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

First Published: August 01, 2014

Last Modified: May 13, 2016

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.

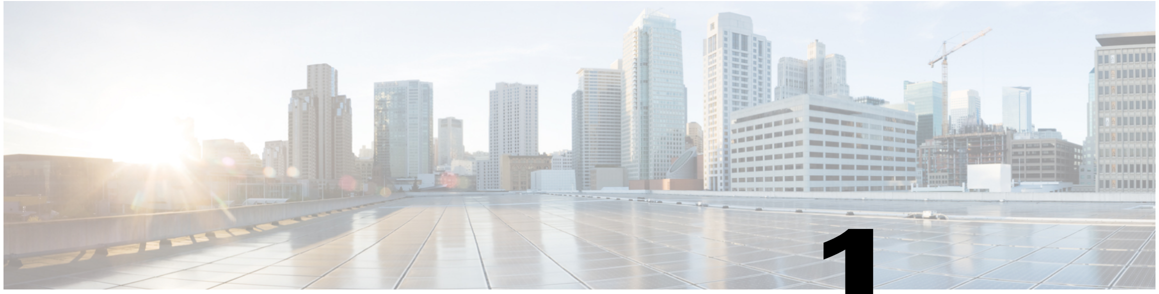
IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: http://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2014-2016 Cisco Systems, Inc. All rights reserved.

CONTENTS



CHAPTER 1

New and Changed Information

This chapter contains the following sections:

New and Changed Information

Table 1: New and Changed Features

Item	Description	Changed in Release	Where Documented
VXLAN Gateway	Starting with Release 5.2(1)SK3(1.1), Cisco Nexus 1000V for KVM does not support the VXLAN Gateway feature.	5.2(1)SK3(1.1)	—

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.

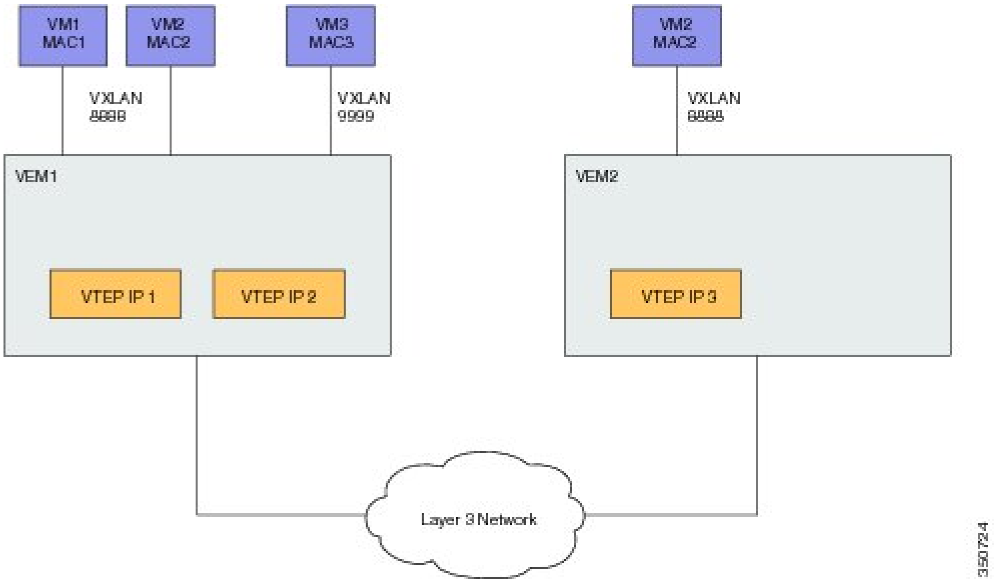
Note You must consistently use OpenStack for all VM network, subnet, and port configurations. If you create VM networks, subnets, and ports directly on the VSM, the configuration is lost when the OpenStack synchronization occurs.

Information About VXLANs

The Virtual Extensible LAN (VXLAN) technology enables you to create virtual domains by running a Layer 2 overlay network on top of Layer 3 with MAC-in-UDP encapsulation and a 24-bit VXLAN ID. The original Layer 2 frame from a Virtual Machine (VM) is encapsulated from within the Virtual Ethernet Module (VEM). Each VEM is assigned at least one IP address that is used as the source IP address when the encapsulated MAC frames are sent to other VEMs over the network.

The IP addresses, which are known as VXLAN Tunnel End Point (VTEP) IP addresses, are assigned to selected interfaces on the corresponding VEM. The encapsulation carries the VXLAN ID to scope the MAC address of the payload frame. The VM's VXLAN ID is indicated within the port profile configuration of the vNIC and is applied when the VM connects to the network.

Figure 1: VXLAN Overview



A VXLAN supports two different modes for flood traffic:

- Multicast mode—A VXLAN uses an IP multicast network to send broadcast, multicast, and unknown unicast flood frames. Each multicast mode VXLAN has an assigned multicast group IP address. When a new VM is deployed on a host in a multicast mode VXLAN, a VEM joins the assigned multicast group IP address by sending IGMP join messages. Flood traffic, broadcast, multicast and unknown unicast from the VM is encapsulated and is sent using the assigned multicast group IP address as the destination IP address. Packets sent to known unicast MAC addresses are encapsulated and sent directly to the destination server VTEP IP addresses.

  When an encapsulated packet is received from the network, it is decapsulated. The source MAC address of the inner frame and VXLAN ID are added to the Layer 2 table as the lookup key, and the source IP address of the encapsulation header is added as the remote IP address for the table entry.

- Unicast-only mode—The unicast-only mode is a mechanism to support traditional broadcast domain environment without the multicast mode requirement on the physical infrastructure. The packet path functions in the same way as multicast mode, except for the handling of flooded packets (for example, unknown unicast, multicast, and broadcast).

  A VXLAN uses each VEM's single unicast IP address as the destination IP address to send broadcast, multicast, and unknown unicast flood frames. The frames contain the designated VTEP of each VEM that has at least one VM in the corresponding VXLAN. When a new VM is deployed on the host in a unicast-mode VXLAN, a designated VTEP is selected for receiving flood traffic on that host. This designated VTEP is communicated to all other hosts through the Virtual Supervisor Module (VSM). Flood traffic (broadcast, multicast, and unknown unicast) is replicated on each VEM's designated VTEP in that VXLAN by encapsulating it with a VXLAN header. Packets are sent only to VEMs with a VM in that VXLAN. Packets that have a known unicast MAC address are encapsulated and sent directly to the destination server's VTEP IP address.

  You configure VXLANs as VM subnets using OpenStack.

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.

VXLAN Tunnel Endpoints

Each VEM requires at least one IP/MAC address pair to terminate VXLAN packets. This IP/MAC address pair is known as the VXLAN Tunnel End Point (VTEP) IP/MAC addresses. The VEM supports IPv4 addressing for this purpose. You can have a maximum of four VTEPs in a single VEM.

Note You must create the VTEP on the host by defining it in the Red Hat Enterprise Linux OpenStack Platform graphical user interface. For more information, see the *Cisco Nexus 1000V for KVM Installation Guide*.

One VTEP per VXLAN segment is designated to receive all broadcast, multicast, and unknown unicast flood traffic for the VEM.

When encapsulated traffic is destined to a VEM that is connected to a different subnet, the VEM does not use the Linux host routing table. Instead, it can use either Proxy Address Resolution Protocol (ARP) or a default gateway.

- To use Proxy ARP, you must configure the upstream router for Proxy ARP. The VTEPs initiate the ARP for remote VEM VTEP IP addresses. If the VTEPs in the different VEMs are in different subnets, the upstream router can respond using Proxy ARP.
- To use a default gateway, you must configure the VTEP with the **transport ip address external** command to specify the netmask and gateway IP address for the VTEP to use. For example, from the interface command mode, enter **transport ip address external netmask 255.255.255.0 gateway 1.2.3.4**.

Multicast Mode

When a VM attaches to a VEM, if it is the first to join the particular VXLAN segment on the VEM, an IGMP Join is issued for the VXLAN's assigned multicast group. When the VM transmits a packet on the network segment, a lookup is made in the L2 table using the destination MAC of the frame and the VXLAN identifier. If the result is a hit, the L2 table entry will contain the remote IP address to use to encapsulate the frame and the frame will be transmitted within an IP packet destined to the remote IP address. If the result is a miss (broadcast/multicast/unknown unicasts fall into this bucket), the frame is encapsulated with the destination IP address set to be the VXLAN segment's assigned IP multicast group.

When an encapsulated packet is received from the network, it is decapsulated and the source MAC address of the inner frame and VXLAN ID, is added to the L2 table as the lookup key and the source IP address of the encapsulation header will be added as the remote IP address for the table entry.

Unicast-only Mode

The unicast-only mode is a mechanism to support traditional broadcast domain environment without the multicast mode requirement on the physical infrastructure. The packet path functions in the same way as multicast mode, except for the handling of flooded packets (for example, unknown unicast, multicast, and broadcast). Each host understands all the VXLAN Tunnel End Points (VTEPs) of a VXLAN through the Virtual Supervisor Module (VSM) and the source host makes multiple encapsulated copies of the packet, one for each VTEP that is an active member of the VXLAN.

Fragmentation

The VXLAN encapsulation overhead is 50 bytes. In order to prevent performance degradation due to fragmentation, the entire interconnection infrastructure between all VEMs that exchange VXLAN packets must be configured to carry 50 bytes more than what the VM VNICs are configured to send. For example, if the default VNIC configuration is 1500 bytes, the VEM uplink port profile, upstream physical switch port, and interswitch links, and any routers if present, must be configured to carry a maximum transmission unit (MTU) of at least 1550 bytes. If that is not possible, we recommend that you configure the MTU within the guest VMs to be smaller by 50 bytes.

If you do not configure a smaller MTU, the VEM attempts to notify the VM if it performs Path MTU (PMTU) Discovery. If the VM does not send packets with a smaller MTU, the VM fragments the IP packets. Fragmentation occurs only at the IP layer. If the VM sends a frame that is too large, the frame will be dropped after VXLAN encapsulation and if the frame does not contain an IP packet.

Jumbo Frames

Jumbo frames are supported by the Cisco Nexus 1000V if there is space on the frame to accommodate the VXLAN encapsulation overhead of at least 50 bytes, and the physical switch/router infrastructure has the capability to transport these jumbo-sized IP packets.

VXLAN Feature Disabled

As a safety precaution, do not use the **no feature segmentation** command if there are any ports associated with a VXLAN port profile. You must remove all associations before you can disable this feature. You can use the **no feature segmentation** command to remove all the VXLAN bridge domain configurations on the Cisco Nexus 1000V.

CHAPTER 3

Configuring VXLANs

This chapter contains the following sections:

Prerequisites for VXLANs

VXLANs have the following prerequisites:

- You must create a VTEP on the host by defining it in the Red Hat Enterprise Linux OpenStack Platform Installer graphical user interface during the Openstack deployment. For more information, see the *Cisco Nexus 1000V for KVM Software Installation Guide*.
- If you plan to configure multiple VTEPs in virtual port channel host mode (vPC-HM) for load balancing in the same subnet, you need to set the vteps_in_the_same_subnet parameter to true in the Red Hat Enterprise Linux OpenStack Platform Installer graphical user interface before installing the Cisco Nexus 1000V for KVM. For more information, see the *Cisco Nexus 1000V for KVM Software Installation Guide*.
- The Cisco Nexus 1000V uplink port profiles and all interconnecting switches and routers between the KVM hosts must have their supported maximum transmission unit (MTU) set to at least 50 bytes larger than the MTU of the Virtual Machines (VMs). For example, the VMs default to using a 1500 byte MTU (same as the uplinks and physical devices), so you must set them to at least 1550 bytes. If this configuration is not possible, you should lower all VM vNICs MTU to 50 bytes smaller than what the physical network supports, such as 1450 bytes. For more information, see the *Cisco Nexus 1000V Port Profile Configuration Guide*.
- If the Cisco Nexus 1000V is using a port channel for its uplinks, you should set the load distribution algorithm to a 5-tuple hash (IP/Layer 4/Layer 4 ports). Use the same setting for any port channels on the physical switches. For more information, see the *Cisco Nexus 1000V Interface Configuration Guide*.

- By default, VXLAN uses MAC in IP (UDP) with a destination port of 8472. However, you can change this setting to the IANA assigned value of 4789 or any value between 1024 through 65535. Whichever port you use, you must allow it through any intermediate firewall.
- If you are using the VXLAN multicast mode, you must configure an IGMP querier in the VXLAN transport VLANs.

Guidelines and Limitations for VXLANs

VXLAN has the following configuration guidelines and limitations:

- You must configure and make all changes to VXLANs in OpenStack.

  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.
- When encapsulated traffic is destined to a VEM that is connected to a different subnet, the VEM does not use the Linux host routing table. Instead, it can use either Proxy Address Resolution Protocol (ARP) or a default gateway.
  - To use Proxy ARP, you must configure the upstream router for Proxy ARP. With ARP configured, if the remote VTEP is in the same subnet as the VXLAN Gateway, the VEM uses ARP to obtain the IP address of the remote VTEP. If the remote VTEP is in a different subnet than the VXLAN Gateway, the VEM uses ARP to obtain the IP address of the VXLAN Gateway.
  - To use a default gateway, you must configure the VTEP with the **transport ip address external** command to specify the netmask and gateway IP address for the VTEP to use. For example, from the interface command mode, enter **transport ip address external netmask 255.255.255.0 gateway 1.2.3.4**.
- If you configure load-balancing with a VPC-HM where multiple VTEPS exist in the same subnet on the KVM platform, you might experience a Linux kernel issue where ARP responses from the Linux kernel for the VTEPs might have the wrong MAC address. This situation could adversely affect the flow of VXLAN traffic.
- VXLANs in unicast-only mode are supported only between VTEPs that are managed by a single VSM. A VXLAN in unicast-only mode cannot be shared across two different distributed virtual switches.

Default Settings for VXLANs

The following table lists the default settings for VXLAN parameters.

Table 2: Default VXLAN Parameters

Parameter	Default
Feature Segmentation	Enabled

Configuring VXLANs

Steps to Configure VXLANs

You can configure a VXLAN using the OpenStack CLI or Horizon dashboard.

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.

Before You Begin

- Ensure that all prerequisites are met. For information, see Prerequisites for VXLANs, on page 9.
- Follow all guidelines and limitations. For information, see Guidelines and Limitations for VXLANs, on page 10.

Procedure

	Command or Action	Purpose
Step 1	Enable the Segmentation feature in the VSM using the CLI.	See Enabling the VXLAN Segmentation Feature, on page 12.
Step 2	Ensure that the VTEP on the host has been defined in the Red Hat Enterprise Linux OpenStack Platform Installer graphical user interface during the OpenStack deployment.	See *Cisco Nexus 1000V for KVM Software Installation Guide*.
Step 3	If you plan to configure multiple VTEPs in virtual port channel host mode (vPC-HM) for load balancing in the same subnet, you need to set the vteps_in_the_same_subnet parameter to true in the Red Hat Enterprise Linux OpenStack Platform Installer graphical user interface before installing the Cisco Nexus 1000V for KVM.	(Optional) See the *Cisco Nexus 1000V for KVM Software Installation Guide*.
Step 4	Using the VSM CLI, configure a vEthernet port profile with VXLAN capability.	
Step 5	Using the OpenStack CLI, create a multicast or unicast VXLAN network profile. **Example:** **neutron cisco-network-profile-create** *name* **vxlan --subtype multicast --segment_range** *segment-range* **--multicast_ip_range** *ip-range* **Example:**	For more information or to perform these steps using the OpenStack Horizon Dashboard, see the *Cisco Nexus 1000V for KVM Virtual Network Configuration Guide*.

	Command or Action	Purpose
	neutron cisco-network-profile-create *name* **vxlan --subtype unicast --segment_range** *segment-range*	
Step 6	Using the OpenStack CLI, create a network and associate it with a Cisco Nexus 1000V switch network profile. **Example:** **neutron net-create** *name* **--n1kv:profile_id** *profileId*	For more information or to perform these steps using the OpenStack Horizon Dashboard, see the *Cisco Nexus 1000V for KVM Virtual Network Configuration Guide*.
Step 7	Using the VSM CLI, verify that the VXLAN has been configured on the VSM. **Example:** **show bridge-domain brief**	This command lists all bridge domains and their corresponding status and ports.
Step 8	Save the configuration on the VSM. **Example:** **copy running-config startup-config**	—

Enabling the VXLAN Segmentation Feature

If you have installed the Cisco Nexus 1000V for KVM on a VM, the segmentation feature is enabled by default. However, if you have installed the Cisco Nexus 1000V for KVM on a Cloud Services Platform, you must enable the segmentation feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **show feature | grep segmentation**	(Optional) Displays whether the VXLAN feature is enabled.
Step 3	switch(config)# **feature segmentation**	Enables the VXLAN segmentation feature.
Step 4	switch(config)# **show feature | grep segmentation**	(Optional) Displays whether the VXLAN feature is enabled.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable the VXLAN segmentation feature:

```
switch# configure terminal
switch(config)# show feature | grep segmentation
network-segmentation 1 disabled
segmentation         1 disabled
switch(config)# feature segmentation
switch(config)# show feature | grep segmentation
network-segmentation 1 disabled
segmentation         1 enabled
switch(config)# copy running-config startup-config
```

Configuring a VTEP Profile for VXLAN Encapsulation

Before You Begin

- Identify a VLAN to be used for transporting VXLAN-encapsulated traffic.
- Ensure that the VLAN is configured on the uplink port profile for all VEMs on which the VXLAN can be configured.
- Create the VTEP on the host by defining it in the Red Hat Enterprise Linux OpenStack Platform Installer graphical user interface during the OpenStack deployment. For details, see the *Cisco Nexus 1000V for KVM Software Installation Guide*.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile type veth** *profilename*	Enters port profile configuration mode for the named port profile. If the port profile does not already exist, it is created using the following characteristics: • *profilename*—The port profile name can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V. **Note** If a port profile is configured as an Ethernet type, it cannot be used to configure VTEPs.
Step 3	switch(config-port-prof)# **switchport mode access**	Designates the interfaces as switch access ports (the default).
Step 4	switch(config-port-prof)# **switchport access vlan** *id*	Assigns a VLAN ID to this port profile. **Note** A VLAN ID must be created and should be in the active state.
Step 5	switch(config-port-prof)# **capability vxlan**	Assigns the VXLAN capability to the port profile to ensure that the interfaces that inherit this port profile are used as sources for VXLAN-encapsulated traffic.
Step 6	switch(config-port-prof)# **transport ip address external netmask** *netmask* [**gateway** *gw-ip*]	(Optional) Configures the VTEP with the netmask and gateway IP address to use to reach a VEM that is connected to a different

	Command or Action	Purpose
		subnet. Alternatively, you can configure the default router for Proxy ARP. For more information, see Guidelines and Limitations for VXLANs, on page 10. **Note** You must create the VTEP on the host by defining it in the Red Hat Enterprise Linux OpenStack Platform Installer graphical user interface during the OpenStack deployment. For details, see the *Cisco Nexus 1000V for KVM Software Installation Guide*.
Step 7	switch(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.
Step 8	switch(config-port-prof)# **state enabled**	Sets the operational state of a port profile.
Step 9	switch(config-port-prof)# **publish port-profile**	Pushes the port profile to the OpenStack controller.
Step 10	switch(config-port-prof)# **show port-profile name** *profilename*	Displays the port profile configuration.
Step 11	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure an interface for VXLAN encapsulation:

```
switch# configure terminal
switch(config)# port-profile type veth vxlan-pp
switch(config-port-prof)# switchport mode access
switch(config-port-prof)# switchport access vlan 100
switch(config-port-prof)# capability vxlan
switch(config-port-prof)# no shutdown
switch(config-port-prof)# state enabled
switch(config-port-prof)# publish port-profile
switch(config-port-prof)# show port-profile name vxlan-pp
port-profile vxlan-pp
type: Vethernet
description:
status: enabled
max-ports: 32
min-ports: 1
inherit:
config attributes:
switchport mode access
switchport access vlan 100
capability vxlan
no shutdown
evaluated config attributes:
switchport mode access
switchport access vlan 100
capability vxlan
no shutdown
assigned interfaces:
port-group: vmknic-pp
system vlans: none
```

```
capability l3control: no
capability iscsi-multipath: no
capability vxlan: yes
capability l3-vservice: no
port-profile role: none
port-binding: static

switch(config-port-prof)#
switch(config-port-prof)# copy running-config startup-config
```

Changing the UDP Port for VXLAN Encapsulation

You can change the default UDP port number to another port number.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vxlan udp port** *port-number*	Changes the UDP port to the specified port number. The default UDP port number is 8472. Valid port numbers are in the range 1024 to 65535.
Step 3	switch(config)# **show running-config | inc "vxlan udp"**	Displays the VXLAN UDP port number.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to change the UDP port to 4789:

```
switch# configure terminal
switch(config)# vxlan udp port 4789
switch(config)# show running-config | inc "vxlan udp"
vxlan udp port 5656
switch(config)# copy running-config startup-config
```

Disabling the VXLAN Segmentation Feature

If you have enabled the segmentation feature on a Cloud Services Platform, you can disable it. If you have installed the Cisco Nexus 1000V for KVM on a VM, the feature is enabled by default and cannot be disabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **show bridge-domain**	Displays all bridge domains.

	Command or Action	Purpose
		Note You must identify all bridge domains with nonzero port counts.
Step 3	switch(config)# **show running port-profile**	(Optional) Displays the running configuration for all port profiles. **Note** You must use this command to identify which port profiles have bridge domains identified in Step 2 configured.
Step 4	switch(config)# **port-profile** *name*	Enters configuration mode for the specified port profile.
Step 5	switch(config-port-prof)# **no switchport access bridge-domain** *name-string*	Removes the VXLAN bridge domain from the port profile and moves the ports to VLAN1.
Step 6	switch(config-port-prof)# **show port-profile usage**	(Optional) Displays a list of interfaces that inherited a port profile.
Step 7	switch(config-port-prof)# **show bridge-domain**	(Optional) Displays all bridge domains.
Step 8	switch(config-port-prof)# **no feature segmentation**	Removes the segmentation feature.
Step 9	switch(config-port-prof)# **show feature | grep segmentation**	(Optional) Displays if the segmentation feature is running or not running.
Step 10	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to disable segmentation:

```
switch# configure terminal
switch(config)# show bridge-domain

Global Configuration:
Mode: Unicast-only
MAC Distribution: Disable

Bridge-domain tenant-red (4 ports in all)
Segment ID: 4096 (Manual/Active)
Mode: Unicast-only
MAC Distribution: Disable
Group IP: NULL
State: UP Mac learning: Enabled
Veth1, Veth2, Veth4, Veth11

switch(config)# show running-config port-profile
port-profile default max-ports 32
port-profile default port-binding static
port-profile type ethernet Unused_Or_Quarantine_Uplink
```

```
vmware port-group
shutdown
description Port-group created for Nexus1000V internal usage. Do not use.
state enabled
port-profile type vethernet Unused_Or_Quarantine_Veth
vmware port-group
shutdown
description Port-group created for Nexus1000V internal usage. Do not use.
state enabled
port-profile type vethernet tenant-profile
vmware port-group
switchport mode access
switchport access bridge-domain tenant-red
no shutdown
state enabled

switch(config)#
switch(config-port-prof)# show port-profile usage

port-profile Unused_Or_Quarantine_Uplink

port-profile Unused_Or_Quarantine_Veth

port-profile tenant-profile
Vethernet1
Vethernet2
Vethernet4
Vethernet11

switch(config-port-prof)# show bridge-domain

Global Configuration:
Mode: Unicast-only
MAC Distribution: Disable

Bridge-domain tenant-red (0 ports in all)
Segment ID: 4096 (Manual/Active)
Mode: Unicast-only
MAC Distribution: Disable
Group IP: NULL
State: UP Mac learning: Enabled

switch(config-port-prof)#
switch(config-port-prof)# no feature segmentation
switch(config-port-prof)# 2013 May 23 05:34:42 switch-cy %SEG_BD-2-SEG_BD_DISABLED: Feature
 Segmentation disabled

switch(config-port-prof)# show feature | grep seg_bd
- NR - 1 - seg_bd
```

Verifying the VXLAN Configuration

You create a bridge domain on the VSM when you create a VXLAN network on the OpenStack controller. For more information, see the *Cisco Nexus 1000V for KVM Virtual Network Configuration Guide*.

To display the VXLAN configuration information, perform one of the following tasks:

Command	Purpose
show feature | grep segmentation	Displays if the segmentation feature is running.
show bridge-domain	Displays all bridge domains with the mode.

Command	Purpose
show bridge-domain brief	Lists all bridge domains and their corresponding status and ports.
show bridge-domain vteps	Displays the bridge domain-to-VTEP mappings that are maintained by the VSM and are pushed to all VEMs.
show run bridge-domain	Displays the running bridge domain.
show bridge-domain *bd-name*	Displays the specified bridge domain.
show bridge-domain *bd-name* **vteps**	Displays the specific bridge domain-to-VTEP mappings that are maintained by the VSM and are pushed to all VEMs.
show interface brief	Displays a short version of the interface configuration.
show interface switchport	Displays information about switchport interfaces.
show module vteps	Displays the IP addresses available on each module that can be used for VXLAN Tunnel Endpoints.

Feature History for VXLAN

Feature Name	Releases	Feature Information
VXLAN	Release 5.2(1)SK1(2.1)	Introduced the Virtual Extensible Local Area Network (VXLAN) feature, including the enhanced VXLAN commands.