシナリオでは、NetScaler アプライアンスは HTTP/1.1 接続で受信した gRPC コンテンツをシームレスにブリッジし、HTTP/2 経由でバックエンド gRPC サーバーに転送します。

次の図は、gRPC ブリッジ構成でコンポーネントが互いにどのように相互作用するかを示しています。

1. gRPC リクエストが送信されると、NetScaler アプライアンスは接続が HTTP/1.1 で、コンテンツタイプがアプリケーション/grpc であるかどうかを確認します。HTTP/1.1 リクエストは、次の疑似ヘッダーに変換されます。
2. Content-Type ヘッダーで示されているように HTTP/1.1 接続で gRPC リクエストを受信すると、ADC アプライアンスはそのリクエストを以下に示すように HTTP/2 経由の gRPC に変換します。

```
    :method: Method-name in HTTP/1.1 request
    :path: Path is HTTP/1.1 request
    content-type: application/grpc
<!--NeedCopy-->
```

1. ポリシー評価に基づいて、負荷分散仮想サーバー（gRPC サービスがバインドされている）は要求を終了するか、HTTP/2 フレームを介してバックエンド gRPC サーバーに転送します。
2. gRPC サーバーから HTTP/2 接続で応答を受信すると、アプライアンスは HTTP/2 トレーラーを受信するまでバッファリングを行い、次に gRPC ステータスコードを確認します。gRPC エラーステータスがゼロ以外の場合、アプライアンスはマッピング HTTP ステータスコードを探し、適切な HTTP/1.1 エラー応答を送信します。

CLI を使用して gRPC ブリッジングを設定する

gRPC ブリッジを設定するには、次の手順を実行する必要があります。

1. HTTP/2 および HTTP/2 ダイレクトが有効な状態で HTTP プロファイルを追加
2. HTTP パラメータでグローバルバックエンド HTTP/2 サポートを有効にする
3. SSL/HTTP タイプの負荷分散仮想サーバーを追加し、HTTP プロファイルを設定します
4. gRPC エンドポイントのサービスを追加し、HTTP プロファイルを設定します
5. gRPC エンドポイントサービスを負荷分散仮想サーバーにバインド
6. gRPC ステータスコードを 0 以外の gRPC ステータスの HTTP レスポンスにマッピングします
7. gRPC バッファリングを時間またはサイズで設定

HTTP/2 および HTTP/2 ダイレクトが有効な状態で HTTP プロファイルを追加します

設定を開始するには、HTTP プロファイルで HTTP/2 機能を有効にする必要があります。クライアントが HTTP 1.1 リクエストを送信すると、アプライアンスはリクエストをブリッジしてバックエンドサーバーに転送します。

コマンドプロンプトで入力します：

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct
 ( ENABLED | DISABLED )]
```

例:

```
add ns httpProfile http2gRPC -http2Direct ENABLED -http2 ENABLED
```


HTTP パラメータでグローバルバックエンド **HTTP/2** サポートを有効にする

NetScaler コマンドラインを使用して、サーバー側で HTTP/2 サポートをグローバルに有効にします。

コマンドプロンプトで入力します:

```
set ns httpParam -http2ServerSide( ON | OFF )
```

例:

```
set ns httpParam -http2ServerSide ON
```

SSL/HTTP タイプの負荷分散仮想サーバーを追加し、**HTTP** プロファイルを設定します

NetScaler コマンドインターフェイスを使用して負荷分散仮想サーバーを追加するには

コマンドプロンプトで入力します:

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-
httpProfileName <string>]
```

例:

```
add lb vserver lb-grpc HTTP 10.10.10.10 80 -httpProfileName http2gRPC
```

> 注:
>
> SSL タイプの負荷分散仮想サーバーを使用している場合は、サーバー証明書をバインドする必要があります。詳細については、「サーバー証明書のバインド」のトピックを参照してください。

gRPC エンドポイントのサービスを追加し、**HTTP** プロファイルを設定します

NetScaler のコマンドインターフェイスを使用して、HTTP プロファイルで gRPC サービスを追加するには

コマンドプロンプトで入力します:

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-
httpProfileName <string>]
```

例:

```
add service svc-grpc 10.10.10.10 HTTP 80 -httpProfileName http2gRPC
```

gRPC エンドポイントサービスを負荷分散仮想サーバーにバインド

CLI を使用して gRPC エンドポイントサービスを負荷分散仮想サーバーにバインドすること。

コマンドインターフェイスで、次のように入力します。

```
bind lb vserver <name> <serviceName>
```

例**:**

```
bind lb vserver lb-grpc svc-grpc
```

gRPC ステータスコードを **HTTP/1.1** レスポンスの **HTTP** ステータスコードにマッピングします

gRPC ブリッジシナリオでは、gRPC サービスはリクエストに gRPC ステータスコードで応答します。アプライアンスは gRPC ステータスコードを対応する HTTP 応答コードと理由フレーズにマッピングします。マッピングは以下の表に基づいて行われます。送信時の NetScaler ADC アプライアンス HTTP/1.1 クライアントへの応答は、HTTP ステータスコードと理由フレーズを送信します。

gRPC ステータスコード	**HTTP** レスポンスステータスコード	**HTTP** 応答理由フレーズ
OK = 0	200	OK
キャンセル済み = 1	499	*
不明 = 2	500	内部サーバーエラー
引数が無効です = 3	400	不正な要求
期限超過 = 4	504	ゲートウェイタイムアウト
見つかりません = 5	404	*
すでに存在する = 6	409	コンフリクト
アクセス拒否 = 7	403	禁止
認証されていない = 16	401	無許可
リソースを使い果たした = 8	429	*
失敗 _ 前提条件 = 9	400	不正な要求
中止されました = 10	409	コンフリクト
範囲外 = 11	400	不正な要求
実装されていない = 12	501	未実装
インターナル = 13	500	内部サーバーエラー
ご利用いただけません = 14	503	サービスは利用できません
データ損失 = 15	500	内部サーバーエラー

gRPC バッファリングを時間またはサイズで設定

NetScaler アプライアンスは、レスポンストレーラーが受信されるまで、バックエンドサーバーからの gRPC 応答をバッファリングします。これにより、双方向の gRPC 呼び出しが中断されます。また、gRPC 応答が大きい場合は、応答を完全にバッファリングするために大量のメモリを消費します。この問題を解決するために、gRPC ブリッジ設定が拡張され、バッファリングを時間やサイズによって制限できるようになりました。バッファサイズまたは時間制限がしきい値を超えると、いずれかの制限がトリガーされた場合でも（設定されたバッファサイズ内でトレーラーが受信されないか、設定されたタイムアウトが発生した場合）、アプライアンスはバッファリングを停止し、クライアントに応答を転送します。その結果、設定されたポリシーとその表現（grpc-status コードに基づく）が期待どおりに機能しません。

CLI で gRPC バッファリングを時間やサイズによって制限するには、新しい HTTP プロファイルを追加するタイミングを設定するか、既存のプロファイルを変更するときに設定できます。

コマンドプロンプトで入力します:

```
add ns httpProfile http2gRPC [-grpcHoldLimit <positive_integer>] [-grpcHoldTimeout <positive_integer>]
```

または

```
set ns httpProfile http2gRPC [-grpcHoldLimit <positive_integer>] [-grpcHoldTimeout <positive_integer>]
```

各項目の意味は次のとおりです。

`grpcholdlimit`。トレーラーが受信されるまで gRPC パケットをバッファできる最大サイズ (バイト単位)。パラメータと任意のパラメータの両方を設定できます。

デフォルト値:131072
最小値:0
最大値:33554432

`grpcholdtimeout`。トレーラーが受信されるまでの gRPC パケットのバッファリングに許容される最大時間 (ミリ秒単位)。値は 100 の倍数でなければなりません。
デフォルト値:1000
最小値:0
最大値:180000

例**:**

```
add httpprofile http2gRPC -grpcholdlimit 1048576 -grpcholdtimeout 5000
set httpprofile http2gRPC -grpcholdlimit 1048576 -grpcholdtimeout 5000
```


GUI を使用して gRPC ブリッジを設定

NetScaler GUI を使用して gRPC ブリッジを構成するには、次の手順を実行します。

HTTP/2 および HTTP/2 ダイレクトが有効な状態で HTTP プロファイルを追加

1. **System > Profiles** に移動して **HTTP Profiles** をクリックします。
2. HTTP プロファイルで **HTTP/2** を選択します。

HTTP パラメータでグローバルバックエンド HTTP/2 サポートを有効にする

1. [システム**] > [**設定**] > [HTTP** パラメータ] に移動します。
2.「**HTTP** パラメータの設定」ページで、「サーバー側の **HTTP/2** 」オプションを選択します。
3.［**OK**］をクリックします。

SSL/HTTP タイプの負荷分散仮想サーバーを追加し、HTTP プロファイルを設定します

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. [追加] をクリックして gRPC トラフィック用の負荷分散仮想サーバーを作成します。
3. **Load Balancing Virtual Server** ページで **Profiles** をクリックします。
4.「プロファイル」セクションで、プロファイルの種類を「HTTP」として選択します。
5. [**OK]**、[完了**]** の順にクリックします。

gRPC エンドポイント用サービスの追加と HTTP プロファイルの設定

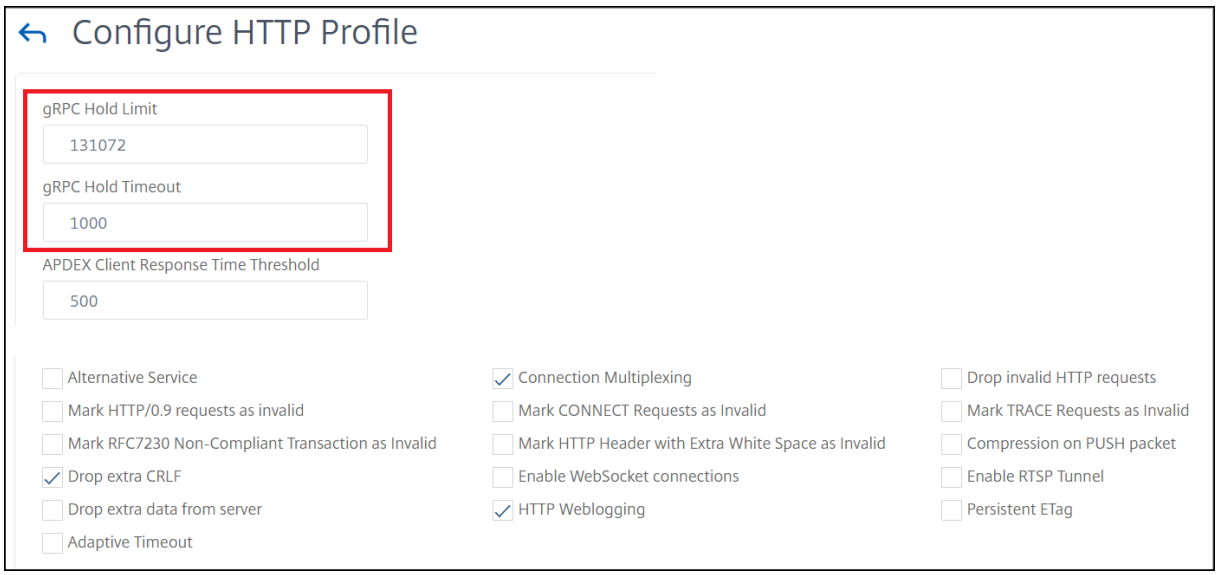
1.［**Traffic Management**］**>**［**Load Balancing**］**>**［**Services**］の順に移動します。
2. [追加] をクリックして gRPC トラフィック用のアプリケーションサーバーを作成します。
3. 負荷分散サービスのページで、プロファイルセクションに移動します。
4. [プロファイル] に gRPC エンドポイントの **HTTP** プロファイルを追加します。
5. [**OK]**、[完了**]** の順にクリックします。

gRPC エンドポイントを負荷分散仮想サーバーにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. [追加] をクリックして gRPC トラフィック用の負荷分散仮想サーバーを作成します。
3. [負荷分散仮想サーバー] ページで、[サービスとサービスグループ] セクションをクリックします。
4. 負荷分散仮想サーバーサービスバインディングページで、バインドする gRPC サービスを選択します。
5.「閉じる」をクリックし、「完了」をクリックします。

GUI を使用して **gRPC** バッファリングを時間とサイズで設定します

1. **System > Profiles** に移動して **HTTP Profiles** をクリックします。
2. HTTP プロファイルで **HTTP/2** を選択します。
3.「**HTTP** プロファイルの設定」ページで、次のパラメータを設定します。
   a) GRPC ホールドタイムアウト。トレーラーが受信されるまで gRPC パケットをバッファする時間をミリ秒単位で入力します。
   b) GRP/ホールドリミット。トレーラーが受信されるまで gRPC パケットをバッファする最大サイズをバイト単位で入力します。
4.「**OK**」をクリックして「閉じる」をクリックします。



サービスと負荷分散仮想サーバーをバインドするための GUI 手順の詳細については、「負荷分散 」のトピックを参照してください。

gRPC リバースブリッジング

January 11, 2024

このシナリオでは、NetScaler アプライアンスは HTTP/2 接続で受信した gRPC コンテンツをシームレスにブリッジし、HTTP/1.1 を介してバックエンド gRPC サーバーに転送します。

リバースブリッジの仕組み

次の図は、gRPC ブリッジ構成でコンポーネントが互いにどのように相互作用するかを示しています。

1. クライアントは、HTTP/2 フレームの gRPC ヘッダーと proto-buf ペイロードを使用して HTTP/2 接続で gRPC リクエストを送信します。
2. ポリシー評価に基づいて、負荷分散仮想サーバー（gRPC サービスがバインドされている）はリクエストを変換し、HTTP/1.1 接続を介してバックエンドサーバーに転送します。
3. HTTP/1.1 応答を受信したときに、応答に grpc-status コードが含まれていない場合、ADC は HTTP 応答コードから grpc ステータスケースを導出します。
4. 次に、アプライアンスは gRPC ヘッダーを HTTP/2 トレーラーに挿入してから、応答をクライアントに転送します。

CLI を使用して gRPC リバースブリッジを設定する

gRPC リバースブリッジを設定するには、次の手順を実行する必要があります。

- 仮想サーバーの負荷分散用に HTTP/2 および HTTP/2 ダイレクトを有効にした HTTP プロファイル 1 を追加します
- バックエンドサーバーに HTTP/2 を無効にした状態で HTTP プロファイル 2 を追加
- SSL/HTTP タイプの負荷分散仮想サーバーを追加し、HTTP プロファイル 1 に設定します
- gRPC エンドポイントのサービスを追加し、HTTP プロファイル 2 に設定
- gRPC エンドポイントを負荷分散仮想サーバーにバインドする
- 応答に grpc ステータスコードが含まれていない場合は、HTTP ステータスコードを gRPC ステータスコードにマッピングします。

仮想サーバーの負荷分散用に HTTP/2 および HTTP/2 ダイレクトを有効にした HTTP プロファイル 1 を追加します

リバースブリッジの設定を開始するには、2 つの HTTP プロファイルを追加する必要があります。1 つは gRPC クライアントリクエストの HTTP/2 を有効にするためのプロファイルで、もう 1 つは gRPC サーバー応答の HTTP/2 を無効にするプロファイルです。

コマンドプロンプトで入力します：

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct
 ( ENABLED | DISABLED )]
```


例:

add ns httpProfile profile1 -http2 ENABLED -http2Direct ENABLED

gRPC エンドポイントでサービスを追加し HTTP プロファイル 2 を設定する

NetScaler コマンドラインを使用して、バックエンドサーバー応答の HTTP プロファイルでの HTTP/2 サポートを無効にするには。

コマンドプロンプトで入力します:

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct
 ( ENABLED | DISABLED )]
```

例:

add ns httpProfile profile2 -http2 DISABLED http2Direct DISABLED

SSL/HTTP タイプの負荷分散仮想サーバーを追加し、HTTP プロファイル 1 に設定します

NetScaler のコマンドインターフェイスを使用して負荷分散仮想サーバーを追加します。

コマンドプロンプトで入力します:

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-
httpProfileName <string>]
```

例:

add lb vserver lb-grpc HTTP 10.10.10.10 80 -httpProfileName profile1

> 注:
>
> SSL タイプの負荷分散仮想サーバーを使用している場合は、サーバー証明書をバインドする必要があります。詳細については、「サーバー証明書のバインド」のトピックを参照してください。

gRPC エンドポイントのサービスを追加し、HTTP プロファイル 2 に設定

gRPC エンドポイントでサービスを追加し、NetScaler ADC コマンドインターフェイスを使用して HTTP プロファイル 2 を設定します。

コマンドプロンプトで入力します:

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-
httpProfileName <string>]
```

例:

```
add service svc-grpc 10.10.10.11 HTTP 80 -httpProfileName profile2
```


gRPC エンドポイントのサービスを負荷分散仮想サーバーにバインドする

NetScaler コマンドインターフェイスを使用して gRPC サービスを負荷分散仮想サーバーにバインドすること。

コマンドインターフェイスで、次のように入力します。

```
bind lb vserver <name> <serviceName>
```

例**:**

```
bind lb vserver lb-grpc svc-grpc
```

HTTP 応答コードを **gRPC** ステータスコードにマッピング

サーバーが gRPC ステータスコードを生成しない場合、NetScaler アプライアンスは受信した HTTP 応答に基づいて適切な gRPC ステータスコードを生成します。ステータスコードは以下のマッピングテーブルにリストされています。

HTTP レスポンスステータスコード	**gRPC** ステータスコード
200	OK
400	インターナル = 13
403	アクセス拒否 = 7
401	認証されていない = 16
429, 502, 503, 504	ご利用いただけません = 14
404	実装されていない = 12

GUI を使用して **gRPC** リバースブリッジを設定する

仮想サーバーの負荷分散用に **HTTP/2** および **HTTP/2** ダイレクトを有効にした **HTTP** プロファイル **1** を追加します

1. System > Profiles に移動して HTTP Profiles をクリックします。
2. **HTTP** プロファイル **1** で **HTTP/2** オプションを有効にします。

gRPC エンドポイントでサービスを追加し **HTTP** プロファイル **2** を設定する

1. **System > Profiles** に移動して **HTTP Profiles** をクリックします。
2. HTTP プロファイルの **HTTP/2** オプションを有効にする 2.
3. [**OK**] をクリックします。

SSL/HTTP タイプの負荷分散仮想サーバーを追加し、**HTTP** プロファイル **1** に設定します

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. [追加] をクリックして gRPC トラフィック用の負荷分散仮想サーバーを作成します。
3. **Load Balancing Virtual Server** ページで **Profiles** をクリックします。
4.「プロファイル」セクションで、プロファイルの種類を「HTTP」として選択します。
5. [**OK]**、[完了**]** の順にクリックします。

gRPC エンドポイントを使用してサービスを追加し、**HTTP** プロファイル **2** に設定します

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
2. [追加] をクリックして gRPC トラフィック用のアプリケーションサーバーを作成します。
3. 負荷分散サービスのページで、プロファイルセクションに移動します。
4. [プロファイル] に gRPC エンドポイントの **HTTP** プロファイルを追加します。
5. [**OK]**、[完了**]** の順にクリックします。

gRPC エンドポイントを負荷分散仮想サーバーにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. [追加] をクリックして gRPC トラフィック用の負荷分散仮想サーバーを作成します。
3. 負荷分散仮想サーバーページで、サービスとサービスグループセクションをクリックします。
4. 負荷分散仮想サーバーサービスバインディングページで、バインドする gRPC サービスを選択します。
5.「閉じる」をクリックし、「完了」をクリックします。

GUI の手順の詳細については、負荷分散のトピックを参照してください 。

gRPC コールターミネーション

August 15, 2023

NetScaler アプライアンスにレート制限や Web App Firewall セキュリティなどのポリシーが設定されていて、ポリシーが true と評価されると、アプライアンスは通話を終了し、計算可能な gRPC エラーメッセージをクライアントに送信できます。

書き換えポリシーを使用した **gRPC**

August 15, 2023

gRPC with rewrite policy のユースケースでは、NetScaler ADC アプライアンスが gRPC 要求または応答の一部の情報を書き換える際にどのように機能するかを説明しています。次の図は、コンポーネントの相互作用を示しています。

次の図は、書き換えポリシーが設定された gRPC でコンポーネントが互いにどのように相互作用するかを示しています。

1. アプライアンスの書き換え機能を有効にします。
2. gRPC ヘッダーを変更、追加、または削除する書き換えアクションを設定します。
3. アクションを実行する必要がある gRPC リクエスト（トラフィック）を決定するための書き換えポリシーを設定します。
4. 書き換えポリシーを負荷分散仮想サーバーにバインドして、トラフィックがポリシー表現と一致するかどうかを確認します。
5. リライトポリシーを使用すると、gRPC ステータスコードに基づいて次の操作を実行できます。

    a) gRPC Web サーバーからの応答を変更します。
    b) gRPC ヘッダーを変更、追加、または削除します。
    c) gRC サーバーへのリクエストの URL を変更します。

書き換えポリシーによる **gRPC** コール終了の設定

リライトポリシーで gRPC コールターミネーションを設定するには、次の手順を実行する必要があります。

1. 書き換え機能を有効にする
2. 書き換えポリシーを追加する
3. リライトポリシーを負荷分散仮想サーバーにバインドする

書き換え機能を有効にする

書き換え機能を使用するには、まずそれを有効にする必要があります。

コマンドプロンプトで入力します。

```
enable ns rewrite
```


書き換えポリシーを追加する

リライトアクションを構成した後、次にリライトポリシーを構成して、NetScaler ADC アプライアンスがリライトする必要のある gRPC リクエストを選択する必要があります。

コマンドプロンプトで入力します。

```
add rewrite policy <name> <expression> <action> [<undefaction>]-
appFlowaction <actionName>
```

例:

```
add rewrite policy grpc-rewr_pol1 "http.res.header(\"grpc-status\").
NE(\"0\")"RESET
```

リライトポリシーを負荷分散仮想サーバーにバインドする

ポリシーを有効にするには、gRPC サービスを使用してそのポリシーを負荷分散仮想サーバーにバインドする必要があります。

コマンドプロンプトで入力します。

```
bind rewrite global <policyName> <priority> [<gotoPriorityExpression>
 [-type <type>] [-invoke (<labelType> <labelName>)]
```

例:

```
bind lb vserver lb-grpc -policyName grpc-rewr_pol1 -priority 100
```

レスポンダーポリシーを持つ gRPC

August 15, 2023

レスポンダー付きの gRPC ポリシー構成では、NetScaler ADC アプライアンスが HTTP/2 プロトコルを介して gRPC 要求に対して異なる応答を提供する方法について説明します。ユーザーが Web サイトのホームページを要求するときは、各ユーザーの場所やユーザーが使用しているブラウザーに応じて、異なるホームページを提供することをお勧めします。

次の図は、相互作用するコンポーネントを示しています。

1. アプライアンスでレスポンダ機能を有効にします。
2. レスポンダーアクションを設定して、カスタムレスポンスを生成したり、リクエストを別の Web ページにリダイレクトしたり、接続をリセットしたりします。
3. アクションを実行する必要がある gRPC リクエスト (トラフィック) を決定するためのレスポンダーポリシーを設定します。
4. レスポンダーポリシーを負荷分散仮想サーバーにバインドして、トラフィックがポリシー式に一致するかどうかを調べます。
5. レスポンダーポリシーを使用すると、gRPC ステータスコードに基づいて以下を実行できます。

CLI を使用して **gRPC** コール終了をレスポンダポリシーで設定します

レスポンダポリシーを使用して gRPC コールの終了を設定するには、次の手順を実行する必要があります。

1. レスポンダー機能を有効にする
2. レスポンダーアクションを追加する
3. レスポンダーポリシーを追加し、レスポンダーアクションを関連付ける
4. レスポンダーポリシーを負荷分散仮想サーバーにバインドする

レスポンダー機能を有効にする

レスポンダー機能を使用するには、最初にそれを有効にする必要があります。

コマンドプロンプトで入力します。

```
enable ns responder
```

レスポンダーアクションを追加する

この機能を有効にした後、バックエンドサーバーから返されたステータスコードに基づいて gRPC 応答を処理するためのレスポンダーアクションを構成する必要があります。

コマンドプロンプトで入力します。


```
add responder action <name> <type>
```

例:

```
add responder action grpc-act respondwith "HTTP/1.1 200 OK\r\nServer:
 NS-Responder\r\nContent-Type:application/grpc\r\ngrpc-status: 12\r\
ngrpc-message: Not Implemented\r\n\r\n"+ "Method: "+ HTTP.REQ.URL+ "
is not implemented."
```

レスポンダーポリシーの追加

レスポンダーアクションを構成したら、次にレスポンダーポリシーを構成して、NetScaler アプライアンスが応答する必要がある gRPC リクエストを選択する必要があります。

コマンドプロンプトで入力します。

```
add responder policy <name> <expression> <action> [<undefaction>]-
appFlowaction <actionName>
```

例:

```
add responder policy grpc-resp-pol1 HTTP.REQ.URL.NE( “/helloworld.
Greeter/SayHello” )grpc-act
```

レスポンダーポリシーを負荷分散仮想サーバーにバインドする

ポリシーを有効にするには、gRPC サービスを使用してそのポリシーを負荷分散仮想サーバーにバインドする必要があります。

コマンドプロンプトで入力します。

```
bind responder global <policyName> <priority> [<gotoPriorityExpression
> [-type <type>] [-invoke (<labelType> <labelName>)]
```

例:

```
bind lb vserver lb-grpc svc-grpc -policyName grpc-resp-pol1 -priority
 100
```

レスポンダーポリシーの詳細については、「レスポンダーポリシー」トピックを参照してください。

gRPC プロトコルバッファフィールドを照合するためのポリシー式

NetScaler ADC アプライアンスは、gRPC 構成で次のポリシー式をサポートしています。

- **gRPC** プロトコルバッファフィールドアクセス。任意の gRPC API 呼び出しは、メッセージフィールド番号と新しいポリシー式と一致します。PI 構成では、一致は「フィールド番号」と「API パス」のみを使用して行われます。
- **gRPC** ヘッダーフィルタリング。gRPC の「HttpProfile」パラメータは、gRPC 解析のデフォルトの動作（gRPC ポリシー式を含む）を調整するために使用されます。gRPC ポリシー式には、次のパラメータが適用されます。
  - **gRPCLengthDelimitation**。これはデフォルトで有効になっており、プロトコルバッファに長さ区切りのメッセージが表示されることを想定しています。
  - **grpCholdLimit**。デフォルト値は 131072 です。これは、プロトコルバッファメッセージの最大サイズ（バイト単位）です。また、文字列の最大長と最大‘byte’フィールド長でもあります。

CLI を使用して **gRPC** アドバンスポリシー式を設定します

コマンドプロンプトで入力します。

```
set ns httpProfile <name> -http2 \( ENABLED | DISABLED ) -
    gRPCLengthDelimitation \( ENABLED | DISABLED ) -gRPCHoldLimit <int>
```

例**:**

```
set ns httpProfile http2gRPC -http2 ENABLED -gRPCLengthDelimitation
    ENABLED -gRPCHoldLimit 131072
```

GUI を使用して **gRPC** ヘッダーフィルタリングパラメータを設定する

1. **System > Profiles** に移動して **HTTP Profiles** をクリックします。
2. [**HTTP** プロファイルの作成] ページで、[**HTTP/3**] セクションまでスクロールダウンし、[**gRPC** の長さの区切り] を選択します。

次のポリシー式の例は、メッセージ 5、サブメッセージ 4、およびフィールド 3 の値を示しています。これは 2 に等しい 32 ビットの int です。

```
http.req.body(1000).grpc.message(5).message(4).int32(3).eq(2)
```

gRPC プロトコルバッファメッセージフィールドを番号で照合するために、次のポリシー式が追加されます。

- message
- ダブル
- フロート
- int32
- int64

- uint32
- uint64
- sint64
- sint32
- fixed32
- fixed64
- sfixed32
- sfixed64
- bool
- string
- 列挙型
- bytes

API パスマッチング

API パスマッチングは、複数の API が使用されている場合に、正しい gRPC API 呼び出しを照合するために使用されます。API パスと一致します。これは HTTP リクエストの ‘: path’ 疑似ヘッダーにあります。

例**:**

```
http.req.header(":path").eq("acme.inventory.v1/ListBooks")
```

gRPC ヘルスチェックモニター

January 9, 2024

gRPC ヘルスモニタは、gRPC サーバのヘルスステータスを調査します。gRPC ヘルスモニタは、gRPC サービスの全体的な健全性または特定のサービスの健全性をチェックします。現在、NetScaler アプライアンスはチェック方法のみをサポートしています。

NetScaler アプライアンスでは、HTTP2 モニター構成で、gRPCHealthCheck gRPCStatusCode、gRPCServiceName、httprequestなどのgRPCパラメーターを設定して、ヘルスチェックモニターを構成します。プロトコルを実装しているクライアントは、サーバーのステータス (正常、正常ではない、不明、またはサービスが実装されていない) を照会し、サービスからのステータス応答を期待します。

次の表に、新しいgRPCパラメータとその説明の詳細を示します。

gRPC parameters	Value	説明
GRPCHEalthCheck	はい/いいえ	gRPC ヘルスチェックプローブを有効または無効にします。
grpcStatusCode	符号なし整数 (0-65535)、デフォルト:12	最大 16 個の gRPC ステータスコードを設定します。アプライアンスはステータス応答でステータスコード 0 を検索します。0 を受信しなかった場合、16 個のコードのいずれかがサービスステータスと一致する場合、サービスは up に設定できます。
grpcServiceName	二重引用符で囲まれたサービス名。デフォルト = 「」(空の文字列)	特定のサービスの正常性をチェックします。

コマンドインターフェイスを使用して **HTTP/2** で **gRPC** ヘルスモニタを設定する

gRPC ヘルスチェックプローブを実行するには、ヘルスチェックサービスを有効にし、gRPC ステータスコードを設定し、gRPCヘルスチェックを実行する必要のあるgRPCサービス名を指定する必要があります。コマンドプロンプトで入力します:

```
add lb monitor <monitor_name> HTTP2 -httpRequest <string> -grpcHealthCheck
 ( YES | NO )- grpcStatusCode <positive_integer> - grpcServiceName
string>]
```

例:

```
add lb monitor http2 HTTP2 -httprequest "POST /grpc.health.v1.Health/
Check"- gRPCHealthCheck Yes -gRPCStatusCode 0 -grpcServiceName "ECHO"
```

GUI を使用して **HTTP/2** で **gRPC** ヘルスモニタを設定する

1. [トラフィック管理] **>** [負荷分散] **>** [モニター] に移動します。
2. [追加] をクリックします。
3. **[**モニターの作成] ページで、次のパラメーターを設定します。
    a) Name: gRPC ヘルスモニターの名前。
    b) タイプ。サービスタイプを HTTP/2 として選択します。
    c) gRPC ヘルスチェック。gRPC ヘルスチェックプローブを有効にします。

d) gRPC ステータスコード。gRPCサービスステータスが「UP」になるのは、gRPC ステータスコードが 0 または設定値である場合のみです。ステータスコードがゼロ以外の値、または設定されている値の場合、ステータスは「ダウン」になります。
e) gRPC サービス名。ヘルスチェックが実行されるサービス。

4. 作成 [作成]。

QUIC

August 15, 2023

Quick UDP Internet Protocol (QUIC) は、UDP に実装されている (TCP+TLS+HTTP/2) プロトコルの組み合わせである。QUIC トランスポートプロトコルは、UDP を使用して 2 つのエンドポイント間の接続を多重化します。また、他のプロトコルと比較すると、QUIC はセキュリティ、トラフィックの高速配信、および低遅延の点で高いパフォーマンスを提供します。

QUIC ブリッジは、QUIC クライアントと QUIC バックエンドサーバー間の QUIC トラフィックの負荷分散のために、NetScaler ADC アプライアンスで構成されます。QUIC ブリッジを使用すると、NAT リバインドまたは接続の移行がある場合に、クライアントとサーバ間で永続的な QUIC 接続を確立できます。ただし、この設定はデータを処理しません。これは、NetScaler ADC アプライアンスを介した QUIC トラフィックの負荷分散にのみ使用されます。

QUIC パケットには接続 ID が含まれており、エンドポイントがパケットを異なるアドレスまたは 4 タプルで同じ接続に関連付けることができます。接続 ID には、NetScaler ADC アプライアンスおよびバックエンドサーバーと共有されるサーバー ID の詳細が含まれます。NetScaler ADC アプライアンスは、サーバー ID の接続 ID の詳細を抽出し、トラフィックをバックエンドサーバーに送信します。接続 ID は保護されたパケットの中にあり、接続の移行時に接続を堅牢にします。

> 重要
>
> バックエンドサーバーは、QUIC 接続 ID でサーバー ID をエンコードするためのサポートが必要です。

QUIC ブリッジのメリット

NetScaler ADC アプライアンスの QUIC ブリッジは、次の理由で優先されます。

- 高価な暗号操作はありません。
- ステートレスルーティングが可能です（4 タプルベースのロードバランシングなし）。

QUIC の暗号オフロードサポート

NetScaler ADC アプライアンスに SSL ハードウェアチップが搭載されている場合、暗号化アクセラレーションを透過的に行い、QUIC トランザクションを高速化します。このアクセラレーションは、暗号処理をソフトウェアからハ

ードウェアにオフロードすることによって行われます。このサポートには明示的な設定は必要ありません。QUIC トランザクションのアクセラレーションは、Intel Coleto ハードウェアを備えた NetScaler ADC アプライアンスでサポートされています。

QUIC のブリッジ構成

January 11, 2024

QUIC ブリッジを設定するには、次の作業を完了する必要があります。

- QUIC ブリッジプロファイルの追加
- QUIC バックエンドサーバーの追加
- アプライアンスに QUIC サービスを追加する
- QUIC ブリッジタイプの負荷分散仮想サーバーの追加
- QUIC ブリッジタイプの QUIC ブリッジの負荷分散仮想サーバーにバインドする

> 重要
>
> QUIC ブリッジを設定する前に、まずアプライアンスで負荷分散機能を有効にしてください。詳細については、「基本的な負荷分散の設定」を参照してください。

CLI を使用した QUIC ブリッジの設定

次のセクションは、CLI を使用して設定する必要があります。

QUIC ブリッジプロファイルを追加する

QUIC ブリッジプロファイルを追加します。

コマンドプロンプトで入力します:

```
add quicBridge profile <name> -routingAlgorithm <PLAINTEXT> -
    serveridlen <value>
```

例:

```
add quicBridge profile q1 -routingAlgorithm PLAINTEXT -serveridlen 6
```

> 注
>
> この例で設定されるserveridlenパラメータは、IP および PORT の 16 進数の文字列であるカスタムサーバー ID の長さです。

QUIC バックエンドアプリケーションサーバーの追加

QUIC バックエンドアプリケーションサーバーを追加します。

コマンドプロンプトで入力します:

```
1 -  add server <name> (<IPAddress>)
2 -  add server <name> (<IPAddress>)
```

例:

```
1 -  add server s1 192.0.2.20
2 -  add server s2 192.0.2.30
```

QUIC ブリッジサービスを追加する

QUIC ブリッジサービスをアプリケーションサーバに追加する必要があります。

コマンドプロンプトで入力します:

```
1 -  add service  <name>  \(<IP> | <serverName>) <serviceType> <port> \[-
     CustomServerID <string>]
2
3 -  add service  <name>  \(<IP> | <serverName>) <serviceType> <port> \[-
     CustomServerID <string>]
```

例:

```
1 -  add service src1 s1 QUIC_BRIDGE 443 -CUSTOMSERVERID C0A8026401BB
2
3 -  add service src2 s2 QUIC_BRIDGE 443 -CUSTOMSERVERID C0A802C801BB
```

> 注
>
> 前の例で設定したCustomServerIDパラメータは、対応する IP の 16 進文字列とサーバーの PORT（s1 および s2）です。QUIC ブリッジ機能では、16 進文字列形式でのみCustomServerIDパラメータを構成することをお勧めします。

QUIC ブリッジタイプの負荷分散仮想サーバーを追加する

QUIC ブリッジタイプの負荷分散仮想サーバーを追加する必要があります。

コマンドプロンプトで入力します:

```
1 add lb vserver <name> [<IPAddress>@ <port>] [-persistenceType <
     persistenceType >] [-lbMethod < lbMethod >] [-rule <rule>] [-
     cltTimeout <secs>] [-quickBridgeProfilename <name>]
```

例:


```
add lb vserver quic_bridge_vip QUIC_BRIDGE 192.0.2.10 443 -
    persistenceType CUSTOMSERVERID -lbMethod TOKEN -rule QUIC.
    CONNECTIONID -cltTimeout 120 -quicBridgeProfilename q1
```

> 注
>
> QUIC Bridge 仮想サーバーを設定する際には、persistenceTypeパラメーターをCUSTOMSERVERIDとして、ruleパラメーターをQUIC.CONNECTIONIDとして、LbMethodパラメーターをTOKENとして 構成する必要があります。

QUIC ブリッジサービスをタイプ **QUIC** ブリッジの負荷分散仮想サーバーにバインドします

QUIC ブリッジサービスは、QUIC ブリッジタイプの負荷分散仮想サーバーにバインドする必要があります。

コマンドプロンプトで入力します:

```
-  bind lb vserver <name> (<serviceName>)

-  bind lb vserver <name> (<serviceName>)
```

例**:**

```
-  bind lb vserver quic_bridge_vip src1

-  bind lb vserver quic_bridge_vip src2
```

サービスグループの **QUIC** ブリッジの設定

サービスグループに QUIC ブリッジ機能を設定することもできます。次の手順では、サービスグループに QUIC ブリッジを設定する手順を示します。

サービスグループに QUIC ブリッジを設定するには、次の作業を完了する必要があります。

QUIC ブリッジプロファイルの追加

コマンドプロンプトで入力します:

```
add quicBridge profile <name> -routingAlgorithm <PLAINTEXT> -
    serveridlen <value>
```

例**:**

```
add quicBridge profile q1 -routingAlgorithm PLAINTEXT -serveridlen 6
```


QUIC タイプのサーバーを追加する

コマンドプロンプトで入力します:

```
1 -  add server <name> (<IPAddress>)
2 -  add server <name> (<IPAddress>)
```

例:

```
1 -  add server s1 192.0.2.20
2 -  add server s2 192.0.2.30
```

QUIC ブリッジサービスグループの追加

コマンドプロンプトで入力します:

```
1 add serviceGroup <serviceGroupName> \(<IP> | <serverName>) <serviceType
     >
```

例:

```
1 add serviceGroup svg1 QUIC_BRIDGE
```

QUIC サーバをサービスグループにバインドします

コマンドプロンプトで入力します:

```
1 -  bind serviceGroup <serviceGroupName> \(<IP>@ | \(<serverName>) \[-
     CustomServerID <string>]
2 -  bind serviceGroup <serviceGroupName> \(<IP>@ | \(<serverName>) \[-
     CustomServerID <string>]
```

例:

```
1 -  bind serviceGroup svg1 s1 443 -customServerID C0A8026401BB
2 -  bind serviceGroup svg1 s2 443 -customServerID C0A802C801BB
```

QUIC ブリッジタイプの負荷分散仮想サーバーの追加

コマンドプロンプトで入力します:

```
1 add lb vserver <name> [<IPAddress>@ <port> [-persistenceType <
     persistenceType >] [-lbMethod < lbMethod > [-cltTimeout <secs>] [-
     quickBridgeProfilename <name>]
```

例:


```
add lb vserver quic_bridge_vip QUIC_BRIDGE 192.0.2.10 443 -
    persistenceType CUSTOMSERVERID -lbMethod TOKEN -cltTimeout 120 -
    quicBridgeProfilename q1
```

QUIC ブリッジタイプの負荷分散仮想サーバーをサービスグループにバインドします

コマンドプロンプトで入力します:

```
bind lb vserver <name>@ (<serviceName>@ <serviceGroupName>
```

例**:**

```
bind lb vserver quic_bridge_vip svg1
```

GUI を使用した **QUIC** ブリッジの設定

GUI を使用して QUIC ブリッジを設定するには、次の手順を実行します。

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動します。
2. [仮想サーバー] ページで、[追加] をクリックします。
3. [負荷分散仮想サーバー] ページで、[プロトコル] を QUIC_BRIDGE として選択し、詳細を入力します。
4. [続行] と [完了] をクリックします。

GUI を使用してサービスの負荷分散を構成します

GUI を使用してサービスの負荷分散を設定するには、次の手順を実行します。

1. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。[サービス] ページで、[追加] をクリックします。
2. [負荷分散サービス] ページで、詳細を入力し、[**OK**] をクリックします。
3. [仮想サーバー] ページで、サービスをバインドするために作成された仮想サーバーを選択します。
4. [負荷分散仮想サーバー] ページを下にスクロールして、[サービスとサービスグループ] を選択します。
5. [サービスバインド] 画面で、[サービスフィールドの選択] をクリックします。
6. [サービス] 画面で、負荷分散仮想サーバーにバインドするサービスを選択し、[選択] をクリックします。
7. 「サービスバインディング」画面で、作成したサービスを選択して「バインド」をクリックします。
8. [負荷分散仮想サーバー] ページで、[完了] をクリックします。

QUIC ブリッジの統計情報の表示

QUIC ブリッジは、QUIC ブリッジ統計情報の詳細なサマリーを表示する statistics コマンドをサポートしています。

次のコマンドは、QUIC ブリッジ統計情報の詳細なサマリーを表示します。コマンドプロンプトで、次のように入力します:

- `stat quicbridge`
- `stat quicbridge -detail`

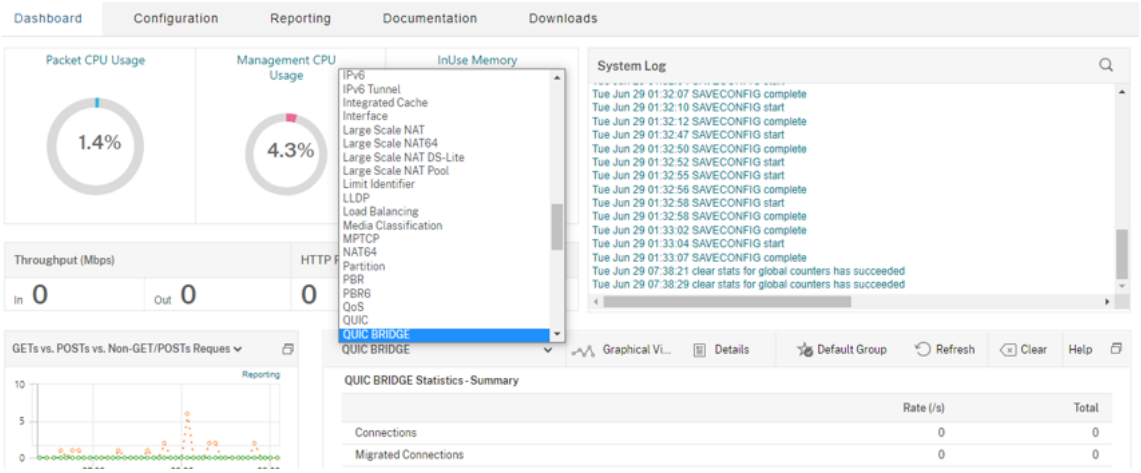
統計情報の表示をクリアするには、次のいずれかを入力します。

- `stat quicbridge -clearstats basic`
- `stat quicbridge -clearstats full`

GUI を使用して **QUIC** ブリッジ統計情報を表示する

QUIC ブリッジ統計情報を表示するには、次の手順を実行します。

1. [ダッシュボード] タブで、[システムの概要] セクションにマウスを移動します。
2. [システム概要] をクリックし、ドロップダウンリストから [QUIC BRIDGE] を選択します。



プロキシプロトコル

January 11, 2024

プロキシプロトコルは、NetScaler アプライアンスを介してクライアントからサーバーにクライアントの詳細を安全に転送します。アプライアンスは、クライアントの詳細を含むプロキシプロトコルヘッダーを追加し、バックエンドサーバーに転送します。以下は、NetScaler アプライアンスでのプロキシプロトコルの使用シナリオの一部です。

- 元のクライアント IP アドレスの学習
- Web サイトの言語を選択する
- 選択した IP アドレスの一覧表示をブロックする
- 統計のロギングと収集。

以下に、3 つの動作モードを示します。

- [挿入]。アプライアンスはクライアントの詳細を挿入し、バックエンドサーバーに送信します。
- フォワード。アプライアンスは、クライアントの詳細をバックエンドサーバに転送します。
- 剥ぎ取られた。アプライアンスは、ログ用にクライアントの詳細を保存します。また、プロキシプロトコルがバックエンドサーバーでサポートされていない場合は、書き換えポリシー設定を使用してクライアントの詳細をサーバーに送信します

次の表は、さまざまなプロキシプロトコルモードでの LB 仮想サーバーとサービスのステータスに関する情報を示しています。

プロキシプロトコルモード	LB 仮想サーバー	サービス
Ins	無効	有効
進	有効	有効
Stripped	有効	無効

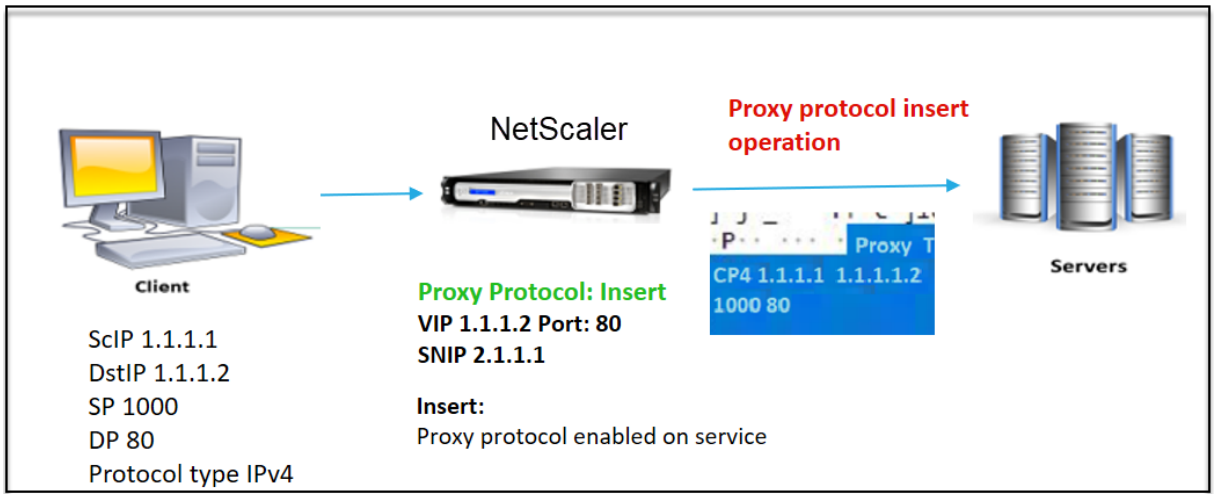
制限事項

プロキシプロトコルは、TCP Fast Open (TFO) およびマルチパス TCP 機能ではサポートされていません。この機能は、NetScaler アプライアンスが TCP 接続終了を行うサービスでのみサポートされます。他のサービス（「任意」など）はサポートされていません。

NetScaler アプライアンスでのプロキシプロトコルのしくみ

次のフロー図は、挿入、転送、および削除操作のために NetScaler アプライアンス間でプロキシプロトコルを構成する方法を示しています。

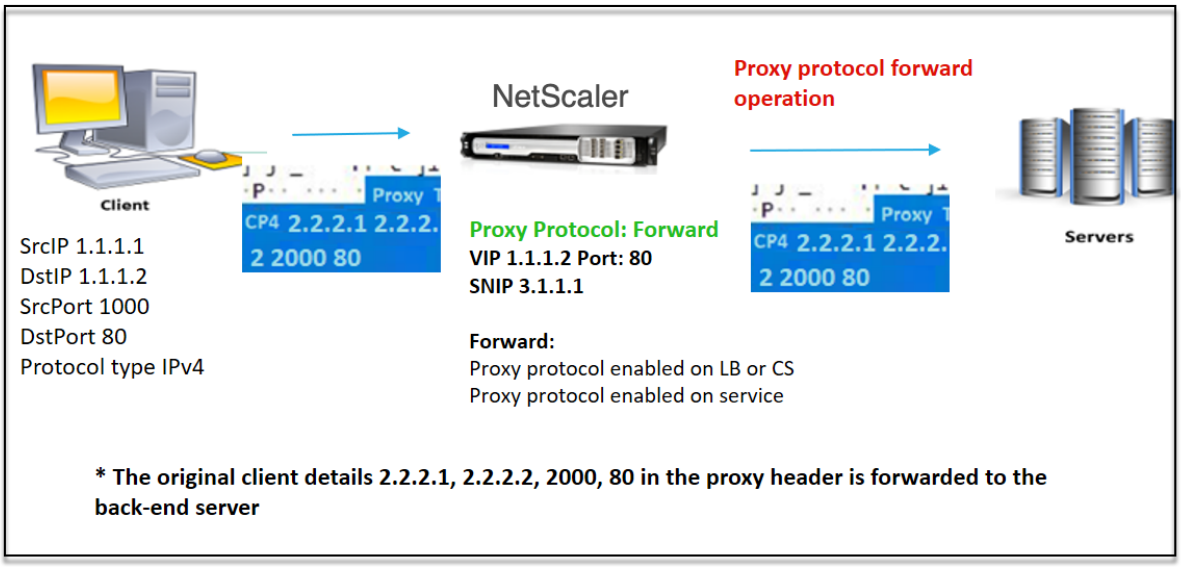
挿入操作



コンポーネントの相互作用は次のとおりです。

- NetScaler インスタンスでは、ネットプロファイルでプロキシプロトコルを有効にし、サービスにバインドする必要があります。
- 挿入操作では、NetScaler はクライアント接続の詳細を含むプロキシヘッダーを追加し、バックエンドサーバーに転送します。
- 送信側では、アプライアンスは CLI 設定に基づいてプロキシプロトコルのバージョンを決定します。

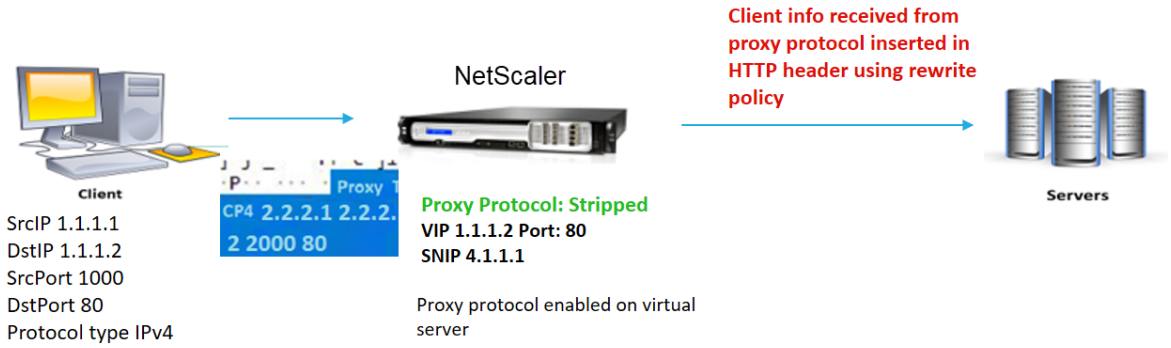
フォワード操作



コンポーネントの相互作用は次のとおりです。

- クライアントは、プロキシヘッダーとともにリクエストを NetScaler に送信します。アプライアンスはバージョンを動的に識別します。
- NetScaler アプライアンスでは、転送操作です。プロキシプロトコルは、負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーで有効になり、サービスでは有効になります。アプライアンスはプロキシヘッダーを受信し、ヘッダーの詳細をバックエンドサーバーに転送します。
- プロキシヘッダーの詳細が無効な形式の場合、アプライアンスは接続をリセットします。
- 送信側では、アプライアンスは CLI 設定に基づいてプロキシプロトコルのバージョンを決定します。

操作を剥ぎ取った



コンポーネントの相互作用は次のとおりです。

- クライアントは、プロキシヘッダーとともに要求を NetScaler アプライアンスに送信します。
- NetScaler アプライアンスでは、ストリップ操作の場合、アプライアンスはプロキシプロトコルから取得したクライアント情報を転送し、書き換えポリシー式を使用して HTTP ヘッダーに挿入します。
- 送信元 IP アドレス、宛先 IP アドレス、送信元ポート、宛先ポートなどのクライアントの詳細は、書き換えポリシー式を使用して HTTP ヘッダーに追加されます。書き換えポリシーによって式が評価され、「true」の場合、対応する書き換えポリシーアクションがトリガーされます。クライアントの詳細は HTTP ヘッダーでバックエンドサーバーに転送されます。
- プロキシヘッダーの詳細が無効な形式の場合、アプライアンスは接続をリセットします。

プロキシプロトコルのバージョン形式

プロキシプロトコルバージョンには 2 つの形式があります。アプライアンスは、着信データの長さに基づいてフォーマットを使用することを決定します。詳細については、プロキシプロトコル RFP を参照してください。

1. プロキシプロトコルバージョン 1 形式

   ```
   PROXY TCP4/TCP6/UNKNOWN <SRC IP> <DST IP> <SRC PORT> <DST PORT>
   ```

   - PROXY-> プロキシヘッダーバージョン -1 の一意の文字列形式。

- プロトコル TCP over IPv4 および TCP over IPv6 をサポートします。残りのプロトコルでは、これは UNKNOWN です。
- SRC IP：パケットの送信元 IP（元のクライアント IP）アドレス。
- DST IP：パケットの宛先 IP アドレス。
- SRC ポート：パケットの送信元ポート。
- DST ポート：パケットの宛先ポート。

2. プロキシプロトコルバージョン 2 形式

```
0D 0A 0D 0A 00 0D 0A 51 55 49 54 0A <13th byte> <14th byte>
<15-16th byte> <17th byte onwards>
```

- D 0A 0D 0A 00 0D 0A 51 55 49 54 0A-> プロキシヘッダバージョン-2 の一意のバイナリ文字列。
- プロトコル TCP over IPv4 および TCP over IPv6 をサポートします。残りのプロトコルでは、これは UNKNOWN です。
- 13 バイト：プロトコルのバージョンとコマンド。
- 14 バイト：アドレスとプロトコルファミリ。
- 15-16 バイト—ネットワークオーダーのアドレス長。
- 17 バイト目以降：ネットワーク順に存在するアドレス情報（src IP、DST IP、src ポート、dst ポート）。

レスポンダーポリシーインフラストラクチャ表現サポート

プロキシプロトコルは、TCP および HTTP タイプの仮想サーバーの次のレスポンダーポリシーインフラストラクチャ表現をサポートします。

1. CLIENT.PROXY.SRCIP_STR
2. CLIENT.PROXY.DSTIP_STR
3. CLIENT.PROXY.SRCPORT
4. CLIENT.PROXY.DSTPORT
5. CLIENT.PROXY.ETHERTYPE

> 注
>
> NetScaler は、NetScaler リリース 13.1-48.x 以降の TCP タイプの仮想サーバー上のプロキシプロトコルのレスポンダーポリシーインフラストラクチャ表現をサポートしています。

NetScaler アプライアンスでプロキシプロトコルを構成する

NetScaler アプライアンスでプロキシプロトコルを構成するには、次の手順を実行します。

1. プロキシプロトコルをグローバルとして有効にします。
2. 挿入操作のプロキシプロトコルを設定します。
3. 転送操作用のプロキシプロトコルを設定します。

4. ストリップ操作のプロキシプロトコルを設定します。

プロキシプロトコルをグローバルとして有効にする

コマンドプロンプトで、次のように入力します:

```
set ns param -proxyProtocol ENABLED
```

挿入操作用のプロキシプロトコルの設定

挿入操作のプロキシプロトコルを構成するには、負荷分散仮想サーバーでプロトコルを無効にし、サービスのプロトコルを有効にする必要があります。

負荷分散仮想サーバーのプロキシプロトコルを無効にしたネットプロファイルを追加する　コマンドプロンプトで、次のように入力します:

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

例:

```
Add netprofile proxyprofile-1 -proxyProtocol DISABLED -proxyprotocoltxversion
 V1
```

> 注:
>
> アプライアンスでプロキシプロトコルを無効にする場合は、プロトコルバージョンパラメータを設定する必要はありません。

サービスに対してプロキシプロトコルが有効になっているネットプロファイルを追加する　コマンドプロンプトで、次のように入力します:

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

例:

```
add netprofile proxyprofile-2 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

プロキシレイヤーに **NetScaler** アプライアンスの負荷分散仮想サーバーを追加します　コマンドプロンプトで、次のように入力します:

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```


例:

```
add lb vserver lbvserver-1 http 1.1.1.1 80
```

プロキシレイヤーに **NetScaler** アプライアンスの **HTTP** サービスを追加します　コマンドプロンプトで、次のように入力します:

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```

例:

```
Add service http-service-1 2.2.2.1 http 80
```

NetScaler アプライアンスで負荷分散仮想サーバーを使用してネットプロファイルを設定する　コマンドプロンプトで、次のように入力します:

```
set lb vserver <vserver name> -netprofile <name>
```

例:

```
set lb vserver lbvserver-1 -netprofile proxyProfile-1
```

NetScaler アプライアンスで **HTTP** サービスを使用してネットプロファイルを設定する　コマンドプロンプトで、次のように入力します:

```
set service <service name> -netprofile <name>
```

例:

```
set service http-service-1 -netprofile proxyProfile-2
```

負荷分散仮想サーバーをサービスにバインドする　コマンドプロンプトで、次のように入力します:

```
bind lb vserver <vserver name> <service name>
```

例:

```
bind lb vserver lbvserver-1 http-service-1
```

転送操作用のプロキシプロトコルの設定

プロキシレイヤーの次の NetScaler インスタンスの転送操作用のプロキシプロトコルを構成するには、プロトコルを有効にして仮想サーバーまたはサービスにバインドする必要があります。

> 注:
>
> 負荷分散仮想サーバー用に作成されたネットプロファイルは、サービスにも使用できます。

負荷分散仮想サーバーでプロキシプロトコルが有効になっているネットプロファイルを追加する　コマンドプロンプトで、次のように入力します:

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

例:

```
add netprofile proxyprofile-3 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

サービスに対してプロキシプロトコルが有効になっているネットプロファイルを追加する　コマンドプロンプトで、次のように入力します:

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

例:

```
add netprofile proxyprofile-4 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

プロキシレイヤーに **NetScaler** アプライアンスの負荷分散仮想サーバーを追加します　コマンドプロンプトで、次のように入力します:

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

例:

```
add lb vserver lbvserver-2 http 2.2.2.2 80
```

プロキシレイヤーに **NetScaler** アプライアンスの **HTTP** サービスを追加します　コマンドプロンプトで、次のように入力します:

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```

例:

```
Add service http-service-2 3.3.3.1 http 80
```

NetScaler アプライアンスで負荷分散仮想サーバーを使用してネットプロファイルを設定する　コマンドプロンプトで、次のように入力します:

```
set lb vserver <vserver name> -netprofile <name>
```

例:

```
set lb vserver lbvserver-2 -netprofile proxyProfile-3
```


NetScaler アプライアンスで **HTTP** サービスを使用してネットプロファイルを設定する　コマンドプロンプトで、次のように入力します:

```
set service <service name> -netprofile <name>
```

例:

```
set service http-service-2 -netprofile proxyProfile-4
```

負荷分散仮想サーバーをサービスにバインドする　コマンドプロンプトで、次のように入力します:

```
bind lb vserver <vserver name> <service name>
```

例:

```
bind lb vserver lbvserver-2 http-service-2
```

ストリップ操作用のプロキシプロトコルの設定

ストリップ操作のプロキシプロトコルを構成するには、負荷分散仮想サーバーでプロキシプロトコルを有効にし、サービスのプロキシプロトコルを無効にする必要があります。

仮想サーバーでプロキシプロトコルが有効になっているネットプロファイルを追加する　コマンドプロンプトで、次のように入力します:

```
add netprofile <name> -proxyProtocol ENABLED> -proxyprotocoltxversion
 <V1/V2>
```

例:

```
add netprofile proxyprofile-5 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

NetScaler アプライアンスの負荷分散またはコンテンツスイッチング仮想サーバーをプロキシレイヤーに追加するコマンドプロンプトで、次のように入力します:

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

例:

```
add lb vserver lbvserver-3 http 2.2.2.2 80
```


プロキシレイヤーに **NetScaler** アプライアンスの **HTTP** サービスを追加します　コマンドプロンプトで、次のように入力します:

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```

例:

```
Add service http-service-3 3.3.3.1 http 80
```

NetScaler アプライアンスの負荷分散またはコンテンツスイッチング仮想サーバーを使用したネットプロファイルの設定　コマンドプロンプトで、次のように入力します:

```
set lb vserver <vserver name> -netprofile <name>
```

例:

```
set lb vserver lbvserver-3 -netprofile proxyProfile-5
```

負荷分散仮想サーバーをサービスにバインドする　コマンドプロンプトで、次のように入力します:

```
bind lb vserver <vserver name> <service name>
```

例:

```
bind lb vserver lbvserver-3 http-service-3
```

CLI を使用してプロキシプロトコルのレスポンダーポリシーインフラストラクチャ表現を設定します

レスポンダーポリシーを構成するには、コマンドプロンプトで次のように入力します。

```
add responder policy <name> <expression> <action>
```

例:

```
1  > add responder policy resppol_proxy_srcip "CLIENT.PROXY.SRCIP_STR.EQ("
       10.106.26.83")" RESET
2  Done
3  <!--NeedCopy-->
```

レスポンダーポリシーを負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
bind lb vserver <name> -policyname <string> -priority <positive_integer
> -gotoPriorityExpression <expression> -type <type>
```

例:

```
1  > bind lb vserver lb_tcp1 -policyName resppol_proxy_srcip -priority 10
       -gotoPriorityExpression END -type REQUEST
2  Done
3  <!--NeedCopy-->
```


エンドツーエンド構成の例

```
> add ns tcpProfile tcp-proxy-profile -tcpmode ENDPOINT

> add netprofile net_proxyv1 -MBF DISABLED -proxyProtocol
ENABLED

> enable ns mode l2

> enable ns mode l3 usnip

> add ns ip 10.106.26.146 255.255.255.0 -type SNIP
Done
> add ns ip 10.106.26.144 255.255.255.0 -type SNIP
Done

> add lb vserver lb_tcp1 TCP 10.106.26.141 80
> add service s1 10.106.26.82 TCP 8080

> bind lb vserver lb_tcp1 s1

> set lb vserver lb_tcp1 -tcpProfileName tcp_proxy -netProfile
   net_proxyv1

> set ns param -proxyProtocol ENABLED

> add responder policy resppol_proxy_srcip "CLIENT.PROXY.SRCIP_STR.EQ("
   10.106.26.83")" RESET

> bind lb vserver lb_tcp1 -policyName resppol_proxy_srcip -priority 10
   -gotoPriorityExpression END -type REQUEST
Done
<!--NeedCopy-->
```

NetScaler GUI を使用してプロキシプロトコルを構成する

1. [システム] > [設定**] > [**グローバルシステム設定の変更] に移動します。
2. [グローバルシステム設定の構成**]** パラメータページで、[プロキシプロトコル] チェックボックスをオンにします。
3. 「**OK**」をクリックして「閉じる」をクリックします。
4. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
5. 詳細ウィンドウで、[追加] をクリックして、負荷分散仮想サーバーのネットプロファイルを作成します。
6. [ネットプロファイル] ページで、次のパラメータを設定します。

    a) 名前: ネットプロファイルの名前。
    b) プロキシプロトコル: 負荷分散仮想サーバーのプロキシプロトコルを有効または無効にします。
    c) プロキシプロトコル **TX** バージョン: 受信データ形式に基づいて、プロキシプロトコルバージョンを V1 または V2 に設定します。

7. [**OK**] をクリックします。
8. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
9. 詳細ペインで、[追加] をクリックします。
10. [負荷分散仮想サーバー] ページで、基本パラメータを設定します。
11. **[**詳細設定**]** セクションで、[プロファイル] を選択します。
12. [プロファイル] セクションで、鉛筆アイコンをクリックします。
13. ネットプロファイルを選択し、「**OK**」をクリックします。
14. [完了] をクリックします。
15. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
16. 詳細ペインで、[追加] をクリックします。
17. [負荷分散サービス] ページで、基本パラメータを設定します。
18. **[**詳細設定**]** セクションで、[プロファイル] を選択します。
19. [プロファイル] セクションで、鉛筆アイコンをクリックします。
20. ネットプロファイルを選択し、「**OK**」をクリックします。
21. [完了] をクリックします。

> 注:
>
> プロキシレイヤーの一部として複数の NetScaler アプライアンスがある場合は、転送操作のために各アプライアンスでプロキシプロトコル構成を設定する必要があります。

TCP オプションのクライアント IP アドレス

August 15, 2023

NetScaler ADC アプライアンスは、さまざまな方法でクライアント情報をバックエンドサーバーに送信します。そのような方法の 1 つに、TCP オプションでクライアント IP アドレスを送信する方法があります。バックエンドサーバが TCP オプションを使用してクライアント IP アドレスを読み取る場合、アプライアンスは TCP プロファイルの TCP オプション番号を使用します。

NetScaler アプライアンスは、次のパケットでのみクライアント IP アドレスを TCP オプションヘッダーで送信します。

- スリーウェイハンドシェイクの最終 ACK パケット
- 最初のデータパケット。

NetScaler アプライアンスの TCP オプション構成の使用シナリオの一部を以下に示します。

- 元のクライアント IP アドレスの学習
- Web サイトの言語を選択する
- 選択した IP アドレスの一覧表示をブロックする

TCP オプションでクライアント IP アドレスを送信する場合、次の 2 つの動作モードがあります。

- **[挿入]**。挿入モードでは、アプライアンスはクライアントの詳細を TCP オプション 28（設定可能ですが、推奨値は 28）フィールドに追加し、バックエンドサーバに送信します。
- フォワード。転送モードでは、仮想サーバは TCP オプションのクライアント IP 詳細をプロキシデバイスから受信します。仮想サーバでは、プロキシデバイスがクライアント IP の詳細を送信するために使用したのと同じ TCP オプションを設定する必要があります。

  アプライアンスは、TCP オプションフィールドのクライアント詳細をバックエンドサーバに送信します。バックエンドサーバーを表すサービスには、任意の TCP オプションを設定できますが、推奨値は 28 です。

NetScaler アプライアンスは、挿入モード構成用の TCP オプションでのクライアントポートの送信もサポートしています。

> 注:
>
> - バインドされた TCP プロファイルで [クライアント IP TCP] オプションが有効になっている場合、仮想サーバで受信したトラフィックの多重化はサポートされません。
> - TCP または HTTP 仮想サーバの場合、TCP オプション番号は、この機能がトランスペアレントモードで有効かどうかにかかわらず転送されます。

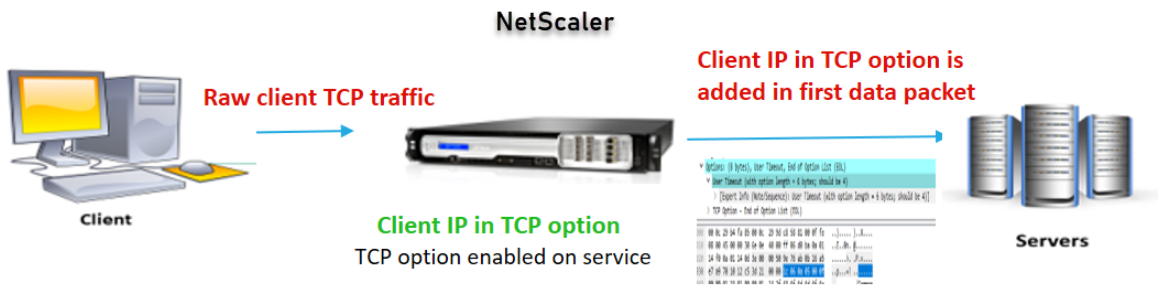
制限事項

TCP オプション設定機能は、TFO、マルチパス TCP、および HTTP2 機能ではサポートされていません。

NetScaler ADC アプライアンスでの TCP オプションの構成方法

次のフロー図は、NetScaler ADC アプライアンスで挿入および転送操作用に TCP オプションを構成する方法を示しています。

挿入操作**:**

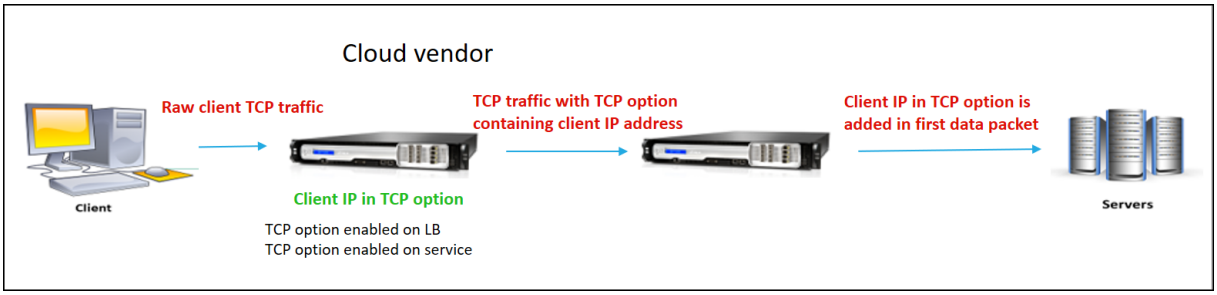


コンポーネントの相互作用は次のとおりです。

- クライアントが NetScaler ADC にリクエストを送信します。
- 挿入操作では、NetScaler アプライアンスは、次のパケットで構成された TCP オプションにクライアントの IP アドレスとポートをバックエンドサーバーに挿入します。
  - スリーウェイハンドシェイクの最終 ACK パケット
  - 第 1 データパケット

> 注:
>
> 着信トラフィックが HTTPS の場合、TCP オプションのクライアント IP アドレスとクライアントポートは SSL クライアント hello メッセージで送信されます。これは TCP レベルの最初のデータパケットです。

フォワードオペレーション:



コンポーネントの相互作用は次のとおりです。

- クライアントは、HTTP/HTTPS 要求を NetScaler ADC アプライアンスに送信します。
- Forward 操作では、負荷分散仮想サーバーまたはコンテンツスイッチ仮想サーバーで TCP オプションが有効になり、サービスでも有効になります。アプライアンスは、仮想サーバで指定された TCP オプション番号でクライアントの詳細を受信します。
- 次に、NetScaler アプライアンスは、バックエンドサーバーへの次のパケットの構成済み TCP オプション（サービス用）にクライアントの IP アドレスとポートを挿入します。
  - スリーウェイハンドシェイクの最終 ACK パケット
  - 第 1 データパケット

挿入操作の **TCP** オプションの設定

挿入操作の TCP オプションの設定は、次の手順で構成されます。

1. TCP プロファイルを設定します。Client IP TCP オプション（`clientIpTcpOption`）を有効にし、TCP オプション番号（`clientIpTcpOptionNumber`）を指定します。オプションで、`sendClientPortInTcpOption`が TCP オプションヘッダー内のクライアントポートを送信できるようにします。

> 注:
>
> TCP プロファイルでは、TCP オプション番号を 28 に設定することをお勧めします。

2. TCP プロファイルをサービスにバインドする

CLI を使用して **TCP** プロファイルを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- `add tcpprofile <name> -clientIpTcpOption (ENABLED | DISABLED)-clientIpTcpOptionNumber <positive_integer> -sendClientPortInTcpOption (ENABLED | DISABLED)`
- `show tcpprofile <name>`

CLI を使用して **TCP** プロファイルをサービスにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- `set service <name> -tcpprofileName <name>`
- `show service <name>`

設定例

```
1 add tcpprofile TCP-PROFILE-1 -clientIpTcpOption ENABLED -
     clientIpTcpOptionNumber 28 -sendClientPortInTcpOption ENABLED
2 set service SERVICE-1  - tcpprofileName TCP-PROFILE-1
3 <!--NeedCopy-->
```

転送操作の **TCP** オプションを設定する

転送動作の TCP オプションの設定は、次の手順で構成されます。

1. TCP プロファイルを設定します。Client IP TCP オプション（`clientIpTcpOption`）を有効にし、TCP オプション番号（`clientIpTcpOptionNumber`）を指定します。
2. TCP プロファイルを負荷分散仮想サーバーまたはコンテンツスイッチ仮想サーバーにバインドする
3. TCP プロファイルをサービスにバインドします。

CLI を使用して **TCP** プロファイルを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- `add tcpprofile <name> -clientIpTcpOption (ENABLED | DISABLED)-clientIpTcpOptionNumber <positive_integer>`
- `show tcpprofile <name>`

CLI を使用して **TCP** プロファイルを負荷分散仮想サーバーまたはコンテンツスイッチ仮想サーバーにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- `set lb vserver <name> -tcpprofileName <name>`
- `show lb vserver <name>`

CLI を使用して **TCP** プロファイルをサービスにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- `set service <name> -tcpprofileName p1`
- `show service <name>`

設定例

```
add tcpprofile TCP-PROFILE-2 -clientIpTcpOption ENABLED -
    clientIpTcpOptionNumber 29
set lb vserver LBVS-2  - tcpprofileName TCP-PROFILE-2
set service SERVICE-2 -tcpprofileName TCP-PROFILE-2
<!--NeedCopy-->
```

NetScaler GUI を使用して **TCP** オプションを構成する

1. **System** > **Profiles** に移動します。
2. [**TCP** プロファイル] タブページで、[追加] をクリックします。
3. **[TCP** プロファイルの設定] ページで、次のパラメータを設定します。
   - **clientpTCPOption**。TCP オプションを有効にして、クライアント IP アドレスを送受信します。
   - **clientiptcptionnumber**。TCP オプション番号を設定します。
   - **SendClientPortIntcpOption** 挿入モードの設定用に、クライアントポートを TCP オプションで送信します。
4. 「**OK**」をクリックして「閉じる」をクリックします。

SNMP

August 15, 2023

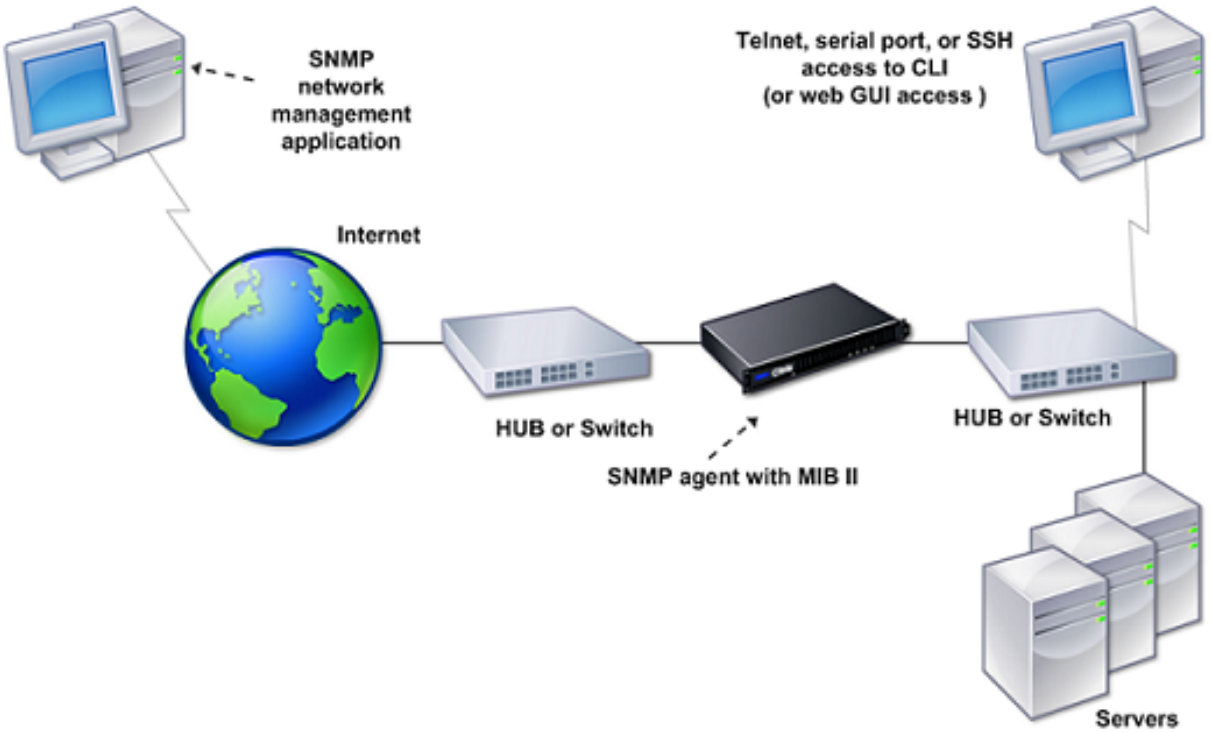
簡易ネットワーク管理プロトコル（*SNMP*）を使用して、トラップと呼ばれる非同期イベントを生成するように *NetScaler* アプライアンスの *SNMP* エージェントを構成できます。トラップは、NetScaler に異常な状況が発生す

るたびに生成されます。その後、トラップはトラップリスナーと呼ばれるリモートデバイスに送信され、NetScaler アプライアンスの異常状態を知らせます。または、*SNMP* マネージャーと呼ばれるリモートデバイスから、*SNMP* エージェントにシステム固有の情報を問い合わせることができます。次に、エージェントは管理情報ベース（MIB）で要求されたデータを検索し、そのデータを SNMP マネージャに送信します。

NetScaler ADC 上の SNMP エージェントは、SNMPv1、SNMPv2、SNMPv3 に準拠したトラップを生成できます。クエリでは、SNMP エージェントは SNMP バージョン 1（SNMPv1）、SNMP バージョン 2（SNMPv2）、および SNMP バージョン 3（SNMPv3）をサポートします。

SNMP パラメータ、トラップ、およびその説明については、NetScaler SNMP OID リファレンスを参照してください。

次の図は、SNMP が有効で構成されている NetScaler ADC を使用するネットワークを示しています。この図では、各 SNMP ネットワーク管理アプリケーションが SNMP を使用して NetScaler 上の SNMP エージェントと通信しています。SNMP エージェントは、管理情報ベース（MIB）を検索して、SNMP マネージャによって要求されたデータを収集し、その情報をアプリケーションに提供します。



> 重要
>
> NetScaler ADC アプライアンスの SNMP モジュールは、SNMP OID の最大 128 バイト（RFC 3416 に準拠している）をサポートします。オブジェクトの長いインデックス変数名を使用すると、SNMP OID の長さが 128 バイトを超える場合があります。
>
> この問題を解決するために、NetScaler SNMP モジュールは、インデックス変数名に対して最大 31 文字をサポートします。インデックス変数名の長さが 31 文字を超える場合、ハッシュアルゴリズムを使用する SNMP モジュールは、名前を 31 文字のハッシュ値に変換します。このハッシュ値は、その変数の SNMP OID で使用

> されます。
>
> 元のインデックス変数名は、次の名前の形式を持つ別の変数に格納されます。<variable type>FullName。たとえば、負荷分散仮想サーバーの名前が 31 文字を超える場合、vserverName SNMP OID にはハッシュ値が含まれ、vsvrFullName SNMP OID には仮想サーバーの完全な（元の）名前が含まれます。
>
> 同様に、SNMP トラップの場合、インデックス変数にはハッシュ値が表示されます。<variable type>FullNameは、元のインデックス変数名のフルネームを格納し、トラップメッセージの一部でもあります。

SNMP マネージャおよびトラップリスナーへの MIB ファイルのインポート

NetScaler ADC アプライアンスを監視するには、MIB オブジェクト定義ファイルをダウンロードする必要があります。NetScaler アプライアンスは、次のエンタープライズ固有の MIB をサポートしています。

- 標準 **MIB-2** グループのサブセット。MIB-2 グループの SYSTEM、IF、ICMP、UDP、および SNMP を提供します。
- システムエンタープライズ **MIB**。システム固有の設定と統計情報を提供します。

MIB オブジェクト定義ファイルは、/netscaler/snmp ディレクトリまたは GUI の [ダウンロード] タブから取得できます。

SNMP トラップを生成するように NetScaler ADC を構成する

August 15, 2023

NetScaler ADC アプライアンスは、トラップと呼ばれる非同期イベントを生成するように構成できます。トラップは、アプライアンスに異常な状態が発生するたびに生成されます。トラップは、トラップ リスナーと呼ばれるリモートデバイスに送信されます。管理者は、アプライアンスを監視し、問題に対して迅速に対応できます。

NetScaler ADC アプライアンスは、*SNMP* アラームと呼ばれる一連の条件エンティティを提供します。いずれかの SNMP アラームの条件が満たされると、アプライアンスは設定されたトラップリスナーに送信される SNMP トラップメッセージを生成します。たとえば、LOGIN-FAILURE アラームを有効にすると、アプライアンスでログインに失敗するたびにトラップメッセージが生成され、トラップリスナーに送信されます。

トラップを生成するように NetScaler ADC アプライアンスを設定するには、アラームを有効にして設定する必要があります。次に、アプライアンスが生成したトラップメッセージを送信するトラップリスナーを指定します。

SNMP アラームの有効化

NetScaler ADC アプライアンスは、有効になっている SNMP アラームに対してのみトラップを生成します。一部のアラームはデフォルトで有効になっていますが、無効にすることもできます。

SNMP アラームを有効にすると、何らかのイベントが発生すると、アプライアンスは対応するトラップメッセージを生成します。一部のアラームは、デフォルトで有効になっています。

CLI を使用して SNMP アラームを有効にするには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `enable snmp alarm <trapName>`
- `show snmp alarm <trapName>`

GUI を使用して SNMP アラームを有効にするには

1. [システム] > [SNMP] > [アラーム] に移動し、アラームを選択します。
2. [アクション] をクリックし、[有効] を選択します。

アラームの設定

NetScaler ADC アプライアンスは、*SNMP* アラームと呼ばれる一連の条件エンティティを提供します。SNMP アラームに設定された条件が満たされると、アプライアンスは設定されたトラップリスナーに送信される SNMP トラップメッセージを生成します。たとえば、LOGIN-FAILURE アラームを有効にすると、アプライアンスでログインに失敗するたびにトラップメッセージが生成され、トラップリスナーに送信されます。

SNMP アラームには重大度レベルを割り当てることができます。これを行うと、対応するトラップメッセージにその重大度が割り当てられます。

以下は、アプライアンス上で定義されている重要度レベルで、重大度の高い順に示されています。

- 重大
- 重要
- 軽度
- 警告
- 情報

たとえば、LOGIN-FAILURE という名前の SNMP アラームの警告重大度を設定すると、ログイン障害があるときに生成されるトラップメッセージに警告重大度が割り当てられます。

> 注
>
> NetScaler ADC は、さまざまな SNMP アラームをサポートしています。詳細については、「SNMP アラーム」を参照してください。

SNMP アラームを設定して、そのアラームの条件が満たされたときに生成される対応するトラップメッセージをログに記録することもできます。

CLI を使用して **SNMP** アラームを設定するには

コマンドプロンプトで次のコマンドを入力して、SNMP アラームを設定し、設定を確認します。

- `set snmp alarm <trapName> [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-time <secs>] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`
- `show snmp alarm <trapName>`

各項目の意味は次のとおりです。

しきい値: 上限しきい値の値。NetScaler ADC アプライアンスは、アラームに関連する属性の値が指定された高しきい値以上になると、SNMP トラップメッセージを生成します。

標準値: 標準しきい値の値。各属性の値が高しきい値を超えた後にこの値以下になると、トラップメッセージが生成されます。

GUI を使用して **SNMP** アラームを設定するには

[システム] **> [SNMP] > [**アラーム] に移動し、アラームを選択して、アラームパラメータを設定します。

SNMPv1 トラップまたは **SNMPv2** トラップの設定

アラームを設定したら、アプライアンスがトラップメッセージを送信するトラップリスナーを指定する必要があります。IP アドレスや IPv6 アドレスやトラップリスナーの宛先ポートなどのパラメータを指定する以外に、トラップのタイプ (汎用または固有) と SNMP バージョンを指定できます。

汎用または専用のトラップを受信するために、最大 20 のトラップリスナーを構成できます。

NetScaler IP（NSIP または NSIP6）アドレス以外の送信元 IP アドレスで SNMP トラップメッセージを特定のトラップリスナーに送信するようにアプライアンスを構成することもできます。IPv4 アドレスを持つトラップリスナーの場合、送信元 IP を、アプライアンス上で構成されたマッピング IP（MIP）アドレスまたはサブネット IP（SNIP）アドレスのいずれかに設定できます。IPv6 アドレスを持つトラップリスナーの場合、アプライアンスに構成されたサブネット IPv6（SNIP6）アドレスにソース IP を設定できます。

また、重大度に基づいてトラップリスナーにトラップメッセージを送信するようにアプライアンスを設定することもできます。たとえば、トラップリスナーに対して重大度を Minor に設定すると、重大度レベルが Minor（Minor、Major、Critical）以上のすべてのトラップメッセージがトラップリスナーに送信されます。

トラップリスナーにコミュニティストリングを定義した場合は、リスナーに送信するトラップごとにコミュニティストリングも指定する必要があります。コミュニティストリングが定義されているトラップリスナーは、トラップリスナーで定義されているコミュニティストリングと一致するコミュニティストリングを含むトラップメッセージのみを受け入れます。その他のトラップメッセージはドロップされます。

CLI を使用して **SNMP** トラップを追加するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp trap <trapClass> <trapDestination> -version ( V1 | V2 )-destPort <port> -communityName <string> -srcIP <ip_addr> -severity <severity>`
- `show snmp trap`

例:

```
> `add snmp trap specific 192.0.2.10 -version V2 -destPort 162 -communityName com1 -severity Major`
<!--NeedCopy-->
```

GUI を使用して **SNMP** トラップを設定するには

[システム] **> [SNMP] > [**トラップ] に移動し、SNMP トラップを作成します。

SNMPv3 トラップの設定

SNMPv3 は、SNMP ユーザーの認証情報を使用して、認証や暗号化などのセキュリティ機能を提供します。SNMP マネージャは、SNMP ユーザに割り当てられたパスワードが設定されている場合に限り、SNMPv3 トラップメッセージを受信できます。

トラップ宛先は、SNMPv1、SNMPv2、および SNMPv3 トラップメッセージを受信できるようになりました。

CLI を使用して **SNMPv3** トラップを設定するには

コマンドプロンプトで、次の操作を行います。

1. SNMPv3 トラップを追加します。


```
add snmp trap <trapClass> <trapDestination> -version ( V1 | V2 |
 V3)-destPort <port> -communityName <string> -srcIP <ip_addr> -
severity <severity>
```

> 注
>
> いったん設定すると、SNMP トラップのバージョンは変更できません。

例

```
1  > add snmp trap specific 192.0.2.10 -version V3 -destPort 162 -
       communityName com1 -severity Major
2  <!--NeedCopy-->
```

2. SNMP ユーザーを追加します。

   ```
   add snmp user <name> -group <string> [ -authType ( MD5 | SHA ){ -
   authPasswd } [-privType ( DES | AES ){ -privPasswd } ]]
   ```

   例

   ```
   1  > add snmp user edocs_user -group edocs_group
   2  <!--NeedCopy-->
   ```

3. SNMPv3 トラップを SNMP ユーザにバインドします。

```
bind snmp trap <trapClass> <trapDestination> [-version <version>] (-
userName <string> [-securityLevel <securityLevel>])
```

例

```
1      > bind snmp trap specific 192.0.2.10 -version V3 -userName
          edocs_user -securityLevel authPriv
2  <!--NeedCopy-->
```

GUI を使用して **SNMPv3** トラップを設定するには

1. SNMPv3 トラップを追加します。

   [システム] **>** **[SNMP]** **>** [トラップ] に移動し、SNMP バージョンとして V3 を選択して SNMP トラップを作成します。

2. SNMP ユーザーを追加します。

   [システム] **>** **[SNMP]** **>** [ユーザ] に移動し、SNMP ユーザを作成します。

3. SNMPv3 トラップを SNMP ユーザにバインドします。

   - [システム] **>** **[SNMP]** **>** [トラップ] に移動し、SNMP バージョン 3 トラップを選択します。
   - トラップをバインドするユーザーを選択し、適切なセキュリティ・レベルを定義します。

SNMP トラップロギング

NetScaler ADC アプライアンスは、SNMP トラップロギングオプションを有効にし、アプライアンスに少なくとも 1 つのトラップリスナーが設定されている場合、（ロギング機能が有効な SNMP アラームの）SNMP トラップメッセージを記録できます。これで、外部ログサーバーに送信されるトラップメッセージの監査ログレベルを指定できます。デフォルトのログレベルは「情報」です。設定可能な値は、緊急、アラート、クリティカル、エラー、警告、デバッグ、通知です。

たとえば、ログオン失敗によって生成される SNMP トラップメッセージの監査ログレベルを Critical に設定できます。この情報は、NSLOG サーバまたは SYSLOG サーバでトラブルシューティングに使用できます。

CLI を使用して SNMP トラップロギングを有効にし、トラップログレベルを設定するには

コマンドプロンプトで次のコマンドを入力して、SNMP トラップロギングを構成し、構成を確認します。

- `set snmp option [-snmpTrapLogging (ENABLED | DISABLED)][-snmpTrapLoggingLe <snmpTrapLoggingLevel>]`
- `show snmp option`

GUI を使用して SNMP トラップロギングを有効にし、SNMP トラップログレベルを構成するには

[システム] > [SNMP] に移動し、[SNMP オプションの変更] をクリックし、次のパラメータを設定します。

1. SNMP トラップロギング：アプライアンスに少なくとも 1 つのトラップリスナーが設定されている場合に SNMP トラップロギングを有効にするには、このチェックボックスをオンにします。
2. [SNMP トラップロギングレベル]：SNMP トラップの監査ログレベルを選択します。デフォルトでは、SNMP トラップの監査レベルは「情報」に設定されています。

SNMP v1 および v2 クエリ用の NetScaler 構成

August 15, 2023

SNMP マネージャーと呼ばれるリモートデバイスから、*NetScaler SNMP* エージェントにシステム固有の情報を問い合わせることができます。次に、エージェントは管理情報ベース（MIB）で要求されたデータを検索し、そのデータを SNMP マネージャに送信します。

SNMP エージェントでは、次のタイプの SNMP v1 および v2 クエリがサポートされています。

- GET
- GET NEXT

- ALL
- GET BULK

コミュニティストリングと呼ばれる文字列を作成し、それぞれをクエリタイプに関連付けることができます。各クエリタイプに 1 つ以上のコミュニティストリングを関連付けることができます。コミュニティストリングはパスワードで、SNMP マネージャーからの SNMP クエリの認証に使用されます。

たとえば、**abc や bcd などの 2 つのコミュニティストリングをクエリタイプ GET**NEXT** に関連付けると、**NetScaler** アプライアンスの **SNMP** エージェントは、**abc** または **bcd** を含む **GET** NEXT SNMP クエリパケットのみをコミュニティストリングと見なします。**

SNMP マネージャーの指定

適切な SNMP マネージャーがクエリを実行できるように NetScaler アプライアンスを構成する必要があります。また、SNMP マネージャーに必要な NetScaler 固有の情報を提供する必要があります。最大 100 の SNMP マネージャーまたはネットワークを追加できます。

IPv4 SNMP マネージャでは、マネージャの IP アドレスの代わりにホスト名を指定できます。その場合は、SNMP マネージャのホスト名を IP アドレスに解決する DNS ネームサーバを追加する必要があります。最大 5 つのホスト名ベースの SNMP マネージャーを追加できます。

> 注: アプライアンスは
> 、IPv6 アドレスを持つ SNMP マネージャーのホスト名の使用をサポートしていません。IPv6 アドレスを指定する必要があります。

少なくとも 1 つの SNMP マネージャーを設定しない場合、アプライアンスはネットワーク上のすべての IP アドレスからの SNMP クエリを受け入れて応答します。1 つ以上の SNMP マネージャを設定すると、アプライアンスはその特定の IP アドレスからの SNMP クエリのみを受け入れ、応答します。

構成から SNMP マネージャーを削除すると、そのマネージャーはアプライアンスにクエリを実行できなくなります。

コマンドラインインターフェイスを使用して **IP** アドレスを指定して **SNMP** マネージャーを追加するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp manager <IPAddress> ... [-netmask <netmask>]`
- `show snmp manager`

例 `> add snmp manager 10.102.29.10 10.102.29.15 10.102.29.30`

コマンドラインインターフェイスを使用してホスト名を指定して **SNMP** マネージャーを追加するには

重要:IP アドレスの代わりに SNMP マネージャのホスト名を指定する場合は、ホスト名を SNMP マネージャの IP アドレスに解決するように DNS ネームサーバを構成する必要があります。詳細については、「ネームサーバーの追加」を参照してください。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp manager <IPAddress> [-domainResolveRetry ****<integer>]`
- `show snmp manager`

例

```
add nameserver 10.103.128.15
add snmp manager engwiki.eng.example.net -domainResolveRetry 10
```

GUI を使用して **SNMP** マネージャーを追加するには

1. [システム] > [**SNMP**] > [マネージャー] に移動し、SNMP マネージャーを作成します。

   > 重要:
   >
   > IPv4 アドレスの代わりに SNMP マネージャーのホスト名を指定する場合は、ホスト名を SNMP マネージャーの IP アドレスに変換する DNS ネームサーバーを構成する必要があります。

> 注: アプライアンスは
> 、IPv6 アドレスを持つ SNMP マネージャーのホスト名をサポートしていません。

SNMP コミュニティの指定

コミュニティストリングと呼ばれる文字列を作成して、アプライアンス上の次の SNMP クエリタイプに関連付けることができます。

- GET
- GET NEXT
- ALL
- GET BULK

各クエリタイプに 1 つ以上のコミュニティストリングを関連付けることができます。たとえば、abc と bcd のような **2** つのコミュニティストリングをクエリタイプ **GET NEXT** に関連付けると、アプライアンスの **SNMP** エージェントは、abc または **bcd** を含む **GET NEXT SNMP** クエリパケットのみをコミュニティストリングと見なします。

クエリータイプにコミュニティストリングを関連付けない場合、SNMP エージェントはそのタイプのすべての SNMP クエリに応答します。

コマンドラインインターフェイスを使用して **SNMP** コミュニティを指定するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp community <communityName> <permissions>`
- `show snmp community`

例 `> add snmp community com all`

GUI を使用して **SNMP** コミュニティストリングを設定するには

[システム] > [**SNMP**] > [コミュニティ] に移動し、SNMP コミュニティを作成します。

SNMPv3 クエリ用の **NetScaler** 構成

August 15, 2023

簡易ネットワーク管理プロトコルバージョン 3（SNMPv3）は、SNMPv1 と SNMPv2 の基本構造とアーキテクチャに基づいています。ただし、SNMPv3 では基本アーキテクチャが強化され、認証、アクセスコントロール、データ整合性チェック、データ発信元検証、メッセージの適時性チェック、データ機密性などの管理機能とセキュリティ機能が組み込まれています。

メッセージレベルのセキュリティとアクセス制御を実装するために、SNMPv3 ではユーザーベースのセキュリティモデル（USM）とビューベースのアクセス制御モデル（VACM）を導入しています。

- ユーザーベースのセキュリティモデル。ユーザーベースのセキュリティモデル (USM) は、メッセージレベルのセキュリティを提供します。SNMP エージェントと SNMP マネージャーのユーザーとセキュリティパラメータを設定できます。USM には次の機能があります。
  - データ整合性**:** ネットワーク経由の送信中にメッセージが変更されないように保護します。
  - データオリジンの検証**:** メッセージリクエストを送信したユーザーを認証します。
  - メッセージの適時性**:** メッセージの遅延や再生を防ぐため。
  - データの機密性**:** メッセージの内容が権限のない団体や個人に開示されないようにするため。
- ビューベースのアクセス制御モデル。ビューベースのアクセス制御モデル (VACM) では、セキュリティレベル、セキュリティモデル、ユーザー名、ビュータイプなどのさまざまなパラメータに基づいて、MIB の特定のサブツリーへのアクセス権を設定できます。これにより、さまざまなマネージャに MIB へのさまざまなレベルのアクセスを提供するようにエージェントを設定できます。

NetScaler は、SNMPv3 のセキュリティ機能を実装できる以下のエンティティをサポートしています。

- SNMP エンジン

- SNMP ビュー
- SNMP グループ
- SNMP ユーザ

これらのエンティティは連携して機能し、SNMPv3 セキュリティ機能を実装します。MIB のサブツリーにアクセスできるように、ビューが作成されます。次に、必要なセキュリティレベルと定義済みのビューへのアクセス権を持つグループが作成されます。最後に、ユーザーが作成され、グループに割り当てられます。

> 注記:
>
> ビュー、グループ、およびユーザー構成は同期され、高可用性 (HA) ペアのセカンダリノードに伝達されます。ただし、エンジン ID は各 NetScaler アプライアンスに固有であるため、伝播も同期もされません。

メッセージ認証とアクセス制御を実装するには、以下を実行する必要があります。

エンジン **ID** の設定

SNMP エンジンは、SNMP エージェントに存在するサービスプロバイダーです。メッセージの送信、受信、認証などのサービスを提供します。SNMP エンジンはエンジン ID を使用して一意に識別されます。

NetScaler アプライアンスには、いずれかのインターフェースの MAC アドレスに基づいて固有の EngineID が割り当てられます。engine ID をオーバーライドする必要はありません。ただし、エンジン ID を変更したい場合はリセットできます。

コマンドラインインターフェイスを使用してエンジン **ID** を設定するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `set snmp engineId <engineID>`
- `show snmp engineId`

例 `> set snmp engineId 8000173f0300c095f80c68`

GUI を使用してエンジン **ID** を設定するには

[システム] > [**SNMP**] > [ユーザー] に移動し、[エンジン **ID** の設定] をクリックしてエンジン ID を入力します。

ビューを設定

SNMP ビューは、ユーザアクセスを MIB の特定の部分に制限します。SNMP ビューはアクセス制御の実装に使用されます。

コマンドラインインターフェイスを使用して **SNMP** ビューを追加するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp view <name> <subtree> -type ( included | excluded )`
- `show snmp view <name>`
- `rm snmp view <name> <subtree>`

各項目の意味は次のとおりです。

Name: SNMPv3 ビューの名前。大文字と小文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、アンダースコア (_) を含む 1 ～31 文字で構成できます。SNMPv3 ビューを識別しやすい名前を選択する必要があります。

サブツリー。この SNMPv3 ビューに関連付けたい MIB ツリーの特定のブランチ（サブツリー）。サブツリーは SNMP OID として指定する必要があります。これは最大長が 99 の引数です。

type。subtree パラメータで指定されたサブツリーをこのビューに含めるか、このビューから除外します。この設定は、A などのサブツリーを SNMPv3 ビューに含め、B などの A の特定のサブツリーを SNMPv3 ビューから除外する場合に便利です。これは必須の議論です。指定できる値: 含む、除外。

例　add snmp view SNMPv3test 1.1.1.1 -type included
sh snmp view SNMPv3test
rm snmp view SNMPv3test 1.1.1.1

GUI を使用して **SNMP** ビューを設定するには

[システム] > [**SNMP**] > [ビュー] に移動し、SNMP ビューを作成します。

グループを設定

SNMP グループは、SNMP ユーザーを論理的に集約したものです。これらは、アクセス制御の実装とセキュリティレベルの定義に使用されます。SNMP グループを設定して、そのグループに割り当てられたユーザーにアクセス権を設定することで、ユーザーを特定のビューに制限できます。

SNMP グループを設定して、そのグループに割り当てられたユーザーのアクセス権を設定する必要があります。

コマンドラインインターフェイスを使用して **SNMP** グループを追加するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp group <name> <securityLevel> -readViewName <string>`

- `show snmp group <name> <securityLevel>`

各項目の意味は次のとおりです。

Name: SNMPv3 グループの名前。大文字と小文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、アンダースコア (_) を含む 1 ～31 文字で構成できます。SNMPv3 グループを識別しやすい名前を選択してください。

securityLevel. NetScaler アプライアンスとグループに属する SNMPv3 ユーザー間の通信に必要なセキュリティレベル。次のオプションのいずれかを指定します。

noAuthNoPriv. 認証も暗号化も必要ありません。

authNoPriv. 認証は必須ですが、暗号化は必須ではありません。

authPriv. 認証と暗号化が必要です。注: 認証を指定する場合、SNMPv3 ユーザーをグループに割り当てるときに暗号化アルゴリズムを指定する必要があります。暗号化も指定する場合は、グループメンバーごとに認証と暗号化アルゴリズムの両方を割り当てる必要があります。これは必須の議論です。指定できる値: 認証権限なし、認証権限なし、認証権限なし

readViewName. この SNMPv3 グループにバインドする設定済みの SNMPv3 ビューの名前。このグループにバインドされている SNMPv3 ユーザは、この SNMPv3 ビューに INCLUDED タイプとしてバインドされているサブツリーにはアクセスできますが、Excluded タイプのサブツリーにはアクセスできません。NetScaler アプライアンスに同じ名前の複数の SNMPv3 ビューエントリがある場合、そのようなエントリはすべて SNMPv3 グループに関連付けられます。これは必須の議論です。Maximum Length: 31

GUI を使用して **SNMP** グループを設定するには

[システム] > [**SNMP**] > [グループ] に移動し、SNMP グループを作成します。

ユーザーを設定する

SNMP ユーザーは、エージェントが MIB へのアクセスを許可する SNMP マネージャーです。各 SNMP ユーザーは SNMP グループに割り当てられます。

エージェントでユーザーを設定し、各ユーザーをグループに割り当てる必要があります。

コマンドラインインターフェイスを使用してユーザーを設定するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp user <name> -group <string> [-authType ( MD5 | SHA ){ -authPasswd } [-privType ( DES | AES ){ -privPasswd } ]]`
- `show snmp user <name>`

各項目の意味は次のとおりです。

AuthType は、ユーザーを設定する際に使用できる認証オプションです。認証には MD5 と SHA の 2 種類があります。

PrivType は、ユーザーを設定する際に使用できる暗号化オプションです。暗号化には、キーサイズ 128 ビットの DES とキーサイズ 128 ビットの AES の 2 つのタイプがあります。

例

```
> add snmp user edocs_user -group edocs_group
<!--NeedCopy-->
```

GUI を使用して **SNMP** ユーザーを設定するには

[システム] > [**SNMP**] > [ユーザー] に移動し、SNMP ユーザーを作成します。

レート制限用の **SNMP** アラームの設定

October 25, 2023

NetScaler アプライアンスにはレート制限があります。各プラットフォームで使用可能なさまざまなモデルの詳細については、データシートを参照してください。このデータシートは www.citrix.comで入手できます。「製品」をクリックします。[**App Delivery and Security**] で、[**NetScaler**] をクリックします。「プラットフォーム」**>**「物理アプライアンス」をクリックし、「**NetScaler MPX/SDX** データシート」をクリックします。

最大スループット (Mbps) と 1 秒あたりのパケット数 (PPS) は、アプライアンス用に購入したライセンスによって決まります。レート制限のあるプラットフォームでは、スループットと PPS が上限に近づいて正常に戻ったときに通知を送信するように SNMP トラップを設定できます。

スループットと PPS は 7 秒ごとに監視されます。トラップには、高しきい値と標準しきい値を設定できます。これらの値は、ライセンス制限のパーセンテージとして表されます。次に、アプライアンスは、スループットまたは PPS が高いしきい値を超えるとトラップを生成し、監視対象パラメータが通常のしきい値に下がると 2 つ目のトラップを生成します。NetScaler ADC アプライアンスは、構成された宛先デバイスにトラップを送信することに加えて、トラップに関連するイベントを「EVENT ALERTSTARTED」および「EVENT ALERTENDED」として /var/log/ns.log ファイルに記録します。

スループット制限を超えると、パケットが失われる可能性があります。パケット損失を報告するように SNMP アラームを設定できます。

SNMP アラームとトラップの詳細については、「SNMP v1 および v2 トラップを生成するように NetScaler ADC を構成する」を参照してください。

このドキュメントでは、次の詳細について説明します。

- スループットまたは PPS の SNMP アラームの設定
- ドロップされたパケットの SNMP アラームの設定

スループットまたは **PPS** の **SNMP** アラームの設定

スループットと PPS の両方を監視するには、個別のアラームを設定し、PPS のしきい値を Mbps 単位で設定する必要があります。

CLI を使用してスループットレートの **SNMP** アラームを設定するには

コマンドプロンプトで次のコマンドを入力して SNMP アラームを設定し、しきい値を Mbps 単位で設定し、構成を確認します。

- `set snmp alarm PF-RL-RATE-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`
- `show snmp alarm PF-RL-RATE-THRESHOLD`

例

```
> set snmp alarm PF-RL-RATE-THRESHOLD -thresholdValue 70 -normalValue
    50
<!--NeedCopy-->
```

CLI を使用して **PPS** の **SNMP** アラームを設定するには

コマンドプロンプトで、次のコマンドを入力して PPS の SNMP アラームを構成し、構成を確認します。

- `set snmp alarm PF-RL-PPS-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`
- `show snmp alarm PF-RL-PPS-THRESHOLD`

例

```
> set snmp alarm PF-RL-PPS-THRESHOLD -thresholdValue 70 -normalValue 50
<!--NeedCopy-->
```

GUI を使用してスループットまたは **PPS** の **SNMP** アラームを設定するには

1. [システム**] > [SNMP] > [**アラーム] に移動し、[**PF-RL-RATE-THRESHOLD**] (スループットレートの場合) または **PF-RL-PPS-THRESHOLD** (1 秒あたりのパケット数) を選択します。
2. アラームパラメータを設定し、選択した SNMP アラームを有効にします。

ドロップされたパケットの **SNMP** アラームの設定

スループット制限を超えた結果としてドロップされたパケットのアラームと、PPS 制限を超えた結果としてドロップされたパケットのアラームを設定できます。

CLI を使用して、スループットが高すぎたためにドロップされたパケットの **SNMP** アラームを設定するには

コマンドプロンプトで入力します。

```
set snmp alarm PF-RL-RATE-PKTS-DROPPED [-state (ENABLED | DISABLED)]
[-severity <severity>] [-logging ( ENABLED | DISABLED )]
```

CLI を使用して **PPS** が多すぎたためにドロップされたパケットの **SNMP** アラームを設定するには

コマンドプロンプトで入力します。

```
set snmp alarm PF-RL-PPS-PKTS-DROPPED [-state (ENABLED | DISABLED)]
[-severity <severity>] [-logging ( ENABLED | DISABLED )]
```

GUI を使用してドロップされたパケットの **SNMP** アラームを設定するには

1. [システム] **> [SNMP] > [**アラーム] に移動し、[**PF-RL-RATE-PKTS-DROPPED**]（スループットが高すぎるためにドロップされたパケットの場合）または **PF-RL-PPS-PKTS-DROPPS** （PPS が過剰であるためにドロップされたパケットの場合）を選択します。
2. アラームパラメータを設定し、選択した SNMP アラームを有効にします。

FIPS モードでの **SNMP** の設定

August 15, 2023

FIPS モードには、認証とプライバシー (AuthPriv) オプション付きの簡易ネットワーク管理プロトコルバージョン 3 (SNMPv3) が必要です。SNMP バージョン 1 とバージョン 2 は、コミュニティストリングメカニズムを使用して管理データへの安全なアクセスを提供します。コミュニティストリングは、SNMP マネージャーと SNMP エージェントの間でクリアテキストとして送信されます。この種の通信は安全ではないため、侵入者はネットワーク上の SNMP 情報にアクセスできます。

SNMPv3 プロトコルは、ユーザーベースのセキュリティモデル（USM）とビューベースのアクセス制御モデル（VACM）を使用して、SNMP メッセージングデータへの管理アクセスを認証および制御します。SNMPv3 には、認証なしプライバシーなし（NoAuthnoPriv）、認証とプライバシーなし（AuthnoPriv）、認証とプライバシー（AuthPriv）の 3 つのセキュリティレベルがあります。

FIPS モードを有効にして NetScaler アプライアンスを再起動すると、次の SNMP 構成がアプライアンスから削除されます。

1. SNMPv1 および SNMPv2 プロトコルのコミュニティ設定。
2. noAuthnoPriv または AuthnoPriv セキュリティレベルオプションを使用して設定された SNMPv3 グループ。
3. SNMPv1 または SNMPv2、または noAuthnoPriv セキュリティレベルオプションを使用して SNMPv3 に設定されたトラップ。

アプライアンスを再起動したら、authPriv オプションを使用して SNMPv3 を設定します。SMNP v3 で AuthPriv オプションを構成する方法の詳細については、SNMPV3 のトピックを参照してください。

> 注記:
>
> FIPS モードを有効にしてアプライアンスを再起動すると、次の SNMP トラップおよびグループコマンドの実行がブロックされます。

```
1.  add snmp community <communityName> <permissions>

2.  add snmp trap <trapClass> <trapDestination> ... [-version: v1/
    v2]   [-td <positive_integer>] [-destPort <port>] [-
    communityName <string>] [-srcIP <ip_addr|ipv6_addr>] [-severity
    <severity>] [-allPartitions ( ENABLED | DISABLED )]

3.  add snmp group <name> <securityLevel : noAuthNoPriv/ authNoPriv
    > -readViewName <string>

4.  bind snmp trap specific <TrapIp>-userName <v3 user name> -
    securityLevel <noAuthNoPriv/ authNoPriv>
<!--NeedCopy-->
```

監査ロギング

August 15, 2023

> 重要
>
> Citrix では、メンテナンス時またはダウンタイム時にのみ SYSLOG または NSLOG の構成を更新することをお勧めします。セッションの作成後に構成を更新した場合、変更は既存のセッションログには適用されません。

監査とは、状態または状況を系統的に調査またはレビューすることです。監査ログ機能を使用すると、さまざまなモジュールによって収集された NetScaler ADC の状態とステータス情報をログに記録できます。ログ情報は、カーネルとユーザーレベルのデーモンに格納できます。監査ログには、SYSLOG プロトコル、ネイティブ NSLOG プロトコル、またはその両方を使用できます。

SYSLOG はロギング用の標準プロトコルです。これには次の 2 つのコンポーネントがあります。

- **SYSLOG** 監査モジュール。 NetScaler ADC アプライアンスで実行されます。
- **SYSLOG** サーバ。 NetScaler アプライアンスの基盤となる FreeBSD オペレーティングシステム（OS）またはリモートシステム上で実行されます。

SYSLOG はデータ転送にユーザーデータプロトコル (UDP) を使用します。

同様に、ネイティブ NSLOG プロトコルには次の 2 つのコンポーネントがあります。

- **NSLOG** 監査モジュール。 NetScaler ADC アプライアンスで実行されます。
- **NSLOG** サーバー。 NetScaler ADC アプライアンスの基盤となる FreeBSDOS またはリモートシステムで実行されます。

NSLOG はデータ転送に TCP を使用します。

SYSLOG サーバーまたは NSLOG サーバーを実行すると、NetScaler ADC アプライアンスに接続されます。その後、NetScaler ADC アプライアンスはすべてのログ情報を SYSLOG または NSLOG サーバーに送信し始めます。また、サーバーはログファイルに保存する前にログエントリをフィルタリングします。NSLOG または SYSLOG サーバーは、複数の NetScaler ADC アプライアンスからログ情報を受信します。NetScaler ADC アプライアンスは、ログ情報を複数の SYSLOG サーバーまたは NSLOG サーバーに送信します。

複数の SYSLOG サーバーが構成されている場合、NetScaler ADC アプライアンスは、構成されたすべての外部ログサーバーに SYSLOG イベントとメッセージを送信します。その結果、メッセージが重複して保存され、システム管理者の監視が困難になります。この問題に対処するために、NetScaler ADC アプライアンスは負荷分散アルゴリズムを提供します。アプライアンスは、メンテナンスとパフォーマンスを向上させるために、外部ログサーバー間で SYSLOG メッセージの負荷分散を行うことができます。サポートされている負荷分散アルゴリズムには、ラウンドロビン、最小帯域幅、カスタムロード、最小パケット、および監査ログハッシュが含まれます。

> 注
>
> NetScaler ADC アプライアンスは、最大 16 KB の監査ログメッセージを外部 SYSLOG サーバーに送信できます。

SYSLOG または NSLOG サーバーが NetScaler ADC アプライアンスから収集するログ情報は、メッセージ形式でログファイルに保存されます。通常、これらのメッセージには次の情報が含まれています。

- ログメッセージを生成した NetScaler ADC アプライアンスの IP アドレス。
- タイムスタンプ
- メッセージの種類
- 定義済みのログレベル (重大、エラー、通知、警告、情報、デバッグ、アラート、緊急)
- メッセージの情報

監査ログを構成するには、まず NetScaler ADC アプライアンスで監査モジュールを構成します。アプライアンスには、監査ポリシーの作成と、NSLOG サーバーまたは SYSLOG サーバーの情報の指定が含まれます。次に、NetScaler ADC アプライアンスの基盤となる FreeBSD OS またはリモートシステムに SYSLOG または NSLOG サーバーをインストールして構成します。

> 注
>
> SYSLOG はプログラムメッセージロギングの業界標準であり、さまざまなベンダーがサポートを提供しています。ドキュメントには SYSLOG サーバの設定情報は含まれていません。

NSLOG サーバーには独自の構成ファイル (auditlog.conf) があります。構成ファイル (auditlog.conf) にさらに変更を加えることで、NSLOG サーバーシステムのロギングをカスタマイズできます。

> 注
>
> ネットワーク内の Syslog アクションで Syslog サーバーを FQDN として使用する場合は、Syslog サーバーへの ICMP アクセスが必須です。環境内で ICMP アクセスがブロックされている場合は、負荷分散された Syslog サーバーとして設定し、set service コマンドの HealthMonitor パラメータの値を NO に設定します。
> ICMP の設定については、「SYSLOG サーバーの負荷分散」を参照してください。

監査ログ用の **NetScaler** アプライアンスの構成

January 11, 2024

> 警告:
>
> 従来のポリシー表現とその使用法は、NetScaler 12.0 ビルド 56.20 以降では廃止されました（使用は推奨されませんが、引き続きサポートされています）。代わりに、高度なポリシーを使用することをお勧めします。詳しくは、「高度なポリシー」を参照してください。

監査ログには、管理者がイベント履歴を時系列で確認できるように、さまざまなモジュールのステータス情報が表示されます。監査フレームワークの主な構成要素は、「監査アクション」と「監査ポリシー」です。「監査アクション」は監査サーバーの設定情報を記述し、「監査ポリシー」はバインドエンティティを「監査アクション」にリンクします。監査ポリシーは、「クラシック・ポリシー・エンジン」(CPE) フレームワークまたはプログレス・インテグレーション (PI) フレームワークを使用して、「監査アクション」を「システム・グローバル・バインド・エンティティ」にリンクします。

ただし、監査ログポリシーをグローバルエンティティにバインドする点で、ポリシーフレームワークは互いに異なります。以前は、監査モジュールは従来のポリシー表現と高度なポリシー表現のみをサポートしていました。現在、高度な表現を使用すると、監査ログポリシーをシステムグローバルエンティティにのみバインドできます。

> 注
>
> ポリシーをグローバルエンティティにバインドする場合は、同じ式のシステムグローバルエンティティにポリシーをバインドする必要があります。たとえば、クラシックポリシーを高度なグローバルエンティティにバインドしたり、高度なポリシーをクラシックグローバルエンティティにバインドしたりすることはできません。
>
> また、従来の監査ログポリシーと高度な監査ログポリシーの両方を負荷分散仮想サーバーにバインドすること

> はできません。

従来のポリシー表現による監査ログポリシーの設定

クラシックポリシーでの監査ログの設定は、次の手順で構成されます。

1. 監査ログアクションの設定。監査アクションは、異なるサーバーおよび異なるログレベルに対して設定できます。「監査アクション」は監査サーバーの設定情報を記述しますが、「監査ポリシー」はバインドエンティティを「監査アクション」にリンクします。デフォルトでは、SYSLOG はデータ転送にユーザーデータプロトコル (UDP) を使用し、NSLOG は TCP のみを使用してログ情報をログサーバーに転送します。TCP は、完全なデータを転送するために UDP よりも信頼性が高いです。SYSLOG に TCP を使用する場合、NetScaler アプライアンスのバッファ制限を設定してログを保存できます。バッファ制限に達すると、ログは SYSLOG サーバに送信されます。
2. 監査ログポリシーの設定。メッセージを SYSLOG サーバーに記録する SYSLOG ポリシーを構成するか、NSLOG サーバーにメッセージを記録する NSLOG ポリシーのいずれかを構成できます。各ポリシーには、メッセージをログに記録する、**true**または`ns_true`に設定されたルールと、SYSLOG または NSLOG アクションが含まれます。
3. 監査ログポリシーをグローバルエンティティにバインドします。監査ログポリシーは、システム、VPN、NetScaler AAA などのグローバルエンティティにグローバルにバインドする必要があります。これを実行して、すべての NetScaler システムイベントのログを有効にすることができます。優先度レベルを定義すると、監査サーバーロギングの評価順序を設定できます。優先度 0 が最高で、最初に評価されます。プライオリティ番号が大きいほど、評価のプライオリティは低くなります。

これらの各手順については、次のセクションで説明します。

監査ログアクションの設定

CLI を使用して高度なポリシーインフラストラクチャで SYSLOG アクションを設定します。

> 注
>
> NetScaler アプライアンスでは、SYSLOG サーバーの IP アドレスとポートに対して 1 つの SYSLOG アクションのみを構成できます。アプライアンスでは、同じサーバ IP アドレスおよびポートに対して複数の SYSLOG アクションを設定することはできません。

syslog アクションには、syslog サーバへの参照が含まれます。ログに記録する情報を指定し、その情報を記録する方法を説明します。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します:

```
1 -  add audit syslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )] [-
     transport ( TCP | UDP )]`
2 -  show audit syslogAction [<name>]
```



```
3
4 <!--NeedCopy-->
```

CLI を使用して高度なポリシーインフラストラクチャで NSLOG アクションを構成するには。

ns ログアクションには、nslog サーバーへの参照が含まれています。ログに記録する情報を指定し、その情報を記録する方法を説明します。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します:

```
1 - add audit nslogAction <name> <serverIP> [-serverPort <port>] -
    logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )]
2 - show audit nslogAction [<name>]
3 <!--NeedCopy-->
```

監査ログポリシーの設定

CLI を使用して、従来のポリシーインフラストラクチャで監査ログポリシーを設定します。

コマンドプロンプトで入力します:

```
1 - add audit syslogpolicy <name> <-rule> <action>
2 - add audit nslogpolicy <name> <-rule> <action>
3 <!--NeedCopy-->
```

監査 **syslog** ポリシーを監査 **syslog** グローバルにバインドする

CLI を使用して、監査ログポリシーを高度なポリシーフレームワークにバインドします。

コマンドプロンプトで入力します:

```
bind syslogGlobal -policyName <policyName> -priority <priority>
```

```
unbind syslogGlobal -policyName <policyName> -priority <priority>
```

CLI を使用して、監査ログポリシーをクラシックポリシーフレームワークにバインドします。

コマンドプロンプトで入力します:

```
bind systemglobal <policy Name> <Priority>
```

```
unbind systemglobal <policy Name> <Priority>
```

高度なポリシー表現を使用した監査ログポリシーの設定

詳細ポリシーでの監査ログの設定は、次の手順で構成されます。

1. 監査ログアクションの設定。監査アクションは、異なるサーバーおよび異なるログレベルに対して設定できます。「監査アクション」は監査サーバーの設定情報を記述しますが、「監査ポリシー」はバインドエンティティを「監査アクション」にリンクします。デフォルトでは、SYSLOG はデータ転送にユーザーデータプロトコル (UDP) を使用し、NSLOG は TCP のみを使用してログ情報をログサーバーに転送します。TCP は、完全なデータを転送するために UDP よりも信頼性が高いです。SYSLOG に TCP を使用する場合、NetScaler アプライアンスのバッファ制限を設定してログを保存できます。バッファ制限に達すると、ログは SYSLOG サーバに送信されます。
2. 監査ログポリシーの設定。メッセージを SYSLOG サーバーに記録する SYSLOG ポリシーを構成するか、NSLOG サーバーにメッセージを記録する NSLOG ポリシーのいずれかを構成できます。各ポリシーには、メッセージをログに記録する、**true**または`ns_true`に設定されたルールと、SYSLOG または NSLOG アクションが含まれます。
3. 監査ログポリシーをグローバルエンティティにバインドします。すべての NetScaler システムイベントのログを有効にするには、監査ログポリシーを SYSTEM グローバルエンティティにグローバルにバインドする必要があります。優先度レベルを定義すると、監査サーバーロギングの評価順序を設定できます。優先度 0 が最高で、最初に評価されます。プライオリティ番号が大きいほど、評価のプライオリティは低くなります。

> 注
>
> NetScaler アプライアンスは、true にバインドされているすべてのポリシーを評価します。

監査ログアクションの設定

CLI を使用して高度なポリシーインフラストラクチャで syslog アクションを設定するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します:

```
1 -  add audit syslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )] [-
     transport ( TCP | UDP )]
2 -  show audit syslogAction [<name>]
3 <!--NeedCopy-->
```

CLI を使用して、高度なポリシーインフラストラクチャの NSLOG アクションを設定します。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します:

```
1 -  add audit nslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )]
2 -  show audit nslogAction [<name>]
3 <!--NeedCopy-->
```

監査ログポリシーの設定

CLI を使用して Syslog 監査アクションを追加するには。

コマンドプロンプトで入力します:


```
1     add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
          domainResolveRetry <integer>])
2     | -lbVserverName <string>))[-serverPort <port>] -logLevel <logLevel
          >[-dateFormat <dateFormat>]
3      [-logFacility <logFacility>][-tcp ( NONE | ALL )] [-acl ( ENABLED
           | DISABLED )]
4     [-timeZone ( GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog ( YES |
          NO )]
5      [-appflowExport ( ENABLED | DISABLED )] [-lsn ( ENABLED | DISABLED
           )][-alg ( ENABLED | DISABLED )]
6     [-subscriberLog ( ENABLED | DISABLED )][-transport ( TCP | UDP )]
          [-tcpProfileName <string>][-maxLogDataSizeToHold]
7 <!--NeedCopy-->
```

例

```
1     > add audit syslogaction audit-action1 10.102.1.1 -loglevel
          INFORMATIONAL -dateformat MMDDYYYY
2     > add audit nslogAction nslog-action1 10.102.1.3 -serverport 520 -
          loglevel INFORMATIONAL -dateFormat MMDDYYYY
3     > add audit syslogpolicy syslog-pol1 TRUE audit-action1
4     > add audit nslogPolicy nslog-pol1 TRUE nslog-action1
5     > bind system global nslog-pol1 -priority 20
6 <!--NeedCopy-->
```

CLI を使用して nslog 監査アクションを追加します。

コマンドプロンプトで入力します:

```
1     add audit nslogAction <name> (<serverIP> | (<serverDomainName>[-
          domainResolveRetry <integer>])) [-serverPort <port>]       -
          logLevel <logLevel> ... [-dateFormat <dateFormat>][-logFacility
          <logFacility>] [-tcp ( NONE | ALL )][-acl ( ENABLED | DISABLED )
          ] [-timeZone ( GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog (
          YES | NO )][-appflowExport ( ENABLED | DISABLED )] [-lsn (
          ENABLED | DISABLED )][-alg ( ENABLED | DISABLED )] [-
          subscriberLog ( ENABLED | DISABLED )]`
2 <!--NeedCopy-->
```

監査ログポリシーをグローバルエンティティにバインドする

CLI を使用して Syslog 監査ログポリシーを高度なポリシーフレームワークにバインドします。

コマンドプロンプトで入力します:

```
bind audit syslogGlobal <policyName> [-globalBindType <globalBindType
```

```
unbind audit syslogGlobal <policyName>[-globalBindType <globalBindType
>]
```


GUI による監査ログポリシーの設定

1. [設定] > [システム] > [監査] > [**Syslog**] に移動します。
2.「サーバー」タブを選択します。
3.［追加］をクリックします。
4. [監査サーバーの作成] ページで、関連するフィールドに入力し、[作成] をクリックします。
5. ポリシーを追加するには、**[**ポリシー**]** タブを選択し、**[**** 追加] をクリックします。******
6. [監査 **Syslog** ポリシーの作成] ページで、関連するフィールドに値を入力し、[作成] をクリックします。
7. ポリシーをグローバルにバインドするには、ドロップダウンリストから [詳細ポリシー**] [**グローバルバインディング] を選択します。**best_syslog_policy_ever** ポリシーを選択します。［**Select**］をクリックします。
8. ドロップダウンリストから、**SYSTEM_GLOBAL** としてバインドポイントを選択し、[バインド] をクリックし、[完了] をクリックします。

ポリシーベースのロギングの設定

書き換えポリシーとレスポンダーポリシーのポリシーベースのロギングを設定できます。監査メッセージは、ポリシーの規則が TRUE と評価されたときに、定義された形式で記録されます。ポリシーベースのロギングを設定するには、高度なポリシー表現を使用して監査メッセージの形式を指定する監査メッセージアクションを設定します。アクションをポリシーに関連付けます。ポリシーは、グローバルにバインドすることも、負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにバインドすることもできます。監査メッセージアクションを使用して、syslog 形式のみ、または syslog 形式と新しい nslog 形式の両方で、さまざまなログレベルでメッセージをログに記録できます。

前提条件

- ユーザー設定可能なログメッセージ (UserDefinedAuditLog) オプションは、定義された形式でログを送信する監査アクションサーバーを設定するときに有効になります。
- 関連する監査ポリシーは、システムグローバルにバインドされます。

監査メッセージアクションの設定

監査メッセージアクションは、syslog 形式のみ、または syslog 形式と新しい ns ログ形式の両方で、さまざまなログレベルでメッセージをログに記録するように設定できます。監査メッセージアクションでは、式を使用して監査メッセージの形式を指定します。

CLI を使用して監査メッセージアクションを作成する　コマンドプロンプトで入力します:

```
add audit messageaction <name> <logLevel> <stringBuilderExpr> [-
   logtoNewnslog (YES|NO)]
<!--NeedCopy-->
```

```
add audit messageaction log-act1 CRITICAL '"Client:"+CLIENT.IP.SRC+"
   accessed "+HTTP.REQ.URL'
<!--NeedCopy-->
```

GUI を使用して監査メッセージアクションを設定する　[システム] **>** [監査] **>** [メッセージアクション] に移動し、監査メッセージアクションを作成します。

監査メッセージアクションをポリシーにバインドする

監査メッセージアクションを作成したら、それをリライトポリシーまたはレスポンダーポリシーにバインドする必要があります。ログメッセージアクションをリライトポリシーまたはレスポンダーポリシーにバインドする方法の詳細については、「[書き換えまたはレスポンダー](/ja-jp/citrix-adc/13-1/appexpert/responder.html)」を参照してください。

NSLOG サーバーのインストールと設定

August 15, 2023

インストール中に、NSLOG サーバーの実行ファイル (auditserver) が他のファイルとともにインストールされます。auditserver 実行ファイルには、NSLOG サーバーの実行や停止など、NSLOG サーバー上でいくつかのアクションを実行するためのオプションが含まれています。さらに、auditserver 実行可能ファイルを使用して、NSLOG サーバーがログの収集を開始する NetScaler ADC アプライアンスの IP アドレスで NSLOG サーバーを構成します。構成設定は NSLOG サーバ設定ファイル (auditlog.conf) で適用されます。

次に、監査サーバーの実行ファイルを実行して NSLOG サーバーを起動します。NSLOG サーバの設定は、設定ファイルの設定に基づいています。NSLOG サーバ設定ファイル (auditlog.conf) にさらに変更を加えることで、NSLOG サーバシステム上のロギングをさらにカスタマイズできます。

> 注意:
>
> NSLOG サーバーパッケージのバージョンは、NetScaler ADC のバージョンと同じである必要があります。たとえば、NetScaler のバージョンが 10.1 Build 125.9 の場合、NSLOG サーバーも同じバージョンである必要があります。

次の表は、NSLOG サーバがサポートされているオペレーティングシステムの一覧です。

オペレーティングシステム	ソフトウェア要件	注釈
Windows	Windows XP Professional、Windows Server 2003、Windows 2000/NT、Windows Server 2008、Windows Server 2008 R2	
Linux	RedHat Linux 4 以降、SUSE Linux エンタープライズ 9.3 以降	
FreeBSD	FreeBSD 6.3 またはそれ以降	NetScaler 10.5 の場合は、FreeBSD 8.4 のみを使用してください。
Mac OS	Mac OS 8.6 またはそれ以降	NetScaler 10.1 以降のリリースではサポートされていません。

NSLOG サーバを実行するプラットフォームの最小ハードウェア仕様は次のとおりです。

- プロセッサー-Intel x86 ～501 メガヘルツ (MHz)
- RAM-512 メガバイト (MB)
- コントローラ-SCSI

Linux オペレーティングシステムへの **NSLOG** サーバーのインストール

Linux システムに管理者としてログオンします。次の手順を使用して、NSLOG サーバーの実行ファイルをシステムにインストールします。

NSLOG サーバーパッケージを **Linux** オペレーティングシステムにインストールするには

1. Linux コマンドプロンプトで、次のコマンドを入力して NSauditserver.rpm ファイルを一時ディレクトリにコピーします。

   ```
   cp <path_to_cd>/Utilities/auditserver/Linux/NSauditserver.rpm /tmp
   ```

2. 次のコマンドを入力して NSauditserver.rpm ファイルをインストールします。

   rpm-i NSauditserver.rpm

   このコマンドは、ファイルを抽出し、次のディレクトリにインストールします。

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

Linux オペレーティングシステムで **NSLOG** サーバーパッケージをアンインストールするには

1. コマンドプロンプトで次のコマンドを入力して、監査サーバーのログ機能をアンインストールします。

   ```
   rpm -e NSauditserver
   ```

2. NSAuditServer RPM ファイルの詳細については、以下のコマンドを使用してください。

   ```
   rpm -qpi \*.rpm
   ```

3. インストールされている監査サーバーファイルを表示するには、次のコマンドを使用します。

   ```
   rpm -qpl *.rpm
   ```

   `*.rpm`: ファイル名を指定します。

FreeBSD オペレーティングシステムへの **NSLOG** サーバーのインストール

NSLOG サーバーをインストールする前に、NetScaler ADC 製品 CD から NSLOG パッケージをコピーするか、www.citrix.com からダウンロードする必要があります。NSLOG パッケージの名前の形式は次のとおりです。

```
AuditServer_<release number>-<build number>.zip
```

たとえば、次のようになります: `AuditServer_10.5-58.11.zip`

このパッケージには、サポートされているすべてのプラットフォーム (Linux、Windows、FreeBSD) 用のファイルが含まれています。FreeBSD オペレーティングシステムに、次の名前形式の NSLOG パッケージをインストールします。

```
audserver_bsd-<release number>-<build number>.tgz
```

たとえば、次のようになります: `audserver_bsd-10.5-58.11.tgz`

www.citrix.com から NSLOG パッケージをダウンロードするには:

1. ウェブブラウザで www.citrix.com にアクセスしてください。
2. メニューバーで、[ログイン] をクリックします。
3. ログイン認証情報を入力し、[ログイン] をクリックします。
4. メニューバーで、[ダウンロード] をクリックします。
5. **[**製品 の選択] リストから、[**NetScaler]** を選択します。
6. **NetScaler** ページで、**NSLOG** パッケージをダウンロードするリリース（リリース **10.5** など）を選択し、「ファームウェア」を選択します。
7. [ファームウェア**]** で、NSLOG パッケージをダウンロードするビルド番号の NetScaler ADC ファームウェアを選択します。
8. 表示されるページで、下にスクロールして「**Audit Servers**」を選択し、ダウンロードするパッケージの横にある「**Download File** 」をクリックします。

NSLOG サーバーパッケージを FreeBSD オペレーティングシステムにインストールするには

1. NSLOG パッケージAuditServer_<release number>-<build number>.zip（AuditServer_9.3-51.5.zipなど）をダウンロードしたシステムで、パッケージからFreeBSD NSLOG server **package** audserver_bsd-<release number>-<build number>.tgz（たとえばaudserver_bsd-9.3-51.5.tgz）を抽出します。
2. FreeBSD NSLOG サーバーパッケージ audserver_bsd-<release number>-<build number>.tgz (たとえば、audserver_bsd-9.3-51.5.tgz) を FreeBSD OS を実行しているシステム上のディレクトリにコピーします。
3. FreeBSD NSLOG サーバーパッケージがコピーされたディレクトリのコマンドプロンプトで、次のコマンドを実行してパッケージをインストールします。

   ```
   pkg_add audserver_bsd-<release number>-<build number>.tgz
   ```

   例:

   ```
   1 pkg_add audserver_bsd-9.3-51.5.tgz
   2 <!--NeedCopy-->
   ```

   次のディレクトリが抽出されます。

   - <root directory extracted from the FreeBSD NSLOG server **package** tgz file>NetScalerbin(たとえば、/var/auditserver/netscaler/bin)
   - <root directory extracted from the FreeBSD NSLOG server **package** tgz file>netscaler/etc（たとえば、/var/auditserver/netscaler/etc）
   - <root directory extracted from the FreeBSD NSLOG server **package** tgz file>\netscaler\samples（たとえば、/var/auditserver/samples）
4. コマンドプロンプトで次のコマンドを入力して、パッケージがインストールされていることを確認します。

   ```
   pkg_info | grep NSaudserver
   ```

FreeBSD オペレーティングシステムで **NSLOG** サーバーパッケージをアンインストールするには

コマンドプロンプトで以下を入力します:

```
pkg_delete NSaudserver
```

Windows オペレーティングシステムへの **NSLOG** サーバーファイルのインストール

NSLOG サーバーをインストールする前に、NetScaler ADC 製品 CD から NSLOG パッケージをコピーするか、www.citrix.com からダウンロードする必要があります。NSLOG パッケージの名前の形式は次のと

おりです: AuditServer _<release number>-<build number>.zip（AuditServer_9.3-51.5.zipなど）。このパッケージには、サポートされているすべてのプラットフォーム用の NSLOG インストールパッケージが含まれています。

www.Citrix.com から **NSLOG** パッケージをダウンロードするには

1. ウェブブラウザで www.citrix.com にアクセスしてください。
2. メニューバーで、[ログイン] をクリックします。
3. ログイン資格情報を入力し、[ログイン] をクリックします。
4. メニューバーで、[ダウンロード] をクリックします。
5. 適切なリリース番号とビルドが記載されているページを検索してください。
6. そのページの「Audit Servers」で、「Download」をクリックして、フォーマットAuditServer_<release number>-<build number>.zipの NSLOG パッケージをローカルシステム（たとえば、AuditServer_9.3-51.5.zip）にダウンロードします。

NSLOG サーバを **Windows** オペレーティングシステムにインストールするには

1. NSLOG パッケージAuditServer_<release number>-<build number>.zip（AuditServer_9.3-51.5.zipなど）をダウンロードしたシステムで、パッケージからaudserver_win-<release number>-<build number>.zip（たとえば、audserver_win-9.3-51.5.zip）抽出します。
2. 抽出したファイルaudserver_<release number>-<build number>.zip(audserver_win-9.3-51.5.zipなど）を、NSLOG サーバをインストールする Windows システムにコピーします。
3. audserver_<release number>-<build number>.zipファイルを解凍します (例: audserver_win-9.3-51.5.zip)。
4. 次のディレクトリが抽出されます。
   a) <root directory extracted from the Windows NSLOG server **package** zip file>\bin(たとえば、C:\audserver_win-9.3-51.5\bin)
   b) <root directory extracted from the Windows NSLOG server **package** zip file>\etc（たとえば、C:\audserver_win-9.3-51.5\etc）
   c) <root directory extracted from the Windows NSLOG server **package** zip file>\samples（たとえば、C:\audserver_win-9.3-51.5\samples）
5. コマンドプロンプトで、<root directory extracted from the Windows NSLOG server **package** zip file>\bin pathから次のコマンドを実行します。


```
audserver -install -f <directorypath>\auditlog.conf
```

<directorypath>: 設定ファイル (auditlog.conf) へのパスを指定します。デフォルトでは、log.confは\<root directory extracted from Windows NSLOG server package zip file\>\samplesディレクトリの下にあります。ただし、auditlog.conf を目的のディレクトリにコピーすることはできます。

Windows オペレーティングシステムで **NSLOG** サーバをアンインストールするには

コマンドプロンプトで、<root directory extracted from Windows NSLOG server package zip file>\bin パスから以下を実行します。

```
audserver -remove
```

NSLOG サーバーコマンドオプション

NSLOG サーバコマンドの詳細については、監査サーバオプションを参照してください。

監査サーバーの実行可能ファイルが存在するディレクトリから audserver コマンドを実行します。

- Windows の場合: \ns\bin
- Solaris および Linux の場合: \usr\local\netscaler\bin

監査サーバー設定ファイルは次のディレクトリにあります。

- Windows の場合: \ns\etc
- Linux の場合: \usr\local\netscaler\etc

監査サーバーの実行ファイルは Linux および FreeBSD ./auditserver と同様に起動されます。

NSLOG サーバーへの **NetScaler ADC** アプライアンスの **IP** アドレスの追加

構成ファイル（
auditlog.conf）に、イベントをログに記録する必要がある NetScaler ADC アプライアンスの IP アドレスを追加します。

NetScaler ADC アプライアンスの **IP** アドレスを追加するには

コマンドプロンプトで、次のコマンドを入力します。

```
audserver -addns -f <directorypath\>\auditlog.conf
```

<directorypath>: 設定ファイル (auditlog.conf) へのパスを指定します。

次のパラメータの情報を入力するよう求められます。

NSIP: NetScaler アプライアンスの IP アドレス（10.102.29.1 など）を指定します。

ユーザー ID: ユーザー名を指定します (たとえば、nsroot)。

パスワード: パスワードを指定します (例:nsroot)。

複数の NetScaler ADC IP アドレス（NSIP）を追加し、後ですべての NetScaler ADC アプライアンスイベントの詳細をログに記録したくない場合は、auditlog.conf ファイルの最後にある NSIP ステートメントを削除して NSIP を手動で削除できます。高可用性（HA）セットアップでは、audserver コマンドを使用して、プライマリとセカンダリの NetScaler IP アドレスの両方を auditlog.conf に追加する必要があります。IP アドレスを追加する前に、ユーザー名とパスワードがシステムに存在することを確認してください。

NSLOG サーバ設定ファイルの検証

設定ファイル (audit log.conf) の構文が正しいかどうかを確認して、ロギングが起動して正しく機能するようにしてください。

構成を確認するには、コマンドプロンプトで次のコマンドを入力します。

```
audserver -verify -f <directorypath>\auditlog.conf
```

<directorypath>: Specifies the path to the configuration file (audit log.conf)。

NSLOG サーバーを実行しています

August 15, 2023

監査サーバーのログ記録を開始するには

コマンドプロンプトで以下のコマンドを入力します。

audserver -start -f <directorypath>\auditlog.conf

<directorypath>: 設定ファイル (audit log.conf) へのパスを指定します。

FreeBSD または **Linux** でバックグラウンドプロセスとして開始される監査サーバーのログ記録を停止するには

次のコマンドを入力します。

```
audserver -stop
```


Windows でサービスとして開始される監査サーバーのログ記録を停止するには

次のコマンドを入力します。

```
audserver -stopservice
```

NSLOG サーバーのロギングのカスタマイズ

August 15, 2023

NSLOG サーバー設定ファイル (log.conf) にさらに変更を加えることで、NSLOG サーバーのログをカスタマイズできます。テキストエディタを使用して、サーバーシステムの log.conf 設定ファイルを変更します。

ロギングをカスタマイズするには、設定ファイルを使用してフィルタとログプロパティを定義します。

- ログフィルタ。NetScaler ADC アプライアンスまたは一連の NetScaler ADC アプライアンスからのログ情報をフィルタリングします。
- ログプロパティ。各フィルタには、関連する一連のログプロパティがあります。ログプロパティでは、フィルタリングされたログ情報の保存方法を定義します。

このドキュメントでは、次の詳細について説明します。

- フィルターの作成
- ログプロパティの指定

フィルタの作成

設定ファイル (audit log.conf) にあるデフォルトのフィルター定義を使用するか、フィルターを変更するか、新しいフィルターを作成できます。複数のログフィルタを作成できます。

> 注:
> 統合ロギングでは、フィルター定義のないログトランザクションが発生した場合、デフォルトのフィルターが使用されます (有効になっている場合)。すべての NetScaler アプライアンスの統合ログを構成する唯一の方法は、デフォルトのフィルターを定義することです。

フィルタを作成するには

コマンドプロンプトで、構成ファイル (auditlog.conf) に次のコマンドを入力します。

```
filter <filterName> [IP <ip>] [NETMASK <mask>] ON | OFF]
<!--NeedCopy-->
```


フィルター名: フィルターの名前を指定します (最大 64 文字の英数字)。

ip: IP アドレスを指定します。

マスク: サブネットで使用するサブネットマスクを指定します。

ON を指定してフィルターを有効にしてトランザクションをログに記録するか、OFF を指定してフィルターを無効にします。引数が指定されていない場合、フィルターは ON です。

例:

```
filter F1 IP 192.168.100.151 ON
<!--NeedCopy-->
```

フィルター F2 を IP アドレス 192.250.100.1 から 192.250.100.254 の IP アドレスに適用するには:

```
filter F2 IP 192.250.100.0 NETMASK 255.255.255.0 ON
<!--NeedCopy-->
```

FilterName は、IP アドレスや IP アドレスとネットマスクの組み合わせなど、他のオプションパラメータを使用してフィルタを定義する場合に必須のパラメータです。

ログプロパティを指定する

フィルターに関連するログプロパティは、フィルターに存在するすべてのログエントリに適用されます。次の例に示すように、ログプロパティの定義は BEGIN というキーワードで始まり、END で終わります。

```
BEGIN <filtername>
    logFilenameFormat ...
    logDirectory ...
    logInterval ...
    logFileSizeLimit ....
END
<!--NeedCopy-->
```

定義には、次の項目を含めることができます。

- **LogFileNameFormat** は、ログファイルのファイル名形式を指定します。ファイル名には次の種類があります。
  - 静的: 絶対パスとファイル名を指定する定数文字列。
  - ダイナミック: 次の形式指定子を含む式:
    * 日付 (% {フォーマット} t)
    * NSIP でファイル名を作成

  例:

  ```
  LogFileNameFormat Ex%`{
  `%m%d%y }
  ```



```
t.log
<!--NeedCopy-->
```

これにより、最初のファイル名が Exmmddyy.log として作成されます。新しいファイルには、exmmddyy.log.0、
exmmddyy.log.1 などの名前が付けられます。次の例では、ファイルサイズが 100 MB に達すると、新しいファイルが作成されます。

例:

```
LogInterval size
LogFileSize 100
LogFileNameFormat Ex%`{
`%m%d%y }
t
<!--NeedCopy-->
```

> 注意
>
> LogFileNameFormat パラメーターで指定された日付形式%t は、そのフィルターのログ間隔プロパティよりも優先されます。指定したログファイルサイズに達したときではなく、毎日新しいファイルが作成されないようにするには、LogFileNameFormat パラメーターに%t を使用しないでください。

- **LogDirectory** は、ログファイルのディレクトリ名形式を指定します。ファイルの名前は、次のいずれかになります。
  - 静的: 絶対パスとファイル名を指定する定数文字列です。
  - ダイナミック: 次の形式指定子を含む式です。
    - 日付 (% {フォーマット} t)
    - NSIP でディレクトリを作成

  ディレクトリセパレータはオペレーティングシステムによって異なります。Windows では、ディレクトリセパレータを使用してください。

  例**:**

  ```
  LogDirectory dir1\dir2\dir3
  <!--NeedCopy-->
  ```

  他のオペレーティングシステム (Linux、FreeBSD など) では、ディレクトリセパレータを使用してください。

- **LogInterval** は、新しいログファイルが作成される間隔を指定します。次のいずれかの値を使用します。
  - Hourly: 1 時間ごとにファイルが作成されます。デフォルト値です。
  - 毎日: ファイルは毎日午前 0 時に作成されます。
  - Weekly: 毎週日曜日の午前 0 時にファイルが作成されます。
  - 毎月: ファイルは、その月の最初の日の午前 0 時に作成されます。
  - なし: ファイルは、監査サーバーのロギングが開始されたときに 1 回だけ作成されます。

- サイズ: ファイルは、ログファイルのサイズ制限に達した場合にのみ作成されます。

例:

```
LogInterval Hourly
<!--NeedCopy-->
```

- **LogFileSizeLimit** は、ログファイルの最大サイズ (MB 単位) を指定します。上限に達すると、新しいファイルが作成されます。

> 注
>
> loginterval プロパティは、値としてサイズを指定することでオーバーライドできます。

デフォルトのログファイルサイズ制限は 10 MB です。

例:

```
LogFileSizeLimit 35
<!--NeedCopy-->
```

TCP を介した SYSLOG

August 15, 2023

syslog は、イベント通知メッセージを送信するための標準です。これらのメッセージは、ローカルまたは外部のログサーバーに保存できます。Syslog を使用すると、ネットワーク管理者はログメッセージを統合し、収集されたデータから洞察を引き出すことができます。

Syslog はもともと UDP 上で動作するように設計されており、パケット損失を最小限に抑えながら同じネットワーク内で大量のデータを送信できます。ただし、通信事業者は、ネットワーク間で信頼性の高い、順序付けられたデータ伝送が必要なため、TCP 経由で syslog データを送信することを好みます。たとえば、電話会社はユーザーのアクティビティを追跡し、TCP はネットワーク障害発生時に再送信を提供します。

TCP を介した Syslog の仕組み

syslog over TCP がどのように機能するかを理解するために、次の 2 つの仮説的なケースを考えてみましょう。

ネットワーク管理者の Sam は、重要なイベントを外部の syslog サーバに記録したいと考えています。

ISP である XYZ Telecom は、政府規制に準拠するために、大量のデータを syslog サーバーに送信して保存する必要があります。

どちらの場合も、ログメッセージは信頼できるチャネルを介して送信し、外部の syslog サーバに安全に保存する必要があります。UDP とは異なり、TCP は接続を確立し、メッセージを安全に送信し、ネットワーク障害のために破損または失われたデータを（送信者から受信者に）再送信します。

NetScaler ADC アプライアンスは、UDP を介してローカルの syslog デーモンにログメッセージを送信し、ログメッセージを TCP または UDP 経由で外部の syslog サーバーに送信します。

Syslog の **SNIP** サポート

監査ログモジュールが syslog メッセージを生成するとき、外部 syslog サーバーにメッセージを送信するための送信元アドレスとして NetScaler ADC IP（NSIP）アドレスを使用します。SNIP を送信元アドレスとして設定するには、それを netProfile オプションの一部にして、netProfile を syslog アクションにバインドする必要があります。

> 注**:**
>
> TCP は SNIP を使用して監視プローブを送信し、接続をチェックし、NSIP 経由でログを送信します。したがって、syslog サーバは SNIP 経由で到達可能である必要があります。Net プロファイルを使用して、すべての TCP syslog トラフィックを SNIP 経由で完全にリダイレクトできます。

SNIP アドレスの使用は、内部ログではサポートされていません。

完全修飾ドメイン名監査ログのサポート

以前は、audit-log モジュールは、ログメッセージの送信先である外部 syslog サーバの宛先 IP アドレスを使用して設定されていました。現在、監査ログサーバーは、宛先 IP アドレスの代わりに完全修飾ドメイン名 (FQDN) を使用します。FQDN 設定は、syslog サーバの設定済みドメイン名を、監査ログモジュールからログメッセージを送信するための対応する宛先 IP アドレスに解決します。ドメイン名を解決し、ドメインベースのサービスの問題を回避するには、ネームサーバーを適切に設定する必要があります。

> 注
>
> FQDN を構成する場合、syslog アクションまたは nslog アクションでの同じ NetScaler ADC アプライアンスのサーバードメイン名構成はサポートされていません。

コマンドラインインターフェイスを使用して **Syslog over TCP** を構成する

コマンドラインインターフェイスを使用して TCP 経由で syslog メッセージを送信するように NetScaler ADC アプライアンスを構成するには

コマンドプロンプトで入力します。


```
1     add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
          domainResolveRetry <integer>]) | -lbVserverName<string>))[-
          serverPort <port>] -logLevel <logLevel>[-dateFormat <dateFormat
          >] [-logFacility <logFacility>]   [-tcp ( NONE | ALL )] [-acl (
          ENABLED | DISABLED )][-timeZone ( GMT_TIME | LOCAL_TIME )][-
          userDefinedAuditlog ( YES | NO )][-appflowExport ( ENABLED |
          DISABLED )] [-lsn ( ENABLED | DISABLED )][-alg ( ENABLED |
          DISABLED )] [-subscriberLog ( ENABLED | DISABLED )][-transport (
           TCP | UDP )] [-tcpProfileName <string>][-maxLogDataSizeToHold <
          positive_integer>][-dns ( ENABLED | DISABLED )] [-netProfile <
          string>]
2 <!--NeedCopy-->
```

```
1     add audit syslogaction audit-action1 10.102.1.1 -loglevel
          INFORMATIONAL -dateformat MMDDYYYY -transport TCP
2 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **SNIP IP** アドレスをネットプロファイルオプションに追加する

コマンドラインインターフェイスを使用して SNIP IP アドレスをネットプロファイルに追加するには

コマンドプロンプトで入力します。

```
1     add netProfile <name> [-td <positive_integer>] [-srcIP <string>][-
          srcippersistency ( ENABLED | DISABLED )][-overrideLsn ( ENABLED
          | DISABLED )]add syslogaction <name> <serverIP>  -loglevel all
           -netprofile net1
2 <!--NeedCopy-->
```

```
1     add netprofile net1  -srcip 10.102.147.204`
2 <!--NeedCopy-->
```

ここで、srcIP は SNIP です。

コマンドラインインターフェイスを使用して **Syslog** アクションに **net** プロファイルを追加する

コマンドラインインターフェイスを使用して syslog アクションに netProfile オプションを追加するには

コマンドプロンプトで入力します。

```
1      add audit syslogaction <name> (<serverIP> | -lbVserverName <string
           >)  -logLevel <logLevel>
2     -netProfile <string> …
3
4 <!--NeedCopy-->
```

```
1     add syslogaction sys_act1 10.102.147.36  -loglevel all  -netprofile
          net1
2 <!--NeedCopy-->
```


ここで、-netprofile は設定されたネットプロファイルの名前を指定します。SNIP アドレスは netProfile の一部として設定され、この netProfile オプションは syslog アクションにバインドされます。

> 注:
>
> NetProfile は、SYSLOGUDP または SYSLOGTCP 負荷分散仮想サーバーにバインドされている SYSLOGUDP または SYSLOGTCP サービスに常にバインドする必要があります。

コマンドラインインターフェイスを使用した **FQDN** サポートの設定

コマンドラインインターフェイスを使用して Syslog アクションにサーバードメイン名を追加するには

コマンドプロンプトで入力します。

```
1 add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
     domainResolveRetry <integer>]) | -lbVserverName <string>)) -logLevel
      <logLevel> ...
2     set audit syslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>]-
        serverDomainName <string>] [-lbVserverName <string>]-
        domainResolveRetry <integer>] [-domainResolveNow]
3 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して Nslog アクションにサーバードメイン名を追加するには。

コマンドプロンプトで入力します。

```
1     add audit nslogAction <name> (<serverIP> | (<serverDomainName>[-
        domainResolveRetry <integer>]))  -logLevel <logLevel> ...
2     set audit nslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>][-
        serverDomainName <string>] [-domainResolveRetry <integer>][-
        domainResolveNow]
3 <!--NeedCopy-->
```

ここで、serverDomainName。ログサーバーのドメイン名。serverIP/lbvServerName と相互に排他的です。

domainResolverRetry 整数。DNS 解決が失敗した後、ドメイン名を解決する次の DNS クエリを送信する前に、NetScaler ADC アプライアンスが待機する時間（秒単位）。

domainResolveNow。サーバーのドメイン名を解決するために DNS クエリをすぐに送信する必要がある場合に含まれます。

GUI を使用した **TCP** 経由の **syslog** の設定

GUI を使用して TCP 経由で Syslog メッセージを送信するように NetScaler ADC アプライアンスを構成するには

1. [システム] > [監査] > [**Syslog**] に移動し、[サーバー] タブを選択します。
2. [追加] をクリックし、[転送タイプ] として [**TCP**] を選択します

GUI を使用して **SNIP** サポートのネットプロファイルを構成する　GUI を使用して SNIP サポートのネットプロファイルを構成するには

1. **System > Auditing > Syslog** に移動して **Servers** タブを選択します。
2. [追加] をクリックし、リストからネットプロファイルを選択します。

GUI を使用した **FQDN** の設定　GUI を使用して FQDN を構成するには

1. [システム] > [監査] > [**Syslog**] に移動し、[サーバー] タブを選択します。
2.「追加」をクリックし、リストから「サーバーの種類」と「サーバーのドメイン名」を選択します。

SYSLOG サーバーの負荷分散

August 15, 2023

NetScaler ADC アプライアンスは、構成されたすべての外部ログサーバーに SYSLOG イベントとメッセージを送信します。その結果、冗長なメッセージが保存され、システム管理者の監視が困難になります。この問題に対処するために、NetScaler ADC アプライアンスは、メンテナンスとパフォーマンスを向上させるために外部ログサーバー間で SYSLOG メッセージを負荷分散できる負荷分散アルゴリズムを提供します。サポートされる負荷分散アルゴリズムには、ラウンドロビン、最小帯域幅、カスタムロード、最小接続、最小パケット、および AuditLogHash が含まれます。

コマンドラインインターフェイスを使用した SYSLOG サーバーの負荷分散

コマンドプロンプトで入力します。

1. サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

   ```
   add service <name>(<IP> | <serverName>)<serviceType (SYSLOGTCP |
   SYSLOGUDP)> <port>
   ```

2. 負荷分散仮想サーバーを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定し、負荷分散方式を AUDITLOGHASH として指定します。

   ```
   add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-
   lbMethod <AUDITLOGHASH>]
   ```

3. サービスを負荷分散仮想サーバーにバインドします。

   ```
   Bind lb vserver <name> <serviceName>
   ```

4. SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つ負荷分散サーバー名を指定します。

   ```
   add syslogaction <name> <serverIP> [-lbVserverName <string>] [-
   logLevel <logLevel>]
   ```


5. ルールとアクションを指定して SYSLOG ポリシーを追加します。

   ```
   add syslogpolicy <name> <rule> <action>
   ```

6. ポリシーを有効にするには、SYSLOG ポリシーをシステムグローバルにバインドします。

   ```
   bind system global <policyName>
   ```

GUI を使用した SYSLOG サーバーの負荷分散

1. サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

   [トラフィック管理] > [サービス] に移動し、[追加] をクリックして、プロトコルとして [**SYLOGTCP**] または [**SYSLOGUDP**] を選択します。

2. 負荷分散仮想サーバーを追加し、サービスタイプを SYSLOGTCP または SYSLOGTCP、負荷分散方法を AUDITLOGHASH として指定します。

   [トラフィック管理] > [仮想サーバー] に移動し、[追加] をクリックして、プロトコルとして **SYLOGTCP** または SYSLOGUDP** を選択します。

3. サービスを負荷分散仮想サーバーにバインドします。

   [トラフィック管理] > [仮想サーバー] に移動し、仮想サーバーを選択し、[負荷分散方法**]** で **[**AUDITLOGHASH**] を選択します。**

4. SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つ負荷分散サーバー名を指定します。

   [システム] > [監査] に移動し、[サーバー] をクリックし、[サーバー] で [**LB Vserver**] オプションを選択してサーバーを追加します。

5. ルールとアクションを指定して SYSLOG ポリシーを追加します。

   [システム] > [**Syslog**] に移動し、[ポリシー] をクリックして、SYSLOG ポリシーを追加します。

6. ポリシーを有効にするには、SYSLOG ポリシーをシステムグローバルにバインドします。

   [システム] > [**Syslog**] に移動し、SYSLOG ポリシーを選択して [アクション] をクリックし、[グローバルバインディング] をクリックして、ポリシーをシステムグローバルにバインドします。

例:

次の構成では、AUDITLOGHASH を負荷分散方法として使用して、外部ログサーバー間で SYSLOG メッセージの負荷分散を指定します。AUDITLOGHASH メソッドは、監査エージェントからの入力ハッシュ値に基づいてトラフィックの負荷を分散します。エージェントは、NetScaler ADC アプライアンスで監査ログを生成するモジュールです。たとえば、エージェント LSN がクライアント IP アドレスに基づいて監査ログをロードバランシングする場合、LSN モジュールは clientIP に基づいてハッシュ値を生成し、そのハッシュ値を監査ログモジュールに渡します。監査ログモジュールは、同じハッシュ値を持つ監査ログメッセージを外部 syslog サーバーに送信します。

NetScaler ADC アプライアンスは、サービス、service1、service2、およびサービス 3 の間で負荷分散される SYSLOG イベントとメッセージを生成します。


```
add service service1 192.0.2.10 SYSLOGUDP 514
add service service2 192.0.2.11 SYSLOGUDP 514
add service service3 192.0.2.11 SYSLOGUDP 514
add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
bind lb vserver lbvserver1 service1
bind lb vserver lbvserver1 service2
bind lb vserver lbvserver1 service3
add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
add syslogpolicy syspol1 ns_true sysaction1
bind system global syspol1
<!--NeedCopy-->
```

ICMP パケットがブロックされている場合、次のコマンドを使用して LB サーバと FQDN を使用して SYSLOG を設定します。

```
set service service1 -healthMonitor NO
```

制限事項:

- NetScaler ADC アプライアンスは、ログサーバー間で SYSLOG メッセージを負荷分散する外部負荷分散仮想サーバーをサポートしていません。

ログプロパティのデフォルト設定

August 15, 2023

以下は、ログプロパティのデフォルト設定を含むデフォルトフィルターの例です。

```
begin default
 logInterval Hourly
 logFileSizeLimit 10
 logFilenameFormat  auditlog%`{
 `%y%m%d }
 t.log
 end default
<!--NeedCopy-->
```

次に、デフォルトフィルタを定義する 2 つの例を示します。

例 **1**:

```
Filter f1 IP 192.168.10.1
<!--NeedCopy-->
```

これにより、NSI 192.168.10.1 のログファイルが作成され、そのログのデフォルト値が有効になります。

例 **2**:


```
Filter f1 IP 192.168.10.1
begin f1
    logFilenameFormat logfiles.log
end f1
<!--NeedCopy-->
```

これにより、NSIP 192.168.10.1 のログファイルが作成されます。ログファイル名の形式が指定されているので、他のログプロパティのデフォルト値が有効になります。

Sample configuration file (audit.conf)

December 8, 2023

Following is a sample configuration file:

```
##############################
# This is the Auditserver configuration file
# Only the default filter is active
# Remove leading # to activate other filters
##############################
MYIP <NSAuditserverIP>
MYPORT 3023
#   Filter filter_nsip  IP <Specify the  IP address to filter on > ON
#   begin filter_nsip
#       logInterval         Hourly
#       logFileSizeLimit    10
#       logDirectory         logdir\%A\
#       logFilenameFormat   nsip%\\{
 \\%d%m%Y }
 t.log
#   end filter_nsip
Filter default
begin default
    logInterval      Hourly
    logFileSizeLimit    10
    logFilenameFormat   auditlog%\{
 \%y%m%d }
 t.log
end default
<!--NeedCopy-->
```

Web サーバーロギング

August 15, 2023

Web サーバーのロギング機能を使用して、HTTP および HTTPS リクエストのログをクライアントシステムに送信し、保存および取得できます。この機能には次の 2 つの要素があります。

- NetScaler 上で稼働する Web ログサーバー。
- クライアントシステム上で実行される NetScaler Web ロギング（NSWL）クライアント。

NetScaler Web ロギング（NSWL）クライアントを実行すると、

1. NetScaler に接続します。
2. NetScaler は、HTTP と HTTPS のリクエストログエントリをクライアントに送信する前にバッファリングします。
3. クライアントは、エントリを保存する前にエントリをフィルタリングできます。

Web サーバーのロギングを構成するには、まず NetScaler で Web ログ機能を有効にし、ログエントリを一時的に保存するためのバッファのサイズを設定します。次に、NSWL をクライアントシステムにインストールします。次に、NetScaler IP アドレス（NSIP）を NSWL 構成ファイルに追加します。これで、NSWL クライアントを起動してロギングを開始する準備ができました。NSWL 設定ファイル (log.conf) にさらに変更を加えることで Web サーバーのロギングをカスタマイズできます。

Web サーバーロギング用の **NetScaler** の構成

April 15, 2024

NetScaler を Web サーバーログ用に構成するには、Web サーバーログ機能のみを有効にする必要があります。オプションで、次の構成を実行できます。

- NetScaler Web ロギング（NSWL）クライアントに送信される前に、ログ情報を保存するバッファのサイズ（デフォルトサイズは 16 MB）を変更します。
- NSWL クライアントにエクスポートするカスタム HTTP ヘッダーを指定します。最大 2 つの HTTP 要求と 2 つの HTTP 応答ヘッダー名を設定できます。

NetScaler Web サーバーログ（NSWL）の構成方法について詳しくは、次のビデオを参照してください：

これは埋め込みビデオです。リンクをクリックしてビデオを見る

コマンドラインインターフェイスを使用して **Web** サーバーのロギングを設定するには

コマンドプロンプトで、次の操作を実行します。

- Web サーバロギング機能を有効にします。

```
enable ns feature WL
```


- [オプション] ログに記録された情報を保存するためのバッファサイズを変更します。

  ```
  set ns weblogparam -bufferSizeMB <size>
  ```

  > 注:
  >
  > 変更を有効にするには、Web サーバーのログ機能を無効にしてから再度有効にする必要があります。

- [オプション] エクスポートするカスタム HTTP ヘッダー名を指定します。

  ```
  set ns weblogparam [-customReqHdrs <string> ...] [-customRspHdrs
  <string> ...]
  ```

```
1      > enable ns feature WL
2      Done
3      > set ns weblogparam -bufferSizeMB 60
4      Done
5      > show ns weblogparam
6          Web Logging parameters:
7          Log buffer size:  60MB
8          Custom HTTP request headers: (none)
9          Custom HTTP response headers: (none)
10     Done
11     > set ns weblogparam -customReqHdrs req1 req2 -customRspHdrs res1
           res2
12     Done
13     > show ns weblogparam
14         Web Logging parameters:
15         Log buffer size:  60MB
16         Custom HTTP request headers: req1, req2
17         Custom HTTP response headers: res1, res2
18     Done
19 <!--NeedCopy-->
```

GUI を使用して **Web** サーバーのロギングを設定するには

1. [システム] > **[設定]** に移動し、次の操作を実行します。

   a) Web サーバーのロギング機能を有効にするには、「拡張機能の変更」をクリックして「**Web** ロギング」を選択します。

   b) バッファサイズを変更するには、「グローバルシステム設定の変更」をクリックし、「**Web** ログ」にバッファサイズを入力します。

   c) エクスポートするカスタム HTTP ヘッダーを指定するには、「グローバルシステム設定の変更」をクリックし、「**Web** ログ」でヘッダー値を指定します。

NetScaler Web ロギング（NSWL）クライアントのインストール

August 15, 2023

NSWL をインストールすると、クライアント実行可能ファイル（NSWL）が他のファイルと一緒にインストールされます。NSWL 実行可能ファイルには、使用できるオプションのリストが表示されます。詳細については、「NSWL クライアントの設定」を参照してください。

> 注意:
>
> NSWL クライアントのバージョンは NetScaler と同じである必要があります。たとえば、NetScaler のバージョンが 10.1 Build 125.9 の場合、NSWL クライアントも同じバージョンである必要があります。また、Web ロギング (NSWL) クライアントは 32 ビットと 64 ビットの両方のサーバーマシンで動作します。ダウンロードページには 32 ビットのウェブログクライアントしかありません。64 ビットの Web ログクライアントはリクエストに応じて提供されます。詳細については NetScaler サポートに問い合わせることをお勧めします。

次の表は、NSWL クライアントをインストールできるオペレーティングシステムの一覧です。

|** オペレーティングシステム **|** バージョン **|** ハードウェア要件 **|** 注釈 **|
|–|–|–|–|
|Windows|Windows Server 2016 以降 | プロセッサー-x86/amd64 CPU (1 GHz 以上)、RAM-4 GB (またはそれ以上) | |macOS|macOS 8.6 以降 |NetScaler 10.1 以降のリリースではサポートされていません。
|
|Linux|Ubuntu、SUSE Linux、CentOS、Red Hat Enterprise Linux が 2016 年以降にリリースされた | プロセッサー-x86/amd64 CPU (1 GHz 以上)、RAM-4 GB (またはそれ以上) |
|Solaris|Solaris Sun OS 5.6 以降 | プロセッサー-UltrasParc-III 400 MHz、RAM-512 MB、コントローラー-SCSI|NetScale ではサポートされていません R 10.5 およびそれ以降のリリース。|
|FreeBSD|FreeBSD 6.3 以降 | プロセッサ-x86/amd64 CPU (1 GHz 以上)、RAM-4 GB (またはそれ以上) |NetScaler 10.5 の場合は、FreeBSD 8.4 のみを使用してください。| AIX|AIX 6.1||NetScaler 10.5 以降のリリースではサポートされていません。
|

NSWL クライアントシステムが CPU の制限のためにログトランザクションを処理できない場合、Web ログバッファがオーバーランし、ロギングプロセスが再開されます。

> 注意
>
> ロギングを再開すると、ログトランザクションが失われる可能性があります。

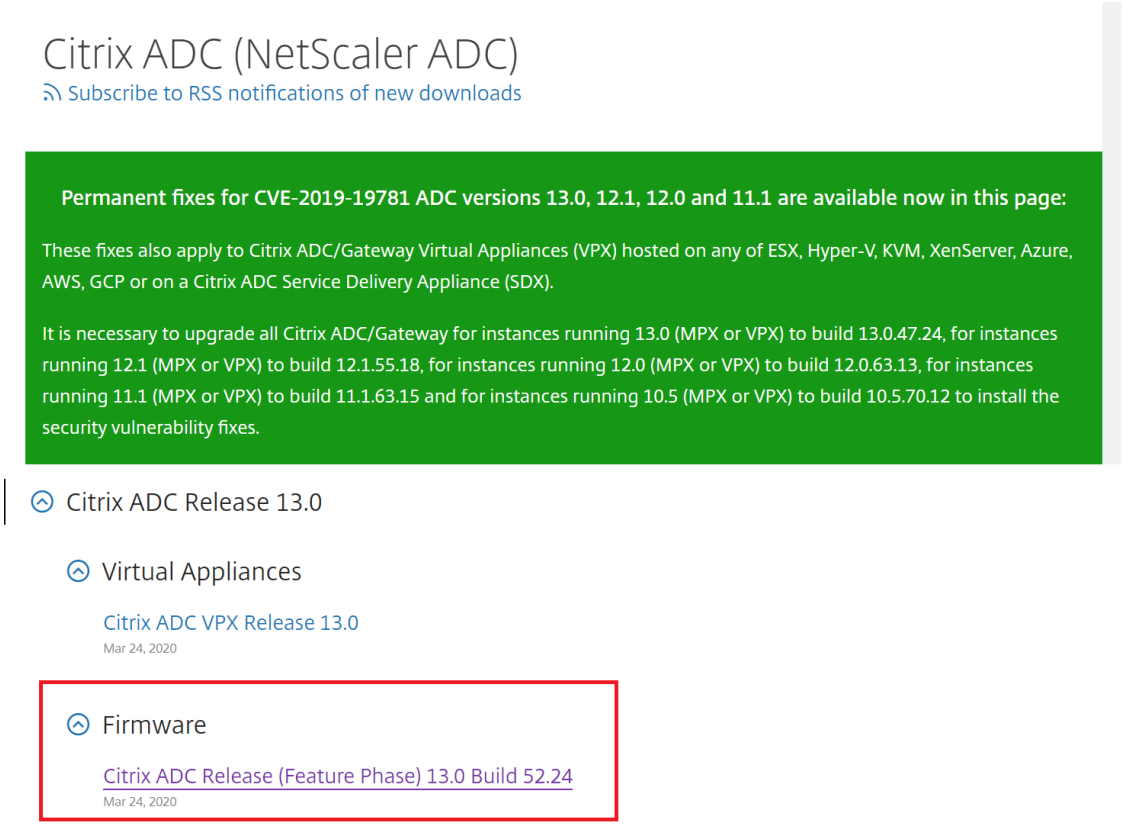
CPU の制限による NSWL クライアントシステムのボトルネックを一時的に解決するには、NetScaler アプライアンスの Web サーバーのロギングバッファサイズを調整します。この問題を解決するには、サイトのスループットを処理できるクライアントシステムが必要です。

NSWL クライアントのダウンロード

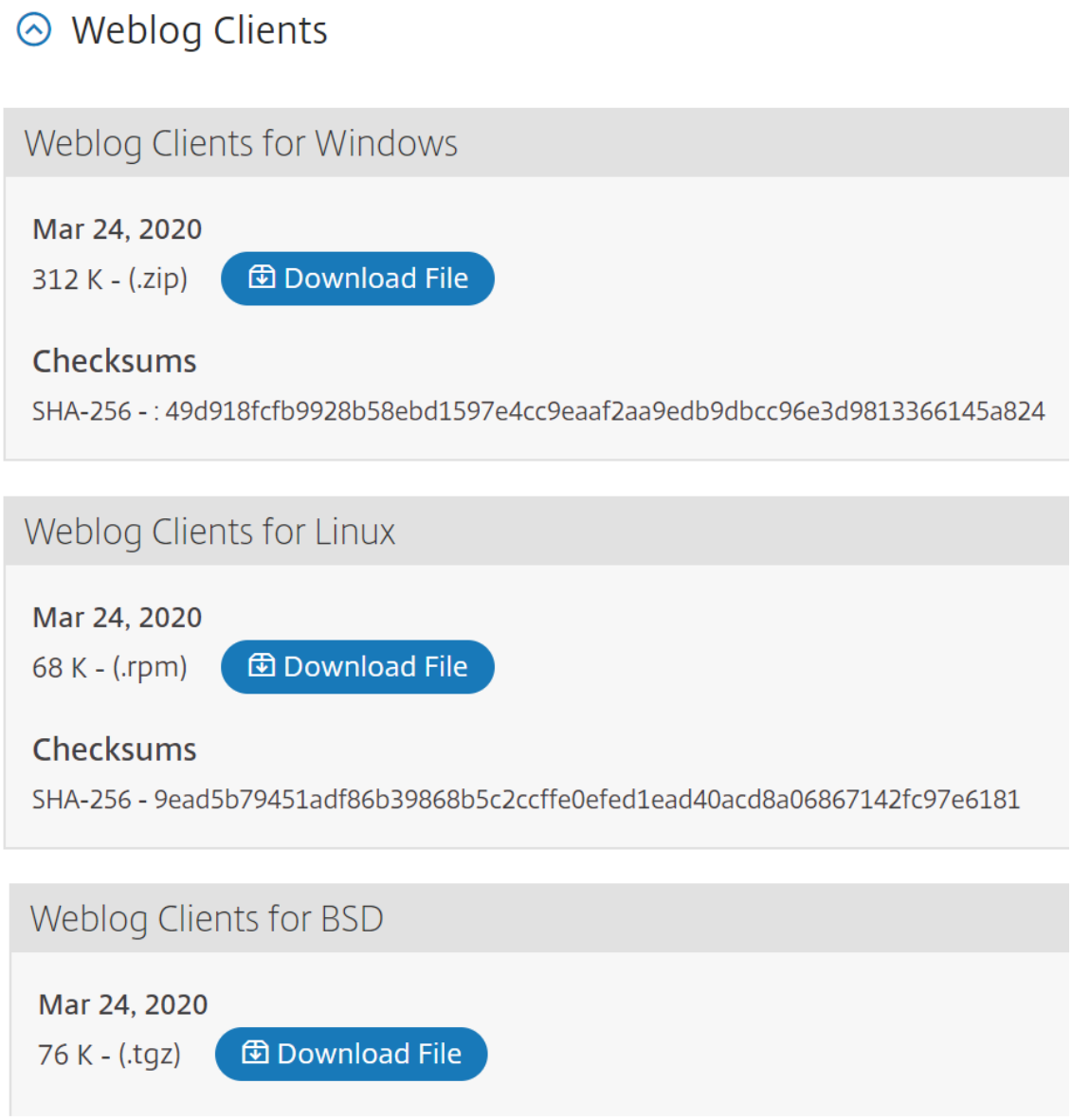
NSWL クライアントパッケージは、NetScaler 製品 CD または NetScaler ダウンロードサイトのいずれかから入手できます。パッケージには、サポートされているプラットフォームごとに個別のインストールパッケージがあります。

Citrix の **Web** サイトから **NSWL** クライアントをダウンロードするには

1. URL `https://www.citrix.com/downloads/citrix-adc/`にアクセスして Citrix にログオンします。
2. 特定の NetScaler リリースバージョンに移動し、そのファームウェアを探します。
3. 「ファームウェア」をクリックします（たとえば、NetScaler リリース（機能フェーズ）13.0 ビルド 52.24）。



4. **NetScaler** リリース（機能フェーズ）ビルドページで、「**Weblog** クライアント」セクションに移動します。
5. このセクションでは、Windows、Linux、および BSD 用のウェブログクライアントをダウンロードできます。



Solaris に **NSWL** クライアントをインストール

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. `nswl_solaris-<release number>-<build number>.tar file` をパッケージから抽出します。
2. 抽出したファイルを、NSWL クライアントをインストールする Solaris システムにコピーします。
3. 次のコマンドを使用して tar ファイルからファイルを抽出します。

   ```
   tar xvf nswl_solaris-9.3-51.5.tar
   ```


一時ディレクトリに Weblog というディレクトリが作成され、ファイルが Welog ディレクトリに抽出されます。

- 以下のコマンドでパッケージをインストールします。

  ```
  pkgadd -d
  ```

- 利用可能なパッケージのリストが表示されます。次の例では、1 つの Welog パッケージが表示されています。

  ```
  1 NSweblog NetScaler Weblogging (SunOS,sparc)7.0
  ```

パッケージを選択するように求められます。インストールするウェブログのパッケージ番号を選択します。

パッケージ番号を選択して **Enter** キーを押すと、ファイルが抽出され、次のディレクトリにインストールされます。

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

1. NSWL パッケージがインストールされているかどうかを確認するには、次のコマンドを実行します。

   ```
   pkginfo | grep NSweblog
   ```

2. NSWL パッケージをアンインストールするには、次のコマンドを実行します。

   ```
   pkgrm NSweblog
   ```

Linux に NSWL クライアントをインストールする

> 重要
>
> Linux に NSWL クライアントをインストールすると、構成ファイルが置き換えられます。インストールする前にバックアップを取らなければなりません。

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. nswl_linux-<release number>-<build number>.rpm パッケージからファイルを抽出します。
2. 抽出したファイルを、NSWL クライアントをインストールする Linux OS を実行しているシステムにコピーします。
3. NSWL パッケージをインストールするには、次のコマンドを実行します。

   ```
   rpm -i nswl_linux-9.3-51.5.rpm
   ```

このコマンドは、ファイルを抽出し、次のディレクトリにインストールします。

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin

- /usr/local/netscaler/samples

1. NSWL パッケージをアンインストールするには、次のコマンドを実行します。

   ```
   rpm -e NSweblog
   ```

2. Weblog RPM ファイルの詳細情報を取得するには、次のコマンドを実行します。

   ```
   rpm -qpi *.rpm
   ```

3. インストールされている Web サーバーのログファイルを表示するには、次のコマンドを実行します。

   ```
   rpm -qpl *.rpm
   ```

FreeBSD に NSWL クライアントをインストールする

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. nswl_bsd-<release number>-<build number>.tgz パッケージからファイルを抽出します。
2. 解凍したファイルを、NSWL クライアントをインストールする FreeBSD OS を実行しているシステムにコピーします。
3. NSWL パッケージをインストールするには、次のコマンドを実行します。

   ```
   pkg_add nswl_bsd-9.3-51.5.tgz
   ```

このコマンドは、ファイルを抽出し、次のディレクトリにインストールします。

```
1 -  /usr/local/netscaler/etc
2 -  /usr/local/netscaler/bin
3 -  /usr/local/netscaler/samples
```

1. NSWL パッケージをアンインストールするには、次のコマンドを実行します。

   ```
   pkg_delete NSweblog
   ```

2. パッケージがインストールされていることを確認するには、次のコマンドを実行します。

   ```
   pkg_info | grep NSweblog
   ```

NSWL クライアントを Mac にインストール

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. nswl_macos-<release number>-<build number>.tgz パッケージからファイルを抽出します。

2. 抽出したファイルを、NSWL クライアントをインストールする macOS を実行しているシステムにコピーします。
3. NSWL パッケージをインストールするには、次のコマンドを実行します。

```
pkg_add nswl_macos-9.3-51.5.tgz
```

このコマンドは、ファイルを抽出し、次のディレクトリにインストールします。

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/ローカル/ネットスケーラー/サンプル

1. NSWL パッケージをアンインストールするには、次のコマンドを実行します。

```
pkg_delete NSweblog
```

2. パッケージがインストールされていることを確認するには、次のコマンドを実行します。

```
pkg_info | grep NSweblog
```

Windows に NSWL クライアントをインストール

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. nswl_win-<release number>-<build number>.zip パッケージからファイルを抽出します。
2. 抽出したファイルを、NSWL クライアントをインストールする Windows システムにコピーします。
3. Windows システムでは、ファイルをディレクトリ (<NSWL-HOME>と呼ばれる) に解凍します。次のディレクトリが抽出されます: /bin、/etc 、/samples
4. コマンドプロンプトで、<NSWL-HOME>\bin directory:から次のコマンドを実行します。

```
nswl -install -f <directorypath>\log.conf
```

各項目の意味は次のとおりです。

ディレクトリパスは、設定ファイル (log.conf) のパスを指します。デフォルトでは、ファイルは <NSWL-HOME>および /etc ディレクトリにあります。設定ファイルは他のディレクトリにコピーできます。

> 注
>
> NSWL クライアントをアンインストールするには、コマンドプロンプトで、<NSWL-HOME>\bin directory:から次のコマンドを実行します。

```
1 > nswl -remove
```


NSWL クライアントを AIX システムにインストールします

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. nswl_aix-<release number>-<build number>.rpm パッケージからファイルを抽出します。
2. 抽出したファイルを、NSWL クライアントをインストールする AIX OS を実行しているシステムにコピーします。
3. NSWL パッケージをインストールするには、次のコマンドを実行します。

   ```
   rpm -i nswl_aix-9.3-51.5.rpm
   ```

このコマンドは、ファイルを抽出し、次のディレクトリにインストールします。

- /usr/local/netscaler/etc
- /usr/local/netscaler/
- usr/local/netscaler/samples

1. NSWL パッケージをアンインストールするには、次のコマンドを実行します。

   ```
   rpm -e NSweblog
   ```

2. Weblog RPM ファイルの詳細情報を取得するには、次のコマンドを実行します。

   ```
   rpm -qpi *.rpm
   ```

3. インストールされている Web サーバーのログファイルを表示するには、次のコマンドを実行します。

   ```
   rpm -qpl *.rpm
   ```

NSWL クライアントの構成

August 15, 2023

NSWL クライアントをインストールしたら、nswl 実行可能ファイルを使用して NSWL クライアントを構成できます。これらの構成は、NSWL クライアント構成ファイル (log.conf) に保存されます。

> 注:
>
> NSWL 構成ファイル（log.conf）にさらに変更を加えることで、NSWL クライアントのロギングをさらにカスタマイズできます。詳細については、「NSWL クライアントシステムでのロギングのカスタマイズ」を参照してください。

次の表では、NSWL クライアントの構成に使用できるコマンドについて説明します。

NSWL コマンド	**Specifies**
nswl-help	使用可能な NSWL ヘルプオプション。
nswl-addns-f <path-to-configuration-file>	ログトランザクションデータを収集するシステム。NetScaler ADC アプライアンスの IP アドレスを入力するように求められます。有効なユーザ名とパスワードを入力します。
nswl -verify -f <path-to-configuration-file>	設定ファイルに構文エラーまたはセマンティックエラーがないかチェックします。
nswl -start -f <path-to-configuration-file>	構成ファイルの設定に基づいて NSWL クライアントを起動します。注:Solaris および Linux の場合:Web サーバーのロギングをバックグラウンドプロセスとして開始するには、コマンドの最後にアンパサンド記号 (&) を入力します。
nswl-stop (ソラリスおよび Linux のみ)	NSWL クライアントがバックグラウンドプロセスとして起動された場合は停止します。それ以外の場合は、Ctrl+C キーを押して Web サーバのロギングを停止します。
nswl-install-f <path-to-configuration-file> (Windows のみ)	Windows で NSWL クライアントをサービスとしてインストールします。
nswl-サービス開始 (Windows のみ)	nswl インストールオプションで指定されている構成ファイルの設定を使用して、NSWL クライアントを起動します。NSWL Client は、［スタート］>［コントロールパネル］**>**［サービス］からも起動できます注:NSWL ログファイルは C:\Windows\SysWOW64. に作成されます
nswl-stopservice (Windows のみ)	NSWL クライアントを停止します。
nswl-削除	NSWL クライアントサービスをレジストリから削除します。

NSWL 実行可能ファイルが配置されているディレクトリから次のコマンドを実行します。

- Windows: \ns\bin
- Solaris and Linux: \usr\local\netscaler\bin

Web サーバーロギング設定ファイルは、次のディレクトリパスにあります。

- Windows: \ns\etc
- Solaris and Linux: \usr\local\netscaler\etc

NSWL 実行可能ファイルはとして起動されます。\ nswl は Linux と Solaris で使えます。

NetScaler ADC アプライアンスの IP アドレスを追加します

NSWL クライアント構成ファイル（log.conf）に、NSWL クライアントがログの収集を開始する NetScaler ADC IP アドレス（NSIP）を追加します。

NetScaler ADC アプライアンスの NSIP アドレスを追加するには

1. クライアントシステムのコマンドプロンプトで、次のように入力します。

   ```
   nswl -addns -f < directorypath > \log.conf
   < directorypath >: Specifies the path to the configuration file (
   log.conf)。
   ```

2. 次のプロンプトで、次の情報を入力します。

   - **NSIP**：NetScaler ADC アプライアンスの IP アドレスを指定します。
   - ユーザー名とパスワード：NetScaler ADC アプライアンスの nsroot ユーザー資格情報を指定します。

     > 注：
     >
     > ロギング権限が有効になっているすべてのシステムユーザがこの機能をサポートします。

> 注：
>
> 複数の NetScaler ADC IP アドレス（NSIP）を追加し、後ですべての NetScaler ADC システムログの詳細をログに記録したくない場合は、log.conf ファイルの最後にある NSIP ステートメントを削除して、NSIP を手動で削除できます。フェイルオーバーのセットアップ中に、コマンドを使用して、プライマリとセカンダリの両方の NetScaler ADC IP アドレスを log.conf に追加する必要があります。IP アドレスを追加する前に、ユーザー名とパスワードが NetScaler ADC アプライアンスに存在することを確認してください。

NSWL 構成ファイルの検証

ロギングが正しく機能することを確認するには、クライアントシステムの NSWL 構成ファイル (log.conf) に構文エラーがないか確認します。

NSWL 構成ファイル内の構成を確認するには

クライアントシステムのコマンドプロンプトで、次のように入力します。

```
nswl -verify -f <directorypath>\log.conf
```

< directorypath>: 設定ファイル (log.conf) へのパスを指定します。

NSWL クライアントを実行する

Web サーバーロギングの開始

クライアントシステムのコマンドプロンプトで、次のように入力します。

```
nswl -start -f <directorypath>\log.conf
```

<directorypath>: 設定ファイル (log.conf) へのパスを指定します。

Solaris または Linux オペレーティングシステムでバックグラウンドプロセスとして開始された Web サーバーロギングの停止

コマンドプロンプトで入力します。

```
nswl -stop
```

Windows オペレーティングシステムでサービスとして開始された Web サーバーのロギングを停止するには

コマンドプロンプトで入力します。

```
nswl -stopservice
```

NSWL クライアントシステムでのログ記録のカスタマイズ

August 15, 2023

NSWL クライアント構成ファイル（log.conf）をさらに変更することで、NetScaler Web ロギング（NSWL）クライアントシステムでのロギングをカスタマイズできます。テキスト・エディタを使用して、クライアント・システム上の log.conf 構成ファイルを変更します。

ロギングをカスタマイズするには、設定ファイルを使用してフィルタとログプロパティを定義します。

- ログフィルタ。Web サーバーのホスト IP アドレス、ドメイン名、およびホスト名に基づいてログ情報をフィルタリングします。
- ログプロパティ。各フィルタには、関連する一連のログプロパティがあります。ログプロパティでは、フィルタリングされたログ情報の保存方法を定義します。

サンプル設定ファイル

次に、設定ファイルの例を示します。

```
1 ##########
2 # This is the NSWL configuration file
3 # Only the default filter is active
4 # Remove leading # to activate other filters
5 ##########
6 ##########
7 # Default filter (default on)
8 # W3C Format logging, new file is created every hour or on reaching 10
     MB file size,
```



```
# and the file name is Exyymmdd.log
##########
Filter default
begin default
        logFormat               W3C
        logInterval             Hourly
        logFileSizeLimit        10
        logFilenameFormat       Ex%`{
 `%y%m%d }
 t.log
end default
##########
# NetScaler caches example
# CACHE_F filter covers all the transaction with HOST name www.
    netscaler.com and the listed server ip's
##########
#Filter CACHE_F HOST www.netscaler.com IP 192.168.100.89 192.168.100.95
     192.168.100.52 192.168.100.53 ON
##########
# netscaler origin server example
# Not interested in Origin server to Cache traffic transaction logging
##########
#Filter ORIGIN_SERVERS IP 192.168.100.64 192.168.100.65 192.168.100.66
    192.168.100.67 192.168.100.225 192.168.100.226 192.168.
100.227 192.168.100.228 OFF
##########
# netscaler image server example
# all the image server logging.
##########
#Filter IMAGE_SERVER HOST www.netscaler.images.com IP 192.168.100.71
    192.168.100.72 192.168.100.169 192.168.100.170 192.168.10
0.171 ON
##########
# NCSA Format logging, new file is created every day midnight or on
    reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.
    log.
# Exclude objects that ends with .png .jpg .jar.
##########
#begin ORIGIN_SERVERS
#       logFormat               NCSA
#       logInterval             Daily
#       logFileSizeLimit        40
#       logFilenameFormat       /datadisk5/ORGIN/log/%v/NS%`{
 `%m%d%y }
 t.log
#       logExclude              .png .jpg .jar
#end ORIGIN_SERVERS

##########
# NCSA Format logging, new file is created every day midnight or on
    reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.
```



```
      log with log record timestamp as GMT.
55 ##########
56 #begin CACHE_F
57 #       logFormat               NCSA
58 #       logInterval             Daily
59 #       logFileSizeLimit        20
60 #       logFilenameFormat /datadisk5/netscaler/log/%v/NS%`{
61  `%m%d%y }
62  t.log
63 #       logtime                 GMT
64 #end CACHE_F
65
66 ##########
67 # W3C Format logging, new file on reaching 20MB and the log file path
      name is
68 # atadisk6/netscaler/log/server's ip/Exmmyydd.log with log record
      timestamp as LOCAL.
69 ##########
70 #begin IMAGE_SERVER
71 #       logFormat               W3C
72 #       logInterval             Size
73 #       logFileSizeLimit        20
74 #       logFilenameFormat /datadisk6/netscaler/log/%AEx%`{
75  `%m%d%y }
76  t
77 #       logtime                 LOCAL
78 #end IMAGE_SERVER
79
80 ##########
81 # Virtual Host by Name firm, can filter out the logging based on the
      host name by,
82 ##########
83
84 #Filter VHOST_F IP 10.101.2.151 NETMASK 255.255.255.0
85 #begin VHOST_F
86 #       logFormat               W3C
87 #       logInterval             Daily
88 #       logFileSizeLimit        10
89 logFilenameFormat /ns/prod/vhost/%v/Ex%`{
90  `%m%d%y }
91  t
92 #end VHOST_F
93
94 ########## END FILTER CONFIGURATION ##########
95 <!--NeedCopy-->
```

フィルターの作成

設定ファイル (log.conf) でデフォルトのフィルタ定義を使用するか、フィルタを変更するか、フィルタを作成することもできます。複数のログフィルタを作成できます。

> 注:
>
> フィルタが定義されていないトランザクションをログに記録する統合ロギングは、有効な場合にデフォルトのフィルタを使用します。すべてのサーバーの統合ログは、デフォルトのフィルタのみを定義することによって実行できます。

サーバーが複数の Web サイトをホストし、各 Web サイトに独自のドメイン名があり、各ドメインが仮想サーバーに関連付けられている場合は、Web サーバーログを構成して Web サイトごとに個別のログディレクトリを作成できます。次の表に、フィルタを作成するためのパラメータを示します。

表 1. フィルターを作成するためのパラメータ

パラメーター	Specifies
filterName	フィルタの名前。フィルタ名には英数字を含めることができ、59 文字を超えることはできません。59 文字を超えるフィルタ名は 59 文字に切り捨てられます。
ホスト名	トランザクションがログに記録されるサーバーのホスト名。
IP `ip`	トランザクションがログに記録されるサーバーの IP アドレス (たとえば、サーバーに 1 つの IP アドレスを持つドメインが複数ある場合)。
IP ip 2…ip N	複数の IP アドレス（たとえば、サーバドメインに複数の IP アドレスがある場合)。
ip6 IP	トランザクションがログに記録されるサーバの IPv6 アドレス。
IP ip NETMASK mask	サブネットで使用される IP アドレスとネットマスクの組み合わせ。
`ON` | `OFF`	フィルタを有効または無効にしてトランザクションをログに記録します。引数が選択されていない場合、フィルタは有効 (オン) になります。

フィルタを作成するには、log.conf ファイルに次のコマンドを入力します。

- `filter <filterName> <HOST name> | [IP<ip> ] | [IP<ip 2...ip n> ] | <IP ip NETMASK mask> [ON | OFF]`
- `filter <filterName> <HOST name> | [IP6 ip/<prefix length>] [ON | OFF]`

仮想サーバーのフィルターを作成する

仮想サーバ用のフィルタを作成するには、log.conf ファイルに次のコマンドを入力します。


```
filter <filterName> <VirtualServer IP address>
```

例

次の例では、IP アドレス 192.168.100.0、ネットマスク 255.255.255.0 を指定しています。このフィルタは、192.168.100.1 から 192.168.100.254 までの IP アドレスに適用されます。

```
Filter F1 HOST www.netscaler.com ON
Filter F2 HOST www.netscaler.com IP 192.168.100.151 ON
Filter F3 HOST www.netscaler.com IP 192.168.100.151 192.165.100.152 ON
Filter F4 IP 192.168.100.151
Filter F5 IP 192.168.100.151 HOST www.netscaler.com OFF
Filter F6 HOST www.netscaler.com HOST www.xyz.com HOST www.abcxyz.com
   IP 192.168.100.200 ON
Filter F7 IP 192.250.100.0 NETMASK 255.255.255.0
Filter F8 HOST www.xyz.com IP 192.250.100.0 NETMASK 255.255.255.0 OFF
For creating filters for servers having IPv6 addresses.
Filter F9 2002::8/112 ON
Filter F10 HOST www.abcd.com IP6 2002::8 ON

<!--NeedCopy-->
```

ログプロパティを指定する

ログプロパティは、フィルタに関連付けられているすべてのログエントリに適用されます。次の例に示すように、ログプロパティの定義はキーワードBEGINで始まり、ENDで終わります。

```
BEGIN <filtername>
 logFormat ...
 logFilenameFormat ...
 logInterval ...
 logFileSize ....
 logExclude ....
 logTime ….
 END
<!--NeedCopy-->
```

定義には、次の項目を含めることができます。

- **LogFormat** は、NCSA、W3C 拡張、およびカスタムログファイル形式をサポートする Web サーバーのログ機能を指定します。

デフォルトでは、logformatプロパティは w3c です。上書きするには、設定ファイルに custom または NCSA と入力します。次に例を示します。

```
LogFormat NCSA
<!--NeedCopy-->
```


> 注:
>
> NCSA およびカスタムログ形式では、トランザクションのタイムスタンプとファイルのローテーションにローカル時間が使用されます。

- **LogInterval** は、新しいログファイルが作成される間隔を指定します。次のいずれかの値を使用します。
  - Hourly: 1 時間ごとにファイルが作成されます。
  - 毎日: 毎日午前 0 時にファイルが作成されます。デフォルト値です。
  - Weekly: 毎週日曜日の午前 0 時にファイルが作成されます。
  - 毎月: 月の初日の午前 0 時にファイルが作成されます。
  -

例:

```
LogInterval Daily
<!--NeedCopy-->
```

LogFileSizeLimit は、ログファイルの最大サイズを MB 単位で指定します。任意のログ間隔 (毎週、毎月など) で使用できます。ファイルは、最大ファイルサイズ制限に達したとき、または定義されたログ間隔が経過したときに作成されます。

この動作をオーバーライドするには、ログファイルのサイズ制限に達したときにのみファイルが作成されるように、サイズを`logintervalプロパティとして指定します。

デフォルトの`LogFileSizeLimitは 10 MB です。

例:

```
LogFileSizeLimit 35
<!--NeedCopy-->
```

- **LogFileNameFormat** は、ログファイルのファイル名形式を指定します。ファイル名には次の種類があります。
  - Dynamic: 次の形式を含むエクスプレッションを指定します。
    - サーバ IP アドレス
    - 日付 (% {フォーマット} t)
    - URL サフィックス (%x)
    - ホスト名 (%v)

例:

```
LogFileNameFormat Ex%`{
 `%m%d%y }
 t.log
<!--NeedCopy-->
```


このコマンドは、最初のファイル名を Exmmddyy.log として作成し、1 時間ごとにファイル名でファイルを作成します。exmmddyy.log.0、exmmDDYY.log.1、…、exmmDDYY.log.n。

例:

```
      LogInterval size
      LogFileSize 100
      LogFileNameFormat Ex%`{
 `%m%d%y }
 t
<!--NeedCopy-->
```

> 注意:
>
> LogFilenameFormat コマンドで指定された日付形式%t は、そのフィルタのログ間隔プロパティを上書きします。指定したログファイルのサイズに達したときではなく、毎日新しいファイルが作成されないようにするには、LogFilenameFormat で%t を使用しないでください。

- **LogExclude** は、指定したファイル名拡張子を持つトランザクションのログを防止します。

例:

```
LogExclude.html
<!--NeedCopy-->
```

このコマンドは、*.html ファイルのログトランザクションを除外したログファイルを作成します。

LogTime は、ログ時間を GMT または LOCAL として指定します。

既定値は次のとおりです。

- NCSA ログファイル形式:LOCAL
- W3C ログファイル形式:GMT

NCSA と **W3C** のログ形式を理解する

NetScaler は、次の標準ログファイル形式をサポートしています。

- NCSA 共通ログ形式
- W3C 拡張ログ形式

NCSA 共通ログ形式

ログファイル形式が NCSA の場合、ログファイルには次の形式でログ情報が表示されます。

```
Client_IP_address -User_Name [Date:Time -TimeZone] "Method Object
    HTTP_version" HTTP_StatusCode BytesSent
<!--NeedCopy-->
```


NCSA 共通ログ形式を使用するには、`log.conf` ファイルの LogFormat 引数に **NCSA** と入力します。

次の表に、NCSA Common ログの形式を示します。

引数	Specifies
Client_ip_address	クライアントコンピュータの IP アドレス。
ユーザー名	ユーザー名。
日付	トランザクションの日付。
時間	トランザクションが完了した時刻。
タイムゾーン	タイムゾーン (グリニッジ標準時または現地時間)。
方法	リクエストメソッド (GET、POST など)。
オブジェクト	URL。
HTTP_version	クライアントが使用する HTTP のバージョン。
HTTP_StatusCode	レスポンスのステータスコード。
送信済みバイト数	サーバから送信されたバイト数。

W3C 拡張ログ形式

拡張ログファイルには、ラインフィード (LF) またはシーケンスのキャリッジリターンラインフィード (CRLF) のいずれかで終了する ASCII 文字を含む一連の行が含まれます。ログファイルジェネレータは、そのジェネレータが実行されるプラットフォームの行終了規則に従う必要があります。

ログアナライザは、LF 形式または CRLF 形式を受け入れる必要があります。各行には、ディレクティブまたはエントリのいずれかを含めることができます。W3C 拡張ログ形式を使用する場合は、log.conf ファイルの Log-Format 引数として W3C と入力します。

既定では、標準の W3C ログ形式は、次のようにカスタムログ形式として内部的に定義されています。

```
%`{
 `%Y-%m-%d%H:%M:%S }
 t %a %u %S %A %p %m %U %q %s %j %J %T %H %+{
 user-agent }
 i %+{
 cookie }
  i%+{
 referer }
 i
<!--NeedCopy-->
```

この W3C ログ形式では、順序を変更したり、一部のフィールドを削除したりすることもできます。次に例を示します:


```
logFormat W3C %`{
 `%Y-%m-%d%H:%M:%S }
 t %m %U
<!--NeedCopy-->
```

W3C ログエントリは、次の形式で作成されます。

```
#Version: 1.0
#Fields: date time cs-method cs-uri
#Date: 12-Jun-2001 12:34
2001-06-12 12:34:23 GET /sports/football.html 2001-06-12 12:34:30
GET /sports/football.html
<!--NeedCopy-->
```

エントリー

エントリは、単一の HTTP トランザクションに関連する一連のフィールドで構成されます。フィールドは空白で区切られます。タブ文字の使用することをお勧めします。特定のエントリのフィールドが使用されていない場合、省略されたフィールドはダッシュ（-）でマークされます。

ディレクティブ

ロギングプロセスの詳細については、ディレクティブ （Directives）の表を参照してください。シャープ記号 (#) で始まる行には、ディレクティブが含まれます。

例**:**

次のサンプルログファイルには、W3C Extended ログ形式でログエントリが表示されます。

```
#Version: 1.0
#Fields: time cs-method cs-uri
#Date: 12-Jan-1996 00:00:00
00:34:23 GET /sports/football.html
12:21:16 GET /sports/football.html
12:45:52 GET /sports/football.html
12:57:34 GET /sports/football.html
<!--NeedCopy-->
```

フィールド

Fields ディレクティブは，各エントリに記録される情報を指定する一連の項目識別子をリストします。フィールド識別子には、次のいずれかの形式があります。

- **identifier**：トランザクション全体に関連します。
- **prefix-identifier:** 値プレフィックスによって定義されるパーティ間の情報転送に関連します。

- **prefix (header):** 値プレフィックスで定義されるパーティ間の転送用の HTTP ヘッダーフィールドヘッダーの値を指定します。この方法で指定されたフィールドには、常に型があります。

次の表に、定義済みの接頭辞を示します。

前	Specifies
c	Client
s	サーバー
r	リモート
cs	クライアントからサーバへ
sc	サーバからクライアントへ
sr	サーバーからリモートサーバー (プロキシが使用するプレフィックス)
rs	リモートサーバーからサーバー (プロキシが使用するプレフィックス)
x	アプリケーション固有識別子

例:

次の例は、接頭辞を使用する定義済み識別子です。

cs-method: クライアントからサーバーに送信されるリクエストのメソッド。

sc (リファラー**):** 返信の`Referer`フィールド。

c-ip: クライアントの IP アドレス。

[識別子**]**

次の表に、プレフィックスを必要としない W3C 拡張ログ形式 ID を示します。

識別子	説明
日付	トランザクションが行われた日付。
時間	トランザクションが完了した時刻。
時間かかった	トランザクションの完了にかかった時間 (秒単位)。
bytes	転送されたバイト数。
キャッシュされます	キャッシュヒットが発生したかどうかを記録します。ゼロはキャッシュミスを示します。

次の表に、プレフィックスが必要な W3C 拡張ログ形式 ID を示します。

識別子	説明
IP	IP アドレスとポート番号。
DNS	DNS 名。
状態	ステータスコード。
comment	コメントはステータスコードとともに返されました。
method	メソッド。
url	URL。
url-stem	URL の語幹部分。
URL クエリ	URL のクエリ部分。

W3C 拡張ログファイル形式では、ログフィールドを選択できます。これらのフィールドを次の表に示します。

フィールド	説明
日付	トランザクションが完了した日付。
時間	トランザクションが完了した時刻。
クライアント IP	クライアントの IP アドレス。
ユーザー名	ユーザー名。
サービス名	サービス名。常に HTTP です。
サーバー IP	サーバの IP アドレス。
サーバーポート	サーバのポート番号
方法	リクエストメソッド (GET、POST など)。
URL ステム	URL ステム。
URL クエリー	URL のクエリ部分。
HTTP ステータス	レスポンスのステータスコード。
送信済みバイト数	サーバーに送信されたバイト数 (HTTP ヘッダーを含むリクエストサイズ)。
受信バイト数	サーバーから受信したバイト数 (HTTP ヘッダーを含むレスポンスサイズ)。
Time Taken	トランザクションが完了するまでの時間（秒）。
Protocol Version	クライアントが使用している HTTP のバージョン番号。

フィールド	説明
ユーザー エージェント	HTTP プロトコルの **User-Agent** フィールド。
クッキー	HTTP プロトコルの **Cookie** フィールド。
Referer	HTTP プロトコルのReferer フィールド。

カスタムログ形式を作成する

ログファイルデータの表示形式は、手動で、または NSWL ライブラリを使用してカスタマイズできます。カスタムログ形式を使用すると、Apache が現在サポートしているほとんどのログ形式を取得できます。

NSWL ライブラリを使用してカスタムログ形式を作成する

NSWL 実行可能ファイルが Windows または Solaris のホスト・コンピュータにインストールされているかどうかに応じて、次の NSWL ライブラリのいずれかを使用します。

- **Windows:** システムマネージャホストコンピュータ上の\ns\binディレクトリにある nswl.lib ライブラリ。
- **Solaris:** usr/local/netscaler/binのlibnswl.a NET のライブラリ。

1. システムで定義された次の 2 つの C 関数を C ソースファイルに追加します。

   ns_userDefFieldName (): この関数は、ログレコードにカスタムフィールド名として追加する必要がある文字列を返します。

   ns_userDefFieldVal (): この関数は、カスタムフィールド値を実装し、それを文字列として返します。この値はログレコードの最後に追加する必要があります。

2. ファイルをオブジェクトファイルにコンパイルします。
3. オブジェクトファイルを NSWL ライブラリ (およびオプションでサードパーティのライブラリ) にリンクして、新しい NSWL 実行可能ファイルを作成します。
4. 設定ファイル (log.conf) の LogFormat 文字列の最後に%d 文字列を追加します。

例**:**

```
##########
# A new file is created every midnight or on reaching 20MB file size,
# and the file name is
/datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log and create
digital
#signature field for each record.
BEGIN CACHE_F
    logFormat custom "%a - "%{
```



```
 user-agent }
 i" [%d/%B/%Y %T -%g] "%x"
%s %b%{
 referrer }
 i "%{
 user-agent }
 i" "%{
 cookie }
 i" %d "
    logInterval Daily
    logFileSizeLimit 20
    logFilenameFormat
/datadisk5/netscaler/log/%v/NS%`{
 `%m%d%y }
 t.log
END CACHE_F
<!--NeedCopy-->
```

カスタムログ形式を手動で作成する

ログファイルデータを表示する形式をカスタマイズするには、**LogFormat** ログプロパティ定義の引数として文字列を指定します。次に、文字列を使用してログ形式を作成する例を示します。

```
LogFormat Custom ""%a - "%{
 user-agent }
 i" %[%d/%m/%Y]t %U %s %b %T"
<!--NeedCopy-->
```

- 文字列には、新しい行とタブを表す「c」タイプの制御文字\ n と\ t を含めることができます。
- リテラル引用符とバックスラッシュで Esc キーを使用します。

リクエストの特性は、フォーマット文字列に% ディレクティブを配置することで記録されます。このディレクティブは、ログファイルでは値で置き換えられます。

%v (ホスト名) または%x (URL サフィックス) の形式指定子がログファイル名の形式文字列に存在する場合、ファイル名の次の文字がログ設定ファイル名のアンダースコア記号に置き換えられます。

```
" *. /: < > ? \ |
```

ASCII 値が 0 ～31 の範囲にある文字は、次のように置き換えられます。

```
%<ASCII value of character in hexadecimal>。
```

たとえば、ASCII 値 22 の文字は%16 に置き換えられます。

> 注意:
>
> %v 形式指定子がログファイル名の形式文字列に存在する場合、仮想ホストごとに個別のファイルが開きます。継続的なログ記録を確保するには、プロセスが開くことができるファイルの最大数が十分に大きい必要があります。開くことができるファイル数を変更する手順については、オペレーティングシステムのマニュアルを参

> 照してください。

Apache ログ形式を作成する

Apache が現在サポートしているほとんどのログ形式をカスタムログから引き出すことができます。Apache ログ形式に一致するカスタムログ形式は次のとおりです。

NCSA/combined: LogFormat custom %h %l %u [%t] “%r” %s %B “%{referer}i” “%{user-agent}i”

NCSA/Common: LogFormat custom %h %l %u [%t] “%r” %s %B

Referer Log: LogFormat custom “%{referer}i” -> %U

User agent: LogFormat custom %{user-agent}i

同様に、カスタム形式から他のサーバーログ形式を派生させることができます。

カスタムログ形式を定義するための引数

次の表に、カスタムログ形式を示します。

引数	指定
%a	リモート IPv4 アドレス。
%A	ローカル IPv4 アドレス。
%a6	リモート IPv6 アドレス。
%A6	ローカル IPv6 アドレス。
%B	HTTP ヘッダー (応答サイズ) を除いた送信バイト数。
%b	HTTP ヘッダー (要求サイズ) を除いた受信バイト数。
%d	ユーザー定義フィールド。
%K	クライアントポート情報。
%e1	最初のカスタム HTTP リクエストヘッダーの値。
%e2	2 番目のカスタム HTTP リクエストヘッダーの値。
%E1	最初のカスタム HTTP 応答ヘッダーの値。
%E2	2 番目のカスタム HTTP 応答ヘッダーの値。注：カスタム HTTP ヘッダーをエクスポートする方法については、「Web サーバーログ用の NetScaler の構成」を参照してください。
%g	グリニッジ標準時のオフセット (たとえば、太平洋標準時は -0800)。

引数	指定
%h	リモートホストの IPv4 アドレス。
%h6	リモートホストの IPv6 アドレス。
H	要求プロトコル。
% {Foobar}i	サーバーに送信されたリクエストの Foobar: ヘッダー行の内容。システムは、ユーザーエージェント、リファラー、Cookie ヘッダーをサポートしています。この形式の% の後ろの + は、ロギングクライアントに、+ を単語の区切り文字として使用するように通知します。
%j	ヘッダー (要求サイズ) を含む受信バイト数。
%J	ヘッダー (応答サイズ) を含む送信バイト数。
%l	リモートログ名 (指定されている場合 identd から)。
%m	リクエストメソッド。
%M	リクエストを処理するのにかかった時間 (マイクロ秒単位)。
% {Foobar}o	Foobar の内容: 返信のヘッダー行。USER-AGENT、リファラー、および Cookie ヘッダー (セット Cookie ヘッダーを含む) がサポートされています。
%p	リクエストを処理するサーバーの標準ポート。
%P	管理パーティション。
%q	クエリ文字列 (疑問符 (?) クエリ文字列が存在する場合)。
%r	要求の最初の行。
%s	内部的にリダイレクトされたリクエスト。これは元のリクエストのステータスです。
%t	時間、一般的なログ形式 (標準の英語の時間形式)。
% {format}t	format で与えられる形式の Time は strftime (3) フォーマットでなければならない。フォーマットの説明については、時間フォーマットの定義を参照してください。
%T	リクエストの処理にかかった時間 (秒単位)。
%u	リモートユーザ (認証から。リターンステータス (%s) が 401 の場合は偽物かもしれない)。
%U	URL パスが要求されました。
%v	リクエストを処理するサーバーの正規名。

引数	指定
%V6	負荷分散、コンテンツスイッチング、キャッシュリダイレクトが使用されている場合、システム内の仮想サーバー IPv6 アドレス。
%D	HTTP トランザクション ID を出力します。
%L	トランザクション時間 (ミリ秒)。
%R	ステータスコードにマップされた HTTP Reason 文字列。
%f	ソースポートロギング。
%V	仮想サーバーの IPv4 アドレス。

> 注
>
> カスタム HTTP ヘッダーをエクスポートする方法については、「Web サーバーロギング用の NetScaler ADC 構成」を参照してください。

たとえば、ログ形式を%+{ user-agent } iとして定義し、ユーザーエージェントの値が NetScaler ADC システムの Web クライアントの場合、情報は NetScaler ADC システム +Web+ クライアントとして記録されます。代わりに、二重引用符を使用することもできます。たとえば、「% {user-agent} i」は、それを「NetScaler ADC システム Web クライアント」として記録します。%.. .r, %. . .i および%. . .o の文字列には\<Esc\>キーを使用しないでください。これは、共通ログ形式の要件に準拠しています。クライアントはログに制御文字を挿入できます。したがって、raw ログファイルを操作するときは注意が必要です。

時間形式の定義

次の表は、カスタムログ形式の表に記載されている%{ format } t文字列の形式部分について知るための時間形式の定義を示しています。角括弧 ([]) 内の値は、表示される値の範囲を示します。たとえば、次の表の%d の説明の [1,31] は、1 から 31 までの%d の範囲を示しています。

引数	指定
———	——-
%%	% と同じです。
%a	ロケールの曜日の略称です。
%A	ロケールの曜日のフルネーム
%b	ロケールの月の略称です。
%B	ロケールの月のフルネーム。
%C	世紀番号 (年を 100 で割り、10 進数として整数に切り捨てる [1,99])。1 桁の先頭には 0 が付きます。
%d	ユーザー定義フィールド。

%K	世紀番号 (年を 100 で割り、10 進数として整数に切り捨てる [1,99])。1 桁の先頭には 0 が付きます。
%e	月の日 [1,31]。一桁の数字の前に空白が付きます。
%h	ロケールの月の略称です。
%H	時 (24 時間制) [0,23]。1 桁の前に 0 が付きます。
%I	時 (12 時間表記) [1,12]。1 桁の前に 0 が付きます。
%j	年の日の数字 [1,366]。一桁の前に 0 が付きます。
%k	時 (24 時間制) [0,23]。一桁の前に空白が付きます。
%l	時 (12 時間制) [1,12]。一桁の前に空白が付きます。
%m	年の月の番号 [1,12]。一桁の前に 0 が付きます。
%M	分 [00,59]; 先頭の 0 は許可されますが、必須ではありません。
%n	新しい行を挿入します。
%p	ロケールの午前または午後に相当します。
%r	%p を含む 12 時間形式の適切な時間表現
%S	秒 [00,61]。値の範囲は、たまのうるう秒と二重うるう秒を許容するため[00,59]ではなく[00,61]です。
%3	ミリ秒 [000,999]。値の範囲は[000,999]です。
%6	マイクロ秒 [000000,999999]。値の範囲は[000000,999999]です。
%9	ナノ秒 [000000000,999999999]。値の範囲は[000000000,999999999]です。
%t	タブを挿入します。
%u	10 進数で表した曜日 [1,7]。1 は
日曜日、2 は火曜日などを表します。|
%U	年の週の番号を 10 進数[00,53]で表し、日曜日を第 1 週の最初の日とします。

> 注:
>
> 前の表で説明した変換のいずれにも対応しない変換を指定した場合、または次の段落で示した変換指定を変更した場合、動作は未定義になり、0 が返されます。

%Uと%W（および変更されたコンバージョン%OUと%OWの間）の差は、週の最初の日と見なされる日です。週番号 1 は、1 月の最初の週です（%Uは日曜日、%Wの場合は月曜日から始まる）。週番号 0 には、%Uおよび%Wの 1 月の第 1 日曜日または月曜日の前の日が含まれます。

サーバーログを表示する

NSWL 機能を構成して、コンソールにサーバーログを表示したり、サーバーログを NetScaler ADC アプライアンス上のディレクトリにリダイレクトしたりできます。

コンソールにログを表示する方法は 2 つあります (標準出力)。
オプション 1: コンソールにすべてのログを表示する。
オプション 2: 選択したログのみ`logfilenameformat`を`STDOUT`で絞り込んでコンソールに表示します。

Call Home

August 15, 2023

ソフトウェアまたはハードウェアの問題が原因で、アプライアンスが正常に動作しないことがあります。このような場合、NetScaler は、お客様のサイトに潜在的な影響が生じる前に、データを収集して問題を解決する必要があります。NetScaler アプライアンスで Call Home を有効にすることで、エラー通知プロセスを自動化できます。サポートチームが問題をトラブルシューティングする前に、NetScaler サポートに電話したり、サービスリクエストを送信したり、システムデータをアップロードしたりする必要がなくなるだけでなく、サポートは問題が発生する前に問題を特定して対処できます。Call Home はアプライアンスを定期的に監視し、Citrix テクニカルサポートサーバーにデータを自動的にアップロードします。さらに、受信した Call Home データから、NetScaler の使用状況に関する洞察が得られます。Citrix 内の複数のチームがこのデータを使用して、NetScaler の設計、サポート、および実装を改善できます。

デフォルトでは、Call Home はすべてのプラットフォームと NetScaler（MPX、VPX、SDX）のすべてのフレーバーで有効になっています。この機能を有効にすると、NetScaler ADC の展開とテレメトリデータを収集して、実装とサポートサービスを向上させることができます。

> 注
>
> また、Call Home に関する詳細については、Call Home FAQ ページを参照してください。

長所

Call Home には次の利点があります。

- ハードウェアおよびソフトウェアのエラー状態を監視します。詳細については、「重大なエラー状態の監視」セクションを参照してください。
- ネットワークに影響する重大なイベントを通知します。
- パフォーマンスデータとシステム使用率の詳細を Citrix に送信して、次の宛先に送信します。
  - 製品の品質を分析し、改善します。
  - リアルタイムのトラブルシューティング情報を提供して、問題をプロアクティブに特定し、問題を迅速に解決します。

プラットフォームサポート

Call Home 機能は、すべての NetScaler プラットフォームとすべてのアプライアンスモデル（MPX、VPX、SDX）でサポートされています。

- NetScaler MPX: すべての MPX モデル。

- NetScaler VPX：外部または中央のライセンスプールからライセンスを取得する VPX アプライアンスを含むすべての VPX モデル。
- NetScaler SDX: ディスクドライブと割り当てられた SSL チップを監視し、エラーや障害がないか確認します。ただし、VPX インスタンスは電源ユニット (PSU) にアクセスできないため、ステータスは監視されません。SDX プラットフォームでは、Call Home を個々のインスタンスに直接設定することも、SVM を介して設定することもできます。

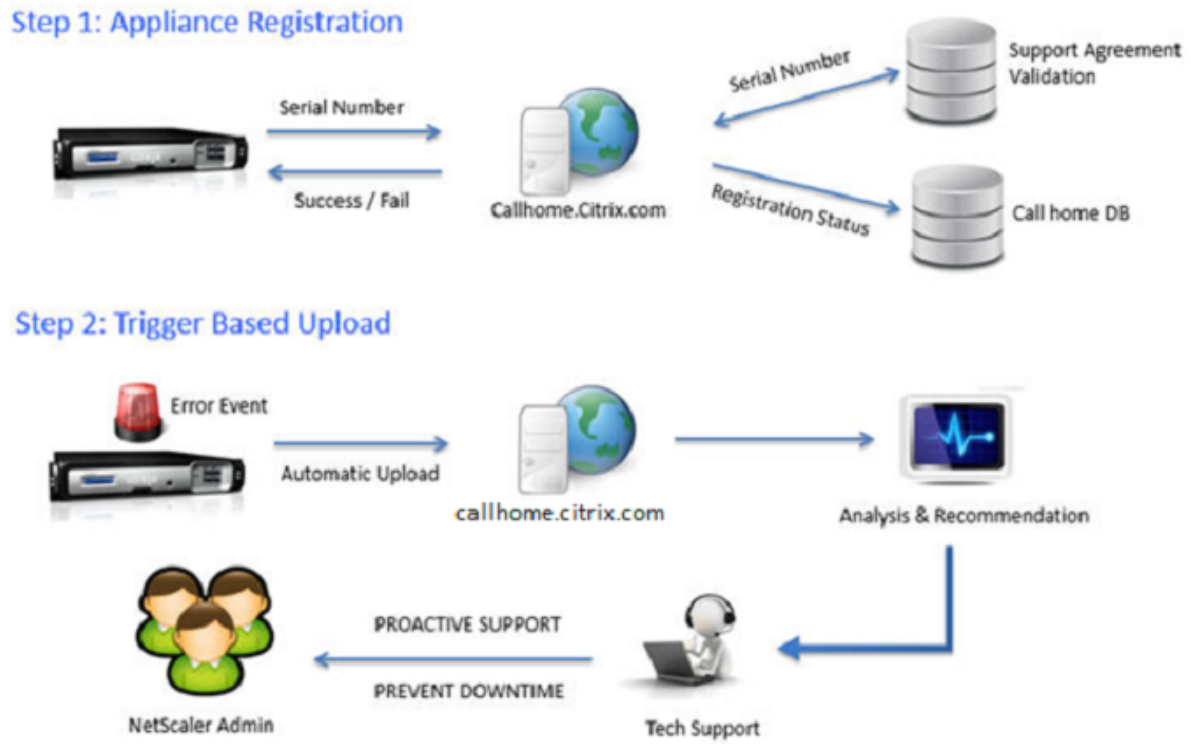
前提条件

Call Home を使用するには、NetScaler アプライアンスに次のものが必要です。

- インターネット接続。Call Home では、NetScaler が NetScaler サポートサーバーに接続してデータアーカイブをアップロードするためのインターネット接続が必要です。
- **URL**。Call Home は、双方向トラフィック用にポート 443 を使用して`callhome.citrix.com` over SSL/TLS プロトコルでトラフィックを交換することで機能します。

Call Home の仕組み

次の図は、お客様のサイトに展開された NetScaler ADC アプライアンスでの Call Home の基本的なワークフローを示しています。



Call Home のワークフローは次のとおりです。

1. インターネット接続をセットアップします。Call Home でシステムデータをアップロードするには、アプライアンスにインターネット接続が必要です。そうでない場合は、インターネット接続を提供するようにプロキシサーバ設定を構成できます。詳細については、「Call Home の設定」の項を参照してください。

2. Call Home を有効にします。NetScaler のコマンドインターフェイスまたは GUI を使用してアプライアンスを最新のソフトウェアにアップグレードすると、Call Home はデフォルトで有効になり、システムは登録プロセスを 24 時間遅らせます。この間、この機能を手動で無効にすることもできますが、Citrix では有効にしておくことをお勧めします。

> 注
>
> Call Home が明示的に無効になっている古いバージョンからアプライアンスをアップグレードする場合、システムはデフォルトで引き続きこの機能を有効にし、初回ログイン時に通知メッセージを表示します。

また、インターネット接続の設定を変更する場合は、Call Home を無効にしてから有効にする必要があります。これにより、Call Home は障害エラーなしで Citrix Insight Services（CIS）サーバーに登録できます。

3. NetScaler アプライアンスを **NetScaler** サポートサーバーに登録します。Call Home がアプライアンスを NetScaler サポートサーバーに登録すると、サーバーはデータベースでアプライアンスのシリアル番号の有効性を確認します。シリアル番号が有効な場合、サーバはアプライアンスを Call Home サービスに登録し、登録に成功した応答を送信します。それ以外の場合、サーバは登録失敗メッセージを返信します。基本的なシステム情報は、個別のメッセージとして送信されます。このデータには、メモリと CPU 使用率の詳細とスループット数が含まれます。デフォルトでは、データは 7 日ごとにハートビートメッセージの一部として定期的に送信されます。ただし、頻繁なアップロードは役に立たないため、5 日未満の値は推奨されません。

4. 重大なエラー状態を監視します。登録が完了すると、Call Home はアプライアンスの監視を開始します。次の表に、Call Home がアプライアンスで監視できる状態を示します。

クリティカルエラー状態	説明	Call Home モニタリング間隔	対応する SNMP アラーム名
コンパクトフラッシュドライブのエラー	アプライアンスの Compact Flash ドライブで、読み取りまたは書き込みのエラーが発生しました。	24 時間	COMPACT-FLASH-ERRORS
ハードディスクドライブのエラー	アプライアンスのハードドライブで読み取りまたは書き込みの障害が発生しました。	24 時間	HARD-DISK-DRIVE-ERRORS
電源装置に障害が発生しました	NetScaler アプライアンスの電源ユニットの 1 つに障害が発生しました。	7 秒	POWER-SUPPLY-FAILURE

クリティカルエラー状態	説明	Call Home モニタリング間隔	対応する SNMP アラーム名
SSL カード障害	NetScaler アプライアンスの SSL カードの 1 つに障害が発生しました。	7 秒	SSL-CARD-FAILED
ウォームリスタート	システムプロセスに障害が発生したため、アプライアンスはウォームリスタートしました。	NetScaler アプライアンスを再起動するたびに。	WARM-RESTART-EVENT
メモリ異常エラー	メモリ使用率は、通常の制限を超えて徐々に増加し、しきい値を超えます。	1 日	SNMP アラームなし
レートリミットパケットドロップ	スループット制限または 1 秒あたりのパケット数 (pps) の上限に達した。	7 秒	PF-RL-PPS-PKTS-DROPPED, PF-RL-RATE-PKTS-DROPPED

5. Call Home データをアップロードします。アプライアンス上で前述の重大な状態のいずれかが確認された場合、Call Home 機能は NetScaler サポートに自動的に通知します。サポートアーカイブは NetScaler サポートサーバーにアップロードされます。また、Call Home のアップロードが発生するたびに SNMP アラートを生成するように CALLHOME-UPLOAD-EVENT SNMP アラームを設定することもできます。SNMP アラートは、重要なイベントについてローカル管理者に通知します。

> 注
>
> Call Home は Call Home tar ファイルを作成し、前回の再起動以降に特定のエラー状態が最初に発生した場合にのみ Call Home tar ファイルを Citrix テクニカルサポートサーバーにアップロードします。特定のエラー状態が発生するたびにアプライアンスがアラートを送信するようにするには、エラー条件に対応する SNMP アラームを設定します。

6. サービスリクエストを作成します。Call Home は、すべての重要なハードウェア関連イベントに対するサービスリクエストを自動的に作成します。イベントは、電源障害、SSL カード障害、ハードディスクドライブエラー、コンパクトフラッシュエラーに分類されます。その他のエラーについては、システムログを確認した後、NetScaler サポートチームに連絡して調査を依頼できます。

Call Home の設定

Call Home を設定するには、アプライアンスのインターネット接続を確認し、DNS ネームサーバーが設定されていることを確認します。インターネットに接続されていない場合は、プロキシサーバーまたはサービスを構成します。次に、アプライアンスで Call Home を有効にして、アプライアンスの NetScaler サポートサーバーへの登録ステー

タスを確認します。登録が完了すると、Call Home はデータを監視およびアップロードできます。また、SNMP アラームを設定して、カスタマーサイトの管理者に通知することもできます。

Call Home を構成するには、NetScaler コマンドインターフェイスまたは GUI を使用して次のタスクを実行できます。

- Call Home を有効にします。
- オプションのプロキシサーバパラメータに Call Home を設定します。
- Call Home の登録ステータスを確認します。
- エラーとタイムスタンプの詳細を表示します。
- SNMP アラームを設定します。

NetScaler ADC コマンドインターフェイスを使用して **Call Home** を構成するには

NetScaler のコマンドインターフェイスでは、次の操作を行うことができます。

Enabling Call Home

コマンドプロンプトで入力します。

```
enable ns feature callhome
```

オプションのプロキシサーバパラメータに対する Call Home の設定

Call Home では、オプションのプロキシサーバをインターネット接続用に設定できます。IP アドレスとポートを使用してプロキシサーバを構成することも、一方向認証または双方向認証を使用するプロキシ認証サービスを構成することもできます。

To configure optional proxy server with IP address and port

コマンドプロンプトで入力します。

```
set callhome -proxyMode ( YES | NO )[-IPAddress <ip_addr|ipv6_addr
|*>] [-port <port |*>]
```

```
1 set callhome  - proxyMode YES  - IPAddress 10.102.167.33  - port 80
2 <!--NeedCopy-->
```

> 注
>
> Call Home がプロキシサーバを使用するのは、proxy-mode パラメータを YES に設定した場合だけです。NO に設定すると、IP アドレスとポートが設定されていても、プロキシ機能は動作しません。ポート番号は HTTPS サービス用ではなく、HTTP サービス用である必要があります。

オプションのプロキシ認証サービスを構成するには

このモードでは、一方向と双方向の 2 種類のセキュリティ認証が提供されます。どちらのタイプも設定するには、SSL サービスを設定する必要があります。詳細については、「SSL サービスの構成」トピックを参照してください。

一方向認証では、NetScaler アプライアンスのみがプロキシサーバーを認証します。双方向認証では、NetScaler アプライアンスがプロキシサーバーを認証し、プロキシサーバーがアプライアンスを認証します。

プロキシ認証サービスを設定するには

コマンドプロンプトで入力します。

```
set callhome -proxyMode ( YES | NO )[-proxyAuthService <string>]
```

```
1 set callhome  - proxyMode YES  - proxyAuthService callhome_proxy
2 <!--NeedCopy-->
```

一方向プロキシサーバ認証を構成するには

一方向プロキシサーバ認証を設定するには、次のタスクを実行します。

1. SSL サービスを作成します。
2. CA 証明書をサービスにバインドします。
3. HTTPS モニターをサービスにバインドします。
4. SSL サービスを使用するように Call Home を設定します。

双方向プロキシサーバ認証を構成するには

双方向プロキシサーバ認証を設定するには、次のタスクを実行します。

1. SSL サービスを作成する
2. CA 証明書をサービスにバインドします。
3. クライアント証明書をバインドします。
4. HTTPS モニターをサービスにバインドします。
5. SSL サービスを使用するように Call Home を設定します。

Call Home 登録ステータスの確認

コマンドプロンプトで入力します。

```
1  show callhome
2
3      show callhome
4
5       Registration with Citrix upload server SUCCESSFUL
6
7       Mode: Default
8
9       Contact email address: exampleadmin@example.com
10
11      Heartbeat Custom Interval (days): 7
12
13      Proxy Mode: Yes
14
15         Proxy IP Address:10.102.29.200
16
```



```
17          Proxy Authentication Service:
18
19          Proxy Port: 80
20
21      Trigger event                   State   First occurrence
                        Latest occurrence
22
23       -------------                   -----   ----------------
                         -----------------
24
25       1) Warm boot                    Enabled N/A
                                         ..
26
27       2) Compact flash errors         Enabled ..
                                           ..
28
29       3) Hard disk drive errors       Enabled ..
                                           ..
30
31       4) SSL card failure             N/A     N/A
                                         N/A
32
33       5) Power supply unit failure    N/A     N/A
                                         N/A
34
35       6) Rate limit packet drops      Enabled ..
                                           ..
36
37       7) Memory anomaly               Enabled ..
                                           ..
38
39       Done
40  <!--NeedCopy-->
```

> 注
>
> Call Home が CIS への登録に失敗すると、アプライアンスはエラーメッセージを表示します。

SNMP アラームの有効化

NetScaler アプライアンスは、*SNMP* アラームと呼ばれるエラー条件エンティティのセットを提供します。SNMP アラームのエラー条件が満たされると、アプライアンスは SNMP トラップメッセージを生成し、設定されたトラップリスナーに送信されます。たとえば、SSL-CARD-FAILED アラームが有効の場合、トラップメッセージが生成され、トラップリスナーに送信されます。アプライアンスで SSL カードに障害が発生すると、トラップメッセージが送信されます。詳細については、SNMPを参照してください。

コマンドプロンプトで入力します。

```
enable snmp alarm <trapName>

show snmp alarm <trapName>
```


GUI を使用して **Call Home** を設定するには

GUI で Call Home 機能がデフォルトで有効になっているかどうかを確認するには

1. 構成 > システム > 設定に移動します。
2. 詳細ウィンドウで、[高度な機能の構成] リンクをクリックします。
3. **[**拡張機能の設定] ページで、[**Call Home**] オプションが有効になっている必要があります。

GUI を使用して Call Home を有効にするには

1. 構成 > システム > 設定に移動します。
2. 詳細ウィンドウで、[拡張機能の構成] リンクをクリックし、[**Callhome**] オプションを選択します。

GUI を使用してオプションのプロキシモード認証用に Call Home を設定するには

1. [Call Home] ページにアクセスするには、次の 2 つの方法のいずれかを使用できます。
   a) [システム] > [システム情報] に移動します。
   b) [システム] > [診断**]** に移動します。
      i. 詳細ペインの [テクニカルサポートツール] で、[**Call Home**] を選択します。
2. [**Call Home** の設定] ページで、次のパラメータを設定します。
   a) **[**モード**]**。Call Home モードの動作。可能な種類：デフォルト、Citrix Service Provider（CSP）展開。

      > 注
      >
      > このオプションはユーザが設定できません。モードは、NetScaler ADC 展開のタイプに基づいて自動的に決定および設定されます。

   b) 電子メールアドレス。カスタマーサイトの連絡先管理者の電子メールアドレス。
   c) **CallHome** ハートビート間隔 **(**日数**)**。Call Home ハートビートのモニタリング間隔（日数）。最小値は 1 で、最大値は 30 です。
   d) **Call Home** を有効にします。Call Home 機能を有効または無効にして、NetScaler サポートサーバー上のアプライアンス登録のステータスを表示します。
   e) プロキシモード。インターネットに接続できない場合は、プロキシモードを有効にして、オプションのプロキシパラメータを設定します。
   f) プロキシサーバー。プロキシサーバを使用してプロキシモードを設定する場合は、サーバの IP アドレスを指定します。
      i. プロキシサービス。プロキシサービスを使用してプロキシモードを設定する場合は、サービス名を指定します。
      ii. **IP** アドレス。プロキシサーバの IP アドレス。
      iii. ポート。プロキシサーバのポート番号。
      iv. プロキシ認証 **SSL** サービス。プロキシモード認証を提供するプロキシサービスの名前。

3. [**OK**] をクリックし、[完了] をクリックします。

GUI を使用してプロキシサーバー認証用の SSL サービスを構成するには

GUI を使用した SSL サービスの構成の詳細については、「SSL サービスの構成 」トピックを参照してください。

GUI を使用して Call Home の登録ステータスを確認するには

1. [**Call Home**] ページにアクセスするには、次の 2 つの方法のいずれかを使用できます。
   a) [システム] > [システム情報] に移動します。
   b) [システム] > [診断**]** に移動します。
      i. 詳細ペインの [テクニカルサポートツール] で、[**Call Home**] を選択します。
2. [**Call Home** の構成] ページの **Citrix** アップロードサーバーへの登録] フィールドに登録ステータスが表示されます。

SNMP アラームを設定するには

1. **[**システム**]** > **[SNMP]** > **[**アラーム**]** に移動します。
2. 詳細ペインで、アラームを選択し、そのパラメータを設定します。
3. 「**OK**」をクリックして「閉じる」をクリックします。

Citrix Service Provider (CSP) 展開のサポート

NetScaler ADC サービスが VPX インスタンスに展開されている Citrix Service Provider（CSP）環境では、Call Home はライセンス固有の情報を監視および追跡し、その情報を Citrix Insight Services（CIS）に安全に送信できます。CIS は、アカウンティング目的および CSP のお客様がライセンス使用状況を確認できるように、ライセンス使用状況インサイト (LUI) ポータルに情報を送信します。現在、CSP 環境は NetScaler サービスを VPX インスタンスのみでサポートしており、MPX または SDX アプライアンスではサポートしていません。VPX インスタンスは、スタンドアロンモードでも高可用性モードでもデプロイできます。

レポートツール

August 15, 2023

Citrix® NetScaler® レポートツールを使用して、NetScaler ADC のパフォーマンス統計データをレポートとして表示します。統計データは`nscollect`ユーティリティによって収集され、データベースに保存されます。ある期間にわたって特定のパフォーマンスデータを表示する場合、レポートツールはデータベースから指定されたデータを取り出し、グラフに表示します。

レポートは、グラフのコレクションです。レポートツールには、組み込みのレポートと、カスタムレポートを作成するオプションが用意されています。レポートでは、グラフを変更したり、新しいグラフを追加できます。また、データ収集ユーティリティ`nscollect`の操作を変更し、その操作を停止または開始することもできます。

レポート作成ツールの使用

レポートツールは、Citrix® NetScaler® アプライアンスからアクセスする Web ベースのインターフェイスです。レポート作成ツールを使用して、パフォーマンス統計データをグラフを含むレポートとして表示します。組み込みレポートを使用するほかに、カスタムレポートを作成して、いつでも変更できます。レポートには 1 つから 4 つのグラフを含めることができます。最大 256 のカスタムレポートを作成できます。任意の数のエンティティのカスタムレポートを作成できます。

レポートツールを起動する

1. 任意の Web ブラウザを使用して、NetScaler ADC の IP アドレス（例: `http://10.102.29.170/`）に接続します。[Web ログオン] 画面が表示されます。
2. ［ユーザー名］テキストボックスに、NetScaler に割り当てられたユーザー名を入力します。
3. 「パスワード」テキスト・ボックスにパスワードを入力します。
4. [開始場所] ドロップダウンリストボックスで、[レポート] を選択します。[ログイン] をクリックします。

次のスクリーンショットは、レポートツールバーとチャートツールバーを示しています。このツールバーは、このドキュメントで頻繁に参照されています。

図 1: レポートツールバー

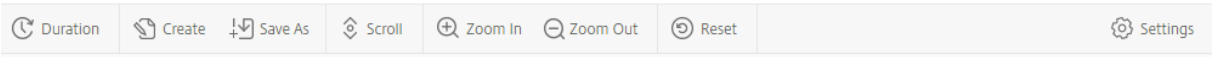
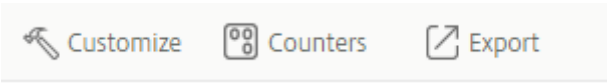


図 2: *[グラフ]* ツールバー



レポートの操作

NetScaler ADC で構成されたさまざまな機能グループの統計を、指定した時間間隔でプロットおよび監視できます。レポートを使用すると、アプライアンスの動作をトラブルシューティングまたは分析できます。レポートには、組み込みレポートとカスタムレポートの 2 種類があります。組み込みレポートまたはカスタムレポートのレポートコンテンツは、グラフ形式または表形式で表示できます。グラフィカルビューは、最大 32 セットのデータ（カウンタ）を表示できる折れ線グラフ、面グラフ、および棒グラフで構成されます。表形式ビューには、データが列と行に表示されます。このビューは、エラーカウンターのデバッグに役立ちます。

レポートツールに表示されるデフォルトのレポートは、CPU 対メモリ使用量と HTTP リクエスト率です。既定のレポートビューを変更するには、目的のレポートを既定のビューとして表示し、[既定のレポート] をクリックします。

過去 1 時間、前日、先週、先月、昨年のレポートを生成することも、期間をカスタマイズすることもできます。

レポートでは次のことができます。

- データの表形式表示とデータのグラフ表示を切り替えます。
- 棒グラフや折れ線グラフなどのグラフィカル表示タイプを変更します。
- レポート内のグラフをカスタマイズします。
- グラフを Excel コンマ区切り値 (CSV) ファイルとしてエクスポートします。
- ズームイン、ズームアウト、またはドラッグ操作 (スクロール) を使用して、グラフの詳細を表示します。
- ログオン時に表示する既定のレポートとしてレポートを設定します。
- カウンタを追加または削除します。
- レポートを印刷します。
- レポートを更新して最新のパフォーマンスデータを表示します。

組み込みレポートを使用する

レポートツールには、頻繁に閲覧されるデータに関する組み込みレポートが用意されています。組み込みレポートは、システム、ネットワーク、SSL、圧縮、統合キャッシュ、NetScaler ADC ゲートウェイ、および NetScaler ADC アプリケーションファイアウォールの機能グループで使用できます。デフォルトでは、ビルトインレポートは最終日に表示されます。ただし、過去 1 時間、先週、先月、または昨年のレポートを表示できます。

> 注記:
>
> 組み込みレポートへの変更を保存することはできませんが、変更した組み込みレポートをカスタムレポートとして保存することはできます。

組み込みレポートを表示する

1. レポートツールの左側のペインの [組み込みレポート] で、グループ (SSL など) を展開します。
2. レポートをクリックします ([**SSL] > [**すべてのバックエンド暗号**]** など**)**。

レポートの作成と削除

独自のカスタムレポートを作成し、ユーザー定義の名前を付けて保存して再利用できます。要件に応じて、グループごとに異なるカウンタをプロットできます。最大 256 のカスタムレポートを作成できます。

レポートを作成することも、組み込みレポートをカスタムレポートとして保存することもできます。デフォルトでは、新しく作成されたカスタムレポートには System Overview という名前のグラフが 1 つ含まれています。このグラフには、前日にプロットされた CPU 使用率が表示されます。レポート・ツールバーから間隔をカスタマイズし、データ・ソースとタイム・ゾーンを設定できます。

カスタムレポートの作成

1. レポートツールのレポートツールバーで、[作成] をクリックします。既存のレポートに基づいてカスタムレポートを作成する場合は、既存のレポートを開いて [名前を付けて保存] をクリックします。

2. [レポート名] ボックスに、カスタムレポートの名前を入力します。
3. 次のいずれかを行います:

    - レポートを既存のフォルダに追加するには、「作成先」または「保存先」で下矢印をクリックして既存のフォルダを選択し、「**OK**」をクリックします。
    - レポートを保存する新しいフォルダを作成するには、「クリックしてフォルダを追加」アイコンをクリックし、「フォルダ名」にフォルダの名前を入力し、「作成先」で新しいフォルダを階層内のどこに置くかを指定して、「**OK**」をクリックします。

> **
>
> 注 **: 最大 128 個のフォルダを作成できます。

カスタムレポートを削除する

1. レポートツールの左ペインで、[カスタムレポート] の横にある [クリックしてカスタムレポートを管理する] アイコンをクリックします。
2. 削除するレポートに対応するチェックボックスをオンにし、[削除] をクリックします。

    > 注:
    >
    > フォルダを削除すると、そのフォルダの内容がすべて削除されます。

時間間隔の変更

デフォルトでは、組み込みレポートには前日のデータが表示されます。ただし、組み込みレポートの時間間隔を変更する場合は、レポートをカスタムレポートとして保存できます。新しい間隔は、レポート内のすべてのグラフに適用されます。次の表では、時間間隔オプションについて説明します。

時間間隔を変更

1. レポートツールの左ペインで、レポートをクリックします。
2. レポートツールバーで、[期間] をクリックし、時間間隔をクリックします。

データソースとタイムゾーンの設定

さまざまなデータソースからデータを取得してレポートに表示できます。また、レポートのタイムゾーンを定義し、現在表示されているレポートの時間選択を組み込みレポートを含むすべてのレポートに適用することもできます。

データソースとタイムゾーンを設定する

1. レポートツールのレポートツールバーで、[設定] をクリックします。
2. [設定] ダイアログボックスの [データソース] で、カウンター情報を取得するデータソースを選択します。

3. 次のいずれか、または両方を行います。
    - チャートがプロットされた期間をツールに記憶させたい場合は、[チャートの時間選択を記憶する] チェックボックスをオンにします。
    - レポートに NetScaler ADC アプライアンスの時間設定を使用する場合は、「アプライアンスのタイムゾーンを使用」チェックボックスを選択します。

カスタムレポートのエクスポートとインポート

レポートをエクスポートすることで、他の NetScaler ADC 管理者とレポートを共有できます。レポートをインポートすることもできます。

カスタムレポートをエクスポートまたはインポートする

1. レポートツールの左ペインで、「カスタムレポート」の横にある「クリックしてカスタムレポートを管理」アイコンをクリックします。
2. エクスポートまたはインポートするレポートに対応するチェックボックスを選択し、**[エクスポート]** または **[** ** インポート] をクリックします。**

    > 注:
    >
    > ファイルをエクスポートすると、.gz ファイル形式でエクスポートされます。

グラフでの作業

チャートを使用して、カウンターまたはカウンターのグループをプロットおよび監視します。1 つのレポートに最大 4 つのグラフを含めることができます。各チャートでは、最大 32 個のカウンターをプロットできます。チャートでは、さまざまなグラフィカルフォーマット (面グラフやバーなど) を使用できます。レポート内でグラフを上下に移動したり、グラフ内の各カウンターの色や表示をカスタマイズしたり、監視したくない場合はグラフを削除したりできます。

どのレポートグラフでも、横軸は時間を表し、縦軸はカウンターの値を表します。

グラフを追加する

レポートにグラフを追加すると、過去 1 日の CPU 使用率カウンタがプロットされた System Overview グラフが表示されます。

> 注記:
>
> 組み込みレポートにグラフを追加し、そのレポートを保持したい場合は、レポートをカスタムレポートとして保存する必要があります。

レポートにチャートを追加するには、以下の手順に従います。

レポートにグラフを追加

1. レポートツールの左ペインで、レポートをクリックします。
2. 新しいチャートを追加するチャートの下で、追加アイコンをクリックします。

チャートの変更

統計が表示される機能グループを変更したり、別のカウンタを選択したりすることで、チャートを変更できます。

グラフを変更

1. レポートツールの左ペインで、レポートをクリックします。
2. 変更するグラフの下にある [カウンター] をクリックします。
3. 表示されるダイアログボックスの [タイトル] ボックスに、グラフの名前を入力します。
4. のプロットチャートの横で、次のいずれかを実行します。
    - 統合キャッシュや圧縮などのグローバルカウンタのカウンタをプロットするには、[システムグローバル統計] をクリックします。
    - 負荷分散や GSLB などのエンティティタイプのエンティティカウンタをプロットするには、[システムエンティティの統計] をクリックします。
5. [選択] グループで、目的のエンティティをクリックします。
6. [カウンタ] の [使用可能] で、プロットする 1 つ以上のカウンタ名をクリックし、[>] ボタンをクリックします。
7. 手順 4 で [システムエンティティの統計] を選択した場合は、[エンティティ] タブの [使用可能] で、プロットする 1 つまたは複数のエンティティインスタンス名をクリックし、[>] ボタンをクリックします。
8. 「OK」をクリックします。

グラフを表示する

グラフにプロットされるカウンタのグラフィカルな形式を指定できます。チャートは、折れ線グラフ、スプラインチャート、ステップラインチャート、散布図、エリアチャート、バーチャート、積層面チャート、積み上げ横棒チャートとして表示できます。チャートのプロットエリア内をズームイン、ズームアウト、スクロールすることもできます。すべてのデータソースを 1 時間、1 日、1 週間、1 か月、1 年、3 年間拡大または縮小できます。

グラフのビューをカスタマイズするその他のオプションには、グラフの軸のカスタマイズ、プロット領域の背景とエッジの色の変更、グリッドの色とサイズのカスタマイズ、グラフ内の各データセット (カウンター) の表示のカスタマイズなどがあります。

データセット 1 などのデータセット番号は、グラフ内のカウンターがグラフの下部に表示される順序に対応しています。たとえば、グラフの一番下に CPU 使用率とメモリ使用量が第 1 順と 2 番目に表示されている場合、CPU 使用量はデータセット 1 と等しく、メモリ使用量はデータセット 2 に等しくなります。

組み込みレポートを変更するたびに、そのレポートをカスタムレポートとして保存して変更を保存する必要があります。

チャートのグラフタイプを変更する

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のペインで、表示したいチャートの下にあるチャートツールバーで、[カスタマイズ] をクリックします。
3. [グラフ] タブの [カテゴリ] で [プロットの種類] をクリックし、グラフに表示するグラフの種類をクリックします。グラフを 3D で表示する場合は、「3D を使用」チェックボックスを選択します。

詳細なデータでチャートに再び焦点を合わせる

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、レポートツールバーの [拡大] をクリックし、次のいずれかまたは両方の操作を行います。
   - 特定のタイムウィンドウのデータを表示するようにチャートを再フォーカスするには、カーソルを開始時刻から終了時刻までドラッグします。たとえば、特定の日の 1 時間分のデータを表示できます。
   - グラフにフォーカスしてデータポイントのデータを表示するには、ズームインするグラフを一度クリックして、詳細情報を取得します。
3. 詳細データを表示する時間範囲が整ったら、レポートツールバーの [表形式表示] をクリックします。表形式ビューでは、データが行と列の数値形式で表示されます。

グラフの数値データを表示する

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のペインのレポートツールバーで、[表形式ビュー] をクリックします。グラフィカルビューに戻るには、「グラフィカルビュー」をクリックします。

   > 注: グリッドラインのノッチにカーソルを合わせると、グラフィカルビューで数値データを表示することもできます。

チャートの時間をスクロールする

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、レポートツールバーの [スクロール] をクリックし、グラフ内をクリックし、新しい期間のデータを表示する方向にカーソルをドラッグします。たとえば、過去のデータを表示する場合は、左にドラッグします。

グラフの背景色とテキスト色の変更

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のペインで、軸をカスタマイズするグラフの下にある [カスタマイズ] をクリックします。
3. [グラフ] タブの **[**カテゴリ**]** で、次の 1 つまたは複数をクリックします。
   - 背景色を変更するには、「背景色」をクリックし、色、透明度、および効果のオプションを選択します。

- テキストの色を変更するには、「テキストの色」をクリックし、色、透明度、および効果のオプションを選択します。

チャートの軸をカスタマイズする

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のペインで、軸をカスタマイズするグラフの下にある [カスタマイズ] をクリックします。
3.「チャート」タブの「カテゴリ」で、次のいずれか 1 つ以上をクリックします。

   - 左の Y 軸のスケールを変更するには、[左の **Y** 軸] をクリックし、目的のスケールを選択します。
   - 右の Y 軸のスケールを変更するには、[右の Y 軸] をクリックし、プロットするデータセットで日付セットを選択し、目的のスケールを選択します。

     > 注:
     > データセット 1 などのデータセット番号は、グラフの下部に表示されるカウンタの順序に対応しています。たとえば、グラフの一番下に CPU 使用率とメモリ使用量が第 1 順と 2 番目に表示されている場合、CPU 使用量はデータセット 1 と等しく、メモリ使用量はデータセット 2 に等しくなります。

   - 各データセットを独自の非表示の Y 軸にプロットするには、[複数軸] をクリックし、[有効化] をクリックします。

グラフのプロット領域の背景色、エッジ色、およびグリッド線の変更

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のペインで、プロットエリアをカスタマイズするグラフの下にある [カスタマイズ] をクリックします。
3. [プロットエリア] タブの [カテゴリ] で、次の 1 つ以上をクリックします。

   - グラフの背景色とエッジの色を変更するには、[背景色**]** と **[**エッジの色**]** をクリックし、色、透明度、および効果のオプションを選択します。
   - グラフの水平グリッドまたは垂直グリッドを変更するには、[水平グリッド] または [垂直グリッド] をクリックし、グリッド、グリッド幅、グリッドの色、透明度、および効果の表示オプションを選択します。

データセットの色とグラフタイプの変更

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のペインで、データセット (カウンター) の表示をカスタマイズするグラフの下にある [カスタマイズ] をクリックします。
3. [データセット] タブの [データセットの選択] で、グラフィカル表示をカスタマイズするデータセット (カウンター) を選択します。
   注: データセット 1 などのデータセット番号は、グラフ内のカウンターがグラフの下部に表示される順序に対応しています。たとえば、グラフの一番下に

CPU
使用率とメモリ使用量が第 1 順と 2 番目に表示されている場合、
CPU
使用量はデータセット 1 と等しく、

メモリ使用量はデータセット 2 に等しくなります。

4. [カテゴリ] で、次のいずれかを行います。

   - 背景色を変更するには、「色」をクリックし、色、透明度、および効果のオプションを選択します。
   - グラフタイプを変更するには、[プロットタイプ] をクリックし、データセットに表示するグラフタイプを選択します。グラフを 3D で表示する場合は、「3D を使用」チェックボックスを選択します。

チャートデータの **Excel** へのエクスポート　さらにデータを分析するために、グラフをカンマ区切り値 (CSV) 形式で Excel にエクスポートできます。

チャートデータを Excel にエクスポートするには

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、Excel にエクスポートするデータを含むグラフの下にある [エクスポート] をクリックします。

グラフを削除する

グラフを使用したくない場合は、レポートから削除できます。チャートを完全に削除できるのは、カスタムレポートからのみです。組み込みレポートからグラフを削除し、変更を保存したい場合は、レポートをカスタムレポートとして保存する必要があります。

グラフを削除する

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のペインで、削除するチャートの下にある削除アイコンをクリックします。

例

過去 **1** 週間の **CPU** 使用率とメモリ使用量のトレンドレポートを表示します

1. レポートツールの左ウィンドウの [組み込みレポート] で、[システム] を展開します。
2. CPU 対メモリ使用量および HTTP リクエスト率のレポートをクリックします。
3. 右側のウィンドウで、レポートツールバーの [期間] をクリックし、[先週] をクリックします。

過去 **1** 週間の **2** つのインターフェイス間の受信バイトレートとバイト転送レートを比較します

1. 右側のウィンドウで、レポートツールバーの [作成] をクリックします。
2. 「レポート名」ボックスに、カスタムレポートの名前 (Custom_Interfaces など) を入力し、「**OK**」をクリックします。レポートはデフォルトのシステム概要グラフで作成されます。このグラフには、過去 1 時間にプロットされた CPU 使用率が表示されます。
3. チャートツールバーの [システム概要] で、[カウンター] をクリックします。
4. カウンター選択ペインの [タイトル] に、グラフの名前 (たとえば、インターフェイスバイトデータ) を入力します。
5. [グラフのプロット] で、[システムエンティティの統計] をクリックし、[グループの選択] で [インターフェイス] を選択します。
6. [エンティティ] タブで、プロットする 1 つまたは複数のインタフェース名 (1/1 と 1/2 など) をクリックし、[>] ボタンをクリックします。
7. [カウンタ] タブで、[受信バイト数 (レート)] および [送信バイト数 (レート)] をクリックし、[>] ボタンをクリックします。
8. [**OK**] をクリックします。
9. レポートツールバーの [期間] をクリックし、[先週] をクリックします。

データ収集ユーティリティの停止と起動

データ収集ユーティリティ`nscollect`は、NetScaler ADC を起動すると自動的に実行されます。このユーティリティは、アプリケーション・パフォーマンス・データを取得し、ADC 上のデータ・ソース形式で保存します。最大 32 個のデータソースを作成できます。既定のデータソースは`/var/log/db/default`です。

データ収集ユーティリティは、グローバルカウンタとエンティティ固有のカウンタ用のデータベースを作成し、このデータを使用してレポートを生成します。グローバルカウンタデータベースは、`/var/log/db/<DataSourceName>`で作成されます。エンティティ固有のデータベースは、NetScaler ADC で構成されたエンティティに基づいて作成され、`/var/log/db/<DataSourceName/EntityNameDB>`のエンティティタイプごとに個別のフォルダが作成されます。

`nscollect`は 5 分ごとに 1 回データを取得します。1 日間、過去 30 日間は 1 時間ごと、3 年間は 1 日ごとに 5 分単位でデータを保持します。

データが正確に更新されない場合や、レポートに破損したデータが表示される場合は、データ収集ユーティリティを停止して再起動する必要があります。

nscollect の停止

コマンドプロンプトで入力します。

```
/netscaler/nscollect stop
```

NetScaler への現在の **SSH** セッションで **nscollect** を起動します。

コマンドプロンプトで入力します。

```
/netscaler/nscollect start
```

ローカルシステムで **nscollect** を起動します。

コマンドプロンプトで入力します。

```
/netscaler/nscollect start &
```

CloudBridge Connector

August 15, 2023

注：現在の NetScaler 1000V リリースでは、この機能はサポートされていません。

NetScaler アプライアンスの CloudBridge Connector 機能は、企業のデータセンターを外部クラウドやホスティング環境に接続し、クラウドを企業ネットワークの安全な拡張機能にします。クラウドでホストされるアプリケーションは、1 つの連続したエンタープライズネットワークで実行されているかのように見えます。Citrix CloudBridge Connector を使用すると、クラウドプロバイダーが提供する容量と効率性でデータセンターを強化できます。

CloudBridge Connector を使用すると、アプリケーションをクラウドに移行してコストを削減し、信頼性を高めることができます。

データセンターとクラウド間で CloudBridge Connector を使用するだけでなく、2 つのデータセンターを接続して、大容量の安全で高速なリンクを実現することもできます。

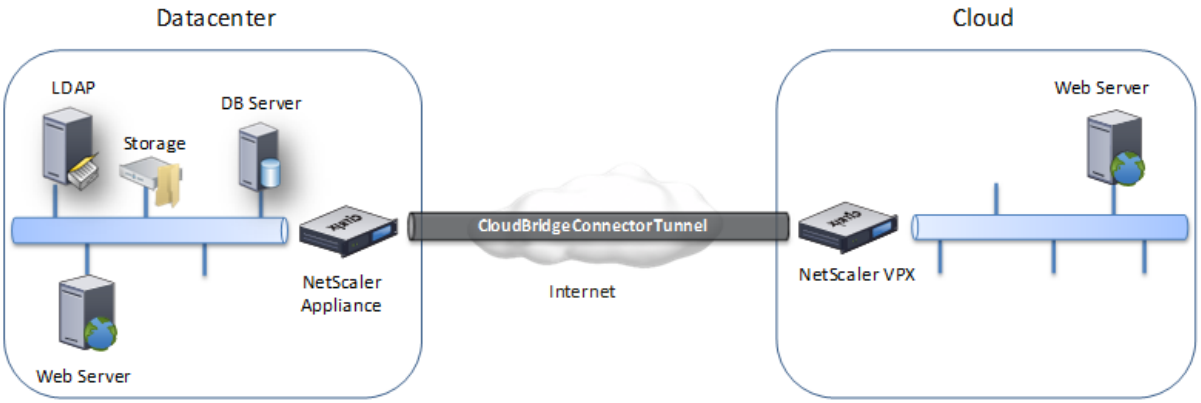
CloudBridge Connector について

Citrix CloudBridge Connector ソリューションを実装するには、CloudBridge Connector Bridge コネクタトンネルと呼ばれるトンネルを設定して、データセンターを別のデータセンターまたは外部クラウドに接続します。

データセンターを別のデータセンターに接続するには、各データセンターに 1 つずつ、合計 2 つの NetScaler アプライアンス間に CloudBridge Connector トンネルを設定します。

データセンターを外部クラウド（Amazon AWS クラウドなど）に接続するには、データセンター内の NetScaler ADC アプライアンスとクラウドに存在する仮想アプライアンス（VPX）の間に CloudBridge Connector トンネルを設定します。リモートエンドポイントは、CloudBridge Connector またはプレミアムライセンスを持つ NetScaler ADC VPX にすることができます。

次の図は、データセンターと外部クラウドの間に設定された CloudBridge Connector トンネルを示しています。



CloudBridge Connector トンネルがセットアップされているアプライアンスは、CloudBridge Connector トンネルのエンドポイントまたはピアと呼ばれます。

CloudBridge Connector トンネルは以下のプロトコルを使用します。

- 汎用ルーティングカプセル化 (GRE) プロトコル
- オープンスタンダード IPsec プロトコルスイート、トランスポートモード

GRE プロトコルは、さまざまなネットワークプロトコルからのパケットをカプセル化し、別のプロトコルで転送するメカニズムを提供します。GRE は次の目的で使用されます。

- 非 IP プロトコルやルーティング不可能なプロトコルを実行するネットワークを接続します。
- ワイドエリアネットワーク (WAN) を経由するブリッジ。
- 変更せずに別のネットワークに送信する必要があるあらゆる種類のトラフィック用のトランスポートトンネルを作成します。

GRE プロトコルは、GRE ヘッダーと GRE IP ヘッダーをパケットに追加することによってパケットをカプセル化します。

インターネットプロトコルセキュリティ (IPsec) プロトコルスイートは、CloudBridge Connector トンネル内のピア間の通信を保護します。

CloudBridge Connector トンネルでは、IPsec により以下が保証されます。

- データインテグリティ
- データオリジン認証
- データの機密保持 (暗号化)
- リプレイ攻撃からの保護

IPsec は、GRE カプセル化されたパケットが暗号化されるトランスポートモードを使用します。暗号化は、カプセル化セキュリティペイロード (ESP) プロトコルによって行われます。ESP プロトコルは、HMAC ハッシュ関数を使用してパケットの完全性を保証し、暗号化アルゴリズムを使用して機密性を保証します。パケットが暗号化され、

HMAC が計算されると、ESP ヘッダーが生成されます。ESP ヘッダーは GRE IP ヘッダーの後に挿入され、ESP トレーラーは暗号化されたペイロードの最後に挿入されます。

CloudBridge Connector トンネル内のピアは、インターネットキーエクスチェンジバージョン（IKE）プロトコル（IPSec プロトコルスイートの一部）を使用して、次のように安全な通信をネゴシエートします。

- 2 つのピアは、次のいずれかの認証方法を使用して相互に認証します。
  - 事前共有キー認証。事前共有キーと呼ばれるテキスト文字列は、各ピアで手動で設定されます。ピアの事前共有鍵は相互に照合されて認証されます。したがって、認証を成功させるには、各ピアに同じ事前共有キーを設定する必要があります。
  - デジタル証明書認証。発信側（送信者）ピアは秘密鍵を使用してメッセージ交換データに署名し、もう一方の受信側ピアは送信者の公開鍵を使用して署名を検証します。通常、公開鍵は X.509v3 証明書を含むメッセージで交換されます。この証明書は、証明書に記載されているピアの ID が特定の公開鍵に関連付けられていることをある程度保証します。
- その後、同僚は交渉し、以下について合意に達します。
  - 暗号化アルゴリズム。
  - 一方のピアでデータを暗号化し、もう一方のピアでデータを復号するための暗号キー。

セキュリティプロトコル、暗号化アルゴリズム、および暗号鍵に関するこの合意は、セキュリティアソシエーション（SA）と呼ばれます。SA は一方向（シンプレックス）です。たとえば、CB1 と CB2 の 2 つのピアがコネクタトンネルを介して通信している場合、CB1 には 2 つのセキュリティアソシエーションがあります。一方の SA はアウトバウンドパケットの処理に使用され、もう 1 つの SA はインバウンドパケットの処理に使用されます。

SA は、ライフタイムと呼ばれる指定された時間が経過すると期限切れになります。2 つのピアは、インターネットキー交換 (IKE) プロトコル (IPsec プロトコルスイートの一部) を使用して新しい暗号鍵をネゴシエートし、新しい SA を確立します。限られた寿命の目的は、攻撃者が鍵をクラックするのを防ぐことです。

次の表は、NetScaler アプライアンスでサポートされている一部の IPsec プロパティを示しています。

IPSec のプロパティ	対応タイプ
IKE バージョン	V1, V2
IKE DH グループ	NetScaler ADC アプライアンスは、IKEv1 と IKEv2 の両方で DH グループ 2（1024 ビット MODP アルゴリズム）のみをサポートします。
IKE 認証方法	事前共有キー認証、デジタル証明書認証
暗号化アルゴリズム	AES (128 ビット)、AES 256 (256 ビット)、3DES
ハッシュアルゴリズム	HMAC SHA1, HMAC SHA256, HMAC SHA384, HMAC SHA512, HMAC MD5

CloudBridge Connector トンネルの監視

August 15, 2023

CloudBridge Connector トンネルのパフォーマンスを監視するための統計情報を表示できます。NetScaler アプライアンスの CloudBridge Connector トンネル統計を表示するには、GUI または NetScaler コマンドラインを使用します。

次の表は、NetScaler アプライアンス上の CloudBridge Connector トンネルを監視するために使用できる統計カウンターを示しています。

統計カウンター	**Specifies**
受信バイト数	アプライアンスを最後に起動してから、NetScaler アプライアンスが構成済みのすべての CloudBridge Connector トンネルを介して受信したバイトの総数。
送信済みバイト数	アプライアンスを最後に起動してから、NetScaler アプライアンスが構成済みのすべての CloudBridge Connector トンネルを介して送信したバイトの総数。
受信済みパケット	アプライアンスを最後に起動してから、NetScaler アプライアンスが構成済みのすべての CloudBridge Connector トンネルを介して受信したパケットの総数。
送信済みパケット	アプライアンスを最後に起動してから、NetScaler アプライアンスが構成済みのすべての CloudBridge Connector トンネルを介して送信したパケットの総数。
バイト受信レート	NetScaler アプライアンスが構成済みのすべての CloudBridge Connector トンネルを介して受信した 1 秒あたりのバイト数。
バイト送信レート	NetScaler アプライアンスが構成済みのすべての CloudBridge Connector トンネルを介して送信する 1 秒あたりのバイト数
パケット受信率	NetScaler アプライアンスが構成済みのすべての CloudBridge Connector トンネルを介して受信した 1 秒あたりのバイト数
パケット送信レート	NetScaler アプライアンスが構成済みのすべての CloudBridge Connector トンネルを介して受信した 1 秒あたりのバイト数

NetScaler アプライアンスを再起動すると、これらのカウンタはすべて 0 にリセットされます。次の段階では増加しません。

- 設定済みの任意の CloudBridge Connector トンネルでのインターネットキー交換（IKE）認証（事前共有キー）フェーズ。
- 設定済みの任意の CloudBridge Connector トンネルでの IKE セキュリティアソシエーション (SA) 確立フェーズ。

NetScaler コマンドラインを使用して CloudBridge Connector のトンネル統計を表示するには

コマンドプロンプトで入力します。

- **IPSec** カウンタを統計する

GUI を使用して CloudBridge Connector のトンネルの統計情報を表示するには

1. Web ブラウザーを使用して GUI にアクセスし、NetScaler アプライアンスの IP アドレスに接続します。
2.「設定」タブで、「システム」>「**CloudBridge**Connector」に移動します。
3.「CloudBridge Connector」ページで、「CloudBridge Connector の作成/監視」をクリックします。「**IPSec Bytes** 」および「**IPSec Packets** 」チャートには、NetScaler ADC アプライアンスで設定されているすべての CloudBridge Connector トンネルのバイト受信レート、バイト送信レート、パケット受信レート、およびパケット送信レートが表示されます。

```
    > stat ipsec counters
    Secure tunnel(s) summary
                          Rate (/s)        Total
    Bytes Received    0    2811248
    Bytes Sent          0    157460630
    Packets Received    0    56787
    Packets Sent        0    200910
    Done
    >
<!--NeedCopy-->
```

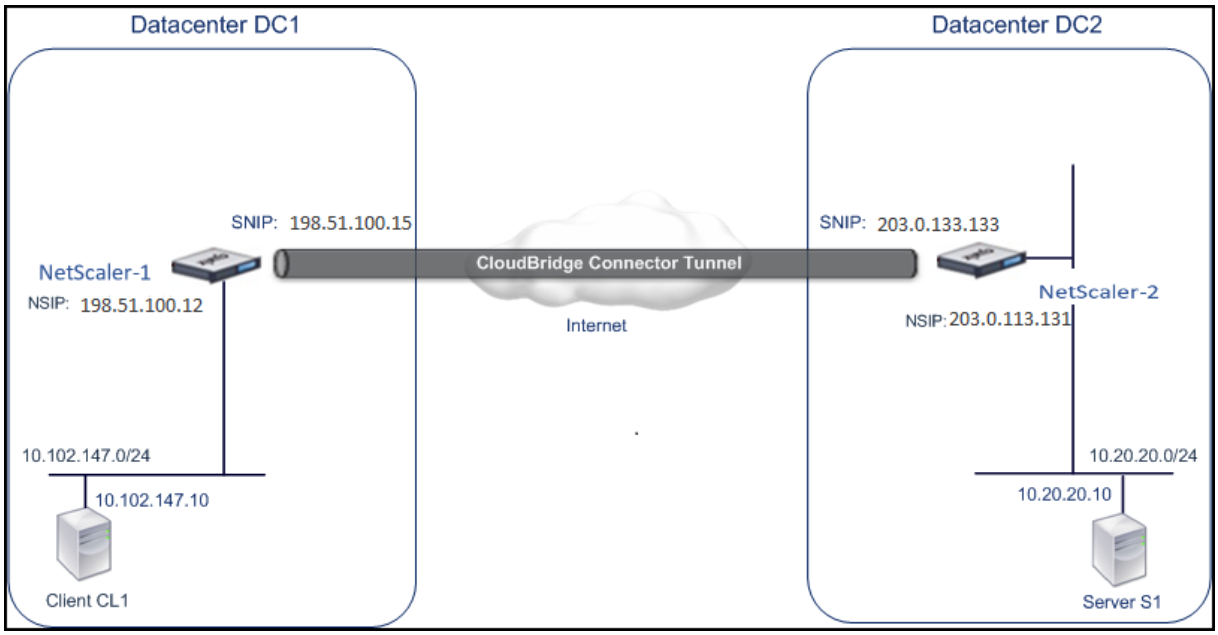
2 つのデータセンター間の **CloudBridge Connector** トンネルの設定

August 15, 2023

2 つの異なるデータセンター間で CloudBridge Connector トンネルを設定すると、再構成せずにネットワークを拡張し、2 つのデータセンターの機能を活用できます。地理的に離れた 2 つのデータセンター間の CloudBridge Connector トンネルにより、冗長性を実装し、セットアップを障害から守ることができます。CloudBridge Connector トンネルは、データセンター全体でインフラストラクチャとリソースを最適に活用するのに役立ちます。2 つのデータセンターで利用可能なアプリケーションは、ユーザーに対してローカルとして表示されます。

データセンターを別のデータセンターに接続するには、一方のデータセンターの NetScaler アプライアンスと別のデータセンターの NetScaler アプライアンスの間に CloudBridge Connector トンネルを設定します。

データセンター間の CloudBridge Connector トンネルの図として、データセンター DC1 の NetScaler アプライアンス NS_Appliance-1 とデータセンター DC2 の NetScaler アプライアンス NS_Appliance-2 の間に CloudBridge Connector トンネルが設定されている例を考えてみましょう。



NS_ アプライアンス 1 と NS_Appliance-2 はどちらも L2 モードと L3 モードで機能します。これにより、データセンター DC1 と DC2 のプライベートネットワーク間の通信が可能になります。L3 モードでは、NS_Appliance-1 と NS_Appliance-2 により、CloudBridge Connector トンネルを介してデータセンター DC1 のクライアント CL1 とデータセンター DC2 のサーバー S1 間の通信が可能になります。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

クライアント CL1 とサーバ S1 は異なるプライベートネットワーク上にあるため、NS_Appliance-1 と NS_Appliance-2 では L3 モードが有効になり、ルートは次のように更新されます。

- CL1 には、S1 に到達するための NS_Appliance-1 へのルートがあります。
- NS_Appliance-1 には、S1 に到達するための NS_Appliance-2 へのルートがあります。
- S1 には、CL1 に到達するための NS_Appliance-2 へのルートがあります。
- NS_Appliance-2 には、CL1 に到達するための NS_Appliance-1 へのルートがあります。

次の表は、データセンター DC1 の NetScaler アプライアンス NS_Appliance-1 の設定を示しています。

次の表は、データセンター DC2 の NetScaler アプライアンス NS_Appliance-2 の設定を示しています。

エンティティ	名前	詳細
NSIP アドレス		198.51.100.12
SNIP アドレス		198.51.100.15

エンティティ	名前	詳細
CloudBridge Connector トンネル	Cloud_Connector_DC1-DC2	1. CloudBridge Connector トンネルのローカルエンドポイント IP アドレス:198.51.100.15, 2. CloudBridge Connector トンネルのリモートエンドポイント IP アドレス:203.0.113.133。GRE トンネル詳細名 = Cloud_Connector_DC1-DC2、IPsec プロファイル詳細名 = Cloud_Connector_DC1-DC2、暗号化アルゴリズム = AES、ハッシュアルゴリズム = HMAC SHA1

CloudBridge Connector トンネルを設定する際に考慮すべき点

CloudBridge Connector トンネルをセットアップする前に、次のタスクが完了していることを確認してください。

1. NetScaler アプライアンスを 2 つのデータセンターのそれぞれに展開してセットアップします。
2. CloudBridge Connector のトンネルエンドポイントの IP アドレスが相互にアクセスできることを確認してください。

設定手順

あるデータセンターにある NetScaler アプライアンスと別のデータセンターにある別の NetScaler アプライアンスの間に CloudBridge Connector トンネルを設定するには、いずれかの NetScaler アプライアンスの GUI またはコマンドラインインターフェイスを使用します。

GUI を使用すると、最初の NetScaler アプライアンスで作成された CloudBridge Connector トンネル構成は、CloudBridge Connector トンネルのもう一方のエンドポイント（他の NetScaler アプライアンス）に自動的にプッシュされます。そのため、他の NetScaler アプライアンスの GUI にアクセスして、対応する CloudBridge Connector トンネル構成をそのアプライアンス上で作成する必要はありません。

各 NetScaler アプライアンスの CloudBridge Connector トンネル構成は、次のエンティティで構成されています。

- **IPsec** プロファイル—IPsec プロファイルエンティティは、CloudBridge Connector トンネル内の IPsec プロトコルが使用する IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、PSK などの IPsec プロトコルパラメータを指定します。
- **GRE** トンネル—IP トンネルは、ローカル IP アドレス（ローカル NetScaler アプライアンスで構成されたパブリック SNIP アドレス）、リモート IP アドレス（リモート NetScaler アプライアンスで構成されたパブリック SNIP アドレス）、CloudBridge Connector トンネルのセットアップに使用されるプロトコル（GRE）、および IPsec プロファイルエンティティを指定します。
- **PBR** ルールを作成し、それに **IP** トンネルを関連付けます。PBR エンティティは、条件セットと IP トンネルエンティティを指定します。送信元 IP アドレスの範囲と宛先 IP の範囲は、PBR エンティティの条件です。送信元 IP アドレス範囲と宛先 IP アドレス範囲を設定して、トラフィックが CloudBridge Connector トンネルを通過するサブネットを指定する必要があります。たとえば、最初のデータセンターのサブネット上のクライアントから送信され、2 番目のデータセンターのサブネット上のサーバー宛てのリクエストパケットを考えてみましょう。このパケットが最初のデータセンターの NetScaler アプライアンス上の PBR エンティティの送信元および宛先 IP アドレス範囲と一致する場合、PBR エンティティに関連付けられた CloudBridge Connector トンネルを介して送信されます。

コマンドラインインターフェイスを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで入力します。

- `add ipsec profile <name> [-ikeVersion (V1 | V2)] [-encAlgo (AES | 3DES)...] [-hashAlgo <hashAlgo\> ...] [-lifetime <positive_integer>] (-psk | (-publickey<string> -privatekey <string>-peerPublicKey <string>))[-livenessCheckInterval <positive_intege>][-replayWindowSize \<positive_integer>] [-ikeRetryInterval <positive_integer>] [-retransmissiontime <positive_integer>]`
- `show ipsec profile <name>`

コマンドラインインターフェイスを使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote><remoteSubnetMask> <local> [-protocol <protocol>] [-ipsecProfileName <string>]`
- `show ipTunnel <name>`

コマンドラインインターフェイスを使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

コマンドプロンプトで入力します。

- add ns pbr <pbr_name> ALLOW -srcIP = <local_subnet_range> -destIP = <remote_subnet_range> -ipTunnel <tunnel_name>
- apply ns pbrs
- show ns pbr <pbr_name>

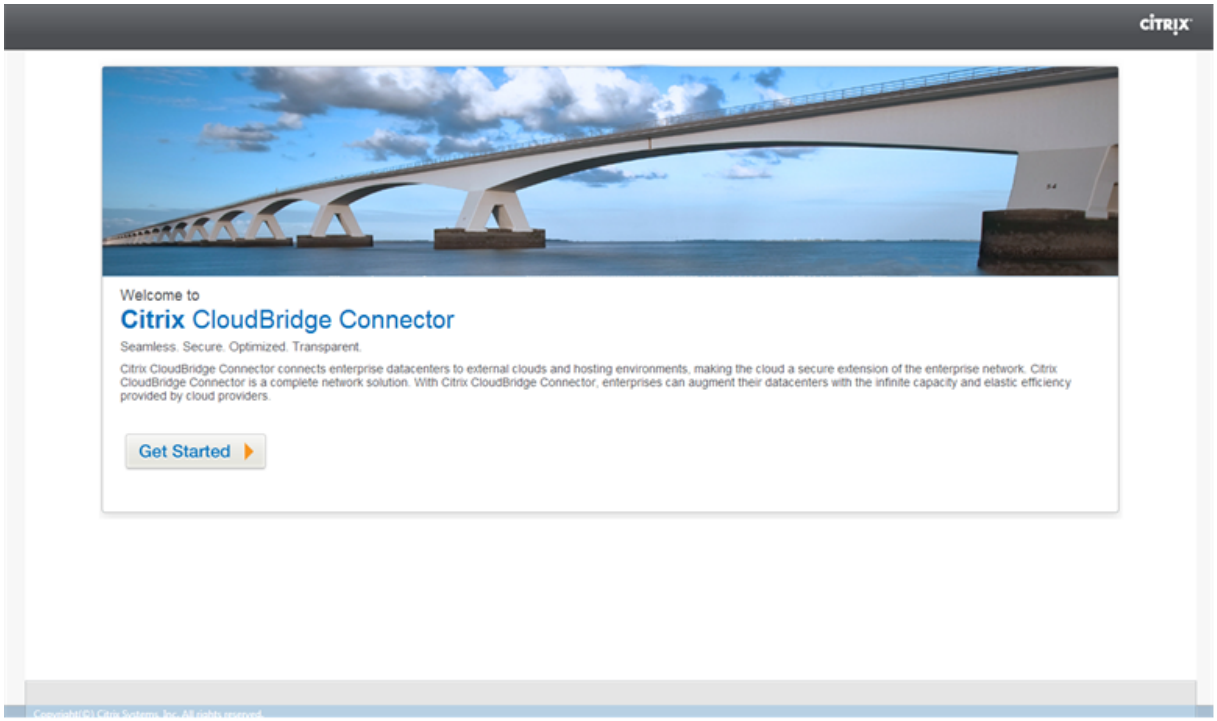
例

```
1   add ipsec profile Cloud_Connector_DC1-DC2  -encAlgo AES -hashAlgo
       HMAC_SHA1
2      Done
3      > add ipTunnel Cloud_Connector_DC1-DC2 203.0.113.133
          255.255.255.255 198.51.100.15 -protocol GRE -ipsecProfileName
          Cloud_Connector_DC1-DC2
4
5      Done
6      > add ns pbr PBR-DC1-DC2 ALLOW -srcIP 198.51.100.15 -destIP
          203.0.113.133 ipTunnel Cloud_Connector_DC1-DC2
7
8      Done
9      > apply ns pbrs
10
11     Done
12  <!--NeedCopy-->
```

GUI を使用して NetScaler アプライアンスの CloudBridge Connector トンネルを構成するには

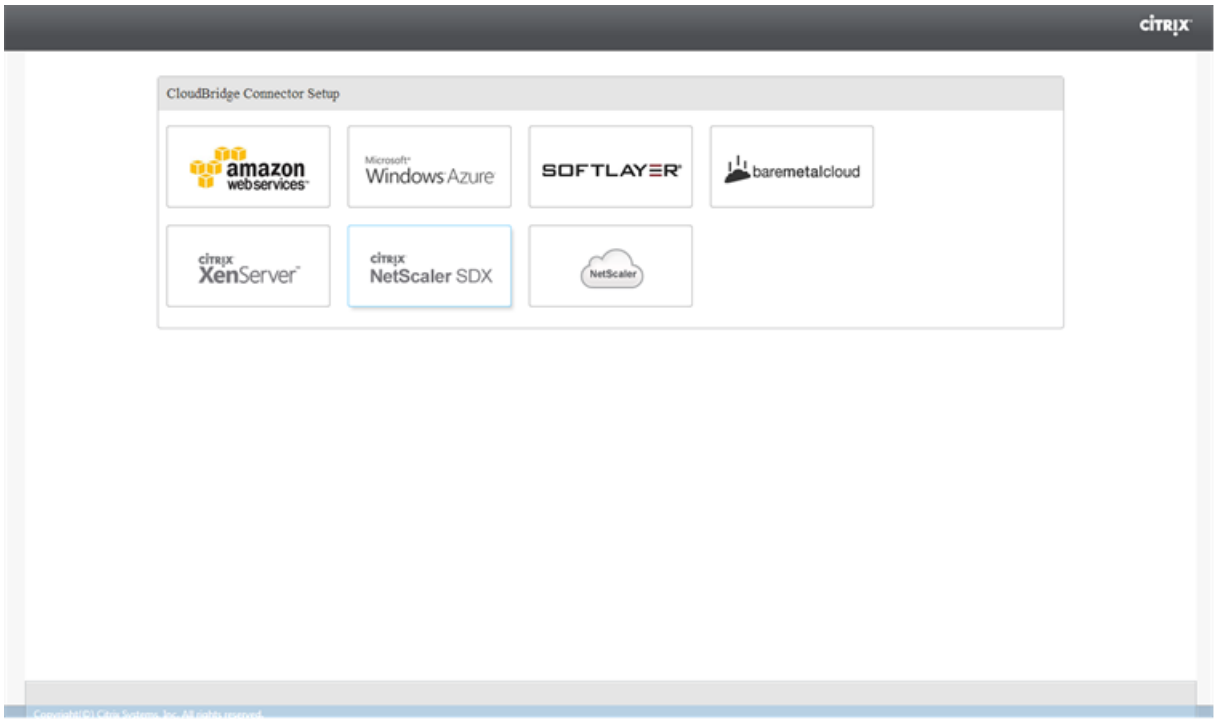
1. Web ブラウザーのアドレス行に NetScaler アプライアンスの NSIP アドレスを入力します。
2. アプライアンスのアカウント認証情報を使用して、NetScaler アプライアンスの GUI にログオンします。
3. [システム] > [**CloudBridge**Connector] に移動します。
4. 右側のペインの「はじめに」で、「CloudBridge の作成/監視」をクリックします。

   アプライアンスで CloudBridge Connector トンネルを初めて設定すると、ウェルカム画面が表示されます。
5. 「ようこそ」画面で、「はじめに」をクリックします。



> 注記:
>
> NetScaler アプライアンスですでに CloudBridge Connector トンネルを構成している場合は、「ようこそ」画面は表示されないので、「はじめに」をクリックしないでください。

1. **CloudBridge Connector** のセットアップペインで、**NetScaler** をクリックします。



1. NetScaler ペインで、リモート NetScaler アプライアンスのアカウント認証情報を入力します。［続行］をクリックします。
2. **CloudBridge Connector** 設定ペインで、次のパラメータを設定します。
   - **CloudBridge Connector** 名—ローカルアプライアンス上の CloudBridge Connector 設定の名前。ASCII アルファベット文字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) のみを含める必要があります。CloudBridge Connector の設定を作成した後は変更できません。
3. 「ローカル設定」で、次のパラメータを設定します。
   - サブネット **IP**—CloudBridge Connector トンネルのローカルエンドポイントの IP アドレス。
4. 「リモート設定」で、次のパラメータを設定します。
   - サブネット **IP**—CloudBridge Connector トンネルのピアエンドポイントの IP アドレス。
5. **PBR** 設定で、次のパラメータを設定します。
   - 操作—等しい (=) または等しくない (!=) 論理演算子。
   - 送信元 **IP Low**：発信 IPv4 パケットの送信元 IP アドレスと一致する最も低い送信元 IP アドレス。
   - 送信元 **IP** 高：発信 IPv4 パケットの送信元 IP アドレスと一致する最大の送信元 IP アドレス。
   - 操作—等しい (=) または等しくない (!=) 論理演算子。
   - 宛先 IP Low*：発信 IPv4 パケットの宛先 IP アドレスと一致する最も低い宛先 IP アドレス。
   - 宛先 **IP** 高：発信 IPv4 パケットの宛先 IP アドレスと一致する最大の宛先 IP アドレス。
6. (オプション) [セキュリティ設定] で、CloudBridge Connector トンネルの次の IPsec プロトコルパラメータを設定します。
   - 暗号化アルゴリズム—CloudBridge トンネルの IPsec プロトコルで使用される暗号化アルゴリズム。
   - ハッシュアルゴリズム—CloudBridge トンネルの IPsec プロトコルで使用されるハッシュアルゴリズム。
   - キー—次の IPsec 認証方法のいずれかを選択して、2 つのピアが相互認証に使用します。
     - 自動生成キー—ローカルアプライアンスによって自動的に生成される事前共有キー (PSK) と呼ばれるテキスト文字列に基づく認証。ピアの PSK キーは相互に照合されて認証されます。
     - 特定キー：手動で入力した PSK に基づく認証。ピアの PSK は相互に照合されて認証されます。
       * 事前共有セキュリティキー—事前共有キーベースの認証用に入力されるテキスト文字列。
     - 証明書のアップロード—デジタル証明書に基づく認証。
       * 公開鍵—IPsec セキュリティアソシエーションを確立する前に、ローカルの NetScaler アプライアンスをピアに対して認証するために使用されるローカルデジタル証明書です。同じ証明書が存在し、ピアのピア公開鍵パラメータに設定されている必要があります。
       * 秘密鍵—ローカルデジタル証明書の秘密鍵。

* ピア公開鍵：ピアのデジタル証明書。IPsec セキュリティアソシエーションを確立する前に、ローカルエンドポイントへのピアを認証するために使用されます。ピアの Public key パラメータにも同じ証明書が存在し、設定されている必要があります。

7. ［完了］をクリックします。

両方の NetScaler アプライアンスの新しい CloudBridge Connector トンネル構成は、それぞれの GUI の [ホーム] タブに表示されます。CloudBridge Connector トンネルの現在のステータスは、設定済みの CloudBridge Connector ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、NetScaler ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。NetScaler アプライアンスでの CloudBridge Connector のトンネル統計の表示について詳しくは、「CloudBridge Connector トンネルの監視」を参照してください。

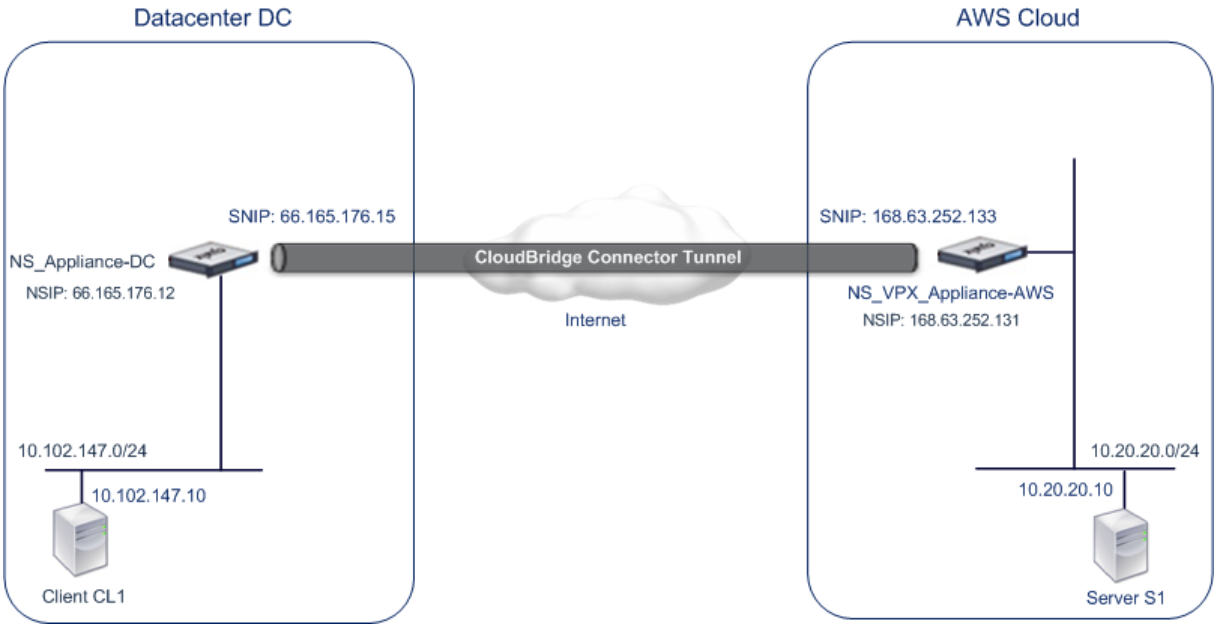
データセンターと **AWS** クラウド間の **CloudBridge Connector** の設定

August 15, 2023

データセンターと AWS クラウドの間に CloudBridge Connector トンネルを設定して、データセンターと AWS クラウドのインフラストラクチャとコンピューティング機能を活用できます。AWS を使用すると、初期投資や拡張されたネットワークインフラストラクチャの維持費用をかけずに、ネットワークを拡張できます。インフラストラクチャは必要に応じてスケールアップまたはスケールダウンできます。たとえば、需要が増えたときには、より多くのサーバー機能をリースできます。

データセンターを AWS クラウドに接続するには、データセンターにある NetScaler アプライアンスと AWS クラウドにある NetScaler 仮想アプライアンス（VPX）の間に CloudBridge Connector トンネルを設定します。

データセンターと Amazon AWS クラウド間の CloudBridge Connector トンネルの例として、データセンター DC の NetScaler アプライアンス NS_Appliance-DC と NetScaler 仮想アプライアンス（VPX）NS_VPX_Appliance-AWS の間に CloudBridge Connector トンネルが設定されている例を考えてみましょう。



NS_Appliance-DC と NS_VPX_Appliance-AWS はどちらも L3 モードで機能します。データセンター DC のプライベートネットワークと AWS クラウド間の通信を可能にします。NS_Appliance-DC と NS_VPX_Appliance-AWS は、CloudBridge Connector トンネルを介して、データセンター DC のクライアント CL1 と AWS クラウド内のサーバー S1 間の通信を可能にします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

> 注記:
>
> AWS は L2 モードをサポートしていないため、両方のエンドポイントで L3 モードのみを有効にする必要があります。

CL1 と S1 間の通信が正しく行われるように、NS_Appliance-DC と NS_VPX_Appliance-AWS では L3 モードが有効になっており、ルートは次のように更新されます。

- CL1 には、S1 に到達するための NS_Appliance-DC へのルートがあります。
- NS_Appliance-DC には S1 に到達するための NS_VPX_Appliance-AWS へのルートがあります。
- S1 には CL1 に到達するための NS_VPX_Appliance-AWS へのルートが必要です。
- NS_VPX_Appliance-AWS には、CL1 に到達するための NS_Appliance-DC へのルートがあります。

次の表は、データセンター DC の NetScaler アプライアンス NS_Appliance-DC の設定を示しています。

エンティティ	名前	詳細
NSIP アドレス		66.165.176.12
SNIP アドレス		66.165.176.15

エンティティ	名前	詳細
CloudBridge Connector トンネル	CC_Tunnel_DC-AWS	CloudBridge Connector トンネルのローカルエンドポイント IP アドレス:66.165.176.15、CloudBridge Connector トンネルのリモートエンドポイント IP アドレス:168.63.252.133、GRE トンネルの詳細-名前 = CC_Tunnel_DC-AWS

次の表は、AWS クラウド上の NetScaler VPX NS_VPX_Appliance-AWS の設定を示しています。

エンティティ	名前	詳細
NSIP アドレス		10.102.25.30
NSIP アドレスにマップされたパブリック EIP アドレス		168.63.252.131
SNIP アドレス		10.102.29.30
SNIP アドレスにマップされたパブリック EIP アドレス		168.63.252.133
CloudBridge Connector トンネル	CC_Tunnel_DC-AWS	CloudBridge Connector トンネルのローカルエンドポイント IP アドレス:168.63.252.133、CloudBridge Connector トンネルのリモートエンドポイント IP アドレス:66.165.176.15; **GRE** トンネルの詳細名前 = CC_Tunnel_DC-AWS、IPsec プロファイルの詳細、名前 = CC_Tunnel_DC-AWS、暗号化アルゴリズム = AES、ハッシュアルゴリズム = HMAC SHA1

前提条件

CloudBridge Connector トンネルをセットアップする前に、次のタスクが完了していることを確認してください。

1. AWS クラウド上で NetScaler ADC 仮想アプライアンス（VPX）のインスタンスをインストール、設定、起動

します。NetScaler VPX を AWS にインストールする手順については、「AWS での NetScaler ADC VPX インスタンスのデプロイ」を参照してください。

2. NetScaler ADC 物理アプライアンスを展開して構成するか、またはデータセンター内の仮想化プラットフォームで NetScaler ADC 仮想アプライアンス（VPX）を Provisioning して構成します。
3. CloudBridge Connector のトンネルエンドポイントの IP アドレスが相互にアクセスできることを確認してください。

NetScaler VPX ライセンス

インスタンスの初期起動後、NetScaler VPX for AWS にはライセンスが必要です。独自のライセンス (BYOL) を持参する場合は、http://support.citrix.com/article/CTX122426の VPX ライセンスガイドを参照してください。

次の操作を実行する必要があります。

1. Citrix Web サイト内のライセンスポータルを使用して、有効なライセンスを生成します。
2. ライセンスをインスタンスにアップロードします。

有料 マーケットプレイスインスタンスの場合は、ライセンスをインストールする必要はありません。該当する機能セットとパフォーマンスが自動的にアクティブ化されます。

構成の手順

データセンターにある NetScaler アプライアンスと AWS クラウド上にある NetScaler 仮想アプライアンス（VPX）の間に CloudBridge Connector トンネルを設定するには、NetScaler アプライアンスの GUI を使用します。

GUI を使用すると、NetScaler アプライアンスで作成された CloudBridge Connector トンネル構成は、CloudBridge Connector トンネルの他のエンドポイントまたはピア（AWS 上の NetScaler VPX）に自動的にプッシュされます。そのため、AWS 上の NetScaler VPX GUI（GUI）にアクセスして、対応する CloudBridge Connector トンネル構成を作成する必要はありません。

両方のピア（データセンターにある NetScaler アプライアンスと AWS クラウドにある NetScaler 仮想アプライアンス（VPX））の CloudBridge Connector トンネル構成は、次のエンティティで構成されています。

- **IPsec** プロファイル—IPsec プロファイルエンティティは、CloudBridge Connector トンネルの両方のピアの IPsec プロトコルが使用する IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、PSK などの IPsec プロトコルパラメータを指定します。
- **GRE** トンネル—IP トンネルは、ローカル IP アドレス (ローカルピアに設定されたパブリック SNIP アドレス)、リモート IP アドレス (リモートピアに設定されたパブリック SNIP アドレス)、CloudBridge Connector トンネルのセットアップに使用されるプロトコル (GRE)、および IPsec プロファイルエンティティを指定します。

- **PBR** ルールを作成し、それに **IP** トンネルを関連付けます。PBR エンティティは、条件セットと IP トンネルエンティティを指定します。送信元 IP アドレスの範囲と宛先 IP の範囲は、PBR エンティティの条件です。送信元 IP アドレス範囲と宛先 IP アドレス範囲を設定して、トラフィックが CloudBridge Connector トンネルを通過するサブネットを指定する必要があります。たとえば、データセンター内のサブネット上のクライアントから発信され、AWS クラウド内のサブネット上のサーバー宛てのリクエストパケットがあるとします。このパケットがデータセンターの NetScaler アプライアンス上の PBR エンティティの送信元および宛先 IP アドレス範囲と一致する場合、PBR エンティティに関連付けられた CloudBridge Connector トンネルを介して送信されます。

コマンドラインインターフェイスを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで入力します。

- `add ipsec profile <name> [-**ikeVersion** (V1 | V2)] [-**encAlgo** (AES | 3DES)...] [-**hashAlgo** <hashAlgo> ...] [-**lifetime** <positive_integer>] (-**psk** | (-**publickey** <string> -**privatekey** <string> -**peerPublicKey** <string>)) [-**livenessCheckInterval** <positive_integer>] [-**replayWindowSize** <positive_integer>] [-**ikeRetryInterval** <positive_integer>] [-**retransmissiontime** <positive_integer>]`
- `**show ipsec profile** <name>`

コマンドラインインターフェイスを使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote><remoteSubnetMask> <local> [-protocol <protocol>] [-ipsecProfileName <string>]`
- `show ipTunnel <name>`

コマンドラインインターフェイスを使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

コマンドプロンプトで入力します。

- `add ns pbr <pbr_name> ALLOW -srcIP = <local_subnet_range> -destIP = <remote_subnet_range> -ipTunnel <tunnel_name>`
- `apply ns pbrs`
- `show ns pbr <pbr_name>`

例

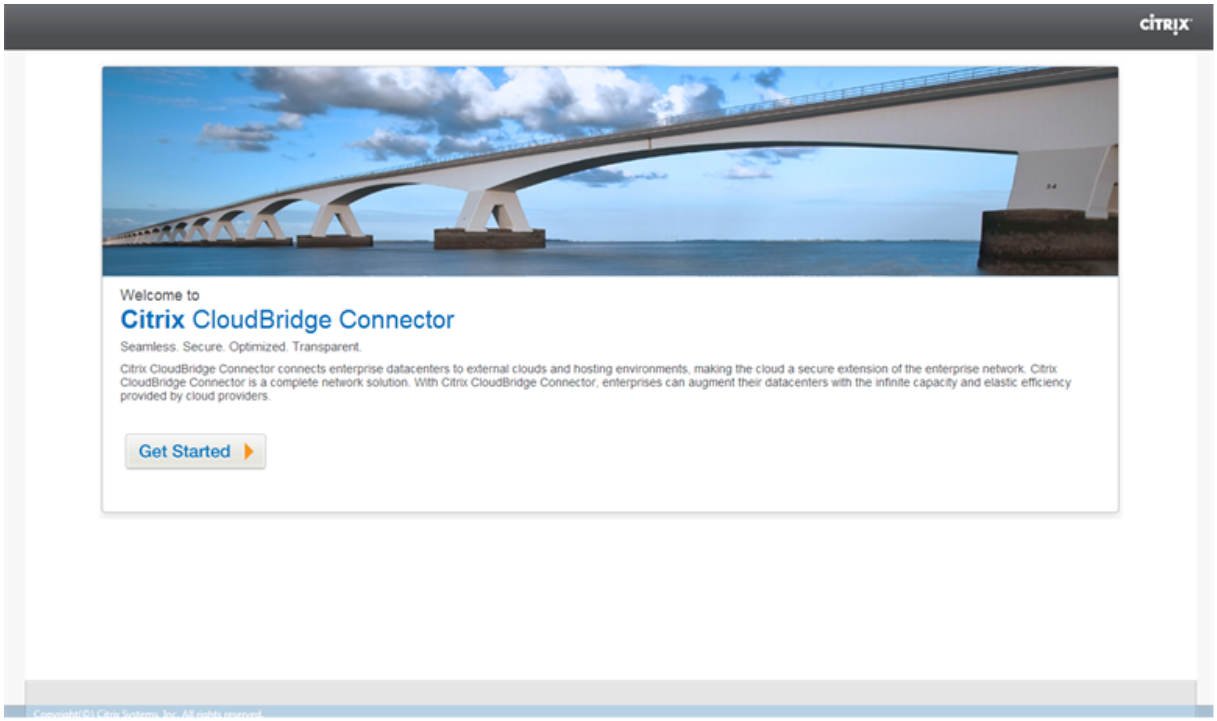
```
1      > add ipsec profile CC_Tunnel_DC-AWS -encAlgo AES -hashAlgo
          HMAC_SHA1
2
```



```
3       Done
4       > add ipTunnel CC_Tunnel_DC-AWS 168.63.252.133 255.255.255.0
            66.165.176.15  -protocol GRE -ipsecProfileName CC_Tunnel_DC-AWS
5
6       Done
7       > add ns pbr PBR-DC-AWS ALLOW  -srcIP 66.165.176.15  -destIP
            168.63.252.133 ipTunnel CC_Tunnel_DC-AWS
8
9       Done
10      > apply ns pbrs
11
12      Done
13  <!--NeedCopy-->
```

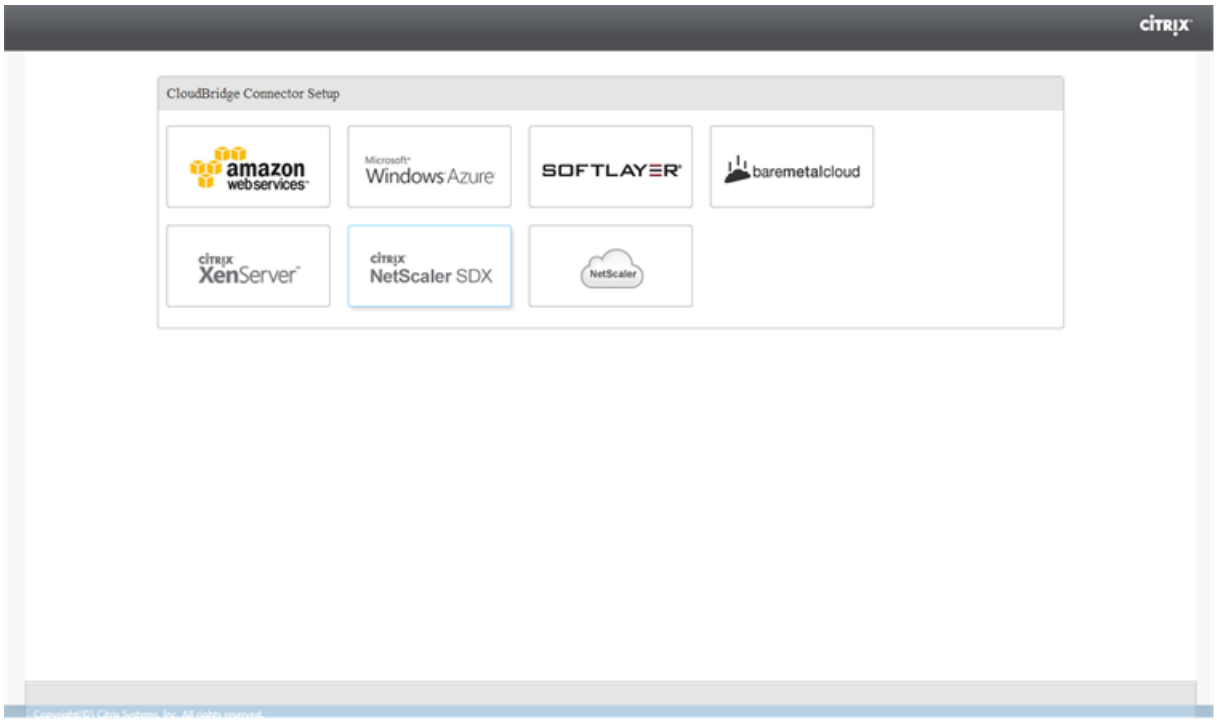
GUI を使用して NetScaler アプライアンスの CloudBridge Connector トンネルを構成するには

1. Web ブラウザーのアドレス行に NetScaler アプライアンスの NSIP アドレスを入力します。
2. アプライアンスのアカウント認証情報を使用して、NetScaler アプライアンスの GUI にログオンします。
3. [システム] > [**CloudBridge**Connector] に移動します。
4. 右側のペインの「はじめに」で、「CloudBridge の作成/監視」をクリックします。
5. アプライアンスで CloudBridge Connector トンネルを初めて設定すると、ウェルカム画面が表示されます。
6. 「ようこそ」画面で、「はじめに」をクリックします。



> 注記:
>
> NetScaler アプライアンスですでに CloudBridge Connector トンネルを構成している場合は、「ようこそ」画面は表示されないので、「はじめに」をクリックしないでください。

1. **CloudBridge Connector** のセットアップペインで、**Amazon** ウェブサービスをクリックします



1. **Amazon** ペインで、AWS アカウントの認証情報 (AWS アクセスキー ID と AWS シークレットアクセスキー) を入力します。これらのアクセスキーは AWS GUI コンソールから取得できます。［続行］をクリックします。

> 注
>
> 以前は、別のリージョンを選択しても、セットアップウィザードは常に同じ AWS リージョンに接続していました。その結果、以前は、選択した AWS リージョンで実行されている NetScaler VPX への CloudBridge Connector トンネルの設定が失敗していました。この問題は修正されました。

1. **NetScaler** ペインで、AWS 上で実行されている NetScaler 仮想アプライアンスの NSIP アドレスを選択します。次に、NetScaler 仮想アプライアンスのアカウント認証情報を入力します。［続行］をクリックします。
2. **CloudBridge Connector** 設定ペインで、次のパラメータを設定します。
   - **CloudBridge Connector** 名—ローカルアプライアンス上の CloudBridge Connector 設定の名前。ASCII アルファベット文字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、およびハイフン (-) のみを含める必要があります。CloudBridge Connector の設定を作成した後は変更できません。
3. 「ローカル設定」で、次のパラメータを設定します。

- サブネット **IP**—CloudBridge Connector トンネルのローカルエンドポイントの IP アドレス。SNIP タイプのパブリック IP アドレスでなければなりません。

4. 「リモート設定」で、次のパラメータを設定します。

   - サブネット **IP**—AWS 側の CloudBridge Connector トンネルエンドポイントの IP アドレス。AWS 上の NetScaler VPX インスタンス上の SNIP タイプの IP アドレスである必要があります。
   - **NAT**—AWS 上の NetScaler VPX インスタンスで構成された SNIP にマップされている AWS のパブリック IP アドレス（EIP）。

5. [**PBR** 設定**]** で、次のパラメータを設定します。

   - 操作—等しい (=) または等しくない (!=) 論理演算子。
   - 送信元 **IP Low**：発信 IPv4 パケットの送信元 IP アドレスと一致する最も低い送信元 IP アドレス。
   - 送信元 **IP** 高：発信 IPv4 パケットの送信元 IP アドレスと一致する最大の送信元 IP アドレス。
   - 操作—等しい (=) または等しくない (!=) 論理演算子。
   - 宛先 **IP Low**：発信 IPv4 パケットの宛先 IP アドレスと一致する最も低い宛先 IP アドレス。
   - 宛先 **IP** 高：発信 IPv4 パケットの宛先 IP アドレスと一致する最大の宛先 IP アドレス。

6. (オプション) [セキュリティ設定**]** で、CloudBridge Connector トンネルの次の IPsec プロトコルパラメータを設定します。

   - 暗号化アルゴリズム—CloudBridge トンネルの IPsec プロトコルで使用される暗号化アルゴリズム。
   - ハッシュアルゴリズム—CloudBridge トンネルの IPsec プロトコルで使用されるハッシュアルゴリズム。
   - キー—次の IPsec 認証方法のいずれかを選択して、2 つのピアが相互認証に使用します。
     - 自動生成キー—ローカルアプライアンスによって自動的に生成される事前共有キー (PSK) と呼ばれるテキスト文字列に基づく認証。ピアの PSK キーは相互に照合されて認証されます。
     - 特定キー：手動で入力した PSK に基づく認証。ピアの PSK は相互に照合されて認証されます。
       * 事前共有セキュリティキー—事前共有キーベースの認証用に入力されるテキスト文字列。
     - 証明書のアップロード—デジタル証明書に基づく認証。
       * 公開鍵：IPsec セキュリティアソシエーションを確立する前に、ローカルピアをリモートピアに対して認証するために使用するローカルデジタル証明書。同じ証明書が存在し、ピアのピア公開鍵パラメータに設定されている必要があります。
       * 秘密鍵—ローカルデジタル証明書の秘密鍵。
       * ピア公開鍵：ピアのデジタル証明書。IPsec セキュリティアソシエーションを確立する前に、ローカルエンドポイントへのピアを認証するために使用されます。ピアの Public key パラメータにも同じ証明書が存在し、設定されている必要があります。

7. ［完了］をクリックします。

データセンターの NetScaler アプライアンス上の新しい CloudBridge Connector トンネル構成は、GUI の [ホーム] タブに表示されます。AWS クラウドの NetScaler VPX アプライアンス上の対応する新しい CloudBridge

Connector トンネル構成が GUI に表示されます。CloudBridge Connector トンネルの現在のステータスは、設定済みの CloudBridge ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、NetScaler ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。NetScaler アプライアンスでの CloudBridge Connector のトンネル統計の表示について詳しくは、「CloudBridge Connector トンネルの監視」を参照してください。

NetScaler アプライアンスと **AWS** 上の仮想プライベートゲートウェイ間の **CloudBridge Connector** トンネルの設定

August 15, 2023

データセンターを Amazon Web Services（AWS）に接続するには、データセンターの NetScaler アプライアンスと AWS 上の仮想プライベートゲートウェイの間に CloudBridge Connector トンネルを構成できます。NetScaler アプライアンスと仮想プライベートゲートウェイは CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

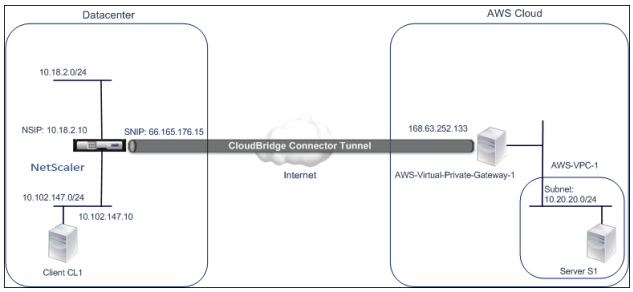
> 注:
>
> また、データセンター内の NetScaler ADC アプライアンスと（仮想プライベート Gateway ではなく）AWS 上の NetScaler ADC VPX インスタンスとの間に CloudBridge Connector トンネルを設定することもできます。詳細については、「データセンターと AWS クラウド間の CloudBridge Connector の設定」を参照してください。

AWS の仮想プライベートゲートウェイは、CloudBridge Connector トンネルに対して次の IPSec 設定をサポートしています。そのため、CloudBridge Connector トンネル用に NetScaler アプライアンスを構成するときには、同じ IPsec 設定を指定する必要があります。

IPSec のプロパティ	設定
IPsec モード	トンネルモード
IKE バージョン	バージョン 1
IKE 認証方法	事前共有キー
暗号化アルゴリズム	AES
ハッシュアルゴリズム	HMAC SHA1

CloudBridge Connector のトンネル構成とデータフローの例

CloudBridge Connector トンネル内のトラフィックフローの図として、データセンター内の NetScaler アプライアンス NS_Appliance-1 と AWS クラウド上の仮想プライベートゲートウェイ AWS-Virtual-Private-Gateway-1 の間に CloudBridge Connector トンネルが設定されている例を考えてみましょう。



NS_Appliance-1 は L3 ルーターとしても機能し、データセンターのプライベートネットワークが CloudBridge Connector トンネルを経由して AWS クラウドのプライベートネットワークに到達できるようにします。NS_Appliance-1 はルーターとして、CloudBridge Connector トンネルを介してデータセンターのクライアント CL1 と AWS クラウドのサーバー S1 間の通信を可能にします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector のトンネル設定には、ns_AWS_IPsec_Profile という名前の IPsec プロファイルエンティティ、ns_AWS_Tunnel という名前の CloudBridge Connector トンネルエンティティ、および ns_AWS_PBR という名前のポリシーベースルーティング (PBR) エンティティが含まれます。

IPsec プロファイルエンティティ ns_AWS_IPsec_Profile は、CloudBridge Connector トンネル内の IPsec プロトコルが使用する IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズムなどの IPsec プロトコルパラメータを指定します。NS_AWS_IPSec_Profile は IP トンネルエンティティ NS_AWS_Tunnel にバインドされています。

CloudBridge Connector のトンネルエンティティ NS_AWS_Tunnel は、ローカル IP アドレス（NetScaler アプライアンスで構成されたパブリック IP-SNIP アドレス）、リモート IP アドレス（AWS-仮想-プライベートゲートウェイ-1 の IP アドレス）、および CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPsec）を指定します。NS_AWS_Tunnel はポリシーベースルーティング (PBR) エンティティ ns_AWS_PBR にバインドされています。

PBR エンティティ ns_AWS_PBR は、一連の条件と CloudBridge Connector トンネルエンティティ (ns_AWS_Tunnel) を指定します。送信元 IP アドレス範囲と宛先 IP アドレス範囲が NS_aws_PBR の条件です。送信元 IP アドレス範囲と宛先 IP アドレス範囲は、それぞれデータセンターのサブネットと AWS クラウドのサブネットとして指定されます。データセンターのサブネット内のクライアントから送信され、AWS クラウド上のサブネット内のサーバー宛てのリクエストパケットはすべて、NS_AWS_PBR の条件と一致します。その後、このパケットは CloudBridge Connector の処理の対象と見なされ、PBR エンティティにバインドされた CloudBridge Connector トンネル（NS_AWS_Tunnel）を介して送信されます。

次の表に、この例で使用されている設定の一覧を示します。

データセンター側の CloudBridge Connector トンネルエンドポイント (NS_Appliance-1) の IP アドレス	66.165.176.15
AWS の CloudBridge Connector トンネルエンドポイント (AWS-仮想プライベートゲートウェイ-1) の IP アドレス	168.63.252.133
トラフィックが CloudBridge Connector トンネルを通過するデータセンターのサブネット	10.102.147.0/24
トラフィックが CloudBridge Connector トンネルを通過する AWS サブネット	10.20.20.0/24

Amazon AWS の設定

カスタマーゲートウェイ	AWS-Customer-Gateway-1	ルーティング = 固定、IP アドレス = インターネットでルーティング可能な CloudBridge Connector のトンネルエンドポイントの NetScaler 側の IP アドレス = 66.165.176.15
仮想プライベートゲートウェイ	AWS-Virtual-Private-Gateway-1	関連する VPC = AWS-VPC-1
VPN コネクション	AWS-VPN-Connection-1	カスタマーゲートウェイ = AWS-カスタマーゲートウェイ-1、仮想プライベートゲートウェイ = 仮想プライベートゲートウェイ-1、ルーティングオプション: タイプ = 静的、静的 IP プレフィックス = NetScaler 側のサブネット = 10.102.147.0/24

データセンター **1** の **NetScaler** アプライアンス **NS_Appliance-1** の設定:

アプライアンス	設定
SNIP1 (参照のみを目的としています)	66.165.176.15
IPSec profile	NS_AWS_IPSec_Profile	IKE version = v1, Encryption algorithm = AES, Hash algorithm = HMAC SHA1
CloudBridge Connector tunnel	NS_AWS_Tunnel	Remote IP = 168.63.252.133, Local IP= 66.165.176.15, Tunnel protocol = IPSec, IPSec profile= NS_AWS_IPSec_Profile
Policy based route	NS_AWS_Pbr	Source IP range = Subnet in the datacenter =10.102.147.0-10.102.147.255, Destination IP range =Subnet in AWS =10.20.20.0-10.20.20.255, IP Tunnel = NS_AWS_Tunnel

CloudBridge Connector のトンネル設定について考慮すべきポイント

NetScaler アプライアンスと AWS ゲートウェイ間の CloudBridge Connector トンネルを構成する前に、次の点を考慮してください。

1. AWS は、CloudBridge Connector トンネルの次の IPsec 設定をサポートしています。そのため、CloudBridge Connector トンネル用に NetScaler アプライアンスを構成するときには、同じ IPsec 設定を指定する必要があります。

    - IKE バージョン = v1
    - 暗号化アルゴリズム = AES
    - ハッシュアルゴリズム = HMAC SHA1

2. 次のことを許可するには、NetScaler 側でファイアウォールを構成する必要があります。

    - ポート 500 の任意の UDP パケット
    - ポート 4500 の任意の UDP パケット
    - 任意の ESP（IP プロトコル番号 50）パケット

3. AWS でトンネル構成を設定すると、トンネルの AWS エンド（ゲートウェイ）と PSK のパブリック IP アドレスが自動的に生成されるため、NetScaler でトンネル構成を指定する前に Amazon AWS を構成する必要があります。この情報は、NetScaler アプライアンスのトンネル構成を指定するために必要です。

4. AWS ゲートウェイは、スタティックルートとルート更新用の BGP プロトコルをサポートしています。NetScaler アプライアンスは、AWS ゲートウェイへの CloudBridge Connector トンネル内の BGP プロトコルをサポートしていません。したがって、トンネルを介してトラフィックを適切にルーティングするには、CloudBridge Connector トンネルの両側で適切な静的ルートを使用する必要があります。

CloudBridge Connector トンネル用の **Amazon AWS** の設定

Amazon AWS で CloudBridge Connector トンネル設定を作成するには、Amazon AWS マネジメントコンソールを使用します。これは Amazon AWS でリソースを作成および管理するためのウェブベースのグラフィカルインターフェイスです。

AWS クラウドで CloudBridge Connector のトンネル設定を開始する前に、次のことを確認してください。

- Amazon AWS クラウドのユーザーアカウントを持っています。
- 仮想プライベートクラウドがあり、そのネットワークを CloudBridge Connector トンネルを介して NetScaler 側のネットワークに接続する必要があります。
- Amazon AWS マネジメントコンソールに精通している。

> 注記:
>
> CloudBridge Connector のトンネル用に Amazon AWS を設定する手順は、Amazon AWS のリリースサイクルに応じて時間の経過とともに変化する可能性があります。最新の手順については、Amazon AWS のドキ

> ュメントを参照することをお勧めします 。

NetScaler と AWS Gateway 間の CloudBridge Connector トンネルを設定するには、AWS マネジメントコンソールで以下のタスクを実行します。

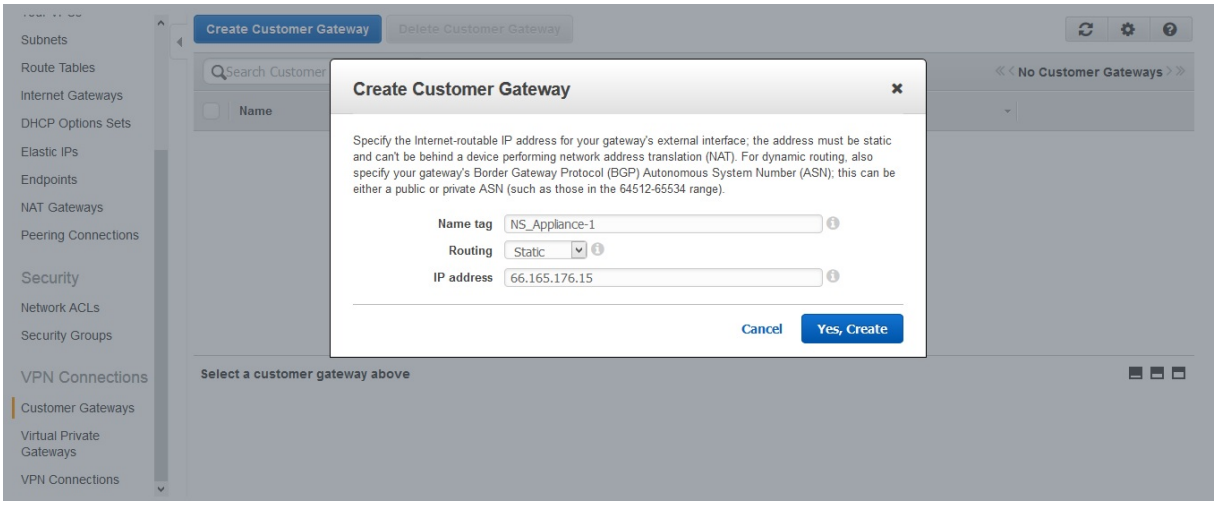
- カスタマーゲートウェイを作成します。カスタマーゲートウェイは、CloudBridge Connector のトンネルエンドポイントを表す AWS エンティティです。NetScaler アプライアンスと AWS ゲートウェイ間の CloudBridge Connector トンネルの場合、カスタマーゲートウェイは AWS 上の NetScaler アプライアンスを表します。カスタマーゲートウェイは、名前、トンネルで使用されるルーティングのタイプ（静的または BGP）、および NetScaler 側の CloudBridge Connector トンネルエンドポイントの IP アドレスを指定します。IP アドレスは、インターネットでルーティング可能な NetScaler 所有のサブネット IP（SNIP）アドレスでも、NetScaler アプライアンスが NAT デバイスの背後にある場合は、SNIP アドレスを表すインターネットルーティング可能な NAT IP アドレスでもかまいません。
- 仮想プライベートゲートウェイを作成して **VPC** にアタッチします。仮想プライベートゲートウェイは AWS 側の CloudBridge Connector トンネルエンドポイントです。仮想プライベートゲートウェイを作成するときは、そのゲートウェイに名前を割り当てるか、AWS に名前の割り当てを許可します。次に、仮想プライベートゲートウェイを VPC に関連付けます。この関連付けにより、VPC のサブネットは、CloudBridge Connector トンネルを介して NetScaler 側のサブネットに接続できます。
- **VPN** 接続を作成します。VPN 接続は、カスタマーゲートウェイと仮想プライベートゲートウェイを指定し、その間に CloudBridge Connector トンネルを作成します。また、NetScaler 側のネットワークの IP プレフィックスも指定します。（静的ルートエントリを通じて）仮想プライベートゲートウェイに認識されている IP プレフィックスだけが、トンネルを介して VPC からトラフィックを受信できます。また、仮想プライベートゲートウェイは、指定された IP プレフィックスを宛先としないトラフィックをトンネル経由でルーティングしません。VPN 接続を設定した後、接続が作成されるまで数分待たなければならない場合があります。
- ルーティングオプションを設定します。VPC のネットワークが CloudBridge Connector トンネルを介して NetScaler 側のネットワークに到達するには、NetScaler 側のネットワークへのルートを含めて、それらのルートが仮想プライベートゲートウェイを指すように VPC のルーティングテーブルを構成する必要があります。次のいずれかの方法で VPC のルーティングテーブルにルートを含めることができます。
  - ルート伝播を有効にします。ルーティングテーブルのルート伝播を有効にすると、ルートが自動的にテーブルに伝播されます。VPN 構成用に指定した固定 IP プレフィックスは、VPN 接続を作成した後にルーティングテーブルに伝達されます。
  - スタティックルートを手動で入力します。ルート伝播を有効にしない場合は、NetScaler 側のネットワークの静的ルートを手動で入力する必要があります。
- 設定をダウンロードします。AWS で CloudBridge Connector トンネル (VPN 接続) 設定が作成されたら、VPN 接続の設定ファイルをローカルシステムにダウンロードします。NetScaler アプライアンスで CloudBridge Connector トンネルを構成するには、構成ファイル内の情報が必要になる場合があります。

カスタマーゲートウェイを作成するには

1. https://console.aws.amazon.com/vpc/で Amazon VPC コンソールを開きます。
2. [**VPN** 接続] > [カスタマーゲートウェイ] に移動し、[カスタマーゲートウェイの作成] をクリックします。

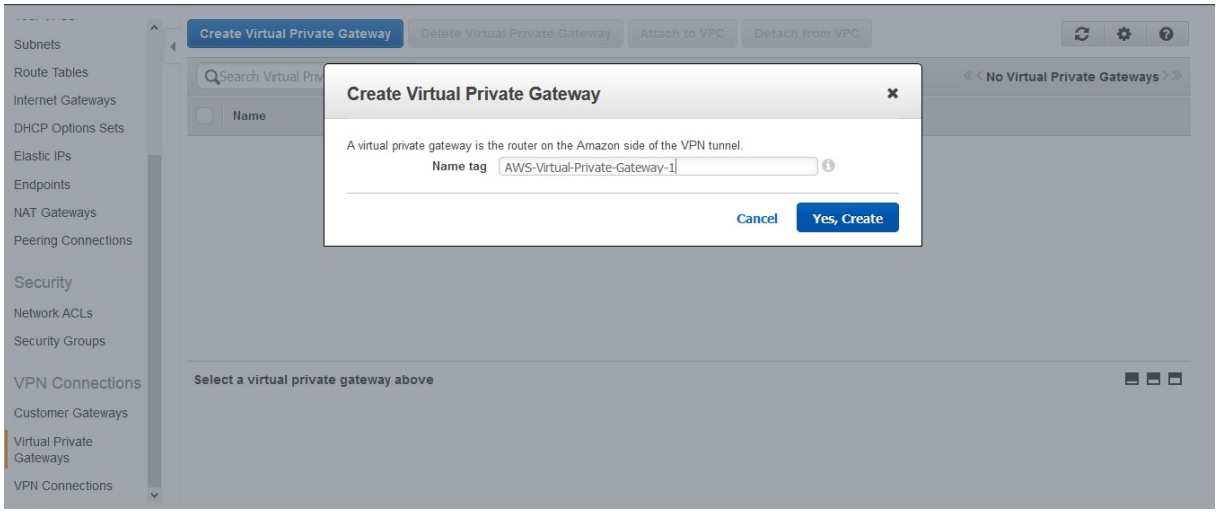
3. 「カスタマーゲートウェイの作成」ダイアログで、次のパラメータを設定し、「はい、作成」をクリックします。

   - ネームタグ。カスタマーゲートウェイの名前。
   - ルーティングリスト。CloudBridge Connector トンネルを介して相互にルートをアドバタイズするための NetScaler アプライアンスと AWS 仮想プライベートゲートウェイ間のルーティングのタイプ。** ルーティングリストから「スタティックルーティング ** 」を選択します。注：NetScaler アプライアンスは、AWS ゲートウェイへの CloudBridge Connector トンネル内の BGP プロトコルをサポートしていません。したがって、トンネルを介してトラフィックを適切にルーティングするには、CloudBridge Connector トンネルの両側で適切な静的ルートを使用する必要があります。
   - **IP** アドレス。NetScaler 側のインターネットルーティング可能な CloudBridge Connector のトンネルエンドポイント IP アドレス。IP アドレスは、インターネットでルーティング可能な NetScaler 所有のサブネット IP（SNIP）アドレスでも、NetScaler アプライアンスが NAT デバイスの背後にある場合は、SNIP アドレスを表すインターネットルーティング可能な NAT IP アドレスでもかまいません。

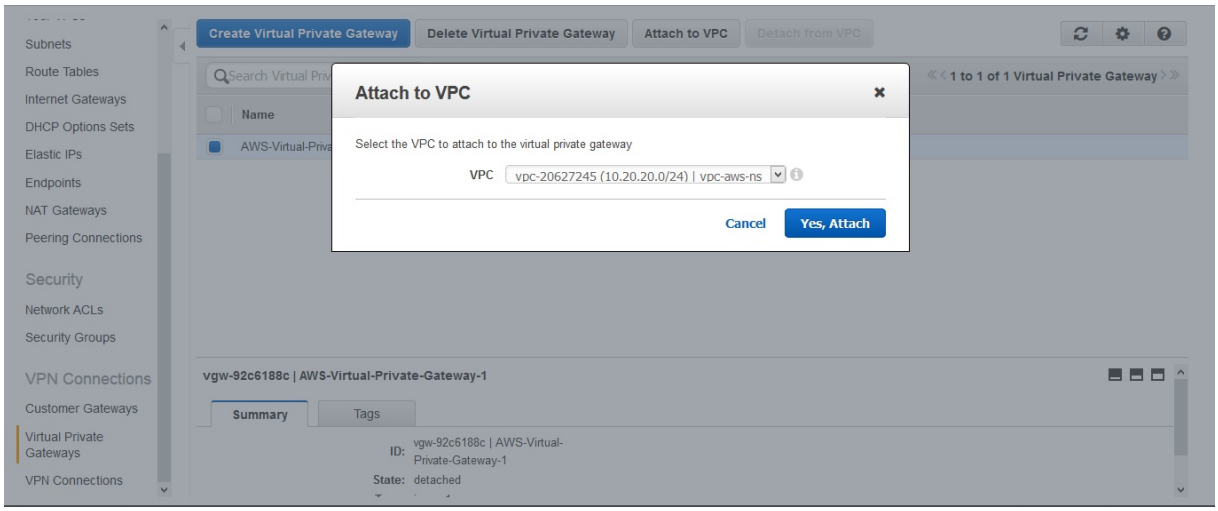


仮想プライベートゲートウェイを作成して VPC にアタッチするには

1. [**VPN** 接続] > [仮想プライベートゲートウェイ] に移動し、[仮想プライベートゲートウェイの作成] をクリックします。
2. 仮想プライベートゲートウェイの名前を入力し、[Yes, Create] をクリックします。



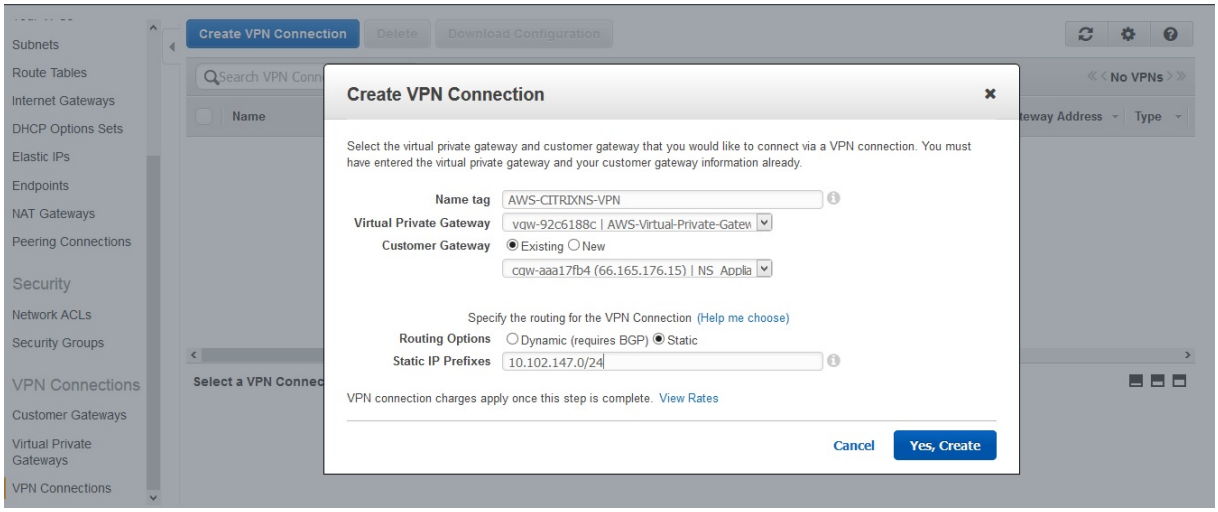
1. 作成した仮想プライベートゲートウェイを選択し、[Attach to VPC] をクリックします。
2. [VPC に接続] ダイアログボックスで、一覧から VPC を選択し、[Yes, Attach] を選択します。



VPN 接続を作成するには:

1. [VPN 接続] > [VPN 接続] に移動し、[VPN 接続の作成] をクリックします。
2. 「VPN 接続の作成」ダイアログボックスで、次のパラメータを設定し、「はい、作成」を選択します。

   - ネームタグ。VPN 接続の名前。
   - 仮想プライベートゲートウェイ。前に作成した仮想プライベートゲートウェイを選択します。
   - カスタマーゲートウェイ。「既存」を選択します。次に、ドロップダウンリストから、先ほど作成したカスタマーゲートウェイを選択します。
   - ルーティングオプション。仮想プライベートゲートウェイとカスタマーゲートウェイ（NetScaler アプライアンス）間のルーティングのタイプ。「スタティック」を選択します。静的 IP プレフィックスフィールドで、NetScaler 側のサブネットの IP プレフィックスをカンマで区切って指定します。



ルート伝播を有効にするには:

1. ルートテーブルに移動し、トラフィックが CloudBridge Connector トンネルを通過するサブネットに関連付けられているルーティングテーブルを選択します。

> 注
>
> デフォルトでは、これが VPC のメインルーティングテーブルです。

1. 詳細ペインの [ルート伝達] タブで [編集] を選択し、仮想プライベートゲートウェイを選択してから [保存] を選択します。

スタティックルートを手動で入力するには:

1. 「ルートテーブル」に移動し、ルーティングテーブルを選択します。
2. [ルート] タブで、[編集] をクリックします。
3. 宛先フィールドに、CloudBridge Connector トンネル (VPN 接続) で使用される静的ルートを入力します。
4. ターゲットリストから仮想プライベートゲートウェイ ID を選択し、[保存] をクリックします。

設定ファイルをダウンロードするには:

1. [**VPN** 接続] に移動し、VPN 接続を選択し、[設定のダウンロード] をクリックします。
2. ダウンロード設定ダイアログボックスで、次のパラメータを設定し、「はい、ダウンロード」をクリックします。
   - ベンダー。「ジェネリック」を選択します。
   - プラットフォーム。「ジェネリック」を選択します。
   - ソフトウェア。「ベンダー・アグノスティック」を選択します。

CloudBridge Connector トンネル用の **NetScaler ADC** アプライアンスの構成

NetScaler アプライアンスと AWS クラウド上の仮想プライベートゲートウェイ間の CloudBridge Connector トンネルを構成するには、NetScaler アプライアンスで次のタスクを実行します。
NetScaler のコマンドラインまたは GUI のどちらかを使用できます。

- **IPsec** プロファイルを作成します。IPSec プロファイルエンティティは、CloudBridge Connector トンネル内の IPsec プロトコルで使用される IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、PSK などの IPsec プロトコルパラメータを指定します。
- **IPsec** プロトコルを使用する **IP** トンネルを作成し、**IPsec** プロファイルをそれに関連付けます。IP トンネルは、ローカル IP アドレス（NetScaler アプライアンスで構成された SNIP アドレス）、リモート IP アドレス（AWS の仮想プライベートゲートウェイのパブリック IP アドレス）、CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPsec）、および IPsec プロファイルエンティティを指定します。作成された IP トンネルエンティティは、CloudBridge Connector トンネルエンティティとも呼ばれます。
- **PBR** ルールを作成して **IP** トンネルに関連付けます。PBR エンティティは、ルールセットと IP トンネル (CloudBridge Connector トンネル) エンティティを指定します。送信元 IP アドレス範囲と宛先 IP アドレス範囲は PBR エンティティの条件です。送信元 IP アドレス範囲を設定してトラフィックがトンネルを通過する NetScaler 側のサブネットを指定し、宛先 IP アドレス範囲を設定してトラフィックが CloudBridge Connector トンネルを通過する AWS VPC サブネットを指定します。NetScaler 側のサブネット内のクライアントから送信され、AWS クラウドサブネット内のサーバーを宛先とし、PBR エンティティの送信元と宛先の IP 範囲と一致するすべてのリクエストパケットは、PBR エンティティに関連付けられた CloudBridge Connector トンネルを介して送信されます。

NetScaler コマンドラインを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで、次のように入力します。

- `add ipsec profile <name> -psk <string> -**ikeVersion** v1`
- `show ipsec profile** <name>`

NetScaler コマンドラインを使用して IPSEC トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

コマンドプロンプトで、次のように入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

NetScaler コマンドラインを使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

コマンドプロンプトで、次のように入力します。

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP** <subnet-range> -*ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

次のコマンドは、「CloudBridge Connector の構成とデータフローの例」で使用されている NetScaler アプライアンス NS_Appliance-1 のすべての設定を作成します。


```
1       > add ipsec profile NS_AWS_IPSec_Profile -psk
            DkiMgMdcbqvYREEuIvxsbKkW0Foyabcd -ikeVersion v1  - lifetime
            31536000
2       Done
3       > add iptunnel NS_AWS_Tunnel 168.63.252.133 255.255.255.255
            66.165.176.15  - protocol IPSEC  - ipsecProfileName
            NS_AWS_IPSec_Profile
4
5       Done
6       > add pbr NS_AWS_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP
            10.20.0.0-10.20.255.255  - ipTunnel NS_AWS_Tunnel
7       Done
8
9       > apply pbrs
10
11      Done
12  <!--NeedCopy-->
```

GUI を使用して IPSEC プロファイルを作成するには

1. [システム] > [**CloudBridge Connector**] > [**IPSec** プロファイル] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **IPsec** プロファイルの追加ダイアログボックスで、次のパラメータを設定します。
   - 名前
   - 暗号化アルゴリズム
   - ハッシュアルゴリズム
   - IKE プロトコルバージョン (V1 を選択)
4. 事前共有キー認証方法を選択し、事前共有キーが存在するパラメーターを設定します。
5. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2. [**IPv4** トンネル] タブで、[追加] をクリックします。
3. [**IP** トンネルの追加] ダイアログボックスで、次のパラメータを設定します。
   - 名前
   - リモート IP
   - リモートマスク
   - ローカル IP タイプ (「ローカル IP タイプ」ドロップダウンリストで、「サブネット IP」を選択します)。
   - ローカル IP (選択した IP タイプの設定済み IP がすべてローカル IP ドロップダウンリストに表示されます。リストから目的の IP を選択します。)
   - プロトコル

- IPSec プロファイル

4. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3. **Create PBR** ダイアログボックスで、次のパラメータを設定します。

   - 名前
   - アクション
   - ネクストホップタイプ (IP トンネルの選択)
   - IP トンネル名
   - 送信元 IP アドレスが低い
   - ソース IP ハイ
   - デスティネーション IP フロー
   - デスティネーション IP ハイ

4. [作成] をクリックし、[閉じる] をクリックします。

NetScaler アプライアンス上の対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。

CloudBridge Connector トンネルの現在のステータスは、設定済みの CloudBridge Connector ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、NetScaler ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。
NetScaler アプライアンスでの CloudBridge Connector のトンネル統計の表示について詳しくは、「CloudBridge Connector トンネルの監視」を参照してください。

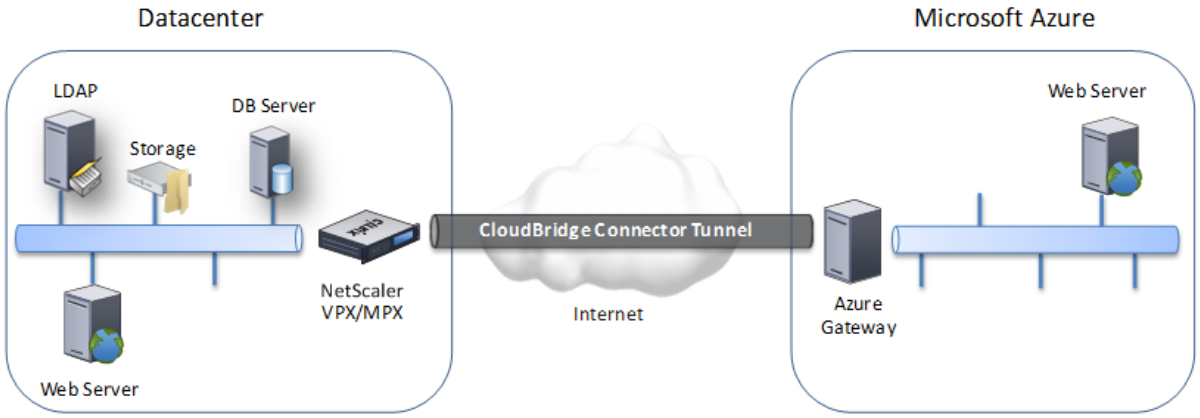
データセンターと **Azure** クラウド間の **CloudBridge Connector** トンネルの設定

August 15, 2023

NetScaler アプライアンスは、企業のデータセンターと Microsoft のクラウドホスティングプロバイダーである Azure との接続を提供するため、Azure はエンタープライズネットワークのシームレスな拡張となります。NetScaler は、エンタープライズデータセンターと Azure クラウド間の接続を暗号化して、両者間で転送されるすべてのデータを安全にします。

CloudBridge Connector トンネルの仕組み

データセンターを Azure クラウドに接続するには、データセンターにある NetScaler アプライアンスと Azure クラウドにあるゲートウェイの間に CloudBridge Connector トンネルを設定します。データセンターの NetScaler アプライアンスと Azure クラウドのゲートウェイは、CloudBridge Connector トンネルのエンドポイントであり、CloudBridge Connector トンネルのピアと呼ばれます。



データセンターと Azure クラウド間の CloudBridge Connector トンネルは、オープンスタンダードのインターネットプロトコルセキュリティ (IPsec) プロトコルスイートをトンネルモードで使用して、CloudBridge Connector トンネル内のピア間の通信を保護します。CloudBridge Connector トンネルでは、IPsec により以下が保証されます。

- データインテグリティ
- データオリジン認証
- データの機密保持 (暗号化)
- リプレイ攻撃からの保護

IPsec は、IP パケット全体を暗号化してからカプセル化するトンネルモードを使用します。暗号化には、HMAC ハッシュ関数を使用してパケットの整合性を確保し、暗号化アルゴリズムを使用して機密性を確保するカプセル化セキュリティペイロード (ESP) プロトコルを使用します。ESP プロトコルは、ペイロードを暗号化して HMAC を計算した後、ESP ヘッダーを生成し、暗号化された IP パケットの前に挿入します。ESP プロトコルは ESP トレーラーも生成し、パケットの最後に挿入します。

次に、IPsec プロトコルは ESP ヘッダーの前に IP ヘッダーを追加して、結果のパケットをカプセル化します。IP ヘッダーでは、宛先 IP アドレスは CloudBridge Connector ーピアの IP アドレスに設定されます。

CloudBridge Connector トンネル内のピアは、インターネットキーエクスチェンジバージョン 1（IKEv1）プロトコル（IPSec プロトコルスイートの一部）を使用して、次のように安全な通信をネゴシエートします。

1. 2 つのピアは、事前共有キー認証を使用して相互に認証します。事前共有キー認証では、ピアが事前共有キー (PSK) と呼ばれるテキスト文字列を交換します。事前共有鍵は相互に照合されて認証されます。したがって、認証を成功させるには、各ピアに同じ事前共有キーを設定する必要があります。
2. その後、同僚は交渉し、以下について合意に達します。

- 暗号化アルゴリズム
- 一方のピアでデータを暗号化し、もう一方のピアでデータを復号するための暗号キー。

セキュリティプロトコル、暗号化アルゴリズム、および暗号鍵に関するこの合意は、セキュリティアソシエーション（SA）と呼ばれます。SA は一方向（シンプレックス）です。たとえば、データセンターの NetScaler アプライアンスと Azure クラウド内のゲートウェイの間に CloudBridge Connector トンネルを設定すると、データセンターアプライアンスと Azure ゲートウェイの両方に 2 つの SA があります。一方の SA はアウトバウンドパケットの処理に使用され、もう 1 つの SA はインバウンドパケットの処理に使用されます。SA は、ライフタイムと呼ばれる指定された期間が経過すると期限切れになります。

CloudBridge Connector のトンネル構成とデータフローの例

CloudBridge Connector トンネルの例として、データセンターの NetScaler アプライアンス CB_Appliance-1 と Azure クラウドのゲートウェイ Azure_Gateway-1 の間に CloudBridge Connector トンネルが設定されている例を考えてみましょう。

CB_Appliance-1 は L3 ルーターとしても機能し、データセンターのプライベートネットワークが CloudBridge Connector トンネルを介して Azure クラウドのプライベートネットワークに到達できるようにします。CB_Appliance-1 はルーターとして、CloudBridge Connector トンネルを介してデータセンターのクライアント CL1 と Azure クラウドのサーバー S1 間の通信を可能にします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

CB_Appliance-1 では、CloudBridge Connector のトンネル設定には、CB_Azure_IPsec_Profile という名前の IPsec プロファイルエンティティ、CB_Azure_Tunnel という名前の CloudBridge Connector トンネルエンティティ、CB_azure_PBR という名前のポリシーベースルーティング (PBR) エンティティが含まれます。

PSec profile entity CB_Azure_IPSec_Profile は、CloudBridge Connector トンネル内の IPsec プロトコルが使用する IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズムなどの IPsec プロトコルパラメータを指定します。CB_Azure_IPSec_Profile は IP トンネルエンティティ CB_Azure_Tunnel にバインドされています。

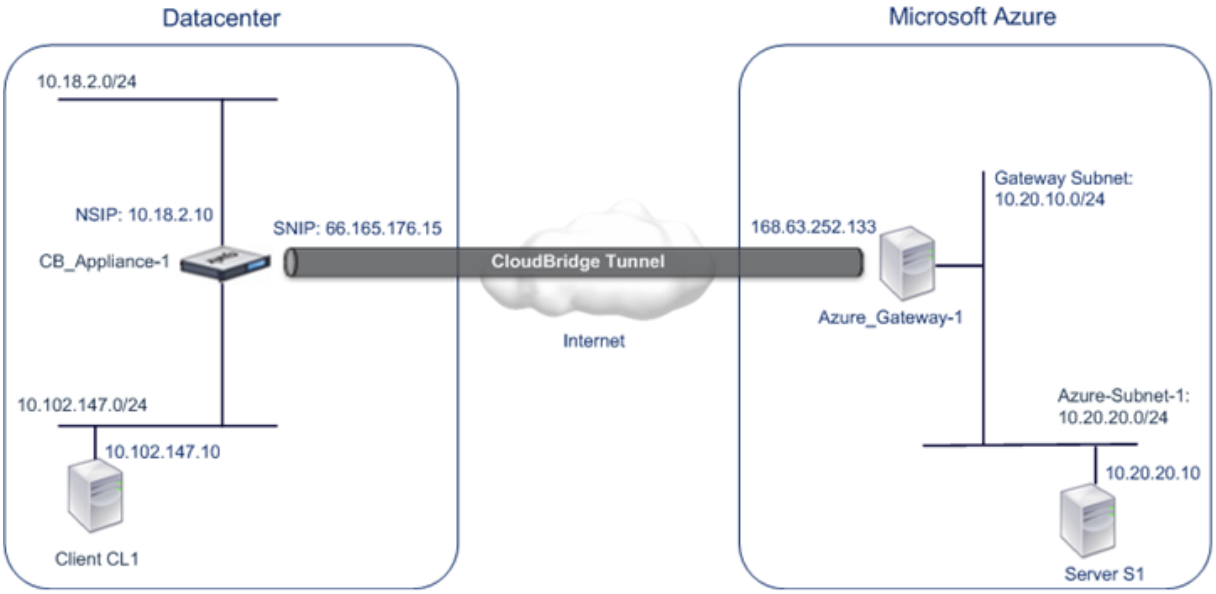
CloudBridge Connector のトンネルエンティティ CB_Azure_Tunnel は、ローカル IP アドレス（NetScaler アプライアンスで構成されたパブリック IP（SNIP）アドレス）、リモート IP アドレス（Azure_Gateway-1 の IP アドレス）、および CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPsec）を指定します。CB_Azure_Tunnel は PBR エンティティ CB_Azure_PBR にバインドされています。

PBR エンティティ CB_Azure_PBR は、条件セットと CloudBridge Connector トンネルエンティティ (CB_Azure_Tunnel) を指定します。送信元 IP アドレス範囲と宛先 IP アドレス範囲が CB_Azure_PBR の条件です。送信元 IP アドレス範囲と宛先 IP アドレス範囲は、それぞれデータセンターのサブネットと Azure クラウドのサブネットとして指定されます。データセンターのサブネット内のクライアントから送信され、Azure クラウド上のサブネット内のサーバー宛てのリクエストパケットはすべて、CB_Azure_PBR の条件と一致します。その後、このパケットは CloudBridge の処理対象と見なされ、PBR エンティティにバインドされた CloudBridge Connector トンネル (CB_Azure_Tunnel) を介して送信されます。

Microsoft Azure では、CloudBridge Connector のトンネル構成には、My-Datacenter-Network という名前のローカルネットワークエンティティ、Azure-Network-for-CloudBridge-Tunnel という名前の仮想ネットワークエンティティ、および Azure_Gateway-1 という名前のゲートウェイが含まれます。

ローカル（Azure のローカル）ネットワークエンティティである My-Datacenter-Network は、データセンター側の NetScaler アプライアンスの IP アドレスと、トラフィックが CloudBridge Connector トンネルを通過するデータセンターのサブネットを指定します。仮想ネットワークエンティティの Azure-Network-for-CloudBridge-Tunnel は、Azure に Azure-Subnet-1 という名前のプライベートサブネットを定義しています。サブネットのトラフィックは CloudBridge Connector トンネルを通過します。サーバー S1 はこのサブネットにプロビジョニングされます。

ローカルネットワークエンティティの My-Datacenter-Network は、仮想ネットワークエンティティ Azure-Network-for-CloudBridge-Tunnel に関連付けられています。この関連付けは、Azure の CloudBridge Connector トンネル設定のリモートネットワークとローカルネットワークの詳細を定義します。ゲートウェイ Azure_Gateway-1 は、このアソシエーションが CloudBridge Connector トンネルの Azure 側の CloudBridge エンドポイントになるために作成されました。



設定の詳細については、CloudBridge Connector トンネル設定 pdf を参照してください。

CloudBridge Connector のトンネル設定について考慮すべきポイント

データセンターの NetScaler アプライアンスと Microsoft Azure の間に CloudBridge Connector トンネルを構成する前に、次の点を考慮してください。

1. CloudBridge Connector トンネルのトンネルエンドポイントアドレスとして使用するには、NetScaler アプライアンスに公開用の IPv4 アドレス（SNIP タイプ）が必要です。また、NetScaler アプライアンスは NAT デバイスの背後に置かないでください。

2. Azure は、CloudBridge Connector トンネルの次の IPsec 設定をサポートしています。そのため、CloudBridge Connector トンネル用の NetScaler を構成する際には、同じ IPsec 設定を指定する必要があります。

    - IKE バージョン = v1
    - 暗号化アルゴリズム = AES
    - ハッシュアルゴリズム = HMAC SHA1

3. 以下を許可するには、データセンターのエッジでファイアウォールを設定する必要があります。

    - ポート 500 の任意の UDP パケット
    - ポート 4500 の任意の UDP パケット
    - 任意の ESP（IP プロトコル番号 50）パケット

4. 新しい SA を確立するために CloudBridge Connector トンネルエンドポイント間で新しい暗号キーを再ネゴシエートする IKE 再キーイングはサポートされていません。セキュリティアソシエーション (SA) の有効期限が切れると、トンネルはダウン状態になります。そのため、SA の有効期間には非常に大きな値を設定する必要があります。
5. Azure でトンネル構成を設定すると、トンネルの Azure 側（ゲートウェイ）のパブリック IP アドレスと PSK が自動的に生成されるため、NetScaler でトンネル構成を指定する前に Microsoft Azure を構成する必要があります。この情報は、NetScaler でトンネル構成を指定するために必要です。

CloudBridge Connector トンネルの設定

データセンターと Azure の間に CloudBridge Connector トンネルをセットアップするには、データセンターに CloudBridge VPX/MPX をインストールし、Microsoft Azure を CloudBridge Connector トンネル用に構成してから、データセンターの NetScaler アプライアンスを CloudBridge Connector トンネル用に構成する必要があります。

データセンターの NetScaler アプライアンスと Microsoft Azure 間の CloudBridge Connector トンネルの構成は、次のタスクで構成されます。

1. データセンターで **NetScaler** アプライアンスをセットアップします。このタスクには、NetScaler 物理アプライアンス（MPX）の展開と構成、またはデータセンターの仮想化プラットフォームでの NetScaler 仮想アプライアンス（VPX）のプロビジョニングと構成が含まれます。
2. **CloudBridge Connector** トンネル用の **Microsoft Azure** の設定このタスクには、Azure にローカルネットワーク、仮想ネットワーク、ゲートウェイエンティティを作成することが含まれます。ローカルネットワークエンティティは、データセンター側の CloudBridge Connector トンネルエンドポイント（NetScaler アプライアンス）の IP アドレスと、トラフィックが CloudBridge Connector トンネルを通過するデータセンターのサブネットを指定します。仮想ネットワークは Azure 上のネットワークを定義します。仮想ネットワークの作成には、トラフィックが CloudBridge Connector トンネルを通過して形成されるサブネットを定義することが含まれます。次に、ローカルネットワークを仮想ネットワークに関連付けます。最後に、CloudBridge Connector トンネルの Azure 側のエンドポイントとなるゲートウェイを作成します。

3. データセンター内の **NetScaler** アプライアンスを **CloudBridge Connector** トンネル用に構成します。このタスクには、データセンターの NetScaler アプライアンスに IPsec プロファイル、IP トンネルエンティティ、および PBR エンティティを作成することが含まれます。IPsec プロファイルエンティティは、CloudBridge Connector トンネルで使用する IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、PSK などの IPsec プロトコルパラメータを指定します。IP トンネルは、CloudBridge Connector トンネルエンドポイント（データセンターの NetScaler アプライアンスと Azure のゲートウェイ）と CloudBridge Connector トンネルで使用されるプロトコルの両方の IP アドレスを指定します。次に、IPsec プロファイルエンティティを IP トンネルエンティティに関連付けます。PBR エンティティは、CloudBridge Connector トンネルを介して相互に通信する 2 つのサブネット (データセンターと Azure クラウド) を指定します。次に、IP トンネルエンティティを PBR エンティティに関連付けます。

CloudBridge Connector トンネル用の **Microsoft Azure** の設定

Microsoft Azure で CloudBridge Connector トンネル構成を作成するには、Microsoft Windows Azure 管理ポータルを使用してください。これは Microsoft Azure 上でリソースを作成および管理するための Web ベースのグラフィカルインターフェイスです。

Azure クラウドで CloudBridge Connector のトンネル設定を開始する前に、次のことを確認してください。

- Microsoft Azure のユーザーアカウントを持っています。
- Microsoft Azure の概念を理解している。
- Microsoft Windows Azure 管理ポータルについてはよくご存知でしょう。

データセンターと Azure クラウドの間の CloudBridge Connector トンネルを設定するには、Microsoft Windows Azure 管理ポータルを使用して Microsoft Azure で次のタスクを実行します。

- ローカルネットワークエンティティを作成します。データセンターのネットワーク詳細を指定するためのローカルネットワークエンティティを Windows Azure に作成します。ローカルネットワークエンティティは、データセンター側の CloudBridge Connector トンネルエンドポイント（NetScaler）の IP アドレスと、トラフィックが CloudBridge Connector トンネルを通過するデータセンターのサブネットを指定します。
- 仮想ネットワークを作成します。Azure 上のネットワークを定義する仮想ネットワークエンティティを作成します。このタスクには、プライベートアドレス空間の定義が含まれます。ここで、アドレス空間に指定された範囲に属するプライベートアドレスとサブネットの範囲を指定します。サブネットのトラフィックは CloudBridge Connector トンネルを通過します。次に、ローカルネットワークエンティティを仮想ネットワークエンティティに関連付けます。この関連付けにより、Azure は仮想ネットワークとデータセンターネットワーク間の CloudBridge Connector トンネルの構成を作成できます。この仮想ネットワーク用の（作成予定の）Azure ゲートウェイは、CloudBridge Connector トンネルの Azure 側にある CloudBridge エンドポイントになります。次に、作成するゲートウェイのプライベートサブネットを定義します。このサブネットは、仮想ネットワークエンティティのアドレス空間で指定された範囲に属します。
- **Windows Azure** でゲートウェイを作成します。CloudBridge Connector トンネルの Azure 側のエンドポイントとなるゲートウェイを作成します。Azure は、パブリック IP アドレスのプールから、作成したゲートウェイに IP アドレスを割り当てます。

- ゲートウェイのパブリック **IP** アドレスと事前共有キーを収集します。Azure 上の CloudBridge Connector トンネル設定の場合、ゲートウェイのパブリック IP アドレスと事前共有キー (PSK) は Azure によって自動的に生成されます。この情報を書き留めておきます。データセンターの NetScaler で CloudBridge Connector トンネルを構成するために必要になります。

> 注:
>
> CloudBridge Connector トンネル用に Microsoft Azure を構成するための手順は、Microsoft Azure のリリースサイクルによって時間が経つにつれ、変更されることがあります。最新の手順については、Microsoft Azure のドキュメントを参照してください。

CloudBridge Connector トンネル用のデータセンターでの **NetScaler ADC** アプライアンスの構成

データセンターと Azure クラウド間の CloudBridge Connector トンネルを構成するには、データセンターの NetScaler で次のタスクを実行します。NetScaler のコマンドラインまたは GUI のどちらかを使用できます。

- **IPsec** プロファイルを作成します。IPSec プロファイルエンティティは、CloudBridge Connector トンネル内の IPsec プロトコルで使用される IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、PSK などの IPsec プロトコルパラメータを指定します。
- **IPsec** プロトコルを使用して **IP** トンネルを作成し、**IPsec** プロファイルをそれに関連付けます。IP トンネルは、ローカル IP アドレス（NetScaler アプライアンスで構成されたパブリック SNIP アドレス）、リモート IP アドレス（Azure のゲートウェイのパブリック IP アドレス）、CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPsec）、および IPsec プロファイルエンティティを指定します。作成された IP トンネルエンティティは、CloudBridge Connector トンネルエンティティとも呼ばれます。
- **PBR** ルールを作成し、**IP** トンネルをそれに関連付けます。PBR エンティティは一連の条件と IP トンネル (CloudBridge Connector トンネル) エンティティを指定します。送信元 IP アドレスの範囲と宛先 IP の範囲は、PBR エンティティの条件です。送信元 IP アドレスの範囲を設定してトラフィックがトンネルを通過するデータセンターのサブネットを指定し、宛先 IP アドレスの範囲を設定してトラフィックが CloudBridge Connector トンネルを通過する Azure サブネットを指定する必要があります。データセンターのサブネット内のクライアントから送信され、Azure クラウド上のサブネット内のサーバー宛てのリクエストパケットはすべて、PBR エンティティの送信元と宛先 IP 範囲と一致します。その後、このパケットは CloudBridge Connector のトンネル処理の対象と見なされ、PBR エンティティに関連付けられた CloudBridge Connector トンネルを介して送信されます。

GUI では、これらすべてのタスクが CloudBridge Connector ウィザードと呼ばれる 1 つのウィザードにまとめられています。
NetScaler コマンドラインを使用して IPSEC プロファイルを作成するには:

コマンドプロンプトで、次のように入力します。

```
add ipsec profile <name> -psk <string> -ikeVersion v1
```

NetScaler コマンドラインを使用して IPSEC トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには:

コマンドプロンプトで、次のように入力します。

```
add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol
IPSEC -ipsecProfileName <string>
```

NetScaler コマンドラインを使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

```
add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range>
ipTunnel <tunnelName> apply pbrs
```

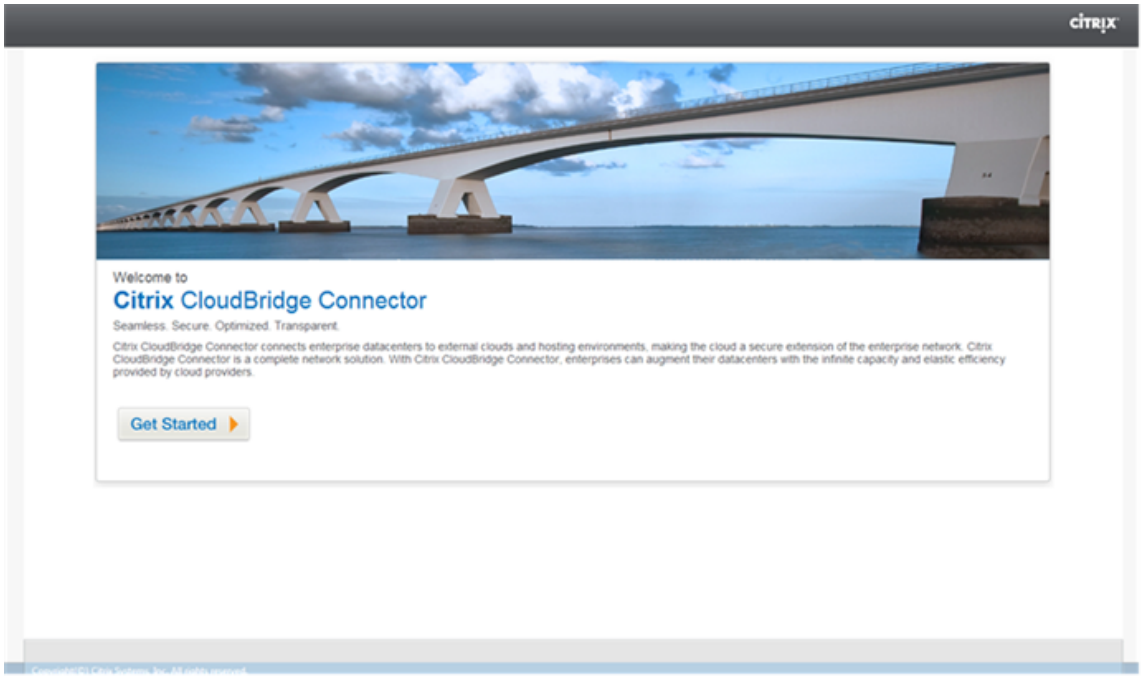
構成例

次のコマンドは、「CloudBridge Connector の構成とデータフローの例」で使用されている NetScaler アプライアンス CB_Appliance-1 のすべての設定を作成します。

```
1  > add ipsec profile CB_Azure_IPSec_Profile -psk
       DkiMgMdcbqvYREEuIvxsbKkW0FOyDiLM  -ikeVersion v1  - lifetime 31536000
2  Done
3
4  >  add iptunnel CB_Azure_Tunnel 168.63.252.133 255.255.255.255
       66.165.176.15  - protocol IPSEC  - ipsecProfileName
       CB_Azure_IPSec_Profile
5  Done
6
7  > add pbr CB_Azure_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP
       10.20.0.0-10.20.255.255  - ipTunnelCB_Azure_Tunnel
8  Done
9
10 > apply pbrs
11 Done
12 <!--NeedCopy-->
```

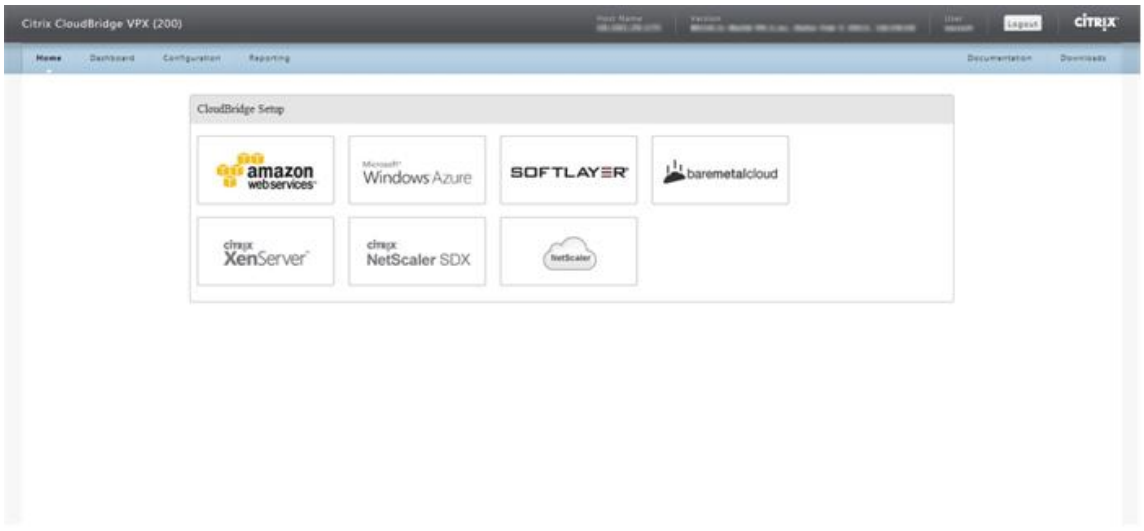
GUI を使用して NetScaler アプライアンスの CloudBridge Connector トンネルを構成するには

1. Web ブラウザーを使用して GUI にアクセスし、データセンターの NetScaler アプライアンスの IP アドレスに接続します。
2. [システム] > [**CloudBridge**Connector] に移動します。
3. 右側のペインの「はじめに」で、「CloudBridge の作成/監視」をクリックします。
4. 「始める」をクリックします。

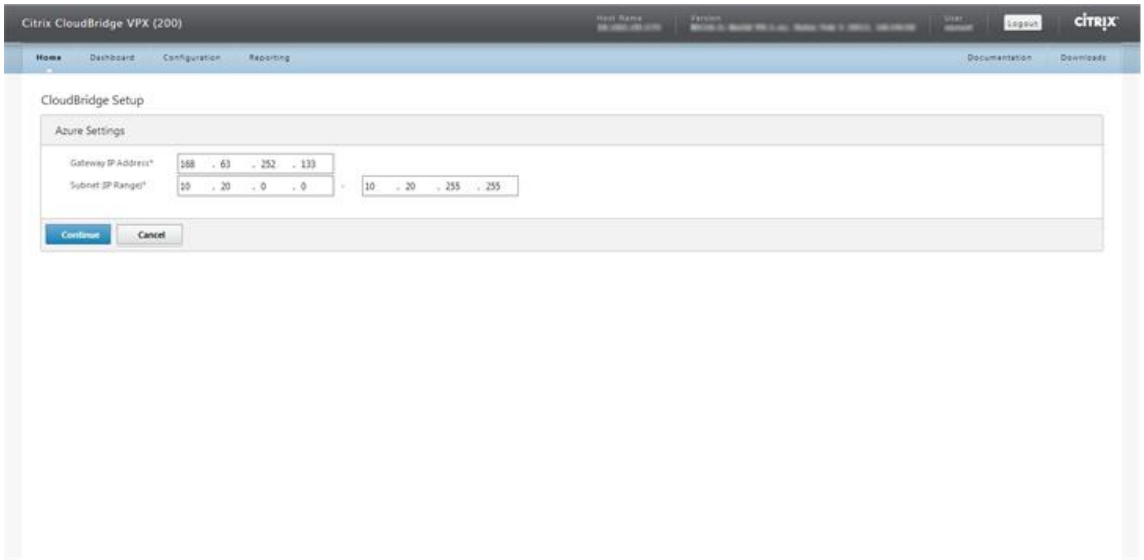


> 注：NetScaler アプライアンスで CloudBridge Connector トンネルがすでに構成されている場合、この画面は表示されず、CloudBridge Connector のセットアップペインが表示されます。

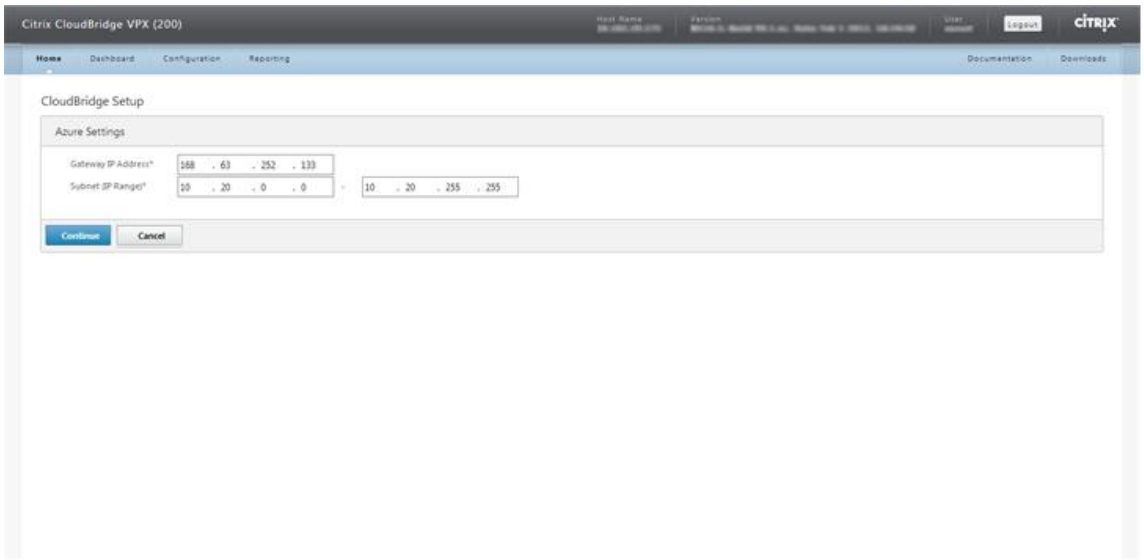
5. CloudBridge セットアップペインで、**Microsoft Windows**Azure をクリックします。



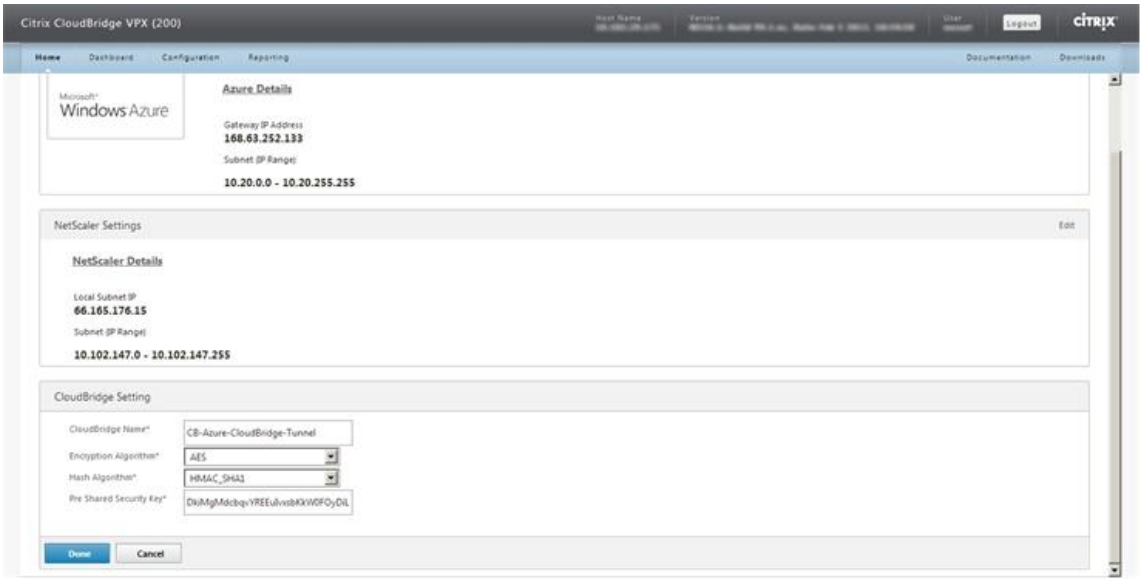
6. Azure 設定ペインの「ゲートウェイ **IP** アドレス」フィールドに、Azure ゲートウェイの IP アドレスを入力します。次に、NetScaler アプライアンスとゲートウェイの間に CloudBridge Connector トンネルが設定されます。サブネット（**IP** 範囲）テキストボックスに、トラフィックが CloudBridge Connector トンネルを通過するサブネット範囲（Azure クラウド内）を指定します。［続行］をクリックします。



7. [NetScaler ADC 設定] ペインの [ローカルサブネット **IP]** ドロップダウンリストから、NetScaler ADC アプライアンスで構成されているパブリックにアクセス可能な SNIP アドレスを選択します。サブネット **(IP** 範囲**)** テキストボックスに、トラフィックが CloudBridge Connector トンネルを通過するローカルサブネット範囲を指定します。［続行］をクリックします。



8.「**CloudBridge** 設定」ペインの「CloudBridge 名」テキストボックスに、作成する CloudBridge の名前を入力します。



9. 「暗号化アルゴリズム」と「ハッシュアルゴリズム」ドロップダウンリストから、それぞれ AES と HMAC_SHA1 アルゴリズムを選択します。事前共有セキュリティキーテキストボックスに、セキュリティキーを入力します。
10. ［完了］をクリックします。

CloudBridge Connector トンネルの監視

データセンターの NetScaler アプライアンスと Microsoft Azure の間の CloudBridge Connector トンネルのパフォーマンスを監視するための統計情報を表示できます。NetScaler アプライアンスの CloudBridge Connector トンネル統計を表示するには、GUI または NetScaler コマンドラインを使用します。Microsoft Azure の CloudBridge Connector トンネルの統計情報を表示するには、Microsoft Windows Azure 管理ポータルを使用してください。

NetScaler ADC アプライアンスでの **CloudBridge Connector** トンネルの統計情報の表示

NetScaler ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、Cloud-Bridge Connector トンネルのモニタリングを参照してください。

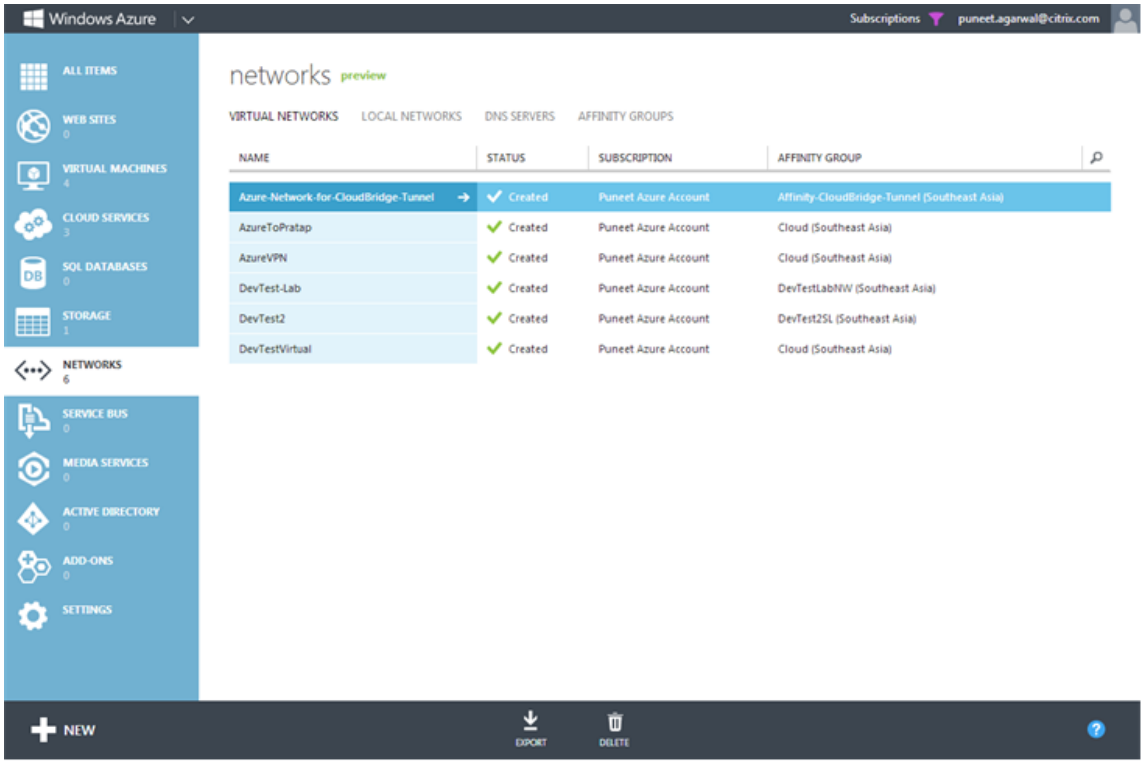
Microsoft の **Azure** での **CloudBridge Connector** のトンネルの統計情報の表示

次の表は、Microsoft Azure の CloudBridge Connector トンネルの監視に使用できる統計カウンタの一覧です。

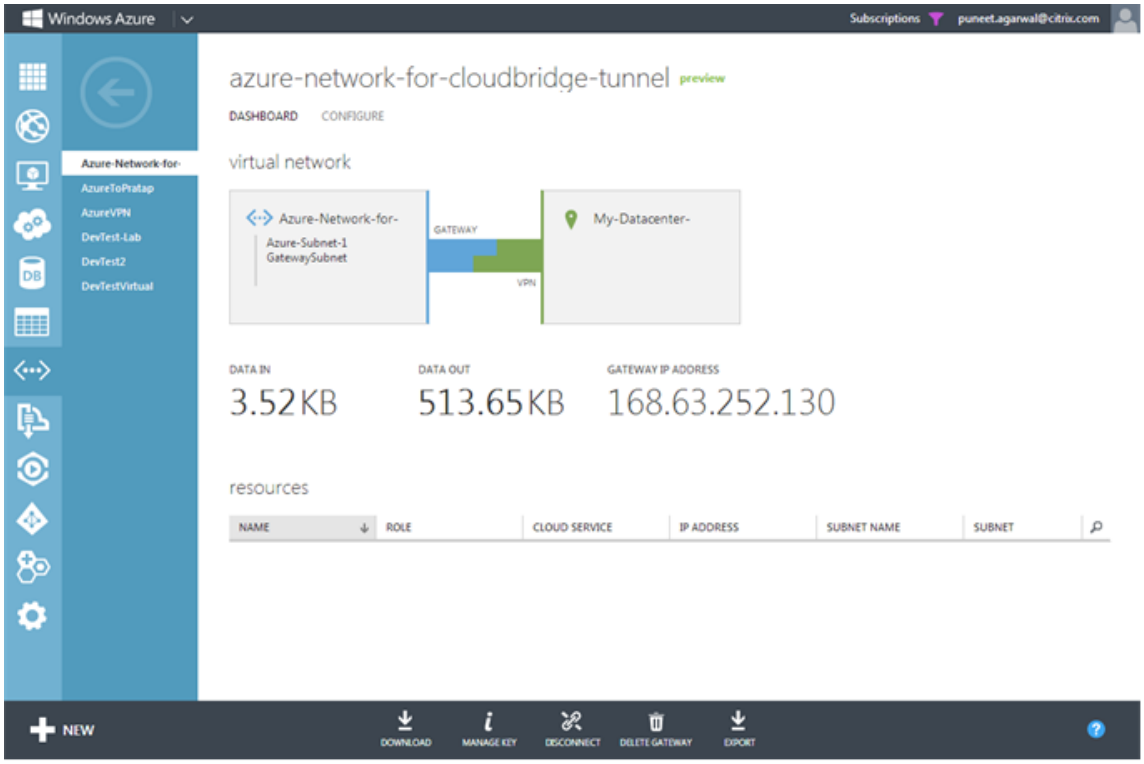
統計カウンタ	Specifies
データイン	ゲートウェイが作成されてから、Azure ゲートウェイが CloudBridge Connector トンネルを介して受信したキロバイトの総数。
データアウト	ゲートウェイが作成されてから、Azure ゲートウェイが CloudBridge Connector トンネルを介して送信したキロバイトの総数。

Microsoft の Windows Azure 管理ポータルを使用して CloudBridge Connector のトンネルの統計情報を表示するには

1. Microsoft Azure アカウントの認証情報を使用して、Windows Azure 管理ポータルにログオンします 。
2. 左側のウィンドウで、[ネットワーク] をクリックします。
3. 「仮想ネットワーク」タブの「名前」列で、統計情報を表示したい CloudBridge Connector トンネルに関連する仮想ネットワークエンティティを選択します。



4. 仮想ネットワークのダッシュボードページで、CloudBridge Connector トンネルのデータ入力カウンターとデータ出力カウンターを表示します。



データセンターとソフトレイヤーエンタープライズクラウド間の **CloudBridge Connector** トンネルの設定

August 15, 2023

この GUI には、データセンターの NetScaler アプライアンスと SoftLayer エンタープライズクラウド上の NetScaler VPX インスタンス間の CloudBridge Connector トンネルを簡単に構成するのに役立つウィザードが含まれています。

データセンターで NetScaler アプライアンスのウィザードを使用すると、NetScaler アプライアンスで作成された CloudBridge Connector トンネル構成は、CloudBridge Connector トンネルの他のエンドポイントまたはピア（SoftLayer 上の NetScaler VPX）に自動的にプッシュされます。

データセンターの NetScaler アプライアンスのウィザードを使用して、次の手順を実行して CloudBridge Connector トンネルを構成します。

1. ユーザーのログオン認証情報を指定して Softlayer エンタープライズクラウドに接続します。
2. NetScaler VPX アプライアンスを実行している Citrix XenServer を選択します。
3. NetScaler VPX アプライアンスを選択します。
4. CloudBridge Connector のトンネルパラメータを以下に提供してください。

    - GRE トンネルを設定します。

- GRE トンネルに IPsec を設定します。
- 名前を指定して、CloudBridge Connector を論理的に表現したネットブリッジを作成します。
- GRE トンネルをネットブリッジにバインドします。

GUI を使用して CloudBridge Connector トンネルを設定するには

1. アプライアンスのアカウント認証情報を使用して、データセンターの NetScaler アプライアンスの GUI にログオンします。
2. **[**** システム] > [**CloudBridge Connector **]** に移動します。
3. 右側のペインの「はじめに」で、「CloudBridge Connector の作成/監視」をクリックします。
4. [開始] をクリックします。

> 注記:
>
> NetScaler アプライアンスで CloudBridge Connector トンネルがすでに構成されている場合、この画面は表示されず、CloudBridge Connector のセットアップペインが表示されます。

1. CloudBridge Connector のセットアップペインで「Softlayer」をクリックし、ウィザードの指示に従います。

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、NetScaler ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。NetScaler アプライアンスでの CloudBridge Connector のトンネル統計の表示について詳しくは、「CloudBridge Connector トンネルの監視」を参照してください。

NetScaler アプライアンスと Cisco IOS デバイス間の CloudBridge Connector トンネルの構成

August 15, 2023

NetScaler アプライアンスと Cisco デバイスの間に CloudBridge Connector トンネルを構成して、2 つのデータセンターを接続したり、ネットワークをクラウドプロバイダーに拡張したりできます。NetScaler アプライアンスと Cisco IOS デバイスは、CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

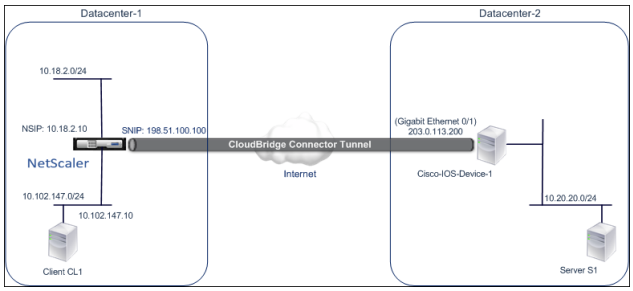
CloudBridge Connector のトンネル構成とデータフローの例

CloudBridge Connector トンネル内のトラフィックフローの図として、次のデバイス間に CloudBridge Connector トンネルが設定されている例を考えてみましょう。

- データセンター 1 として指定されたデータセンターにある NetScaler アプライアンス NS_Appliance-1
- データセンター-2 として指定されたデータセンター内の Cisco IOS デバイス Cisco-IOS-Device-1

NS_Appliance-1 と Cisco-IOS-Device-1 により、CloudBridge Connector トンネルを介したデータセンター 1 とデータセンター 2 のプライベートネットワーク間の通信が可能になります。この例では、NS_Appliance-1 と Cisco-IOS-Device-1 により、CloudBridge Connector トンネルを介したデータセンター 1 のクライアント CL1 とデータセンター 2 のサーバー S1 間の通信が可能になります。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector のトンネル構成には IPsec プロファイルエンティティ NS_Cisco_IPsec_Profile、CloudBridge Connector トンネルエンティティ NS_Cisco_Tunnel、およびポリシーベースルーティング (PBR) エンティティ ns_Cisco_PBR が含まれます。



詳細については、NetScaler ADC アプライアンスと Cisco IOS デバイスの設定間の CloudBridge Connector トンネルを参照してください 。

CloudBridge Connector のトンネル設定について考慮すべきポイント

NetScaler アプライアンスと Cisco IOS デバイス間の CloudBridge Connector トンネルを構成する前に、次の点を考慮してください。

- NetScaler アプライアンスと Cisco IOS デバイス間の CloudBridge Connector トンネルでは、次の IPsec 設定がサポートされています。

IPSec のプロパティ	設定
IPsec モード	トンネルモード
IKE バージョン	バージョン 1
IKE DH グループ	DH グループ 2 (1024 ビット MODP アルゴリズム)

IPSec のプロパティ	設定
IKE 認証方式	事前共有キー
IKE 暗号化アルゴリズム	AES, 3DES
IKE ハッシュアルゴリズム	HMAC SHA1, HMAC SHA256, HMAC SHA384, HMAC SHA512, HMAC MD5
ESP 暗号化アルゴリズム	AES, 3DES
ESP ハッシュアルゴリズム	HMAC SHA1、HMAC SHA256、HMAC SHA256、HMAC SHA256、HMAC MD5

- CloudBridge Connector の両端にある NetScaler アプライアンスと Cisco IOS デバイスでも同じ IPsec 設定を指定する必要があります。
- NetScaler には、IKE ハッシュアルゴリズムと ESP ハッシュアルゴリズムを指定するための共通パラメータ（IPSec プロファイル）が用意されています。また、IKE 暗号化アルゴリズムと ESP 暗号化アルゴリズムを指定するための、もう 1 つの共通パラメータも用意されています。そのため、Cisco デバイスでは、IKE（IKE ポリシーの作成中）と ESP（IPsec トランスフォームセットの作成中）に同じハッシュアルゴリズムと同じ暗号化アルゴリズムを指定する必要があります。
- 次のことを許可するには、NetScaler 側と Cisco デバイス側でファイアウォールを構成する必要があります。
  - ポート 500 の任意の UDP パケット
  - ポート 4500 の任意の UDP パケット
  - 任意の ESP（IP プロトコル番号 50）パケット

CloudBridge Connector トンネル用の **Cisco IOS** デバイスの設定

Cisco IOS デバイスで CloudBridge Connector トンネルを設定するには、Cisco IOS コマンドラインインターフェイスを使用します。これは、Cisco デバイスの設定、モニタリング、および保守のための主要なユーザインターフェイスです。

Cisco IOS デバイスで CloudBridge Connector のトンネル設定を開始する前に、次のことを確認してください。

- Cisco IOS デバイスに管理者認証情報を持つユーザアカウントがあります。
- Cisco IOS コマンドラインインターフェイスに精通している。
- Cisco IOS デバイスは稼働中で、インターネットに接続されています。また、CloudBridge Connector トンネルを介してトラフィックを保護するプライベートサブネットにも接続されています。

> 注記:
>
> Cisco IOS デバイスで CloudBridge Connector トンネルを設定する手順は、Cisco のリリースサイクルによって時間が経つにつれ、変更されることがあります。詳細については、Cisco の公式製品ドキュメントに従うこ

> とをお勧めします。詳細については、「IPsec VPN トンネルの構成 」を参照してください。

NetScaler ADC アプライアンスと **Cisco IOS** デバイスの間に **CloudBridge Connector** トンネルを設定するには、**Cisco** デバイスの **IOS** コマンドラインで次のタスクを実行します。

- IKE ポリシーを作成します。
- IKE 認証用の事前共有キーを設定します。
- トランスフォームセットを定義し、トンネルモードで IPSec を設定します。
- 暗号アクセスリストを作成する
- クリプトマップを作成する
- インターフェイスにクリプトマップを適用する

次の手順の例では、「CloudBridge Connector の設定とデータフローの例」で説明されている`Cisco IOS device Cisco-IOS-Device-1`の設定を作成します。

IKE ポリシーを作成するには、IKE ポリシー pdf を参照してください。

Cisco IOS コマンドラインを使用して事前共有キーを設定するには、次の手順を実行します。

Cisco IOS デバイスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。

コマンド	例	コマンドの説明
crypto isakmp identity address	Cisco-ios-device-1(config)# crypto isakmp identity address	IKE ネゴシエーション中にピア（NetScaler アプライアンス）と通信するときに使用する Cisco IOS デバイスの ISAKMP ID（アドレス）を指定します。この例では、IP アドレス 203.0.113.200（Cisco-IOS-Device-1 のギガビットイーサネットインターフェイス 0/1）をデバイスの ID として使用するアドレスキーワードを指定します。

コマンド	例	コマンドの説明
crypto isakmp key keystringaddress peer-address	Cisco-IOS-Device-1 (config) # クリプトは KMP キーの例事前共有キーアドレス 198.51.100.100	IKE 認証用の事前共有キーを指定します。この例では、NetScaler アプライアンス NS_Appliance-1（198.51.100.100）で使用する共有キーの例/事前共有キーを構成します。Cisco IOS デバイスと NetScaler アプライアンス間で IKE 認証を成功させるには、NetScaler アプライアンスで同じ事前共有キーを設定する必要があります。

Cisco IOS コマンドラインを使用して暗号アクセスリストを作成するには、次の手順を実行します。

Cisco IOS デバイスのコマンドプロンプトで、グローバル構成モードで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
access-listaccess-list-number permit IPsource source-wildcard destination destination-wildcard	Cisco-ios-device-1(config)# access-list 111 permit ip 10.20.20.0 0.0.0.255 10.102.147.0 0.0.0.255	CloudBridge Connector トンネルで IP トラフィックを保護するサブネットを決定する条件を指定します。この例では、サブネット 10.20.0/24（Cisco-IOS-デバイス 1 側）および 10.102.147.0/24（NS_ アプライアンス 1 側）からのトラフィックを保護するように、アクセスリスト 111 を設定します。

Cisco IOS コマンドラインを使用して、トランスフォームを定義して **IPSec** トンネルモードを設定するには、次の手順を実行します。

Cisco IOS デバイスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。
コマンド	例	コマンドの説明
暗号化 ipsec transform-setname esp_authentication_Transform Esp_Authentication_transform 注:Esp_Authentication_Transform は次の値を取ることができます:esp-sha-hmac, esp-sha256-hmac, esp-sha512-hmac, esp-md5-hmac。Esp_encryption_Transform は次の値を取ることができます:esp-aes または esp-3des	Cisco-ios-device-1(config)# crypto ipsec transform-set NS-CISCO-TS esp-sha256-hmac esp-3des	トランスフォームセットを定義し、CloudBridge Connector トンネルピア間のデータ交換時に使用す

る ESP ハッシュアルゴリズム (認証用) と ESP 暗号化アルゴリズムを指定します。次の例では、トランスフォームセット NS-CISCO-TS を定義し、ESP 認証アルゴリズムを esp-sha256-hmac、ESP 暗号化アルゴリズムを esp-3des に指定しています。|
| モードトンネル |Cisco-iOS-device-1（設定暗号トランス）# モードトンネル |IPSec をトンネルモードに設定します。|
|exit|Cisco-iOS-device-1 (設定-crypto-trans) # 終了、Cisco-iOS-device-1 (設定) #| グローバル構成モードに戻ります。|

Cisco IOS コマンドラインを使用してクリプトマップを作成するには、次の手順を実行します。

Cisco IOS デバイスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。

コマンド	例	コマンドの説明
–	-	-
crypto mapmap-name seq-num ipsec-isakmp	Cisco-ios-device-1 (config)# crypto map NS-CISCO-CM 2 ipsec-isakmp	クリプトマップ構成モードを開始し、クリプトマップのシーケンス番号を指定し、IKE を使用してセキュリティアソシエーション（SA）を確立するようにクリプトマップを設定します。この例では、クリプトマップ NS-CISCO-CM のシーケンス番号 2 と IKE を設定しています。
ピア IP アドレスを設定	Cisco-ios-device-1 (config-crypto-map)# set peer 172.23.2.7	ピア（NetScaler アプライアンス）を IP アドレスで指定します。この例では、NetScaler アプライアンスの CloudBridge Connector エンドポイント IP アドレスである 198.51.100.100 を指定しています。
match addressaccess-list-id	Cisco-ios-device-1 (config-crypto-map)# match address 111	拡張アクセスリストを指定してください。このアクセスリストは、CloudBridge Connector トンネルを介して IP トラフィックを保護するサブネットを決定する条件を指定します。この例では、アクセスリスト 111 を指定します。
set transform-set transform-set-name	Cisco-ios-device-1 (config-crypto-map)# set transform-set NS-CISCO-TS	このクリプトマップエントリに使用できるトランスフォームセットを指定します。この例では、トランスフォームセット NS-CISCO-TS を指定しています。
exit	Cisco-ios-device-1 (config-crypto-map)# exit
Cisco-ios-device-1 (config)#|Exit back to global configuration mode.|

Cisco IOS コマンドラインを使用してインターフェイスにクリプトマップを適用するには、次の手順を実行します。

Cisco IOS デバイスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。

コマンド	例	コマンドの説明
interfaceinterface-ID	Cisco-ios-device-1(config)# interface GigabitEthernet 0/1	クリプトマップを適用する物理インターフェイスを指定し、インターフェイス構成モードを開始します。この例では、Cisco デバイス Cisco-IOS-Device-1 のギガビットイーサネットインターフェイス 0/1 を指定しています。IP アドレス 203.0.113.200 は、このインターフェイスに既に設定されています。
crypto mapmap-name	Cisco-ios-device-1 (config-if)# crypto map NS-CISCO-CM	クリプトマップを物理インターフェイスに適用します。この例では、クリプトマップ NS-CISCO-CM を適用しています。
exit	Cisco-ios-device-1 (config-if)# exit, Cisco-ios-device-1 (config)#	グローバル構成モードに戻ります。

CloudBridge Connector トンネル用の **NetScaler ADC** アプライアンスの構成

NetScaler アプライアンスと Cisco IOS デバイス間の CloudBridge Connector トンネルを構成するには、NetScaler アプライアンスで次のタスクを実行します。NetScaler コマンドラインまたは NetScaler グラフィカルユーザーインターフェイス（GUI）のいずれかを使用できます。

- IPsec プロファイルを作成します。
- IPsec プロトコルを使用する IP トンネルを作成し、IPsec プロファイルをそれに関連付けます。
- PBR ルールを作成して IP トンネルに関連付けます。

NetScaler コマンドラインを使用して **IPSEC** プロファイルを作成するには:

コマンドプロンプトで、次のように入力します。

- `add ipsec profile <name> -psk <string> -ikeVersion v1`
- `show ipsec profile <name>`

NetScaler コマンドラインを使用して **IPSEC** トンネルを作成し、**IPSEC** プロファイルをそのトンネルにバインドするには:

コマンドプロンプトで、次のように入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`

- add ipTunnel <name>

NetScaler コマンドラインを使用して **PBR** ルールを作成し、**IPSEC** トンネルをそのルールにバインドするには:

コマンドプロンプトで、次のように入力します。

- add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>
- apply pbrs
- show pbrs <pbrName>

次のコマンドは、「**CloudBridge Connector** の設定とデータフローの例」で説明されている**NetScaler appliance NS_Appliance-1**の設定を作成します。

```
1       >  add ipsec profile NS_Cisco_IPSec_Profile -psk
            examplepresharedkey -ikeVersion v1  - lifetime 315360  - encAlgo 3
            DES
2       Done
3       >  add iptunnel NS_Cisco_Tunnel 203.0.113.200 255.255.255.255
            198.51.100.100  - protocol IPSEC  - ipsecProfileName
            NS_Cisco_IPSec_Profile
4
5       Done
6       > add pbr NS_Cisco_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP
            10.20.0.0-10.20.255.255  - ipTunnel NS_Cisco_Tunnel
7
8       Done
9       > apply pbrs
10
11      Done
12  <!--NeedCopy-->
```

GUI を使用して **IPSEC** プロファイルを作成するには:

1. [システム] > [**CloudBridge Connector**] > [**IPSec** プロファイル] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **IPsec** プロファイルの追加ダイアログボックスで、次のパラメータを設定します。
    - 名前
    - 暗号化アルゴリズム
    - ハッシュアルゴリズム
    - IKE プロトコルバージョン
4. **2** つの **CloudBridge Connector** トンネルピアが相互認証に使用する IPsec 認証方法を設定します。事前共有キー認証方法を選択し、事前共有キーが存在するパラメーターを設定します。
5. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して **IP** トンネルを作成し、**IPSEC** プロファイルをそのトンネルにバインドするには:

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2.「IPv4 トンネル」タブで、「追加」をクリックします。
3. [**IP** トンネルの追加] ダイアログボックスで、次のパラメータを設定します。

    - 名前
    - リモート IP
    - リモートマスク
    - ローカル IP タイプ (「ローカル IP タイプ」ドロップダウンリストで、「サブネット IP」を選択します)。
    - ローカル IP (選択した IP タイプの設定済み IP がすべてローカル IP ドロップダウンリストに表示されます。リストから目的の IP を選択します。)
    - プロトコル
    - IPSec プロファイル

4. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3. **Create PBR** ダイアログボックスで、次のパラメータを設定します。

    - 名前
    - アクション
    - ネクストホップタイプ (IP トンネルの選択)
    - IP トンネル名
    - 送信元 IP アドレスが低い
    - ソース IP ハイ
    - デスティネーション IP フロー
    - デスティネーション IP ハイ

4. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して **PBR** を適用するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. **[**PBR**] タブで **PBR** を選択し、[アクション**]** リストで [適用] を選択します。**

NetScaler アプライアンス上の対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。CloudBridge Connector トンネルの現在のステータスは、設定済みの CloudBridge Connector ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、NetScaler ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。NetScaler アプライアンスでの CloudBridge Connector のトンネル統計の表示について詳しくは、「CloudBridge Connector トンネルの監視」を参照してください。

NetScaler アプライアンスとフォーティネットの FortiGate アプライアンス間の CloudBridge Connector トンネルの設定

August 15, 2023

NetScaler アプライアンスとフォーティネット FortiGate アプライアンスの間に CloudBridge Connector トンネルを構成して、2 つのデータセンターを接続したり、ネットワークをクラウドプロバイダーに拡張したりできます。NetScaler アプライアンスと FortiGate アプライアンスは CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

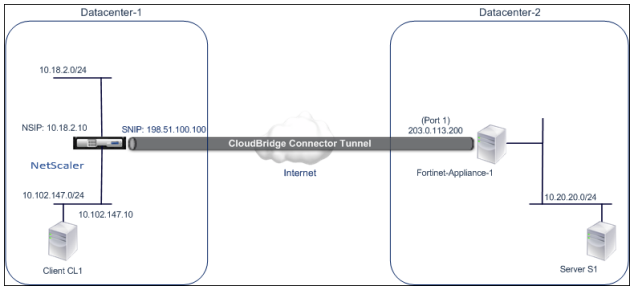
CloudBridge Connector のトンネル設定の例

CloudBridge Connector トンネル内のトラフィックフローの図として、次のデバイス間に CloudBridge Connector トンネルが設定されている例を考えてみましょう。

- データセンター 1 として指定されたデータセンターにある NetScaler アプライアンス NS_Appliance-1
- データセンター 2 として指定されたデータセンターにある FortiGate アプライアンス FortiGate アプライアンス-1

NS_Appliance-1 と FortiGate-Appliance-1 は、CloudBridge Connector トンネルを介してデータセンター 1 とデータセンター 2 のプライベートネットワーク間の通信を可能にします。この例では、NS_Appliance-1 と FortiGate-Appliance-1 により、CloudBridge Connector トンネルを介してデータセンター 1 のクライアント CL1 とデータセンター 2 のサーバー S1 間の通信が可能になります。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector のトンネル構成には、IPSec プロファイルエンティティ NS_Fortinet_IPSec_Profile、CloudBridge Connector トンネルエンティティ NS_Fortinet_Tunnel、およびポリシーベースルーティング（PBR）エンティティ NS_Fortinet_PBR が含まれます。



詳細については、CloudBridge Connector トンネル設定表 pdf を参照してください。

データセンター 2 の Fortinet FortiGate-Appliance-1 の設定については、表を参照してください。

CloudBridge Connector のトンネル設定について考慮すべきポイント

NetScaler ADC アプライアンスと FortiGate アプライアンスの間に CloudBridge Connector トンネルを構成する前に、次の点を考慮してください。

- 以下の IPSec 設定は、NetScaler ADC アプライアンスと FortiGate アプライアンス間の CloudBridge Connector トンネルでサポートされています。

IPSec のプロパティ	設定
IPsec モード	トンネルモード
IKE バージョン	バージョン 1
IKE DH グループ	DH グループ 2 (1024 ビット MODP アルゴリズム)
IKE 認証方式	事前共有キー
IKE 暗号化アルゴリズム	AES
IKE ハッシュアルゴリズム	HMAC SHA1
ESP 暗号化アルゴリズム	AES
ESP ハッシュアルゴリズム	HMAC SHA1

- CloudBridge Connector の両端にある NetScaler アプライアンスと FortiGate アプライアンスで同じ IPsec 設定を指定する必要があります。
- NetScaler には、IKE ハッシュアルゴリズムと ESP ハッシュアルゴリズムを指定するための共通パラメータ（IPSec プロファイル）が用意されています。また、IKE 暗号化アルゴリズムと ESP 暗号化アルゴリズムを指定するためのもう 1 つの共通パラメータも用意されています。そのため、FortiGate アプライアンスでは、IKE（フェーズ 1 構成）と ESP（フェーズ 2 構成）で同じハッシュアルゴリズムと同じ暗号化アルゴリズムを指定する必要があります。

- 次のことを許可するには、NetScaler 側と FortiGate 側でファイアウォールを構成する必要があります。
  - ポート 500 の任意の UDP パケット
  - ポート 4500 の任意の UDP パケット
  - 任意の ESP（IP プロトコル番号 50）パケット
- FortiGate アプライアンスは、ポリシーベースとルートベースの 2 種類の VPN トンネルをサポートしています。FortiGate アプライアンスと NetScaler アプライアンスの間では、ポリシーベースの VPN トンネルのみがサポートされています。

CloudBridge Connector トンネル用の **FortiGate** アプライアンスの設定

FortiGate アプライアンスで CloudBridge Connector トンネルを設定するには、FortiGate アプライアンスの構成、監視、保守を行うための主要なユーザーインターフェイスであるフォーティネットの Web ベースのマネージャーを使用します。

FortiGate アプライアンスで CloudBridge Connector のトンネル構成を開始する前に、次のことを確認してください。

- FortiGate アプライアンスの管理者認証情報を持つユーザーアカウントを持っています。
- フォーティネットの Web ベースマネージャーについてはよくご存知でしょう。
- FortiGate アプライアンスは稼働中で、インターネットに接続されています。また、CloudBridge Connector トンネルを介してトラフィックを保護するプライベートサブネットにも接続されています。

> 注
>
> FortiGate アプライアンスで CloudBridge Connector トンネルを設定する手順は、Fortinet のリリースサイクルによって時間が経つにつれ、変更されることがあります。IPsec VPN トンネルの構成に関する公式の Fortinet 製品マニュアルに従うことをお勧めします。

NetScaler ADC アプライアンスと FortiGate アプライアンス間の CloudBridge Connector トンネルを構成するには、Fortinet ウェブベースのマネージャーを使用して、FortiGate アプライアンスで以下のタスクを実行します。

- ポリシーベースの **IPsec VPN** 機能を有効にします。この機能を有効にすると、FortiGate アプライアンスでポリシーベースの VPN トンネルを作成できます。FortiGate アプライアンスと NetScaler アプライアンスの間では、ポリシーベースの VPN トンネルのみがサポートされています。FortiGate アプライアンスのポリシーベースの VPN トンネル構成には、フェーズ 1 の設定、フェーズ 2 の設定、および IPsec セキュリティポリシーが含まれます。
- フェーズ **1** のパラメータを定義します。フェーズ 1 のパラメーターは、NetScaler アプライアンスへの安全なトンネルを形成する前に、FortiGate アプライアンスによって IKE 認証に使用されます。
- フェーズ **2** のパラメータを定義します。FortiGate アプライアンスはフェーズ 2 のパラメーターを使用して、IKE セキュリティアソシエーション（SA）を確立することで NetScaler アプライアンスへの安全なトンネルを形成します。

- プライベートサブネットを指定します。IP トラフィックがトンネルを介して転送される FortiGate 側と NetScaler 側のプライベートサブネットを定義します。
- トンネルの **IPsec** セキュリティポリシーを定義します。セキュリティポリシーにより、IP トラフィックが FortiGate アプライアンスのインターフェイス間を通過することが許可されます。IPSec セキュリティポリシーは、プライベートサブネットへのインターフェイスと、トンネルを介して NetScaler アプライアンスを接続するインターフェイスを指定します。

フォーティネットの Web ベースマネージャーを使用してポリシーベースの IPsec VPN 機能を有効にするには

1. [システム] > [設定] > [機能] に移動します。
2. [機能設定] ページで [詳細を表示] を選択し、ポリシーベースの **IPsec**VPN をオンにします。

フォーティネットの Web ベースマネージャーを使用してフェーズ 1 のパラメータを定義するには

1. [**VPN**] > [**IPsec**] > [自動キー **(IKE)**] に移動し、[フェーズ **1** の作成] をクリックします。
2.「新規フェーズ **1** 」ページで、次のパラメータを設定します。

    - 名前: このフェーズ 1 構成の名前を入力します。
    - リモートゲートウェイ: 固定 *IP* アドレスを選択します。
    - モード: メイン *(ID* 保護*)* を選択します。
    - 認証方法: 事前共有キーを選択します。
    - 事前共有キー: 事前共有キーを入力します。NetScaler アプライアンスでも同じ事前共有キーを構成する必要があります。
    - ピアオプション: NetScaler アプライアンスを認証するための次の IKE パラメーターを設定します。
        - IKE バージョン: *1* を選択します。
        - モード設定: このオプションが選択されている場合はオフにします。
        - ローカルゲートウェイ IP: メインインターフェイス *IP* を選択します。
        - P1 提案: NetScaler アプライアンスへの安全なトンネルを形成する前に、IKE 認証用の暗号化および認証アルゴリズムを選択してください。
            * 1-暗号化: *AES128* を選択します。
            * 認証: *SHA1* を選択します。
            * キーライフ: フェーズ 1 のキーの有効期間の長さ (秒単位) を入力します。
            * DH グループ: 2 を選択します。
        - *X-Auth:*「無効化」を選択します。
        - Deed Peer Detection: このオプションを選択してください。

3. [**OK**] をクリックします。

フォーティネットの Web ベースマネージャーを使用してプライベートサブネットを指定するには

1. **[**** ファイアウォールオブジェクト] > [アドレス] > [アドレス] に移動し、[新規作成] を選択します。**
2.「新規アドレス」ページで、次のパラメータを設定します。

    - 名前:FortiGate 側のサブネットの名前を入力します。

- タイプ:「サブネット」を選択します。
- サブネット/IP 範囲:FortiGate 側のサブネットのアドレスを入力します。
- インターフェイス: このサブネットへのローカルインターフェイスを選択します。

3. [**OK**] をクリックします。
4. 手順 1〜3 を繰り返して、NetScaler 側のサブネットを指定します。

フォーティネットの Web ベースマネージャーを使用してフェーズ 2 のパラメータを定義するには

1. [**VPN**] > [**IPsec**] > [自動キー **(IKE)**] に移動し、[フェーズ **2** の作成] をクリックします。
2. 「新規フェーズ **2** 」ページで、次のパラメータを設定します。

- 名前: このフェーズ 2 構成の名前を入力します。
- フェーズ 1: ドロップダウンリストからフェーズ 1 の構成を選択します。

3. 「詳細設定」をクリックし、次のパラメータを設定します。

- P2 提案: NetScaler アプライアンスへの安全なトンネルを形成するための暗号化および認証アルゴリズムを選択してください。
  - 1-暗号化: *AES128* を選択します。
  - 認証: *SHA1* を選択します。
  - リプレイ検出を有効にする: このオプションを選択します。
  - Perfect Forward Secrecy (PFS) を有効にする: このオプションを選択します。
  - DH グループ: *2* を選択します。
- キーライフ: フェーズ 2 のキーの有効期間の長さ (秒単位) を入力します。
- Autokey Keep Alive: Select this option.
- オートネゴシエーション: このオプションを選択してください。
- クイックモードセレクター: トラフィックがトンネルを通過する FortiGate 側と NetScaler 側のプライベートサブネットを指定します。
  - ソースアドレス: ドロップダウンリストから FortiGate 側のサブネットを選択します。
  - 送信元ポート: *0* を入力します。
  - 宛先アドレス: ドロップダウンリストから NetScaler 側のサブネットを選択します。
  - 宛先ポート: *0* を入力します。
  - プロトコル: *0* を入力します。

4. [**OK**] をクリックします。

フォーティネットの Web ベースマネージャーを使用して IPsec セキュリティポリシーを定義するには

1. **[**ポリシー**] > [**** ポリシー] > [ポリシー] に移動し、[新規作成] をクリックします。**
2. 「ポリシーの編集」ページで、次のパラメータを設定します。

- ポリシータイプ: *VPN* を選択します。
- ポリシーサブタイプ: *IPsec* を選択します。

- ローカルインターフェース: 内部 (プライベート) ネットワークへのローカルインターフェースを選択します。
- ローカル保護サブネット: トラフィックがトンネルを通過する FortiGate 側のサブネットをドロップダウンリストから選択します。
- 発信 VPN インターフェイス: 外部 (パブリック) ネットワークへのローカルインターフェイスを選択します。
- リモート保護サブネット： トラフィックがトンネルを通過する NetScaler 側のサブネットをドロップダウンリストから選択します。
- スケジュール: 特定の要件を満たすために変更が必要でない限り、デフォルト設定を (常に) そのまま使用してください。
- サービス: 特定の要件を満たすために変更が必要でない限り、デフォルト設定 *(ANY)* のままにしてください。
- VPN トンネル:[既存を使用] を選択し、ドロップダウンリストからトンネルを選択します。
- リモートサイトからのトラフィックの開始を許可: リモートネットワークからのトラフィックによるトンネルの開始を許可するかどうかを選択します。

3. [**OK**] をクリックします。

CloudBridge Connector トンネル用の **NetScaler ADC** アプライアンスの構成

NetScaler アプライアンスと FortiGate アプライアンス間の CloudBridge Connector トンネルを構成するには、NetScaler アプライアンスで次のタスクを実行します。NetScaler コマンドラインまたは NetScaler グラフィカルユーザーインターフェイス（GUI）のいずれかを使用できます。

- **IPsec** プロファイルを作成します。IPSec プロファイルエンティティは、IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、CloudBridge Connector トンネル内の IPsec プロトコルで使用される認証方法などの IPsec プロトコルパラメータを指定します。
- **IPsec** プロトコルを使用する **IP** トンネルを作成し、**IPsec** プロファイルをそれに関連付けます。IP トンネルは、ローカル IP アドレス（NetScaler アプライアンス上で構成された CloudBridge Connector のトンネルエンドポイント IP アドレス（SNIP タイプ)）、リモート IP アドレス（FortiGate アプライアンス上で構成された CloudBridge Connector のトンネルエンドポイント IP アドレス)、CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPsec)、および IPsec プロファイルエンティティを指定します。作成された IP トンネルエンティティは、CloudBridge Connector トンネルエンティティとも呼ばれます。
- **PBR** ルールを作成して **IP** トンネルに関連付けます。PBR エンティティは、ルールセットと IP トンネル (CloudBridge Connector トンネル) エンティティを指定します。送信元 IP アドレス範囲と宛先 IP アドレス範囲は PBR エンティティの条件です。送信元 IP アドレス範囲を設定してトラフィックをトンネルで保護する NetScaler 側のサブネットを指定し、宛先 IP アドレス範囲を設定してトラフィックをトンネルで保護する FortiGate アプライアンス側のサブネットを指定します。

NetScaler コマンドラインを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで入力します。

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecy ENABLE`
- `show ipsec profile <name>`

NetScaler コマンドラインを使用して IPSEC トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName** <string>`
- `show ipTunnel <name>`

NetScaler コマンドラインを使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

コマンドプロンプトで入力します。

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

GUI を使用して IPSEC プロファイルを作成するには

1. [システム] > [**CloudBridge Connector**] > [**IPSec** プロファイル] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **IPsec** プロファイルの追加ページで、次のパラメータを設定します。
    - 名前
    - 暗号化アルゴリズム
    - ハッシュアルゴリズム
    - IKE プロトコルバージョン
    - Perfect Forward Secrecy (このパラメータを有効にする)
4. 2 つの CloudBridge Connector トンネルピアが相互認証に使用する IPsec 認証方法を設定します。事前共有キー認証方法を選択し、事前共有キーが存在するパラメータを設定します。
5. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2.「IPv4 トンネル」タブで、「追加」をクリックします。
3. **IP** トンネルの追加ページで、次のパラメータを設定します。
    - 名前
    - リモート IP

- リモートマスク
- ローカル IP タイプ (「ローカル IP タイプ」ドロップダウンリストで、「サブネット *IP*」を選択します)。
- ローカル IP（選択した IP タイプの設定済み IP アドレスがすべてローカル IP ドロップダウンリストに表示されます。リストから目的の IP を選択します。)
- プロトコル
- IPSec プロファイル

4. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3.「**PBR** の作成」ページで、次のパラメータを設定します。

- 名前
- アクション
- ネクストホップタイプ (*IP* トンネルの選択)
- IP トンネル名
- 送信元 IP アドレスが低い
- ソース IP ハイ
- デスティネーション IP フロー
- デスティネーション IP ハイ

4. [作成] をクリックし、[閉じる] をクリックします。

NetScaler アプライアンス上の対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。

CloudBridge Connector トンネルの現在のステータスは、設定済みの CloudBridge Connector ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

次のコマンドは、「CloudBridge Connector 構成の例」の NetScaler アプライアンス NS_Appliance-1 の設定を作成します。

```
1      > add ipsec profile NS_Fortinet_IPSec_Profile -psk
          examplepresharedkey -ikeVersion v1 -encAlgo AES -hashalgo
          HMAC_SHA1 -lifetime 315360 -perfectForwardSecrecy ENABLE
2
3       Done
4      > add iptunnel NS_Fortinet_Tunnel 203.0.113.200 255.255.255.255
          198.51.100.100 -protocol IPSEC -ipsecProfileName
          NS_Fortinet_IPSec_Profile
5
6       Done
7      > add pbr NS_Fortinet_Pbr -srcIP 10.102.147.0-10.102.147.255 -
          destIP 10.20.0.0-10.20.255.255 -ipTunnel NS_Fortinet_Tunnel
8
```



```
 9       Done
10     > apply pbrs
11
12       Done
13 <!--NeedCopy-->
```

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、NetScaler ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。NetScaler アプライアンスでの CloudBridge Connector のトンネル統計の表示について詳しくは、「CloudBridge Connector トンネルの監視」を参照してください。

CloudBridge Connector トンネルの診断とトラブルシューティング

August 15, 2023

CloudBridge Connector のトンネル構成に問題がある場合は、トンネルを設定する前にすべての前提条件が満たされていることを確認してください。もしそうなら、問題はトンネルのエンドポイントの IP アドレス、NAT 構成、トンネルの設定方法、またはデータトラフィックにある可能性があります。

CloudBridge Connector トンネルのトラブルシューティング

CloudBridge Connector トンネルが正しく機能しない場合は、トンネルの確立またはデータトラフィックに問題がある可能性があります。発生している問題の種類が不明な場合は、ログファイルでエラーメッセージを探し、そのエラーメッセージがトンネル確立の問題のリストに含まれているかどうかを確認してください。エラーメッセージが見つからない場合は、データトラフィックに関連して発生する可能性のある問題の一覧を確認してください。

トンネル設置に関する問題

IPsec トンネルを構成するための要件を満たし、CloudBridge Connector トンネルを構成した後、トンネルのステータスが「UP」でない場合は、トンネルエンドポイントとして構成されている一方または両方の NetScaler アプライアンスの iked.log ファイルでデバッグ情報を探します。

いずれかのアプライアンスで、NetScaler ADC シェルプロンプトで次のコマンドを入力します。

```
cat /tmp/iked.debug | tee /var/iked.log
```

トラブルシューティング pdf には、一般的なエラーとその解決策が記載されています。

データトラフィックに関連する問題

CloudBridge Connector トンネル内のデータがトンネルのエンドポイント間で適切に交換されない場合は、以下を実行してください。

- GRE および IPsec プロトコルを使用する CloudBridge Connector トンネルの場合:
  - 両方の CloudBridge Connector トンネルエンドポイントで L2 モードが有効になっていることを確認します。L2 モードを有効にするには、NetScaler コマンドラインインターフェイスで次のコマンドを入力します。

    ```
    enable mode L2
    ```

    * CloudBridge Connector のトンネルエンドポイントの 1 つが CloudBridge 仮想アプライアンス（VPX）で、VMware ESXi ハイパーバイザー上にプロビジョニングされている場合は、CloudBridge VPX アプライアンスに関連付けられた vSwitch の無差別モードが Accept に設定されていることを確認してください。
  - VLAN が CloudBridge Connector トンネルを介して拡張されている場合は、各トンネルエンドポイントの拡張 VLAN エンティティで 1 対 1 のマッピングを確認してください
  - IP トンネルエンティティが各トンネルエンドポイントの正しい netbridge エンティティにバインドされていることを確認します。
  - NetScaler コマンドラインインターフェイスで次のコマンドを入力して、ピア CloudBridge Connector トンネルエンドポイントの ARP エントリがローカルトンネルエンドポイントに存在することを確認します。

    ```
    show arp
    ```
  - 出力に不完全な ARP エントリが表示されている場合、双方向トラフィックはトンネルを通過していません。双方向トラフィックが流れている場合、ARP エントリにはトンネルの反対側にあるデバイスのトンネルインターフェイスの名前が表示されます。
  - 両方のトンネルエンドポイントから IP トンネルエンティティを削除し、同じパラメータで IPsec プロファイルを NONE に設定して、トンネルが GRE プロトコルのみを使用するように再度追加します。

    IP トンネル（GRE プロトコルを使用する）で次のことを確認したら、各トンネルエンドポイントの各 IP トンネルエンティティに有効な IPsec プロファイルを指定して、IPsec パラメータを使用してトンネルを設定します。

    トンネルを通る適切な PING または TCP フロー。

    トンネルを通るデータトラフィックの適切なフロー。

    設定したトンネル（GRE および IPsec プロトコルを使用する）が UP 状態になった後、データトラフィックがトンネルを正常に通過せず、NAT デバイスがトンネルエンドポイントのどちらかまたは両方の前に配置されている場合は、NAT デバイス上の入力パケットと出力パケットを分析します。
- NetScaler アプライアンスがルーターまたはゲートウェイとして使用されている場合。
  - NetScaler アプライアンスで L3 モードが有効になっていることを確認します。L3 モードを有効にするには、CloudBridge コマンドラインで次のコマンドを実行します。

  - 有効モード L3
  - サブネットがネットブリッジエンティティにバインドされている場合は、正しい IP トンネルエンティティもネットブリッジにバインドされていることを確認してください。
  - NetScaler コマンドラインで次のコマンドを実行して、パケット（入力と出力）がドロップされる場所を確認します。
    ```
    stat ipsec counters
    ```
  - 両方のトンネルエンドポイントに正しいルートが設定されていることを確認してください。
  - NetScaler アプライアンスの前に NAT デバイスが展開されていない場合は、ポート 4500 のすべての ESP（IP プロトコル番号 50）パケットと UDP パケットを許可するようにファイアウォールが構成されていることを確認してください。

上記のいずれの方法でもトンネルエンドポイント間のトラフィック交換が成功しない場合は、Citrix のテクニカルサポートに連絡してください。

Citrix のテクニカルサポートに連絡する前のチェックリスト

迅速に解決するには、Citrix のテクニカルサポートに連絡する前に、次のアイテムを用意しておいてください。

- 展開とネットワークトポロジの詳細。
- NetScaler シェルプロンプトで次のコマンドを入力して収集されたログファイル。
  ```
  cat /tmp/iked.debug | tee /var/log/iked.log
  ```
- NetScaler コマンドラインで次のコマンドを入力してキャプチャしたテクニカルサポートバンドル。
  ```
  show techsupport
  ```
- 両方の CloudBridge Connector トンネルエンドポイントでキャプチャされたパケットトレース。パケットトレースを開始するには、NetScaler コマンドラインで次のコマンドを入力します。
  ```
  start nstrace -size 0
  ```
  パケットトレースを停止するには、NetScaler コマンドラインで次のコマンドを入力します。
  ```
  stop nstrace
  ```
- NetScaler のコマンドプロンプトで入力した次のコマンドの出力。
  ```
  show arp
  ```

CloudBridge Connector の相互運用性—StrongSwan

August 15, 2023

StrongSwan は Linux プラットフォーム用のオープンソースの IPsec 実装です。NetScaler アプライアンスと StrongSwan アプライアンスの間に CloudBridge Connector トンネルを構成して、2 つのデータセンターを接続したり、ネットワークをクラウドプロバイダーに拡張したりできます。NetScaler アプライアンスと StrongSwan アプライアンスは CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

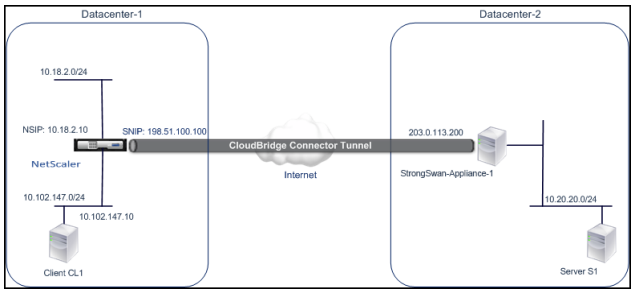
CloudBridge Connector のトンネル設定の例

CloudBridge Connector トンネル内のトラフィックフローの図として、次のデバイス間に CloudBridge Connector トンネルが設定されている例を考えてみましょう。

- データセンター 1 として指定されたデータセンターにある NetScaler アプライアンス NS_Appliance-1
- データセンター 2 として指定されたデータセンター内の StrongSwan アプライアンス StrongSwan アプライアンス-1

NS_Appliance-1 と StrongSwan-Appliance-1 は、CloudBridge Connector トンネルを介してデータセンター 1 とデータセンター 2 のプライベートネットワーク間の通信を可能にします。この例では、NS_Appliance-1 と StrongSwan-Appliance-1 により、CloudBridge Connector トンネルを介してデータセンター 1 のクライアント CL1 とデータセンター 2 のサーバー S1 間の通信が可能になります。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector のトンネル構成には IPsec プロファイルエンティティ NS_strongSwan_IPsec_Profile、CloudBridge Connector トンネルエンティティ NS_StrongSwan_Tunnel、およびポリシーベースルーティング (PBR) エンティティ NS_strongSwan_PBR が含まれます。



次の表に、この例で使用されている設定の一覧を示します。

CloudBridge Connector のトンネル設定の主な設定

エンティティ	詳細
データセンター 1 の CloudBridge Connector トンネルエンドポイント (NS_Appliance-1) の IP アドレス	198.51.100.100
データセンター 2 の CloudBridge Connector トンネルエンドポイント (StrongSwan-Appliance-1) の IP アドレス	203.0.113.200

エンティティ	詳細
データセンター—CloudBridge Connector トンネルを介してトラフィックを保護する 1 のサブネット	10.102.147.0/24
データセンター—CloudBridge Connector トンネルを介してトラフィックを保護する 2 のサブネット	10.20.20.0/24

データセンター 1 の NetScaler アプライアンス NS_ アプライアンス 1 の設定

SNIP1 (参照のみを目的としています)	198.51.100.100
IPSec profile	NS_StrongSwan_IPSec_Profile	IKE version: v1, Encryption algorithm: AES, Hash algorithm: HMAC_SHA1
psk = examplepresharedkey (Note: This is an example of a pre-share key, for illustration. NetScaler では、この文字列を CloudBridge Connector 構成に使用することはお勧めしません)
CloudBridge Connector トンネル |NS_strongswan_Tunnel| リモート IP = 203.0.113.200、ローカル IP= 198.51.100.100、トンネルプロトコル = IPSEC、IPsec プロファイル = NS_strongSwan_IPsec_profile|	ポリシーベースのルート |NS_strongSwan_PBRR	送信元 IP 範囲 = データセンターのサブネット -1=10.102.147.255、宛先 IP 範囲 = データセンターのサブネット -2=10.20.20.0-10.20.255、IP トンネル = NS_strongswan_Tunnel

CloudBridge Connector のトンネル設定について考慮すべきポイント

CloudBridge Connector トンネルの設定を開始する前に、次のことを確認してください。

- Linux の構成に関する基本的な知識があります。
- IPsec プロトコルスイートに関する基本的な知識があります。
- StrongSwan アプライアンスは稼働中で、インターネットに接続されています。また、CloudBridge Connector トンネルを介してトラフィックを保護するプライベートサブネットにも接続されています。
- NetScaler アプライアンスは稼働中で、インターネットに接続されています。また、CloudBridge Connector トンネルを介してトラフィックを保護するプライベートサブネットにも接続されています。
- NetScaler アプライアンスと StrongSwan アプライアンスの間の CloudBridge Connector トンネルでは、次の IPsec 設定がサポートされています。
  - IPsec モード: トンネルモード
  - IKE バージョン: バージョン 1
  - IKE 認証方法: 事前共有キー
  - IKE 暗号化アルゴリズム:AES
  - IKE ハッシュアルゴリズム:HMAC SHA1
  - ESP 暗号化アルゴリズム:AES
  - ESP ハッシュアルゴリズム:HMAC SHA1

- CloudBridge Connector トンネルの両端にある NetScaler アプライアンスと StrongSwan アプライアンスで同じ IPsec 設定を指定する必要があります。
- NetScaler には、IKE ハッシュアルゴリズムと ESP ハッシュアルゴリズムを指定するための共通パラメータ（IPSec プロファイル）が用意されています。また、IKE 暗号化アルゴリズムと ESP 暗号化アルゴリズムを指定するためのもう 1 つの共通パラメータも用意されています。そのため、StrongSwan アプライアンスでは、IPSec.conf ファイルの IKE パラメータと ESP パラメータに同じハッシュアルゴリズムと同じ暗号化アルゴリズムを指定する必要があります。
- 次のことを許可するには、NetScaler 側と StrongSwan 側でファイアウォールを構成する必要があります。
  - ポート 500 の任意の UDP パケット
  - ポート 4500 の任意の UDP パケット
  - 任意の ESP（IP プロトコル番号 50）パケット

CloudBridge Connector トンネル用の **StrongSwan** の設定

NetScaler アプライアンスと StrongSwan アプライアンスの間に CloudBridge Connector トンネルを構成するには、StrongSwan アプライアンスで次のタスクを実行します。

- **ipsec.conf** ファイルに **IPsec.conf** 接続情報を指定します。**ipsec.conf** ファイルには、**StrongSwan** アプライアンス内の **IPsec** 接続のすべての制御および構成情報が定義されます。
- **ipsec.secrets** ファイルに事前共有キーを指定します。**ipsec.secrets** ファイルには、**StrongSwan** アプライアンスでの **IPsec** 接続の **IKE/IPsec** 認証のシークレットが定義されています。

StrongSwan アプライアンスで IPsec VPN（CloudBridge Connector トンネル）を設定する手順は、StrongSwan のリリースサイクルによって時間とともに変化する可能性があります。IPsec VPN トンネルの構成に関する公式の StrongSwan ドキュメントに従うことをお勧めします。

ipsec.conf ファイルのサンプル抜粋に続いて、IPsec VPN トンネルを設定するための IPsec 情報を指定します。詳細については、「CloudBridge Connector 設定の例」トピックを参照してください。詳細については、「CloudBridge Connector の設定 」を参照してください。

ipsec.secrets ファイルのサンプル抜粋に続いて、IPsec VPN トンネルを設定するための IKE 認証事前共有キーを指定します（「CloudBridge Connector 設定の例」トピックを参照）。

`/etc/ipsec.secrets` PSK ‘examplepresharedkey’ #pre-shared key for IPsec IKE authentication

CloudBridge Connector トンネル用の **NetScaler ADC** アプライアンスの構成

NetScaler アプライアンスと StrongSwan アプライアンスの間に CloudBridge Connector トンネルを構成するには、NetScaler アプライアンスで次のタスクを実行します。NetScaler コマンドラインまたは NetScaler グラフィカルユーザーインターフェイス（GUI）のいずれかを使用できます。

- **IPsec** プロファイルを作成します。IPSec プロファイルエンティティは、IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、CloudBridge Connector トンネル内の IPsec プロトコルで使用される認証方法などの IPsec プロトコルパラメータを指定します。
- **IPsec** プロトコルを使用する **IP** トンネルを作成し、**IPsec** プロファイルをそれに関連付けます。IP トンネルは、ローカル IP アドレス（NetScaler アプライアンス上で構成された CloudBridge Connector のトンネルエンドポイント IP アドレス（SNIP タイプ））、リモート IP アドレス（StrongSwan アプライアンス上で構成された CloudBridge Connector のトンネルエンドポイント IP アドレス）、CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPsec）、および IPsec プロファイルエンティティを指定します。作成された IP トンネルエンティティは、CloudBridge Connector トンネルエンティティとも呼ばれます。
- **PBR** ルールを作成して **IP** トンネルに関連付けます。PBR エンティティは、ルールセットと IP トンネル (CloudBridge Connector トンネル) エンティティを指定します。送信元 IP アドレス範囲と宛先 IP アドレス範囲は PBR エンティティの条件です。送信元 IP アドレス範囲を設定してトラフィックをトンネルで保護する NetScaler 側のサブネットを指定し、宛先 IP アドレス範囲を設定してトラフィックをトンネルで保護する StrongSwan 側のサブネットを指定します。

NetScaler コマンドラインを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで入力します。

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1`
- `show ipsec profile <name>`

NetScaler コマンドラインを使用して IPSEC トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

NetScaler コマンドラインを使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

コマンドプロンプトで入力します。

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

GUI を使用して IPSEC プロファイルを作成するには

1. **[**** システム] > [**CloudBridge Connector ******] > [**IPsec** プロファイル] に移動します。**

2. 詳細ペインで、[追加] をクリックします。
3. **IPsec** プロファイルの追加ページで、次のパラメータを設定します。

    - 名前
    - 暗号化アルゴリズム
    - ハッシュアルゴリズム
    - IKE プロトコルバージョン

4. 2 つの CloudBridge Connector トンネルピアが相互認証に使用する IPSec 認証方法を設定します。事前共有キー認証方法 を選択し、事前共有キーが存在 パラメータを設定します。
5. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2.「IPv4 トンネル」タブで、「追加」をクリックします。
3. IP トンネルの追加ページで、次のパラメータを設定します。

    - 名前
    - リモート IP
    - リモートマスク
    - ローカル IP タイプ (「ローカル IP タイプ」ドロップダウンリストで、「サブネット *IP*」を選択します)。
    - ローカル IP（選択した IP タイプの設定済み IP アドレスがすべてローカル IP ドロップダウンリストに表示されます。リストから目的の IP を選択します。)
    - プロトコル
    - IPSec プロファイル

4. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3.「**PBR** の作成」ページで、次のパラメータを設定します。

    - 名前
    - アクション
    - ネクストホップタイプ (*IP* トンネルの選択)
    - IP トンネル名
    - 送信元 IP アドレスが低い
    - ソース IP ハイ
    - デスティネーション IP フロー
    - デスティネーション IP ハイ

4. [作成] をクリックし、[閉じる] をクリックします。

NetScaler アプライアンス上の対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。CloudBridge Connector トンネルの現在のステータスは、設定済みの CloudBridge Connector ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。
次のコマンドは、「CloudBridge Connector 構成の例」の NetScaler アプライアンス NS_Appliance-1 の設定を作成します。

```
    > add ipsec profile NS_StrongSwan_IPSec_Profile -psk
      examplepresharedkey -ikeVersion v1  - encAlgo AES  - hashalgo
      HMAC_SHA1


    Done

    > add iptunnel NS_StrongSwan_Tunnel 203.0.113.200 255.255.255.255
      198.51.100.100  - protocol IPSEC  - ipsecProfileName
      NS_StrongSwan_IPSec_Profile


    Done

    > add pbr NS_StrongSwan_Pbr -srcIP 10.102.147.0-10.102.147.255  -
      destIP 10.20.0.0-10.20.255.255  - ipTunnel NS_StrongSwan_Tunnel


    Done

    > apply pbrs


    Done
<!--NeedCopy-->
```

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、NetScaler ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。NetScaler アプライアンスでの CloudBridge Connector のトンネル統計の表示について詳しくは、「CloudBridge Connector トンネルの監視」を参照してください。

CloudBridge Connector の相互運用性—F5 BIG-IP

August 15, 2023

NetScaler アプライアンスと F5 BIG-IP アプライアンスの間に CloudBridge Connector トンネルを構成して、2つのデータセンターを接続したり、ネットワークをクラウドプロバイダーに拡張したりできます。NetScaler アプラ

イアンスと F5 BIG-IP アプライアンスは CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

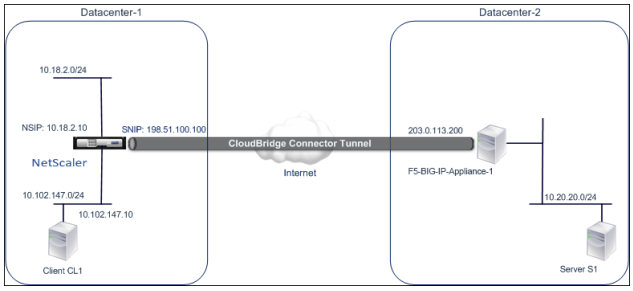
CloudBridge Connector のトンネル設定の例

CloudBridge Connector トンネル内のトラフィックフローの図として、次のデバイス間に CloudBridge Connector トンネルが設定されている例を考えてみましょう。

- データセンター 1 として指定されたデータセンターにある NetScaler アプライアンス NS_Appliance-1
- データセンター 2 として指定されたデータセンター内の F5 BIG-IP アプライアンス F5-BIG-IP アプライアンス-1

NS_Appliance-1 と F5-BIG-IP-Appliance-1 は、CloudBridge Connector トンネルを介してデータセンター 1 とデータセンター 2 のプライベートネットワーク間の通信を可能にします。この例では、NS_Appliance-1 と F5-BIG-IP-Appliance-1 により、CloudBridge Connector トンネルを介してデータセンター 1 のクライアント CL1 とデータセンター 2 のサーバー S1 間の通信が可能になります。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector のトンネル設定には IPsec プロファイルエンティティ NS_F5-BIG-IP_IPSec_Profile、CloudBridge Connector トンネルエンティティ NS_F5-BIG-IP_Tunnel、およびポリシーベースルーティング (PBR) エンティティ NS_f5-BIG-IP_PBR が含まれます。



詳細については、F5 big IP pdf を参照してください。

CloudBridge Connector のトンネル設定について考慮すべきポイント

- NetScaler アプライアンスは稼働中で、インターネットに接続されています。また、CloudBridge Connector トンネルを介してトラフィックを保護するプライベートサブネットにも接続されています。
- F5 BIG-IP アプライアンスは稼働中で、インターネットに接続されています。また、CloudBridge Connector トンネルを介してトラフィックを保護するプライベートサブネットにも接続されています。
- NetScaler アプライアンスと F5 BIG-IP アプライアンスの間の CloudBridge Connector トンネルでは、次の IPsec 設定がサポートされています。
  - IPsec モード: トンネルモード
  - IKE バージョン: バージョン 1

  - IKE 認証方法: 事前共有キー
  - IKE 暗号化アルゴリズム:AES
  - IKE ハッシュアルゴリズム:HMAC SHA1
  - ESP 暗号化アルゴリズム:AES
  - ESP ハッシュアルゴリズム:HMAC SHA1

- CloudBridge Connector トンネルの両端にある NetScaler アプライアンスと F5 BIG-IP アプライアンスで同じ IPsec 設定を指定する必要があります。
- NetScaler には、IKE ハッシュアルゴリズムと ESP ハッシュアルゴリズムを指定するための共通パラメータ（IPSec プロファイル）が用意されています。また、IKE 暗号化アルゴリズムと ESP 暗号化アルゴリズムを指定するためのもう 1 つの共通パラメータも用意されています。そのため、F5 BIG-IP アプライアンスでは、IKE（フェーズ 1 構成）と ESP（フェーズ 2 構成）で同じハッシュアルゴリズムと同じ暗号化アルゴリズムを指定する必要があります。
- 次のことを許可するには、NetScaler 側と F5 BIG-IP 側のファイアウォールを構成する必要があります。

  - ポート 500 の任意の UDP パケット
  - ポート 4500 の任意の UDP パケット
  - 任意の ESP（IP プロトコル番号 50）パケット

CloudBridge Connector トンネル用の **F5 BIG-IP** の設定

NetScaler アプライアンスと F5 BIG-IP アプライアンス間の CloudBridge Connector トンネルを構成するには、F5 BIG-IP アプライアンスで次のタスクを実行します。

- **IPSec** 用の転送仮想サーバーを作成します。転送仮想サーバーは、IPsec トンネルの IP トラフィックをインターセプトします。
- **IKE** ピアを作成します。IKE ピアは、ローカルとリモートの IPsec トンネルエンドポイントを指定します。また、IPSec IKE フェーズ 1 に使用するアルゴリズムと認証情報も指定しています。
- カスタム **IPsec** ポリシーを作成します。ポリシーは、IPsec トンネルの形成に使用する IPsec プロトコル (ESP) とモード (トンネル) を指定します。また、IKE IPsec フェーズ 2 に使用するアルゴリズムとセキュリティパラメータも指定します。
- 双方向 **IPsec** トラフィックセレクターを作成します。トラフィックセレクターは、IP トラフィックが IPsec トンネルを通過する F5 BIG-IP 側と NetScaler 側のサブネットを指定します。

F5 BIG-IP アプライアンスで IPSec VPN（CloudBridge Connector トンネル）を設定する手順は、F5 のリリースサイクルによっては、時間の経過とともに変わる可能性があります。Citrix では、次の URL にある F5 BIG-IP の公式ドキュメントに従って、IPsec VPN トンネルを構成することをお勧めします。

https://f5.com

F5 BIG-IP GUI を使用して IPsec 用の転送仮想サーバーを作成するには

1. [メイン] タブで、[ローカルトラフィック] > [仮想サーバー] をクリックし、[作成] をクリックします。

2.「新規仮想サーバーリスト」画面で、次のパラメーターを設定します。

- **Name**：仮想サーバーの一意の名前を入力します。
- 種類。[転送 **(IP)**] を選択します。
- 宛先アドレス。任意のトラフィックを受け入れるには、ワイルドカードネットワークアドレスを CIDR 形式で入力します。たとえば、IPv4 の場合は 0.0.0.0/0 です。
- サービスポート。リストから [すべてのポート] を選択します。
- プロトコルリスト。リストから [すべてのプロトコル] を選択します。
- **VLAN** とトンネルトラフィック。デフォルトの [すべての **VLAN** とトンネル] のままにします。

3.「完了」をクリックします。

F5 BIG-IP GUI を使用してカスタム IPsec ポリシーを作成するには

1. **[**** メイン] タブで、[ネットワーク] > [**IPsec**] > [**IPsec** ポリシー] をクリックし、[作成] をクリックします。**
2.「新規ポリシー」画面で、次のパラメータを設定します。

- **Name**：ポリシーの固有の名前を入力します。
- **IPsec** プロトコル。デフォルトの選択である ESP のままにします。
- **[**モード**]**。「トンネル」を選択します。画面が更新され、その他の関連設定が表示されます。
- トンネルのローカルアドレス。ローカル IPsec トンネルエンドポイントの IP アドレス（F5 BIG-IP アプライアンスで設定）を入力します。
- トンネルリモートアドレス。リモート IPsec トンネルエンドポイントの IP アドレス（NetScaler アプライアンスで構成）を入力します。

3. IKE フェーズ 2 のパラメータについては、デフォルト値のままにするか、展開に適したオプションを選択してください。
4.「完了」をクリックします。

F5 BIG-IP GUI を使用して双方向 IPsec トラフィックセレクターを作成するには

1. **[**** メイン] タブで、[ネットワーク] > [**IPsec**] > [トラフィックセレクター] をクリックし、[作成] をクリックします。**
2.「新規トラフィックセレクター」画面で、次のパラメータを設定します。

- **Name**：トラフィックセレクターに固有の名前を入力します。
- オーダー。デフォルト値 **(First)** のままにします。この設定では、トラフィックセレクターリスト画面にトラフィックセレクターが表示される順序を指定します。

3.「構成」リストから「詳細」を選択し、次のパラメータを設定します。

- 送信元 **IP** アドレス。****** ホストまたはネットワークをクリックし、****** アドレスフィールドに、IPsec トンネルでトラフィックを保護する F5 BIG-IP 側サブネットのアドレスを入力します。
- 送信元ポート。***** すべてのポートを選択します。

- 宛先 **IP** アドレス。「ホスト」をクリックし、「アドレス」フィールドに、IPsec トンネルでトラフィックを保護する NetScaler 側サブネットのアドレスを入力します。
- 宛先ポート。***** すべてのポートを選択します。
- プロトコル。*** [**すべてのプロトコル] を選択します。
- 方向。[両方] を選択します。
- アクション。[保護] を選択します。**IPsec** ポリシー名の設定が表示されます。
- **IPsec** ポリシー名。作成したカスタム IPsec ポリシーの名前を選択します。

4. 「完了」をクリックします。

CloudBridge Connector トンネル用の **NetScaler ADC** アプライアンスの構成

NetScaler アプライアンスと F5 BIG-IP アプライアンス間の CloudBridge Connector トンネルを構成するには、NetScaler アプライアンスで次のタスクを実行します。NetScaler コマンドラインまたは NetScaler グラフィカルユーザーインターフェイス（GUI）のいずれかを使用できます。

- **IPsec** プロファイルを作成します。IPSec プロファイルエンティティは、IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、CloudBridge Connector トンネル内の IPsec プロトコルで使用される認証方法などの IPsec プロトコルパラメータを指定します。
- **IPsec** プロトコルを使用する **IP** トンネルを作成し、**IPsec** プロファイルをそれに関連付けます。IP トンネルは、ローカル IP アドレス（NetScaler アプライアンス上で構成された CloudBridge Connector のトンネルエンドポイント IP アドレス（SNIP タイプ））、リモート IP アドレス（F5 BIG-IP アプライアンスで構成された CloudBridge Connector のトンネルエンドポイント IP アドレス）、CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPsec）、および IPsec プロファイルエンティティを指定します。作成された IP トンネルエンティティは、CloudBridge Connector トンネルエンティティとも呼ばれます。
- **PBR** ルールを作成して **IP** トンネルに関連付けます。PBR エンティティは、ルールセットと IP トンネル (CloudBridge Connector トンネル) エンティティを指定します。送信元 IP アドレス範囲と宛先 IP アドレス範囲は PBR エンティティの条件です。送信元 IP アドレス範囲を設定してトラフィックをトンネルで保護する NetScaler 側のサブネットを指定し、宛先 IP アドレス範囲を設定してトラフィックをトンネルで保護する F5 BIG-IP 側のサブネットを指定します。

NetScaler コマンドラインを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで入力します。

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecyENABLE`
- `show ipsec profile** <name>`

NetScaler コマンドラインを使用して IPSEC トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

NetScaler コマンドラインを使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

コマンドプロンプトで入力します。

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

GUI を使用して IPSEC プロファイルを作成するには

1. **[**** システム] > [**CloudBridge Connector**] > [IPsec プロファイル] に移動します。******
2. 詳細ペインで、[追加] をクリックします。
3. **IPsec** プロファイルの追加ページで、次のパラメータを設定します。
    - 名前
    - 暗号化アルゴリズム
    - ハッシュアルゴリズム
    - IKE プロトコルバージョン
4. 2 つの CloudBridge Connector トンネルピアが相互認証に使用する IPSec 認証方法を設定します。事前共有キー認証方法 を選択し、事前共有キーが存在 パラメータを設定します。
5. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2.「IPv4 トンネル」タブで、「追加」をクリックします。
3. **IP** トンネルの追加ページで、次のパラメータを設定します。
    - 名前
    - リモート IP
    - リモートマスク
    - ローカル IP タイプ (「ローカル IP タイプ」ドロップダウンリストで、「サブネット *IP*」を選択します)。
    - ローカル IP（選択した IP タイプの設定済み IP アドレスがすべてローカル IP ドロップダウンリストに表示されます。リストから目的の IP を選択します。)
    - プロトコル
    - IPSec プロファイル
4. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して PBR ルールを作成し、IPSEC トンネルをそのルールにバインドするには

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3.「**PBR** の作成」ページで、次のパラメータを設定します。

    - 名前
    - アクション
    - ネクストホップタイプ (*IP* トンネルの選択)
    - IP トンネル名
    - 送信元 IP アドレスが低い
    - ソース IP ハイ
    - デスティネーション IP フロー
    - デスティネーション IP ハイ

4. [作成] をクリックし、[閉じる] をクリックします。

NetScaler アプライアンス上の対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。CloudBridge Connector トンネルの現在のステータスは、設定済みの CloudBridge Connector ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

次のコマンドは、「CloudBridge Connector 構成の例」の NetScaler アプライアンス NS_Appliance-1 の設定を作成します。:

```
1      > add ipsec profile NS_F5-BIG-IP_IPSec_Profile -psk
           examplepresharedkey -ikeVersion v1 -encAlgo AES -hashalgo
           HMAC_SHA1 -lifetime 315360 -perfectForwardSecrecy ENABLE
2
3
4      Done
5
6      > add iptunnel NS_F5-BIG-IP_Tunnel 203.0.113.200 255.255.255.255
           198.51.100.100 -protocol IPSEC -ipsecProfileName NS_F5-BIG-
           IP_IPSec_Profile
7
8
9      Done
10
11     > add pbr NS_F5-BIG-IP_Pbr -srcIP 10.102.147.0-10.102.147.255 -
           destIP 10.20.0.0-10.20.255.255 -ipTunnel NS_F5-BIG-IP_Tunnel
12
13
14     Done
15
16     > apply pbrs
17
18
19     Done
20 <!--NeedCopy-->
```


CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、NetScaler ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。NetScaler アプライアンスでの CloudBridge Connector のトンネル統計の表示について詳しくは、「CloudBridge Connector トンネルの監視」を参照してください。

CloudBridge Connector の相互運用性—Cisco ASA

August 15, 2023

NetScaler アプライアンスと Cisco ASA アプライアンスの間に CloudBridge Connector トンネルを構成して、2 つのデータセンターを接続したり、ネットワークをクラウドプロバイダーに拡張したりできます。NetScaler アプライアンスと Cisco ASA アプライアンスは CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

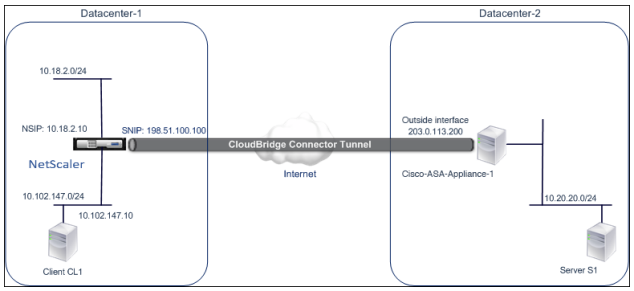
CloudBridge Connector のトンネル設定の例

CloudBridge Connector トンネル内のトラフィックフローの図として、次のアプライアンス間に CloudBridge Connector トンネルが設定されている例を考えてみましょう。

- データセンター 1 として指定されたデータセンターにある NetScaler アプライアンス NS_Appliance-1
- データセンター-2 として指定されたデータセンター内の Cisco ASA アプライアンス Cisco-ASA アプライアンス-1

NS_Appliance-1 と Cisco-asa-Appliance-1 により、CloudBridge Connector トンネルを介したデータセンター 1 とデータセンター 2 のプライベートネットワーク間の通信が可能になります。この例では、NS_Appliance-1 と Cisco-ASA-Appliance-1 により、CloudBridge Connector トンネルを介したデータセンター 1 のクライアント CL1 とデータセンター 2 のサーバー S1 間の通信が可能になります。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector のトンネル構成には IPsec プロファイルエンティティ NS_Cisco-ASA_IPsec_Profile、CloudBridge Connector トンネルエンティティ NS_Cisco-ASA_Tunnel、およびポリシーベースルーティング (PBR) エンティティ ns_Cisco-ASA_PBR が含まれます。



CloudBridge Connector のトンネル設定について考慮すべきポイント

CloudBridge Connector トンネルの設定を開始する前に、次のことを確認してください。

- NetScaler アプライアンスと Cisco ASA アプライアンスの間の CloudBridge Connector トンネルでは、次の IPsec 設定がサポートされています。

IPSec のプロパティ	設定
IPsec モード	トンネルモード
IKE バージョン	バージョン 1
IKE 認証方式	事前共有キー
IKE 暗号化アルゴリズム	AES, 3DES
IKE ハッシュアルゴリズム	HMAC SHA1、HMAC MD5
ESP 暗号化アルゴリズム	AES, 3DES
ESP ハッシュアルゴリズム	HMAC SHA1、HMAC MD5

- CloudBridge Connector トンネルの両端にある NetScaler アプライアンスと Cisco ASA アプライアンスで同じ IPsec 設定を指定する必要があります。
- NetScaler には、IKE ハッシュアルゴリズムと ESP ハッシュアルゴリズムを指定するための共通パラメータ（IPSec プロファイル）が用意されています。また、IKE 暗号化アルゴリズムと ESP 暗号化アルゴリズムを指定するためのもう 1 つの共通パラメータも用意されています。そのため、Cisco ASA アプライアンスでは、IKE（フェーズ 1 設定）と ESP（フェーズ 2 設定）で同じハッシュアルゴリズムと同じ暗号化アルゴリズムを指定する必要があります。
- 次のことを許可するには、NetScaler 側と Cisco ASA 側でファイアウォールを構成する必要があります。
  - ポート 500 の任意の UDP パケット
  - ポート 4500 の任意の UDP パケット
  - 任意の ESP（IP プロトコル番号 50）パケット

CloudBridge Connector トンネル用の **Cisco ASA** の設定

Cisco ASA アプライアンスで CloudBridge Connector トンネルを設定するには、Cisco ASA アプライアンスを設定、監視、および保守するための主要なユーザインターフェイスである Cisco ASA コマンドラインインターフェイスを使用します。

Cisco ASA アプライアンスで CloudBridge Connector トンネル設定を開始する前に、次のことを確認してください。

- Cisco ASA アプライアンスの管理者認証情報を持つユーザアカウントを持っています。

- Cisco ASA コマンドラインインターフェイスに精通している。
- Cisco ASA アプライアンスは稼働中で、インターネットに接続されています。また、CloudBridge Connector トンネルを介してトラフィックを保護するプライベートサブネットにも接続されています。

> 注
>
> Cisco ASA アプライアンスで CloudBridge Connector トンネルを設定する手順は、Cisco のリリースサイクルによっては、時間の経過とともに変わる可能性があります。Citrix では、次の URL にある IPsec VPN トンネルの設定に関する Cisco ASA の公式製品マニュアルに従うことを推奨しています。

- http://www.cisco.com

NetScaler アプライアンスと Cisco ASA アプライアンス間の CloudBridge Connector トンネルを構成するには、Cisco ASA アプライアンスのコマンドラインで次のタスクを実行します。

- **IKE** ポリシーを作成します。IKE ポリシーは、IKE ネゴシエーション（フェーズ 1）中に使用するセキュリティパラメータの組み合わせを定義します。たとえば、IKE ネゴシエーションで使用されるハッシュアルゴリズム、暗号化アルゴリズム、認証方法などのパラメータは、このタスクで設定されます。
- 外部インターフェイスで **IKE** を有効にします。トンネルトラフィックがトンネルピアに流れる外部インターフェイスで IKE を有効にします。
- トンネルグループを作成します。トンネルグループは、トンネルのタイプと事前共有キーを指定します。トンネルタイプは ipsec-l2l に設定する必要があります。これは IPsec LAN to LAN の略です。事前共有キーはテキスト文字列で、CloudBridge Connector トンネルのピアが相互に認証するために使用します。事前共有キーは相互に照合され、IKE 認証が行われます。そのため、認証を成功させるには、Cisco ASA アプライアンスと NetScaler アプライアンスに同じ事前共有キーを設定する必要があります。
- トランスフォームセットを定義します。トランスフォームセットは、IKE ネゴシエーションが成功した後の CloudBridge Connector トンネルを介したデータ交換に使用されるセキュリティパラメータ（フェーズ 2）の組み合わせを定義します。
- アクセスリストを作成します。暗号アクセスリストを使用して、IP トラフィックを CloudBridge トンネルで保護するサブネットを定義します。アクセスリストのソースとターゲットのパラメータには、CloudBridge Connector トンネルを介して保護される Cisco アプライアンス側と NetScaler 側のサブネットを指定します。アクセスリストは permit に設定する必要があります。Cisco アプライアンス側のサブネット内のアプライアンスから送信され、NetScaler 側のサブネット内のアプライアンスを宛先とするリクエストパケットで、アクセスリストの送信元と宛先のパラメーターと一致するリクエストパケットは、CloudBridge Connector トンネルを介して送信されます。
- クリプトマップを作成します。クリプトマップは、セキュリティアソシエーション (SA) の IPsec パラメータを定義します。これらには、CloudBridge トンネルを介してトラフィックを保護するサブネットを識別する暗号アクセスリスト、IP アドレスによるピア（NetScaler）識別、およびピアセキュリティ設定と一致するトランスフォームセットが含まれます。
- クリプトマップを外部インターフェイスに適用します。このタスクでは、トンネルトラフィックがトンネルピアに流れる外部インターフェイスにクリプトマップを適用します。クリプトマップをインターフェイスに適用すると、Cisco ASA アプライアンスは、すべてのインターフェイストラフィックをクリプトマップセットと照

らし合わせて評価し、接続またはセキュリティアソシエーションネゴシエーション中に指定されたポリシーを使用するように指示します。

以下の手順の例では、CloudBridge Connector の設定とデータフローの例で使用されている Cisco ASA アプライアンス Cisco-ASA-Appliance-1 の設定を作成します。

Cisco ASA コマンドラインを使用して IKE ポリシーを作成するには

Cisco ASA アプライアンスのコマンドプロンプトで、次のコマンドをグローバル構成モードから順番に入力します。

コマンド	例	コマンドの説明
crypto ikev1 policy priority	Cisco-ASA-appliance-1(config)# crypto ikev1 policy 1	IKE ポリシー構成モードを開始し、作成するポリシーを指定します。（各ポリシーは、割り当てた優先度番号によって一意に識別されます。）この例では、ポリシー 1 を設定しています。
encryption (3des | aes)	Cisco-ASA-appliance-1 (config-ikev1-policy)# encryption 3des	暗号化アルゴリズムを指定します。この例では、3DES アルゴリズムを設定します。
hash (sha | md5)	Cisco-ASA-appliance-1 (config-ikev1-policy)# hash sha	ハッシュアルゴリズムを指定します。この例では SHA を設定します。
authenticationpre-share	Cisco-ASA-appliance-1 (config-ikev1-policy)# authentication pre-share	事前共有認証方法を指定します。
グループ 2	Cisco-ASA-appliance-1 (config-ikev1-policy)# group 2	1024 ビットの Diffie-Hellman グループ識別子 (2) を指定します。
ライフタイム秒	Cisco-ASA-appliance-1 (config-ikev1-policy)# lifetime 28800	セキュリティアソシエーションの有効期間を秒単位で指定します。この例では、NetScaler アプライアンスのライフタイムのデフォルト値である 28800 秒を構成しています。

Cisco ASA コマンドラインを使用して外部インターフェイスで IKE を有効にするには

Cisco ASA アプライアンスのコマンドプロンプトで、次のコマンドをグローバル構成モードから順番に入力します。

コマンド	例	コマンドの説明
crypto ikev1 enable outside	Cisco-ASA-appliance-1(config)# crypto ikev1 enable outside	トンネルトラフィックがトンネルピアに流れるインターフェイスで IKEv1 を有効にします。この例では、outside という名前のインターフェイスで IKEv1 を有効にします。

To create a tunnel group by using the Cisco ASA command line

Cisco ASA アプライアンスのコマンドプロンプトで、Cisco ASA コマンドラインを使用して、接続された pdf トンネルグループの show のように、グローバル構成モードで次のコマンドを入力します。

Cisco ASA コマンドラインを使用してクリプトアクセスリストを作成するには

Cisco ASA アプライアンスのコマンドプロンプトで、グローバル構成モードで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
access-list access-list-number permit IP source source-wildcard destination destination-wildcard	Cisco-ASA-appliance-1(config)# access-list 111 permit ip 10.20.20.0 0.0.0.255 10.102.147.0 0.0.0.255	CloudBridge Connector トンネルで IP トラフィックを保護するサブネットを決定する条件を指定します。この例では、サブネット 10.20.20.0/24（Cisco-ASA-Appliance-1 側）と 10.102.147.0/24（NS_Appliance-1 側）からのトラフィックを保護するようにアクセスリスト 111 を設定しています。

Cisco ASA コマンドラインを使用してトランスフォームセットを定義するには

Cisco ASA アプライアンスのコマンドプロンプトで、グローバル構成モードで次のコマンドを入力します。ASA コマンドラインテーブルを使用したトランスフォームセット pdf を参照してください。

Cisco ASA コマンドラインを使用してクリプトマップを作成するには

Cisco ASA アプライアンスのコマンドプロンプトで、グローバル構成モードから次のコマンドを順番に入力します。

コマンド	例	コマンドの説明
crypto map map-name seq-num match address access-list-name	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 match address 111	クリプトマップを作成し、それにアクセスリストを指定します。この例では、シーケンス番号 1 のクリプトマップ NS-CISCO-CM を設定し、NS-CISCO-CM にアクセスリスト 111 を割り当てています。
crypto map map-name seq-num set peer ip-address	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 set peer 198.51.100.100	ピア（NetScaler アプライアンス）を IP アドレスで指定します。この例では、NetScaler アプライアンスのトンネルエンドポイント IP アドレスである 198.51.100.100 を指定しています。
crypto map map-name seq-num set ikev1 transform-set transform-set-name	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 set ikev1 transform-set NS-CISCO-TS	このクリプトマップエントリに使用できるトランスフォームセットを指定します。この例では、トランスフォームセット NS-CISCO-TS を指定しています。

Cisco ASA コマンドラインを使用してクリプトマップをインターフェイスに適用するには

Cisco ASA アプライアンスのコマンドプロンプトで、グローバル構成モードから次のコマンドを順番に入力します。

コマンド	例	コマンドの説明
crypto map map-nameinterface interface-name	Cisco-ASA-appliance-1(config)# crypto map NS-CISCO-CM interface outside	CloudBridge Connector トンネルトラフィックが通過するインターフェイスにクリプトマップを適用します。この例では、クリプトマップ NS-CISCO-CM を外部のインターフェイスに適用します。

CloudBridge Connector トンネル用の **NetScaler ADC** アプライアンスの構成

NetScaler アプライアンスと Cisco ASA アプライアンス間の CloudBridge Connector トンネルを構成するには、NetScaler アプライアンスで次のタスクを実行します。NetScaler コマンドラインまたは NetScaler グラフィカルユーザーインターフェイス（GUI）のいずれかを使用できます。

- IPsec プロファイルを作成します。

- IPsec プロトコルを使用する IP トンネルを作成し、IPsec プロファイルをそれに関連付けます。
- PBR ルールを作成して IP トンネルに関連付けます。

NetScaler コマンドラインを使用して **IPSEC** プロファイルを作成するには:

コマンドプロンプトで入力します。

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecy ENABLE`
- `show ipsec profile <name>`

NetScaler コマンドラインを使用して **IPSEC** トンネルを作成し、**IPSEC** プロファイルをそのトンネルにバインドするには:

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

NetScaler コマンドラインを使用して **PBR** ルールを作成し、**IPSEC** トンネルをそのルールにバインドするには:

コマンドプロンプトで入力します。

- `**add pbr** <pbrName> **ALLOW** -**srcIP** <subnet-range> -**destIP** <subnet-range>`
- `**ipTunnel** <tunnelName>`
- `**apply pbrs**`
- `**show pbr** <pbrName>`

GUI を使用して **IPSEC** プロファイルを作成するには**:**

1. [システム] > [**CloudBridge Connector**] > [**IPSec** プロファイル] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. **IPsec** プロファイルの追加ページで、次のパラメータを設定します。
    - 名前
    - 暗号化アルゴリズム
    - ハッシュアルゴリズム
    - IKE プロトコルバージョン
    - Perfect Forward Secrecy (このパラメータを有効にする)
4. **2** つの **CloudBridge Connector** トンネルピアが相互認証に使用する **IPsec** 認証方法を設定します。事前共有キー認証方法を選択し、事前共有キーが存在するパラメータを設定します。
5. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して **IP** トンネルを作成し、**IPSEC** プロファイルをそのトンネルにバインドするには:

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2. **[**IPv4** トンネル] タブで、[追加] をクリックします。**
3. **IP** トンネルの追加ページで、次のパラメータを設定します。

    - 名前
    - リモート IP
    - リモートマスク
    - ローカル IP タイプ (「ローカル IP タイプ」ドロップダウンリストで、「サブネット IP」を選択します)。
    - ローカル IP（選択した IP タイプの設定済み IP アドレスがすべてローカル IP ドロップダウンリストに表示されます。リストから目的の IP を選択します。)
    - プロトコル
    - IPSec プロファイル

4. [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して **PBR** ルールを作成し、**IPSEC** トンネルをそのルールにバインドするには**:**

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3. [**PBR** の 作成**]** ページで、次のパラメーターを設定します。

    - 名前
    - アクション
    - ネクストホップタイプ (IP トンネルの選択)
    - IP トンネル名
    - 送信元 IP アドレスが低い
    - ソース IP ハイ
    - デスティネーション IP フロー
    - デスティネーション IP ハイ

4. [作成] をクリックし、[閉じる] をクリックします。

NetScaler アプライアンス上の対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。CloudBridge Connector トンネルの現在のステータスは、設定済みの CloudBridge Connector ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

次のコマンドは、「CloudBridge Connector 構成の例」の NetScaler アプライアンス NS_Appliance-1 の設定を作成します。

```
1 >  add ipsec profile NS_Cisco-ASA_IPSec_Profile -psk
     examplepresharedkey -ikeVersion v1  - encAlgo AES  - hashalgo
     HMAC_SHA1  - lifetime 315360 -perfectForwardSecrecy ENABLE
2
```



```
Done

>  add iptunnel NS_Cisco-ASA_Tunnel 203.0.113.200 255.255.255.255
   198.51.100.100  -protocol IPSEC  -ipsecProfileName NS_Cisco-
   ASA_IPSec_Profile


Done

> add pbr NS_Cisco-ASA_Pbr -srcIP 10.102.147.0-10.102.147.255  -destIP
   10.20.0.0-10.20.255.255  -ipTunnel NS_Cisco-ASA_Tunnel


Done

> apply pbrs

Done

<!--NeedCopy-->
```

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、NetScaler ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。NetScaler アプライアンスでの CloudBridge Connector のトンネル統計の表示について詳しくは、「CloudBridge Connector トンネルの監視」を参照してください。

高可用性

August 15, 2023

2 台の NetScaler ADC アプライアンスを高可用性（HA）で展開すると、どのトランザクションでも中断のない操作を実現できます。一方のアプライアンスをプライマリノードとして、もう一方をセカンダリノードとして構成すると、プライマリノードが接続を受け入れてサーバーを管理し、セカンダリノードがプライマリノードを監視します。何らかの理由でプライマリノードが接続を受け付けることができなくなると、セカンダリノードが処理を引き継ぎます。

セカンダリノードは、プライマリノードが接続を受け付けているかどうかを判断するために、定期的なメッセージ（ハートビートメッセージまたはヘルスチェックとも呼ばれます）を送信してプライマリを監視します。ヘルスチェックが失敗した場合、セカンダリノードは指定された期間接続を再試行し、その後、プライマリノードが正常に機能していないと判断します。次に、セカンダリノードがプライマリ（フェールオーバーと呼ばれるプロセス）を引き継ぎます。

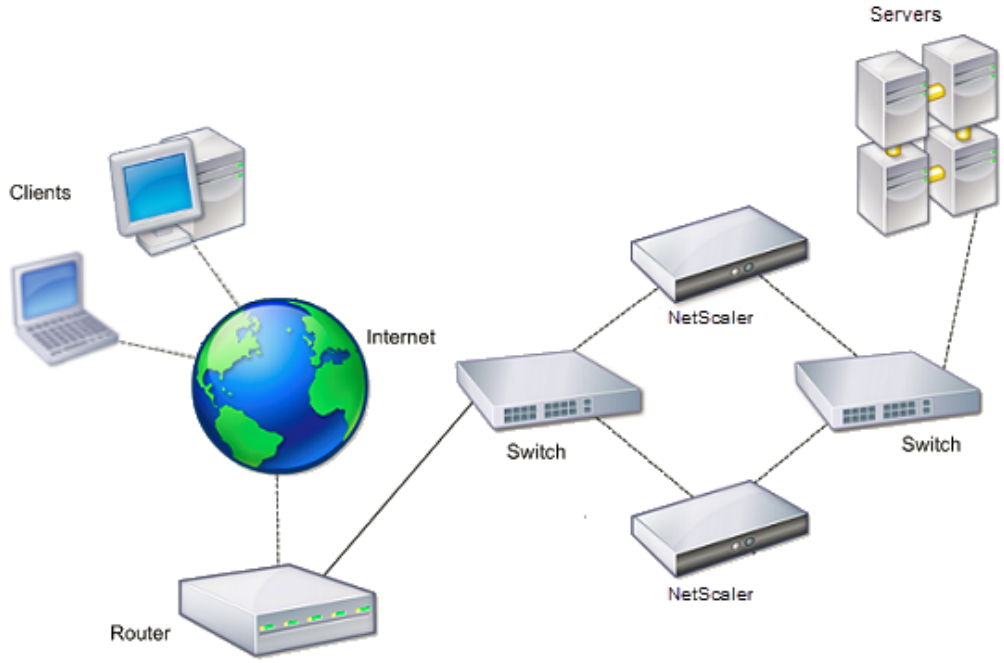
フェイルオーバー後、すべてのクライアントは管理対象サーバーへの接続を再確立する必要がありますが、セッション永続性ルールはフェイルオーバー前と同じように維持されます。

Web サーバーのロギングパーシステンスを有効にすると、フェールオーバーによってログデータが失われることはありません。ロギングパーシステンスを有効にするには、ログサーバーの設定で log.conf ファイルに両方のシステムのエントリが含まれている必要があります。

> 注:
>
> 場合によっては、プライマリノードがセカンダリノードのプロキシとして使用されます。

次の図は、HA ペアのネットワーク構成を示しています。

図 1: 高可用性構成の NetScaler アプライアンス



HA を設定するには、まず両方のノードを同じサブネットに設定して基本設定を作成するとよいでしょう。その後、ノードがヘルスチェック情報を通信する間隔、ノードが同期を維持するプロセス、プライマリからセカンダリへのコマンドの伝達をカスタマイズできます。フェイルセーフモードを設定して、どちらのノードもプライマリではない状況を防ぐことができます。ご使用の環境に NetScaler 無償の ARP メッセージを受け入れないデバイスが含まれている場合は、仮想 MAC アドレスを設定する必要があります。より複雑な構成の準備ができたら、HA ノードをさまざまなサブネットに設定できます。

HA 設定の信頼性を高めるために、ルートモニターを設定し、冗長リンクを作成できます。トラブルシューティングやメンテナンスタスクの実行など、状況によっては、ノードを強制的にフェイルオーバーする（プライマリステータスを他のノードに割り当てる）場合や、セカンダリノードを強制的にセカンダリにしたり、プライマリノードをプライマリにしたりしたい場合があります。

高可用性セットアップの考慮事項

August 15, 2023

> 注
>
> HA セットアップでシステムを設定するには、次の要件が必要です。

- HA 構成では、プライマリとセカンダリの NetScaler アプライアンスは同じモデルでなければなりません。異なる NetScaler モデルは HA ペアではサポートされていません。また、異なるモデルに展開された NetScaler VPX は、HA ペアではサポートされません。HA ペアを形成できるのは、同じモデルにデプロイされた NetScaler VPX だけです。
- HA セットアップでは、両方のノードで同じバージョンの NetScaler を実行する必要があります。
- プライマリシステムとセカンダリシステムの両方で構成ファイル（ns.conf）のエントリが一致する必要があります。ただし、次の例外があります。
  - プライマリシステムとセカンダリシステムは、それぞれ固有の IP アドレス (NSIP) で構成する必要があります。
  - HA ペアでは、一方のノードのノード ID と関連する IP アドレスがもう一方のノードを指している必要があります。たとえば、NS1 と NS2 というノードがある場合、NS1 には一意のノード ID と NS2 の IP アドレスを、NS2 には一意のノード ID と NS1 の IP アドレスで構成する必要があります。
- GUI や CLI を直接経由しない方法（SSL 証明書のインポート、スタートアップスクリプトへの変更など）を使用していずれかのノードに設定ファイルを作成する場合は、設定ファイルを他のノードにコピーするか、そのノードに同じファイルを作成する必要があります。
- 最初は、すべての NetScaler アプライアンスが同じ RPC ノードパスワードで構成されます。RPC ノードは、構成およびセッション情報のシステム間通信に使用される内部システムエンティティです。セキュリティ上の理由から、デフォルトの RPC ノードパスワードを変更する必要があります。

  各 NetScaler には 1 つの RPC ノードが存在します。このノードにはパスワードが保存され、連絡先システムから提供されたパスワードと照合されます。他のシステムと通信するには、各 NetScaler が、それらのシステムでの認証方法など、それらのシステムに関する知識を必要とします。RPC ノードは、他のシステムの IP アドレスや認証に必要なパスワードなど、この情報を保持します。

  RPC ノードは、ノードを追加するとき、またはグローバルサーバー負荷分散 (GSLB) サイトを追加するときに暗黙的に作成されます。RPC ノードを手動で作成または削除することはできません。

  > 注:
  >
  > 高可用性セットアップの NetScaler アプライアンスがワンアームモードで構成されている場合は、スイッチまたはハブに接続されているものを除くすべてのシステムインターフェイスを無効にする必要があります。

IPv6 HA 構成には、次の考慮事項が適用されます。

- IPv6pt ライセンスを両方の NetScaler アプライアンスにインストールする必要があります。
- IPv6pt ライセンスをインストールしたら、GUI またはコマンドラインインターフェイスを使用して IPv6 機能を有効にします。
- どちらの NetScaler アプライアンスにもグローバル NSIP IPv6 アドレスが必要です。さらに、2 つのノード間のネットワークエンティティ (スイッチやルーターなど) は IPv6 をサポートしている必要があります。

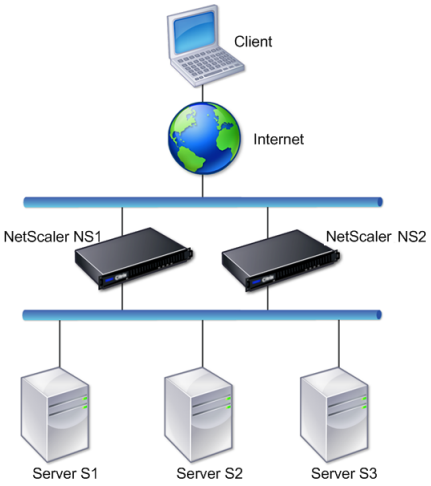
高可用性の設定

August 15, 2023

高可用性構成を設定するには、2 つのノードを作成します。各ノードは、もう一方の NetScaler ADC IP（NSIP）アドレスをリモートノードとして定義します。まず、高可用性を構成する 2 つの NetScaler ADC アプライアンスのいずれかにログオンし、ノードを追加します。他のアプライアンスの NetScaler ADC IP（NSIP）アドレスを新しいノードのアドレスとして指定します。次に、もう一方のアプライアンスにログオンし、最初のアプライアンスの NSIP アドレスを持つノードを追加します。アルゴリズムは、どのノードがプライマリになり、どのノードがセカンダリになるかを決定します。

次の図は、両方のノードが同じサブネットにある単純な HA 設定を示しています。

図 1：高可用性構成で接続された 2 つの NetScaler アプライアンス



高可用性を設定する手順

2 つの NetScaler ADC アプライアンスの高可用性ペアをセットアップするには、両方のアプライアンスで次のタスクを実行します。

- ノードを追加します。アプライアンス（たとえば N1）で、一意のノード ID とアプライアンス（N2）の NSIP アドレスを指定して、もう 1 つのアプライアンス（たとえば N2）を追加します。ピアノード ID には、1 ～64 の範囲の任意の整数を指定できます。

  セルフノードで指定されたピアノード ID はセルフノードにのみ適用され、ピアノードには関係ありません。たとえば、N1 のピアノードとして N2 を追加し、N2 のノード ID を 33 と指定したとします。N2 のノード ID が 33 に設定されているのは N1 にのみ適用され、N2 の構成には影響しません。

  両方のノードで指定されているピアノード ID は、同じ値である必要はなく、変更できます。どちらのノードでも、セルフノード ID は 0 にハードコードされており、変更できません。

- 未使用のインターフェイスの **HA** モニターを無効にします。セルフノードでは、接続されていない、またはトラフィックに使用されていない各インターフェイスの HA モニタを無効にする必要があります。未使用のインターフェイスの HA モニタを無効にすると、これらの未使用のインターフェイスのいずれかの状態が DOWN になったときに発生する HA フェールオーバーを防ぐことができます。

> 注記:
>
> 高可用性構成の各ノードの設定が同じになるようにするには、SSL 証明書、起動スクリプト、およびその他の構成ファイルをプライマリノードのものと同期させる必要があります。

CLI のプロシージャ

CLI を使用して 2 つの NetScaler ADC アプライアンスの高可用性ペアをセットアップするには、2 つのアプライアンスのそれぞれで次のタスクを実行します。

CLI を使用してノードを追加するには**:**

コマンドプロンプトで入力します。

- `add ha node <id> <IPAddress>`
- `show ha node`

CLI を使用して未使用のインターフェイスの **HA** モニタを無効にするには:

コマンドプロンプトで入力します。

- `set interface <ifNum> [-haMonitor ( ON | OFF )]`
- `show interface <ifNum>`

例**:**


```
> add ha node 33 203.0.113.33

> set interface 1/3 -haMonitor OFF
 Done
<!--NeedCopy-->
```

GUI プロシージャ

NetScaler GUI には、ピアノードを追加するタスクと、セルフノード上の未使用のインターフェイスで HA モニターを無効にするタスクを組み合わせた画面が表示されます。この画面には、HA セットアップ用にピアノードを自動的に構成するオプションもあり、ピアノードを手動で構成する必要がなくなります。

GUI を使用して **2** つの **NetScaler ADC** アプライアンスの高可用性ペアをセットアップするには:

1. いずれかのアプライアンスの GUI にログインします。
2. [システム] > [高可用性] > [ノード] に移動し、[リモートノード **IP** アドレス**]** フィールドにピアノードの **NSIP** アドレスを入力します。
3. [停止している **HA** モニターのインターフェイス**/**チャネルをオフにする] を選択します。
4. [高可用性セットアップに参加するようにリモートシステムを設定] を選択し、ピアノードのログイン認証情報を入力します。
5. [作成] をクリックします。

ノードの無効化または有効化

セカンダリノードのみを無効または有効にできます。セカンダリノードを無効にすると、プライマリノードへのハートビートメッセージの送信が停止するため、プライマリノードはセカンダリのステータスを確認できなくなります。ノードを有効にすると、そのノードは高可用性構成に参加します。

コマンドラインインターフェイスを使用してノードを無効または有効にするには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `set ha node -hastatus DISABLED`
- `set ha node -hastatus ENABLED`

GUI を使用してノードを無効または有効にするには

1. [システム] > [高可用性] に移動し、[ノード] タブでノードを開きます。
2. [高可用性ステータス] リストで、**[**有効**] (HA** にアクティブに参加する**)** または [無効**] (HA** に参加しない**)** を選択します。

通信間隔の設定

August 15, 2023

hello 間隔は、ハートビートメッセージがピアノードに送信される間隔です。デッドインターバルは、ハートビートパケットが受信されない場合にピアノードが DOWN とマークされるまでの時間間隔です。ハートビートメッセージは、HA ペアの他のノードのポート 3003 に送信される UDP パケットです。デッドインターバルは hello インターバルの倍数として設定する必要があります。デフォルトでは、hello 間隔は 200 ミリ秒に設定され、デッド間隔は 3 秒に設定されています。

コマンドラインインターフェイスを使用して **hello** 間隔とデッドインターバルを設定するには

コマンドプロンプトで入力します。

- `set HA node [-helloInterval <msecs>] [-deadInterval <secs>]`
- `show HA node <id>`

GUI を使用して **hello** インターバルとデッドインターバルを設定するには

1. [システム] > [高可用性] に移動し、[ノード] タブでノードを開きます。
2. 次のパラメーターを設定します。
    - ハロー間隔 (ミリ秒)
    - デッド間隔 (秒)

同期の設定

December 8, 2023

同期は、プライマリノードの設定をセカンダリノードに複製するプロセスです。同期の目的は、発生するフェイルオーバーの数に関係なく、プライマリノードとセカンダリノード間で構成情報が失われないようにすることです。同期には TCP ポート 3008 または 3010 が使用されます。

次の条件のいずれかが満たされると、同期がトリガーされます:

- HA セットアップのセカンダリノードは、再起動後に起動します。
- プライマリノードは、フェールオーバー後にセカンダリになります。

自動同期はデフォルトで有効になっています。同期を強制することもできます。

> 注:
>
> - HA 同期中は、コマンドの伝播が失敗する原因となる可能性のあるコマンド設定の競合を防ぐため、コマンド伝播は無効になっています。
> - HA 同期中、セカンダリノードは`clear ns config`コマンドを実行して既存の設定をクリアし、プライマリノードから取得した新しい設定をロードします。ただし、クリア構成では、セカンダリノードに設定されているデフォルトのスタティックルートは削除されません。このデフォルトの静的ルートが誤ったゲートウェイを指している場合、サービスのダウンタイムが発生する可能性があります。

同期の無効化または有効化

自動 HA 同期は、HA ペアの各ノードでデフォルトで有効になっています。どちらのノードでも有効または無効にできます。

コマンドラインインターフェイスを使用して自動同期を無効または有効にするには

コマンドプロンプトで入力します:

- `set HA node -haSync DISABLED`
- `set HA node -haSync ENABLED`

GUI を使用して同期を無効または有効にするには

1. [**System**] > [**High Availability**] に移動します。
2. [HA 同期] で、[セカンダリノード] を選択解除または選択すると、[プライマリ] オプションから構成が取得されます。

セカンダリノードを強制的にプライマリノードと同期させる

自動同期に加えて、NetScaler は強制同期をサポートしています。プライマリノードとセカンダリノードのどちらからでも強制的に同期できます。セカンダリノードから強制的に同期すると、プライマリノードとの設定の同期が開始されます。

ただし、同期が既に進行中の場合、強制同期は失敗し、システムは警告を表示します。強制同期は、次のいずれかの状況でも失敗します。

- スタンドアロンシステムで同期を強制します。
- セカンダリノードは無効です。
- HA 同期はセカンダリノードで無効になっています。

コマンドラインインターフェイスを使用して強制的に同期させるには

コマンドプロンプトで入力します：

```
force HA sync
```

GUI を使用して強制的に同期させるには

1. ［**System**］ > ［**High Availability**］に移動します。
2.「ノード」タブの「アクション」リストで、「強制同期」をクリックします。

高可用性セットアップの設定ファイルの同期

August 15, 2023

高可用性セットアップでは、すべての構成ファイルが 1 分間隔でプライマリノードからセカンダリノードに自動的に同期されます。構成ファイルの同期は、プライマリノードまたはセカンダリノードのいずれかでコマンドラインインターフェイスまたは GUI を使用して手動で実行できます。

セカンダリに固有の（プライマリに存在しない）セカンダリ上にあるファイルは、同期中に削除されません。

コマンドラインインターフェイスを使用して高可用性セットアップでファイルを同期するには

コマンドプロンプトで入力します。

```
sync HA files <mode>
```

例

```
1 > sync HA files all
2  Done
3 <!--NeedCopy-->
```

```
1 > sync HA files ssl
2  Done
3 <!--NeedCopy-->
```

パラメータの説明（**CLI** プロシージャにリストされているコマンドの）

```
sync ha files <mode>
```

```
mode
```

次のいずれかの同期モードを指定します。

- **all** -システム構成、Access Gateway のブックマーク、SSL 証明書、SSL CRL リスト、およびアプリケーションファイアウォール XML オブジェクトに関連するファイルを同期します。
- **bookmarks** -すべての Access Gateway ブックマークを同期します。
- **ssl-SSL** 機能のすべての証明書、キー、および CRL を同期します。
- **import** -アプリケーションファイアウォール用に設定されたすべての XML オブジェクト (WSDL、スキーマ、エラーページなど) を同期します。
- **misc** -すべてのライセンスファイルと rc.conf ファイルを同期します。
- **all_plus_misc** -システム構成、Access Gateway のブックマーク、SSL 証明書、SSL CRL リスト、アプリケーションファイアウォールの XML オブジェクト、ライセンス、および rc.conf ファイルに関連するファイルを同期します。

GUI を使用して高可用性セットアップのファイルを同期するには

[システム] > [診断] に移動し、[ユーティリティ] グループで [**HA** ファイル同期の開始] をクリックします。

コマンド伝播の設定

December 8, 2023

HA セットアップでは、プライマリノードで発行されたコマンドはすべてセカンダリノードに伝播されます。伝播されたコマンドは、プライマリノードで実行される前にセカンダリノードで実行されます。セカンダリでコマンドの伝播または実行が失敗した場合、プライマリノードはコマンドを実行してエラーを記録します。コマンド伝播には TCP ポート 3008 または 3010 が使用されます。

HA ペア構成では、コマンドの伝播はプライマリノードとセカンダリノードの両方でデフォルトで有効になっています。HA ペアのいずれかのノードでコマンドの伝播を有効または無効にできます。1 次ノードでコマンド伝達を無効にすると、コマンドは 2 次ノードに伝播されません。2 次ノードでコマンド伝達を無効にすると、1 次ノードから伝播されたコマンドは 2 次ノードでは実行されません。

> 注
>
> 伝播を再度有効にしたら、必ず同期を強制してください。

伝播を無効にしているときに同期が行われると、伝播を無効にする前に行った構成関連の変更は、すべてセカンダリノードと同期されます。これは、同期の進行中に伝播が無効になっている場合にも当てはまります。

コマンドラインインターフェイスを使用してコマンドの伝播を無効または有効にするには

コマンドプロンプトで入力します:

- HA ノードを設定-HAProp を無効化
- HA ノードを設定-HAProp を有効にする

GUI を使用してコマンドの伝播を無効または有効にするには

1. [システム] > [高可用性] に移動し、[ノード] タブでノードを開きます。
2. プライマリノードを選択または選択すると、構成がセカンダリオプションに反映されます。

> 注:
>
> HA 同期中は、コマンドの伝播が失敗する原因となる可能性のあるコマンド設定の競合を防ぐため、コマンド伝播は無効になっています。

VLAN への高可用性同期トラフィックの制限

December 8, 2023

高可用性 (HA) 展開では、HA 構成の維持に関連するトラフィックが 2 つの HA ノード間を流れます。このトラフィックには次のタイプがあります。

- 設定同期
- コンフィグ・プロパゲーション
- 接続ミラーリング
- 負荷分散永続性設定の同期
- 永続的なセッション同期
- セッション状態同期

この HA 関連トラフィックを 2 つのノード間で適切にフローさせることは、HA デプロイメントの機能にとって重要です。通常、HA 関連のトラフィックの量は少ないですが、フェイルオーバー中に非常に多くなることがあります。ステートフル接続フェイルオーバーが有効になっていて、フェイルオーバー前にプライマリだったノードが多数の接続を処理していた場合は、非常に高くなります。

デフォルトでは、HA 関連のトラフィックは NSIP アドレスがバインドされている VLAN を経由します。このトラフィックの急増に備えて、HA 関連のトラフィックを管理トラフィックから分離し、そのフローを別の VLAN に制限することができます。この VLAN は HA シンク VLAN と呼ばれます。

> 注:
>
> HA セットアップでは、`tag all` パラメータが有効になっている場合、HA パケットにも VLAN トラフィックのタグが付けられます。ただし、高可用性通信には問題がある可能性があります。そのため、HA トラフィックには NSVLAN または HA SYNC VLAN を設定することをお勧めします。

> - NSVLAN の設定については、「NSVLAN の設定」を参照してください。
> - HA シンク VLAN の設定については、「HA シンク VLAN の設定」を参照してください。

HA シンク VLAN を設定する前に考慮すべきポイント

- HA SYNC VLAN の設定は、伝播も同期もされません。つまり、HA SYNC VLAN はノード固有であり、各ノードで個別に設定されます。
- フルモードだけで設定をクリアすると、HA SYNC VLAN 設定は削除されます。
- 両方のノードがプライマリノードとして機能しないように、HA SYNC VLAN に含まれるインターフェイスでは HA MON を OFF に設定する必要があります。
- HA 関連のトラフィックが管理インターフェイスを経由しないように、管理インターフェイス（0/1 や 0/2 など）を HA SYNC VLAN の一部にしないでください。
- Citrix では、管理インターフェイスでは高可用性ハートビートメッセージを無効にし、HA SYNC VLAN インターフェイスでは有効にすることをお勧めします。これらの推奨事項を満たすと、データインターフェイスで高可用性ハートビートメッセージも有効にできます。

  インターフェイスでの高可用性ハートビートメッセージの無効化の詳細については、NetScaler アプライアンスでの高可用性ハートビートメッセージの管理を参照してください。

高可用性同期 VLAN の設定

NetScaler ノードで HA SYNC VLAN を構成するには、ローカルノードエンティティの HA SYNC VLAN パラメータを使用して、設定済みの VLAN を指定します。

コマンドラインを使用してローカルノードで **HA SYNC VLAN** を設定するには、次の手順を実行します。

コマンドプロンプトで入力します：

- `set ha node -syncvlan <VLANID>`
- `show node`

パラメータの説明:

syncvlan（同期 **VLAN**）：HA 関連のトラフィックが送信される VLAN。これには、同期、伝播、接続ミラーリング、ロードバランシングの永続性、設定の同期、永続セッション同期、およびセッション状態の同期のためのトラフィックが含まれます。ただし、HA ハートビートは任意のインターフェイスを使用できます。

GUI を使用してノードに **HA SYNC VLAN** を設定するには、次の手順を実行します。

1. ［**System**］ > ［**High Availability**］に移動します。
2. ローカルノードの変更中に **Sync VLAN** パラメータを設定します。

フェイルセーフモードの設定

December 8, 2023

HA 構成では、フェイルセーフモードにより、両方のノードがヘルスチェックに失敗した場合でも、一方のノードが常にプライマリになります。これは、ノードの一部しか使用できない場合に、トラフィックを可能な限り最適に処理できるバックアップ方法を有効にするためです。HA フェイルセーフモードは、プライマリノードとセカンダリノードで個別に設定されます。

次の表に、フェイルセーフのケースをいくつか示します。NOT_UP 状態は、ノードがヘルスチェックに失敗したものの、部分的に使用可能であることを意味します。UP 状態は、ノードがヘルスチェックに合格したことを意味します。

ノード A (プライマリ) のヘルス状態	ノード B (セカンダリ) のヘルス状態	デフォルトの HA 動作	フェイルセーフを有効にした HA の動作	説明
NOT_UP (前回失敗しました)	NOT_UP (最初に失敗した)	A (二次)、B (二次)	A (プライマリ)、B (セカンダリ)	両方のノードが次々と故障しても、最後のプライマリであったノードがプライマリのままです。
NOT_UP (最初に失敗した)	NOT_UP (前回失敗しました)	A (二次)、B (二次)	A (セカンダリ)、B (プライマリ)	両方のノードが次々と故障しても、最後のプライマリであったノードがプライマリのままです。
上へ	上へ	A (プライマリ)、B (セカンダリ)	A (プライマリ)、B (セカンダリ)	両方のノードがヘルスチェックに合格した場合、フェイルセーフを有効にした場合の動作は変更されません。
上へ	NOT_UP	A (プライマリ)、B (セカンダリ)	A (プライマリ)、B (セカンダリ)	セカンダリノードだけに障害が発生した場合、フェイルセーフを有効にした場合の動作は変更されません。

ノード A (プライマリ) のヘルス状態	ノード B (セカンダリ) のヘルス状態	デフォルトの HA 動作	フェイルセーフを有効にした HA の動作	説明
NOT_UP	上へ	A (セカンダリ)、B (プライマリ)	A (セカンダリ)、B (プライマリ)	プライマリだけに障害が発生した場合、フェイルセーフを有効にした場合の動作は変更されません。
NOT_UP	UP (STAYSECONDARY)	A (二次)、B (二次)	A (プライマリ)、B (セカンダリ)	セカンダリが STAYSECONDARY として設定されている場合、プライマリは障害が発生してもプライマリのままです。

コマンドラインインターフェイスを使用してフェールセーフモードを有効にするには

コマンドプロンプトで入力します:

```
set HA node [-failSafe ( **ON** | **OFF** )]
```

例

```
set ha node -failsafe ON
<!--NeedCopy-->
```

GUI を使用してフェールセーフモードを有効にするには

1. [システム] > [高可用性] に移動し、[ノード] タブでノードを開きます。
2. 「フェイルセーフモード」で、「両方のノードに異常があっても **1** つのプライマリノードを維持する」オプションを選択します。

仮想 **MAC** アドレスの構成

August 15, 2023

仮想 MAC アドレスは、HA セットアップのプライマリノードとセカンダリノードで共有されるフローティングエンティティです。

HA セットアップでは、プライマリノードは MIP、SNIP、VIP などのフローティング IP アドレスをすべて所有します。プライマリノードは、これらの IP アドレスに対するアドレス解決プロトコル (ARP) 要求に自身の MAC アドレスで応答します。その結果、外部デバイス（アップストリームルータなど）の ARP テーブルが Floating IP アドレスとプライマリノードの MAC アドレスで更新されます。

フェイルオーバーが発生すると、セカンダリノードが新しいプライマリノードとして引き継がれます。次に、Gratuitous ARP（GARP）を使用して、プライマリから取得したフローティング IP アドレスをアドバタイズします。ただし、新しいプライマリがアドバタイズする MAC アドレスは、自身のインターフェイスの MAC アドレスです。

一部のデバイス（特に一部のルーター）は、NetScaler アプライアンスによって生成された GARP メッセージを受け入れません。その結果、一部の外部デバイスでは、古いプライマリノードがアドバタイズしていた古い IP と MAC のマッピングが維持されます。これにより、サイトがダウンする可能性があります。

この問題は、HA ペアの両方のノードに仮想 MAC を設定することで解決できます。これにより、両方のノードに同じ MAC アドレスが割り当てられます。したがって、フェールオーバーが発生しても、セカンダリノードの MAC アドレスは変更されず、外部デバイスの ARP テーブルを更新する必要はありません。

仮想 MAC を作成するには、まず仮想ルータ ID (VRID) を作成し、それをインターフェイスにバインドする必要があります。(HA セットアップでは、VRID を両方のノードのインターフェイスにバインドする必要があります)。VRID がインターフェイスにバインドされると、システムは VRID を最後のオクテットとする仮想 MAC を生成します。

ここでは、次の詳細について説明します。

- IPv4 仮想 MAC の設定
- IPv6 仮想 MAC6 の設定

IPv4 仮想 MAC の設定

IPv4 仮想 MAC アドレスを作成してインターフェイスにバインドすると、インターフェイスから送信される IPv4 パケットはすべて、インターフェイスにバインドされた仮想 MAC アドレスを使用します。インターフェイスに IPv4 仮想 MAC がバインドされていない場合は、そのインターフェイスの物理 MAC アドレスが使用されます。

汎用仮想 MAC の形式は次のとおりです `00:00:5e:00:01:<VRID>`。たとえば、値が 60 の VRID を作成してインターフェイスにバインドした場合、生成される仮想 MAC は 00:00:5 e: 00:01:3 c になります。ここで、3c は VRID の 16 進表現です。1 ～255 の値で 255 個の VRIDs を作成できます。

IPv4 仮想 MAC の作成または変更

IPv4 仮想 MAC を作成するには、仮想ルーター ID を割り当てます。その後、仮想 MAC をインターフェイスにバインドできます。複数の VRID を同じインターフェイスにバインドすることはできません。仮想 MAC の設定を確認するには、仮想 MAC と仮想 MAC にバインドされているインターフェイスを表示して確認する必要があります。

コマンドラインインターフェイスを使用して仮想 **MAC** を追加するには　コマンドプロンプトで入力します。

- `add vrID`
- `bind vrid <id> -ifnum <interface_name>`
- `show vrID`

例

```
> add vrID 100
 Done
> bind vrid 100 -ifnum 1/1 1/2 1/3
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して仮想 **MAC** からインターフェイスをバインド解除するには　コマンドプロンプトで入力します。

- `unbind vrid <id> -ifnum <interface_name>`
- `show vrID`

GUI を使用して仮想 **MAC** を設定するには　[システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC**] タブで新しい仮想 MAC を追加するか、既存の仮想 MAC を編集します。

IPv4 仮想 **MAC** を削除する

IPv4 仮想 MAC を削除するには、その仮想ルーター ID を削除します。

コマンドラインインターフェイスを使用して **IPv4** 仮想 **MAC** を削除するには　コマンドプロンプトで入力します。

`rm vrid <id>`

例

```
rm vrid 100s
<!--NeedCopy-->
```

GUI を使用して **IPv4** 仮想 **MAC** を削除するには　[システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC**] タブで IPv4 仮想 MAC を削除します。

IPv6 仮想 MAC6 の設定

NetScaler は、IPv6 パケット用の仮想 MAC6 をサポートしています。IPv4 仮想 MAC がインターフェイスにバインドされている場合でも、任意のインターフェイスを仮想 MAC6 にバインドできます。インターフェイスから送信されるすべての IPv6 パケットは、そのインターフェイスにバインドされた仮想 MAC6 を使用します。インターフェイスにバインドされた仮想 MAC6 がない場合、IPv6 パケットは物理 MAC を使用します。

仮想 MAC6 の作成または変更

IPv6 仮想 MAC を作成するには、IPv6 仮想ルーター ID を割り当てます。その後、仮想 MAC をインターフェイスにバインドできます。1 つのインターフェイスに複数の IPv6 VRID をバインドすることはできません。仮想 MAC6 の設定を確認するには、仮想 MAC6 と仮想 MAC6 にバインドされたインターフェイスを表示して確認する必要があります。

コマンドラインインターフェイスを使用して仮想 **MAC6** を追加するには　コマンドプロンプトで入力します。

- `add vrID6 <id>`
- `bind vrID6 <id> -ifnum <interface_name>`
- `show vrID6`

例

```
> add vrID6 100
Done
> bind vrID6 100 -ifnum 1/1 1/2 1/3
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して仮想 **MAC6** からインターフェイスをバインド解除するには　コマンドプロンプトで入力します。

- `unbind vrID6 <id> -ifnum <interface_name>`
- `show vrID6`

GUI を使用して仮想 **MAC6** を設定するには　[システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC6]** タブで新しい仮想 **MAC6** を追加するか、既存の仮想 MAC6 を編集します。

仮想 MAC6 を削除する

IPv4 仮想 MAC を削除するには、その仮想ルーター ID を削除します。

コマンドラインインターフェイスを使用して仮想 **MAC6** を削除するには　コマンドプロンプトで入力します。

```
rm vrid6 <id>
```

例

```
1 rm vrid6 100s
2 <!--NeedCopy-->
```

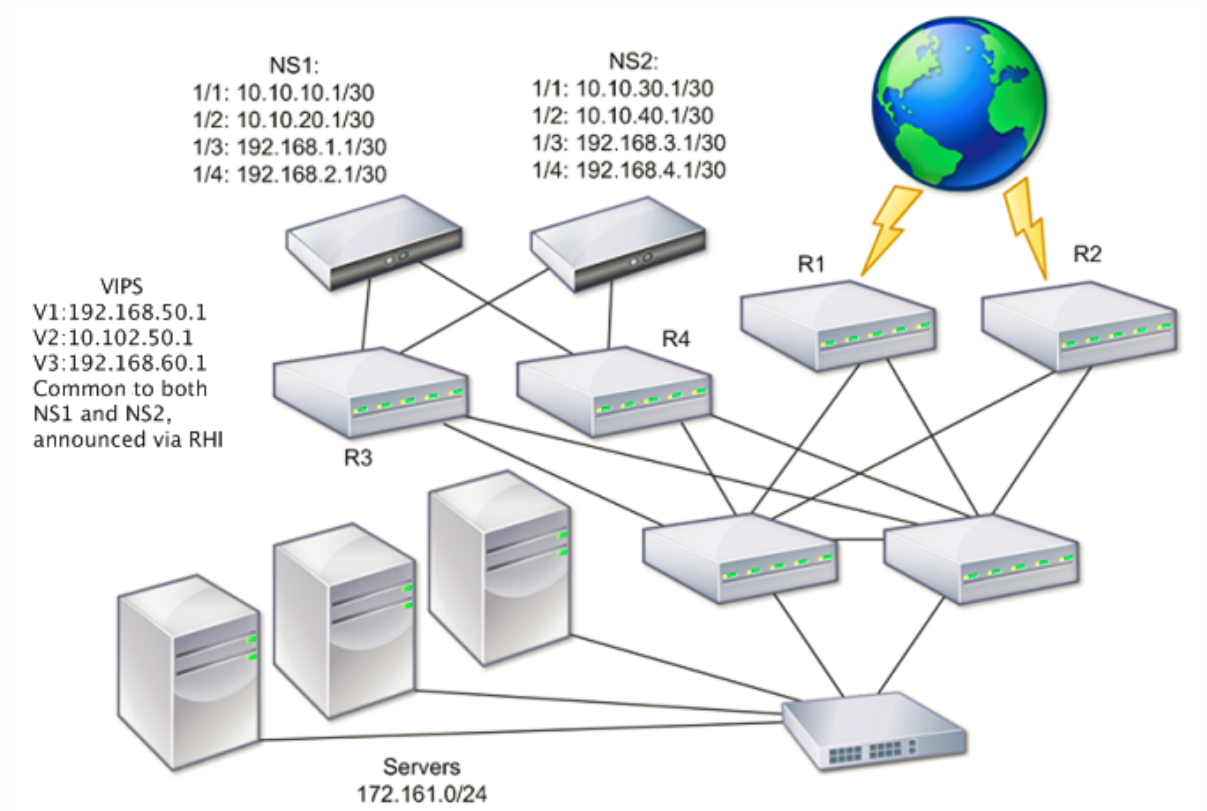
GUI を使用して仮想 **MAC6** を削除するには　[システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC6**] タブで仮想ルーター ID を削除します。

異なるサブネットに高可用性ノードを構成する

August 15, 2023

次の図は、2 つのシステムが異なるサブネットにある HA デプロイメントを示しています。

図 1: ルーテッドネットワークでの高可用性



この図では、システム NS1 と NS2 は、2 つの異なるサブネット上の 2 つの別々のルータ、R3 と R4 に接続されています。NetScaler アプライアンスは、ルーターを介してハートビートパケットを交換します。この構成は、任意の数

のインターフェイスを含む展開に対応するように拡張できます。

> 注記:
>
> ネットワークでスタティックルーティングを使用する場合は、ハートビートパケットが正常に送受信されるように、すべてのシステム間にスタティックルートを追加する必要があります。(システムでダイナミックルーティングを使用する場合、スタティックルートは不要です)。

HA ペアのノードが 2 つの別々のネットワーク上にある場合、プライマリノードとセカンダリノードには独立したネットワーク構成が必要です。つまり、異なるネットワーク上のノードは SNIP アドレス、VLAN、ルートなどのエンティティを共有できません。HA ペアのノードで設定可能なパラメータが異なるこのタイプの構成は、独立ネットワーク構成 (INC) または対称ネットワーク構成 (SNC) と呼ばれます。

次の表は、INC の設定可能なエンティティとオプションをまとめたもので、各ノードでどのように設定する必要があるかを示しています。

NetScaler エンティティ	オプション
IP (スニップ/スニップ)	ノード固有。そのノードでのみ有効です。
VIP	フローティング。
VLAN	ノード固有。そのノードでのみ有効です。
ルート	ノード固有。そのノードでのみ有効です。リンクロードバランシングルートはフローティングです。
ACL	フローティング (共通)。両方のノードでアクティブです。
動的ルーティング	ノード固有。そのノードでのみ有効です。また、セカンダリノードはルーティングプロトコルを実行し、アップストリームルーターとピアリングする必要があります。
L2 モード	フローティング (共通)。両方のノードでアクティブです。
L3 モード	フローティング (共通)。両方のノードでアクティブです。
リバース NAT (RNAT)	VIP アドレスがフローティング（共通）であるため、NAT IP アドレスを仮想サーバの IP アドレス（VIP）に設定した RNAT 設定。

同じサブネット内の HA ノードを構成する場合と同様に、異なるサブネットに HA ノードを構成するには、2 つの NetScaler ADC アプライアンスのそれぞれにログオンし、もう一方のアプライアンスを表すリモートノードを追加します。

リモートノードの追加

HA ペアの 2 つのノードが異なるサブネットにある場合、各ノードは異なるネットワーク構成を持つ必要があります。そのため、2 つの独立したシステムを HA ペアとして機能するように設定するには、設定プロセス中に INC モードを

指定する必要があります。

HA ノードを追加するときは、接続されていない、またはトラフィックに使用されていないインターフェイスごとに HA モニタを無効にする必要があります。CLI ユーザの場合、これは別の手順です。

コマンドラインインターフェイスを使用してノードを追加するには

コマンドプロンプトで入力します。

- `add ha node <id> <IPAddress> -inc ENABLED`
- `show ha node`

例

```
> add ha node 3 10.102.29.170 -inc ENABLED
 Done
> add ha node 3 1000:0000:0000:0000:0005:0600:700a:888b
  Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **HA** モニタを無効にするには

コマンドプロンプトで入力します。

- `set interface <ifNum> [-haMonitor ( **ON** | **OFF** )]`
- `show interface <ifNum>`

例

```
> set interface 1/3 -haMonitor OFF
 Done
<!--NeedCopy-->
```

GUI を使用してリモートノードを追加するには

1. [システム] > [高可用性] に移動し、[ノード] タブで新しいリモートノードを追加します。
2. 必ず、ダウンしているインターフェイス/チャネルでは HA モニタをオフにし、セルフモードオプションで INC（独立ネットワーク設定）モードをオンにしてください。

ノードを削除する

ノードを削除すると、そのノードは高可用性構成ではなくなります。

コマンドラインインターフェイスを使用してノードを削除するには

コマンドプロンプトで入力します。

```
rm ha node <id>
```

例

```
> rm ha node 2
  Done
<!--NeedCopy-->
```

GUI を使用してノードを削除するには

[システム] > [高可用性] に移動し、[ノード] タブでノードを削除します。

> 注記:
>
> ネットワークビジュアライザーを使用して、高可用性（HA）ペアとして構成されている NetScaler ADC アプライアンスを表示し、高可用性構成タスクを実行できます。

ルートモニタの設定

September 25, 2023

ルートモニターを使用すると、テーブルに動的に学習されたルートやスタティックなルートが含まれているかどうかにかかわらず、HA の状態を内部ルーティングテーブルに依存させることができます。HA 構成では、各ノードのルートモニターが内部ルーティングテーブルを監視して、特定のネットワークに到達するためのルートエントリが常に存在することを確認します。ルートエントリが存在しない場合、ルートモニタの状態は DOWN に変わります。

NetScaler アプライアンスにネットワークに到達するための静的ルートしかなく、ネットワークのルートモニターを作成する場合は、静的ルートの監視対象静的ルート（MSR）を有効にする必要があります。MSR は、到達不能なスタティックルートを内部ルーティングテーブルから削除します。スタティックルートで MSR を無効にすると、到達不能なスタティックルートが内部ルーティングテーブルに残り、ルートモニタの目的が果たせなくなる可能性があります。

ルートモニターは、非 INC モードと INC モードの両方でサポートされています。

非 INC モードの HA のルートモニター	INC モードの HA のルートモニター
ルートモニタはノードによって伝達され、同期中に交換されます。	ルートモニタは、ノードによって伝播されず、同期中に交換されることもありません。

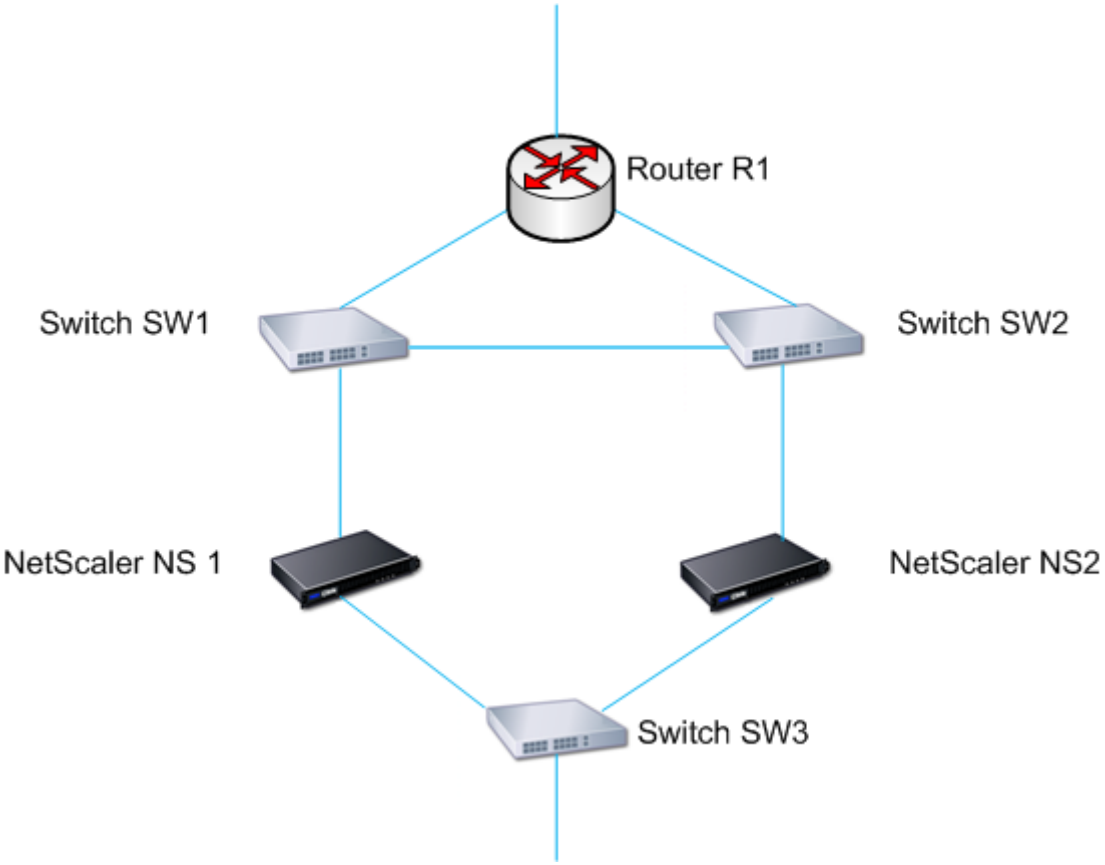
非 INC モードの HA のルートモニター	INC モードの HA のルートモニター
ルートモニタは、現在のプライマリノードでのみアクティブです。 NetScaler アプライアンスは、ルートエントリが内部ルーティングテーブルに存在するかどうかに関係なく、常にルートモニターの状態を UP と表示します。 リブート、フェールオーバー、v6 ルートの場合は set route6 コマンド、v4 ルートの set route msr enable/disable コマンド、新しいルートモニターの追加などの場合、ルートモニターは 180 秒後にルートのモニタリングを開始します（この処理には 180 秒かかる場合があります）。	ルートモニタは、プライマリノードとセカンダリノードの両方でアクティブです。 NetScaler ADC アプライアンスは、対応するルートエントリが内部ルーティングテーブルに存在しない場合、ルートモニターの状態を DOWN と表示します。 -

ルートモニターは、プライマリノードからゲートウェイにアクセスできないことを HA フェイルオーバーの条件の 1 つにしたい非 INC モードの HA 構成で役立ちます。

NetScaler アプライアンス NS1 と NS2 が同じサブネットにあり、ルーター R1 とスイッチ SW1、SW2、SW3 を備えたツーアームトポロジでの非 INC モードの HA セットアップの例を考えてみましょう。

この設定では R1 が唯一のルータなので、現在のプライマリノードから R1 にアクセスできないときはいつでも HA セットアップをフェールオーバーさせる必要があります。各ノードにルートモニタ（それぞれ RM1 と RM2）を設定して、そのノードからの R1 の到達可能性を監視できます。

図 1:



NS1 を現在のプライマリノードとして使用すると、実行フローは次のようになります。

1. NS1 上のルートモニタ RM1 は、NS1 の内部ルーティングテーブルを監視し、ルータ R1 のルートエントリの存在を確認します。NS1 と NS2 は、スイッチ SW1 または SW3 を介して定期的にハートビートメッセージを交換します。
2. スイッチ SW1 がダウンすると、NS1 のルーティングプロトコルは R1 に到達できないことを検出し、内部ルーティングテーブルから R1 のルートエントリを削除します。NS1 と NS2 は、スイッチ SW3 を介してハートビートメッセージを定期的に交換します。
3. R1 のルートエントリが内部ルーティングテーブルに存在しないことを検出すると、RM1 はフェールオーバーを開始します。NS1 と NS2 の両方から R1 へのルートがダウンしている場合、いずれかのアプライアンスが R1 に到達して接続を回復できるようになるまで、180 秒ごとにフェイルオーバーが発生します。

高可用性ノードへのルートモニターの追加

1 つのプロシージャでルートモニターが作成され、HA ノードにバインドされます。

> 注:
>
> 管理パーティションを設定している場合は、必ずデフォルトパーティションからルートモニターを追加してください。

コマンドラインインターフェイスを使用してルートモニターを追加するには

コマンドプロンプトで入力します。

- bind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])
- show HA node

例

```
> bind HA node 0 -routeMonitor 10.102.71.0 255.255.255.0
 Done
> bind HA node 0 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
 Done
<!--NeedCopy-->
```

GUI を使用してルートモニターを追加するには

[システム] > [高可用性] に移動し、[ルートモニター] タブで [設定] をクリックします。

ルートモニターの削除

コマンドラインインターフェイスを使用してルートモニターを削除するには

コマンドプロンプトで入力します。

- unbind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])
- show ha node

例

```
unbind HA node 3 -routeMonitor 10.102.71.0 255.255.255.0
unbind HA node 3 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
<!--NeedCopy-->
```

GUI を使用してルートモニターを削除するには

[システム] > [高可用性] に移動し、[ルートモニター] タブでルートモニターを削除します。

非 **INC** モードでのルートモニタによるフェイルオーバーの制限

August 15, 2023

非 INC モードの HA 構成では、両方のノードでルートモニターに障害が発生した場合、いずれかのノードがそれぞれのルートモニターで監視されているすべてのルートにアクセスできるようになるまで、180 秒ごとにフェイルオーバーが発生します。

ただし、ノードの場合は、ノードの [最大フリップ数] と [最大フリップ時間] パラメータを設定することで、特定の間隔のフェイルオーバー数を制限できます。いずれかの制限に達すると、それ以上フェイルオーバーは発生せず、そのノードでルートモニターに障害が発生しても、ノードはプライマリとして割り当てられます（ただし、ノードの状態は NOT UP）。プライマリとしての HA 状態と NOT UP としてのノード状態のこの組み合わせは、スティックプライマリ状態と呼ばれます。

その後、ノードが監視対象のすべてのルートに到達できた場合、次の監視障害により、ノードの [最大フリップ数] と [最大フリップ時間] パラメーターがリセットされ、[最大フリップ時間] パラメーターで指定された時間が開始されます。

これらのパラメータは各ノードで独立して設定されるため、伝播も同期もされません。

フェイルオーバーの数を制限するためのパラメータ

- 最大フリップ数 **(**最大フリップ**)**

  ルートモニターの障害が原因でフェイルオーバーが発生した場合、非 INC モードの HA ノードで、最大フリップ時間間隔内で許容されるフェイルオーバーの最大数。

- 最大フリップ時間 **(**最大フリップ時間**)**

  ルートモニターの障害によるフェイルオーバーが、非 INC モードの HA のノードで許可される時間（秒単位）。

コマンドラインインターフェイスを使用してフェイルオーバーの数を制限するには

コマンドプロンプトで入力します。

- `set HA node [-maxFlips < positive_integer>] [-maxFlipTime < positive_integer>]`
- `show HA node [< id>]`

GUI を使用してフェイルオーバーの数を制限するには

1. [システム] > [高可用性] に移動し、[ノード] タブでローカルノードを開きます。
2. 次のパラメーターを設定します。

- 最大フリップ数
- 最大フリップ時間

```
    > set ha node -maxFlips 30 -maxFlipTime 60
    Done
    > sh ha node
    1) Node ID: 0
    IP: 10.102.169.82 (NS)
    Node State: UP
    Master State: Primary
```



```
    Fail-Safe Mode: OFF
    INC State: DISABLED
    Sync State: ENABLED
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    Interfaces on which heartbeats are not seen :None
    Interfaces causing Partial Failure:None
    SSL Card Status: NOT PRESENT
    Hello Interval: 200 msecs
    Dead Interval: 3 secs
    Node in this Master State for: 0:4:24:1 (days:hrs:min:sec)

    2) Node ID: 1
    IP: 10.102.169.81
    Node State: UP
    Master State: Secondary
    Fail-Safe Mode: OFF
    INC State: DISABLED
    Sync State: SUCCESS
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    Interfaces on which heartbeats are not seen : None
    Interfaces causing Partial Failure: None
    SSL Card Status: NOT PRESENT

    Local node information:
    Configured/Completed Flips: 30/0
    Configured Flip Time: 60
    Critical Interfaces: 1/1

    Done
<!--NeedCopy-->
```

スティッキプライマリステートの **SNMP** アラーム

ノードがスティッキプライマリになったときにアラートを受け取りたい場合は、高可用性セットアップのノードで HA-STICKY-PRIMARY SNMP アラームを有効にします。ノードがスティッキプライマリになると、トラップメッセージ（stickyPrimary（1.3.6.1.4.1.5951.1.0.138））を生成して警告し、設定されたすべての SNMP トラップ宛先に送信します。SNMP アラームとトラップ先の構成の詳細については、「SNMPv1 および SNMPv2 トラップを生成するように NetScaler ADC を構成する」を参照してください。

よくある質問

2 つの NetScaler アプライアンス NS-1 と NS-2 を非 INC モードで高可用性セットアップした例を考えてみましょう。両方のノードの最大フリップ数と最大フリップ時間は同じ値で設定されています。

次の表に、この例で使用されている設定の一覧を示します。

エンティティ	詳細
NS-1 の IP アドレス	10.102.173.211
NS-2 の IP アドレス	10.102.173.212
最大フリップ数	2
最大フリップ時間	200

フリップの最大数と最大フリップ時間の設定については、PDF を参照してください。

フェールオーバーインターフェイスセットの設定

August 15, 2023

フェールオーバーインターフェイスセット (FIS) は、インターフェイスの論理的なグループです。HA 構成では、FIS を使用すると、1 つのインターフェイスに障害が発生しても、機能している他のインターフェイスが引き続き使用できるようにインターフェイスをグループ化することでフェールオーバーを防ぐことができます。FIS は、NetScaler ADC クラスターのノードにも構成できます。

FIS にバインドされていない HA MON インターフェイスは、いずれかのインターフェイスに障害が発生するとフェイルオーバーがトリガーされるため、クリティカルインターフェイス (CI) と呼ばれます。

> 注記:
>
> FIS では、アクティブ構成とスタンバイ構成は作成されません。また、同じ VLAN にリンクに接続するときのブリッジングループも防止されません。

FIS を作成または修正する

コマンドラインインターフェイスを使用して **FIS** を追加し、それにインターフェイスをバインドするには

コマンドプロンプトで入力します。

- `add fis <name>`

- bind fis \<name\> \<ifnum\> ...
- show fis \<name\>

例

```
> add fis fis1
 Done
> bind fis fis1 1/3 1/5
 Done
<!--NeedCopy-->
```

バインドされていないインターフェイスは、有効で HA MON がオンになっている場合、クリティカルインターフェイス（CI）になります。

コマンドラインインターフェイスを使用して **FIS** からインターフェイスをバインド解除するには

コマンドプロンプトで入力します。

- unbind fis \<name\> \<ifnum\> ...
- show fis \<name\>

例

```
> unbind fis fis1 1/3
 Done
<!--NeedCopy-->
```

GUI を使用して **FIS** を設定するには

[システム] > [高可用性] に移動し、[フェールオーバーインターフェイスセット] タブで新しい FIS を追加するか、既存の FIS を編集します。

FIS を削除する

FIS を削除すると、そのインターフェイスはクリティカルインターフェイスとしてマークされます。

コマンドラインインターフェイスを使用して **FIS** を削除するには

コマンドプロンプトで入力します。

```
rm fis <name>
```

例


```
> rm fis fis1
  Done
<!--NeedCopy-->
```

GUI を使用して **FIS** を削除するには

[システム] > [高可用性] に移動し、[フェールオーバーインターフェイスセット] タブで FIS を削除します。

フェイルオーバーの原因を理解する

August 15, 2023

次のイベントにより、ハイアベイラビリティ設定でフェールオーバーが発生する可能性があります。

1. セカンダリノードが、セカンダリに設定されたデッドインターバルを超える期間、プライマリからハートビートパケットを受信しない場合。(注 1 を参照)。
2. プライマリノードで SSL カードのハードウェア障害が発生します。
3. プライマリノードは、ネットワークインターフェイス上で 3 秒間ハートビートパケットを受信しません。
4. プライマリノードで、フェールオーバーインターフェイスセット（FIS）またはリンク集約（LA）チャネルの一部ではなく、HA モニタ（HAMON）が有効になっているネットワークインターフェイスで障害が発生します。(注 2 を参照)。
5. プライマリノードでは、FIS 内のすべてのインターフェイスで障害が発生します。(注 2 を参照)。
6. プライマリノードで、HAMON が有効になっている LA チャネルで障害が発生します。(注 2 を参照)。
7. プライマリノードでは、すべてのインターフェイスで障害が発生します（注 2 を参照)。この場合、フェールオーバーは HAMON の設定に関係なく発生します。
8. プライマリノードでは、すべてのインターフェイスが手動で無効になります。この場合、フェールオーバーは HAMON の設定に関係なく発生します。
9. いずれかのノードで force failover コマンドを発行して、フェールオーバーを強制します。
10. プライマリノードにバインドされているルートモニタがダウンします。

> 注 **1**:
>
> デッドインターバルの設定の詳細については、「通信間隔の設定」を参照してください。ノードがピアノードからハートビートパケットを受信しない原因としては、次のようなものがあります。
>
> - ネットワーク構成の問題により、ハートビートが HA ノード間でネットワークを通過するのを防ぎます。
> - ピアノードでハードウェアまたはソフトウェア障害が発生し、その原因でフリーズする（ハング)、リブートしたり、ハートビートパケットの処理や転送を停止したりします。
>
> 注 **2**:

> この場合、fail は、show interface コマンドまたは GUI からわかるように、インターフェイスが有効になっていて DOWN 状態になったことを意味します。有効なインターフェイスがダウン状態になる原因としては、LINK DOWN および TXSTAL があります。

ノードを強制的にフェイルオーバーさせる

August 15, 2023

たとえば、プライマリノードを交換またはアップグレードする必要がある場合は、フェールオーバーを強制的に実行できます。プライマリノードまたはセカンダリノードのいずれかから強制的にフェールオーバーできます。強制フェールオーバーは継承されたり、同期されたりしません。強制フェイルオーバー後の同期ステータスを表示するには、ノードのステータスを表示できます。

次の状況では、強制フェールオーバーを実行できません。

- スタンドアロンシステムにフェールオーバーを強制する。
- セカンダリノードは無効です。
- セカンダリノードはセカンダリのままになるように設定されます。

強制フェイルオーバーコマンドを実行したときに潜在的な問題が検出されると、NetScaler アプライアンスは警告メッセージを表示します。メッセージには警告の要因に関する情報が含まれており、手順を進める前に確認が求められます。

プライマリノード、セカンダリノード、およびノードがリッスンモードのときにフェールオーバーを強制できます。

- プライマリノードでのフェールオーバーの強制実行。

  プライマリノードでフェールオーバーを強制すると、プライマリがセカンダリになり、セカンダリがプライマリになります。強制フェールオーバーは、プライマリノードがセカンダリノードが稼働中であると判断できる場合にのみ可能です。

  セカンダリノードがダウンしている場合、force failover コマンドは次のエラーメッセージを返します。「ピアの状態が無効なため、操作できません。修正して再試行してください。」

  セカンダリシステムが要求状態または非アクティブの場合、次のエラーメッセージが表示されます。

  ```
  Operation not possible now. Please wait for the system to stabilize
   before retrying.
  ```

- セカンダリ・ノードでのフェイルオーバーの強制実行。

  セカンダリノードから force failover コマンドを実行すると、セカンダリノードはプライマリノードになり、プライマリノードはセカンダリノードになります。強制フェールオーバーは、セカンダリノードの正常性が良好で、セカンダリとして構成されていない場合にのみ発生します。

セカンダリノードがプライマリノードになれない場合、またはセカンダリノードが（STAYSECERNDARY オプションを使用して）セカンダリとして構成されている場合、ノードは次のエラーメッセージが表示されます。

```
Operation not possible as my state is invalid. View the node for
more information.
```

- ノードがリッスンモードである場合のフェールオーバーの強制実行。

  HA ペアの 2 つのノードが異なるバージョンのシステムソフトウェアを実行している場合、上位バージョンを実行しているノードがリッスンモードに切り替わります。このモードでは、コマンドの伝播も同期も機能しません。

  両方のノードでシステムソフトウェアをアップグレードする前に、いずれかのノードで新しいバージョンをテストします。これを行うには、すでにアップグレードされているシステムでフェイルオーバーを強制する必要があります。アップグレードされたシステムはプライマリノードとして引き継がれますが、コマンドの伝播や同期は行われません。また、すべての接続を再確立する必要があります。

> 重要:
>
> HA 同期処理の進行中にフェールオーバーを強制すると、HA セットアップのアクティブなデータセッションの一部が失われる可能性があります。したがって、HA 同期処理が完了するのを待ってから、強制フェールオーバー操作を実行します。

コマンドラインインターフェイスを使用してノードでフェイルオーバーを強制するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
force HA failover
```

GUI を使用してノードでフェイルオーバーを強制するには、次の手順を実行します。

[システム] > [高可用性] に移動し、[ノード] タブでノードを選択し、[アクション] リストで [強制フェイルオーバー] を選択します。

セカンダリノードを強制的にセカンダリに維持する

August 15, 2023

HA セットアップでは、プライマリノードの状態に関係なく、セカンダリノードをセカンダリのまま強制的に維持できます。

たとえば、プライマリノードをアップグレードする必要があり、処理に数秒かかるとします。アップグレード中、プライマリノードが数秒間停止することがありますが、セカンダリノードに引き継がれることは望ましくありません。プライマリノードで障害が検出された場合でも、セカンダリノードのままにしておきたいものです。

セカンダリノードを強制的にセカンダリのまま維持すると、プライマリノードがダウンしても、セカンダリのまま維持されます。HA ペアの一方のノードのステータスをセカンダリのまま強制的に維持すると、そのノードは、HA 状態マシン遷移には参加しません。ノードのステータスは、STAYSECONDARY として表示されます。

ノードをセカンダリのまま強制的に維持する方法は、スタンドアロンノードとセカンダリノードのどちらでも機能します。スタンドアロンノードでは、ノードを追加して HA ペアを作成する前に、このオプションを使用する必要があります。新しいノードを追加すると、既存のノードはトラフィックの処理を停止し、セカンダリノードになります。新しいノードがプライマリノードになります。

> 注記:
>
> システムをセカンダリのまま強制的に維持する場合、その強制を実施するプロセスは、伝播も同期もされません。コマンドを実行するノードのみが対象となります。

コマンドラインインターフェイスを使用してセカンダリノードを強制的にセカンダリのままにするには

コマンドプロンプトで入力します。

```
set ha node -hastatus STAYSECONDARY
```

GUI を使用してセカンダリノードを強制的にセカンダリのままにするには

[システム] > [高可用性] に移動し、[ノード] タブで、ローカルノードを開き、[**STAY SECONDARY**] を選択します。

プライマリノードを強制的にプライマリのままにする

August 15, 2023

HA セットアップでは、フェールオーバー後も正常なプライマリノードを強制的にプライマリに保つことができます。このオプションは、HA ペアのプライマリノードのいずれかで有効にできます。このオプションを使用すると、プライマリノードが正常である限り、プライマリノードがプライマリ状態になります。

スタンドアロンノードでは、ノードを追加して HA ペアを作成する前に、このオプションを使用する必要があります。新しいノードを追加しても、既存のノードがプライマリノードとして機能し続け、新しいノードがセカンダリノードになります。

コマンドラインインターフェイスを使用してプライマリノードを強制的にプライマリノードのままにするには

コマンドプロンプトで入力します。

```
set ha node -hastatus STAYPRIMARY
```


GUI を使用してプライマリノードを強制的にプライマリのままにするには

[システム] > [高可用性] に移動し、[ノード] タブで、ローカルノードを開き、[**STAY PRIMARY**] を選択します。

高可用性に関するよくある質問

December 8, 2023

1. HA 構成のノード間で HA 関連情報を交換するために使用するさまざまなポートは何ですか。

   HA 構成では、両方のノードが次のポートを使用して HA 関連情報を交換します:

   - ハートビートパケットを交換するための UDP ポート 3003。
   - TCP ポート 3008 または 3010 (同期およびコマンド伝播用)。

2. 同期をトリガーする条件は何か?

   同期は、次のいずれかの条件によってトリガーされます。

   - セカンダリが受信したプライマリノードのインカネーション番号が、セカンダリノードのインカネーション番号と一致しません。
     注:HA 構成内の両方のノードには、
     ノードの構成ファイル内の構成数をカウントするインカーネーション番号と呼ばれるカウンタがあります。各ノードは、ハートビートメッセージでそのインカネーション番号を他のノードに送信します。次のコマンドでは、インカーネーション番号は増加しません:
     a) すべての HA 設定関連コマンド。たとえば、ha ノードを追加し、ha ノードを設定し、ha ノードをバインドします。
     b) すべてのインターフェイス関連コマンド。たとえば、インターフェイスを設定したり、インターフェイスを設定解除したりします。
     c) チャンネル関連のすべてのコマンド。たとえば、チャネルの追加、チャネルの設定、チャネルのバインドなどを行います。
   - セカンダリノードは再起動後に起動します。
   - プライマリノードは、フェールオーバー後にセカンダリになります。

3. INC モードまたは非 INC モードの HA 構成で同期または伝播されない構成はどれですか?

   次のコマンドは、セカンダリノードに伝播も同期もされません:

   - すべてのノード固有の HA 構成コマンド。たとえば、ha ノードを追加し、ha ノードを設定し、ha ノードをバインドします。
   - インターフェイス関連のすべての構成コマンド。たとえば、インターフェイスを設定したり、インターフェイスを設定解除したりします。

- チャンネル関連のすべての設定コマンド。たとえば、チャネルの追加、チャネルの設定、チャネルのバインドなどを行います。

> 注:
>
> 以下の構成は、INC モードの HA でのみ同期も伝達もされません。各ノードにはそれぞれ独自のものがあります。
>
> - SNIP
> - VLAN
> - ルート (LLB ルートを除く)
> - ルートモニター
> - RNAT ルール (NAT IP として VIP を使用するすべての RNAT ルールを除く)
> - ダイナミックルーティング構成
> - ネットプロファイル

4. セカンダリノードに追加された構成はプライマリノードで同期されますか?

   いいえ、セカンダリノードに追加された構成はプライマリノードと同期されません。

5. HA 構成で両方のノードがプライマリであると主張する理由は何でしょうか。

   最も可能性の高い理由は、プライマリノードとセカンダリノードは両方とも正常ですが、セカンダリはプライマリからハートビートパケットを受信しないことです。問題は、ノード間のネットワークにある可能性があります。

6. 異なるシステムクロック設定で 2 つのノードを展開すると、HA 構成で問題が発生しますか?

   2 つのノードでシステムクロック設定が異なると、次の問題が発生する可能性があります。

   - ログファイルエントリのタイムスタンプが一致しません。このような状況では、ログエントリに問題がないか分析するのが難しくなります。
   - フェイルオーバー後、どのような種類の Cookie ベースのパーシスタンスでも、ロードバランシングの問題が発生する可能性があります。時間が大きく異なると、Cookie が予想よりも早く期限切れになり、永続化セッションが終了する可能性があります。
   - ノードでの時間関連の決定にも同様の考慮事項が当てはまります。

7. *Force HA sync* コマンドが失敗した場合の条件は何ですか。

   強制同期は、次のいずれかの状況で失敗します。

   - 同期が既に進行中の場合は、強制的に同期します。
   - スタンドアロンの NetScaler アプライアンスで同期を強制します。
   - セカンダリノードは無効です。
   - HA 同期は現在のセカンダリノードでは無効になっています。
   - 現在のプライマリノードでは HA 伝播が無効になっているため、プライマリノードからの同期を強制します。

8. *sync HA files* コマンドの失敗条件は何ですか。

   設定ファイルの同期は、次のいずれかの状況で失敗します。

   - スタンドアロンシステム上。
   - セカンダリノードを無効にした状態。

9. HA 構成では、セカンダリノードがプライマリを引き継いだ場合、元のプライマリがオンラインに戻ったときにセカンダリステータスに戻りますか。

   いいえ。セカンダリノードがプライマリを引き継いだ後は、元のプライマリノードが再びオンラインに戻っても、プライマリノードはプライマリのままです。ノードのプライマリステータスとセカンダリステータスを交換するには、*force failover* コマンドを実行します。

10. force failover コマンドが失敗する条件を教えてください。

    次の状況では、強制フェールオーバーを実行できません。

    - スタンドアロンシステムにフェールオーバーを強制する。
    - セカンダリノードは無効です。
    - セカンダリノードはセカンダリのままになるように設定されます。
    - プライマリノードはプライマリのままであるように構成されています。
    - ピアノードの状態が不明です。

高可用性問題のトラブルシューティング

August 15, 2023

高可用性に関する最も一般的な問題は、高可用性機能がまったく機能しない、または断続的にしか機能しないことです。高可用性に関する一般的な問題と、考えられる原因と解決策を以下に示します。

- 問題点高可用性セットアップでは
  、NetScaler アプライアンスが NetScaler アプライアンスとペアリングできない。

  - 原因

    ネットワーク接続

    解像度

    両方のアプライアンスがスイッチに接続され、インターフェイスが有効になっていることを確認します。

  - 原因

    デフォルトの管理者アカウントのパスワードが一致しない

    解像度

    両方のアプライアンスのパスワードが同じであることを確認します。

- 原因

  IP の競合

  解像度

  両方のアプライアンスに固有の NetScaler IP（NSIP）アドレスが割り当てられていることを確認します。アプライアンスの NSIP アドレスは同じであってはなりません。

- 原因

  ノード ID の不一致

  解像度

  両方のアプライアンスのノード ID 構成が固有であることを確認します。アプライアンスのノード ID 設定は同じであってはなりません。さらに、ノード ID には 1 ～64 の値を割り当てる必要があります。

- 原因

  RPC ノードのパスワードの不一致

  解像度

  両方のノードの RPC ノードパスワードが同じであることを確認します。

- 原因

  管理者がリモートノードを無効にしました

  解像度

  リモートノードを有効にします。

- 原因

  ファイアウォールアプリケーションがハートビートパケットをブロックしました

  解像度

UDP ポート 3003 が許可されていることを確認します。

- 問題両方のアプライアンスがプライマリアプライアンスであると主張しています

  。

  - 原因

    アプライアンス間でハートビートパケットが見つからない

    解像度

    UDP ポート 3003 がアプライアンス間の通信でブロックされていないことを確認します。

- 問題

  ： NetScaler アプライアンスは構成を同期できません。

  - 原因

    ファイアウォールアプリケーションが、必要なポートをブロックしています。

    解像度

    UDP ポート 3010（またはセキュア同期の場合は UDP ポート 3008）がアプライアンス間の通信でブロックされていないことを確認します。

- 原因

  管理者が同期を無効にしました。

  解像度

  問題のあるアプライアンスで同期を有効にします。

- 原因

  アプライアンスには、さまざまな NetScaler リリースまたはビルドがインストールされています。

  解像度

  アプライアンスを同じ NetScaler リリースまたはビルドにアップグレードします。

- 問題点

  : アプライアンス間でコマンドの伝播が失敗する。

  - 原因

    ファイアウォールアプリケーションがポートをブロックしています。

    解像度

    UDP ポート 3011（または安全な伝播が可能な UDP ポート 3009）がアプライアンス間の通信でブロックされていないことを確認します。

  - 原因

    管理者がコマンド伝達を無効にしました。

    解像度

    問題のあるアプライアンスでコマンドの伝播を有効にします。

  - 原因

    アプライアンスには、さまざまな NetScaler リリースまたはビルドがインストールされています。

    解像度

    アプライアンスを同じ NetScaler リリースまたはビルドにアップグレードします。

- 問題高可用性ペアの **NetScaler** アプライアンスは

  、強制フェイルオーバープロセスを実行できません。

  - 原因

    セカンダリノードは無効です。

    解像度

    2 次ノードを有効にします。

  - 原因

    セカンダリノードはセカンダリのままであるように構成されています。

    解像度

    セカンダリノードのセカンダリ高可用性ステータスを Stay Secondary から Enable に設定します。

- 問題フェイルオーバー処理後

  、セカンダリアプライアンスはトラフィックを受信しません。

- 原因

  アップストリームルーターは NetScaler アプライアンスの GARP メッセージを認識しません。

  解像度

  セカンダリアプライアンスで仮想 MAC アドレスを設定します。

NetScaler ADC アプライアンスでの高可用性ハートビートメッセージの管理

August 15, 2023

高可用性構成の 2 つのノードは、有効になっているすべてのインターフェイスで相互にハートビートメッセージを送受信します。ハートビートメッセージは、これらのインターフェイスの HA MON 設定に関係なく流れます。NSVLAN またはその両方（NSVLAN と SYNC）がアプライアンスに設定されている場合、ハートビートメッセージは、NSVLAN および SYNCVLAN の一部である有効なインターフェイスを介してのみ流れます。

ノードが有効なインターフェイスでハートビートメッセージを受信しない場合、指定された SNMP マネージャに重大なアラートが送信されます。これらの重大なアラートは、ピアノードへの接続の一部として構成されていないインターフェイスについて、誤警報を発し、管理者から不必要な注意を引きます。

この問題を解決するために、インターフェイスとチャネルの HAHeartbeat オプションを使用して、それらの HA ハートビートメッセージフローを有効または無効にします。

コマンドラインインターフェイスを使用してインターフェイス上の高可用性ハートビートメッセージを管理するには

コマンドプロンプトで入力します。

- `set interface <ID> [-HAHeartBeat ( ON | OFF )]`
- `show interface <ID>`

コマンドラインインターフェイスを使用してチャネル上の高可用性ハートビートメッセージを管理するには

コマンドプロンプトで入力します。

- `set channel <ID> [-HAHeartBeat ( ON | OFF )]`
- `show channel <ID>`

GUI を使用してインターフェイスの高可用性ハートビートメッセージを管理するには

1. **[**システム**]** > **[**ネットワーク**]** > **[**インターフェイス**]** に移動します。
2. **HA** ハートビート パラメータを有効または無効にします。

GUI を使用してチャネル上の高可用性ハートビートメッセージを管理するには

1. [システム] > [ネットワーク] > [チャネル] に移動します。
2. **HA** ハートビート パラメータを有効または無効にします。

高可用性セットアップでの **NetScaler ADC** の削除と交換

August 15, 2023

このトピックは、RMA の代替品に取り組むのに役立ちます。また、このトピックでは、構成のバックアップ方法、出荷されているソフトウェアバージョンのアップグレードまたはダウングレード、ADC での RPC パスワードの設定方法についても説明します。

考慮すべきポイント

次の構成は、INC（独立ネットワーク構成）または非 INC モードの高可用性構成では同期も伝達もされません。

- すべてのノード固有の HA 構成コマンド。たとえば、ha ノードを追加し、ha ノードを設定し、ha ノードをバインドします。
- インターフェイス関連のすべての構成コマンド。たとえば、インターフェイスを設定したり、インターフェイスを設定解除したりします。
- チャンネル関連のすべての設定コマンド。たとえば、チャネルの追加、チャネルの設定、チャネルのバインドなどを行います。
- すべてのインターフェイス HA モニタリング設定コマンド。

次の構成は、INC モード (独立ネットワーク構成) の HA 構成では同期も伝達もされません。

- SNIP
- VLAN
- ルート (LLB ルートを除く)
- ルートモニター
- RNAT ルール (NAT IP として VIP を使用するすべての RNAT ルールを除く)
- ダイナミックルーティング構成

手順

高可用性セットアップの NetScaler を交換するには、次の手順を実行してください。

- アクティブな NetScaler セカンダリノードを削除する
- 交換用セカンダリノードの設定
- 交換用 ADC のソフトウェアビルドの検証と更新
- 新しいセカンダリのパスワードをプライマリと同じに設定
- 交換用 ADC へのライセンスの追加
- プライマリノードと新しいセカンダリノード間の HA ペアの作成

アクティブなセカンダリノードを削除する

1. 両方の ADC にログオンし、次のコマンドを実行して、どのノードがプライマリでどのノードがセカンダリかを確認します。

```
show ha node
<!--NeedCopy-->
```

2. プライマリ ADC にログオンし、プライマリノードの設定をバックアップし、変更前にファイルを ADC からコピーします。これらのファイルは「/var /ns_sys_backup/」ディレクトリにあります。

   手順は次のとおりです。

   a) ADC の実行構成をメモリに保存します。

```
save ns config
<!--NeedCopy-->
```

   b) フルバックアップファイルパッケージの作成:

```
create system backup -level full
<!--NeedCopy-->
```

   c) 基本的なバックアップファイルパッケージを作成します。

```
create system backup -level basic
<!--NeedCopy-->
```

3. すべてのバックアップファイルが生成されたら、次に進む前に必ずそれらをデバイスからコピーしてください。

   Windows ターミナルから、コマンドプロンプトを開き、バックアップファイルを ADC からローカルハードドライブにコピーします。これは次のコマンドを使用して実行できます。

```
pscp <username>@<NSIP>:<Target file source>  <Target file
   destination>
<!--NeedCopy-->
```

   例:

```
pscp nsroot@10.125.245.78:/var/ns_sys_backup/backup_basic_10
   .125.245.78_2016_09_14_15_08.tgz c:\nsbackup\backup_basic_10
   .125.245.78_2016_09_14_15_08.tgz
<!--NeedCopy-->
```

   プロンプトが表示されたら、指定した管理者アカウントのパスワードを入力し、Enter キーを押します。先に進む前に、すべてのバックアップバンドルがローカル PC にコピーされるまで、これらの手順を繰り返します。

4. セカンダリ ADC に SSH 接続し、ユニットを「STAYSECONDARY」ステータスに設定します。これにより、スワップ中に障害が検出されても、ユニットはプライマリーロールを引き継ごうとしなくなります。この手順を実行する前に、セカンダリ ADC に接続されていることを確認してください。


```
set ha node  -haStatus <state>
set ha node  -haStatus STAYSECONDARY
<!--NeedCopy-->
```

5. セカンダリ **ADC** のノードステートに **STAYSECONDARY** が正常に表示されたら、プライマリ ADC に切り替えてセカンダリノードを削除し、次のコマンドを実行します。

```
save ns config
<!--NeedCopy-->
```

プライマリ ADC にログインした状態で、次のコマンドを実行します。

a) 次のコマンドを実行して、どの数値がセカンダリ HA ノードを表すかを確認します。

```
show ha node
<!--NeedCopy-->
```

b) 次のコマンドを実行して、セカンダリ ADC をプライマリ HA ペアから削除します。

```
rm ha node <node ID>
<!--NeedCopy-->
```

c) 次のコマンドを実行して設定を保存します。

```
save ns config
<!--NeedCopy-->
```

d) セカンダリ ADC を取り外したら、セカンダリ ADC をシャットダウンして切断し、ネットワークから取り外します。

注記。切断する前に、必ずすべての接続にラベルを付けてください。

交換用セカンダリノードの設定

1. 交換用 ADC を取り付けたら、新しいデバイスの電源を入れます。この時点では、ネットワーク接続を接続しないでください。
2. 起動が完了したら、コンソールポートを使用して ADC に接続し、ユニットへの接続に使用する NSIP を設定します。
3. プロンプトが表示されたら、**4** を選択します。

   注。この例では、交換用の ADC に別の NSIP を使用しています。元のセカンダリ・ユニットの IP を使用する場合は、新しい ADC をプライマリ HA ユニットにバインドする前に、交換時に IP を変更できます。
4. これで ADC が起動するはずです。次に、管理トラフィックに使用するネットワークインターフェイスを接続し、ネットワークから IP アドレスにアクセスできることを確認します。

交換用 **ADC** のソフトウェアビルドの検証と更新

新しいユニットをプライマリ ADC に同期する前に、両方の ADC が同じビルドを実行していることを確認する必要があります。

1. ADC でバージョンを確認するには、次のコマンドを実行します。

```
show version
<!--NeedCopy-->
```

2. 新しいセカンダリ ADC で、アップグレードに使用するサブフォルダを **/var** に作成します。
3. NetScaler のダウンロードに移動し 、プライマリ ADC で実行されているビルドバージョンと一致する適切なパッケージをダウンロードします。
4. .tgz ファイルをダウンロードし、展開します。

```
tar -xvzf "file.tgz"
<!--NeedCopy-->
```

5. 抽出したファイルをセカンダリ ADC にコピーします。Windows ターミナルで「コマンドプロンプト」を開き、抽出された.tgz ビルドパッケージを含むディレクトリに移動し、次の pscp コマンドを実行します。

```
pscp <Target file source>  <username>@<NSIP>:<Target file
   destination>
<!--NeedCopy-->
```

例:

```
C:\inetpub>pscp c:\inetpub\build-12.1-47.14_nc.tgz nsroot@10
   .20.245.80:/var/NS_upg_12.1_47.14/build-12.1-47.14_nc.tgz
<!--NeedCopy-->
```

6. ファイルが転送されたら、セカンダリ ADC に戻り、アップグレードします。詳細な手順については、「Citrix ADX スタンドアロンアプライアンスのアップグレード」を参照してください。
7. 新しいセカンダリがリブートしたら、SSH でユニットに戻り、アップグレードが成功し、ビルドがプライマリのビルドと一致することを確認します。

交換用セカンダリノードのパスワードをプライマリノードと一致するように設定

注: この時点で新しいセカンダリ ADC の管理 IP (NSIP) アドレスを変更したい場合は、先に進む前に変更してください。

新しいセカンダリ ADC のパスワードを、現在プライマリ ADC に設定されているパスワードと一致するように変更します。

1. デフォルトの管理者 (nsroot) アカウントのパスワードがプライマリ ADC と同じであることを確認します。これは、SSH 経由で新しいセカンダリユニットにログインした状態で次のコマンドを使用して実行されます。


```
set system user <user> <password>
<!--NeedCopy-->
```

このコマンドは、指定されたユーザーのパスワードを設定/リセットします。

2. プライマリ ADC と新しいセカンダリ ADC に SSH 接続し、パスワードが一致することを確認します。

交換用セカンダリノードへのライセンスの追加

新しい ADC が更新され、ペアリングの準備ができたら、交換用ノード用の適切なライセンスをダウンロードしてインストールします。

1. https://www.citrix.com に移動して、新しい交換ユニットのライセンスをリクエストおよびダウンロードしてください。
2. 適切なライセンスをすべてダウンロードしたら、新しいセカンダリ ADC に SSH 接続し、次のコマンドを入力して現在のライセンスの状態を確認します。

```
show license
<!--NeedCopy-->
```

3. Windows ターミナルのコマンドプロンプトから、次のコマンドを使用してライセンスファイルを新しいセカンダリ ADC にアップロードする必要があります。

   注記。複数のライセンスをお持ちの場合は、すべてのライセンスがアップロードされるまでこの手順を繰り返します。

```
pscp <Target file source>  <username>@<NSIP>:<Target file
   destination>
<!--NeedCopy-->
```

   例:

```
C:\inetpub>pscp c:\inetpub\NS-VPX-3K-LIC-020030ad0024.lic
   nsroot@10.125.245.80:/nsconfig/license/NS-VPX-3K-LIC-020030
   ad0024.lic
<!--NeedCopy-->
```

4. 新しいセカンダリ ADC に SSH 接続し、次のコマンドを使用してウォームリブートを実行します。

```
reboot  -w
<!--NeedCopy-->
```

   ユニットが再起動したら、ユニットに SSH 接続して show license コマンドをもう一度実行します。この時点で、ライセンスを適用する必要があります。

プライマリノードと新しいセカンダリノード間の高可用性の設定

この時点で、NetScaler ADC ユニットを高可用性ペアに加入する準備が整いました。詳細については、「高可用性の構成」を参照してください。

再試行を要求する

August 15, 2023

NetScaler アプライアンスが HTTP リクエストを受信したが、バックエンドサーバーとの接続に失敗した場合、アプライアンスは再試行ディレクティブを使用します。リクエストの再試行は、接続障害のシナリオに対処し、アプライアンスが次に使用可能なサービスを選択してリクエストを転送できるようにします。リクエストの再試行を行うことで、クライアントはラウンドトリップ時間 (RTT) を節約できます。

再試行要求機能は、次の接続障害シナリオに適用できます。

- HTTP リクエストの受信時にバックエンドサーバーが TCP 接続をリセットした場合。詳細については、「再試行のリクエスト」を参照してください。
- 接続の確立中にバックエンドサーバーが TCP 接続をリセットした場合。詳細については、「再試行のリクエスト」を参照してください。
- アプライアンスが HTTP リクエストを送信したときに、バックエンドサーバーからの応答が（設定されたタイムアウト値に基づいて）タイムアウトした場合。詳細については、「再試行のリクエスト」を参照してください。

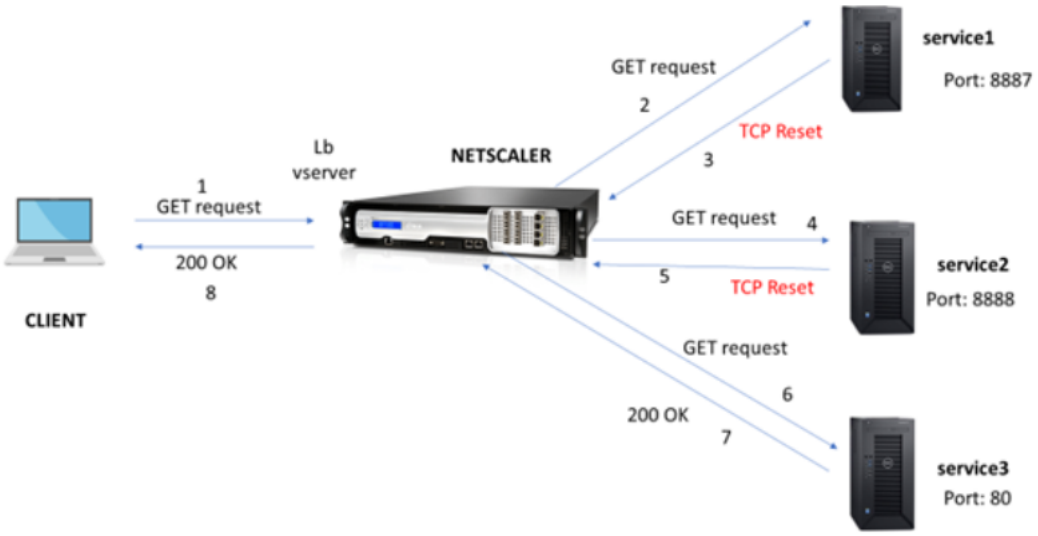
バックエンドサーバーが **TCP** 接続をリセットした場合に再試行をリクエストする

January 11, 2024

バックエンドサーバーが TCP 接続をリセットすると、要求の再試行機能は、リセットをクライアントに送信する代わりに、次に使用可能なサーバーに要求を転送します。リロードバランシングを行うことで、アプライアンスが次に使用可能なサービスへの同じリクエストを開始したときに、クライアントは RTT を節約できます。

バックエンドサーバーが **TCP** 接続をリセットしたときの要求再試行の仕組み

次の図は、コンポーネントが互いにどのように相互作用するかを示しています。



1. このプロセスでは、まずアプライアンスの appqoe 機能を有効にします。
2. クライアントが HTTP または HTTPS 要求を送信すると、負荷分散仮想サーバーはその要求をバックエンドサーバーに送信します。
3. 要求されたサービスが利用できない場合、バックエンドサーバーは TCP 接続をリセットします。
4. appqoe 設定で「再試行」が有効になっていて、必要な再試行回数が指定されている場合、負荷分散仮想サーバーは、設定された負荷分散アルゴリズムを使用して、次に利用可能なアプリケーションサーバーに要求を転送します。
5. 負荷分散仮想サーバーが応答を受信した後、アプライアンスは応答をクライアントに転送します。
6. 使用可能なバックエンドサーバーが再試行回数と同じかそれ以下で、すべてのサーバーがリセットを送信した場合、アプライアンスは 500 の内部サーバーエラーを返します。使用可能なサーバーが 5 つあり、再試行回数が 6 に設定されているシナリオを考えてみます。5 台のサーバーすべてが接続をリセットすると、アプライアンスはクライアントに 500 内部サーバーエラーを返します。
7. 同様に、バックエンドサーバーの数が再試行回数よりも多く、バックエンドサーバーが接続をリセットした場合、アプライアンスはリセットをクライアントに転送します。3 つのバックエンドサーバーがあり、再試行回数が 2 に設定されているシナリオを考えてみます。3 台のサーバーが接続をリセットすると、アプライアンスはリセット応答をクライアントに送信します。

GET メソッドの要求再試行を構成します

GET メソッドのリトライ機能を設定するには、次の手順を完了する必要があります。

1. AppQoE を有効にする
2. AppQoE アクションの追加
3. AppQoE ポリシーの追加
4. AppQoE ポリシーを負荷分散仮想サーバーにバインドする

AppQoE を有効にする　コマンドプロンプトで入力します:

```
enable ns feature appqoe
```

AppQoE アクションの追加　AppQoE アクションを設定して、TCP リセット後にアプライアンスを再試行するかどうか、および再試行回数を指定する必要があります。

```
add appqoe action reset_action -retryOnReset ( YES | NO )-numretries
<positive_integer>]
```

例:

```
add appqoe action reset_action -retryOnReset YES -numretries 5
```

ここで、
リセット時に再試行してください。バックエンドサーバーが TCP 接続をリセットした場合は、再試行を有効にします。
数字。再試行回数。

AppQoE ポリシーの追加　AppQoE を実装するには、特定のキューで受信する HTTP または SSL リクエストに優先順位を付けるように AppQoE ポリシーを設定する必要があります。

コマンドプロンプトで入力します:

```
add appqoe policy <name> -rule <expression> -action <string>
```

例:

```
add appqoe policy reset_policy -rule http.req.method.eq(get)-action
reset_action
```

appqoe ポリシーを負荷分散仮想サーバーにバインドする　バックエンドサーバーが TCP パケット要求をリセットし、負荷分散仮想サーバーがその要求を次に利用可能なサービスに転送するようにする場合は、負荷分散仮想サーバーを AppQoE ポリシーにバインドする必要があります。

コマンドプロンプトで入力します:

```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-
priority <positive_integer>] [-gotoPriorityExpression <expression
>] [-type ( REQUEST | RESPONSE )]
```

例:

```
bind lb vserver v1 -policyName reset_policy -type REQUEST -priority 1
```


POST リクエストのリクエストリトライの設定

バックエンドサーバーにデータを書き込むバランスリクエストをリロードする場合は、常に注意が必要です。このようなリクエストには、コンテンツの長さが短いことを確認してください。コンテンツの長さが長いと、リソースを消費する可能性があります。POST リクエストのリ負荷分散を設定するには、以下の手順に従ってください。

1. AppQoE を有効にする
2. AppQoE アクションの追加
3. AppQoE ポリシーの追加
4. AppQoE ポリシーを負荷分散仮想サーバーにバインドする

AppQoE を有効にする　コマンドプロンプトで入力します:

```
enable ns feature appqoe
```

Apple アクションを追加　TCP リセットと再試行回数の後に再試行するには、AppQoE アクションを追加する必要があります。

```
add appqoe action reset_action -retryOnReset ( YES | NO )-numretries
<positive_integer>]
```

例**:**

```
add appqoe action reset_action -retryOnReset YES -numretries 5
```

アップルポリシーの追加　AppQoE を実装するには、AppQoE ポリシーを設定して、特定のキュー内の接続をキューに入れる方法を定義する必要があります。

コマンドプロンプトで入力します:

```
add appqoe policy <name> -rule <expression> -action <string>
```

例**:**

```
add appqoe policy reset_policy -rule HTTP.REQ.CONTENT_LENGTH.le(2000)
-action reset_action
```

> 注:
>
> この構成は、リクエストのリトライ機能をコンテンツの長さが 2000 未満に制限したい場合に使用できます。

負荷分散仮想サーバーを **AppQoE** ポリシーにバインドする　バックエンドサーバーが TCP パケット要求をリセットしたときに、負荷分散仮想サーバーが特定のキューを介して次の利用可能なサービスに要求を転送するようにするには、負荷分散仮想サーバーを AppQoE ポリシーにバインドする必要があります。

コマンドプロンプトで入力します:


```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-
priority <positive_integer>] [-gotoPriorityExpression <expression
>] [-type ( REQUEST | RESPONSE )]
```

例:

```
bind lb vserver v1 -policyName reset_policy -type REQUEST -priority 1
```

NetScaler GUI を使用してリクエストリトライ用の **AppQoE** ポリシーを構成する

1. **[**AppExpert] > [**AppQoE****] > [ポリシー] に移動します。**
2. 「**AppQoE** ポリシー」ページで、「追加」をクリックします。
3. 「**AppQoE** ポリシーの作成」ページで、
   次のパラメータを設定します。a. 名前。AppQoE ポリシー名
   b. アクション。アクションを追加または編集します。アクションを作成するには、。
   c. 式。`HTTP.REQ.CONTENT_LENGTH.le (2000)` ポリシー表現を選択または入力します。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

NetScaler GUI を使用してリクエストリトライバランシング用の **AppQoE** アクションを構成する

1. **[**AppExpert] > [**AppQoE****] > [アクション] に移動します。**
2. 「**AppQoE** アクション」ページで、「追加」をクリックします。
3. 「**AppQoE** アクションの作成」ページで、TCP リセット時に再試行するための次のパラメータを設定します。
   a. TCP リセットで再試行してください。このチェックボックスを選択すると、TCP リセットのリトライアクションが有効になります。
   b. 再試行回数。再試行回数を入力します。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

TCP SYN の確立時にバックエンドサーバーがリセットされたときに、**GET** メソッドの要求の再試行を構成します

CLI と GUI の構成は、GET メソッドの場合と同様の手順です。詳細については、「GET メソッドのリクエスト試行を設定する」セクションを参照してください。バックエンドサーバーが接続セクションをリセットしたとき。

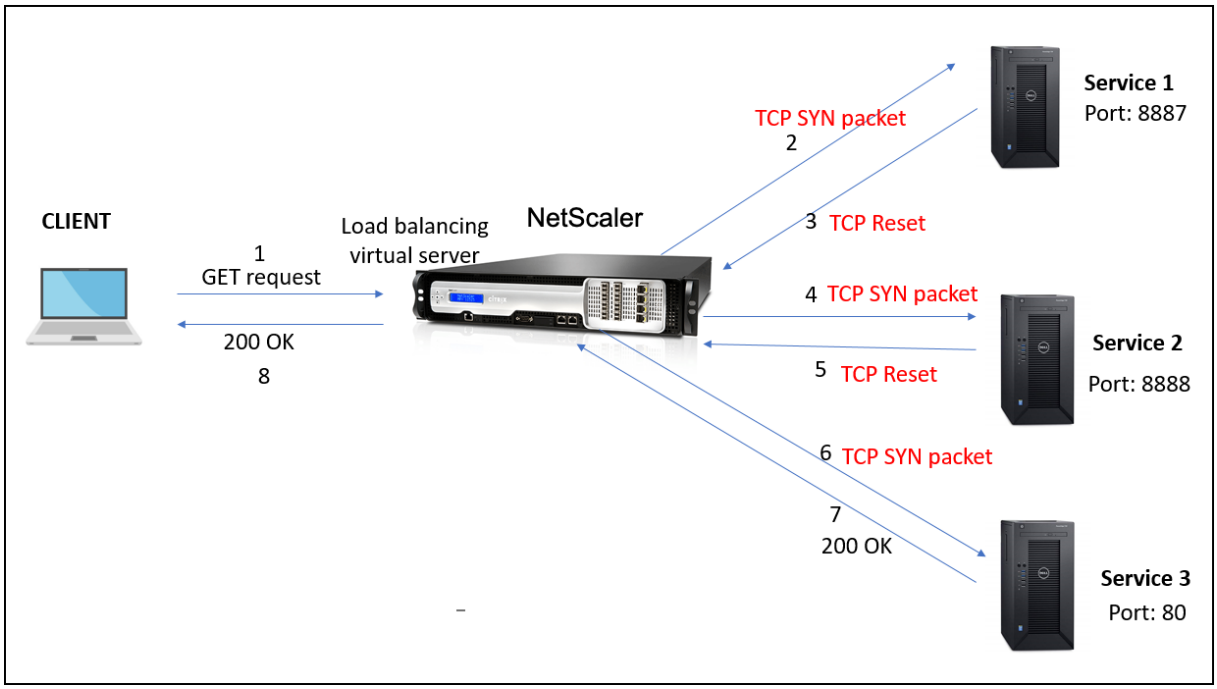
接続確立中にバックエンドサーバーが **TCP** 接続をリセットした場合に再試行を要求する

August 15, 2023

接続確立中にバックエンドサーバーが TCP 接続をリセットすると、要求再試行機能により、リセットをクライアントに送信する代わりに、次に利用可能なサーバーに要求が転送されます。リロードバランシングを行うことで、アプライアンスが次に使用可能なサービスへの同じリクエストを開始したときに、クライアントは RTT を節約できます。

SYN 確立時にバックエンドサーバーが **TCP** 接続をリセットしたときの要求再試行の仕組み

次の図は、コンポーネントが互いに相互作用することを示しています。



1. このプロセスでは、まずアプライアンスの appqoe 機能を有効にします。
2. クライアントが HTTP または HTTPS リクエストを送信すると、負荷分散仮想サーバーはバックエンドサーバーへの接続を開始します。
3. TCP SYN 確立時に要求されたサービスが利用できない場合、バックエンドサーバーは TCP 接続をリセットします。
4. appqoe 設定で「再試行」が有効になっていて、必要な再試行回数が指定されている場合、負荷分散仮想サーバーは、設定された負荷分散アルゴリズムを使用して、次に利用可能なアプリケーションサーバーに要求を転送します。
5. 負荷分散仮想サーバーが応答を受信した後、アプライアンスは応答をクライアントに転送します。
6. 使用可能なバックエンドサーバーが再試行回数と同じかそれ以下で、すべてのサーバーがリセットを送信した場合、アプライアンスは 500 の内部サーバーエラーを返します。使用可能なサーバーが 5 つあり、再試行回数が 6 に設定されているシナリオを考えてみます。5 台のサーバーすべてが接続をリセットすると、アプライアンスはクライアントに 500 内部サーバーエラーを返します。
7. 同様に、バックエンドサーバーの数が再試行回数を超え、バックエンドサーバーが TCP SYN 確立時に接続をリセットした場合、アプライアンスはリセットをクライアントに転送します。3 つのバックエンドサーバーが

あり、再試行回数が 2 に設定されているシナリオを考えてみます。3 台のサーバーが接続をリセットすると、アプライアンスはクライアントにリセットパケットを送信します。

TCP SYN 確立時にバックエンドサーバーがリセットされるときのリクエスト再試行 **(GET** メソッドと **POST** メソッド**)** を設定します

CLI および GUI の構成は、GET および POST メソッドの場合と同様の手順です。詳細については、「GET メソッドのリクエスト再試行の構成」の「バックエンドサーバーが接続セクションをリセットするときの POST メソッドのリクエスト再試行を構成する」を参照してください。tyuu

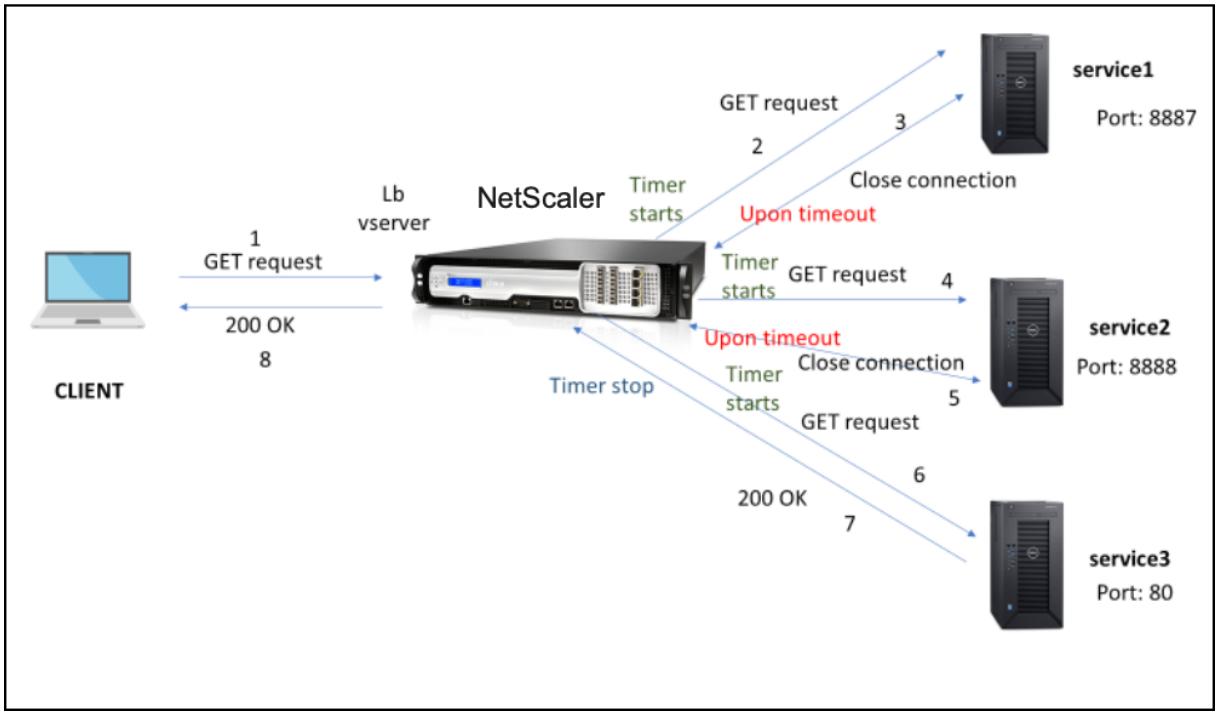
バックエンドサーバーの応答がタイムアウトになったら再試行をリクエストする

January 11, 2024

リクエスト再試行は、バックエンドサーバーがリクエストへの応答に時間がかかる場合、アプライアンスがタイムアウト時にリロードバランシングを実行し、そのリクエストを次の使用可能なサーバーに転送するもう 1 つのシナリオで使用できます。

バックエンドサーバーの応答がタイムアウトしたときのリクエストリトライの仕組み

次の図は、コンポーネントが互いに相互作用することを示しています。



1. このプロセスでは、まずアプライアンスの appqoe 機能を有効にします。
2. appqoe 設定には、ミリ秒単位の「retryOnTimeout」パラメータがあります。
3. アプライアンスが要求を送信し、サーバーの応答に時間がかかる場合、アプライアンスは設定されたタイムアウト値に基づいて再負荷分散を実行します。アプライアンスは、サーバーの応答を待つ代わりに、接続をリセットし、別のサービスを選択して要求を転送します。
4. 負荷分散仮想サーバーが応答を受信した後、アプライアンスは応答をクライアントに転送します。タイムアウトパラメータを使用すると、アプライアンスがサーバーの応答を待つ必要がなくなり、RTT が増加します。
5. 使用可能なバックエンドサーバーが再試行回数と同じかそれ以下で、すべてのサーバーが要求に対してタイムアウトした場合、アプライアンスは 500 の内部サーバーエラーを返します。使用可能なサーバーが 5 つあり、再試行回数が 6 に設定されているシナリオを考えてみます。5 台のサーバーすべてが要求に対してタイムアウトした場合、アプライアンスはクライアントに 500 内部サーバーエラーを返します。
6. 同様に、バックエンドサーバーの数がリトライ回数よりも多く、バックエンドサーバーがリクエストでタイムアウトした場合、アプライアンスはサーバーが応答を送信するか、クライアントのアイドル接続がタイムアウトするまで、最後のサービスを待機し続けます。3 つのバックエンドサーバーがあり、再試行回数が 2 に設定されているシナリオを考えてみます。要求に応じて 3 台のサーバーすべてがタイムアウトした場合、アプライアンスは、サーバーが応答を送信するか、クライアントのアイドル接続がタイムアウトするまで、3 番目のサービスを待機し続けます。

バックエンドサーバーの応答がタイムアウトしたときの要求再試行 **(GET** および **POST** メソッド**)** の設定

タイムアウト時に GET メソッドのリクエスト再試行を設定するには、次の手順を完了する必要があります。

1. アプリを有効にする
2. AppQoE アクションの設定
3. 適用ポリシーの追加
4. appqoe ポリシー を負荷分散仮想サーバーにバインドする

> 注:
>
> タイムアウト時のリクエスト再試行シナリオは、POST メソッドにも適用できます。

アプリを有効にする

コマンドプロンプトで入力します:

```
enable ns feature appqoe
```

タイムアウト用の **appqoe** アクションを追加

タイムアウト時に再試行するように appqoe アクションを設定し、再試行回数を定義する必要があります。

コマンドプロンプトで入力します:


```
add appqoe action <name> -retryOnTimeout <msecs> -numRetries <positive_integer
>
```

例:

```
add appqoe action appact1 -retryOnTimeout 35 -numRetries 5
```

適用ポリシーの追加

appqoe を実装するには、appqoe ポリシーを設定して、接続をキューに入れる方法を定義する必要があります。

コマンドプロンプトで入力します:

```
add appqoe policy <name> -rule <rule> -action <name>
```

例:

```
add appqoe policy timeout_policy -rule http.req.method.eq(get)-action
 appact1
```

appqoe ポリシーを負荷分散仮想サーバーにバインドする

バックエンドサーバーの応答に時間がかかり、負荷分散仮想サーバーから次の利用可能なサービスに要求を転送したい場合は、appqoe ポリシーを分散仮想サーバーにバインドする必要があります。

コマンドプロンプトで入力します:

```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-
priority <positive_integer>] [-gotoPriorityExpression <expression
>] [-type ( REQUEST | RESPONSE )]
```

例:

```
bind lb vserver v1 -policyName timeout_policy -type REQUEST -priority
 1
```

NetScaler GUI を使用してタイムアウト時に再負荷分散を行うための **AppQoE** ポリシーを構成する

1. [**AppExpert] > [AppQoE] [**ポリシー] に移動します。
2. 「**AppQoE** ポリシー」ページで、「追加」をクリックします。
3. [**AppQoE** ポリシー の作成] ページで、次のパラメーターを設定します。

   a. 名前。AppQoE ポリシー名
   b. アクション。アクションを追加または編集します。新しいアクションを作成するには、「AppQoE アクションの作成」セクションを参照してください。
   c. 式。「http.req.method.eq (get)」ポリシーエクスプレッションを選択または入力します。

4. **[**作成**]** して **[**閉じる**]** をクリックします。

NetScaler GUI を使用してリクエスト再試行用の **AppQoE** アクションを構成する

1. [**AppExpert] > [AppQoE] [**アクション] に移動します。
2.「**AppQoE** アクション」ページで、「追加」をクリックします。
3.「**AppQoE** アクションの作成」ページで、バックエンドサーバーの応答タイムアウト時に再試行する次のパラメータを設定します。a.
   タイムアウト時に再試行してください。バックエンドサーバーにリクエストを送信したときに、リクエストのタイムアウト (ミリ秒単位) で再試行します。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

TCP の最適化

August 15, 2023

TCP は、データ伝送におけるネットワークの輻輳を回避するために、以下の最適化手法と輻輳制御戦略（またはアルゴリズム）を使用します。

渋滞制御戦略

TCP は、インターネット接続の確立と管理、転送エラーの処理、および Web アプリケーションとクライアントデバイスの円滑な接続に長い間使用されてきました。しかし、パケットロスはネットワークの輻輳だけに依存するわけではなく、輻輳が必ずしもパケットロスを引き起こすわけではないため、ネットワークトラフィックの制御はより困難になっています。したがって、輻輳を測定するには、TCP アルゴリズムはパケット損失と帯域幅の両方に焦点を当てる必要があります。

比例レート回復 **(PRR)** アルゴリズム

TCP 高速回復メカニズムは、パケット損失によるウェブの遅延を低減します。新しい比例レート回復 (PRR) アルゴリズムは、損失回復中に TCP データを評価する高速回復アルゴリズムです。輻輳制御アルゴリズムで選択したターゲットウィンドウに適した割合を使用して、Rate-Halving を模したパターンになっています。これにより、ウィンドウの調整が最小限に抑えられ、リカバリ終了時の実際のウィンドウサイズは Slow-Start のしきい値 (ssthresh) に近い値になります。

TCP ファストオープン **(TFO)**

TCP Fast Open（TFO）は、TCP の最初のハンドシェイク中にクライアントとサーバー間で迅速かつ安全なデータ交換を可能にする TCP メカニズムです。この機能は、NetScaler アプライアンスの仮想サーバーにバインドされた TCP プロファイルの TCP オプションとして使用できます。TFO は、NetScaler アプライアンスが生成する TCP Fast Open Cookie（セキュリティクッキー）を使用して、仮想サーバーへの TFO 接続を開始するクライアントを検証および認証します。この TFO メカニズムを使用すると、1 回のフルラウンドトリップに必要な時間だけアプリケーションのネットワーク遅延を減らすことができます。これにより、短い TCP 転送で発生する遅延が大幅に減少します。

TFO の仕組み

クライアントが TFO 接続を確立しようとすると、最初の SYN セグメントに TCP Fast Open Cookie が含まれ、それ自体が認証されます。認証が成功すると、NetScaler アプライアンス上の仮想サーバーは、スリーウェイハンドシェイクの最後の ACK セグメントを受信していなくても、SYN-ACK セグメントにデータを含めることができます。これにより、データを交換する前に三者間のハンドシェイクを必要とする通常の TCP 接続と比較して、最大 1 回の往復を節約できます。

クライアントとバックエンドサーバーは、次の手順を実行して TFO 接続を確立し、最初の TCP ハンドシェイク中にデータを安全に交換します。

1. クライアント自体を認証するための TCP ファストオープンクッキーがない場合、クライアントは SYN パケットでファストオープンクッキーリクエストを NetScaler アプライアンス上の仮想サーバーに送信します。
2. 仮想サーバにバインドされた TCP プロファイルで TFO オプションが有効になっている場合、アプライアンスは（クライアントの IP アドレスを秘密鍵で暗号化することにより）Cookie を生成し、生成された Fast Open Cookie を TCP オプションフィールドに含む SYN-ACK でクライアントに応答します。
3. クライアントは、アプライアンス上の同じ仮想サーバーへの今後の TFO 接続に備えて Cookie をキャッシュします。
4. クライアントが同じ仮想サーバーへの TFO 接続を確立しようとすると、キャッシュされた Fast Open Cookie（TCP オプションとして）を含む SYN を HTTP データとともに送信します。
5. NetScaler アプライアンスは Cookie を検証し、認証が成功すると、サーバーは SYN パケット内のデータを受け入れ、SYN-ACK、TFO クッキー、HTTP レスポンスでイベントを確認します。

> 注:
>
> クライアント認証が失敗した場合、サーバーはデータをドロップし、セッションタイムアウトを示す SYN のみでイベントを確認します。

1. サーバー側では、サービスにバインドされた TCP プロファイルで TFO オプションが有効になっている場合、NetScaler アプライアンスは、接続しようとしているサービスに TCP Fast Open Cookie が存在するかどうかを判断します。
2. TCP Fast Open Cookie が存在しない場合、アプライアンスは SYN パケットで Cookie リクエストを送信します。

3. バックエンドサーバーが Cookie を送信すると、アプライアンスはその Cookie をサーバー情報キャッシュに保存します。
4. アプライアンスに特定の宛先 IP ペアの Cookie が既にある場合は、古い Cookie が新しい Cookie に置き換えられます。
5. 仮想サーバーが同じ SNIP アドレスを使用して同じバックエンドサーバーに再接続しようとしたときに Cookie がサーバー情報キャッシュに存在する場合、アプライアンスは SYN パケット内のデータを Cookie と結合し、バックエンドサーバーに送信します。
6. バックエンドサーバーは、データと SYN の両方を使用してイベントを確認します。

注：サーバーが SYN セグメントのみでイベントを確認した場合、NetScaler アプライアンスは元のパケットから SYN セグメントと TCP オプションを削除した直後にデータパケットを再送信します。

TCP ファストオープンの設定

TCP Fast Open（TFO）機能を使用するには、関連する TCP プロファイルで TCP Fast Open オプションを有効にし、TFO Cookie Timeout パラメータをそのプロファイルのセキュリティ要件に適した値に設定します。

CLI を使用して **TFO** を有効または無効にする　コマンドプロンプトで、次のコマンドのいずれかを入力して、新規または既存のプロファイルの TFO を有効または無効にします。

注**:** デフォルト値は DISABLED です。

```
1     add tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
2     set tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
3     unset tcpprofile <TCP Profile Name> - tcpFastOpen
4     Examples
5     add tcpprofile Profile1 - tcpFastOpen
6     Set tcpprofile Profile1 - tcpFastOpen Enabled
7     unset tcpprofile Profile1 - tcpFastOpen
8 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **TCP Fast Open Cookie** のタイムアウト値を設定するには

コマンドプロンプトで入力します。

```
1     set tcpparam - tcpfastOpenCookieTimeout <Timeout Value>
2     Example
3     set tcpprofile - tcpfastOpenCookieTimeout 30secs
4 <!--NeedCopy-->
```

GUI を使用して **TCP** ファストオープンを設定するには

1. [構成] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。

2.「**TCP** プロファイルの設定」ページで、「**TCPFast Open** 」チェックボックスを選択します。
3. [**OK]**、[完了**]** の順にクリックします。

GUI を使用して **TCP** ファストクッキーのタイムアウト値を設定するには

[構成] > [システム] > [設定**] > [TCP** パラメータの変更] に移動し、次に [**TCP** パラメータの設定] ページに移動して TCP Fast Open Cookie のタイムアウト値を設定します。

TCP ハイスタート

新しい TCP プロファイルパラメーター HyStart により、HyStart アルゴリズムが有効になります。HyStart アルゴリズムは、終了する安全なポイント (ssthresh) を動的に決定するスロースタートアルゴリズムです。これにより、大量のパケットロスを発生させることなく、輻輳回避への移行が可能になります。この新しいパラメータはデフォルトでは無効になっています。

混雑が検出されると、HyStart は輻輳回避フェーズに入ります。これを有効にすると、パケット損失の多い高速ネットワークでのスループットが向上します。このアルゴリズムは、トランザクションの処理中に最大帯域幅に近い状態を維持するのに役立ちます。そのため、スループットを向上させることができます。

TCP ハイスタートの設定

HyStart 機能を使用するには、関連する TCP プロファイルで Cubic HyStart オプションを有効にします。

コマンドラインインターフェイス **(CLI)** を使用して **HyStart** を設定するには

コマンドプロンプトで、次のコマンドのいずれかを入力して、新規または既存の TCP プロファイルで HyStart を有効または無効にします。

```
add tcpprofile <profileName> -hystart ENABLED
set tcpprofile <profileName> -hystart ENABLED
unset tcprofile <profileName> -hystart
<!--NeedCopy-->
```

例:

```
    add tcpprofile profile1 -hystart ENABLED
    set tcpprofile profile1 -hystart ENABLED
    unset tcprofile profile1 -hystart
<!--NeedCopy-->
```

GUI を使用して HyStart サポートを設定するには

1. [構成] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。

2.「**TCP** プロファイルの設定」ページで、「**Cubic Hystart** 」チェックボックスを選択します。
3. [**OK]**、[完了**]** の順にクリックします。

TCP バーストレート制御

TCP 制御メカニズムにより、高速モバイルネットワークでバーストトラフィックフローが発生し、ネットワーク全体の効率に悪影響を与える可能性があることが確認されています。データの輻輳やレイヤ 2 再送信などのモバイルネットワークの状況により、TCP 確認応答がまとまって送信者に届き、大量の送信がトリガーされます。これらの連続したパケットのグループが、短いパケット間ギャップで送信されます。これを TCP パケットバーストと呼びます。トラフィックの急増を防ぐため、NetScaler アプライアンスは TCP バーストレート制御技術を使用しています。この手法では、データがバーストに送信されないように、ラウンドトリップ時間全体にわたってデータをネットワークに均等に配置します。このバーストレート制御技術を使用すると、スループットを向上させ、パケットドロップ率を下げることができます。

TCP バーストレート制御の仕組み

NetScaler アプライアンスでは、この手法により、ラウンドトリップ時間（RTT）の全期間にわたってパケットの送信が均等に分散されます。これは、TCP スタックとネットワークパケットスケジューラを使用してさまざまなネットワーク条件を識別し、進行中の TCP セッションのパケットを出力してバーストを減らすことで実現されます。

送信側では、確認応答を受信したらすぐにパケットを送信する代わりに、パケットの送信を遅延させて、スケジューラ（動的構成）または TCP プロファイル（固定構成）で定義されたレートでパケットを分散させることができます。

TCP バーストレート制御の設定

関連する TCP プロファイルの TCP バーストレート制御オプションを使用して、バーストレート制御パラメータを設定します。

コマンドラインを使用して **TCP** バーストレート制御を設定するには

コマンドプロンプトで、次のいずれか 1 つの TCP Burst Rate Control コマンドを新規または既存のプロファイルで設定します。

注**:** デフォルト値は DABLED です。

```
1 add tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic
      | Fixed
2
3 set tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic
      | Fixed
4
```



```
unset tcpprofile <TCP Profile Name> -burstRateControl Disabled |
    Dynamic | Fixed
<!--NeedCopy-->
```

各項目の意味は次のとおりです。

無効—バーストレート制御が無効になっている場合、NetScaler アプライアンスは MaxBurst 設定以外のバースト管理を実行しません。

固定—TCP バーストレート制御が Fixed の場合、アプライアンスは TCP プロファイルに記載されている TCP 接続ペイロード送信レート値を使用します。

ダイナミック—バーストレート制御が「ダイナミック」の場合、TCP バーストを減らすために、さまざまなネットワーク条件に基づいて接続が調整されます。このモードは、TCP 接続が ENDPOINT モードの場合にのみ機能します。ダイナミックバーストレート制御が有効な場合、TCP プロファイルの maxBurst パラメータは有効になりません。

```
add tcpProfile  profile1 -burstRateControl Disabled

set tcpProfile profile1 -burstRateControl Dynamic

unset tcpProfile profile1 -burstRateControl Fixed
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **TCP** バーストレート制御パラメータを設定するには　コマンドプロンプトで入力します。

```
    set ns tcpprofile nstcp_default_profile  - burstRateControl <type of
        burst rate control>  - tcprate <TCP rate> -rateqmax <maximum
       bytes in queue>

    T1300-10-2> show ns tcpprofile nstcp_default_profile
            Name: nstcp_default_profile
            Window Scaling status:  ENABLED
            Window Scaling factor: 8
            SACK status: ENABLED
            MSS: 1460
            MaxBurst setting: 30 MSS
            Initial cwnd setting: 16 MSS
            TCP Delayed-ACK Timer: 100 millisec
            Nagle's Algorithm: DISABLED
            Maximum out-of-order packets to queue: 15000
            Immediate ACK on PUSH packet: ENABLED
            Maximum packets per MSS: 0
            Maximum packets per retransmission: 1
            TCP minimum RTO in millisec: 1000
            TCP Slow start increment: 1
            TCP Buffer Size: 8000000 bytes
            TCP Send Buffer Size: 8000000 bytes
            TCP Syncookie: ENABLED
            Update Last activity on KA Probes: ENABLED
            TCP flavor: BIC
```



```
24                TCP Dynamic Receive Buffering: DISABLED
25                Keep-alive probes: ENABLED
26                Connection idle time before starting keep-alive probes: 900
                      seconds
27                Keep-alive probe interval: 75 seconds
28                Maximum keep-alive probes to be missed before dropping
                     connection: 3
29                Establishing Client Connection: AUTOMATIC
30                TCP Segmentation Offload: AUTOMATIC
31                TCP Timestamp Option: DISABLED
32                RST window attenuation (spoof protection): ENABLED
33                Accept RST with last acknowledged sequence number: ENABLED
34                SYN spoof protection: ENABLED
35                TCP Explicit Congestion Notification: DISABLED
36                Multipath TCP: DISABLED
37                Multipath TCP drop data on pre-established subflow:
                     DISABLED
38                Multipath TCP fastopen: DISABLED
39                Multipath TCP session timeout: 0 seconds
40                DSACK: ENABLED
41                ACK Aggregation: DISABLED
42                FRTO: ENABLED
43                TCP Max CWND : 4000000 bytes
44                FACK: ENABLED
45                TCP Optimization mode: ENDPOINT
46                TCP Fastopen: DISABLED
47                HYSTART: DISABLED
48                TCP dupack threshold: 3
49                Burst Rate Control: Dynamic
50                TCP Rate: 0
51                TCP Rate Maximum Queue: 0
52  <!--NeedCopy-->
```

GUI を使用して **TCP** バーストレート制御を設定するには

1. [構成] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。
2. **TCP** プロファイルの設定ページで、ドロップダウンリストから **TCP** バーストコントロールオプションを選択します。

   a) BurstRateCntrl
   b) CreditBytePrms
   c) RateBytePerms
   d) RateSchedulerQ

3. [**OK]**、[完了**]** の順にクリックします。

ラップシーケンスからの保護 **(PAWS)** アルゴリズム

デフォルトの TCP プロファイルで TCP タイムスタンプオプションを有効にすると、NetScaler アプライアンスはラップシーケンス保護（PAWS）アルゴリズムを使用して、シーケンス番号が現在の TCP 接続の受信ウィンドウ内にある古いパケットを識別して拒否します。これは、シーケンスが「ラップ」した（最大値に達し、0 から再起動した）ためです。

ネットワークの混雑により SYN 以外のデータパケットが遅延し、パケットが到着する前に新しい接続を開くと、シーケンス番号のラップによって新しい接続がパケットを有効として受け入れてしまい、データが破損する可能性があります。ただし、TCP タイムスタンプオプションが有効な場合、パケットは破棄されます。

デフォルトでは、TCP タイムスタンプオプションは無効になっています。これを有効にすると、アプライアンスはパケットのヘッダーの TCP タイムスタンプ (Seg.tsval) を最近のタイムスタンプ (ts.Recent) の値と比較します。Seg.tsVal が ts.Recent と同じかそれより大きい場合、パケットは処理されます。それ以外の場合、アプライアンスはパケットをドロップし、修正確認を送信します。

PAWS の仕組み

PAWS アルゴリズムは、同期接続のすべての着信 TCP パケットを次のように処理します。

1. If `SEG.TSval` < `Ts.recent`: 受信パケットは受け入れられません。PAWS は（RFC-793 で指定されているように）確認応答を送信し、パケットをドロップします。注: ハーフオープン接続を検出して回復する TCP のメカニズムを維持するには、ACK セグメントの送信が必要です。
2. パケットがウィンドウの外にある場合:PAWS は、通常の TCP 処理と同様にパケットを拒否します。
3. `SEG.TSval` If > `Ts.recent`: `PAWS` がパケットを受け入れて処理します。
4. `SEG.TSval` <=`Last.ACK.sent` (到着セグメントが満たす場合): PAWS は`SEG.TSval`値を`Ts.recent`にコピーします。
5. パケットが順番に並んでいる場合:PAWS はパケットを受け入れます。
6. パケットが順番に並んでいない場合: パケットは通常のウィンドウ内順序外の TCP セグメントとして扱われます。たとえば、後で配信するためにキューに入れられている場合があります。
7. `Ts.recent` 値が 24 日以上アイドル状態の場合:PAWS タイムスタンプチェックが失敗すると、`Ts.recent` の有効性が確認されます。ts.Recent の値が無効であることが判明した場合、そのセグメントは受け入れられ、`PAWS rule`が新しいセグメントの`Ts.recent`のを TSval 値で更新します。

コマンドラインインターフェイスを使用して **TCP** タイムスタンプを有効または無効にするには

コマンドプロンプトで入力します。

```
`set nstcpprofile nstcp_default_profile -TimeStamp (ENABLED | DISABLED)
    `
```

GUI を使用して TCP タイムスタンプを有効または無効にするには

[システム] > [プロファイル] > [**TCP** プロファイル] に移動し、デフォルトの TCP プロファイルを選択して [編集] をクリックし、**TCP** タイムスタンプチェックボックスをオンまたはオフにします。

最適化手法

TCP では、以下の最適化手法と方法を使用してフロー制御を最適化します。

ポリシーベースの **TCP** プロファイル選択

今日のネットワークトラフィックは、かつてないほど多様で帯域幅を大量に消費しています。トラフィックの増加に伴い、サービス品質 (QoS) が TCP のパフォーマンスに与える影響は大きくなります。QoS を強化するために、ネットワークトラフィックのクラスごとに異なる TCP プロファイルを使用して AppQoE ポリシーを設定できるようになりました。AppQoE ポリシーは、仮想サーバーのトラフィックを分類して、3G、4G、LAN、WAN などの特定のタイプのトラフィックに最適化された TCP プロファイルを関連付けます。

この機能を使用するには、TCP プロファイルごとにポリシーアクションを作成し、AppQoE ポリシーにアクションを関連付け、負荷分散仮想サーバーにポリシーをバインドします。

サブスクライバ属性を使用して TCP 最適化を実行する方法については、ポリシーベースの TCP プロファイルを参照してください。

ポリシーベースの **TCP** プロファイル選択の設定

ポリシーベースの TCP プロファイル選択の設定は、次のタスクで構成されます。

- AppQoE を有効にする。TCP プロファイル機能を設定する前に、AppQoE 機能を有効にする必要があります。
- AppQoE アクションを追加します。AppQoE 機能を有効にしたら、TCP プロファイルを使用して AppQoE アクションを設定します。
- AppQoE ベースの TCP プロファイル選択の設定さまざまなクラスのトラフィックに TCP プロファイル選択を実装するには、NetScaler が接続を識別し、正しい AppQoE アクションを各ポリシーにバインドできる AppQoE ポリシーを構成する必要があります。
- AppQoE ポリシーを仮想サーバにバインドします。AppQoE ポリシーを構成したら、それらを 1 つ以上の負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーにバインドする必要があります。

コマンドラインインターフェイスを使用した構成

コマンドラインインターフェイスを使用して **AppQoE** を有効にするには

コマンドプロンプトで次のコマンドを入力して機能を有効にし、有効になっていることを確認します。

- enable ns feature appqoe
- show ns feature

コマンドラインインターフェイスを使用して **AppQoE** アクションを作成する際に **TCP** プロファイルをバインドするには

コマンドプロンプトで、次の AppQoE action コマンドとオプションを入力します。tcpprofiletobind

```
add appqoe action <name> [-priority <priority>] [-respondWith ( ACS |
 NS )[<CustomFile>] [-altContentSvcName <string>] [-altContentPath <
string>] [-maxConn <positive_integer>] [-delay <usecs>]] [-polqDepth
<positive_integer>] [-priqDepth <positive_integer>] [-dosTrigExpression
 <expression>] [-dosAction ( SimpleResponse |HICResponse )] [-tcpprofiletobind
 <string>]
show appqoe action
```

コマンドラインインターフェイスを使用して **AppQoE** ポリシーを構成するには

コマンドプロンプトで入力します。

```
add appqoe policy <name> -rule <expression> -action <string>
```

コマンドラインインターフェイスを使用して **AppQoE** ポリシーを負荷分散、キャッシュリダイレクト、またはコンテンツスイッチング仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind cs vserver cs1 -policyName <appqoe_policy_name> -priority <
priority>
bind lb vserver <name> - policyName <appqoe_policy_name> -priority <
priority>
bind cr vserver <name> -policyName <appqoe_policy_name> -priority <
priority>
```

例

```
1    add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -nagle
        ENABLED -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 500
         -slowStartIncr 1 -bufferSize 4194304 -flavor BIC -KA ENABLED -
        sendBuffsize 4194304 -rstWindowAttenuate ENABLED -spoofSynDrop
        ENABLED -dsack enabled -frto ENABLED -maxcwnd 4000000 -fack
        ENABLED -tcpmode ENDPOINT
2    add appqoe action appact1 -priority HIGH -tcpprofile tcp1
```



```
3      add appqoe policy apppol1 -rule "client.ip.src.eq(10.102.71.31)" -
          action appact1
4      bind lb vserver lb2 -policyName apppol1 -priority 1 -
          gotoPriorityExpression END -type REQUEST
5      bind cs vserver cs1 -policyName apppol1 -priority 1 -
          gotoPriorityExpression END -type REQUEST
6  <!--NeedCopy-->
```

GUI を使用したポリシーベースの **TCP** プロファイリングの設定

GUI を使用して AppQoE を有効にするには

1. [システム] > [設定**]** に移動します。
2. 詳細ウィンドウで、「拡張機能の設定」をクリックします。
3.「拡張機能の設定」ダイアログで、「**AppQoE** 」チェックボックスを選択します。
4. [**OK**] をクリックします。

GUI を使用して **AppQoE** ポリシーを設定するには

1. **[**** アプリエキスパート**] > [**AppQoE****] > [アクション] に移動します。**
2. 詳細ウィンドウで、次のいずれかの操作を行います。
3. アクションを作成するには、[追加] をクリックします。
4. 既存のアクションを変更するには、アクションを選択し、[編集] をクリックします。
5.「**AppQoE** アクションの作成」または「**AppQoE** アクションの設定」画面で、パラメータの値を入力または選択します。ダイアログボックスの内容は、「AppQoE アクションを構成するためのパラメーター」で説明されているパラメーターに次のように対応します (アスタリスクは必須パラメーターを示します)。

   a) 名前—名前
   b) アクションタイプ—次の式で応答
   c) 優先度—優先度
   d) ポリシーキューの深さ: POLQ の深さ
   e) キューの深さ—PRIQ の深さ
   f) DOS アクション—ディスコクション

6. [作成] をクリックします。

GUI を使用して **AppQoE** ポリシーをバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、サーバーを選択して [編集] をクリックします。
2.「ポリシー」セクションで、(+) をクリックして AppQoE ポリシーをバインドします。
3.「ポリシー」スライダーで、次の操作を行います。

a) ドロップダウンリストから AppQoE としてポリシータイプを選択します。
b) ドロップダウンリストからトラフィックタイプを選択します。

4.「ポリシーバインディング」セクションで、次の操作を行います。

a)「新規」をクリックして AppQoE ポリシーを作成します。
b) 既存のポリシーをクリックして、ドロップダウンリストから AppQoE ポリシーを選択します。

5. バインディングの優先順位を設定し、ポリシーを仮想サーバにバインドをクリックします。
6.［完了］をクリックします。

SACK ブロック生成

1 つのデータウィンドウで複数のパケットが失われると、TCP のパフォーマンスが低下します。このようなシナリオでは、選択的確認応答（SACK）メカニズムと選択的繰り返し再送信ポリシーを組み合わせることで、この制限を克服できます。順不同のパケットが受信されるたびに、SACK ブロックを生成する必要があります。

順序が狂ったパケットが再構成キューブロックに収まる場合は、そのブロックにパケット情報を挿入し、完全なブロック情報を SACK-0 に設定します。順序が狂ったパケットが再構成ブロックに収まらない場合は、そのパケットを SACK-0 として送信し、先の SACK ブロックを繰り返します。順序が合っていないパケットが重複していて、パケット情報が SACK-0 に設定されている場合は、ブロックを D-SACK します。

注**:** 確認済みのパケット、または既に受信された順序の悪いパケットの場合、そのパケットは D-SACK と見なされます。

クライアントリネージング

NetScaler アプライアンスは、SACK ベースのリカバリ中のクライアント更新を処理できます。

PCB 上の **end_point** をマーキングするためのメモリチェックでは、使用可能なメモリの合計が考慮されない

NetScaler アプライアンスでは、使用可能な合計メモリを使用せずに、メモリ使用量のしきい値を 75% に設定すると、新しい TCP 接続が TCP 最適化をバイパスします。

SACK ブロックの欠落による不必要な再送信

非エンドポイントモードで DUPACKS を送信するときに、順序が狂ったパケットのいくつかで SACK ブロックが欠落していると、サーバからの再送信が増えます。

接続の **SNMP** が過負荷のため最適化をバイパスしました

過負荷が原因で TCP 最適化をバイパスした接続数を追跡するために、次の SNMP ID が NetScaler アプライアンスに追加されました。

1. 1.3.6.1.4.1.5951.4.1.46.131 (TCP 最適化が有効)。TCP 最適化で有効になっている接続の総数を追跡します。
2. 1.3.6.1.4.1.5951.4.1.46.132 (TCP 最適化バイパス)。接続の総数を追跡するには、TCP 最適化をバイパスしました。

ダイナミック受信バッファ

TCP パフォーマンスを最大化するために、NetScaler ADC アプライアンスは TCP 受信バッファサイズを動的に調整できるようになりました。

テール・ロス・プローブ・アルゴリズム

再送信タイムアウト (RTO) は、トランザクションの最後でセグメントが失われることです。RTO は、特に短いウェブトランザクションで、アプリケーションのレイテンシーに問題がある場合に発生します。トランザクションの終了時に失われたセグメントを回復するために、TCP は Tail Loss Probe (TLP) アルゴリズムを使用します。
TLP は送信者専用アルゴリズムです。TCP 接続で一定期間確認応答がない場合、TLP は最後の未確認パケット（損失プローブ）を送信します。元の送信でテールロスが発生した場合、損失プローブからの確認応答により SACK または FACK のリカバリがトリガーされます。

テールロスプローブの設定

Tail Loss Probe (TLP) アルゴリズムを使用するには、TCP プロファイルで TLP オプションを有効にし、パラメータをそのプロファイルのセキュリティ要件に適した値に設定する必要があります。

コマンドラインを使用して **TLP** を有効にする　コマンドプロンプトで、次のコマンドのいずれかを入力して、新規または既存のプロファイルの TLP を有効または無効にします。

> 注記:
>
> デフォルト値は無効です。

```
add tcpprofile <TCP Profile Name> - taillossprobe ENABLED | DISABLED

set tcpprofile <TCP Profile Name> - taillossprobe ENABLED | DISABLED
```



```
unset tcpprofile <TCP Profile Name> - taillossprobe
```

例:

```
add tcpprofile nstcp_default_profile - taillossprobe
set tcpprofile nstcp_default_profile -taillossprobe Enabled
unset tcpprofile nstcp_default_profile -taillossprobe
```

NetScaler GUI を使用してテールロスプローブアルゴリズムを構成する

1. [構成] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。
2. 「**TCP** プロファイルの設定」ページで、「**Tail Loss Probe** 」チェックボックスを選択します。
3. [**OK]**、[完了**]** の順にクリックします。

NetScaler のトラブルシューティングソリューション

August 15, 2023

このトピックでは、アプライアンスで発生する問題を解決するために必要な基本的なトラブルシューティングソリューションをいくつか紹介します。NetScaler アプライアンスと、NetScaler アプライアンスがネットワークとどのように統合されるか、基本的なシステム機能にどのような問題があるかを理解できます。

NetScaler でパケットトレースを記録する方法

December 8, 2023

NetScaler GUI を使用してパケットトレースを記録できます。トレースは`nstrace.cap`に保存されます。

1. [システム**] > [**診断] に移動します。
2. [テクニカルサポートツール] の [新しいトレースを開始] をクリックします。
3. 「トレース開始」ページで、次のフィールドを更新します:

    a) パケットサイズ -トレース中にキャプチャするパケットのサイズを入力します。フルパケットトレースを行うには 0 を入力します。

    - デフォルト値:164
    - 最小値:0
    - 最大値:1514

b) **.pcap** 形式でトレースをキャプチャ -パケットトレースを nstrace (.cap) 形式または TCP ダンプ (.pcap) 形式でキャプチャするオプションがあります。デフォルトでは、パケットトレースは nstrace 形式 (.cap) でキャプチャされ、推奨形式です。TCP ダンプ形式でトレースをキャプチャするには、[**.pcap** 形式でトレースをキャプチャ] を選択します。

c) **SSL** マスターキーのキャプチャ -トレースをより適切に分析するには、「**SSL** マスターキーをキャプチャ」オプションを有効にします。この設定では、暗号化されたデータを復号化するために必要な、現在のセッションの SSL キーがキャプチャされます。SSL キーは`nstrace.sslkeys`という名前のファイルに保存されます。

- [**SSL** マスターキーをキャプチャ] オプションを有効にして [開始] をクリックしてトレースを開始すると、セキュリティ警告メッセージが表示されます。この警告を確認して続行してください。
- 秘密鍵が利用できない、または共有されていない場合は、秘密鍵の代わりに SSL セッション鍵をエクスポートすることを検討してください。

d) トレースファイルの数 -トレース中に生成されるトレースファイルの数を入力します。

- デフォルト値:24
- 最小値:1
- 最大値:100

e) トレースファイル名 -トレースファイルの名前を入力します。

f) トレースファイル **ID** -トレースファイルのファイル ID を入力します。

g) ファイルあたりのデータ時間 **(**秒**)** -各トレースファイルのデータをキャプチャする時間 (秒単位) を入力します。

- デフォルト値:3600
- 最小値:1

h) ファイルサイズ -各トレースファイルのファイルサイズ (MB 単位) を入力します。

- デフォルト値:1024
- 最小値:0
- 最大値:10240

トレースが指定されたファイルサイズに達すると、新しいトレースが開始されます。空きディスク容量が 2 GB 未満の場合、トレースは停止します。

i) **Trace Buffers** -パケットキャプチャを保存するトレースバッファ（一時ストレージ）の数を入力します。各バッファは約 16 KB です。

- デフォルト値:5000
- 最小値:1000

4. フィルター式を入力します。

IP アドレス、ポート、VLAN、またはインターフェイスにフィルター式を追加すると、関連するトラフィックのみがキャプチャされ、パケットトレース中の NetScaler の負荷が軽減されます。

5. リストから「マージ」オプションを選択します。

   - **ONSTOP** -一時トレースファイルが 1 つのトレースファイルにマージされます。
   - **NOMERGE** -トレースファイルはマージされません。
   - **ONTHEFLY** -トレースファイルは一時ファイルを作成せずにマージされます。

   デフォルト値: オンストップ

6. 使用可能な追加のパケットキャプチャオプションから関連するオプションを選択します。

   デフォルト値: 実行時クリーンアップを実行

7. キャプチャモードで必要なオプションを選択します。

   デフォルトでは、送信用にバッファリングされたパケット **(TXB)** と **NIC** パイプライン処理後の受信パケット **(NEW_RX**) が選択されます。秘密鍵なしでトレースを復号化するには、「復号化された **SSL** パケット (SSLPLAIN)」を選択します。

8. [開始] をクリックして、ネットワークパケットトレースの記録を開始します。
9. 「**Stop Trace** 」ページで「**Stop and Download** 」をクリックして、テスト完了後にネットワークパケットトレースの記録を停止します。
10. **[**** トレースファイルの削除**/**ダウンロード**]** ページで、ファイルを選択して [ダウンロード**]** をクリックし、**[**閉じる] をクリックします。**

    Wireshark ユーティリティでトレースファイルを開き、ファイルの内容を表示します。

次の Web ページにある自動ビルドセクションにある最新の Wireshark バージョンを使用することをお勧めします:. http://www.wireshark.org/download/automated。

仮想サーバー **IP** フィルタ（フロントエンドとバックエンドの両方）でパケットトレースをキャプチャするユースケース

仮想サーバーの IP アドレスのフィルターを使用し、CLI で「–link」オプションを有効にするか、GUI で「フィルターされた接続ピアトラフィックをトレースする」オプションを選択すると、IP アドレスのフロントエンドトラフィックとバックエンドトラフィックの両方をキャプチャできます。

```
start nstrace -size 0 -filter "CONNECTION.IP.EQ(1.1.1.1)" -link ENABLED

show nstrace
        State:  RUNNING          Scope:  LOCAL            TraceLocation
           :  "/var/nstrace/24Mar2017_16_00_19/..." Nf:  24
                          Time:  3600              Size:  0
                         Mode:  TXB NEW_RX
        Traceformat:  NSCAP      PerNIC:  DISABLED        FileName:  24
           Mar2017_16_00_19 Filter:  "CONNECTION.IP.EQ(1.1.1.1)" Link:
            ENABLED           Merge:  ONSTOP           Doruntimecleanup
           :  ENABLED
```



```
6          TraceBuffers:  5000      SkipRPC:  DISABLED       Capsslkeys:
              DISABLED    InMemoryTrace:  DISABLED
7  <!--NeedCopy-->
```

周期的なトレースのキャプチャ

断続的な問題のトラブルシューティングは常に困難です。周期的なトレースは、断続的な問題に最適です。トレースは、問題が発生するまでに数時間または数日にわたって実行できます。また、特定のフィルタを使用して、長期間実行する前に生成されるトレースファイルのサイズを評価することもできます。

CLI から以下のコマンドを実行します。

```
1  start nstrace -nf 60 -time 30 -size 0
2  This particular trace will create 60 files each of them for 30 sec.
      This means the files will start getting overwritten after 60 trace
      files or 30 mins
3  Show nstrace - To check the status of the nstrace
4  Stop nstrace - To stop the nstrace.
5
6  <!--NeedCopy-->
```

ベストプラクティス

1 秒あたり GB のトラフィックを処理するユニットでは、トラフィックのキャプチャは非常にリソースを大量に消費するプロセスです。リソースへの影響は、主に CPU とディスク容量の点にあります。ディスク容量への影響は、フィルタリング式を使用することで軽減できます。ただし、アプライアンスはパケットをキャプチャする前にフィルタに従ってパケットを処理する必要があるため、CPU への影響は残り、わずかに増加することがあります。

トレースのベストプラクティスは次のとおりです。

1. 対象のパケットが確実にキャプチャされる場合は、トレースを実行する期間をできるだけ制限する必要があります。
2. 営業時間外など、ユーザー数 (したがってトラフィック) が大幅に減少したときにトレースアクティビティが発生するようにスケジュールします。

/var ディレクトリの空き容量を増やす方法

March 20, 2024

次の記事では、管理者が Citrix /var ADC アプライアンスのディレクトリから領域を解放する方法について説明します。GUI にアクセスできない場合は、以下の手順に従ってください。

アプライアンスの /var ディレクトリのディスク容量が少ないと、GUI にサインインできないことがあります。このシナリオでは、古いログファイルを削除して /var ディレクトリに空き領域を作成できます。

確認事項

- ファイルをアプライアンスから削除する前に、必ずファイルをバックアップしてください。

NetScaler アプライアンスの/varディレクトリ内の領域を解放するには、次の手順を実行します。

1. SSH を使用して NetScaler の CLI にログオンします。このタスクを完了する方法の詳細については、NetScaler ドキュメントを参照してください。
2. NetScaler CLI にログオンした後、次のコマンドを使用してシェルプロンプトに切り替えます。`shell`
3. 次のコマンドを実行して、NetScaler アプライアンスの空き容量を確認します。`df -h`
4. /varディレクトリのメモリ容量が最大 90% 満たされている場合は、このディレクトリからファイルを削除する必要があります。

   - 次のコマンドを実行して、/var ディレクトリの内容を表示します。

     ```
     cd /var
     ls -l
     ```

   通常、目的のディレクトリは次のとおりです。

   ```
   /var/nstrace - This directory contains trace files.This is the most common reason for HDD being filled on the NetScaler appliance. This is due to an nstrace being left running for indefinite amount of time. All traces that are not of interest can and should be deleted. To stop an nstrace, go back to the CLI and issue stop nstrace command.

   /var/log - This directory contains system specific log files.

   /var/nslog - This directory contains NetScaler log files.

   /var/tmp/support - This directory contains technical support files, also known as, support bundles. All files not of interest should be deleted.

   /var/core - Core dumps are stored in this directory. There will be directories within this directory and they will be labeled with numbers starting with 1. These files can be quite large in size. Clear all files unless the core dumps are recent and investigation is required.

   /var/crash - Crash files, such as process crashes are stored in this directory. Clear all files unless the crashes are recent and investigation is required.

   ```



```
/var/nsinstall -  Firmware is placed in this directory when
    upgrading. Clear all files, except the firmware that is
    currently being used.
```

- ディレクトリのいずれかがより多くのスペースを使用しているかどうかを確認します。

```
du -hs *
44k     cache
2.0k    clusterd
2.0k    configdb
6.0k    core
989M    crash
4.0k    cron
2.0k    dev
6.0k    download
2.0k    gui
2.0k    install
2.0k    krb
2.0k    learnt_data
122M    log
366M    NetScaler
14k     ns_gui
86k     ns_sys_backup
631M    nsinstall
883M    nslog
32k     nsproflog
2.0k    nssynclog
16k     nstemplates
36k     nstmp
4.5G    nstrace
8.1M    opt
6.0k    pubkey
52k     run
28M     safenet
72M     tmp
2.0k    vmtools
14k     vpn
```

- 不要なファイルを削除します。

```
rm -r nstrace/*
```

ファイル削除についての詳細は FreeBSD のマニュアルページを参照してください。

- 不要なファイルを削除します。

  `rm -r nstrace/*`

  ファイル削除についての詳細は FreeBSD のマニュアルページを参照してください。

- ログまたは`nslog`ディレクトリの容量が多い場合は、次のコマンドを実行してログディレクトリを開き、その内容を表示します。

```
cd /var/log
```



```
ls -l
cd /var/nslog
ls -l
```

1. すべてのファイルが圧縮されていることを確認します。これは、.tar.gz ファイル名の拡張子で示されます。

   ファイルが圧縮されていない場合は、以下を実行してください。

   ファイルを***.gz*** 形式に圧縮するには***:***

   ```
   cd /var/log
   gzip <filename>
   ```

   圧縮ファイルは /var/log に置かれます

   ファイルを***.tar.gz*** 形式に圧縮するには***:***

   ```
   cd /var/nslog
   tar -cz <filename>.tar.gz <filename>
   ```

   圧縮ファイルは /var/nslog に保存されます

2. NetScaler コンソールを使用している場合は、/var/ns_system_backup ディレクトリを確認してください。NetScaler Console が作成したバックアップファイルを必ず消去してください。

その他のリソース

前述の手順で説明したコマンドの詳細については、http://ss64.com/bash/を参照してください。

NetScaler ADC アプライアンスからコアファイルまたはクラッシュしたファイルをダウンロードする方法

August 15, 2023

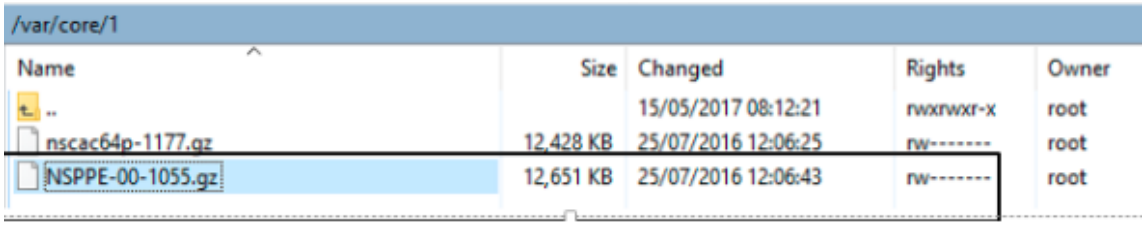
このトラブルシューティング記事では、管理者が NetScaler アプライアンスからコアファイルまたはクラッシュファイルをダウンロードする方法について説明します。

SFTP クライアントを使用して **NetScaler** アプライアンスからコアファイルまたはクラッシュファイルをダウンロードする

NetScaler アプライアンスからコアファイルまたはクラッシュファイルをダウンロードするには、次の手順を実行します。

1. WinSCP を開き、NetScaler 管理 IP アドレスにログオンします。

2. /var/core/1 に移動してファイルをダウンロードします。



> 注:
>
> 最新のクラッシュファイルまたはコアファイルをダウンロードするには、コマンドインターフェイスから WinSCP ツールを使用することもできます。ファイルは、コアディレクトリまたはクラッシュディレクトリのいずれかに配置できます。

パフォーマンス統計とイベントログを収集する方法

January 11, 2024

/var/nslogディレクトリにあるアーカイブのnewnslogファイルから、仮想サーバーと関連サービスのパフォーマンス統計を収集できます。newnslogファイルは/netscaler/nsconmsgの実行によって解釈されます。

CLI を使用してパフォーマンス統計とイベントログを収集する

NetScaler ADC シェルプロンプトからnsconmsgコマンドを実行して、イベントを報告できます。

コマンドプロンプトで入力します:

```
/netscaler/nsconmsg -K /var/nslog/newnslog -d event
```

```
1 Displaying event information
2 NetScaler V20 Performance Data
3 NetScaler NS10.5: Build 57.7.nc, Date: May 14 2015, 07:35:21
4 rtime: Relative time between two records in milliseconds
5 seqno rtime event-message                              event-time
6 11648 16310 PPE-0 MonServiceBinding_10.104.20.110:443_(tcp-default)
7 <!--NeedCopy-->
```

指定した「**newnslog**」ファイルの対象期間を表示する

コマンドプロンプトで入力します:

```
/netscaler/nsconmsg -K /var/nslog/newnslog -d setime
```


現在のデータが/var/nslog/newnslogファイルに追加されます。NetScaler は、デフォルトでは 2 日ごとにnewnslogファイルを自動的にアーカイブします。アーカイブされたデータを読み取るには、次の例のようにアーカイブを抽出する必要があります。

cd /var/nslog: コマンドを実行して、NetScaler シェルプロンプトから特定のディレクトリに移動します。

tar xvfz newnslog.100.tar.gz: tar ファイルを抽出するコマンド。

/netscaler/nsconmsg -K newnslog.100 -d setime: 特定のファイルの対象期間を確認するコマンド (この例ではnewnslog.100) 。

ls -l: コマンドは、それらのファイルに関連付けられているすべてのログファイルとタイムスタンプをチェックします。

```
root@NETSCALER# cd /var/nslog
root@NETSCALER# ls -l
```

```
 wheel    461544 Aug  7  2014 newnslog.1.tar.gz
-rw-r--r--   1 root       wheel    191067 Aug  7  2014 newnslog.10.tar.
   gz
-rw-r--r--   1 root       wheel  11144873 Apr 26 22:04 newnslog.100.tar
   .gz
-rw-r--r--   1 root       wheel  11095053 Apr 28 22:04 newnslog.101.tar
   .gz
-rw-r--r--   1 root       wheel  11114284 Apr 30 22:04 newnslog.102.tar
   .gz
-rw-r--r--   1 root       wheel  11146418 May  2 22:04 newnslog.103.tar
   .gz
-rw-r--r--   1 root       wheel  11104227 May  4 22:04 newnslog.104.tar
   .gz
-rw-r--r--   1 root       wheel  11297419 May  6 22:04 newnslog.105.tar
   .gz
-rw-r--r--   1 root       wheel  11081212 May  8 22:04 newnslog.106.tar
   .gz
-rw-r--r--   1 root       wheel  11048542 May 10 22:04 newnslog.107.tar
   .gz
-rw-r--r--   1 root       wheel  11101869 May 12 22:04 newnslog.108.tar
   .gz
-rw-r--r--   1 root       wheel  11378787 May 14 22:04 newnslog.109.tar
   .gz
-rw-r--r--   1 root       wheel  44989298 Apr 11  2014 newnslog.11.gz
<!--NeedCopy-->
```

ファイル内のタイムスパンを表示

以下の例に示すように、nsconmsg コマンドを使用して、指定したファイル内の一定期間だけを表示します。

```
/netscaler/nsconmsg -K /var/nslog/newnslog -s time=22Mar2007:20:00 -T
 7 -s ConLb=2 -d oldconmsg
```

各項目の意味は次のとおりです。

s：時間 =2007 年 3 月 22 日 20:00:00 は 2007 年 3 月 22 日のちょうど 20:00 に始まります。

T　7：7 秒間のデータを表示します

s：ロードバランシング統計の詳細レベルを表示します。

d：統計情報を表示します。

> 注:
>
> ADC リリース 12.1 からは、「時間」の秒数にも追加する必要があります。つまり、2007 年 3 月 22 日 20:00:00

-d oldconmsg パラメータによって提供される統計情報は 7 秒ごとに記録されます。以下は出力例です。

```
VIP(10.128.58.149:80:UP:WEIGHTEDRR): Hits(38200495, 18/sec) Mbps(1.02)
   Pers(OFF) Err(0)
Pkt(186/sec, 610 bytes) actSvc(4) DefPol(NONE) override(0)
Conn: Clt(253, 1/sec, OE[252]) Svr(3)
S(10.128.49.40:80:UP) Hits(9443063, 4/sec, P[2602342, 0/sec]) ATr(5)
   Mbps(0.23) BWlmt(0 kbits) RspTime(112.58 ms)
Other: Pkt(36/sec, 712 bytes) Wt(10000) RHits(31555)
Conn: CSvr(42, 0/sec) MCSvr(20) OE(16) RP(11) SQ(0)
S(10.128.49.39:80:UP) Hits(9731048, 4/sec, P[2929279, 0/sec]) ATr(9)
   Mbps(0.27) BWlmt(0 kbits) RspTime(161.69 ms)
Other: Pkt(41/sec, 756 bytes) Wt(10000) RHits(31555)
Conn: CSvr(32, 0/sec) MCSvr(19) OE(13) RP(4) SQ(0)
S(10.128.49.38:80:UP) Hits(9341366, 5/sec, P[2700778, 0/sec]) ATr(4)
   Mbps(0.27) BWlmt(0 kbits) RspTime(120.50 ms)
Other: Pkt(42/sec, 720 bytes) Wt(10000) RHits(31556)
Conn: CSvr(37, 0/sec) MCSvr(19) OE(13) RP(9) SQ(0)
S(10.128.49.37:80:UP) Hits(9685018, 4/sec, P[2844418, 0/sec]) ATr(3)
   Mbps(0.23) BWlmt(0 kbits) RspTime(125.38 ms)
Other: Pkt(38/sec, 670 bytes) Wt(10000) RHits(31556)
Conn: CSvr(32, 0/sec) MCSvr(20) OE(10) RP(7) SQ(0)
<!--NeedCopy-->
```

> 注:
>
> 個々のサービスのクライアント接続数は、仮想サーバーのクライアント接続数に加算されません。その理由は、NetScaler ADC アプライアンスとバックエンドサービス間のセッション再利用のためです。

仮想サーバー出力

```
VIP(10.128.58.149:80:UP:WEIGHTEDRR): Hits(38200495, 18/sec) PHits(5)
   Mbps(1.02) Pers(OFF) Err(0) LConn_Best [Idx:SubIdx] 0:0
   PrimVserverDownBackupHits(0)
Pkt(186/sec, 610 bytes) actSvc(4) DefPol(NONE) override(0) newlyUP(0)
Conn: Clt(253, 1/sec, OE[252]) Svr(3) SQ(Total: 0 OnVserver: 0
   OnServices: 0)
slimit_SO: (Sothreshhold: 0 [Ex: 0]  Consumed: [Ex: 0 Borrowed: 0
   TotActiveConn: 0] Available: 0)
<!--NeedCopy-->
```


次のリストは、仮想サーバーの統計情報を示しています。

1. IP (IP address:port:state:Load balancing method): 設定した仮想 IP アドレスの IP アドレスとポート。仮想サーバの状態または仮想 IP アドレスが UP、DOWN、または OUT OF SERVICE です。仮想 IP アドレスに設定されている負荷分散方法。
2. Hits (#): 仮想サーバーに到達したリクエストの数。
3. Mbps (#): 仮想サーバ上の総トラフィック量（Rx+Tx）をメガビット/秒に換算しました。
4. Pers: パーシステンスのタイプが設定されています。
5. Err (#): 仮想サーバーによってエラーページが生成された回数。
6. Pkt (#/sec, # bytes): 仮想サーバを通過するネットワークトラフィック（パケット）の量と仮想サーバを通過する平均パケットサイズ。
7. actSvc(#): 仮想サーバにバインドされているアクティブなサービスの数。
8. DefPol (RR): デフォルトの負荷分散方法がアクティブかどうかを示します。一部の初期リクエストには、他の方法の動作をスムーズにするために、デフォルトの負荷分散方法が使用されます。
9. Clt (#, #/sec): 仮想サーバレートへの現在のクライアント接続数。
10. OE [#]: オープン確立状態の仮想サーバからのサーバ接続の数。
11. Svr (#): 仮想サーバーからの現在のサーバー接続の数。
12. PHits (#): パーシスタンスヒットの数。
13. SO: スピルオーバーが発生した回数。
14. LConn_Best [Idx:SubIdx] (port:#)。最小接続方法を使用した場合の、最適なサーバーのインデックスサブスロット。
15. PrimVserverDownBackupHits (#): プライマリサーバがダウンしているときに仮想サーバをバックアップするヒット数。
16. Override (#): MaxCLT の L2Conn に基づいて次善のサーバが選択された回数。
17. newlyUP (#): 新たに稼働している現在のサービスの数。
18. SQ(Total:OnVserver:OnServices:): 現在のサージキューの長さ。
19. slimit_SO: (Sothreshhold:Exclusive:Consumed: [Exclusive:Borrowed: TotActiveConn:] Available: (#)): スピルオーバーの共有限度に関する独占的かつ共有された情報。

上記の出力では、Svr(3)はコマンドが統計サンプルを収集していることを示しています。合計で 4 つのサービスがありますが、仮想サーバーからバックエンドサーバーへのアクティブな接続は 3 つあります。クライアントが仮想サーバーとの接続を確立すると、コマンドが情報を収集するときにクライアントがトラフィックを送受信する必要はありません。そのため、Svrカウンターが数値OE[]よりも低くなるのが一般的です。Svr カウンタは、データをアクティブに送受信しているアクティブな接続の数を表します。サブネット IP アドレス (SNIP) は、関連するバックエンドサーバーに接続されます。また、NetScaler ADC は、バックエンドサーバーに接続された仮想サーバーを追跡し、カウンターを計算します。

仮想サービス出力


```
S(10.128.49.40:80:UP) Hits(9443063, 4/sec, P[2602342, 0/sec]) ATr(5) Mbps(0.23) BWlmt(0 kbits) RspTime(112.58 ms) Load(0) LConn_Best [Idx:SubIdx] (C:0; V:0,I:1, B:0, X:0, SI:0)
Other: Pkt(36/sec, 712 bytes) Wt(10000) Wt(Reverse Polarity)(10000) RHits(31555) Conn: CSvr(42, 0/sec) MCSvr(20) OE(16) E(5) RP(11) SQ(0)
slimit_maxClient: (MaxClt: 2 [Ex: 0]  Consumed: [Ex: 0 Borrowed: 0 TotActiveConn: 0] Available: 2)
newlyUP_mode: NO, Pending: 0, update: 0x0, incr_time: 0x0, incr_count: 0
<!--NeedCopy-->
```

次のリストは、サービス統計を示しています。

1. S (IP address:port:state): IP アドレス、ポート、およびサービスの状態（DOWN、UP、OUT OF SERVICE など）
2. Hits (#, P[#]): サービスに送信されたリクエストの数、設定されたサーバパーシスタンスによりサービスに送信されたリクエストの数。
3. ATr (#): サービスへのアクティブな接続の数。

   > 注:
   >
   > アクティブな接続には、サービスへの未処理のリクエストがあるか、現在トラフィックアクティビティがあります。

4. Mbps (#.##): サービスの総トラフィック量（Rx+Tx）をメガビット/秒に換算しました。
5. BWlmt (# kbits): 帯域幅制限が定義されています。
6. RspTime (# ms): サービスの平均応答時間 (ミリ秒)。
7. Pkt(#/sec, #bytes): サービスに送信される 1 秒あたりのパケット数で表したトラフィック量。パケットの平均サイズ。
8. Wt (#): 負荷分散アルゴリズムで使用されるウェイトインデックス。

   > 注:
   >
   > この値を 10,000 で割ると、サービスの実際に設定されている重みがわかります。

9. RHits (#): ラウンドロビン負荷分散アルゴリズムで使用される実行リクエストカウンタ。
10. CSvr (#, #/sec): サービスレートへの接続数。
11. MCSvr (#): サービスへの最大接続数。
12. OE (#): オープン状態および確立状態のサービスへの接続の数。
13. E (#): 確立された状態のサービスへの接続数。
14. RP (#): 再利用プールにあるサービスへの接続数。

15. SQ (#): サージキューで待機中のサービスへの接続数。
16. Load (#): サービスをロードします。
17. LConn_Idx: (Current index(#); current virtual index(#),I:(#), base virtual slot index(#), transaction (#), Sub slot index(#)): 最も少ない接続方法が使用されている場合のサーバーのインデックス。
18. Wt(Reverse Polarity): 負荷分散アルゴリズムで使用されるリバースウェイトインデックス。
19. slimit_maxClient: (MaxClient [Exclusinve] Consumed: [Exclusive: Borrowed:TotActiveConnection:] Available: (#)): 最大クライアント数の共有制限に関する独占情報および共有情報
20. newlyUP_mode: (No, pending (#), update (#*#), incr_time (#*#), incr_count (#)): サービスが新しく起動したかどうか、および新しいサービスで許可されたヒット数に対応する統計情報を示します。また、このサービスのウェイトが更新される時刻でもあります。

NetScaler GUI を使用してパフォーマンス統計とイベントログを収集する

1. [システム] **>** [診断] **>** [メンテナンス] **>** [ログファイルの削除/ダウンロード] に移動します。
2. ファイルを選択して **[**ダウンロード] をクリックしてファイルをダウンロードします。

ログファイルのローテーションを設定する方法

August 15, 2023

NetScaler アプライアンスは、複数のディレクトリにさまざまな形式でログを生成します。これらのログの一部はデフォルトではローテーションされず、サイズが大きくなりすぎてディスク容量を消費する可能性があります。付属のログローテーションユーティリティ (newsyslog) を使用すると、関連情報のみを保持して管理や管理が容易になるため、これらのログを一貫して管理できます。

NetScaler ファームウェアに含まれるnewsyslogユーティリティは、ログファイルをアーカイブし、システムログをローテーションして、ローテーション中は現在のログが空になるようにします。システムの crontab はこのユーティリティを 1 時間おきに実行し、ローテーションするファイルと条件を指定する設定ファイルを読み取ります。アーカイブされたファイルは、必要に応じて圧縮される場合があります。

既存の設定は/etc/newsyslog.confにあります。ただし、このファイルはメモリファイルシステムにあるため、/nsconfig/newsyslog.conf NetScaler を再起動しても構成が存続できるように、管理者は変更を保存する必要があります。

このファイルに含まれるエントリの形式は次のとおりです。

```
logfilename [owner:group] mode count size when flags [/pid_file] [
sig_num]
```


> 注:
>
> 角括弧内のフィールドはオプションで、省略できます。

ファイルの各行はログファイルと、ローテーションを行う必要がある条件を表しています。

この例では、size フィールドのサイズは 100 KB ns.log と表示されています。countフィールドには、アーカイブされたns.logファイルの数が 25 と表示されます。サイズは 100 K、カウントは 25 がデフォルトのサイズとカウント値です。

> 注:
>
> フィールドがアスタリスク (*) で設定されている場合、ns.log ファイルは時間に基づいてローテーションされません。1 時間ごとに crontab newsyslog ジョブがユーティリティを実行し、ns.log のサイズがこのファイルで設定されているサイズ以上かどうかをチェックします。この例では、ファイルが 100 K 以上の場合、そのファイルをローテーションします。

```
root@ns# cat /etc/newsyslog.conf
# Netscaler newsyslog.conf

# This file is present in the memory filesystem by default, and any
    changes
# to this file will be lost following a reboot. If changes to this file
# require persistence between reboots, copy this file to the /nsconfig
# directory and make the required changes to that file.
#
# logfilename [owner:group] mode count size when flags [/pid_file] [
    sig_num]
/var/log/cron 600 3 100 * Z
/var/log/amd.log 644 7 100 * Z
/var/log/auth.log 600 7 100 * Z
/var/log/ns.log 600 25 100 * Z
<!--NeedCopy-->
```

sizeフィールドを変更してns.logファイルの最小サイズを変更することも、特定の時間に基づいてns.logファイルをローテーションするようにフィールドを変更することもできます。

日単位、週単位、月単位の仕様は、それぞれ [Dhh]、、、[Dhh [Mdd]]の形式で示されます。時間フィールドはオプションで、デフォルトは午前 0 時です。これらの仕様の範囲と意味は次のとおりです。

```
Hh hours, range 0 ... 23
w day of week, range 0 ... 6, 0 = Sunday
dd day of month, range 1 ... 31, or the letter L or l to specify the
    last day of the month.
<!--NeedCopy-->
```

例:

デフォルトでローテーションされるログの説明を含む例をいくつか示します。

```
/var/log/auth.log 600 7 100 * Z
```


ファイルが 100 K に達すると、認証ログはローテーションされ、auth.log の最後の 7 つのコピーが gzip (Z フラグ) でアーカイブおよび圧縮され、生成されたアーカイブには次の権限—rw——-が割り当てられます。

```
/var/log/all.log 600 7 * @T00 Z
```

キャッチオールログは毎晩深夜に 7 回ローテーションされ（@T00）、gzip で圧縮されます。作成されたアーカイブには、次の権限が割り当てられます—rw-r—-。

```
/var/log/weekly.log 640 5 * $W6D0 Z
```

週次ログは、毎週月曜日の午前 0 時に 5 回ローテーションされます。作成されたアーカイブには権限が割り当てられます。

一般的な回転パターン**:**

- D0. 毎晩深夜にローテーションします
- D23. 毎日 23:00 にローテーションします
- W0D23. 毎週日曜日の 23:00 にローテーションします
- W5. 毎週金曜日の深夜にローテーションします
- MLD6. 毎月最終日の 6:00 にローテーション
- M5. 毎月 5 日おきに深夜にローテーションします

間隔と時間の両方を指定する場合は、両方の条件を満たす必要があります。つまり、ファイルは指定された間隔と同じかそれよりも古く、現在の時刻は時間指定と一致している必要があります。

最小ファイルサイズを制御することはできますが、newsyslog 次の時間帯にユーティリティが処理を開始するまでのファイルサイズに制限はありません。

デバッグニュース/システムログ**:**

newsyslog ユーティリティの動作をデバッグするには、verbose フラグを追加します。

```
root@dj_ns# newsyslog -v
/var/log/cron <3Z>: size (Kb): 31 [100] --> skipping
/var/log/amd.log <7Z>: does not exist, skipped.
/var/log/auth.log <7Z>: size (Kb): 2 [100] --> skipping
/var/log/kerberos.log <7Z>: does not exist, skipped.
/var/log/lpd-errs <7Z>: size (Kb): 0 [100] --> skipping
/var/log/maillog <7Z>: --> will trim at Tue Mar 24 00:00:00 2009
/var/log/sendmail.st <10>: age (hr): 0 [168] --> skipping
/var/log/messages <5Z>: size (Kb): 7 [100] --> skipping
/var/log/all.log <7Z>: --> will trim at Tue Mar 24 00:00:00 2009
/var/log/slip.log <3Z>: size (Kb): 0 [100] --> skipping
/var/log/ppp.log <3Z>: does not exist, skipped.
/var/log/security <10Z>: size (Kb): 0 [100] --> skipping
/var/log/wtmp <3>: --> will trim at Wed Apr 1 04:00:00 2009
/var/log/daily.log <7Z>: does not exist, skipped.
/var/log/weekly.log <5Z>: does not exist, skipped.
/var/log/monthly.log <12Z>: does not exist, skipped.
/var/log/console.log <5Z>: does not exist, skipped.
/var/log/ns.log <5Z>: size (Kb): 18 [100] --> skipping
```



```
/var/log/nsvpn.log <5Z>: size (Kb): 0 [100] --> skipping
/var/log/httperror.log <5Z>: size (Kb): 1 [100] --> skipping
/var/log/httpaccess.log <5Z>: size (Kb): 1 [100] --> skipping
root@dj_ns#
<!--NeedCopy-->
```

NetScaler アプライアンスの/flash ディレクトリの空き容量を増やす方法

August 15, 2023

このトラブルシューティング記事では、管理者が NetScaler アプライアンスの/flash ディレクトリからスペースを解放する方法について説明します。

NetScaler アプライアンスの/flash ディレクトリのスペースを解放する手順

1. SSH を使用して NetScaler ADC の CLI にログオンします。
2. NetScaler CLI にログオンした後、次のコマンドを使用してシェルプロンプトに切り替えます。`shell`。
3. `df -h` コマンドを実行して、NetScaler アプライアンスの空き容量を確認します。
4. /flash ディレクトリの容量が 90% を超えているか少ない場合は、このディレクトリからいくつかのファイルを削除する必要があります。
5. 次のコマンドを実行して、/flash ディレクトリの内容を表示します。

   ```
   cd /flash
   ls -l
   ```

6. NetScaler ソフトウェアリリースのさまざまなバージョンの複数のファイルが見つかる場合があります。この場所にあるファイルが、アプライアンス上の NetScaler ソフトウェアの現在のバージョンに適用できるファイルであることを確認してください。次のコマンドを実行して、アプライアンスから他のファイルを削除します。

   ```
   rm <filename>
   ```

> 注
>
> 古いバージョンのカーネルのみを削除してください。/flash ディレクトリには、NetScaler ソフトウェアリリースの現在のバージョンまたはビルドで使用されているファイルと kernel.gz ファイルが含まれている必要があります。Citrix では、これらのファイルを/flash ディレクトリから削除しないことをお勧めします。

参考資料

August 15, 2023

このリファレンス情報を使用して、次の NetScaler ADC コンポーネントについて詳しく理解してください。

NetScaler SNMP OID -NetScaler ADC アプライアンスから情報を取得するために使用できる SNMP OID の詳細。

NetScaler Syslog メッセージ -NetScaler アプライアンスによって提供される Syslog メッセージの詳細。

NetScaler CLI コマンド -CLI を使用して NetScaler ADC アプライアンスを構成するために使用できるコマンドの詳細。「man <ns-command-name>」コマンドを入力して、CLI で各コマンドの詳細を表示することもできます。

API リファレンス -REST API を使用して NetScaler ADC アプライアンスで実行できるすべての操作の詳細。

NetScaler ADC の詳細ポリシー式 -高度なポリシーの定義に使用できる式の詳細。