

NetScaler BLX 14.1

Contents

NetScaler BLX について	7
NetScaler BLX と他のソフトウェアフォームファクターの違いは何ですか？	7
NetScaler BLX を使用する理由？	7
一般的なアーキテクチャ	8
サポートされているNetScaler®機能	9
NetScaler BLX ライセンス	10
NetScaler BLX のホスト ID を取得する	10
新しいライセンスをインストールする	11
システム要件	11
サポートされているLinuxディストリビューション	11
Linux ホストのハードウェア要件	12
オープンソースパッケージ	13
自動インストールされる依存関係	14
手動でインストールされる依存関係	15
NetScaler BLX を展開する	17
Linux ホストに NetScaler BLX パッケージをダウンロードする	17
NetScaler BLX をダウンロードする	17
次のステップ	18
DebianベースのLinuxホストにNetScaler BLXをインストールする	18
始める前に	18
NetScaler BLXをインストールする	18
次のステップ	20
RPMベースのLinuxホストにNetScaler BLXをインストールする	20
始める前に	20
NetScaler BLXをインストールする	21
次のステップ	22

NetScaler BLX を構成する	23
前提条件	23
NetScaler BLX を構成する	24
次のステップ	27
NetScaler BLX 設定ファイル	27
NetScaler BLXにおける起動時のカスタマイズスクリプト	29
起動スクリプトの構成	30
NetScaler BLX の圧縮コアダンプを設定する	30
<code>blx.conf</code> ファイルを使用してコアダンプを有効にする	30
<code>blx.conf</code> ファイルを使用してコアダンプを無効にする	31
NetScaler BLXマネージドホストの設定	32
LinuxホストへのSSHアクセスでNetScaler BLXマネージドホストを有効にする	33
NSIP と Linux ホストに異なる IP アドレスを設定する	34
DPDKサポートなしのNetScaler BLX向けnsdrvdドライバーの構成	36
前提条件	38
<code>nsdrvd</code> ドライバを有効にする	38
<code>nsdrvd</code> ドライバーを無効にする	39
NetScaler BLXとFastly Next-Gen WAFエージェントを統合する	39
主な利点	39
アーキテクチャの概要	40
Fastly NextGen WAFエージェントと統合するようにNetScaler BLXを構成する方法	40
NetScaler BLX の起動	42
NetScaler BLX のステータスを確認する	42
NetScaler BLXを停止する	43
次のステップ	43
NetScaler BLX にアクセスし、NetScaler 機能を構成する	44
開始する前に	44

パスワードの要件	44
NetScaler BLXにアクセスし、NetScaler CLIを使用してNetScaler機能を構成する	45
NetScaler BLXにアクセスし、NetScaler GUIを使用してNetScaler機能を構成する	45
GUI を使用して NetScaler BLX にアクセスする	46
NITRO API を使用して NetScaler BLX にアクセスし、NetScaler の機能を構成する	46
NetScaler BLXへのアクセス	46
次のステップ	47
NetScaler BLX クラスターをセットアップする	47
始める前に	47
NetScaler BLXクラスターの制限事項	48
NetScaler BLXクラスターのセットアップ	49
アップグレードとダウングレード	50
開始する前に	50
NetScaler BLXのアップグレード	51
NetScaler BLXのダウングレード	51
アップグレードまたはダウングレード後の NetScaler BLX でのエンティティステータスの確認	52
ユースケース	53
AWSへのNetScaler BLXの展開	53
AWS上でのNetScaler BLX™のデプロイ	53
AWS用語	53
AWSでのNetScaler BLXの仕組み	56
前提条件	57
制限事項と使用ガイドライン	58
AWS上にスタンドアロンのNetScaler BLX™インスタンスを展開する	59
デプロイ手順	59
EC2 Linuxインスタンスを作成する	59
NetScaler BLXのダウンロード	63
NetScaler BLXのインストール	63

BLXを構成する	63
NetScaler BLXを起動する	63
NetScaler BLXにアクセスする	63
ライセンス	63
AWSでGSLBを使用してNetScaler BLX™を展開する	63
ドメインベースサービス概要	64
ELBとのDBS	64
前提条件	64
blx.confファイルのサンプル	65
GSLBの構成	66
よくある質問	67
NetScaler BLXとは何ですか？	67
なぜNetScaler BLXが必要なのですか？	67
NetScaler BLX、CPX、VPXの違い？	67
NetScaler BLXを使用するタイミング？	67
ハイパーバイザーやコンテナがないことによって、どのような違いがありますか？	68
NetScaler BLXはどのサーバーハードウェアでも実行できますか？	68
LinuxサーバーにNetScaler BLXをデプロイするにはどうすればよいですか？	68
NetScaler BLXソフトウェアの展開を自動化できますか？	69
NetScaler BLXがLinux OSを搭載したサーバーにインストールされている場合の、他の標準的なLinuxパッケージやアプリケーションのインストール？	69
NetScaler BLXライセンスの購入方法？	69
現在のNetScaler VPXライセンスをNetScaler BLXに使用できますか？	69
NetScaler BLXをワンアームモードとツーアームモードで展開できますか？	69
NetScaler BLXはどのネットワークスタックを使用しますか？	70
NetScaler BLXは高可用性をサポートしていますか？	70
NetScaler BLXとNetScaler VPXまたはCPXの間で高可用性ペアをセットアップできますか？	70
DPDKを搭載した仮想化LinuxマシンでNetScaler BLXを実行できますか？	70

ARMプラットフォームでNetScaler BLXを実行できますか？	70
NetScaler BLXの管理IPアドレスは何ですか？	70
NetScaler BLX のメモリ使用率が常に高いのはなぜですか？	71
トラブルシューティング	72
<code>blx.conf</code> ファイル内の <code>ipaddress</code> (NSIP) を変更しましたが、NetScaler BLX は古い NSIP アドレスを使い続けています	72
<code>systemctl start blx</code> コマンドを使用して NetScaler BLX を起動しましたが、起動に失敗します	72
NetScaler BLX が DPDK ポートで起動しない	72
NetScaler BLX の制限事項と使用ガイドライン	73
高可用性	73
NetScaler BLX クラスター	73
NetScaler ゲートウェイ	73
LA および LACP チャンネル	73
ウェブアプリケーションファイアウォール (WAF)	73
DPDKポートを使用するNetScaler BLX	74
Mellanoxポート	74
その他の制限とガイドライン	74
NetScaler BLX でサポートされていない NetScaler 機能	78
<code>_fix_</code>	79

NetScaler BLX について

NetScaler BLX は、NetScaler のソフトウェアフォームファクターの 1 つです。商用オフザシェルフサーバー (COTS) 上のベアメタル Linux でネイティブに実行できます。

NetScaler は、アプリケーション固有のトラフィック分析を実行して、Web アプリケーションのレイヤー 4 ～ レイヤー 7 (L4～L7) ネットワークトラフィックをインテリジェントに分散、最適化、および保護するアプリケーションデリバリーコントローラーです。たとえば、NetScaler は、長期間存続する TCP 接続ではなく、個々の HTTP リクエストに対する決定をロードバランスします。詳細については、「[NetScaler の理解](#)」を参照してください。

NetScaler BLX と他のソフトウェアフォームファクターの違いは何ですか？

NetScaler BLX、CPX、VPX は NetScaler のソフトウェアフォームファクターです。

- NetScaler BLX は、Linux システムでネイティブに実行される軽量ソフトウェアパッケージです。NetScaler BLX は、ハイパーバイザーやコンテナのオーバーヘッドなしでシンプルさを提供し、パフォーマンスを向上させます。NetScaler BLX は、選択したハードウェア上で Linux プロセスとして実行されます。
- CPX は NetScaler® のコンテナ化されたバージョンであり、コンテナ内で実行する必要があります。
- VPX は仮想アプライアンスであり、サーバーにインストールされたハイパーバイザー上で実行する必要があります。

NetScaler BLX を使用する理由？

NetScaler BLX を使用する利点は次のとおりです。

- **クラウド対応** - NetScaler BLX は、クラウドでの実行に対するデイズロサポートを提供します。クラウド上にプロビジョニングされた Linux 仮想マシン上でソフトウェアアプリケーションとして実行されるため、クラウドで実行するための認定は必要ありません。
- **簡単な管理** - Linux オペレーティングシステムの標準ツールを使用して、NetScaler BLX を監視および管理できます。また、NetScaler BLX を既存のオーケストレーション設定に簡単に組み込むこともできます。

- **シームレスなサードパーティツール統合** - Linux 環境でサポートされているオープンソースツールを NetScaler BLX とシームレスに統合できます。統合ごとに個別のプラグインを開発する必要はありません。
- **他のアプリケーションとの共存** - NetScaler BLX はソフトウェアアプリケーションとして実行されます。他の Linux アプリケーションも同じホスト上で実行できます。
- **DPDK サポート** - NetScaler BLX は、パフォーマンス向上のために Data Plane Development Kit (DPDK) 統合をサポートしています。DPDK オープンソースライブラリを使用して、パフォーマンスを向上させ、パケット処理における Linux カーネルのボトルネックを克服します。

一般的なアーキテクチャ

NetScaler BLXはNetScalerのソフトウェアフォームファクターであり、他のフォームファクターと同じ機能を提供します。Linuxホスト上でユーザースペースアプリケーションとして動作します。

NetScaler BLXは、パケットの送受信およびNICポートの管理にLinuxドライバーを使用します。起動フェーズ中に作成される仮想イーサネット (veths) インターフェース `blx0` および `blx1` は、LinuxホストとNetScaler BLX間の通信に使用されます。例えば、NetScaler BLXはvethsを使用して、Linuxホスト上のsyslogデーモンにログ情報を送信します。

NetScaler BLXでは、`/etc/blx/blx.conf` (NetScaler BLXの起動設定ファイル) ファイルで指定されたLinuxホストのNICポートは、NetScaler BLX専用です。これらのNICポートは、ホスト上で実行されている他のアプリケーションと共有されません。専用のNICポートはNetScaler BLXのみが認識できます。

IPアドレス指定スキームは、従来のNetScalerと同様です。NetScaler BLXは、NetScaler IP (NSIP)、仮想サーバーIP (VIP)、およびサブネットIP (SNIP) に異なるIPアドレスを持つことができます。

NetScaler BLXは、外部ネットワークからパケットを受信し、受信したパケットを処理し、設定された専用のLinux NICポートを介して直接応答します。LinuxカーネルのTCP/IPスタックをバイパスしてパケットを処理するための本格的なTCP/IPスタックを備えています。NetScaler BLXは、Linuxカーネルドライバーと直接連携して、NICポートから生パケットを取得します。

NetScaler BLXはLinuxカーネルのネットワークスタックをバイパスしますが、Linuxカーネルメモリとユーザースペースメモリ間でパケットを転送する際に依然としてオーバーヘッドが発生します。このオーバーヘッドは、パケット処理の全体的なパフォーマンスに影響を与えます。高いパケット処理パフォーマンスを実現するために、Data Plane Development Kit (DPDK) 互換のNICを

使用することをお勧めします。NetScaler BLXでサポートされているDPDK互換NICのリストについては、「[Linuxホストのハードウェア要件](#)」を参照してください。



DPDKは、ネットワークパフォーマンスを向上させるために使用されるオープンソースのLinuxライブラリとネットワークインターフェースコントローラーのセットです。DPDKの詳細については、公式DPDKウェブサイト <https://www.dpdk.org/> を参照してください。

DPDKは、カーネルメモリをバイパスし、パケットを処理のためにユーザースペースメモリに直接配信するのに役立ちます。DPDKはLinux UIOモジュールと組み合わせることで、NetScaler BLXがカーネルメモリからユーザースペースメモリへのパケットコピーというLinuxカーネルのオーバーヘッドを伴わずにパケットを送受信できるようにします。メモリが割り当てられると、DPDKはバッファを管理してパフォーマンスを向上させます。

注:

以下のいずれかの条件が満たされた場合、NetScaler BLXはDPDKサポートなしで起動する場合があります。

- NetScaler BLXが専用のDPDK互換NICポートをサポートしていない。
- DPDKが専用のNICポートをサポートしていない。

サポートされているNetScaler®機能

NetScaler BLXはNetScalerのソフトウェアフォームファクターであり、他のNetScalerフォームファクターと同じ機能を提供します。

NetScalerの機能は、特定の要件に対応するために個別に、または組み合わせて構成できます。一部の機能は複数のカテゴリに当てはまりますが、多数のNetScaler機能は一般的に次のように分類できます。

- [アプリケーション切り替えおよびトラフィック管理機能](#)
- [アプリケーション高速化機能](#)
- [アプリケーションセキュリティおよびファイアウォール機能](#)
- [アプリケーション可視化機能](#)
- [NetScaler Gatewayアプリケーション](#)
- [スクリプト可能なモニターのサポート](#)

注記:

- NetScaler BLXに固有の機能制限については、「[NetScaler BLXの制限事項と使用ガイドライン](#)」を参照してください。
- NetScaler BLXでサポートされていない機能のリストについては、「[NetScaler BLXでサポートされていないNetScaler機能](#)」を参照してください。
- NetScalerリリース14.1ビルド25.x以降、NetScaler BLXはNetScaler Gatewayをサポートしています。詳細については、「[NetScaler Gatewayのドキュメント](#)」を参照してください。
- NetScalerリリース14.1ビルド34.x以降、BLXはDynamic Cluster Link Aggregation (DCLA)をサポートしています。詳細については、「[Dynamic Cluster Link Aggregation](#)」を参照してください。

NetScaler BLX ライセンス

NetScaler® は、お客様の組織の要件を満たす幅広い製品エディションとライセンスモデルを提供しています。

NetScaler BLX は、以下の種類のライセンスをサポートしています。

- **Express ライセンス** - デフォルトでは、NetScaler BLX には Express ライセンスが付属しており、ライセンスファイルは不要です。詳細については、「[NetScaler Express ライセンス](#)」を参照してください。
- **固定帯域幅サブスクリプション** - 固定帯域幅サブスクリプションは、期間ベースのライセンスであり、NetScaler BLX に許可される最大スループットを適用します。詳細については、「[NetScaler 固定帯域幅ライセンス](#)」を参照してください。
- **NetScaler プール容量ライセンス** - プール容量ライセンスを使用すると、NetScaler のさまざまなフォームファクター間で帯域幅またはインスタンスライセンスを共有できます。詳細については、「[NetScaler プール容量](#)」を参照してください。

NetScaler BLX のホスト ID を取得する

固定スループットライセンスおよび NetScaler プール容量ライセンスには、NetScaler BLX のホスト ID が必要です。NetScaler BLX のホスト ID を取得するには、NetScaler CLI で次のコマンドを実行します。

```
show hardware
```

出力例:



新しいライセンスをインストールする

NetScaler BLX にライセンスをインストールするには、自動または手動の方法を使用できます。

NetScaler BLX でのライセンスのインストールに関する詳細については、「[NetScaler ライセンスガイド](#)」を参照してください。

システム要件

NetScaler BLX を展開する前に、このドキュメントに記載されている要件を確認してください。

サポートされているLinuxディストリビューション

LINUXディストリビューション	NETSCALER BLX 14.1	NETSCALER BLX 13.1
レッドハット エンタープライズ リナックス (RHEL) 9.x	はい	はい
RHEL 8.x	はい	はい
RHEL 7.5～7.9	はい (ビルド 56.74 以前)	はい
オラクル Linux 8.x	はい	はい
オラクル Linux 7.5～7.9	はい (ビルド 56.74 以前)	はい
ウブントゥ 22.04	はい (ビルド 12.30 以降)	いいえ
ウブントゥ 20.04	はい	はい
ウブントゥ 18.04	はい	はい
オラクルクラウド インフラストラクチャ (OCI) 上のオラクルクラウド リナックス	はい	はい
アマゾン Linux 2	はい	はい

LINUXディストリビューション	NETSCALER BLX 14.1	NETSCALER BLX 13.1
ロッキー Linux 8	はい	いいえ

注:

- Amazon Linux では、DPDK を使用する NetScaler BLX は Elastic Network Adapter (ENA) でのみサポートされています。
- ホストシステムに GNU C Library (GLIBC) バージョン 2.23 より前のものがある場合、NetScaler BLX のインストールはブロックされます。NetScaler リリース 14.1.60.x 以降、RHEL 7.5 から 7.9 および Oracle Linux 7.5 から 7.9 ディストリビューションは、最小 GLIBC 要件を満たさないためサポートされません。

Linux ホストのハードウェア要件

カテゴリ	DPDK を使用する NETSCALER BLX のサポート	NETSCALER BLX
プロセッサ	Intel または AMD x86-64 (64 ビット) プロセッサ	Intel または AMD x86-64 (64 ビット) プロセッサ
最小RAM	2 GB	2 GB
ドライバー	Amazon EC2 エラスティックネットワークアダプタ ポート: <code>ena</code>	すべての Linux サポート対象ドライバー
	グーグル クラウド プラットフォーム: <code>gVNIC</code> , <code>virtIO</code>	
	Intel ポート: <code>igb</code> 、 <code>ixgbe</code> 、 <code>i40e</code> 、および <code>E810 (ice)</code>	
	KVM準仮想化ドライバー: <code>virtIO</code>	
	Mellanoxポート: ConnectX-4、ConnectX-5、	

カテゴリ	DPDK を使用する NETSCALER BLX のサポート	NETSCALER BLX
	およびConnectX-6 (<code>mlx5_core</code>)	
	VMware仮想化プラットフォームネットワークアダプターポート: <code>vmxnet3</code>	
仮想機能	インテル XL710	—
	インテル 82599	
	Mellanox ConnectX-5 SR-IOV インターフェース	
	Mellanox ConnectX-6 SR-IOV インターフェース	

注記:

- NetScaler BLXは、最大9つのNICポート（DPDK NICポート、非DPDK NICポート、またはその両方の組み合わせ）をサポートします。
- Linuxホストは、DPDKをインストールするための最小システム要件を満たしている必要があります。DPDKの最小要件の詳細については、[公式DPDKドキュメント](#)を参照してください。
- NetScalerリリース14.1ビルド43.x以降、NetScaler BLXはDPDKモードでIPv6をサポートします。サポートされているドライバーは、`virtio_net`、`i40e`、`ixgb`、`igb`、`af_packet`、`vmxnet`、および`mlx5_core`（物理NIC用）です。
- NetScalerリリース14.1ビルド72.x以降、NetScaler BLXはIntel E810 (ice) NICをサポートします。

オープンソースパッケージ

Linuxホストには以下の依存関係が必要です。

- 自動インストールされる依存関係
- 手動でインストールされた依存関係

自動インストールされる依存関係

NetScaler BLX をインストールすると、標準の公開リポジトリから以下の依存関係が Linux ホストに自動的にインストールされます。

RPM ベースの LINUX ホスト	DEBIAN ベースの LINUX ホスト
<code>crontabs</code>	<code>build-essential:amd64</code>
<code>ethtool</code>	<code>coreutils</code>
<code>gcc</code>	<code>cpanminus:amd64</code>
<code>glibc</code> (x86-32) (バージョン 2.17 以降)	<code>cron:amd64</code>
<code>glibc</code> (x86-64) (バージョン 2.17–196 以降)	<code>ethtool</code>
<code>Requpsd</code>	<code>gcc:amd64</code>
<code>ibdb</code> (x86-64) (バージョン 5.3–21 以降)	<code>lib32gcc1:amd64</code> (バージョン 4.9 以降)
<code>libgcc</code> (x86-32) (バージョン 4.8.5 以降)	<code>lib32stdc++6:amd64</code> (バージョン 4.8.4 以降)
<code>libstdc++</code> (x86-32) (バージョン 4.8.5 以降)	<code>lib32z1:amd64</code> (バージョン 1.2.8 以降)
<code>libstdc++</code> (x86-64) (バージョン 4.8.5 以降)	<code>libbsd0</code> (バージョン 0.8.2 以降)
<code>make</code>	<code>libc6</code> (バージョン 2.17 以降)
<code>openssl-devel</code>	<code>libc6:amd64</code> (バージョン 2.19 以降)
<code>pciutils</code>	<code>libc6:i386</code> (バージョン 2.19 以降)
<code>perl</code>	<code>libc6-i386:amd64</code> (バージョン 2.19 以降)
<code>perl-App-cpanminus</code>	<code>libdb5.3:amd64</code> (バージョン 5.3.28 以降)
<code>perl-core</code>	<code>libhttp-message-perl:amd64</code>
<code>perl-CPAN</code>	<code>libio-socket-ssl-perl:amd64</code>
<code>perl-IO-Socket-SSL</code>	<code>liblwp-protocol-https-perl:amd64</code>

RPM ベースの LINUX ホスト	DEBIAN ベースの LINUX ホスト
<code>perl-libwww-perl</code>	<code>libnuma1</code> (バージョン 2.0.11 以降)
<code>perl-LWP-Protocol-https</code>	<code>libssl-dev:amd64</code>
<code>perl-Sys-Syslog</code> (x86-64) (バージョン 0.33 以降)	<code>libstdc++6</code> (バージョン 4.8.5 以降)
<code>perl</code> (x86-64) (バージョン 5.16.3 以降)	<code>libswitch-perl:amd64</code>
<code>perl-XML-Writer</code>	<code>libwww-perl:amd64</code>
<code>ping</code>	<code>libxml-parser-lite-perl:amd64</code>
<code>procps-ng</code>	<code>libxml-writer-perl:amd64</code>
<code>python3</code>	<code>perl:amd64</code> (バージョン 5.16.3 以降)
<code>rsyslog</code>	<code>iputils-ping</code>
<code>sqlite-devel</code> (x86-64) (バージョン 3.7.17 以降)	<code>python3</code>
<code>systemd</code>	<code>rsyslog</code>
<code>tcpdump</code> (x86-64) (バージョン 4.9.2 以降)	<code>sqlite3:amd64</code> (バージョン 3.11.0 以降)
<code>zlib</code> (x86-32) (バージョン 1.2.7 以降)	<code>systemd</code>
	<code>tcpdump:amd64</code> (バージョン 4.9.2 以降)

手動でインストールされる依存関係

以下の依存関係は、標準の公開リポジトリからLinuxホストに手動でインストールする必要があります。

- RPMベースのLinuxシステムの場合、Extra Packages for Enterprise Linux (EPEL) リポジトリをインストールします

EPELリポジトリのインストールについては、[EPEL](#)を参照してください。

- Ubuntuバージョン18以降を実行しているDebianベースのLinuxシステムの場合、`libc6:i386` リポジトリをインストールします

パッケージをインストールするには、Linuxシェルで次のコマンドを実行します。

- `dpkg --add-architecture i386`
- `apt update`
- `apt install libc6:i386`

- **AWS上のAmazon Linux 2の場合、EPELリポジトリをインストールします**

EPELリポジトリをインストールするには、Amazon Linux 2シェルで次の3つのコマンドを実行します。

- a. `amazon-linux-extras install epel -y`
- b. `yum-config-manager --enable epel`
- c. `yum update`

EPELリポジトリのインストールに関する詳細については、[AWS公式ドキュメント](#)を参照してください。

NetScaler BLX を展開する

Linux ホスト上での NetScaler BLX の展開は、以下の手順で構成されます。

1. Linux ホストに NetScaler BLX パッケージをダウンロードする(/ja-jp/blx/current-release/deploy-blx/download-blx.html)。
2. Linux ホストに NetScaler BLX をインストールします。
 - Debian ベースの Linux ホストに NetScaler BLX をインストールする(/ja-jp/blx/current-release/deploy-blx/install-blx-debian.html)。
 - RPM ベースの Linux ホストに NetScaler BLX をインストールする(/ja-jp/blx/current-release/deploy-blx/install-blx-rpm.html)。

Linux ホストに NetScaler BLX パッケージをダウンロードする

NetScaler BLX インストールパッケージは、ダウンロードページでホストされています。インストールパッケージは、NetScaler 機能パッケージと DPDK サポート用パッケージで構成されています。

NetScaler BLX インストールパッケージは TAR ファイルであり、以下の命名規則に従います。

- Debian ベースのパッケージの場合、 `blx-deb-<release number>-<build-number>.tar.gz`。
- RPM ベースのパッケージの場合、 `blx-rpm-<release number>-<build-number>.tar.gz`。

例:

```
blx-deb-14.1-4.42.tar.gz
```

```
blx-rpm-14.1-4.42.tar.gz
```

NetScaler BLX をダウンロードする

1. Web ブラウザで [ダウンロード](#) ページを開きます。
2. ダウンロードページで、ダウンロードしたい **NetScaler BLX リリース** を展開します。
3. NetScaler BLX ビルドリンクをクリックします。

4. **ファイルのダウンロード** をクリックして、NetScaler BLX ビルドパッケージをダウンロードします。

注:

チェックサムは、ダウンロードしたビルドパッケージがウェブサイトでホストされている実際のパッケージと一致することを確認するために提供されます。チェックサムは、正しいビットであることを確認するための重要なチェックです。

次のステップ

- [DebianベースのLinuxホストにNetScaler BLXをインストールする](#)
- [RPMベースのLinuxホストにNetScaler BLXをインストールする](#)

DebianベースのLinuxホストにNetScaler BLXをインストールする

始める前に

- NetScaler BLXパッケージがLinuxホストで利用可能であることを確認します。NetScaler BLXのダウンロード方法については、「[LinuxホストにNetScaler BLXパッケージをダウンロードする](#)」を参照してください。
- 必要な依存関係をインストールするために、Linuxホストがインターネットにアクセスできることを確認します。自動インストールされる依存関係の詳細については、「[NetScaler BLXの自動インストールされる依存関係](#)」を参照してください。
- LinuxホストにNetScaler BLXをインストールするためのrootまたはsudo権限があることを確認します。

NetScaler BLXをインストールする

1. NetScaler BLXインストールパッケージを解凍し、作業ディレクトリを抽出されたNetScaler BLXインストールディレクトリに変更します。

```
tar -xvzf blx-deb-<release number>-<build-number>.tar.gz
```

```
cd <path to the extracted NetScaler BLX installation directory>
```

出力例:

次の出力例は、Linuxホストの `/var/blxinstall` ディレクトリにダウンロードされた NetScaler BLXインストールパッケージ `blx-deb-14.1-4.42.tar.gz` が解凍されたことを示しています。その後、作業ディレクトリは抽出されたディレクトリ `blx-deb-14.1-4.42` に変更されます。

```
# cd /var/blxinstall

# tar -xvzf blx-deb-14.1-4.42.tar.gz

# cd blx-deb-14.1-4.42

# pwd

/var/blxinstall/blx-deb-14.1-4.42
```

2. NetScaler BLXをインストールするには、次のコマンドを実行します。

```
apt install ./blx*.deb
```

注記:

Ubuntuバージョン18以降を実行しているDebianベースのLinuxホストでは、次の依存関係エラーによりNetScaler BLXのインストールが失敗する可能性があります。

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19) but it is not installable
```

回避策: NetScaler BLXをインストールする前に、LinuxホストのCLIで次のコマンドを実行します。

- `dpkg --add-architecture i386`
- `apt update`

- `apt install libc6:i386`

3. これはオプションの手順です。カスタムモニター機能を使用するには、`nsall.pl` および `nsntlm-lwp.pl` スクリプトを使用する前に、システムにComprehensive Perl Archive Network (CPAN)を使用してperl(Authen::NTLM)を手動でインストールする必要があります。

```
# cpan install Authen::NTLM
```

4. 次のコマンドを実行して、NetScaler BLXのステータスを確認します。

```
systemctl status blx
```

デフォルトでは、NetScaler BLXは非アクティブ状態です。

LinuxホストからNetScaler BLXをアンインストールするには：

- Linuxホストで `apt remove blx` コマンドを実行してNetScaler BLXをアンインストールし、NetScaler BLX構成ファイルを保持します。
- Linuxホストで `apt purge blx` コマンドを実行してNetScaler BLXをアンインストールし、NetScaler BLX構成ファイルを削除します。

注:

Linuxホストは、一部のNetScaler BLX関連システムファイルが削除されていないという警告メッセージを表示する場合があります。しかし、`apt purge blx` コマンドを実行すると、すべてのNetScaler BLXファイルが削除されます。

次のステップ

- [NetScaler BLXを構成する](#)

RPMベースのLinuxホストにNetScaler BLXをインストールする

始める前に

- NetScaler BLXパッケージがLinuxホストで利用可能であることを確認してください。
NetScaler BLXのダウンロード方法については、「[LinuxホストにNetScaler BLXパッケージをダウンロードする](#)」を参照してください。

- Linuxホストが必要な依存関係をインストールするためにインターネットにアクセスできることを確認してください。自動インストールされる依存関係の詳細については、「[NetScaler BLXの自動インストールされる依存関係](#)」を参照してください。
- LinuxホストにNetScaler BLXをインストールするためのrootまたはsudo権限があることを確認してください。
- NetScaler BLXバージョン14.1ビルド17.x以降では、Red HatベースのLinuxホストにNetScaler BLXをインストールすると、SELinuxモジュールがLinuxホストで利用可能な場合、SELinuxポリシーが適用されます。このポリシーにより、NetScaler BLXはLinuxホスト上で実行できません。SELinuxポリシーの詳細については、「[SELinuxポリシー](#)」を参照してください。

NetScaler BLXをインストールする

1. NetScaler BLXインストールパッケージを解凍し、作業ディレクトリを解凍されたNetScaler BLXインストールディレクトリに変更します。

```
tar -xvzf blx-rpm-<release number>-<build-number>.tar.gz

cd <path to the extracted NetScaler BLX installation directory>
```

出力例:

以下の出力例は、Linuxホストの `/var/blxinstall` ディレクトリにすでにダウンロードされているNetScaler BLXインストールパッケージ `blx-rpm-14.1-4.42.tar.gz` が解凍されたことを示しています。その後、作業ディレクトリは解凍されたディレクトリ `blx-rpm-14.1-4.42` に変更されます。

```
# cd /var/blxinstall

# tar -xvzf blx-rpm-14.1-4.42.tar.gz

# cd blx-rpm-14.1-4.42
```

```
# pwd

/var/blxinstall/blx-rpm-14.1-4.42
```

2. Red HatベースのLinuxシステムの場合、Extra Packages for Enterprise Linux (EPEL) をインストールします。EPELのインストール方法の詳細については、「[EPELドキュメント](#)」を参照してください。
3. これはオプションの手順です。カスタムモニター機能を使用するには、`nsall.pl` および `nsntlm-lwp.pl` スクリプトを使用する前に、システムにComprehensive Perl Archive Network (CPAN) を使用してperl(Authen::NTLM)を手動でインストールする必要があります。

```
# cpan install Authen::NTLM
```

4. NetScaler BLXをインストールするには、次のコマンドを実行します。

```
yum install ./blx*.rpm
```

5. 次のコマンドを実行して、NetScaler BLXのステータスを確認します。

```
systemctl status blx
```

デフォルトでは、NetScaler BLXは非アクティブ状態です。

LinuxホストからNetScaler BLXをアンインストールするには:

NetScaler BLXをアンインストールするには、Linuxホストで `yum remove blx` コマンドを実行します。

次のステップ

- [NetScaler BLXを構成する](#)

NetScaler BLX を構成する

NetScaler BLX 専用の Linux ホストの NIC ポートは、Linux ホスト上の他のアプリケーションと共有されません。

高いパケット処理性能のために、DPDK互換NICの使用を推奨します。NetScaler BLXでサポートされているDPDK互換NICのリストについては、[Linuxホストのハードウェア要件](#)を参照してください。

`blx.conf` ファイルで次のパラメータを指定することにより、NetScaler BLXを構成できます。

- `worker-processes` - 起動するワーカプロセスの数。
- `interfaces` - NetScaler BLX専用のLinuxホストのNICポート。
- `ipaddress` - NetScaler BLXのNSIPアドレス。
- `default` - NetScaler BLXのデフォルトルート。

`interfaces` パラメータで指定されたNICポートがDPDK互換であり、NetScaler BLXでサポートされている場合、NICポートは自動的にDPDK VFIOモジュールにバインドされます。NICポートがDPDK互換でない場合、NICポートは非DPDKポートとして追加されます。NetScaler BLXを起動すると、`interfaces` パラメータで指定されたすべてのポートが、NetScaler BLXの専用ポートとして追加されます。

注：

NetScaler BLXは、一度に1種類のDPDK NICポートのみをサポートします。たとえば、すべてのMellanoxポートまたはすべてのIntelポートのいずれかです。

前提条件

- LinuxホストでIOMMUサポートが有効になっていることを確認してください。IOMMUを有効にする方法については、Linuxホストのハードウェアドキュメントを参照してください。
- NetScaler BLXでサポートされているDPDK互換のMellanoxポートの場合、Mellanox OpenFabrics Enterprise Distribution (OFED) パッケージがLinuxホストにインストールされていることを確認してください。Mellanox OFEDパッケージのインストール方法については、[OFEDドキュメント](#)を参照してください。

NetScaler BLX を構成する

NetScaler BLX を構成するには、Linux ホスト CLI を使用する必要があります。

1. 次のコマンドを実行して、`blx.conf` ファイルを開きます。

注:

任意のテキストエディタを使用して `blx.conf` ファイルを編集できます。

```
nano /etc/blx/blx.conf
```

2. `worker-processes` パラメータのコメントを解除し、NetScaler BLX のパケットエンジンの数を指定します。

注:

NetScaler BLX でサポートされている VMXNET3 DPDK ポートの場合、ワーカースレッドの数を 2 の累乗 (2^n) で指定する必要があります。たとえば、1、2、4、8 などです。

```
blx-system-config

{
    ...

    worker-processes: <number of worker processes>

    ...
}
```

3. `interfaces` パラメータのコメントを解除し、NetScaler BLX に割り当てる Linux ホストの NIC ポートを指定します。

注:

- Linux ホスト CLI に表示されているポート名をスペースで区切って指定する必要があります。
- **AMD プロセッサの場合**、1 つ以上の IOMMU グループのすべての DPDK NIC ポートを指定する必要があります。IOMMU グループのすべての NIC ポートを指定しない場合、その IOMMU グループの DPDK 互換 NIC ポートは、非 DPDK 専用ポートとして NetScaler BLX に追加されます。

```
blx-system-config
{
    ...

    interfaces: <interface1 interface2>

    ...
}
```

4. **ipaddress** パラメータのコメントを解除し、NetScaler BLX の NSIP アドレスを指定します。

```
blx-system-config
{
    ...

    ipaddress: <IP address>

    ...
}
```

5. (オプション) `total-hugepage-mem` パラメータのコメントを解除し、DPDK Huge pages に割り当てるメモリを指定します。DPDK Huge pages の詳細については、[DPDK ドキュメント](#) を参照してください。

注:

Huge pages の合計サイズは、`megabytes (MB or M)` または `gigabytes (GB or G)` で指定できます。たとえば、1024MB、1024M、1GB、1G などです。

```
blx-system-config
{
    ...

    total-hugepage-mem: <memory size>

    ...
}
```

6. (オプション) デフォルトのnsrootユーザーを非rootユーザーに変更するには、`non-root-nsroot: yes` パラメーターを追加します。

非root nsrootモードでNetScaler BLXを起動した後、root nsrootユーザーに戻したり、CLI経由で `showtechsupport` コマンドを実行したりすることはできません。

詳細については、[NetScaler BLXの制限事項と使用ガイドライン](#)を参照してください。

7. `default` パラメーターのコメントを解除し、専用インターフェイスのデフォルトルートを設定します。

```
static-routes
{
    ...

    default <gateway IP address>

    ...
}
```

```
}
```

8. `blx.conf` ファイルを保存します。

次のステップ

- [NetScaler BLXを起動する。](#)

NetScaler BLX 設定ファイル

NetScaler BLX のインストールの一部として、設定ファイル (`blx.conf`) が Linux ホストに追加されます。 `blx.conf` ファイルには、NetScaler BLX の設定に使用できるさまざまなパラメータがあります。

デフォルトでは、NetScaler BLX 設定ファイルのすべてのパラメータはコメントアウトされています (`#` 記号がプレフィックスとして付いています)。パラメータのコメントを解除 (`#` プレフィックスを削除) し、カスタム値を設定することで、特定のパラメータを有効にできます。

`blx.conf` ファイルのパラメータを次の表に示します。

パラメータ	設定可能な値	デフォルト	説明
<code>worker-processes</code>	1~28	1	起動するワーカプロセスの数。
<code>cpu-yield</code>	yes、enable、または 1	無効	CPUイールドを設定します。CPUイールドを有効にすると、NSPPEは他のプロセスにCPUを譲ります。
<code>core-dumps</code>	yes、enable、または 1	無効	NetScaler BLXのコアダンプを設定します。
<code>syslog</code>	yes、enable、または 1	無効	Linuxホストのポート 514/UDPでsyslogがリッスンできるようにします。NetScaler BLXは、Linuxホスト

パラメータ	設定可能な値	デフォルト	説明
			のポート514/UDPでリスンしている syslog にログを送信します。
<code>ipaddress</code>	IPアドレス	NetScaler BLXは、Linuxホストに設定されているすべてのIPアドレスでリスンします	NetScaler BLXの NSIPアドレスを設定します。
<code>blx-managed-host</code>	1	無効	NetScaler BLX を介した Linux ホストへの SSH アクセスを設定します。
<code>host-ipaddress</code>	IPアドレス	なし	NetScaler BLX を介して Linux ホストへの SSH アクセスを許可する IP アドレスを設定します。 注: このパラメーターは <code>blx-managed-host</code> パラメーターと組み合わせて使用する必要があります。
<code>total-hugepage-mem</code>	最小: 1G、最大: Linux ホストで利用可能な容量	1G	NetScaler BLX の DPDK Huge Page メモリを設定します。
<code>interfaces</code>	Linux ホスト CLI に表示される NIC ポート名	NetScaler BLX は、すべてのインターフェースとホストトラフィックを共有します。	Linux ホストの指定された NIC ポートは NetScaler BLX 専用です。
<code>default</code>	IPアドレス	なし	専用インターフェースのデフォルトルートを設定します。

パラメータ	設定可能な値	デフォルト	説明
<code>cli-cmds</code>	NetScaler CLIコマンド	なし	NetScaler BLXの起動時に実行するNetScaler CLIコマンドをリストします。
<code>nsdrvd</code>	1、2、または3	無効	パケット処理性能を向上させるために <code>nsdrvd</code> ドライバーを設定します。

注記：

パラメータを無効にする場合は、`blx.conf` ファイル内のパラメータを `#` でコメントアウトし、`systemctl restart blx` コマンドを使用してNetScaler BLXを再起動します。変更はNetScaler BLXの再起動後に適用されます。

NetScaler BLXにおける起動時のカスタマイズスクリプト

NetScaler BLXは、NetScaler BLXの起動前後に実行されるスクリプトを介して、起動時のカスタマイズメカニズムを提供します。特定のニーズに合わせてNetScaler BLX環境を調整するには、`nsbefore.sh` および `nsafter.sh` という名前の起動スクリプトを作成します。これらの起動時のカスタマイズスクリプトを使用することで、タスクを自動化し、構成標準を適用し、特定のユースケースに合わせてNetScaler BLX環境を最適化できます。

- **nsbefore.sh:** - このスクリプトはNetScaler BLXの起動前に実行されます。一般的なユースケースは次のとおりです。
 - 依存するサービスに必要な環境変数を設定する。
 - カスタム設定を適用するために構成ファイルを変更する。
 - NetScaler BLXが依存する前提サービスを開始する。
- **nsafter.sh:** - このスクリプトはNetScaler BLXが完全に初期化された後に実行されます。一般的なユースケースは次のとおりです。
 - NetScaler BLXが正しく実行されていることを確認するためにシステムヘルスチェックを実行する。
 - ログローテーションやアーカイブなど、ログファイルを管理する。

- NetScaler BLX インスタンスに依存する追加のアプリケーションまたはサービスを起動する。

起動スクリプトの構成

1. NetScaler BLX が起動時にそれらを認識して実行できるように、`/nsconfig/` ディレクトリに `nsbefore.sh` および `nsafter.sh` スクリプトを作成します。
2. スクリプトが実行に必要な権限を持っていることを確認します (例: `chmod +x nsbefore.sh`)。
3. これらのスクリプト内で行われた変更は、本番環境に展開する前に、管理された環境で徹底的にテストしてください。

NetScaler BLX の圧縮コアダンプを設定する

NetScaler BLX のコアダンプは、`blx.conf` ファイル内の `core-dumps` パラメータを使用して有効にできます。

コアダンプは、Linux ホスト上の `core_pattern` ファイル内のパターンに従って生成されます。

```
/proc/sys/kernel/core_pattern
```

`core_pattern` ファイルにパターンが存在しない場合、コアダンプ用に次のパターンがファイルに追加されます。

```
/var/core/core-%e-sig%s-user%u-group%g-pid%p-time%t
```

`blx.conf` ファイルを使用してコアダンプを有効にする

コアダンプを有効にするには、Linux ホストの CLI を使用する必要があります。

1. 次のコマンドを実行して、`blx.conf` ファイルを開きます。

注記:

`blx.conf` ファイルを編集するには、任意のテキストエディタを使用できます。

```
nano /etc/blx/blx.conf
```

2. `core-dumps` パラメータのコメントを解除し、`1`、`enable`、または `yes` に設定します。

```
blx-system-config
{
    ...
    core-dumps: yes
    ...
}
```

3. `blx.conf` ファイルを保存します。
4. NetScaler BLX を再起動します。

```
systemctl restart blx
```

NetScaler BLX の再起動後、NetScaler BLX のコアダンプが有効になります。

`blx.conf` ファイルを使用してコアダンプを無効にする

コアダンプを無効にするには、Linux ホストの CLI を使用する必要があります。

注記:

Linuxホストでコアダンプを有効にすると、`blx.conf` ファイルで `core-dumps` パラメータがコメントアウト（無効化）されていても、NetScaler BLXのコアダンプが生成されます。

1. 次のコマンドを実行して、`blx.conf` ファイルを開きます。

注記:

任意のテキストエディタを使用して、`blx.conf` ファイルを編集できます。

```
nano /etc/blx/blx.conf
```

2. `core-dumps` パラメータをコメントアウトします。

```
blx-system-config

{

    ...

    # core-dumps: yes

    ...

}
```

3. `blx.conf` ファイルを保存します。
4. NetScaler BLXを再起動します。

```
systemctl restart blx
```

NetScaler BLXが再起動すると、NetScaler BLXのコアダンプは無効になります。

NetScaler BLXマネージドホストの設定

NetScaler BLXマネージドホスト機能を使用して、NetScaler BLX経由でLinuxホストを管理できます。この機能は、LinuxホストのすべてのNICポートを専用ポートとしてNetScaler BLXに自動的に追加します。ポートがDPDK互換でNetScaler BLXでサポートされている場合、それらはLinuxホスト上のDPDK VFIOモジュールにバインドされます。

NetScaler BLXは、Linuxホスト上で最も優先順位の高いデフォルトルートを持つ専用NICポートの1つを選択します。選択されたポートのIPアドレスとデフォルトルートは、NetScaler BLXのNSIPアドレスおよびデフォルトルートとして追加されます。

Linuxホスト上のNICポートにデフォルトルートが設定されていない場合、NetScaler BLXはIPアドレスが割り当てられた専用ポートをランダムに選択します。選択されたポートのIPアドレスは、NetScaler BLXのNSIPアドレスとして追加されます。

デフォルトでは、LinuxホストへのSSHアクセスはNSIPアドレスのポート9022で有効になっています。

注記:

- NetScaler BLXはLinuxホストのボンダイインターフェイス（リンクアグリゲーションチャンネル）を自動的に追加しませんが、ボンダイインターフェイスのすべてのメンバーをNetScaler BLXに追加します。
- デフォルトポートに複数のIPアドレスが割り当てられている場合、NetScaler BLXはCLIにエラーメッセージを表示し、`blx.conf` ファイルでNSIPアドレスを手動で設定するよう促します。
- NetScaler BLXを再起動すると、LinuxホストへのすべてのアクティブなSSHセッションが閉じられます。接続を復元するには、ホストへの接続を再試行する必要があります。
- `blx.conf` ファイルでNSIPアドレスを手動で設定した場合、Linuxホストで利用可能なデフォルトルートはBLXに自動的に追加されません。
- `ns.conf` ファイルの構成は、`blx.conf` ファイルよりも優先されます。

LinuxホストへのSSHアクセスでNetScaler BLXマネージドホストを有効にする

NetScaler BLXマネージドホストを有効にするには、LinuxホストのCLIを使用する必要があります。

1. 次のコマンドを実行して、`blx.conf` ファイルを開きます。

注:

`blx.conf` ファイルを編集するには、任意のテキストエディターを使用できます。

```
nano /etc/blx/blx.conf
```

2. `blx-managed-host` パラメーターのコメントを解除し、1 に設定します。

```
blx-system-config
{
    ...
}
```

```

    blx-managed-host: 1

    ...

}

```

3. `blx.conf` ファイル内の他のパラメーターがコメントアウトされていることを確認します。
4. `blx.conf` ファイルを保存します。
5. NetScaler BLX を再起動します。

```
systemctl restart blx
```

NetScaler BLX の再起動後、SSH クライアントを使用して、以下の IP アドレスで Linux ホストと NetScaler BLX にアクセスできます。

- Linux ホスト - `<NSIP address>:9022`
- NetScaler BLX - `<NSIP address>:22`

ポート 22 で Linux ホストへの SSH アクセスが必要な場合は、NSIP と Linux ホストに異なる IP アドレスを手動で設定できます。詳細については、「[NSIP と Linux ホストに異なる IP アドレスを設定する](#)」を参照してください。

NSIP と Linux ホストに異なる IP アドレスを設定する

前のセクションで述べた構成に加えて、NSIP と Linux ホスト (ホスト IP) に異なる IP アドレスを設定するには、`ipaddress`、`default`、および `host-ipaddress` パラメーターを使用する必要があります。

ポート 22 で SSH アクセスを有効にするには、Linux ホストの CLI を使用する必要があります。

1. 次のコマンドを実行して、`blx.conf` ファイルを開きます。

注:

`blx.conf` ファイルを編集するには、任意のテキストエディターを使用できます。

```
nano /etc/blx/blx.conf
```

2. NetScaler BLXにアクセスしたい `ipaddress` パラメータのコメントを解除し、NSIPアドレスを指定します。

```
blx-system-config
{
    ...
    ipaddress: <IP address>
    ...
}
```

3. Linuxホストにアクセスしたい `host-ipaddress` パラメータのコメントを解除し、ホストIPアドレスを指定します。

注:

IPアドレスはNSIPサブネット内にある必要があります。

```
blx-system-config
{
    ...
    host-ipaddress: <IP address>
    ...
}
```

4. `default` パラメータのコメントを解除し、デフォルトルートを指定します。

```
static-routes
{
```

```
...  
  
default <gateway IP address>  
  
...  
  
}
```

5. `blx.conf` ファイルを保存します。

6. NetScaler BLXを再起動します。

```
systemctl restart blx
```

7. NetScaler BLXの再起動後、NetScaler CLIで `show nsip` コマンドを実行してホストIPアドレスを確認します。



SSHクライアントを使用して、次のIPアドレスでLinuxホストとNetScaler BLXにアクセスできます。

- Linuxホスト - `<Host IP address>:22`
- NetScaler BLX - `<NSIP address>:22`

DPDKサポートなしのNetScaler BLX向けnsdrvdドライバーの構成

NetScaler BLXはLinuxカーネルのネットワークスタックをバイパスしますが、Linuxカーネルメモリとユーザースペースメモリ間でパケットを転送する際にオーバーヘッドが発生します。このオーバーヘッドは、パケット処理の全体的なパフォーマンスに影響を与えます。

高いパケット処理パフォーマンスを得るには、DPDK互換NICの使用をお勧めします。NetScaler BLXでサポートされているDPDK互換NICのリストについては、[Linuxホストのハードウェア要件](#)を参照してください。

DPDK互換NICがない場合は、DPDKサポートなしでパケット処理のパフォーマンスを向上させるために、`nsdrvd` ドライバーを使用できます。

nsdrvd ドライバーは、パケットの受信と送信に関するLinuxカーネルとのすべてのやり取りを処理します。また、トラフィックをPEに分散します。**blx.conf** ファイル内の **nsdrvd** パラメーターを使用して、**nsdrvd** ドライバーを構成できます。次の表は、**nsdrvd** パラメーターの可能な値について説明しています。

可能な値	説明
nsdrvd: 1	専用ポートごとに1つのドライバプロセスが作成されます。RxとTxは順次実行されます。
nsdrvd: 2	専用ポートごとに1つのRxプロセスと1つのTxスレッドが作成されます。
nsdrvd: 3	専用ポートごとに1つのRxプロセスと2つのTxスレッドが作成されます。

nsdrvd ドライバーを使用するには、Linuxホストは以下の計算に基づいて、少なくともn個のコアを持っている必要があります。

$$n \geq WP + (INT * P) + 1$$

説明:

- **WP** - NetScaler BLXのワーカープロセス（パケットエンジン）の数。**blx.conf** ファイル内の **worker-processes** パラメータは、NetScaler BLXのパケットエンジンの数を指定します。
- **INT** - NetScaler BLX専用のLinuxホストNICポートの数。**blx.conf** ファイル内の **interface** パラメータは、NetScaler BLX専用のLinuxホストNICポートを指定します。
- **P** - NetScaler BLXの **nsdrvd** ドライバプロセスの数。**blx.conf** ファイル内の **nsdrvd** パラメータは、**nsdrvd** ドライバプロセスの数を指定します。

例: 以下の構成を持つNetScaler BLXには、少なくとも10個のコアが必要です。

- **WP** = 3つのパケットエンジン
- **INT** = 2つの専用インターフェース
- **P** = 3つの **nsdrvd** プロセス

$$n = WP + (INT * P) + 1 = (3 + 2 * 3 + 1) = 10$$

前提条件

- 専用NICポートが、NetScaler BLXでサポートされているDPDK互換NICにリストされていないことを確認してください。詳細については、「[Linuxホストのハードウェア要件](#)」を参照してください。

nsdrvd ドライバを有効にする

nsdrvd ドライバを有効にするには、LinuxホストCLIを使用する必要があります。

1. 次のコマンドを実行して、**blx.conf** ファイルを開きます。

注記:

blx.confファイルを編集するには、任意のテキストエディターを使用できます。

```
nano /etc/blx/blx.conf
```

2. **nsdrvd** パラメータのコメントを解除し、値を 1、2、または 3 に設定します。

```
blx-system-config  
  
{  
  
    ...  
  
    nsdrvd: <number of process>  
  
    ...  
  
}
```

3. **blx.conf** ファイルを保存します。
4. NetScaler BLX を再起動します。

```
systemctl restart blx
```

NetScaler BLX の再起動後、NetScaler BLX で **nsdrvd** ドライバーが有効になります。

nsdrvd ドライバーを無効にする

nsdrvd ドライバーを無効にするには、Linux ホスト CLI を使用する必要があります。

1. **nsdrvd** ドライバーを無効にするには、**blx.conf** ファイル内の **nsdrvd** パラメータをコメントアウトします。

```
blx-system-config  
  
{  
  
    ...  
  
    # nsdrvd: 2  
  
    ...  
  
}
```

2. NetScaler BLX を再起動します。

```
systemctl restart blx
```

NetScaler BLX の再起動後、NetScaler BLX で **nsdrvd** ドライバーが無効になります。

NetScaler BLXとFastly Next-Gen WAFエージェントを統合する

NetScaler BLXは、Fastly Next-Gen WAF（旧称Signal Sciences）エージェントを単一のLinux OS ホスト上で連携させることで、アプリケーション配信とセキュリティのための統合されたエンタープライズグレードのソリューションを提供します。このソリューションは、密接に結合された高性能なWAFソリューションを提供することで、顧客の重要な要件を直接満たします。この独自の設計により、複雑なマルチVMや代替ベンダーソリューションと比較して、低遅延が保証され、展開が劇的に簡素化され、総所有コスト（COGS）が大幅に削減されます。最高のパフォーマンスを得るために、このソリューションは通常、ベアメタルサーバーに展開され、DPDKテクノロジーを使用して高スループットと低遅延を保証します。

主な利点

この統合ソリューションは以下を提供します。

- **最適化されたパフォーマンスと低遅延:** NetScaler BLXとFastly Next-Gen WAFエージェントは、最大の効率を得るために同じサーバー上で動作します。
- **コスト削減:** 同居により、Fastlyエージェント用の個別のVMが不要になります。
- **展開と管理の簡素化:** NetScaler BLXは、同じサーバー上の他のアプリケーションと共存できます。
- **セキュリティの強化:** Fastly Next-Gen WAFの保護（DDoS、SQLインジェクション、XSS）とNetScalerの機能を組み合わせます。

アーキテクチャの概要

次の図は、Fastly NextGen WAFとNetScaler BLXの統合アーキテクチャを示しています。

Fastly WAFとNetScaler BLXのアーキテクチャ(/en-us/blx/media/blx-waf-architecture-diagram.png)

Fastly NextGen WAFエージェントと統合するようにNetScaler BLXを構成する方法

1. Fastlyエージェントをインストールします。

詳細な手順については、[Fastlyエージェントインストールガイド](#)を参照してください。

2. NetScaler BLXホスト上でFastlyエージェントを構成します。
 - a. `/etc/sigsci/agent.conf` ファイルを、[Fastly エージェントセットアップガイド](#) に記載されているとおりに編集します。
 - b. 次のパラメータを追加します。

```
accesskeyid = "<replace this with accesskeyid>"

secretaccesskey = "<replace this with secretaccesskey>"

rpc-address = "tcp:0.0.0.0:5555"
```

3. Fastly エージェントをNetScaler BLXと統合します。
 - a. 負荷分散仮想サーバーとサービスを設定します。
 - b. Web Assembly (WASM) モジュールを作成し、リクエストトラフィックとレスポンストラフィックの両方に対してコンテンツ検査プロファイル (`wasmProfiles` と呼ばれる) を追加します。
 - c. Fastly エージェントのサービスを追加します。

- d. リクエストとレスポンスのコンテンツ検査アクションを追加します。
- e. リクエストとレスポンスのコンテンツ検査ポリシーを追加します。
- f. リクエストおよびレスポンスポリシーを負荷分散仮想サーバーにバインドします。

設定例:

HTTP 負荷分散仮想サーバーの名前が v1 で、サービスの名前が s1 の場合は、次のコマンドを使用します。

```
add wasm module ns_fastly -moduleFile ns_sigsci.wasm -signatureFile  
ns_sigsci_wasm.sig  
  
add contentinspection wasmProfile ciresprof -maxBodyLen 0 -wasmModule  
ns_fastly  
  
add contentinspection wasmProfile cireqprof -wasmModule ns_fastly  
  
add service fastly_agent 192.0.0.3 TCP 5555  
  
add contentinspection action ciresact -type WASM -wasmProfileName  
ciresprof -serverName fastly_agentthec
```

NetScaler BLX の起動

NetScaler BLX は、Linux ホスト上で動作するアプリケーションです。NetScaler BLX をインストールして構成した後、Linux ホスト CLI で次のコマンドを実行して NetScaler BLX を起動する必要があります。

```
systemctl start blx
```

注:

- NetScaler BLX の起動には最大 45 秒かかる場合があります。
- CentOS バージョン 8.x または Oracle Linux バージョン 8.x に展開された NetScaler BLX バージョン 14.1 ビルド 12.35 以前では、Linux ホストで **SELinux** ポリシーが有効になっている場合、正常に起動または機能しない可能性があります。

回避策: Linux ホストで **SELinux** を無効にします。

- a. コマンド `nano /etc/selinux/config` を使用して、Linux ホスト上の **SELinux** 設定ファイルを開きます。
- b. **SELINUX=disabled** を設定し、ファイルを保存します。
- c. コマンド `systemctl restart blx` を使用して NetScaler BLX を再起動します。

NetScaler BLX のステータスを確認する

- Linux ホストの CLI で次のコマンドを実行して、NetScaler BLX のステータスを確認できます。

```
systemctl status blx
```

NetScaler BLX のステータスは **active (exited)** である必要があります。



- Linux ホストで実行されている NetScaler® プロセスを確認するには、Linux ホストの CLI で次のコマンドを実行します。

```
ps aux | grep ns
```

注:

Linuxホストで `nspe` プロセスが実行されていることを確認します。

例: `root 68332 2.7 5.5 485264 442084 ? Ss 16:25 0:02 /usr/sbin/nspe 1`

NetScaler BLXを停止する

NetScaler BLXと関連プロセスを停止するには、LinuxホストのCLIで次のコマンドを実行します。

```
systemctl stop blx
```

注:

Linuxホストを再起動すると、再起動前に停止していた場合でも、Linuxホストの再起動後にNetScaler BLXが自動的に起動します。Linuxホストの再起動後にNetScaler BLXが自動的に起動しないようにするには、`systemctl disable blx` コマンドを実行する必要があります。

次のステップ

- [NetScaler BLXにアクセスしてNetScaler機能を構成する](#)

NetScaler BLX にアクセスし、NetScaler 機能を構成する

以下のいずれかの方法で NetScaler BLX にアクセスできます。

- NetScaler® CLI
- NetScaler GUI
- NetScaler NITRO REST API

開始する前に

- NetScaler BLX が Linux ホストで稼働していることを確認してください。NetScaler BLX の起動方法の詳細については、「[NetScaler BLX の起動](#)」を参照してください。

パスワードの要件

- 初めてデフォルトの管理者 (`nsroot`) パスワードでログインすると、NetScaler BLX はセキュリティ上の理由からパスワードの変更を促します。パスワードを変更した後、設定を保存する必要があります。設定が保存されずに NetScaler BLX が再起動した場合、再度デフォルトのパスワードでログインする必要があります。
- NetScaler BLX では、すべてのローカルシステムユーザーに対して、強力なパスワードの適用がデフォルトで有効になっています。強力なパスワードのデフォルトの最小長は 4 文字です。強力なパスワードには、以下が含まれている必要があります。
 - 小文字を 1 文字。
 - 大文字を 1 文字。
 - 数字を 1 文字。
 - セット (`!`、`@`、`#`、`(`、`)`、`$`、`%`、`\^`、`&`、および `*`) から特殊文字を 1 文字。

注:

NetScaler BLXの各システムユーザーのパスワードが、強力なパスワード基準に合致していることを確認してください。

- 強力なパスワード基準の詳細については、[NetScalerでパスワードの複雑さを強制する方法](#)を参照してください。

NetScaler BLXにアクセスし、NetScaler CLIを使用してNetScaler機能を構成する

NetScaler BLXにはコマンドラインインターフェース (CLI) があり、NetScaler CLIコマンドを実行してNetScaler BLXのNetScaler機能を構成できます。

ワークステーションからセキュアシェル (SSH) を介して接続することで、NetScaler BLXにリモートでアクセスできます。

NetScaler CLIを使用してNetScaler BLXにアクセスするには：

1. ワークステーションからSSHクライアントを開きます。
2. NetScaler BLXのCLIが利用可能なIPアドレスとポートを指定し、CLIに接続します。
3. NetScaler BLXのログイン資格情報を使用してNetScaler BLXにログインします。

NetScaler CLIコマンドの詳細については、[NetScalerコマンドリファレンスガイド](#)を参照してください。

NetScaler BLXにアクセスし、NetScaler GUIを使用してNetScaler機能を構成する

NetScaler GUIには、構成ユーティリティとダッシュボードユーティリティが含まれています。

次の表に、NetScaler GUIが利用可能なデフォルトのIPアドレスとポートを示します。

アクセスタイプ	NETSCALER GUIにアクセスするためのIPアドレスとポート
HTTP	<NetScaler IP address (NSIP)>:80
HTTPS	<NetScaler IP address (NSIP)>:443

これらのデフォルトポート番号は `blx.conf` ファイルで変更できます。 `blx.conf` ファイルを変更した後、NetScaler BLXを再起動する必要があります。

注:

- HTTPおよびHTTPSのデフォルト管理ポート番号は、`set ns param` コマンドを使用して変更できます。ただし、`unset ns param` または `clear config full` コマンドを使用しても、デフォルトのポート番号は復元されません。

NetScaler GUIは、NetScaler BLXのログイン資格情報を要求します。GUIにログインした後、NetScaler GUIを使用してNetScalerの機能を構成できます。

GUI を使用して NetScaler BLX にアクセスする

1. Webブラウザを開きます。
2. 次のいずれかのアクセス方法を使用します。
 - HTTP アクセスの場合は、URL フィールドに次のように入力します: `<NetScaler BLX IP address (NSIP)>:<HTTP port>`
 - HTTPS アクセスの場合は、URL フィールドに次のように入力します: `<NetScaler BLX IP address (NSIP)>:<HTTPS port>`
3. ログインページで、NetScaler BLX のログイン資格情報を入力し、**ログイン**をクリックします。

NITRO API を使用して NetScaler BLX にアクセスし、NetScaler の機能を構成する

NetScaler NITRO APIを使用して、NetScaler機能を構成できます。NITROは、Representational State Transfer (REST) インターフェースを介してその機能を提供します。したがって、NITROアプリケーションは任意のプログラミング言語で開発できます。また、Java、.NET、またはPythonで開発する必要があるアプリケーションの場合、NITRO APIは、個別のソフトウェア開発キット (SDK) としてパッケージ化された関連ライブラリを介して公開されます。

NetScaler GUIと同様に、NITRO APIリクエストは、NetScaler BLX管理IPアドレスのHTTPまたはHTTPSポートに送信する必要があります。

NetScaler BLXへのアクセス

- WebブラウザでNITRO APIを使用してNetScaler BLXを構成するには、次のように入力します。

```
http://<NetScaler BLX IP address (NSIP)>:<HTTP port>/nitro/v1/config/  
<resource-type>
```

```
https://<NetScaler BLX IP address (NSIP)>:<HTTPS port>/nitro/v1/config/  
<resource-type>
```

- WebブラウザでNITRO APIを使用してNetScaler BLXの統計情報を取得するには、次のように入力します。

```
http://<NetScaler BLX IP address (NSIP)>:<HTTP port>/nitro/v1/stats/  
<resource-type>
```

```
https://<NetScaler BLX IP address (NSIP)>:<HTTPS port>/nitro/v1/stats/  
<resource-type>
```

NetScaler NITRO APIの使用方法的の詳細については、[NetScaler BLX NITRO API](#)を参照してください。

次のステップ

- [NetScaler機能を構成する](#)

NetScaler BLX クラスターをセットアップする

NetScaler BLX クラスターは、単一のシステムとして連携するNetScaler BLXインスタンスのグループです。各NetScaler BLXインスタンスはノードと呼ばれます。NetScaler BLX クラスターは、1つのインスタンスから最大32のインスタンスをノードとして持つことができます。

始める前に

- NetScalerクラスター機能を理解していることを確認してください。詳細については、[NetScaler Cluster](#)を参照してください。
- すべてのNetScaler BLXインスタンスのLinuxホストに、以下の構成が存在することを確認してください。
 - 各LinuxホストでNTPが構成されていること。

注:

- Oracle LinuxでのNTP構成については、[Oracle Linux documentation](#)を参照してください。
- Ubuntu LinuxでのNTP構成については、[Ubuntu Linux documentation](#)を参照してください。
- CentOS LinuxでのNTP構成については、[CentOS Linux documentation](#)を参照してください。

- NetScaler BLXログのロギングと `rsyslog` 設定が構成されていること。
- すべてのNetScaler BLXインスタンスでコアダンプが有効になっていることを確認してください。コアダンプの有効化の詳細については、[Configure compressed core dumps for NetScaler BLX](#)を参照してください。
- クラスターはNetScaler BLXインスタンスのみでサポートされます。
- NetScalerクラスターのすべての一般的な前提条件は、NetScaler BLXクラスターにも適用されます。

注:

NetScalerクラスターのセットアップに関する一般的な前提条件の詳細については、[NetScalerクラスターの一般的な前提条件](#)を参照してください。

- NetScaler BLXクラスターでサポートされているNetScaler機能の詳細については、[NetScaler BLXクラスターのNetScaler機能サポートマトリックス](#)を参照してください。
- Terraformを使用してNetScalerの展開を自動化する方法については、以下を参照してください。
 - [GitHub上のNetScaler Terraformプロバイダー](#)
 - [GitHub上のNetScaler Terraform自動化スクリプト](#)
- NetScaler BLXクラスターのセットアップは、パブリッククラウドプラットフォームではサポートされていません。たとえば、AWSクラウドなどです。

NetScaler BLXクラスターの制限事項

NetScaler BLXクラスターには、次の制限があります。

- CLAGベースのトラフィック分散はサポートされていません。
- スタンドアロンのNetScaler BLXのすべての制限は、NetScaler BLXクラスターにも適用されます。

スタンドアロンNetScaler BLXの制限事項の詳細については、「NetScaler BLXの制限事項」([/ja-jp/blx/current-release/limitations-blx.html](#))を参照してください。

NetScaler BLXクラスターのセットアップ

NetScaler BLXクラスターをセットアップするには、「NetScalerクラスター」([/ja-jp/citrix-adc/current-release/clustering/cluster-setup](#))に記載されているNetScalerクラスターのセットアップに関する一般的な手順に従ってください。

アップグレードとダウングレード

各NetScaler BLXリリースでは、機能が強化された新しい機能と更新された機能が提供されます。新しい機能とバグ修正を利用するために、NetScaler BLXを最新リリースにアップグレードすることをお勧めします。強化点、既知の問題、バグ修正の包括的なリストは、すべてのリリース発表に付随する[リリースノート](#)に含まれています。

開始する前に

- 組織のサポート契約を評価する必要があります。NetScaler®テクニカルサポートまたはNetScaler認定パートナーからのサポートに関するサポート契約と連絡先の詳細を文書化してください。
- アップグレードする前に、ライセンスフレームワークと使用できるライセンスの種類を理解することも重要です。詳細については、「[NetScaler BLXライセンス](#)」を参照してください。
- [新規および非推奨のコマンド、パラメータ、SNMP OID](#)のトピックを確認する必要があります。
- NetScaler BLXの構成ファイルをバックアップします。バックアップと復元の方法については、「[NetScalerをバックアップおよび復元して失われた構成を回復する方法](#)」を参照してください。

注：

バックアップされるファイルの詳細については、「[バックアップされるファイルの一覧](#)」を参照してください。

- NetScaler BLXのアップグレードまたはダウングレードは、NetScaler BLXのインストールと同じです。Linuxホストのパッケージマネージャーは、ビルド番号に基づいてアップグレードまたはダウングレード操作を管理します。
- 必要に応じて、テスト環境でNetScaler BLXのアップグレードまたはダウングレードを試すことができます。

NetScaler BLXのアップグレード

1. アップグレードするNetScaler BLXリリースパッケージをダウンロードします。詳細については、「[LinuxホストにNetScaler BLXパッケージをダウンロードする](#)」を参照してください。
2. ダウンロードしたパッケージをLinuxホストのCLIを使用してインストールします。詳細については、「[DebianベースのLinuxホストにNetScaler BLXをインストールする](#)」または「[RPMベースのLinuxホストにNetScaler BLXをインストールする](#)」を参照してください。

注：

NetScaler BLX管理ホストを設定している場合は、ダウンロードしたパッケージをインストールする前に、次の手順を実行します。

- ホストIPアドレスにログインします。
- `systemctl stop blx` コマンドを使用してNetScaler BLXを停止します。
- ホストIPアドレスに再接続します。

NetScaler BLXのダウングレード

1. ダウングレードするNetScaler BLXリリースパッケージをダウンロードします。詳細については、「[LinuxホストでNetScaler BLXパッケージをダウンロードする](#)」を参照してください。
2. LinuxホストCLIを使用して、ダウンロードしたパッケージをインストールします。詳細については、「[DebianベースのLinuxホストにNetScaler BLXをインストールする](#)」または「[RPMベースのLinuxホストにNetScaler BLXをインストールする](#)」を参照してください。

注:

- NetScaler BLX管理ホストを設定している場合は、ダウンロードしたパッケージをインストールする前に、次の手順を実行します。
 - ホストIPアドレスにログインします。
 - `systemctl stop blx` コマンドを使用してNetScaler BLXを停止します。
 - ホストIPアドレスに再接続します。

3. 自動構成機能を使用して構成されたNetScaler BLX管理ホストの場合、NetScaler BLXをバージョン13.1ビルド45.64以降からバージョン13.1ビルド42.47以前にダウングレードするときは、次の手順を実行します。
 - a. `/etc/blx/` ディレクトリにある `blx.conf` ファイルを開きます。
 - b. `interfaces` パラメータのコメントを解除し、`blx.conf` ファイルでNetScaler BLXに割り当てるインターフェイスを指定します。
 - c. `ip-address` パラメータのコメントを解除し、NetScaler BLX の NSIP アドレスを指定します。
 - d. `default` パラメータのコメントを解除し、NetScaler BLX のデフォルトルートを指定します。
 - e. `blx.conf` ファイルを保存します。
 - f. `systemctl restart blx` コマンドを使用して NetScaler BLX を再起動します。

アップグレードまたはダウングレード後の NetScaler BLX でのエンティティステータスの確認

NetScaler BLX のアップグレードまたはダウングレード後、以下を確認します。

- 仮想サーバーが UP 状態であること
- モニターが UP 状態であること
- すべての証明書が NetScaler BLX に存在すること
- すべてのライセンスが NetScaler BLX に存在すること

ユースケース

NetScaler® BLXは、Linuxホスト上でユーザー空間アプリケーションとしてネイティブに動作するNetScalerのソフトウェアフォームファクターです。BLXは標準的なLinuxホスト上で動作するため、さまざまなビジネスニーズに対応するために多様な環境に展開できます。

以下のユースケースでは、特定のシナリオでNetScaler BLXを展開および使用方法について説明します。

AWSへのNetScaler BLXの展開

AWSで利用可能なLinuxインスタンスにNetScaler BLXを展開できます。AWSに展開されたBLXを使用すると、AWSのクラウドコンピューティング機能とNetScalerの機能を組み合わせて、ビジネスニーズに対応できます。

- [Deploy NetScaler BLX on AWS](#) — AWSでのNetScaler BLXの動作概要、AWSの用語、および展開の前提条件。
- [Deploy a standalone NetScaler BLX instance on AWS](#) — AWS上にEC2 Linuxインスタンスを作成し、その上にNetScaler BLXをインストールします。
- [Deploy NetScaler BLX with GSLB on AWS](#) — NetScaler BLXでグローバルサーバーロードバランシング (GSLB) を使用して、AWS上のデータセンター間でトラフィックを分散します。

AWS上でのNetScaler BLX™のデプロイ

AWSで利用可能なLinuxインスタンスにNetScaler® BLXを展開できます。AWSにデプロイされたBLXを使用すると、ビジネスニーズに合わせてAWSのクラウドコンピューティング機能とNetScalerの機能を利用できます。

AWS用語

このセクションでは、一般的に使用されるAWSの用語とフレーズについて説明します。詳細については、[AWS用語集](#)を参照してください。

用語	定義
Amazon マシンイメージ (AMI)	インスタンス（クラウド内の仮想サーバー）を起動するために必要な情報を提供するマシンイメージ。
エラスティック・ブロック・ストア	AWSクラウド内のAmazon EC2インスタンスで使用するための永続的なブロックストレージボリュームを提供します。
シンプル・ストレージ・サービス (S3)	インターネット用のストレージ。開発者がウェブスケールのコンピューティングをより簡単にできるように設計されています。
エラスティック・コンピュート・クラウド (EC2)	クラウド内で安全かつサイズ変更可能なコンピューティング能力を提供するウェブサービス。開発者がウェブスケールのクラウドコンピューティングをより簡単にできるように設計されています。
エラスティック・ロード・バランシング (ELB)	複数のアベイラビリティゾーンにある複数のEC2インスタンスに、受信アプリケーショントラフィックを分散します。これにより、アプリケーションの耐障害性が向上します。
弾力性ネットワークインターフェイス (ENI)	仮想プライベートクラウド (VPC) 内のインスタンスにアタッチできる仮想ネットワークインターフェイス。
弾力性IP (EIP) アドレス	Amazon EC2 または Amazon VPC で割り当て、インスタンスにアタッチした静的パブリック IPv4 アドレス。弾力性IPアドレスは、特定のインスタンスではなく、アカウントに関連付けられています。必要に応じて簡単に割り当て、アタッチ、デタッチ、解放できるため、弾力性があります。
インスタンスタイプ	Amazon EC2 は、さまざまなユースケースに合わせて最適化された幅広いインスタンスタイプを提供しています。インスタンスタイプは、CPU、メモリ、ストレージ、ネットワーク容量のさまざまな組み合わせで構成されて

用語	定義
	おり、アプリケーションに適したリソースの組み合わせを柔軟に選択できます。
ID およびアクセス管理 (IAM)	AWS でのIDができることとできないことを決定するアクセス許可ポリシーを持つ AWS ID。IAM ロールを使用して、EC2 インスタンスで実行されているアプリケーションが AWS リソースに安全にアクセスできるようにすることができます。
インターネットゲートウェイ	ネットワークをインターネットに接続します。VPC の外部にある IP アドレスのトラフィックをインターネットゲートウェイにルーティングできます。
キーペア	電子的に身元を証明するために使用するセキュリティ認証情報のセット。キーペアは、秘密鍵と公開鍵で構成されます。
ルートテーブル	ルートテーブルに関連付けられているサブネットから出るトラフィックを制御するルーティングルールのセット。複数のサブネットを単一のルートテーブルに関連付けることができますが、サブネットは一度に1つのルートテーブルにのみ関連付けることができます。
セキュリティグループ	インスタンスに対して許可されるインバウンドネットワーク接続の名前付きセット。
サブネット	EC2インスタンスをアタッチできるVPCのIPアドレス範囲のセグメント。セキュリティおよび運用上のニーズに応じてインスタンスをグループ化するためにサブネットを作成できます。
仮想プライベートクラウド (VPC)	定義した仮想ネットワーク内でAWSリソースを起動できる、AWSクラウドの論理的に分離されたセクションをプロビジョニングするためのウェブサービス。

用語	定義
オートスケーリング	ユーザー定義のポリシー、スケジュール、およびヘルスチェックに基づいてAmazon EC2インスタンスを自動的に起動または終了するウェブサービス。
CloudFormation	関連するAWSリソースを単位としてまとめて作成および削除するテンプレートを作成または変更するためのサービス。

AWSでのNetScaler BLXの仕組み

NetScaler BLXは、Linuxシステム上でネイティブに動作する軽量ソフトウェアパッケージです。BLXでサポートされており、AWSマーケットプレイスで利用可能な任意のLinux AMIにBLXをインストールできます。サポートされているLinuxディストリビューションの詳細については、「[サポートされているLinuxディストリビューション](#)」を参照してください。

BLXは、AWS VPC内のEC2 Linuxインスタンス上でLinuxプロセスとして実行されます。Linux AMI インスタンスには、最低2つの仮想CPUと2GBのメモリが必要です。AWS VPC内で起動されたEC2インスタンスは、複数のインターフェイスまたはインターフェイスごとに複数のIPアドレスを持つことができます。各BLXインスタンスには、少なくとも3つのIPサブネットが必要です。

- 管理サブネット (NSIP)
- クライアント向けサブネット (VIP)
- バックエンド向けサブネット (SNIP)

注:

AWSでの標準的なBLX展開には、3つのネットワークインターフェイスを推奨します。

AWSは現在、AWS VPC内で実行されているインスタンスに対してのみマルチIP機能をサポートしています。VPC内のBLXインスタンスは、EC2インスタンスで実行されているサーバーの負荷分散に使用できます。Amazon VPCを使用すると、独自のIPアドレス範囲、サブネット、ルートテーブル、ネットワークゲートウェイを含む仮想ネットワーク環境を作成および制御できます。

注:

デフォルトでは、各AWSアカウントにつきAWSリージョンごとに最大5つのVPCインスタンスを作成できます。Amazonの[リクエストフォーム](#)を送信することで、より高いVPC制限をリクエストできます。

次の図は、Linux AMIに展開されたBLXを持つAWS VPCのシンプルなトポロジを示しています。



AWS VPCには次のものがあります。

- VPC内外のトラフィックをルーティングするための単一のインターネットゲートウェイ
- インターネットゲートウェイとインターネット間のネットワーク接続
- 管理用、クライアント用、サーバー用にそれぞれ1つずつ、計3つのサブネット
- インターネットゲートウェイと2つのサブネット（管理用とクライアント用）間のネットワーク接続
- 各サブネットに3つのENIがアタッチされたLinuxインスタンスにインストールされたスタンドアロンBLXインスタンス

前提条件

AWSでインスタンスを作成する前に、次の点を確認してください。

- EC2インスタンスが[BLXシステム要件](#)を満たしていることを確認してください。
- パフォーマンス向上のため、m5.xlarge以上のインスタンスタイプを作成することをお勧めします。
- NSIP、VIP、およびSNIPを設定するには、3つのIPアドレスが必要です。

注:

VIPおよびSNIPとして設定されたIPアドレスには、パブリックIPアドレスが関連付けられている必要があります。

- AWS Virtual Private Cloud (VPC) でLinux AMIを起動するには、AWSアカウントが必要です。aws.amazon.comで無料でAWSアカウントを作成できます。
- ユーザーのAWSサービスとリソースへのアクセスを安全に制御するには、AWS Identity and Access Management (IAM) ユーザーアカウントが必要です。IAMユーザーアカウントの作成方法の詳細については、[IAMユーザーの作成 \(コンソール\)](#)を参照してください。

- ターミナルプログラムからAWS Management Consoleが提供するすべての機能を使用できます。詳細については、[AWS CLIユーザーガイド](#)を参照してください。ネットワークインターフェースタイプをSR-IOVに変更するには、AWS CLIも必要です。
- Elastic Network Adapter (ENA) ドライバーが有効なインスタンスタイプ（M5、C5インスタンスなど）の場合、ファームウェアバージョンは13.0以降である必要があります。

制限事項と使用ガイドライン

AWSにNetScaler BLXインスタンスを展開する場合、以下の制限事項と使用ガイドラインが適用されます。

- データトラフィックと管理トラフィックのENIは、異なるサブネットに存在する必要があります。
- 管理ENIにはNSIPアドレスのみが存在する必要があります。
- NSIPにEIPを割り当てる代わりに、セキュリティのためにNATインスタンスを使用する場合、適切なVPCレベルのルーティング変更が必要です。VPCレベルのルーティング変更を行う手順については、[シナリオ2: パブリックサブネットとプライベートサブネットを持つVPC](#)を参照してください。
- ENIに複数のIPアドレスを割り当てることができます。ENIあたりのIPアドレスの最大数は、EC2インスタンスタイプによって決まります。詳細については、[Elastic Network Interfaces](#)の「IP Addresses Per Network Interface Per Instance Type」セクションを参照してください。

注:

ENIに割り当てる前に、AWSでIPアドレスを割り当てる必要があります。詳細については、[Elastic Network Interfaces](#)を参照してください。

- AWSの制限により、以下の機能はサポートされていません。
 - 無償ARP (GARP)
 - L2モード
 - タグ付きVLAN
 - 動的ルーティング
 - 仮想MAC

- RNATが機能するには、**Source/Destination**チェックが無効になっていることを確認してください。詳細については、[Elastic Network Interfaces](#)の「Changing the Source/Destination Checking」を参照してください。

AWS上にスタンドアロンのNetScaler BLX™ インスタンスを展開する

このトピックでは、AWS上にEC2 Linuxインスタンスを作成し、そのLinuxインスタンスにBLXをインストールする手順について説明します。

デプロイを開始する前に、以下のトピックをお読みください。

- [前提条件](#)
- [制限事項と使用ガイドライン](#)

デプロイ手順

以下の手順を実行します。

1. [EC2 Linuxインスタンスを作成する](#)
2. [NetScaler® BLXをダウンロードする](#)
3. [NetScaler BLXをインストールする](#)
4. [BLXを設定する](#)
5. [NetScaler BLXを開始する](#)
6. [NetScaler BLXにアクセスする](#)
7. [ライセンス](#)

EC2 Linuxインスタンスを作成する

AWSウェブコンソールを使用して、AWS上にEC2 Linuxインスタンスを作成するには、次の手順を実行します。

1. [キーペアを作成する](#)

Amazon EC2は、ログイン情報を暗号化および復号化するためにキーペアを使用します。インスタンスにログオンするには、キーペアを作成し、インスタンスを起動するときにキーペアの名前を指定し、インスタンスに接続するときに秘密鍵を提供する必要があります。

AWS起動インスタンスウィザードを使用してインスタンスを確認して起動すると、既存のキーペアを使用するか、新しいキーペアを作成するように求められます。キーペアの作成方法の詳細については、「[Amazon EC2キーペア](#)」を参照してください。

2. VPCを作成する

NetScaler VPCインスタンスは、AWS VPC内にデプロイされます。VPCを使用すると、AWS アカウント専用の仮想ネットワークを定義できます。AWS VPCの詳細については、「[Amazon VPCの開始方法](#)」を参照してください。

Linuxインスタンス用のVPCを作成する際は、次の点に留意してください。

- AWSアベイラビリティゾーンにAWS VPCを作成するには、VPCを「**単一のパブリックサブネットのみ**」オプションで使します。
- Citrixは、少なくとも3つのサブネットを作成することを推奨しています。すべてのサブネットは同じアベイラビリティゾーンにある必要があります。
 - 管理トラフィック用のサブネットを1つ。このサブネットに管理IP (NSIP) を配置します。デフォルトでは、エラスティックネットワークインターフェース (ENI) eth0が管理IPに使用されます。
 - クライアントアクセス (ユーザーからNetScaler BLXへ) トラフィック用のサブネットを1つ以上。クライアントは、NetScalerロードバランシング仮想サーバーに割り当てられた1つ以上の仮想IP (VIP) アドレスに接続します。
 - サーバーアクセス (BLXからサーバーへ) トラフィック用のサブネットを1つ以上。サーバーは、BLXが所有するサブネットIP (SNIP) アドレスに接続します。

3. サブネットを追加する

VPCウィザードを使用すると、1つのサブネットしか作成されません。要件に応じて、さらに多くのサブネットを作成したい場合があります。追加のサブネットの作成方法の詳細については、「[VPCへのサブネットの追加](#)」を参照してください。

4. セキュリティグループとセキュリティルールを作成する

インバウンドおよびアウトバウンドトラフィックを制御するには、セキュリティグループを作成し、グループにルールを追加します。グループの作成方法とルールの追加方法の詳細については、[VPC のセキュリティグループ](#)を参照してください。

EC2 ウィザードは、AWS Marketplace が生成する Linux インスタンス用のデフォルトのセキュリティグループを提供します。ただし、要件に基づいてさらにセキュリティグループを作成できます。

セキュリティグループで、SSH、HTTP、および HTTPS アクセス用に次のポートを開く必要があります。

アクセスタイプ	ポート番号
SSH	22
HTTP	80
HTTPS	443

5. ルートテーブルを追加する

ルートテーブルには、ネットワークトラフィックの転送先を決定するために使用されるルートと呼ばれる一連のルールが含まれています。VPC 内の各サブネットは、ルートテーブルに関連付けられている必要があります。ルートテーブルの作成方法の詳細については、[ルートテーブル](#)を参照してください。

6. インターネットゲートウェイを作成する

インターネットゲートウェイには2つの目的があります。VPC ルートテーブルでインターネットルーティング可能なトラフィックのターゲットを提供することと、パブリック IPv4 アドレスが割り当てられているインスタンスのネットワークアドレス変換 (NAT) を行うことです。

インターネットトラフィック用のインターネットゲートウェイを作成します。インターネットゲートウェイの作成方法の詳細については、「[インターネットゲートウェイのアタッチ](#)」セクションを参照してください。

7. AWS EC2サービスを使用してLinuxインスタンスを作成する

- a. AWSダッシュボードから、**Compute > EC2 > Launch Instance > AWS Marketplace**に移動します。

Launch Instanceをクリックする前に、**Launch Instance**の下に表示されるメモを確認して、リージョンが正しいことを確認してください。



- b. **Search AWS Marketplace**バーで、キーワード「Linux」で検索します。
- c. デプロイするAMIを選択し、**Select**をクリックします。

インスタンス起動ウィザードが開始されます。ウィザードに従ってインスタンスを作成します。ウィザードは次のことを求めます。

- **インスタンスタイプを選択**

- インスタンスを設定
- ストレージを追加
- タグを追加
- セキュリティグループを設定
- 確認



8. 追加のネットワークインターフェースを作成してアタッチする

VIPとSNIP用にさらに2つのネットワークインターフェースを作成します。ネットワークインターフェースの作成方法の詳細については、[ネットワークインターフェースの作成セクション](#)を参照してください。

ネットワークインターフェースを作成したら、それらをLinuxインスタンスにアタッチする必要があります。インターフェースをアタッチする前に、Linuxインスタンスをシャットダウンし、インターフェースをアタッチして、インスタンスの電源をオンにします。ネットワークインターフェースのアタッチ方法の詳細については、[インスタンス起動時のネットワークインターフェースのアタッチセクション](#)を参照してください。

9. Elastic IPアドレスの割り当てと関連付け

EC2インスタンスにパブリックIPアドレスを割り当てた場合、そのアドレスはインスタンスが停止するまでのみ割り当てられたままになります。その後、アドレスはプールに解放されます。インスタンスを再起動すると、新しいパブリックIPアドレスが割り当てられます。

対照的に、Elastic IP (EIP) アドレスは、インスタンスから関連付けが解除されるまで割り当てられたままになります。

管理NICにElastic IPを割り当てて関連付けるには、[Elastic IPアドレスの割り当て](#)を参照してください。

これらの手順で、AWS上にLinuxインスタンスを作成する手順は完了です。インスタンスが準備できるまで数分かかる場合があります。インスタンスがステータスチェックに合格したことを確認してください。この情報は、**Instances**ページの**Status Checks**列で確認できます。

10. Linuxインスタンスへの接続

Linuxインスタンスを作成したら、AWS管理コンソールからインスタンスに接続できます。

- a. Linuxインスタンスを選択し、**Connect**をクリックします。
- b. **Connect to Your Instance**ページに表示される指示に従ってください。

NetScaler BLXのダウンロード

Linux AMIにBLXパッケージをダウンロードします。詳細については、「」([/ja-jp/blx/current-release/deploy-blx/download-blx.html](#))を参照してください。

NetScaler BLXのインストール

- DebianベースのLinuxホストにBLXをインストールするには、「」([/ja-jp/blx/current-release/deploy-blx/install-blx-debian.html](#))を参照してください。
- RPMベースのLinuxホストにBLXをインストールするには、[RPMベースのLinuxホストにNetScaler BLXをインストールする](#)を参照してください。

BLXを構成する

BLXをインストールした後、構成ファイルを編集してBLXを起動します。詳細については、[NetScaler BLXの構成](#)を参照してください。

NetScaler BLXを起動する

blx.confファイルを編集した後、BLXを起動します。詳細については、[NetScaler BLXの起動](#)を参照してください。

NetScaler BLXにアクセスする

以下のいずれかの方法でBLXにアクセスできます。

- NetScaler CLI
- NetScaler GUI
- NetScaler NITRO REST API

詳細については、[NetScaler BLXへのアクセス](#)を参照してください。

ライセンス

NetScalerは、組織の要件を満たす幅広い製品エディションとライセンスモデルを提供しています。詳細については、[NetScaler BLXライセンス](#)を参照してください。

AWSでGSLBを使用してNetScaler BLX™を展開する

NetScaler® BLXはNetScalerのソフトウェアフォームファクターであり、他のフォームファクターと同じ機能を提供します。Linuxホスト上でユーザースペースアプリケーションとして動作します。

AWSでBLXのGSLBを設定するには、NetScalerが属するVPCの外部にあるサーバー（別の可用性リージョンの別のVPC内やオンプレミスデータセンターなど）にトラフィックをロードバランシングするようにNetScalerを構成します。

AWSでのBLX GSLB展開の図(</en-us/blx/media/blx-aws-gslb-deployment.png>)

ドメインベースサービス概要

NetScaler GSLBは、クラウドロードバランサー向けのドメインベースサービス (DBS) をサポートしており、動的なクラウドサービスの自動検出を可能にします。この構成により、NetScalerはアクティブ-アクティブ環境でグローバルサーバーロードバランシングを実装できます。DBSは、DNS検出を通じてAWS環境におけるバックエンドリソースのスケールリングを可能にします。

ELBとのDBS

GSLB DBSは、ユーザーのElastic Load Balancer (ELB) のFQDNを使用して、AWS内で作成および削除されるバックエンドサーバーを含むようにGSLBサービスグループを動的に更新します。AWSのバックエンドサーバーまたはインスタンスは、ネットワーク需要またはCPU使用率に基づいてスケールリングするように構成できます。この機能を構成するには、NetScalerをELBにポイントして、AWS内でインスタンスが作成および削除されるたびにNetScalerを手動で更新することなく、AWS内の異なるサーバーに動的にルーティングします。GSLBサービスグループ向けのNetScaler DBS機能は、DNS対応サービス検出を使用して、Autoscale™グループで識別されたDBS名前空間のメンバーサービスリソースを決定します。

前提条件

- 異なるAWS VPCに2つのスタンドアロンBLXインスタンスを展開します。展開の詳細については、「[AWSにスタンドアロンNetScaler BLXインスタンスを展開する](#)」を参照してください。

注:

`blx.conf` ファイルのcli-cmdsセクションに以下のコマンドを指定していることを確認してください。

- `add ns ip <VIP address> <netmask> -type VIP`
- `add ns ip <SNIP address> <netmask> -type SNIP -mgmtAccess ENABLED`
- GSLB機能をサポートするNetScalerライセンスが必要です。詳細については、「[ライセンス](#)」を参照してください。
- AWS上で2つのLinuxインスタンスが利用可能であることを確認してください。

注:

パフォーマンス向上のため、m5.xlarge以上のインスタンスタイプを作成することをお勧めします。

- NSIP、VIP、SNIPとして構成するために、3つのIPアドレスが必要です。

注:

VIPおよびSNIPとして構成されたIPアドレスには、パブリックIPアドレスが関連付けられている必要があります。

- セキュリティグループで以下のポートを開く必要があります。
 - 53/UDP
 - 22/TCP
 - 3008/TCP
 - 3009/TCP

blx.confファイルのサンプル

```
blx-system-config

{
    core-dumps: yes

    syslog: yes

    ipaddress: 10.0.12.245/20

    interfaces: eth1
}

static-routes

{

    default 10.0.0.1
```

```
}  
  
cli-cmds  
  
{  
  
    add ns ip 11.0.12.245/20 -type VIP`  
  
    add ns ip 12.0.12.245/20 -type SNIP -mgmtAccess ENABLED`  
  
}
```

GSLBの構成

GSLBで構成されたNetScalerは、WANにおける障害点から保護することで、ディザスタリカバリを提供し、アプリケーションの継続的な可用性を確保します。GSLBは、クライアント要求を最も近い、または最もパフォーマンスの高いデータセンター、あるいは障害発生時には稼働中のデータセンターに誘導することで、データセンター間の負荷を分散します。

GSLBの展開タイプと構成の詳細については、[GSLBドキュメント](#)を参照してください。

よくある質問

NetScaler BLXとは何ですか？

NetScaler BLXは、Linuxホスト上でネイティブアプリケーションとして動作するNetScalerのベアメタルソフトウェアバージョンです。詳細については、[NetScaler BLXについて](#)を参照してください。

なぜNetScaler BLXが必要なのですか？

NetScaler BLXはNetScalerのベアメタルバージョンであり、仮想マシンのオーバーヘッドなしでシンプルさを提供し、パフォーマンスを向上させます。また、NetScaler BLXはお好みのサーバーハードウェアで実行できます。詳細については、[NetScaler BLXが必要な理由](#)を参照してください。

NetScaler BLX、CPX、VPXの違い？

NetScaler BLX、CPX、VPXは、NetScalerのソフトウェアフォームファクターです。

NetScaler VPX™は仮想アプライアンスであり、サーバーにインストールされたハイパーバイザー上で実行する必要があります。

NetScaler CPXはNetScalerのコンテナ化されたバージョンであり、コンテナ内で実行する必要があります。

NetScaler BLXは、Linuxシステム上でネイティブに動作するソフトウェアパッケージです。

NetScaler BLXを使用するタイミング？

NetScaler VPX、CPX、およびNetScaler BLXは、ハイブリッドマルチクラウドへの移行をサポートするために、業界で最も包括的なソフトウェア中心のADCラインナップを代表しています。以下の表は、違いとユースケースに関するガイダンスを提供します。

製品	ユースケース	特徴
NetScaler VPX (ハイパーバイザー上で動作)	ハードウェアインフラストラクチャの仮想化、共通インフ	ハードウェアとOSに依存しない、完全な分離、マルチテナンシーのサポート

製品	ユースケース	特徴
	ラストラクチャ上でのワークロードの統合	
NetScaler CPX (コンテナ内で動作)	DevOps、マイクロサービス、自動ステージング、テスト、デプロイ、East-Westトラフィック	軽量、省フットプリント、APIゲートウェイ機能、マイクロサービス中心、認証
NetScaler BLX (ベアメタルサーバー上で動作)	高トラフィック負荷、ミッションクリティカルなアプリケーション、レイテンシーに敏感なワークロード、North-Southトラフィック	ネイティブLinuxソフトウェアパッケージ、VMオーバーヘッドなし

ハイパーバイザーやコンテナがないことによって、どのような違いがありますか？

ハイパーバイザーの変換レイヤーやコンテナがないため、NetScaler BLXソフトウェアは基盤となるハードウェアをより詳細に制御でき、パフォーマンスが向上します。また、ハイパーバイザーソフトウェアに追加費用はかかりません。

NetScaler BLXはどのサーバーハードウェアでも実行できますか？

はい、NetScaler BLXはどのサーバーハードウェアでも実行できます。ただし、より高いパフォーマンスを得るには、NetScaler BLXでサポートされているDPDK互換NICを使用することをお勧めします。ハードウェア要件の詳細については、「[Linuxホストのハードウェア要件](#)」を参照してください。

LinuxサーバーにNetScaler BLXをデプロイするにはどうすればよいですか？

NetScaler BLXは、以下のいずれかのパッケージディストリビューションをサポートする任意のLinuxサーバーにデプロイできます。

- **.rpm** (RPMベースのパッケージ)
- **.deb** (Debianベースのパッケージ)

NetScaler BLXの展開については、[NetScaler BLXの展開](#)を参照してください。

NetScaler BLXソフトウェアの展開を自動化できますか？

はい、`.rpm` または `.deb` パッケージをサポートする任意のソフトウェア展開ツールを使用して、NetScaler BLXを展開できます。

たとえば、Terraformを使用してNetScaler BLXを展開できます。詳細については、[Terraformを使用したNetScaler BLXの展開](#)を参照してください。

NetScaler BLXがLinux OSを搭載したサーバーにインストールされている場合の、他の標準的なLinuxパッケージやアプリケーションのインストール？

はい、他の標準的なLinuxパッケージやアプリケーションはNetScaler BLXと並行して実行できます。

NetScaler BLXライセンスの購入方法？

デフォルトでは、NetScaler BLXにはエクスプレスライセンスが付属しています。NetScaler BLXを無料で試用できます。

製品にご満足いただけましたら、サブスクリプションベースのローカルライセンスまたはNetScaler®プール容量ライセンスにアップグレードできます。

NetScaler BLXライセンスの詳細については、[NetScaler BLXライセンス](#)を参照してください。

現在のNetScaler VPXライセンスをNetScaler BLXに使用できますか？

はい、現在のVPXライセンスをNetScaler BLXに使用できます。詳細については、[NetScaler BLXライセンス](#)を参照してください。

NetScaler BLXをワンアームモードとツーアームモードで展開できますか？

はい、NetScaler BLXはワンアームモードまたはツーアームモードのいずれかで展開できます。詳細については、「[物理展開モード](#)」を参照してください。

NetScaler BLXはどのネットワークスタックを使用しますか？

NetScaler BLXは独自のネットワークスタックを使用します。NetScaler BLXのアーキテクチャの詳細については、「[一般的なアーキテクチャ](#)」を参照してください。

NetScaler BLXは高可用性をサポートしていますか？

NetScaler BLXは高可用性をサポートしています。高可用性セットアップの詳細については、「[高可用性](#)」を参照してください。

NetScaler BLXとNetScaler VPXまたはCPXの間で高可用性ペアをセットアップできますか？

いいえ、NetScaler BLXとVPXまたはCPXの間で高可用性ペアをセットアップすることはできません。

DPDKを搭載した仮想化LinuxマシンでNetScaler BLXを実行できますか？

はい、NetScaler BLXは以下のシナリオで仮想化Linuxマシン上で実行できます。

- BLXは、お客様の要件に応じて、カスタマイズされたホストイメージでNetScaler®を実行する必要がある場合に活用できます。
- BLXは、NetScaler VPXが公式にサポートされていない環境においても、推奨される仮想マシンソリューションです。

ARMプラットフォームでNetScaler BLXを実行できますか？

いいえ、NetScaler BLXはIntelまたはAMD x86-64 (64ビット) Linuxプラットフォームでのみサポートされています。

NetScaler BLXの管理IPアドレスは何ですか？

NSIPアドレスはNetScaler BLXの管理IPアドレスです。これは `blx.conf` の `ipaddress` パラメータを使用して構成されます。 `ipaddress` パラメータでNSIPアドレスを指定せずにNetScaler BLX管理ホストを構成した場合、LinuxホストIPが自動的にNetScaler BLXのNSIPアドレスとして追加されます。

NetScaler BLX のメモリ使用率が常に高いのはなぜですか？

メモリ使用率は、NetScaler BLX の現在のメモリ使用量と、NetScaler BLX がオペレーティングシステム (OS) から自身に割り当てる総メモリ量を比較することによって決定されます。NetScaler BLX は、起動と基本的な操作のために、OS から必要最小限のメモリを割り当てることから開始します。当初、NetScaler BLX はその操作のために必要最小限のメモリを割り当てるため、メモリ消費率は高くなります。最終的に、NetScaler BLX の操作の規模が拡大するにつれてより多くのリソースが必要になると、NetScaler BLX は OS から追加のメモリを段階的に割り当て、メモリ使用量も高くなり、結果としてメモリ使用率が高くなります。

このように、NetScaler BLX の動的なメモリ割り当て戦略により、NetScaler BLX は通常、高いメモリ使用率を示しますが、これは利用可能なメモリの不足ではなく、リソースの効率的な使用を反映しています。

トラブルシューティング

blx.conf ファイル内の `ipaddress` (NSIP) を変更しましたが、NetScaler BLX は古い NSIP アドレスを使い続けています

NetScaler BLX の管理 IP アドレスは、以下のいずれかの方法で設定されていない限り、常に

`blx.conf` ファイルの `ipaddress` パラメータで設定された IP アドレスです。

- **NetScaler BLX CLI:** NetScaler BLX CLI で `set ns config` コマンドを実行し、管理 IP アドレスを変更します。行われた設定変更は、NetScaler BLX の保存済み設定ファイル (`/nsconfig/ns.conf`) に保存されます。
- **NetScaler BLX GUI:** NetScaler BLX GUI の設定ユーティリティ画面で、右上隅の歯車アイコンをクリックし、**NSIP アドレス**ペインをクリックして、管理 IP アドレスを変更します。

行われた設定変更は、NetScaler BLX の保存済み設定ファイル (`/nsconfig/ns.conf`) に保存されます。

`ns.conf` ファイルの変更は、常に `blx.conf` ファイルよりも優先されます。

`systemctl start blx` コマンドを使用して NetScaler BLX を起動しましたが、起動に失敗します

- NetScaler BLX の起動ログファイル (`/var/log/blx-boot.log`) で、NetScaler BLX 設定ファイル (`/etc/blx/blx.conf`) の解析エラーに関連するログを探します。
- SYSLOG ファイル (`/var/log/messages`) で、クラッシュ関連のログまたはエラーログを探します。

NetScaler BLX が DPDK ポートで起動しない

- `blx.conf` ファイルの `interfaces` パラメータで指定されたポートが、NetScaler BLX でサポートされている DPDK ポートであることを確認してください。NetScaler BLX でサポートされている DPDK ポートについては、「[Linux ホストのハードウェア要件](#)」を参照してください。

NetScaler BLX の制限事項と使用ガイドライン

以下の制限事項と使用ガイドラインは、NetScaler BLX に関連しています。

高可用性

- Amazon Web Services (AWS) や Oracle Cloud Infrastructure (OCI) などのパブリッククラウドプラットフォームでは、高可用性はサポートされていません。
- `nsinternal` ユーザーログインが無効になっている場合、高可用性はサポートされません。

NetScaler BLX クラスター

- CLAGベースのトラフィック分散はサポートされていません。

NetScaler ゲートウェイ

- MAC および Linux SSO VPN クライアントはサポートされていません。
- RDP プロキシ機能はサポートされていません。

LA および LACP チャンネル

- LA/LACP チャンネルは、専用の NIC インターフェイスまたは DPDK NIC インターフェイス間でのみサポートされます。
- LA/LACP チャンネルは、`blx1` および `ns1` 仮想インターフェイスではサポートされていません。

ウェブアプリケーションファイアウォール (WAF)

- Web Application Firewall (WAF) が有効になっている場合、BLX Gateway にはアクセスできません。

DPDKポートを使用するNetScaler BLX

- DPDKポートを使用するNetScaler BLXは、Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60 GHzやCPU E5504 @ 2.00 GHzなどの一部の古いCPUモデルでLinuxホストが実行されている場合、起動に失敗する可能性があります。
- NetScaler BLXの実行中にDPDKモジュールにバインドされているNICポートのバインドを解除すると、Linuxホストがクラッシュする可能性があります。
- DPDKポートを使用するNetScaler BLXは、DPDKポートを使用しないNetScaler BLXよりも再起動に時間がかかります。
- DPDKにバインドされたすべてのLinuxポートは、自動的にNetScaler BLX専用となり、他のDPDK Linuxアプリケーションには使用できません。
- NetScaler BLXでサポートされているVMXNET3 DPDKポートの場合、ワーカプロセスの数を2のべき乗 (2^n) で指定する必要があります。たとえば、1、2、4、8などです。
- NetScaler BLXは、DPDKポートに対してのみトランクモードまたはVLANタギングをサポートします。

Mellanoxポート

- NetScaler BLXは、一度に1種類のDPDKポートのみをサポートします。たとえば、すべてのMellanoxポート、またはすべてのIntelポートのいずれかです。
- NetScaler BLXは、Mellanoxポートに対してMLX5 DPDKドライバーのみをサポートします。
- MLX5 DPDKドライバーとその制限の詳細については、公式の[MLX5 DPDKドキュメント](#)を参照してください。
- Mellanox NICとその制限の詳細については、公式の[Mellanoxドキュメント](#)を参照してください。

その他の制限とガイドライン

- NetScaler BLX管理ホスト機能で構成されたNetScaler BLXを再起動すると、LinuxホストへのすべてのアクティブなSSHセッションが閉じられます。接続を復元するには、ホストへの接続を再試行する必要があります。
- `blx.conf` ファイルで指定された管理HTTPまたはHTTPSポート（`mgmt-http-port` または `mgmt-https-port`）は無視されます。デフォルトでは、ポート番号80と443はHTTPおよびHTTPS管理アクセス専用です。NetScaler BLXでこれらのポートを変更するには、次のNetScaler CLIコマンドを使用する必要があります。

```
set ns param (-mgmthttpport <value> | -mgmthttpsport <value>)
```

例: 次のコマンドは、管理HTTPポートを2080に変更します。

```
set ns param -mgmthttpport 2080
```

- Linuxホストでファイアウォールが有効になっている場合、NetScaler BLX管理ポートとSYSLOGポートの例外を追加する必要がある場合があります。
- NetScaler BLXの起動には最大45秒かかる場合があります。
- NetScaler BLXの構成は `/nsconfig/ns.conf` ファイルに保存されます。構成をセッション間で利用できるようにするには、構成を変更するたびに構成を保存する必要があります。

◦ **NetScaler® CLIを使用して実行中の構成を表示するには**

コマンドプロンプトで、次のように入力します。

```
show ns runningConfig
```

◦ **NetScaler CLIを使用して構成を保存するには**

コマンドプロンプトで、次のように入力します。

```
save ns config
```

- `/nsconfig/ns.conf` ファイル内のNetScaler BLX構成は、`/etc/blx/blx.conf` ファイル内の構成よりも優先されます。
- ワーカープロセスあたり1GB未満のメモリが割り当てられている場合、NetScaler BLXは起動しません。
- NetScaler BLXをインストールすると、Linuxホストで `ip_forward` パラメーターが1に設定されます。
- NetScaler BLX をアンインストールした後、設定ファイル (`blx.conf`) は保持され、`blx.conf.rpmsave` としてバックアップされます。このバックアップ設定ファイルを同じLinuxホストに新しくインストールされたNetScaler BLXに適用するには、ファイルを`blx.conf` に手動で名前変更する必要があります。
- NetScaler BLX はパケットドロップ関連の問題に遭遇する可能性があるため、以下のUbuntuバージョンでNetScaler BLXを実行することはお勧めしません。

```
Ubuntu version 16.04.5 with kernel version 4.4.0-131-generic
```

- NetScaler BLX は、最大9つのNICポート（DPDK NICポート、非DPDK NICポート、またはその両方の組み合わせ）をサポートします。
- 以下の条件が満たされている場合、NetScaler BLX が起動しない、または正常に機能しない可能性があります。

- Linux ホストで SELinux ポリシーが有効になっています。SELinux は、systemd プロセスが一部の NetScaler BLX システムファイルを実行するのを防ぎます。

回避策: Linux ホストで SELinux を無効にします。

注:

NetScaler BLX 14.1-17.x 以降では、Red Hat ベースの Linux ホストに NetScaler BLX をインストールすると、SELinux モジュールが Linux ホストで利用可能な場合、SELinux ポリシーが適用されます。このポリシーにより、NetScaler BLX は Linux ホストで実行できます。SELinux ポリシーの詳細については、[SELinux policy](#) を参照してください。

- NetScaler BLX 14.1-25.x は、`/var/netscaler/gui/vpn` ディレクトリに `pluginlist.xml` ファイルが存在しないため、フルトンネルモードで起動できません。**回避策:** 次のコマンドを実行します。

```
enable ns feature RESPONDER
```

```
add responder action pluginlist_respond respondwith q{"HTTP/1.1 200
OK\r\nDate: Thu, 30 May 2024 12:00:51 GMT\r\nServer: Apache\r\nX-Frame-Options: SAMEORIGIN\r\nLast-Modified: Thu, 30 May 2024 11:45:52
GMT\r\nETag: \"60b-619aa68c07aea\"\r\nAccept-Ranges: bytes\r\nContent-Length: 1547\r\nFeature-Policy: camera 'none'; microphone 'none';
geolocation 'none'\r\nReferrer-Policy: no-referrer\r\nX-XSS-Protection: 1; mode=block\r\nX-Content-Type-Options: nosniff\r\nContent-Type: application/xml; charset=utf-8\r\nKeep-Alive: timeout=15, max=100\r\nConnection: Keep-Alive\r\nCache-Control: no-cache, no-store\r\nPlugin-Upgrade: epa_win:Never;epa_mac:Never;epa_linux:Never;vpn_win:Never;vpn_mac:Never;vpn_linux:Never;\r\n\r\n<repositories>\n\t<repository name=\"default\">\n\t\t<plugin\n\t\t\t\t\tname=\"Netscaler Gateway EPA plug-in for Windows (32 bit)\" type=\"WIN-EPA\"\n\t\t\t\t\tversion=\"23.8.1.11\" path=\"/epa/scripts/win/nsepa_setup.exe\"\n\t\t\t\t\tcompatibleFrom=\"12.1.0.0\" compatibleTill=\"\"\n\t\t\t\t/>\n\t\t<plugin\n\t\t\t\t\tname=\"Netscaler Gateway EPA plug-in for Windows (64 bit)\" type=\"WIN-
```

```
EPA64\"n\t\t\t\tversion=\"23.8.1.11\" path=\"/epa/scripts/win/
nsepa_setup.exe\"n\t\t\t\tcompatibleFrom=\"12.1.0.0\"
compatibleTill=\"\"n\t\t\t\t/>n\t\t\t\t<plugin\n\t\t\t\t\tname=\"Netscaler
Gateway VPN plug-in for Windows\" type=\"WIN-
VPN\"n\t\t\t\t\tversion=\"23.8.1.11\" path=\"/vpns/scripts/vista/
AGEE_setup.exe\"n\t\t\t\t\tcompatibleFrom=\"12.1.0.0\"
compatibleTill=\"\"n\t\t\t\t\t/>n\t\t\t\t\t<plugin\n\t\t\t\t\t\tname=\"EPA scanning
Engine (OpSWAT) for Windows\" type=\"WIN-EPA-
ENGINE\"n\t\t\t\t\t\tversion=\"23.8.1.11\" path=\"/epa/scripts/win/
epaPackage.exe\" opswatVersion=\"4.3.3635.0\"n\t\t\t\t\t\t/
>n\t\t\t\t\t\t<plugin\n\t\t\t\t\t\t\tname=\"Netscaler Gateway EPA plug-in for Mac\"
type=\"MAC-EPA\"n\t\t\t\t\t\t\tversion=\"22.11.3\" path=\"/epa/scripts/mac/
Citrix_Endpoint_Analysis.dmg\"n\t\t\t\t\t\t\tcompatibleFrom=\"22.11.3\"
compatibleTill=\"\"n\t\t\t\t\t\t\t/>n\t\t\t\t\t\t\t<plugin\n\t\t\t\t\t\t\t\tname=\"Netscaler
Gateway VPN plug-in for Mac\" type=\"MAC-VPN\"n\t\t\t\t\t\t\t\tversion=\"4.4.8
(518)\" path=\"/vpns/scripts/mac/
Citrix_Access_Gateway.dmg\"n\t\t\t\t\t\t\t\tcompatibleFrom=\"4.4.8 (518)\"
compatibleTill=\"\"n\t\t\t\t\t\t\t\t/>n\t\t\t\t\t\t\t\t<plugin\n\t\t\t\t\t\t\t\t\tname=\"EPA scanning
Engine (OpSWAT) for Mac\" type=\"MAC-EPA-
ENGINE\"n\t\t\t\t\t\t\t\t\tversion=\"1.3.5.7\" path=\"/epa/scripts/mac/
MacLibs.zip\" opswatVersion=\"4.3.2138.0\"n\t\t\t\t\t\t\t\t\t/
>n\t\t\t\t\t\t\t\t\t<plugin\n\t\t\t\t\t\t\t\t\t\tname=\"Netscaler Gateway RfWeb GUI\"
type=\"RFWEB-GUI\"n\t\t\t\t\t\t\t\t\t\tversion=\"23.8.1.11\" path=\"/logon/
logonPoint/\"n\t\t\t\t\t\t\t\t\t\t/>n\t\t\t\t\t\t\t\t\t\t</repository>n</repositories>n\"}
```

```
add responder policy pluginlist_respond_pol
"HTTP.REQ.URL.CONTAINS(\"pluginlist.xml\")" pluginlist_respond
```

```
bind vpn vserver <VSERVER_NAME> -policy pluginlist_respond_pol
-priority 1 -gotoPriorityExpression END -type AAA_REQUEST
```

- a. 非ルート nsroot モードで NetScaler BLX を起動すると、ルート nsroot ユーザーモードに戻ることはできません。これを行うには、NetScaler BLX を再インストールする必要があります。 `showtechsupport` コマンドを実行するには、root としてホストにログインし、`showtechsupport.pl` スクリプトを実行する必要があります。

NetScaler BLX でサポートされていない NetScaler 機能

- 管理パーティション
- コンテンツ最適化
- ハードウェア SSL オフロード
- Intermediate System-to-Intermediate System (IS-IS) ルーティングプロトコル
- アイピーセック
- ジャンボフレーム

注：

- NetScalerリリース14.1-66.x以降、NetScaler BLXはIntel XL710 NIC（i40e DPDKドライバー）でジャンボフレームをサポートしています。
- NetScalerリリース14.1-72.x以降、NetScaler BLXはIntel E810およびMellanox ConnectX NIC（ConnectX-4、ConnectX-5、ConnectX-6）でジャンボフレームをサポートしています。

- プレシジョンタイムプロトコル (PTP)
- サービス品質 (QoS)
- ルーティング情報プロトコル (RIP)
- ルーティング情報プロトコル次世代 (RIPng)
- URLフィルタリング

fix

This is the content for the sample heading topic. It is hidden from the main table of contents but remains accessible via links.