



# Configuring VXLANs

---

This chapter contains the following sections:

- [Overview, page 1](#)
- [Configuring VXLAN Traffic Forwarding, page 7](#)
- [Verifying the VXLAN Configuration, page 13](#)
- [Displaying MAC Addresses, page 15](#)
- [Clearing MAC Addresses, page 19](#)

## Overview

### VXLAN Overview

The Cisco Nexus 3100 Series switches are designed for a hardware-based Virtual Extensible LAN (VXLAN) function. These switches can extend Layer 2 connectivity across the Layer 3 boundary and integrate between VXLAN and non-VXLAN infrastructures. Virtualized and multitenant data center designs can be shared over a common physical infrastructure.

VXLANs enable you to extend Layer 2 networks across the Layer 3 infrastructure by using MAC-in-UDP encapsulation and tunneling. In addition, you can use a VXLAN to build a multitenant data center by decoupling tenant Layer 2 segments from the shared transport network.

When deployed as a VXLAN gateway, the Cisco Nexus 3100 Series switches can connect VXLAN and classic VLAN segments to create a common forwarding domain so that tenant devices can reside in both environments.

A VXLAN has the following benefits:

- Flexible placement of multitenant segments throughout the data center.

It extends Layer 2 segments over the underlying shared network infrastructure so that tenant workloads can be placed across physical pods in the data center.

- Higher scalability to address more Layer 2 segments.

A VXLAN uses a 24-bit segment ID called the VXLAN network identifier (VNID). The VNID allows a maximum of 16 million VXLAN segments to coexist in the same administrative domain. (In comparison, traditional VLANs use a 12-bit segment ID that can support a maximum of 4096 VLANs.)

- Utilization of available network paths in the underlying infrastructure.

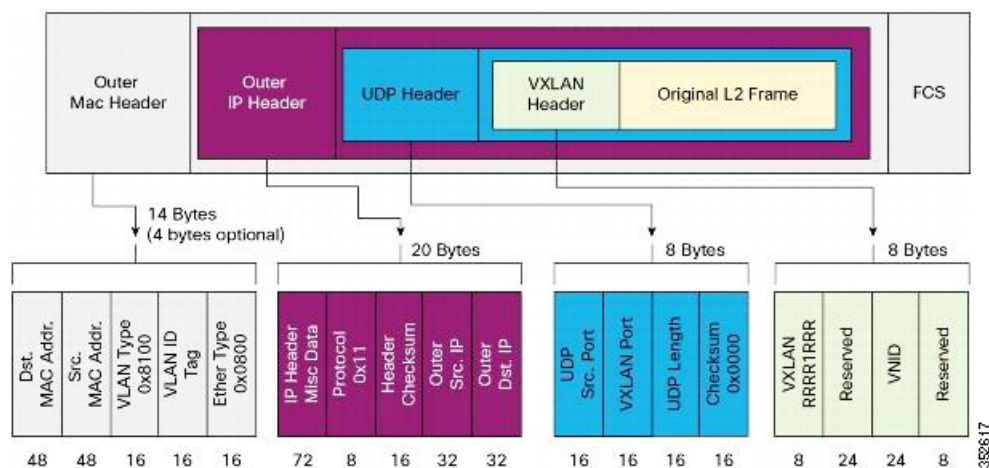
VXLAN packets are transferred through the underlying network based on its Layer 3 header. It uses equal-cost multipath (ECMP) routing and link aggregation protocols to use all available paths.

## VXLAN Encapsulation and Packet Format

A VXLAN is a Layer 2 overlay scheme over a Layer 3 network. It uses MAC-in-UDP encapsulation to extend Layer 2 segments across the data center network. The transport protocol over the physical data center network is IP plus UDP.

A VXLAN defines a MAC-in-UDP encapsulation scheme where the original Layer 2 frame has a VXLAN header added and is then placed in a UDP-IP packet. With this MAC-in-UDP encapsulation, VXLAN tunnels Layer 2 network over the Layer 3 network. The VXLAN packet format is shown in the following figure.

**Figure 1: VXLAN Packet Format**



A VXLAN uses an 8-byte VXLAN header that consists of a 24-bit VNID and a few reserved bits. The VXLAN header and the original Ethernet frame are in the UDP payload. The 24-bit VNID identifies the Layer 2 segments and maintains Layer 2 isolation between the segments. A VXLAN can support 16 million LAN segments.

## VXLAN Tunnel Endpoints

A VXLAN uses VXLAN tunnel endpoint (VTEP) devices to map tenants' end devices to VXLAN segments and to perform VXLAN encapsulation and deencapsulation. Each VTEP device has two types of interfaces:

- Switch port interfaces on the local LAN segment to support local endpoint communication through bridging
- IP interfaces to the transport network where the VXLAN encapsulated frames will be sent

A VTEP device is identified in the IP transport network by using a unique IP address, which is a loopback interface IP address. The VTEP device uses this IP address to encapsulate Ethernet frames and transmits the

encapsulated packets to the transport network through the IP interface. A VTEP device learns the remote VTEP IP addresses and the remote MAC address-to-VTEP IP mapping for the VXLAN traffic that it receives.

The VXLAN segments are independent of the underlying network topology; conversely, the underlying IP network between VTEPs is independent of the VXLAN overlay. The IP network routes the encapsulated packets based on the outer IP address header, which has the initiating VTEP as the source IP address and the terminating VTEP or multicast group IP address as the destination IP address.

## VXLAN Packet Forwarding Flow

A VXLAN uses stateless tunnels between VTEPs to transmit traffic of the overlay Layer 2 network through the Layer 3 transport network.

## VXLAN Implementation on Cisco Nexus 3100 Series Switches

The Cisco Nexus 3100 Series switches support the hardware-based VXLAN function that extends Layer 2 connectivity across the Layer 3 transport network and provides a high-performance gateway between VXLAN and non-VXLAN infrastructures.

## Layer 2 Mechanisms for Broadcast, Unknown Unicast, and Multicast Traffic

A VXLAN on the Cisco Nexus 3100 Series switches uses flooding and dynamic MAC address learning to do the following:

- Transport broadcast, unknown unicast, and multicast traffic
- Discover remote VTEPs
- Learn remote host MAC addresses and MAC-to-VTEP mappings for each VXLAN segment

A VXLAN can forward these traffic types as follows:

- Using multicast in the core—IP multicast reduces the flooding of the set of hosts that are participating in the VXLAN segment. Each VXLAN segment, or VNID, is mapped to an IP multicast group in the transport IP network. The Layer 2 gateway uses Protocol Independent Multicast (PIM) to send and receive traffic from the rendezvous point (RP) for the IP multicast group. The multicast distribution tree for this group is built through the transport network based on the locations of participating VTEPs.
- Using ingress replication—Each VXLAN segment or VXLAN network identifier (VNI) is mapped to a remote unicast peer. The Layer 2 frame is VXLAN encapsulated with the destination IP address as the remote unicast peer IP address and is sent out to the IP transport network where it gets unicast routed or forwarded to the remote destination.

## Layer 2 Mechanisms for Unicast-Learned Traffic

The Cisco Nexus 3100 Series switches perform MAC address lookup-based forwarding for VXLAN unicast-learned traffic.

When Layer 2 traffic is received on the access side, a MAC address lookup is performed for the destination MAC address in the frame. If the lookup is successful, VXLAN forwarding is done based on the information

retrieved as a result of the lookup. The lookup result provides the IP address of the remote VTEP from which this MAC address is learned. This Layer 2 frame is then UDP/IP encapsulated with the destination IP address as the remote VTEP IP address and is forwarded out of the appropriate network interface. In the Layer 3 cloud, this IP packet is forwarded to the remote VTEP through the route to that IP address in the network.

For unicast-learned traffic, you must ensure the following:

- The route to the remote peer is known through a routing protocol or through static routes in the network.
- Adjacency is resolved.

## VXLAN Layer 2 Gateway as a Transit Multicast Router

A VXLAN Layer 2 gateway must terminate VXLAN-multicast traffic that is headed to any of the groups to which VNIs are mapped. In a network, a VXLAN Layer 2 gateway can be a multicast transit router for the downstream multicast receivers that are interested in the group's traffic. A VXLAN Layer 2 gateway must do some additional processing to ensure that VXLAN multicast traffic that is received is both terminated and multicast routed. This traffic processing is done in two passes:

- 1 The VXLAN multicast traffic is multicast routed to all network receivers interested in that group's traffic.
- 2 The VXLAN multicast traffic is terminated, decapsulated, and forwarded to all VXLAN access side ports.

## ECMP and LACP Load Sharing with VXLANs

Encapsulated VXLAN packets are forwarded between VTEPs based on the native forwarding decisions of the transport network. Most data center transport networks are designed and deployed with multiple redundant paths that take advantage of various multipath load-sharing technologies to distribute traffic loads on all available paths.

A typical VXLAN transport network is an IP-routing network that uses the standard IP equal cost multipath (ECMP) to balance the traffic load among multiple best paths. To avoid out-of-sequence packet forwarding, flow-based ECMP is commonly deployed. An ECMP flow is defined by the source and destination IP addresses and optionally, the source and destination TCP or UDP ports in the IP packet header.

All the VXLAN packet flows between a pair of VTEPs have the same outer source and destination IP addresses, and all VTEP devices must use one identical destination UDP port that can be either the Internet Assigned Numbers Authority (IANA)-allocated UDP port 4789 or a customer-configured port. The only variable element in the ECMP flow definition that can differentiate VXLAN flows from the transport network standpoint is the source UDP port. A similar situation for Link Aggregation Control Protocol (LACP) hashing occurs if the resolved egress interface that is based on the routing and ECMP decision is an LACP port channel. LACP uses the VXLAN outer-packet header for link load-share hashing, which results in the source UDP port being the only element that can uniquely identify a VXLAN flow.

In the Cisco Nexus 3100 Series switches implementation of VXLANs, a hash of the inner frame's header is used as the VXLAN source UDP port. As a result, a VXLAN flow can be unique. The IP address and UDP port combination is in its outer header while the packet traverses the underlay transport network.

## Guidelines and Limitations for VXLANs

VXLAN has the following guidelines and limitations:

- The Cisco Nexus 3100 Series switches do not support VTEP under vPC.
- The Cisco Nexus 3100 series switches do not support anycast RP with VxLAN.
- IGMP snooping is not supported on VXLAN VLANs.
- VXLAN routing is not supported. The default Layer 3 gateway for VXLAN VLANs must be provisioned on a different device.
- Ensure that the network can accommodate an additional 50 bytes for the VXLAN header.
- Only one Network Virtualization Edge (NVE) interface is supported on a switch.
- Layer 3 VXLAN uplinks are not supported in a nondefault virtual and routing forwarding (VRF) instance.
- Only one VXLAN IP adjacency is possible per physical interface.
- Switched virtual interfaces (SVIs) are not supported on VXLAN VLANs.
- Switched Port Analyzer (SPAN) Tx for VXLAN-encapsulated traffic is not supported for the Layer 3 uplink interface.
- Access control lists (ACLs) and quality of service (QoS) for VXLAN traffic to access direction are not supported.
- SNMP is not supported on the NVE interface.
- Native VLANs for VXLAN are not supported.
- For ingress replication configurations, multiple VNIs can now have the same remote peer IP configured.
- Use the **ip pim spt-threshold infinity group-list** command to ensure that Shortest Path Tree (SPT) is not selected for the VXLAN multicast group.
- The VXLAN source UDP port is determined based on the VNID and source and destination IP addresses.
- The UDP port configuration must be done before the NVE interface is enabled. If the UDP configuration must be changed while the NVE interface is enabled, you must shut down the NVE interface, make the UDP configuration change, and then reenabling the NVE interface.
- When a VN-Segment is mapped to a native VLAN, if traffic is sent on any normal VLAN on that port instead of getting switched in the VLAN, it gets forwarded in the VXLAN tunnel for the native VLAN.

## Considerations for VXLAN Deployment

The following are some of the considerations while deploying VXLANs:

- A loopback interface IP is used to uniquely identify a VTEP device in the transport network.
- To establish IP multicast routing in the core, an IP multicast configuration, PIM configuration, and Rendezvous Point (RP) configuration are required.
- You can configure VTEP-to-VTEP unicast reachability through any IGP protocol.
- You can configure a VXLAN UDP destination port as required. The default port is 4789.
- The default gateway for VXLAN VLANs should be provisioned on a different upstream router.
- VXLAN multicast traffic should always use the RPT shared tree.

- An RP for the multicast group on the VTEP is a supported configuration. However, you must configure the RP for the multicast group at the spine layer/upstream device. Because all multicast traffic traverses the RP, it is more efficient to have this traffic directed to a spine layer/upstream device.

## vPC Guidelines and Limitations for VXLAN Deployment

- You must bind NVE to a loopback address that is separate from other loopback addresses required by Layer 3 protocols. Use a dedicated loopback address for VXLAN.
- vPC peers must have identical configurations for the following:
  - Consistent VLAN to VN-Segment mapping.
  - Consistent NVE binding:
    - Using the same source interface IP address.
    - Using consistent VNI to group mapping.
- For multicast, the vPC node that receives the (S,G) join from the RP becomes the designated forwarder (DF). On the DF node, both encapsulation and decapsulation routes are installed for multicast. The other vPC node does not initiate or terminate multicast traffic.
- Multicast traffic on a vPC that is hashed toward the non-DF switch traverses the multichassis EtherChannel trunk (MCT) and is encapsulated on the DF node.
- When MCT is shut, the loopback interface on the secondary vPC is brought down and the status is **Admin Shut**. The route to the loopback is withdrawn on the upstream and the upstream can divert all traffic to the primary vPC.



### Note

Orphans that are connected to the secondary vPC experience a loss of traffic when the MCT is shut down. This situation is similar to Layer 2 orphans in a secondary vPC of a traditional vPC setup.

- In a VXLAN vPC, consistency checks are performed to ensure that NVE configurations and VN-Segment configurations are identical across vPC peers.
- The router ID for unicast routing protocols must be different from the loopback IP address used for VTEP.
- When MCT is no-shut, the NVE loopback interface is brought up again and the route is advertised upstream to attract traffic.
- Configure an SVI between vPC peers and advertise routes between the vPC peers by using a routing protocol with higher routing metric. This action ensures that the IP connectivity of the vPC node does not go down if one vPC node fails.

## Configuring VXLAN Traffic Forwarding

There are two options for forwarding broadcast, unknown unicast and multicast traffic on a VXLAN Layer 2 gateway. [Layer 2 Mechanisms for Broadcast, Unknown Unicast, and Multicast Traffic](#), on page 3 provides more information about these two options.

Before you enable and configure VXLANs, ensure that the following configurations are complete:

- For IP multicast in the core, ensure that the IP multicast configuration, the PIM configuration, and the RP configuration are complete, and that a routing protocol exists.
- For ingress replication, ensure that a routing protocol exists for reaching unicast addresses.



### Note

On a Cisco Nexus 3100 Series switch that functions as a VXLAN Layer 2 gateway, note that traffic that is received on the access side cannot trigger an ARP on the network side. ARP for network side interfaces should be resolved either by using a routing protocol such as BGP, or by using static ARP. This requirement is applicable for ingress replication cases alone, not for multicast replication cases.

## Enabling and Configuring the PIM Feature

Before you can access the PIM commands, you must enable the PIM feature.

This is a prerequisite only for multicast replication.

### Before You Begin

Ensure that you have installed the LAN Base Services license.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>feature pim</b>	Enables PIM. By default, PIM is disabled.
<b>Step 3</b>	switch(config)# <b>ip pim spt-threshold infinity group-list route-map-name</b>	Creates the IPv4 Protocol Independent Multicast (PIM) (*, G) state only. Allows selection of the RPT only and not the SPT.
<b>Step 4</b>	switch(config)# <b>show running-config pim</b>	(Optional) Shows the running-configuration information for PIM, including the <b>feature</b> command.
<b>Step 5</b>	switch(config)# <b>copy running-config startup-config</b>	(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 PIM feature:

```
switch# configure terminal
switch(config)# feature pim
switch(config)# ip pim spt-threshold infinity group-list rp_name
switch(config)# show running-config pim

!Command: show running-config pim
!Time: Wed Mar 26 08:04:23 2014

version 6.0(2)U3(1)
feature pim

ip pim spt-threshold infinity group-list rp_name
```

## Configuring a Rendezvous Point

You can configure a rendezvous point (RP) by configuring the RP address on every router that will participate in the PIM domain.

This is a prerequisite only for multicast replication.

### Before You Begin

Ensure that you have installed the LAN Base Services license and enabled PIM.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>ip pim rp-address</b> <i>rp-address</i> [ <b>group-list</b> <i>ip-prefix</i>   <b>route-map</b> <i>policy-name</i> ]	Configures a PIM RP address for a multicast group range. You can specify a route-map policy name that lists the group prefixes to use with the <b>match ip multicast</b> command. The default mode is ASM. The default group range is 224.0.0.0 through 239.255.255.255.
<b>Step 3</b>	switch(config)# <b>show ip pim group-range</b> [ <i>ip-prefix</i> ] [ <b>vrf</b> { <i>vrf-name</i>   <b>all</b>   <b>default</b>   <b>management</b> }]	(Optional) Displays PIM modes and group ranges.
<b>Step 4</b>	switch(config)# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure an RP:

```
switch# configure terminal
switch(config)# ip pim rp-address 111.1.1.1 group-list 224.0.0.0/4
```



## Enabling a VXLAN

Enabling VXLANs involves the following:

- Enabling the VXLAN feature
- Enabling VLAN to VN-Segment mapping

### Before You Begin

Ensure that you have installed the VXLAN Enterprise license.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>[no] feature nv overlay</b>	Enables the VXLAN feature.
<b>Step 3</b>	switch (config)# <b>[no] feature vn-segment-vlan-based</b>	Configures the global mode for all VXLAN bridge domains.  Enables VLAN to VN-Segment mapping. VLAN to VN-Segment mapping is always one-to-one.
<b>Step 4</b>	switch(config)# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable a VXLAN and configure VLAN to VN-Segment mapping:

```
switch# configure terminal
switch(config)# feature nv overlay
switch(config)# feature vn-segment-vlan-based
switch(config)# copy running-config startup-config
```

## Mapping a VLAN to a VXLAN VNI

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vlan <i>vlan-id</i></b>	Specifies a VLAN.

	Command or Action	Purpose
<b>Step 3</b>	switch(config-vlan)# <b>vn-segment</b> <i>vnid</i>	Specifies the VXLAN virtual network identifier (VNID).

This example shows how to map a VLAN to a VXLAN VNI:

```
switch# configure terminal
switch(config)# vlan 3100
switch(config-vlan)# vn-segment 5000
```

## Configuring a Routing Protocol for NVE Unicast Addresses

Configuring a routing protocol for unicast addresses involves the following:

- Configuring a dedicated loopback interface for NVE reachability.
- Configuring the routing protocol network type.
- Specifying the routing protocol instance and area for an interface.
- Enabling PIM sparse mode in case of multicast replication.



### Note

Open shortest path first (OSPF) is used as the routing protocol in the examples.

This is a prerequisite for both multicast and ingress replication.

Guidelines for configuring a routing protocol for unicast addresses are as follows:

- For ingress replication, you can use a routing protocol that can resolve adjacency, such as BGP.
- When using unicast routing protocols in a vPC topology, explicitly configure a unique router ID for the vPC peers to avoid the VTEP loopback IP address (which is the same on the vPC peers) being used as the router ID.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface loopback</b> <i>instance</i>	Creates a dedicated loopback interface for the NVE interface. The instance range is from 0 to 1023.
<b>Step 3</b>	switch(config-if)# <b>ip address</b> <i>ip-address/length</i>	Configures an IP address for this interface.
<b>Step 4</b>	switch(config-if)# <b>ip ospf network</b> { <b>broadcast</b>   <b>point-to-point</b> }	Configures the OSPF network type to a type other than the default for an interface.

	Command or Action	Purpose
<b>Step 5</b>	switch(config-if)# <b>ip router ospf</b> <i>instance-tag area area-id</i>	Specifies the OSPF instance and area for an interface.
<b>Step 6</b>	switch(config-if)# <b>ip pim sparse-mode</b>	Enables PIM sparse mode on this interface. The default is disabled.  Enable the PIM sparse mode in case of multicast replication.

This example shows how to configure a routing protocol for NVE unicast addresses:

```
switch# configure terminal
switch(config)# interface loopback 10
switch(config-if)# ip address 222.2.2.1/32
switch(config-if)# ip ospf network point-to-point
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# ip pim sparse-mode
```

## Creating a VXLAN Destination UDP Port

The UDP port configuration should be done before the NVE interface is enabled.



### Note

If the configuration must be changed while the NVE interface is enabled, ensure that you shut down the NVE interface, make the UDP configuration change, and then reenables the NVE interface.

Ensure that the UDP port configuration is done network-wide before the NVE interface is enabled on the network.

The VXLAN UDP source port is determined based on the VNID and source and destination IP addresses.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vlan udp port</b> <i>number</i>	Specifies the destination UDP port number for VXLAN encapsulated packets. The default destination UDP port number is 4789.

This example shows how to create a VXLAN destination UDP port:

```
switch# configure terminal
switch(config)# vlan udp port 4789
```

## Creating and Configuring an NVE Interface

An NVE interface is the overlay interface that initiates and terminates VXLAN tunnels. You can create and configure an NVE (overlay) interface.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface nve</b> <i>instance</i>	Creates a VXLAN overlay interface that initiates and terminates VXLAN tunnels.  <b>Note</b> Only one NVE interface is allowed on the switch.
<b>Step 3</b>	switch(config-if-nve)# <b>source-interface loopback</b> <i>instance</i>	Specifies a source interface.  The source interface must be a loopback interface that is configured on the switch with a valid /32 IP address. This /32 IP address must be known by the transit routers in the transport network and the remote VTEPs.

This example shows how to create and configure an NVE interface:

```
switch# configure terminal
switch(config)# interface nve 1
switch(config-if-nve)# source-interface loopback 10
```

## Configuring Replication for a VNI

Replication for VXLAN network identifier (VNI) can be configured in one of two ways:

- Multicast replication
- Ingress replication

### Configuring Multicast Replication

#### Before You Begin

- Ensure that the NVE interface is created and configured.
- Ensure that the source interface is specified.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	switch(config-if-nve)# <b>member vni</b> { <i>vniid</i> <b>mcast-group</b> <i>multicast-group-addr</i>   <i>vniid- range</i> <b>mcast-group</b> <i>start-addr</i> [ <i>end-addr</i> ]}	Maps VXLAN VNIs to the NVE interface and assigns a multicast group to the VNIs.

This example shows how to map a VNI to an NVE interface and assign it to a multicast group:

```
switch(config-if-nve)# member vni 5000 mcast-group 225.1.1.1
```

## Configuring Ingress Replication

### Before You Begin

- Ensure that the NVE interface is created and configured.
- Ensure that the source interface is specified.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	switch(config-if-nve)# <b>member vni</b> <i>vniid</i>	Maps VXLAN VNIs to the NVE interface.
<b>Step 2</b>	switch(config-if-nve-vni)# <b>ingress-replication</b> <i>ip-address</i>	Enables ingress replication for the VNI to the specified unicast address.  <b>Note</b> <ul style="list-style-type: none"> <li>• A VNI can be associated only with a single IP address.</li> <li>• An IP address can be associated only with a single VNI.</li> </ul>

This example shows how to map a VNI to an NVE interface and create a unicast tunnel:

```
switch(config-if-nve)# member vni 5001
switch(config-if-nve-vni)# ingress-replication 111.1.1.1
```

## Verifying the VXLAN Configuration

Use one of the following commands to verify the VXLAN configuration:

Command	Purpose
<b>show nve interface</b> <i>nve id</i>	Displays the configuration of an NVE interface.

Command	Purpose
<b>show nve vni</b>	Displays the VNI that is mapped to an NVE interface.
<b>show nve peers</b>	Displays peers of the NVE interface.
<b>show interface nve id counters</b>	Displays all the counters for an NVE interface.
<b>show nve vxlan-params</b>	Displays the VXLAN UDP port configured.

This example shows how to display the configuration of an NVE interface:

```
switch# show nve interface nve 1
Interface: nve1, State: up, encapsulation: VXLAN
Source-interface: loopback10 (primary: 111.1.1.1, secondary: 0.0.0.0)
```

This example shows how to display the VNI that is mapped to an NVE interface for multicast replication:

```
switch# show nve vni
Interface      VNI      Multicast-group  VNI State
-----
nve1           5000      225.1.1.1        Up
```

This example shows how to display the VNI that is mapped to an NVE interface for ingress replication:

```
switch# show nve vni
Interface      VNI      Multicast-group  VNI State
-----
nve1           5000      0.0.0.0          Up
```

This example shows how to display the peers of an NVE interface:

```
switch# show nve peers
Interface      Peer-IP      Peer-State
-----
nve1           111.1.1.1    Up
```

This example shows how to display the counters of an NVE interface:

```
switch# show interface nv 1 counter
```

```
-----
Port           InOctets      InUcastPkts
-----
nve1           0              0

-----
Port           InMcastPkts    InBroadcastPkts
-----
nve1           0              0

-----
Port           OutOctets      OutUcastPkts
-----
nve1           0              0

-----
Port           OutMcastPkts    OutBroadcastPkts
-----
nve1           0              0
```

This example shows how to display the VXLAN UDP port configured:

```
switch# show nve vxlan-params
VxLAN Dest. UDP Port: 4789
```

## Displaying MAC Addresses

Enter one of these commands to display VXLAN and VLAN MAC addresses:

Command	Purpose
<b>show mac address-table</b>	Displays both VLAN and VXLAN MAC addresses.
<b>show mac address-table vlan</b> <i>vlan-id</i>	Displays all the VxLAN MAC addresses that are learned on the specified VLAN. For VN-Segment mapped VLANs, it displays both local and remote MAC addresses.
<b>show mac address-table local</b>	Displays only locally learned MAC addresses on all VLANs that are mapped to VN-Segments.
<b>show mac address-table local vlan</b> <i>vlan-id</i>	Displays only locally learned MAC addresses on the specified VLAN, which is mapped to a VN-Segment.
<b>show mac address-table interface nve</b> <i>nve-id</i>	Displays all remote MAC addresses learned on NVE.
<b>show mac address-table interface nve</b> <i>nve-id vni vni-id</i>	Displays all remote MAC addresses learned on the VNI.
<b>show mac address-table interface ethernet</b> <i>slot/port</i> <b>vlan</b> <i>vlan-id</i>	Displays all MAC addresses learned on the VLAN on this interface.
<b>show mac address-table interface nve</b> <i>nve-id peer ip-address</i> <b>show mac address-table interface nve</b> <i>nve-id peer vrf vrf-name ip-address</i>	Displays all MAC addresses learned on NVE from the specified peer.
<b>show mac address-table interface nve</b> <i>nve-id peer ip-address vni vni-id</i> <b>show mac address-table interface nve</b> <i>nve-id peer vrf vrf-name ip-address vni vni-id</i>	Displays all MAC addresses learned on NVE from the specified peer on the specified VNI.
<b>show mac address-table count local</b>	Displays the number of locally learned MAC address table entries.
<b>show mac address-table count local vlan</b> <i>vlan-id</i>	Displays the number of locally learned MAC address table entries on the specified VLAN, which is mapped to a VN-segment.

Command	Purpose
<b>show mac address-table count interface nve nve-id</b>	Displays the number of remote MAC address table entries learned on NVE.
<b>show mac address-table count interface nve nve-id vni vni-id</b>	Displays the number of remote MAC address table entries learned on the VNI.
<b>show mac address-table count interface nve nve-id peer ip-address</b> <b>show mac address-table count interface nve nve-id peer ip-address vrf vrf-name</b>	Displays the number of MAC address table entries learned on NVE from the specified peer.
<b>show mac address-table count interface nve nve-id peer ip-address vni vni-id</b> <b>show mac address-table count interface nve nve-id peer ip-address vrf vrf-name vni vni-id</b>	Displays the number of MAC address table entries learned on NVE from the specified peer on the specified VNI.

This example shows how to display both VLAN and VXLAN MAC addresses:

```
switch# show mac address-table
```

Legend:

```

      * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
      age - seconds since first seen, + - primary entry using vPC Peer-Link
      VLAN      MAC Address      Type      age      Secure NTFY      Ports/SWID.SSID.LID
-----+-----+-----+-----+-----+-----+-----
* 109          0000.0410.0902    dynamic    470          F      F      Po2233
* 109          0000.0410.0912    dynamic    470          F      F      Po2233
* 109          0000.0410.0912    dynamic    470          F      F      nve1(1.1.1.200)
* 108          0000.0410.0802    dynamic    470          F      F      Po2233
* 108          0000.0410.0812    dynamic    470          F      F      Po2233
* 107          0000.0410.0702    dynamic    470          F      F      Po2233
* 107          0000.0410.0712    dynamic    470          F      F      Po2233
* 107          0000.0410.0712    dynamic    470          F      F      nve1(1.1.1.200)
* 106          0000.0410.0602    dynamic    470          F      F      Po2233
* 106          0000.0410.0612    dynamic    470          F      F      Po2233
* 105          0000.0410.0502    dynamic    470          F      F      Po2233
* 105          0000.0410.0512    dynamic    470          F      F      Po2233
* 105          0000.0410.0512    dynamic    470          F      F      nve1(1.1.1.200)
* 104          0000.0410.0402    dynamic    470          F      F      Po2233
* 104          0000.0410.0412    dynamic    470          F      F      Po2233

```

This example shows how to display all the VXLAN MAC addresses learned on the specified VLAN:

```
switch# show mac address-table vlan 107
```

Legend:

```

      * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
      age - seconds since first seen, + - primary entry using vPC Peer-Link
      VLAN      MAC Address      Type      age      Secure NTFY      Ports/SWID.SSID.LID
-----+-----+-----+-----+-----+-----+-----
* 107          0000.0410.0702    dynamic    470          F      F      Po2233
* 107          0000.0410.0712    dynamic    470          F      F      Po2233
* 107          0000.0410.0712    dynamic    470          F      F      nve1(1.1.1.200)

```

This example shows how to display only locally learned MAC addresses on all VLANs that are mapped to VN-Segments:

```
switch# show mac address-table local
```

Legend:

```

      * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
      age - seconds since first seen, + - primary entry using vPC Peer-Link
      VLAN      MAC Address      Type      age      Secure NTFY      Ports/SWID.SSID.LID
-----+-----+-----+-----+-----+-----+-----

```



```

* 109      0000.0410.0902      dynamic  470      F      F      Po2233
* 109      0000.0410.0912      dynamic  470      F      F      Po2233
* 108      0000.0410.0802      dynamic  470      F      F      Po2233
* 108      0000.0410.0812      dynamic  470      F      F      Po2233
* 107      0000.0410.0702      dynamic  470      F      F      Po2233
* 107      0000.0410.0712      dynamic  470      F      F      Po2233
* 106      0000.0410.0602      dynamic  470      F      F      Po2233
* 106      0000.0410.0612      dynamic  470      F      F      Po2233
* 105      0000.0410.0502      dynamic  470      F      F      Po2233
* 105      0000.0410.0512      dynamic  470      F      F      Po2233
* 104      0000.0410.0402      dynamic  470      F      F      Po2233
* 104      0000.0410.0412      dynamic  470      F      F      Po2233
* 103      0000.0410.0302      dynamic  470      F      F      Po2233
* 103      0000.0410.0312      dynamic  470      F      F      Po2233
* 102      0000.0410.0202      dynamic  470      F      F      Po2233
* 102      0000.0410.0212      dynamic  470      F      F      Po2233
* 101      0000.0410.0102      dynamic  470      F      F      Po2233
* 101      0000.0410.0112      dynamic  470      F      F      Po2233
* 100      0000.0410.0002      dynamic  470      F      F      Po2233
* 100      0000.0410.0012      dynamic  470      F      F      Po2233

```

switch#

This example shows how to display only locally learned MAC addresses on the specified VLAN, which is mapped to a VN-Segment:

```
switch# show mac address-table local vlan 107
```

Legend:

```

* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since first seen,+ - primary entry using vPC Peer-Link
      VLAN      MAC Address      Type      age      Secure NTFY      Ports/SWID.SSID.LID
-----+-----+-----+-----+-----+-----+-----
* 107      0000.0410.0702      dynamic  480      F      F      Po2233
* 107      0000.0410.0712      dynamic  480      F      F      Po2233

```

switch#

This example shows how to display all remote MAC addresses learned on NVE:

```
switch# show mac address-table interface nve 1
```

Legend:

```

* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since first seen,+ - primary entry using vPC Peer-Link
      VN_SEGMENT      MAC Address      Type      Age      Remote Vtep/VxLAN Port
-----+-----+-----+-----+-----+-----
* 4100      0000.0110.0002      dynamic  1180      nve1(1.1.1.200)
* 4100      0000.0110.0012      dynamic  1180      nve1(1.1.1.200)
* 4101      0000.0110.0102      dynamic  1180      nve1(1.1.1.200)
* 4101      0000.0110.0112      dynamic  1180      nve1(1.1.1.200)
* 4102      0000.0110.0202      dynamic  1180      nve1(1.1.1.200)
* 4102      0000.0110.0212      dynamic  1180      nve1(1.1.1.200)
* 4103      0000.0110.0302      dynamic  1180      nve1(1.1.1.200)
* 4103      0000.0110.0312      dynamic  1180      nve1(1.1.1.200)
* 4104      0000.0110.0402      dynamic  1180      nve1(1.1.1.200)
* 4104      0000.0110.0412      dynamic  1180      nve1(1.1.1.200)
* 4105      0000.0110.0502      dynamic  1180      nve1(1.1.1.200)
* 4105      0000.0110.0512      dynamic  1180      nve1(1.1.1.200)
* 4106      0000.0110.0602      dynamic  1180      nve1(1.1.1.200)
* 4106      0000.0110.0612      dynamic  1180      nve1(1.1.1.200)
* 4107      0000.0110.0702      dynamic  1180      nve1(1.1.1.200)
* 4107      0000.0110.0712      dynamic  1180      nve1(1.1.1.200)
* 4108      0000.0110.0802      dynamic  1180      nve1(1.1.1.200)
* 4108      0000.0110.0812      dynamic  1180      nve1(1.1.1.200)
* 4109      0000.0110.0902      dynamic  1180      nve1(1.1.1.200)
* 4109      0000.0110.0912      dynamic  1180      nve1(1.1.1.200)

```

switch#

This example shows how to display all remote MAC addresses learned on the VNI:

```
switch# show mac address-table interface nve 1 vni 4100
```

Legend:

```

* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since first seen,+ - primary entry using vPC Peer-Link
      VN_SEGMENT      MAC Address      Type      Age      Remote Vtep/VxLAN Port
-----+-----+-----+-----+-----+-----
* 4100      0000.0110.0002      dynamic  1230      nve1(1.1.1.200)

```

```
* 4100          0000.0110.0012    dynamic    1230      nve1(1.1.1.200)
switch#
```

This example shows how to display all MAC addresses learned on NVE from the specified peer:

```
switch# show mac address-table interface nve 1 peer 1.1.1.200
```

Legend:

```

* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since first seen, + - primary entry using vPC Peer-Link
  VN_SEGMENT  MAC Address  Type      Age      Remote Vtep/VxLAN Port
-----+-----+-----+-----+-----
* 4100        0000.0110.0002    dynamic    1400      nve1(1.1.1.200)
* 4100        0000.0110.0012    dynamic    1400      nve1(1.1.1.200)
* 4101        0000.0110.0102    dynamic    1400      nve1(1.1.1.200)
* 4101        0000.0110.0112    dynamic    1400      nve1(1.1.1.200)
* 4102        0000.0110.0202    dynamic    1400      nve1(1.1.1.200)
* 4102        0000.0110.0212    dynamic    1400      nve1(1.1.1.200)
* 4103        0000.0110.0302    dynamic    1400      nve1(1.1.1.200)
* 4103        0000.0110.0312    dynamic    1400      nve1(1.1.1.200)
* 4104        0000.0110.0402    dynamic    1400      nve1(1.1.1.200)
* 4104        0000.0110.0412    dynamic    1400      nve1(1.1.1.200)
* 4105        0000.0110.0502    dynamic    1400      nve1(1.1.1.200)
* 4105        0000.0110.0512    dynamic    1400      nve1(1.1.1.200)
* 4106        0000.0110.0602    dynamic    1400      nve1(1.1.1.200)
* 4106        0000.0110.0612    dynamic    1400      nve1(1.1.1.200)
* 4107        0000.0110.0702    dynamic    1400      nve1(1.1.1.200)
* 4107        0000.0110.0712    dynamic    1400      nve1(1.1.1.200)
* 4108        0000.0110.0802    dynamic    1400      nve1(1.1.1.200)
* 4108        0000.0110.0812    dynamic    1400      nve1(1.1.1.200)
* 4109        0000.0110.0902    dynamic    1400      nve1(1.1.1.200)
* 4109        0000.0110.0912    dynamic    1400      nve1(1.1.1.200)
switch#
```

This example shows how to display all MAC addresses learned on NVE from the specified peer on the specified VNI:

```
switch# show mac address-table interface nve 1 peer 1.1.1.200 vni 4100
```

Legend:

```

* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since first seen, + - primary entry using vPC Peer-Link
  VN_SEGMENT  MAC Address  Type      Age      Remote Vtep/VxLAN Port
-----+-----+-----+-----+-----
* 4100        0000.0110.0002    dynamic    1420      nve1(1.1.1.200)
* 4100        0000.0110.0012    dynamic    1420      nve1(1.1.1.200)
switch#
```

This example shows how to display the number of locally learned MAC address table entries:

```
switch# show mac address-table count local
```

```
MAC Entries for all vlans:
Dynamic Address Count: 20
Static Address (User-defined) Count: 0
Multicast MAC Address Count: 0
Total MAC Addresses in Use: 20
```

```
Total PVLAN Clone MAC Address Count: 0
switch#
```

This example shows how to display the number of locally learned MAC address table entries on the specified VLAN, which is mapped to a VN-Segment:

```
switch# show mac address-table count local vlan 107
```

```
MAC Entries for all vlans:
Total MAC Addresses in Use: 2
switch#
```

This example shows how to display the number of remote MAC address table entries learned on NVE:

```
switch# show mac address-table count interface nve 1
```

```
MAC entries for all vlans:
Total MAC Address in use: 20
switch#
```

This example shows how to display the number of remote MAC address table entries learned on the VNI:

```
switch# show mac address-table count interface nve 1 vni 4100
```

```
MAC entries for all vlans:
```

```
Total MAC Address in use: 2
switch#
```

This example shows how to display the number of MAC address table entries learned on NVE from the specified peer:

```
switch# show mac address-table count interface nve 1 peer 1.1.1.200
MAC entries for all vlans:
Total MAC Address in use: 20
switch#
```

This example shows how to display the number of MAC address table entries learned on NVE from the specified peer on the specified VNI:

```
switch# show mac address-table count interface nve 1 peer 1.1.1.200 vni 4100
MAC entries for all vlans:
Total MAC Address in use: 2
switch#
```

## Clearing MAC Addresses

Use one of the following commands to clear the address entries from the MAC address table:

Command	Purpose
<b>clear mac address-table dynamic</b>	Clears all MAC address entries in the MAC address table.
<b>clear mac address-table dynamic vlan</b> <i>vlan-id</i>	Clears all VLAN and VXLAN MAC address entries from the MAC address table.
<b>clear mac address-table dynamic local</b>	Clears all locally learned MAC address entries on all VLANs mapped to VN -Segments.
<b>clear mac address-table dynamic local vlan</b> <i>vlan-id</i>	Clears all locally learned MAC address entries on the specified VLAN.
<b>clear mac address-table dynamic interface nve</b> <i>nve-id</i>	Clears all overlay learned MAC addresses.
<b>clear mac address-table dynamic interface nve</b> <i>nve-id vni vni-id</i>	Clears all network-learned MAC addresses on the specified VNI.
<b>clear mac address-table dynamic interface Ethernet</b> <i>slot/port vlan vlan-id</i>	Clears all MAC addresses on the specified interface and VLAN.
<b>clear mac address-table dynamic interface nve</b> <i>nve-id peer ip-address</i> <b>clear mac address-table dynamic interface nve</b> <i>nve-id peer ip-address vrf vrf-name</i>	Clears all MAC addresses on the NVE interface for the specified peer.
<b>clear mac address-table dynamic interface nve</b> <i>nve-id peer ip-address vni vni-id</i> <b>clear mac address-table dynamic interface nve</b> <i>nve-id peer ip-address vrf vrf-name vni vni-id</i>	Clears all MAC addresses on the NVE interface from the specified peer on the specified VNI.

This example shows how to clear all MAC address entries in the MAC address table:

```
switch# clear mac address-table dynamic
switch#
```

This example shows how to clear all VLAN and VXLAN MAC address entries from the MAC address table:

```
switch# clear mac address-table dynamic vlan 3100
switch#
```

This example shows how to clear all locally learned MAC address entries on all VLANs mapped to VN-Segments:

```
switch# clear mac address-table dynamic local
switch#
```

This example shows how to clear all locally learned MAC address entries on the specified VLAN:

```
switch# clear mac address-table dynamic local vlan 3100
switch#
```

This example shows how to clear all overlay learned MAC addresses:

```
switch# clear mac address-table dynamic interface nve 1
switch#
```

This example shows how to clear all network-learned MAC addresses on the specified VNI:

```
switch# clear mac address-table dynamic interface nve 1 vni 5000
switch#
```

This example shows how to clear all MAC addresses on the specified interface and VLAN:

```
switch# clear mac address-table dynamic interface Ethernet 1/1 vlan 3100
switch#
```

This example shows how to clear all MAC addresses on the NVE interface for the specified peer:

```
switch# clear mac address-table dynamic interface nve 1 peer 222.1.1.1 vrf default
switch#
```

This example shows how to clear all MAC addresses on the NVE interface from the specified peer on the specified VNI:

```
switch# clear mac address-table dynamic interface nve 1 peer 222.1.1.1 vrf default vni 5000
switch#
```