



NetScaler ADC CPX 12.1

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	2
Licence NetScaler CPX	6
Déploiement d'une instance NetScaler CPX dans Docker	14
Ajouter des instances NetScaler CPX à Citrix ADM	21
Configuration de NetScaler CPX	24
Configuration de NetScaler CPX à l'aide d'un fichier de configuration	28
Configuration des pilotes de journalisation Docker	30
Support du routage dynamique dans NetScaler CPX	31
Mise à niveau d'une instance NetScaler CPX	34
Utilisation de serveurs virtuels génériques dans une instance NetScaler CPX	36
Déployer NetScaler CPX en tant que proxy pour permettre un flux de trafic Est-Ouest	37
Déployer NetScaler CPX sur un réseau hôte unique	41
Déployer NetScaler CPX dans un réseau multi-hôtes	42
Déployez NetScaler CPX avec un accès direct au réseau	48
Configurer NetScaler CPX dans Kubernetes à l'aide de ConfigMaps	49
Déployer le proxy NetScaler CPX sur Google Compute Engine	52
Résolution des problèmes liés à NetScaler CPX	72

À propos de NetScaler CPX

November 23, 2023

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.

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
- Sécurité des applications
 - Réécriture et répondeur L7
- Facilité de gestion simple
 - Journalisation Web
 - AppFlow

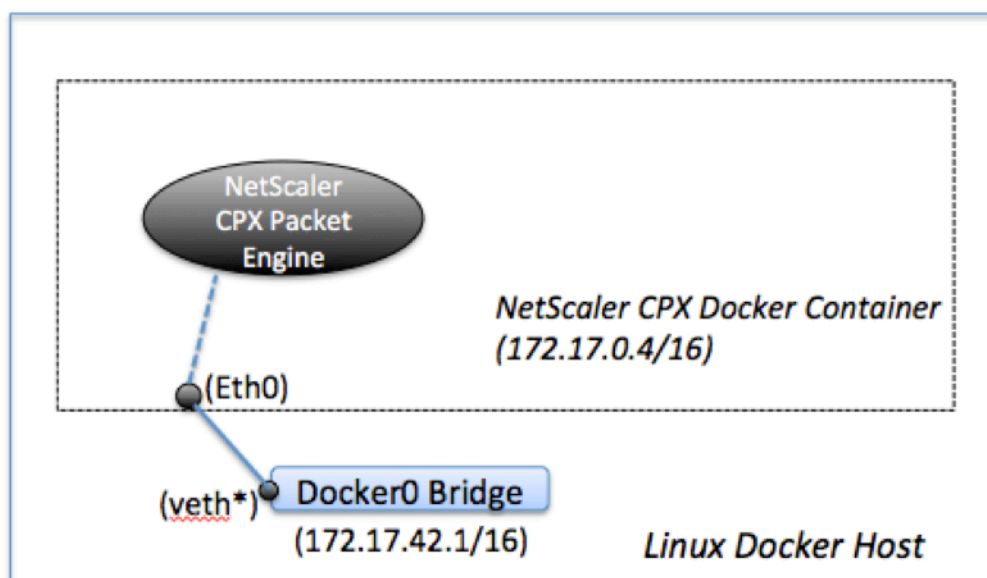
Architecture et flux de trafic

November 23, 2023

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

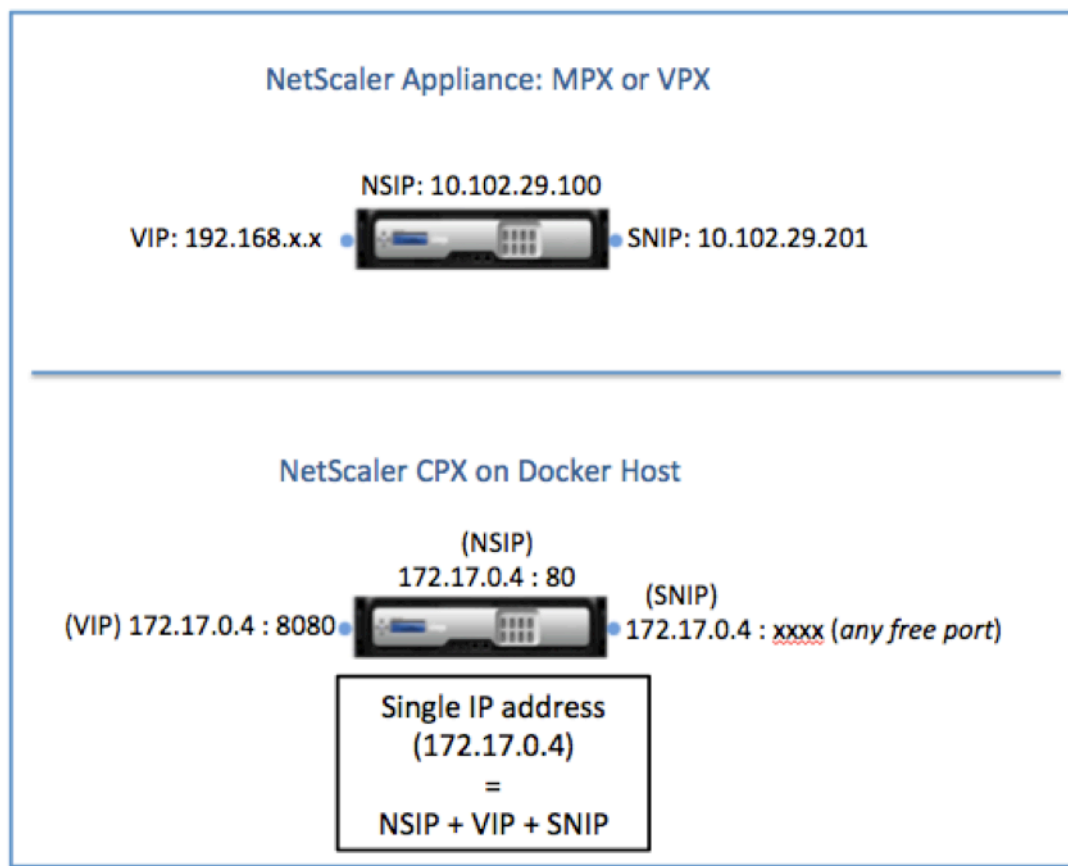
Une appliance NetScaler MPX ou VPX standard nécessite au moins trois adresses IP pour fonctionner :

- 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.

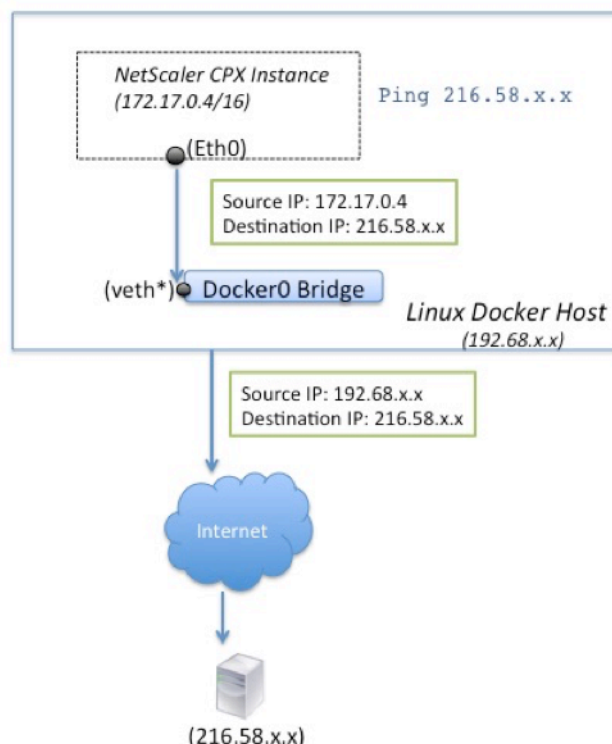
L'image suivante illustre comment une seule adresse IP est utilisée pour exécuter les fonctions de NSIP, SNIP et VIP.



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.



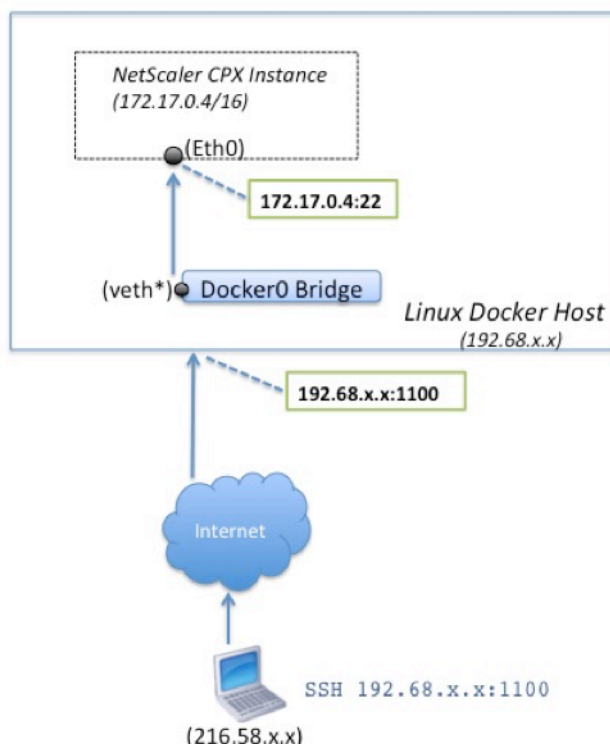
Dans cet exemple, la demande ping est envoyée par le moteur de paquets sur l'interface eth0 avec l'adresse IP source comme adresse IP NetScaler CPX (172.17.0.4). L'hôte Docker effectue ensuite une traduction d'adresse réseau (NAT) pour ajouter l'adresse IP de l'hôte (192.68.x.x) en tant qu'adresse IP 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

March 21, 2024

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 également disponible sur site et dans le cloud. Vous pouvez utiliser Citrix ADM pour gérer les licences de capacité groupée pour tous les facteurs de forme NetScaler ADC.

Pour plus d'informations sur Citrix ADM local, consultez [Citrix ADM local](#). Pour plus d'informations sur le service Citrix ADM, consultez la section [Service Citrix ADM](#).

Types de licences NetScaler CPX

NetScaler CPX prend en charge les licences de bande passante et de pool de processeurs virtuels (cœur) 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. Pour plus d'informations sur le pool de bande passante, consultez [Licence basée sur la bande passante](#).

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 vCPU, consultez la section [Licences de processeur virtuel NetScaler ADC](#).

Capacité groupée prise en charge pour les instances NetScaler CPX

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

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](#), il s'agit de la capacité par défaut disponible pour les POC qui ne nécessitent pas de fichier de licence. Elle est dotée des fonctionnalités suivantes :

- Bande passante 20 Mbps
- 250 sessions SSL maximum
- Débit SSL de 20 Mbps

Vous devez attribuer une licence à votre instance NetScaler CPX pour la mettre à niveau afin d'améliorer les performances et les déploiements de production.

Modèles de licence NetScaler CPX

Citrix propose une gamme de modèles de licences de produits pour NetScaler CPX afin de répondre aux exigences de votre entreprise. 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

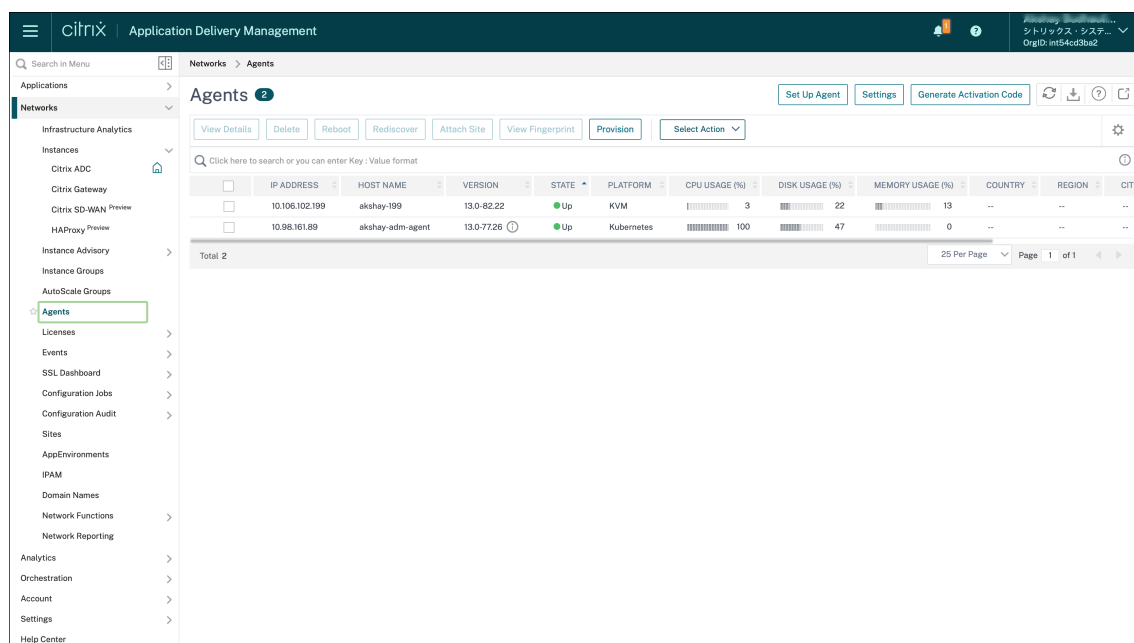
Fournissez des licences basées sur la bande passante et le vCPU à partir du service Citrix ADM pour NetScaler CPX

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.

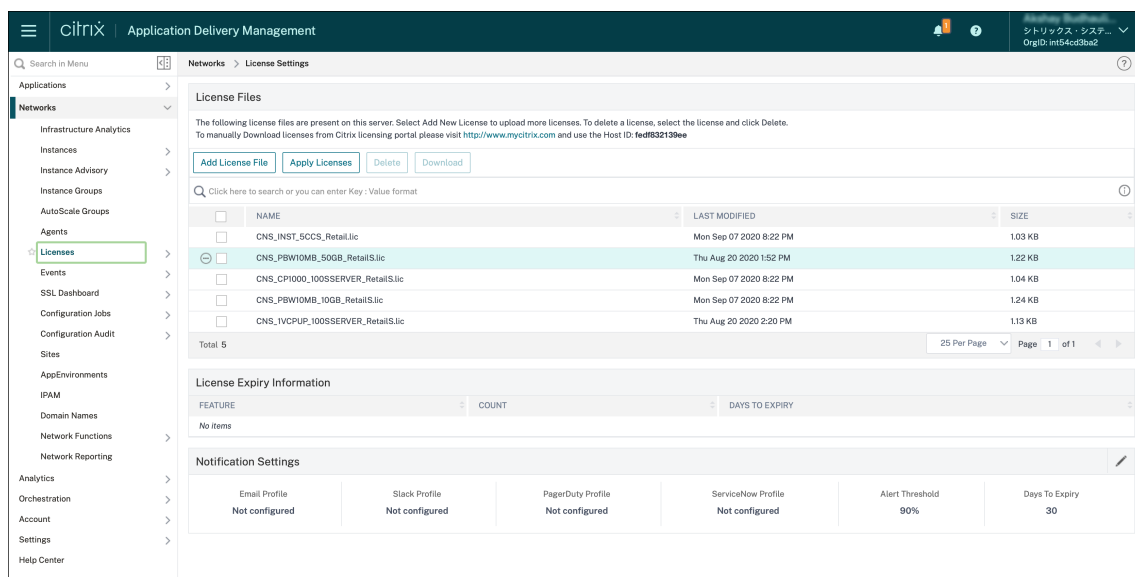
Assurez-vous que la configuration du service Citrix ADM est opérationnelle avec l'agent Citrix ADM. Vous devez disposer d'un service Citrix ADM et d'un compte d'agent Citrix ADM pour que la licence NetScaler CPX soit fonctionnelle. [Pour plus d'informations sur la configuration du service Citrix ADM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

Remarque : Dans cette procédure, une configuration de l'agent Citrix ADM d'un hyperviseur (sur site) est utilisée. L'image suivante montre l'agent local utilisé pour l'octroi de licences NetScaler CPX. 10.106.102.199



2. Ajoutez le pool de licences d'instance NetScaler ADC 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 sur Citrix ADM, consultez [Configurer la capacité mise en pool](#). Dans l'image suivante, `CNS_INST_200CC_Retail.lic` est utilisé comme bande passante et pool de licences d'instance.



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 :

- name: "LS_IP"
value: "10.105.158.166"//IP de l'agent ADM comme indiqué à l'étape 1
- name: "LS_PORT"
value: "27000"// port sur lequel le serveur de licences ADM écoute
- name: "BANDWIDTH"
value: "3000"//la capacité en Mbps que vous voulez allouer à CPX

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

- name: "LS_IP"
value: "10.102.216.173"//IP de l'agent ADM comme indiqué à l'étape 1
- name: "LS_PORT"
value: "27000"// port sur lequel le serveur de licences ADM écoute
- name: "CPX_CORES"
value: "4"// nombre de cœurs que vous souhaitez allouer

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

```
1 kubectl create namespace bandwidth
```

```
2  wget https://raw.githubusercontent.com/citrix/cloud-native-  
    getting-started/master/cpx-licensing/manifest/cpx-bandwidth-  
    license-adm-service.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-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

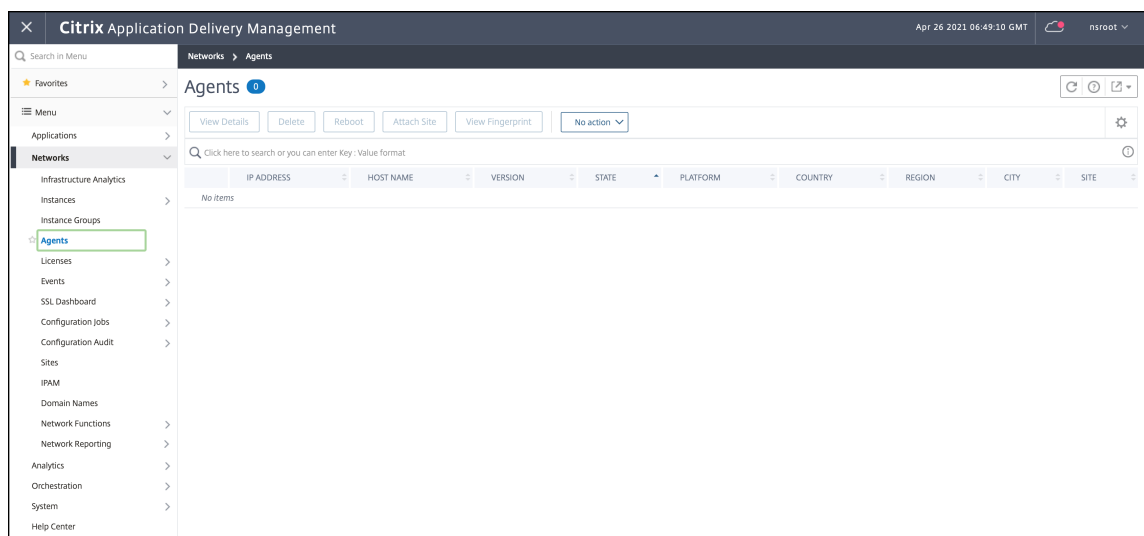
Procédez comme suit pour approvisionner NetScaler CPX en fonction de la bande passante et du vCPU à partir de Citrix ADM sur site.

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.

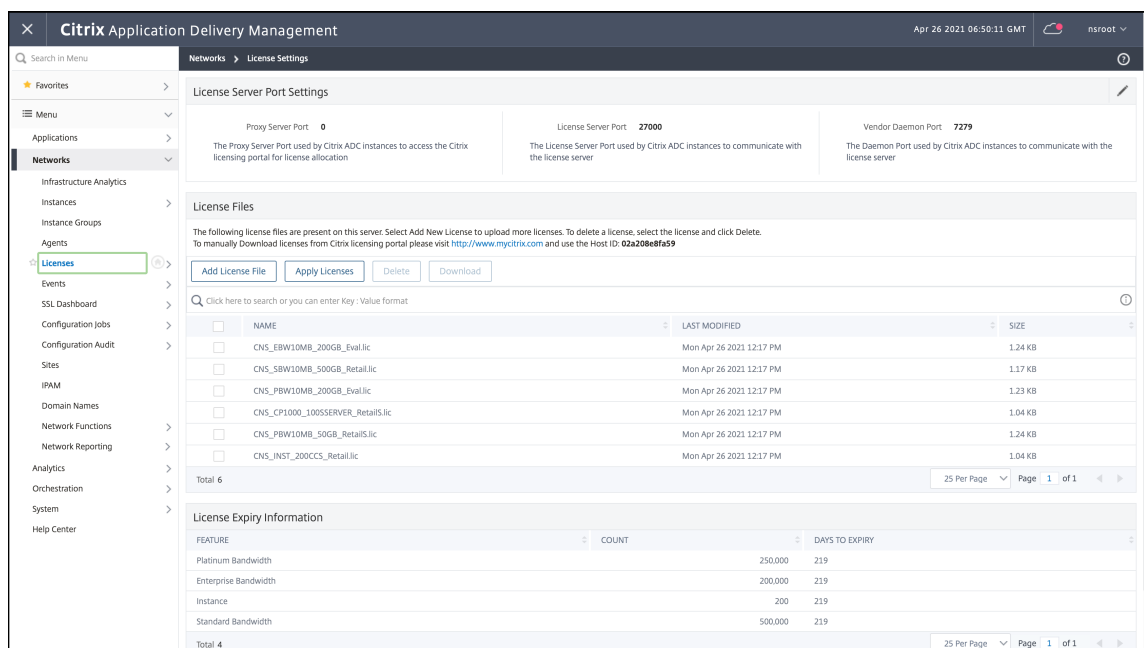
[Pour plus d'informations sur la configuration locale de Citrix ADM et de l'agent Citrix ADM, consultez le service Citrix ADM.](#)

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.

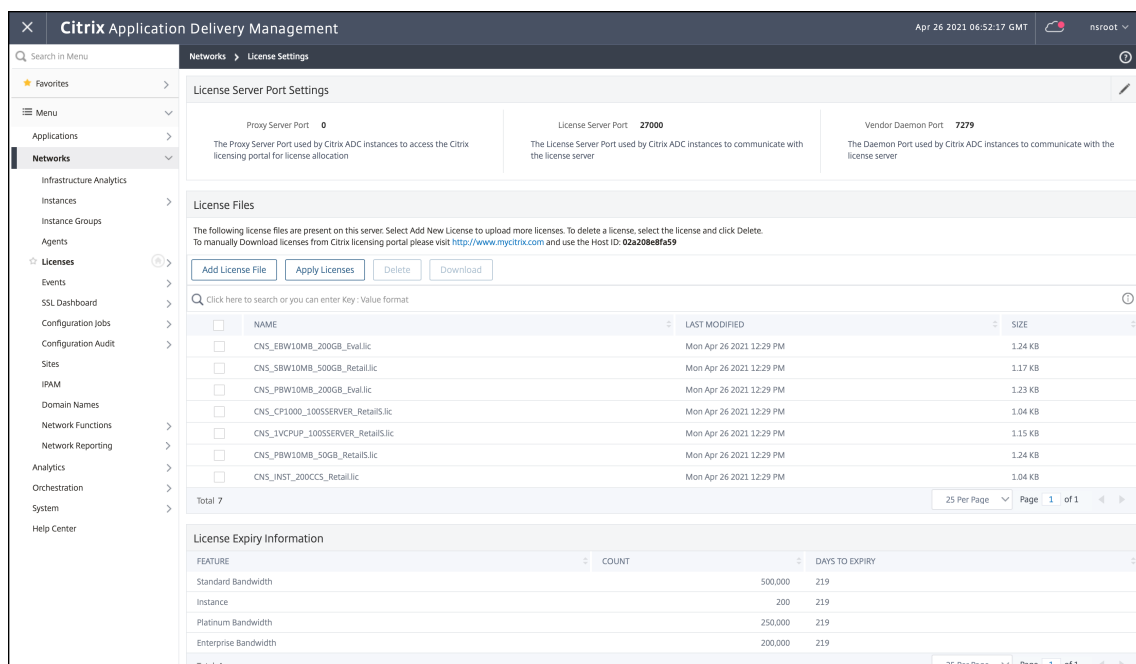


2. Ajoutez le pool de licences d'instance NetScaler ADC à ADM local.

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 sur Citrix ADM, consultez la section [Système 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, CP1000 est utilisé comme pool de licences vCPU.



- 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 de Citrix ADM local, spécifiez les variables d'environnement suivantes dans le fichier YAML :

- name: "LS_IP"
value: "10.105.158.144"// IP de l'instance locale d'ADM, si vous avez déployé l'agent ADM, il s'agit de l'adresse IP de votre agent comme décrit à l'étape 1
- name: "LS_PORT"
value: "27000"// port sur lequel le serveur de licences ADM écoute
- name: "BANDWIDTH"
value: "3000"//la capacité en Mbps que vous voulez allouer à CPX

Pour les licences basées sur VCPU de Citrix ADM local, spécifiez les variables d'environnement suivantes dans le fichier YAML :

- name: "LS_IP"
value: "10.105.158.144"// IP de l'instance locale d'ADM, si vous avez déployé l'agent ADM, il s'agira de votre adresse IP d'agent comme décrit à l'étape 1
- name: "LS_PORT"
value: "27000"// port sur lequel le serveur de licences ADM écoute
- name: "CPX_CORES"
value: "4"// le nombre de cœurs que vous souhaitez allouer

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-native-getting-started/master/cpx-licensing/manifest/cpx-bandwidth-license-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

November 23, 2023

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.

Conditions préalables

Assurez-vous que :

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

Remarque : Pour améliorer les performances de 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 la [documentation 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.

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:tag
```

Dans cette commande, la balise spécifie l'image Citrix NetScaler CPX.

Par exemple, si vous souhaitez télécharger la version 12.1-51.16, utilisez la commande suivante :

```
1 docker pull quay.io/citrix/citrix-k8s-cpx-ingress:12.1-51.16
```


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

```
1 root@ubuntu:~# docker images | grep 'citrix-k8s-cpx-ingress'
2 quay.io/citrix/citrix-k8s-cpx-ingress 12.1-51.16
952a04e73101 2 months ago 469 MB
```

Remarque : Il est recommandé d'utiliser la dernière image NetScaler CPX du registre de conteneurs Quay pour bénéficier des dernières fonctionnalités de NetScaler CPX.

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.

Important

:

Si vous avez téléchargé NetScaler CPX Express depuis <https://www.citrix.com/products/netscaler-adc/cpx-express.html>, assurez-vous de lire et de comprendre le contrat de licence utilisateur final (CLUF) disponible sur :<https://www.citrix.com/products/netscaler-adc/cpx-express.html> et acceptez le CLUF lors du déploiement de l'instance NetScaler CPX.

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

```
1 docker run -dt -P --privileged=true --net=host -e NS_NETMODE="HOST" \
  -e CPX_CORES=<number of cores> --name <container_name> --ulimit core \
  =-1 -e CPX_NW_DEV='<INTERFACES>' -e CPX_CONFIG=' {
2   "YIELD" : "NO" }
3   ' -e LS_IP=<LS_IP_ADDRESS> -e LS_PORT=<LS_PORT> -e PLATFORM=CP1000 -v
  <host_dir>:/cpx <REPOSITORY>:<TAG>
4 <!--NeedCopy-->
```

```
1 docker run -dt --privileged=true --net=host -e NS_NETMODE="HOST" -e
  CPX_NW_DEV='eth1 eth2' -e CPX_CORES=5 -e CPX_CONFIG='{
2   "YIELD": "No" }
3   ' -e LS_IP=10.102.38.134 -e PLATFORM=CP1000 -v /var/cpx:/cpx --name
  cpx_host cpx:12.1-48.xx
4 <!--NeedCopy-->
```

Cet exemple crée un conteneur nommé mycpx sur la base 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 que les ports 9080, 22, 9443 et 161/UDP sont mappés aux ports de l'hôte Docker sélectionnés aléatoirement 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 avec plusieurs cœurs, vous devez fournir tous les privilèges système au NetScaler CPX. Si vous souhaitez exécuter NetScaler CPX avec un seul cœur, vous devez utiliser cette option au lieu de cette option. `--cap-add=NET_ADMIN` Cette `--cap-add=NET_ADMIN` option vous permet d'exécuter le conteneur NetScaler CPX avec des privilèges réseau complets.

****`--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

Si vous exécutez NetScaler CPX sur un pont ou sur un réseau sans réseau, ignorez cette option.

`-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 quatre règles iptable par défaut sur la 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

Si vous exécutez NetScaler CPX en mode pont ou sans réseau, ignorez cette variable d'environnement.

–e `CPX_CORES` Il 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.

Remarque

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.

–e `EULA = yes` est 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 (EULA) disponible à l'adresse : <https://www.citrix.com/products/netscaler-adc/cpx-express.html>

Le –e `PLATFORM=CP1000` paramètre 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:12.0-53.x  
2 <!--NeedCopy-->
```

–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'environnement `CPX_CONFIG` 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 docker run -dt --privileged=true --net=host -e NS_NETMODE="HOST" -e
  EULA=yes -e CPX_CORES=5 -e CPX_CONFIG='{
2   "YIELD": "No" }
3   ' -e PLATFORM=CP1000 --name cpx_host cpx:12.0-51.x
4 <!--NeedCopy-->
```

```
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:12.0-51.xx
4 <!--NeedCopy-->
```

Le `-v` paramètre 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`. 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.

```
1 docker run -dt --privileged=true --net=host -e NS_NETMODE="HOST" -e
  CPX_CORES=5 -e CPX_CONFIG='{
2   "YIELD": "No" }
3   ' -e LS_IP=10.102.38.134 -e PLATFORM=CP1000 --name cpx_host cpx
  :12.0-51.xx
4 <!--NeedCopy-->
```

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'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 Compose, vous devez d'abord écrire un fichier Compose. 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>
4  ports:
5      - 22
6      - 9080
7      - 9443
8      - 161/udp
9      - 35021-35030
10  tty: true
11  cap_add:
12      - NET_ADMIN
13  ulimits:
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     '
23 <!--NeedCopy-->
```

```
1     CPX_0:
2     container_name: CPX_0
3     image: cpx:12.0-53.xx
4     ports:
5         - 9443
6         - 22
7         - 9080
8         - 161/udp
9         - 35021-35030
10    tty: true
11    cap_add:
12        - NET_ADMIN
13    ulimits:
14        core: -1
15    volumes:
16        - /root/test:/cpx
17    environment:
18        - CPX_CORES=2
19        - EULA=yes
20 <!--NeedCopy-->
```

Si vous souhaitez provisionner une seule instance NetScaler CPX, vous devez ajouter la ligne suivante au fichier de composition : `container_name:<name_of_container>`

Exécutez la commande suivante pour provisionner plusieurs instances NetScaler CPX :

```
docker-compose -f <compose_file_name> scale <service-name>=<number of
instances> up -d
docker-compose -f docker-compose.yml scale cpxlb=3
up -d
```

Si vous souhaitez provisionner une seule instance NetScaler CPX, exécutez la commande suivante :

```
docker-compose -f <compose_file_name> up -d
```

Ajouter des instances NetScaler CPX à Citrix ADM

November 23, 2023

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 de la ou des instances que vous souhaitez ajouter à Citrix ADM. Pour chaque type d'instance, un profil par défaut est disponible. Par exemple, le profil `ns-root-profile` est le profil par défaut pour les instances NetScaler ADC. 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'une instance après la découverte de l'instance, vous devez modifier le profil d'instance ou créer un nouveau profil, puis redécouvrir l'instance.

Conditions préalables

Assurez-vous d'avoir :

- Installez le logiciel Citrix ADM sur Citrix XenServer - effectué. 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, tapez 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 > Citrix ADC** 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 clé-valeur vous permet de rechercher plus tard l'instance.

← Add Citrix ADC CPX

☒ Enter Device IP Address ☐ Import from file

Enter one or more hostnames, IP addresses , and/or a range of IP addresses (for example, 10.102.40.30-10.102.40.45) using a comma separator.

Routable IP/ Docker IP*

?

Profile Name*

Site*

Agent

>

Tags

+

6. Cliquez sur **OK**.

Remarque

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

Ajouter des 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.)

Voici un exemple de `docker run` commande pour 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
2
3  <!--NeedCopy-->
```

Configuration de NetScaler CPX

November 23, 2023

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@10:~# cli_script.sh "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**.

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>`

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>\`

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 la tâche.

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

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

Select Configuration

Select Instances

Specify Variable Values

Job Name*

cpx-single-host

Instance Type*

NetScaler

Configuration Editor

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

LBVariablesTemplate

1 SSH

2 SSH

3 SSH

4 SSH

5 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

bind lb vserver cpx-vip db2

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

Select Configuration

Select Instances

Specify Variable Values

Select the target instances on which you want to run the configuration.

Add Instances

Delete

☒

IP Address

Name

☐

172.17.0.150

10.102.31.190

Cancel

Back

Next

Save and Exit

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**.

The screenshot shows the 'Execute' step of a configuration process. At the top, there are four tabs: 'Select Configuration', 'Select Instances', 'Specify Variable Values', and 'Execute' (which is active). Below the tabs, a message states: 'You can either run the commands now or schedule to run the commands at a later time.' The main configuration area includes:

- On Command Failure*:** A dropdown menu set to 'Ignore error and continue'.
- Execution Mode*:** A dropdown menu set to 'Now'.
- Execution Settings:**
 - ☒ Execute in Sequence
 - ☐ Execute in Parallel
- Receive Execution Report Through:**
 - ☐ Email

At the bottom, there are four buttons: 'Cancel', 'Back', 'Finish' (highlighted in blue), and 'Save and Exit'.

Configuration de la diffusion de journaux sur une instance NetScaler CPX

Vous pouvez configurer le streaming des journaux sur une instance NetScaler CPX pour collecter des données de performance des pages Web, des informations sur le flux et le niveau de session utilisateur, ainsi que des 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 les rapports historiques et en temps réel pour toutes vos applications.

Pour configurer la diffusion des journaux sur une instance NetScaler CPX, vous devez d'abord activer la fonctionnalité AppFlow et le `ulfd` démon sur l'instance NetScaler CPX. Lorsque vous activez le `ulfd` démon, vous devez spécifier l'adresse IP de Citrix ADM sur laquelle vous souhaitez surveiller les rapports en temps réel et historiques. Vous devez ensuite configurer un collecteur, une action, une stratégie AppFlow et lier la stratégie AppFlow globalement.

Le démon `ulfd` envoie tous les enregistrements de flux à Citrix ADM dans un format de journalisation unifié unique.

Vous pouvez configurer la diffusion des journaux soit à l'aide de l'interface de ligne de commande d'une instance NetScaler CPX, soit à l'aide de la fonctionnalité Jobs de Citrix ADM.

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 Citrix ADM.

Pour configurer la diffusion en continu des journaux sur une instance NetScaler CPX :

1. Activez la fonctionnalité AppFlow en exécutant la commande suivante : `enable ns feature AppFlow`
2. Activez le démon ulfd à l'aide de la commande suivante : `set ns param -ulfd ENABLED -loggerip <NUMS_IP_Address>`
3. Exécutez les commandes suivantes pour configurer un collecteur, une action et une stratégie AppFlow, et pour lier la stratégie globalement :

```

1  add appflow collector <name> -IPAddress <ipaddress>`
2
3  set appflow param -templateRefresh 3600 -httpUrl ENABLED -
    httpCookie ENABLED -httpReferer ENABLED -httpMethod ENABLED -
    httpHost ENABLED -httpUserAgent ENABLED -httpContentType ENABLED
    -httpAuthorization ENABLED -httpVia ENABLED -httpXForwardedFor
    ENABLED -httpLocation ENABLED -httpSetCookie ENABLED -
    httpSetCookie2 ENABLED -connectionChaining ENABLED -httpDomain
    Enabled
4
5  add appflow action <name> --collectors<string> ... [-
    clientSideMeasurements (Enabled|Disabled) ]
6
7  add appflow policy <name> true <action>
8
9  bind appflow global <policyName> <priority> [<
    gotoPriorityExpression [-type <type>]
10 <!--NeedCopy-->
```

Remarque :

Vous devez configurer le collecteur AppFlow avec une adresse IP fictive.

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 NetScaler Management and Analytics (MAS) 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 ci-dessus, 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
7 bind lb vserver v1 s1
8
9 #Shell Commands
10
11 touch /etc/a.txt
12
13 echo "this is a" > /etc/a.txt
14
15 #NetScaler Commands
16
17 add lb vserver v2 http
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
28 <!--NeedCopy-->
```

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/netscalercpx
  :12.1-48.13
2 <!--NeedCopy-->
```

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 de Docker, reportez-vous à <https://docs.docker.com/config/containers/logging/configure/>.

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/netscalercpx:12.1-48.13
2 <!--NeedCopy-->
```

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/netscalercpx:12.1-48.13  
2 <!--NeedCopy-->
```

Support du routage dynamique dans NetScaler CPX

November 23, 2023

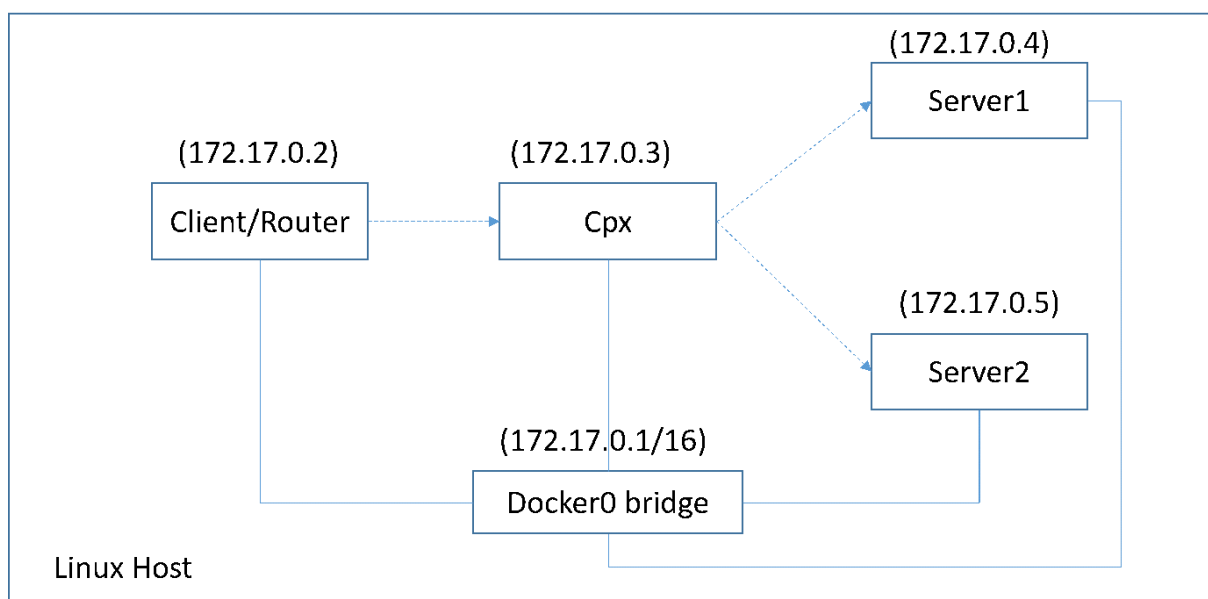
NetScaler CPX prend en charge le protocole de routage dynamique BGP. L'objectif principal du protocole de routage dynamique est d'annoncer l'adresse IP du serveur virtuel en fonction de l'état des services liés au serveur virtuel. Il aide un routeur en amont à choisir le meilleur itinéraire parmi plusieurs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

Remarque : Le protocole de routage dynamique BGP est pris en charge à partir de la version 52.15 de NetScaler ADC 12.1 et versions ultérieures.

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 :

```
1 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 vservers <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 :

```
1 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 :

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

Spécifiez la commande comme suit :

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

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

```
1 curl -u username:password http://<NSIP>:9080/nitro/v1/config/ --
   data 'object={
2   "params":{
3     "action":"apply" }
4   , "routerDynamicRouting": {
5     "commandstring" : "ns route-install bgp" }
6   }
7   '
```

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  {
3    "errorCode": 0, "message": "Done", "severity": "NONE", "
   routerDynamicRouting":{
4    "bgpRouter":[{
5      "localAS": 100, "routerId": "172.17.0.3", "afParams": [ {
6        "addressFamily": "ipv4" }
7    , {
8      "addressFamily": "ipv6" }
9    ], "neighbor": [ {
10     "address": "172.17.0.2", "remoteAS": 101, "ASOriginationInterval
        ": 15, "advertisementInterval": 30, "holdTimer": 90, "
        keepaliveTimer": 30, "state": "Connect", "singlehopBfd":
        false, "multihopBfd": false, "afParams": [ {
11      "addressFamily": "ipv4", "activate": true }
12    , {
13      "addressFamily": "ipv6", "activate": false }
14    ]

```

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`.

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:12.1-48.xx
2 <!--NeedCopy-->

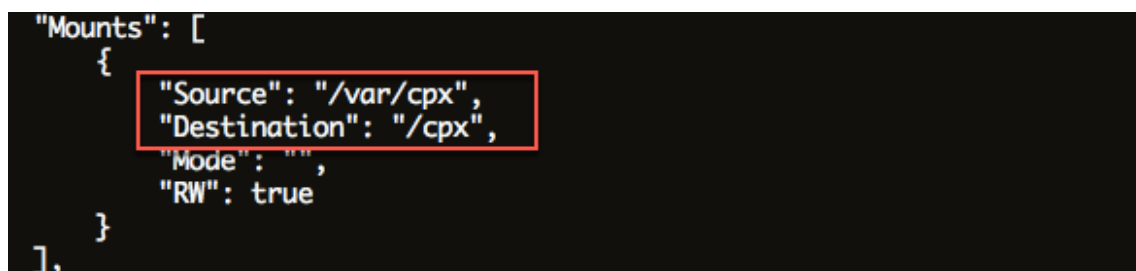
```

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.



```

"Mounts": [
  {
    "Source": "/var/cpx",
    "Destination": "/cpx",
    "Mode": "",
    "RW": true
  }
],

```

- Téléchargez le dernier fichier image Docker NetScaler CPX à partir de, <https://www.microloadbalancer.com/get-it-now> puis chargez l'image Docker NetScaler CPX. 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-12.0-41.10.gz
2
3 root@ubuntu:~# docker images
4
5 REPOSITORY TAG IMAGE ID CREATED VIRTUAL SIZE
6
7 cpx 12.0-41.10 2e97aadf918b 43 hours ago 414.5 MB
8 <!--NeedCopy-->

```

Pour mettre à niveau une instance NetScaler CPX

- 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
3 <!--NeedCopy-->

```

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:12.0-41.10
2 <!--NeedCopy-->

```

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

```

1 root@ubuntu:~# docker ps
2
3 CONTAINER ID          IMAGE               COMMAND             PORTS
4 ead12ec4e965          cpx:12.0-41.10     "/bin/sh -c 'bash -C " 5
5 seconds ago          Up 5 seconds       22/tcp, 80/tcp, 443/
tcp, 161/udp          latestcpx
6 <!--NeedCopy-->

```

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
3 <!--NeedCopy-->

```

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, le NSIP étant attribué à l'instance NetScaler CPX par le moteur Docker.

Dans les déploiements NetScaler CPX haute disponibilité, vous pouvez ajouter des serveurs virtuels génériques sur l'une des instances NetScaler CPX et copier le fichier **ns.conf** de l'instance vers d'autres instances NetScaler CPX du déploiement et vous assurer que la configuration NetScaler est cohérente sur tous les Instances NetScaler CPX dans le déploiement au lieu d'identifier l'adresse IP unique attribuée par le moteur Docker à l'instance NetScaler et de créer les serveurs virtuels LB ou CS en fonction de l'adresse IP unique sur toutes les instances NetScaler CPX du déploiement.

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
4 <!--NeedCopy-->
```

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

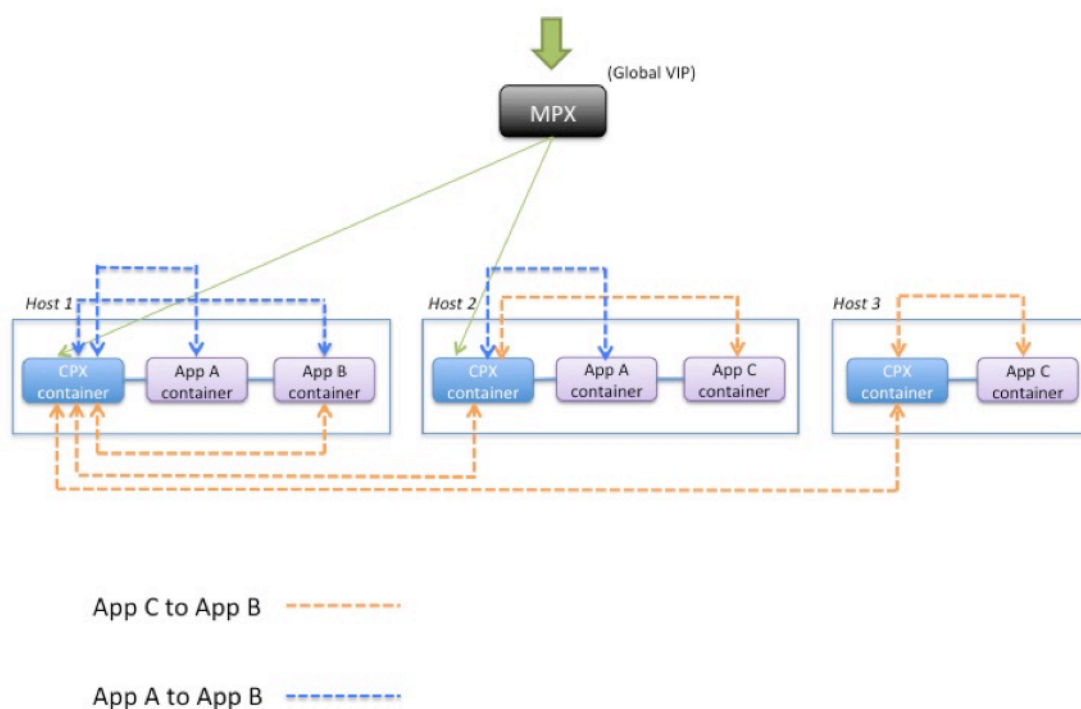
```
1      add cs vserver <name> <serviceType> 127.0.0.1 <port>
2
3      add cs vserver testcsvserver HTTP 127.0.0.1 30000
4 <!--NeedCopy-->
```

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.

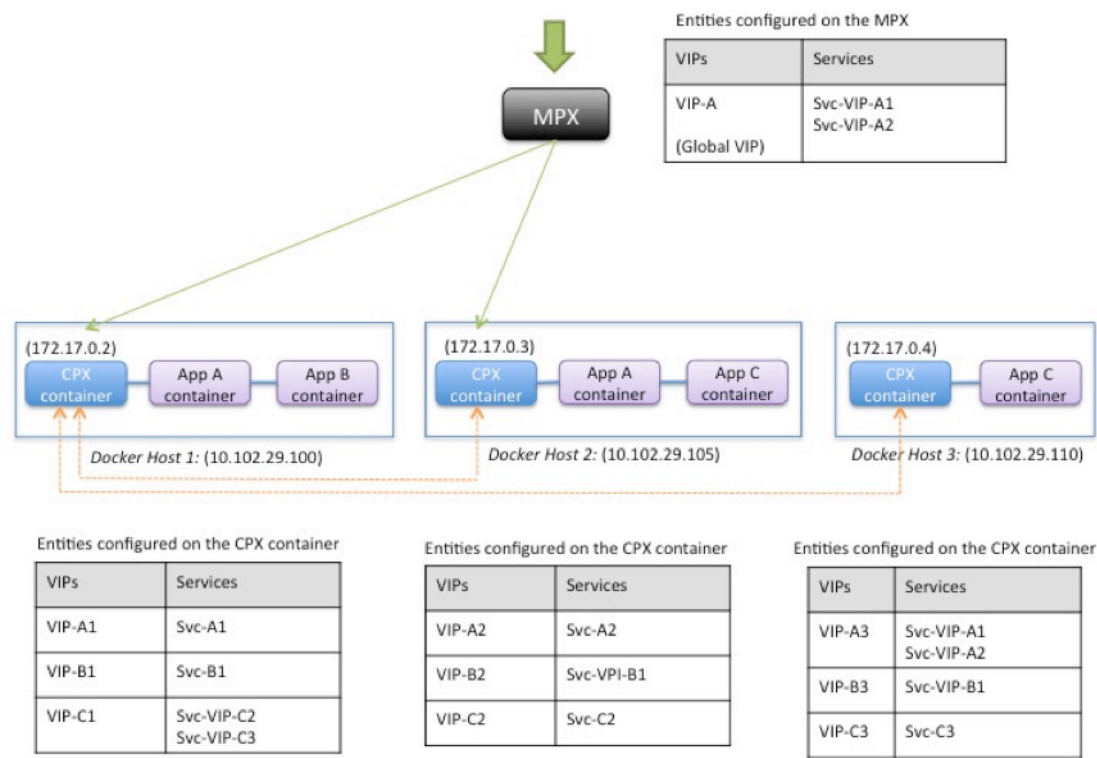
L'image suivante illustre le flux de trafic entre deux applications via les instances NetScaler CPX.



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.



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 Bound to the VIP	VIPs	Services Bound to the VIP	VIPs	Services Bound to the VIP
VIP-A1 172.17.0.2:30000	SVC-A1 10.102.29.100:80	VIP-A2 172.17.0.3:30000	SVC-A2 10.102.29.105:80	VIP-A3 172.17.0.4:30000	SVC-VIP-A1 10.102.29.100:30000
					SVC-VIP-A2 10.102.29.105:30000
VIP-B1 172.17.0.2:30001	SVC-B1 10.102.29.100:90	VIP-B2 172.17.0.3:30001	SVC-VIP-B1 10.102.29.100:30001	VIP-B3 172.17.0.4:30001	SVC-VIP-B1 10.102.29.100:30001
VIP-C1 172.17.0.2:30002	SVC-VIP-C2 10.102.29.105:30002	VIP-C2 172.17.0.3:70	SVC-C2 10.102.29.105:70	VIP-C3 172.17.0.4:70	SVC-C3 10.102.29.110:70
	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=true cpx:6.2
2 <!--NeedCopy-->
```

Exécutez les commandes suivantes soit à l'aide de la fonctionnalité Tâches de NetScaler MAS, soit à l'aide des API NITRO.

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

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

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
10 <!--NeedCopy-->
```

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

```
1 add lb vserver VIP-A3 HTTP 172.17.0.4 30000
2 add service svc-VIP-A1 10.102.29.100 HTTP 30000
3 add service svc-VIP-A2 10.102.29.105 HTTP 30000
4 bind lb vserver VIP-A3 svc-VIP-A1
5 bind lb vserver VIP-A3 svc-VIP-A2
6 add lb vserver VIP-B3 HTTP 172.17.0.4 30001
7 add service svc-VIP-B1 10.102.29.100 HTTP 30001
8 bind lb vserver VIP-B3 svc-VIP-B1
9 add lb vserver VIP-C3 HTTP 172.17.0.4 30002
10 add service svc-C3 10.102.29.110 HTTP 70
11 bind lb vserver VIP-C3 svc-C3
12 <!--NeedCopy-->
```

Déployer NetScaler CPX sur un réseau hôte unique

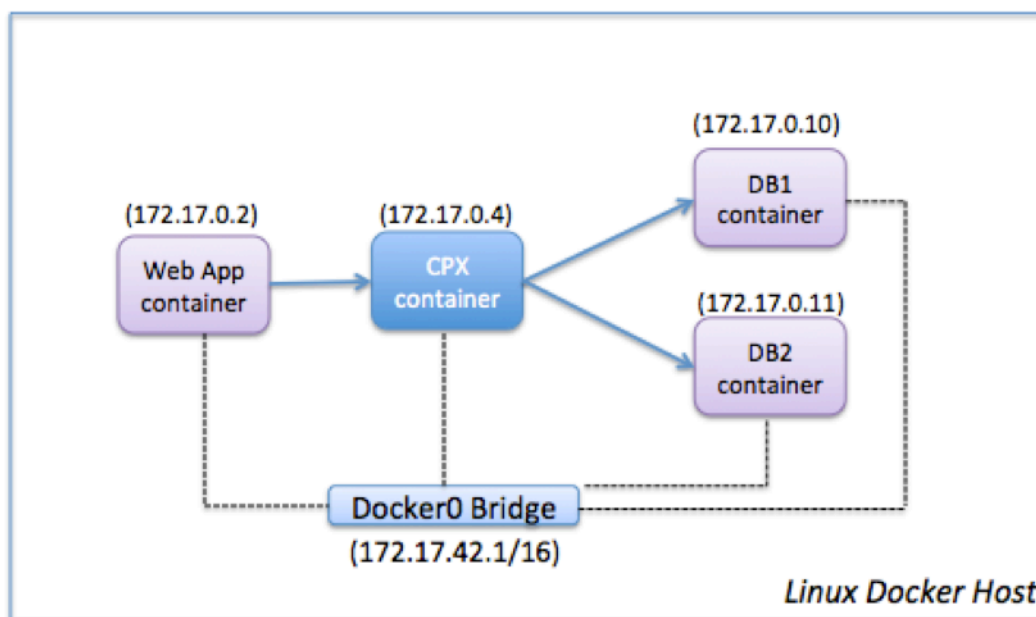
November 23, 2023

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.

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 soit à l'aide de la fonctionnalité Tâches de NetScaler MAS, soit à l'aide des API NITRO :

```
1 add service db1 HTTP 172.17.0.10 80
2 add service db2 HTTP 172.17.0.11 80
3 add lb vserver cpx-vip HTTP 172.17.0.4 81
4 bind lb vserver cpx-vip db1
5 bind lb vserver cpx-vip db2
6 <!--NeedCopy-->
```

Déployer NetScaler CPX dans un réseau multi-hôtes

November 23, 2023

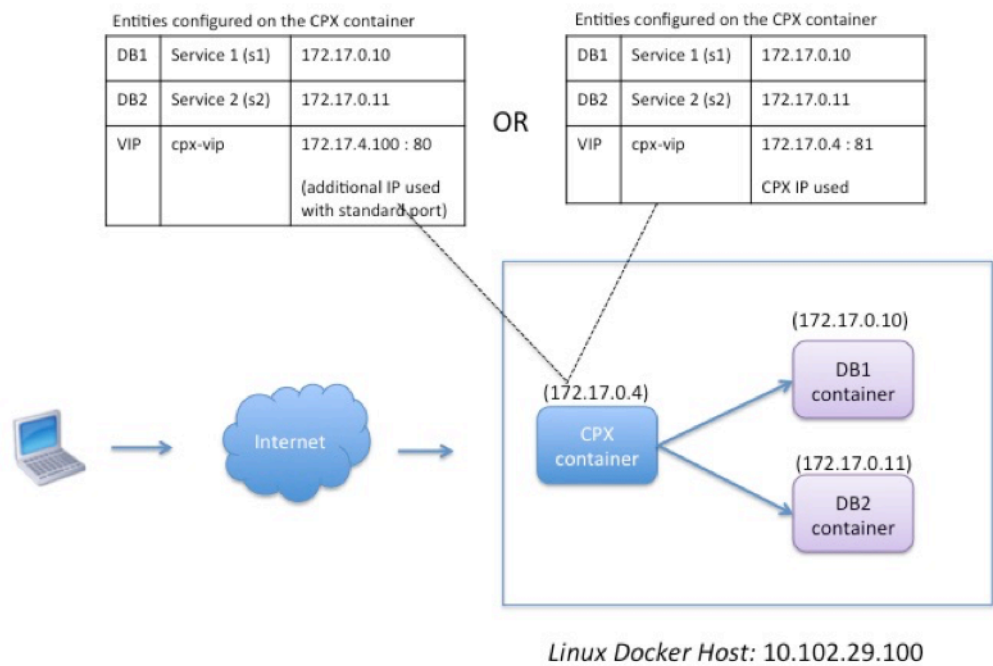
Une instance NetScaler CPX dans un réseau multi-hôtes peut être configurée dans un déploiement de production dans le centre de données où elle fournit des fonctions d'équilibrage de charge. Il peut également fournir des fonctions de surveillance et des données analytiques.

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.



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 Tâches de NetScaler MAS ou 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
6 <!--NeedCopy-->

```

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 :

```

1      ip addr add 10.102.29.103/24 dev eth0
2      iptables -t nat -A PREROUTING -p ip -d 10.102.29.103 -j DNAT --to-
      destination 172.17.4.100
3 <!--NeedCopy-->

```

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
2
3 <!--NeedCopy-->

```

Une fois l'instance NetScaler CPX provisionnée, exécutez les commandes suivantes à l'aide de la fonctionnalité Tâches de NetScaler MAS ou 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 172.17.0.4 81
4      bind lb vserver cpx-vip s1
5      bind lb vserver cpx-vip s2
6 <!--NeedCopy-->

```

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 :

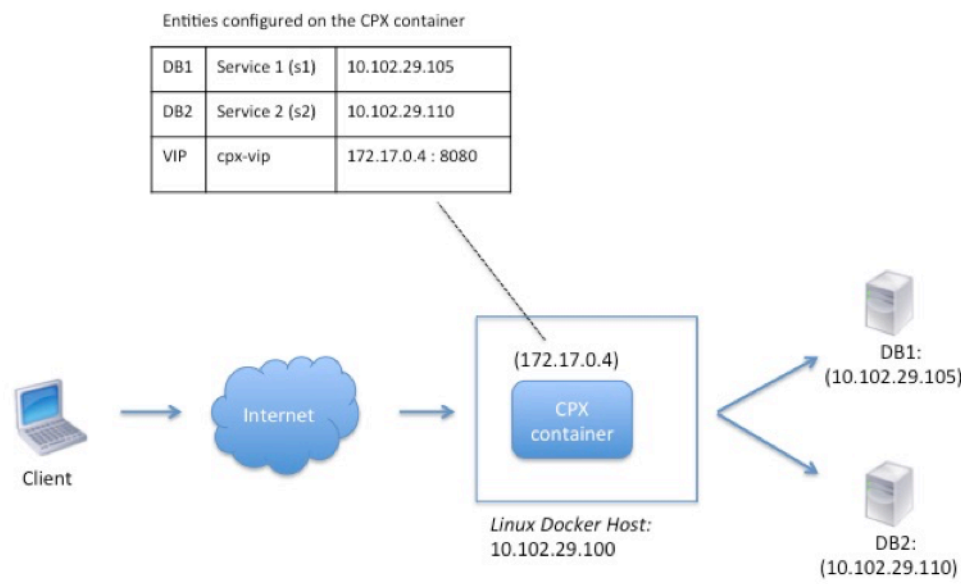
```
iptables -t nat -A PREROUTING -p tcp -m addrtype --dst-type LOCAL -m tcp --dport 50000 -j DNAT --to-destination 172.17.0.4:81
```

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 :

```
1     docker run -dt -p 22 -p 80 -p 161/udp -p 8080:8080 --ulimit core=-1
      --privileged=true cpx:6.2
2 <!--NeedCopy-->
```

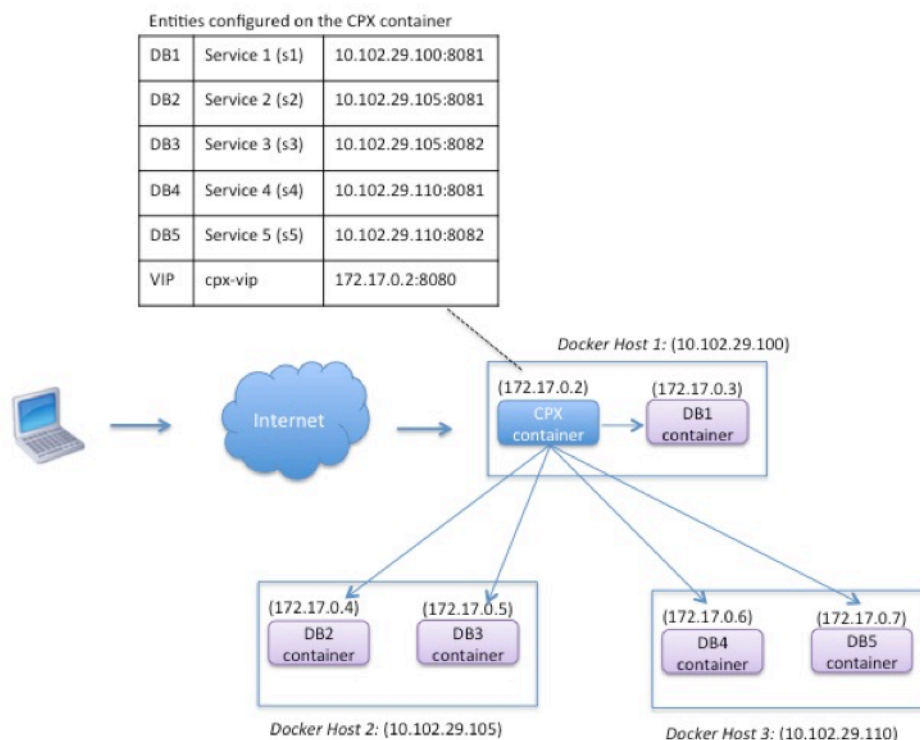
Ensuite, exécutez les commandes suivantes à l'aide de la fonctionnalité Tâches de NetScaler MAS ou des API NITRO :

```
1     add service s1 HTTP 10.102.29.105 80
2     add service s2 HTTP 10.102.29.110 80
3     add lb vserver cpx-vip HTTP 172.17.0.4 8080
4     bind lb vserver cpx-vip s1
5     bind lb vserver cpx-vip s2
6 <!--NeedCopy-->
```

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 :

1. 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 :

```
1      docker run -dt -p 22 -p 80 -p 161/udp -p 8080:8080 --ulimit core=-1
      --privileged=true cpx:6.2
2 <!--NeedCopy-->
```

Exécutez les commandes suivantes soit à l'aide de la fonctionnalité Tâches de NetScaler MAS, soit à l'aide des API NITRO :

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

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 :

```
1      ip addr add 10.102.29.100/24 dev docker0;
2      ip addr del 10.102.29.100/24 dev eth0;
3      brctl addif docker0 eth0;
4      ip route del default;
```

```

5 ip route add default via 10.102.29.1 dev docker0
6 <!--NeedCopy-->

```

En utilisant la fonctionnalité Tâches de NetScaler MAS ou en utilisant les API NITRO, exécutez les commandes suivantes :

```

1 add service s1 172.17.0.8 http 80
2 add service s2 172.17.0.9 http 80
3 add lb vserver cpx-vip HTTP 10.102.29.102 80
4 bind lb vserver cpx-vip s1
5 bind lb vserver cpx-vip s2
6 <!--NeedCopy-->

```

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
14 <!--NeedCopy-->

```

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
2 <!--NeedCopy-->

```

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
28 <!--NeedCopy-->

```

Vous pouvez spécifier la ConfigMap créée `cpx-config` dans le fichier YAML utilisé pour déployer l'instance NetScaler CPX, comme indiqué ci-dessous :

```

1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: cpx-1
5    labels:
6      app: cpx-daemon
7    annotations:
8      NETSCALER_AS_APP: "True"
9  spec:
10   hostNetwork: true
11   containers:
12   - name: cpx
13     image: "quay.io/citrix/citrix-k8s-cpx-ingress:12.1-51.16"
14     securityContext:
15       privileged: true
16     volumeMounts:
17     - name: config-volume
18       mountPath: /cpx/conf
19     env:
20     - name: "EULA"
21       value: "yes"
22     - name: "NS_NETMODE"
23       value: "HOST"
24     - name: "kubernetes_url"
25       value: "https://10.90.248.101:6443"
26     - name: "NS_MGMT_SERVER"
27       value: "10.90.248.99"
28     - name: "NS_MGMT_FINGER_PRINT"
29       value: "19:71:A3:36:85:0A:2B:62:24:65:0F:7E:72:CC:DC:AD:B8:BF
30         :53:1E"
31     - name: "NS_ROUTABLE"
32       value: "FALSE"
33     - name: "KUBERNETES_TASK_ID"
34       valueFrom:
35         fieldRef:
36           fieldPath: metadata.name
37     imagePullPolicy: Never
38   volumes:
39   - name: config-volume
40     configMap:
41       name: cpx-config
42 <!--NeedCopy-->

```

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.

Déployer le proxy NetScaler CPX sur Google Compute Engine

November 23, 2023

Ce guide de déploiement explique comment déployer NetScaler CPX avec Docker sur Google Compute Engine (GCE) de Google Cloud avec NetScaler MAS s'exécutant sur le réseau de l'entreprise. Dans ce déploiement, NetScaler CPX installé sur GCE équilibre la charge de deux serveurs principaux, et NetScaler MAS 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. NetScaler MAS fournit des solutions de gestion, de licence et d'analyse. En tant que serveur de licences, NetScaler MAS 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. Lorsqu'elle est concédée sous licence par NetScaler MAS et installée, 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.

Conditions préalables

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

Remarque

Pour plus d'informations sur la façon de déployer NetScaler MAS, consultez [Déploiement de NetScaler MAS](#).

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 NetScaler CPX Image.
4. Créez une instance CPX.
5. Accordez une licence NetScaler CPX via NetScaler MAS.
6. Configurez les services d'équilibrage de charge sur NetScaler CPX et vérifiez la configuration.
 - a) Installez les serveurs Web NGINX.
 - b) Configurez CPX pour l'équilibrage de charge et vérifiez la distribution de la charge sur 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 $ sudo curl -ssl https://get.docker.com/ | sh
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: xn--ssl-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 + echo -----BEGIN PGP PUBLIC KEY BLOCK-----
9 Version: GnuPG v1
10
11 mQINBFWln24BEADrBl5p99uKh8+rpvqJ48u4eTtjeXAWbslJotmC/CakbNSq0b9o
12 ddfzRvGVeJVERT/Q/mlvEqgnyTQy+e6oEYN2Y2kqXceUhXagThnqCoxcEJ3+KM4R
13 mYdoe/BJ/J/6rH0jq70mk24z2qB3RU1uAv57iY5VGw5p45uZB4C4pNNsBJXoCvPn
14 TGAs/7IrekFZDDgVraPx/hdiwopQ8NltSfZCyU/jPpWFK28TR8yfVlzYFwibj5WK
15 dHM7ZTqlA1tHIG+agyPf3Rae0jPMsHR6q+arXVwMccy0i+ULU0z8mHUJ3iEMirpT
16 X+80KaN/ZjibfsB0CjcfiJSB/acn4nxQQgNZigna32velafhQivsNREFeJpzENiG
17 HOoyC6qVe0gKrRiKxzymj0FIMLru/iFF5pSWcQB7PYlt8J0G80lAcPr6VCiN+4c
18 NKv03SdvA69dC0j79Pu09IIvQsJXsSq96HB+TeEmmL+xSdpGtGdCJHm1fDeCqkZ
19 hT+RtBGQL2SEdWjxbF43oQopocT8cHvyX6Zaltn0svoGs+wX3Z/H6/8P5anog43U
20 65c0A+64Jj00rNDR8j31izhtQMRo892kGeQAaaxg4Pz6HnS7hRC+cOMHUU4HA7iM
21 zHrouAdYeTZeZEQA7SxtCME9ZnGwe2grxPXh/U/80WJGkzLFNcTKdv+rwARAQAB
22 tDdEb2NrZXIgmUmVsZWZzZSBub29sIChyZWxlYXNlZG9ja2VyKSA8ZG9ja2VyQGRv
23 Y2tldi5jb20+iQIcBBABCgAGBQJWw7vdAAoJEFyzYeVS+w0QHysP/i37m4Syo0CV
24 cnybl18vzwBEcp4VCRbXvHvOXtylgccVIV8/aJqNKgBV97LY3vrp0yiIeB8ETQeg
25 srxFE7t/Gz0rsL0bqfLEHdmn5iBJRkhLfCpzje0nyB3Z0IJB6Uog0/msQVYe5CXJ
26 l6uwr0AmoiCBLrVlDAktxVh9RWch0l0KZR2FpHu8h+uM0/zySqIdlyfLa3y5oH
27 scU+nGU1i6ImwDTD3ysZC5jp9aVfvUmcESyAb4vvdCAHR+bXhA/RW8QHeeMFLiWw
28 7Z2jYHyyHmDnWG2yUrnCqAJTTrWV+OfKRIzzJFBs4e88ru5h2ZIXdRepw/+COYj34
29 LyzXR2cxr2u/xvxwXcKsMe7F4KZaphD+1ws61FhnUMi/PERMYfTFuvPrCkq4gyBj
30 t3fFpZ2NR/fKW87Q0eVcn1ivXl9id3MMs9KXJsg7QasT7mCsee2VIFsrxkFQ2jNp
31 D+JAERRn9Fj4ArHL5TbwkkFbZZvSi6fr5h2GbCAXIGhIXKnjjorPY/YDX6X8AaH0
32 W1zblWy/CFr6VFL963jrjJgag0G6tNtBZLrclZgWh0QpeZZ5Lbvz2ZA5CqRrFAVc

```

```
33 wPNW1f0bFIRtqV6vuVluFOPCMAAnOnqR02w9t17iVQj03oVN0mbQi9vjuExXh1Yo
34 ScVeti06LSmlQfVEVRTqHLMgXyR/EMo7iQIcBBABCgAGBQJXSWBLAAoJEFyzYeVS
35 +w0QeH0QAI6btAfYwYPuAjfRUy9qlnPhZ+xt1rnwsUzsbmo8K3XTNh+l/R08nu0d
36 sczw30Q1wju28fh1N8ay223+69f0+yICaXqR18AbGgFGKX7vo0gfEVaxdItUN3eH
37 NydGFzmeOKbAlrxIMECnSTG/TkFVY09Ntlv9vSN2BupmTagTRExLZKnVsWRzp+X
38
39 -----END PGP PUBLIC KEY BLOCK-----
40
41 OK
42 + sudo -E sh -c mkdir -p /etc/apt/sources.list.d
43 + dpkg --print-architecture
44 + sudo -E sh -c echo deb \[arch=amd64\] https://apt.dockerproject.org
    /repo ubuntu-yakkety main > /etc/apt/sources.list.d/docker.list
45 + sudo -E sh -c sleep 3; apt-get update; apt-get install -y -q docker-
    engine
46 Hit:1 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety InRelease
47 Get:2 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates
    InRelease [102 kB]
48 Get:3 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-backports
    InRelease [102 kB]
49 Get:4 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/restricted
    Sources [5,376 B]
50 Get:5 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/multiverse
    Sources [181 kB]
51 Get:6 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/universe
    Sources [8,044 kB]
52 Get:7 http://archive.canonical.com/ubuntu yakkety InRelease [11.5 kB]
53 Get:8 http://security.ubuntu.com/ubuntu yakkety-security InRelease [102
    kB]
54 Get:9 https://apt.dockerproject.org/repo ubuntu-yakkety InRelease [47.3
    kB]
55 Get:10 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/main
    Sources [903 kB]
56 Get:11 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
    restricted Sources [2,688 B]
57 Get:12 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
    universe Sources [57.9 kB]
58 Get:13 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
    multiverse Sources [3,172 B]
59 Get:14 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
    main Sources [107 kB]
60 Get:15 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
    main amd64 Packages [268 kB]
61 Get:16 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
    main Translation-en [122 kB]
62 Get:17 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
    universe amd64 Packages [164 kB]
63 Get:18 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
    universe Translation-en [92.4 kB]
64 Get:19 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
    multiverse amd64 Packages [4,840 B]
65 Get:20 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety-updates/
    multiverse Translation-en [2,708 B]
```

```
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 Sources  
    [47.7 kB]  
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  
    Sources [20.7 kB]  
75 Get:30 http://security.ubuntu.com/ubuntu yakkety-security/multiverse  
    Sources [1,140 B]  
76 Get:31 http://security.ubuntu.com/ubuntu yakkety-security/restricted  
    Sources [2,292 B]  
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  
    Translation-en [47.3 kB]  
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.  
92 Need to get 21.2 MB of archives.  
93 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]  
95 Get:2 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/universe  
    amd64 cgroupfs-mount all 1.3 [5,778 B]  
96 Get:3 http://us-west1.gce.archive.ubuntu.com/ubuntu yakkety/main amd64  
    libltdl7 amd64 2.4.6-1 [38.6 kB]  
97 Get:4 https://apt.dockerproject.org/repo ubuntu-yakkety/main amd64  
    docker-engine amd64 17.05.0~ce-0~ubuntu-yakkety [21.1 MB]  
98 Fetched 21.2 MB in 1s (19.8 MB/s)
```



```
99  Selecting previously unselected package aufs-tools.
100  (Reading database ... 63593 files and directories currently installed.)
101  Preparing to unpack .../aufs-tools_1%3a3.2+20130722-1.1ubuntu1_amd64.
    deb ...
102  Unpacking aufs-tools (1:3.2+20130722-1.1ubuntu1) ...
103  Selecting previously unselected package cgroupfs-mount.
104  Preparing to unpack .../cgroupfs-mount_1.3_all.deb ...
105  Unpacking cgroupfs-mount (1.3) ...
106  Selecting previously unselected package libltdl7:amd64.
107  Preparing to unpack .../libltdl7_2.4.6-1_amd64.deb ...
108  Unpacking libltdl7:amd64 (2.4.6-1) ...
109  Selecting previously unselected package docker-engine.
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-yakkety) ...
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) ...
116  Processing triggers for systemd (231-9ubuntu4) ...
117  Setting up libltdl7:amd64 (2.4.6-1) ...
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) ...
125  + sudo -E sh -c docker version
126  Client:
127  Version: 17.05.0-ce
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
136  API version: 1.29 (minimum version 1.12)
137  Go version: go1.7.5
138  Git commit: 89658be
139  Built: Thu May 4 22:15:36 2017
140  OS/Arch: linux/amd64
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
146  sudo usermod -aG docker albert_lee
```

```
147
148 Remember that you will have to log out and back in for this to take
    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/security/#docker-
    daemon-attack-surface
154 for more information.
155
156 $
157
158 \*\*$ sudo docker info\*\*
159 Containers: 0
160 Running: 0
161 Paused: 0
162 Stopped: 0
163 Images: 0
164 Server Version: 17.05.0-ce
165 Storage Driver: aufs
166 Root Dir: /var/lib/docker/aufs
167 Backing Filesystem: extfs
168 Dirs: 0
169 Dirperm1 Supported: true
170 Logging Driver: json-file
171 Cgroup Driver: cgroupfs
172 Plugins:
173 Volume: local
174 Network: bridge host macvlan null overlay
175 Swarm: inactive
176 Runtimes: runc
177 Default Runtime: runc
178 Init Binary: docker-init
179 containerd version: 9048e5e50717ea4497b757314bad98ea3763c145
180 runc version: 9c2d8d184e5da67c95d601382adf14862e4f2228
181 init version: 949e6fa
182 Security Options:
183 apparmor
184 seccomp
185 Profile: default
186 Kernel Version: 4.8.0-51-generic
187 Operating System: Ubuntu 16.10
188 OSType: linux
189 Architecture: x86_64
190 CPUs: 1
191 Total Memory: 3.613GiB
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
196 Debug Mode (server): false
```

```

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
203 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 $
213 <!--NeedCopy-->

```

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. Cela est nécessaire pour que NetScaler MAS puisse communiquer avec l'instance Docker.

```

1 $
2 $ \*\*cd /etc/systemd/system\*\*
3 $ \*\*sudo vi docker-tcp.socket\*\*
4 $
5
6 $ \*\*cat docker-tcp.socket\*\*
7 [Unit]
8 \*\*Description=Docker Socket for the API
9 [Socket]
10 ListenStream=4243
11 BindIPv6Only=both
12 Service=docker.service
13 [Install]
14 WantedBy=sockets.target\*\*
15 $
16
17 $ \*\*sudo systemctl enable docker-tcp.socket\*\*
18 Created symlink /etc/systemd/system/sockets.target.wants/docker-tcp.
   socket → /etc/systemd/system/docker-tcp.socket.
19 $ \*\*sudo systemctl enable docker.socket\*\*
20 $ \*\*sudo systemctl stop docker\*\*
21 $ \*\*sudo systemctl start docker-tcp.socket\*\*
22 $ \*\*sudo systemctl start docker\*\*
23 $ \*\*sudo systemctl status docker\*\*
24 • docker.service - Docker Application Container Engine
25 Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor
   preset: enabled)
26 Active: \*\*active (running)\*\* since Wed 2017-05-31 12:52:17 UTC; 2s

```

```
ago
27 Docs: https://docs.docker.com
28 Main PID: 4133 (dockerd)
29 Tasks: 16 (limit: 4915)
30 Memory: 30.1M
31 CPU: 184ms
32 CGroup: /system.slice/docker.service
33 └─4133 /usr/bin/dockerd -H fd://
34 └─4137 docker-containerd -l unix:///var/run/docker/libcontainerd/docker
    -containerd.sock --metrics-interval=0 --start-timeout 2m -
35
36 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
37 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
38 May 31 12:52:17 docker-7 dockerd[4133]: time="2017-05-31T12
    :52:17.301681794Z" level=info msg="Loading containers: start."
39 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
40 May 31 12:52:17 docker-7 dockerd[4133]: time="2017-05-31T12
    :52:17.465011600Z" level=info msg="Loading containers: done."
41 May 31 12:52:17 docker-7 dockerd[4133]: time="2017-05-31T12
    :52:17.484747909Z" level=info msg="Daemon has completed
    initialization"
42 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
43 May 31 12:52:17 docker-7 systemd[1]: Started Docker Application
    Container Engine.
44 May 31 12:52:17 docker-7 dockerd[4133]: time="2017-05-31T12
    :52:17.503832254Z" level=info msg="API listen on /var/run/docker.
    sock"
45 May 31 12:52:17 docker-7 dockerd[4133]: time="2017-05-31T12
    :52:17.504061522Z" level=info msg="API listen on [::]:4243"
46 $
47
48 (external)$ \*\curl 104.199.209.157:4243/version\*\*
49 {
50   "Version":"17.05.0-ce","ApiVersion":"1.29","MinAPIVersion":"1.12","
    GitCommit":"89658be","GoVersion":"go1.7.5","Os":"linux","Arch":"
    amd64","KernelVersion":"4.8.0-52-generic","BuildTime":"2017-05-04
    T22:15:36.071254972+00:00" }
51
52 (external)$
53 <!--NeedCopy-->
```

Étape 3 : Installation de NetScaler CPX Image

Téléchargez l'image NetScaler CPX sur Docker App Store. Notez que le CPX Express et le CPX ont la même image. Lorsqu'elle est concédée sous licence par NetScaler MAS et qu'elle est installée, l'image devient une instance CPX complète avec 1 Gbit/s de performances. Sans licence, l'image devient une instance CPX Express prenant en charge 20 Mbit/s et 250 connexions SSL.

```

1 $ \*\*sudo docker pull store/citrix/netscalercpx:12.0-41.16\*\*
2 12.0-41.16: Pulling from store/citrix/netscalercpx
3 4e1f679e8ab4: Pull complete
4 a3ed95caeb02: Pull complete
5 2931a926d44b: Pull complete
6 362cd40c5745: Pull complete
7 d10118725a7a: Pull complete
8 1e570419a7e5: Pull complete
9 d19e06114233: Pull complete
10 d3230f008ffd: Pull complete
11 22bdb10a70ec: Pull complete
12 1a5183d7324d: Pull complete
13 241868d4ebff: Pull complete
14 3f963e7ae2fc: Pull complete
15 fd254cf1ea7c: Pull complete
16 33689c749176: Pull complete
17 59c27bad28f5: Pull complete
18 588f5003e10f: Pull complete
19 Digest: sha256:31
    a65cfa38833c747721c6fbc142faec6051e5f7b567d8b212d912b69b4f1ebe
20 Status: Downloaded newer image for store/citrix/netscalercpx:12.0-41.16
21 $
22
23 $ \*\*sudo docker images\*\*
24 REPOSITORY TAG IMAGE ID CREATED SIZE
25 store/citrix/netscalercpx 12.0-41.16 6fa57c38803f 3 weeks ago 415MB
26 $
27 <!--NeedCopy-->

```

Étape 4 : Création d'une instance CPX

Installez l'image CPX sur l'hôte Docker. Ouvrez des ports pour des services spécifiques, comme illustré dans l'exemple ci-dessous, et spécifiez une adresse IP pour NetScaler MAS :

```

1 bash-2.05b# \*\*CHOST=${
2 1:-localhost }
3 \*\*
4 bash-2.05b# \*\*echo | openssl s_client -connect $CHOST:443 | openssl
    x509 -fingerprint -noout | cut -d'=' -f2\*\*
5 depth=0 C = US, ST = California, L = San Jose, O = Citrix NetScaler, OU
    = Internal, CN = Test Only Cert
6 verify error:num=18:self signed certificate
7 verify return:1

```

```

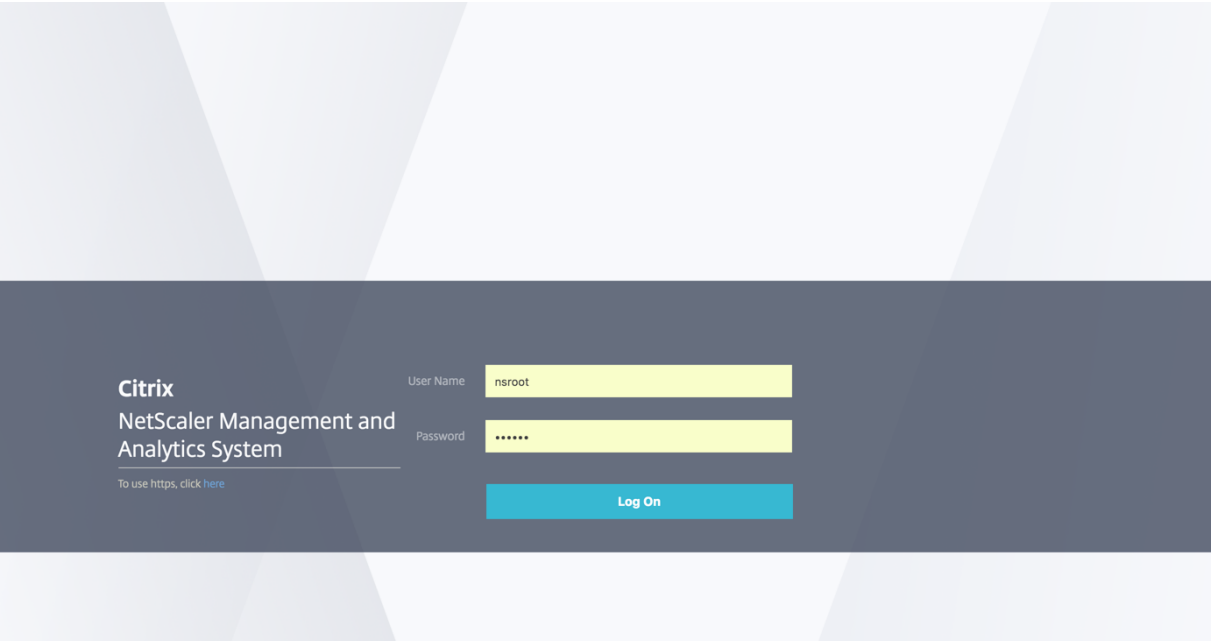
8 depth=0 C = US, ST = California, L = San Jose, O = Citrix NetScaler, OU
  = Internal, CN = Test Only Cert
9 verify return:1
10 DONE
11 24:AA:8B:91:7B:72:5E:6E:C1:FD:86:FA:09:B6:42:49:FC:1E:86:A4
12 bash-2.05b#
13
14 $ \*\*sudo docker run -dt -p 50000:88 -p 5080:80 -p 5022:22 -p 5443:443
    -p 5163:161/udp -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.
    xx -e PLATFORM=CP1000 --privileged=true --ulimit core=-1 -e
    NS_MGMT_SERVER=xx.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/
    netscalercpx:12.0-41.16\*\*
15 44calc6c0907e17a10ffcb9ffe33cd3e9f71898d8812f816e714821870fa3538
16 $
17
18 $ \*\*sudo docker ps\*\*
19 CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
20 44calc6c0907 store/citrix/netscalercpx:12.0-41.16 "/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 $
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
29 root@44calc6c0907:~#
30 root@44calc6c0907:~#
31 root@44calc6c0907:~# \*\*cli_script.sh 'show ns ip'\*\*
32 exec: show ns ip
33 Ippaddress Traffic Domain Type Mode Arp Icmp Vserver State
34 -----
35 1) 172.17.0.2 0 NetScaler IP Active Enabled Enabled NA Enabled
36 2) 192.0.0.1 0 SNIP Active Enabled Enabled NA Enabled
37 Done
38 root@44calc6c0907:~# \*\*cli_script.sh 'show licenseserver'\*\*
39 exec: show licenseserver
40 1) ServerName: xx.xx.xx.xxPort: 27000 Status: 1 Grace: 0 Gptimeleft: 0
41 Done
42 root@44calc6c0907:~# cli_script.sh 'show capacity'
43 exec: show capacity
44 Actualbandwidth: 1000 Platform: CP1000 Unit: Mbps Maxbandwidth: 3000
    Minbandwidth: 20 Instancecount: 0
45 Done
46 root@44calc6c0907:~#
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
56 Chain OUTPUT (policy ACCEPT)
57 target prot opt source destination
58 DOCKER all -- 0.0.0.0/0 !127.0.0.0/8 ADDRTYPE match dst-type LOCAL
59
60 Chain POSTROUTING (policy ACCEPT)
61 target prot opt source destination
62 MASQUERADE all -- 172.17.0.0/16 0.0.0.0/0
63 MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:443
64 MASQUERADE udp -- 172.17.0.2 172.17.0.2 udp dpt:161
65 MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:88
66 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)
70 target prot opt source destination
71 RETURN all -- 0.0.0.0/0 0.0.0.0/0
72 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5443 to:172.17.0.2:443
73 DNAT udp -- 0.0.0.0/0 0.0.0.0/0 udp dpt:5163 to:172.17.0.2:161
74 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:50000 to:172.17.0.2:88
75 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5080 to:172.17.0.2:80
76 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5022 to:172.17.0.2:22
77 $
78 <!--NeedCopy-->
```

Étape 5 : Accordez une licence NetScaler CPX via NetScaler MAS

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



×

Citrix NetScaler Management and Analytics System

May 31 2017 13:14:20 GMT

nsroot

Search here

Applications

Networks

Dashboard

Instances

Instance Groups

Licenses

System Licenses

Third Party Licenses

CPX Licenses

Events

SSL Dashboard

Configuration Jobs

Configuration Audit

Sites

Network Functions

Network Reporting

Analytics

Orchestration

System

Downloads

Networks > License Settings

License Server Port Settings

Proxy Server Port

0

License Server Port

27000

Vendor Daemon Port

7279

License Files

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**.

Add License File

Apply Licenses

Delete

Download

<input type="checkbox"/>	Name	Last Modified	Size
<input type="checkbox"/>	FID_3bf0b423_15ba7640cc6_2664.lic	2017-05-23 21:30:15	1.10 KB
<input type="checkbox"/>	FID_3bf0b423_15ba7640cc6_2672.lic	2017-05-23 21:30:15	1.10 KB
<input type="checkbox"/>	FID_2a2386a8_15b93284902_487e.lic	2017-05-23 21:30:15	1.10 KB
<input type="checkbox"/>	FID_2a2386a8_15b93284902_4878.lic	2017-05-23 21:30:15	1.10 KB
<input type="checkbox"/>	FID_3bf0b423_15ba7640cc6_5281.lic	2017-05-23 21:30:15	1.10 KB
<input type="checkbox"/>	FID_2a2386a8_15b93284902_4870.lic	2017-05-23 21:30:15	1.10 KB
<input type="checkbox"/>	FID_3bf0b423_15ba7640cc6_527b.lic	2017-05-23 21:30:15	1.10 KB
<input type="checkbox"/>	FID_2a2386a8_15b93284902_486a.lic	2017-05-23 21:30:15	1.10 KB
<input type="checkbox"/>	FID_3bf0b423_15ba7640cc6_5275.lic	2017-05-23 21:30:15	1.10 KB
<input type="checkbox"/>	FID_2a2386a8_15b93284902_4864.lic	2017-05-23 21:30:15	1.10 KB

License Expiry Information

Feature	Count	Days To Expiry
No items		

×

Citrix NetScaler Management and Analytics System

Jun 05 2017 15:09:41 GMT

nsroot

Search here

Applications

Networks

Dashboard

Instances

NetScaler MPX

NetScaler VPX

NetScaler SDX

NetScaler CPX

NetScaler Gateway

NetScaler SD-WAN WO

Networks > Instances > NetScaler CPX

NetScaler CPX

Instances 3

Docker Host 0

Add

Provision

Delete

Dashboard

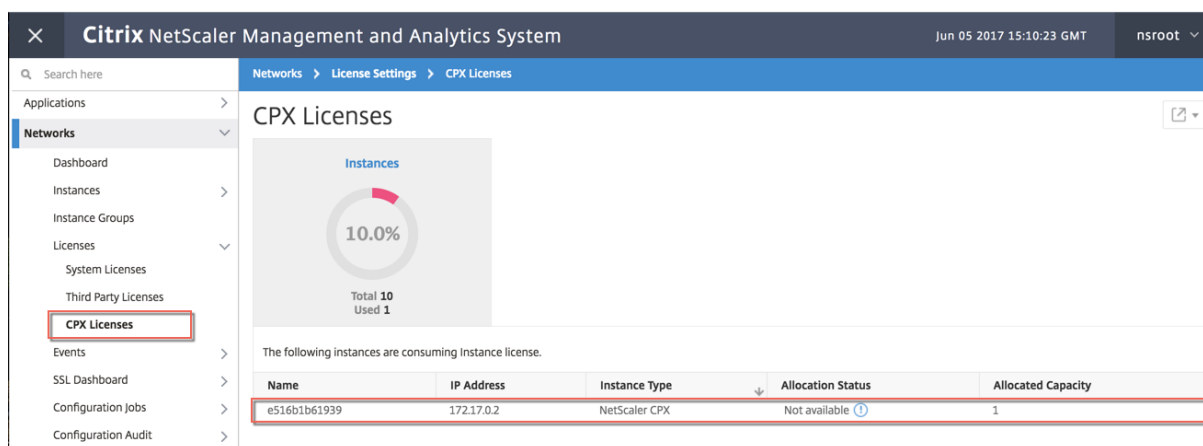
Reboot

Action

Search

Settings

<input type="checkbox"/>	IP Address	Host Name	State	Docker Host	Port Range	SSH Port	HTTP Port	HTTPS Port	SNMP Port
<input type="checkbox"/>	172.17.0.2	-NA-	Out of Service	104.196.190.229		32770	32769	32768	32768
<input type="checkbox"/>	172.17.0.5	-NA-	Out of Service	10.10.15.159	88-88	32785	32784	32783	32773
<input type="checkbox"/>	172.17.0.2	-NA-	Up	104.199.209.157		5022	5080	5443	5163



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 les serveurs Web NGINX.

```

1 $ sudo docker pull nginx
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\*\*
9 098a77974818f451c052ecd172080a7d45e446239479d9213cd4ea6a3678616f
10
11
12 \*\*$ sudo docker run -d -p 82:80 nginx\*\*
13 bbdac2920bb4085f70b588292697813e5975389dd546c0512daf45079798db65
14
15
16 \*\*$ sudo iptables -t nat -L -n\*\*
17 Chain PREROUTING (policy ACCEPT)
18 target prot opt source destination
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

```

```

26 DOCKER all -- 0.0.0.0/0 !127.0.0.0/8 ADDRTYPE match dst-type LOCAL
27
28 Chain POSTROUTING (policy ACCEPT)
29 target prot opt source destination
30 MASQUERADE all -- 172.17.0.0/16 0.0.0.0/0
31 MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:443
32 MASQUERADE udp -- 172.17.0.2 172.17.0.2 udp dpt:161
33 MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:88
34 MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:80
35 MASQUERADE tcp -- 172.17.0.2 172.17.0.2 tcp dpt:22
36 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
41 RETURN all -- 0.0.0.0/0 0.0.0.0/0
42 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5443 to:172.17.0.2:443
43 DNAT udp -- 0.0.0.0/0 0.0.0.0/0 udp dpt:5163 to:172.17.0.2:161
44 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:50000 to:172.17.0.2:88
45 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5080 to:172.17.0.2:80
46 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5022 to:172.17.0.2:22
47 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:81 to:172.17.0.3:80
48 DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:82 to:172.17.0.4:80
49 $
50 <!--NeedCopy-->

```

Configurez CPX pour l'équilibrage de charge et vérifiez la répartition de la charge sur 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/
6 Last login: Mon Jun 5 18:58:54 2017 from 172.17.0.1
7 root@44calc6c0907:~#
8 root@44calc6c0907:~#
9 root@44calc6c0907:~#
10 root@44calc6c0907:~#
11 root@44calc6c0907:~# \*\cli_script.sh "add service web1 172.17.0.3
    HTTP 80"\*\*
12 exec: add service web1 172.17.0.3 HTTP 80
13 Done
14 root@44calc6c0907:~# \*\cli_script.sh "add service web2 172.17.0.4
    HTTP 80"\*\*
15 exec: add service web2 172.17.0.4 HTTP 80
16 Done
17 root@44calc6c0907:~# \*\cli_script.sh "add lb vserver cpx-vip HTTP
    172.17.0.2 88"\*\*
18 exec: add lb vserver cpx-vip HTTP 172.17.0.2 88
19 Done
20 root@44calc6c0907:~# \*\cli_script.sh "bind lb vserver cpx-vip web1
    "\*\*
21 exec: bind lb vserver cpx-vip web1

```

```
22 Done
23 root@44ca1c6c0907:~# \*\*cli_script.sh "bind lb vserver cpx-vip web2
   \*\*
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
30
31 cpx-vip (172.17.0.2:88) - HTTP Type: ADDRESS
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
35 Effective State: UP
36 Client Idle Timeout: 180 sec
37 Down state flush: ENABLED
38 Disable Primary Vserver On Down : DISABLED
39 Appflow logging: ENABLED
40 Port Rewrite : DISABLED
41 No. of Bound Services : 2 (Total) 2 (Active)
42 Configured Method: LEASTCONNECTION
43 Current Method: Round Robin, Reason: A new service is bound
   BackupMethod: ROUNDROBIN
44 Mode: IP
45 Persistence: NONE
46 Vserver IP and Port insertion: OFF
47 Push: DISABLED Push VServer:
48 Push Multi Clients: NO
49 Push Label Rule: none
50 L2Conn: OFF
51 Skip Persistency: None
52 Listen Policy: NONE
53 IcmpResponse: PASSIVE
54 RHISTate: PASSIVE
55 New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0
56 Mac mode Retain Vlan: DISABLED
57 DBS_LB: DISABLED
58 Process Local: DISABLED
59 Traffic Domain: 0
60 TROFS Persistence honored: ENABLED
61 Retain Connections on Cluster: NO
62
63 2) web1 (172.17.0.3: 80) - HTTP State: UP Weight: 1
64 3) web2 (172.17.0.4: 80) - HTTP State: UP Weight: 1
65 Done
66 root@44ca1c6c0907:~#
67
68 (external)$ \*\*curl 104.199.209.157:50000\*\*
69 \<\!\DOCTYPE html\>
70 \<html\>
71 \<head\>
72 \<title\>Welcome to nginx\!\</title\>
```

```
73 \<style\>
74 body {
75
76 width: 35em;
77 margin: 0 auto;
78 font-family: Tahoma, Verdana, Arial, sans-serif;
79 }
80
81 \</style\>
82 \</head\>
83 \<body\>
84 \<h1\>Welcome to nginx\!\</h1\>
85 \<p\>If you see this page, the nginx web server is successfully
86 installed and
87 working. Further configuration is required.\</p\>
88
89 \<p\>For online documentation and support please refer to
90 \<a href="http://nginx.org/"\>nginx.org\</a\>.\<br/\>
91 Commercial support is available at
92 \<a href="http://nginx.com/"\>nginx.com\</a\>.\</p\>
93
94 \<p\>\<em\>Thank you for using nginx.\</em\>\</p\>
95 \</body\>
96 \</html\>
97 (external)$
98 ![localized image](/en-us/cpx/12-1/media/cpx-image-5.png)
99
100 (external)$ for i in {
101 1..100 }
102 ; \*\*do curl http://104.199.209.157:50000 -o /dev/null ; done\*\*
103
104 % Total      % Received % Xferd  Average Speed   Time    Time       Time
105      Current
106
107      Speed
108 100    612    100    612      0      0    1767      0 --:--:-- --:--:--
109      --:--:--    1768
110
111 % Total      % Received % Xferd  Average Speed   Time    Time       Time
112      Current
113
114      Speed
115 100    612    100    612      0      0    1893      0 --:--:-- --:--:--
116      --:--:--    1894
117
118 % Total      % Received % Xferd  Average Speed   Time    Time       Time
119      Current
```

118								Dload	Upload	Total	Spent	Left
119	Speed											
120	100	612	100	612	0	0	1884	0	--:--:--	--:--:--		
121	--:--:-- 1883											
122	% Total		% Received		% Xferd		Average	Speed		Time	Time	Time
123	Current											
124							Dload	Upload	Total	Spent	Left	
125	Speed											
126	100	612	100	612	0	0	1917	0	--:--:--	--:--:--		
127	--:--:-- 1924											
128	% Total		% Received		% Xferd		Average	Speed		Time	Time	Time
129	Current											
130							Dload	Upload	Total	Spent	Left	
131	Speed											
132	100	612	100	612	0	0	1877	0	--:--:--	--:--:--		
133	--:--:-- 1883											
134	% Total		% Received		% Xferd		Average	Speed		Time	Time	Time
135	Current											
136							Dload	Upload	Total	Spent	Left	
137	Speed											
138	100	612	100	612	0	0	1852	0	--:--:--	--:--:--		
139	--:--:-- 1848											
140	% Total		% Received		% Xferd		Average	Speed		Time	Time	Time
141	Current											
142							Dload	Upload	Total	Spent	Left	
143	Speed											
144	100	612	100	612	0	0	1860	0	--:--:--	--:--:--		
145	--:~:~:~ 1865											
146	% Total		% Received		% Xferd		Average	Speed		Time	Time	Time
147	Current											
148							Dload	Upload	Total	Spent	Left	
149	Speed											
150	100	612	100	612	0	0	1887	0	--:~:~:~ 1888	--:~:~:~		
151												
152	% Total		% Received		% Xferd		Average	Speed		Time	Time	Time
	Current											


```
189
190 (external)$
191
192
193
194
195
196 root@44ca1c6c0907:~# \*\*cli_script.sh 'stat lb vserver cpx-vip'\*\*
197
198 exec: stat lb vserver cpx-vip
199
200
201
202 Virtual Server Summary
203
204           vsvrIP  port  Protocol  State  Health
205   actSvcs
206   cpx-vip      172.17.0.2    88      HTTP      UP      100
207           2
208
209   inactSvcs
210   cpx-vip      0
211
212
213
214
215 Virtual Server Statistics
216
217           Rate (/s)
218   Total
219   Vserver hits      0
220   101
221
222   Requests      0
223   101
224
225   Responses      0
226   101
227
228   Request bytes      0
229   8585
230
231   Response bytes      0
232   85850
233
234   Total Packets rcvd      0
235   708
236
237   Total Packets sent      0
```

233	408	
234	Current client connections	--
	0	
235		
236	Current Client Est connections	--
	0	
237		
238	Current server connections	--
	0	
239		
240	Current Persistence Sessions	--
	0	
241		
242	Requests in surge queue	--
	0	
243		
244	Requests in vserver's surgeQ	--
	0	
245		
246	Requests in service's surgeQs	--
	0	
247		
248	Spill Over Threshold	--
	0	
249		
250	Spill Over Hits	--
	0	
251		
252	Labeled Connection	--
	0	
253		
254	Push Labeled Connection	--
	0	
255		
256	Deferred Request	0
	0	
257		
258	Invalid Request/Response	--
	0	
259		
260	Invalid Request/Response Dropped	--
	0	
261		
262	Vserver Down Backup Hits	--
	0	
263		
264	Current Multipath TCP sessions	--
	0	
265		
266	Current Multipath TCP subflows	--
	0	
267		


```

268 Apdex for client response times.      --
    1.00
269
270 Average client TTLB                  --
    0
271
272 web1          172.17.0.3    80      HTTP      UP      51
    0/s
273
274 web2          172.17.0.4    80      HTTP      UP      50
    0/s
275
276 Done
277
278 root@44ca1c6c0907:~#
279 <!--NeedCopy-->

```

Résolution des problèmes liés à NetScaler CPX

November 23, 2023

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.

- 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`

```

1   containers:
2     - name: cpx-ingress
3       image: "quay.io/citrix/citrix-k8s-cpx-ingress:13.0-58.30"
4       tty: true
5       securityContext:
6         privileged: true

```

```

7      env:
8
9      <!--NeedCopy-->

```

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  root@localhost# kubectl exec cpx-ingress1-55b9b6fc75-t5kc6 -c cpx
    -n citrix-adc /var/netscaler/bins/cli_script.sh "show
    techsupport"
2  exec: show techsupport
3  Scope:  NODE
4  Done
5  root@localhost# kubectl cp cpx-ingress1-55b9b6fc75-t5kc6:var/tmp/
    support/collector_P_192.168.29.232_31Aug2020_07_30.tar.gz /tmp
    /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-r--r-- 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 :

```

1  kubectl describe pods <pod-name> [-c <container-name>] [-n <
    namespace-name>]

```

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 :

```
1 kubectl cp <pod-name>:<absolute-src-path> <dst-path> [-c <
  container-name>] [-n <namespace-name>]
```

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 :

```
1 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-ingress1-69b9b8c648-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 commande `ping <ADM-IP>` 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,

```

1 root@cpx-ingress-596d56bb6-zbx6h:/cpx/log# cli_script.sh "show
  capacity"
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 :

```
1 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 : 0 Mbps
6 Standard Bandwidth Available : 0 Mbps
7 Enterprise Bandwidth Total : 0 Mbps
8 Enterprise Bandwidth Available : 0 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
14 <!--NeedCopy-->

```

- 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 :

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

Exemple :

```

1 ng472 | grep -i port
2 Ports: 80/TCP, 443/TCP, 9080/TCP, 9443/TCP
3 Host Ports: 0/TCP, 0/TCP, 0/TCP, 0/TCP
4 NS_HTTP_PORT: 9080

```

```

5      NS_HTTPS_PORT:          9443
6      Port:                   <none>
7      Host Port:              <none>
8      NS_PORT:                 80
9      <!--NeedCopy-->

```

- 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 ?

Vous pouvez trouver les journaux relatifs à l'enregistrement de NetScaler CPX auprès de Citrix ADM 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=<ADM-IP>`
- `NS_MGMT_FINGER_PRINT =<ADM-Fingerprint>`
- `HOST=<Node-IP>`
- `NS_HTTP_PORT=<Mapped-HTTP-Port-on-Node>`
- `NS_HTTPS_PORT=<Mapped-HTTPS-Port-on-Node>`
- `NS_SSH_PORT=<Mapped-SSH-Port-on-Node>`
- `NS_SNMP_PORT=<Mapped-SNMP-Port-on-Node>`
- `NS_ROUTABLE=0` (L'adresse IP du pod NetScaler CPX n'est pas routable depuis l'extérieur.)

- 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:

```
1 root@087a1e34642d:/# cli_script.sh "show ns ip"
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
10 Ipaddress      Traffic Domain      Type      Mode
      Arp      Icmp      Vserver  State
11 -----
12 172.17.0.3      0
      Enabled Enabled NA      Enabled NetScaler IP Active
13 192.0.0.1      0
      Enabled Enabled NA      Enabled SNIP Active
14 Done
```



© 2024 Cloud Software Group, Inc. All rights reserved. Cloud Software Group, the Cloud Software Group logo, and other marks appearing herein are property of Cloud Software Group, Inc. and/or one or more of its subsidiaries, and may be registered with the U.S. Patent and Trademark Office and in other countries. All other marks are the property of their respective owner(s).
