net>scaler

NetScaler CPX 14.1

Machine translated content

Disclaimer

La version officielle de ce document est en anglais. Certains contenus de la documentation Cloud Software Group ont été traduits de façon automatique à des fins pratiques uniquement. Cloud Software Group n'exerce aucun contrôle sur le contenu traduit de façon automatique, qui peut contenir des erreurs, des imprécisions ou un langage inapproprié. Aucune garantie, explicite ou implicite, n'est fournie quant à l'exactitude, la fiabilité, la pertinence ou la justesse de toute traduction effectuée depuis l'anglais d'origine vers une autre langue, ou quant à la conformité de votre produit ou service Cloud Software Group à tout contenu traduit de façon automatique, et toute garantie fournie en vertu du contrat de licence de l'utilisateur final ou des conditions d'utilisation des services applicables, ou de tout autre accord avec Cloud Software Group, quant à la conformité du produit ou service à toute documentation ne s'applique pas dans la mesure où cette documentation a été traduite de façon automatique. Cloud Software Group ne pourra être tenu responsable de tout dommage ou problème dû à l'utilisation de contenu traduit de façon automatique.

Contents

À propos de NetScaler CPX	2
Architecture et flux de trafic	4
Licence NetScaler CPX	8
Déploiement d'une instance NetScaler CPX dans Docker	16
Ajouter des instances NetScaler CPX à Citrix ADM	24
Agrégateur de licences NetScaler CPX	29
Configuration de NetScaler CPX	35
Configuration d'AppFlow sur une instance NetScaler CPX	39
Configuration de NetScaler CPX à l'aide d'un fichier de configuration	41
Support du routage dynamique dans NetScaler CPX	43
Configuration des pilotes de journalisation Docker	46
Mise à niveau d'une instance NetScaler CPX	47
Utilisation de serveurs virtuels génériques dans une instance NetScaler CPX	49
Déployer NetScaler CPX en tant que proxy pour permettre un flux de trafic Est-Ouest	50
Déployer NetScaler CPX sur un réseau hôte unique	54
Déployer NetScaler CPX dans un réseau multi-hôtes	55
Déployez NetScaler CPX avec un accès direct au réseau	60
Configurer NetScaler CPX dans Kubernetes à l'aide de ConfigMaps	60
Déployez les CPX NetScaler en tant que caches DNS locaux pour les nœuds Kubernetes	63
Déployer le proxy NetScaler CPX sur Google Compute Engine	68
Résolution des problèmes liés à NetScaler CPX	88

À propos de NetScaler CPX

July 22, 2024

NetScaler CPX est un contrôleur de diffusion d'applications basé sur des conteneurs qui peut être provisionné sur un hôte Docker. NetScaler CPX permet aux clients de tirer parti des fonctionnalités du moteur Docker et d'utiliser les fonctionnalités d'équilibrage de charge et de gestion du trafic NetScaler pour les applications basées sur des conteneurs. Vous pouvez déployer une ou plusieurs instances NetScaler CPX en tant qu'instances autonomes sur un hôte Docker.

Une instance NetScaler CPX fournit un débit allant jusqu'à 1 Gbit/s.

En tant que facteur de forme conteneurisé de NetScaler, NetScaler CPX s'intègre parfaitement à l' environnement Kubernetes et fait partie intégrante de la solution native cloud NetScaler. La solution cloud native de NetScaler vous aide à créer et à fournir des applications logicielles avec rapidité, agilité et efficacité dans un environnement Kubernetes. Grâce à la solution cloud native NetScaler, vous pouvez garantir une fiabilité et une sécurité de niveau professionnel pour votre environnement Kubernetes.

Pour plus d'informations, consultez la solution cloud native NetScaler.

Ce document suppose que vous connaissiez Docker et son fonctionnement. Pour plus d'informations sur Docker, consultez la documentation e Docker sur https://docs.docker.com.

Fonctionnalités prises en charge

NetScaler CPX prend en charge les fonctionnalités suivantes :

- Disponibilité des applications
 - Équilibrage de charge L4 et commutation de contenu L7
 - Déchargement SSL
 - Traduction du protocole IPv6
 - Équilibrage de charge Microsoft SQL, MySQL
 - Contrôles de taux AppExpert
 - Pilotage du trafic sensible aux abonnés
 - Protection contre les surtensions et file d'attente prioritaire
 - Protocoles de routage dynamiques
- Accélération application
 - Optimisations TCP pour les clients et les serveurs
 - Redirection du cache
 - AppCompress

- AppCache
- Sécurité des applications
 - Réécriture et répondeur L7
 - Défenses DoS L4
 - Défenses DoS L7
 - Pare-feu d'application Web (WAF). NetScaler CPX prend en charge toutes les fonctionnalités WAF qui sont prises en charge sur d'autres formats NetScaler. Pour plus d'informations sur les fonctionnalités WAF prises en charge, consultez NetScaler WebApp Firewall.
 - Authentification, autorisation et audit (AAA) pour le trafic des applications
- Optimisation du protocole TCP
 - TCP à chemins multiples
 - Contrôle de congestion d'augmentation binaire (BIC) et TCP cubique
- Facilité de gestion simple
 - Journalisation Web
 - AppFlow
 - NetScaler Application Delivery Management
 - Analyse des actions
- Optimisation des applications
 - Mise en cache intégrée
- Routage BGP et injection d'intégrité de routage (RHI)

Remarque :

Les fonctionnalités d'interface telles que Rx, Tx, GRO, GSO et LRO sont désactivées pour les interfaces (hôte Linux) allouées à l'appliance NetScaler CPX. Ces fonctionnalités restent désactivées même après l'arrêt de l'appliance NetScaler CPX. En outre, le MTU est remplacé par 1500 octets pour de telles interfaces.

Plates-formes prises en charge

NetScaler CPX est pris en charge sur les plateformes suivantes :

- Kubernetes
- Red Hat OpenShift
- Clouds publics
 - Amazon Elastic Kubernetes Service (EKS)

- Azure Kubernetes Service (AKS)
- Google Kubernetes Engine (GKE)
- Rancher
- Pivotal Container Service (PKS)
- Docker version 1.12 et supérieure

Architecture et flux de trafic

November 23, 2023

Cette section décrit l'architecture en mode pont de NetScaler CPX et le flux de trafic. NetScaler CPX peut également être déployé en mode hôte.

Lorsque vous provisionnez une instance NetScaler CPX sur un hôte Docker, le moteur Docker crée une interface virtuelle, eth0, sur l'instance CPX. Cette interface eth0 est directement connectée à une interface virtuelle (veth*) sur le pont docker0. Le moteur Docker attribue également une adresse IP à l'instance NetScaler CPX sur le réseau 172.17.0.0/16.

La passerelle par défaut de l'instance CPX est l'adresse IP du pont docker0, ce qui signifie que toute communication avec l'instance NetScaler CPX se fait via le réseau Docker. Tout le trafic entrant reçu du pont docker0 est reçu par l'interface eth0 sur l'instance NetScaler CPX et traité par le moteur de paquets NetScaler CPX.

La figure suivante illustre l'architecture d'une instance NetScaler CPX sur un hôte Docker.



Fonctionnement de l'adresse IP unique sur NetScaler CPX

Une appliance NetScaler MPX ou VPX standard nécessite au moins trois adresses IP pour fonctionner :

- Adresse IP de gestion appelée adresse IP NetScaler (NSIP)
- Adresse IP du sous-réseau (SNIP) pour communiquer avec la batterie de serveurs
- Adresse (s) IP (VIP) du serveur virtuel pour accepter les demandes des clients

Une instance NetScaler CPX fonctionne avec une seule adresse IP utilisée pour la gestion ainsi que pour le trafic de données.

Lors du provisionnement, une seule adresse IP privée (adresse IP unique) est attribuée à une instance NetScaler CPX par le moteur Docker. Les trois fonctions IP d'une instance NetScaler sont multiplexées sur une adresse IP. Cette adresse IP unique utilise différents numéros de port pour fonctionner en tant que NSIP, SNIP et VIP.

L'image suivante illustre comment une seule adresse IP est utilisée pour exécuter les fonctions de NSIP, SNIP et VIP.



Flux de trafic pour les demandes provenant de l'instance NetScaler CPX

Docker configure implicitement des tables IP et une règle NAT pour diriger le trafic provenant de l' instance NetScaler CPX vers l'adresse IP docker0.

La figure suivante illustre comment une demande ping provenant d'une instance NetScaler CPX atteint la destination.



Dans cet exemple, la demande ping est envoyée par le moteur de paquets sur l'interface eth0 avec l' adresse IP source comme adresse IP NetScaler CPX (172.17.0.4). L'hôte Docker effectue ensuite une traduction d'adresse réseau (NAT) pour ajouter l'adresse IP de l'hôte (192.68.x.x) en tant qu'adresse IP source et envoie la demande à la destination (216.58.x.x). La réponse de l'adresse IP de destination suit le même chemin en sens inverse. L'hôte Docker effectue une NAT sur la réponse et transmet la réponse à l'instance NetScaler CPX sur l'interface eth0.

Flux de trafic pour les demandes provenant du réseau externe

Pour activer la communication externe, lors du provisionnement de NetScaler CPX, vous devez définir des paramètres tels que Docker expose certains ports tels que 80, 22 et tout autre port de votre choix. Si vous n'avez défini aucun port à exposer pendant le provisionnement, vous devez configurer des règles NAT sur l'hôte Docker pour rendre ces ports disponibles.

La demande client provenant d'Internet est reçue par l'hôte Docker, qui effectue ensuite une traduction d'adresse de port (PAT) pour mapper l'adresse IP publique et le port à l'adresse IP et au port uniques de l'instance NetScaler CPX, et transmet le trafic à l'instance.

La figure suivante montre comment l'hôte Docker effectue la traduction d'adresse de port pour diriger le trafic vers l'adresse IP unique et le port NetScaler CPX.



Dans cet exemple, l'adresse IP de l'hôte Docker est 192.68.x.x et l'adresse IP unique de l'instance NetScaler CPX est 172.17.0.4. Le port SSH 22 de l'instance NetScaler CPX est mappé au port 1100 sur l'hôte Docker. La demande SSH du client est reçue sur l'adresse IP 192.68.x.x sur le port 1100. L'hôte Docker effectue une traduction d'adresse de port pour mapper cette adresse et ce port à l'adresse IP unique 172.17.0.4 sur le port 22 et transmet la demande du client.

Licence NetScaler CPX

April 1, 2025

NetScaler CPX est un contrôleur de diffusion d'applications basé sur des conteneurs qui peut être provisionné sur un hôte Docker pour équilibrer la charge des applications basées sur des microservices. Vous avez besoin d'une licence CPX pour améliorer les performances de livraison des applications. NetScaler CPX prend en charge les licences de pool. Citrix ADM peut agir en tant que serveur de licences pour vos instances NetScaler CPX.

Citrix ADM est disponible à la fois sur site et en tant que service cloud. Vous pouvez utiliser Citrix ADM pour gérer les licences de capacité groupées pour tous les formats NetScaler.

Pour plus d'informations sur NetScaler ADM sur site, consultez NetScaler ADM sur site. Pour plus d'

informations sur Citrix ADM sur site, voir Citrix ADM sur site. Pour plus d'informations sur le service Citrix ADM, consultez le service NetScalerADM.

Types de licences NetScaler CPX

NetScaler CPX prend en charge les licences de bande passante et de pool de processeurs virtuels (core) pour les déploiements sur site et dans le cloud.

Pool de bande passante : les licences NetScaler CPX peuvent être attribuées en fonction de la consommation de bande passante par les instances. Vous pouvez utiliser les licences groupées pour optimiser l'utilisation de la bande passante en garantissant l'allocation de bande passante nécessaire à une instance et pas plus que ses exigences. Actuellement, NetScaler CPX ne prend en charge que les licences de pool de bande passante premium.

Pool de processeurs virtuels : dans la licence basée sur l'utilisation du processeur virtuel, la licence spécifie le nombre de processeurs auxquels une instance NetScaler CPX particulière est autorisée. Ainsi, le NetScaler CPX peut récupérer des licences uniquement pour le nombre de processeurs virtuels à partir du serveur de licences. NetScaler CPX vérifie les licences en fonction du nombre de processeurs en cours d'exécution sur le système. Pour plus d'informations sur le pool de processeurs virtuels, consultez la section Licences de processeurs virtuels NetScaler.

Produit	Bande passante maximale	Bande passante minimale	Instances minimales	Nombre maximum d' instances	Unité de bande passante minimale
NetScaler CPX	40000 Remarque : Cela dépend de la fréquence du processeur, de la génération, etc.	20 Mbits/s	1	16	10 Mbit/s

Capacité groupée prise en charge pour les instances NetScaler CPX

Remarque : Citrix travaille actuellement sur un modèle de licence NetScaler CPX basé sur la consommation ou basé sur le paiement au fur et à mesure de la croissance pour les offres basées sur le cloud public. Une fois prêt, il sera disponible sur le marché du cloud public pour être consommé.

Comment fonctionnent les licences NetScaler CPX ?

Capacité groupée NetScaler CPX : pool de licences commun à partir duquel votre instance NetScaler CPX peut récupérer une licence d'instance et uniquement la quantité de bande passante dont elle a besoin. Lorsque l'instance n'a plus besoin de ces ressources, elle les réintègre dans le pool commun, ce qui rend les ressources disponibles pour les autres instances qui ont besoin de ces licences.

Licences**d'enregistrement et de départ NetScaler CPX : Citrix ADM attribue des licences** aux instances NetScaler CPX à la demande. Une instance NetScaler CPX peut récupérer la licence auprès de Citrix ADM lorsqu'une instance NetScaler CPX est provisionnée et vérifier sa licence auprès de Citrix ADM lorsqu'une instance est détruite.

Comportement de NetScaler CPX : une seule instance NetScaler CPX extrait un débit allant jusqu' à 1 Gbit/s, extrait uniquement du pool d'instances et non du pool de licences de bande passante. NetScaler CPX fonctionne de cette manière jusqu'à 1 Gbit/s d'utilisation de la bande passante. Par exemple, si une instance CPX consomme une bande passante de 200 Mbit/s, elle utilise le pool d' instances de licence, au lieu du pool de bande passante. Toutefois, si une instance NetScaler CPX consomme 1 200 Mbit/s de débit, les 1 000 premiers Mbit/s sont utilisés depuis le pool d'instances et les 200 Mbit/s restants sont consommés depuis le pool de bande passante.

NetScaler CPX Express

NetScaler CPX Express est une édition logicielle gratuite pour les déploiements sur site et dans le cloud. Lorsque vous téléchargez une instance NetScaler CPX depuis le référentiel Quay, la licence NetScaler CPX Express est appliquée à l'instance CPX par défaut. Toutes les fonctionnalités de NetScaler CPX sont disponibles avec la licence NetScaler CPX Express. Pour plus d'informations sur les fonctionnalités prises en charge sur CPX, consultez À propos de NetScaler CPX. Le support de la communauté NetScaler est disponible avec NetScaler CPX Express mais pas le support pour les entreprises.

La licence NetScaler CPX Express offre une capacité de bande passante de 100 Mbit/s. Vous devez obtenir une licence pour votre instance NetScaler CPX pour une capacité accrue, un déploiement en production et pour bénéficier du support d'entreprise.

Remarque

À partir de NetScaler CPX version 14.1 build 14.1-17.38, la bande passante de la licence CPX Express est augmentée de 20 Mbps à 100 Mbps. Cette bande passante accrue permet aux développeurs de Kubernetes d'essayer le proxy NetScaler.

Modèles de licence NetScaler CPX

NetScaler propose une gamme de modèles de licences de produits pour NetScaler CPX afin de répondre aux exigences de votre organisation. Vous pouvez sélectionner des options telles que vCPU ou bande passante et sur site ou cloud.

En fonction de vos besoins, vous pouvez choisir l'un des modèles suivants :

- Licence basée sur la bande passante pour NetScaler CPX à partir du service ADM
- Licence basée sur un processeur virtuel pour NetScaler CPX à partir du service ADM
- Licence basée sur la bande passante pour NetScaler CPX auprès d'ADM sur site
- Licences basées sur vCPU pour NetScaler CPX auprès d'ADM sur site

Fournissez des licences basées sur la bande passante et sur vCPU à partir du service Citrix ADM pour NetScaler CPX

Procédez comme suit pour fournir une licence basée sur la bande passante et une licence basée sur vCPU pour NetScaler CPX à partir du service Citrix ADM.

1. Configurez Citrix ADM.

Assurez-vous que la configuration du service Citrix ADM est opérationnelle avec l'agent Citrix ADM. Vous devez disposer d'un service Citrix ADM et d'un compte d'agent Citrix ADM pour que la licence NetScaler CPX soit fonctionnelle. Pour plus d'informations sur la configuration du service Citrix ADM et de l'agent Citrix ADM, consultez le service Citrix ADM.

Remarque : Dans cet exemple, un agent ADM intégré avec ADM local est utilisé. Dans l'image suivante, vous pouvez voir qu'aucun agent n'est déployé.

Q Search in Menu	< :	Networks > Ag	rents									-	
Applications	>	Agonto							Set Up Agent	Satting	a Activation Code	<u>a</u> + (2 5
Networks	\sim	Agents	9						Dec of Agent	General	e Activation code		
Infrastructure Analytics		View Details	Delete Reboo	t Rediscover Att	ach Site View Fit	ngerprint	Provision	Select Action $$					¢
Instances Citrix ADC	۵	Q Click here to	search or you can enter	Key : Value format									0
Citrix Gateway			IP ADDRESS	HOST NAME	VERSION 0	STATE *	PLATFORM 0	CPU USAGE (%)	DISK USAGE (%)	MEMORY USAGE (%) COUNTRY C	REGION	CIT
Citrix SD-WAN Preview			10.106.102.199	akshay-199	13.0-82.22	Up	KVM	3	22	15	-	-	
HAProxy Preview			10.98.161.89	akshay-adm-agent	13.0-77.26 (• Up	Kubernetes	100	47		-		
Instance Advisory	>	Total 2								25	Per Page V Page	1 of 1	< >
Instance Groups													
AutoScale Groups													
☆ Agents													
Licenses	>												
Events	>												
SSL Dashboard	>												
Configuration Jobs	>												
Configuration Audit	>												
Sites													
AppEnvironments													
IPAM													
Domain Names													
Network Functions	>												
Network Reporting													
Analytics	>												
Orchestration	>												
Account	>												
Settings	>												
Help Center													

2. Ajoutez un pool de licences d'instance NetScaler au service Citrix ADM.

Il est supposé que vous disposez d'un pool de licences de bande passante disponibles pour le service ADM. Pour plus d'informations sur le téléchargement d'un fichier de licence vers Citrix ADM, consultez la section Configurer la capacité groupée. Dans l'image suivante, CNS_INST_200CC_Retail.lic est utilisé comme bande passante et pool de licences d' instance. Dans l'image suivante, CNS_INST_200CC_Retail.lic est utilisé comme bande passante et pool de licences d'instance.

Q Search in Menu	<:	Networks > License Settings						?
Applications	>	License Files						
Networks	\sim	2100110011100						
Infrastructure Analytics		The following license files are preser To manually Download licenses from	nt on this server. Select Add New License to Citrix licensing portal please visit http://ww	o upload more licenses. To delete a license, selec ww.mycitrix.com and use the Host ID: fedf832136	t the license and click Delete. Nee			
Instances	>							
Instance Advisory	>	Add License File Apply Licen	Delete Download					
Instance Groups		Q Click here to search or you can er	iter Key : Value format					1
AutoScale Groups		NAME			LAST MODIFIED		© SIZE	
Agents		CNS_INST_5CCS_Re	taiLlic		Mon Sep 07 2020 8:22 PM		1.03 KB	
☆ Licenses	>	CNS_PBW10MB_500	B_RetailS.lic		Thu Aug 20 2020 1:52 PM		1.22 KB	
Events	>	CNS_CP1000_100SS	ERVER_RetailS.lic		Mon Sep 07 2020 8:22 PM 1.04 KB			
SSL Dashboard	>	CNS_PBW10MB_10G	B_RetailS.lic		Mon Sep 07 2020 8:22 PM 1.24 KB			
Configuration Jobs	>	CNS_1VCPUP_100SS	ERVER_RetailS.lic		Thu Aug 20 2020 2:20 PM		1.13 KB	
Configuration Audit	>	Total 5				25 Per Page	✓ Page 1 of 1	
Sites								
AppEnvironments		License Expiry Information						
IPAM		FEATURE	COU	NT	DAYS TO EXPIRY			
Domain Names		No items						
Network Functions	>							
Network Reporting		Notification Settings						/
Analytics	>							
Orchestration	>	Email Profile	Slack Profile	PagerDuty Profile	ServiceNow Profile	Alert Threshold	Days To Expiry	
Account	>	Not configured	Not comgarea	Not comiguieu	Not compared	0'0'70	30	
Settings	>							
Help Center								
1								

3. Déployez l'instance NetScaler CPX dans le cluster Kubernetes. Assurez-vous que les variables d'environnement suivantes sont ajoutées au fichier YAML NetScaler CPX pour attribuer une licence à l'instance NetScaler CPX.

Pour les licences basées sur la bande passante du service Citrix ADM, spécifiez les variables d' environnement suivantes dans le fichier YAML :

- nom : « LS_IP » valeur : « 10.105.158.166 » // IP de l'agent ADM comme mentionné à l' étape 1
- nom : « LS_PORT » valeur : « 27000 » // port sur lequel le serveur de licences ADM écoute
- nom : « BANDE PASSANTE » valeur : « 3 000 » // capacité en Mbps que l'on souhaite allouer à CPX
- nom : « ÉDITION » valeur : « Standard » ou « Enterprise » // pour choisir une édition de licence particulière qui inclut Standard, Platinum et Enterprise. Par défaut, Platine est sélectionné.

Pour les licences basées sur vCPU du service Citrix ADM, spécifiez les variables d'environnement suivantes dans le fichier YAML :

- nom : « LS_IP » valeur : « 10.102.216.173 » // IP de l'agent ADM comme mentionné à l' étape 1
- nom : « LS_PORT » valeur : « 27000 » // port sur lequel le serveur de licences ADM écoute
- nom : « CPX_CORES » valeur : « 4 » // nombre de cœurs que vous souhaitez allouer
- nom : « PLATEFORME » valeur : « CP1000 » // nombre de cœurs. Le nombre de sorties est égal au nombre de cœurs.

4. Téléchargez le cpx-bandwidth-license-adm-service.yaml fichier à l'aide de la commande suivante :



5. Déployez le YAML modifié dans le cluster Kubernetes à l'aide de la commande suivante :

```
1 kubectl create -f cpx-bandwidth-license-adm-service.yaml -n
bandwidth
```

6. Connectez-vous à NetScaler CPX pour vérifier les informations d'instanciation à l'aide de la commande suivante :

1 kubectl exec -it 'cpx-pod-ip-name' bash -n bandwidth

7. Pour afficher les informations de licence pour l'instance NetScaler CPX donnée, exécutez les commandes suivantes :

```
1 cli_script.sh "show licenseserver"
2 cli_script.sh "show capacity"
```

Vous pouvez suivre la bande passante allouée et la capacité du processeur virtuel dans le portail de services ADM.

Fournir des licences basées sur la bande passante et des licences basées sur le processeur virtuel pour NetScaler CPX à partir de Citrix ADM sur site

Procédez comme suit pour approvisionner NetScaler CPX en fonction de la bande passante et du vCPU à partir de Citrix ADM sur site.

1. Configurez Citrix ADM.

Assurez-vous que la configuration locale d'ADM est prête. Assurez-vous que Citrix ADM sur site avec ou sans déploiement de l'agent ADM pour les licences NetScaler CPX fonctionne.

Pour plus d'informations sur la configuration de Citrix ADM sur site et de l'agent Citrix ADM, consultez le service Citrix ADM.

Remarque : Dans cette procédure, une configuration de l'agent Citrix ADM d'un hyperviseur (sur site) est utilisée. Dans l'image suivante, vous pouvez voir qu'aucun agent n'est déployé.

Q. Search in Menu		Networks > Agents		
* Favorites	>	Agents 💿	C 0	[2] *
≡ Menu	\sim	View Details Delete Reboot Attach Site View Finoerorint No artion		ö
Applications	>			~
Networks	\sim	Q Click here to search or you can enter Key : Value format		0
Infrastructure Analytics		IP ADDRESS O HOST NAME O VERSION O STATE O PLATFORM O COUNTRY O REGION O CITY	© SITE	¢
Instances	>	No items		
Instance Groups				
û Agents				
Licenses	>			
Events	>			
SSL Dashboard	>			
Configuration Jobs	>			
Configuration Audit	>			
Sites				
IPAM				
Domain Names				
Network Functions	>			
Network Reporting	>			
Analytics	>			
Orchestration	>			
System	>			
Help Center				

2. Ajoutez un pool de licences d'instance NetScaler à ADM sur site.

Il est supposé que vous disposez d'un pool de licences de bande passante disponible pour ADM sur site. Pour en savoir plus sur le téléchargement d'un fichier de licence vers Citrix ADM, consultez la section Gestion de licences. Dans l'image suivante, CNS_INST_200CC_Retail. lic est utilisé comme bande passante et pool de licences d'instance. Dans l'image suivante, CNS_INST_200CC_Retail.lic est utilisé comme bande passante et pool de licences d' instance.

Q. Search in Menu		Networks > License Settings		0
★ Favorites	>	License Server Port Settings		1
≔ Menu	~	Proxy Server Port 0	License Server Port 27000	Vendor Daemon Port 7279
Applications Networks	~	The Proxy Server Port used by Citrix ADC instances to access the Citrix licensing portal for license allocation	The License Server Port used by Citrix ADC instances to communicate with the license server	The Daemon Port used by Citrix ADC instances to communicate with the license server
Infrastructure Analytics				
Instances	>	License Files		
Agents		The following license files are present on this server. Select Add New License to uplo To manually Download licenses from Citrix licensing portal please visit http://www.m	ad more licenses. To delete a license, select the license and click Delete. wcitrix.com and use the Host ID: 02a208e8fa59	
☆ Licenses	ه>	Add License File Apply Licenses Delete Download		
Events SSL Dashboard	>	Q Click here to search or you can enter Key : Value format		0
Configuration Jobs	<i></i> >	NAME	LAST MODIFIED	≎ SIZE ≎
Configuration Audit	>	CNS_EBW10MB_200GB_Eval.lic	Mon Apr 26 2021 12:17 PM	1.24 KB
Sites		CNS_SBW10MB_500GB_Retail.lic	Mon Apr 26 2021 12:17 PM	1.17 KB
IPAM		CNS_PBW10MB_200GB_Eval.lic	Mon Apr 26 2021 12:17 PM	1.23 KB
Domain Names		CNS_CP1000_100SSERVER_RetailS.lic	Mon Apr 26 2021 12:17 PM	1.04 KB
Network Functions	>	CNS_PBW10MB_50GB_RetailS.lic	Mon Apr 26 2021 12:17 PM	1.24 KB
Network Reporting	~	CNS_INST_200CCS_Retail.lic	Mon Apr 26 2021 12:17 PM	1.04 KB
Analytics	>	Total 6		25 Per Page 💙 Page 1 of 1 🔺 🕨
System	>	License Evning Information		
Help Center		FFATURE	COUNT 0	DAYS TO EXPIRY
		Platinum Bandwidth	250,000	219
		Enterprise Bandwidth	200.000	219
		Instance	200 2	219
		Standard Bandwidth	500,000	219
		Total 4		25 Per Page 🗸 Page 1 of 1 🚽 🕨

Dans l'image suivante, CP1000 est utilisé comme pool de licences vCPU.

Q Search in Menu		Networks > License Settings		0			
★ Favorites	>	License Server Port Settings		1			
≔ Menu	~						
Applications	>	Proxy Server Port 0	License Server Port 27000	Vendor Daemon Port 7279			
Networks	~	licensing portal for license allocation	the license server	license server			
Infrastructure Analytics							
Instances	>	License Files					
Instance Groups Agents		The following license files are present on this server. Select Add New License to upload more licenses. To delete a license, select the license and click Delete. To manually Download licenses from Citrix licensing portal please with http://www.mycitrix.com and use the Hox ID 02a208e8fa59					
1 Licenses	ه>	Add License File Apply Licenses Delete Developed					
Events	>	Add License File Apply Licenses Delete Download					
SSL Dashboard	>	Q Click here to search or you can enter Key : Value format		٢			
Configuration Jobs	>	NAME	LAST MODIFIED	♦ SIZE ♦			
Configuration Audit	>	CNS_EBW10MB_200GB_Eval.lic	Mon Apr 26 2021 12:29 PM	1.24 KB			
Sites		CNS_SBW10MB_500GB_Retail.lic	Mon Apr 26 2021 12:29 PM	1.17 KB			
IPAM		CNS_PBW10MB_200GB_Eval.lic	Mon Apr 26 2021 12:29 PM	1.23 KB			
Domain Names		CNS_CP1000_100SSERVER_RetailS.lic	Mon Apr 26 2021 12:29 PM	1.04 KB			
Network Functions	>	CNS_1VCPUP_100SSERVER_RetailS.lic	Mon Apr 26 2021 12:29 PM	1.15 KB			
Network Reporting	>	CNS_PBW10MB_50GB_RetailS.lic	Mon Apr 26 2021 12:29 PM	1.24 KB			
Analytics	>	CNS_INST_200CCS_Retail.lic	Mon Apr 26 2021 12:29 PM	1.04 KB			
Orchestration	>	Total 7		25 Per Page 🗸 Page 1 of 1 🔺 🕨			
System Holo Canter							
help center		License Expiry Information					
		FEATURE	COUNT COUNT	AYS TO EXPIRY 0			
		Standard Bandwidth	500,000 2	19			
		Instance	200 2	19			
		Platinum Bandwidth	250,000 2	19			
		Enterprise Bandwidth	200,000 2	19			
		Total 4		25 Per Page ∨ Page 1 of 1 < >			

3. Déployez l'instance NetScaler CPX dans le cluster Kubernetes. Assurez-vous que les variables d'environnement suivantes sont ajoutées au fichier YAML NetScaler CPX pour attribuer une licence à l'instance NetScaler CPX.

Pour les licences basées sur la bande passante depuis Citrix ADM sur site, spécifiez les variables d'environnement suivantes dans le fichier YAML :

- nom : « LS_IP » valeur : « 10.105.158.144 » // IP de l'instance ADM sur site. Si vous avez déployé l'agent ADM, il s'agit de l'adresse IP de votre agent, comme décrit à l'étape 1.
- nom : « LS_PORT » valeur : « 27000 » // port sur lequel le serveur de licences ADM écoute
- nom : « BANDE PASSANTE » valeur : « 3 000 » // capacité en Mbps que l'on souhaite allouer à CPX

Pour les licences basées sur vCPU depuis Citrix ADM sur site, spécifiez les variables d'environnement suivantes dans le fichier YAML :

- nom : « LS_IP » valeur : « 10.105.158.144 » // IP de l'instance ADM sur site. Si vous disposez d'un déploiement d'agent ADM, il s'agira de l'IP de votre agent, comme décrit à l'étape 1.
- nom : « LS_PORT » valeur : « 27000 » // port sur lequel le serveur de licences ADM écoute
- nom : « CPX_CORES » valeur : « 4 » // le nombre de cœurs que vous souhaitez allouer
- nom : « PLATEFORME » valeur : « CP1000 » // nombre de cœurs. Le nombre de sorties est égal au nombre de cœurs.
- 4. Téléchargez le cpx-bandwidth-license-adm-onprem.yaml fichier à l'aide de la commande suivante :

1 kubectl create namespace bandwidth

- 2 wget https://raw.githubusercontent.com/citrix/cloud-nativegetting-started/master/cpx-licensing/manifest/cpx-bandwidthlicense-adm-onprem.yaml
- 5. Déployez le YAML modifié dans le cluster Kubernetes à l'aide de la commande suivante :

```
1 kubectl create -f cpx-bandwidth-license-adm-onprem.yaml -n
bandwidth
```

6. Connectez-vous à NetScaler CPX pour vérifier les informations d'instanciation à l'aide de la commande suivante :

```
1 kubectl exec -it <cpx-pod-ip-name> bash -n bandwidth
```

7. Pour consulter les informations de licence de l'instance NetScaler CPX, exécutez les commandes suivantes :

```
1 cli_script.sh "show licenseserver"
2 cli_script.sh "show capacity"
```

Vous pouvez suivre la bande passante allouée et la capacité du processeur virtuel sur le portail local ADM.

Commandes de nettoyage des déploiements

Vous pouvez utiliser les commandes suivantes pour nettoyer les différents déploiements YAML :

```
1 kubectl delete -f cpx-bandwidth-license-adm-service.yaml -n bandwidth
2 kubectl delete -f cpx-core-license-adm-service.yaml -n core
3 kubectl delete -f cpx-bandwidth-license-adm-onprem.yaml -n bandwidth
4 kubectl delete -f cpx-core-license-adm-onprem.yaml -n core
5 kubectl delete namespace bandwidth
6 kubectl delete namespace core
```

Déploiement d'une instance NetScaler CPX dans Docker

April 1, 2025

Les instances NetScaler CPX sont disponibles sous forme de fichier image Docker dans le registre de conteneurs Quay. Pour déployer une instance, téléchargez l'image NetScaler CPX depuis le registre de conteneurs Quay, puis déployez l'instance à l'aide de la docker run commande ou de l'outil de composition Docker.

Prérequis

Assurez-vous que :

- Le système hôte Docker possède au moins :
 - 1 PROCESSEUR
 - 2 Go de RAM

Remarque :

Pour de meilleures performances NetScaler CPX, vous pouvez définir le nombre de moteurs de traitement que vous souhaitez que l'instance NetScaler CPX démarre. Pour chaque moteur de traitement supplémentaire que vous ajoutez, assurez-vous que l'hôte Docker contient le nombre équivalent de vCPU et la quantité de mémoire en Go. Par exemple, si vous souhaitez ajouter 4 moteurs de traitement, l'hôte Docker doit contenir 4 vCPU et 4 Go de mémoire.

- Le système hôte Docker exécute Linux Ubuntu version 14.04 ou ultérieure.
- La version 1.12 de Docker est installée sur le système hôte. Pour plus d'informations sur l'installation de Docker sur Linux, consultez ladocumentation Docker.
- L'hôte Docker dispose d'une connexion Internet.

Remarque : NetScaler CPX rencontre des problèmes lors de l'exécution sur Ubuntu version 16.04.5, version du noyau 4.4.0-131-generic. Il n'est donc pas recommandé d'exécuter NetScaler CPX sur le noyau Ubuntu version 16.04.5 version 4.4.0-131-generic.

Remarque: Les versions suivantes de kubelet et kube-proxy présentent certaines vulnérabilités de sécurité et il n'est pas recommandé d'utiliser NetScaler CPX avec ces versions :

- kubelet/kube-proxy v1.18.0-1.18.3
- kubelet/kube-proxy v1.17.0-1.17.6
- kubelet/kube-proxy <=1.16.10</p>

Pour plus d'informations sur la façon d'atténuer cette vulnérabilité, consultez Atténuer cette vulnérabilité.

Téléchargement de l'image NetScaler CPX depuis Quay

Vous pouvez télécharger l'image NetScaler CPX depuis le registre de conteneurs Quay à l'aide de la docker pull commande et la déployer sur votre environnement. Utilisez la commande suivante pour télécharger l'image NetScaler CPX depuis le registre de conteneurs Quay :

1 docker pull quay.io/citrix/citrix-k8s-cpx-ingress:13.0-xx.xx

Par exemple, si vous souhaitez télécharger la version 13.0-64.35, utilisez la commande suivante :

1

docker pull quay.io/citrix/citrix-k8s-cpx-ingress:13.0-64.35

Utilisez la commande suivante pour vérifier si l'image NetScaler CPX est installée dans les images Docker :

1root@ubuntu:~# docker images | grep 'citrix-k8s-cpx-ingress'2quay.io/citrix/citrix-k8s-cpx-ingress13.0-64.35952a04e731012 months ago469 MB

Vous pouvez déployer la dernière image NetScaler CPX à partir du registre de conteneurs Quay.

Déploiement de l'instance NetScaler CPX à l'aide de la commande docker run

Sur l'hôte, vous pouvez installer une instance NetScaler CPX dans le conteneur Docker à l'aide de l'image Docker NetScaler CPX que vous avez chargée sur l'hôte. À l'aide de la docker run commande, installez l'instance NetScaler CPX avec la configuration NetScaler CPX par défaut.

Installez l'instance NetScaler CPX sur le conteneur Docker à l'aide de la commande docker run suivante :

Cet exemple crée un conteneur nommé enmycpxfonction de l'image Docker NetScaler CPX.

Le paramètre – P est obligatoire. Il indique à Docker de mapper les ports exposés dans le conteneur par l'image Docker NetScaler CPX. Cela signifie mapper les ports 9080, 22, 9443 et 161/UDP aux ports de l'hôte Docker sélectionnés de manière aléatoire dans la plage définie par l'utilisateur. Ce mappage est fait pour éviter les conflits. Si vous créez ultérieurement plusieurs conteneurs NetScaler CPX sur le même hôte Docker. Les mappages de ports sont dynamiques et sont définis chaque fois que le conteneur est démarré ou redémarré. Les ports sont utilisés comme suit :

- 9080 est utilisé pour HTTP
- 9443 est utilisé pour les HTTPs
- 22 utilisés pour SSH

• 161/UDP est utilisé pour le SNMP.

Si vous souhaitez des mappages de ports statiques, utilisez le paramètre -p pour les définir manuellement.

L'option --privileged=**true** permet d'exécuter le conteneur en mode privilégié. Si vous exécutez le NetScaler CPX en mode de déploiement hôte, vous devez fournir tous les privilèges système au NetScaler CPX.

Pour exécuter le NetScaler CPX en mode pont avec un ou plusieurs cœurs, vous pouvez utiliser l'option --cap-add=NET_ADMIN au lieu du mode privilégié. --cap-add=NET_ADMIN L' option --cap-add=NET_ADMIN vous permet d'exécuter le conteneur NetScaler CPX avec des privilèges réseau complets. Avec l'option --cap-add=NET_ADMIN, utilisez l'option --sysctl kernel.shmmax=1073741824 --sysctl net.ipv6.conf.default.accept_dad

=0 --sysctl kernel.core_pattern=/var/crash/core.%e.%p.%s de la commande docker run pour définir manuellement les paramètres de contrôle système supplémentaires. Ces paramètres de contrôle système supplémentaires ne sont pas définis automatiquement dans les modes non privilégiés.

**--net=host Il s'agit d'une option de commande d'exécution du menu fixe standard qui spécifie que le conteneur est en cours d'exécution dans la pile du réseau hôte et a accès à tous les périphériques réseau.

Remarque

Ignorez cette option si vous exécutez NetScaler CPX sur un pont ou sur aucun réseau.

-e NS_NETMODE="HOST" Il s'agit d'une variable d'environnement spécifique à NetScaler CPX qui vous permet de spécifier que NetScaler CPX est démarré en mode hôte. Une fois que NetScaler CPX démarre en mode hôte, il configure 4 règles iptables par défaut sur une machine hôte pour l'accès de gestion au NetScaler CPX. Il utilise les ports suivants :

- 9995 pour HTTP
- 9996 pour HTTPS
- 9997 pour SSH
- 9998 pour SNMP

Si vous souhaitez spécifier différents ports, vous pouvez utiliser les variables d'environnement suivantes :

- -e NS_HTTP_PORT=
- -e NS_HTTPS_PORT=
- -e NS_SSH_PORT=
- -e NS_SNMP_PORT=

Remarque

Ignorez cette variable d'environnement si vous exécutez NetScaler CPX sur un pont ou sur aucun réseau.

– e CPX_CORESII s'agit d'une variable d'environnement optionnelle spécifique à NetScaler CPX.
 Vous pouvez l'utiliser pour améliorer les performances de l'instance NetScaler CPX en définissant le nombre de moteurs de traitement que vous souhaitez que le conteneur NetScaler CPX démarre.

Remarques:

- Pour chaque moteur de traitement supplémentaire que vous ajoutez, assurez-vous que l' hôte Docker contient le nombre équivalent de vCPU et la quantité de mémoire en Go. Par exemple, si vous souhaitez ajouter 4 moteurs de traitement, l'hôte Docker doit contenir 4 vCPU et 4 Go de mémoire.
- Le NetScaler CPX peut prendre en charge de 1 à 16 cœurs.

La variable d'environnement-e EULA = yesest une variable d'environnement spécifique à NetScaler CPX obligatoire, qui est requise pour vérifier que vous avez lu et compris le contrat de licence utilisateur final (CLUF) disponible à l'adresse :Les composants du système de licences Citrix.

Le-e PLATFORM=CP1000paramètre spécifie le type de licence NetScaler CPX.

Si vous exécutez Docker sur un réseau hôte, vous pouvez attribuer des interfaces réseau dédiées au conteneur NetScaler CPX à l'aide de la variable d'environnement. –e CPX_NW_DEV Vous devez définir les interfaces réseau séparées par un espace blanc. Les interfaces réseau que vous définissez sont conservées par le conteneur NetScaler CPX jusqu'à ce que vous le désinstalliez. Lorsque le conteneur NetScaler CPX est provisionné, toutes les interfaces réseau attribuées sont ajoutées à l'espace de noms réseau NetScaler.

Remarque

Si vous exécutez NetScaler CPX dans un réseau de pont, vous pouvez modifier le réseau de conteneurs, par exemple configurer une autre connexion réseau au conteneur ou supprimer un réseau existant. Assurez-vous ensuite de redémarrer le conteneur NetScaler CPX pour utiliser le réseau mis à jour.

1 docker run -dt --privileged=true --net=host -e NS_NETMODE="HOST" -e
EULA=yes -e CPX_NW_DEV='eth1 eth2' -e CPX_CORES=5 -e PLATFORM=
CP1000 --name cpx_host cpx:13.0-x.x

-e CPX_CONFIG Il s'agit d'une variable d'environnement spécifique à NetScaler CPX qui vous permet de contrôler les performances de débit du conteneur NetScaler CPX. Lorsque le NetScaler CPX ne reçoit aucun trafic entrant à traiter, il libère le processeur pendant cette période d'inactivité, ce qui se traduit par de faibles performances de débit. Vous pouvez utiliser la variable d'environnement CPX_CONFIG pour contrôler les performances de débit du conteneur NetScaler CPX dans de tels scénarios. Vous devez fournir les valeurs suivantes à la variable d'CPX_CONFIG environnement au format JSON :

- Si vous souhaitez que le conteneur NetScaler CPX génère du processeur dans des scénarios d' inactivité, définissez { "YIELD" : "Yes" }
- Si vous souhaitez que le conteneur NetScaler CPX évite de surcharger le processeur dans des scénarios d'inactivité afin d'obtenir des performances de débit élevées, définissez { "YIELD

```
": "No" }
```

1

2

3

```
docker run -dt --privileged=true --net=host -e NS_NETMODE="HOST" -e
EULA=yes -e CPX_CORES=5 - e CPX_CONFIG='{
"YIELD":"No" }
' -e PLATFORM=CP1000 --name cpx_host cpx:13.0-x.x
```

```
1 docker run -dt --privileged=true --net=host -e NS_NETMODE="HOST" -e
	EULA=yes -e CPX_CORES=5 -e CPX_CONFIG='{
2 "YIELD":"Yes" }
3 ' -e PLATFORM=CP1000 --name cpx_host cpx:13.0-x.x
```

Le – vparamètre est un paramètre facultatif qui spécifie le point de montage du répertoire de montage NetScaler CPX./cpx Un point de montage est un répertoire sur l'hôte, dans lequel vous montez le répertoire /cpx. Un point de montage est un répertoire sur l'hôte, dans lequel vous montez le répertoire/cpx. Le répertoire /cpx stocke les journaux, les fichiers de configuration, les certificats SSL et les fichiers de vidage de mémoire. Dans l'exemple, le point de montage est/var/cpx et le répertoire de montage NetScaler CPX est./cpx

Si vous avez acheté une licence ou que vous possédez une licence d'évaluation, vous pouvez télécharger la licence sur un serveur de licences et spécifier l'emplacement du serveur de licences à l'aide de la commande docker run, à l'aide du paramètre – e LS_IP=<LS_IP_ADDRESS> – e LS_PORT=<LS_PORT>. Dans ce cas, il n'est pas nécessaire d'accepter le CLUF.

Où:

- LS_IP_ADDRESS est l'adresse IP du serveur de licences.
- LS_PORT est le port du serveur de licences.

Vous pouvez afficher les images exécutées sur votre système et les ports mappés aux ports standard à l'aide de la commande : docker ps

Déploiement d'une version allégée de NetScaler CPX à l'aide de la commande docker run

NetScaler fournit une version allégée de NetScaler CPX qui consomme moins de mémoire d'exécution. La version allégée de NetScaler CPX peut être déployée en tant que sidecar dans les déploiements de maillage de services.

La version allégée de NetScaler CPX prend en charge les fonctionnalités suivantes :

- Disponibilité des applications
 - Équilibrage de charge L4 et commutation de contenu L7
 - Déchargement SSL
 - Traduction du protocole IPv6
- Sécurité des applications
 - Réécriture et répondeur L7
- Facilité de gestion simple
 - Journalisation Web
 - AppFlow

Pour instancier la version allégée de NetScaler CPX, définissez la variable d'environnement NS_CPX_LITE lors de l'exécution de la commande Docker run.

L'exemple suivant crée un conteneur léger basé sur l'image NetScaler CPX.

Par défaut, l'utilisation de la journalisation newnslog est désactivée sur la version allégée de NetScaler CPX. Pour l'activer, vous devez définir la variable d'NS_ENABLE_NEWNSLOG environnement sur 1 tout en faisant apparaître la version allégée de NetScaler CPX.

L'exemple suivant montre comment activer la journalisation lors du déploiement de la version allégée de NetScaler CPX. newnslog

Remarque : La version allégée de CPX ne prend en charge que le monocœur (CPX_CORES=1).

Déploiement d'instances NetScaler CPX à l'aide de Docker Compose

Vous pouvez utiliser l'outil Compose de Docker pour mettre en service une seule instance NetScaler CPX ou plusieurs instances NetScaler CPX. Pour provisionner des instances NetScaler CPX à l'aide de Docker Compose, vous devez d'abord écrire un fichier de composition. Ce fichier spécifie l'image NetScaler CPX, les ports que vous souhaitez ouvrir pour l'instance NetScaler CPX et les privilèges de votre instance NetScaler CPX.

Important

Vérifiez que vous avez installé l'outil Docker Compose sur l'hôte.

Pour provisionner plusieurs instances NetScaler CPX :

- 1. Écrivez un fichier de composition, où :
- <service-name> est le nom du service que vous souhaitez mettre en service.
- image:<repository>:<tag> indique le référentiel et les versions de l'image NetScaler CPX.
- privileged : true fournit tous les privilèges root à l'instance NetScaler CPX.
- **cap_add** fournit des privilèges réseau à l'instance NetScaler CPX.
- <host_directory_path> indique le répertoire sur l'hôte Docker que vous souhaitez monter pour l'instance NetScaler CPX.
- <number_processing_engine> est le nombre de moteurs de traitement que vous souhaitez faire démarrer par l'instance NetScaler CPX. Pour chaque moteur de traitement supplémentaire, assurez-vous que l'hôte Docker contient le nombre équivalent de vCPU et la quantité de mémoire en Go. Par exemple, si vous souhaitez ajouter 4 moteurs de traitement, l'hôte Docker doit contenir 4 vCPU et 4 Go de mémoire.

Le fichier de composition suit généralement un format similaire à :

```
1
         <service-name>:
2
         container_name:
3
         image: <repository>:<tag>
         ports:
4
5
                 22
6
              - 9080
7
              - 9443
8
              - 161/udp
              _
9
                 35021-35030
         tty: true
11
         cap_add:
12
              - NET ADMIN
         ulimits:
13
14
             core: -1
15
         volumes:
16
             - <host_directory_path>:/cpx
17
         environment:
18

    EULA=yes
```

```
19 - CPX_CORES=<number_processing_engine>
20 - CPX_CONFIG='{
21 "YIELD":"Yes" }
22 '
```

```
CPX_0:
1
2
          image: quay.io/citrix/citrix-k8s-cpx-ingress:13.1-37.38
3
          ports:
4
                 9443
              _
5
                 22
6
              _
                 9080
7
                 161/udp
          tty: true
8
9
          cap_add:
10

    NET_ADMIN

11
          ulimits:
12
              core: -1
13
          volumes:
14
              - /root/test:/cpx
15
        environment:
16
              - CPX_CORES=2
17
              - EULA=yes
```

Ajouter des instances NetScaler CPX à Citrix ADM

March 21, 2024

Vous devez ajouter les instances NetScaler CPX installées sur un hôte Docker au logiciel NetScaler Application Delivery Management (ADM) si vous souhaitez gérer et surveiller ces instances.

Vous pouvez ajouter des instances lors de la configuration d'ADM pour la première fois ou ultérieurement.

Pour ajouter des instances, vous devez créer un profil d'instance et spécifier le nom d'hôte ou l' adresse IP de chaque instance, ou une plage d'adresses IP. Ce profil d'instance contient le nom d' utilisateur et le mot de passe des instances que vous souhaitez ajouter à Citrix ADM. Pour chaque type d'instance, un profil par défaut est disponible. Par exemple, ns-root-profile est le profil par défaut pour les instances NetScaler. Ce profil est défini par les informations d'identification de l'administrateur ADC par défaut. Si vous avez modifié les informations d'identification d'administrateur par défaut de vos instances, vous pouvez définir des profils d'instance personnalisés pour ces instances. Si vous modifiez les informations d'identification d'aue instance après sa découverte, vous devez modifier le profil d'instance ou créer un profil, puis redécouvrir l'instance.

Conditions préalables

Assurez-vous d'avoir :

- Installation du logiciel Citrix ADM sur Citrix XenServer. Pour plus d'informations, consultez la documentation Citrix ADM.
- Vous avez installé les instances NetScaler CPX sur un hôte Docker.

Pour ajouter des instances NetScaler CPX à ADM :

- 1. Dans un navigateur Web, saisissez l'adresse IP de **NetScaler Application Delivery** Management (par exemple, http://192.168.100.1).
- 2. Dans les champs **Nom d'utilisateur** et **Mot de passe**, entrez les informations d'identification de l'administrateur. Les informations d'identification de l'administrateur par défaut sont **nsroot** et **nsroot**.
- 3. Accédez à Réseaux > Instances > NetScaler et cliquez sur l'onglet CPX.
- 4. Cliquez sur **Ajouter** pour ajouter de nouvelles instances CPX dans Citrix ADM.
- 5. La page Ajouter NetScaler CPX s'ouvre. Entrez les valeurs pour les paramètres suivants :
 - a) Vous pouvez ajouter des instances CPX en fournissant l'adresse IP accessible de l'instance CPX ou l'adresse IP du conteneur Docker où l'instance CPX est hébergée.
 - b) Sélectionnez le profil de l'instance CPX.
 - c) Sélectionnez le site sur lequel les instances doivent être déployées.
 - d) Sélectionnez l'agent.
 - e) En option, vous pouvez entrer la paire clé-valeur de l'instance. L'ajout d'une paire clévaleur vous permet de rechercher facilement l'instance ultérieurement.

Enter Device IP Address	Import f	rom file					
Enter one or more hostnames	, IP addresses , a	and/or a range	of IP addresses	(for example,	10.102.40.30-10.1	102.40.45) using a	comma separat
🗸 Enable Device addition on	first time login f	ailure					
Routable IP/ Docker IP*							
172.31.32.161							
Profile Name*							
Docker-profile	\sim	Add	Edit				
Site*							
Ohio-site	\checkmark	Add	Edit]			
Agent*							
	>						
Tags							
Kev		Value			+		

6. Cliquez sur **OK**.

Remarque

Si vous souhaitez redécouvrir une instance, choisissez **Réseaux > Instances > NetScaler CPX**, sélectionnez l'instance que vous souhaitez redécouvrir, puis dans la liste déroulante **Sélectionner une action**, cliquez sur **Redécouvrir**.

Ajout d'instances NetScaler CPX à Citrix ADM à l'aide de variables d'environnement

Vous pouvez également ajouter les instances NetScaler CPX à Citrix ADM à l'aide de variables d'environnement. Pour ajouter des instances, vous devez configurer les variables d'environnement suivantes pour l'instance NetScaler CPX.

- NS_MGMT_SERVER Adresse IP ADM/FQDN
- HOST Adresse IP du nœud
- NS_HTTP_PORT Port HTTP mappé sur le nœud
- NS_HTTPS_PORT- Port HTTPS mappé sur le nœud
- NS_SSH_PORT Port SSH mappé sur le nœud
- NS_SNMP_PORT Port SNMP mappé sur le nœud
- NS_ROUTABLE (L'adresse IP du pod NetScaler CPX n'est pas routable depuis l'extérieur.)
- NS_MGMT_USER Nom d'utilisateur ADM
- NS_MGMT_PASS Mot de passe ADM

Voici un exemple de docker run commande permettant d'ajouter une instance NetScaler CPX à Citrix ADM.

```
1 docker run -dt --privileged=true -p 9080:9080 -p 9443:9443 -p 9022:22
-p 9161:161 -e EULA=yes -e NS_MGMT_SERVER=abc-mgmt-server.com -e
HOST=10.1.1.1 -e NS_HTTP_PORT=9080 -e NS_HTTPS_PORT=9443 -e
NS_SSH_PORT=9022 -e NS_SNMP_PORT=9161 -e NS_ROUTABLE=0 --ulimit
core=-1 - name test cpx:latest
```

Ajouter des instances NetScaler CPX à Citrix ADM à l'aide de Kubernetes ConfigMaps

NetScaler CPX prend en charge l'enregistrement auprès de Citrix ADM en utilisant des fichiers montés en volume via Kubernetes ConfigMaps. Pour activer ce mode d'enregistrement, NetScaler CPX nécessite certaines variables d'environnement qui doivent être spécifiées ainsi que certains fichiers montés en volume via ConfigMaps et Secrets.

Voici les variables d'environnement requises et leur description :

- NS_HTTP_PORT Spécifie le port HTTP mappé sur le nœud.
- NS_HTTPS_PORT Spécifie le port HTTPS mappé sur le nœud.
- NS_SSH_PORT Spécifie le port SSH mappé sur le nœud.
- NS_SNMP_PORT- Spécifie le port SNMP mappé sur le nœud.

Outre les variables d'environnement répertoriées, NetScaler CPX a besoin d'informations sur l'agent ADM auprès duquel il doit s'enregistrer. Ces informations contiennent l'adresse IP ou le nom de domaine complet de l'agent ADM ainsi que les informations d'identification. NetScaler CPX acquiert ces informations à partir des fichiers montés sur le volume. Un ConfigMap contenant l'adresse IP ou le nom de domaine complet est monté sous forme de fichier dans le système de fichiers de l'instance NetScaler CPX. Un secret Kubernetes contenant les informations d'identification de l'agent ADM est également monté sous forme de fichier dans le système de l'instance NetScaler CPX. Avec toutes les informations requises pour l'enregistrement, NetScaler CPX tente de s'enregistrer auprès de l'agent ADM.

Voici un exemple d'extrait de fichier NetScaler CPX YAML avec les fichiers ConfigMap et Secret montés sous forme de fichiers :

```
1
2
         env:
         - name: "EULA"
3
4
          value: "yes"
        - name: "NS_HTTP_PORT"
5
          value: "9080"
6
        - name: "NS_HTTPS_PORT"
7
          value: "9443"
8
        - name: "NS_SSH_PORT"
9
        value: "22"
10
```

```
- name: "NS_SNMP_PORT"
           value: "161"
12
13
         - name: "KUBERNETES_TASK_ID"
           value: ""
14
15
16
         volumeMounts:
17
            - mountPath: /var/adminfo/server/
18
             name: adm-agent-config
19
            - mountPath: /var/adminfo/credentials/
             name: adm-agent-user
22
23
     volumes:
24
25
     - name: adm-agent-config
26
       configMap:
27
         name: adm-agent-config
28
     - name: adm-agent-user
29
       secret:
         secretName: adm-secret
```

Dans l'exemple précédent, un ConfigMap nommé adm-agent-config et un secret adm-agentuser sont consommés. Voici un exemple de création des ConfigMap et Secret requis.

ConfigMap : Le ConfigMap est créé à partir d'un fichier nommé adm_reg_envs. Le fichier nécessite l'adresse IP ou le nom de domaine complet de l'agent ADM au format suivant :

```
1 NS_MGMT_SERVER=adm-agent
```

Dans le format précédent, adm-agent il s'agit du nom de domaine complet de l'agent ADM sur lequel l'instance NetScaler CPX doit être enregistrée.

Utilisez la commande suivante pour créer un ConfigMap :

```
1 kubectl create configmap adm-agent-config --from-file=adm_reg_envs
```

Remarque : Le nom du fichier doit comporter la variable adm_reg_envs et il doit être monté sur le chemin d'accès : /var/adminfo/server/.

Secret : utilisez la commande suivante pour créer un secret Kubernetes. Dans la commande suivante, user123 représente le nom d'utilisateur de l'agent ADM et pass123 le mot de passe.

Une instance NetScaler CPX peut être déployée dans un cluster Kubernetes avec les variables d'environnement et les fichiers montés en volume requis avant même de déployer l'agent ADM dans le cluster. Si vous déployez une instance NetScaler CPX avant de déployer l'agent ADM, NetScaler CPX continue d'essayer de s'enregistrer jusqu'à ce que l'agent ADM soit déployé. Une fois l'agent ADM déployé, l'instance NetScaler CPX utilise les données de configuration fournies par le biais des variables d'environnement et des fichiers montés en volume pour s'enregistrer auprès de l'agent ADM. Cela vous permet d'éviter le redéploiement de NetScaler CPX avec les informations de configuration.

Une instance NetScaler CPX, qui est déjà enregistrée auprès d'un agent ADM, peut modifier dynamiquement l'enregistrement vers un autre agent ADM après une modification de la configuration. Pour cela, vous pouvez mettre à jour les informations de configuration dans le ConfigMap et le Secret du NetScaler CPX déjà déployé. Vous devez mettre à jour le fichier à partir duquel le ConfigMap est créé avec l'adresse IP ou le nom de domaine complet du nouvel agent ADM, supprimer l'ancien ConfigMap, puis créer un nouveau ConfigMap. De même, le secret existant doit être supprimé et un nouveau secret doit être créé avec les informations d'identification du nouvel agent ADM.

Agrégateur de licences NetScaler CPX

April 1, 2025

Actuellement, les CPX de NetScaler obtiennent des licences auprès du serveur Citrix ADM. Dans un environnement Kubernetes, les CPX de NetScaler peuvent augmenter ou diminuer de manière dynamique. Si un NetScaler CPX tombe en panne de manière inattendue, le serveur Citrix ADM met quelques minutes à récupérer la licence. Le serveur Citrix ADM doit être en mesure de récupérer ces licences immédiatement au fur et à mesure que les CPX de NetScaler tombent en panne afin que la même licence puisse être allouée à un autre NetScaler CPX à venir. De plus, si le serveur Citrix ADM n' est pas accessible pour quelque raison que ce soit, vous ne pouvez pas attribuer de licence aux nouveaux CPX NetScaler dans le cluster.

NetScaler CPX License Aggregator est un service Kubernetes fourni par NetScaler. Ce service agit en tant que fournisseur local pour les licences NetScaler CPX au sein d'un cluster Kubernetes. Le service NetScaler CPX License Aggregator déployé dans un cluster Kubernetes peut servir d'intermédiaire entre NetScaler CPX et le serveur de licences ADM et assurer le suivi des CPX NetScaler et des licences attribuées. Grâce au service NetScaler CPX License Aggregator, le serveur Citrix ADM peut récupérer des licences immédiatement lorsque NetScaler CPXS tombe en panne.

Dans un cluster Kubernetes, le service NetScaler CPX License Aggregator prend en charge à la fois NetScaler CPX en tant que sidecar et les déploiements autonomes.

Remarque

L'octroi de licences à l'aide de NetScaler CPX License Aggregator nécessite NetScaler CPX 13.1-30.x ou version ultérieure. L'agrégateur de licences NetScaler CPX ne prend pas en charge les licences pour les anciennes versions de NetScaler CPX.

Principaux avantages de l'agrégateur de licences NetScaler CPX

Les principaux avantages de l'utilisation de NetScaler CPX License Aggregator sont les suivants :

- Capacité à monter en charge: Un serveur de licences NetScaler ADM ne peut prendre en charge que jusqu'à 10 000 déploiements NetScaler CPX. Avec l'introduction du service NetScaler CPX License Aggregator, chaque cluster Kubernetes peut agir comme un client unique pour le serveur de licences Citrix ADM. Vous pouvez donc faire évoluer de nombreux CPX NetScaler avec un seul serveur de licences Citrix ADM.
- Optimisation des ressources : Le service NetScaler CPX License Aggregator prend également en charge la capacité de licence à l'échelle du cluster et peut également extraire des licences du serveur NetScaler ADM en fonction des besoins. L'agrégateur de licences NetScaler CPX peut renvoyer des licences au serveur Citrix ADM. L'agrégateur de licences NetScaler CPX peut gérer la résiliation abusive de NetScaler CPX et récupérer les licences de ces CPX NetScaler après la période d'attente configurée.

Topologie de déploiement

Le schéma suivant montre un déploiement de NetScaler CPX License Aggregator au sein d'un cluster Kubernetes.



Dans ce schéma :

- CPX signifie NetScaler CPX
- CPX License Aggregator signifie NetScaler CPX License Aggregator

Procédez comme suit pour déployer l'agrégateur de licences NetScaler CPX à l'aide de diagrammes Helm en fonction du type de licence NetScaler CPX. Pour plus d'informations sur les différents types de licences NetScaler CPX, consultez la section Licences NetScalerCPX.

Déployez l'agrégateur de licences NetScaler CPX à l'aide de Helm charts

Prérequis

Les conditions préalables suivantes s'appliquent :

- Vous avez besoin de Kubernetes version 1.16 et ultérieure.
- Vous avez besoin de la version 3.x ou ultérieure de Helm.
- Vous devez obtenir l'adresse IP du serveur de licences qui possède la licence pour NetScaler CPX.
- Vous devez fournir un mot de passe pour la base de données Redis dans NetScaler CPX License Aggregator. Vous pouvez fournir le mot de passe de la base de données à l'aide du secret Kubernetes et la commande suivante peut être utilisée pour créer le secret :

```
1 kubectl create secret generic dbsecret --from-literal=password
=<custom-password>
```

Déploiement à l'aide de graphiques Helm

Procédez comme suit pour déployer NetScaler CPX License Aggregator à l'aide des graphiques Helm en fonction du type de licence NetScaler CPX. Pour plus d'informations sur les différents types de licences NetScaler CPX, consultez Licences NetScaler CPX.

Installation du graphique Helm Ajoutez le référentiel graphique NetScaler CPX License Aggregator Helm à l'aide de la commande suivante :

helm repo add Citrix https://citrix.github.io/citrix-helm-charts/

Installation de l'agrégateur de licences NetScaler CPX pour gérer les licences groupées de bande passante Utilisez l'une des commandes suivantes en fonction du type de licence groupée NetScaler CPX dont vous disposez. Dans ces commandes, my-release est utilisé comme nom de version.

Remarque

1

Par défaut, le graphique Helm installe les rôles et les liaisons de rôles RBAC recommandés.

Pour une licence de bande passante platine :

1	helm install my-release citrix/cpx-license-aggregatorset
	licenseServer.address= <license-server-ip-or-fqdn>,redis.</license-server-ip-or-fqdn>
	secretName= <kubernetes-secret-for-DB-password>,</kubernetes-secret-
	licenseAggregator.username= <unique-id-for-CLA>,licenseInfo.</unique-id-
	instanceQuantum= <quantum>,licenseInfo.instanceLowWatermark=<</quantum>
	LOW WATERMARK>,licenseInfo.bandwidthPlatinumQuantum= <quantum-< th=""></quantum-<>
	in-Mbps>,licenseInfo.bandwidthPlatinumLowWatermark= <low< th=""></low<>
	WATERMARK-in-Mbps>

Pour l'édition Enterprise Bandwidth :

1	helm install my-release citrix/cpx-license-aggregatorset
	licenseServer.address= <license-server-ip-or-fqdn>,redis.</license-server-ip-or-fqdn>
	secretName= <kubernetes-secret-for-DB-password>,</kubernetes-secret-
	licenseAggregator.username= <unique-id-for-CLA>,licenseInfo.</unique-id-
	instanceQuantum= <quantum>,licenseInfo.instanceLowWatermark=<</quantum>
	LOW WATERMARK>,licenseInfo.bandwidthEnterpriseQuantum= <quantum< th=""></quantum<>
	-in-Mbps>,licenseInfo.bandwidthEnterpriseLowWatermark= <low< th=""></low<>
	WATERMARK-in-Mbps>

Pour l'édition à bande passante standard :

1	helm install my-release citrix/cpx-license-aggregatorset
	licenseServer.address= <license-server-ip-or-fqdn>,redis.</license-server-ip-or-fqdn>
	secretName= <kubernetes-secret-for-DB-password>,</kubernetes-secret-
	licenseAggregator.username= <unique-id-for-CLA>,licenseInfo.</unique-id-
	instanceQuantum= <quantum>,licenseInfo.instanceLowWatermark=<</quantum>
	LOW WATERMARK>,licenseInfo.bandwidthStandardQuantum= <quantum-< th=""></quantum-<>
	in-Mbps>,licenseInfo.bandwidthStandardLowWatermark= <low< th=""></low<>
	WATERMARK-in-Mbps>

Ces commandes déploient l'agrégateur de licences NetScaler CPX sur le cluster Kubernetes avec la configuration par défaut. Vous pouvez configurer les paramètres au moment de l'installation. Pour plus d'informations, consultez la section de **configuration de NetScaler CPX License Aggregator** dans le référentiel GitHub Helm Chart qui répertorie les paramètres obligatoires et facultatifs que vous pouvez configurer lors de l'installation.

Installation de l'agrégateur de licences NetScaler CPX pour gérer les licences vCPU

Utilisez l'une des commandes suivantes en fonction du type de licence NetScaler CPX vCPU dont vous disposez. Dans ces commandes, my-release est utilisé comme nom de version.

Remarque : Par défaut, le graphique Helm installe les rôles et les liaisons de rôles RBAC recommandés.

Pour l'édition Platinum vCPU :

```
helm install my-release citrix/cpx-license-aggregator --set
licenseServer.address=<License-Server-IP-or-FQDN>,redis.
secretName=<Kubernetes-Secret-for-DB-password>,
```

licenseAggregator.username=<unique-ID-for-CLA>,licenseInfo. vcpuPlatinumQuantum=<QUANTUM>,licenseInfo. vcpuPlatinumLowWatermark=<LOW WATERMARK>

Pour l'édition Enterprise vCPU :

1 helm install my-release citrix/cpx-license-aggregatorset
licenseServer.address= <license-server-ip-or-fqdn>,redis.</license-server-ip-or-fqdn>
secretName= <kubernetes-secret-for-DB-password>,</kubernetes-secret-
licenseAggregator.username= <unique-id-for-CLA>,licenseInfo.</unique-id-
<pre>vcpuEnterpriseQuantum=<quantum>,licenseInfo.</quantum></pre>
vcpuEnterpriseLowWatermark= <low watermark=""></low>

Pour l'édition vCPU standard :

1	helm install my-release citrix/cpx-license-aggregatorset
	licenseServer.address= <license-server-ip-or-fqdn>,redis.</license-server-ip-or-fqdn>
	secretName= <kubernetes-secret-for-DB-password>,</kubernetes-secret-
	licenseAggregator.username= <unique-id-for-CLA>,licenseInfo.</unique-id-
	<pre>vcpuStandardQuantum=<quantum>,licenseInfo.</quantum></pre>
	vcpuStandardLowWatermark= <low watermark=""></low>

Installation de NetScaler CPX License Aggregator pour gérer plusieurs licences

Si vous avez besoin de l'agrégateur de licences NetScaler CPX pour gérer plusieurs types de licences, les arguments pertinents de ces licences doivent être spécifiés dans la commande Helm.

Par exemple :

Pour déployer NetScaler CPX License Aggregator pour et des licences pooled platinum bandwidth edition et vCPU platinum edition, procédez comme suit:

1	helm install demo citrix/cpx-license-aggregatorset
	licenseServer.address= <license-server-ip-or-fqdn>,redis.</license-server-ip-or-fqdn>
	secretName= <kubernetes-secret-for-DB-password>,</kubernetes-secret-
	licenseAggregator.username= <unique-id-for-CLA>,licenseInfo.</unique-id-
	instanceQuantum= <quantum>,licenseInfo.instanceLowWatermark=<</quantum>
	LOW WATERMARK>,licenseInfo.bandwidthPlatinumQuantum= <quantum< th=""></quantum<>
	-in-Mbps>,licenseInfo.bandwidthPlatinumLowWatermark= <low< th=""></low<>
	WATERMARK-in-Mbps>,licenseInfo.vcpuPlatinumQuantum= <quantum< th=""></quantum<>
	>,licenseInfo.vcpuPlatinumLowWatermark=LOW WATERMARK>

Configuration de NetScaler CPX pour obtenir une licence auprès de NetScaler CPX License Aggregator

Lorsque vous utilisez NetScaler CPX License Aggregator pour obtenir une licence NetScaler CPX, la variable CLA d'environnement doit être fournie dans le YAML du déploiement de NetScaler CPX.

Le ipaddress ou domainname à l'aide duquel NetScaler CPX License Aggregator est accessible doit être fourni dans cette variable d'environnement comme suit :

```
1 env:
2
3 - name: "CLA"
4 value: "192.0.2.2"
```

Ou

```
1 env:
2
3 - name: "CLA"
4 value: "local-cla.org"
```

Vous devez également fournir les variables d'environnement suivantes dans le NetScaler CPX YAML.

- POD_NAME: spécifie le nom du pod. Le nom du pod est exposé à NetScaler CPX en tant que variable d'environnement.
- POD_NAMESPACE: spécifie l'espace de noms du pod. L'espace de noms du pod est exposé à NetScaler CPX en tant que variable d'environnement.
- Bandwidth: Spécifie la bande passante en Mbit/s à allouer à NetScaler CPX.
- Edition: spécifie l'édition de licence. Les valeurs prises en charge incluent Standard, Platinum et Enterprise.
- CPX_CORES : Spécifie le nombre de cœurs que vous souhaitez exécuter pour NetScaler CPX.

Pour plus d'informations sur les différentes options de licence NetScaler CPX, consultez la section Licences NetScalerCPX.

La figure suivante montre un exemple de configuration avec ces variables d'environnement :

```
- name: POD NAME
1
2
             valueFrom:
3
                fieldRef:
4
                  apiVersion: v1
5
                  fieldPath: metadata.name
6

    name: POD_NAMESPACE

7
             valueFrom:
8
9
                fieldRef:
10
                  apiVersion: v1
                  fieldPath: metadata.namespace
11
12
            - name: "BANDWIDTH "
13
             value: 1000
14
15
16
             name: "CPX_CORES"
17
             value: 1
18
             name: "EDITION "
19
```

20 value: PLATINUM

Vous devez également ajouter l'étiquette suivante au NetScaler CPX YAML :

```
1 labels:
2 adc: citrix
```

Pour un exemple de déploiement de NetScaler CPX License Aggregator, voir NetScaler CPX License Aggregator : Exemple de déploiement.

Configuration de NetScaler CPX

March 21, 2024

Vous pouvez configurer une instance NetScaler CPX en accédant à l'invite CLI via l'hôte Linux Docker ou en utilisant les API NetScaler NITRO.

Configuration d'une instance NetScaler CPX à l'aide de l'interface de ligne de commande

Accédez à l'hôte Docker et connectez-vous à l'invite SSH de l'instance, comme illustré dans la figure suivante. Les informations d'identification d'administrateur par défaut pour se connecter à une instance NetScaler CPX sont root/linux.

```
root@ubuntu:~# ssh -p 32777 root@127.0.0.1
root@127.0.0.1's password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic x86_64)
* Documentation: https://help.ubuntu.com/
Last login: Tue Dec 15 02:45:42 2015 from 172.17.0.1
root@10:~#
```

Tapez la commande suivante pour utiliser l'invite de ligne de commande de l'instance afin d'exécuter les commandes CLI : **cli_script.sh** "<command>"

Exemple:

root@1	l0:~# cli_script.s	h "show ip"	
exec:	show ip		
	Ipaddress	Traffic Domain	Type
1)	172.17.0.4	0	NetScaler IP VIP
2)	192.0.0.1	0	SNIP
Pour vous déconnecter de l'invite d'instance, tapez se déconnecter.

Prise en charge de l'utilisation d'un mot de passe autre que celui par défaut dans NetScaler CPX

NetScaler CPX prend en charge l'utilisation d'un mot de passe autre que celui par défaut pour le compte root, c'est-à-dire. nsroot Un mot de passe par défaut est généré et attribué à l'utilisateur une fois que NetScaler CPX a été déployé. Ce mot de passe par défaut est également mis à jour pour les utilisateurs SSH : root et nsroot. Vous pouvez modifier ce mot de passe par défaut manuellement. Vous pouvez également réinitialiser le mot de passe SSH par défaut pour root et les comptes nsroot d'utilisateurs manuellement. Citrix recommande de modifier ce mot de passe manuellement pour préserver la sécurité du système.

Une fois votre mot de passe réinitialisé, le nouveau mot de passe est utilisé pour les communications et les cli_script.sh exécutions de l'API NITRO.

Le mot de passe du compte root par défaut est enregistré en texte brut dans le fichier /var/deviceinfo/random_id du système de fichiers NetScaler CPX.

Utilisez la syntaxe suivante pour exécuter cli_script.sh avec les informations d'identification :

cli_script.sh "<command>"":<user>:<password>"

Par exemple, cli_script.sh pour exécuter l'affichage des adresses IP avec un utilisateur ns root et un mot de passe Citrix123, utilisez ce qui suit :

1 cli_script.sh "show ns ip" ":nsroot:Citrix123"

Configuration d'une instance NetScaler CPX à l'aide de l'API NITRO

Vous pouvez utiliser l'API NetScaler NITRO pour configurer les instances NetScaler CPX.

Pour configurer des instances NetScaler CPX à l'aide de l'API Nitro, dans un navigateur Web, tapez :

http://<host_IP_address>:<port>/nitro/v1/config/<resource-type\</pre>

Pour récupérer des statistiques à l'aide de l'API Nitro, dans un navigateur Web, tapez :

http://\<host_IP_address\>:\<port\>/nitro/v1/stat/\<resource-type
\</pre>

Pour plus d'informations sur l'utilisation de l'API NITRO, consultez Services Web REST. Pour NetScaler CPX, utilisez CPX IP address:port where netscaler-ip-address is mentioned.

Configuration d'une instance NetScaler CPX à l'aide de travaux

Vous pouvez configurer des instances NetScaler CPX en créant et en exécutant des tâches dans Citrix ADM. Vous pouvez utiliser les configurations des modèles de configuration, extraire les configurations disponibles sur d'autres appareils et utiliser des configurations enregistrées dans des fichiers texte. Vous pouvez également enregistrer les configurations effectuées à l'aide de l'utilitaire de configuration d'autres instances. Citrix ADM affiche ensuite les commandes CLI correspondantes que vous pouvez utiliser sur votre instance NetScaler CPX. Après avoir sélectionné la configuration, vous devez sélectionner les **instances NetScaler CPX** sur lesquelles vous souhaitez charger la configuration, spécifier les valeurs des variables et exécuter le travail.

Pour configurer des instances NetScaler CPX à l'aide de Jobs :

- 1. Connectez-vous à Citrix ADM à l'aide des informations d'identification administratives.
- 2. Accédez à Réseaux > Tâches de configuration, puis cliquez sur Créer une tâche.
- 3. Spécifiez les valeurs requises et sélectionnez la source de configuration. Vous pouvez également saisir les commandes que vous souhaitez exécuter.

Select Configuration Select In	nstances	Specify Variable Values
b Name*	Instance Type	*
px-single-host NetScaler V		
Configuration Source	=	New
Configuration Source Configuration Template	1 SSH •	New add service db1 172.17.0.10 HTTP 80
Configuration Source Configuration Template Drag and drop the template to the Commands	1 SSH • 2 SSH •	New add service db1 172.17.0.10 HTTP 80 add service db2 172.17.0.11 HTTP 80
Configuration Source Configuration Template Drag and drop the template to the Commands field in the right pane. You can also edit the configuration and save the template with a	1 SSH ▼ 2 SSH ▼ 3 SSH ▼	New add service db1 172.17.0.10 HTTP 80 add service db2 172.17.0.11 HTTP 80 add lb vserver cpx-vip HTTP 172.17.0.4 81
Configuration Source Configuration Template Drag and drop the template to the Commands field in the right pane. You can also edit the configuration and save the template with a different name	1 SSH ▼ 2 SSH ▼ 3 SSH ▼ 4 SSH ▼	New add service db1 172.17.0.10 HTTP 80 add service db2 172.17.0.11 HTTP 80 add lb vserver cpx-vip HTTP 172.17.0.4 81 bind lb vserver cpx-vip db1

4. Sélectionnez les instances NetScaler CPX sur lesquelles vous souhaitez exécuter la configuration et cliquez sur Suivant.

Select t	he target instances on which you want	to run the configuration.	
Add	Instances Delete		
	IP Address		Name
	172.17.0.150		10.102.31.190

5. Spécifiez les paramètres d'exécution et cliquez sur Terminer pour exécuter les commandes sur l'instance NetScaler CPX. Si vous souhaitez enregistrer la configuration et l'exécuter ultérieurement, cliquez sur **Enregistrer et quitter**.

Select Configuration		Select Instances	Specify Variable Values	<pre>Execute</pre>
You can either run the commands	now or schedu	le to run the commands at	a later time.	
On Command Failure*				
Ignore error and continue	•			
Execution Mode*				
Now	•			
Execution Settings Execute in Sequence				
Execute in Parallel				
A STATE OF CALL AND A STATE OF CALL				
0				
Receive Execution Report Throu	gh			

Configuration d'AppFlow sur une instance NetScaler CPX

November 23, 2023

Vous pouvez configurer la fonctionnalité AppFlow sur une instance NetScaler CPX pour collecter les données de performance des pages Web, les informations de flux et de niveau de session utilisateur, ainsi que les informations de base de données requises pour la surveillance et l'analyse des performances des applications. Ces enregistrements de données sont envoyés à Citrix ADM où vous pouvez consulter des rapports historiques et en temps réel pour toutes vos applications.

Pour configurer AppFlow, vous devez d'abord activer la fonctionnalité AppFlow. Ensuite, vous spécifiez les collecteurs auxquels les enregistrements de flux sont envoyés. Ensuite, vous définissez des actions, qui sont des ensembles de collecteurs configurés. Vous configurez ensuite une ou plusieurs stratégies et associez une action à chaque stratégie. La stratégie indique au NetScaler CPX de sélectionner les demandes dont les enregistrements de flux sont envoyés à l'action associée. Enfin, vous liez chaque stratégie soit globalement, soit au serveur virtuel spécifique pour la mettre en œuvre.

Vous pouvez également définir les paramètres AppFlow pour spécifier l'intervalle d'actualisation du modèle et pour activer l'exportation de httpURL, httpCookie et des informations de httpReferer. Sur chaque collecteur, vous devez spécifier l'adresse IP NetScaler CPX comme adresse de l'exportateur.

L'utilitaire de configuration fournit des outils qui aident les utilisateurs à définir les stratégies et les actions. Il détermine exactement comment NetScaler CPX exporte les enregistrements d'un flux particulier vers un ensemble de collecteurs (action). L'interface de ligne de commande fournit un ensemble correspondant de commandes basées sur l'interface de ligne de commande pour les utilisateurs expérimentés qui préfèrent utiliser la ligne de commande.

Avant de pouvoir surveiller les enregistrements, vous devez ajouter l'instance NetScaler CPX à Citrix ADM. Pour plus d'informations sur l'ajout d'une instance NetScaler CPX à Citrix ADM, consultez Installation d'une instance NetScaler CPX à l'aide de NetScalerADM.

Activer AppFlow

Pour utiliser la fonctionnalité AppFlow, vous devez d'abord l'activer.

Pour activer la fonctionnalité AppFlow à l'aide de l'interface de ligne de commande :

Exécutez les commandes suivantes :

1 enable ns feature AppFlow 2 enable ns mode ulfd

Spécifier un collecteur

Un collecteur reçoit les enregistrements AppFlow générés par NetScaler. Pour envoyer les enregistrements AppFlow, vous devez spécifier au moins un collecteur. Par défaut, le collecteur écoute les messages IPFIX sur le port UDP 4739. Vous pouvez modifier le port par défaut lors de la configuration du collecteur.

Pour spécifier un collecteur à l'aide de l'interface de ligne de commande :

Utilisez les commandes suivantes pour ajouter un collecteur :

Pour vérifier la configuration, utilisez la commande suivante :

```
1 show appflow collector <name>
```

Pour spécifier plusieurs collecteurs à l'aide de l'interface de ligne de commande :

Utilisez les commandes suivantes pour ajouter et envoyer les mêmes données à plusieurs collecteurs :

```
1 add appflow collector <collector1> -IPAddress <IP> -Transport Logstream
2
3 add appflow collector <collector2> -IPAddress <IP> -Transport Logstream
4
5 add appflow action <action> -collectors <collector1> <collector2> -
    Transport Logstream
6
7 add appflow policy <policy> true <action> -Transport Logstream
8
9 bind lbvserver <lbvserver> -policy <policy> -priority <priority> -
    Transport Logstream
9 bind lbvserver <lbvserver> -policy <policy> -priority <priority> -
    Transport Logstream
9 bind lbvserver
```

Configuration d'une action AppFlow

Une action AppFlow est un collecteur d'ensembles auquel les enregistrements de flux sont envoyés si la stratégie AppFlow associée correspond.

Utilisez les commandes suivantes pour configurer une action AppFlow :

Pour vérifier la configuration, utilisez la commande suivante :

```
1 show appflow action
```

Configuration d'une stratégie AppFlow

Après avoir configuré une action AppFlow, vous devez ensuite configurer une stratégie AppFlow. Une stratégie AppFlow est basée sur une règle, qui consiste en une ou plusieurs expressions.

Pour configurer une stratégie AppFlow à l'aide de l'interface de ligne de commande :

À l'invite de commandes, tapez la commande suivante pour ajouter une stratégie AppFlow et vérifier la configuration :

```
1 add appflow policy <name> <rule> <action>
2
3 show appflow policy <name>
```

Liaison d'une stratégie AppFlow

Pour appliquer une stratégie, vous devez la lier soit globalement, afin qu'elle s'applique à tout le trafic qui passe par le NetScaler CPX.

Pour lier globalement une stratégie AppFlow à l'aide de l'interface de ligne de commande :

Utilisez la commande suivante pour lier globalement une stratégie AppFlow :

1 bind appflow global <policyName> <priority> [<gotoPriorityExpression [type <type>] [-invoke (<labelType> <labelName>)]

Vérifiez la configuration à l'aide de la commande suivante :

```
1 show appflow global
```

Configuration de NetScaler CPX à l'aide d'un fichier de configuration

November 23, 2023

Au lieu d'utiliser l'interface de ligne de commande (cli_script.sh), l'API NITRO ou les tâches de configuration Citrix ADM pour configurer le NetScaler CPX, vous pouvez configurer le NetScaler CPX à l'aide d'un fichier de configuration statique lors du déploiement de l'instance NetScaler CPX.

Vous pouvez fournir un fichier de configuration statique en tant que fichier d'entrée lors du déploiement du conteneur NetScaler CPX. Lors du démarrage du conteneur NetScaler CPX, le conteneur est configuré en fonction de la configuration spécifiée dans le fichier de configuration statique. Cette configuration inclut une configuration spécifique à NetScaler et des commandes bash shell que vous pouvez exécuter dynamiquement sur le conteneur NetScaler CPX.

Structure du fichier de configuration statique

Comme indiqué précédemment, lorsque NetScaler CPX est déployé, il est configuré en fonction des configurations spécifiées dans le fichier de configuration statique.

Le fichier de configuration statique est un fichier .conf qui inclut deux balises, #NetScaler Commands et #Shell Commands. Sous la #NetScaler Commands balise, vous devez ajouter toutes les commandes NetScaler pour configurer la configuration spécifique à NetScaler sur NetScaler CPX. Sous la #Shell Commands balise, vous devez ajouter les commandes shell que vous souhaitez exécuter sur NetScaler CPX.

Lors du déploiement du conteneur NetScaler CPX, les commandes NetScaler et les commandes shell sont exécutées sur le conteneur dans l'ordre spécifié dans le fichier de configuration.

Important

•

- Les balises peuvent être répétées plusieurs fois dans le fichier de configuration.
- Les balises ne sont pas sensibles à la casse.
- Le fichier de configuration doit être présent dans le répertoire /etc en tant que fichier cpx
 . conf dans le système de fichiers du conteneur.
- Le fichier de configuration peut également inclure des commentaires. Vous devez ajouter un caractère « **#** » avant vos commentaires.
- S'il existe des scénarios d'échec lors du déploiement du conteneur NetScaler CPX avec le fichier de configuration, les échecs sont enregistrés dans le ns.log fichier du conteneur.
- Lorsque vous redémarrez le conteneur NetScaler CPX, le fichier de configuration est réappliqué sur le conteneur.

1 #NetScaler Commands

```
2
3
  add lb vserver v1 http 1.1.1.1 80
4
5
  add service s1 2.2.2.2 http 80
6
   bind lb vserver v1 s1
7
8
   #Shell Commands
9
10
  touch /etc/a.txt
11
12
13 echo "this is a" > /etc/a.txt
14
15 #NetScaler Commands
16
  add lb vserver v2 http
17
18
19 #Shell Commands
```

```
20
21 echo "this is a 1" >> /etc/a.txt
22
23 #NetScaler Commands
24
25 add lb vserver v3 http
26
27 #This is a test configuration file
```

Pour installer un conteneur NetScaler CPX et configurer dynamiquement le conteneur NetScaler CPX en fonction d'un fichier de configuration, montez le fichier de configuration statique à l'aide de l'option – v de la commande docker run:

```
1 docker run -dt --privileged=true -e EULA=yes --ulimit core=-1 -v /tmp/
cpx.conf:/etc/cpx.conf --name mycpx store/citrix/citrixadccpx:13.0-x
.x
```

Support du routage dynamique dans NetScaler CPX

April 1, 2025

NetScaler CPX prend en charge le protocole de routage dynamique BGP. L'objectif principal du protocole de routage dynamique est d'annoncer l'adresse IP du serveur virtuel en fonction de l'état des services liés au serveur virtuel. Il aide un routeur en amont à choisir le meilleur itinéraire parmi plusieurs itinéraires vers un serveur virtuel réparti topographiquement.

Pour plus d'informations sur le mot de passe autre que celui par défaut dans NetScaler CPX, consultez la Support **for** using a non-**default** password in NetScaler CPX section du document Configuration de NetScaler CPX.

Dans un réseau hôte unique, le client, les serveurs et l'instance NetScaler CPX sont déployés en tant que conteneurs sur le même hôte Docker. Tous les conteneurs sont reliés par le pont docker0. Dans cet environnement, l'instance NetScaler CPX agit en tant que proxy pour les applications provisionnées en tant que conteneurs sur le même hôte Docker. Pour plus d'informations sur le déploiement du mode réseau hôte NetScaler CPX, consultez la section Mode réseau hôte.

La figure suivante illustre la topologie à hôte unique.



Dans cette topologie, les serveurs virtuels sont configurés et annoncés (en fonction de l'état des services) sur le réseau ou le routeur en amont à l'aide du protocole BGP.

Effectuez les étapes suivantes pour configurer BGP sur NetScaler CPX sur un seul hôte Docker avec le mode réseau de pont.

Configurer l'injection d'intégrité de route basée sur BGP à l'aide de l'API REST sur NetScaler CPX

1. Créez un conteneur à partir de l'image NetScaler CPX à l'aide de la commande suivante :

```
1 docker run -dt --privileged=true -p 22 -p 80 -p 161 -e EULA=yes
--ulimit core=-1 cpx: <tag>
```

Par exemple :

1 docker run -dt --privileged=true -p 22 -p 80 -p 161 -e EULA=yes --ulimit core=-1 cpx:12.1-50.16

2. Connectez-vous au conteneur à l'aide de la commande suivante :

docker exec -it <container id> bash

3. Activez la fonctionnalité BGP à l'aide de la commande suivante :

1 cli_script.sh "enable ns feature bgp"

4. Obtenez le NSIP à l'aide de la commande show ns ip:

1 cli_script.sh "show ns ip"

5. Ajoutez le serveur virtuel à l'aide de la commande suivante :

1 cli_script.sh "add lb vserver <vserver_name> http <VIP> <PORT>"

- 6. Ajoutez des services et liez des services au serveur virtuel.
- 7. Activez hostroute pour l'adresse IP virtuelle à l'aide de la commande suivante :

cli_script.sh "set ns ip <VIP> -hostroute enabled "

Déconnectez-vous du conteneur et envoyez des commandes BGP NITRO de l'hôte au NSIP sur le port 9080.

8. Configurez le routeur BGP :

Par exemple, si vous souhaitez configurer :

```
    router bgp 100
    Neighbour 172.17.0.2 remote-as 101
    Redistribute kernel
```

Spécifiez la commande comme suit :

```
curl -u username:password http://<NSIP>:9080/nitro/v1/config/ -
1
         X POST --data 'object={
2
    "routerDynamicRouting": {
   "bgpRouter" : {
3
4
   "localAS":100, "neighbor": [{
5
    "address": "172.17.0.2", "remoteAS": 101 }
   ], "afParams":{
6
    "addressFamily": "ipv4", "redistribute": {
7
8
    "protocol": "kernel" }
9
     }
10
     }
11
     }
12
     }
13
    ŧ.
```

9. Installez les routes BGP apprises dans le PE à l'aide de la commande NITRO suivante :

10. Vérifiez l'état d'adjacence BGP à l'aide de la commande NITRO suivante :

```
1 curl -u username:password http://<NSIP>:9080/nitro/v1/config/
routerDynamicRouting/bgpRouter
```

Exemple de sortie :

```
1
       root@ubuntu:~# curl -u username:password http://
           172.17.0.3:9080/nitro/v1/config/routerDynamicRouting/
          bgpRouter
2
       {
      "errorcode": 0, "message": "Done", "severity": "NONE", "
3
         routerDynamicRouting":{
     "bgpRouter":[{
4
      "localAS": 100, "routerId": "172.17.0.3", "afParams": [ {
5
      "addressFamily": "ipv4" }
6
    , {
    "addressFamily": "ipv6" }
    inthese". [ {
7
8
      ], "neighbor": [ {
9
      "address": "172.17.0.2", "remoteAS": 101, "ASOriginationInterval
10
         ": 15, "advertisementInterval": 30, "holdTimer": 90, "
         keepaliveTimer": 30, "state": "Connect", "singlehopBfd":
false, "multihopBfd": false, "afParams": [ {
      "addressFamily": "ipv4", "activate": true }
11
     , {
    "addressFamily": "ipv6", "activate": false }
12
13
14
      1
```

11. Vérifiez que les routes apprises par le biais de BGP sont installées dans le moteur de paquets à l'aide de la commande suivante :

1 cli_script.sh "show route"

12. Enregistrez la configuration à l'aide de la commande suivante :

1 cli_script.sh "save config"

La configuration du routage dynamique est enregistrée dans le fichier /nsconfig/ZebOS.conf

Configuration des pilotes de journalisation Docker

November 23, 2023

Docker inclut des mécanismes de journalisation appelés « pilotes de journalisation » pour vous aider à obtenir des informations à partir des conteneurs en cours d'exécution. Vous pouvez configurer un conteneur NetScaler CPX pour transmettre les journaux qu'il génère aux pilotes de journalisation Docker. Pour plus d'informations sur les pilotes de journalisation Docker, consultez Configurer les pilotes de journalisation.

Par défaut, tous les journaux générés par le conteneur NetScaler CPX sont stockés dans un /cpx/log /ns.log fichier sur l'hôte Docker. Lorsque vous démarrez le conteneur NetScaler CPX à l'aide de la commande docker run, vous pouvez le configurer pour transférer tous les journaux générés vers un pilote de journalisation docker à l'aide de cette option. --log-driver Si le pilote de journalisation possède des paramètres configurables, vous pouvez les définir à l'aide de l'option --log-opt < NAME>=<VALUE>.

Dans l'exemple suivant, le conteneur NetScaler CPX est configuré pour transmettre tous les journaux générés en utilisant syslog comme pilote de journalisation.

```
1 docker run -dt --privileged=true --log-driver syslog --log-opt syslog-
address=udp://10.106.102.190:514 -e EULA=yes --ulimit core=-1 --name
test store/citrix/cpx:12.1-48.13
```

De même, dans l'exemple suivant, le conteneur NetScaler CPX est configuré pour transmettre tous les journaux générés à l'aide de Splunk comme pilote de journalisation.

```
1 docker run -dt --privileged=true --log-driver=splunk --log-opt splunk-
token=176FCEBF-4CF5-4EDF-91BC-703796522D20 --log-opt splunk-url=
https://splunkhost:8088 -e EULA=yes --ulimit core=-1 --name test
store/citrix/cpx:12.1-48.13
```

Mise à niveau d'une instance NetScaler CPX

November 23, 2023

Vous pouvez mettre à niveau une instance NetScaler CPX en l'arrêtant, en installant la dernière version sur le même point de montage, puis en supprimant l'ancienne instance. Un point de montage est un répertoire dans lequel vous montez le répertoire **/cpx** sur l'hôte.

Par exemple, pour monter le répertoire **/cpx** de l'instance NetScaler CPX existante dans le répertoire **/var/cpx** de l'hôte, le point de montage est **/var/cpx** et le répertoire de montage NetScaler CPX est **/cpx** comme indiqué ci-dessous :

```
1 root@ubuntu:~# docker run -dt -e EULA=yes --name mycpx -v /var/cpx
:/cpx --ulimit core=-1 cpx:13.0-x.x
```

Conditions préalables

Assurez-vous que vous avez :

 Détails du répertoire d'hôte dans lequel vous avez monté le répertoire /cpx de l'instance NetScaler CPX existante. Vous pouvez utiliser la commande docker inspect <containerName>, où <containerName> est le nom du conteneur NetScaler CPX, pour afficher des informations sur le répertoire hôte. La sortie de la commande fournit les détails des configurations de conteneur, y compris les volumes. Dans l'entrée "**Mounts**", la sous-entrée "**Source**"indique l'emplacement du répertoire hôte sur l'hôte.

"Mo	unts {	": [
	Č.,	"Source": "/var/cpx",
		"Destination": "/cpx",
		"Mode": "",
		"RW": true
	}	
],		

 Téléchargez le dernier fichier image NetScaler CPX Docker et chargez l'image NetScaler CPX Docker. Pour charger l'image, accédez au répertoire dans lequel vous avez enregistré le fichier image Docker. Utilisez la commande docker load -i <image_name> pour charger l'image. Une fois l'image NetScaler CPX chargée, vous pouvez entrer la commande docker images pour afficher les informations relatives à l'image :

1	root@ubuntu:	~# docker	load -i cpx-1	3.0-x.x.gz	
2	rootaubuntu	~# docker	images		
4	roo ceuburicu.	in docker	Inages		
5	REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
6 7	срх	13.0-x.x	2e97aadf918b	43 hours ago	414.5 MB

Pour mettre à niveau une instance NetScaler CPX

1. Arrêtez l'instance NetScaler CPX existante en saisissant la commande docker stop < containerName>, où <containerName> est le nom de l'instance NetScaler CPX.

```
1 root@ubuntu:~# docker stop mycpx
2 mycpx
```

2. À l'aide de la commande docker run, déployez la dernière instance NetScaler CPX à partir de l'image NetScaler CPX que vous avez chargée sur l'hôte. Assurez-vous de déployer l'instance au même point de montage (par exemple /var/cpx:/cpx) que celui que vous avez utilisé pour l'instance NetScaler CPX existante.

```
1 root@ubuntu:~# docker run -dt -P -e CPX_CORES=1 --name latestcpx
--ulimit core=-1 -e EULA=yes -v /var/cpx:/cpx --cap-add=
NET_ADMIN cpx:13.0-x.x
```

Vous pouvez entrer la commande docker ps afin de vérifier que l'instance NetScaler CPX déployée est la dernière version.

```
2
      root@ubuntu:~# docker ps
3
      CONTAINER ID
                          IMAGE
                                             COMMAND
4
          CREATED
                              STATUS
                                                             PORTS
                           NAMES
5
                    cpx:13.0-x.x "/bin/sh -c 'bash -C "
6
      ead12ec4e965
                                                             5
          seconds ago
                            Up 5 seconds
                                                22/tcp, 80/tcp, 443/
          tcp, 161/udp latestcpx
  ...
7
```

3. Après avoir vérifié que vous avez déployé la bonne instance NetScaler CPX, entrez la commande **docker rm** <containerName> pour supprimer l'ancienne instance.

```
1 root@ubuntu:~# docker rm mycpx
2 mycpx
```

Utilisation de serveurs virtuels génériques dans une instance NetScaler CPX

November 23, 2023

Lorsque vous provisionnez une instance NetScaler, une seule adresse IP privée (adresse IP unique) est attribuée à une instance NetScaler CPX par le moteur Docker. Les trois fonctions IP d'une instance NetScaler sont multiplexées sur une adresse IP. Cette adresse IP unique utilise différents numéros de port pour fonctionner en tant que NSIP, SNIP et VIP.

L'adresse IP unique attribuée par le moteur Docker est dynamique. Ajoutez les serveurs virtuels d' équilibrage de charge (LB) ou de commutation de contenu (CS) à l'aide de l'adresse IP unique ou de l'adresse IP 127.0.0.1. Les serveurs virtuels créés à l'aide de 127.0.0.1 sont appelés serveurs virtuels Wildcard. Par défaut, lorsque vous créez un serveur virtuel générique, le NetScaler CPX remplace l' adresse IP attribuée au serveur virtuel générique. L'adresse IP attribuée est 127.0.0.1, qui est remplacée par le NSIP attribué à l'instance NetScaler CPX par le moteur Docker.

Dans les déploiements NetScaler CPX à haute disponibilité, vous pouvez ajouter des serveurs virtuels génériques sur l'instance principale de NetScaler CPX. La synchronisation des configurations entre les nœuds configure le serveur virtuel Wildcard sur l'instance NetScaler CPX secondaire. Cela élimine le besoin de configurer le serveur virtuel sur le NSIP attribué par le moteur Docker aux instances NetScaler CPX.

Points à noter :

- Assurez-vous que le numéro de port que vous attribuez au serveur virtuel générique n'est utilisé par aucun autre serveur virtuel du déploiement.
- L'ajout de serveur virtuel générique échoue si le numéro de port que vous attribuez au serveur virtuel générique est déjà utilisé par les services internes.
- Le serveur virtuel générique ne prend pas en charge le caractère *.

Pour créer un serveur virtuel d'équilibrage de charge générique, entrez la commande suivante à l' invite de commandes :

```
1 add lb vserver <name> <serviceType> 127.0.0.1 <port>
2
3 add lb vserver testlbvserver HTTP 127.0.0.1 30000
```

Pour créer un serveur virtuel de commutation de contenu générique, à l'invite de commandes, entrez la commande suivante :

```
1 add cs vserver <name> <serviceType> 127.0.0.1 <port>
2
3 add cs vserver testcsvserver HTTP 127.0.0.1 30000
```

Déployer NetScaler CPX en tant que proxy pour permettre un flux de trafic Est-Ouest

November 23, 2023

Dans ce déploiement, l'instance NetScaler CPX agit en tant que proxy pour permettre la communication entre les conteneurs d'applications résidant sur plusieurs hôtes. L'instance CPX NetScaler est mise en service avec les applications sur plusieurs hôtes et fournit le chemin de communication le plus court.

L'image suivante illustre le flux de trafic entre deux applications via les instances NetScaler CPX.



Cette image montre le flux de trafic entre l'application C et l'application B et entre l'application A et l'application B. Lorsque l'application C (sur l'un des hôtes) envoie une demande à B, la demande est d'abord reçue sur le conteneur NetScaler CPX sur le même hôte que l'application C. Ensuite, le conteneur NetScaler CPX transmet le trafic à le conteneur NetScaler CPX hébergé sur le même hôte que l'application B, puis le trafic est transféré à l'application B. Un chemin de trafic similaire est suivi lorsque l'application A envoie une demande à l'application B.

Dans cet exemple, un NetScaler MPX est également déployé pour autoriser le trafic vers les applications depuis Internet via une adresse IP virtuelle globale. Le trafic provenant du NetScaler MPX est reçu sur les conteneurs NetScaler CPX, qui répartit ensuite le trafic entre les conteneurs d'applications.

Le schéma suivant illustre cette topologie avec les configurations qui doivent être définies pour que la communication ait lieu.



Entities configured on the CPX container

VIPs	Services
VIP-A1	Svc-A1
VIP-B1	Svc-B1
VIP-C1	Svc-VIP-C2 Svc-VIP-C3

Entities configured on	the CPX container
------------------------	-------------------

VIPs	Services
VIP-A2	Svc-A2
VIP-B2	Svc-VPI-B1
VIP-C2	Svc-C2

ntities configured on the	CPX	container
---------------------------	-----	-----------

F

VIPs	Services
VIP-A3	Svc-VIP-A1 Svc-VIP-A2
VIP-B3	Svc-VIP-B1
VIP-C3	Svc-C3

Le tableau suivant répertorie les adresses IP et les ports configurés sur les instances NetScaler CPX dans cet exemple de configuration.

Docker Host 1		Docker Host 2		Docker Host 3	
VIPs	Services	VIPs	Services	VIPs	Services
	Bound to the		Bound to the		Bound to the
	VIP		VIP		VIP
VIP-A1	SVC-A1	VIP-A2	SVC-A2	VIP-A3	SVC-VIP-A1
172.17.0.2:3	10.102.29.100:	172.17.0.3:3	10.102.29.105:	172.17.0.4:3	10.102.29.100:
0000	80	0000	80	0000	30000
					SVC-VIP-A2
					10.102.29.105:
					30000
VIP-B1	SVC-B1	VIP-B2	SVC-VIP-B1	VIP-B3	SVC-VIP-B1
172.17.0.2:3	10.102.29.100:	172.17.0.3:3	10.102.29.100:	172.17.0.4:3	10.102.29.100:
0001	90	0001	30001	0001	30001
VIP-C1	SVC-VIP-C2	VIP-C2	SVC-C2	VIP-C3	SVC-C3
172.17.02:30	10.102.29.105:	172.17.0.3:3	10.102.29.105:	172.17.0.4:3	10.102.29.110:
002	30002	0002	70	0002	70
	SVC-VIP-C3				
	10.102.29.110:				
	30002				

Pour configurer cet exemple de scénario, exécutez la commande suivante à l'invite du shell Linux lors de la création du conteneur NetScaler CPX sur les trois hôtes Docker :

```
1 docker run -dt -p 22 -p 80 -p 161/udp -p 30000-30002: 30000-30002 --
ulimit core=-1 --privileged=truecpx:6.2
```

Exécutez les commandes suivantes à l'aide de la fonctionnalité Jobs de Citrix ADM ou à l'aide des API NITRO.

Sur l'instance NetScaler CPX sur l'hôte Docker 1 :

```
add lb vserver VIP-A1 HTTP 172.17.0.2 30000
1
2
       add service svc-A1 10.102.29.100 HTTP 80
       bind lb vserver VIP-A1 svc-A1
3
       add lb vserver VIP-B1 HTTP 172.17.0.2 30001
4
       add service svc-B1 10.102.29.100 HTTP 90
5
6
       bind lb vserver VIP-B1 svc-B1
       add lb vserver VIP-C1 HTTP 172.17.0.2 30002
7
       add service svc-VIP-C2 10.102.29.105 HTTP 30002
8
9
       add service svc-VIP-C3 10.102.29.110 HTTP 30002
       bind lb vserver VIP-C1 svc-VIP-C2
10
       bind lb vserver VIP-C1 svc-VIP-C3
11
```

Sur l'instance NetScaler CPX sur l'hôte Docker 2 :

1	add lb vserver VIP-A2 HTTP 172.17.0.3 30000
2	add service svc-A2 10.102.29.105 HTTP 80
3	bind lb vserver VIP-A2 svc-A2
4	add lb vserver VIP-B2 HTTP 172.17.0.3 30001
5	add service svc-VIP-B1 10.102.29.100 HTTP 30001
6	bind lb vserver VIP-B2 svc-VIP-B1
7	add lb vserver VIP-C2 HTTP 172.17.0.3 30002
8	add service svc-C2 10.102.29.105 HTTP 70
9	bind lb vserver VIP-C2 svc-C2

Sur l'instance NetScaler CPX sur l'hôte Docker 3 :

add lb vserver VIP-A3 HTTP 172.17.0.4 30000
add service svc-VIP-A1 10.102.29.100 HTTP 30000
add service svc-VIP-A2 10.102.29.105 HTTP 30000
bind lb vserver VIP-A3 svc-VIP-A1
bind lb vserver VIP-A3 svc-VIP-A2
add lb vserver VIP-B3 HTTP 172.17.0.4 30001
add service svc-VIP-B1 10.102.29.100 HTTP 30001
bind lb vserver VIP-B3 svc-VIP-B1
add lb vserver VIP-C3 HTTP 172.17.0.4 30002
add service svc-C3 10.102.29.110 HTTP 70
bind lb vserver VIP-C3 svc-C3

Déployer NetScaler CPX sur un réseau hôte unique

November 23, 2023

Dans un réseau hôte unique, l'instance NetScaler CPX agit en tant que proxy entre les conteneurs d' applications sur le même hôte. À ce titre, l'instance NetScaler CPX fournit évolutivité et sécurité aux applications basées sur des conteneurs. En outre, il optimise les performances et fournit également un aperçu des données de télémétrie.

Dans un réseau hôte unique, le client, les serveurs et l'instance NetScaler CPX sont déployés en tant que conteneurs sur le même hôte Docker. Tous les conteneurs sont reliés par un pont docker0.

Dans cet environnement, l'instance NetScaler CPX agit en tant que proxy pour les applications provisionnées en tant que conteneurs sur le même hôte Docker.

(172.17.0.10) (172.17.0.2) (172.17.0.4) (172.17.0.11) DB2 container (172.17.0.11) DD2 container

La figure suivante illustre la topologie à hôte unique.

Dans cet exemple, un conteneur d'application Web (172.17.0.2) est le client et les deux conteneurs de base de données, DB1 (172.17.0.10) et DB2 (172.17.0.11), sont les serveurs. Le conteneur NetScaler CPX (172.17.0.4) se trouve entre le client et les serveurs agissant en tant que proxy.

Pour permettre à l'application Web de communiquer avec les conteneurs de base de données via NetScaler CPX, vous devez d'abord configurer deux services sur le conteneur NetScaler CPX pour représenter les deux serveurs. Configurez ensuite un serveur virtuel à l'aide de l'adresse IP NetScaler CPX et d'un port HTTP non standard (tel que 81), car NetScaler CPX réserve le port HTTP standard 80 pour la communication NITRO. Dans cette topologie, vous n'avez pas à configurer de règles NAT car le client et le serveur se trouvent sur le même réseau.

Pour configurer ce scénario, exécutez les commandes suivantes à l'aide de la fonctionnalité Jobs de Citrix ADM ou à l'aide des API NITRO :

```
    add service db1 HTTP 172.17.0.10 80
    add service db2 HTTP 172.17.0.11 80
    add lb vserver cpx-vip HTTP 172.17.0.4 81
    bind lb vserver cpx-vip db1
    bind lb vserver cpx-vip db2
```

Déployer NetScaler CPX dans un réseau multi-hôtes

November 23, 2023

Une instance NetScaler CPX dans un réseau multi-hôtes peut être configurée dans un déploiement de production dans le centre de données où elle fournit des fonctions d'équilibrage de charge. Il peut également fournir des fonctions de surveillance et des données analytiques.

Dans un réseau multi-hôtes, les instances NetScaler CPX, les serveurs principaux et les clients sont déployés sur différents hôtes. Vous pouvez utiliser des topologies multi-hôtes dans les déploiements de production où l'instance NetScaler CPX équilibre la charge d'un ensemble d'applications et de serveurs basés sur des conteneurs, voire des serveurs physiques.

Topologie 1 : serveurs NetScaler CPX et backend sur le même hôte ; client sur un réseau différent

Dans cette topologie, l'instance NetScaler CPX et les serveurs de base de données sont provisionnés sur le même hôte Docker, mais le trafic client provient d'un autre emplacement sur le réseau. Cette topologie peut être utilisée dans un déploiement de production dans lequel l'instance NetScaler CPX équilibre la charge d'un ensemble d'applications ou de serveurs basés sur des conteneurs.

Le schéma suivant illustre cette topologie.



Linux Docker Host: 10.102.29.100

Dans cet exemple, l'instance NetScaler CPX (172.17.0.4) et les deux serveurs, DB1 (172.17.0.10) et DB2 (172.17.0.11) sont provisionnés sur le même hôte Docker avec l'adresse IP 10.102.29.100. Le client réside ailleurs sur le réseau.

Les demandes du client provenant d'Internet sont reçues sur l'adresse IP virtuelle configurée sur l' instance NetScaler CPX, qui distribue ensuite les demandes sur les deux serveurs.

Vous pouvez utiliser deux méthodes pour configurer cette topologie :

Méthode 1 : Utilisation d'une adresse IP supplémentaire et d'un port standard pour le VIP

- 1. Configurez le VIP sur le conteneur NetScaler CPX à l'aide d'une adresse IP supplémentaire.
- 2. Configurez une adresse IP supplémentaire pour l'hôte Docker.
- 3. Configurez les règles NAT pour transférer tout le trafic reçu sur l'adresse IP supplémentaire de l'hôte Docker vers l'adresse IP supplémentaire du VIP.
- 4. Configurez les deux serveurs en tant que services sur l'instance NetScaler CPX.
- 5. Enfin, liez les services au VIP.

Notez que dans cet exemple de configuration, le réseau 10.x.x.x désigne un réseau public.

Pour configurer cet exemple de scénario, exécutez les commandes suivantes à l'aide de la fonctionnalité Jobs de Citrix ADM ou à l'aide des API NITRO :

1 add service s1 172.17.0.10 HTTP 80

```
2 add service s2 172.17.0.11 HTTP 80
3 add lb vserver cpx-vip HTTP 172.17.4.100 80
4 bind lb vserver cpx-vip s1
5 bind lb vserver cpx-vip s2
```

Configurez une adresse IP publique supplémentaire pour l'hôte Docker et une règle NAT en exécutant les commandes suivantes à l'invite du shell Linux :

```
ip addr add 10.102.29.103/24 dev eth0
iptables -t nat -A PREROUTING -p ip -d 10.102.29.103 -j DNAT --to-
destination 172.17.4.100
```

Méthode 2 : Utilisation de l'adresse IP NetScaler CPX pour le VIP et configurer le mappage des ports :

- 1. Configurez l'adresse IP virtuelle et les deux services sur l'instance NetScaler CPX. Utilisez un port non standard, 81, avec le VIP.
- 2. Liez les services au VIP.
- 3. Configurez une règle NAT pour transférer tout le trafic reçu sur le port 50000 de l'hôte Docker vers le VIP et le port 81.

Pour configurer cet exemple de scénario, exécutez la commande suivante à l'invite du shell Linux lors de la création du conteneur NetScaler CPX sur les trois hôtes Docker :

```
1 docker run -dt -p 22 -p 80 -p 161/udp -p 50000:81 --ulimit core=-1
--privileged=true cpx:6.2
```

Une fois l'instance NetScaler CPX provisionnée, exécutez les commandes suivantes à l'aide de la fonctionnalité Jobs de Citrix ADM ou à l'aide des API NITRO :

```
    add service s1 172.17.0.10 http 80
    add service s2 172.17.0.11 http 80
    add lb vserver cpx-vip HTTP 172.17.0.4 81
    bind lb vserver cpx-vip s1
    bind lb vserver cpx-vip s2
```

Remarque:

Si vous n'avez pas configuré le mappage des ports lors du provisionnement de l'instance NetScaler CPX, configurez une règle NAT en exécutant les commandes suivantes à l'invite du shell Linux :

iptables -t nat -A PREROUTING -p tcp -m addrtype –dst-type LOCAL -m tcp –dport 50000 -j DNAT –to-destination 172.17.0.4:81

Topologie 2 : NetScaler CPX avec serveurs physiques et client

Dans cette topologie, seule l'instance NetScaler CPX est provisionnée sur un hôte Docker. Le client et les serveurs ne sont pas basés sur des conteneurs et résident ailleurs sur le réseau.

Dans cet environnement, vous pouvez configurer l'instance NetScaler CPX pour équilibrer la charge du trafic entre les serveurs physiques.

La figure suivante illustre cette topologie.

Dans cet exemple, le conteneur NetScaler CPX (172.17.0.4) se trouve entre le client et les serveurs physiques agissant en tant que proxy. Les serveurs, DB1 (10.102.29.105) et DB2 (10.102.29.110), résident en dehors d'un hôte Docker sur le réseau. La demande du client provient d'Internet et est reçue sur le NetScaler CPX, qui la distribue entre les deux serveurs.

Pour activer cette communication entre le client et les serveurs via NetScaler CPX, vous devez d'abord configurer le mappage des ports lors de la création du conteneur NetScaler CPX. Configurez ensuite les deux services sur le conteneur NetScaler CPX pour représenter les deux serveurs. Enfin, configurez un serveur virtuel à l'aide de l'adresse IP NetScaler CPX et du port HTTP non standard mappé 8080.

Notez que dans l'exemple de configuration, le réseau 10.x.x.x désigne un réseau public.

Pour configurer cet exemple de scénario, exécutez la commande suivante à l'invite du shell Linux lors de la création du conteneur NetScaler CPX :

Exécutez ensuite les commandes suivantes à l'aide de la fonctionnalité Jobs de Citrix ADM ou à l'aide des API NITRO :

```
    add service s1 HTTP 10.102.29.105 80
    add service s2 HTTP 10.102.29.110 80
    add lb vserver cpx-vip HTTP 172.17.0.4 8080
    bind lb vserver cpx-vip s1
    bind lb vserver cpx-vip s2
```

Topologie 3 : NetScaler CPX et serveurs provisionnés sur différents hôtes

Dans cette topologie, l'instance NetScaler CPX et les serveurs de base de données sont provisionnés sur différents hôtes Docker, et le trafic client provient d'Internet. Cette topologie peut être utilisée dans un déploiement de production dans lequel l'instance NetScaler CPX équilibre la charge d'un ensemble d'applications ou de serveurs basés sur des conteneurs.

Le schéma suivant illustre cette topologie.

Dans cet exemple, l'instance NetScaler CPX et un serveur (DB1) sont provisionnés sur le même hôte Docker dont l'adresse IP est 10.102.29.100. Quatre autres serveurs (DB2, DB3, DB4 et DB5) sont provisionnés sur deux hôtes Docker différents, 10.102.29.105 et 10.102.29.110.

Les demandes du client provenant d'Internet sont reçues sur l'instance NetScaler CPX, qui les distribue ensuite sur les cinq serveurs. Pour activer cette communication, vous devez configurer les éléments suivants :

- Définissez le mappage des ports lors de la création de votre conteneur NetScaler CPX. Dans cet exemple, cela signifie que vous devez transférer le port 8080 du conteneur vers le port 8080 sur l'hôte. Lorsque la demande du client arrive sur le port 8080 de l'hôte, elle correspond au port 8080 du conteneur CPX.
- 2. Configurez les cinq serveurs en tant que services sur l'instance NetScaler CPX. Vous devez utiliser une combinaison de l'adresse IP de l'hôte Docker et du port mappé respectifs pour définir ces services.
- 3. Configurez un VIP sur l'instance NetScaler CPX pour recevoir la demande du client. Cette adresse IP virtuelle doit être représentée par l'adresse IP NetScaler CPX et le port 8080 mappés au port 8080 de l'hôte.
- 4. Enfin, liez les services au VIP.

Notez que dans l'exemple de configuration, le réseau 10.x.x.x désigne un réseau public.

Pour configurer cet exemple de scénario, exécutez la commande suivante à l'invite du shell Linux lors de la création du conteneur NetScaler CPX :

Exécutez les commandes suivantes à l'aide de la fonctionnalité Jobs de Citrix ADM ou à l'aide des API NITRO :

```
1
       add service s1 10.102.29.100 HTTP 8081
       add service s2 10.102.29.105 HTTP 8081
2
3
       add service s3 10.102.29.105 HTTP 8082
       add service s4 10.102.29.110 HTTP 8081
4
5
       add service s5 10.102.29.110 HTTP 8082
       add lb vserver cpx-vip HTTP 172.17.0.2 8080
6
7
       bind lb vserver cpx-vip s1
8
       bind lb vserver cpx-vip s2
9
       bind lb vserver cpx-vip s3
       bind lb vserver cpx-vip s4
10
       bind lb vserver cpx-vip s5
11
```

Déployez NetScaler CPX avec un accès direct au réseau

November 23, 2023

En mode réseau de pont, vous pouvez configurer l'instance NetScaler CPX pour avoir un accès direct au réseau. Dans ce scénario, le trafic entrant est directement reçu sur l'IP du serveur virtuel NetScaler CPX (VIP).

Pour activer cette communication, vous devez d'abord configurer une adresse IP publique sur le pont docker0. Supprimez ensuite l'adresse IP publique du port réseau eth0 et liez le port réseau au pont docker0.

Configurez l'équilibrage de charge en ajoutant les deux services, puis configurez une adresse IP publique réseau en tant qu'adresse IP virtuelle sur l'instance NetScaler CPX. Les demandes des clients sont reçues directement sur le VIP.

Dans l'exemple de configuration, le réseau 10.x.x.x désigne un réseau public.

Pour configurer ce scénario, exécutez la commande suivante à l'invite du shell Linux :

```
ip addr add 10.102.29.100/24 dev docker0;
p addr del 10.102.29.100/24 dev eth0;
brctl addif docker0 eth0;
ip route del default;
j route add default via 10.102.29.1 dev docker0
```

À l'aide de la fonctionnalité Jobs de Citrix ADM ou à l'aide des API NITRO, exécutez les commandes suivantes :

```
    add service s1 172.17.0.8 http 80
    add service s2 172.17.0.9 http 80
    add lb vserver cpx-vip HTTP 10.102.29.102 80
    bind lb vserver cpx-vip s1
    bind lb vserver cpx-vip s2
```

Configurer NetScaler CPX dans Kubernetes à l'aide de ConfigMaps

November 23, 2023

Dans Kubernetes, vous pouvez configurer l'instance NetScaler CPX à l'aide de ConfigMaps. À l'aide de ConfigMaps, vous pouvez configurer dynamiquement l'instance NetScaler CPX lors du démarrage de l'instance.

Créez un fichier cpx.conf de configuration qui inclut la configuration spécifique à NetScaler et les commandes bash shell que vous souhaitez exécuter dynamiquement sur l'instance NetScaler CPX. La

structure du fichier de configuration nécessite deux types de balises, #NetScaler Commands et #Shell Commands. Sous la #NetScaler Commands balise, vous devez ajouter toutes les commandes NetScaler pour configurer la configuration spécifique à NetScaler sur l'instance NetScaler CPX. Sous la #Shell Commands balise, vous devez ajouter les commandes shell que vous souhaitez exécuter sur l'instance NetScaler CPX.

Important

- Les balises peuvent être répétées plusieurs fois dans le fichier de configuration.
- Le fichier de configuration peut également inclure des commentaires. Ajoutez un caractère « # » avant les commentaires.
- Les balises ne sont pas sensibles à la casse.
- S'il existe des scénarios d'échec lors du déploiement du conteneur NetScaler CPX avec le fichier de configuration, les échecs sont enregistrés dans le fichier ns.log.
- Après le démarrage de l'instance NetScaler CPX, si vous modifiez le ConfigMap, la configuration mise à jour est appliquée uniquement lorsque l'instance NetScaler CPX est redémarrée.

Voici un exemple de fichier de configuration :

```
1 #NetScaler Commands
2 add lb vserver v1 http 1.1.1.1 80
3 add service s1 2.2.2.2 http 80
4 bind lb vserver v1 s1
5 #Shell Commands
6 touch /etc/a.txt
7 echo "this is a" > /etc/a.txt
8 #NetScaler Commands
9 add lb vserver v2 http
10 #Shell Commands
11 echo "this is a 1" >> /etc/a.txt
12 #NetScaler Commands
13 add lb vserver v3 http
```

Une fois que vous avez créé le fichier de configuration, vous devez créer un ConfigMap à partir du fichier de configuration à l'aide de la commande kubectl create configmap.

1 kubectl create configmap cpx-config --from-file=cpx.conf

Dans l'exemple ci-dessus, vous pouvez créer un ConfigMap, cpx-config basé sur le fichier de configuration cpx.conf. Vous pouvez ensuite utiliser ce ConfigMap dans le fichier YAML utilisé pour déployer l'instance NetScaler CPX.

Vous pouvez afficher le ConfigMap créé à l'aide de la commande kubectl get configmap. root@node1:~/yaml# kubectl get configmap cpx-config -o yaml

Échantillon :

1	apiVersion: v1
2	data:
3	cpx.conf:
4	#NetScaler Commands
5	add lb vserver v1 http 1.1.1.1 80
6	add service s1 2.2.2.2 http 80
7	bind lb vserver v1 s1
8	#Shell Commands
9	touch /etc/a.txt
10	echo "this is a" > /etc/a.txt
11	echo "this is the file" >> /etc/a.txt
12	ls >> /etc/a.txt
13	#NetScaler Commands
14	add lb vserver v2 http
15	#Shell Commands
16	echo "this is a 1" >> /etc/a.txt
17	#NetScaler Commands
18	add lb vserver v3 http
19	#end of file
20	kind: ConfigMap
21	metadata:
22	creationTimestamp: 2017-12-26T06:26:50Z
23	name: cpx-config
24	namespace: default
25	resourceVersion: "8865149"
26	selfLink: /api/v1/namespaces/ default /configmaps/cpx-config
27	uid: c1c7cb5b-ea05-11e7-914a-926745c10b02

Vous pouvez spécifier le ConfigMap créé cpx-config dans le fichier YAML utilisé pour déployer l' instance NetScaler CPX comme suit :

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4 name: cpx-1
5 labels:
6
    app: cpx-daemon
7 annotations:
     NETSCALER_AS_APP: "True"
8
9 spec:
  hostNetwork: true
10
   containers:
11
   - name: cpx
image: "quay.io/citrix/citrix-k8s-cpx-ingress:13.1-33.47"
     securityContext:
14
15
      privileged: true
     volumeMounts:
16
17
      - name: config-volume
      mountPath: /cpx/bootup_conf
18
19
      env:
      - name: "EULA"
21
        value: "yes"
```

```
- name: "NS_NETMODE"
23
           value: "HOST"
         - name: "kubernetes_url"
24
           value: "https://10.90.248.101:6443"
25
         - name: "NS_MGMT_SERVER"
26
27
           value: "10.90.248.99"
         - name: "NS_MGMT_FINGER_PRINT"
28
           value: "19:71:A3:36:85:0A:2B:62:24:65:0F:7E:72:CC:DC:AD:B8:BF
               :53:1E"
         - name: "NS_ROUTABLE"
           value: "FALSE"
31
         - name: "KUBERNETES_TASK_ID"
32
33
           valueFrom:
34
              fieldRef:
                fieldPath: metadata.name
       imagePullPolicy: Never
37
     volumes:
     - name: config-volume
39
       configMap:
40
         name: cpx-config
```

Une fois que l'instance NetScaler CPX est déployée et démarre, la configuration spécifiée dans le ConfigMap cpx-config est appliquée à l'instance NetScaler CPX.

Déployez les CPX NetScaler en tant que caches DNS locaux pour les nœuds Kubernetes

April 1, 2025

Les espaces d'application d'un cluster Kubernetes dépendent du DNS pour communiquer avec les autres espaces d'application. Les demandes DNS provenant d'applications à l'intérieur d'un cluster Kubernetes sont gérées par Kubernetes DNS (kube-dns). En raison de l'adoption plus large des architectures de microservices, les taux de requêtes DNS au sein d'un cluster Kubernetes augmentent. En conséquence, le DNS Kubernetes (kube-dns) est surchargé. Vous pouvez désormais déployer NetScaler CPX en tant que cache DNS local sur chaque nœud Kubernetes et transférer les requêtes DNS provenant des espaces d'applications du nœud vers NetScaler CPX. Vous pouvez ainsi résoudre les demandes DNS plus rapidement et réduire considérablement la charge sur le DNS Kubernetes.

Pour déployer NetScaler CPX, une entité Kubernetes DaemonSet est utilisée pour planifier les pods NetScaler CPX sur chaque nœud du cluster Kubernetes. Un daemonSet Kubernetes garantit qu'il existe une instance de NetScaler CPX sur chaque nœud Kubernetes du cluster. Pour que les pods d' applications dirigent le trafic vers les pods DNS CPX, vous devez créer un service Kubernetes avec des points de terminaison sous forme de pods NetScaler CPX. L'adresse IP du cluster de ce service est utilisée comme point de terminaison DNS pour les espaces d'application. Pour vous assurer que les modules d'applications utilisent l'adresse IP du cluster de services NetScaler CPX pour la résolution DNS, vous devez mettre à jour le fichier de configuration Kubelet sur chaque nœud avec l'adresse IP du cluster de services NetScaler CPX.

Les variables d'environnement suivantes sont introduites pour prendre en charge le déploiement de NetScaler CPX en tant que cache DNS NodeLocal :

- KUBE_DNS_SVC_IP: Spécifie l'adresse IP du cluster du kube-dns service qui est un argument obligatoire pour déclencher la configuration sur un pod NetScaler CPX. Le pod NetScaler CPX dirige les requêtes DNS vers cette adresse IP lorsque la réponse à la requête DNS n'est pas disponible dans le cache NetScaler CPX.
- CPX_DNS_SVC_IP: Spécifie l'adresse IP du cluster du service NetScaler CPX. La variable d' CPX_DNS_SVC_IP environnement est utilisée pour configurer le DNS local sur les nœuds. Lorsque vous configurez cette variable, une iptables règle est ajoutée pour diriger les requêtes DNS provenant des espaces d'applications vers le pod NetScaler CPX local à l'intérieur du nœud.
- NS_DNS_FORCE_TCP: cette variable d'environnement force l'utilisation de TCP pour les requêtes DNS même si les requêtes sont reçues via UDP.
- NS_DNS_EXT_RESLV_IP: spécifie l'adresse IP du serveur de noms externe pour diriger les demandes DNS pour un domaine spécifique.
- NS_DNS_MATCH_DOMAIN: spécifie la chaîne de domaine externe à comparer pour diriger les requêtes vers le serveur de noms externe.

Déployez des CPX NetScaler sous forme de caches DNS sur des nœuds

Le déploiement de NetScaler CPX en tant que cache DNS local pour un cluster Kubernetes inclut les tâches suivantes :

Sur le nœud maître :

- Création d'un service Kubernetes avec des points de terminaison sous forme de pods NetScaler
 CPX
- Création d'une ConfigMap pour définir des variables d'environnement pour les pods NetScaler
 CPX
- Planifiez des pods NetScaler CPX sur chaque nœud du cluster Kubernetes à l'aide d'un Kubernetes DaemonSet.

Sur les nœuds de travail :

• Modifiez le fichier de configuration Kubelet avec l'adresse IP du cluster du service NetScaler CPX pour transférer les requêtes DNS vers NetScaler CPX.

Configuration sur le nœud maître Kubernetes

Effectuez les étapes suivantes sur le nœud principal Kubernetes pour déployer NetScaler CPX en tant que cache DNS local pour les nœuds :

1. Créez un service avec des pods NetScaler CPX comme points de terminaison à l'aide du fichier. cpx_dns_svc.yaml

```
1 kubectl apply -f cpx_dns_svc.yaml
```

Le fichier cpx_dns_svc.yaml est fourni comme suit :

1	apiVersion: v1
2	kind: Service
3	metadata:
4	name: cpx-dns-svc
5	labels:
6	app: cpxd
7	spec:
8	ports:
9	
10	- protocol: UDP
11	port: 53
12	name: dns
13	
14	- protocol: TCP
15	port: 53
16	name: dns-tcp
17	selector:
18	app: cpx-daemon

2. Obtenez l'adresse IP du service NetScaler CPX.

1 kubectl get svc cpx-dns-svc

3. Obtenez l'adresse IP du service DNS Kube.

1 kubectl get svc -n kube-system

4. Créez une ConfigMap pour définir les variables d'environnement pour les pods NetScaler CPX. Ces variables d'environnement sont utilisées pour transmettre les adresses IP du service NetScaler CPX et du service DNS Kube. Au cours de cette étape, un exemple de ConfigMap cpx-dns-cache est créé à l'aide des variables d'environnement spécifiées en tant que données (paires clé-valeur) dans un fichier.

1 kubectl create configmap cpx-dns-cache --from-file <path-to-file
>

Voici un exemple de fichier contenant les variables d'environnement sous forme de paires clévaleur.

```
1 CPX_DNS_SVC_IP: 10.111.95.145
2 EULA: "yes"
3 KUBE_DNS_SVC_IP: 10.96.0.10
4 NS_CPX_LITE: "1"
5 NS_DNS_EXT_RESLV_IP: 10.102.217.142
6 NS_DNS_MATCH_DOMAIN: citrix.com
7 PLATFORM: CP1000
```

Voici un exemple de ConfigMap :

```
apiVersion: v1
1
     data:
2
3
       CPX_DNS_SVC_IP: 10.111.95.145
4
       EULA: "yes"
       KUBE_DNS_SVC_IP: 10.96.0.10
5
       NS_CPX_LITE: "1"
6
7
       NS_DNS_EXT_RESLV_IP: 10.102.217.142
8
       NS_DNS_MATCH_DOMAIN: citrix.com
9
       PLATFORM: CP1000
10
     kind: ConfigMap
11
     metadata:
12
      creationTimestamp: "2019-10-15T07:45:54Z"
13
      name: cpx-dns-cache
14
      namespace: default
15
       resourceVersion: "8026537"
16
       selfLink: /api/v1/namespaces/default/configmaps/cpx-dns-cache
       uid: 8d06f6ee-133b-4e1a-913c-9963cbf4f48
17
```

5. Créez un DaemonSet Kubernetes pour NetScaler CPX sur le nœud principal.

1 kubectl apply -f cpx_daemonset.yaml

Le fichier cpx_daemonset.yaml est fourni comme suit :

```
apiVersion: apps/v1
1
2
     kind: DaemonSet
3
     metadata:
4
       name: cpx-daemon
5
       labels:
6
              app: cpxd
7
     spec:
8
        selector:
9
              matchLabels:
                app: cpx-daemon
11
     template:
12
        metadata:
13
              labels:
14
                app: cpx-daemon
15
         spec:
16
             containers:
17
18
             - name: cpxd
19
              imagePullPolicy: IfNotPresent
```

```
20
              image: localhost:5000/dev/cpx
21
              volumeMounts:
22
               - mountPath: /netns/default/
23
24
              name: test-vol
25
              ports:
26
              - containerPort: 53
27
28
        envFrom:
29
             - configMapRef:
31
            name: cpx-dns-cache
32
        securityContext:
        privileged: true
34
        allowPrivilegeEscalation: true
        capabilities:
         add: ["NET_ADMIN"]
        volumes:
38
39
            - name: test-vol
40
           hostPath:
41
            path: /proc/1/ns
42
            type: Directory
```

Configuration sur les nœuds de travail dans le cluster Kubernetes

Une fois la configuration terminée sur le nœud maître, effectuez l'étape suivante sur les nœuds de travail :

- Modifiez le fichier de configuration Kubelet afin que les pods d'applications puissent utiliser l'adresse IP du cluster de services NetScaler CPX pour la résolution DNS en suivant l'une des étapes suivantes :
 - Suivez les étapes de la section Reconfigurer le kubelet d'un nœud et modifiez la valeur de l'argument --cluster-dns au format suivant.

```
1 --cluster-dns=<CPX_DNS_SVC_IP>,<KUBE_DNS_SVC_IP>
```

ou

- Modifiez le fichier /etc/systemd/system/kubelet.service.d/10-kubeadm. conf et modifiez l'argument --cluster-dns en suivant les étapes suivantes.
 - a) Modifiez la configuration du kubelet et spécifiez l'adresse IP du cluster du service NetScaler CPX et l'adresse IP du service kube-dns pour l'argument --clusterdns.

```
1 root@node:~# cat /etc/systemd/system/kubelet.service.d
/10-kubeadm.conf | grep KUBELET\_DNS\_ARGS
```

3	Environment="KUBELET_DNS_ARGS=cluster-dns
	=10.111.95.145,10.96.0.10cluster-domain=cluster.
	local"
4	ExecStart=/usr/bin/kubelet
	<pre>\$KUBELET_CONFIG_ARGS \$KUBELET_DNS_ARGS</pre>

- b) Rechargez le kubelet de nœuds à l'aide des commandes suivantes :
 - 1 # systemctl daemon-reload 2 # service kubelet restart

Déployer le proxy NetScaler CPX sur Google Compute Engine

April 1, 2025

Ce guide de déploiement explique comment déployer NetScaler CPX avec Docker sur Google Compute Engine (GCE) de Google Cloud avec Citrix ADM s'exécutant au sein du réseau de l'entreprise. Dans ce déploiement, NetScaler CPX installé sur GCE équilibre la charge de deux serveurs principaux, et Citrix ADM fournit des solutions de licences et d'analyse.

NetScaler CPX est un proxy basé sur des conteneurs qui prend en charge toutes les fonctionnalités de couche 7, le déchargement SSL, plusieurs protocoles et l'API NITRO. Citrix ADM fournit des solutions de gestion, de licences et d'analyse. En tant que serveur de licences, Citrix ADM fournit des droits d'accès aux instances NetScaler CPX qui s'exécutent sur site ou dans le cloud.

CPX et CPX Express sont les mêmes images. Lorsque vous achetez une licence et installez l'image CPX à l'aide de Citrix ADM, l'image CPX dans le Docker App Store (version 11 ou 12) devient une instance CPX complète. Sans licence, l'image CPX devient une instance CPX Express prenant en charge 20 Mbit/s et 250 connexions SSL.

Prérequis

- 2 Go de mémoire et 1 processeur virtuel dédiés à NetScaler CPX
- Open Source Docker disponible auprès de GCE
- · Citrix ADM s'exécutant sur site avec une connexion Internet ou VPN à GCE

Remarque

Pour plus d'informations sur le déploiement de Citrix ADM, consultez la section Déploiement de Citrix ADM.

Étapes de configuration

Vous devez suivre les étapes suivantes pour configurer ce déploiement.

- 1. Installez Docker sur une machine virtuelle GCE.
- 2. Configurez la communication d'API distante avec l'instance Docker.
- 3. Installez l'image NetScaler CPX.
- 4. Créez une instance CPX.
- 5. Accordez une licence NetScaler CPX via Citrix ADM.
- 6. Configurez les services d'équilibrage de charge sur NetScaler CPX et vérifiez la configuration.
 - a) Installez les serveurs Web NGINX.
 - b) Configurez NetScaler CPX pour l'équilibrage de charge et vérifiez la répartition de la charge entre les deux services Web.

Étape 1 : installer Docker sur une machine virtuelle GCE

À partir de GCE, créez une machine virtuelle Linux Ubuntu. Ensuite, installez Docker sur la machine virtuelle à l'aide des commandes illustrées dans l'exemple suivant :

1	<pre>\$ sudo curl - ssl https://get.docker.com/ sh</pre>
2	% Total % Received % Xferd Average Speed Time Time Time Current
3	Dload Upload Total Spent Left Speed
4	0 0 0 0 0 0 0 0::: 0curl: (6) Could not
	resolve host: xnssl-1n0a
5	100 17409 100 17409 0 0 21510 0::: 21492
6	apparmor is enabled in the kernel and apparmor utils were already
	installed
7	\+ sudo -E sh -c apt-key add -
8	<pre>\+ echoBEGIN PGP PUBLIC KEY BLOCK</pre>
9	Version: GnuPG v1
10	
11	mQINBFWln24BEADrBl5p99uKh8+rpvqJ48u4eTtjeXAWbslJotmC/CakbNSqOb9o
12	ddfzRvGVeJVERt/Q/mlvEqgnyTQy+e6oEYN2Y2kqXceUhXagThnqCoxcEJ3+KM4R
13	mYdoe/BJ/J/6rHOjq70mk24z2qB3RU1uAv57iY5VGw5p45uZB4C4pNNsBJXoCvPn
14	TGAs/7IrekFZDDgVraPx/hdiwopQ8NltSfZCyu/jPpWFK28TR8yfVlzYFwibj5WK
15	dHM7ZTqlA1tHIG+agyPf3Rae0jPMsHR6q+arXVwMccy0i+ULU0z8mHUJ3iEMIrpT
16	X+80KaN/ZjibfsBOCjcfiJSB/acn4nxQQgNZigna32velafhQivsNREFeJpzENiG
17	HOoyC6qVeOgKrRiKxzymj0FIMLru/iFF5pSWcBQB7PYlt8J0G80lAcPr6VCiN+4c
18	NKv03SdvA69dC0j79Pu09IIvQsJXsSq96HB+TeEmmL+xSdpGtGdCJHHM1fDeCqkZ
19	hT+RtBGQL2SEdWjxbF43oQopocT8cHvyX6Zaltn0svoGs+wX3Z/H6/8P5anog43U
20	65c0A+64Jj00rNDr8j31izhtQMRo892kGeQAaaxg4Pz6HnS7hRC+cOMHUU4HA7iM
21	zHrouAdYeTZeZEQOA7SxtCME9ZnGwe2grxPXh/U/80WJGkzLFNcTKdv+rwARAQAB
22	tDdEb2NrZXIgUmVsZWFzZSBUb29sIChyZWxlYXNlZG9ja2VyKSA8ZG9ja2VyQGRv
23	Y2tlci5jb20+iQIcBBABCgAGBQJWw7vdAAoJEFyzYeVS+w0QHysP/i37m4SyoOCV

24 25 26 27 28 29 30 31 32 33 34 35 36	cnybll8vzwBEcp4VCRbXvHv0Xty1gccVIV8/aJqNKgBV97lY3vrp0yiIeB8ETQeg srxFE7t/Gz0rsLObqfLEHdmn5iBJRkhLfCpzjeOnyB3Z0IJB6Uog0/msQVYe5CXJ l6uwr0AmoiCBLrVlDAktxVh9RWch0l0KZRX2FpHu8h+uM0/zySqIidlYfLa3y5oH scU+nGU1i6ImwDTD3ysZC5jp9aVfvUmcESyAb4vvdcAHR+bXhA/RW8QHeeMFliWw 7Z2jYHyuHmDnWG2yUrnCqAJTrWV+0fKRIzzJFBs4e88ru5h2ZIXdRepw/+COYj34 LyzxR2cxr2u/xvxwXCkSMe7F4KZAphD+1ws61FhnUMi/PERMYfTFuvPrCkq4gyBj t3fFpZ2NR/fKW87Q0eVcn1ivXl9id3MMs9KXJsg7QasT7mCsee2VIFsxrkFQ2jNp D+JAERRn9Fj4ArHL5TbwkkFbZZvSi6fr5h2GbCAXIGhIXKnjjorPY/YDX6X8AaH0 W1zblWy/CFr6VFl963jrjJgag0G6tNtBZLrclZgWh0QpeZZ5Lbvz2ZA5CqRrfAVc wPNW1f0bFIRtqV6vuVluF0PCMAAnOnqR02w9t17iVQj03oVN0mbQi9vjuExXh1Yo ScVeti06LSmlQfVEVRTqHLMgXyR/EMo7iQIcBBABCgAGBQJXSWBLAAoJEFyzYeVS \+w0QeH0QAI6btAfYwYPuAjfRUy9qlnPhZ+xt1rnwsUzsbmo8K3XTNh+l/R08nu0d sczw30Q1wju28fh1N8ay223+69f0+yICaXqR18Ab6gFGKX7vo0gfEVaxdItUN3eH
37 38	NydGFzmeOKbAlrxIMECnSTG/TkFVYO9Ntlv9vSN2BupmTagTRErxLZKnVsWRzp+X
39	\END PGP PUBLIC KEY BLOCK
40	
41	
42	<pre>\+ sudo -E sn -c mkdir -p /etc/apt/sources.list.d }</pre>
43 ЛЛ	<pre>\+ upkgprint-architecture \+ sudo -E sh -c echo deb \\\[arch=amd64\\\] https://apt</pre>
44	<pre>dockerproject.org/repo ubuntu-yakkety main \> /etc/apt/sources. list.d/docker.list</pre>
45	<pre>\+ sudo -E sh -c sleep 3; apt-get update; apt-get install -y -q docker-engine</pre>
46	Hit:1 http://us-west1.gce.archive.ubuntu.com/ubuntu_vakkety_InRelease
47	Get:2 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates
48	Get:3 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-backports
49	Get:4 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/
50	Get:5 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/
51	Get:6 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/universe
52	Get:7 http://archive.canonical.com/ubuntu yakkety InRelease \[11.5 kB
53	Get:8 http://security.ubuntu.com/ubuntu yakkety-security InRelease
54	Get:9 https://apt.dockerproject.org/repo ubuntu-yakkety InRelease
55	Get:10 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/main
56	Get:11 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
57	Get:12 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
58	Get:13 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
	multiverse Sources \[3,172 B\]
59	<pre>Get:14 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/ main Sources \[107 kB\]</pre>
60	<pre>Get:15 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/</pre>

61	main amd64 Packages \[268 kB\] Get:16 http://us-west1 gce_archive_ubuntu_com/ubuntu_vakketv-undates/
01	main Translation-en $[122 \text{ kB}]$
62	<pre>Get:17 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/ universe amd64 Packages \[164 kB\]</pre>
63	<pre>Get:18 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/ universe Translation-en \[92.4 kB\]</pre>
64	<pre>Get:19 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/ multiverse amd64 Packages \[4.840 B\]</pre>
65	<pre>Get:20 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/ multiverse Translation-en \[2,708 B\]</pre>
66	Get:21 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety- backports/universe Sources \[2.468 B\]
67	Get:22 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety- backports/main Sources \[2,480 B\]
68	Get:23 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety- backports/main amd64 Packages \[3,500 B\]
69	Get:24 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety- backports/universe amd64 Packages \[3.820 B\]
70	Get:25 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety- backports/universe Translation-en \[1.592 B\]
71	Get:26 http://archive.canonical.com/ubuntu yakkety/partner amd64 Packages \[2.480 B\]
72	Get:27 http://security.ubuntu.com/ubuntu yakkety-security/main
73	Get:28 https://apt.dockerproject.org/repo ubuntu-yakkety/main amd64 Packages \[2,453 B\]
74	Get:29 http://security.ubuntu.com/ubuntu yakkety-security/universe
75	<pre>Get:30 http://security.ubuntu.com/ubuntu yakkety-security/multiverse Sources \[1.140 B\]</pre>
76	<pre>Get:31 http://security.ubuntu.com/ubuntu yakkety-security/restricted Sources \[2,292 B\]</pre>
77	Get:32 http://security.ubuntu.com/ubuntu yakkety-security/main amd64 Packages \[150 kB\]
78	Get:33 http://security.ubuntu.com/ubuntu yakkety-security/main Translation-en \[68.0 kB\]
79	Get:34 http://security.ubuntu.com/ubuntu yakkety-security/universe amd64 Packages \[77.2 kB\]
80	Get:35 http://security.ubuntu.com/ubuntu yakkety-security/universe
81	Get:36 http://security.ubuntu.com/ubuntu yakkety-security/multiverse amd64 Packages \[2.832 B\]
82	Fetched 10.8 MB in 2s (4,206 kB/s)
83	Reading package lists Done
84	Reading package lists
85	Building dependency tree
86	Reading state information
87	The following additional packages will be installed:
88	aufs-tools cgroupfs-mount libltdl7
89	The following NEW packages will be installed:
90	aufs-tools cgroupfs-mount docker-engine libltdl7
91	0 upgraded, 4 newly installed, 0 to remove and 37 not upgraded.
```
Need to get 21.2 MB of archives.
      After this operation, 111 MB of additional disk space will be used.
94
      Get:1 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/universe
         amd64 aufs-tools amd64 1:3.2+20130722-1.1ubuntu1 \[92.9 kB\]
      Get:2 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/universe
         amd64 cgroupfs-mount all 1.3 \[5,778 B\]
      Get:3 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/main
         amd64 libltdl7 amd64 2.4.6-1 \[38.6 kB\]
      Get:4 https://apt.dockerproject.org/repo ubuntu-yakkety/main amd64
         docker-engine amd64 17.05.0~ce-0~ubuntu-yakkety \[21.1 MB\]
      Fetched 21.2 MB in 1s (19.8 MB/s)
      Selecting previously unselected package aufs-tools.
100
      (Reading database ...
                              63593 files and directories currently
         installed.)
      Preparing to unpack .../aufs-tools\_1%3a3.2+20130722-1.1ubuntul\
101
         _amd64.deb ...
102
      Unpacking aufs-tools (1:3.2+20130722-1.1ubuntu1) ...
      Selecting previously unselected package cgroupfs-mount.
103
104
      Preparing to unpack .../cgroupfs-mount\_1.3\_all.deb ...
105
      Unpacking cgroupfs-mount (1.3) ...
106
      Selecting previously unselected package libltdl7:amd64.
      Preparing to unpack .../libltdl7\_2.4.6-1\_amd64.deb ...
107
108
      Unpacking libltdl7:amd64 (2.4.6-1) ...
      Selecting previously unselected package docker-engine.
109
110
      Preparing to unpack .../docker-engine\_17.05.0~ce-0~ubuntu-yakkety\
         amd64.deb ...
111
      Unpacking docker-engine (17.05.0~ce-0~ubuntu-vakkety) ...
112
      Setting up aufs-tools (1:3.2+20130722-1.1ubuntu1) ...
113
      Processing triggers for ureadahead (0.100.0-19) ...
114
      Setting up cgroupfs-mount (1.3) ...
115
      Processing triggers for libc-bin (2.24-3ubuntu2) ...
      Processing triggers for systemd (231-9ubuntu4) ...
116
      Setting up libltdl7:amd64 (2.4.6-1) ...
117
118
      Processing triggers for man-db (2.7.5-1) ...
119
      Setting up docker-engine (17.05.0~ce-0~ubuntu-yakkety) ...
120
      Created symlink /etc/systemd/system/multi-user.target.wants/docker.
         service → /lib/systemd/system/docker.service.
121
      Created symlink /etc/systemd/system/sockets.target.wants/docker.
         socket → /lib/systemd/system/docker.socket.
122
      Processing triggers for ureadahead (0.100.0-19) ...
123
      Processing triggers for libc-bin (2.24-3ubuntu2) ...
124
      Processing triggers for systemd (231-9ubuntu4) ...
      \+ sudo -E sh -c docker version
125
126
      Client:
      Version: 17.05.0-ce
127
128
      API version: 1.29
129
      Go version: go1.7.5
130
      Git commit: 89658be
131
      Built: Thu May 4 22:15:36 2017
132
      OS/Arch: linux/amd64
133
134
      Server:
135
      Version: 17.05.0-ce
```

```
API version: 1.29 (minimum version 1.12)
136
137
      Go version: go1.7.5
      Git commit: 89658be
138
139
      Built: Thu May 4 22:15:36 2017
      OS/Arch: linux/amd64
140
141
      Experimental: false
142
143
      If you would like to use Docker as a non-root user, you should now
          consider
144
      adding your user to the "docker" group with something like:
145
      sudo usermod -aG docker albert\_lee
146
147
      Remember that you will have to log out and back in for this to take
148
         effect.
149
150
      WARNING: Adding a user to the "docker" group will grant the ability
         to run
151
      containers which can be used to obtain root privileges on the
152
      docker host.
153
      Refer to https://docs.docker.com/engine/security/\#docker-
          daemon-attack-surface
154
      for more information.
155
156
      $
157
158
      **$ sudo docker info**
159
      Containers: 0
160
      Running: 0
      Paused: 0
161
162
      Stopped: 0
      Images: 0
163
      Server Version: 17.05.0-ce
164
165
      Storage Driver: aufs
166
      Root Dir: /var/lib/docker/aufs
167
      Backing Filesystem: extfs
168
      Dirs: 0
      Dirperm1 Supported: true
169
170
      Logging Driver: json-file
      Cgroup Driver: cgroupfs
171
172
      Plugins:
173
      Volume: local
      Network: bridge host macvlan null overlay
174
175
      Swarm: inactive
176
      Runtimes: runc
      Default Runtime: runc
177
178
      Init Binary: docker-init
      containerd version: 9048e5e50717ea4497b757314bad98ea3763c145
179
      runc version: 9c2d8d184e5da67c95d601382adf14862e4f2228
180
      init version: 949e6fa
181
182
      Security Options:
183
      apparmor
184
      seccomp
```

```
Profile: default
186
      Kernel Version: 4.8.0-51-generic
187
      Operating System: Ubuntu 16.10
      OSType: linux
188
189
   Architecture: x86\_64
190
      CPUs: 1
      Total Memory: 3.613GiB
191
192
      Name: docker-7
193
     ID: R5TW:VKXK:EKGR:GHWM:UNU4:LPJH:IQY5:X77G:NNRQ:HWBY:LIUD:4ELQ
194
      Docker Root Dir: /var/lib/docker
195
      Debug Mode (client): false
      Debug Mode (server): false
196
197
      Registry: https://index.docker.io/v1/
198
      Experimental: false
199
      Insecure Registries:
200
     127.0.0.0/8
201
    Live Restore Enabled: false
202
      WARNING: No swap limit support
204
      $
205
206
      **$ sudo docker images**
207
      REPOSITORY TAG IMAGE ID CREATED SIZE
208
      $
209
210
      **$ sudo docker ps**
211
      CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
212
      Ś
```

Étape 2 : Configurer la communication d'API distante avec l'instance Docker

Ouvrez le port 4243 pour la communication de l'API avec l'instance Docker. Ce port est requis pour que Citrix ADM puisse communiquer avec l'instance Docker.

```
1
2
      **cd /etc/systemd/system**
3
      **sudo vi docker-tcp.socket**
4
      **cat docker-tcp.socket**
5
     \left[ \text{Unit} \right]
     **Description=Docker Socket for the API
6
     \[Socket\]
7
     ListenStream=4243
8
9
     BindIPv60nly=both
10
     Service=docker.service
11
     \[Install\]
     WantedBy=sockets.target**
12
13
14
     $ **sudo systemctl enable docker-tcp.socket**
15
     Created symlink /etc/systemd/system/sockets.target.wants/docker-tcp.
         socket → /etc/systemd/system/docker-tcp.socket.
      **sudo systemctl enable docker.socket**
16
```

```
**sudo systemctl stop docker**
18
      **sudo systemctl start docker-tcp.socket**
      **sudo systemctl start docker**
19
     $ **sudo systemctl status docker**
20
21

    docker.service - Docker Application Container Engine

     Loaded: loaded (/lib/system/docker.service; enabled; vendor
        preset: enabled)
     Active: **active (running)** since Wed 2017-05-31 12:52:17 UTC; 2s
23
        ago
24
     Docs: https://docs.docker.com
25
     Main PID: 4133 (dockerd)
     Tasks: 16 (limit: 4915)
26
27
     Memory: 30.1M
     CPU: 184ms
28
     CGroup: /system.slice/docker.service
29
      -4133 /usr/bin/dockerd -H fd://
     4137 docker-containerd -l unix:///var/run/docker/libcontainerd/
31
        docker-containerd.sock --metrics-interval=0 --start-timeout 2m -
32
33
     May 31 12:52:17 docker-7 dockerd\[4133\]: time="2017-05-31T12
        :52:17.300890402Z" level=warning msg="Your kernel does not support
         cgroup rt peri
34
     May 31 12:52:17 docker-7 dockerd\[4133\]: time="2017-05-31T12
        :52:17.301079754Z" level=warning msg="Your kernel does not support
         cgroup rt runt
     May 31 12:52:17 docker-7 dockerd\[4133\]: time="2017-05-31T12
        :52:17.301681794Z" level=info msg="Loading containers: start."
     May 31 12:52:17 docker-7 dockerd\[4133\]: time="2017-05-31T12
        :52:17.417539064Z" level=info msg="Default bridge (docker0) is
        assigned with an I
37
     May 31 12:52:17 docker-7 dockerd\[4133\]: time="2017-05-31T12
        :52:17.465011600Z" level=info msg="Loading containers: done."
     May 31 12:52:17 docker-7 dockerd\[4133\]: time="2017-05-31T12
38
        :52:17.484747909Z" level=info msg="Daemon has completed
        initialization"
     May 31 12:52:17 docker-7 dockerd\[4133\]: time="2017-05-31T12
        :52:17.485119478Z" level=info msg="Docker daemon" commit=89658be
        graphdriver=aufs
     May 31 12:52:17 docker-7 systemd\[1\]: Started Docker Application
40
        Container Engine.
     May 31 12:52:17 docker-7 dockerd\[4133\]: time="2017-05-31T12
41
        :52:17.503832254Z" level=info msg="API listen on /var/run/docker.
        sock"
     May 31 12:52:17 docker-7 dockerd\[4133\]: time="2017-05-31T12
42
        :52:17.504061522Z" level=info msg="API listen on \[::\]:4243"
43
     $
ΔΔ
45
     (external)$ **curl 104.199.209.157:4243/version**
46
     {
    "Version":"17.05.0-ce","ApiVersion":"1.29","MinAPIVersion":"1.12","
47
       GitCommit":"89658be","GoVersion":"go1.7.5","0s":"linux","Arch":"
       amd64", "KernelVersion": "4.8.0-52-generic", "BuildTime": "2017-05-04
       T22:15:36.071254972+00:00" }
```

48 49 (external)\$

Étape 3 : Installation de NetScaler CPX Image

Téléchargez l'image NetScaler CPX sur Docker App Store. Le CPX Express et le CPX ont la même image. Toutefois, lorsque vous achetez une licence et installez l'image CPX à l'aide de Citrix ADM, l'image devient une instance CPX complète avec des performances de 1 Gbit/s. Sans licence, l'image devient une instance CPX Express prenant en charge 20 Mbit/s et 250 connexions SSL.

```
$ **sudo docker pull store/citrix/citrixadccpx:13.0-36.29**
1
     13.0-36.29: Pulling from store/citrix/citrixadccpx
2
3
     4e1f679e8ab4: Pull complete
4
     a3ed95caeb02: Pull complete
5
     2931a926d44b: Pull complete
     362cd40c5745: Pull complete
6
     d10118725a7a: Pull complete
7
8
     1e570419a7e5: Pull complete
9
     d19e06114233: Pull complete
10 d3230f008ffd: Pull complete
     22bdb10a70ec: Pull complete
11
    1a5183d7324d: Pull complete
12
     241868d4ebff: Pull complete
13
     3f963e7ae2fc: Pull complete
14
15
     fd254cf1ea7c: Pull complete
     33689c749176: Pull complete
16
   59c27bad28f5: Pull complete
17
18
     588f5003e10f: Pull complete
19
     Digest: sha256:31
        a65cfa38833c747721c6fbc142faec6051e5f7b567d8b212d912b69b4f1ebe
     Status: Downloaded newer image for store/citrix/citrixadccpx
        :13.0-36.29
     $
21
22
     $ **sudo docker images**
23
24
     REPOSITORY TAG IMAGE ID CREATED SIZE
     store/citrix/citrixadccpx:13.0-36.29 6fa57c38803f 3 weeks ago 415MB
25
26
     $
```

Étape 4 : Création d'une instance NetScaler CPX

Installez l'image NetScaler CPX sur l'hôte Docker. Ouvrez des ports pour des services spécifiques, comme indiqué dans l'exemple suivant, et spécifiez une adresse IP pour Citrix ADM :

```
1 bash-2.05b\# **CHOST=${
2 1:-localhost }
3 **
```

```
bash-2.05b\# **echo | openssl s\_client -connect $CHOST:443 | openssl
4
         x509 -fingerprint -noout | cut -d'=' -f2**
5
     depth=0 C = US, ST = California, L = San Jose, 0 = NetScaler, 0U =
        Internal, CN = Test Only Cert
6
     verify error:num=18:self signed certificate
7
     verify return:1
     depth=0 C = US, ST = California, L = San Jose, 0 = NetScaler, 0U =
8
        Internal, CN = Test Only Cert
9
     verify return:1
10
     DONE
11
     24:AA:8B:91:7B:72:5E:6E:C1:FD:86:FA:09:B6:42:49:FC:1E:86:A4
12
     bash-2.05b
13
     $ **sudo docker run -dt -p 50000:88 -p 5080:80 -p 5022:22 -p 5443:443
14
         -p 5163:161/udp -e NS\ HTTP\_PORT=5080 -e NS\_HTTPS\_PORT=5443 -e
         NS\_SSH\_PORT=5022 -e NS\_SNMP\_PORT=5163 -e EULA=yes -e LS\_IP=
        xx.xx.xx. - e PLATFORM=CP1000 --privileged=true --ulimit core=-1
        -e NS\_MGMT\_SERVER=xx.xx.xx:xxxx -e NS\_MGMT\_FINGER\_PRINT
        =24:AA:8B:91:7B:72:5E:6E:C1:FD:86:FA:09:B6:42:49:FC:1E:86:A4 --env
         NS\_ROUTABLE=false --env HOST=104.199.209.157 store/citrix/
        citrixadccpx:13.0-36.29**
     44ca1c6c0907e17a10ffcb9ffe33cd3e9f71898d8812f816e714821870fa3538
15
16
     $
17
18
     $ **sudo docker ps**
19
     CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
     44ca1c6c0907 store/citrix/citrixadccpx:13.0-36.29 "/bin/sh -c 'bash
        ..." 19 seconds ago Up 17 seconds 0.0.0.0:5022-\>22/tcp,
        0.0.0.0:5080-\>80/tcp, 0.0.0.0:50000-\>88/tcp, 0.0.0.0:5163-\>161/
        udp, 0.0.0.0:5443-\>443/tcp gifted\_perlman
21
     Ś
22
23
     $ **ssh -p 5022 root@localhost**
24
     root@localhost's password:
25
     Welcome to nsoslx 1.0 (GNU/Linux 4.8.0-52-generic x86\_64)
26
27
     \* Documentation: https://www.citrix.com/
28
     Last login: Mon Jun 5 18:58:51 2017 from xx.xx.xx.xx
     root@44ca1c6c0907:~\#
29
     root@44ca1c6c0907:~\#
31
     root@44ca1c6c0907:~\# **cli\_script.sh 'show ns ip'**
32
     exec: show ns ip
     Ipaddress Traffic Domain Type Mode Arp Icmp Vserver State
     \-----
34
35
     1\) 172.17.0.2 0 NetScaler IP Active Enabled Enabled NA Enabled
     2\) 192.0.0.1 0 SNIP Active Enabled Enabled NA Enabled
37
     Done
     root@44ca1c6c0907:~\# **cli\_script.sh 'show licenseserver'**
38
     exec: show licenseserver
39
     1) ServerName: xx.xx.xx.Port: 27000 Status: 1 Grace: 0 Gptimeleft:
40
         0
     Done
41
     root@44ca1c6c0907:~\# cli\_script.sh 'show capacity'
42
```

```
43
     exec: show capacity
44
     Actualbandwidth: 1000 Platform: CP1000 Unit: Mbps Maxbandwidth: 3000
        Minbandwidth: 20 Instancecount: 0
45
     Done
     root@44ca1c6c0907:~\#
46
47
48
     $ **sudo iptables -t nat -L -n**
49
     Chain PREROUTING (policy ACCEPT)
50
     target prot opt source destination
51
     DOCKER all -- 0.0.0.0/0 0.0.0.0/0 ADDRTYPE match dst-type LOCAL
52
53
     Chain INPUT (policy ACCEPT)
54
     target prot opt source destination
55
     Chain OUTPUT (policy ACCEPT)
56
57
     target prot opt source destination
     DOCKER all -- 0.0.0.0/0 \!127.0.0.0/8 ADDRTYPE match dst-type LOCAL
58
59
     Chain POSTROUTING (policy ACCEPT)
61
     target prot opt source destination
62
     MASQUERADE all -- 172.17.0.0/16 0.0.0.0/0
     MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:443
63
64
     MASQUERADE udp -- 172.17.0.2 172.17.0.2 udp dpt:161
     MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:88
     MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:80
67
     MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:22
68
69
     Chain DOCKER (2 references)
     target prot opt source destination
70
     RETURN all -- 0.0.0.0/0 0.0.0.0/0
71
     DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5443 to:172.17.0.2:443
72
     DNAT udp -- 0.0.0.0/0 0.0.0.0/0 udp dpt:5163 to:172.17.0.2:161
73
     DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:50000 to:172.17.0.2:88
74
     DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5080 to:172.17.0.2:80
75
     DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5022 to:172.17.0.2:22
76
77
     $
```

Étape 5 : Accordez une licence NetScaler CPX via Citrix ADM

En supposant que Citrix ADM s'exécute sur site, vous devriez être en mesure de vérifier que NetScaler CPX communique avec Citrix ADM et envoie des informations. Les images suivantes montrent que NetScaler CPX récupère une licence auprès de Citrix ADM.

NetScaler CPX 14.1

onco Cr	anvar Port Sattinar							
ense se	erver Port Settings							
	Proxy Server Port		License Server Port			Vendor	Daemon Port	
0 27000			27000				7279	
ense Fi	iles							
e followi	ng license files are present on this server. Se	lect Add New Lice	nse to upload more license	es. To delete a license,	select the licens	e and click Delete .		
Add Lice	nse File Apply Licenses Dele	Downlo	ad					
	Name			Last Mo	dified		Size	
	FID3bf0b423_15ba7640cc6_2664.lic			2017-05	-23 21:30:15		1.10 KB	
	FID3bf0b423_15ba7640cc6_2672.lic			2017-05	-23 21:30:15		1.10 KB	
	FID2a2386a8_15b93284902_487e.lic			2017-05	-23 21:30:15		1.10 KB	
	FID2a2386a8_15b93284902_4878.lic			2017-05	-23 21:30:15		1.10 KB	
	FID3bf0b42315ba7640cc65281.lic			2017-05	-23 21:30:15		1.10 KB	
	FID2a2386a8_15b93284902_4870.lic			2017-05-23 21:30:15			1.10 KB	
	FID3bf0b423_15ba7640cc6_527b.lic			2017-05	-23 21:30:15		1.10 KB	
	FID2a2386a8_15b93284902_486a.lic			2017-05	-23 21:30:15		1.10 KB	
FID3bf0b423_15ba7640cc6_5275.lic			2017-05-23 21:30:15 1.10			1.10 KB		
	FID2a2386a8_15b93284902_4864.lic			2017-05	-23 21:30:15		1.10 KB	
ense Ex	xpiry Information							
ature		Count		Days To Ex	piry			
			No iter	ns				
/orks 🕻	Instances > NetScaler CPX							
								(
etSc	aler CPX							
	Desker Hert							
lances	J DOCKER MOST 0							
Add	Provision Delete Dash	board Reb	Action	•			Search	• {
	IP Address Host Name S	tate	Docker Host	Port Range	SSH Port	HTTP Port	HTTPS Port	SNMP F
	172.17.0.2 -NA-	Out of Service	104.196.190.229		32770	32769	32768	32
	172.17.0.5 -NA-	Out of Service	10.10.15.159	88-88	32785	32784	32783	32
	1721702 NA	Up	10/ 100 200 157		5022	5090	5443	5

Networks > License Settings > CPX Licenses						
CPX Licenses				~ \		
Instances 10.0%						
Total 10 Used 1						
The following instances are const	uming Instance license.	Instance True	Allo sobiem Status	Allocated Canacity		
e516b1b61939	172.17.0.2	NetScaler CPX	Not available ()	1		

Étape 6 : configurer les services d'équilibrage de charge sur NetScaler CPX et vérifier la configuration

Tout d'abord, installez les serveurs Web NGINX sur l'hôte Docker. Configurez ensuite l'équilibrage de charge sur NetScaler CPX pour équilibrer la charge des deux serveurs Web et testez la configuration.

Installation des serveurs Web NGINX Utilisez les commandes illustrées dans l'exemple suivant pour installer des serveurs Web NGINX.

```
$ sudo docker pull nginx
1
2
     Using default tag: latest
3
     latest: Pulling from library/nginx
4
     Digest: sha256:41
        ad9967ea448d7c2b203c699b429abe1ed5af331cd92533900c6d77490e0268
5
     Status: Image is up to date for nginx:latest
6
7
8
     **$ sudo docker run -d -p 81:80 nginx**
     098a77974818f451c052ecd172080a7d45e446239479d9213cd4ea6a3678616f
9
10
11
     **$ sudo docker run -d -p 82:80 nginx**
13
     bbdac2920bb4085f70b588292697813e5975389dd546c0512daf45079798db65
14
15
16
     **$ sudo iptables -t nat -L -n**
17
     Chain PREROUTING (policy ACCEPT)
     target prot opt source destination
18
19
     DOCKER all -- 0.0.0.0/0 0.0.0.0/0 ADDRTYPE match dst-type LOCAL
20
21
     Chain INPUT (policy ACCEPT)
22
     target prot opt source destination
23
24
     Chain OUTPUT (policy ACCEPT)
25
     target prot opt source destination
     DOCKER all -- 0.0.0.0/0 \!127.0.0.0/8 ADDRTYPE match dst-type LOCAL
26
27
28
     Chain POSTROUTING (policy ACCEPT)
29
     target prot opt source destination
     MASQUERADE all -- 172.17.0.0/16 0.0.0.0/0
     MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:443
31
32
     MASQUERADE udp -- 172.17.0.2 172.17.0.2 udp dpt:161
     MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:88
     MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:80
34
     MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:22
     MASQUERADE tcp -- 172.17.0.3 172.17.0.3 tcp dpt:80
37
     MASQUERADE tcp -- 172.17.0.4 172.17.0.4 tcp dpt:80
38
39
     Chain DOCKER (2 references)
40
     target prot opt source destination
     RETURN all -- 0.0.0.0/0 0.0.0.0/0
41
```

DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5443 to:172.17.0.2:443 42 43 DNAT udp -- 0.0.0.0/0 0.0.0.0/0 udp dpt:5163 to:172.17.0.2:161 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:50000 to:172.17.0.2:88 44 DNAT tcp -- 0.0.0.0/0 0.0.0/0 tcp dpt:5080 to:172.17.0.2:80 45 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5022 to:172.17.0.2:22 46 47 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:81 to:172.17.0.3:80 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:82 to:172.17.0.4:80 48 49 \$

Configurer NetScaler CPX pour l'équilibrage de charge et vérifier la répartition de la charge entre les deux services Web

```
1
     $ **ssh -p 5022 root@localhost**
2
     root@localhost's password:
3
     Welcome to nsoslx 1.0 (GNU/Linux 4.8.0-52-generic x86\_64)
4
5
     \* Documentation: https://www.citrix.com/
     Last login: Mon Jun 5 18:58:54 2017 from 172.17.0.1
6
7
     root@44ca1c6c0907:~\#
     root@44ca1c6c0907:~\#
8
     root@44ca1c6c0907:~\#
9
     root@44ca1c6c0907:~\#
     root@44ca1c6c0907:~\# **cli\_script.sh "add service web1 172.17.0.3
        HTTP 80"**
     exec: add service web1 172.17.0.3 HTTP 80
12
13
     Done
     root@44ca1c6c0907:~\# **cli\ script.sh "add service web2 172.17.0.4
14
        HTTP 80"**
     exec: add service web2 172.17.0.4 HTTP 80
15
     Done
17
     root@44ca1c6c0907:~\# **cli\_script.sh "add lb vserver cpx-vip HTTP
        172.17.0.2 88"**
     exec: add lb vserver cpx-vip HTTP 172.17.0.2 88
18
19
     Done
20
     root@44ca1c6c0907:~\# **cli\_script.sh "bind lb vserver cpx-vip web1
        "**
21
     exec: bind lb vserver cpx-vip web1
     Done
     root@44ca1c6c0907:~\# **cli\_script.sh "bind lb vserver cpx-vip web2
23
        "**
24
     exec: bind lb vserver cpx-vip web2
25
     Done
26
     root@44ca1c6c0907:~\#
27
28
     root@44ca1c6c0907:~\# **cli\_script.sh 'show lb vserver cpx-vip'**
29
     exec: show lb vserver cpx-vip
     cpx-vip (172.17.0.2:88) - HTTP Type: ADDRESS
31
32
     State: UP
33
     Last state change was at Mon Jun 5 19:01:49 2017
34
     Time since last state change: 0 days, 00:00:42.620
     Effective State: UP
     Client Idle Timeout: 180 sec
     Down state flush: ENABLED
37
```

```
Disable Primary Vserver On Down : DISABLED
39
     Appflow logging: ENABLED
     Port Rewrite : DISABLED
40
           of Bound Services : 2 (Total) 2 (Active)
41
     No.
     Configured Method: LEASTCONNECTION
42
43
     Current Method: Round Robin, Reason: A new service is bound
         BackupMethod: ROUNDROBIN
44
     Mode: IP
     Persistence: NONE
45
46
     Vserver IP and Port insertion: OFF
47
     Push: DISABLED Push VServer:
     Push Multi Clients: NO
48
49
     Push Label Rule: none
50
     L2Conn: OFF
     Skip Persistency: None
51
52
     Listen Policy: NONE
53
     IcmpResponse: PASSIVE
54
     RHIstate: PASSIVE
     New Service Startup Request Rate: 0 PER\ SECOND, Increment Interval:
        0
     Mac mode Retain Vlan: DISABLED
     DBS\_LB: DISABLED
57
58
     Process Local: DISABLED
59
     Traffic Domain: 0
     TROFS Persistence honored: ENABLED
61
     Retain Connections on Cluster: NO
62
     2\) web1 (172.17.0.3: 80) - HTTP State: UP Weight: 1
63
     3\) web2 (172.17.0.4: 80) - HTTP State: UP Weight: 1
64
     Done
     root@44ca1c6c0907:~\#
     (external)$ **curl 104.199.209.157:50000**
68
69
     \<\!DOCTYPE html\>
     \<html\>
71
     72
     \<title\>Welcome to nginx\!\</title\>
73
     <style>
74
     body {
75
76
     width: 35em;
77
     margin: 0 auto;
     font-family: Tahoma, Verdana, Arial, sans-serif;
78
79
      }
81
     \</style>
82
     \langle /head \rangle
83
     \langle dy \rangle
     \<h1\>Welcome to nginx\!\</h1\>
84
     \>If you see this page, the nginx web server is successfully
        installed and
     working. Further configuration is required.\
```

```
>For online documentation and support please refer to
88
89
     \<a href="http://nginx.org/"\>nginx.org\</a\>.\<br/\>
     Commercial support is available at
90
91
     \<a href="http://nginx.com/"\>nginx.com\</a\>.\</p\>
92
93
     ><em>>Thank you for using nginx.</em>>>
94
     \langle /body \rangle
     \</html\>
     (external)$
96
97
99
     (external)$ for i in {
100
    1..100 }
     ; **do curl http://104.199.209.157:50000 -o /dev/null ; done**
102
103
       % Total
                % Received % Xferd Average Speed Time
                                                          Time
      Time Current
104
105
                                    Dload Upload Total
                                                          Spent
      Left Speed
106
                                            0 --:--:--:--:--
     100 612 100 612
                          0 0
                                   1767
107
        --:-- 1768
108
                % Received % Xferd Average Speed Time
109
       % Total
                                                          Time
      Time Current
110
111
                                    Dload Upload Total
                                                          Spent
      Left Speed
112
113
     100 612 100 612
                          0
                               Θ
                                    1893
                                            0 --:--:--:--:--
        --:-- 1894
114
115
       % Total
                 % Received % Xferd Average Speed Time
                                                          Time
      Time Current
116
117
                                    Dload Upload
                                                  Total
                                                          Spent
      Left Speed
118
     100 612 100
                   612
                          0 0
                                   1884 0 --:--:--
119
        --:-- 1883
120
                % Received % Xferd Average Speed
121
       % Total
                                                  Time
                                                          Time
      Time Current
122
123
                                    Dload Upload Total
                                                          Spent
      Left Speed
124
125
     100 612 100 612
                           0
                                Θ
                                    1917 0 --:--:--
        --:-- 1924
126
                % Received % Xferd Average Speed Time
127
       % Total
                                                          Time
      Time Current
```

128 129 Dload Upload Total Spent Left Speed 130 100 612 100 612 0 0 1877 0 --:--:--131 --:-- 1883 132 % Total % Received % Xferd Average Speed Time Time 133 Time Current 134 135 Dload Upload Total Spent Left Speed 136 100 612 100 612 0 0 1852 0 --:--:--137 --:-- 1848 138 % Total % Received % Xferd Average Speed Time 139 Time Time Current 140 Dload Upload Total Spent 141 Left Speed 142 100 612 100 612 0 0 1860 0 --:--:--143 --:-- 1865 144 145 % Total % Received % Xferd Average Speed Time Time Time Current 146 Dload Upload Total Spent 147 Left Speed 148 100 612 100 612 0 0 1887 0 --:--:--149 --:-- 1888 150 % Total % Received % Xferd Average Speed Time 151 Time Time Current 152 Dload Upload Total Spent 153 Left Speed 154 100 612 100 612 0 0 1802 0 --:-:--155 --:-- 1800 156 % Total % Received % Xferd Average Speed Time Time 157 Time Current 158 Dload Upload Total Spent 159 Left Speed 160 100 612 100 612 0 0 1902 0 --:--:--161 --:-- 1906 162 % Total % Received % Xferd Average Speed Time Time 163

Time Current 164 Dload Upload Total Spent 165 Left Speed 166 100 612 100 612 0 0 1843 0 --:--:--167 --:-- 1848 168 169 170 % Total % Received % Xferd Average Speed Time Time 171 Time Current 172 Dload Upload Total Spent 173 Left Speed 174 100 612 100 612 0 0 1862 0 --:--:--175 --:-- 1860 176 % Total % Received % Xferd Average Speed Time 177 Time Time Current 178 Dload Upload Total Spent 179 Left Speed 180 100 612 100 612 0 0 1806 0 --:--:--181 --:-- 1810 % Total % Received % Xferd Average Speed Time 183 Time Time Current 184 Dload Upload Total Spent 185 Left Speed 186 187 100 612 100 612 0 0 1702 0 --:--:----:-- 1704 188 189 (external)\$ 190 191 192 193 194 195 root@44ca1c6c0907:~\# **cli_script.sh 'stat lb vserver cpx-vip'** 196 197 exec: stat lb vserver cpx-vip 198 199 Virtual Server Summary 201 202 vsvrIP port Protocol State 203 Health actSvcs

204						
205	cpx-vip 2	172.17.0.2	88	HTTP	UP	100
206						
207						
208						
200	in	actSvcs				
205		actoves				
210	and other	0				
211	cpx-vip	0				
212						
213						
214						
215	Virtual Serve	r Statistics				
216						
217				Rate (/s)		
	Total					
218						
219	Vserver hits 101			Θ		
220						
221	Requests			Θ		
	101					
222						
223	Responses			Θ		
		101				
224						
225	Request bytes			Θ		
	8585					
226						
227	Response byte	S		Θ		
	85850					
228						
229	Total Packets	rcvd		0		
	708					
230						
231	Total Packets	sent		0		
	408					
232						
233	Current clien	t connections				
200	current ceren	0				
234		Ŭ				
235	Current Clien	+ Est connections				
200	current ctren					
236		0				
230	Current sorve	r connections				
251	current serve					
220		U				
238	Current Dave	atomaa Caratana				
239	Current Persi	scence Sessions				
0.46		Θ				
240						
241	Requests in s	urge queue				
		Θ				
242						

243	Requests in vserver's	surgeQ 0					
244							
245	Requests in service's	surgeQs 0					
246							
247	Spill Over Threshold	Θ					
248							
249	Spill Over Hits	Θ					
250							
251	Labeled Connection	Θ					
252							
253	Push Labeled Connection	on 0					
254							
255	Deferred Request	0			Θ		
256							
257	Invalid Request/Respon	nse 0					
258	T 1.1 D 1/D	5					
259	Invalid Request/Respon	o 0	I				
260		:+					
261	vserver bown Backup H	0					
262	Current Multipath TCP	sossions					
264		0					
265	Current Multipath TCP	subflows					
266		0					
267	Andex for client resp	onse times.					
268	1	.00					
269	Average client TTLR						
		0					
270							
271	web1 172	.17.0.3	80	HTTP		UP	51
272	0/s						
273	web2 172	.17.0.4	80	HTTP		UP	50
	0/s						
274							
275	Done						
276							
277	root@44ca1c6c0907:~\#						

Résolution des problèmes liés à NetScaler CPX

July 22, 2024

Ce document explique comment résoudre les problèmes que vous pouvez rencontrer lors de l'utilisation de NetScaler CPX. À l'aide de ce document, vous pouvez collecter des journaux pour déterminer les causes et appliquer des solutions de contournement à certains des problèmes courants liés à l' installation et à la configuration de NetScaler CPX.

• Pourquoi le pourcentage d'utilisation de la mémoire est-il toujours élevé dans NetScaler CPX ?

Le pourcentage d'utilisation de la mémoire est déterminé en comparant l'utilisation actuelle de la mémoire de NetScaler CPX à la mémoire totale que NetScaler CPX s'alloue à partir du système d'exploitation (OS). NetScaler CPX commence par allouer un minimum de mémoire nécessaire au système d'exploitation pour le démarrage et les opérations de base. Au départ, étant donné que NetScaler CPX alloue le minimum de mémoire nécessaire à ses opérations, le pourcentage de consommation de mémoire est élevé. Finalement, à mesure que les besoins en ressources augmentent avec la mise à l'échelle des opérations NetScaler CPX, NetScaler CPX alloue de manière incrémentielle de la mémoire supplémentaire depuis le système d'exploitation et l'utilisation de la mémoire est également élevée, ce qui se traduit par un pourcentage d'utilisation de la mémoire élevé.

Ainsi, en raison de la stratégie d'allocation de mémoire dynamique de NetScaler CPX, NetScaler CPX affiche généralement un pourcentage d'utilisation de la mémoire élevé, reflétant son utilisation efficace des ressources plutôt qu'un manque de mémoire disponible.

• Comment puis-je consulter les journaux NetScaler CPX ?

Vous pouvez consulter les journaux de NetScaler CPX à l'aide de la kubectl logs commande si NetScaler CPX est déployé avec cette option. tty:true Vous pouvez exécuter la commande suivante pour afficher les journaux :

```
1 kubectl logs <pod-name> [-c <container-name>] [-n <namespace-name
>]
```

Exemple,

1 kubectl logs cpx-ingress1-69b9b8c648-t8bgn -c cpx -n citrix-adc

Voici un exemple de déploiement du pod NetScaler CPX avec l'option : tty:true

```
containers:
    - name: cpx-ingress
    image: "quay.io/citrix/citrix-k8s-cpx-ingress:13.0-58.30"
    tty: true
    securityContext:
    privileged: true
```

7 env:

Vous trouverez d'autres journaux de démarrage dans le fichier /*cpx/log/boot.log* du système de fichiers NetScaler CPX.

Remarque : Pour obtenir le nom du pod, exécutez la commande kubectl get pods -o wide.

• Comment puis-je obtenir le pack de support technique auprès de NetScaler CPX ?

Vous pouvez exécuter la commande suivante sur l'interface shell du nœud principal Kubernetes pour collecter le bundle de support technique NetScaler CPX :

```
1 kubectl exec <cpx-pod-name> [-c <cpx-container-name>] [-n <
            namespace-name>] /var/netscaler/bins/cli_script.sh "show
            techsupport"
```

Vous pouvez consulter le bundle de support technique dans le répertoire /var/tmp/support du système de fichiers de NetScaler CPX. Utilisez scp ou copiez kubectl cp le bundle de support technique depuis NetScaler CPX vers la destination souhaitée.

Exemple :

1	<pre>root@localhost# kubectl exec cpx-ingress1-55b9b6fc75-t5kc6 -c cpx</pre>
	-n citrix-adc /var/netscaler/bins/cli_script.sh "show
	techsupport"
2	exec: show techsupport
3	Scope: NODE
4	Done
5	<pre>root@localhost# kubectl cp cpx-ingress1-55b9b6fc75-t5kc6:var/tmp/</pre>
	<pre>support/collector_P_192.168.29.232_31Aug2020_07_30.tar.gz /tmp</pre>
	/collector_P_192.168.29.232_31Aug2020_07_30.tar.gz -c cpx
6	root@localhost# ll /tmp/collector_P_192.168.29.232
	_31Aug2020_07_30.tar.gz
7	-rw-rr 1 root root 1648109 Aug 31 13:23 /tmp/collector_P_192
	.168.29.232_31Aug2020_07_30.tar.gz

• Pourquoi le pod NetScaler CPX est-il bloqué lors du démarrage ?

Vous pouvez vérifier l'état de l'espace à l'aide de la commande kubectl describe pods. Exécutez la commande suivante pour connaître l'état de l'espace :

Exemple :

1 kubectl describe pods cpx-ingress1-69b9b8c648-t8bgn

Si les événements de l'espace indiquent que le conteneur est démarré, vous devez vérifier les journaux de l'espace.

• Comment copier des fichiers entre le pod NetScaler CPX et le nœud principal Kubernetes ?

Il est recommandé d'utiliser la fonction de montage de volume du menu fixe pour monter le répertoire / cpx sur le système de fichiers de l'hôte. Si un conteneur NetScaler CPX quitte les fichiers core-dumps, les journaux et autres données importantes sont disponibles sur le point de montage.

Vous pouvez utiliser l'une des commandes suivantes pour copier des fichiers entre le pod NetScaler CPX et le nœud principal Kubernetes :

kubectl cp : Vous pouvez exécuter la commande suivante pour copier des fichiers d'un espace à un autre :

Exemple :

```
1 root@localhost:~# kubectl cp cpx-ingress-596d56bb6-zbx6h:cpx/log/
	boot.log /tmp/cpx-boot.log -c cpx-ingress
2 root@localhost:~# ll /tmp/cpx-boot.log
3 -rw-r--r-- 1 root root 7880 Sep 11 00:07 /tmp/cpx-boot.log
```

scp : Vous pouvez utiliser la commande pour copier des fichiers entre le pod NetScaler CPX et le nœud Kubernetes. Exécutez la commande suivante pour copier des fichiers d'un espace vers un nœud. Lorsqu'il vous demande le mot de passe, indiquez le mot de passe de l'utilisateur SSH :

scp <user>@<pod-ip>:<absolute-src-path> <dst-path>

Exemple :

```
1 root@localhost:~# scp nsroot@192.168.29.198:/cpx/log/boot.log /
    tmp/cpx-boot.log
2 nsroot@192.168.29.198's password:
3 boot.log
4 100% 7880 5.1MB/s 00:00
5 root@localhost:~#
```

Comment puis-je capturer des paquets sur NetScaler CPX ?

Pour capturer des paquets sur NetScaler CPX, lancez l'interface shell de NetScaler CPX à l'aide de la commande kubectl exec. Exécutez la commande suivante pour lancer l'interface shell du pod NetScaler CPX :

```
1 kubectl exec -it pod-name [-c container-name] [-n namespace-
name] bash
```

Exemple :

1 kubectl exec -it cpx-ingress1-**69**b9b8c648-t8bgn -c cpx -n citrix-adc bash

Ensuite, exécutez la commande suivante pour commencer la capture de paquets :

1 cli_script.sh "start nstrace -size 0"

Si vous souhaitez arrêter la capture de paquets en cours, exécutez la commande suivante :

1 cli_script.sh "stop nstrace"

Vous pouvez consulter les paquets capturés dans un fichier *.cap* situé dans le répertoire */cpx/nstrace/time-stamp* du système de fichiers NetScaler CPX.

 Pourquoi le serveur de licences n'est-il pas configuré même lorsque NetScaler CPX est déployé avec la variable d'environnement LS_IP=<ADM-IP> ?

Assurez-vous que le serveur de licences est accessible depuis le nœud sur lequel NetScaler CPX est déployé. Vous pouvez utiliser la ping <ADM-IP> commande pour vérifier la connectivité entre le nœud NetScaler CPX et Citrix ADM.

Si Citrix ADM est accessible depuis le nœud, vous devez vérifier les journaux de configuration du serveur de licences dans le fichier /cpx/log/boot.log. Vous pouvez également vérifier la configuration du serveur de licences à l'aide de la commande suivante sur l'interface shell du pod NetScaler CPX :

```
1 cli_script.sh "show licenseserver"
```

Exemple :

```
1 root@cpx-ingress-596d56bb6-zbx6h:/cpx/log# cli_script.sh "show
licenseserver"
2 exec: show licenseserver
3 ServerName: 10.106.102.199Port: 27000 Status: 1 Grace: 0
Gptimeleft: 720
4 Done
```

• Pourquoi la licence groupée n'est-elle pas configurée sur NetScaler CPX même après une configuration réussie du serveur de licences sur NetScaler CPX ?

Vérifiez les journaux de configuration des licences dans le fichier /*cpx/log/boot.log*. Vous pouvez également vérifier la licence groupée configurée sur NetScaler CPX à l'aide de la commande suivante sur l'interface shell du pod NetScaler CPX :

1 cli_script.sh "show capacity"

Exemple,

```
2 exec: show capacity
3 Actualbandwidth: 1000 MaxVcpuCount: 2 Edition: Platinum
Unit: Mbps Bandwidth: 0` `Maxbandwidth: 40000
Minbandwidth: 20 Instancecount: 1
4 Done
```

Assurez-vous également que les fichiers de licences requis sont téléchargés sur le serveur de licences. Vous pouvez également vérifier les licences disponibles sur le serveur de licences une fois qu'il est correctement configuré sur NetScaler CPX à l'aide de la commande suivante. Exécutez la commande sur l'interface shell du pod NetScaler CPX :

cli_script.sh "sh licenseserverpool"

Exemple :

1	root@cpx-ingress-596d56bb6-zbx6h:/cpx/log# cli_script.sh "show					
	licenseserverpool"					
2	exec: show licenseserverpool					
3	Instance Total	: 5				
4	Instance Available	: 4				
5	Standard Bandwidth Total	: O Mbps				
6	Standard Bandwidth Availabe	: O Mbps				
7	Enterprise Bandwidth Total	: O Mbps				
8	Enterprise Bandwidth Available	: O Mbps				
9	Platinum Bandwidth Total	: 10.00 Gbps				
10	Platinum Bandwidth Available	: 9.99 Gbps				
11	CP1000 Instance Total	: 100				
12	CP1000 Instance Available	: 100				
13	Done					

 Pourquoi les appels d'API NITRO reçoivent-ils une réponse de refus de connexion de la part de NetScaler CPX ?

Le port par défaut pour les API NITRO est 9080 (non sécurisé) et 9443 (sécurisé) à partir de la version 12.1 de NetScaler CPX. Assurez-vous que le port NITRO de NetScaler CPX auquel vous essayez d'accéder est exposé sur le pod. Vous pouvez exécuter la commande kubectl describe pour afficher le port exposé et mappé du conteneur NetScaler CPX dans la section Conteneur NetScaler CPX :

```
kubectl describe pods <pod-name> | grep -i port
```

Exemple :

1 ng472 | grep -i port
 Ports:
 80/TCP, 443/TCP, 9080/TCP, 9443/TCP

 Host Ports:
 0/TCP, 0/TCP, 0/TCP, 0/TCP
 2 3 9080 4 NS_HTTP_PORT: 5 NS_HTTPS_PORT: 9443 6 Port: <none> 7 Host Port: <none> 8 NS_PORT: 80

• Pourquoi le processus NSPPE de NetScaler CPX consomme-t-il la majeure partie du processeur, même en cas de faible trafic ou d'absence de trafic ?

Si NetScaler CPX est déployé avec la variable d'environnement CPX_CONFIG='{ "YIELD": "NO" } ', le processus NSPPE utilise 100 % du processeur, même lorsque le trafic est nul ou faible. Si vous souhaitez que le processus NSPPE n'utilise pas le processeur, vous devez déployer NetScaler CPX sans la variable d'environnement. CPX_CONFIG='{ "YIELD": "NO" } Par défaut, le processus NSPPE dans CPX est configuré pour ne pas surcharger ou consommer l'utilisation du processeur.

• Pourquoi NetScaler CPX n'est-il pas répertorié dans Citrix ADM alors qu'il a été déployé avec les variables d'environnement requises pour l'enregistrement auprès de Citrix ADM ?

Les journaux relatifs à l'enregistrement de NetScaler CPX auprès de Citrix ADM se trouvent dans le fichier */cpx/log/boot.log* du système de fichiers NetScaler CPX.

Vous pouvez vérifier l'accessibilité de l'adresse IP Citrix ADM à partir du pod NetScaler CPX à l' aide de la commande. ping Assurez-vous également que toutes les variables d'environnement requises pour l'enregistrement de Citrix ADM sont configurées pour le conteneur NetScaler CPX.

- NS_MGMT_SERVER: spécifie l'adresse IP ou le nom de domaine complet ADM-IP.
- HOST: spécifie l'adresse IP du nœud.
- NS_HTTP_PORT: spécifie le port HTTP mappé sur le nœud.
- NS_HTTPS_PORT: spécifie le port HTTPS mappé sur le nœud.
- NS_SSH_PORT: spécifie le port SSH mappé sur le nœud.
- NS_SNMP_PORT: spécifie le port SNMP mappé sur le nœud.
- NS_ROUTABLE: L'adresse IP du pod NetScaler CPX n'est pas routable depuis l'extérieur.
- NS_MGMT_USER: spécifie le nom d'utilisateur ADM.
- NS_MGMT_PASS: spécifie le mot de passe ADM.
- Pourquoi cli_script.sh affiche le message d'erreur *Nom d'utilisateur ou mot de passe non valide* affiche-t-il après la modification du mot de passe de l'utilisateur nsroot ?

La commande cli_script.sh est un utilitaire d'encapsulation pour NSCLI sur NetScaler CPX. Il exécute le premier argument en tant que chaîne de commande ou chemin de fichier et le second argument est facultatif, qui est des informations d'identification. Si le mot de passe de l'utilisateur nsroot est modifié, vous devez fournir des informations d'identification en tant que deuxième argument de cli_script.sh. Vous pouvez exécuter la commande suivante pour exécuter NSCLI avec des informations d'identification :

1 cli_script.sh " <command> " ":<username>:<password> "

Exemple :

```
root@087a1e34642d:/# cli_script.sh "show ns ip"
1
2
   exec: show ns ip
3
4
   ERROR: Invalid username or password
5
6 root@087a1e34642d:/# cli_script.sh "show ns ip" ":nsroot:
    nsroot123"
7
8
   exec: show ns ip
9
   IpaddressTraffic DomainTypeArpIcmpVserverState
10
                                                     Mode
   _____
                                                     ___
11
                 ----
      2.17.0.3 0
12 172.17.0.3
                                   NetScaler IP Active
     Enabled Enabled NA Enabled
22.0.0.1 0 SI
13 192.0.0.1
                             SNIP
                                                 Active
    Enabled Enabled NA Enabled
14 Done
```

• Pourquoi le SSH vers NetScaler CPX échoue-t-il avec l'utilisateur root et nsroot?

À partir de la version 13.0 à 64.35, NetScaler CPX génère un mot de passe par défaut et le met à jour pour les utilisateurs SSH - et. rootnsroot Si vous n'avez pas modifié le mot de passe manuellement, le mot de passe des utilisateurs SSH se trouve dans /var/deviceinfo/ random_id le système de fichiers de NetScaler CPX.

net>scaler

© 2025 Cloud Software Group, Inc. All rights reserved. This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc. This and other products of Cloud Software Group may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal. Citrix, the Citrix logo, NetScaler, and the NetScaler logo and other marks appearing herein are either registered trademarks or trademarks of Cloud Software Group, Inc. and/or its subsidiaries in the United States and/or other countries. Other marks are the property of their respective owner(s) and are mentioned for identification purposes only. Please refer to Cloud SG's Trademark Guidelines and Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.