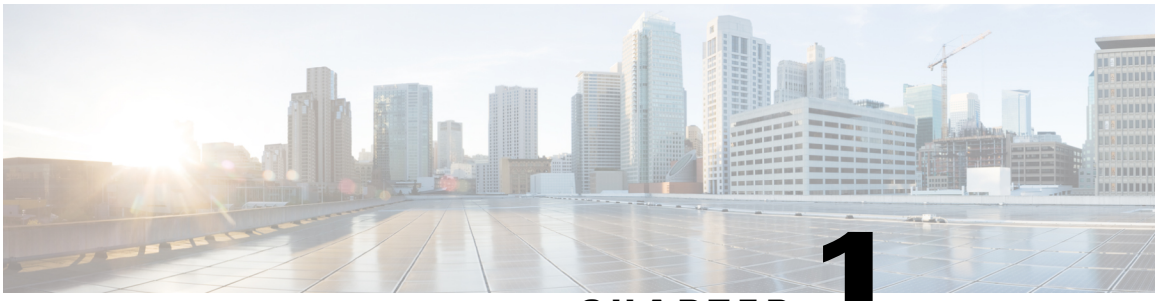
Cisco Nexus 1000V for VMware vSphere VXLAN Configuration Guide, Release 5.x

First Published: August 22, 2014

Last Modified: May 16, 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

CONTENTS



CHAPTER 1

New and Changed Information

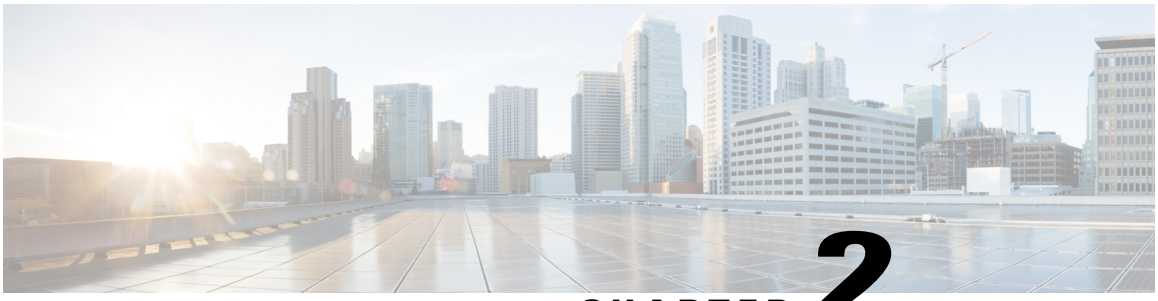
This chapter contains the following sections:

New and Changed Information

Table 1: New and Changed Features

Item	Description	Changed in Release	Where Documented
VXLAN UDP Port Number	Updated prerequisite information and added a procedure about how to change the VXLAN UDP port number.	5.2(1)SV3(1.1)	Prerequisites for VXLANs, on page 9 Changing the UDP Port for VXLAN Encapsulation, on page 20

New and Changed Information



CHAPTER 2

Overview

This chapter contains the following sections:

Information About VXLANs

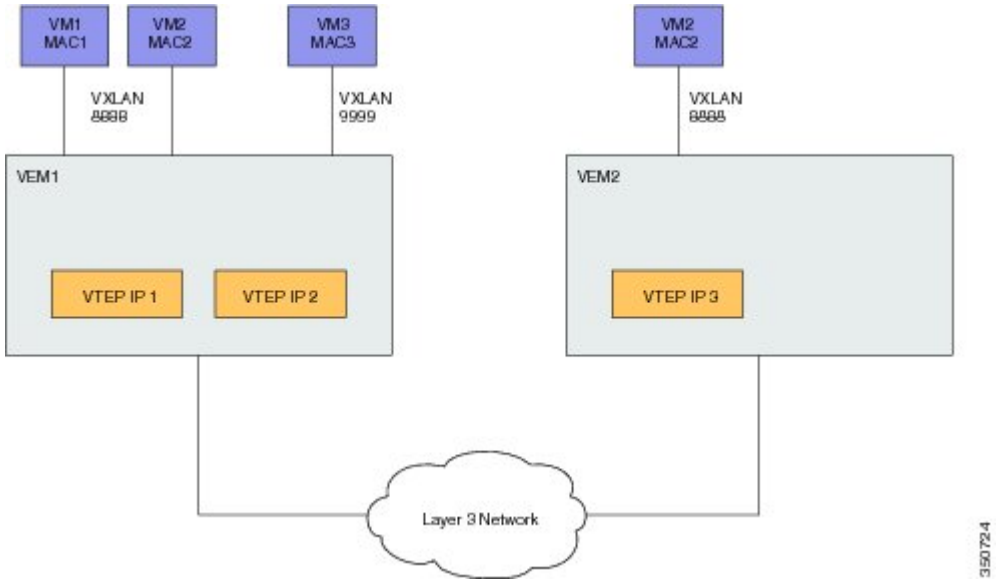
Overview of VXLANs

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. See the following figure.

The IP addresses, which are known as VXLAN Tunnel End Point (VTEP) IP addresses, are assigned to selected vmknics 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 three 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 joins 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.
- Unicast-only mode—A VXLAN uses each VEM's single unicast IP address as the destination IP address to send broadcast, multicast, and unknown unicast flood frames of the designated VTEP on each VEM that has at least one VM in the corresponding VXLAN. When a new VM joins 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 unicast MAC address are encapsulated and sent directly to the destination server's VTEP IP address.
- MAC distribution mode (supported only in unicast mode)—In this mode, unknown unicast flooding in the network is eliminated. The VSM learns all the MAC addresses from the VEMs in all the VXLANs and distributes those MAC addresses with VTEP IP mappings to other VEMs. Therefore, there is no unknown unicast MAC address in the network when the VMs on the VEMs are communicating and controlled by the same VM.

Note MAC distribution works only for static MAC addresses. If dynamic MAC addresses are found on ports that use VXLANs that operate in MAC distribution mode, syslogs are generated to indicate that MAC distribution does not work with dynamic MAC addresses.

Note You can configure the above modes globally and override them for each bridge domain.

By default, if you upgrade the VSM from an earlier version of the Cisco Nexus 1000V to the current version with the segmentation feature enabled, all the VXLANs continue to operate in multicast mode. If you enable the feature when the VSM is running the current version of the Cisco Nexus 1000V, by default, the bridge domains change to unicast-only mode. You must explicitly enable MAC distribution mode because it is disabled by default.

After completing the software upgrade, you need to explicitly configure the segment mode to multicast mode.

Note During an upgrade, you cannot enable unicast-only mode unless you upgrade all VEMs and the VEM level.

Border Gateway Protocol Control Plane Feature

The Border Gateway Protocol (BGP) control plane feature enables the Cisco Nexus 1000V to exchange the VXLAN information collected on the VSM-VTEP flood list across VSMs. The Cisco Nexus 1000V supports BGP peering between 16 VSMs to allow VXLAN segments to reach across servers. BGP runs on the VSM and can exchange VXLAN information with the BGP on any other Cisco Nexus 1000V. The Cisco Nexus 1000V can also be used as a route reflector to exchange VTEP list between VSMs.

The BGP control plane feature extends the unicast-only mode to a multi-VSM environment using a L2VPN EVPN address family. The VTEP information is not exchanged with the VSMs that are running the old version. They will continue to work in multicast mode (VXLAN 1.0) or unicast-only mode in a single Cisco Nexus 1000V (VXLAN 1.5).

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. The IP/MAC address that the VTEP uses is configured when you enter the **capability vxlan** command. You can have a maximum of four VTEPs in a single VEM.

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 VMware host routing table. Instead, it can use one of the following:

- **Proxy Address Resolution Protocol (ARP)**: 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.
- **Default 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**.

Note VMs brought up behind VEMs cannot use the transport VLAN of the VTEP, because VLANs used on VTEPs are isolated and reserved for VXLAN traffic only.

VXLAN Gateway

VXLAN termination (encapsulation and decapsulation) is supported on virtual switches. As a result, the only endpoints that can connect into VXLANs are VMs that are connected to a virtual switch. Physical servers cannot be in VXLANs and routers or services that have traditional VLAN interfaces cannot be used by VXLAN networks. The only way that VXLANs can currently interconnect with traditional VLANs is through VM-based software routers.

The supported gateways are as follows:

- VMware vShield Edge
- Cisco VXLAN Gateway
- Cisco ASA1000V

The configuration for such VLAN-VXLAN translation/mappings for the VXLAN gateway must be configured from the VSM and must always be a 1:1 mapping for each Layer 2 domain. Each VXLAN gateway can support multiple VLAN-VXLAN mappings.

VXLAN Trunks

A VXLAN trunk allows you to trunk multiple VXLANs on a single virtual Ethernet interface. In order to achieve this configuration, you must encapsulate a VLAN-VXLAN mapping on the virtual Ethernet interface.

VLAN-VXLAN mappings are applied on a virtual Ethernet interface using a port profile. A single port profile can support multiple VLAN-VXLAN mappings.

Multi-MAC Capability

You must use the multi-MAC capability feature to mark a virtual Ethernet interface as capable of sourcing packets from multiple MAC addresses. For example, you can use this feature if you have a virtual Ethernet port and you have enabled VXLAN trunking on it and the VM that is connected to the port bridges packets that are sourced from multiple MAC addresses.

By using this feature, you can easily identify such multi-MAC capable ports and handle live migration scenarios correctly for those ports.

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.

Scalability

Maximum Number of VXLANs

The Cisco Nexus 1000V supports up to 6000 VLANs and 6000 VXLANs with a combined maximum of 12000.

Supported Features

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.

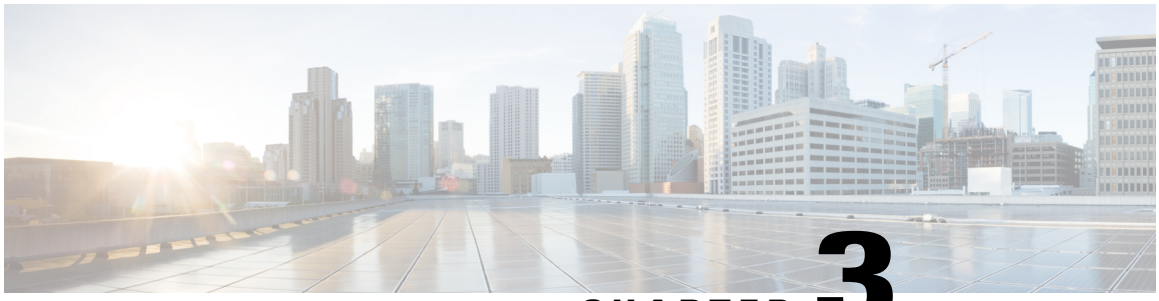
VXLAN Feature Disabled

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.

VXLAN Offload

The Cisco Nexus 1000V supports offloading VXLAN checksum and TSO computations of inner packets for VXLAN-encapsulated packets. The VXLAN offload feature is supported only if an adapter supports the offload feature and VMware supports the offload feature on that adapter. To verify if the Cisco Nexus 1000V supports the VXLAN offload feature on an adapter, use the **vemcmd show pd-port** command on the host.

The V flag in the Flags column indicates that the VXLAN offload feature is supported. The TSO computations are automatically offloaded when the VXLAN offload feature is supported.



CHAPTER 3

Configuring VXLANs

This chapter contains the following sections:

Information About VXLANs

Prerequisites for VXLANs

VXLANs have the following prerequisites:

- The Cisco Nexus 1000V uplink port profiles and all interconnecting switches and routers between the ESX 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*.
- If you are using the VXLAN multicast mode, you must configure an IGMP querier in the VXLAN transport VLANs.
- VXLAN uses MAC in user datagram protocol (MAC-in-UDP) destination port 4789. You must permit this port through any intermediate firewall.

- The VXLAN UDP port number is used in VXLAN encapsulation. In Cisco Nexus 1000V for VMware Release 4.2(1)SV2(2.1) and earlier, the default UDP port number was 8472. Beginning with Release 5.2(1)SV3(1.1), the default UDP port number has changed to the recently IANA-approved UDP port number 4789. This change affects the Cisco Nexus 1000V for VMware software installation, upgrade, and VXLAN configuration in the following ways:
  - When you upgrade to Release 5.2(1)SV3(1.1) from an earlier release that has VXLAN configured, the switch retains the UDP port number of 8472. You are not required to change the UDP number to the IANA approved UDP port number 4789; however if you decide to change it, make sure that the VEMs are upgraded to the Release 5.2(1)SV3(1.1) as well. Otherwise the **vxlan udp port** command is not available and you cannot change the UDP number.
  - When you upgrade to Release 5.2(1)SV3(1.1) from an earlier release that does not have VXLAN configured, and then you configure VXLAN, the switch is configured with the default, IANA approved UDP port number 4789.
  - When you perform a fresh Cisco Nexus 1000V for VMware installation of Release 5.2(1)SV3(1.1) and you configure VXLAN, the switch is configured with the default, IANA approved UDP port number 4789.
  - You can change the UDP port number at any time using the **vxlan udp port** command. However, when upgrading to Release 5.2(1)SV3(1.1) from an earlier release, ensure that the VSM and VEMs are at the same release before using the **vxlan udp port** command. Otherwise the **vxlan udp port** command is not available and you cannot change the UDP number.

Guidelines and Limitations for VXLANs

VXLAN has the following configuration guidelines and limitations:

- VMs brought up behind VEMs cannot use the transport VLAN of the VTEP, because VLANs used on VTEPs are isolated and reserved for VXLAN traffic only.
- When a VXLAN is configured in the unicast-only mode with MAC distribution enabled, the VXLAN gateway does not register any MAC addresses that it learns on the VLAN side. If these MAC addresses have not been learned yet, the traffic to these MAC addresses is delivered by replicating of unknown unicast packets to the VXLAN gateway. This is the only scenario where unknown unicast packets are replicated in the MAC distribution mode.
- Microsoft Network Load Balancing (NLB) servers in unicast mode require unknown unicast packets to be delivered to all the server ports, because the shared MAC address of the NLB servers is never discovered. This solution will break the unknown unicast semantics of unicast-only mode with MAC distribution. We recommend that you use either multicast mode or unicast-only mode without MAC distribution.
- You cannot enable the MAC distribution mode and the multi-MAC capability feature together. You must use either the MAC distribution or the muti-MAC capability feature.
- The Cisco Nexus 1000V switch in ESXi 5.5 supports VXLAN offload NICs. The Cisco Nexus 1000V switch is designed to assume that either all or none of the physical NICs (PNICs) in a port channel support the VXLAN offload capability.

VXLAN has the following configuration guidelines and limitations for changing the VXLAN configuration:

- Use the **segment control-protocol bgp** command to enable or disable the BGP control plane feature. You can configure each bridge domain (if present) to override the global configuration.
- Use the **segment mode unicast-only** command to change the global configuration mode from multicast to unicast. This command affects all bridge domains with no overrides.
- You can use multicast or unicast mode if you override the global configuration for the bridge domain by entering the **segment mode unicast-only** or **no segment mode unicast-only** commands.
- You can enable the segment distribution MAC command only after entering the **segment mode unicast-only** command.
- You can disable the segment distribution MAC address configuration globally by entering the **no segment distribution mac** command.
- You cannot use the **no segment mode unicast-only** command if you already entered the **segment distribution MAC** command.
- You must configure a multicast IP address that is required for a VXLAN that is in the multicast mode.
- If you remove the multicast IP address while VXLAN is in the multicast mode, the ports that use that VXLAN go to the inactive state.

Note Ports become inactive if you change the mode from unicast to multicast if a multicast IP address is not configured or a segment ID is removed.

Default Settings for VXLANs

The following table lists the default settings for VXLAN parameters.

Table 2: Default VXLAN Parameters

Parameter	Default
Feature Segmentation	Disabled

Configuring VXLANs

Initial Enabling of VXLANs

To enable a VXLAN, you must perform the following two procedures when you first configure a VXLAN.

- Configuring vmknics for VXLAN Encapsulation, on page 13
- Enabling VXLANs, on page 12

Enabling VXLANs

Before You Begin

Enter the **show system vem feature level** command to confirm that the feature level is 5.2(1)SV3(1.1) or a later release. If the feature level is not or 5.2(1)SV3(1.1) or a later release, see the *Cisco Nexus 1000V Installation and Upgrade Guide*.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature segmentation**
3. (Optional) switch(config)# **show feature** | **grep segmentation**
4. switch (config)# [**no**] **segment mode unicast-only**
5. switch (config)# [**no**] **segment distribution mac**
6. (Optional) switch (config)# **segment control-protocol bgp**
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature segmentation**	Enables the VXLAN.
Step 3	switch(config)# **show feature** | **grep segmentation**	(Optional) Displays whether the VXLAN is enabled.
Step 4	switch (config)# [**no**] **segment mode unicast-only**	Configures the global configuration mode for all VXLAN bridge domains. If the configuration mode is not entered, the global mode is unicast-only mode without MAC distribution.
Step 5	switch (config)# [**no**] **segment distribution mac**	Enables or disables MAC distribution globally. All bridge domains with default MAC distribution mode will inherit this configuration and enable or disable MAC distribution.
Step 6	switch (config)# **segment control-protocol bgp**	(Optional) Use the segment control-protocol bgp command to enable or disable BGP control plane feature, with per bridge domain configuration (if present) overriding the global configuration.
Step 7	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 segmentation feature:

```
switch# configure terminal
switch(config)# feature segmentation
switch(config)# show feature | grep segmentation
network-segmentation 1 disabled
segmentation         1 enabled
switch(config)# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
```

Configuring vmknics for VXLAN Encapsulation

Before You Begin

- Identify a VLAN to be used for transporting VXLAN-encapsulated traffic.
- Ensure that it is configured on the uplink port profile for all VEMs on which the VXLAN can be configured.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-profile type veth** *profilename*
3. switch(config-port-prof)# **vmware port-group** *name*
4. switch(config-port-prof)# **switchport mode access**
5. switch(config-port-prof)# **switchport access vlan** *id*
6. switch(config-port-prof)# **capability vxlan**
7. (Optional) switch(config-port-prof)# **capability l3control**
8. switch(config-port-prof)# **no shutdown**
9. switch(config-port-prof)# **state enabled**
10. switch(config-port-prof)# **show port-profile name** *profilename*
11. (Optional) switch(config-port-prof)# **copy running-config startup-config**

DETAILED STEPS

	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 VMware virtual ports.

	Command or Action	Purpose
Step 3	switch(config-port-prof)# **vmware port-group** *name*	Designates the port profile as a VMware port group. The port profile is mapped to a VMware port group of the same name unless you specify a name here. When you connect the VSM to vCenter Server, the port group is distributed to the virtual switch on vCenter Server.
Step 4	switch(config-port-prof)# **switchport mode access**	Designates the interfaces as switch access ports (the default).
Step 5	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 6	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 7	switch(config-port-prof)# **capability l3control**	(Optional) Assigns the l3control capability to the port profile.
Step 8	switch(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.
Step 9	switch(config-port-prof)# **state enabled**	Sets the operational state of a port profile.
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 a vmknic for VXLAN encapsulation:

```
switch# configure terminal
switch(config)# port-profile type veth vmknic-pp
switch(config-port-prof)# vmware port-group
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)# show port-profile name vmknic-pp
port-profile vmknic-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
[########################################] 100%
Copy complete, now saving to disk (please wait)...
```

What to Do Next

The vSphere administrator must create a new vmknic on each ESX/ESXi host and assign the previously created port profile to this vmknic. IP address and netmask should be assigned to the vmknic. This IP address will be used for VXLAN packet encapsulation. Use the **show module vteps** to view the interfaces on the VSM.

Creating a Bridge Domain

You are limited to creating a maximum of 6000 VXLAN bridge domains.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **bridge-domain** *name-string*
3. switch(config-bd)# **segment id** [*number*]
4. (Optional) switch(config-bd)# **group** *ipaddr*
5. (Optional) switch(config-bd)# **show bridge-domain** *name-string*
6. (Optional) switch(config-bd)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **bridge-domain** *name-string*	Creates a VXLAN and associates an identifying name to it.
Step 3	switch(config-bd)# **segment id** [*number*]	Specifies the VXLAN segment ID. Only one bridge domain can use a particular segment ID value. Valid values are from 4096 to 16000000. (1 to 4095 are reserved for VLANs.)
Step 4	switch(config-bd)# **group** *ipaddr*	(Optional) Associates the multicast group for broadcasts and floods.

	Command or Action	Purpose
		Note Reserved multicast addresses are not allowed.
Step 5	switch(config-bd)# **show bridge-domain** *name-string*	(Optional) Displays bridge domain information.
Step 6	switch(config-bd)# **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 create a VXLAN:

```
switch# configure terminal
switch(config)# bridge-domain tenant-red
switch(config-bd)# segment id 4096
switch(config-bd)# group 239.1.1.1
switch(config-bd)# show bridge-domain vxlan-bd-9
Bridge-domain vxlan-bd-9 (2 ports in all)
Segment ID: 10009 (Manual/Active)
Mode: Unicast-only (override)
MAC Distribution: Enable (override)
BGP control mode: Enable
Group IP: NULL
State: UP               Mac learning: Enabled
Veth16, Veth17
switch(config-bd)#
switch(config-bd)# copy running-config startup-config
```

Configuring the Bridge Domain Mode

You can configure a bridge domain in the bridge-domain mode or global mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch# **bridge-domain** *bd-name*
3. switch (config-bd)# **segment id** *[number]*
4. switch (config-bd)# [**no**] **segment mode unicast-only** | **default segment mode**
5. switch (config-bd)# [no] **segment distribution mac** | **default segment distribution mac**
6. switch (config-bd)# [no] **segment control-protocol bgp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch# **bridge-domain** *bd-name*	Creates a bridge domain.

	Command or Action	Purpose
Step 3	switch (config-bd)# **segment id** *[number]*	Specifies the VXLAN segment ID. Only one bridge domain can use a particular segment ID value. Valid values are from 4096 to 16000000. (1 to 4095 are reserved for VLANs.)
Step 4	switch (config-bd)# [**no**] **segment mode unicast-only** | **default segment mode**	Configures the segment mode as unicast only. The mode can be configured globally or for a specific bridge domain. When configured under a specific bridge domain, the mode is treated as an override to the global configuration for that specific bridge domain. Any change in the global configuration affects all the bridge domains that do not have overrides. The mode configuration on a specific bridge domain overwrites the global bridge domain. The overrides configured on the bridge domain can be removed by using the **default segment mode**. **Note** Use the **no segment mode unicast-only** command to override the configuration under a bridge domain. If you have unicast enabled globally, the bridge domain can use the multicast mode. To override, use the **default segment mode** command. **Note** This command cannot be performed globally or under a bridge domain if the **segment distribution MAC** feature is configured.
Step 5	switch (config-bd)# [no] **segment distribution mac** | **default segment distribution mac**	Enables MAC distribution for the bridge domain. **Note** To configure an override under a bridge domain, you must enter the **segment mode unicast-only** command as an override first.
Step 6	switch (config-bd)# [no] **segment control-protocol bgp**	Enables BGP control plane.The BGP control plane can be configured globally or for a specific bridge domain. When configured under a specific bridge domain, it is treated as an override to the global configuration for that specific bridge domain. Any change in the global configuration affects all the bridge domains that do not have overrides. The configuration on a specific bridge domain overwrites the global bridge domain. The overrides configured on the bridge domain can be removed by using the default setting.

This example shows how to configure a bridge domain:

Note The ports are inactive if a segment ID is not configured for a bridge domain and if a multicast IP address is not configured when global configuration or a bridge domain override has the **no segment mode unicast-only** configuration.

```
config terminal
bridge-domain domain-660
segment id 4097
    segment mode unicast-only
    segment distribution mac
```

Creating a Port Profile Configured to Use a VXLAN

Alternatively, you can associate ports with a bridge domain by modifying the configuration of an existing virtual Ethernet port profile to use VXLANs instead of VLANs. To do so, enter the **switchport access bridge-domain name** command on a profile with switchport mode access configured.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-profile** [**type** {**vethernet**}] *name*
3. switch(config-port-prof)# **vmware port-group** [*pg_name*]
4. switch(config-port-prof)# **switchport mode access**
5. switch(config-port-prof)# **switchport access bridge-domain** *bridge-domain name*
6. switch(config-port-prof)# **no shutdown**
7. switch(config-port-prof)# **state enabled**
8. (Optional) switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]
9. (Optional) switch(config-port-prof)# **show running-config bridge-domain**
10. (Optional) switch(config-port-prof)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**vethernet**}] *name*	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: • name—The port profile name can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V. • type—The port profile type is virtual Ethernet. Once configured, the type cannot be changed. The default is the virtual Ethernet type. Defining a port profile type as Ethernet allows the port profile to be used for physical (Ethernet) ports. In vCenter Server, the corresponding port group can be selected and assigned to physical ports (PNICs). **Note** If a port profile is configured as an Ethernet type, it cannot be used to configure VMware virtual ports.
Step 3	switch(config-port-prof)# **vmware port-group** [*pg_name*]	Designates the port profile as a VMware port group. The port profile is mapped to a VMware port group of the same name unless you specify a name here. When you connect the VSM to vCenter Server, the port group is distributed to the virtual switch on vCenter Server.

	Command or Action	Purpose
Step 4	switch(config-port-prof)# **switchport mode access**	Designates that the interfaces are to be used as trunking ports. A trunk port transmits untagged packets for the native VLAN and transmits encapsulated, tagged packets for all other VLANs.
Step 5	switch(config-port-prof)# **switchport access bridge-domain** *bridge-domain name*	Assigns a VXLAN bridge domain to this port profile. You must configure the bridge domain with its segment ID for the port to be active. You should configure a multicast IP address if you prefer multicast mode. The multicast mode is displayed in the running configuration as **no segment mode unicast-only**.
Step 6	switch(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.
Step 7	switch(config-port-prof)# **state enabled**	Sets the operational state of a port profile.
Step 8	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	(Optional) Displays the configuration for verification.
Step 9	switch(config-port-prof)# **show running-config bridge-domain**	(Optional) Displays the segmentation configuration.
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 create a port profile configured to use a VXLAN:

```
switch# configure terminal
switch(config)# port-profile tenant-profile
switch(config-port-prof)# vmware port-group
switch(config-port-prof)# switchport mode access
switch(config-port-prof)# switchport access bridge-domain tenant-red
switch(config-port-prof)# no shutdown
switch(config-port-prof)# state enabled
switch(config-port-prof)# show port-profile name tenant-profile
port-profile tenant-profile
type: Vethernet
description:
status: enabled
max-ports: 32
min-ports: 1
inherit:
config attributes:
switchport mode access
switchport access bridge-domain tenant-red
no shutdown
evaluated config attributes:
switchport mode access
switchport access bridge-domain tenant-red
no shutdown
assigned interfaces:
port-group: tenant-profile
system vlans: none
capability l3control: no
capability iscsi-multipath: no
capability vxlan: no
capability l3-vservice: no
```

```
port-profile role: none
port-binding: static

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

!Command: show running-config bridge-domain
!Time: Tue Apr 15 00:55:58 2014

version 5.2(1)SV3(1.1)
feature segmentation
segment mode unicast-only
segment control-protocol bgp

bridge-domain tenant-red
  segment id 4096
  group 239.1.1.1


switch(config-port-prof)# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
```

Changing the UDP Port for VXLAN Encapsulation

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vxlan udp port** *port-number*
3. switch(config)# **show running-config | inc "vxlan udp"**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	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 4789. Valid port numbers are in the range 1024 to 65535. **Note** You must permit this port through any intermediate firewall.
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)# show running-config | inc "vxlan udp"
vxlan udp port 8472
switch(config)# vxlan udp port 4789
switch(config)# show running-config | inc "vxlan udp"
vxlan udp port 4789
switch(config)# copy running-config startup-config
```

Removing Ports from a VXLAN

By performing this procedure, you move the ports to the default VLAN.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-profile** [**type** {**vethernet**}] *name*
3. switch(config-port-prof)# **no switchport access bridge-domain**
4. (Optional) switch(config-port-prof)# **show port-profile usage**
5. (Optional) switch(config-port-prof)# **show bridge-domain**
6. (Optional) switch(config-port-prof)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**vethernet**}] *name*	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: • **name—**The port profile name can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V. • **type—**The port profile type is vEthernet. Once configured, the type cannot be changed. The default is the vEthernet type. Defining a port profile type as Ethernet allows the port profile to be used for physical (Ethernet) ports. In vCenter Server, the corresponding port group can be selected and assigned to physical ports (PNICs). **Note** If a port profile is configured as an Ethernet type, it cannot be used to configure VMware virtual ports.
Step 3	switch(config-port-prof)# **no switchport access bridge-domain**	Removes the VXLAN bridge domain from this port profile.
Step 4	switch(config-port-prof)# **show port-profile usage**	(Optional) Displays a list of interfaces that inherited a port profile.

	Command or Action	Purpose
Step 5	switch(config-port-prof)# **show bridge-domain**	(Optional) Displays all bridge domains.
Step 6	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 remove ports from a VXLAN:

```
switch# configure terminal
switch(config)# port-profile tenant-profile
switch(config-port-prof)# no switchport access bridge-domain tenant-red
switch(config-port-prof)# show port-profile usage

port-profile N1K_Cloud_Default_Trunk

port-profile tenant-profile

port-profile Unused_Or_Quarantine_Uplink

port-profile Unused_Or_Quarantine_Veth

port-profile UPLINK_AIPC_INBAND

port-profile UPLINK_HOST1
 port-channel1
 Ethernet3/2
 Ethernet3/3

port-profile UPLINK_HOST3
 port-channel2
 Ethernet4/2
 Ethernet4/3

port-profile vmknic-pp

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

Global Configuration:
Mode: Unicast-only
MAC Distribution: Disable
BGP control mode: Enable

Bridge-domain tenant-red (0 ports in all)
Segment ID: 4096 (Manual/Active)
Mode: Unicast-only
MAC Distribution: Disable
BGP control mode: Enable
Group IP: 239.1.1.1
State: UP               Mac learning: Enabled


switch(config-port-prof)# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
```

Deleting a VXLAN

When you delete an existing bridge domain with ports on it, all the ports are moved to a down state and traffic stops flowing.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no bridge-domain group-red**
3. (Optional) switch(config-bd)# **show bridge-domain**
4. (Optional) switch(config-bd)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no bridge-domain group-red**	Deletes a VXLAN.
Step 3	switch(config-bd)# **show bridge-domain**	(Optional) Displays all bridge domains.
Step 4	switch(config-bd)# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to delete a VXLAN:

```
switch# configure terminal
switch(config)# no bridge-domain group-red
switch(config)# show bridge-domain
switch(config)# copy running-config startup-config
```

Disabling Segmentation

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **show bridge-domain**
3. (Optional) switch(config)# **show running port-profile**
4. switch(config)# **port-profile** *name*
5. switch(config-port-prof)# **no switchport access bridge-domain** *name-string*
6. (Optional) switch(config-port-prof)# **show port-profile usage**
7. (Optional) switch(config-port-prof)# **show bridge-domain**
8. switch(config-port-prof)# **no feature segmentation**
9. (Optional) switch(config-port-prof)# **show feature | grep segmentation**
10. (Optional) switch(config-port-prof)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **show bridge-domain**	Displays all bridge domains. **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*	Names the port profile and enters port profile configuration mode. If the port profile does not already exist, it is created using the following characteristics: *name*—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 VMware virtual ports.
Step 5	switch(config-port-prof)# **no switchport access bridge-domain** *name-string*	Removes the VXLAN bridge domain from this 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.

	Command or Action	Purpose
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

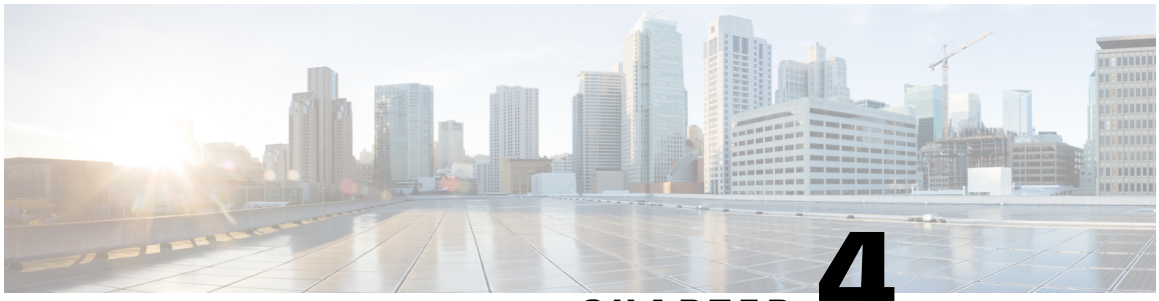
To display the VXLAN configuration information, use one of the following commands:

Command	Purpose
show feature | grep segmentation	Displays if the segmentation feature is running.
show bridge-domain	Displays all bridge domains with the mode.
show bridge-domain vteps	Displays the bridge domain-to-VTEP mappings that are maintained by the VSM and are pushed to all VEMs. Remote Cisco Nexus 1000V VTEPs that are learned through BGP are designated with the **Remote** keyword.
show bridge-domain *bd-name mac*	Displays all the MAC addresses that are learned by the VSMs on VXLANs that are configured with the MAC distribution feature.
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.

Command	Purpose
show bridge-domain module	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
Enhanced VXLAN	4.2(1)SV2(2.1)	Added the enhanced VXLAN commands.
VXLAN	4.2(1)SV1(5.1)	Introduced the Virtual Extensible Local Area Network (VXLAN) feature.
BGP Control Plane	5.2(1)SV3(1.1)	Introduced the Border Gateway Protocol (BGP) Control Plane feature.



CHAPTER 4

Configuring BGP Control Plane

This chapter contains the following sections:

Information About BGP Control Plane

The Border Gateway Protocol (BGP) control plane feature enables the Cisco Nexus 1000V to exchange the VXLAN information collected on the VSM-VTEP flood list across VSMs. The Cisco Nexus 1000V supports BGP peering between 16 VSMs to allow VXLAN segments to reach across servers. BGP runs on the VSM and can exchange VXLAN information with the BGP on any other Cisco Nexus 1000V. The Cisco Nexus 1000V can also be used as a route reflector to exchange VTEP list between VSMs.

The BGP control plane feature extends the unicast-only mode to a multi-VSM environment using a L2VPN EVPN address family. The VTEP information is not exchanged with the VSMs that are running the old version. They will continue to work in multicast mode (VXLAN 1.0) or unicast-only mode in a single Cisco Nexus 1000V (VXLAN 1.5).

Prerequisites for BGP Control Plane

The BGP control plane has the following prerequisites:

- The VSM must be running the latest release. The VTEP information is not exchanged with VSMs that are running older release. They will continue to work in multicast mode or unicast only mode in a single Cisco Nexus 1000V.

- On the control0 interface, an IP address is configured for BGP peering.
- The Switch must have a valid NEXUS_1000V_ADVANCED_PACKAGE license (version 3.0) is installed.

Configuring BGP

You must configure a BGP routing process and BGP peers.

Enabling BGP

You must enable BGP before you can configure BGP.

Before You Begin

Ensure that you are in the correct VDC.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature bgp**	Enables BGP. Use the **no feature bgp** command to disable BGP and remove all associated configuration.
Step 3	switch(config)# **interface control0**	Enters the interface control0 phase.
Step 4	switch(config-if)# **ip address 14.17.199.1/24**	Assigns IP address.
Step 5	switch(config-if)# **vrf context default**	Assigns default vrf context.
Step 6	switch(config-if)# **ip route 0.0.0.0/0 14.17.199.254**	Assigns an IP route.
Step 7	switch(config-if)# **exit**	(Optional) Exits the interface mode.
Step 8	switch(config)# **show feature**	(Optional) Displays enabled and disabled features.
Step 9	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 BGP:

```
switch# configure terminal
switch(config)# feature bgp
switch(config)# interface control0
```

```
switch(config-if)# ip address 14.17.199.1/24
switch(config-if)# vrf context default
switch(config-if)# ip route 0.0.0.0/0 14.17.199.254
switch(config-if)# exit
switch(config)# show feature
Feature Name                  Instance  State
----------------------------  --------  --------
bgp                           1         enabled
```

Creating a BGP Instance

You can create a BGP instance and assign a router ID to the BGP instance. Cisco NX-OS supports 2-byte or 4-byte autonomous system (AS) numbers in plain-text notation or as.dot notation.

Before You Begin

- You must enable BGP.
- BGP must be able to obtain a router ID (for example, a configured loopback address).

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
Step 3	switch(config-router)# **router-id** *ip-address*	Configures the BGP router ID. This IP address identifies this BGP speaker.
Step 4	switch(config-router-af)# **address-family l2vpn evpn**	Enters router family configuration mode for the l2vpn evpn address family.
Step 5	switch(config-router-af)# **show bgp all**	(Optional) Displays information about all BGP address families.
Step 6	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 BGP with the l2vpn evpn address family and manually add one network to advertise:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# router-id 192.169.67.11
switch(config-router)# address-family l2vpn evpn
switch(config-router-af)# copy running-config startup-config
```

Restarting a BGP Instance

You can restart a BGP instance and clear all peer sessions for the instance.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config)# **restart bgp** *instance-tag*	Restarts the BGP instance and resets or reestablishes all peering sessions.

Shutting Down BGP

You can shut down the BGP protocol and gracefully disable BGP and retain the configuration.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router)# **shutdown**	Gracefully shuts down BGP.

Configuring BGP Peers

You can configure a BGP peer within a BGP process. Each BGP peer has an associated keepalive timer and hold timers. You can set these timers either globally or for each BGP peer. A peer configuration overrides a global configuration.

Note You must configure the address family under neighbor configuration mode for each peer.

Before You Begin

- You must enable BGP.
- Ensure that you are in the correct VDC.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
Step 3	switch(config-router)# **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *as-number*	Configures the IP address and AS number for a remote BGP peer. The ip-address format is x.x.x.x. The ipv6-address format is A:B::C:D.
Step 4	switch(config-router-neighbor)# **password** *shared-password*	(Optional) Specifies a password for MD5 authentication per TCP packet.
Step 5	switch(config-router-neighbor)# **description** *text*	(Optional) Adds a description for the neighbor. The description is an alphanumeric string up to 80 characters.
Step 6	switch(config-router-neighbor)# **timers** *keepalive-time hold-time*	(Optional) Adds the keepalive and hold time BGP timer values for the neighbor. The range is from 0 to 3600 seconds. The default is 60 seconds for the keepalive time and 180 seconds for the hold time.
Step 7	switch(config-router-neighbor)# **address-family l2vpn evpn**	Enters neighbor address family configuration mode for the l2vpn evpn address family.
Step 8	switch(config-router-neighbor)# **send-community extended**	
Step 9	switch(config-router-neighbor)# **show bgp session**	(Optional) Displays all the running BGP sessions.
Step 10	switch(config-router-neighbor)# **show bgp l2vpn evpn neighbors**	(Optional) Displays information about BGP peers.
Step 11	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 configure a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1 remote-as 64497
switch(config-router)# password password1
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family l2vpn evpn
switch(config-router-neighbor-af)# send-community extended
switch(config-router-neighbor-af)# copy running-config startup-config
```

Peer Templates

The Cisco Nexus 1000V implements three types of peer templates:

- The *peer-session* template defines BGP peer session attributes, such as the transport details, remote autonomous system number of the peer, and session timers. A peer-session template can also inherit attributes from another peer-session template (with locally defined attributes that override the attributes from an inherited peer-session).
- A *peer-policy* template defines the address-family dependent policy aspects for a peer including the inbound and outbound policy, filter lists, and prefix lists. A peer-policy template can inherit from a set of peer-policy templates. The Cisco Nexus 1000V evaluates these peer-policy templates in the order specified by the preference value in the inherit configuration. The lowest number is preferred over higher numbers.
- The *peer* template can inherit the peer-session and peer-policy templates to allow for simplified peer definitions. It is not mandatory to use a peer template but it can simplify the BGP configuration by providing reusable blocks of configuration.

Configuring BGP Session Templates

You can use BGP session templates to simplify the BGP configuration for multiple BGP peers with similar configuration needs. BGP templates allow you to reuse common configuration blocks. You configure BGP templates first and then apply these templates to BGP peers.

With BGP session templates, you can configure session attributes such as inheritance, passwords, timers, and security.

A peer-session template can inherit from one other peer-session template. You can configure the second template to inherit from a third template. The first template also inherits this third template. This indirect inheritance can continue for up to seven peer-session templates.

Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Before You Begin

- You must enable BGP.
- Ensure that you are in the correct VDC.

Note When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the **default** form of the command to reset that attribute to the default state.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **template peer-session** *template-name*	Enters peer-session template configuration mode.
Step 4	switch(config-router-stmp)# **password** *number password*	(Optional) Adds the clear text password test to the neighbor. The password is stored and displayed in type 3 encrypted form (3DES).
Step 5	switch(config-router-stmp)# **timers** *keepalive hold*	(Optional) Adds the BGP keepalive and holdtimer values to the peer-session template. The default keepalive interval is 60. The default hold time is 180.
Step 6	switch(config-router-stmp)# **exit**	Exits peer-session template configuration mode.
Step 7	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	switch(config-router-neighbor)# **inherit peer-session** *template-name*	Applies a peer-session template to the peer.
Step 9	switch(config-router-neighbor)# **description** *text*	(Optional) Adds a description for the neighbor.
Step 10	switch(config-router-neighbor)# **show bgp peer-session** *template-name*	(Optional) Displays the peer-policy template.
Step 11	switch(config-router-neighbor)# **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 a BGP peer-session template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BaseSession
switch(config-router-stmp)# timers 30 90
switch(config-router-stmp)# exit
switch(config-router)#  neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# description Peer Router A
switch(config-router-neighbor)# copy running-config startup-config
```

Configuring BGP Peer-Policy Templates

You can configure a peer-policy template to define attributes for a particular address family. You assign a preference to each peer-policy template and these templates are inherited in the order specified, for up to five peer-policy templates in a neighbor address family.

The Cisco Nexus 1000V evaluates multiple peer policies for an address family by using the preference value. The lowest preference value is evaluated first. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Peer-policy templates can configure address family-specific attributes such as AS-path filter lists, prefix lists, route reflection, and soft reconfiguration.

Before You Begin

- You must enable BGP.
- Ensure that you are in the correct VDC.

Note When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the **default** form of the command to reset that attribute to the default state.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **template peer-policy** *template-name*	Creates a peer-policy template.
Step 4	switch(config-router-ptmp)# **advertise-active-only**	(Optional) Advertises only active routes to the peer.
Step 5	switch(config-router-ptmp)# **maximum-prefix** *number*	(Optional) Sets the maximum number of prefixes allowed from this peer.
Step 6	switch(config-router-ptmp)# **exit**	Exits peer-policy template configuration mode.
Step 7	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	switch(config-router-neighbor)# **address-family l2vpn evpn**	Enters neighbor address family configuration mode for the l2vpn evpn address family.

	Command or Action	Purpose
Step 9	switch(config-router-neighbor-af)# **inherit peer-policy** *template-name preference*	Applies a peer-policy template to the peer address family configuration and assigns the preference value for this peer policy.
Step 10	switch(config-router-neighbor-af)# **show bgp peer-policy** *template-name*	(Optional) Displays the peer-policy template.
Step 11	switch(config-router-neighbor-af)# **copy running-config startup-config**	(Optional) Saves this configuration change.

This example shows how to configure a BGP peer-policy template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BasePolicy
switch(config-router-ptmp)# maximum-prefix 20
switch(config-router-ptmp)# exit
switch(config-router)# neighbor 192.168.1.1 remote-as 65536
switch(config-router-neighbor)# address-family l2vpn evpn
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring BGP Peer Templates

You can configure BGP peer templates to combine session and policy attributes in one reusable configuration block. Peer templates can also inherit peer-session or peer-policy templates. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template. You configure only one peer template for a neighbor, but that peer template can inherit peer-session and peer-policy templates.

Peer templates support session and address family attributes, such as eBGP multihop time-to-live, maximum prefix, next-hop self, and timers.

Before You Begin

- You must enable BGP.
- Ensure that you are in the correct VDC.

Note When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the **default** form of the command to reset that attribute to the default state.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **template peer** *template-name*	Enters peer template configuration mode.
Step 4	switch(config-router-neighbor)# **inherit peer-session** *template-name*	(Optional) Inherits a peer-session template in the peer template.
Step 5	switch(config-router-neighbor-af)# **inherit peer-policy** *template-name*	(Optional) Applies a peer template to the neighbor address family configuration.
Step 6	switch(config-router-neighbor-af)# **exit**	Exits BGP neighbor address family configuration mode.
Step 7	switch(config-router-neighbor)# **timers** *keepalive hold*	(Optional) Adds the BGP timer values to the peer. These values override the timer values in the peer-session template, BaseSession.
Step 8	switch(config-router-neighbor)# **exit**	Exits BGP peer template configuration mode.
Step 9	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 10	switch(config-router-neighbor)# **inherit peer** *template-name*	Inherits the peer template.
Step 11	switch(config-router-neighbor)# **timers** *keepalive hold*	(Optional) Adds the BGP timer values to this neighbor. These values override the timer values in the peer template and the peer-session template.
Step 12	switch(config-router-neighbor-af)# **show bgp peer-template** *template-name*	(Optional) Displays the peer template.
Step 13	switch(config-router-neighbor-af)# **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 a BGP peer template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer BasePeer
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
```

```
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer BasePeer
switch(config-router-neighbor)# copy running-config startup-config
```

Route Reflector

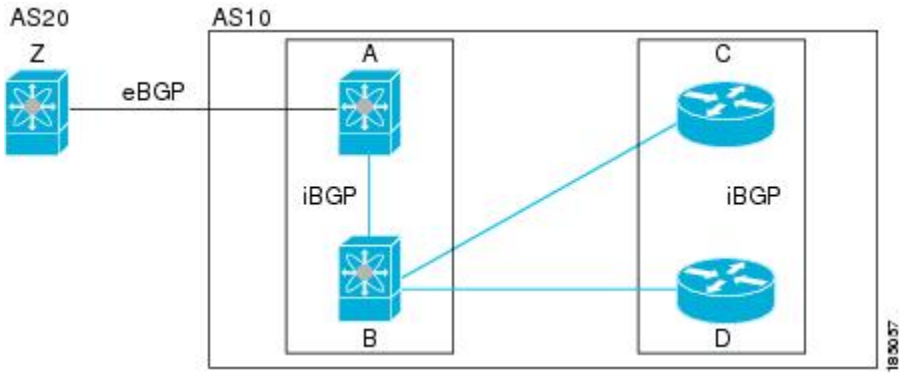
You can reduce the Internal BGP (iBGP) mesh by using a route reflector configuration where route reflectors pass learned routes to neighbors so that all iBGP peers do not need to be fully meshed.

The figure below shows a simple iBGP configuration with four meshed iBGP speakers (routers A, B, C, and D.) Without these route reflectors, when router A receives a route from an external neighbor, it advertises the route to all three iBGP neighbors.

When you configure an iBGP peer to be a route reflector, it becomes responsible for passing iBGP learned routes to a set of iBGP neighbors.

In the figure, router B is the route reflector. When the route reflector receives routes advertised from router A, it advertises (reflects) the routes to routers C and D. Router A no longer has to advertise to both routers C and D.

Figure 2: Route Reflector



The route reflector and its client peers form a cluster. You do not have to configure all iBGP peers to act as client peers of the route reflector. You must configure any nonclient peer as fully meshed to guarantee that complete BGP updates reach all peers.

Configuring Route Reflector

You can configure iBGP peers as route reflector clients to the local BGP speaker, which acts as the route reflector. Together, a route reflector and its clients form a cluster. A cluster of clients usually has a single route reflector. In such instances, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure in the network, you can configure a cluster with more than one route reflector. You must configure all route reflectors in the cluster with the same 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster.

Before You Begin

- You must enable BGP.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **cluster-id** *cluster-id*	Configures the local router as one of the route reflectors that serve the cluster. You specify a cluster ID to identify the cluster. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 4	switch(config-router)# **address-family l2vpn evpn**	Enters router address family configuration mode for the specified address family.
Step 5	switch(config-router-af)# **client-to-client reflection**	(Optional) Configures client-to-client route reflection. This feature is enabled by default. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 6	switch(config-router-neighbor)# **exit**	Exits router address configuration mode.
Step 7	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Configures the IP address and AS number for a remote BGP peer.
Step 8	switch(config-router-neighbor)# **address-family l2vpn evpn**	Enters neighbor address family configuration mode for the l2vpn evpn address family.
Step 9	switch(config-router-neighbor-af)# **route-reflector-client**	Configures the device as a BGP route reflector and configures the neighbor as its client. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 10	switch(config-router-neighbor-af)# **show bgp l2vpn evpn neighbors**	(Optional) Displays the BGP peers.
Step 11	switch(config-router-neighbor-af)# **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 the router as a route reflector and add one neighbor as a client:

```
switch(config)# configure terminal
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.10 remote-as 65536
switch(config-router-neighbor)# address-family l2vpn evpn
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# copy running-config startup-config
```

Verifying the BGP Control Plane Configuration

To display the BGP Control Plane configuration, use one of the following commands:

Command	Purpose
show bgp session	Displays the BGP sessions.
show bgp l2vpn evpn summary	Displays the BGP peers, status, and number of prefixes received from the peers.
show bgp l2vpn evpn	Displays the VTEPs that are learned through the BGP.
show bgp l2vpn evpn rd	Displays the detailed output for a specific segment ID or RD.
show bgp convergence	Displays the BGP convergence time.
show bgp l2vpn evpn evi all vtep	Displays the VTEP list for a specific VXLAN segment ID or all segments.
show bgp l2vpn evpn evi *id* **vtep** *VTEP IP address*	Displays the VTEP list for a specific VXLAN segment ID or all segments.
show bridge-domain vteps	Displays the bridge domain-to-VTEP mappings that are maintained by the VSM and are pushed to all VEMs. Remote Cisco Nexus 1000V VTEPs that are learned through BGP are designated with the Remote keyword.

This example shows how to display the BGP sessions:

```
vsm# show bgp session
Total peers 1, established peers 1
ASN 65000
VRF default, local ASN 65000
peers 1, established peers 1, local router-id 1.1.1.1
State: I-Idle, A-Active, O-Open, E-Established, C-Closing, S-Shutdown

Neighbor ASN Flaps LastUpDn|LastRead|LastWrit St Port(L/R) Notif(S/R)
14.17.199.2 65000 0 00:04:05|00:00:04|00:00:04 E 61467/179 0/0
```

This example shows how to display BGP peers, status, and number of prefixes received from the peers:

```
vsm# show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 172.23.181.67, local AS number 1
BGP table version is 10, L2VPN EVPN config peers 1, capable peers 1
4 network entries and 4 paths using 484 bytes of memory
BGP attribute entries [3/384], BGP AS path entries [0/0]
BGP community entries [0/0], BGP clusterlist entries [0/0]

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.23.181.68   4     1      19      28       10    0    0      00:13:01   1
```

.

TThis example shows how to display the VTEPs that are learned through the BGP:

```
vsm# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 10, local router ID is 172.23.181.67
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 172.23.181.67:5000     (EVI 5000)      # RD = <Router-id>:<segment-id>
*>l[3]:[5000]:[4]:[192.168.69.3]/88     #Local VTEP 192.168.69.3
                      0.0.0.0                            100      32768 i
*>i[3]:[5000]:[4]:[192.168.69.104]/88     #VTEP 192.168.69.104 that are learned from peer
172.23.181.68
                      172.23.181.68                      100          0 i
```

This example shows how to display the detailed output for a specific segment ID or RD:

```
vsm# show bgp l2vpn evpn rd 172.23.181.67:5000
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 10, local router ID is 172.23.181.67
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 172.23.181.67:5000     (EVI 5000)
BGP routing table entry for [3]:[5000]:[4]:[192.168.69.3]/88, version 4
Paths: (1 available, best #1)
Flags: (0x00000a) on xmit-list, is not in l2rib/evpn

  Path type: local, path is valid, is best path
  AS-Path: NONE, path locally originated
    0.0.0.0 (metric 0) from 0.0.0.0 (0.0.0.0)
      Origin IGP, MED not set, localpref 100, weight 32768
      Extcommunity:
          RT:1:5000

  Advertised to peers:
    172.23.181.68
BGP routing table entry for [3]:[5000]:[4]:[192.168.69.104]/88, version 10
Paths: (1 available, best #1)
Flags: (0x00001a) on xmit-list, is in l2rib/evpn

  Path type: internal, path is valid, is best path
             Imported from 172.23.181.68:5000:[3]:[5000]:[4]:[192.168.69.104]/88
  AS-Path: NONE, path sourced internal to AS
    172.23.181.68 (metric 0) from 172.23.181.68 (172.23.181.68)
      Origin IGP, MED not set, localpref 100, weight 0
      Extcommunity:
          RT:1:5000

  Not advertised to any peer
```

This example shows how to display the BGP convergence time:

```
vsm# show bgp convergence
Global settings:
BGP start time 01:46:06
Config processing completed 0.280000 after start
BGP out of wait mode 0.280000 after start

Information for VRF default
Initial-bestpath timeout: 300 sec, configured 0 sec
First peer up 00:00:24 after start
Bestpath timer not running

   IPv4 Unicast:
   First bestpath signalled 00:00:06 after start
   First bestpath completed 00:00:06 after start
```

```
   L2VPN EVPN:
   First bestpath signalled 00:00:06 after start
   First bestpath completed 00:00:06 after start #First Convergence event.
```

This example shows how to display the VTEP list for a specific VXLAN segment id or all segments:

```
vsm# show bgp l2vpn evpn evi all vtep
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 17, local router ID is 192.168.66.10
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 192.168.66.10:5000     (EVI 5000)
*>i66.100.0.1         192.168.66.100                   100          0 i
*>l192.168.69.101     0.0.0.0                          100      32768 i
*>i192.168.69.201     192.168.67.10                    100          0 i
```

This example shows how to display the VTEP list for a specific VXLAN segment id or all segments:

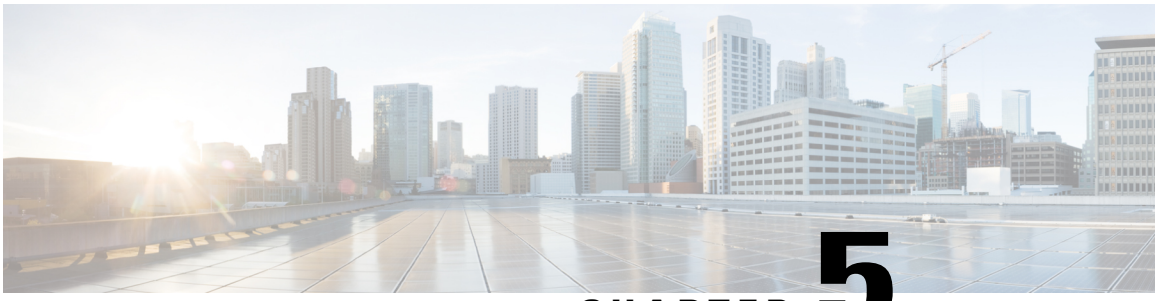
```
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.168.66.10:5000     (EVI 5000)
BGP routing table entry for [3]:[5000]:[4]:[192.168.69.101]/88, version 2
Paths: (1 available, best #1)
Flags: (0x00000a) on xmit-list, is not in l2rib/evpn

  Path type: local, path is valid, is best path
  AS-Path: NONE, path locally originated
    0.0.0.0 (metric 0) from 0.0.0.0 (0.0.0.0)
      Origin IGP, MED not set, localpref 100, weight 32768
      Extcommunity:
          RT:1:5000

  Advertised to peers:
    192.168.66.100     192.168.67.10
```

This example shows how to display the bridge domain-to-VTEP mappings that are maintained by the VSM and are pushed to all VEMs:

```
vsm# show bridge-domain vteps

D: Designated VTEP     I:Forwarding Publish Incapable VTEP

Note: (*) Denotes active gateway module

Bridge-domain: vxlan-5000
VTEP Table Version: 13
Port         Module   VTEP-IP Address   VTEP-Flags
-------------------------------------------------------------------------------
Veth5        3        192.168.69.101    (D)
Remote       -        66.100.0.1        (DI)
Remote       -        192.168.69.201    (DI)
```

Feature History for BGP Control Plane

Feature Name	Releases	Feature Information
BGP Control Plane	5.2(1)SV3(1.1)	BGP Control Plane was introduced.



CHAPTER 5

Configuring Advanced Features

This chapter contains the following sections:

Information about VXLAN Advanced Features

You can use VXLAN trunk is used to configure ports that can individually map to an IEEE 802.1Q tag value to a VXLAN. Ports that receive the packets with the 802.1Q tag are used to map the packet into a VXLAN. You can use the VXLAN trunk feature to support network service Virtual Machines (VMs) that need access to more VXLANs than the number of NICs that VMware supports on a VM.

Configuring VXLAN Trunks

You can trunk multiple VXLANs on a single vEthernet interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-profile type vethernet** *name*
3. (Optional) switch(config-port-prof)# **switchport mode access**
4. switch(config -port-profile)# **switchport access bridge-domain** *name-string*
5. switch(config -port-profile)# [**no**] **service instance** *place holder*
6. switch(config -port-profile)# **encapsulation profile***encapsulation-profile-name*
7. (Optional) switch(config-port-prof)# **no shutdown**
8. (Optional) switch(config-port-prof)# **state enabled**
9. (Optional) switch(config-port-prof)# **end**
10. (Optional) switch# **show run port-profile csr-access**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile type vethernet** *name*	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: • **name—**The port profile name can be an alphanumeric name up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V. • **type—**(Optional) The port profile type can be Ethernet or vEthernet.
Step 3	switch(config-port-prof)# **switchport mode access**	(Optional) Designates that the interfaces are to be used as a trunking ports. A trunk port transmits untagged packets for the native VLAN and transmits encapsulated, tagged packets for all other VLANs.
Step 4	switch(config -port-profile)# **switchport access bridge-domain** *name-string*	Assigns a VXLAN bridge domain to this profile for non-dot1q traffic.
Step 5	switch(config -port-profile)# [**no**] **service instance** *place holder*	Defines a place holder for mappings. The range is from 1 to 4096.
Step 6	switch(config -port-profile)# **encapsulation profile***encapsulation-profile-name*	Reference to an encapsulation profile comprising mapping between VLAN and VXLAN segments. **Note** **[no]encapsulation dot1q** *mappings* **bridge-domain** *name* command is no longer supported.
Step 7	switch(config-port-prof)# **no shutdown**	(Optional) Administratively enables all ports in the profile.

	Command or Action	Purpose
Step 8	switch(config-port-prof)# **state enabled**	(Optional) Enables the port profile and applies its configuration to the assigned ports.
Step 9	switch(config-port-prof)# **end**	(Optional) Exits the global configuration mode.
Step 10	switch# **show run port-profile csr-access**	(Optional) Displays the configuration of the port profile.

This example shows how to configure a VXLAN trunk:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# port-profile type vethernet csr-access
switch(config-port-prof)# switchport mode access
switch(config-port-prof)# switchport access bridge-domain bd-701
switch(config-port-prof)# service instance 10
switch(config-port-prof-srv)# encapsulation dot1q 600 bridge-domain bd-600
switch(config-port-prof-srv)# encapsulation dot1q 601 bridge-domain bd-601
switch(config-port-prof-srv)# encapsulation dot1q 602 bridge-domain bd-602
switch(config-port-prof-srv)# no shutdown
cswitch(config-port-prof-srv)# state enabled
switch(config-port-prof)# end
switch# show run port-profile csr-access
port-profile type vethernet csr-access
  switchport mode access
  switchport access bridge-domain bd-701
  service instance 10
    encapsulation dot1q 600 bridge-domain bd-600
    encapsulation dot1q 601 bridge-domain bd-601
    encapsulation dot1q 602 bridge-domain bd-602
  state enabled
```

This example shows how to display VXLAN trunk interface mappings:

```
switch(config-bd)# show bridge-domain mappings

--------------------------------------------------------------------------------
Interface      Module    Serv Inst   Vlan     Segment           BD-Name
--------------------------------------------------------------------------------
Vethernet923    8           1        1561      6061           vxlan6061
                8           2        1562      6062           vxlan6062
                8           3        1563      6063           vxlan6063
                8           4        1564      6064           vxlan6064

switch(config-bd)#
```

Note The value in the Interface column varies based on the VXLAN gateway or the VXLAN trunk feature. Vethernet<number> in the Interface column indicates the mapping for the VXLAN trunk feature; port-channel<number> in the Interface column indicates the mapping that configured on the VXLAN gateway.

Configuring Multi-MAC Capability

You can mark a virtual Ethernet interface as capable of sourcing packets from multiple MAC addresses.

Before You Begin

Do not use the multi-MAC capability feature if MAC distribution (segment distribution MAC) is enabled.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-profile type vethernet** *name*
3. (Optional) switch(config-port-prof)# **switchport mode access**
4. switch(config-port-prof)# **switchport access bridge-domain** *name-string*
5. switch(config-port-prof)# [**no**] **capability multi- mac**
6. (Optional) switch(config-port-prof)# **no shutdown**
7. (Optional) switch(config-port-prof)# **state enabled**
8. (Optional) switch(config-port-prof)# **end**
9. (Optional) switch# **show running-config port-profile csr-multi-mac-access**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile type vethernet** *name*	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: • **name—**The port profile name can be an alphanumeric name up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V. • **type—**(Optional) The port profile type can be Ethernet or vEthernet.
Step 3	switch(config-port-prof)# **switchport mode access**	(Optional) Designates that the interfaces are to be used as a trunking ports. A trunk port transmits untagged packets for the native VLAN and transmits encapsulated, tagged packets for all other VLANs.
Step 4	switch(config-port-prof)# **switchport access bridge-domain** *name-string*	Assigns a VXLAN bridge domain to this profile for non-dot1q traffic.
Step 5	switch(config-port-prof)# [**no**] **capability multi- mac**	Marks the vEthernet port as multi-MAC capable.

	Command or Action	Purpose
Step 6	switch(config-port-prof)# **no shutdown**	(Optional) Administratively enables all ports in the profile.
Step 7	switch(config-port-prof)# **state enabled**	(Optional) Enables the port profile and applies its configuration to the assigned ports.
Step 8	switch(config-port-prof)# **end**	(Optional) Exits the global configuration mode.
Step 9	switch# **show running-config port-profile csr-multi-mac-access**	(Optional) Displays the configuration of the port profile.

This example shows how to configure the multi-MAC capability feature:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config-port-prof)# port-profile type vethernet csr-multi-mac-access
switch(config-port-prof)# switchport mode access
switch(config-port-prof)# switchport access bridge-domain bd-701
switch(config-port-prof)# capability multi-mac
switch(config-port-prof)# state enabled
switch(config-port-prof)# no shutdown
switch(config-port-prof)# end
switch# show running-config port-profile csr-multi-mac-access
port-profile type vethernet csr-multi-mac-access
  switchport mode access
  switchport access bridge-domain bd-701
  capability multi-mac
  no shutdown
  state enabled
```

Configuring Encapsulation Profiles

SUMMARY STEPS

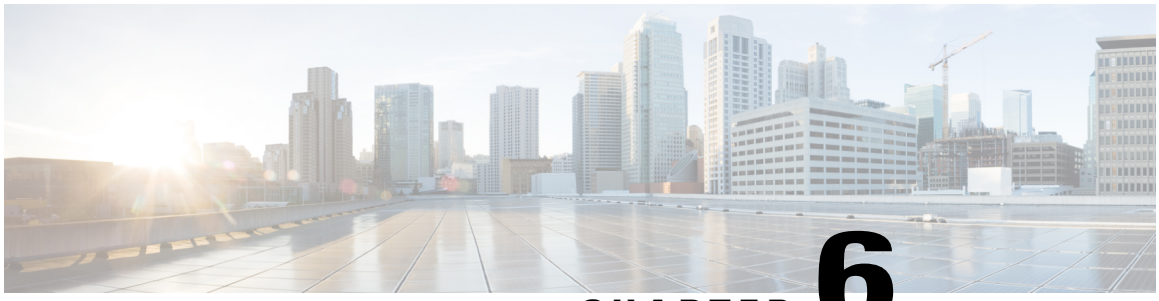
1. switch(config)# **encapsulation profile segment** *encapsulation-profile-name*
2. switch(config)# **dot1q vlan bridge-domain vxlan**
3. switch(config)# **show encapsulation profile**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config)# **encapsulation profile segment** *encapsulation-profile-name*	Creates an encapsulation profile for configure mappings.

	Command or Action	Purpose
Step 2	switch(config)# **dot1q vlan bridge-domain vxlan**	Creates a VLAN-VXLAN mapping. Ensure corresponding VLAN and VXLAN BD is created for mapping to be pushed to VEM data-path.
Step 3	switch(config)# **show encapsulation profile**	Displays all the encapsulation profiles created.

```
switch(config)# encapsulation profile segment test
switch(config-vxlan-encap-prof)# dot1q 1560 bridge-domain vxlan6000
switch(config)# show encapsulation profile
-------------------------------------------------------------------------------
Vlan Bridge-domain
-------------------------------------------------------------------------------
1601 vxlan6101
1602 vxlan6102
1603 vxlan6103
1604 vxlan6104
```



CHAPTER 6

Installing and Configuring VXLAN Gateway

This chapter contains the following sections:

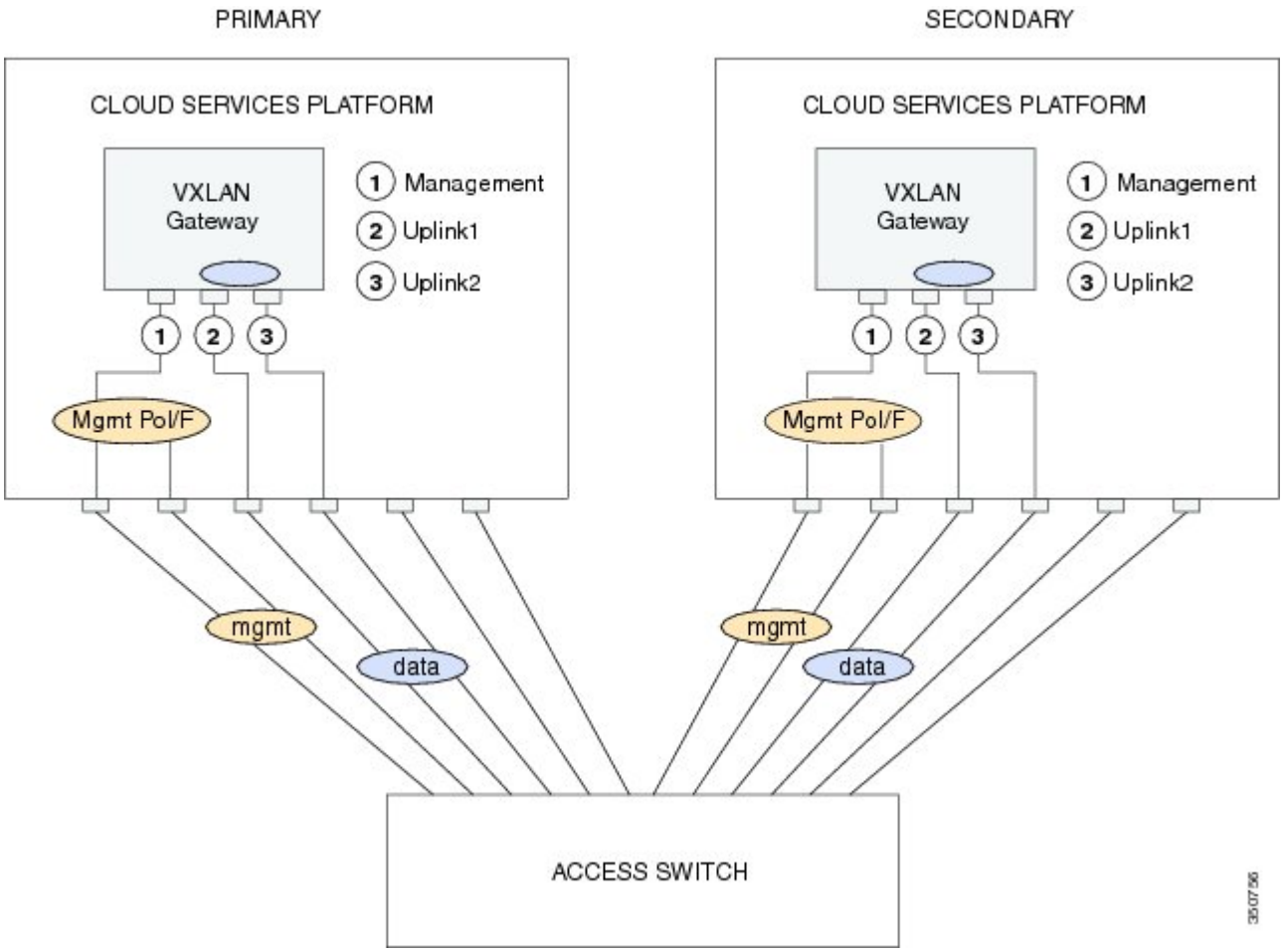
Information About the VXLAN Gateway Deployment

The VXLAN gateway has the following deployment requirements:

- To configure the VXLAN gateway, you must install the Advanced Edition license on the Cisco Nexus 1000V switch.
- You can deploy the VXLAN gateway as a VM or on the Cisco Nexus Cloud Services Platform Release 4.2(1)SP1(6.1) or later releases.
- You must connect the Cloud Services Platform appliance to a switch that supports the Link Aggregation Control Protocol (LACP) based or statically configured port channels and VLAN-based trunk interfaces.
- vCPU and Memory requirements:
  - Three vCPUs for each Virtual Service Blade (VSB)
  - Two vCPUs for each Virtual Machine (VM)
  - 2-GB RAM
  - 3-GB disk space

This figure shows the VXLAN gateway deployment.

Figure 3: VXLAN Gateway Deployment



Guidelines and Limitations

VXLAN gateways have the following configuration guidelines and limitations:

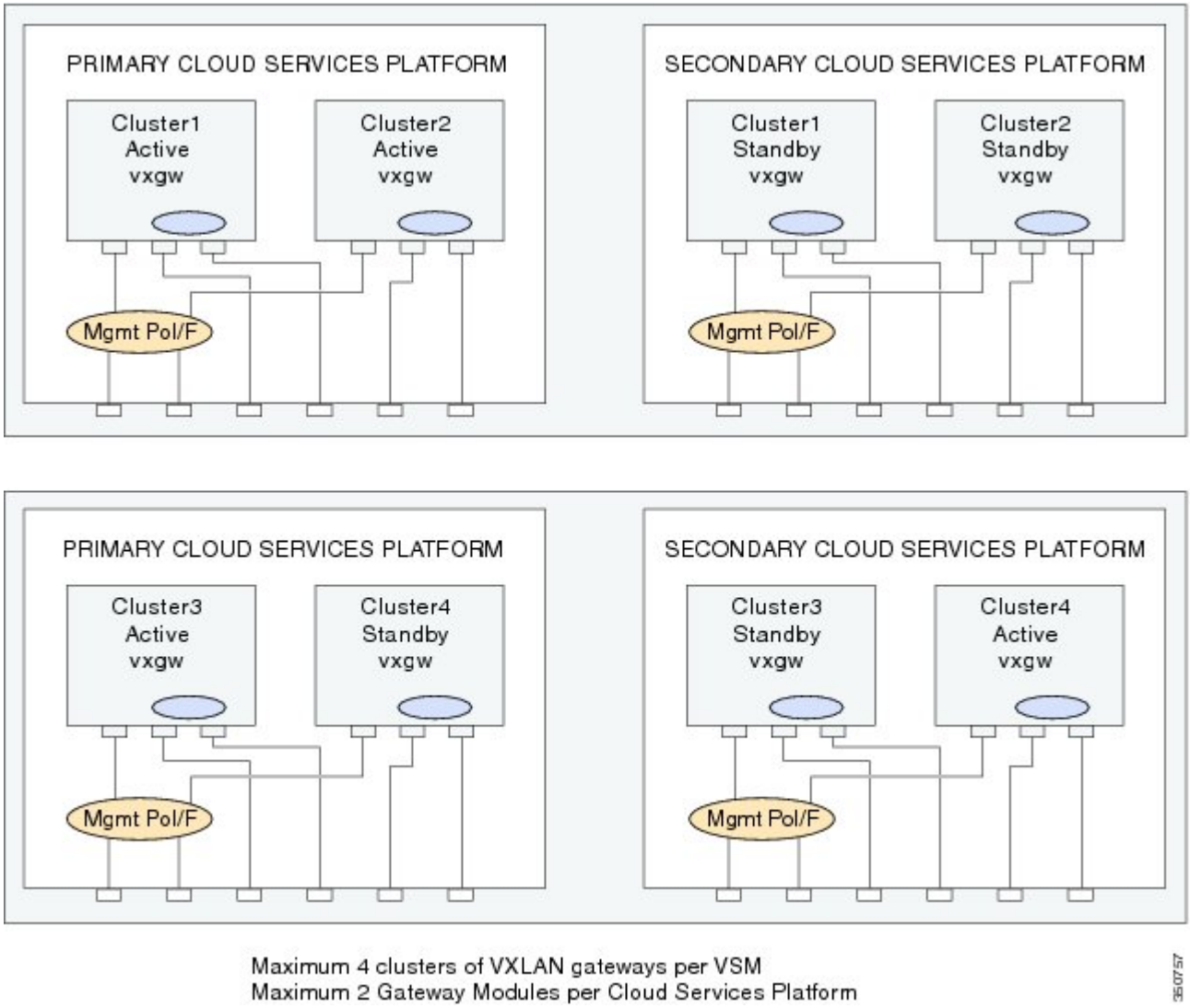
- You must configure the VSM to use the Layer 3 control. We strongly recommend that the VSM Layer 3 control is through mgmt 0. For more information about Layer 3 control, see the *Cisco Nexus 1000V Installation and Upgrade Guide* and *Cisco Nexus 1000V System Management Configuration Guide*.
- You must configure the uplink for the gateway module as a LACP or a static port channel. The VXLAN gateway does not function if gateways are configured in the MAC-pinning mode.
- A single VSM can manage a maximum of eight VXLAN gateway high availability (HA) clusters.
- You must configure the HA mode of the VXLAN gateway as standalone or primary/secondary so that when you bring up the VXLAN gateway, the HA state is either active or standby and the VXLAN-to-VLAN mappings are either active or pending. If you do not configure an HA role for the

VXLAN gateway, when you bring up the VXLAN gateway, the HA state is unknown and VXLAN-to-VLAN traffic is not processed.

- You must configure the underlying Cloud Services Platform with an uplink type that is flexible (type 5). VXLAN gateways use two physical interfaces. You must set the interfaces in the passthrough mode. In addition, you must set at least one physical or a port channel interface must be set up to carry management traffic.
- Ensure that you do not configure a private VLAN (PVLAN) on the VLANs used for VXLAN-VLAN mappings.
- Ensure that the VXGW VTEP VLAN is not also used as a VXLAN segment.
- The VXLAN gateway virtual services blade (VSB) has two uplink data interfaces configured in a port channel group. The VXLAN gateway VM has only one uplink data interface; therefore, it does not require a port channel.
- The VXLAN gateway VSB uses three vCPUs: one vCPU for management traffic and the other two vCPUs for the data interfaces. The VXALN gateway VM uses two vCPUs; one vCPU for management traffic and one vCPU for the data interface.

The following figure shows four Cloud Services Platform devices where each Cloud Services Platform device hosts two VXLAN gateway modules. Four HA clusters of gateway modules are set up with each cluster that consists of an active/standby pair of modules.

Figure 4: VXLAN Gateway HA Pairs



Enabling VXLAN Gateway

You must enable the VXLAN gateway before you can configure it.

Before You Begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature vxlan-gateway**	Enables the VXLAN gateway. Use the **no feature vxlan-gateway** command to disable the VXLAN gateway and remove all associated configuration.
Step 3	switch(config)# **show feature**	(Optional) (Optional) Displays enabled and disabled features.
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.

Configuring Port Profiles on the VSM

You must create port profiles on the VSM before you can install and configure the VXLAN gateway.

Creating Port Profiles for VXLAN gateway as VSB

Configuring a Port Profile for the Uplink on the VXLAN Gateway

Before installing the VXLAN gateway, you must create two port profiles on the switch (VSM), one for the uplinks on the gateway and one for the VXLAN Tunnel Endpoint (VTEP) interface.

Before You Begin

- Ensure that VSM is configured in the Advanced mode by entering the **svs switch edition advanced** configuration command to enable Advanced mode.
- Ensure that LACP is configured by entering the **feature lacp** configuration command on the VSM.
- Offload the LACP operation by entering the **lacp offload** configuration command on the VSM.
- Ensure that VXLAN is enabled on the VSM by entering the **feature segmentation** configuration command to enable VXLANs on the VSM.
- Ensure that VXLAN gateway is enabled on the VSM by entering the **feature vxlan-gateway** configuration command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm# **configure terminal**	Enters global configuration mode.
Step 2	vsm# **encapsulation profile segment** *name*	Creates an encapsulation profile to contain the VLAN-to-VXLAN mappings.
Step 3	vsm(config-vxlan-encap-prof)# **dot1q** *VLAN-ID* **bridge-domain** *bd-name*	Maps a VLAN to a VXLAN. The VXLAN is specified through the bridge-domain name. **Note** The bridge-domain name and VLAN-ID that you provide are not created during the port-profile configuration. The bridge-domain name and the VLAN ID that you provide should be in an active state or the mapping is held in an inactive state until you create the bridge-domain name and VLAN ID. **Note** Repeat this step to specify additional mappings.
Step 4	vsm(config)# **port profile type ethernet** *name*	Creates a port profile of type ethernet for the VXLAN gateway uplink. **Note** You must provide a port-profile name when prompted while executing the setup script to configure the VXLAN gateway.
Step 5	vsm(config-port-prof)# **switchport mode trunk**	Designates that the interfaces are to be used as trunking ports. A trunk port transmits untagged packets for the native VLAN and transmits encapsulated, tagged packets for all other VLANs.
Step 6	vsm(config-port-prof)# **switchport trunk allowed vlan** *vlan list*	Specifies the list of VLANs allowed on the gateway's uplink. This list should consist of all the mapped VLANs and the VLAN for the VTEP virtual interface.
Step 7	vsm(config-port-prof)# **mtu** *mtu size in bytes*	Designates the MTU size. For VXLAN traffic to be functional, you must set the MTU size as 1550. If you do not set the MTU size, the default of 1500 is used. The size must be an even number between 1500 and 9000. The MTU configured on an interface takes precedence over the MTU configured on a port profile.
Step 8	vsm(config-port-prof)# **service instance** *place holder*	(Optional) Defines a place holder for mappings. The range is from 1 to 4096. **Note** You do not need to execute the service instance and the encapsulation command at this stage to bring up the gateway. These commands are optional and you can add the mappings later once the port profiles are configured.
Step 9	vsm(config-port-prof-svc)# **encapsulation profile** *name*	Specifies the encapsulation profile for the port profile.
Step 10	vsm(config-port-prof-srv)# **exit**	(Optional) Exits from the service instance mode.
Step 11	vsm(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.

	Command or Action	Purpose
Step 12	vsm(config-port-prof)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 13	vsm(config-port-prof)# **vmware port-group**	Distributes the port profile. Recommends that this port profile should not be inherited on non VXLAN gateway ports.

This example shows how to configure and display the gateway mappings:

```
vsm# configure terminal
vsm(config)# port-profile type ethernet gw-uplink
vsm(config)# switchport mode trunk
vsm(config)# switchport trunk allowed vlan 1545
vsm(config)# mtu 1550
vsm(config-port-prof)# service instance 1
vsm(config-port-prof-srv)# encapsulation profile segment gw-segment
vsm(config-port-prof-srv)# exit
vsm(config-port-prof)# no shutdown
vsm(config-port-prof)# state enabled
vsm(config-port-prof)# vmware port-group
```

Configuring a Port Profile for the VTEP on the VXLAN Gateway

You can create a port profile that can be applied to the VTEP virtual interface on the VXLAN gateway.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm# **configure terminal**	Enters global configuration mode.
Step 2	vsm(config) # **port-profile type vethernet** *port-profile name*	Configures a port profile for the VTEP on the VXLAN gateway. **Note** You must provide a port profile name when prompted while executing the setup script to configure the VXLAN gateway.
Step 3	vsm(config-port-prof) # **switchport mode access**	Designates that the interfaces are to be used as a trunking ports. A trunk port transmits untagged packets VLAN and transmits encapsulated, tagged packets for all other VLANs.
Step 4	vsm(config-port-prof) # **switchport access vlan** *vlan-id-access*	Assigns an access VLAN ID to this port profile. The VLAN ID provided must be added to the allowed VLAN set of the uplink port profile. This VLAN should not be mapped to any VXLAN. **Note** If you do not specify a VLAN ID, VLAN 1 is used automatically.
Step 5	vsm(config-port-prof) # **capability vxlan**	Configures the capability VXLAN feature on the specified virtual Ethernet port and enables encapsulation and decapsulation of VXLAN packets.

	Command or Action	Purpose
Step 6	vsm(config-port-prof) # **transport ip address** *ip-address* **netmask** *network mask* [**gateway** *ip-address*]	Configures the IP address, netmask, and gateway for the VTEP. **Note** If you have VTEPs that are in different subnets, you must specify the gateway IP address. If a gateway is not provided, the VXLAN gateway uses ARP to reach the remote VTEP.
Step 7	vsm(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.
Step 8	vsm(config-port-prof)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 9	vsm(config-port-prof)# **vmware port-group**	Distributes the port profile. Recommends that this port profile should not be inherited on non VXLAN gateway ports.

This example displays how to configure a VTEP on the VXLAN gateway:

```
vsm# configure terminal
vsm(config)# port-profile type vethernet gw-vtep
vsm(config-port-prof)# switchport mode access
vsm(config-port-prof)# switchport access vlan 760
vsm(config-port-prof)# capability vxlan
vsm(config-port-prof)# transport ip address 192.0.2.1 netmask 255.255.255.0 gateway
192.0.2.254
vsm(config-port-prof)# no shutdown
vsm(config-port-prof)# state enabled
vsm(config-port-prof)# vmware port-group
```

Creating Port Profiles for VXLAN Gateway as VM in VMWare vCenter

Configuring a Port Profile for the Uplink on the VXLAN Gateway

Before installing the VXLAN gateway, you must create two port profiles on the switch (VSM), one for the uplinks on the gateway and one for the VXLAN Tunnel Endpoint (VTEP) interface.

Before You Begin

- Ensure that VSM is configured in the Advanced mode by entering the **svs switch edition advanced** configuration command to enable Advanced mode.
- Ensure that LACP is configured by entering the **feature lacp** configuration command on the VSM.
- Offload the LACP operation by entering the **lacp offload** configuration command on the VSM.
- Ensure that VXLAN is enabled on the VSM by entering the **feature segmentation** configuration command to enable VXLANs on the VSM.
- Ensure that VXLAN gateway is enabled on the VSM by entering the **feature vxlan-gateway** configuration command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm# **configure terminal**	Enters global configuration mode.
Step 2	vsm# **encapsulation profile segment** *name*	Creates an encapsulation profile to contain the VLAN-to-VXLAN mappings.
Step 3	vsm(config-vxlan-encap-prof)# **dot1q** *VLAN-ID* **bridge-domain** *bd-name*	Maps a VLAN to a VXLAN. The VXLAN is specified through the bridge-domain name. **Note** The bridge-domain name and VLAN-ID that you provide are not created during the port-profile configuration. The bridge-domain name and the VLAN ID that you provide should be in an active state or the mapping is held in an inactive state until you create the bridge-domain name and VLAN ID. **Note** Repeat this step to specify additional mappings.
Step 4	vsm(config)# **port profile type ethernet** *name*	Creates a port profile of type ethernet for the VXLAN gateway uplink. **Note** You must provide a port-profile name when prompted while executing the setup script to configure the VXLAN gateway.
Step 5	vsm(config-port-prof)# **switchport mode trunk**	Designates that the interfaces are to be used as trunking ports. A trunk port transmits untagged packets for the native VLAN and transmits encapsulated, tagged packets for all other VLANs.
Step 6	vsm(config-port-prof)# **switchport trunk allowed vlan** *vlan list*	Specifies the list of VLANs allowed on the gateway's uplink. This list should consist of all the mapped VLANs and the VLAN for the VTEP virtual interface.
Step 7	vsm(config-port-prof)# **mtu** *mtu size in bytes*	Designates the MTU size. For VXLAN traffic to be functional, you must set the MTU size as 1550. If you do not set the MTU size, the default of 1500 is used. The size must be an even number between 1500 and 9000. The MTU configured on an interface takes precedence over the MTU configured on a port profile.
Step 8	vsm(config-port-prof)# **service instance** *place holder*	(Optional) Defines a place holder for mappings. The range is from 1 to 4096. **Note** You do not need to execute the service instance and the encapsulation command at this stage to bring up the gateway. These commands are optional and you can add the mappings later once the port profiles are configured.
Step 9	vsm(config-port-prof-svc)# **encapsulation profile** *name*	Specifies the encapsulation profile for the port profile.
Step 10	vsm(config-port-prof-srv)# **exit**	(Optional) Exits from the service instance mode.
Step 11	vsm(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.

	Command or Action	Purpose
Step 12	vsm(config-port-prof)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 13	vsm(config-port-prof)# **vmware port-group**	Distributes the port profile. Recommends that this port profile should not be inherited on non VXLAN gateway ports.

This example shows how to configure and display the gateway mappings:

```
vsm# configure terminal
vsm(config)# port-profile type ethernet gw-uplink
vsm(config)# switchport mode trunk
vsm(config)# switchport trunk allowed vlan 1545
vsm(config)# mtu 1550
vsm(config-port-prof)# service instance 1
vsm(config-port-prof-srv)# encapsulation profile segment gw-segment
vsm(config-port-prof-srv)# exit
vsm(config-port-prof)# no shutdown
vsm(config-port-prof)# state enabled
vsm(config-port-prof)# vmware port-group
```

Configuring a Port Profile for the VTEP on the VXLAN Gateway

You can create a port profile that can be applied to the VTEP virtual interface on the VXLAN gateway.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm# **configure terminal**	Enters global configuration mode.
Step 2	vsm(config) # **port-profile type vethernet** *port-profile name*	Configures a port profile for the VTEP on the VXLAN gateway. **Note** You must provide a port profile name when prompted while executing the setup script to configure the VXLAN gateway.
Step 3	vsm(config-port-prof) # **switchport mode access**	Designates that the interfaces are to be used as a trunking ports. A trunk port transmits untagged packets VLAN and transmits encapsulated, tagged packets for all other VLANs.
Step 4	vsm(config-port-prof) # **switchport access vlan** *vlan-id-access*	Assigns an access VLAN ID to this port profile. The VLAN ID provided must be added to the allowed VLAN set of the uplink port profile. This VLAN should not be mapped to any VXLAN. **Note** If you do not specify a VLAN ID, VLAN 1 is used automatically.
Step 5	vsm(config-port-prof) # **capability vxlan**	Configures the capability VXLAN feature on the specified virtual Ethernet port and enables encapsulation and decapsulation of VXLAN packets.

	Command or Action	Purpose
Step 6	vsm(config-port-prof) # **transport ip address** *ip-address* **netmask** *network mask* [**gateway** *ip-address*]	Configures the IP address, netmask, and gateway for the VTEP. **Note** If you have VTEPs that are in different subnets, you must specify the gateway IP address. If a gateway is not provided, the VXLAN gateway uses ARP to reach the remote VTEP.
Step 7	vsm(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.
Step 8	vsm(config-port-prof)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 9	vsm(config-port-prof)# **vmware port-group**	Distributes the port profile. Recommends that this port profile should not be inherited on non VXLAN gateway ports.

This example displays how to configure a VTEP on the VXLAN gateway:

```
vsm# configure terminal
vsm(config)# port-profile type vethernet gw-vtep
vsm(config-port-prof)# switchport mode access
vsm(config-port-prof)# switchport access vlan 760
vsm(config-port-prof)# capability vxlan
vsm(config-port-prof)# transport ip address 192.0.2.1 netmask 255.255.255.0 gateway
192.0.2.254
vsm(config-port-prof)# no shutdown
vsm(config-port-prof)# state enabled
vsm(config-port-prof)# vmware port-group
```

Configuring a vEthernet Trunk Port Profile for VXLAN Gateway Uplink Port

You can create a vEthernet trunk port profile for VXLAN gateway on vCenter. It is used by vCenter to send data to the VXLAN gateway.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm# **configure terminal**	Enters global configuration mode.
Step 2	vsm(config)# **port-profile type vethernet** *port-profile name*	Configures a port profile for the VXLAN gateway on the VSM.
Step 3	vsm(config-port-prof)# **switchport mode trunk**	Designates that the interfaces are to be used as trunking ports. A trunk port transmits untagged packets for the native VXLAN and transmits encapsulated, tagged packets for all other VXLANs.
Step 4	vsm(config-port-prof)# **switchport trunk allowed vlan** *vlan-id-access*	Assigns an access VLAN ID to this port profile. The VLAN ID provided must be added to the allowed VLAN set of the uplink port profile. This VLAN should not be mapped to any VXLAN. If you do not specify a VLAN ID, VLAN 1 is used automatically.
Step 5	vsm(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.

	Command or Action	Purpose
Step 6	vsm(config-port-prof)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 7	vsm(config-port-prof)# **vmware port-group**	Designates the port profile as a VMware port group.

This example shows how to configure a vEthernet trunk port profile:

```
vsm# configure terminal
vsm(config)# port-profile type vethernet gw-trunk1
vsm(config-port-prof)# switchport mode trunk
vsm(config-port-prof)# switchport trunk allowed vlan 105-109
vsm(config-port-prof)# no shutdown
vsm(config-port-prof)# state enabled
vsm(config-port-prof)# vmware port-group
vsm(config-port-prof)# end
```

Configuring a vEthernet Access Port Profile for VXLAN Gateway Management Port

You can create a vEthernet management access port profile for the VXLAN gateway on vCenter. It is used for vCenter to allow management data to the VXLAN gateway.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm# **configure terminal**	Enters global configuration mode.
Step 2	vsm(config-port-prof)# **port-profile type vethernet** *port-profile name*	Configures a port profile for the VXLAN gateway on the VSM.
Step 3	vsm(config-port-prof)# **switchport mode access**	Designates that the interfaces are to be used as a access ports.
Step 4	vsm(config-port-prof)# **switchport access vlan** *vlan-id*	Assigns an access VLAN ID to this port profile.
Step 5	vsm(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.
Step 6	vsm(config-port-prof)# **system vlan** *vlan-id*	Assigns an access VLAN ID to this port profile.
Step 7	vsm(config-port-prof)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 8	vsm(config-port-prof)# **vmware port-group**	Designates the port profile as a VMware port group.

This example shows how to configure a vEthernet access port profile:

```
vsm# configure terminal
vsm(config)# port-profile type vethernet gwmgmt
vsm(config-port-prof)# switchport mode access
vsm(config-port-prof)# switchport access vlan 233
vsm(config-port-prof)# no shutdown
```

```
vsm(config-port-prof)# system vlan 233
vsm(config-port-prof)# state enabled
vsm(config-port-prof)# vmware port-group
vsm(config-port-prof)# end
```

Installing VXLAN Gateway

Installing the VXLAN Gateway on a Virtual Service Blade

Creating and Deploying a VXLAN Gateway

You can create and deploy a VXLAN gateway as a VSB on all Cisco Nexus Cloud Services Platforms.

Before You Begin

You must be logged in to the Cisco Nexus Cloud Services Platform on which you want to install the VXLAN gateway.

DETAILED STEPS

	Command or Action	Purpose
Step 1	CSP# **copy scp://** *server where the VXGW image is located source path iso image of vxlan gw bootflash:repository >*	Copies the VXLAN gateway image to the bootflash/repository on the CCPA Manager.
Step 2	CSP# **configure terminal**	Enters global configuration mode.
Step 3	CSP(config) # **virtual-service-blade** *name of the VXLAN GW VSB*	Creates a VXLAN gateway VSB.
Step 4	CSP(config-vsb-config) # **virtual-service-blade-type new** *iso image of the vxlan gw*	Deploys the downloaded VXLAN gateway. The image is always populated from the bootflash or repository and there is no need to specify the path.
Step 5	CSP(config-vsb-config) # **interface gw-uplink1 uplink** *Physical-Interface Cloud Services Platform*	Assigns a physical interface on the Cloud Services Platform to the gateway uplink. GigabitEthernet3 through GigabitEthernet6 are available in the flexible mode physical interfaces. You must configure the port channels using LACP on the upstream switches.
Step 6	CSP(config-vsb-config) # **interface gw-uplink1 mode passthrough**	Configures the gateway uplink as passthrough. The corresponding Gigabit Ethernet interface cannot be shared with other VSBs on the Cloud Services Platform.
Step 7	CSP(config-vsb-config) # **interface gw-uplink2 uplink** *Physical-Interface Cloud Services Platform*	Assigns a physical interface on the Cloud Services Platform to the gateway uplink. GigabitEthernet3 through GigabitEthernet6 are available in the flexible mode physical interfaces. You must configure the port channels using LACP on the upstream switches.

	Command or Action	Purpose
Step 8	CSP(config-vsb-config) # **interface gw-uplink2 mode passthrough**	Configures the gateway uplink as passthrough. The corresponding Gigabit Ethernet interface cannot be shared with other VSBs on the Cloud Services Platform.
Step 9	CSP(config-vsb-config) # **interface management vlan** *vlan id*	Allows the specified VLAN ID on the management uplink. The VLAN range is from 1 to 4096.
Step 10	CSP(config-vsb-config) # **interface management uplink** *interface*	Specifies the interface as either a physical interface of the Cloud Services Platform or a port channel interface previously created on the Cloud Services Platform.
Step 11	Use one of the following commands to deploy a gateway: • CSP(config-vsb-config)# **enable** • CSP(config-vsb-config)# **enable primary** • CSP(config-vsb-config)# **enable secondary**	Use the **enable** command to install two VSBs, one on the primary Cloud Services Platform and another on the secondary cloud services platform. Use the **enable primary** command to deploy the gateway in standalone mode on the primary Cloud Services Platform. Use the **enable secondary** command to deploy the gateway in standalone mode on the secondary Cloud Services Platform. Initiates a setup script to configure the VXLAN gateway, IP address, subnet mask, gateway, hostname, and password for the VXLAN gateway VSB. You are also required to specify the details of the VSM's domain ID, IP address, and primary and secondary MAC addresses on the control interface.

This example shows how to bring up a gateway as a VSB on a VSA pair:

```
CSP# configure terminal
CSP(config)# virtual-service-blade VXLAN-GW
CSP(config-vsb-config)# virtual-service-blade-type new vxgw.4.2.1.SV2.2.0.264.iso
CSP(config-vsb-config)# interface gw-uplink1 uplink GigabitEthernet3
CSP(config-vsb-config)# interface gw-uplink2 uplink GigabitEthernet4
CSP(config-vsb-config)# interface gw-uplink1  mode passthrough
CSP(config-vsb-config)# interface gw-uplink2  mode passthrough
CSP(config-vsb-config)# interface management uplink GigabitEthernet1
CSP(config-vsb-config)# interface management vlan 751
CSP(config-vsb-config)# enable
```

Configuring the VXLAN Gateway Using the Setup Script

After you enter enable while installing a VXLAN gateway as a VSB, the setup script to configure the VXLAN gateway is executed. The setup script configures the following parameters on the VXLAN gateway:

- IP address, network mask, and default gateway for both the primary and secondary VXLAN gateway Management interface
- VSM details—Domain ID and the IP address of the VSM control interface

- Port profiles used for the VXLAN gateway uplink and VTEP

Step 1 On the command prompt, enter the VSB image and press Enter.

```
Enter VSB image:x.x.x.x.x.x.x.iso: [vxgw.5.2.1.SK1.2.1.iso]
```

Step 2 Enter the VSM domain ID. The range is from 1 to 1023.

```
Enter domain [1-1023]:405
```

Note You can get the domain ID by entering the **show svs domain** command on the VSM.

Step 3 Enter the management IP version.

```
Management IP version [V4]:v4
```

Step 4 Configure the management IP address to interface mgmt 0 on the VXLAN gateway that is deployed on the primary Cloud Services Platform.

```
Enter management IP address of service module on primary: 192.168.1.104
```

Note If you are deploying the gateway in the standalone mode on the secondary Cloud Services Platform, enter the IP address, network mask, and the default gateway address for the primary as 0.0.0.0.

Step 5 Enter the management subnet mask.

```
Enter management subnet mask of service module on primary: 255.255.255.0
```

Step 6 Enter the management default gateway.

```
Enter default gateway IP address of service module on primary: 192.168.1.1
```

Step 7 Configure the management IP address to interface mgmt 0 on the VXLAN gateway deployed on the secondary Cloud Services Platform.

```
Enter management IP address of service module on secondary: 192.168.1.105
```

Note If you are deploying the gateway in the standalone mode on the primary Cloud Services Platform, enter the IP address, network mask, and the default gateway address for the secondary as 0.0.0.0.

Note In a HA deployment, we recommend that the IP address that you provide is in the same subnet as the one provided in Step 5.

Step 8 Enter the management subnet mask.

```
Enter management subnet mask of service module on secondary: 255.255.255.0
```

Step 9 Enter the management interface default gateway.

```
Enter default gateway IP address of service module on secondary: 192.168.1.1
```

Step 10 Enter the VXLAN gateway hostname.

```
Enter HostName: VXLAN-GW-DOCS
```

Step 11 Enter the login credentials.

```
Enter the password for admin:Sfish123
```

Step 12 Enter the IP address of the VSM.

```
VSM L3 Ctrl IPv4 address:192.168.1.210
```

Step 13 Enter the uplink trunk port profile configured on the VSM.

```
Enter VSM uplink port-profile name: gw-uplink
```

Note Enter the dedicated uplink trunk port profile for the VXLAN gateway pair created on the VSM.

Step 14 Enter the VTEP profile name.

```
Enter VTEP port-profile name: gw-vtep
```

Note Enter the same VTEP port profile name created on the VSM.

This example shows how to bring up the VXLAN gateway:

```
CSP(config-vsb-config)# enable
Enter vsb image: [vxgw.5.2.1.SV3.1.1.iso]
Enter the VSM domain id[1-4095]: 405
Enter Management IP version [V4]: [V4]
Enter Management IP address of service module on primary: 192.168.1.104
Enter Management subnet mask of service module on primary: 255.255.255.0
Enter default gateway IP address of service module on primary: 192.168.1.1
Enter management IP address of service module on secondary: 192.168.1.105
Enter management subnet mask of service module on secondary: 255.255.255.0
Enter default gateway IP address of service module on secondary: 192.168.1.1
Enter HostName: VXLAN-GW-DOCS
Enter the password for 'admin': password
VSM L3 Ctrl IPv4 address : 192.168.1.210
Enter VSM uplink port-profile name: gw-uplink
Enter VTEP port-profile name: gw-vtep
Note: VSB installation is in progress, please use show virtual-service-blade commands to
check the installation status.
CSP(config-vsb-config)#
```

Modifying the Initial Setup Script Parameters

After executing the setup script for the first time, if you need to modify any of the setup parameters, use the following commands on the VSM:

Note If an HA pair is installed, ensure that you apply the same changes individually on both gateway modules.

Note Ensure the port profile that you update is first saved on the VSM.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm(config)# **service** *module* **update port-profile type ethernet name** *VXLAN Gateway Uplink port-profile name*	Modifies the VXLAN gateway uplink port profile from the VSM.
Step 2	vsm(config)# **service** *module* **update port-profile type vethernet name** *VXLAN Gateway VTEP port-profile name*	Modifies the VXLAN gateway VTEP port profile from the VSM.

Installing the VXLAN Gateway as a VM

Installing and Configuring VXLAN Gateway Using .iso Image

Installing the VXLAN Gateway as a VM Using the .iso Image

Before You Begin

- Ensure that the port profiles and bridge domains are configured on the VSM.
- Ensure that the VSM is connected to vCenter and that all the configurations are pushed from VSM to vCenter.
- Ensure that the image is available on the VMware host where the VXLAN is created.

Step 1 Log in to VMware vSphere client using your login credentials.

Step 2 In the left pane, right-click on the host and choose **New Virtual machine**.
Create New Virtual Machine window opens.

Step 3 Under the **Configuration** pane, click the **Custom** radio button.

Step 4 Click **Next**.

Note Click **Next** after each step unless instructed otherwise.

Step 5 In the **Name** field, enter a name for the VXLAN gateway VM.

Step 6 Under the **Storage** pane, choose the data store where the .iso image is copied.

Step 7 Under the **Virutal Machine Version** pane, click the **Virtual Machine Version: 8** radio button.

Step 8 In the **Guest Operating System** list, click the **Linux** radio button. From the **Version** drop-down list, choose **Ubuntu Linux (32 bit)**.

Step 9 Under **CPU** pane, from the **Number of virtual sockets** drop-down list, choose **2**. From the **Number of cores per virtual** drop-down list, choose **1**.

Step 10 Under the **Memory** pane, choose the memory size from the **Memory Size** drop-down list.
The minimum memory size required is 4 GB.

Step 11 Under the **Network** pane, from the **How many NICs do you want to** drop-down list, choose **2**. Do not click **Next**.

Step 12 For the first **NIC** field, from the **Network** drop-down list, choose a vEthernet trunk port that is already created on the VSM and from the **Adapter** drop-down list, choose **VMXNET3**. Do not click **Next**.
See Configuring a vEthernet Access Port Profile for VXLAN Gateway Management Port , on page 62 and Configuring a vEthernet Trunk Port Profile for VXLAN Gateway Uplink Port, on page 61 to configure port profiles.

Step 13 For the second **NIC** field, from the **Network** drop-down list, choose a vEthernet access port that is already created on the VSM and from the **Adapter** drop-down list, choose **VMXNET3**.
See Configuring a vEthernet Access Port Profile for VXLAN Gateway Management Port , on page 62 and Configuring a vEthernet Trunk Port Profile for VXLAN Gateway Uplink Port, on page 61 to configure port profiles.

Step 14 Under the **SCSI Controller** pane, click the **LSI Logic Parallel** radio button.

Step 15 Under the **Select a Disk** pane, click the **Create a new virtual disk** radio button.

Step 16 Under the **Capacity** field, choose the disk size from the **disk** drop-down list. For the **Disk Provisioning** and **Location** fields, keep the default values.

The minimum disk size required is 16 GB.

Step 17 Under the **Advanced Options** pane, keep the default values.

Step 18 Under the **Ready to Complete** pane, check the **Edit the virtual machine settings before** check box.

Step 19 Click **Continue**.
VM name - **Virtual Machine Properties** window appears.

Step 20 In the **Hardware** tab, click the **New CD/DVD (adding)** property.

Step 21 In the right pane, under **Device Type** pane, click the **Destination ISO File** radio button. Click **Browse** and choose the .iso image stored on the host.

Step 22 In the right pane, under the **Device Status** pane, check the **Connect at power on** check box.

Step 23 Click **Finish**.

Step 24 In the right pane of the **vSphere Client** window, right-click the new VM and choose **Power** > **Power On**.

Step 25 In the left pane of the **vSphere Client** window, click the new VM.

Step 26 In the right pane of **vSphere Client** window, click the **Console** tab.

Step 27 Press **Enter**.
Depending on the VM, the boot might take some time. Wait for `Enter the password for "admin"` prompt.

Step 28 Proceed to Configuring the VXLAN Gateway as a VM, on page 68.

Configuring the VXLAN Gateway as a VM

Before You Begin

- Install and power on the VM and verify that it has booted up and you see the `Confirm the password for "admin"` prompt. See Installing the VXLAN Gateway as a VM Using the .iso Image, on page 67 to install the VXLAN as a VM.

Step 1 Set an admin password on the command and press **Enter**. Make a note of this password.
`Enter the password for "admin":` <*password*>

Caution The password is not visible as you enter. Ensure that you do not make any typing errors.

The password should contain the following:

- At least one upper case letter
- At least one lower case letter
- At least one number

Step 2 Reenter the same password at the `Confirm the password for "admin"` prompt and press **Enter**.
`Confirm the password for "admin"` <*password*>

Step 3 Enter the domain ID and press **Enter**.
`Enter the domain id <1-1023>` <*domain id*>

Step 4 Enter **yes** and press **Enter**.

```
Continue with out-of-band (mgmt0) management configuration? (yes/no): yes
```

Step 5 Enter the management IP address and press **Enter**.
```
Mgmt0 IPv4 address: <IPv4 address>
```

Step 6 Enter the management netmask and press **Enter**.
```
Mgmt0 IPv4 netmask: <IPv4 address>
```

Step 7 Enter **y** and press **Enter**.
```
Configure the default gateway? (yes/no) (y): y
```

Step 8 Enter the default gateway and press **Enter**.
```
IPv4 address of the default gateway: <IPv4 address>
```

Step 9 Enter the IP address of the VSM and press **Enter**.
```
VSM L3 Ctrl IPv4 address <IPv4 address>
```

Step 10 Enter the uplink trunk port profile configured on the VSM and press **Enter**.
```
VSM uplink port-profile name <port name>
```
Note Enter the dedicated uplink trunk port profile for the VXLAN Gateway pair created on the VSM.

Step 11 Enter the VXLAN gateway encapsulation port profile configured on the VSM and press **Enter**.
```
Encapsulation port-profile name port profile name
```
Note Enter the dedicated vEthernet encapsulation port profile for the VXLAN Gateway pair created on the VSM.

Step 12 Enter **n** and press **Enter**.
```
Would you like to edit the configuration? (yes/no): n
```

Step 13 Enter **y** and press **Enter**.
```
Use this configuration and save it? (yes/no): y
```
The VM saves the configuration and reboots.

Installing and Configuring the VXLAN Gateway Using OVA

Before You Begin

- Ensure that the port profiles and bridge domains are configured on the VSM.
- Ensure that the VSM is connected to vCenter and that all the configurations are pushed from VSM to vCenter.
- Ensure that the OVA image is also available on the local disk where vCenter is running.

Step 1 Log in to VMware vSphere Client using your login credentials.

Step 2 From the **File** menu, choose **Deploy OVF Template**.

The **Deploy OVF Template** window opens.

Step 3 Click **Browse** and choose the OVF file stored on the host.

Step 4 Click **Next**.

Step 5 Click **Accept** and then **Next**.

Step 6 Under the **Name** field, enter a name for the VXLAN gateway VM.

Step 7 Under the **Inventory Location** pane, choose the datacenter and click **Next**.

Step 8 From the **Configuration** drop-down list, choose **Nexus 1000 vxlan Gateway Installation** and click **Next**.

Step 9 Under the **Host / Cluster** pane, choose the host and click **Next**.

Step 10 Under the **Disk Format** pane, choose the default options and click **Next**.

Step 11 Under the **Map the networks use in this OVF template to networks in your inventory** pane, from the **Destination Networks** drop-down list, choose the port profiles. See Configuring a vEthernet Access Port Profile for VXLAN Gateway Management Port , on page 62 and Configuring a vEthernet Trunk Port Profile for VXLAN Gateway Uplink Port, on page 61 for port profiles.

Step 12 Click **Next**.

Step 13 In the **Enter password** field, enter an admin password.

Caution The password is not visible as you type. Ensure that you do not make any typing errors.

The password should contain the following:

- At least one uppercase letter
- At least one lowercase letter
- At least one number

Step 14 In the **Confirm password** field, reenter the same password.

Step 15 In the **Domain Id** field, enter the domain ID.

Step 16 In the **Mgmt 0 IPV4 Address** field, enter the management IP address.

Step 17 In the **Mgmt 0 IPV4 Subet Mask** field, enter the management subnet mask.

Step 18 In the **IPV4 default gateway** field, enter the default gateway.

Step 19 In the **VSM L3 ctrl IPV4 address** field, enter the IP address of the VSM.

Step 20 In the **VXGW uplink port-profile name** field, enter the uplink trunk port profile configured on the VSM.

Note Enter the dedicated uplink trunk port profile for the VXLAN gateway pair created on the VSM.

Step 21 In the **VXGW encapsulation port-profile name** field, enter the access port profile configured on the VSM.

Note Enter the dedicated encapsulation port profile for the VXLAN gateway pair created on the VSM.

Step 22 Click **Next**.

Step 23 Check the **Power on after deployment** check box and click **Finish**.
VM is created and listed in the datacenter.

Configuring High Availability

VXLAN Gateway and High Availability

The operation of high availability (HA) involves the following terminology:

- Cluster—A cluster is a pair of gateway modules that operate together as a single HA module. Each cluster is distinguished by a unique cluster ID. A gateway module that is deployed in a standalone mode of operation is assigned a dummy cluster ID of 0.
- HA role—The gateway modules that make up an HA cluster are assigned separate roles. One is designated as primary and the other as secondary. This role decides which of the two modules goes to the active state first and which stays in the standby state. These states persist until the active module fails. If the active gateway module fails, the standby gateway module detects the failure and moves to the active state. This way, one of the two modules is always providing active service.
- HA state— At any given time, only one gateway module from a given cluster is actively performing the gateway function. The other module stays in the standby state pending the failure of the active module. A gateway module can be in the active or standby state. In addition, there is a transient initial state called the Init state. In this state, a gateway is either waiting to be assigned a role or negotiating its state with its peer.

After a gateway module is installed and brought up, the VSM assigns a role to the gateway module and can result in one of the following transitions:

- Unconfigured-Init to Standalone-Active
- Unconfigured-Init to Primary-Active
- Unconfigured-Init to Secondary-Standby
- Standalone-Active to Primary-Active
- Standalone-Active to Secondary-Active

For all other combinations, we recommend that you first fall back to the Unconfigured-Init mode by using the **no service** *VXLAN Gateway module* command and then proceed to the desired role or states.

Note You must preassign module numbers in the VSM. When a VXLAN gateway is attached to the VSM on that module, it inherits the role and state that are assigned by the VSM.

You must configure the HA mode of the VXLAN gateway as standalone or primary/secondary so that when you bring up the VXLAN gateway, the HA state is either active or standby and the VXLAN-to-VLAN mappings are either active or pending. If you do not configure an HA role for the VXLAN gateway, when you bring up the VXLAN gateway, the HA state is unknown, and VXLAN-to-VLAN traffic is not processed.

Configuring the VXLAN Gateway HA Mode as Standalone

You can create a service module in a standalone mode. Perform these steps on the VSM.

Before You Begin

You must preassign roles to module numbers in the VSM. When a VXLAN gateway is attached to the VSM on that module, it inherits the role and state that are assigned by the VSM.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm(config)# **service** *mod* **role standalone**	Configures the service module as standalone active.
Step 2	vsm(config)# **show module service**	Displays the service module number, cluster ID, role, HA mode, and status.

This example shows how to display the cluster ID mapping and the details about active, standby, and standalone service modules:

```
vsm(config)# show module service
Mod  Cluster-id  Role            HA Mode       Status
---  ----------  ------------    ----------    -------
36        0      Standalone      Standalone    Active
```

Configuring the VXLAN Gateway as an HA Pair

You can create a service module as a HA pair. Perform these steps on the VSM.

Before You Begin

You must create a second instance of the VXLAN gateway VM.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm(config)# **service** *modNo1* **role** *primary* **ha-cluster** *clusterNo*	Configures the service module in HA and adds a primary service module to a cluster.
Step 2	vsm(config)# **service** modNo2 **role** *secondary* **ha-cluster** *clusterNo*	Configures another service module as secondary in the same cluster.
Step 3	vsm(config)# **show module service-module**	Displays the service module number, cluster ID, role, HA mode, and status.

This example shows how to display the cluster ID mapping and the details about active, standby, and standalone service modules:

```
vsm(config)# show module service
Mod  Cluster-id  Role        HA Mode    Status
--- ---------- ----------- ---------- -------
9    1        Primary          HA       Active
```

```
10    1         Secondary         HA        Standby
```

To switch over between the active and standby VXLAN gateway, enter the following command on the VSM:

```
vsm# service ha-cluster 1 switchover
```

Verifying the VXLAN Gateway Configuration

To display the VXLAN gateway installation and configuration information, use one of the following commands on the VSM:

Command	Purpose
show running-config port-profile gw-uplink	Displays the configuration of the port profile assigned to the VXLAN gateway uplinks.
show running-config port-profile gw-vtep	Displays the configuration of the port profile assigned to the VXLAN VTEP.
show module	Displays the VXLAN gateway service modules.
show module service	Verifies the role of the VXLAN gateway module and displays the cluster ID mapping and the details about active, standby, and standalone service modules.
show vxlan gateway interface	Displays if the VTEPs are configured properly.
show interface vethernet 6	Displays if both the VTEP Virtual Ethernet interfaces are in the up state.
show port-channel summary	Displays if the port channels are up for gateway service modules.
show bridge-domain mappings	Displays VXLAN gateway mappings.
show switch edition	Displays if the VSM is in Advanced mode.
show feature	Displays if the VXLAN gateway is enabled on the VSM.
show virtual-service-blade summary **Note** You must enter this command from the Cloud Services Platform.	Displays the status of the VXLAN gateway VSB as it transitions from the VSB DEPLOY IN PROGRESS to VSB POWERED ON.
show virtual-service-blade **Note** You must enter this command from the Cloud Services Platform.	Displays the VXLAN gateway configuration.

Command	Purpose
show encapsulation profile	Displays the VLAN-to-VXLAN mappings for all encapsulation profiles or for the specified encapsulation profile.

This example shows how to display the status of the VXLAN gateway VSB:

```
CSP# show virtual-service-blade summary
--------------------------------------------------------------------------------
Name HA-Role HA-Status Status Location
--------------------------------------------------------------------------------
VXLAN-GW PRIMARY ACTIVE VSB POWERED ON PRIMARY
VXLAN-GW SECONDARY ACTIVE VSB POWERED ON SECONDARY
```

This example shows how to display the VXLAN gateway configuration:

```
CSP# show virtual-service-blade
virtual-service-blade VXLAN-GW
   Description:
   Slot id: 1
   Host Name: VXLAN-GW-DOCS
   Management IP: 192.168.1.104
   VSB Type Name : vx-gw-1.5
   Configured vCPU: 3
   Operational vCPU: 3
   Configured Ramsize: 2048
   Operational Ramsize: 2048
   Disksize: 3
   Heartbeat: 154764
   Legends: P - Passthrough
   ------------------------------------------------------------------------------
   Interface          Type         MAC          VLAN     State    Uplink-Int
                                                        Pri Sec   Oper Adm
   ------------------------------------------------------------------------------
   VsbEthernet1/1  gw-uplink1    0002.3d71.a303          up up  Gi3(P) Gi3(P)
   VsbEthernet1/2  management    0002.3d71.a302  751     up up  Gi1     Gi1
   VsbEthernet1/3  gw-uplink2    0002.3d71.a304          up up  Gi4(P) Gi4(P)
   internal          NA            NA             NA     up up
   HA Role: Primary
   HA Status: ACTIVE
   Status: VSB POWERED ON
   Location: PRIMARY
   SW version:
   HA Role: Secondary
   HA Status: ACTIVE
   Status: VSB POWERED ON
   Location: SECONDARY
   SW version:
   VSB Info:
   Domain ID : 405
```

This example shows how to display the port-profile configuration assigned to the VXLAN gateway uplinks:

```
vsm# show running-config port-profile gw-uplink

port-profile type ethernet gw-uplink
  switchport mode trunk
  switchport trunk allowed vlan 1,81,751-760
  mtu 1550
  channel-group auto mode active
  no shutdown
  state enabled
```

This example shows how to display the port-profile configuration assigned to the VXLAN VTEP:

```
vsm# show running-config port-profile gw-vtep

port-profile type vethernet gw-vtep
```

```
  switchport mode access
  switchport access vlan 760
  capability vxlan
  transport ip address 182.168.1.253 255.255.255.0 gateway 182.168.1.1
  no shutdown
  state enabled
```

This example shows how to display the VXLAN gateway service modules as soon as they are online:

```
vsm# show module
Mod  Ports  Module-Type                       Model               Status
---  -----  --------------------------------  ------------------  ------------
1    0      Virtual Supervisor Module         Nexus1000V          active *
3    1022   Virtual Ethernet Module           NA                  offline

Mod  Sw                Hw
---  ----------------  ------------------------------------------------
1    5.2(1)SV3(1.1)    0.0
3    5.2(1)SV3(1.1)    VMware ESXi 5.1.0 BETAbuild-802205 (3.1)

Mod  Server-IP        Server-UUID                          Server-Name
---  ---------------  -----------------------------------  --------------------
1    172.23.232.17    NA                                   NA
3    172.23.232.158   3a8fdc56-86d2-9044-969f-e2aea57d0ebf  NA

* this terminal session
```

This example shows how to display the cluster ID mapping and the details about active, standby, and standalone service modules:

```
vsm# show module service
Mod  Cluster-id  Role        HA Mode    Status
--- ---------- ----------- ---------- -------
9    1        Primary          HA       Active
10   1        Secondary        HA       Standby
```

This example shows how to display the module for virtual Ethernet interface binding:

```
vsm(config-if)# show vxlan gateway interface
-------------------------------------------------------------------
Port  IPAddress  Netmask  Gateway Mod Status Role
-------------------------------------------------------------------
Veth6  192.0.2.253  255.255.255.0 192.168.1.1 9  up Active
Veth22  192.0.2.253  255.255.255.0 192.168.1.1 10  up Standby
```

This example shows how to display whether both the VTEP virtual Ethernet interfaces are in the up state:

```
vsm# show interface vethernet 6
Vethernet6 is up
  Port description is VXLANGW VTEP, Network Adapter 1
  Hardware: Virtual, address: 0002.3d71.a303 (bia 0002.3d71.a303)
  Owner is VM "VXLANGW VTEP", adapter is Network Adapter 1
  Active on module 9
  Port-Profile is gw-vtep
  Port mode is access
  5 minute input rate 8 bits/second, 0 packets/second
  5 minute output rate 0 bits/second, 0 packets/second
  Rx
    6 Input Packets 6 Unicast Packets
    0 Multicast Packets 588 Broadcast Packets
    468 Bytes
  Tx
    34321 Output Packets 34321 Unicast Packets
    33609 Multicast Packets 24 Broadcast Packets 33633 Flood Packets
    2193700 Bytes
    0 Input Packet Drops 0 Output Packet Drops

vsm# show interface vethernet 22
Vethernet22 is up
  Port description is VXLANGW VTEP, Network Adapter 1
  Hardware: Virtual, address: 0002.3d71.a383 (bia 0002.3d71.a383)
  Owner is VM "VXLANGW VTEP", adapter is Network Adapter 1
```

```
  Active on module 10
  Port-Profile is gw-vtep
  Port mode is access
  5 minute input rate 8 bits/second, 0 packets/second
  5 minute output rate 0 bits/second, 0 packets/second
  Rx
    6 Input Packets 6 Unicast Packets
    0 Multicast Packets 25 Broadcast Packets
    468 Bytes
  Tx
    33742 Output Packets 33742 Unicast Packets
    33609 Multicast Packets 133 Broadcast Packets 33742 Flood Packets
    2158956 Bytes
    0 Input Packet Drops 0 Output Packet Drops
```

This example shows how to display whether the port channels are up for VXLAN gateway service modules:

```
vsm# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------------------------
1     Po1(SU)     Eth      NONE      Eth3/3(P)    Eth3/4(P)    Eth3/5(P)
                                     Eth3/6(P)
2     Po2(SU)     Eth      NONE      Eth4/3(P)    Eth4/4(P)    Eth4/5(P)
                                     Eth4/6(P)
3     Po3(SU)     Eth      NONE      Eth5/3(P)    Eth5/4(P)    Eth5/5(P)
                                     Eth5/6(P)
4     Po4(SU)     Eth      NONE      Eth6/3(P)    Eth6/4(P)    Eth6/5(P)
                                     Eth6/6(P)
5     Po5(SU)     Eth      NONE      Eth7/3(P)    Eth7/4(P)    Eth7/5(P)
                                     Eth7/6(P)
6     Po6(SU)     Eth      NONE      Eth8/4(P)
7     Po7(SU)     Eth      LACP      Eth9/1(P)    Eth9/3(P)
8     Po8(SU)     Eth      LACP      Eth10/1(P)   Eth10/3(P)
```

This example shows how to display VXLAN gateway mappings:

```
vsm# show bridge-domain mappings
--------------------------------------------------------------------------------
Interface       Module  Serv Inst  Vlan  BD-Name
--------------------------------------------------------------------------------
port-channel7   9       753        753   bd-753
port-channel8   10      753        753   bd-753
```

This example shows how to display the IP address for module binding:

```
vsm(config-if)# show module service mgmt-int
 --------------------------------------------------------------------------------
Mod Interface-Name IP-address Speed MTU
--------------------------------------------------------------------------------
4 Mgmt0 10.10.10.2             0     0
5 Mgmt0 10.10.10.3             0     0
Remember the management IP address user installs gateway with
(in this example 10.10.10.2, which occupies module slot 4)
```

This example shows how to display whether the VSM is in Advanced mode:

```
vsm# show switch edition
Switch Edition: Advanced

Advanced Features
Feature Name    Feature State
-----------------------------
vxlan-gateway   enabled
```

```
Licenses Available: 1020
Licenses In Use: 4
License Expiry Date: 13 Jun 2013
```

This example shows how to display whether the VXLAN gateway is enabled on the VSM:

```
vsm# show feature
Feature Name          Instance  State
--------------------  --------  --------
cts                   1         enabled
dhcp-snooping         1         enabled
http-server           1         enabled
lacp                  1         enabled
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         enabled
vxlan-gateway         1         enabled
```

Perform one of the following tasks on the VXLAN gateway. If your VSM is on Layer 3 through management and your gateway is also on the same management subnet, use the **attach module** *service module number* command to acces the gateway CLI. If your VSM is on Layer 3 through control, you can access the gateway CLI from any machine on that control subnet. This example shows the VSM which is on Layer 3 control.

Command	Purpose
show redundancy config	Displays the high availability status.

This example shows how to display the HA status:

```
gw# show redundancy config

HA Manager Node Information:

    Cluster Node Count: 2

    Local Node:
      state          : Active
      HA mode        : High Availability
      uuid           : 56fa6753-4dc5-4a7d-ad07-cc817114f838
      cluster_id     : 1
      node_priority  : 2
      node_type      : VXLAN Gateway
      ipaddr [mgmt]  : 192.168.1.104

    Peer Node 1:
      state          : Standby
      uuid           : 4cbd05df-b3e5-468a-9497-89aa3fae8153
      node_type      : VXLAN Gateway
      ipaddr [mgmt]  : 192.168.1.105
```

This example shows how to display the VLAN-to-VXLAN mappings for all encapsulation profiles:

```
gw# show encapsulation profile

--------------------------------------------------------------------------------
Vlan Bridge-domain
--------------------------------------------------------------------------------
2100 segment5050
2055 segment5031
```

```
2056 segment5032
2057 segment5033
2058 segment5034
```

Managing the VXLAN-to-VLAN Mappings on the VXLAN Gateway

The VLAN-to-VXLAN mappings that are configured on a gateway module can be managed by editing the port profile applied on the gateway uplink modules. To add or remove a mapping, perform these steps on the VSM.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm# **configure terminal**	Enters global configuration mode.
Step 2	vsm(config)# **encapsulation profile segment** *name*	Creates an encapsulation profile to contain the VLAN-to-VXLAN mappings.
Step 3	vsm(config-vxlan-encap-prof)# **dot1q** *VLAN-ID* **bridge-domain** *bd-name*	Maps a VLAN to a VXLAN. The VXLAN is specified through the bridge-domain name. **Note** The bridge-domain name and VLAN ID that you provide are not created during the port-profile configuration. The bridge-domain name and the VLAN ID that you provide should be in an active state or the mapping is held in an inactive state until you create the bridge-domain name and VLAN ID. **Note** Repeat this step to specify additional mappings. **Note** To remove a mapping, use the **no** form of this command.
Step 4	vsm(config-vxlan-encap-prof)# **exit**	Exits the current configuration mode.
Step 5	vsm(config)# **port-profile** *port-profile-name*	Specifies the name of the port profile applied to the VXLAN Gateway uplink interface.
Step 6	vsm(config-port-prof)# **service instance** *place holder*	Defines a place holder for mappings. The range is from 1 to 4096. **Note** Port profiles that contain the **service instance** keyword cannot be used for a non-VXLAN gateway module.
Step 7	vsm(config-port-prof-srv)# **encapsulation profile** *name*	Assigns the specified encapsulation profile to the port profile.
Step 8	vsm(config-port-prof-srv)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

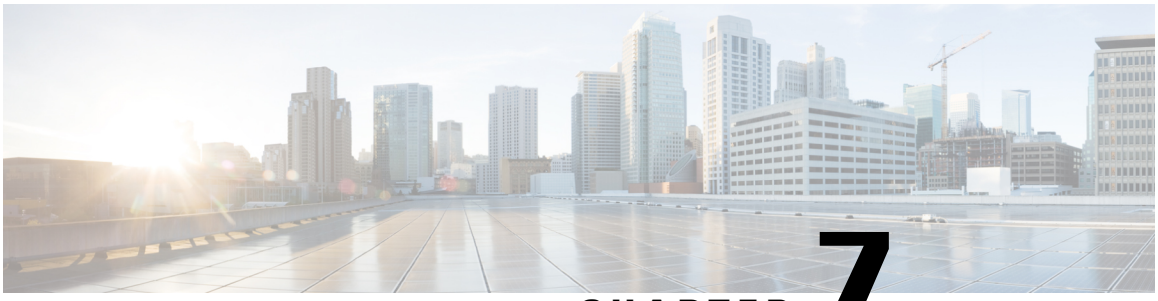
This example shows how to configure VXLAN-to-VLAN mappings on the VXLAN gateway:

```
vsm# configure terminal
```

```
vsm(config)# encapsulation profile segment mgmt_mappings
vsm(config-vxlan-encap-prof)# dot1q 1555 bridge-domain vxlan6000
vsm(config-vxlan-encap-prof)# dot1q 1557 bridge-domain vxlan6002
vsm(config-vxlan-encap-prof)# dot1q 1558 bridge-domain vxlan6003
vsm(config-vxlan-encap-prof)# dot1q 1559 bridge-domain vxlan6004
vsm(config-vxlan-encap-prof)# exit
vsm(config)# port-profile Uplink-All-VXGW
vsm(config-port-prof)# service instance 2
vsm(config-port-prof)# encapsulation profile mgmt_mappings
vsm(config-prot-prof-srv)# copy running-config startup-config
vsm(config)# show run port-profile Uplink-All-VXGW
port-profile type ethernet Uplink-All-VXGW
  switchport mode trunk
  switchport trunk allowed vlan 1545-1575,1577-1605
  mtu 1550
  service instance 2
    encapsulation dot1q 1555 bridge-domain vxlan6000
    encapsulation dot1q 1557 bridge-domain vxlan6002
    encapsulation dot1q 1558 bridge-domain vxlan6003
    encapsulation dot1q 1559 bridge-domain vxlan6004
   no shutdown
  state enabled
vsm(config)# show port-profile usage
port-profile Uplink-All-VXGW
 port-channel1
 port-channel5
 Ethernet7/1
 Ethernet7/3
vsm(config)# show run interface ethernet 7/1 expand-port-profile
interface Ethernet7/1
  switchport mode trunk
  switchport trunk allowed vlan 1545-1575,1577-1605
  mtu 1550
  channel-group auto mode active
  service instance 2
    no shutdown
    encapsulation dot1q 1557 bridge-domain vxlan6002
    encapsulation dot1q 1555 bridge-domain vxlan6000
    encapsulation dot1q 1558 bridge-domain vxlan6003
    no shutdown
```

Feature History for VXLAN Gateways

Feature Name	Releases	Feature Information
VXLAN Gateway	4.2(1)SV2(2.1)	Introduced the Virtual Extensible Local Area Network (VXLAN) gateway feature.
BGP Control Plane	5.2(1)SV3(1.1)	Introduced the Border Gateway Protocol (BGP) Control Plane feature.
VXLAN Gateway as a Virtual Machine	5.2(1)SV3(1.1)	Introduced the VXLAN gateway as a Virtual Machine feature.
VXLAN Gateway	5.2(1)SV3(1.15)	Starting with Release 5.2(1)SV3(1.15), Cisco Nexus 1000V for VMware vSphere does not support the VXLAN Gateway feature.



CHAPTER 7

Upgrading VXLAN Gateway from VSM

This chapter contains the following sections:

Upgrading the VXLAN Gateway Service Module

You can upgrade a VXLAN gateway service module (standalone) from the VSM by using the kickstart and system image or the iso image.

Note

- During the upgrade process, the VXLAN gateway service module is reloaded. Therefore, the VXLAN gateway service module might be unavailable for few minutes which can impact the data path traffic.
- Starting with Release 5.2(1)SV3(1.15), Cisco Nexus 1000V for VMware vSphere does not support the VXLAN Gateway feature.

Before You Begin

Ensure the following:

- The VXLAN gateway module is attached to the VSM.
- The VXLAN gateway module is configured in the standalone mode. For information about configuring the standalone mode, see Configuring the VXLAN Gateway HA Mode as Standalone, on page 71.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm# **install service-module kickstart bootflash:** *kickstart image* **system bootflash:** *system image* **module-num** *number*	Upgrades the VXLAN gateway service module by using the kickstart and system image. The range is from 3 to 130.
Step 2	vsm# **install service-module iso bootflash:** *iso image* **module-num** *number*	Upgrades the VXLAN gateway service module by using the iso image. The range is from 3 to 130.

Upgrading the VXLAN Gateway Cluster

You can upgrade a VXLAN gateway high availability (HA) cluster from the VSM by using the kickstart and system image or the iso image. To ensure that at least one VXLAN gateway service module within a cluster is available to serve the data path traffic during an upgrade, the service modules within a cluster are upgraded in the following sequence:

1. The VXLAN gateway standby service module is upgraded.

   After the VXLAN gateway standby module is upgraded and it is online, it forms the HA cluster with the VXLAN gateway active service module.

2. The VXLAN gateway active service module is upgraded so that the standby service module can serve the datapath traffic.

Note If the standby service module fails to form HA in step 1, the active service module is still upgraded to maintain a uniform software version in the cluster. In such a case, the datapath traffic is impacted.

Note We recommend that you upgrade the VXLAN gateway service module after upgrading the VSM and *before* upgrading the VEM. This recommendation applies to upgrades to Release 5.2(1)SV3(1.1) only.

Before You Begin

Ensure the following:

- The VXLAN gateway modules are attached to the VSM.
- The VXLAN gateway modules are configured in the HA mode. For information about configuring the HA mode, see Configuring the VXLAN Gateway as an HA Pair, on page 72.

DETAILED STEPS

	Command or Action	Purpose
Step 1	vsm# **install service-module kickstart bootflash:** *kickstart image* **system bootflash:** *system image* **cluster-id** *number*	Upgrades the VXLAN gateway cluster by using the kickstart and system image. The cluster ID range is from 1 to 8.
Step 2	vsm# **install service-module iso bootflash:** *iso image* **cluster-id** *number*	Upgrades the VXLAN gateway cluster by using the iso image. The cluster ID range is from 1 to 8.

Example for Upgrading to 5.2(1)SV3(1.1)

On VXLAN gateway upgrade to 5.2(1)SV3(1.1) the VXLAN gateway uplink port (Ethx/2) & VXLAN gateway access/vEth port are down. Recover by doing a 'no shutdown' on the respective ports as shown below:

Ports in NoPortProfile/Down state after VXLAN gateway upgrade:

```
VSM-CY(config)# sh interface brief
--------------------------------------------------------------------------------
Ethernet VLAN Type Mode Status Reason Speed Port
Interface Ch #
--------------------------------------------------------------------------------
Eth6/1 1 eth trunk up none auto 4
Eth6/2 1 eth access down NoPortProfile auto
--------------------------------------------------------------------------------
Vethernet VLAN/ Type Mode Status Reason MTU Module
Segment
--------------------------------------------------------------------------------
Veth38 1 virt access down NoPortProfile 1500 6
```

Recover VXLAN gateway ports by doing a 'no shutdown':

```
VSM-CY(config)# interface Eth6/2
VSM-CY(config-if)# no shutdown
VSM-CY(config)# interface Veth38
VSM-CY(config-if)# no shutdown
VSM-CY(config)# sh interface brief
--------------------------------------------------------------------------------
Ethernet VLAN Type Mode Status Reason Speed Port
Interface Ch #
--------------------------------------------------------------------------------
Eth6/2 1 eth trunk up none auto 4
--------------------------------------------------------------------------------
Vethernet VLAN/ Type Mode Status Reason MTU Module
Segment
--------------------------------------------------------------------------------
Veth38 1546 virt access up none 1500 6 "
```