

NSX Container Plug-In für OpenShift – Installations- und Administratorhandbuch

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



vmware®

Die aktuellste technische Dokumentation finden Sie auf der VMware-Website unter:

<https://docs.vmware.com/de/>

Die VMware-Website enthält auch die neuesten Produkt-Updates.

Falls Sie Anmerkungen zu dieser Dokumentation haben, senden Sie diese an:

docfeedback@vmware.com

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

VMware Global, Inc.
Zweigniederlassung Deutschland
Willy-Brandt-Platz 2
81829 München
Germany
Tel.: +49 (0) 89 3706 17 000
Fax: +49 (0) 89 3706 17 333
www.vmware.com/de

Copyright © 2017–2019 VMware, Inc. Alle Rechte vorbehalten. [Urheberrechts- und Markenhinweise](#).

Inhalt

NSX-T Container Plug-In für OpenShift – Installations- und Administratorhandbuch 4

1 Übersicht über das NSX-T Container Plug-In 5

Kompatibilitätsanforderungen 6

Überblick über die Installation 6

Upgrade von NCP 7

2 Einrichten von NSX-T-Ressourcen 8

Konfigurieren von NSX-T-Ressourcen 8

3 Installieren von NCP 12

Systemvoraussetzungen 12

Vorbereiten der Ansible-Hostdatei 13

4 Lastausgleich 18

Konfigurieren des Lastausgleichs 18

5 Verwalten von NSX Container Plug-in 26

Verwalten von IP-Blöcken über die NSX Manager -GUI 26

Anzeigen von IP-Block-Subnetzen über die GUI von NSX Manager 27

CIF-verknüpfte logische Ports 27

CLI-Befehle 28

Fehlercodes 39

NSX-T Container Plug-In für OpenShift – Installations- und Administratorhandbuch

Dieses Handbuch beschreibt die Installation und Verwaltung von NSX Container Plug-in (NCP) für die Bereitstellung der Integration zwischen NSX-T Data Center und OpenShift.

Zielgruppe

Dieses Handbuch ist für die System- und Netzwerkadministratoren bestimmt. Es wird vorausgesetzt, dass Sie mit der Installation und Verwaltung von NSX-T Data Center und OpenShift vertraut sind.

VMware Technical Publications – Glossar

VMware Technical Publications enthält ein Glossar mit Begriffen, die Ihnen möglicherweise unbekannt sind. Definitionen von Begriffen, die in der technischen Dokumentation von VMware verwendet werden, finden Sie unter <http://www.vmware.com/support/pubs>.

Übersicht über das NSX-T Container Plug-In

1

NSX Container Plug-in (NCP) stellt eine Integration zwischen NSX-T Data Center und der Container-Orchestrierung wie z. B. Kubernetes sowie die Integration zwischen NSX-T Data Center und Container-basierten PaaS-Softwareprodukten (Plattform als Dienst), wie z. B. OpenShift, bereit. Dieses Handbuch beschreibt das Einrichten von NCP mit OpenShift.

Die Hauptkomponente von NCP wird in einem Container ausgeführt und kommuniziert mit NSX Manager und mit der OpenShift-Steuerungsebene. NCP überwacht Änderungen an den Containern und anderen Ressourcen und verwaltet Netzwerkressourcen wie logische Ports, Switches, Router und Sicherheitsgruppen für die Container per Aufruf der NSX API.

Das NSX CNI-Plugin wird auf jedem OpenShift-Knoten ausgeführt. Es überwacht Ereignisse im Container-Lebenszyklus, verbindet eine Containerschnittstelle mit dem Gast-vSwitch und programmiert den Gast-vSwitch für die Kennzeichnung und Weiterleitung des Containerdatenverkehrs zwischen den Containerschnittstellen und der VNIC.

NCP bietet die folgenden Funktionen:

- Es erstellt automatisch eine logische NSX-T-Topologie für einen OpenShift-Cluster und erstellt ein separates logisches Netzwerk für jeden OpenShift-Namespace.
- Es verbindet OpenShift-Pods mit dem logischen Netzwerk und weist IP- und MAC-Adressen zu.
- Es unterstützt Netzwerkadressübersetzung (Network Address Translation – NAT) und weist eine separate SNAT-IP für jeden OpenShift-Namespace zu.

Hinweis Bei der Konfiguration von NAT darf die Gesamtzahl der übersetzten IPs 1000 nicht überschreiten.

- Es implementiert OpenShift-Netzwerkrichtlinien mit verteilter NSX-T-Firewall.
 - Unterstützung für Ingress- und Egress-Netzwerkrichtlinien.
 - Unterstützung für IPBlock-Selektor in Netzwerkrichtlinien.
 - Unterstützung für matchLabels und matchExpression beim Angeben von Bezeichnungsselektoren für Netzwerkrichtlinien.
- Es implementiert eine OpenShift-Route mit NSX-T-Load Balancer der Schicht 7.
 - Unterstützung für HTTP-Route und HTTPS-Route mit TLS-Edge-Beendigung.

- Unterstützung für Routen mit alternativen Backends und Unterdomänen als Platzhalter.
- Es erstellt auf dem logischen NSX-T-Switch-Port Tags für den Namespace, den Pod-Namen und die Bezeichnungen eines Pods und lässt zu, dass der Administrator die NSX-T Data Center-Sicherheitsgruppen und -richtlinien basierend auf den Tags festlegt.

In dieser Version unterstützt NCP einen einzelnen OpenShift-Cluster.

Dieses Kapitel enthält die folgenden Themen:

- [Kompatibilitätsanforderungen](#)
- [Überblick über die Installation](#)
- [Upgrade von NCP](#)

Kompatibilitätsanforderungen

NSX Container Plug-in (NCP) weist die folgenden Kompatibilitätsanforderungen auf.

Softwareprodukt	Version
NSX-T Data Center	2.3, 2.4
Hypervisor für Container-Host-VMs	<ul style="list-style-type: none"> ■ Unterstützte vSphere-Version ■ RHEL KVM 7.4, 7.5, 7.6
Container-Host-Betriebssystem	RHEL 7.4, 7.5, 7.6
Platform as a Service	OpenShift 3.10, 3.11
Container-Host-Open vSwitch	2.10.2 (im Lieferumfang von NSX-T Data Center 2.4 enthalten)

Überblick über die Installation

Das Installieren und Konfigurieren von NCP umfasst die folgenden Schritte. Zur erfolgreichen Durchführung der Schritte müssen Sie mit NSX-T Data Center sowie der Installation und Verwaltung von OpenShift vertraut sein.

- 1 Installieren Sie NSX-T Data Center.
- 2 Erstellen Sie eine Overlay-Transportzone.
- 3 Erstellen Sie einen logischen Overlay-Switch, und verbinden Sie die Knoten mit dem Switch.
- 4 Erstellen Sie einen logischen Tier-0-Router.
- 5 Erstellen Sie IP-Blöcke für die Pods.
- 6 Erstellen Sie IP-Pools für SNAT (Source Network Address Translation).
- 7 Bereiten Sie die Ansible-Hostdatei vor.
- 8 Installieren von NCP und OpenShift mit einem einzelnen Playbook

Upgrade von NCP

In diesem Abschnitt wird beschrieben, wie ein Upgrade von NCP auf 2.4.0 vorgenommen wird.

Verfahren

- 1 Führen Sie ein Upgrade für das CNI-RPM-Paket, das DaemonSet des NSX-Knotenagents und den NCP-ReplicationController durch.
- 2 Bereiten Sie die Ansible-Hostdatei vor.

Auf jedem Knoten muss der Parameter `openshift_node_group_name` angegeben sein. Beispiel:

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

- 3 (Optional) Konfigurieren Sie den Lastausgleich.

Fügen Sie einen Schritt hinzu, um einen anderen IP-Pool für externe IP-Adressen für den LoadBalancer-Dienst anzugeben. Beispiel:

```
external_ip_pools_lb = <nsx ip pool name>
```

Einrichten von NSX-T-Ressourcen

2

Zum Bereitstellen der Netzwerkfunktionen für OpenShift-Knoten müssen NSX-T Data Center-Ressourcen erstellt werden.

Konfigurieren von NSX-T-Ressourcen

Zu den zu konfigurierenden NSX-T Data Center-Ressourcen gehören eine Overlay-Transportzone, ein logischer Tier-0-Router, ein logischer Switch zum Verbinden der virtuellen Maschinen des Knotens, IP-Blöcke für Kubernetes-Knoten und ein IP-Pool für SNAT.

Wichtig Wenn die Ausführung mit NSX-T Data Center 2.4 oder höher erfolgt, müssen Sie mithilfe der Registerkarte **Netzwerk und Sicherheit – Erweitert** NSX-T-Ressourcen konfigurieren.

In der NCP-Konfigurationsdatei `ncp.ini` sind die NSX-T Data Center-Ressourcen unter Verwendung ihrer UUIDs oder Namen angegeben.

Overlay-Transportzone

Melden Sie sich bei NSX Manager an und suchen Sie nach der für Containernetzwerke verwendeten Overlay-Transportzone oder erstellen Sie eine neue.

Geben Sie eine Overlay-Transportzone für einen Cluster an, indem Sie die Option `overlay_tz` im Abschnitt `[nsx_v3]` der Datei `ncp.ini` festlegen. Dieser Schritt ist optional. Wenn Sie `overlay_tz` nicht festlegen, ruft NCP die ID der Overlay-Transportzone automatisch vom Tier-0-Router ab.

Logisches Tier-0-Routing

Melden Sie sich bei NSX Manager an und suchen Sie nach dem für Containernetzwerke verwendeten Router oder erstellen Sie einen neuen.

Geben Sie einen logischen Tier-0-Router für einen Cluster an, indem Sie die Option `tier0_router` im Abschnitt `[nsx_v3]` der Datei `ncp.ini` festlegen.

Hinweis Der Router muss im Aktiv/Standby-Modus erstellt werden.

Logischer Switch

Die vom Knoten für den Netzwerkdatenverkehr verwendeten vNICs müssen mit einem logischen Overlay-Switch verbunden sein. Die Verwaltungsschnittstelle des Knotens muss nicht zwingend mit NSX-T Data Center verbunden sein, obwohl dies die Einrichtung erleichtert. Sie können einen logischen Switch erstellen, indem Sie sich bei NSX Manager anmelden. Erstellen Sie auf dem Switch logische Ports und hängen Sie die vNICs des Knotens daran an. Die logischen Ports müssen die folgenden Tags aufweisen:

- Tag: <cluster_name>, Geltungsbereich: ncp/cluster
- Tag: <node_name>, Geltungsbereich: ncp/node_name

Der Wert für <cluster_name> muss mit dem Wert der Option `cluster` im Abschnitt `[coe]` von `ncp.ini` übereinstimmen.

IP-Blöcke für Kubernetes-Pods

Melden Sie sich bei NSX Manager an und erstellen Sie einen oder mehrere IP-Blöcke. Geben Sie den IP-Block im CIDR-Format an.

Geben Sie IP-Blöcke für Kubernetes-Pods an, indem Sie die Option `container_ip_blocks` im Abschnitt `[nsx_v3]` der Datei `ncp.ini` festlegen.

Sie können IP-Blöcke auch speziell für Nicht-SNAT-Namespace erstellen.

Geben Sie Nicht-SNAT-IP-Blöcke ein, indem Sie die Option `no_snat_ip_blocks` im Abschnitt `[nsx_v3]` der Datei `ncp.ini` festlegen.

Wenn Sie Nicht-SNAT-IP-Blöcke erstellen, während NCP ausgeführt wird, müssen Sie NCP neu starten. Andernfalls verwendet NCP weiterhin die freigegebenen IP-Blöcke, bis sie erschöpft sind.

Hinweis Wenn Sie einen IP-Block erstellen, darf das Präfix nicht größer als der Wert des Parameters `subnet_prefix` in der NCP-Konfigurationsdatei `ncp.ini` sein.

IP-Pool für SNAT

Der IP-Pool wird für die Zuteilung von IP-Adressen verwendet, die der Übersetzung von Pod-IP-Adressen über SNAT-Regeln dienen. Zudem werden sie zum Verfügbarmachen der Ingress-Controller über SNAT/DNAT-Regeln verwendet – genau wie Openstack Floating IP-Adressen. Diese IP-Adressen werden auch als „externe IP-Adressen“ bezeichnet.

Mehrere Kubernetes-Cluster verwenden denselben externen IP-Pool. Jede NCP-Instanz verwendet einen Teil dieses Pools für den Kubernetes-Cluster, den sie verwaltet. Standardmäßig wird dasselbe Subnetzpräfix für Pod-Subnetze verwendet. Wenn Sie eine andere Subnetzgröße verwenden möchten, aktualisieren Sie die `external_subnet_prefix`-Option im `[nsx_v3]`-Abschnitt in `ncp.ini`.

Melden Sie sich bei NSX Manager an und erstellen Sie einen Pool oder suchen Sie einen vorhandenen Pool.

Geben Sie IP-Pools für SNAT an, indem Sie die Option `external_ip_pools` im Abschnitt `[nsx_v3]` der Datei `ncp.ini` festlegen.

Sie können SNAT auch für einen bestimmten Dienst konfigurieren, indem Sie dem Dienst eine Anmerkung hinzufügen. Beispiel:

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

NCP konfiguriert die SNAT-Regel für diesen Dienst. Bei der Quell-IP der Regel handelt es sich um die Gruppe der Backend-Pods. Bei der Ziel-IP handelt es sich um die SNAT-IP, die aus dem angegebenen externen IP-Pool zugeteilt wurde. Beachten Sie Folgendes:

- Der von `ncp/snat_pool` angegebene IP-Pool sollte bereits in NSX-T Data Center vorhanden sein, bevor der Dienst konfiguriert wird. Der IP-Pool muss das Tag `{"ncp/owner": "cluster:<cluster>"}` aufweisen.
- In NSX-T Data Center ist die Priorität der SNAT-Regel für den Dienst höher als die für das Projekt.
- Wenn ein Pod mit mehreren SNAT-Regeln konfiguriert wird, funktioniert nur eine der Regeln.

Sie können durch Hinzufügen des folgenden Tags zum IP-Pool festlegen, welchem Namespace IPs aus dem SNAT-IP-Pool zugeteilt werden können.

- Geltungsbereich: `ncp/owner`, Tag: `ns:<namespace_UUID>`

Sie können die Namespace-UUID mit einem der folgenden Befehle abrufen:

```
oc get ns -o yaml
```

Beachten Sie Folgendes:

- Jedes Tag sollte eine UUID enthalten. Sie können mehrere Tags für denselben Pool erstellen.
- Wenn Sie die Tags ändern, nachdem einigen Namespaces basierend auf den alten Tags IPs zugewiesen wurden, werden diese IPs nicht wiederhergestellt, bis sich die SNAT-Konfigurationen der Dienste ändern oder NCP neu gestartet wird.
- Das Owner-Tag für den Namespace ist optional. Ohne dieses Tag können jedem Namespace IPs aus dem SNAT-IP-Pool zugeteilt werden.

(Optional) Markierte Firewallabschnitte

Damit der Administrator Firewallregeln erstellen kann und diese die von NCP erstellten, auf Netzwerkrichtlinien basierenden Firewallabschnitte nicht beeinträchtigen, melden Sie sich bei NSX Manager an und erstellen Sie zwei Firewallabschnitte.

Geben Sie Firewall-Markierungsabschnitte an, indem Sie die Optionen `bottom_firewall_section_marker` und `top_firewall_section_marker` im Abschnitt `[nsx_v3]` der Datei `ncp.ini` festlegen.

Der untere Firewallabschnitt muss sich unterhalb des oberen Firewallabschnitts befinden. Wenn diese Firewallabschnitte erstellt sind, werden alle von NCP zur Isolierung erstellten Firewallabschnitte oberhalb des unteren Firewallabschnitts und alle von NCP für Richtlinien erstellten Firewallabschnitte unterhalb des oberen Firewallabschnitts erstellt. Wenn diese Markierungsabschnitte nicht erstellt werden, werden alle Isolierungsregeln unten und alle Richtlinienabschnitte oben erstellt. Mehrere markierte Firewallabschnitte mit demselben Wert pro Cluster werden nicht unterstützt und führen zu einem Fehler.

Installieren von NCP

NCP ist vollständig in OpenShift integriert. Wenn Sie die benötigten Parameter in der Ansible-Hostdatei hinzufügen und OpenShift installieren, wird NCP automatisch installiert.

Dieses Kapitel enthält die folgenden Themen:

- [Systemvoraussetzungen](#)
- [Vorbereiten der Ansible-Hostdatei](#)

Systemvoraussetzungen

Stellen Sie vor dem Installieren von OpenShift sicher, dass Ihre Umgebung bestimmte Anforderungen erfüllt.

Allgemeine Anforderungen

- Ansible 2.4 oder höher.

Anforderungen von virtuellen Maschinen

OpenShift-Knoten-VMs müssen zwei vNICs aufweisen:

- Eine Verwaltungs-vNIC, die mit dem logischen Switch verbunden ist, der über einen Uplink zum Tier-1-Verwaltungsrouter verfügt.
- Die zweite vNIC auf allen VMs muss folgende Tags in NSX-T aufweisen, damit NCP weiß, welcher Port als übergeordnete VIF für alle auf diesem bestimmten OpenShift-Knoten ausgeführten PODs verwendet wird.

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

Bare-Metal-Maschinenanforderungen

- Die OpenShift-Knoten müssen NSX-T-Transportknoten sein, und die oben erwähnten Tags müssen anstatt auf die VIFs auf die Transportknoten angewendet werden.
- Die Ansible-Hostdatei muss die folgende Einstellung aufweisen: `nsx_node_type='BAREMETAL'`.

NSX-T-Anforderungen

- Ein Tier-0-Router.
- Eine Overlay-Transportzone.
- Ein IP-Block für das POD-Netzwerk.
- (Optional) Ein IP-Block für das geroutete POD-Netzwerk (keine NAT).
- Ein IP-Pool für SNAT. Standardmäßig ist der IP-Block für das POD-Netzwerk nur innerhalb von NSX-T routingfähig. NCP verwendet diesen IP-Pool, um Konnektivität nach außen bereitzustellen.
- (Optional) Oberer und unterer Firewallabschnitt. NCP platziert Kubernetes-Netzwerkrichtlinienregeln zwischen diesen beiden Abschnitten.
- Open vSwitch- und CNI-Plug-In-RPMs müssen auf einem HTTP-Server gehostet werden, der von den OpenShift-Knoten-VMs aus erreichbar ist.

NCP-Docker-Image

Derzeit ist das NCP-Docker-Image nicht öffentlich verfügbar. Sie müssen das Image `nsx-ncp` in einer lokalen privaten Registrierung vorliegen haben oder wie folgt vorgehen:

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

Auf allen Knoten:

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

Vorbereiten der Ansible-Hostdatei

Sie müssen in der Ansible-Hostdatei NCP-Parameter angeben, damit NCP in OpenShift integriert wird.

Nachdem Sie die folgenden Parameter in der Ansible-Hostdatei angegeben haben, wird beim Installieren von OpenShift NCP automatisch installiert.

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

(Erforderlich) Dies ist erforderlich, da mehrere OpenShift-/Kubernetes-Cluster eine Verbindung zum selben NSX Manager herstellen können.

- `nsx_api_managers='10.10.10.10'`

(Erforderlich) IP-Adressen von NSX Manager. Geben Sie für einen NSX Manager-Cluster die IP-Adressen in Form einer kommagetrennten Liste an.

- `nsx_tier0_router='MyT0Router'`

(Erforderlich) Name oder UUID des Tier-0-Routers, zu dem die Tier-1-Router des Projekts eine Verbindung herstellen.

- `nsx_overlay_transport_zone='my_overlay_tz'`

(Erforderlich) Name oder UUID der Overlay-Transportzone, die zum Erstellen logischer Switches verwendet wird.

- `nsx_container_ip_block='ip_block_for_my_ocp_cluster'`

(Erforderlich) Name oder UUID eines auf NSX-T konfigurierten IP-Blocks. Ausgehend von diesem IP-Block ist ein Subnetz pro Projekt vorhanden. Diese Netzwerke befinden sich hinter SNAT und sind nicht routingfähig.

- `nsx_ovs_uplink_port='ens224'`

(Erforderlich) Sofern im HOSTVM-Modus. NSX-T benötigt eine zweite vNIC für das POD-Netzwerk auf den OCP-Knoten. Diese muss sich von der Verwaltungs-vNIC unterscheiden. Es wird dringend empfohlen, beide vNICs mit logischen NSX-T-Switches zu verbinden. Die zweite (Nicht-Verwaltungs-) vNIC muss hier angegeben werden. Für Bare Metal ist dieser Parameter nicht erforderlich.

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

(Erforderlich) Temporäre Anforderung, bis NCP ein Bootstrapping der Knoten durchführen kann. Darüber hinaus muss `nsx-cni` auf einem HTTP-Server platziert werden.

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

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

(Erforderlich) Temporäre Parameter, bis NCP ein Bootstrapping der Knoten durchführen kann. Kann bei einem Bare-Metal-Setup ignoriert werden.

- `nsx_node_type='HOSTVM'`

(Optional) Die Standardeinstellung lautet HOSTVM. Setzen Sie den Parameter auf BAREMETAL, wenn OpenShift nicht auf den VMs ausgeführt wird.

- `nsx_k8s_api_ip=192.168.10.10`

(Optional) Sofern festgelegt, kommuniziert NCP mit diesen IP-Adressen. Andernfalls erfolgt die Kommunikation mit der Kubernetes-Dienst-IP.

- `nsx_k8s_api_port=192.168.10.10`

(Optional) Die Standardeinstellung für den Kubernetes-Dienst lautet 443. Setzen Sie diese Einstellung auf 8443, wenn Sie ihn in Kombination mit `nsx_k8s_api_ip` verwenden, um die Master-Knoten-IP anzugeben.

- `nsx_insecure_ssl=true`

(Optional) Die Standardeinstellung lautet `true`, da der NSX Manager mit einem nicht vertrauenswürdigen Zertifikat ausgeliefert wird. Wenn Sie das Zertifikat durch ein vertrauenswürdiges Zertifikat ersetzt haben, können Sie die Einstellung in `false` ändern.

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

(Optional) Die Standardeinstellung lautet 24. Dies ist die Subnetzgröße, die pro OpenShift-Projekt reserviert wird. Wenn die Anzahl der PODs die Subnetzgröße überschreitet, wird dem Projekt ein neuer logischer Switch mit derselben Subnetzgröße hinzugefügt.

- `nsx_use_loadbalancer=true`

(Optional) Die Standardeinstellung lautet `true`. Setzen Sie diesen Parameter auf `false`, wenn Sie keine NSX-T-Load Balancer für OpenShift-Routen und Dienste des Typs LoadBalancer verwenden möchten.

- `nsx_lb_service_size='SMALL'`

(Optional) Die Standardeinstellung lautet `SMALL`. Je nach NSX Edge-Größe ist `MEDIUM` oder `LARGE` ebenfalls möglich.

- `nsx_no_snat_ip_block='router_ip_block_for_my_ocp_cluster'`

(Optional) Wenn die Anmerkung `ncp/no_snat=true` in einem Projekt oder Namespace angewendet wird, wird das Subnetz aus diesem IP-Block entnommen und es ist keine SNAT dafür verfügbar. Es wird erwartet, dass das Projekt bzw. der Namespace routingfähig ist.

- `nsx_external_ip_pool='external_pool_for_snat'`

(Erforderlich) IP-Pool für SNAT und Load Balancer, wenn `nsx_external_ip_pool_lb` nicht definiert ist.

- `nsx_external_ip_pool_lb='my_ip_pool_for_lb'`

(Optional) Legen Sie diesen Parameter fest, wenn Sie einen eindeutigen IP-Pool für Router und SvcTypeLB möchten.

- `nsx_top_fw_section='top_section'`

(Optional) Kubernetes-Netzwerkrichtlinienregeln werden in NSX-T-Firewallregeln übersetzt und unterhalb dieses Abschnitts platziert.

- `nsx_bottom_fw_section='bottom_section'`

(Optional) Kubernetes-Netzwerkrichtlinienregeln werden in NSX-T-Firewallregeln übersetzt und oberhalb dieses Abschnitts platziert.

- `nsx_api_cert='/path/to/cert/nsx.crt'`
- `nsx_api_private_key='/path/to/key/nsx.key'`

(Optional) Sofern festgelegt, werden `nsx_api_user` und `nsx_api_password` ignoriert. Das Zertifikat muss auf NSX-T hochgeladen werden, und ein sich mit diesem Zertifikat authentifizierender Prinzipalidentitätsbenutzer muss manuell erstellt werden.

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

(Optional) NSX-T-Load Balancer erfordert ein Standardzertifikat, um SNIs für TLS-basierte Routen erstellen zu können. Dieses Zertifikat wird nur dann präsentiert, wenn keine Route konfiguriert ist. Wenn es nicht bereitgestellt wird, wird ein selbstsigniertes Zertifikat generiert.

Beispiel einer Ansible-Hostdatei

```
[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'
```

Lastausgleich

Der NSX-T Data Center-Load Balancer ist in OpenShift integriert und fungiert als OpenShift-Router.

NCP überwacht OpenShift-Routen- und -Endpoint-Ereignisse und konfiguriert Lastausgleichsregeln auf dem Load Balancer basierend auf der Routenspezifikation. Dies führt dazu, dass der NSX-T Data Center-Load Balancer eingehenden Schicht 7-Datenverkehr basierend auf den Regeln zu den geeigneten Backend-Pods weiterleitet.

Konfigurieren des Lastausgleichs

Das Konfigurieren des Lastausgleichs umfasst das Konfigurieren eines Kubernetes-LoadBalancer-Diensts oder einer OpenShift-Route. Sie müssen auch den NCP ReplicationController konfigurieren. Der LoadBalancer-Dienst wird für den Schicht 4-Datenverkehr und die OpenShift-Route für den Schicht 7-Datenverkehr verwendet.

Wenn Sie einen Kubernetes-LoadBalancer-Dienst konfigurieren, wird diesem eine IP-Adresse aus dem von Ihnen konfigurierten externen IP-Block zugeteilt. Der Load Balancer wird auf dieser IP-Adresse und dem Dienst-Port bereitgestellt. Sie können den Namen oder die ID eines IP-Pools mithilfe der `loadBalancerIP`-Spezifikation in der Definition des LoadBalancer-Diensts angeben. Die IP-Adresse des LoadBalancer-Diensts wird aus diesem IP-Pool zugeteilt. Wenn die `loadBalancerIP`-Spezifikation leer ist, wird die IP-Adresse aus dem externen IP-Block zugeteilt, den Sie konfigurieren.

Der von `loadBalancerIP` angegebene IP-Pool muss das Tag `{"ncp/owner": "cluster:<cluster>"}` aufweisen.

Um den Load Balancer von NSX-T Data Center zu verwenden, müssen Sie den Lastausgleich in NCP konfigurieren. Führen Sie in der Datei `ncp_rc.yml` die folgenden Schritte aus:

- 1 Legen Sie `use_native_loadbalancer = True` fest.
- 2 Legen Sie `pool_algorithm` auf `WEIGHTED_ROUND_ROBIN` fest.
- 3 Legen Sie „`lb_default_cert_path`“ und „`lb_priv_key_path`“ als vollständige Pfadnamen des von der Zertifizierungsstelle signierten Zertifikats und der Datei mit dem privaten Schlüssel fest. Ein Beispielskript zum Generieren eines von einer Zertifizierungsstelle signierten Zertifikats finden Sie unten. Mounten Sie darüber hinaus das Standardzertifikat und den Standardschlüssel in den NCP-Pod. Anweisungen hierzu finden Sie unten.

- 4 (Optional) Legen Sie mit den Parametern `l4_persistence` und `l7_persistence` eine Persistenzeinstellung fest. Die verfügbare Option für die Schicht-4-Persistenz ist „Quell-IP“. Die verfügbaren Optionen für die Schicht-7-Persistenz sind „Cookie“ und „Quell-IP“. Die Standardeinstellung ist `<None>`.
Beispiel:

```
# 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 (Optional) Legen Sie `service_size` = SMALL, MEDIUM oder LARGE fest. Die Standardeinstellung ist SMALL.
- 6 Wenn Sie OpenShift 3.11 ausführen, müssen Sie die folgende Konfiguration ausführen, damit OpenShift dem LoadBalancer-Dienst keine IP zuweist.
- Legen Sie `ingressIPNetworkCIDR` unter `networkConfig` in der Datei `/etc/origin/master/master-config.yaml` auf „0.0.0.0/32“ fest.
 - Starten Sie den API-Server und die API-Controller mit den folgenden Befehlen neu:

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

Für einen Kubernetes-LoadBalancer-Dienst können Sie `sessionAffinity` auch in der Dienstspezifikation angeben, um das Persistenzverhalten für den Dienst zu konfigurieren, wenn die globale Schicht-4-Persistenz deaktiviert, also `l4_persistence` auf `<None>` festgelegt ist. Wenn `l4_persistence` auf `source_ip` festgelegt ist, kann die `sessionAffinity` auf der Dienstspezifikation verwendet werden, um die Persistenz-Zeitüberschreitung für den Dienst anzupassen. Die standardmäßige Persistenz-Zeitüberschreitung von Layer 4 beträgt 10.800 Sekunden (wie in der Kubernetes-Dokumentation für Dienste

(<https://kubernetes.io/docs/concepts/services-networking/service>) angegeben. Alle Dienste mit standardmäßiger Persistenz-Zeitüberschreitung nutzen das gleiche Persistenz-Profil des NSX-T-Lastausgleichs. Für jeden Dienst mit einer nicht standardmäßigen Persistenz-Zeitüberschreitung wird ein dediziertes Profil erstellt.

Hinweis Wenn der Backend-Dienst eines Dateneingangs ein Dienst des Typs LoadBalancer ist, dürfen der virtuelle Server der Schicht 4 für den Dienst und der virtuelle Server der Schicht 7 für den Dateneingang nicht unterschiedliche Persistenzeinstellungen aufweisen, beispielsweise `source_ip` für Schicht 4 und `cookie` für Schicht 7. In einem Szenario dieser Art müssen die Persistenzeinstellungen für beide virtuelle Server identisch sein (`source_ip`, `cookie` oder `None`), oder einer davon hat die Einstellung `None` (in diesem Fall kann die andere Einstellung `source_ip` oder `cookie` lauten). Beispiel für ein Szenario dieser Art:

```
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
```

Router-Sharding

NCP verarbeitet immer TLS-Edge-Beendigungs- und HTTP-Routen und überspringt TLS-Passthrough-Routen und TLS-Neuverschlüsselungsrouten unabhängig von deren Namespaces oder Namespace-Bezeichnungen. Um einen OpenShift-Router auf die ausschließliche Verarbeitung von TLS-Neuverschlüsselungs- und TLS-Passthrough-Routen zu beschränken, müssen Sie die folgenden Schritte ausführen:

- Fügen Sie dem OpenShift-Router eine Namespace-Bezeichnungsauswahl hinzu.
- Fügen Sie dem Ziel-Namespace eine Namespace-Bezeichnung hinzu.
- Erstellen Sie im Ziel-Namespace TLS-Neuverschlüsselungs-/TLS Passthrough-Routen.

Um beispielsweise einen Router mit einer Namespace-Bezeichnungsauswahl zu konfigurieren, führen Sie den folgenden Befehl aus (davon ausgehend, dass der Dienstkontoname des Routers `router` lautet):

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

Der Router verarbeitet jetzt Routen von den ausgewählten Namespaces. Damit diese Auswahl mit einem Namespace übereinstimmt, führen Sie den folgenden Befehl aus (davon ausgehend, dass der Namespace den Namen `ns1` hat):

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

Beispiel für einen Load Balancer auf Schicht 7:

Zur Bereitstellung des Lastausgleichs auf Schicht 7 konfiguriert die folgende YAML-Datei zwei Replikations-Controller (`tea-rc` und `coffee-rc`), zwei Dienste (`tea-svc` und `coffee-svc`) sowie zwei Routen (`cafe-route-multi` und `cafe-route`).

```
# 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

```

Zusätzliche Hinweise

- Nur die Edge-Beendigung wird für HTTPS-Datenverkehr unterstützt.
- Unterdomänen als Platzhalter werden unterstützt. Wenn zum Beispiel `wildcardPolicy` auf **Unterdomäne** und der Hostname auf **wildcard.example.com** festgelegt ist, werden alle Anforderungen an ***.example.com** verarbeitet.
- Wenn NCP aufgrund einer fehlerhaften Konfiguration einen Fehler während der Verarbeitung eines Routen-Ereignisses auslöst, müssen Sie die YAML-Datei der Route korrigieren und die Routen-Resource löschen und neu erstellen.
- NCP erzwingt den Hostnamen-Besitz nicht nach Namespaces.
- Ein LoadBalancer-Dienst wird pro Kubernetes-Cluster unterstützt.
- NSX-T Data Center erstellt für jeden LoadBalancer-Dienst-Port einen virtuellen Server und einen Pool des Load Balancers auf Schicht 4. TCP und UDP werden unterstützt.
- Der Load Balancer von NSX-T Data Center ist in verschiedenen Größen erhältlich. Weitere Informationen zum Konfigurieren eines Load Balancers für NSX-T Data Center finden Sie im *Administratorhandbuch für NSX-T Data Center*.

Nachdem der Load Balancer erstellt wurde, kann seine Größe nicht durch Aktualisierung der Konfigurationsdatei geändert werden. Die Größe kann über die Benutzeroberfläche oder API geändert werden.

- Das automatische Skalieren des Load Balancers auf Schicht 4 wird unterstützt. Wenn ein Kubernetes-LoadBalancer-Dienst erstellt oder geändert wird, sodass er zusätzliche virtuelle Server erfordert, und der vorhandene Load Balancer auf Schicht 4 nicht über ausreichend Kapazität verfügt, wird ein neuer Load Balancer auf Schicht 4 erstellt. NCP löscht einen Load Balancer auf Schicht 4 auch, wenn keine virtuellen Server mehr an ihn angehängt sind. Diese Funktion ist standardmäßig aktiviert. Sie können sie deaktivieren, indem Sie in der NCP-ConfigMap `l4_lb_auto_scaling` auf `false` festlegen.

Beispielskript zum Generieren eines von einer Zertifizierungsstelle signierten Zertifikats

Das nachfolgende Skript generiert ein von einer Zertifizierungsstelle signiertes Zertifikat und einen privaten in den Dateien `<Dateiname>.cert` und `<Dateiname>.key` gespeicherten privaten Schlüssel. Der Befehl `genrsa` generiert einen Zertifizierungsstellen-Schlüssel. Der Zertifizierungsstellen-Schlüssel muss verschlüsselt werden. Sie können eine Verschlüsselungsmethode mit einem Befehl wie zum Beispiel `aes256` angeben.

```
#!/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}.cert -sha256
```

Mounten Sie das Standardzertifikat und den Standardschlüssel in den NCP-Pod

Wenn Sie das Zertifikat und den privaten Schlüssel erstellt haben, platzieren Sie sie im Verzeichnis `/etc/nsx-ujo` auf dem VM-Host. Vorausgesetzt, dass die Zertifikats- und Schlüsseldateien die Namen `lb-default.crt` und `lb-default.key` aufweisen, bearbeiten Sie `ncp-rc.yaml`, sodass diese Dateien auf dem Host in den Pod gemountet werden. Beispiel:

```
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-uj0/lb-default.key
volumes:
...
- name: lb-default-cert
  hostPath:
    path: /etc/nsx-uj0/lb-default.crt
- name: lb-priv-key
  hostPath:
    path: /etc/nsx-uj0/lb-default.key
```

Verwalten von NSX Container Plug-in

5

Sie können NSX Container Plug-in über die NSX Manager-GUI oder über die Befehlszeilenschnittstelle (Command Line Interface, CLI) verwalten.

Hinweis Wenn eine Container-Host-VM auf ESXi 6.5 ausgeführt wird und die VM über vMotion auf einen anderen ESXi 6.5-Host migriert wird, geht die Verbindung von Containern, die auf dem Container-Host ausgeführt werden, mit Containern, die auf anderen Container-Hosts ausgeführt werden, verloren. Sie können das Problem lösen, indem Sie die vNIC des Container-Hosts trennen und erneut verbinden. Dieses Problem tritt bei ESXi 6.5 Update 1 oder höher nicht auf.

Hyperbus reserviert die VLAN-ID 4094 auf dem Hypervisor für die PVLAN-Konfiguration, und die ID kann nicht geändert werden. Um VLAN-Konflikte zu vermeiden, konfigurieren Sie logische VLAN-Switches oder VTEP-vmknics mit derselben VLAN-ID.

Dieses Kapitel enthält die folgenden Themen:

- [Verwalten von IP-Blöcken über die NSX Manager-GUI](#)
- [Anzeigen von IP-Block-Subnetzen über die GUI von NSX Manager](#)
- [CIF-verknüpfte logische Ports](#)
- [CLI-Befehle](#)
- [Fehlercodes](#)

Verwalten von IP-Blöcken über die NSX Manager -GUI

Sie können die Tags für einen IP-Block über die NSX Manager-GUI hinzufügen, löschen, bearbeiten, verwalten und die zugehörigen Details anzeigen.

Verfahren

- 1 Melden Sie sich über einen Browser bei NSX Manager unter `https://<nsx-manager-IP-address-or-domain-name>` an.
- 2 Navigieren Sie zu **Netzwerk > IPAM**.

Es wird eine Liste der vorhandenen IP-Blöcke angezeigt.

3 Führen Sie eine der folgenden Aktionen durch.

Option	Aktion
Hinzufügen eines IP-Blocks	Klicken Sie auf HINZUFÜGEN .
Löschen eines oder mehrerer IP-Blöcke	Wählen Sie einen oder mehrere IP-Blöcke aus, und klicken Sie auf LÖSCHEN .
Bearbeiten eines IP-Blocks	Wählen Sie einen IP-Block aus, und klicken Sie auf BEARBEITEN .
Anzeigen von Details zu einem IP-Block	Klicken Sie auf den Namen des IP-Blocks. Klicken Sie auf die Registerkarte Übersicht , um allgemeine Informationen anzuzeigen. Klicken Sie auf die Registerkarte Subnetze , um die Subnetze dieses IP-Blocks anzuzeigen.
Verwalten von Tags für einen IP-Block	Wählen Sie einen IP-Block aus, und klicken Sie auf AKTIONEN > Tags verwalten .

IP-Blöcke mit zugeteilten Subnetzen können nicht gelöscht werden.

Anzeigen von IP-Block-Subnetzen über die GUI von NSX Manager

Sie können Subnetze für einen IP-Block über die NSX Manager-GUI anzeigen. Es wird nicht empfohlen, IP-Block-Subnetze nach der Installation und Ausführung von NCP hinzuzufügen oder zu löschen.

Verfahren

- 1 Melden Sie sich über einen Browser bei NSX Manager unter `https://<nsx-manager-IP-address-or-domain-name>` an.
- 2 Navigieren Sie zu **Netzwerk > IPAM**.
Es wird eine Liste der vorhandenen IP-Blöcke angezeigt.
- 3 Klicken Sie auf den Namen eines IP-Blocks.
- 4 Klicken Sie auf die Registerkarte **Subnetze**.

CIF-verknüpfte logische Ports

CIFs (Container Interfaces) sind Netzwerkschnittstellen für Container, die mit logischen Ports auf einem Switch verbunden sind. Diese Ports werden als CIF-verknüpfte logische Ports bezeichnet.

Sie können CIF-verknüpfte logische Ports über die NSX Manager-GUI verwalten.

Verwalten von CIF-verknüpften logischen Ports

Navigieren Sie zu **Netzwerk > Switching > Ports**, um alle logischen Ports anzuzeigen, einschließlich der CIF-verknüpften logischen Ports. Klicken Sie auf die Anhangsverknüpfung eines CIF-verknüpften logischen Ports, um die Anlageninformationen anzuzeigen. Klicken Sie auf den Link des logischen Ports, um einen Fensterbereich mit vier Registerkarten zu öffnen: „Übersicht“, „Überwachen“, „Verwalten“ und „Zugehörig“. Wenn Sie auf **Zugehörig > Logische Ports** klicken, wird der zugehörige logische Port auf einem Uplink-Switch angezeigt. Weitere Informationen zu Switch-Ports finden Sie im *NSX-T-Administratorhandbuch*.

Netzwerküberwachungstools

Die folgenden Tools unterstützen CIF-verknüpfte logische Ports. Weitere Informationen zu diesen Tools finden Sie im *NSX-T-Administratorhandbuch*.

- Traceflow
- Portverbindung
- IPFIX
- Die Remote-Portspiegelung mit GRE-Kapselung eines logischen Switch-Ports, der mit einem Container verbunden ist, wird unterstützt. Weitere Informationen finden Sie unter „Grundlegendes zum Switching-Profil für die Portspiegelung“ im *NSX-T-Administratorhandbuch*. Die Portspiegelung des CIF-Ports zum VIF-Port wird jedoch über die Manager-Benutzeroberfläche nicht unterstützt.

CLI-Befehle

Um CLI-Befehle auszuführen, melden Sie sich beim NSX Container Plug-in-Container an, öffnen Sie ein Terminal, und führen Sie den Befehl `nsxcli` aus.

Sie können die CLI-Eingabeaufforderung auch aufrufen, indem Sie den folgenden Befehl für einen Knoten ausführen:

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

Tabelle 5-1. CLI-Befehle für den NCP-Container

Typ	Befehl
Status	<code>get ncp-master status</code>
Status	<code>get ncp-nsx status</code>
Status	<code>get ncp-watcher <Watcher-Name></code>
Status	<code>get ncp-watchers</code>
Status	<code>get ncp-k8s-api-server status</code>
Status	<code>check projects</code>
Status	<code>check project <project-name></code>
Cache	<code>get project-cache <Projektname></code>
Cache	<code>get project-caches</code>
Cache	<code>get namespace-cache <Namensraum-Name></code>
Cache	<code>get namespace-caches</code>
Cache	<code>get pod-cache <Pod-Name></code>
Cache	<code>get pod-caches</code>
Cache	<code>get ingress-caches</code>
Cache	<code>get ingress-cache <ingress-name></code>

Tabelle 5-1. CLI-Befehle für den NCP-Container (Fortsetzung)

Typ	Befehl
Cache	get ingress-controllers
Cache	get ingress-controller <ingress-controller-name>
Cache	get network-policy-caches
Cache	get network-policy-cache <pod-name>
Support	get ncp-log file <Dateiname>
Support	get ncp-log-level
Support	set ncp-log-level <log-level>
Support	get support-bundle file <Dateiname>
Support	get node-agent-log file <Dateiname>
Support	get node-agent-log file <Dateiname> <Knotenname>

Tabelle 5-2. CLI-Befehle für den NSX-Knoten-Agent-Container

Typ	Befehl
Status	get node-agent-hyperbus status
Cache	get container-cache <Containername>
Cache	get container-caches

Tabelle 5-3. CLI-Befehle für den NSX-Kube-Proxy-Container

Typ	Befehl
Status	get ncp-k8s-api-server status
Status	get kube-proxy-watcher <Watcher-Name>
Status	get kube-proxy-watchers
Status	dump ovs-flows

Statusbefehle für den NCP-Container

- Status des NCP-Masters anzeigen

```
get ncp-master status
```

Beispiel:

```
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
```

- Verbindungsstatus zwischen NCP und NSX Manager anzeigen

```
get ncp-nsx status
```

Beispiel:

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

- Watcher-Status für eingehenden Datenverkehr, Namensraum, Pod und Dienst anzeigen

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

Beispiel 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
```

Beispiel 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

```

- Verbindungsstatus zwischen NCP und Kubernetes-API-Server anzeigen

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

Beispiel:

```

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

```

- Alle Projekte oder ein bestimmtes Projekt überprüfen

```

check projects
check project <project-name>

```

Beispiel:

```

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

```

Cachebefehle für den NCP-Container

- Internen Cache für Projekte oder Namensräume abrufen

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

Beispiel:

```
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

```

■ Internen Cache für Pods abrufen

```

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

```

Beispiel:

```

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

```

- Netzwerkrichtlinien-Caches oder einen bestimmten Netzwerkrichtlinien-Cache abrufen

```

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

```

Beispiel:

```

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
```

Supportbefehle für den NCP-Container

■ NCP-Support-Paket im Dateispeicher speichern

Das Support-Paket umfasst die Protokolldateien für alle Container in Pods mit der Bezeichnung **tier:nsx-networking**. Die Paketdatei liegt im TGZ-Format vor und befindet sich im CLI-Standard-Dateispeicherverzeichnis `/var/vmware/nsx/file-store`. Mithilfe des CLI-Befehls „file-store“ können Sie die Paketdatei in eine Remote-Site kopieren.

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

Beispiel:

```
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
```

■ NCP-Protokolle im Dateispeicher speichern

Die Protokolldatei wird im TGZ-Format im CLI-Standard-Dateispeicherverzeichnis `/var/vmware/nsx/file-store` gespeichert. Mithilfe des CLI-Befehls „file-store“ können Sie die Paketdatei in eine Remote-Site kopieren.

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

Beispiel:

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

■ Knoten-Agent-Protokolle im Dateispeicher speichern

Speichern Sie die Knoten-Agent-Protokolle von einem oder allen Knoten. Die Protokolle werden im TGZ-Format im CLI-Standard-Dateispeicherverzeichnis `/var/vmware/nsx/file-store` gespeichert. Mithilfe des CLI-Befehls „file-store“ können Sie die Paketdatei in eine Remote-Site kopieren.

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

Beispiel:

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

- Protokollebene abrufen und einrichten

Zu den verfügbaren Protokollebenen gehören: NOTSET, DEBUG, INFO, WARNING, ERROR und CRITICAL.

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

Beispiel:

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

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

Status-Befehle für den NSX-Knoten-Agent-Container

- Zeigen Sie den Verbindungsstatus zwischen dem Knoten-Agent und HyperBus auf diesem Knoten an.

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

Beispiel:

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

Cache-Befehle für den NSX-Knoten-Agent-Container

- Internen Cache für NSX-Knoten-Agent-Container abrufen

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

Beispiel 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
```

Beispiel 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
```

Status-Befehle für den NSX-Kube-Proxy-Container

- Verbindungsstatus zwischen Kube-Proxy und Kubernetes-API-Server anzeigen

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

Beispiel:

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

- Kube-Proxy-Watcher-Status anzeigen

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

Beispiel 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
```

Beispiel 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

```

■ OVS-Flows für Speicherabbild an einem Knoten

```
dump ovs-flows
```

Beispiel:

```

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 ac-
  tions=ct(table=1)
    cookie=0x0, duration=8.898s, table=0, n_packets=0, n_bytes=0, idle_age=8, priority=0 ac-
    tions=NORMAL
      cookie=0x0, duration=8.759s, table=1, n_packets=0, n_bytes=0, idle_age=8, priori-
      ty=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, priori-
        ty=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, priori-
          ty=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 ac-
            tions=NORMAL
              cookie=0x0, duration=8.856s, table=2, n_packets=0, n_bytes=0, idle_age=8, actions=NORMAL

```

Fehlercodes

In diesem Abschnitt werden die Fehlercodes der verschiedenen Komponenten aufgelistet.

NCP-Fehlercodes

Fehlercode	Beschreibung
NCP00001	Ungültige Konfiguration
NCP00002	Initialisierung fehlgeschlagen
NCP00003	Ungültiger Zustand
NCP00004	Ungültiger Adapter
NCP00005	Zertifikat wurde nicht gefunden
NCP00006	Token wurde nicht gefunden
NCP00007	Ungültige Konfiguration für NSX
NCP00008	Ungültiges NSX-Tag
NCP00009	NSX-Verbindung fehlgeschlagen

Fehlercode	Beschreibung
NCP00010	Knoten-Tag nicht gefunden
NCP00011	Ungültiger logischer Switch-Port des Knotens
NCP00012	Aktualisieren der übergeordneten VIF fehlgeschlagen
NCP00013	VLAN ausgeschöpft
NCP00014	VLAN-Version ist fehlgeschlagen
NCP00015	IP-Pool ist ausgeschöpft
NCP00016	IP-Version ist fehlgeschlagen
NCP00017	IP-Block ausgeschöpft
NCP00018	Erstellen des IP-Subnetzes ist fehlgeschlagen
NCP00019	Löschen des IP-Subnetzes ist fehlgeschlagen
NCP00020	Erstellen des IP-Pools ist fehlgeschlagen
NCP00021	Löschen des IP-Pools ist fehlgeschlagen
NCP00022	Erstellen des logischen Routers ist fehlgeschlagen
NCP00023	Aktualisieren des logischen Routers ist fehlgeschlagen
NCP00024	Löschen des logischen Routers ist fehlgeschlagen
NCP00025	Erstellen des logischen Switches ist fehlgeschlagen

Fehlercode	Beschreibung
NCP00026	Aktualisieren des logischen Switches ist fehlgeschlagen
NCP00027	Löschen des logischen Switches ist fehlgeschlagen
NCP00028	Erstellen des Ports für den logischen Router ist fehlgeschlagen
NCP00029	Löschen des Ports für den logischen Router ist fehlgeschlagen
NCP00030	Erstellen des Ports für den logischen Switch ist fehlgeschlagen
NCP00031	Aktualisieren des Ports für den logischen Switch ist fehlgeschlagen
NCP00032	Löschen des Ports für den logischen Switch ist fehlgeschlagen
NCP00033	Netzwerkrichtlinie wurde nicht gefunden
NCP00034	Erstellen der Firewall ist fehlgeschlagen
NCP00035	Lesen der Firewall ist fehlgeschlagen
NCP00036	Aktualisieren der Firewall ist fehlgeschlagen
NCP00037	Löschen der Firewall ist fehlgeschlagen
NCP00038	Mehrere Firewalls gefunden
NCP00039	Erstellen der NS-Gruppe ist fehlgeschlagen
NCP00040	Löschen der NS-Gruppe ist fehlgeschlagen
NCP00041	Erstellen von IP Set ist fehlgeschlagen
NCP00042	Aktualisieren von IP Set ist fehlgeschlagen

Fehlercode	Beschreibung
NCP00043	Löschen von IP Set ist fehlgeschlagen
NCP00044	Erstellen der SNAT-Regel ist fehlgeschlagen
NCP00045	Löschen der SNAT-Regel ist fehlgeschlagen
NCP00046	Adapter-API-Verbindung ist fehlgeschlagen
NCP00047	Adapter-Watcher-Ausnahme
NCP00048	Löschen des Load Balancer-Diensts ist fehlgeschlagen
NCP00049	Erstellen des virtuellen Servers für den Load Balancer ist fehlgeschlagen
NCP00050	Aktualisieren des virtuellen Servers für den Load Balancer ist fehlgeschlagen

Fehlercode	Beschreibung
NCP00051	Löschen des virtuellen Servers für den Load Balancer ist fehlgeschlagen
NCP00052	Erstellen des Load Balancer-Pools ist fehlgeschlagen
NCP00053	Aktualisieren des Load Balancer-Pools ist fehlgeschlagen
NCP00054	Löschen des Load Balancer-Pools ist fehlgeschlagen
NCP00055	Erstellen der Load Balancer-Regel ist fehlgeschlagen
NCP00056	Aktualisieren des Load Balancer-Pools ist fehlgeschlagen
NCP00057	Löschen der Load Balancer-Regel ist fehlgeschlagen
NCP00058	IP-Freigabe für Load Balancer-Pool ist fehlgeschlagen
NCP00059	Zuordnung von virtuellem Server und Dienstzuordnung für Load Balancer nicht gefunden
NCP00060	Aktualisieren der NS-Gruppe ist fehlgeschlagen
NCP00061	Abrufen der Firewallregeln ist fehlgeschlagen
NCP00062	NS-Gruppe – keine Kriterien
NCP00063	Knoten-VM nicht gefunden
NCP00064	Knoten-VIF nicht gefunden
NCP00065	Zertifikatimport ist fehlgeschlagen
NCP00066	Rückgängigmachen des Zertifikats ist fehlgeschlagen
NCP00067	Aktualisieren der SSL-Bindung ist fehlgeschlagen
NCP00068	SSL-Profil nicht gefunden
NCP00069	IP-Pool nicht gefunden
NCP00070	T0-Edge-Cluster nicht gefunden
NCP00071	Aktualisieren des IP-Pools ist fehlgeschlagen
NCP00072	Dispatcher ist fehlgeschlagen
NCP00073	Löschen der NAT-Regel ist fehlgeschlagen
NCP00074	Abrufen des Ports für den logischen Router ist fehlgeschlagen
NCP00075	NSX-Konfigurationsvalidierung ist fehlgeschlagen

Fehlercode	Beschreibung
NCP00076	Aktualisieren der SNAT-Regel ist fehlgeschlagen
NCP00077	SNAT-Regel überlagert
NCP00078	Hinzufügen der Load Balancer-Endpoints ist fehlgeschlagen
NCP00079	Aktualisieren der Load Balancer-Endpoints ist fehlgeschlagen
NCP00080	Erstellen des Load Balancer-Regelpools ist fehlgeschlagen
NCP00081	Virtueller Server für Load Balancer nicht gefunden
NCP00082	Lesen von IP Set ist fehlgeschlagen
NCP00083	Abrufen von SNAT-Pool ist fehlgeschlagen
NCP00084	Erstellen des Load Balancer-Diensts ist fehlgeschlagen
NCP00085	Aktualisieren des Load Balancer-Diensts ist fehlgeschlagen
NCP00086	Aktualisieren des Ports für den logischen Router ist fehlgeschlagen
NCP00087	Load Balancer-Initialisierung ist fehlgeschlagen
NCP00088	IP-Pool nicht eindeutig
NCP00089	Layer 7 des Load Balancers – Cache-Synchronisierungsfehler
NCP00090	Fehler, da Load Balancer-Pool nicht vorhanden
NCP00091	Fehler beim Initialisieren des Load Balancer-Regelcaches
NCP00092	SNAT-Prozess ist fehlgeschlagen
NCP00093	Fehler bei Load Balancer-Standardzertifikat
NCP00094	Löschen des Load Balancer-Endpoints ist fehlgeschlagen
NCP00095	Projekt nicht gefunden
NCP00096	Pool-Zugriff verweigert
NCP00097	Fehler beim Abrufen eines Load Balancer-Diensts
NCP00098	Fehler beim Erstellen eines Load Balancer-Diensts
NCP00099	Fehler bei Synchronisierung des Load Balancer-Pool-Caches

Fehlercodes für NSX-Knoten-Agent

Fehlercode	Beschreibung
NCP01001	OVS-Uplink nicht gefunden
NCP01002	Host-MAC nicht gefunden
NCP01003	OVS-Porterstellung ist fehlgeschlagen
NCP01004	Keine Pod-Konfiguration
NCP01005	Pod-Konfiguration ist fehlgeschlagen
NCP01006	Aufheben der Pod-Konfiguration ist fehlgeschlagen
NCP01007	CNI-Socket nicht gefunden

Fehlercode	Beschreibung
NCP01008	CNI-Verbindung ist fehlgeschlagen
NCP01009	CNI-Version stimmt nicht überein
NCP01010	CNI-Nachrichtenempfang ist fehlgeschlagen
NCP01011	CNI-Nachrichtenübertragung ist fehlgeschlagen
NCP01012	Hyperbus-Verbindung ist fehlgeschlagen
NCP01013	Hyperbus-Version stimmt nicht überein
NCP01014	Fehler beim Empfang der Hyperbus-Nachricht
NCP01015	Hyperbus-Nachrichtenübertragung ist fehlgeschlagen
NCP01016	GARP-Senden ist fehlgeschlagen
NCP01017	Schnittstellenkonfiguration ist fehlgeschlagen

Fehlercodes für nsx-kube-proxy

Fehlercode	Beschreibung
NCP02001	Ungültiger Gateway-Port des Proxys
NCP02002	Proxy-Befehl ist fehlgeschlagen
NCP02003	Proxy-Validierung ist fehlgeschlagen

CLI-Fehlercodes

Fehlercode	Beschreibung
NCP03001	CLI-Start ist fehlgeschlagen
NCP03002	Erstellen des CLI-Sockets ist fehlgeschlagen
NCP03003	CLI-Socket-Ausnahme
NCP03004	Ungültige Anforderung von CLI-Client
NCP03005	CLI-Server-Übertragung ist fehlgeschlagen
NCP03006	CLI-Server-Empfang ist fehlgeschlagen
NCP03007	CLI-Befehlsausführung ist fehlgeschlagen

Fehlercodes für Kubernetes

Fehlercode	Beschreibung
NCP05001	Kubernetes-Verbindung ist fehlgeschlagen
NCP05002	Ungültige Konfiguration für Kubernetes
NCP05003	Kubernetes-Anforderung ist fehlgeschlagen
NCP05004	Kubernetes-Schlüssel nicht gefunden

Fehlercode	Beschreibung
NCP05005	Kubernetes-Typ nicht gefunden
NCP05006	Ausnahme bei Kubernetes-Wächter
NCP05007	Kubernetes-Ressource weist ungültige Länge auf
NCP05008	Kubernetes-Ressource weist ungültigen Typ auf
NCP05009	Kubernetes-Ressourcen-Handle ist fehlgeschlagen
NCP05010	Kubernetes-Dienst-Handle ist fehlgeschlagen
NCP05011	Kubernetes-Endpoint-Handle ist fehlgeschlagen
NCP05012	Kubernetes-Ingress-Handle ist fehlgeschlagen
NCP05013	Kubernetes-Netzwerkrichtlinien-Handle ist fehlgeschlagen
NCP05014	Kubernetes-Knoten-Handle ist fehlgeschlagen
NCP05015	Kubernetes-Namespace-Handle ist fehlgeschlagen
NCP05016	Kubernetes-Pod-Handle ist fehlgeschlagen
NCP05017	Kubernetes-Secret-Handle ist fehlgeschlagen
NCP05018	Kubernetes-Standard-Backend ist fehlgeschlagen
NCP05019	Nicht unterstützter Übereinstimmungsausdruck für Kubernetes
NCP05020	Aktualisieren des Kubernetes-Status ist fehlgeschlagen
NCP05021	Aktualisieren des Kubernetes-Kommentars ist fehlgeschlagen
NCP05022	Kubernetes-Namespace-Cache nicht gefunden
NCP05023	Kubernetes-Secret nicht gefunden
NCP05024	Kubernetes-Standard-Backend wird verwendet
NCP05025	Kubernetes-LoadBalancer-Dienst-Handle ist fehlgeschlagen

Fehlercodes für OpenShift

Fehlercode	Beschreibung
NCP07001	OC-Routen-Handle ist fehlgeschlagen
NCP07002	Aktualisieren des OC-Routenstatus ist fehlgeschlagen