Cisco Nexus 1000V for KVM Virtual Network Configuration Guide, Release 5.x

First Published: November 21, 2014

Last Modified: May 26, 2015

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

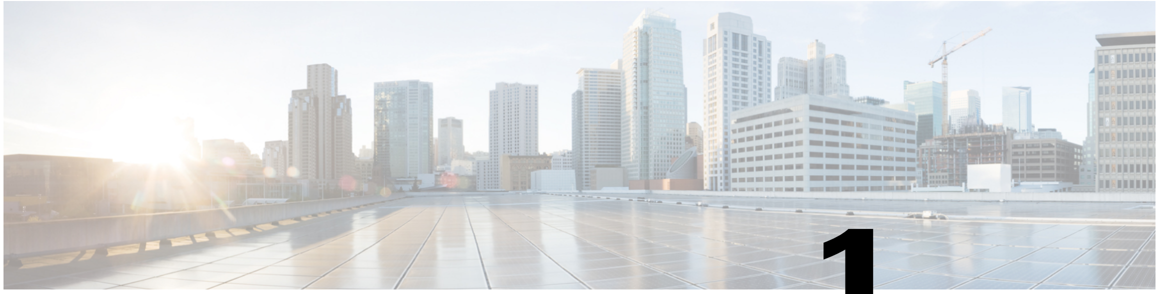
The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

CONTENTS

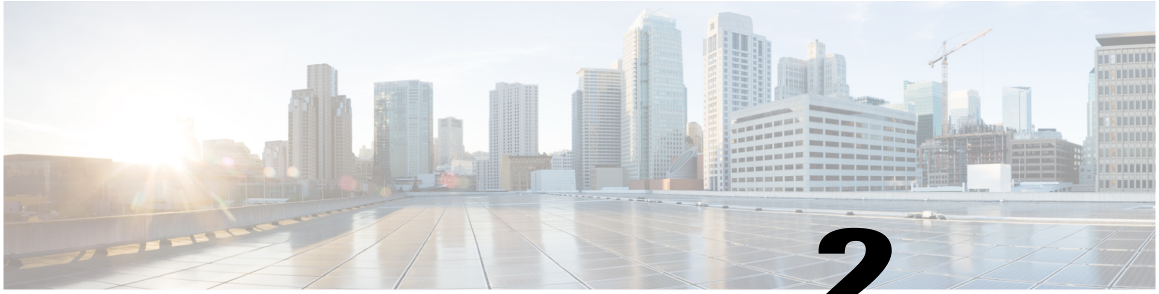


CHAPTER 1

New and Changed Information

Table 1: New and Changed Features

Content	Description	Changed in Release	Where Documented
Layer 3 Forwarding	This feature is introduced.	5.2(1)SK3(2.2)	Layer 3 Forwarding Overview, on page 35
Configuring VLAN Trunk vEthernet Ports	This feature is introduced.	5.2(1)SK3(2.2)	Configuring VLAN Trunk vEthernet Ports, on page 32
Neutron-to-VSM Configuration Synchronization	An automatic state mismatch check between VSM and Neutron is performed.	5.2(1)SK3(2.2)	Neutron-to-VSM Configuration Synchronization, on page 8
OpenStack Commands	cisco-credential-* and cisco-network-profile-* commands are no longer supported.	5.2(1)SK3(2.2)	OpenStack Command Reference, on page 43
Private VLANs	This feature is introduced.	5.2(1)SK3(2.1)	Information About Private VLANs, on page 29



CHAPTER 2

Overview

This chapter contains the following sections:

Cisco Nexus 1000V for KVM and OpenStack

The Cisco Nexus 1000V for KVM consists of two main components:

- Virtual Ethernet Module (VEM)—A software component that is deployed on each KVM host. Each VM on the host is connected to the VEM through virtual Ethernet (vEth) ports.
- Virtual Supervisor Module (VSM)—The Management component that controls multiple VEMs and helps in the definition of VM-focused network policies. It is deployed either as a virtual appliance on any KVM host or on the Cisco Cloud Services Platform appliance.

Each of these components is tightly integrated with the OpenStack environment:

- The VEM is a hypervisor-resident component and is tightly integrated with the KVM architecture.

- The VSM is integrated with OpenStack using the OpenStack Neutron Plug-in.
- The OpenStack Neutron API has been extended to include two additional user-defined resources:
  - Network profiles as logical groupings of network segments.

    Note In Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2), network profiles are created automatically for each network type. Network profile creation by administrators is not supported.

  - Policy profiles group port policy information, including security.

Using OpenStack, you create VMs, networks, and subnets on the Cisco Nexus 1000V for KVM, by defining components such as the following:

- Tenants
- Network segments, such as VLANs, VLAN trunks, and VXLANs
- IP address pools (subnets)

Using the Cisco Nexus 1000V for KVM VSM, you create port profiles (called policy profiles in OpenStack), which define the port policy information, including security settings.

When a VM is deployed, a port profile is dynamically created on the Cisco Nexus 1000V for KVM for each unique combination of policy port profile and network segment. All other VMs deployed with the same policy to this network reuse this dynamic port profile.

Note You must consistently use OpenStack for all VM network, subnet, and port configurations. If you create VM networks, subnets, and ports directly on the VSM, the configuration is lost when the OpenStack synchronization occurs.

Tenants

OpenStack has a concept of identity user management called a tenant (also called a project). A tenant is a container used to group resources and/or identity objects. Depending on the how OpenStack is being deployed, a tenant might correspond to a customer, account, organization, or project.

Network Segments

A network segment is an isolated Layer 2 network with a unique broadcast domain (similar to a VLAN). A network segment also facilitates the availability of the network resources to a virtual machine. In OpenStack, a network segment is a VLAN or VXLAN type of network, which provides isolation on virtual networks.

You create a virtual network on the OpenStack Controller using the OpenStack dashboard or the OpenStack CLI commands. When you create a virtual network of type VLAN or VXLAN on the OpenStack controller, OpenStack triggers the auto-creation of a network segment with VLANs or VXLANs on the VSM.

For information about how to create a virtual network, see one the following chapters:

- Creating a Virtual Network Using the OpenStack Dashboard
- Creating a Virtual Network Using the OpenStack CLI

Policy and Network Separation

In the Cisco Nexus 1000V for OpenStack environment, features and network segments are independently associated with the interfaces. The independent association allows you to assign the same set of features on the interfaces that are spread across multiple dynamically-allocated network segments. With this capability, a network administrator can define the policy profiles and export policy profiles to the OpenStack environment. The OpenStack cloud administrator can allocate the network segments from the network pools dynamically, and associate the virtual machine (VM) interfaces to the policy profile and the allocated network segment. This decoupling provides the flexibility to allocate network segments dynamically while grouping the network features to be applied on the interfaces.

IP Pool Templates

An IP pool template represents a block of IP addresses and other network configuration (for example, default gateways or DNS servers) that can be assigned to VMs on a given network. The IP pool templates are the address templates that are applied to the network segments.

The server administrator manages the IP addresses for the virtual environment and assigns a range of IP addresses to the hosts and to the virtual machines that are running inside the OpenStack-managed environment. When creating a subnet for a VM network, the network administrator assigns a range of IP addresses that can be used by the VMs in the network.

The IP pool templates can be reused in the environments with the same IP Address spacing, for example, the duplicate IP addresses are used on the different network segments.

Port Profiles

A port profile is a collection of the interface-level configuration attributes. The network administrator creates a consistent network policy across the similar VM interfaces by defining the Virtual Ethernet port profiles. The network administrator can also create a port profile for the VM hosts adapters. The profile defines the policy to be applied on the physical Ethernet adapters on the servers.

Dynamic Port Profiles

When a VM is deployed, a port profile is dynamically created on the Cisco Nexus 1000V for each unique combination of the Port Classification, the VM Network, and the VM subnet. All other VMs deployed with the same policy to this network reuse this dynamic port profile. This dynamic port profile is a combination of network isolation and network policy.

Note The auto-generated profile should not be modified, inherited in any other port profiles, or referenced in any other configuration. Any changes or references should be in the port-profile inherited by the dynamic port-profile.

When a port-attach notification is received, the port profile UUID and the network segment UUID are generated. A UUID is a globally unique identifier that is used to provide a unique reference for the port profile and the network segment. When a UUID is generated, a new dynamic port profile is created on the VSM that combines the policy profile and the network segment (VLAN/VXLAN). This automatically created port profile is inherited on the interface. If more than one port uses the same combination of the port profile and the network segment, the port profile is shared. The port profiles are dynamically created during the interface attach process.

Bridge Domain

A bridge domain is a Layer 2 flood domain, used for Layer 2 isolation of ports. A bridge domain is distinguished by an identifier, such as a VXLAN segment ID.

Types of OpenStack Networks

Before creating a network using OpenStack, it is important to understand how OpenStack defines these types of networks:

- Virtual network—An OpenStack networking Layer 2 network (identified by a universally unique identifier [UUID] and optional name) whose ports can be attached as vNICs to OpenStack compute instances and to various OpenStack networking agents.
- Physical network—A network connecting virtualization hosts (i.e. OpenStack compute nodes) with each other and with other network resources. Each physical network may support multiple virtual networks. The provider extension and the plugin configurations identify physical networks using simple string names.
- Tenant network—A typical virtual network created by or for a tenant. The tenant is not aware of how that network is physically realized.
- Provider network—A virtual network administratively created to map to a specific network in the data center, typically to enable direct access to non-OpenStack resources on that network. Tenants can be given access to provider networks.
- VLAN Trunk network—A virtual network realized as packets on a specific physical network containing IEEE 802.1Q headers with a specific VLAN ID (VID) field value. VLAN networks that share the same physical network are isolated from each other at Layer 2, and can even have overlapping IP address spaces. Each distinct physical network that supports VLAN networks is treated as a separate VLAN trunk, with a distinct space of VID values. Valid VID values are 1 through 4094.
- Flat network—A virtual network realized as packets on a specific physical network containing no IEEE 802.1Q header. Each physical network can realize at most one flat network.
- Local network—A virtual network that allows communication within each host, but not across a network. Local networks are intended mainly for single-node test scenarios, but may have other uses.

Comparison of Network Terminology

Cisco Nexus 1000V for KVM and OpenStack use many of the same components and concepts. However, they have given these components and concepts different terminology. The following table defines these components and concept and maps the ones that are different.

Cisco Nexus 1000V for KVM	OpenStack	Description
—	Linux KVM	Linux Kernel-based virtual machine that functions as a hypervisor.
—	OpenStack Controller	Point of management.
—	Neutron	Point of network management.
Logical network	Container object	Server nodes (virtual machines), network nodes, and network services that logically isolate network traffic and partition needed resources.
Network segment pool	Cisco network profile	A container that allows you to associate IP address blocks and other network configuration settings with a neutron network. OpenStack supports VLAN, overlay (VXLAN), and trunk types.
Network segment	Network	Represents an isolated virtual Layer 2 network domain (similar to a VLAN); Can also be regarded as a virtual or logical switch.
IP pool template	Subnet	Represents a block of IP addresses and other network configuration (for example, default gateways or DNS servers) that are assigned to VMs on a given network.
Network vEthernet	—	The combination of a network segment and a port profile policy.
Network vEthernet port	port	Ports that represent virtual (or logical) switch ports on a given network.

Cisco Nexus 1000V for KVM	OpenStack	Description
Dynamic port profile	—	An automatically generated combination of a policy port profile and network segment. Dynamic port profiles have **vmn** as a prefix.
Bridge domain	—	A bridge domain object is created only in the Virtual Supervisor Module (VSM) and not in OpenStack. When a VXLAN network is created, Openstack requests the creation of a bridge domain in VSM. The newly created bridge-domain is used to configure the VXLAN network segment.

Neutron-to-VSM Configuration Synchronization

In order to keep the Neutron service and the VSM configurations in synchronization, the Cisco Nexus 1000V Neutron Plug-in has the capability to restore the configuration in the VSM under certain situations.

Rollback for Neutron Resources

The Neutron service rolls back to the previous configuration if it fails to create or update a resource on the VSM. If the Neutron service sends a resource request to the VSM and the VSM responds with an HTTP error, the Neutron service deletes the resource and all of its associated bindings and logs an exception in the Neutron server logs.

State Synchronization

Note Starting with Release 5.2(1)SK3(2.2), you no longer have to restart the Neutron service to trigger a full synchronization. However, a bridge domain synchronization between the Neutron service and the VSM only happens when the Neutron service restarts.

An automatic state mismatch check on the VSM is performed every five minutes unless you change the default duration. To change the default duration, edit the **sync_interval** parameter located in the /etc/neutron/plugin.ini file. If there is a state mismatch, a create or delete operation on the VSM is performed to get it in sync with the Neutron. The resources that are synchronized include network profiles, networks, subnets, ports, and bridge domains. If there are only certain resources out of sync on the VSM, synchronization will occur only for those resources.

Policy profiles that are missing from the VSM or in use in Neutron are not restored automatically as part of a full synchronization. You must manually create them on the VSM using the same UUID values before you restart the Neutron server to trigger a full synchronization.

Synchronizing a Fresh VSM

Use this procedure to perform a full synchronization of the VSM after a reload or as a part of the VSM recovery (fresh VSM bring-up).

Before You Begin

Before beginning this procedure, you must know or do the following:

- You are logged into the CLI in EXEC mode.

Procedure

Step 1 From the old VSM, copy the running configuration to an external location.
copy running-config tftp://*external-location*

Step 2 Generate the **selective-config-file**.

```
show running config static
```

Step 3 Enable the management communication using either Telnet or Secure Shell (SSH).

a) On the Nova-Cloud-Controller (Neutron Server), locate the `cisco_plugins.ini` configuration file at the following path: `/etc/neutron/plugins/cisco`.
b) In the `cisco_n1K` section, add enable_sync_on_start=True in the `cisco_plugins.ini` file.

Step 4 Restart the Neutron server to start the Neutron-VSM synchronization process.

```
root@ncc:~# service neutron-server restart
neutron-server stop/waiting
neutron-server start/running, process 24157
```

Step 5 Verify if all the configurations and vEthernet interface have come up on the VSM.

```
show running-config
```

CHAPTER 3

Configuring Network Segmentation Manager

This chapter contains the following sections:

Network Segments

A network segment is an isolated Layer 2 network with a unique broadcast domain (similar to a VLAN). A network segment also facilitates the availability of the network resources to a virtual machine. In OpenStack, a network segment is a VLAN or VXLAN type of network, which provides isolation on virtual networks.

You create a virtual network on the OpenStack Controller using the OpenStack dashboard or the OpenStack CLI commands. When you create a virtual network of type VLAN or VXLAN on the OpenStack controller, OpenStack triggers the auto-creation of a network segment with VLANs or VXLANs on the VSM.

For information about how to create a virtual network, see one the following chapters:

- Creating a Virtual Network Using the OpenStack Dashboard
- Creating a Virtual Network Using the OpenStack CLI

Prerequisites

Network Segmentation Manager has the following prerequisites:

- You have installed the Cisco Nexus 1000V software and configured the VSM using the *Cisco Nexus 1000V for KVM Software Installation Guide*.

Guidelines and Limitations

The network segmentation manager feature has the following configuration guidelines and limitations:

- The **network-segmentation-manager** feature is enabled on the VSM by default. Verify the output of the **show feature** command on the VSM to make sure that the **network-segmentation-manager** feature is enabled by default.
- The OpenStack controller should be able to communicate with the Cisco Nexus 1000V using HTTP.
- The **http-server** feature is enabled by default on the Cisco Nexus 1000V to allow web service communication.

Enabling and Disabling the Network Segmentation Manager Feature

The Network Segmentation Manager feature is enabled by default on the VSM. However, if you need to, you can enable or disable it.

Before You Begin

You must be logged in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature network-segmentation-manager**	Enables the Network Segmentation Manager feature. To disable this feature, use the **no feature network-segmentation-manager** command.
Step 3	switch(config)# **show feature**	(Optional) Displays the status for Cisco Nexus 1000V features.

This example shows how to enable the NSM feature and verify that it is enabled:

```
switch# configure terminal
switch(config)# feature network-segmentation-manager
switch# show feature
Feature Name          Instance  State
--------------------  --------  --------
http-server           1         enabled
lacp                  1         disabled
netflow               1         enabled
network-segmentation  1         enabled
port-profile-roles    1         disabled
privilege             1         disabled
scpServer             1         disabled
segmentation          1         enabled
sftpServer            1         disabled
```

```
sshServer            1         enabled
tacacs               1         disabled
telnetServer         1         enabled
vxlan-gateway        1         disabled
switch(config)#
```

Verifying the NSM Configuration

Use one of the following commands to verify the configuration:

- **show nsm ip pool template**
- **show nsm ip pool template filter description** *description*
- **show nsm ip pool template usage network segment**
- **show nsm logical network name <name>**
- **show nsm network segment brief**
- **show nsm network segment filter description** *description*
- **show nsm network segment filter network segment pool name <name>**
- **show nsm network segment filter vlan <vlan_ID>**
- **show nsm network segment filter network segment pool <name>**
- **show nsm network segment name <name>**
- **show nsm network segment pool filter description** *description*
- **show nsm network segment pool name <name>**
- **show dynamic-port-profile**
- **show dynamic-port-profile <name>**
- **show dynamic-port-profile inherit <name>**
- **show dynamic-port-profile network segment <name>**

show nsm ip pool template

This show command displays an IP pool template of a given name.

```
switch# show nsm ip pool template
Name: 00683778-cbd4-4e76-b181-bd562b6a1b3d
  Description: subnet-vlan-39
  IP-address-range: 39.1.1.2-39.1.1.254
  Network: 39.1.1.0
  Subnet mask: 255.255.255.0
  Default router: 39.1.1.1
  Netbios: Disabled
  DHCP: Enabled
  Reserved-ip-list:
  Netbios-name-server-list:
  DNS-server-list:
  DNS-suffix-list:
switch#
```

show nsm ip pool template filter description

This show command displays a specific IP pool template based on its description. (The description in the VSM is the name in OpenStack.)

```
switch# show nsm ip pool template filter description sub-10-1
Name: d259d433-3e5c-491b-afda-787ddc260dea
  Description: sub-10-1
  IP-address-range: 10.10.1.2-10.10.1.254
  Network: 10.10.1.0
  Subnet mask: 255.255.255.0
```

show nsm ip pool template usage network segment

The following show command displays the network using an IP pool template.

```
switch# show nsm ip pool template usage network segment
Ip-pool: 00683778-cbd4-4e76-b181-bd562b6a1b3d
51c652ca-b118-41ea-b3ff-f02bb2ac934b
switch#
```

show nsm logical network name <name>

This command displays the Logical Network of a given name.

```
switch# show nsm logical network name 9a8d49b6-4590-47a5-8ecd-8616276694d2_log_net
Name: 9a8d49b6-4590-47a5-8ecd-8616276694d2_log_net
  Description: seg-pool-11-310
```

show nsm network segment brief

This command displays information about mode, VLAN, publish status, and the system segment status for all the network segments.

```
switch# show nsm network segment brief

--------------------------------------------------------------------------------
Network segment               Mode           VLAN      Pub       Sys
--------------------------------------------------------------------------------
0200362d-0d69-44bc-8f2d-40685f474ddf
                              access         63        1         0
027f02fa-2854-40d2-a0ad-04cd37025cab
                              access         20        1         0
03625912-ce1b-4e53-ae14-88255f2f1de7
                              access         17        1         0

--------------------------------------------------------------------------------
Total          Total Pub      Total Sys
--------------------------------------------------------------------------------
3              3              0
```

show nsm network segment filter description

This command displays a specific network segment based on its description. (The description in the VSM is the name in OpenStack.)

```
switch# show nsm network segment filter description net-10-1
Name: 3a43c169-bbf9-404d-abf0-3580b9a7113e
  Description: net-10-1
  UUID: 3a43c169-bbf9-404d-abf0-3580b9a7113e
  Network segment pool: 39e45a8d-8ecd-4bb0-9666-6ddcec2cfefc
  Mode: switchport mode access
  Vlan: 1090
  System Network Segment: FALSE
  ip pool template: d259d433-3e5c-491b-afda-787ddc260dea
  ip pool template UUID: d259d433-3e5c-491b-afda-787ddc260dea
  Publish-name: 3a43c169-bbf9-404d-abf0-3580b9a7113e
```

show nsm network segment filter network segment pool <name>

This command displays all network segments that are part of a given network segment pool.

```
switch# show nsm network segment filter network segment pool
9a8d49b6-4590-47a5-8ecd-8616276694d2
Name: 0200362d-0d69-44bc-8f2d-40685f474ddf
  Description: vlan-seg-63
  UUID: 0200362d-0d69-44bc-8f2d-40685f474ddf
  Network segment pool: 9a8d49b6-4590-47a5-8ecd-8616276694d2
  Mode: switchport mode access
  Vlan: 63
  System Network Segment: FALSE
  ip pool template: c3a3f619-1a80-402c-b05d-829ce4eaed8f
  ip pool template UUID: c3a3f619-1a80-402c-b05d-829ce4eaed8f
  Publish-name: 0200362d-0d69-44bc-8f2d-40685f474ddf
```

show nsm network segment filter vlan <vlan_ID>

This command displays the network segment that is using a given VLAN ID.

```
switch# show nsm network segment filter vlan 70
Name: 34e94f30-4ed5-48dc-8e60-820e125692d8
  Description: vlan-seg-70
  UUID: 34e94f30-4ed5-48dc-8e60-820e125692d8
  Network segment pool: 9a8d49b6-4590-47a5-8ecd-8616276694d2
  Mode: switchport mode access
  Vlan: 70
  System Network Segment: FALSE
  ip pool template: b5a716d4-b2d6-45fa-b685-806947ed48b0
  ip pool template UUID: b5a716d4-b2d6-45fa-b685-806947ed48b0
  Publish-name: 34e94f30-4ed5-48dc-8e60-820e125692d8
switch#
```

show nsm network segment name <name>

The following show command displays the details of the network segment.

```
switch# show nsm network segment name 1c3046fb-d33c-4156-9b7d-ac0fb74f5891
Name: 1c3046fb-d33c-4156-9b7d-ac0fb74f5891
  Description: vlan-seg-62
  UUID: 1c3046fb-d33c-4156-9b7d-ac0fb74f5891
  Network segment pool: 9a8d49b6-4590-47a5-8ecd-8616276694d2
  Mode: switchport mode access
  Vlan: 62
  System Network Segment: FALSE
  ip pool template: 2e88cb6c-5a7a-4916-a17e-126d1dc370d2
  ip pool template UUID: 2e88cb6c-5a7a-4916-a17e-126d1dc370d2
  Publish-name: 1c3046fb-d33c-4156-9b7d-ac0fb74f5891
switch#
```

show nsm network segment pool filter description

The following show command displays a specific network segment pool based on its description. (The description in the VSM is the name in OpenStack.)

```
switch# show nsm network segment pool filter description vm-pool1
Name: 39e45a8d-8ecd-4bb0-9666-6ddcec2cfefc
  Description: vm-pool1
  UUID: 39e45a8d-8ecd-4bb0-9666-6ddcec2cfefc
  Logical network Name: 39e45a8d-8ecd-4bb0-9666-6ddcec2cfefc_log_net
  Intra Port Communication: Disabled
  Publish-name: 39e45a8d-8ecd-4bb0-9666-6ddcec2cfefc
```

show nsm network segment pool name <name>

The following show command displays which network segments are used by a given network segment pool.

```
switch# show nsm network segment pool name 9a8d49b6-4590-47a5-8ecd-8616276694d2
Name: 9a8d49b6-4590-47a5-8ecd-8616276694d2
  Description: seg-pool-11-310
  UUID: 9a8d49b6-4590-47a5-8ecd-8616276694d2
  Logical network Name: 9a8d49b6-4590-47a5-8ecd-8616276694d2_log_net
  Intra Port Communication: Disabled
  Publish-name: 9a8d49b6-4590-47a5-8ecd-8616276694d2
switch#
```

show dynamic-port-profile

The following show command displays a list of all the dynamically created profiles.

```
switch# show dynamic-port-profile
dynamic-port-profile:
vmn_f58d3545-a0a1-4441-8b7e-1a7c8339524b_0200362d-0d69-44bc-8f2d-40685f474ddf
  inherit port-profile: dhcp_pp
  network segment: 0200362d-0d69-44bc-8f2d-40685f474ddf
dynamic-port-profile:
vmn_f58d3545-a0a1-4441-8b7e-1a7c8339524b_027f02fa-2854-40d2-a0ad-04cd37025cab
  inherit port-profile: dhcp_pp
  network segment: 027f02fa-2854-40d2-a0ad-04cd37025cab
switch#
```

show dynamic-port-profile name <name>

The following show command displays a specific dynamic port profile.

```
switch# show dynamic-port-profile name
vmn_f58d3545-a0a1-4441-8b7e-1a7c8339524b_3ff2d845-e587-4bdd-8737-75044e99a7c7
dynamic-port-profile:
vmn_f58d3545-a0a1-4441-8b7e-1a7c8339524b_3ff2d845-e587-4bdd-8737-75044e99a7c7
  inherit port-profile: dhcp_pp
  network segment: 3ff2d845-e587-4bdd-8737-75044e99a7c7
switch#
```

show dynamic-port-profile inherit <name>

The following show command displays the list of dynamic port profile inheriting a specific vEthernet policy profile.

```
switch# show dynamic-port-profile inherit dhcp_pp
dynamic-port-profile:
vmn_f58d3545-a0a1-4441-8b7e-1a7c8339524b_0200362d-0d69-44bc-8f2d-40685f474ddf
  inherit port-profile: dhcp_pp
  network segment: 0200362d-0d69-44bc-8f2d-40685f474ddf
dynamic-port-profile:
vmn_f58d3545-a0a1-4441-8b7e-1a7c8339524b_027f02fa-2854-40d2-a0ad-04cd37025cab
  inherit port-profile: dhcp_pp
  network segment: 027f02fa-2854-40d2-a0ad-04cd37025cab
switch#
```

show dynamic-port-profile network-segment <name>

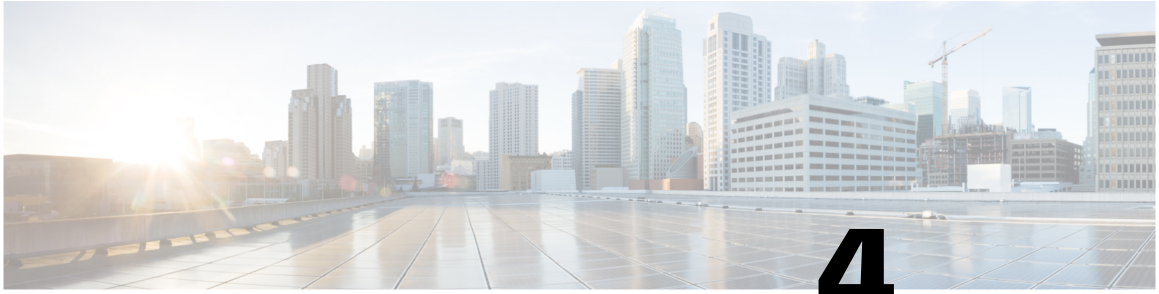
The following show command displays the list of dynamic port profile using a given network segment.

```
switch# show dynamic-port-profile network segment 03625912-ce1b-4e53-ae14-88255f2f1de7
dynamic-port-profile:
vmn_f58d3545-a0a1-4441-8b7e-1a7c8339524b_03625912-ce1b-4e53-ae14-88255f2f1de7
  inherit port-profile: dhcp_pp
  network segment: 03625912-ce1b-4e53-ae14-88255f2f1de7
switch#
```

Feature History for Network Segmentation Manager

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Release	Feature Information
Network Segmentation Manager	5.2(1)SK1(2.1)	Introduced the Network Segmentation Manager (NSM) feature.



CHAPTER 4

Configuring Virtual Networks Using OpenStack

This chapter contains the following sections:

Information About Virtual Networks

This chapter provides general information about using the OpenStack dashboard to create several different types of virtual networks. For specific information about implementing virtual network components to deploy the VXLAN Gateway, see the VXLAN Configuration Guide or the Cisco Nexus 1000V for KVM Installation Guide.

Guidelines and Limitations for the OpenStack Dashboard

The OpenStack dashboard has the following guidelines and limitations when you use it to create virtual networks for Cisco Nexus 1000V for KVM:

- Network profile creation by an administrator is not supported in Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and highter. Network profiles are automatically created for each network type.
- To create a network profile and associate it with a tenant, you must log in to the OpenStack dashboard as a user with admin privileges. Any user can use a network profile that is associated with a tenant.
- You cannot create policy profiles or assign them to a tenant in OpenStack dashboard. You must first create them as part of the port profiles in the VSM. The OpenStack dashboard retrieves them from the VSM and displays them on the **Router** dashboard.
- When there are multiple VSMs, the port profile must be configured on all the VSMs.

Creating a Virtual Network Workflow

This workflow describes how to create a virtual network.

Steps	Notes
1. Create tenants.	See Creating a Tenant Using the OpenStack Dashboard, on page 20.
2. Create policy and port profiles.	See the *Cisco Nexus 1000V for KVM Port Profile Configuration Guide* .
3.Create the network using the OpenStack Dashboard or OpenStack CLI:	
- Create a network using the OpenStack Dashboard	
1. Create a network profile **Note** This step is not necessary for Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2).	See Creating a Network Profile Using the OpenStack Dashboard, on page 21. Network profile creation by an administrator is not supported in Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and highter. Network profiles are automatically created for each network type.
2. Create a network	See Creating a Virtual Network Using OpenStack Dashboard, on page 23.
3. Create a network subnet	See Creating a Subnet for a Network Using the OpenStack Dashboard, on page 23.
4. Create and launch a VM instance	See Creating and Launching a VM Instance Using the OpenStack Dashboard, on page 24.
- Create a network using the OpenStack CLI	See Creating VLAN and VXLAN Networks Using the OpenStack CLI, on page 24.

Creating a Tenant Using the OpenStack Dashboard

In the OpenStack dashboard, tenants are also known as projects.

Procedure

Step 1 In the **Navigation** pane, click **Admin** > **Projects**.

Step 2 In the **Projects** panel, click **Create Project**.

Step 3 In the **Create Project** dialog box, complete the following fields on the **Project Info** tab:

a) In the **Name** field, enter a unique name for the project.
The name can have a maximum length of 255 characters, and can contain uppercase or lowercase characters, numerals, and special characters such as an "at" sign (@), ampersand (&), and exclamation point (!).

b) (Optional) In the **Description** field, enter a description for the project.

c) In the **Enabled** check box, check the box if you want to enable the project.

Step 4 On the **Project Members** tab, click the + button for all members that you want to add to the project.

Step 5 On the **Quota** tab, change the defaults in the fields if desired.

Step 6 Click **Create Project**.

What to Do Next

Create the desired network profiles.

Creating a Network Using the OpenStack Dashboard

Creating a Network Profile Using the OpenStack Dashboard

Note This procedure is not required for Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2). Network profiles are created automatically for each network type.

Before You Begin

- Create one or more policy profiles as part of the port profiles in the VSM.
- Create one or more tenant in the OpenStack dashboard.

Procedure

Step 1 In the **Navigation** pane, click the **Router** dashboard.

Step 2 In the **Cisco Nexus 1000v** panel, click **Create Network Profile**.

Step 3 In the **Create Network Profile** dialog box, do the following:

a) In the **Name** field, enter a unique name for the network profile.
The name can have a maximum length of 255 characters, and can contain uppercase or lowercase characters, numerals, and special characters such as an "at" sign (@), ampersand (&), and exclamation point (!).

b) From the **Segment Type** drop-down list, choose one of the following:

- **VLAN—**For networks as access mode.
- **Overlay—**For VXLAN.
- **Trunk—**For networks as trunk mode.

c) From the **Sub Type** drop-down list, choose a sub type. The sub type that you can choose depends on the segment type that you chose:

Chosen Segment Type	Possible Sub Types
VLAN	**None**
Overlay	• **Enhanced** for unicast VXLAN • **Native VXLAN** for multicast VXLAN
Trunk	**VLAN**

d) In the **Segment Range** field, enter the segment range for the network profile.
Separate the first and last segments in the range with a hyphen (-). For example, enter a range of 80-86. If the segment type is **VLAN**, the range can be from 1 to 3967, or from 4048 to 4093. If the segment type is **Overlay**, the range can be from 4096 to 16000000.

e) In the **Physical Network** field, enter the name of the associated physical network.

f) If you chose the **Overlay** segment type, enter the IP address range in the **Multicast IP Range** field.
Separate the first and last IP addresses in the range with a hyphen (-). The reserved multicast IP address range is 224.0.0.0 to 224.0.0.255.

g) If you chose the **Other** segment type, complete the **Other** field.
Complete this field with a string only if you need to specify a network profile subtype that is not one of the subtypes that is currently supported and available in the drop-down list.

h) From the **Project** check box, check a tenant that you want to associate with this network profile.

i) Click **Create Network Profile**.

OpenStack dashboard creates the network profile and then updates the OpenStack Neutron database and the VSM.

What to Do Next

Create one or more networks.

Creating a Virtual Network Using OpenStack Dashboard

Before You Begin

Create one or more port profiles in the VSM. These port profiles are displayed as policy profiles in OpenStack dashboard. For more information, see *Cisco Nexus 1000V for KVM Port Profile Configuration Guide* .

Procedure

Step 1 If you have not already done so, log in to OpenStack dashboard as a user with admin privileges.

Step 2 Create a tenant.

Note If using Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2), skip Steps 3 and 4. Network profiles are created automatically for each network type.

Step 3 Create a network profile of type trunk.

Step 4 Create a network profile of type VLAN.

Step 5 Create a network.

Step 6 Create a subnet for the network.
You do not need to create a port for the network. OpenStack dashboard creates a port for the network when you launch the instance.

Step 7 Create and launch the virtual machine (VM) instance.

Creating a Subnet for a Network Using the OpenStack Dashboard

Procedure

Step 1 In the **Navigation** pane, click **Admin** > **Networks**.

Step 2 In the **Networks** panel, click the network to which you want to add a subnet.

Step 3 In the **Create Subnet** dialog box, click the **Subnet** tab an do the following:

a) In the **Name** field, enter a unique name for the subnet.
The name can have a maximum length of 255 characters and can contain uppercase or lowercase characters, numerals, and special characters such as an "at" sign (@), ampersand (&), and exclamation point (!).

b) In the **Network Address** field, enter the address for the subnet.
The subnet address must be in classless interdomain routing (CIDR) format. For example, 192.168.0.0/16.

c) From the **IP Version** drop-down list, choose IPv4.

d) (Optional) In the **Gateway IP** field, enter a gateway IP address for the subnet.

Step 4 Optionally, click the **Subnet Detail** tab and do the following:

a) (Optional) Click the **Enable DHCP** checkbox.

b) Enter one or more allocation pools in the **Allocation Pools** text box.

c) Enter one or more name servers in the **DNS Name Servers** text box.

d) Enter one or more host routes in the **Host Routes** text box.

Step 5 Click **Create** to create the subnet.

Creating and Launching a VM Instance Using the OpenStack Dashboard

Procedure

Step 1 From the **Current Project** drop-down list, choose the project in which you created the network.

Step 2 In the **Navigation** pane, click the **Project** dashboard.

Step 3 In the **Instances** panel, click **Launch Instance**.

Step 4 On the **Details** tab of the **Launch Instance** dialog box, do the following:

a) From the **Instance Source** drop-down list, choose **Image**.

b) From the **Image** drop-down list, choose the image you want to associate with the instance.
For VXLAN Gateway, choose the VXLAN Gateway image.

c) In the **Name** field, enter a unique name for the instance.
The name can have a maximum length of 255 characters, and can contain uppercase or lowercase characters, numerals, and special characters such as an "at" sign (@), ampersand (&), and exclamation point (!).

Step 5 On the **Networking** tab of the **Launch Instance** dialog box, do the following:

a) In the **Networks** area, check the check box for the networks that you want to assign to the instance.
The networks should be the networks you created previously.

b) From the **Policy Profiles** drop-down list, choose the policy profile that you want to assign to the network.

Step 6 Click **Launch**.
The OpenStack dashboard creates the instance and launches it.

Creating VLAN and VXLAN Networks Using the OpenStack CLI

You can create a virtual network for VLAN and VXLAN traffic using the using the OpenStack CLI.

Note This procedure is not required for Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2). Network profiles are created automatically for each network type.

Step	Command	Description
1. Create one of the following types of networks:		

Step	Command	Description
• VLAN network	**neutron cisco-network-profile-create** *name* **vlan --segment_range***segment-range* **--physical_network** *network*	Creates a VLAN type network profile. For more information about this command, see the cisco-network-profile-create, on page 48 command reference page.
• Trunk network	**neutron cisco-network-profile-create** *name* **trunk --sub_type vlan**	Creates a trunk type of network profile with a sub type of VLAN. For more information about this command, see the cisco-network-profile-create, on page 48 command reference page.
• Trunk network	**neutron cisco-network-profile-create** *name* **overlay --subtype native_vxlan --segment_range** *segment-range* **--multicast_ip_range** *ip-range*	Creates a multicast VXLAN type network profile. For more information about this command, see the cisco-network-profile-create, on page 48 command reference page.
• Multicast VXLAN network	**neutron cisco-network-profile-create** *name* **overlay --subtype enhanced --segment_range** *segment-range*	Creates a unicast VXLAN type network profile. For more information about this command, see the cisco-network-profile-create, on page 48 command reference page.
• Unicast VXLAN network	**neutron net-create** *name* **--n1kv:profile_id** *networkProfileId* For Release 5.2(1)SK3(2.2) or higher use the following command: **neutron net-create** *name* **--n1kv:profile** *networkProfileID*	Creates a network and associates it with a Cisco Nexus 1000V switch network profile.
3. Create the subnet.	**neutron subnet-create** *network-name IP-address-range* **--name** *subnet-name*	For more information about this command, see the OpenStack documentation

Step	Command	Description
4. Create a port profile.	**neutron port-create** *network-name* **--n1kv:profile_id** *policyProfileID* For Release 5.2(1)SK3(2.2) or higher use the following command: **neutron net-create** *name* **--n1kv:profile** *PolicyProfileID* or *PolicyProfileName*	Creates ports and associates them with either the policy profile UUID or policy profile name.
5. Bring up the virtual machine with the network.	**nova boot --image** *image-id* **--flavor** *flavor-id* **--nic port-id** *=port-id vm-name*	For more information about this command, see the OpenStack documentation.

Note The profile_id in the neutron net-create command refers to the network profile ID. The profile_id in the neutron port-create command refers to the policy profile ID.

The following example shows how to create a VLAN network:

```
$
$ neutron cisco-network-profile-create netprof vlan --segment_range 100-200 --physical_network
 physnet1
$ neutron net-create NetworkOne --n1kv:profile_id a9355268-5aed-8030-f3ab-e367ef4c9acc
$ neutron subnet-create NetworkOne 172.23.181.0/24 --name subnet1
$ neutron port-create NetworkOne --n1kv:profile_id b9b8d5fa-41a3-4e59-bb1e-6a5e296908e1
$ nova boot --image image-name --flavor m1.medium --nic
port-id=d341926c-21ca-48cd-ae18-c51f899f6d3f VM-1
```

The following example shows how to create a VLAN trunk network:

```
$ neutron cisco-policy-profile-update polprofId --add-tenant 1234-1234-1234-1234
$ neutron cisco-network-profile-create trunkprof trunk --sub_type vlan
$ neutron net-create NetworkOne --n1kv:profile_id b9b8d5fa-41a3-4e59-bb1e-6a5e296908e1
$ neutron port-create NetworkOne --n1kv:profile_id a9355268-5aed-8030-f3ab-e367ef4c9acc
$ nova boot --image image-name --flavor m1.medium --nic
port-id=d341926c-21ca-48cd-ae18-c51f899f6d3f --nic
port-id=7acf56b5-2d0d-e35d-def7-bdbe3960ea30 VM-1
```

The following example shows how to create a multicast VXLAN type network:

```
$ neutron cisco-policy-profile-update polprofId --add-tenant 1234-1234-1234-1234
$ neutron cisco-network-profile-create netprof overlay --subtype native_vxlan --segment_range
 5000-5300 --multicast_ip_range 224.99.0.0-224.99.0.1
$ neutron net-create NetworkOne --n1kv:profile_id b9b8d5fa-41a3-4e59-bb1e-6a5e296908e1
$ neutron port-create NetworkOne --n1kv:profile_id a9355268-5aed-8030-f3ab-e367ef4c9acc
$ nova boot --image image-name --flavor flavor-id --nic
port-id=d341926c-21ca-48cd-ae18-c51f899f6d3f VM-1
```

Note To obtain a list of all images and their UUIDs, type `nova image-list`.

The following example shows how to create a unicast VXLAN type network:

```
$ neutron cisco-policy-profile-update polprofId --add-tenant 1234-1234-1234-1234
$ neutron cisco-network-profile-create netprof overlay --subtype enhanced --segment_range
5000-5300
$ neutron net-create NetworkOne --n1kv:profile_id b9b8d5fa-41a3-4e59-bb1e-6a5e296908e1
$ neutron port-create NetworkOne --n1kv:profile_id a9355268-5aed-8030-f3ab-e367ef4c9acc
$ nova boot --image imageid --flavor flavor-id --nic
port-id=d341926c-21ca-48cd-ae18-c51f899f6d3f VM-1
```

CHAPTER 5

Configuring Layer 2 Features on Virtual Networks

This chapter contains the following sections:

Configuring Private VLANs

Information About Private VLANs

Private VLANs are implemented in OpenStack by configuring provider network and policy profiles. The primary VLAN should be specified by the provider network profile, while the secondary VLAN should be specified in the policy profile. When the policy profile is selected for a VM interface, the interface is attached to the corresponding secondary VLAN.

Configuring a Private VLAN

The following section guides you through the private VLAN configuration process. After completing each procedure, return to this section to make sure that you have completed all required procedures in the correct sequence.

Procedure

Step 1 Enable or disable the PVLAN feature globally. For more information, see the *Cisco Nexus 1000V for KVM Layer 2 Configuration Guide*.

Step 2 Configure one or more VLANs as primary VLAN(s) on the VSM. For more information, see the *Cisco Nexus 1000V for KVM Layer 2 Configuration Guide*.

Step 3 Configure a VLAN as a secondary VLAN on the VSM. For more information, see the *Cisco Nexus 1000V for KVM Layer 2 Configuration Guide*.

Step 4 Associate secondary VLANs to a PVLAN. For more information, see the *Cisco Nexus 1000V for KVM Layer 2 Configuration Guide*.

Step 5 Configure PVLAN port profiles for each secondary VLAN on the VSM. For more information, see the *Cisco Nexus 1000V for KVM Port Profile Configuration Guide* .

Step 6 Create a network segment in OpenStack for each PVLAN. For more information, see Creating a Primary VLAN in OpenStack, on page 30.

Guidelines and Limitations

Private VLANs in OpenStack have the following configuration guidelines and limitations:

- Restrict the policy profile scope to tenants unless the usage of policy profiles can be validated by the orchestration system.
- When a PVLAN policy profile is selected, the association between the secondary VLAN in the policy profile to the primary VLAN in the network is not validated.
- Do not publish policy profiles without secondary VLAN configuration on regular tenants in a PVLAN environment unless it can be validated by the orchestration system. Using policy profiles without secondary VLANS on primary VLAN segments result in promiscuous access to the VLAN.

Creating a Primary VLAN in OpenStack

Command	Purpose
For Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) and higher: **neutron cisco-net-create** *name* **--provider:network_type vlan --provider:physical_network** *network* **--provider:***segmentation_id vlan_id* For all other releases: **neutron cisco-network-profile-create** *name* **vlan --segment_range***private-VLAN* **--physical_network** *network*	Creates a VLAN network profile. For more information about this command, see the cisco-network-profile-create, on page 48 command reference page. You can also create the network profile using the OpenStack dashboard. For more information, see Creating a Network Profile Using the OpenStack Dashboard, on page 21.

Command	Purpose
neutron net-create *name* **--n1kv:profile_id** *networkProfileId***--shared** **Note** Starting from Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2), **--n1kv:profile_id** has been replaced with **--n1kv:profile**.	Creates a network with a primary VLAN as the network ID and makes the network available to all tenants. For more information about this command, see the net-create, on page 55 command reference page. You can also create the network using the OpenStack dashboard. For more information, see Creating a Network Using the OpenStack Dashboard.
neutron subnet-create *network-name IP-address-range* **--name** *subnet-name*	Attaches a subnet to the network. For more information about this command, see the OpenStack documentation. You can also create a subnet using the OpenStack dashboard. For more information, see Creating a Subnet for a Network Using the OpenStack Dashboard, on page 23.

Before You Begin

Create a primary VLAN on the VSM. For more information, see the *Cisco Nexus 1000V for KVM Layer 2 Configuration Guide*.

The following example shows how to create a Primary VLAN network of VLAN 100 with subnet 10.10.10.0/24:

```
$
$ neutron cisco-network-profile-create primary100pool vlan --segment_range 100-100
--physical_network physnet1
$ neutron net-create primary100 --n1kv:profile_id a9355268-5aed-8030-f3ab-e367ef4c9acc
--shared
$ neutron subnet-create primary100 10.10.10.0/24 --name subnet1
$
```

Note The profile_id in the neutron net-create command refers to the network profile ID. The profile_id in the neutron port-create command refers to the policy profile ID.

Associating a Feature Profile of a Secondary VLAN to a Tenant

You can limit the scope of a feature profile to selected tenants by setting the restrict_policy_profiles variable in the cisco_plugins.ini file. For more information on how to set this variable in OpenStack, see the OpenStack documentation.

Tenants can access the secondary VLANs that are associated with the other tenants. Hence, in a private VLAN environment, it is recommended to associate a feature profile to selected tenants unless the orchestration system can perform the validation of policy-profile usage by a tenant.

Before You Begin

- Create a feature profile for the secondary VLAN on the VSM. For information on how to create a feature profile, see the Configuring a Port Profile as a Private VLAN section in the *Cisco Nexus 1000V for KVM Port Profile Configuration Guide, Release 5.x*.
- Confirm that the feature profile is available in the cisco_plugins.ini file. For more information on the cisco_plugins.ini file, see the Configuring Additional Parameters in the cisco_plugin.ini File section in the *Cisco Nexus 1000V for KVM Installation Guide, Release 5.2(1)SK3(2.1)*.

Procedure

	Command or Action	Purpose
Step 1	**neutron cisco-policy-profile-update** *feature_profile_name* **--add-tenant** *tenant-id*	Associates the feature profile to the tenant. For more information on this command or to disassociate the feature profile from a tenant, see the cisco-policy-profile-update, on page 54 command reference page. **Note** This command is not available in Cisco Nexus 1000V Release 5.2(1)SK3(2.2).

The following example shows how to associate the secondary101 feature profile to the tenant ID 8d53387cb36e4475813b09bd53beaa00:

```
$
$ neutron cisco-policy-profile-update secondary101 --add-tenant
8d53387cb36e4475813b09bd53beaa00
$
```

Note The same feature profile can be associated with multiple tenants by issuing the same command for each tenant. This is useful if the secondary VLAN is an isolated VLAN.

Feature History for Private VLAN

Feature Name	Release	Feature Information
Private VLANs	5.2(1)SK3(2.1)	This feature was introduced.

Configuring VLAN Trunk vEthernet Ports

The following section guides you through the VLAN trunk configuration process for vEthernet ports. After completing each procedure, return to this section to make sure that you have completed all required procedures in the correct sequence.

Procedure

Step 1 Create a vEthernet trunk port profile. See the "Configuring a Trunk Policy Profile for a vEthernet Port" section in the *Cisco Nexus 1000V Port Profile Configuration Guide* .

Step 2 Create a VLAN network in OpenStack. See Configuring Virtual Networks Using OpenStack, on page 19.

Step 3 Configure a VM or Cisco Cloud Services Router (CSR) interface and specify the trunk port profile and network you configured in the previous steps. Note that the native VLAN of the trunk port will be set to the segment ID of the VLAN network that was created in Step 2.
After the VM or CSR port appears in the VSM and VEM, the port is identified as a trunk port and can carry traffic for all the tenant VLANs.

Note

- The native VLAN of the trunk port will be set to the segment ID of the VLAN network created in Step 2.
- Trunk native VLAN configuration in policy profile is not supported. If configured, the effective native VLAN for a vEthernet trunk port is set to the segment ID of the VLAN network created in Step 2.

CHAPTER 6

Configuring L3 Forwarding

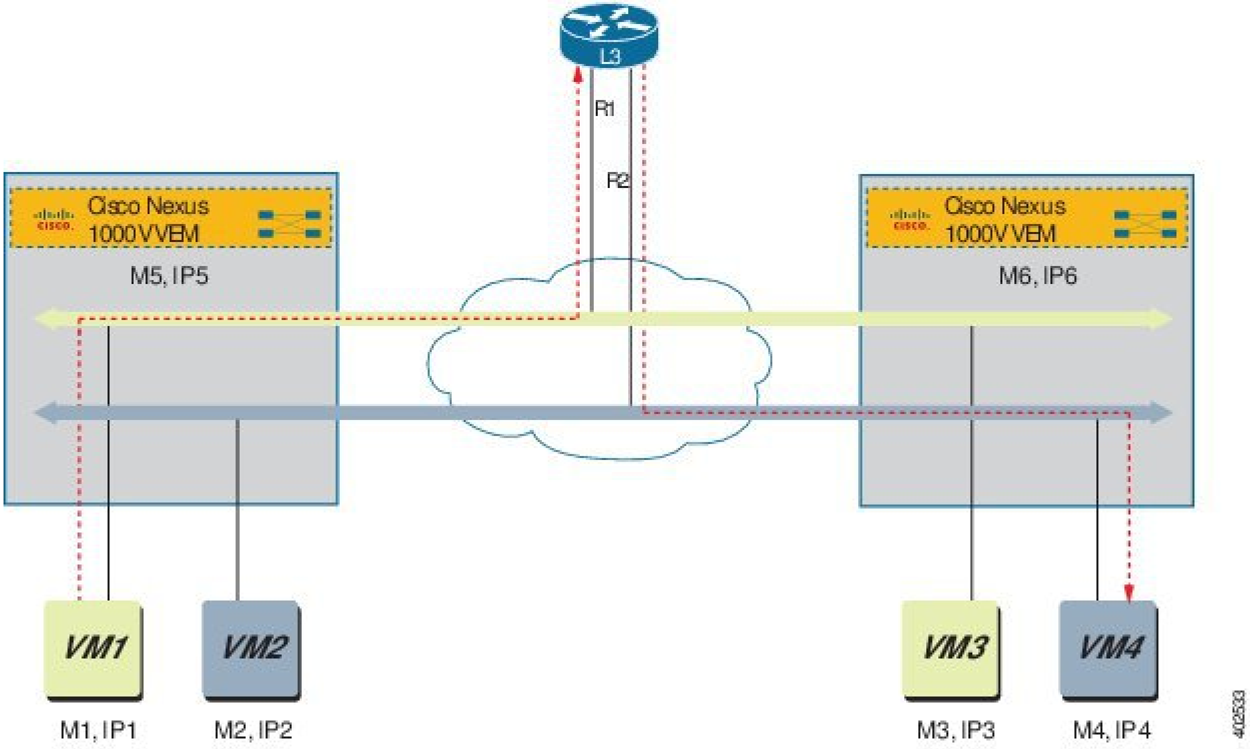
This chapter contains the following sections:

Layer 3 Forwarding Overview

Note Layer 3 Forwarding requires a Cisco Nexus 1000V Advanced Edition license.

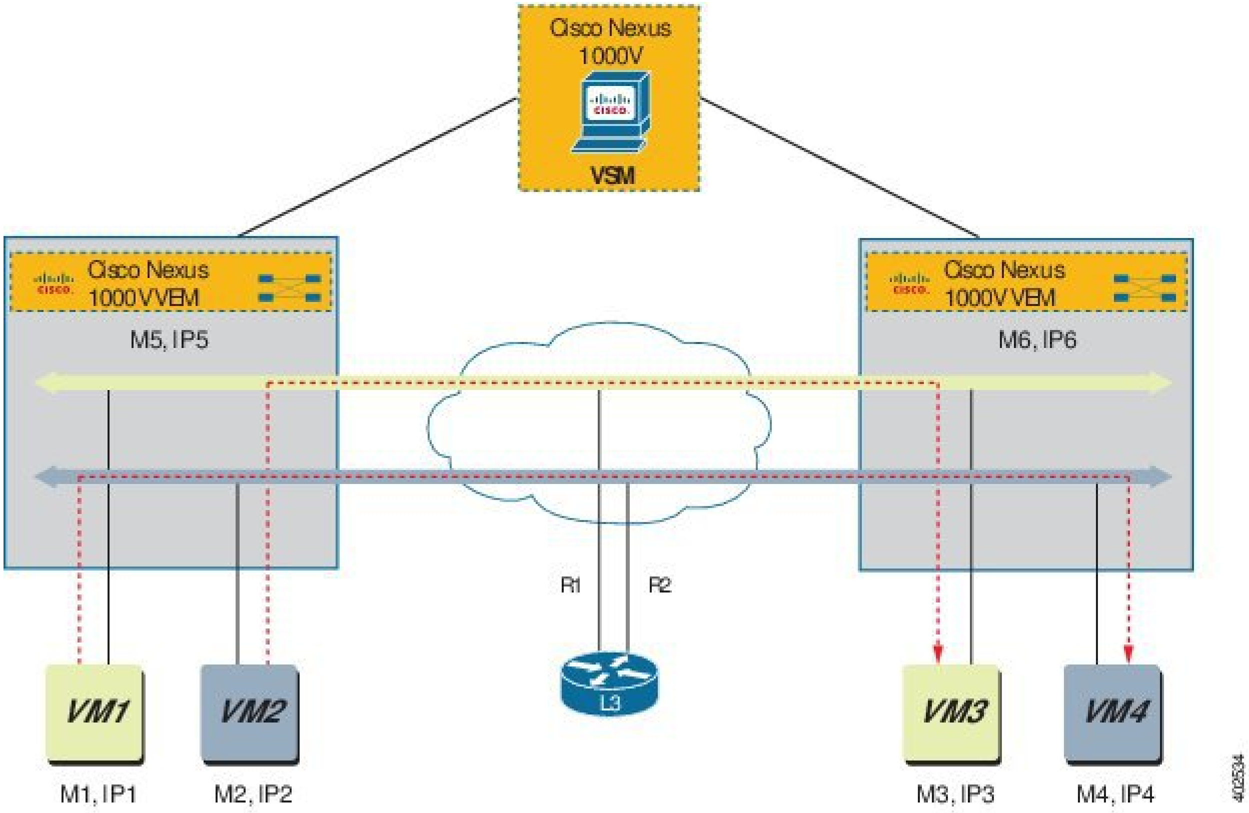
In a typical, centralized Layer 3 forwarding model, a Layer 3 router (virtual and physical) receives packets from a Cisco Nexus 1000V and forwards the traffic across the segments. In this model, the Layer 3 router can become a point of congestion or blockage for the flow of traffic. For example, in the following figure, data packets from VM1 are routed to the Layer 3 router. The Layer 3 router decides where the data packets need to go and forwards the packets to VM4.

Figure 1: Centralized Layer 3 Forwarding Model



In a distributed forwarding model, the VSM manages all the configurations and the VEMs are instantiated on each host to provide packet switching functionality. In this model, the VSM shares the VM packet routing information with the VEMs, so that the VEMs can forward the packets to the correct host. Distributed forwarding reduces the traffic that is sent to the Layer 3 router because the VEMs send the packets directly to the destination VM. For example, in the following figure, the VEM is aware of VM1 and VM2 routing information. The VEM automatically directs the traffic from VM1 to VM4 and VM2 to VM3. There is no longer a need to forward the packet information to the Layer 3 router.

Figure 2: Distributed Layer 3 Forwarding Model



Enabling and Verifying Layer 3 Forwarding

Before You Begin

Log in to the CLI in EXEC mode.

Note Layer 3 Forwarding requires a Cisco Nexus 1000V Advanced Edition license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature l3forwarding**	Enables the Layer 3 forwarding feature.
Step 3	switch(config)# **show feature**	(Optional) Displays the enabled status for Cisco Nexus 1000V features.

This example shows how to enable the Layer 3 forwarding feature and display the output:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# feature l3forwarding
switch(config)# show feature
Feature Name          Instance  State
--------------------  --------  --------
cts                   1         disabled
dhcp-snooping         1         disabled
http-server           1         enabled
lacp                  1         disabled
netflow               1         disabled
network-segmentation  1         enabled
port-profile-roles    1         disabled
private-vlan          1         disabled
segmentation          1         enabled
sshServer             1         enabled
tacacs                1         disabled
telnetServer          1         disabled
vtracker              1         disabled
vxlan-gateway         1         disabled
l3forwarding          1         enabled
switch(config)#
```

Viewing Layer 3 Forwarding Information

Use the following commands to view Layer 3 forwarding information:

Note Make sure that you are logged into the VEM when issuing **vemcmd** commands.

Command	Purpose
vemcmd show ip-forwarding-table	Displays the complete IP forwarding table.
vemcmd show l3-forwarding-table *l3-table-id*	Displays the Layer 3 forwarding table. If a Layer 3 table ID is not specified, then the complete Layer 3 forwarding table is displayed.
vemcmd show l2 segment *segment-id*	Displays the router-mac for that segment.

Command	Purpose
show segment statistics module [**vlan** | **bridge-domain-name**] *number*	Displays segment statistics for the specified VLAN or bridge domain
show l3-segment-attribute-table [**vlan** | **bridge-domain-name**] *number*	Displays the Layer 3 segment attribute table for the specified VLAN or bridge domain.
show interface counters	Displays related interface counter information.

This example shows how to display information about Layer 3 forwarding:

```
switch# show ip-forwarding-table
Flags:(Rtr)=Router MAC; (L)=Local; (R)=Remote;
VLAN/SEGID|L3 TableID| MAC | IP | Flags
----------+----------+--------------------+---------------+----------
1172 9 FA:16:3E:49:88:D6 192.168.72.65 L,Rtr
1170 9 FA:16:3E:2D:87:5B 192.168.70.101 L
1170 9 FA:16:3E:42:8C:AF 192.168.70.50 L,Rtr
1170 9 FA:16:3E:E4:8D:8A 192.168.70.104 L
1171 9 FA:16:3E:1A:06:0A 192.168.71.2 L


VEM# vemcmd show l3-forwarding-table 1
 L3-table-id           IP address          mac address     BD
           1        192.168.1.150    bc:16:65:22:ac:42    130
           1         192.168.1.48    bc:16:65:22:ac:42    130
           1        192.168.1.179    bc:16:65:22:ac:42    130
           1         192.168.1.92    bc:16:65:22:ac:42    130

VEM# vemcmd show flow-mgr l3-flows
Flow-id L3-table-id           IP address          mac address     BD
--------------------------------------------------------------------------------------
  0          5000         10.10.163.20    00:16:3e:a9:03:c8    163
  1          5000         10.10.163.64    00:16:3e:20:a9:b4    163
  2          5000         10.10.162.63    00:16:3e:20:a9:a3    162
  3          5000         10.10.162.10    00:1b:35:ab:45:0e    162

VEM# vemcmd show l2 segment 50001
Bridge domain   11 brtmax 4096, brtcnt 3, timeout 300
Segment ID 50001, swbd 4096, "bd1"
Flags:  P - PVLAN  S - Secure  D - Drop  R - Router-mac

       Type         MAC Address   LTL   timeout   Flags    PVLAN Remote IP    DSN    Slot
     Static   52:54:00:98:b4:ff    65         0                                    0.0.0.0
          0
     Static   52:54:00:62:12:3a    63         0                                    0.0.0.0
          0
     Static   52:54:00:61:13:bd     0         0                 R                  0.0.0.0
     0

switch# show segment statistics module 3
     VLAN/    Rx    Rx    Tx    Tx Missed Missed Dropped Dropped
        BD  Pkts Bytes  Pkts Bytes  Pkts Bytes  Pkts Bytes
         1    0     0     0     0     0     0     0     0
      3972    0     0     0     0     0     0     0     0
      3970    0     0     0     0     0     0     0     0
      3968    0     0     0     0     0     0     0     0
      3971    0     0     0     0     0     0     0     0


switch# show l3-segment-attribute-table
-------------------------------------------------------
Segment-id    Segment-type     Attribute      Value
-------------------------------------------------------
```

```
111127  Vxlan  Router IP            45.11.9.1
111127  Vxlan Router MAC    FA:16:3E:8B:59:05
111127  Vxlan     SUBNET              0.0.0.0/0
111126  Vxlan  Router IP            45.11.8.1
111126  Vxlan Router MAC    FA:16:3E:CD:ED:A1
111126  Vxlan     SUBNET              0.0.0.0/0
111125  Vxlan  Router IP            45.11.7.1
111125  Vxlan Router MAC    FA:16:3E:B9:A7:D2
111125  Vxlan     SUBNET              0.0.0.0/0
111124  Vxlan  Router IP            45.11.6.1
111124  Vxlan Router MAC    FA:16:3E:4D:D9:20
111124  Vxlan     SUBNET              0.0.0.0/0
111123  Vxlan  Router IP            45.11.5.1
111123  Vxlan Router MAC    FA:16:3E:8F:3E:48
```

```
switch# show interface counters

--------------------------------------------------------------------------------
Port                                 InOctets                      InUcastPkts
--------------------------------------------------------------------------------
mgmt0                               846142352                          1456395
Eth3/1                              234693677                            48980
Eth4/1                               14229614                             4606
Eth5/1                              198530588                            21751
Eth5/2                              201360061                            35320
Eth6/1                              276841979                             3298
Eth7/1                               72027394                              153
Eth7/2                               74577517                            22113
Po1                                 276808574                             3298
Po2                                 399811656                            57064
Po3                                 146577970                            22259
Veth1                                  987879                             3671
Veth2                                  343513                             2618
```

Monitoring Layer 3 Forwarding Statistics

Use the following commands to view Layer 3 forwarding statistics:

Note Make sure that you are logged into the VEM when issuing **vemcmd** commands.

Command	Purpose
vemcmd show stats	Displays general Layer 3 forwarding port statistics.
vemcmd show packets	Displays Layer 3 forwarded packets.
vemcmd show bd stats [**vlan** | **segment** | **bridge-domain-name**] *number*	Displays Layer 3 forwarded packets per BD.
vemcmd clear bd stats [**vlan** | **segment** | **bridge-domain-name**] *number*	Clears the bridge domain statistics for the specified VLAN, segment, or bridge domain.
show l3-segment-attribute-table [**vlan** | **bridge-domain-name**] *number*	Displays the Layer 3 segment attribute table for the specified VLAN or bridge domain.

Command	Purpose
show interface counters	Displays related interface counter information.

This example shows how to view Layer 3 forwarding statistics :

```
VEM# vemcmd show stats
LTL  Received        Bytes      Sent        Bytes     RxL3frwd         Bytes     TxL3frwd
    Bytes Txflood    Rxdrop    Txdrop  Name
52   525             50666       483        47182     121              7096       120
    7032  4          0         0       vnet0
53   520             50352       478        46844     119              7085       119
    7085  0          0         0       vnet2

VEM# vemcmd show packets
LTL   RxUcast   TxUcast   RxMcast   TxMcast   RxBcast   TxBcast   RxL3frwd    TxL3frwd
Txflood    Rxdrop    Txdrop    RxJumbo     TxJumbo    Name
52    2026      2000      16        16        18         0        121         120
16         0         0         0           0          vnet0
53    2026      2000      0         0         16         0        119         119
0          0         0         0           0          vnet2

VEM# vemcmd show bd stats vlan 107
BD  L3Rx        Bytes      L3Tx        Bytes     L3Rxmiss        bytes
6   97          6456       95          6359      0               0

L3RxMiss - Miss in the L3 hash table for /32 addresses.
```

Layer 3 Forwarding Guidelines and Limitations

Layer 3 forwarding has the following configuration guidelines and limitations:

- Layer 3 forwarding must be enabled before system host setup or the information in the forwarding tables will be inconsistent. To enable Layer 3 forwarding on active VSMs, you must reload the VSM.
- Layer 3 forwarding is not supported for packets with VXLAN encapsulation received from VMs behind a VEM, such as a VXLAN gateway.
- Same segment Layer 3 forwarding is supported, but ICMP redirect messages are not generated.
- In Anycast forwarding (non-gateway forwarding) mode, external traffic is forwarded using the gateway. Also, packets with a router_mac destination are dropped if there is no matching entry in the Layer 3 forwarding table. An ICMP unreachable message will not be generated.
- There can be only one gateway per segment.
- In Openstack mode, there can be only one subnet in a network. Multiple subnets in one network is not supported.
- QoS and security policies applied to packets on the Layer 3 router are skipped in the distributed Layer 3 forwarding model.
- Destination interface MTU validation is not done in VEM forwarding. There have been no traffic issues observed in testing between VMs on the same VEM.
- There can only be one router per tenant.
- VTEP IP address changes may result in transient packet loss for a brief period.
- The network cannot be changed from shared to non-shared and vice-versa.

- A MAC cannot be associated with multiple IP addresses.
- The following are not supported:
    - PVLAN with Layer 3 forwarding.
    - Localization of Layer 3 forwarding entries in VLAN deployments.
    - IPv6 Layer 3 forwarding.
    - Multicast Layer 3 forwarding.
    - Layer 3 forwarding to and from shared segments.

Feature History for Layer 3 Forwarding

Feature Name	Release	Feature Information
Layer 3 Forwarding	5.2(1)SK3(2.2)	This feature was introduced.

APPENDIX A

OpenStack Command Reference

This chapter contains the following sections:

Additions to the Neutron Command-Line Interface

The Neutron command-line interface now accepts a Cisco Nexus 1000V-related attribute extension for the core Neutron resources. Additionally, new commands have been introduced for the Cisco Nexus 1000V Neutron plug-in's extended resources.

This CLI reference document describes the newly added attribute extension and commands and contains examples to demonstrate how they are used. For a complete list and description of network-related commands and arguments, see http://docs.openstack.org/api/openstack-network/2.0/content/.

Attribute Extension for Core Neutron Resources

The network and port objects have been extended to include the **n1kv:profile_id** attribute extension to enable network and port association with Cisco Nexus 1000V profiles. Use the profile_id extension at network creation to associate a network with a Cisco Nexus 1000V network profile and, at port creation, to associate a port with a Cisco Nexus 1000V policy profile.

Note For Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) and higher:

- The **n1kv:profile_id** attribute extension has been replaced with **n1kv:profile**.
- Only the port object has been extended.
- The network create extension is not required.

Commands and Options for Extended Neutron Resources

Commands have been added to enable extended Neutron resources; these resources include the network profile, policy profile, profile binding, and credentials.

Note The Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher does not support the network profile and credential commands.

Network Profile Commands

Network profile commands enable you to create, update, list, delete, and show Cisco Nexus 1000V network profile details.

Policy Profile Commands

Policy profile commands enable you to list and show details of your Cisco Nexus 1000V policy profile.

Profile Binding Options

Profile binding options enable you to associate or disassociate Cisco Nexus 1000V policy and network profiles with tenants.

Credential Commands

Credential commands enable you to create, update, delete, and show details of your Cisco Nexus 1000V credentials.

cisco-credential-create

To create a Cisco Nexus 1000V credential, use the **neutron cisco-credential-create** command.

neutron cisco-credential-create [--help] *credential-name credential-type* [**--request-format** {*format*}] [**--tenant-id** *tenant-id*] [**--user_name** *username*] [**--password** *password*]

Syntax Description

--help	(Optional) Specifies the help message
credential-name	IP address of the credential. The name is a string with up to 255 characters. Characters can be numbers, upper and lowercase letters, and special characters. The IP address must be in the a.b.c.d format.
credential-type	Type of credential. The credential for the Nexus 1000V is n1kv.
--tenant-id *tenant-id*	(Optional) Specifies the owner's tenant ID. This is a UUID. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.
--user_name *username*	(Optional) Specifies the username of the credential. The username is a string with up to 255 characters. Characters can be numbers, upper and lowercase letters, and special characters.
--request-format *format*	(Optional) Specifies the format of the request. Accepted values are **json** or **xml.**
--password *password*	(Optional) Specifies the password for the credential.

Command Default

None

Command History

Release	Modification
OpenStack Juno	This command has be deprecated and is not supported by Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher.
OpenStack Havana	This command was introduced.

Example

This example shows how to create a credential:

```
$ neutron cisco-credential-create 172.23.181.101 N1KV --user_name admin --password mypwd
```

cisco-credential-delete

To delete a credential, use the **neutron cisco-credential-delete** command.

neutron cisco-credential-delete *credential-id*

Syntax Description

credential-id	ID of the credential to be deleted. This is a UUID. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.

Command Default

None

Command History

Release	Modification
OpenStack Juno	This command has be deprecated and is not supported by Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher.
OpenStack Havana	This command was introduced.

Example

This example shows how to delete a credential:

```
$ neutron cisco-credential-delete 9fff279d-2f3f-4a9c-b0fe-3a0ae91075c5
```

cisco-credential-list

To list all available credentials, use the **neutron cisco-credential-list** command.

neutron cisco-credential-list

This command has no arguments or keywords.

Command Default

None

Command History

Release	Modification
OpenStack Juno	This command has be deprecated and is not supported by Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher.
OpenStack Havana	This command was introduced.

Example

This example shows how to list credentials:

```
$ neutron cisco-credential-list
```

cisco-credential-show

To display the details for the credentials associated with a credential ID, use the **neutron cisco-credential-show** command.

neutron cisco-credential-show *credential-id*

Syntax Description

credential-id	ID of the credential. This is a UUID. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.

Command Default

None

Command History

Release	Modification
OpenStack Juno	This command has be deprecated and is not supported by Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher.
OpenStack Havana	This command was introduced.

Example

This example shows how to display details about the credential:

```
$  neutron cisco-credential-show 9fff279d-2f3f-4a9c-b0fe-3a0ae91075c5
```

cisco-credential-update

To update a credential, use the **neutron cisco-credential-update** command.

neutron cisco-credential-update *credential-id* [**--user_name** *username*] [**--password** *password*]

Syntax Description

credential-id	ID of the credential. This is a UUID. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.
--user_name *username*	(Optional) Specifies the username of the credential. The username is a string with up to 255 characters. Characters can be numbers, upper and lowercase letters, and special characters.
--password *password*	(Optional) Specifies the password for the credential.

Command Default

None

Command History

Release	Modification
OpenStack Juno	This command has be deprecated and is not supported by Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher.
OpenStack Havana	This command was introduced.

Example

This example shows how to update a username and password:

```
$ neutron cisco-credential-update  9fff279d-2f3f-4a9c-b0fe-3a0ae91075c5 --user_name admin
--password mypwd
```

cisco-network-profile-create

To create a Cisco Nexus 1000V network profile, use the **neutron cisco-network-profile-create** command.

neutron cisco-network-profile-create [--sub_type{*type*}**] [--segment_range** *segment-range*] [**--physical_network** *network*] [**--multicast_ip_range** *ip-range*] **[(--add-tenant** *tenant-id*)**...]** **netprofileName***name* {*type*}

Syntax Description

--sub_type *type*	(For Overlay and Trunk only.)Specifies the subtype for a specific type of network profile. The subtype is **native_vxlan** or **enhanced** for an overlay type of network profile and **vlan** for trunk type of network profile.
--segment_range *segment-range*	Specifies the range of the segment for vlan and vxlan types. The range is entered in a lowest to highest hyphen-separated format. The range of valid values for vlan types is 1 to 4095. The range of valid values for vxlan types is 4095 to 16000000.
--physical_network *network*	(For VLAN, only.) Specifies the name of the Layer 2 domain. The name is a string with up to 255 characters. Characters can be numbers, upper and lowercase letters, and special characters.
--multicast_ip_range *ip-range*	Specifies the range of the IP address. This is only applicable for the native_vxlan sub_type. The range is entered in a lowest to highest hyphen-separated format. The range of valid values is from 224.0.1.0 to 239.255.255.255. The range 224.0.0.0 to 224.0.0.255 is reserved on the VSM.
--add-tenant *tenant-id*	Associates a tenant with the network profile. This is a UUID. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN. Can be repeated any number of times to add multiple tenants. When you add a new list of tenants using this keyword, the new list of tenants overwrites the existing list of tenants.
netprofName *name*	Name of the network profile.

{*type*}	Specifies the type of network profile. The type can be one of the following: **vlan**, **overlay**, or **trunk**.

Command Default

None

Command History

Release	Modification
OpenStack Juno	This command has be deprecated and is not supported by Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher.
OpenStack Havana	This command was introduced.

Example

This example shows how to create a Cisco Nexus 1000V network profile:

```
$ neutron cisco-network-profile-create netprof vlan --segment_range 100-200 --physical_network
 physnet1
```

cisco-network-profile-delete

To delete a Cisco Nexus 1000V network profile, use the **neutron cisco-network-profile-delete** command.

neutron cisco-network-profile-delete *network-profile*

Syntax Description

network-profile	ID or name of the network profile. The name is a string with up to 255 characters. Characters can be numbers, upper and lowercase letters, and special characters The ID is a UUID. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.

Command Default

None

Command History

Release	Modification
OpenStack Juno	This command has be deprecated and is not supported by Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher.
OpenStack Havana	This command was introduced.

Example

This example shows how to delete a Cisco Nexus 1000V network profile:

```
$ neutron cisco-network-profile-delete netProf
```

cisco-network-profile-list

To list Cisco Nexus 1000V network profiles, use the **neutron cisco-network-profile-list** command.

neutron cisco-network-profile-list

This command has no arguments or keywords.

Command Default

None

Command History

Release	Modification
OpenStack Juno	This command has be deprecated and is not supported by Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher.
OpenStack Havana	This command was introduced.

Example

This example shows how to list Cisco Nexus 1000V network profiles:

```
$ neutron cisco-network-profile-list
```

cisco-network-profile-show

To show Cisco Nexus 1000V network profile details, use the **neutron cisco-network-profile-show** command.

neutron cisco-network-profile-show *network-profile-id*

Syntax Description

network-profile-id	ID or name of the network profile. The network profile ID is a UUID. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN. The network profile name is a string with up to 255 characters. Characters can be numbers, upper and lowercase letters, and special characters.

Command Default

None

Command History

Release	Modification
OpenStack Juno	This command has be deprecated and is not supported by Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher.
OpenStack Havana	This command was introduced.

Example

This example shows how to view Cisco Nexus 1000V network profile details:

```
$ neutron cisco-network-profile-show netProfId
```

cisco-network-profile-update

To update a Cisco Nexus 1000V network profile information, use the **neutron cisco-network-profile-update** command.

neutron cisco-network-profile-update *network-profile-name* **[--request-format** *format***] [--add-tenant | --remove-tenant]** *tenant-id*

Syntax Description

network-profile-name	UUID or name of the network profile to update. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN. The name is a string with up to 255 characters. Characters can be numbers, upper and lowercase letters, and special characters.
--request-format *format*	(Optional) Specifies the format of the request. Accepted values are: **json** or **xml**.
--add-tenant	(Optional) Associates a tenant with a network profile. Can be repeated any number of times to add multiple tenants. When you add a new list of tenants using this keyword, the new list of tenants overwrites the existing list of tenants.
--remove-tenant	(Optional) Disassociates a tenant from the network.
tenant-id	ID of the tenant being added or removed. This is a UUID. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.

Command History

Release	Modification
OpenStack Juno	This command has be deprecated and is not supported by Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) software and higher.
OpenStack Havana	This command was introduced.

Usage Guidelines

None

Example

This example shows how to associate a tenant with a network profile:

```
$   neutron cisco-network-profile-update mynetprofile VLAN --add-tenant 1234-1234-1234-1234
```

cisco-policy-profile-list

To list available Cisco Nexus 1000V policy profiles, use the **neutron cisco-policy-profile-list** command.

neutron cisco-policy-profile-list

This command has no arguments or keywords.

Command Default

None

Command History

Release	Modification
OpenStack Havana	This command was introduced.

Example

This example shows how to list available Cisco Nexus 1000V policy profiles:

```
  $ neutron policy-profile-list
```

cisco-policy-profile-show

To show Cisco Nexus 1000V policy profile details, use the **neutron cisco-policy-profile-show** command.

neutron cisco-policy-profile-show *policy-profile-id*

Syntax Description

policy-profile-id	UUID of the policy profile. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.

Command Default

None

Command History

Release	Modification
OpenStack Havana	This command was introduced.

Example

This example shows how to view Cisco Nexus 1000V policy profile details:

```
$ neutron cisco-policy-profile-show b9b8d5fa-41a3-4e59-bb1e-6a5e296908e1
```

cisco-policy-profile-update

To update a Cisco Nexus 1000V policy profile and associate or disassociate tenants, use the **neutron cisco-policy-profile-update** command.

neutron cisco-policy-profile-update *policy-profile-id* **[--request-format** *format***] [--add-tenant | --remove-tenant]** *tenant-id*

Syntax Description

policy-profile-id	ID of the policy profile being updated. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.This is a UUID or name of the policy profile published in the OpenStack plugin from the VSM.
--request-format *format*	(Optional) Specifies the format of the request. Accepted values are: **json** or **xml**.
--add-tenant	(Optional) Associates a tenant with a policy profile. Can be repeated any number of times to add multiple tenants.
--remove-tenant	(Optional) Disassociates a tenant from the network.
tenant-id	ID of the tenant being added or removed. This is a UUID. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.

Command Default

None

Command History

Release	Modification
OpenStack Havana	These arguments were introduced.

Example

This example shows how to update a policy profile and associate a tenant:

```
$neutron cisco-policy-profile-update polprofId --add-tenant 1234-1234-1234-1234
```

net-create

To create a network and associate it with a Cisco Nexus 1000V network profile, use the **neutron net-create** command.

Note For Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2) and higher, **--n1kv:profile_id** is replaced with **--n1kv:profile**

neutron net-create *name* **--n1kv:profile_id** *profileId*

Syntax Description

name	Name of the network. The name is a string with up to 255 characters. Characters can be numbers, upper and lowercase letters, and special characters.
--n1kv:profile_id	Associates a network with a Cisco Nexus 1000V network profile.
profileId	UUID of the network profile. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.

Command Default

None

Command History

Release	Modification
OpenStack Havana	This attribute extension was introduced.

Example

This example shows how to create a network and associate the network with a Cisco Nexus 1000V network profile:

```
$ neutron net-create NetworkOne --n1kv:profile_id b9b8d5fa-41a3-4e59-bb1e-6a5e296908e1
$ neutron subnet-create NetworkOne 172.23.181.0/24 --name SubnetOne
```

port-create

To create a port and associate it with a Cisco Nexus 1000V policy profile, use the **neutron port-create** command.

neutron port-create *name* **--n1kv:profile_id** *profile-id*

Syntax Description

name	Name of the network. The name is a string with up to 255 characters. Characters can be numbers, upper and lowercase letters, and special characters.
--n1kv:profile_id	Associates a network with a Cisco Nexus 1000V policy profile.
profile-id	UUID of the policy profile. The value is 36 hexadecimal digits and hyphens in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN.

Command Default

None

Command History

Release	Modification
OpenStack Havana	This attribute extension was introduced.

Example

This example shows how to create a port and associate it with a Cisco Nexus 1000V policy profile:

```
$ neutron port-create NetworkOne --n1kv:profile_id b9b8d5fa-41a3-4e59-bb1e-6a5e296908e1
```

Related Cisco Nexus 1000V Configuration Options

cisco_n1k Configuration Options

The following configuration options appear in the cisco_n1k section in the cisco_plugins.ini file located at `/etc/neutron/plugin.ini`.

Configuration Option	Description
integration_bridge = br-int	Specify the name of the integration bridge to which the VIFs are attached.
default_policy_profile = For example, **default_policy_profile = service_profile**	The name of the policy profile that needs to be associated with a port, when a policy profile is not specified during port creation.
network_node_policy_profile = For example, **network_node_policy_profile = dhcp_pp**	The name of the policy profile that needs to be associated with a port owned by the network node (dhcp, router).
default_network_profile = For example, **default_network_profile = network_pool**	The name of the network profile to be associated with a network, when a network profile is not specified during network creation. The administrator must pre-create a network profile with this name.
poll_duration = 60	The time (in seconds) for which the plug-in polls the VSM for updates in the policy profiles. The default value is 60.
restrict_policy_profiles = For example, **restrict_policy_profiles = False**	Specifies if tenants are restricted from accessing all the policy profiles. The default value is False, indicating that all tenants can access all the policy profiles.
http_pool_size = 4	The number of threads that needs to be used to make HTTP requests to the VSM.
http_timeout = 30	The time (in seconds) for which the plug-in waits for the VSM to respond.
enable_sync_on_start = False	Specifies if the plug-in should attempt to synchronize with the VSM when neutron is started. The default value is False, indicating that no full sync will be performed when neutron is started.
enable_sync_on_error = False	Specifies if the plug-in should attempt to synchronize with the VSM when there is a connection failure to the VSM. The default value is False, indicating that no full sync will be performed when there is a connection failure to the VSM.
max_vsm_retries	Number of VSM request retries the Neutron plug-in attempts before timing out. The default value is 2.

Configuration Option	Description
sync_interval	Number of seconds between checks of state between the plugin and VSM. The default value is 300 seconds.

ml2_cisco_n1kv Configuration Options

The following configuration options appear in the ml2_cisco_n1kv section in ml2_conf_cisco.ini file located at `/etc/neutron/plugins/ml2/ml2_conf_cisco.ini`.

Configuration Option	Description
default_policy_profile	Name of the policy profile to be associated with a port when a port is created. The default value is **default-pp**. For example: `default_policy_profile = default-pp`
default_vlan_network_profile	Name of the VLAN network profile to be associated with a network. The default value is **default-vlan-np**. For example: `default_vlan_network_profile = default-vlan-np`
default_vxlan_network_profile	Name of the VXLAN network profile to be associated with a network. The default value is **default-vxlan-np**. For example: `default_vxlan_network_profile = default-vxlan-np`
poll_duration	Time in seconds for which the plugin polls the VSM for updates in policy profiles. The default value is 60 seconds. For example: `poll_duration = 60`
http_timeout	Timeout duration in seconds for the HTTP request. The default value is 15 seconds. For example: `http_timeout = 15`
restrict_policy_profiles	Specifies whether tenants are restricted from accessing all of the policy profiles. The default value is **false**, indicating that all tenants can access all policy profiles. For example: `restrict_policy_profiles = false`

Configuration Option	Description
n1kv_vsm_ips	Specifies the IP addresses in order for the plugin to connect to the VSM. You can enter multiple IP addresses separated by commas. For example: `n1kv_vsm_ips = 192.0.2.1, 192.0.2.2`
username	Specifies the username in order for the plugin to log into the VSM. For example: `username = user`
password	Specifies the password in order for the plugin to log into the VSM. For example: `password = secret`