NSX Container Plug-in pour OpenShift - Guide d'installation et d'administration

VMware NSX Container Plug-in 2.4
VMware NSX-T Data Center 2.4

vmware®

Vous trouverez la documentation technique la plus récente sur le site Web de VMware, à l'adresse :

https://docs.vmware.com/fr/

Les dernières mises à jour produit se trouvent également sur le site Web de VMware.

Si vous avez des commentaires à propos de cette documentation, envoyez-les à l'adresse suivante :

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

VMware France SAS.
Tour Franklin
100-101 Terrasse Boieldieu
92042 Paris La Défense 8 Cedex
France
www.vmware.com/fr

Table des matières

NSX-T Container Plug-in pour OpenShift - Guide d'installation et d'administration

Ce guide décrit comment installer et administrer NSX Container Plug-in (NCP) pour fournir l'intégration entre NSX-T Data Center et OpenShift.

Public visé

Ce guide est destiné aux administrateurs système et réseau. Il est impératif de connaître les procédures d'installation et d'administration de NSX-T Data Center et d'OpenShift.

Glossaire VMware Technical Publications

VMware Technical Publications fournit un glossaire de termes pouvant ne pas vous être familiers. Pour consulter la définition des termes utilisés dans la documentation technique VMware, visitez le site Web http://www.vmware.com/support/pubs.

Présentation de NSX-T Container Plug-in

1

NSX Container Plug-in (NCP) fournit l'intégration entre NSX-T Data Center et les orchestrateurs de conteneur, tels que Kubernetes, ainsi que l'intégration entre NSX-T Data Center et les logiciels PaaS (plate-forme en tant que service) basés sur un conteneur, tels qu'OpenShift. Ce guide décrit comment configurer NCP avec OpenShift.

Le composant principal de NCP s'exécute dans un conteneur et communique avec NSX Manager et avec le plan de contrôle OpenShift. NCP surveille les modifications apportées aux conteneurs et d'autres ressources et gère des ressources réseau, telles que des ports logiques, des commutateurs, des routeurs et des groupes de sécurité pour les conteneurs en appelant NSX API.

Le plug-in NSX CNI s'exécute sur chaque nœud OpenShift. Il surveille les événements de cycle de vie des conteneurs, connecte une interface de conteneur au vSwitch invité et programme le vSwitch invité pour marquer et transférer le trafic de conteneur entre les interfaces de conteneur et la carte réseau virtuelle.

NCP fournit les fonctionnalités suivantes :

- Création automatique d'une topologie logique NSX-T pour un cluster OpenShift et création d'un réseau logique distinct pour chaque espace de noms OpenShift.
- Connexion des espaces OpenShift au réseau logique et allocation des adresses IP et MAC.
- Prise en charge de la traduction d'adresse réseau (NAT) et allocation d'une adresse IP SNAT distincte pour chaque espace de noms OpenShift.

  Note Lors de la configuration de NAT, le nombre total d'adresses IP traduites ne peut pas dépasser 1 000.

- Implémentation de stratégies réseau OpenShift avec pare-feu distribué NSX-T.
  - Prise en charge des stratégies réseau d'entrée et de sortie.
  - Prise en charge de sélecteur d'`IPBlock` dans les stratégies réseau.
  - Prise en charge de `matchLabels` et de `matchExpression` lors de la spécification de sélecteurs d'étiquette pour les stratégies réseau.
- Implémentation de route OpenShift avec un équilibrage de charge de couche 7 NSX-T.
  - Prise en charge des routes HTTP et HTTPS avec terminaison Edge TLS.

    - Prise en charge des routes avec d'autres serveurs principaux et sous-domaines génériques.
- Création de balises sur le port de commutateur logique NSX-T pour l'espace de noms, le nom de l'espace et les étiquettes d'un espace, et définition par l'administrateur de groupes de sécurité et de stratégies NSX-T Data Center basées sur les balises.

Dans cette version, NCP prend en charge un seul cluster OpenShift.

Ce chapitre contient les rubriques suivantes :

Conditions de compatibilité

NSX Container Plug-in (NCP) présente les conditions de compatibilité suivantes.

Produit logiciel	Version
NSX-T Data Center	2.3, 2.4
Hyperviseur pour machines virtuelles d'hôte de conteneur	■ Version de vSphere prise en charge ■ RHEL KVM 7.4, 7.5, 7.6
Système d'exploitation d'hôte de conteneur	RHEL 7.4, 7.5, 7.6
Plateforme sous forme de service	OpenShift 3.10, 3.11
Open vSwitch d'hôte de conteneur	2.10.2 (fourni avec NSX-T Data Center 2.4)

Présentation de l'installation

L'installation et la configuration de NCP impliquent les étapes suivantes. Pour effectuer la procédure correctement, vous devez être familiarisé avec l'installation et l'administration de NSX-T Data Center et d'OpenShift.

1 Installez NSX-T Data Center.

2 Créez une zone de transport de superposition.

3 Créez un commutateur logique de superposition et connectez les nœuds au commutateur.

4 Créez un routeur logique de niveau 0.

5 Créez des blocs d'adresses IP pour les espaces.

6 Créez des pools d'adresses IP pour SNAT (Source Network Address Translation, traduction d'adresse réseau source).

7 Préparez le fichier hosts Ansible.

8 Installez NCP et OpenShift à l'aide d'un seul playbook.

Mettre à niveau NCP

Cette section décrit comment mettre à niveau NCP vers la version 2.4.0.

Procédure

1. Mettez à niveau le module RPM CNI, le modèle DaemonSet de l'agent de nœud NSX et ReplicationController de NCP.
2. Préparez le fichier hosts Ansible.

   Le paramètre `openshift_node_group_name` doit être spécifié pour chaque nœud. Par exemple,

   ```
   [nodes]
   config-master.example.com openshift_hostname=config-master.example.com
   openshift_node_group_name=config-master
   ```

3. (Facultatif) Configurez l'équilibrage de charge

   Ajoutez une étape pour spécifier un pool d'adresses IP différent pour les adresses IP externes pour le service d'équilibrage de charge. Par exemple,

   `external_ip_pools_lb = <nsx ip pool name>`

Configuration des ressources de NSX-T 2

Des ressources de NSX-T Data Center doivent être créées pour fournir la mise en réseau aux nœuds OpenShift.

Configuration des ressources de NSX-T

Les ressources de NSX-T Data Center que vous devez configurer incluent une zone de transport de superposition, un routeur logique de niveau 0, un commutateur logique pour connecter les machines virtuelles du nœud, des blocs d'adresses IP pour les nœuds Kubernetes et un pool d'adresses IP pour SNAT.

Important Si vous exécutez avec NSX-T Data Center 2.4 ou version ultérieure, vous devez configurer les ressources NSX-T à l'aide de l'onglet **Mise en réseau et sécurité avancées**.

Dans le fichier de configuration NCP `ncp.ini`, les ressources NSX-T Data Center sont spécifiées à l'aide de leurs UUID ou de leurs noms.

Zone de transport de superposition

Connectez-vous à NSX Manager et recherchez la zone de transport de superposition utilisée pour la mise en réseau de conteneur ou créez-en une.

Spécifiez une zone de transport de superposition pour un cluster en définissant l'option `overlay_tz` dans la section `[nsx_v3]` de `ncp.ini`. Cette étape est facultative. Si vous ne définissez pas `overlay_tz`, NCP récupère automatiquement l'ID de zone de transport de superposition à partir du routeur de niveau 0.

Routage logique de niveau 0

Connectez-vous à NSX Manager et recherchez le routeur utilisé pour la mise en réseau de conteneur ou créez-en un.

Spécifiez un routeur logique de niveau 0 pour un cluster en définissant l'option `tier0_router` dans la section `[nsx_v3]` de `ncp.ini`.

Note Le routeur doit être créé en mode actif-en veille.

Commutateur logique

Les cartes réseau virtuelles utilisées par le nœud pour le trafic de données doivent être connectées à un commutateur logique de superposition. Il n'est pas obligatoire que l'interface de gestion du nœud soit connectée à NSX-T Data Center, bien que cela facilite la configuration. Vous pouvez créer un commutateur logique en vous connectant à NSX Manager. Sur le commutateur, créez des ports logiques et attachez-y les cartes réseau virtuelles du nœud. Les ports logiques doivent avoir les balises suivantes :

- balise : `<cluster_name>`, étendue : `ncp/cluster`
- balise : `<node_name>`, étendue : `ncp/node_name`

La valeur `<cluster_name>` doit correspondre à la valeur de l'option `cluster` dans la section `[coe]` de `ncp.ini`.

Blocs d'adresses IP pour des espaces Kubernetes

Connectez-vous à NSX Manager et créer un ou plusieurs blocs d'adresses IP. Spécifiez le bloc d'adresses IP au format CIDR.

Spécifiez les blocs d'adresses IP pour les espaces Kubernetes en définissant l'option `container_ip_blocks` dans la section `[nsx_v3]` de `ncp.ini`.

Vous pouvez également créer des blocs d'adresses IP spécialement pour les espaces de noms non SNAT.

Spécifiez les blocs d'adresses IP non SNAT en définissant l'option `no_snat_ip_blocks` dans la section `[nsx_v3]` de `ncp.ini`.

Si vous créez des blocs d'adresses IP non SNAT alors que NCP est en cours d'exécution, vous devez redémarrer NCP. Sinon, NCP continuera d'utiliser les blocs d'adresses IP partagés jusqu'à leur épuisement.

Note Lorsque vous créez un bloc d'adresses IP, le préfixe ne doit pas être supérieur à la valeur du paramètre `subnet_prefix` dans le fichier de configuration de NCP `ncp.ini`.

Pool d'adresses IP pour SNAT

Le pool d'adresses IP sert à allouer des adresses IP qui seront utilisées pour la traduction d'adresses IP d'espace via des règles SNAT et pour l'exposition de contrôleurs d'entrée via des règles SNAT/DNAT, comme des adresses IP flottantes OpenStack. Ces adresses IP sont également appelées adresses IP externes.

Plusieurs clusters Kubernetes utilisent le même pool d'adresses IP externes. Chaque instance NCP utilise un sous-ensemble de ce pool pour le cluster Kubernetes qu'il gère. Par défaut, le même préfixe de sous-réseau pour les sous-réseaux d'espace sera utilisé. Pour utiliser une taille de sous-réseau différente, mettez à jour l'option `external_subnet_prefix` dans la section `[nsx_v3]` dans `ncp.ini`.

Connectez-vous à NSX Manager, et créez un pool ou recherchez un pool existant.

Spécifiez les pools d'adresses IP pour SNAT en définissant l'option `external_ip_pools` dans la section `[nsx_v3]` de `ncp.ini`.

Vous pouvez également configurer SNAT pour un service spécifique en ajoutant une annotation au service. Par exemple,

```
apiVersion: v1
kind: Service
metadata:
  name: svc-example
  annotations:
    ncp/snat_pool: <external IP pool ID or name>
  selector:
    app: example
...
```

NCP configurera la règle SNAT pour ce service. L'adresse IP source de la règle correspond à l'ensemble des espaces de serveur principal. L'adresse IP de destination est l'adresse IP SNAT allouée à partir du pool d'adresses IP externe spécifié. Notez les points suivants :

- Le pool d'adresses IP spécifié par `ncp/snat_pool` doit déjà exister dans NSX-T Data Center avant que le service ne soit configuré. Le pool d'adresses IP doit avoir la balise **`{"ncp/owner": cluster:<cluster>}`**.
- Dans NSX-T Data Center, la priorité de la règle SNAT pour le service est supérieure à celle du projet.
- Si un espace est configuré avec plusieurs règles SNAT, une seule fonctionnera.

Vous pouvez spécifier l'espace de noms auquel les adresses IP sont allouées à partir du pool d'adresses IP SNAT en ajoutant la balise suivante au pool d'adresses IP.

- étendue : `ncp/owner`, balise : `ns:<namespace_UUID>`

Vous pouvez obtenir l'UUID de l'espace de noms à l'aide de l'une des commandes suivantes :

```
oc get ns –o yaml
```

Notez les points suivants :

- Chaque balise doit spécifier un UUID. Vous pouvez créer plusieurs balises pour le même pool.
- Si vous modifiez les balises après l'allocation d'adresses IP à des espaces de noms selon les anciennes balises, ces adresses IP sont récupérées uniquement après la modification des configurations SNAT des services ou le redémarrage de NCP.
- La balise de propriétaire d'un espace de noms est facultative. Sans cette balise, les adresses IP du pool IP SNAT peuvent être allouées à n'importe quel espace de noms.

(Facultatif) Sections de marqueur de pare-feu

Pour permettre à l'administrateur de créer des règles de pare-feu et que celles-ci n'interfèrent pas avec les sections de pare-feu créées par NCP et basées sur des stratégies réseau, connectez-vous à NSX Manager et créer deux sections de pare-feu.

Spécifiez les sections de pare-feu de marqueur en définissant les options `bottom_firewall_section_marker` et `top_firewall_section_marker` dans la section `[nsx_v3]` de `ncp.ini`.

La section de pare-feu inférieure doit se trouver sous la section de pare-feu supérieure. Une fois ces sections de pare-feu créées, toutes les sections de pare-feu créées par NCP pour une isolation sont créées au-dessus de la section de pare-feu inférieure, et toutes les sections de pare-feu créées par NCP pour une stratégie sont créées en dessous de la section de pare-feu supérieure. Si ces sections de marqueur ne sont pas créées, toutes les règles d'isolation sont créées en bas et toutes les sections de stratégie sont créées en haut. L'utilisation de plusieurs sections de pare-feu de marqueur possédant la même valeur par cluster n'est pas prise en charge et provoque une erreur.

Installation de NCP

3

NCP est entièrement intégré à OpenShift. Lorsque vous ajoutez les paramètres nécessaires dans le fichier hosts Ansible et installez OpenShift, NCP est installé automatiquement.

Ce chapitre contient les rubriques suivantes :

Configuration système requise

Avant d'installer OpenShift, assurez-vous que votre environnement satisfait certaines conditions requises.

Conditions générales

- Ansible 2.4 ou version ultérieure.

Spécifications des machines virtuelles

Les machines virtuelles du nœud Openshift doivent disposer de deux cartes réseau virtuelles :

- Une carte réseau virtuelle de gestion connectée au commutateur logique disposant d'une liaison montante vers le routeur de gestion de niveau 1.
- La deuxième carte réseau virtuelle sur toutes les machines virtuelles doit disposer des balises suivantes dans NSX-T afin que NCP sache quel port est utilisé comme VIF parent pour tous les espaces en cours d'exécution sur le nœud OpenShift particulier.

```
{'ncp/node_name':  '<node_name>'}
{'ncp/cluster': '<cluster_name>'}
```

Spécifications des machines bare metal

- Les nœuds OpenShift doivent être des nœuds de transport NSX-T et les balises mentionnées ci-dessus doivent être appliquées aux nœuds de transport plutôt qu'aux VIF.
- Le fichier hosts Ansible doit inclure ce paramètre : `nsx_node_type='BAREMETAL'`.

Configuration requise de NSX-T

- Routeur de niveau 0.
- Zone de transport de superposition.
- Bloc d'adresses IP pour la mise en réseau d'espace.
- (Facultatif) Bloc d'adresses IP pour la mise en réseau d'espace routé (pas de NAT).
- Pool d'adresses IP pour SNAT. Par défaut, le bloc d'adresses IP pour la mise en réseau d'espace est routable uniquement à l'intérieur de NSX-T. NCP utilise ce pool d'adresses IP pour fournir une connectivité vers l'extérieur.
- (Facultatif) Sections de pare-feu supérieure et inférieure. NCP placera les règles de stratégie réseau Kubernetes entre ces deux sections.
- Les RPM de plug-in Open vSwitch et CNI doivent être hébergés sur un serveur HTTP accessible depuis les machines virtuelles du nœud OpenShift.

Image Docker de NCP

Actuellement, l'image docker de NCP n'est pas publiquement accessible. Vous devez disposer de l'image `nsx-ncp` dans un registre privé local ou procéder comme suit :

`ansible-playbook [-i /path/to/inventory] playbooks/prerequisites.yml`

Sur tous les nœuds :

```
docker load -i nsx-ncp-rhel-xxx.yyyyyyyy.tar
docker image tag registry.local/xxx.yyyyyyyy/nsx-ncp-rhel nsx-ncp
ansible-playbook [-i /path/to/inventory] playbooks/deploy_cluster.yml
```

Préparation du fichier hosts Ansible

Vous devez spécifier les paramètres NCP dans le fichier hosts Ansible pour l'instance de NCP à intégrer à OpenShift.

Une fois que vous spécifiez les paramètres suivants dans le fichier hosts Ansible, l'installation d'OpenShift installe automatiquement NCP.

- `openshift_use_nsx=True`
- `openshift_use_openshift_sdn=False`
- `os_sdn_network_plugin_name='cni'`
- `nsx_openshift_cluster_name='ocp-cluster1'`

  (Obligatoire) Cela est obligatoire, car plusieurs clusters Openshift/Kubernetes peuvent se connecter à la même instance de NSX Manager.

- `nsx_api_managers='10.10.10.10'`

  (Obligatoire) Adresse IP de NSX Manager. Pour un cluster NSX Manager, spécifiez les adresses IP séparées par des virgules.

- `nsx_tier0_router='MyT0Router'`

  (Obligatoire) Nom ou UUID du routeur de niveau 0 auquel se connecteront les routeurs de niveau 1 du projet.

- `nsx_overlay_transport_zone='my_overlay_tz'`

  (Obligatoire) Nom ou UUID de la zone de transport de superposition qui sera utilisée pour créer des commutateurs logiques.

- `nsx_container_ip_block='ip_block_for_my_ocp_cluster'`

  (Obligatoire) Nom ou UUID d'un bloc d'adresses IP configuré sur NSX-T. Il y aura un sous-réseau par projet en dehors de ce bloc d'adresses IP. Ces réseaux seront derrière SNAT et non routables.

- `nsx_ovs_uplink_port='ens224'`

  (Obligatoire) Si l'instance de NSX-T est en mode `HOSTVM`, une deuxième carte réseau virtuelle (vNIC) est requise pour la mise en réseau d'espace sur les nœuds OCP, différente de la carte réseau virtuelle de gestion. Il est fortement recommandé que les deux cartes réseaux virtuelles soit connectées à des commutateurs logiques NSX-T. La seconde carte réseau virtuelle (hors gestion) doit être fournie ici. Pour une configuration bare metal, ce paramètre n'est pas nécessaire.

- `nsx_cni_url='http://myserver/nsx-cni.rpm'`

  (Obligatoire) Condition requise temporaire jusqu'à ce que NCP puisse démarrer les nœuds. Nous devons placer `nsx-cni` sur un serveur HTTP.

- `nsx_ovs_url='http://myserver/openvswitch.rpm'`
- `nsx_kmod_ovs_url='http://myserver/kmod-openvswitch.rpm'`

  (Obligatoire) Paramètres temporaires jusqu'à ce que NCP puisse démarrer les nœuds. Peut être ignoré dans une configuration bare metal.

- `nsx_node_type='HOSTVM'`

  (Facultatif) Défini par défaut sur `HOSTVM`. Défini sur `BAREMETAL` si OpenShift ne s'exécute pas dans des machines virtuelles.

- `nsx_k8s_api_ip=192.168.10.10`

  (Facultatif) S'il est défini, NCP accède à cette adresse IP, sinon à l'adresse IP du service Kubernetes.

- `nsx_k8s_api_port=192.168.10.10`

  (Facultatif) Par défaut 443 pour le service Kubernetes. Défini sur 8443 si vous l'utilisez avec `nsx_k8s_api_ip` pour spécifier l'adresse IP du nœud master.

- `nsx_insecure_ssl=true`

(Facultatif) La valeur par défaut est `true`, car NSX Manager dispose d'un certificat non approuvé. Si vous avez remplacé le certificat par un certificat approuvé, vous pouvez le définir sur `false`.

- `nsx_api_user='admin'`
- `nsx_api_password='super_secret_password'`
- `nsx_subnet_prefix=24`

  (Facultatif) Défini par défaut sur 24. Il s'agit de la taille de sous-réseau dédiée par projet Openshift. Si le nombre d'espaces dépasse la taille du sous-réseau, un nouveau commutateur logique ayant la même taille de sous-réseau est ajouté au projet.
- `nsx_use_loadbalancer=true`

  (Facultatif) Défini par défaut sur `true`. Défini sur `false` si vous ne souhaitez pas utiliser d'équilibrages de charge NSX-T pour les routes OpenShift et les services de type LoadBalancer.
- `nsx_lb_service_size='SMALL'`

  (Facultatif) Défini par défaut sur `SMALL`. Selon la taille de NSX Edge, `MEDIUM` ou `LARGE` est également possible.
- `nsx_no_snat_ip_block='router_ip_block_for_my_ocp_cluster'`

  (Facultatif) Si l'annotation `ncp/no_snat=true` est appliquée sur un projet ou un espace de noms, le sous-réseau proviendra de ce bloc d'adresses IP et aucun SNAT ne lui sera dédié. Il est censé être routable.
- `nsx_external_ip_pool='external_pool_for_snat'`

  (Obligatoire) Pool d'adresses IP pour SNAT et l'équilibrage de charge si `nsx_external_ip_pool_lb` n'est pas défini.
- `nsx_external_ip_pool_lb='my_ip_pool_for_lb'`

  (Facultatif) Définissez cette option si vous souhaitez un pool d'adresses IP distinct pour `Router` et pour `SvcTypeLB`.
- `nsx_top_fw_section='top_section'`

  (Facultatif) Les règles de stratégie réseau Kubernetes seront traduites en règles de pare-feu NSX-T et placées en dessous de cette section.
- `nsx_bottom_fw_section='bottom_section'`

  (Facultatif) Les règles de stratégie réseau Kubernetes seront traduites en règles de pare-feu NSX-T et placées au-dessus de cette section.
- `nsx_api_cert='/path/to/cert/nsx.crt'`
- `nsx_api_private_key='/path/to/key/nsx.key`

  (Facultatif) Si ce paramètre est défini, `nsx_api_user` et `nsx_api_password` sont ignorés. Le certificat doit être téléchargé sur NSX-T et une authentification d'utilisateur d'identité de principal avec ce certificat doit être créée manuellement.

- `nsx_lb_default_cert='/path/to/cert/nsx.crt'`
- `nsx_lb_default_key='/path/to/key/nsx.key`

  (Facultatif) L'équilibrage de charge NSX-T nécessite un certificat par défaut afin de pouvoir créer des SNI pour les routes basées sur TLS. Ce certificat sera présenté uniquement si aucune route n'est configurée. S'il n'est pas fourni, un certificat autosigné sera généré.

Exemple de fichier hosts Ansible

```
[OSEv3:children]
masters
nodes
etcd

[OSEv3:vars]
ansible_ssh_user=root
openshift_deployment_type=origin

openshift_master_identity_providers=[{'name': 'htpasswd_auth', 'login': 'true', 'challenge': 'true',
'kind': 'HTPasswdPasswordIdentityProvider'}]
openshift_master_htpasswd_users={'yasen' : 'password'}

openshift_master_default_subdomain=demo.corp.local
openshift_use_nsx=true
os_sdn_network_plugin_name=cni
openshift_use_openshift_sdn=false
openshift_node_sdn_mtu=1500

# NSX specific configuration
nsx_openshift_cluster_name='ocp-cluster1'
nsx_api_managers='192.168.110.201'
nsx_api_user='admin'
nsx_api_password='VMware1!'
nsx_tier0_router='DefaultT0Router'
nsx_overlay_transport_zone='overlay-tz'
nsx_container_ip_block='ocp-pod-networking'
nsx_no_snat_ip_block='ocp-nonat-pod-networking'
nsx_external_ip_pool='ocp-external'
nsx_top_fw_section='openshift-top'
nsx_bottom_fw_section='openshift-bottom'
nsx_ovs_uplink_port='ens224'
nsx_cni_url='http://1.1.1.1/nsx-cni-2.3.2.x86_64.rpm'
nsx_ovs_url='http://1.1.1.1/openvswitch-2.9.1.rhel75-1.x86_64.rpm'
nsx_kmod_ovs_url='http://1.1.1.1/kmod-openvswitch-2.9.1.rhel75-1.el7.x86_64.rpm'

[masters]
ocp-master.corp.local

[etcd]
ocp-master.corp.local

[nodes]
ocp-master.corp.local ansible_ssh_host=10.1.0.10 openshift_node_group_name='node-config-master'
```

```
ocp-node1.corp.local ansible_ssh_host=10.1.0.11 openshift_node_group_name='node-config-infra'
ocp-node2.corp.local ansible_ssh_host=10.1.0.12 openshift_node_group_name='node-config-infra'
ocp-node3.corp.local ansible_ssh_host=10.1.0.13 openshift_node_group_name='node-config-compute'
ocp-node4.corp.local ansible_ssh_host=10.1.0.14 openshift_node_group_name='node-config-compute'
```

Équilibrage de charge

4

L'équilibrage de charge NSX-T Data Center est intégré à OpenShift et est utilisé comme routeur OpenShift.

NCP surveille les événements de routage et de point de terminaison OpenShift, et configure les règles d'équilibrage de charge en fonction de la spécification de la route. En conséquence, l'équilibrage de charge NSX-T Data Center transférera le trafic de couche 7 entrant vers les espaces principaux en fonction des règles.

Configuration de l'équilibrage de charge

La configuration de l'équilibrage de charge implique la configuration d'un service Kubernetes LoadBalancer ou d'une route OpenShift. Vous devez également configurer le contrôleur de réplication NCP. Le service LoadBalancer est pour le trafic de couche 4 et la route OpenShift est pour le trafic de couche 7.

Lorsque vous configurez un service Kubernetes LoadBalancer, une adresse IP est allouée au service à partir du bloc d'adresses IP externe que vous configurez. L'équilibrage de charge est exposé sur cette adresse IP et sur le port de service. Vous pouvez spécifier le nom ou l'ID d'un pool d'adresses IP à l'aide de la spécification `loadBalancerIP` dans la définition de l'équilibrage de charge. L'adresse IP du service d'équilibrage de charge sera allouée à partir de ce pool d'adresses IP. Si la spécification `loadBalancerIP` est vide, l'adresse IP sera allouée à partir du bloc d'adresses IP externe que vous configurez.

Le pool d'adresses IP spécifié par `loadBalancerIP` doit avoir la balise **`{"ncp/owner": cluster:<cluster>}`**.

Pour utiliser l'équilibrage de charge NSX-T Data Center, vous devez configurer l'équilibrage de charge dans NCP. Dans le fichier `ncp_rc.yml`, procédez comme suit :

1 Définissez `use_native_loadbalancer` = `True`.

2 Définissez `pool_algorithm` sur `WEIGHTED_ROUND_ROBIN`.

3 Définissez lb_default_cert_path et lb_priv_key_path comme noms de chemin d'accès complet du fichier de certificat signé par une autorité de certification et du fichier de clé privée, respectivement. Reportez-vous à la section ci-dessous pour un exemple de script permettant de générer un certificat signé par une autorité de certification. En outre, placez le certificat et la clé par défaut dans l'espace NCP. Reportez-vous à la section ci-dessous pour obtenir des instructions.

4 (Facultatif) Spécifiez un paramètre de persistance avec les paramètres `l4_persistence` et `l7_persistence`. Les options disponibles pour la persistance de la couche 4 est l'adresse IP source. Les options disponibles pour la persistance de la couche 7 sont le cookie et l'adresse IP source. La valeur par défaut est `<None>`. Par exemple,

```
# Choice of persistence type for ingress traffic through L7 Loadbalancer.
# Accepted values:
# 'cookie'
# 'source_ip'
l7_persistence = cookie

# Choice of persistence type for ingress traffic through L4 Loadbalancer.
# Accepted values:
# 'source_ip'
l4_persistence = source_ip
```

5 (Facultatif) Définissez `service_size` = `SMALL`, `MEDIUM` ou `LARGE`. La valeur par défaut est `SMALL`.

6 Si vous utilisez OpenShift 3.11, vous devez effectuer la configuration suivante pour qu'OpenShift n'attribue aucune adresse IP au service LoadBalancer.

- Définissez `ingressIPNetworkCIDR` sur 0.0.0.0/32 sous `networkConfig` dans le fichier `/etc/origin/master/master-config.yaml`.
- Redémarrez le serveur d'API et les contrôleurs à l'aide des commandes suivantes :

```
master-restart api
master-restart controllers
```

Pour un service LoadBalancer Kubernetes, vous pouvez également spécifier `sessionAffinity` sur la spécification de service pour configurer le comportement de persistance du service si la persistance de couche 4 globale est désactivée, c'est-à-dire que `l4_persistence` est défini sur `<None>`. Si `l4_persistence` est défini sur `source_ip`, `sessionAffinity` sur la spécification de service peut être utilisé pour personnaliser le délai d'expiration de la persistance pour le service. Le délai d'expiration de la persistance de couche 4 par défaut est de 10 800 secondes (identique à celui spécifié dans la

documentation Kubernetes pour les services) (https://kubernetes.io/docs/concepts/services-networking/service). Tous les services ayant un délai d'expiration de persistance par défaut partageront le même profil de persistance d'équilibrage de charge NSX-T. Un profil dédié sera créé pour chaque service avec un délai d'expiration de persistance non défini par défaut.

Note Si le service du serveur principal d'une entrée est un service de type LoadBalancer, le serveur virtuel de couche 4 pour le service et le serveur virtuel de couche 7 pour l'entrée ne peuvent pas avoir de paramètres de persistance différents, par exemple, `source_ip` pour la couche 4 et `cookie` pour la couche 7. Dans un tel scénario, les paramètres de persistance des deux serveurs virtuels doivent être les mêmes (`source_ip`, `cookie` ou `None`), ou l'un d'entre eux peut être `None` (l'autre peut alors être `source_ip` ou `cookie`). Exemple d'un tel scénario :

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress
spec:
  rules:
  - host: cafe.example.com
    http:
      paths:
      - path: /tea
        backend:
          serviceName: tea-svc
          servicePort: 80
-----
apiVersion: v1
kind: Service
metadata:
  name: tea-svc <==== same as the Ingress backend above
  labels:
    app: tea
spec:
  ports:
  - port: 80
    targetPort: 80
    protocol: TCP
    name: tcp
  selector:
    app: tea
  type: LoadBalancer
```

Partitionnement de routeur

NCP gère toujours la terminaison Edge TLS et les routes HTTP, et ignore les routes de relais TLS. TLS chiffre à nouveau les routes indépendamment de leurs espaces de noms ou de leurs étiquettes d'espaces de noms. Pour limiter un routeur OpenShift à la gestion des routes de rechiffrement et de relais TLS, procédez comme suit :

- Ajoutez un sélecteur d'étiquettes d'espaces de noms pour le routeur Openshift.

- Ajoutez une étiquette d'espace de noms à l'espace de noms cible.
- Créez des routes TLS de rechiffrement/relais dans l'espace de noms cible.

Par exemple, pour configurer un routeur avec un sélecteur d'étiquettes d'espaces de noms, exécutez la commande suivante (en partant du principe que le nom du compte de service de routeur est `router`) :

```
oc set env dc/router NAMESPACE_LABELS="router=r1"
```

Le routeur gérera dorénavant les routes depuis les espaces de noms sélectionnés. Pour que ce sélecteur corresponde à un espace de noms, exécutez la commande suivante (en partant du principe que l'espace de noms est appelé `ns1`) :

```
oc label namespace ns1 "router=r1"
```

Exemple d'équilibrage de charge de couche 7

Le fichier YAML suivant configure deux contrôleurs de réplication (tea-rc et coffee-rc), deux services (tea-svc et coffee-svc) et deux routes (cafe-route-multi et cafe-route) pour assurer l'équilibrage de charge de couche 7.

```
# RC
apiVersion: v1
kind: ReplicationController
metadata:
  name: tea-rc
spec:
  replicas: 2
  template:
    metadata:
       labels:
         app: tea
    spec:
      containers:
      - name: tea
        image: nginxdemos/hello
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: ReplicationController
metadata:
  name: coffee-rc
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: coffee
    spec:
      containers:
      - name: coffee
        image: nginxdemos/hello
```

```
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 80
---
# Services
apiVersion: v1
kind: Service
metadata:
  name: tea-svc
  labels:
    app: tea
spec:
  ports:
  - port: 80
    targetPort: 80
    protocol: TCP
    name: http
  selector:
    app: tea
---
apiVersion: v1
kind: Service
metadata:
  name: coffee-svc
  labels:
    app: coffee
spec:
  ports:
  - port: 80
    targetPort: 80
    protocol: TCP
    name: http
  selector:
    app: coffee
---
# Routes
apiVersion: v1
kind: Route
metadata:
  name: cafe-route-multi
spec:
  host: www.cafe.com
  path: /drinks
  to:
    kind: Service
    name: tea-svc
    weight: 1
  alternateBackends:
  - kind: Service
    name: coffee-svc
    weight: 2
---
apiVersion: v1
kind: Route
metadata:
```

```
    name: cafe-route
  spec:
    host: www.cafe.com
    path: /tea-svc
    to:
      kind: Service
      name: tea-svc
      weight: 1
```

Remarques supplémentaires

- Seule la terminaison Edge est prise en charge pour le trafic HTTPS.
- Un sous-domaine de caractères génériques est pris en charge. Par exemple, si `wildcardPolicy` est défini sur **`Sous-domaine`** et que le nom d'hôte est défini sur **`wildcard.example.com`**, toute demande à **`*.example.com`** est traitée.
- Si NCP génère une erreur lors du traitement d'un événement Route en raison d'une mauvaise configuration, vous devez corriger le fichier Route YAML, puis supprimer et recréer la ressource Route.
- NCP n'applique pas la propriété de nom d'hôte par espaces de noms.
- Un service LoadBalancer est pris en charge par cluster Kubernetes.
- NSX-T Data Center crée un serveur virtuel et un pool d'équilibrage de charge de couche 4 pour chaque port de service LoadBalancer. TCP et UDP sont pris en charge.
- L'équilibrage de charge NSX-T Data Center est fourni dans différentes tailles. Pour plus d'informations sur la configuration d'un équilibrage de charge NSX-T Data Center, consultez le *Guide d'administration de NSX-T Data Center*.

  Après la création de l'équilibrage de charge, la taille d'équilibrage de charge ne peut pas être modifiée en mettant à jour le fichier de configuration. Elle peut être modifiée au moyen de l'interface utilisateur ou de l'API.
- La mise à l'échelle automatique de l'équilibrage de charge de couche 4 est prise en charge. Si un service LoadBalancer Kubernetes est créé ou modifié pour qu'il requiert des serveurs virtuels supplémentaires et que l'équilibrage de charge de couche 4 existant n'a pas la capacité, un nouvel équilibrage de charge de couche 4 sera créé. NCP supprimera également les équilibrages de charge de couche 4 qui n'ont plus aucun serveur virtuel associé. Cette fonctionnalité est activée par défaut. Vous pouvez la désactiver en définissant `l4_lb_auto_scaling` sur **`false`** dans le ConfigMap NCP.

Exemple de script permettant de générer un certificat signé par une autorité de certification

Le script ci-dessous génère un certificat signé par une autorité de certification et une clé privée stockés dans les fichiers <filename>.crt et <finename>.key, respectivement. La commande `genrsa` génère une clé d'autorité de certification. La clé d'autorité de certification doit être chiffrée. Vous pouvez spécifier une méthode de chiffrement avec la commande `aes256`, par exemple.

```
#!/bin/bash
host="www.example.com"
filename=server

openssl genrsa -out ca.key 4096
openssl req -key ca.key -new -x509 -days 365 -sha256 -extensions v3_ca -out ca.crt -subj
"/C=US/ST=CA/L=Palo Alto/O=OS3/OU=Eng/CN=${host}"
openssl req -out ${filename}.csr -new -newkey rsa:2048 -nodes -keyout ${filename}.key -subj
"/C=US/ST=CA/L=Palo Alto/O=OS3/OU=Eng/CN=${host}"
openssl x509 -req -days 360 -in ${filename}.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out $
{filename}.crt -sha256
```

Placer le certificat et la clé par défaut dans l'espace NCP

Une fois le certificat et la clé privée générés, placez-les dans le répertoire `/etc/nsx-ujo` sur la machine virtuelle hôte. En supposant que les fichiers de certificat et de clé sont nommés `lb-default.crt` et `lb-default.key`, respectivement, modifiez `ncp-rc.yaml` afin que ces fichiers, stockés sur l'hôte, soient placés dans l'espace. Par exemple,

```
spec:
  ...
  containers:
  - name: nsx-ncp
    ...
    volumeMounts:
    ...
    - name: lb-default-cert
      # Mount path must match nsx_v3 option "lb_default_cert_path"
      mountPath: /etc/nsx-ujo/lb-default.crt
    - name: lb-priv-key
      # Mount path must match nsx_v3 option "lb_priv_key_path"
      mountPath: /etc/nsx-ujo/lb-default.key
  volumes:
  ...
  - name: lb-default-cert
    hostPath:
      path: /etc/nsx-ujo/lb-default.crt
  - name: lb-priv-key
    hostPath:
      path: /etc/nsx-ujo/lb-default.key
```

Administration de NSX Container Plug-in 5

Vous pouvez administrer NSX Container Plug-in à partir de l'interface utilisateur NSX Manager ou à partir de l'interface de ligne de commande.

Note Si une machine virtuelle d'hôte de conteneur s'exécute sur ESXi 6.5 et si la machine virtuelle est migrée via vMotion vers un autre hôte ESXi 6.5, les conteneurs en cours d'exécution sur l'hôte de conteneur perdent la connectivité aux conteneurs en cours d'exécution sur d'autres hôtes de conteneur. Vous pouvez résoudre ce problème en déconnectant et reconnectant la carte réseau virtuelle de l'hôte de conteneur. Ce problème ne se produit pas avec ESXi 6.5 Update 1 ou version ultérieure.

Hyperbus réserve l'ID de VLAN 4094 sur l'hyperviseur pour la configuration PVLAN et cet ID ne peut pas être modifié. Pour éviter tout conflit de VLAN, ne configurez pas les commutateurs logiques de VLAN ou les vmknics VTEP avec les mêmes ID de VLAN.

Ce chapitre contient les rubriques suivantes :

Gérer les blocs d'adresses IP à partir de l'interface utilisateur de NSX Manager

Vous pouvez ajouter, supprimer, modifier, afficher les détails et gérer les balises d'un bloc d'adresses IP à partir de l'interface utilisateur de NSX Manager.

Procédure

1. Dans un navigateur, connectez-vous au dispositif NSX Manager sur `https://<nsx-manager-IP-address-or-domain-name>`.
2. Accédez à **Mise en réseau > IPAM**.

   La liste des blocs d'adresses IP existants s'affiche.

3 Utilisez l'une des actions suivantes.

Option	Action
Ajouter un bloc d'adresses IP	Cliquez sur **Ajouter**.
Supprimer un ou plusieurs blocs d'adresses IP	Sélectionnez un ou plusieurs blocs d'adresses IP et cliquez sur **SUPPRIMER**.
Modifier un bloc d'adresses IP	Sélectionnez un bloc d'adresses IP et cliquez sur **MODIFIER**.
Afficher des détails sur un bloc d'adresses IP	Cliquez sur le nom du bloc d'adresses IP. Cliquez sur l'onglet **Présentation** pour voir des informations générales. Cliquez sur l'onglet **Sous-réseaux** pour voir les sous-réseaux du bloc d'adresses IP.
Gérer les balises d'un bloc d'adresses IP	Sélectionnez un bloc d'adresses IP et cliquez sur **ACTIONS > Gérer les balises**.

Vous ne pouvez pas supprimer un bloc d'adresses IP dont des sous-réseaux sont alloués.

Voir les sous-réseaux de bloc d'adresses IP à partir de l'interface utilisateur graphique de NSX Manager

Vous pouvez voir les sous-réseaux pour un bloc d'adresses IP à partir de l'interface utilisateur graphique de NSX Manager. Il n'est pas recommandé d'ajouter ou de supprimer des sous-réseaux de bloc d'adresses IP après avoir installé et exécuté NCP.

Procédure

1 Dans un navigateur, connectez-vous au dispositif NSX Manager sur `https://<nsx-manager-IP-address-or-domain-name>`.

2 Accédez à **Mise en réseau > IPAM**.

La liste des blocs d'adresses IP existants s'affiche.

3 Cliquez sur un nom de bloc d'adresses IP.

4 Cliquez sur l'onglet **Sous-réseaux**.

Ports logiques attachés par CIF

Les CIF (interfaces de conteneur) sont des interfaces réseau sur des conteneurs qui sont connectés à des ports logiques sur un commutateur. Ces ports sont appelés ports logiques attachés par CIF.

Vous pouvez gérer des ports logiques attachés par CIF depuis l'interface utilisateur de NSX Manager.

Gestion des ports logiques attachés par CIF

Accédez à **Mise en réseau > Commutation > Ports** pour voir tous les ports logiques, y compris les ports logiques attachés par CIF. Cliquez sur le lien d'attachement d'un port logique attaché par CIF pour afficher les informations de l'attachement Cliquez sur le lien du port logique pour ouvrir un volet de fenêtre avec quatre onglets : Présentation, Surveiller, Gérer et Éléments associés. Cliquez sur **Éléments associés > Ports logiques** pour voir le port logique associé sur un commutateur de liaison montante. Pour plus d'informations sur les ports de commutateur, consultez le *Guide d'administration de NSX-T*.

Outils de surveillance du réseau

Les outils suivants prennent en charge les ports logiques attachés par CIF. Pour plus d'informations sur ces outils, consultez le *Guide d'administration de NSX-T*.

- Traceflow
- Connexion de port
- IPFIX
- La mise en miroir de port distant à l'aide de l'encapsulation GRE d'un port de commutateur logique qui se connecte à un conteneur est prise en charge. Pour plus d'informations, reportez-vous à la section « Comprendre le profil de commutation de mise en miroir de ports » dans le *Guide d'administration de NSX-T*. Toutefois, la mise en miroir des ports CIF-VIF n'est pas prise en charge via l'interface utilisateur du gestionnaire.

Commandes d'interface de ligne de commande

Pour exécuter des commandes d'interface de ligne de commande, connectez-vous au conteneur NSX Container Plug-in, ouvrez un terminal et exécutez la commande `nsxcli`.

Vous pouvez également obtenir l'invite d'interface de ligne de commande en exécutant la commande suivante sur un nœud :

```
kubectl exec –it <pod name> nsxcli
```

Tableau 5-1. Commandes d'interface de ligne de commande pour le conteneur NCP

Type	vdmadmin
Statut	get ncp-master status
Statut	get ncp-nsx status
Statut	get ncp-watcher <watcher-name>
Statut	get ncp-watchers
Statut	get ncp-k8s-api-server status
Statut	check projects
Statut	check project <nom_projet>
Cache	get project-cache <project-name>
Cache	get project-caches
Cache	get namespace-cache <namespace-name>
Cache	get namespace-caches
Cache	get pod-cache <pod-name>
Cache	get pod-caches
Cache	get ingress-caches

Tableau 5-1. Commandes d'interface de ligne de commande pour le conteneur NCP (Suite)

Type	vdmadmin
Cache	get ingress-cache <nom_entrée>
Cache	get ingress-controllers
Cache	get ingress-controller <nom_contrôleur_entrée>
Cache	get network-policy-caches
Cache	get network-policy-cache <nom_espace>
Support	get ncp-log file <filename>
Support	get ncp-log-level
Support	set ncp-log-level <niveau_journal>
Support	get support-bundle file <filename>
Support	get node-agent-log file <filename>
Support	get node-agent-log file <filename> <node-name>

Tableau 5-2. Commandes d'interface de ligne de commande pour le conteneur de l'agent du nœud NSX

Type	vdmadmin
Statut	get node-agent-hyperbus status
Cache	get container-cache <nom-conteneur>
Cache	get container-caches

Tableau 5-3. Commandes d'interface de ligne de commande pour le conteneur du proxy Kube NSX

Type	vdmadmin
Statut	get ncp-k8s-api-server status
Statut	get kube-proxy-watcher <watcher-name>
Statut	get kube-proxy-watchers
Statut	dump ovs-flows

Commandes d'état pour le conteneur NCP

- Afficher l'état du nœud maître NCP

```
get ncp-master status
```

Exemple :

```
kubenode> get ncp-master status
This instance is not the NCP master
Current NCP Master id is a4h83eh1-b8dd-4e74-c71c-cbb7cc9c4c1c
Last master update at Wed Oct 25 22:46:40 2017
```

- Afficher l'état de la connexion entre NCP et NSX Manager

```
get ncp-nsx status
```

Exemple :

```
kubenode> get ncp-nsx status
NSX Manager status: Healthy
```

- Afficher l'état de l'observateur de l'entrée, l'espace de noms, l'espace et le service

```
get ncp-watcher <watcher-name>
get ncp-watchers
```

Exemple 1 :

```
kubenode> get ncp-watcher pod
    Average event processing time: 1174 msec (in past 3600-sec window)
    Current watcher started time: Mar 02 2017 10:47:35 PST
    Number of events processed: 1 (in past 3600-sec window)
    Total events processed by current watcher: 1
    Total events processed since watcher thread created: 1
    Total watcher recycle count: 0
    Watcher thread created time: Mar 02 2017 10:47:35 PST
    Watcher thread status: Up
```

Exemple 2 :

```
kubenode> get ncp-watchers
    pod:
        Average event processing time: 1145 msec (in past 3600-sec window)
        Current watcher started time: Mar 02 2017 10:51:37 PST
        Number of events processed: 1 (in past 3600-sec window)
        Total events processed by current watcher: 1
        Total events processed since watcher thread created: 1
        Total watcher recycle count: 0
        Watcher thread created time: Mar 02 2017 10:51:37 PST
        Watcher thread status: Up

    namespace:
        Average event processing time: 68 msec (in past 3600-sec window)
        Current watcher started time: Mar 02 2017 10:51:37 PST
        Number of events processed: 2 (in past 3600-sec window)
        Total events processed by current watcher: 2
        Total events processed since watcher thread created: 2
```

```
        Total watcher recycle count: 0
        Watcher thread created time: Mar 02 2017 10:51:37 PST
        Watcher thread status: Up

    ingress:
        Average event processing time: 0 msec (in past 3600-sec window)
        Current watcher started time: Mar 02 2017 10:51:37 PST
        Number of events processed: 0 (in past 3600-sec window)
        Total events processed by current watcher: 0
        Total events processed since watcher thread created: 0
        Total watcher recycle count: 0
        Watcher thread created time: Mar 02 2017 10:51:37 PST
        Watcher thread status: Up

    service:
        Average event processing time: 3 msec (in past 3600-sec window)
        Current watcher started time: Mar 02 2017 10:51:37 PST
        Number of events processed: 1 (in past 3600-sec window)
        Total events processed by current watcher: 1
        Total events processed since watcher thread created: 1
        Total watcher recycle count: 0
        Watcher thread created time: Mar 02 2017 10:51:37 PST
        Watcher thread status: Up
```

- Afficher l'état de la connexion entre NCP et Kubernetes API Server

```
get ncp-k8s-api-server status
```

Exemple :

```
kubenode> get ncp-k8s-api-server status
Kubernetes ApiServer status: Healthy
```

- Vérifier tous les projets ou un projet spécifique

```
check projects
check project <project-name>
```

Exemple :

```
kubenode> check projects
    default:
        Tier-1 link port for router 1b90a61f-0f2c-4768-9eb6-ea8954b4f327 is missing
        Switch 40a6829d-c3aa-4e17-ae8a-7f7910fdf2c6 is missing

    ns1:
        Router 8accc9cd-9883-45f6-81b3-0d1fb2583180 is missing

kubenode> check project default
    Tier-1 link port for router 1b90a61f-0f2c-4768-9eb6-ea8954b4f327 is missing
    Switch 40a6829d-c3aa-4e17-ae8a-7f7910fdf2c6 is missing
```

Commandes de cache pour le conteneur NCP

- Obtenir le cache interne pour des projets ou des espaces de noms

```
get project-cache <project-name>
get project-caches
get namespace-cache <namespace-name>
get namespace-caches
```

Exemple :

```
kubenode> get project-caches
    default:
        logical-router: 8accc9cd-9883-45f6-81b3-0d1fb2583180
        logical-switch:
            id: 9d7da647-27b6-47cf-9cdb-6e4f4d5a356d
            ip_pool_id: 519ff57f-061f-4009-8d92-3e6526e7c17e
            subnet: 10.0.0.0/24
            subnet_id: f75fd64c-c7b0-4b42-9681-fc656ae5e435

    kube-system:
        logical-router: 5032b299-acad-448e-a521-19d272a08c46
        logical-switch:
            id: 85233651-602d-445d-ab10-1c84096cc22a
            ip_pool_id: ab1c5b09-7004-4206-ac56-85d9d94bffa2
            subnet: 10.0.1.0/24
            subnet_id: 73e450af-b4b8-4a61-a6e3-c7ddd15ce751

    testns:
        ext_pool_id: 346a0f36-7b5a-4ecc-ad32-338dcb92316f
        labels:
            ns: myns
            project: myproject
        logical-router: 4dc8f8a9-69b4-4ff7-8fb7-d2625dc77efa
        logical-switch:
            id: 6111a99a-6e06-4faa-a131-649f10f7c815
            ip_pool_id: 51ca058d-c3dc-41fd-8f2d-e69006ab1b3d
            subnet: 50.0.2.0/24
            subnet_id: 34f79811-bd29-4048-a67d-67ceac97eb98
        project_nsgroup: 9606afee-6348-4780-9dbe-91abfd23e475
        snat_ip: 4.4.0.3

kubenode> get project-cache default
    logical-router: 8accc9cd-9883-45f6-81b3-0d1fb2583180
    logical-switch:
        id: 9d7da647-27b6-47cf-9cdb-6e4f4d5a356d
        ip_pool_id: 519ff57f-061f-4009-8d92-3e6526e7c17e
        subnet: 10.0.0.0/24
        subnet_id: f75fd64c-c7b0-4b42-9681-fc656ae5e435

kubenode> get namespace-caches
    default:
        logical-router: 8accc9cd-9883-45f6-81b3-0d1fb2583180
```

```
        logical-switch:
            id: 9d7da647-27b6-47cf-9cdb-6e4f4d5a356d
            ip_pool_id: 519ff57f-061f-4009-8d92-3e6526e7c17e
            subnet: 10.0.0.0/24
            subnet_id: f75fd64c-c7b0-4b42-9681-fc656ae5e435

    kube-system:
        logical-router: 5032b299-acad-448e-a521-19d272a08c46
        logical-switch:
            id: 85233651-602d-445d-ab10-1c84096cc22a
            ip_pool_id: ab1c5b09-7004-4206-ac56-85d9d94bffa2
            subnet: 10.0.1.0/24
            subnet_id: 73e450af-b4b8-4a61-a6e3-c7ddd15ce751

    testns:
        ext_pool_id: 346a0f36-7b5a-4ecc-ad32-338dcb92316f
        labels:
            ns: myns
            project: myproject
        logical-router: 4dc8f8a9-69b4-4ff7-8fb7-d2625dc77efa
        logical-switch:
            id: 6111a99a-6e06-4faa-a131-649f10f7c815
            ip_pool_id: 51ca058d-c3dc-41fd-8f2d-e69006ab1b3d
            subnet: 50.0.2.0/24
            subnet_id: 34f79811-bd29-4048-a67d-67ceac97eb98
        project_nsgroup: 9606afee-6348-4780-9dbe-91abfd23e475
        snat_ip: 4.4.0.3

kubenode> get namespace-cache default
    logical-router: 8accc9cd-9883-45f6-81b3-0d1fb2583180
    logical-switch:
        id: 9d7da647-27b6-47cf-9cdb-6e4f4d5a356d
        ip_pool_id: 519ff57f-061f-4009-8d92-3e6526e7c17e
        subnet: 10.0.0.0/24
        subnet_id: f75fd64c-c7b0-4b42-9681-fc656ae5e435
```

- Obtenir le cache interne pour des espaces

```
get pod-cache <pod-name>
get pod-caches
```

Exemple :

```
kubenode> get pod-caches
    nsx.default.nginx-rc-uq2lv:
        cif_id: 2af9f734-37b1-4072-ba88-abbf935bf3d4
        gateway_ip: 10.0.0.1
        host_vif: d6210773-5c07-4817-98db-451bd1f01937
        id: 1c8b5c52-3795-11e8-ab42-005056b198fb
        ingress_controller: False
        ip: 10.0.0.2/24
        labels:
            app: nginx
        mac: 02:50:56:00:08:00
```

```
        port_id: d52c833a-f531-4bdf-bfa2-e8a084a8d41b
        vlan: 1

    nsx.testns.web-pod-1:
        cif_id: ce134f21-6be5-43fe-afbf-aaca8c06b5cf
        gateway_ip: 50.0.2.1
        host_vif: d6210773-5c07-4817-98db-451bd1f01937
        id: 3180b521-270e-11e8-ab42-005056b198fb
        ingress_controller: False
        ip: 50.0.2.3/24
        labels:
            app: nginx-new
            role: db
            tier: cache
        mac: 02:50:56:00:20:02
        port_id: 81bc2b8e-d902-4cad-9fc1-aabdc32ecaf8
        vlan: 3

kubenode> get pod-cache nsx.default.nginx-rc-uq2lv
    cif_id: 2af9f734-37b1-4072-ba88-abbf935bf3d4
    gateway_ip: 10.0.0.1
    host_vif: d6210773-5c07-4817-98db-451bd1f01937
    id: 1c8b5c52-3795-11e8-ab42-005056b198fb
    ingress_controller: False
    ip: 10.0.0.2/24
    labels:
        app: nginx
    mac: 02:50:56:00:08:00
    port_id: d52c833a-f531-4bdf-bfa2-e8a084a8d41b
    vlan: 1
```

- Obtenir les caches de stratégie réseau ou un cache spécifique

```
get network-policy caches
get network-policy-cache <network-policy-name>
```

Exemple :

```
kubenode> get network-policy-caches
    nsx.testns.allow-tcp-80:
        dest_labels: None
        dest_pods:
            50.0.2.3
        match_expressions:
            key: tier
            operator: In
            values:
                cache
        name: allow-tcp-80
        np_dest_ip_set_ids:
            22f82d76-004f-4d12-9504-ce1cb9c8aa00
            np_except_ip_set_ids:
        np_ip_set_ids:
            14f7f825-f1a0-408f-bbd9-bb2f75d44666
```

```
        np_isol_section_id: c8d93597-9066-42e3-991c-c550c46b2270
        np_section_id: 04693136-7925-44f2-8616-d809d02cd2a9
        ns_name: testns
        src_egress_rules: None
        src_egress_rules_hash: 97d170e1550eee4afc0af065b78cda302a97674c
        src_pods:
            50.0.2.0/24
        src_rules:
            from:
                namespaceSelector:
                    matchExpressions:
                        key: tier
                        operator: DoesNotExist
                    matchLabels:
                        ns: myns
            ports:
                port: 80
                protocol: TCP
        src_rules_hash: e4ea7b8d91c1e722670a59f971f8fcc1a5ac51f1


kubenode> get network-policy-cache nsx.testns.allow-tcp-80
    dest_labels: None
    dest_pods:
        50.0.2.3
    match_expressions:
        key: tier
        operator: In
        values:
            cache
    name: allow-tcp-80
    np_dest_ip_set_ids:
        22f82d76-004f-4d12-9504-ce1cb9c8aa00
        np_except_ip_set_ids:
    np_ip_set_ids:
        14f7f825-f1a0-408f-bbd9-bb2f75d44666
    np_isol_section_id: c8d93597-9066-42e3-991c-c550c46b2270
    np_section_id: 04693136-7925-44f2-8616-d809d02cd2a9
    ns_name: testns
    src_egress_rules: None
    src_egress_rules_hash: 97d170e1550eee4afc0af065b78cda302a97674c
    src_pods:
        50.0.2.0/24
    src_rules:
        from:
            namespaceSelector:
                matchExpressions:
                    key: tier
                    operator: DoesNotExist
                matchLabels:
                    ns: myns
```

```
        ports:
            port: 80
            protocol: TCP
    src_rules_hash: e4ea7b8d91c1e722670a59f971f8fcc1a5ac51f1
```

Commandes de support pour le conteneur NCP

- Enregistrer le bundle de support NCP dans le magasin de fichiers

  Le bundle de support est constitué des fichiers journaux de tous les conteneurs d'espaces avec l'étiquette **`tier:nsx-networking`**. Le fichier de bundle est au format tgz et enregistré dans le répertoire du magasin de fichiers par défaut d'interface de ligne de commande `/var/vmware/nsx/file-store`. Vous pouvez utiliser la commande de magasin de fichiers d'interface de ligne de commande pour copier le fichier de bundle sur un site distant.

  ```
  get support-bundle file <filename>
  ```

  Exemple :

  ```
  kubenode>get support-bundle file foo
  Bundle file foo created in tgz format
  kubenode>copy file foo url scp://nicira@10.0.0.1:/tmp
  ```

- Enregistrer les journaux NCP dans le magasin de fichiers

  Le fichier journal est enregistré au format tgz dans le répertoire du magasin de fichiers par défaut d'interface de ligne de commande `/var/vmware/nsx/file-store`. Vous pouvez utiliser la commande de magasin de fichiers d'interface de ligne de commande pour copier le fichier de bundle sur un site distant.

  ```
  get ncp-log file <filename>
  ```

  Exemple :

  ```
  kubenode>get ncp-log file foo
  Log file foo created in tgz format
  ```

- Enregistrer les journaux de l'agent de nœud dans le magasin de fichiers

  Enregistrez les journaux de l'agent de nœud d'un seul nœud ou de tous les nœuds. Les journaux sont enregistrés au format tgz dans le répertoire du magasin de fichiers par défaut d'interface de ligne de commande `/var/vmware/nsx/file-store`. Vous pouvez utiliser la commande de magasin de fichiers d'interface de ligne de commande pour copier le fichier de bundle sur un site distant.

  ```
  get node-agent-log file <filename>
  get node-agent-log file <filename> <node-name>
  ```

Exemple :

```
kubenode>get node-agent-log file foo
Log file foo created in tgz format
```

- Obtenir et définir le niveau de journalisation

  Les niveaux de journalisation disponibles sont `NOTSET`, `DEBUG`, `INFO`, `WARNING`, `ERROR` et `CRITICAL`.

  ```
  get ncp-log-level
  set ncp-log-level <log level>
  ```

  Exemple :

  ```
  kubenode>get ncp-log-level
  NCP log level is INFO

  kubenode>set ncp-log-level DEBUG
  NCP log level is changed to DEBUG
  ```

Commandes d'état pour le conteneur de l'agent du nœud NSX

- Affichez l'état de la connexion entre l'agent de nœud et HyperBus sur ce nœud.

  ```
  get node-agent-hyperbus status
  ```

  Exemple :

  ```
  kubenode> get node-agent-hyperbus status
  HyperBus status: Healthy
  ```

Commandes de cache pour le conteneur de l'agent du nœud NSX

- Obtenir le cache interne pour les conteneurs d'agents du nœud NSX.

  ```
  get container-cache <container-name>
  get container-caches
  ```

  Exemple 1 :

  ```
  kubenode> get container-cache cif104
      ip: 192.168.0.14/32
      mac: 50:01:01:01:01:14
      gateway_ip: 169.254.1.254/16
      vlan_id: 104
  ```

Exemple 2 :

```
kubenode> get container-caches
    cif104:
        ip: 192.168.0.14/32
        mac: 50:01:01:01:01:14
        gateway_ip: 169.254.1.254/16
        vlan_id: 104
```

Commandes d'état pour le conteneur du proxy Kube NSX

- Afficher l'état de la connexion entre Kube Proxy et Kubernetes API Server

```
get ncp-k8s-api-server status
```

Exemple :

```
kubenode> get kube-proxy-k8s-api-server status
Kubernetes ApiServer status: Healthy
```

- Afficher l'état de l'observateur Kube Proxy

```
get kube-proxy-watcher <watcher-name>
get kube-proxy-watchers
```

Exemple 1 :

```
kubenode> get kube-proxy-watcher endpoint
    Average event processing time: 15 msec (in past 3600-sec window)
    Current watcher started time: May 01 2017 15:06:24 PDT
    Number of events processed: 90 (in past 3600-sec window)
    Total events processed by current watcher: 90
    Total events processed since watcher thread created: 90
    Total watcher recycle count: 0
    Watcher thread created time: May 01 2017 15:06:24 PDT
    Watcher thread status: Up
```

Exemple 2 :

```
kubenode> get kube-proxy-watchers
    endpoint:
        Average event processing time: 15 msec (in past 3600-sec window)
        Current watcher started time: May 01 2017 15:06:24 PDT
        Number of events processed: 90 (in past 3600-sec window)
        Total events processed by current watcher: 90
        Total events processed since watcher thread created: 90
        Total watcher recycle count: 0
        Watcher thread created time: May 01 2017 15:06:24 PDT
        Watcher thread status: Up

     service:
```

```
        Average event processing time: 8 msec (in past 3600-sec window)
        Current watcher started time: May 01 2017 15:06:24 PDT
        Number of events processed: 2 (in past 3600-sec window)
        Total events processed by current watcher: 2
        Total events processed since watcher thread created: 2
        Total watcher recycle count: 0
        Watcher thread created time: May 01 2017 15:06:24 PDT
        Watcher thread status: Up
```

- Vider les flux OVS sur un nœud

```
dump ovs-flows
```

Exemple :

```
kubenode> dump ovs-flows
    NXST_FLOW reply (xid=0x4):
    cookie=0x0, duration=8.876s, table=0, n_packets=0, n_bytes=0, idle_age=8, priority=100,ip
actions=ct(table=1)
    cookie=0x0, duration=8.898s, table=0, n_packets=0, n_bytes=0, idle_age=8, priority=0
actions=NORMAL
    cookie=0x0, duration=8.759s, table=1, n_packets=0, n_bytes=0, idle_age=8,
priority=100,tcp,nw_dst=10.96.0.1,tp_dst=443 actions=mod_tp_dst:443
    cookie=0x0, duration=8.719s, table=1, n_packets=0, n_bytes=0, idle_age=8,
priority=100,ip,nw_dst=10.96.0.10 actions=drop
    cookie=0x0, duration=8.819s, table=1, n_packets=0, n_bytes=0, idle_age=8,
priority=90,ip,in_port=1 actions=ct(table=2,nat)
    cookie=0x0, duration=8.799s, table=1, n_packets=0, n_bytes=0, idle_age=8, priority=80,ip
actions=NORMAL
    cookie=0x0, duration=8.856s, table=2, n_packets=0, n_bytes=0, idle_age=8, actions=NORMAL
```

Codes d'erreur

Cette section répertorie les codes d'erreur renvoyés par les différents composants.

Codes d'erreur NCP

Code d'erreur	Description
NCP00001	Configuration non valide
NCP00002	Échec de l'initialisation
NCP00003	État non valide
NCP00004	Adaptateur non valide
NCP00005	Certificat introuvable
NCP00006	Jeton introuvable
NCP00007	Configuration de NSX non valide
NCP00008	Balise NSX non valide
NCP00009	Échec de la connexion à NSX

Code d'erreur	Description
NCP00010	Balise de nœud introuvable
NCP00011	Port de commutateur logique du nœud non valide
NCP00012	Échec de la mise à jour de la VIF parent
NCP00013	VLAN épuisé
NCP00014	Échec de la libération du VLAN
NCP00015	Pool d'adresses IP épuisé
NCP00016	Échec de la libération de l'adresse IP
NCP00017	Bloc d'adresses IP épuisé
NCP00018	Échec de la création du sous-réseau IP
NCP00019	Échec de la suppression du sous-réseau IP
NCP00020	Échec de la création du pool d'adresses IP
NCP00021	Échec de la suppression du pool d'adresses IP
NCP00022	Échec de la création du routeur logique
NCP00023	Échec de la mise à jour du routeur logique
NCP00024	Échec de la suppression du routeur logique
NCP00025	Échec de la création du commutateur logique

Code d'erreur	Description
NCP00026	Échec de la mise à jour du commutateur logique
NCP00027	Échec de la suppression du commutateur logique
NCP00028	Échec de la création du port de routeur logique
NCP00029	Échec de la suppression du port de routeur logique
NCP00030	Échec de la création du port de commutateur logique
NCP00031	Échec de la mise à jour du port de commutateur logique
NCP00032	Échec de la suppression du port de commutateur logique
NCP00033	Stratégie réseau introuvable
NCP00034	Échec de la création du pare-feu
NCP00035	Échec de la lecture du pare-feu
NCP00036	Échec de la mise à jour du pare-feu
NCP00037	Échec de la suppression du pare-feu
NCP00038	Plusieurs pare-feu trouvés
NCP00039	Échec de la création du NSGroup
NCP00040	Échec de la suppression du NSGroup
NCP00041	Échec de la création de l'ensemble d'adresses IP
NCP00042	Échec de la mise à jour de l'ensemble d'adresses IP

Code d'erreur	Description
NCP00043	Échec de la suppression de l'ensemble d'adresses IP
NCP00044	Échec de la création de la règle SNAT
NCP00045	Échec de la suppression de la règle SNAT
NCP00046	Échec de la connexion à l'API d'adaptateur
NCP00047	Exception de l'observateur d'adaptateur
NCP00048	Échec de la suppression du service d'équilibrage de charge
NCP00049	Échec de la création du serveur virtuel d'équilibrage de charge
NCP00050	Échec de la mise à jour du serveur virtuel d'équilibrage de charge

Code d'erreur	Description
NCP00051	Échec de la suppression du serveur virtuel d'équilibrage de charge
NCP00052	Échec de la création du pool d'équilibrage de charge
NCP00053	Échec de la mise à jour du pool d'équilibrage de charge
NCP00054	Échec de la suppression du pool d'équilibrage de charge
NCP00055	Échec de la création de la règle d'équilibrage de charge
NCP00056	Échec de la mise à jour de la règle d'équilibrage de charge
NCP00057	Échec de la suppression de la règle d'équilibrage de charge
NCP00058	Échec de la libération de l'adresse IP de pool d'équilibrage de charge
NCP00059	Serveur virtuel d'équilibrage de charge et association de service introuvables
NCP00060	Échec de la mise à jour du NSGroup
NCP00061	Échec de l'obtention des règles de pare-feu
NCP00062	Aucun critère de NSGroup
NCP00063	VM de nœud introuvable
NCP00064	VIF de nœud introuvable
NCP00065	Échec de l'importation de certificat
NCP00066	Échec de l'annulation de l'importation de certificat
NCP00067	Échec de la mise à jour de la liaison SSL
NCP00068	Profil SSL introuvable
NCP00069	Pool d'adresses IP introuvable
NCP00070	Cluster Edge T0 introuvable
NCP00071	Échec de la mise à jour du pool d'adresses IP
NCP00072	Échec du répartiteur
NCP00073	Échec de la suppression de la règle NAT
NCP00074	Échec de l'obtention du port de routeur logique
NCP00075	Échec de la validation de la configuration de NSX

Code d'erreur	Description
NCP00076	Échec de la mise à jour de la règle SNAT
NCP00077	Chevauchement de la règle SNAT
NCP00078	Échec de l'ajout des points de terminaison d'équilibrage de charge
NCP00079	Échec de la mise à jour des points de terminaison d'équilibrage de charge
NCP00080	Échec de la création du pool de règles d'équilibrage de charge
NCP00081	Serveur virtuel d'équilibrage de charge introuvable
NCP00082	Échec de la lecture de l'ensemble d'adresses IP
NCP00083	Échec de l'obtention du pool SNAT
NCP00084	Échec de la création du service d'équilibrage de charge
NCP00085	Échec de la mise à jour du service d'équilibrage de charge
NCP00086	Échec de la mise à jour du port de routeur logique
NCP00087	Échec de l'initialisation de l'équilibrage de charge
NCP00088	Pool d'adresses IP non unique
NCP00089	Erreur de synchronisation du cache d'équilibrage de charge de couche 7
NCP00090	Erreur : le pool d'équilibrage de charge n'existe pas
NCP00091	Erreur d'initialisation du cache de la règle d'équilibrage de charge
NCP00092	Échec du processus SNAT
NCP00093	Erreur de certificat par défaut d'équilibrage de charge
NCP00094	Échec de la suppression du point de terminaison d'équilibrage de charge
NCP00095	Projet introuvable
NCP00096	Accès au pool refusé
NCP00097	Échec de l'obtention d'un service d'équilibrage de charge
NCP00098	Échec de la création d'un service d'équilibrage de charge
NCP00099	Erreur de synchronisation du cache de pool d'équilibrage de charge

Codes d'erreur de l'agent de nœud NSX

Code d'erreur	Description
NCP01001	Liaison montante OVS introuvable
NCP01002	Adresse MAC de l'hôte introuvable
NCP01003	Échec de la création du port OVS
NCP01004	Aucune configuration de l'espace
NCP01005	Échec de la configuration de l'espace
NCP01006	Échec de l'annulation de la configuration de l'espace
NCP01007	Socket CNI introuvable

Code d'erreur	Description
NCP01008	Échec de la connexion à CNI
NCP01009	Incompatibilité de la version de CNI
NCP01010	Échec de la réception du message CNI
NCP01011	Échec de la transmission du message CNI
NCP01012	Échec de la connexion à Hyperbus
NCP01013	Incompatibilité de la version d'Hyperbus
NCP01014	Échec de la réception du message Hyperbus
NCP01015	Échec de la transmission du message Hyperbus
NCP01016	Échec de l'envoi du paquet GARP
NCP01017	Échec de la configuration de l'interface

Codes d'erreur de nsx-kube-proxy

Code d'erreur	Description
NCP02001	Port de la passerelle de proxy non valide
NCP02002	Échec de la commande de proxy
NCP02003	Échec de la validation du proxy

Codes d'erreur de la CLI

Code d'erreur	Description
NCP03001	Échec du démarrage de la CLI
NCP03002	Échec de la création du socket de la CLI
NCP03003	Exception de socket de la CLI
NCP03004	Demande du client de CLI non valide
NCP03005	Échec de la transmission du serveur de CLI
NCP03006	Échec de la réception du serveur de CLI
NCP03007	Échec de l'exécution de la commande de CLI

Codes d'erreur Kubernetes

Code d'erreur	Description
NCP05001	Échec de la connexion à Kubernetes
NCP05002	Configuration de Kubernetes non valide
NCP05003	Échec de la demande Kubernetes
NCP05004	Clé Kubernetes introuvable

Code d'erreur	Description
NCP05005	Type Kubernetes introuvable
NCP05006	Exception de l'observateur Kubernetes
NCP05007	Longueur de la ressource Kubernetes non valide
NCP05008	Type de ressource Kubernetes non valide
NCP05009	Échec du handle de ressource Kubernetes
NCP05010	Échec du handle de service Kubernetes
NCP05011	Échec du handle de point de terminaison Kubernetes
NCP05012	Échec du handle d'entrée Kubernetes
NCP05013	Échec du handle de stratégie réseau Kubernetes
NCP05014	Échec du handle de nœud Kubernetes
NCP05015	Échec du handle d'espace de noms Kubernetes
NCP05016	Échec du handle d'espace Kubernetes
NCP05017	Échec du handle de secret Kubernetes
NCP05018	Échec du serveur principal Kubernetes par défaut
NCP05019	Expression de correspondance Kubernetes non prise en charge
NCP05020	Échec de la mise à jour de l'état Kubernetes
NCP05021	Échec de la mise à jour de l'annotation Kubernetes
NCP05022	Cache d'espace de noms Kubernetes introuvable
NCP05023	Secret Kubernetes introuvable
NCP05024	Serveur principal Kubernetes par défaut en cours d'utilisation
NCP05025	Échec du handle de service Kubernetes LoadBalancer

Codes d'erreur OpenShift

Code d'erreur	Description
NCP07001	Échec du handle de la route OC
NCP07002	Échec de la mise à jour de l'état de la route OC